



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍA INFORMÁTICA (SEGOVIA)  
GRADO EN INGENIERÍA INFORMÁTICA DE SERVICIOS Y APLICACIONES

# DASELL: PLATAFORMA DE COMPRAVENTA MEDIANTE SERVICIOS AVANZADOS REALIZADO EN FLUTTER

AUTOR: DANIEL MARTÍN CUBERO

TUTOR: FERNANDO DÍAZ GÓMEZ



# AGRADECIMIENTOS

En primer lugar, agradecer a mi familia que siempre me ha dado su apoyo en los momentos más difíciles y me han animado en cada momento, a continuar, y en el apoyo en la realización de este TFG. En especial, a mis abuelos Rosa, Vicente, Atanasio y Tomasa que seguro estarían orgullosos de mí allá donde estén.

Gracias a mis amigos de toda la vida, que han estado siempre ahí para lo bueno y para lo malo, pero con los que sigo pasando grandes momentos.

Gracias a mis compañeros de clase que siempre han estado ahí para poder echarnos unas risas entre clase y clase, y a todas esas amistades que he hecho durante mi etapa universitaria.

Por último, mostrar mi agradecimiento a todos los profesores de la Escuela de Ingeniería de Informática de Segovia, por el tiempo y dedicación que han dedicado y por los conocimientos que nos han transmitido, Mencionar especialmente, a mi tutor de TFG, por todas las indicaciones dadas y ayuda, su guía durante todo el proceso y los valiosos consejos que me ha dado.

En definitiva, muchas gracias a todo el mundo que ha aportado algo en mí y en la realización de este proyecto.



# RESUMEN

El propósito principal de este Trabajo Final de Grado (TFG) se basa en el desarrollo de una plataforma híbrida de compra-venta entre usuarios mediante servicios avanzados dirigida a plataformas móviles, tanto Android como iOS, que permita al usuario la gestión de la publicación de anuncios y compra de productos o servicios en el momento que desee, en cualquier lugar a través de cualquier dispositivo móvil, y facilitado por su sistema de geolocalización de anuncios disponibles.

Por esta razón, el objetivo es que el usuario disponga de una aplicación intuitiva y sencilla de manera integral que facilite las transacciones de objetos entre usuarios de la misma.

En consecuencia, el objetivo de "DaSell" es mejorar la experiencia del usuario en el caso de realizar una transacción con otro usuario sin necesidad de disponer de un intermediario, facilitar una comunicación instantánea entre ambos, la posibilidad de mostrar la geolocalización para la entrega del producto y gestionar las publicaciones para mostrar la información más relevante del producto, y llegar al resto de usuarios con la mayor eficacia posible.

**Palabras claves:** Desarrollo móvil, Flutter, Dart, Firebase, aplicación móvil, producto, venta, compra, Cloud Firestore. transacción.

# ABSTRACT

The main purpose of this Final Degree Project is based on the development of a hybrid platform for buying and selling between users through advanced services aimed at mobile platforms both Android and iOS, which allows the user to manage the publication of advertisements and purchase of products or services at the time they want, anywhere through any mobile device, and facilitated by its geolocation system of available ads.

For this reason, the objective is to provide the user with an intuitive and simple application in a comprehensive manner that facilitates transactions of objects between users of the same.

Consequently, the objective of "DaSell" is to improve the user's experience in the case of a transaction with another user without the need for an intermediary, to facilitate instant communication between the two, the possibility of showing the geolocation for the delivery of the product and managing the publications to show the most relevant information about the product, and to reach the rest of the users as efficiently as possible.

**Keywords:** Mobile development, Flutter, Dart, Firebase, mobile application, product, sale, purchase, Cloud Firestore. transaction.



# ÍNDICE DE CONTENIDOS

1. DESCRIPCIÓN DEL PROYECTO .....	19
<b>1.1 INTRODUCCIÓN</b> .....	21
<b>1.2 MOTIVACIÓN</b> .....	23
<b>1.3 OBJETIVOS Y ALCANCE</b> .....	24
1.3.1 OBJETIVOS .....	24
1.3.2 ALCANCE .....	25
<b>1.4 ENTORNO TECNOLÓGICO</b> .....	28
1.4.1 HERRAMIENTAS HARDWARE .....	28
1.4.2 HERRAMIENTAS SOFTWARE .....	29
1.4.3 TECNOLOGÍAS DE DESARROLLO SOFTWARE .....	30
<b>1.5 ORGANIZACIÓN DEL DOCUMENTO</b> .....	34
<b>1.6 CONTENIDO DEL DIRECTORIO WEB</b> .....	35
2. CONTEXTO DEL PROYECTO .....	36
<b>2.1 ESTADO DEL ARTE</b> .....	38
2.1.1 WALLAPOP .....	38
2.1.2 VIBBO .....	41
2.1.3 EBAY .....	43
<b>2.2 METODOLOGÍA</b> .....	45
3. PLANIFICACIÓN, ESTIMACIÓN Y PRESUPUESTO .....	48
<b>3.1 PLANIFICACIÓN</b> .....	50
<b>3.2 PLANIFICACIÓN TEMPORAL</b> .....	50
<b>3.3 PUNTOS DE FUNCIÓN</b> .....	52
<b>3.4 COSTES</b> .....	62
3.4.1. RECURSOS HUMANOS .....	62
3.4.2. HARDWARE .....	63
3.4.3. SOFTWARE .....	64
<b>3.5. PRESUPUESTO EN BASE A LA PLANIFICACIÓN INICIAL</b> .....	65
<b>3.6 SEGUIMIENTO REAL</b> .....	65
<b>3.7 COSTE REAL</b> .....	67
3.7.1 RECURSOS HUMANOS .....	67
3.7.2 HARDWARE .....	68

3.7.3 SOFTWARE .....	69
3.7.4 COSTE REAL TOTAL .....	69
<b>3.8 CONCLUSIONES</b> .....	69
4. ANÁLISIS .....	71
<b>4.1 ACTORES</b> .....	73
<b>4.2 REGLAS DE NEGOCIO</b> .....	74
<b>4.3 OBJETIVOS DEL SISTEMA</b> .....	75
<b>4.4 CARACTERÍSTICAS</b> .....	76
<b>4.5 REQUISITOS DE USUARIO</b> .....	78
<b>4.6 REQUISITOS FUNCIONALES</b> .....	82
4.6.1 GESTIÓN DE USUARIOS .....	82
4.6.2 GESTIÓN DE PUBLICACIÓN DE ARTÍCULOS .....	84
4.6.3. GESTIÓN DE PUBLICACIONES .....	84
4.6.4. GESTIÓN DE SISTEMA DE VALORACIÓN DE USUARIOS .....	86
4.6.5 GESTIÓN DE SISTEMA DE MENSAJERÍA INSTANTÁNEA .....	87
4.6.6 GESTIÓN DE SISTEMA DE SEGUIMIENTO EN TIEMPO REAL .....	88
<b>4.7 REQUISITOS NO FUNCIONALES</b> .....	89
4.7.1 DISPONIBILIDAD .....	89
4.7.2. ACCESIBILIDAD .....	89
4.7.3 ESCALABILIDAD .....	90
4.7.4. USABILIDAD .....	90
4.7.5. SEGURIDAD .....	90
4.7.6 RESTRICCIONES .....	91
4.7.7. ATRIBUTOS DE CALIDAD .....	91
4.7.8 MANTENIBILIDAD .....	91
4.7.9. FIABILIDAD .....	91
4.7.10 RENDIMIENTO .....	91
4.7.11 ROBUSTEZ .....	91
4.7.12 INTEGRIDAD .....	92
<b>4.8 DIAGRAMA CASOS DE USO</b> .....	92
4.8.1. GESTIÓN DE USUARIOS .....	94
4.8.2. GESTIÓN DE PROCESO DE PUBLICACIÓN .....	95
4.8.3. GESTIÓN DE PUBLICACIONES .....	95
4.8.4. GESTIÓN DE PUBLICACIONES Y GESTIÓN DE SISTEMA DE VALORACIÓN ENTRE USUARIOS .....	96
4.8.5. GESTIÓN DE PUBLICACIONES Y GESTIÓN DE SISTEMA DE MENSAJERÍA INSTANTÁNEA .....	96
4.8.6. GESTIÓN DE SERVICIO DE MENSAJERÍA INSTANTÁNEA .....	97
4.8.7. GESTIÓN DE PUBLICACIONES .....	98

<b>4.9 ESPECIFICACIÓN CASOS DE USO</b> .....	99
4.9.1 CASO DE USO – USUARIO NO REGISTRADO .....	99
4.9.2. CASO DE USO – USUARIO REGISTRADO Y NO IDENTIFICADO .....	101
4.9.3. CASO DE USO – USUARIO REGISTRADO E IDENTIFICADO .....	103
4.9.4. CASO DE USO – USUARIO VENDEDOR – GESTIÓN DE PUBLICACIÓN DE ARTÍCULOS .....	108
4.9.5 CASO DE USO – USUARIO VENDEDOR – GESTIÓN DE PUBLICACIONES.....	122
4.9.6 CASO DE USO - USUARIO CLIENTE – GESTIÓN DE PUBLICACIONES.....	133
4.9.7. CASO DE USO - USUARIO CLIENTE – GESTIÓN DE SERVICIO DE MENSAJERÍA INSTANTÁNEA	143
4.9.8. CASO DE USO - USUARIO CLIENTE – GESTIÓN SISTEMA DE SEGUIMIENTO EN TIEMPO REAL	155
4.9.9 CASO DE USO - USUARIO CLIENTE - GESTIÓN DE PUBLICACIONES .....	171
<b>4.10 REQUISITOS DE INFORMACIÓN</b> .....	183
<b>4.11 MODELO LÓGICO DE DATOS (DIAGRAMA ENTIDAD – RELACIÓN)</b> .....	185
<b>5. DISEÑO</b> .....	189
<b>5.1 ARQUITECTURA LÓGICA</b> .....	191
<b>5.2 ARQUITECTURA FÍSICA</b> .....	194
<b>5.3 MÓDELO LÓGICO DE DATOS</b> .....	195
5.3.1 DISEÑO BASE DE DATOS .....	199
<b>5.4 DICCIONARIO DE DATOS</b> .....	199
5.4.1. USUARIO REGISTRADO E IDENTIFICADO .....	199
5.4.2. USUARIO VENDEDOR.....	200
5.4.3. CATEGORÍA .....	200
5.4.4. PUBLICACIÓN .....	201
5.4.5. VALORACIÓN .....	202
5.4.6. TRANSACCIÓN.....	202
5.4.7. MENSAJE.....	203
5.4.8. CHAT .....	203
5.4.9. UBICACIÓN.....	204
5.4.10. RUTA.....	204
<b>5.5 DIAGRAMAS DE SECUENCIA</b> .....	205
5.5.1. RU-02: REGISTRARSE EN EL SISTEMA .....	206
5.5.2. RU-62: AÑADIR VALORACIÓN .....	207
5.5.3. RU-43: MODIFICAR PUBLICACIÓN .....	208
5.5.4 RU-85: ACCEDER A SISTEMA DE SEGUIMIENTO EN TIEMPO REAL .....	209
<b>5.6 DISEÑO DE LA INTERFAZ DE USUARIO</b> .....	210
<b>6. IMPLEMENTACIÓN</b> .....	211
<b>6.1 REQUISITOS MÍNIMOS</b> .....	212

<b>6.2</b>	<b>HERRAMIENTAS UTILIZADAS</b> .....	212
6.2.1	HERRAMIENTAS DE DESARROLLO SOFTWARE MÓVIL.....	212
6.2.2	HERRAMIENTAS DE SOPORTE.....	215
<b>6.3.</b>	<b>TECNOLOGÍAS IMPLEMENTADAS</b> .....	216
<b>6.4.</b>	<b>LIBRERIAS Y DEPENDENCIAS</b> .....	217
<b>6.5.</b>	<b>DETALLES DE IMPLEMENTACIÓN</b> .....	218
6.5.1.	DESCRIPCIÓN DEL SERVICIO DE BASE DE DATOS.....	219
6.5.2.	IMPLEMENTACIÓN DEL SERVICIO DE BASE DE DATOS .....	223
6.5.3.	ADMINISTRACIÓN DE USUARIOS.....	230
6.5.4.	IMPLEMENTACIÓN TECNOLOGÍAS <i>FRONT-END</i> .....	234
<b>6.6</b>	<b>ESTRUCTURA INTERNA DEL PROYECTO</b> .....	260
7.	PRUEBAS.....	270
<b>7.1</b>	<b>PRUEBAS CAJA BLANCA</b> .....	272
<b>7.2</b>	<b>PRUEBAS CAJA NEGRA</b> .....	273
8.	MANUALES .....	279
8.2.1	INICIO DE SESIÓN.....	281
<b>8.1</b>	<b>MANUAL DE INSTALACIÓN</b> .....	281
<b>8.2</b>	<b>MANUAL DE USUARIO</b> .....	282
8.2.2	INICIO DE SESIÓN MEDIANTE CUENTA DE GOOGLE .....	282
8.2.3	REGISTRO DE UN NUEVO USUARIO .....	283
8.2.4	CONSULTAR INFORMACIÓN DE PERFIL DE USUARIO .....	284
8.2.5	GESTIÓN DE MIS PUBLICACIONES .....	285
8.2.6	PÁGINA DE INICIO.....	288
8.2.7	GESTIÓN DE PUBLICACIÓN .....	291
8.2.8	MODIFICAR UNA PUBLICACIÓN.....	294
8.2.9	GESTIÓN DE SISTEMA DE VALORACIONES ENTRE USUARIOS .....	295
8.2.10	GESTIÓN DE PROCESO DE PUBLICACIÓN.....	298
8.2.11	GESTIÓN DE SERVICIO DE MENSAJERÍA INSTANTÁNEA .....	302
8.2.12	GESTIÓN DE SISTEMA DE SEGUIMIENTO EN TIEMPO REAL.....	306
9.	CONCLUSIONES .....	311
<b>9.1</b>	<b>CONCLUSIONES</b> .....	312
<b>9.2</b>	<b>FUTURAS MEJORAS</b> .....	313
	REFERENCIAS .....	315
<b>10.1</b>	<b>BIBLIOGRAFÍA</b> .....	316
<b>10.2</b>	<b>WEBGRAFÍA</b> .....	316
<b>10.3</b>	<b>CONTENIDO AUDIOVISUAL</b> .....	317

# ÍNDICE DE TABLAS

Tabla 1: Objetivos del sistema.....	25
Tabla 2: Herramientas Hardware utilizadas en el desarrollo del TFG .....	28
Tabla 3: Herramientas Software para el desarrollo de este TFG .....	30
Tabla 4: Tecnologías de Desarrollo Software utilizadas en el desarrollo de este TFG .....	32
Tabla 5: Puntos de función (Entradas de usuario).....	53
Tabla 6: Puntos de función (Salidas).....	55
Tabla 7: Puntos de función (Consultas de usuario) .....	56
Tabla 8: Puntos de función (Ficheros Lógicos Internos) .....	56
Tabla 9: Puntos de función (Ficheros externos) .....	56
Tabla 10: Sumatorio ponderación Puntos de Función .....	57
Tabla 11: Factor de Ajuste .....	58
Tabla 12: Planificación inicial.....	60
Tabla 13: Diagrama de Gantt - Planificación inicial .....	61
Tabla 14: Planificación Inicial - Recursos Humanos.....	63
Tabla 15: Planificación Inicial - Hardware.....	64
Tabla 16: Planificación Inicial - Software .....	65
Tabla 17: Presupuesto total - Planificación Inicial .....	65
Tabla 18: Planificación Seguimiento real.....	66
Tabla 19: Diagrama de Gantt - Planificación Seguimiento real .....	67
Tabla 20: Coste real - Recursos humanos.....	67
Tabla 21: Coste real - Hardware .....	69
Tabla 22: Coste real - Software.....	69
Tabla 23: Coste real .....	69
Tabla 24: Descripción actores del sistema .....	74
Tabla 25: Reglas de Negocio .....	75
Tabla 26: Objetivos del sistema - Gestión de los usuarios .....	75
Tabla 27: Objetivos del sistema - Gestión publicación de artículos .....	75
Tabla 28: Objetivos del sistema - Gestión de sistema de valoración entre usuarios .....	76
Tabla 29: Objetivos del sistema - Gestión de sistema de mensajería instantánea .....	76
Tabla 30: Objetivos del sistema - Gestión de sistema de seguimiento en tiempo real .....	76
Tabla 31: Requisitos de usuario - Usuario no registrado.....	78
Tabla 32: Requisitos de usuario – Usuario registrado y no identificado .....	78
Tabla 33: Requisitos de usuario – Usuario registrado e identificado .....	78
Tabla 34: Requisitos de usuario - Usuario vendedor .....	80
Tabla 35: Requisitos de usuario - Usuario cliente .....	82
Tabla 36: RU-01: Consultar información general de la aplicación.....	99
Tabla 37: RU-02 Registrarse en el sistema .....	100
Tabla 38: RU-03 Registrarse con Usuario de Google .....	101
Tabla 39: RU-04: Iniciar sesión en la aplicación.....	102
Tabla 40: RU-05: Iniciar sesión con cuenta de Google .....	102
Tabla 41: RU-06 Visualizar información de perfil de usuario .....	103
Tabla 42: RU-07 Modificar avatar del perfil de usuario.....	104
Tabla 43: RU-08: Acceder a configuración de perfil de usuario .....	105

Tabla 44: RU-09 Seleccionar tema oscuro .....	105
Tabla 45: RU-10 Activar/desactivar notificaciones.....	106
Tabla 46: RU-11 Visualizar transacciones asociadas al perfil de usuario.....	106
Tabla 47: RU-12 Eliminar cuenta de usuario .....	107
Tabla 48: RU-13 Cerrar sesión .....	107
Tabla 49: RU-14 Realizar publicación de un producto o servicio .....	108
Tabla 50: RU-15 Visualizar listado de categorías.....	109
Tabla 51: RU-16 Seleccionar categoría asociada .....	109
Tabla 52: RU-17 Modificar categoría seleccionada .....	110
Tabla 53: RU-18 Visualizar listado de subcategorías .....	110
Tabla 54: RU-19 Seleccionar subcategoría asociada .....	111
Tabla 55: RU-20 Modificar subcategoría asociada .....	112
Tabla 56: RU-21 Añadir título a una publicación .....	112
Tabla 57: RU-22 Añadir descripción a una publicación .....	113
Tabla 58: RU-23 Activar/desactivar posibilidad de realizar envío .....	113
Tabla 59: RU-24 Añadir estado de publicación.....	114
Tabla 60: RU-25 Añadir imágenes desde la galería del dispositivo .....	115
Tabla 61: RU-26 Añadir imágenes desde la cámara del dispositivo .....	115
Tabla 62: RU-27 Añadir precio a una publicación.....	116
Tabla 63: RU-28 Activar/Desactivar publicación como donación.....	117
Tabla 64: RU-29 Seleccionar ubicación actual .....	117
Tabla 65: RU-30 Seleccionar una ubicación distinta a la actual .....	118
Tabla 66: RU-31 Visualizar mapa con la ubicación seleccionada.....	119
Tabla 67: RU-32 Establecer marcador con nueva ubicación .....	119
Tabla 68: RU-33 Confirmar nueva ubicación establecida.....	120
Tabla 69: RU-34 Confirmar proceso de publicación .....	121
Tabla 70: RU-35 Cancelar proceso de publicación .....	121
Tabla 71: RU-36 Visualizar listado completo de publicaciones .....	122
Tabla 72: RU-37 Visualizar información de una publicación .....	123
Tabla 73: RU-38 Eliminar una publicación .....	123
Tabla 74: RU-39 Confirmar eliminar una publicación.....	124
Tabla 75: RU-40 Marcar una publicación como vendida.....	125
Tabla 76: RU-41 Confirmar eliminar una publicación.....	125
Tabla 77: RU-42 Acceder al mapa con la ubicación.....	126
Tabla 78: RU-43 Visualizar formulario para modificar publicación .....	126
Tabla 79: RU-44 Modificar título de la publicación .....	127
Tabla 80: RU-45 Modificar descripción de publicación .....	128
Tabla 81: RU-46 Modificar categoría de una publicación.....	129
Tabla 82: RU-47 Modificar subcategoría de publicación.....	129
Tabla 83: RU-48 Modificar estado de publicación.....	130
Tabla 84: RU-49 Modificar la posibilidad de envío de la publicación .....	131
Tabla 85: RU-50 Modificar precio de publicación.....	132
Tabla 86: RU-51 Visualizar sistema de valoración personal .....	132
Tabla 87: RU-52 Visualizar publicaciones en venta .....	133
Tabla 88: RU-53 Visualizar listado completo de publicaciones .....	134
Tabla 89: RU-54 Visualizar información de una publicación .....	134
Tabla 90: RU-55 Acceder al mapa con la ubicación.....	135
Tabla 91: RU-56 Seleccionar botón para redirigir a ubicación actual.....	136
Tabla 92: RU-57 Visualizar imágenes de una publicación .....	136
Tabla 93: RU-58 Acceder al sistema de valoración entre usuarios.....	137

Tabla 94: RU-59 Visualizar información de un vendedor .....	137
Tabla 95: RU-60 Visualizar listado de valoraciones del usuario vendedor .....	138
Tabla 96: RU-61 Visualizar valoración a un usuario vendedor .....	138
Tabla 97: RU-62 Añadir valoración a un usuario vendedor .....	139
Tabla 98: RU-63 Modificar comentario de una valoración.....	140
Tabla 99: RU-64 Modificar puntuación de valoración .....	141
Tabla 100: RU-65 Cancelar proceso de añadir una valoración .....	142
Tabla 101: RU-66 Eliminar valoración realizada .....	142
Tabla 102: RU-67 Visualizar listado de publicaciones en venta.....	143
Tabla 103: RU-68 Acceder a servicio de mensajería instantánea.....	144
Tabla 104: RU-69 Acceder al historial de mensajería instantánea .....	145
Tabla 105: RU-70 Visualizar historial de chats abiertos .....	145
Tabla 106: RU-71 Seleccionar conversación de chat .....	146
Tabla 107: RU-72 Visualizar listado de mensajes del chat.....	146
Tabla 108: RU-73 Escribir mensaje .....	147
Tabla 109: RU-74 Ampliar selector de emoticonos .....	148
Tabla 110: RU-75 Seleccionar un emoticono.....	148
Tabla 111: RU-76 Compartir archivo .....	149
Tabla 112: RU-77 Acceder a la cámara del dispositivo .....	149
Tabla 113 RU-78 Modificar configuración de la cámara del dispositivo .....	150
Tabla 114: RU-79 Realizar fotografía de la cámara del dispositivo .....	151
Tabla 115: RU-80 Confirmar imagen realizada mediante la cámara del dispositivo .....	151
Tabla 116: RU-81 Acceder a la galería del dispositivo .....	152
Tabla 117: RU-82 Visualizar listado de álbumes de la galería del dispositivo .....	153
Tabla 118 RU-83 Seleccionar imagen de la galería del dispositivo.....	153
Tabla 119: RU-84 Seleccionar imagen enviada para ampliarla .....	154
Figura 120: RU-85 Acceder a sistema de seguimiento en tiempo real.....	156
Tabla 121: RU-86 Visualizar mapa con ubicación actual .....	156
Tabla 122: RU-87 Ampliar tamaño del mapa .....	157
Tabla 123: RU-88 Reducir tamaño del mapa.....	158
Tabla 124: RU-89 Mostrar ubicación actual en el mapa.....	158
Tabla 125: RU-90 Rotar mapa sobre ubicación actual.....	159
Tabla 126: RU-91 Realizar un desplazamiento sobre el mapa .....	160
Tabla 127: RU-92 Seleccionar barra de búsqueda incluida en el mapa.....	160
Tabla 128: RU-93 Escribir ubicación de destino en la barra de búsqueda .....	161
Tabla 129: RU-94 Realizar búsqueda de la ubicación de destino en la barra de búsqueda. ....	162
Tabla 130: RU-95 Visualizar listado de sugerencias en la barra de búsqueda.....	163
Tabla 131: RU-96 Seleccionar ubicación de destino deseada .....	163
Tabla 132: RU-97 Colocar marcador con la ubicación de destino manualmente.....	164
Tabla 133: RU-98 Confirmar ubicación de destino seleccionada mediante el marcador.....	165
Tabla 134: RU-99 Mostrar/no mostrar recorrido trazado en la ruta por el usuario .....	166
Tabla 135: RU-100 Seleccionar botón que redirija a la ubicación actual .....	166
Tabla 136: RU-101 Compartir ubicación actual.....	167
Tabla 137: RU-102 Iniciar seguimiento en tiempo real .....	168
Tabla 138: RU-103 Acceder a mapa con la ubicación actual .....	169
Tabla 139: RU-104 Visualizar el mapa con el seguimiento en tiempo real .....	170
Tabla 140: RU-105 Dejar de compartir ubicación en tiempo real .....	170
Tabla 141: RU-106 Visualizar listado completo de publicaciones .....	172
Tabla 142: RU-107 Activar/Desactivar publicación como favorita.....	172
Tabla 143: RU-108 Ordenar listado de publicaciones por proximidad.....	173

Tabla 144: RU-109 Ordenar listado de publicaciones por fecha de publicación .....	174
Tabla 145: RU-110 Realizar filtrado por intervalo de precio .....	175
Tabla 146: RU-111 Actualizar listado completo de publicaciones.....	175
Tabla 147: RU-112 Seleccionar barra de búsqueda.....	176
Tabla 148: RU-113 Escribir publicación que se desea buscar.....	176
Tabla 149: RU-114 Visualizar listado coincidente con la búsqueda realizada .....	177
Tabla 150: RU-115 Visualizar listado horizontal de categorías disponibles .....	178
Tabla 151: RU-116 Seleccionar categoría deseada.....	178
Tabla 152: RU-117 Visualizar listado coincidente con la categoría seleccionada.....	179
Tabla 153: RU-118 Visualizar listado de publicaciones propias.....	179
Tabla 154: RU-119 Seleccionar publicación del listado de publicaciones realizadas por el usuario .....	180
Tabla 155: RU-120 Visualizar el listado de publicaciones marcadas como favoritas por el usuario .....	181
Tabla 156: RU-121 Seleccionar publicación del listado de publicaciones marcadas como favoritas .....	181
Tabla 157: RU-122 Visualizar el listado de publicaciones marcadas como vendidas .....	182
Tabla 158: RU-123 Seleccionar publicación del listado de publicaciones marcadas como vendidas.....	182
Tabla 159: Diccionario de datos – Usuario registrado e identificado.....	200
Tabla 160: Diccionario de datos – Usuario vendedor.....	200
Tabla 161: Diccionario de datos - Categoría.....	200
Tabla 162: Diccionario de datos - Publicación .....	202
Tabla 163: Diccionario de datos - Valoración .....	202
Tabla 164: Diccionario de datos - Transacción .....	203
Tabla 165: Diccionario de datos - Mensaje .....	203
Tabla 166: Diccionario de datos - Chat .....	204
Tabla 167: Diccionario de datos - Ubicación .....	204
Tabla 168: Diccionario de datos - Ruta.....	205
Tabla 169: Requisitos mínimos Hardware y Software.....	212
Tabla 170: PCN-01 Registro de usuarios.....	274
Tabla 171: PCN-02 Acceder al sistema .....	274
Tabla 172: PCN-03 Listado de publicaciones .....	274
Tabla 173: PCN-04 Información de la publicación.....	275
Tabla 174: PCN-05 Información de la publicación.....	275
Tabla 175: PCN-06 Buscador de publicaciones.....	275
Tabla 176: PCN-07 Modificar avatar de perfil de usuario .....	276
Tabla 177: PCN-08 Realizar publicación .....	276
Tabla 178: PCN-09 Añadir una nueva valoración .....	277
Tabla 179: PCN-10 Eliminar una publicación.....	277
Tabla 180: PCN-11 Acceso a sistema de seguimiento en tiempo real.....	277
Tabla 181: PCN-12 Cerrar sesión .....	277

# ÍNDICE DE FIGURAS

Figura 1: Comercio Electrónico.....	22
Figura 2: Introducción - Perfil Comprador Online .....	22
Figura 3: Introducción: Dispositivos móviles .....	23
Figura 4: Árbol de características .....	27
Figura 5: Entronco tecnológico del desarrollo de este TFG .....	33
Figura 6: Estado del arte - Wallapop .....	38
Figura 7: Estado del arte - Wallapop Inicio.....	39
Figura 8: Estado del arte - Vibbo .....	41
Figura 9: Estado del arte Vibbo Inicio.....	42
Figura 10: Estado del arte - Ebay .....	43
Figura 11: Estado del arte - eBay Inicio .....	44
Figura 12: Modelo incremental iterativo.....	46
Figura 13: Calendario planificación inicial 2022 .....	50
Figura 14: Calendario planificación Inicial 2023 .....	51
Figura 15: Jerarquía Actores del sistema.....	74
Figura 16: Información Diagramas Casos de uso.....	93
Figura 17: Casos de Uso - Gestión de Usuarios .....	94
Figura 18: Casos de Uso - Gestión de proceso de publicación - Usuario vendedor .....	95
Figura 19: Casos de Uso - Gestión de publicaciones – Usuario vendedor.....	95
Figura 20: Casos de Uso - Gestión de publicaciones + Gestión de sistema de valoraciones - Usuario cliente .....	96
Figura 21: Casos de Uso - Gestión de publicaciones + Gestión de sistema de valoraciones - Usuario cliente .....	96
Figura 22: Casos de Uso - Gestión de servicios de mensajería instantánea – Usuario cliente .....	97
Figura 23: Casos de Uso - Gestión de publicaciones - Usuario cliente .....	98
Figura 24 - Caso de Uso - Usuario no registrado .....	99
Figura 25: Casos de uso – Usuario registrado y no identificado.....	101
Figura 26: Caso de uso – Usuario registrado e identificado .....	103
Figura 27: Caso de Uso Usuario Vendedor - Gestión de publicación de artículos.....	108
Figura 28: Caso de Uso Usuario Vendedor - Gestión de publicaciones.....	122
Figura 29: Caso de uso - Usuario cliente – Gestión de publicaciones .....	133
Figura 30: Caso de Uso - Usuario cliente - Gestión de servicio de mensajería instantánea .....	143
Figura 31: Caso de Uso - Usuario cliente - Gestión de sistema se seguimiento en tiempo real.....	155
Figura 32: Caso de Uso - Usuario cliente - Gestión de publicaciones.....	171
Figura 33: Modelo Entidad-Relación .....	187
Figura 34: Estructura Clean Architecture .....	191
Figura 35: Arquitectura lógica – Aplicación Flutter .....	192
Figura 36: Patrón Modelo-Vista-Controlador .....	194
Figura 37: Arquitectura física .....	195
Figura 38: Estructura documentos, base de datos Cloud Firestore.....	197
Figura 39: Estructura de un documento en Cloud Firestore .....	197
Figura 40: Subcolecciones de Cloud Firestore .....	198
Figura 41: Modelo lógico de datos - DaSell .....	199
Figura 42: Diagrama de secuencia - RU-02 Registrarse en el sistema .....	206
Figura 43: Diagrama se secuencia - Añadir valoración .....	207
Figura 44: Diagrama de secuencia- RU:43 Modificar publicación .....	208
Figura 45: Diagrama de secuencia - Acceder a sistema de seguimiento en tiempo real .....	209

Figura 46: Logo Plataforma Firebase .....	219
Figura 47: Módulos de Firebase .....	220
Figura 48: Logo Cloud Firestore.....	222
Figura 49: Proyectos asociados a la cuenta de desarrollo.....	224
Figura 50: Creación proyecto Firebase	Figura 51: Implementación Google Analytics.....
Figura 52: Organización herramientas Firebase .....	225
Figura 53: Integrar Cloud Firestore en aplicación móvil.....	225
Figura 54: Configuración entorno.....	226
Figura 55: Instalación y ejecución de CLI de FlutterFire .....	226
Figura 56: Inicialización Firebase y agregar complementos .....	227
Figura 57: Crear instancia de FirebaseFirestore .....	227
Figura 58: Método Insertar mensaje de chat en Cloud Firestore.....	228
Figura 59: Método de obtención de colección de datos de publicaciones .....	228
Figura 60: Método de recuperación de datos de la colección "products" .....	229
Figura 61: Método para modificar estado de publicación vendida o no .....	229
Figura 62: Método modificación de estado de conexión .....	229
Figura 63: Método de borrado de una publicación .....	230
Figura 64: Crear instancia de Firebase Authentication.....	230
Figura 65: Método registro de usuarios en el sistema mediante credenciales.....	231
Figura 66: Método para autenticación de usuarios mediante credenciales .....	232
Figura 67: Método para autenticación de usuarios mediante los servicios de Google.....	233
Figura 68: Método de cierre de sesión en la plataforma .....	234
Figura 69: Implementación - Logo Flutter .....	235
Figura 70: Arquitectura lógica a alto nivel aplicación Flutter .....	236
Figura 71: Arquitectura Flutter en un terminal .....	237
Figura 72: Lógica de ejecución de Flutter en iOS.....	237
Figura 73: Lógica de ejecución de Flutter en Android .....	238
Figura 74: Ejemplo código StatelessWidget .....	241
Figura 75: Ejemplo básico StatefulWidget en Flutter .....	242
Figura 76: Ejemplo métodos GestureDetector .....	244
Figura 77: Ejemplo método InkWell .....	244
Figura 78: Ejemplo métodos DragGestureRecognizer.....	244
Figura 79: Árbol de Widgets genérico .....	247
Figura 80: Ejemplo código Árbol de Widgets .....	248
Figura 81: Árbol de Widgets aplicación DaSell .....	249
Figura 82: Implementación - Logo Dart.....	250
Figura 83: Gestor de estados: Patrón Provider .....	251
Figura 84: Implementación Provider – paquete provider .....	252
Figura 85: Implementación Provider - paquete Provider .....	252
Figura 86: Implementación Provider - ChangeNotifier.....	253
Figura 87: Implementación Provider - ChangeNotifierProvider .....	253
Figura 88: Implementación Provider - Gestión estado mediante Provider .....	254
Figura 89: Patrón BLoC Flutter .....	254
Figura 90: Patrón BLOC - Flujo de información .....	255
Figura 91: Implementación BLOC - Dependencias BLOC.....	256
Figura 92: Implementación BLOC – Definición estados.....	257
Figura 93: Implementación Patrón BLOC - Definición de eventos .....	257
Figura 94: Implementación de BLOC - Creación de BLOC.....	257
Figura 95: Implementación BLOC - Implementación en widget.....	258
Figura 96: Estructura interna del proyecto.....	260

Figura 97: Estructura interna - directorio lib .....	263
Figura 98: Directorios Maps - Sistema de seguimiento en tiempo real .....	264
Figura 99: Estructura interna - Screens .....	264
Figura 100: Pruebas de caja blanca .....	272
Figura 101: Pruebas caja negra .....	273
Figura 102: Manual de usuario - Inicio de sesión .....	281
Figura 103: Manual de usuario - Inicio de sesión con Google .....	283
Figura 104: Manual de usuario - Registro de un nuevo usuario.....	284
Figura 105: Manual de usuario - Consultar información de perfil de usuario .....	285
Figura 106: Manual de usuario - Gestión de mis publicaciones - Publicadas .....	286
Figura 107: Manual de usuario - Gestión de mis publicaciones - Favoritas .....	287
Figura 108 Manual de usuario - Gestión de mis publicaciones - Vendidas .....	287
Figura 109: Página de Inicio – Listado completo de publicaciones .....	288
Figura 110: Página de Inicio - Filtrado por intervalo de precio.....	289
Figura 111: Página de Inicio - Ordenar por más cercano o más recientes .....	289
Figura 112: Página de Inicio - Gestor de búsqueda .....	290
Figura 113: Página de Inicio - Filtrado por categoría .....	291
Figura 114: Gestión de una publicación - Información general.....	292
Figura 115: Gestión de publicación - Acceso sistema de valoración usuario vendedor .....	293
Figura 116: Gestión de publicación - Mapa con ubicación actual .....	293
Figura 117: Gestión de Publicación - Acceder a servicio de mensajería instantánea.....	294
Figura 118: Gestión de publicación - Modificar una publicación .....	294
Figura 119: Gestión de sistema de valoración entre usuarios - Listado publicaciones en venta .....	295
Figura 120: Gestión de sistema de valoración entre usuarios - Listado de valoraciones recibidas .....	296
Figura 121: Menú de opciones - Modificar valoración y eliminar valoración .....	297
Figura 122: Añadir una valoración.....	297
Figura 123: Gestión de proceso de publicación - Seleccionar categoría .....	298
Figura 124: Gestión de proceso de publicación - Seleccionar subcategoría .....	299
Figura 125: Gestión del proceso de publicación - Detalles del producto .....	300
Figura 126: Gestión de proceso de publicación - Añadir imágenes del producto .....	301
Figura 127: Gestión de proceso de publicación - Introducir precio y localización .....	302
Figura 128: Gestión de servicio de mensajería instantánea - Listado de chats.....	303
Figura 129: Gestión de servicio de mensajería instantánea – Visualizar chat.....	303
Figura 130: Gestión de servicio de mensajería instantánea - Escribir mensaje .....	304
Figura 131: Gestión de servicio de mensajería instantánea - Añadir emoticonos .....	304
Figura 132: Gestión de servicio de mensajería instantánea – Compartir archivos .....	304
Figura 133: Gestión de servicio de mensajería instantánea - mapa con ubicación actual .....	306
Figura 134: Gestión de sistema de seguimiento en tiempo real - Seleccionar ubicación .....	307
Figura 135: Gestión de sistema de seguimiento en tiempo real - Confirmar ubicación seleccionada ....	308
Figura 136: Gestión de sistema de seguimiento en tiempo real - Menú de opciones .....	309



---

# 1. DESCRIPCIÓN DEL PROYECTO

---



La clave fundamental, de la que vamos a partir, para desarrollar un determinado producto es que debemos ser conscientes de que es lo que queremos llevar a cabo y el camino a seguir para conseguir el citado producto. Por lo tanto, será imprescindible establecer los diferentes elementos que lo forman o definen, que reciben la denominación de requisitos.

## 1.1 INTRODUCCIÓN

Desde la aparición de la red de Internet, donde se sentaron las bases de cómo conectar dos dispositivos de manera remota, hasta el día de hoy, donde prácticamente cualquier dispositivo ya sean dispositivos móviles, ordenadores, automóviles, impresoras, sistemas de video-vigilancia e incluso los electrodomésticos del hogar, conectados mediante la nueva tecnología 5G, se puede concluir que el uso y la forma de la utilización de la conectividad a Internet, ha evolucionado de forma exponencial y con una velocidad que ha establecido la utilización de Internet de forma cotidiana, como algo totalmente normalizado.

Esta evolución de las redes trasforma la manera de realizar nuestras tareas cotidianas. Tan solo con un dispositivo con conectividad podemos disponer de entretenimiento, comunicación y conectividad en cualquier lugar en el que nos encontremos, lo que permite tener a nuestro alcance muchas opciones, entre ellas la disponibilidad del comercio electrónico, donde la compra y venta de productos por Internet supone uno de las mayores fuentes de ingresos, tanto para empresas como vendedores particulares.

En consecuencia, esta transformación tecnológica obliga a que cualquier empresa, independientemente de su tamaño, se vea obligada a adaptarse a la manera de interactuar entre vendedor y cliente. Este desarrollo ha permitido que pequeñas compañías tengan acceso a un mercado más amplio, proveyéndoles de una mayor capacidad para competir, y consecuentemente provocando un mayor impacto económico.

En los últimos años, este desarrollo ha generado también la posibilidad de realizar transacciones de productos, tanto de primera como de segunda mano entre particulares, sin necesidad de ninguna empresa ni negocio que intervenga en el proceso.

El comercio electrónico ha supuesto una revolución mundial en lo que a las transacciones comerciales respecta. En España, aproximadamente el 33% de las compras totales se realizaron a través de Internet durante el 2022. De hecho, en el último trimestre de ese mismo año, la facturación procedente del comercio *online* alcanzó un máximo histórico al superar los 18,9 millones de euros, un 28,8% más que el año anterior, según los datos que ofrece la CNMC<sup>2</sup>.

---

<sup>2</sup> Comisión Nacional de los Mercados y la Competencia. <https://www.cnmc.es/>



Figura 1: Comercio Electrónico

Cabe reseñar, que no todas las ramas de actividad se benefician del mismo modo de la posibilidad de ofertar sus artículos y servicios en la red. La Administración Pública y las tiendas de alimentación son los negocios que menos facturaron respecto al comercio electrónico en España en el último año, dado que no todos los comercios se pueden adaptar a ese estilo de estrategia comercial.

### Perfil del comprador *online* español

Según los estudios más recientes, el comprador *online* habitual tiende a ser un hombre de entre 26 y 44 años, ocupado y probablemente residente en una población de más de 100.000 habitantes. De hecho, más del 40% de los usuarios de Internet que compran a través de la red dentro del territorio nacional viven en este tipo de entorno. Ahora bien, los jóvenes y adolescentes no son para nada ajenos a este tipo de actividad *online*. En 2021, más de tres millones de españoles de entre 16 y 24 años adquirieron algún tipo de producto o servicio en Internet; una cifra que supera los siete millones, si se tienen en cuenta a los individuos de 25 a 34 años.



Figura 2: Introducción - Perfil Comprador Online

En cuanto al gasto medio de los internautas españoles, la gran mayoría dedican entre 25 y 100 euros a sus compras *online*. Debido a estas cantidades, no sorprende que los principales métodos de pago en este tipo de operaciones comerciales sean las tarjetas de débito y la plataforma de pagos *online*, PayPal. Este último es especialmente útil en el caso de realizar compras en ciertos establecimientos con sede física fuera del país desde el que se realiza la compra. Ahora bien, este no es, ni siquiera, uno de los aspectos más valorados por los consumidores *online* de estos medios de pago, e incluso de otros menos usados. La posibilidad de comprar desde casa y a cualquier hora, así como evitar aglomeraciones son las principales razones para decantarse por esta vía de compra.

Se espera que el número de compradores *online* crezca en un 25% hasta alcanzar los 37 millones de usuarios en España en 2025.

A la vista de estos datos tan prometedores, se trata de una excelente oportunidad para las tiendas *online* de cualquier tamaño y sector, por lo que es importante estar bien preparado para tener éxito en los próximos años.

### **Los dispositivos móviles ganan terreno entre los compradores *online***

Según avanza la tecnología, los ordenadores de sobremesa pierden peso entre los usuarios compradores *online* españoles. Y es que, pese a que todavía siguen siendo el dispositivo electrónico más utilizado para comprar en Internet en España, cada vez son más los que se decantan por los *smartphones*. Es más, el aumento del uso de dispositivos móviles en las compras *online* ha sido constante en los últimos años y la pandemia de COVID-19 no parece haber frenado dicho crecimiento. En concreto, el porcentaje de internautas que adquirieron productos o servicios a través de ellos alcanzó el 70% en 2022.



*Figura 3: Introducción: Dispositivos móviles*

## **1.2 MOTIVACIÓN**

La idea de desarrollar esta aplicación móvil surge a raíz de la transformación que ha supuesto el comercio electrónico en los últimos años en España y en el resto del mundo, incrementada por el motivo de la pandemia del COVID-19. Por este motivo, el año 2020 marcó el inicio de la digitalización de miles de negocios que no habían llevado a cabo todavía este proceso y con ello la aparición de nuevos compradores *online*. Un fenómeno que se mantuvo en 2021 y 2022, que persistirá en 2023 y permanecerá ya como un nuevo modelo de negocio en nuestras vidas y con una continua expansión del comercio electrónico de manera exponencial.

Esta causa ha supuesto la necesidad de desarrollar una aplicación móvil híbrida<sup>3</sup>, dado el continuo crecimiento de este tipo de comercio a nivel nacional e internacional, y orientada a la aparición de nuevas formas de estrategia de comercio electrónico como el C2C (*Consumer-to-Consumer*), abreviatura utilizada en comercio electrónico para definir una estrategia de cliente a cliente. Se utiliza este término para definir un modelo de negocio en la red que pretende relacionar comercialmente el usuario final con otro usuario final. Una estrategia C2C es aquella que define un negocio cuyo objetivo es facilitar la comercialización de productos y/o servicios entre usuarios, como por ejemplo **DASELL**, sirviendo la aplicación como una simple intermediaria.

Por estos motivos, se ha propuesto la idea de crear una aplicación móvil que permite centralizar todos los servicios que proporciona la aplicación tanto de publicación de productos, gestión de publicaciones, transacciones de artículos entre usuarios, sistema de mensajería instantánea, sistema de seguimiento en tiempo real integrado en este servicio y un sistema de valoración entre usuarios. El objetivo es qué facilite al usuario un espacio donde poder realizar ventas o compras tanto en productos nuevos como en productos de segunda mano, que permitan dar una segunda utilidad a aquellos objetos que ya se usan o no deseamos tener, sin necesidad de disponer de una tienda o intermediario que nos facilite la transacción. De esta manera el usuario disfruta de todas las ofertas y promociones que desee en el momento al ser desarrollada para dispositivos móviles. A su vez, se ofrecen servicios como venta o alquiler de inmuebles o terrenos, clases particulares, personal trainers, soporte escolar, búsqueda de empleo, etc. La necesidad hace

## 1.3 OBJETIVOS Y ALCANCE

### 1.3.1 OBJETIVOS

La aplicación **DASELL** tendrá como objetivo principal la gestión de las publicaciones de productos y mostrar al usuario toda la información necesaria del producto, así como una constante comunicación instantánea con el usuario vendedor.

Lo que se persigue con este proyecto es el desarrollo una aplicación híbrida que permita y dinamice a los usuarios la venta o compra de objetos nuevos o usados, beneficiándose ambos, vendedores y compradores, ya sea porque se vende un producto que no le da uso o por querer comprar un objeto a un usuario particular en un estado aceptable y a buen precio.

Tras definir el objetivo principal que debe cumplir la aplicación, esta también cuenta con una serie de **objetivos de negocio** asociados que son complementarios al objetivo principal y que permiten dotar al producto de otras funcionalidades muy interesantes para el usuario.

Los **objetivos de negocio** son los beneficios que obtiene el cliente al hacer uso del producto final. En este proyecto se persiguen los siguientes objetivos:

OBJETIVO	DESCRIPCIÓN
OBJ-1	Realizar una implementación de un sistema eficiente de transacciones entre usuarios finales

<sup>3</sup> **Aplicación móvil híbrida:** Las aplicaciones híbridas, a diferencia de las nativas, son aquellas capaces de funcionar en distintos sistemas operativos móviles. Entre ellos: Android, iOS y Windows Phone. De esta manera, una misma app puede utilizarse en cualquier smartphone o tablet, indistintamente de su marca o fabricante. ( Fuente: <https://blog.mdcloud.es/aplicaciones-hibridas-frameworks-ejemplos-y-ventajas/> )

<b>OBJ-2</b>	Buscar mejorar la experiencia de los usuarios que desean realizar la compra o venta de un producto
<b>OBJ-3</b>	Conseguir que la mayor parte de los usuarios utilicen nuestro servicio
<b>OBJ-4</b>	Implementar un sistema de publicación fácil e intuitivo para todo tipo de usuarios, permitiendo al usuario de publicar una amplia diversidad de tipos de productos
<b>OBJ-5</b>	Fomentar la reutilización de objetos, permitiendo dar una segunda vida a objetos en desuso
<b>OBJ-6</b>	Fomentar la publicación de servicios, permitiendo ofrecer venta/alquiler de vivienda, y servicios profesionales sobre distintos sectores
<b>OBJ-7</b>	Reducción de los costes que se producen al realizar una venta de un artículo mediante una tienda física o mediante un intermediario.
<b>OBJ-8</b>	Facilitar una comunicación instantánea e eficaz entre los usuarios implicados en una transacción.
<b>OBJ-9</b>	Optimización de los tiempos, dado la manera rápida e intuitiva de encontrar el producto que necesitas

Tabla 1: Objetivos del sistema

Además de estos objetivos, se debe cumplir otras características para facilitar la interacción entre el usuario-dispositivo.

- ❖ **Diseño de una interfaz sencilla.** Se busca implementar una interfaz sencilla y familiar que sea intuitiva en su uso y manejo, respecto a las aplicaciones más demandas actualmente por los usuarios de dispositivos móviles, permitiendo así, que usuarios no tan habituales al uso de este tipo de herramientas, sean capaces de utilizarlo con fluidez.
- ❖ La aplicación debe contener su propio **manual de usuario** que permita al usuario conocer el funcionamiento de la aplicación de forma detallada.
- ❖ Se implementa un **diseño responsive**, que permita adaptarse a la pantalla del dispositivo, independientemente de su tamaño.
- ❖ La implementación de las distintas *APIs*<sup>4</sup> no supondrá **ningún riesgo en la seguridad** de los datos del usuario, ya que nunca estarán expuestos a terceros.
- ❖ La aplicación mostrará tanto mensajes de confirmación o de error a la hora de selección de ciertas funcionalidades, como notificaciones *push* en el dispositivo.

### 1.3.2 ALCANCE

Esta aplicación tiene como finalidad englobar en un mismo entorno a todos los usuarios que desean realizar tanto una compra de un producto que desee, o una venta de un producto sobre el que

---

<sup>4</sup> **API:** API es el acrónimo de "Application Programming Interface" (Interfaz de Programación de Aplicaciones, en español). Se trata de un conjunto de reglas y protocolos que permiten la comunicación entre distintos componentes de software. Una API define los métodos y formatos de interacción que pueden utilizar dos aplicaciones para intercambiar información y realizar acciones entre sí de forma estandarizada. En resumen, una API proporciona una interfaz para que los desarrolladores puedan acceder y utilizar funcionalidades específicas de una aplicación o servicio sin necesidad de conocer todos los detalles internos de su implementación. Esto facilita la integración de diferentes sistemas y la creación de aplicaciones que interactúen entre sí de manera eficiente.

desea obtener un rendimiento económico. De esta manera, permitimos que los usuarios puedan interactuar fácilmente, tanto para publicar sus productos con el fin de que llegue a una mayor cantidad de posibles compradores, y a su vez, para que el comprador encuentre de una manera muy rápida y sencilla el producto deseado, sin necesidad de tener un intermediario en el proceso.

Por esto, el entorno es una plataforma híbrida, que permita utilizar la aplicación tanto en dispositivos Android como iOS, que tan solo con disponer de conectividad mediante red WIFI o datos móviles, pueda gestionar sus productos desde cualquier lugar, sin necesidad de acudir a una tienda intermediaria o algún espacio físico que actué como intermediario para realizar cualquier transacción.

La aplicación permitirá que todos los usuarios interactúen de manera rápida e intuitiva con la plataforma, la cual se encargará de que esta cumpla con todos los servicios y funcionalidades desarrolladas. El objetivo o alcance que se desea abarcar con esta plataforma es mucho mayor que el punto de desarrollo actual. En un futuro la aplicación podrá soportar funcionalidades que permitan al usuario tener una experiencia mucho más satisfactoria con el entorno. Actualmente, la aplicación contará con seis módulos diferenciados, como se detalla en el siguiente árbol de características:



## 1.4 ENTORNO TECNOLÓGICO

**DASELL** es una aplicación móvil, para dispositivos tanto Android como iOS, desarrollada mediante Flutter, *SDK*<sup>5</sup> de Google, bajo el lenguaje *Dart* para la capa de *Front-end*, y como *Back-end* se implementa una base de datos no relacional mediante *Firebase* mediante el módulo de *Cloud Firestore*, que nos permite disponer la capa de seguridad, persistencia y robustez a la aplicación bajo el entorno de *Google*.

A continuación, se muestra un listado con una breve descripción de las distintas herramientas y tecnologías que se han utilizado durante el proceso de desarrollo hasta alcanzar el producto final deseado.

### 1.4.1 HERRAMIENTAS HARDWARE

En primer lugar, se detalla el equipamiento *hardware* utilizado que ha permitido el desarrollo del proyecto.

DISPOSITIVO	DESCRIPCIÓN
<p>HP Pavilion Power Laptop 15-cb0xx</p> 	<p>Características del dispositivo:</p> <ul style="list-style-type: none"> <li>• Intel(R) Core™ i7-7700HQ CPU @ 2,8 GHz, 2808 MHz</li> <li>• Gráfica NVIDIA® GeForce® GTX 1040</li> <li>• Memoria RAM 16GB</li> </ul>
<p>Xiaomi Mi9T</p> 	<p>Características del dispositivo:</p> <ul style="list-style-type: none"> <li>• Procesador Snapdragon 730</li> <li>• Memoria: RAM 6 GB</li> <li>• Almacenamiento: 128GB</li> <li>• Pantalla AMOLED 6,39" FullHD+ 2340 x 1080 píxeles, 19,5:9</li> </ul>
<p>Monitor HP x2301</p> 	<p>Características del dispositivo:</p> <ul style="list-style-type: none"> <li>• Pantalla: LED 24 pulgadas</li> <li>• Resolución: Full HD (1920 x 1080)</li> <li>• Tiempo De Respuesta: 3 ms</li> </ul>

Tabla 2: Herramientas Hardware utilizadas en el desarrollo del TFG

<sup>5</sup> **SDK (Software Development Kit):** es un conjunto de herramientas de desarrollo de software que proporciona un conjunto de bibliotecas, herramientas, documentación y ejemplos de código para facilitar la creación de aplicaciones y programas para una plataforma específica. Un SDK suele incluir un conjunto de APIs que permiten a los desarrolladores interactuar con las características y funciones de un sistema operativo, un lenguaje de programación, una plataforma de software o un hardware en particular. Los SDKs son utilizados por los desarrolladores para acelerar el proceso de desarrollo y simplificar la creación de aplicaciones compatibles con una plataforma determinada.

## 1.4.2 HERRAMIENTAS SOFTWARE

A continuación, se describe brevemente el *software* utilizado durante el proceso de desarrollo de dicha aplicación para dispositivos móviles:

HERRAMIENTAS	DESCRIPCIÓN
<b>Windows 10 Home</b> 	Windows 10 Home es el sistema operativo implementado en el dispositivo portátil utilizado para el desarrollo, como se ha descrito anteriormente
<b>Visual Studio Code</b> 	Editor de código fuente desarrollado por Microsoft, destaca por ser un software libre y multiplataforma, y por su gran variedad de extensiones que ayudan al desarrollo del código, su buena integración con Git, y el soporte para depurar código. Permite ejecutar código fuente en cualquier lenguaje.
<b>Postman</b> 	Plataforma de desarrollo de API que permite a los desarrolladores probar, depurar y documentar API de manera más eficiente. Proporciona una interfaz gráfica de usuario para enviar solicitudes HTTP a un servidor y recibir respuestas. Postman también permite la creación de colecciones de solicitudes para automatizar el proceso de pruebas y la generación de documentación para API.
<b>Microsoft Project</b> 	Microsoft Project es un software de gestión de proyectos desarrollado por Microsoft que permite a los usuarios planificar, programar y controlar el progreso de proyectos de cualquier tamaño y complejidad. La aplicación proporciona herramientas para la planificación de tareas, la asignación de recursos, el seguimiento de los plazos, el análisis de costos y la gestión de riesgos, lo que permite a los equipos de proyectos colaborar de manera efectiva y alcanzar sus objetivos en tiempo y forma.
<b>GitHub</b>	GitHub es una plataforma en línea que permite a los desarrolladores de software almacenar y compartir su código fuente de manera colaborativa, lo que facilita la colaboración y el control de versiones en proyectos de software. Además, GitHub ofrece herramientas para la gestión de proyectos, seguimiento de problemas y la

	<p>integración continua, lo que hace que sea una plataforma muy popular para el desarrollo de software en equipo.</p>
<p style="text-align: center;"><b>Draw.io</b></p> 	<p>Draw.io es una herramienta de diagramación y creación de gráficos en línea gratuita y de código abierto. Con Draw.io, los usuarios pueden crear diagramas, organigramas, diagramas de flujo y otros tipos de gráficos para visualizar información de manera clara y concisa. La herramienta ofrece una interfaz de usuario fácil de usar y permite la colaboración en tiempo real, lo que la hace popular para proyectos de equipo y colaborativos.</p>
<p style="text-align: center;"><b>FontAwesome</b></p> 	<p>FontAwesome es una biblioteca de iconos vectoriales que se pueden utilizar en aplicaciones web y de escritorio. Los iconos de FontAwesome son escalables y personalizables, lo que permite que los desarrolladores los adapten a diferentes tamaños y estilos en sus proyectos. Esta biblioteca es muy popular entre los desarrolladores web y se utiliza en una gran cantidad de sitios y aplicaciones.</p>

Tabla 3: Herramientas Software para el desarrollo de este TFG

### 1.4.3 TECNOLOGÍAS DE DESARROLLO SOFTWARE

Antes de comenzar el desarrollo del proyecto, se ha realizado un estudio previo de que tecnologías utilizar, el cual ha sido una decisión clave, ya que sobre las tecnologías consideradas tan solo tenía formación básica sobre Flutter y el lenguaje Dart. El motivo de la decisión ha sido el aprendizaje de un nuevo lenguaje para desarrollo software en dispositivos móviles tan demandado en la actualidad. Se describen dichas tecnologías, a continuación:

HERRAMIENTAS	DESCRIPCIÓN
Flutter	Flutter es un marco de desarrollo de software móvil y web de código abierto creado por Google. Permite a los desarrolladores crear aplicaciones móviles nativas y web utilizando un solo código base en el lenguaje de programación Dart. Flutter destaca por su alto rendimiento, su amplia biblioteca de <i>widgets</i> <sup>6</sup> personalizados y su

<sup>6</sup> **Widget:** término ampliamente utilizado para describir cualquier componente visual o de interfaz de usuario en una aplicación Flutter, el cual utiliza un modelo de programación basado en widgets para construir interfaces de usuario, lo que significa que todo en Flutter es un widget. Para más detalle, véase la sección 6.5.4 Implementación Tecnologías Front-end.

	<p>capacidad de construir interfaces de usuario atractivas y animaciones complejas.</p>
<p>Dart</p> 	<p>Dart es un lenguaje de programación de código abierto, desarrollado por Google, que se utiliza principalmente para crear aplicaciones móviles, de escritorio y web. Dart es un lenguaje orientado a objetos y ofrece características de tipado fuerte, así como un sistema de recolección de basura automatizado. Dart también incluye una <i>máquina virtual (VM)</i> <sup>7</sup> y un compilador que permite generar código ejecutable tanto para la web como para dispositivos móviles.</p>
<p>Patrón BLoC</p> 	<p>El patrón <i>BLoC (Business Logic Component)</i> es un patrón de arquitectura de software utilizado en Flutter para separar la lógica de negocio de la interfaz de usuario. En resumen, el patrón BLoC utiliza un componente intermedio (el BLoC) para comunicar la interfaz de usuario con la lógica de negocio, lo que permite una mejor separación de responsabilidades y facilita la reutilización del código. El BLoC se encarga de recibir los eventos de la interfaz de usuario, procesarlos y enviar los resultados de vuelta a la interfaz de usuario para su visualización.</p>
<p>Patrón Provider</p> 	<p>El patrón Provider es un patrón de diseño que se utiliza para manejar el estado global en una aplicación. Básicamente, Provider permite compartir datos entre diferentes <i>widgets</i> de forma eficiente y sin tener que pasar los datos de un <i>widget</i> a otro a través de parámetros. En lugar de eso, Provider mantiene una instancia de los datos compartidos en la memoria y permite a los <i>widgets</i> acceder a estos datos en cualquier parte de la aplicación. Esto mejora el rendimiento y la organización del código, lo que facilita el mantenimiento y la escalabilidad de la aplicación. Herramienta muy útil para manejar el estado global en una aplicación Flutter.</p>

<sup>7</sup> **Máquina virtual (VM):** software o programa que simula un entorno de computadora completo y virtualizado, permitiendo ejecutar y utilizar sistemas operativos y aplicaciones en un entorno aislado y separado del sistema operativo y hardware físico subyacente. Proporciona una capa de abstracción entre el software y el hardware, lo que permite que diferentes sistemas operativos y aplicaciones se ejecuten en la misma máquina física. Esto brinda flexibilidad, portabilidad y eficiencia al permitir la consolidación de varios sistemas en una sola máquina física y facilitar la creación de entornos de prueba, desarrollo y ejecución de aplicaciones aislados.

<p>API Google Maps</p> 	<p>La API de Google Maps permite a los desarrolladores integrar mapas interactivos de Google en sus propias aplicaciones. Proporciona un conjunto de herramientas y funciones que permiten a los desarrolladores agregar mapas personalizados, visualizar información geográfica, proporcionar direcciones y rutas, y crear experiencias de usuario interactivas basadas en la ubicación. También, permite a los desarrolladores incorporar funciones de mapeo y geolocalización en sus aplicaciones.</p>
<p>API MapBox</p> 	<p>Mapbox es una plataforma de mapas interactivos que ofrece una API para desarrolladores. La API de Mapbox permite a los desarrolladores integrar mapas personalizados, datos geoespaciales y capacidades de navegación en sus aplicaciones y sitios web. La API de Mapbox es una herramienta que los desarrolladores pueden utilizar para agregar funcionalidad de mapas a sus proyectos de programación.</p>
<p>Firebase</p> 	<p>Firebase es una plataforma de desarrollo de aplicaciones móviles y web, propiedad de Google, que ofrece una variedad de herramientas y servicios en la nube para ayudar a los desarrolladores a crear aplicaciones de alta calidad de manera rápida y eficiente. Firebase ofrece servicios como autenticación de usuarios, almacenamiento de datos en tiempo real, análisis de datos, notificaciones push, entre otros.</p>
<p>Cloud Firestore</p> 	<p>Cloud Firestore es un servicio de base de datos en la nube proporcionado por Google. Es una base de datos NoSQL (no relacional) que permite almacenar, sincronizar y consultar datos de forma escalable en tiempo real. Cloud Firestore se utiliza comúnmente para aplicaciones web y móviles, y se integra fácilmente con otros servicios de Google Cloud Platform, como Firebase y Google Cloud Functions.</p>

Tabla 4: Tecnologías de Desarrollo Software utilizadas en el desarrollo de este TFG

Para tener una mayor visión de las tecnologías descritas anteriormente, se detalla en la siguiente figura el entorno tecnológico implementado, dividiéndolo en módulos para aportar una mayor claridad.

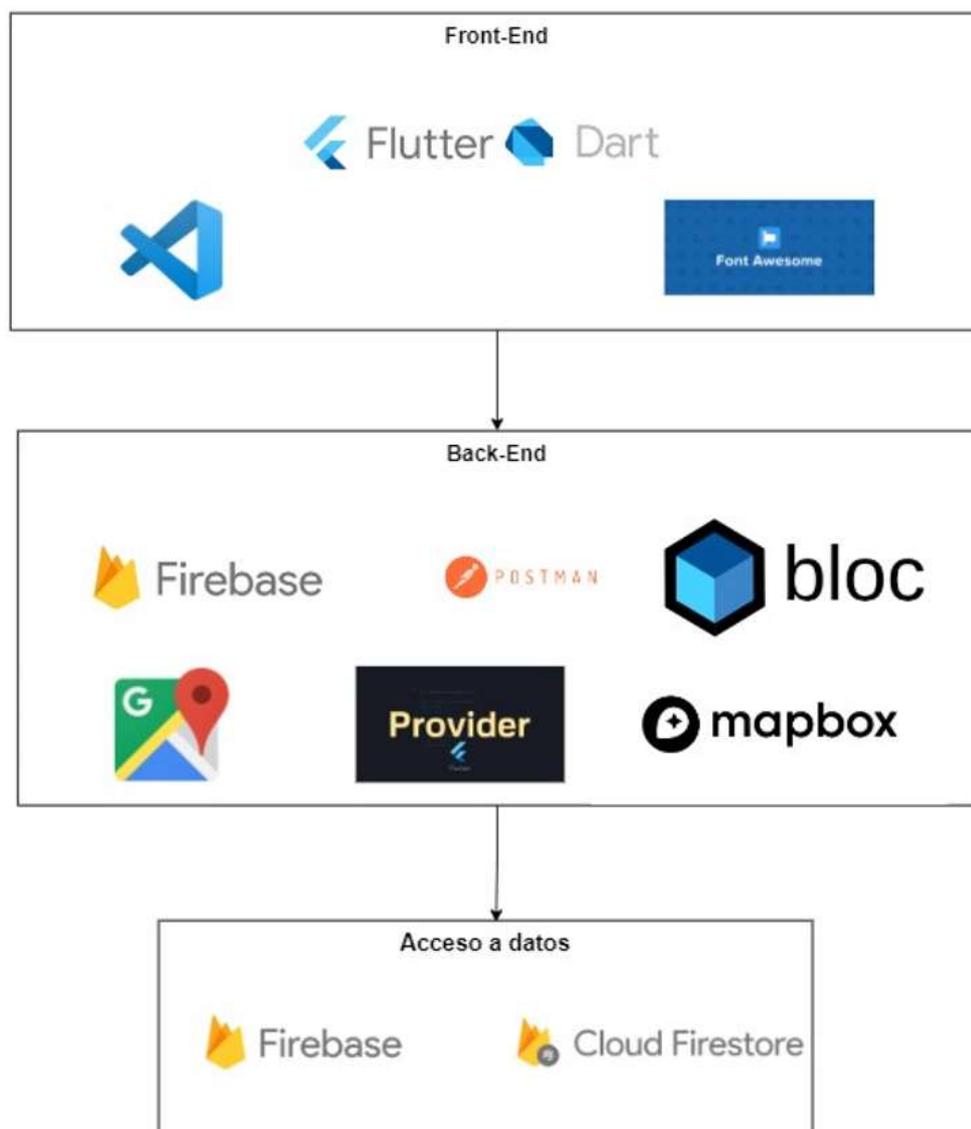


Figura 5: Entrono tecnológico del desarrollo de este TFG

## 1.5 ORGANIZACIÓN DEL DOCUMENTO

En este apartado se define cómo está organizada la estructura principal de este documento, el cual está formado por distintos capítulos. En total, se distribuye en nueve capítulos y dos anexos. Esta memoria está estructurada en los siguientes apartados, que detallan cada uno de ellos distintos aspectos del trabajo realizado:

- **Capítulo 1 – Introducción.** En esta sección se presentan los aspectos generales del proyecto, como son, una pequeña introducción al contexto actual del desarrollo, los objetivos que se buscan cumplir, el alcance, el entorno tecnológico, la motivación y las características principales de la aplicación.
- **Capítulo 2 – Descripción del proyecto.** Esta sección se destina a la obtención de información de cualquier sistema o herramienta similar que se encuentre en el mercado para, posteriormente, realizar una comparativa y un análisis crítico sobre las mismas y el entorno tecnológico que rodea al proyecto. En este caso se realiza sobre varias aplicaciones similares que ofrecen el mismo servicio de compraventa de productos.
- **Capítulo 3 - Planificación y presupuestos.** En este capítulo se muestra el análisis de la metodología de trabajo utilizada en el desarrollo, sumado a la planificación temporal y estimación de costes requeridos por el proyecto. Se detalla una estimación del coste en tiempo y recursos del proyecto.
- **Capítulo 4 - Análisis de Sistema.** Este apartado tiene como objetivo destinado a describir, detalladamente, la información los requisitos propuestos por el usuario correspondientes a cada funcionalidad de la aplicación, junto con la descripción detallada de los actores participantes de la misma, así como, las acciones que debe realizar el usuario con el sistema mediante los casos de uso.
- **Capítulo 5 - Diseño del sistema.** Capítulo en el que se detallan las distintas arquitecturas del proyecto, tanto arquitectura física cómo lógica, necesarias para el correcto funcionamiento de la aplicación y complementadas con sus respectivos diagramas explicativos. Además, se especificará el diseño de la base de datos, las relaciones entre entidades para el almacenamiento de la información en la base de datos y el diseño de la interfaz que posee la aplicación.
- **Capítulo 6 – Implementación.** En este apartado se tratará de analizar los aspectos técnicos del proyecto, como son las tecnologías utilizadas, herramientas, librerías y software implementado en su desarrollo.
- **Capítulo 7 – Pruebas.** Capítulo que contiene las distintas pruebas realizadas para asegurar que la aplicación cumple todos los requisitos y funciona correctamente.
- **Capítulo 8 – Manuales.** En esta sección se documentan las guías y manuales de usuario necesarios para la instalación y el aprendizaje y posterior manejo de todos los aspectos esenciales de la aplicación desarrollada.

- **Capítulo 9 - Conclusiones y posibles mejoras.** En este apartado se define la valoración general final del desarrollo completo de la aplicación y una serie de conclusiones, una vez terminado el proyecto, junto con propuestas destinadas a futuras mejoras que se puedan incorporar en versiones posteriores.
- **Capítulo 10 - Bibliografía y referencias:** Capítulo destinado a recoger todas las referencias físicas y web de documentación externas utilizadas en el desarrollo de la aplicación

## 1.6 CONTENIDO DEL DIRECTORIO WEB

Junto a este documento, se incluye el contenido del directorio web correspondiente al proyecto, el cual se encuentra estructurado de esta manera:

### 1. Carpeta de documentación

- Memoria en PDF del proyecto Fin De Grado: DaSell.

### 2. Carpeta Software

- Se incluye copia de software de la aplicación desarrollada

### 3. Instaladores

- Archivo apk: Ejecutable para la instalación de la aplicación DaSell.

---

## 2. CONTEXTO DEL PROYECTO

---



## 2.1 ESTADO DEL ARTE

Antes de comenzar con el desarrollo de la plataforma, esta sección recoge una búsqueda y análisis sobre las propuestas de herramientas, plataformas o aplicaciones similares que podemos encontrar actualmente en el mercado. Este análisis permite conocer cuáles son las implementaciones y características en este entorno, y cuál es el dominio que puede abarcar nuestro desarrollo, para una vez obtenida esta información, fijar las ideas sobre cuáles son las características relevantes y esenciales que, al menos, debemos desarrollar. Con estas características básicas, se planifica cuáles pueden ser las nuevas implementaciones que incluya nuestra plataforma con el fin de encontrar un hueco en el mercado, ofreciendo una mejora de lo que puede encontrar el usuario en otras aplicaciones e incluyendo nuevas funcionalidades.

En primer lugar, se ha realizado la búsqueda en el ámbito local por el entorno de la provincia de Segovia, con el objetivo de encontrar una plataforma que permita la compraventa de productos entre usuarios. Tan solo se han encontrado puntos de comercio físicos cuya labor, en el caso de productos utilizados, es actuar de intermediario entre vendedor y comprador a cambio de una comisión por la venta realizada o este intermediario realiza la compra del producto y la labor de venta corre ya a su cargo. En el caso de productos nuevos, en este mismo entorno se han encontrado tanto tiendas físicas como tiendas que se han adaptado al comercio electrónico y distribuyen sus productos mediante sus propias plataformas u otras externas.

Por estos motivos, se ha recurrido a realizar el análisis y búsqueda referente al ámbito nacional, en el que nos encontramos una situación similar. Podemos destacar que, en este ámbito, despuntan con mucha fuerza, las plataformas más extendidas en nuestro país, e incluso aplicaciones que permiten una compraventa de forma local entre los usuarios, pero fuera de la provincia de Segovia. Se detallan a continuación las siguientes plataformas con servicios y funcionalidades similares.

### 2.1.1 WALLAPOP



*Figura 6: Estado del arte - Wallapop*

**Wallapop** es una aplicación móvil y sitio web que permite a los usuarios comprar y vender productos de segunda mano de manera fácil y rápida. Fue fundada en 2013 en Barcelona, España, y ha experimentado un gran crecimiento en popularidad desde entonces. Wallapop se ha convertido en una de las aplicaciones más populares en España y otros países como México, Argentina, Colombia y Francia.

Wallapop es una *startup* española que surge gracias a la necesidad de comprar y vender productos a través de una app en nuestro *smartphone*. Se basa en la geolocalización, tanto del comprador como del vendedor, y te muestra los productos que buscas dependiendo de su cercanía a tu posición.

La plataforma se enfoca en la economía circular y promueve la reutilización de productos en lugar de desecharlos. Permite a los usuarios publicar anuncios de manera gratuita y vender sus productos de forma local a personas cercanas a su ubicación. La aplicación utiliza la geolocalización para mostrar los

productos disponibles cerca de la ubicación del usuario, lo que hace que sea fácil y conveniente encontrar productos interesantes en la misma ciudad o área.

La aplicación permite a los usuarios navegar por diferentes categorías de productos, como electrónica, moda, hogar, deportes y mucho más. También cuenta con un sistema de mensajería integrado que permite a los compradores y vendedores comunicarse y acordar los detalles de la compra-venta. Además, la aplicación ofrece funciones para garantizar la seguridad de las transacciones, como la verificación de perfiles y el uso de sistemas de pago seguros.

En resumen, Wallapop es una aplicación de compra-venta de productos de segunda mano que facilita el proceso de encontrar y comprar productos cercanos a tu ubicación. La aplicación promueve la economía circular y el uso responsable de los recursos, y ofrece funciones para garantizar la seguridad de las transacciones entre compradores y vendedores.

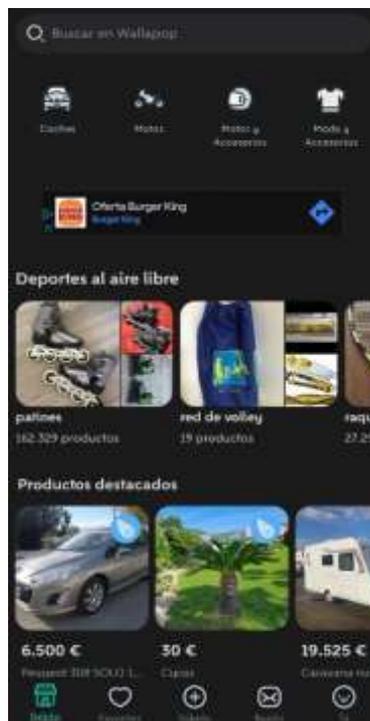


Figura 7: Estado del arte - Wallapop Inicio

Tras realizar el recorrido por esta aplicación móvil, se puede observar que la aplicación dispone de las siguientes características técnicas a destacar:

- **Tecnología avanzada.** Wallapop utiliza tecnología avanzada para permitir la búsqueda y filtrado de productos de manera eficiente. La aplicación utiliza técnicas de aprendizaje automático para mejorar la calidad de los resultados de búsqueda y proporcionar recomendaciones personalizadas.

- **Geolocalización.** La aplicación utiliza la geolocalización para mostrar los productos disponibles cerca de la ubicación del usuario, lo que hace que sea fácil y conveniente encontrar productos interesantes en la misma ciudad o área.
- **Sistema de mensajería integrado.** La aplicación cuenta con un sistema de mensajería integrado que permite a los compradores y vendedores comunicarse y acordar los detalles de la venta de manera segura y sin salir de la aplicación.
- **Verificación de perfiles.** Wallapop verifica los perfiles de los usuarios para garantizar que los compradores y vendedores sean personas reales y confiables. Esto ayuda a reducir el riesgo de fraude y aumenta la confianza en las transacciones.
- **Sistema de pago seguro.** La aplicación cuenta con un sistema de pago seguro que permite a los compradores y vendedores realizar transacciones de manera segura y protegida. Los pagos se realizan a través de la plataforma de pagos de Wallapop, lo que ayuda a evitar fraudes y estafas.
- **Comunidad de usuarios activa.** Wallapop cuenta con una gran comunidad de usuarios activos, lo que significa que hay una gran cantidad de productos disponibles para la venta en la plataforma. Esto aumenta las posibilidades de encontrar lo que se busca y de realizar transacciones exitosas.

En resumen, la aplicación Wallapop cuenta con tecnología avanzada, un sistema de mensajería integrado, verificación de perfiles y sistema de pago seguro, lo que hace que sea fácil y seguro para los usuarios comprar y vender productos de segunda mano. Además, la geolocalización y la gran comunidad de usuarios activos hacen que sea una plataforma muy conveniente y efectiva para la compra y venta de productos.

Por otra parte, se encuentran ciertos aspectos a tener en cuenta:

- La aplicación no dispone de un sistema de seguimiento en tiempo real, que permita conocer la localización del usuario que desea vender un producto en caso que la entrega de este se realice de manera personal.
- El sistema de valoración entre usuarios puede ser poco fiable en el caso de vendedores que no estén verificados por la aplicación.
- Problemas de seguridad, aunque ofrece funciones de seguridad para proteger a los compradores y vendedores, algunos usuarios han reportado fraudes o estafas en la plataforma, lo que puede ser preocupante para los usuarios que buscan realizar transacciones seguras.
- Limitaciones en la entrega de los productos: como la plataforma se enfoca en la venta de productos de segunda mano de manera local, puede ser difícil encontrar productos específicos o que se encuentren lejos de tu ubicación.

## 2.1.2 VIBBO



*Figura 8: Estado del arte - Vibbo*

**Vibbo** es una plataforma digital de compraventa de productos de segunda mano y nuevos, que anteriormente se conocía como Segundamano. La aplicación y la página web de Vibbo permiten a los usuarios comprar y vender todo tipo de productos, desde productos electrónicos y de informática hasta ropa, vehículos, inmuebles y mucho más.

Una de las principales ventajas de Vibbo es que permite realizar búsquedas de productos cercanos a tu ubicación, lo que facilita las transacciones entre compradores y vendedores. La plataforma también ofrece la posibilidad de contactar directamente con el vendedor, a través de la mensajería interna de la aplicación, para negociar el precio o hacer preguntas sobre el producto.

Otra característica destacada de Vibbo es su sistema de valoraciones y opiniones, que permite a los usuarios evaluar a los vendedores y ofrecer *feedback* sobre sus experiencias de compra. Esto ayuda a mantener la integridad de la plataforma y a garantizar que los vendedores cumplan con las expectativas de los compradores.

En definitiva, Vibbo es una plataforma de compraventa que se ha convertido en una herramienta útil y popular para aquellos que quieren comprar o vender productos de segunda mano o nuevos. Su enfoque en la cercanía y la facilidad de uso lo convierte en una opción atractiva para aquellos que buscan una forma rápida y sencilla de realizar transacciones comerciales en línea.



Figura 9: Estado del arte Vibbo Inicio

Tras realizar un análisis de la aplicación móvil se concluye que:

- ❖ **Vibbo** es una herramienta que dispone de un sistema de búsqueda muy avanzado el cual permite buscar publicaciones de productos con mayor exactitud sobre lo que desea encontrar el usuario.
- ❖ Posee también de una página web que muestra los servicios y productos que puede ofrecer a sus usuarios.
- ❖ Ofrece un sistema de verificación de usuarios donde mediante procedimientos de verificación internos, que tratan de verificar la identidad del vendedor para tratar evitar publicaciones falsas o estafas.

### 2.1.3 EBAY



Figura 10: Estado del arte - Ebay

eBay es una plataforma de comercio electrónico en línea que permite a los compradores y vendedores de todo el mundo conectarse y llevar a cabo transacciones comerciales. Los vendedores pueden utilizar la plataforma para ofrecer una amplia variedad de productos, desde artículos de colección y electrónica, hasta moda y productos para el hogar.

Los compradores pueden buscar productos en eBay y realizar compras directamente desde los vendedores. Además, eBay ofrece un sistema de subastas en línea donde los compradores pueden pujar por artículos y ganarlos al mejor postor. Los compradores pueden también hacer uso de la función "Compra inmediata" y adquirir un producto sin tener que esperar a que finalice una subasta.

eBay cuenta con un sistema de retroalimentación y calificaciones que permite a los compradores y vendedores evaluar su experiencia con una transacción específica. Los vendedores pueden mejorar su reputación y aumentar la confianza de los compradores mediante la entrega de productos de alta calidad y un servicio al cliente excepcional.

En resumen, eBay es una plataforma de comercio electrónico que ofrece a los compradores y vendedores una manera segura y fácil de conectarse y llevar a cabo transacciones comerciales en línea.

Analizando esta plataforma, se puede concluir que:

- ❖ No existe un servicio de mensajería instantánea integrado que permita al usuario vendedor y al cliente comunicarse, este sistema de mensajería es mediante mensajes directos, pero no en formato *chat*.
- ❖ La plataforma está más orientada a la venta de artículos nuevos, y no promueve el concepto de C2C (Consumer-to Consumer).
- ❖ También dispone de un sistema de subasta de artículos, donde los usuarios pujan por el producto hasta que finalmente lo adquiere aquel que mayor cantidad ha pujado en el momento que finalice la subasta.



Figura 11: Estado del arte - eBay Inicio

## 2.2 METODOLOGÍA

Tras el análisis de distintas metodologías propuestas a la hora de implementar un modelo de trabajo en el desarrollo del software, se ha optado por **un modelo iterativo incremental**, ya que es el modelo ideal para el desarrollo de aplicaciones para dispositivos móviles, dado que durante el proceso de desarrollo e implementación se produce la aparición de nuevos requisitos y nuevas funcionalidades.

Este tipo de metodología se basa en etapas incrementales que se van repitiendo hasta que obtenemos el resultado final del desarrollo que deseamos. Como razón principal y que fundamenta la elección de esta metodología es la posibilidad de ir desarrollando la aplicación, generando nuevas versiones, completando el resultado de la etapa anterior, hasta el resultado final.

Es decir, una etapa está compuesta por distintas fases desarrolladas de forma iterativa y, según se van realizando etapas, estas van añadiendo nuevas funcionalidades al desarrollo.

Las fases por las que transcurre cada etapa son:

- Análisis de los requisitos del proyecto
- Fase de diseño
- Elaboración de código fuente.
- Prueba de la funcionalidad desarrollada.

Una de las características que conlleva este modelo de trabajo incremental es que los tiempos de entrega se ven reducidos, respecto a otras metodologías como podría ser el modelo en cascada. Además, al tratarse de un modelo incremental, no debemos abordar todos los requisitos al comienzo del proyecto, sino que debemos ir abordando los requisitos cuando surja la aparición de nuevas funcionalidades, donde estas se pueden analizar e incluir en el siguiente incremento.

Otra característica importante que ha permitido decantarse por la utilización de esta metodología es que permite detectar errores y corregirlos, acotando el impacto que pueden producir estos errores durante el desarrollo. La realización de entregas periódicas de los incrementos de la aplicación nos permite estar en contacto con el cliente durante más fases en el proceso de desarrollo, lo que da lugar a que el cliente se vea más implicado en el desarrollo del mismo. Por parte del desarrollador, permite realizar pruebas de funcionalidad constantes, lo que facilita detectar vulnerabilidades en el sistema.

En el caso de esta aplicación móvil, el desarrollo ha sido realizado únicamente por una persona, encargada de realizar todas las etapas del proceso. En definitiva, las etapas de análisis, diseño, codificación, realización de pruebas y documentación han sido realizadas por el mismo desarrollador.

Según podemos observar en la imagen que aparece a continuación, se observa que cuánto más grande es el incremento, más funcionalidad tendrá el sistema a desarrollar.

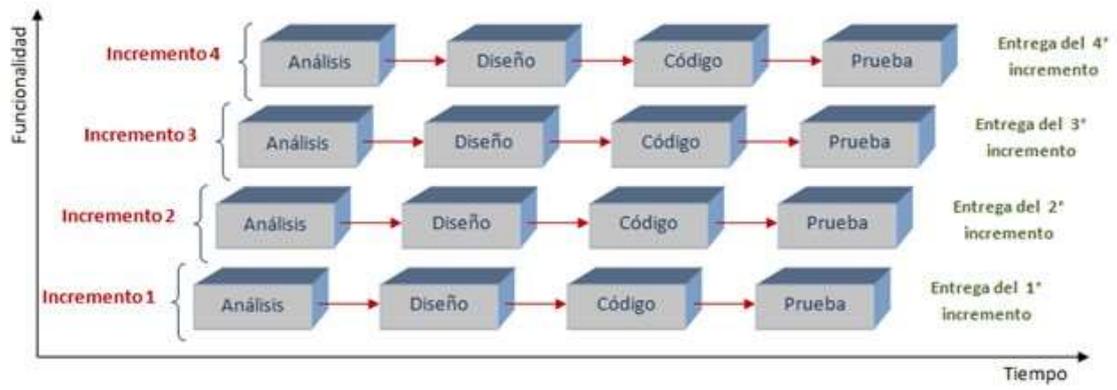


Figura 12: Modelo incremental iterativo



---

### 3. PLANIFICACIÓN, ESTIMACIÓN Y PRESUPUESTO

---



### 3.1 PLANIFICACIÓN

Tras haber establecido, en la sección anterior, la metodología que deseamos utilizar para este proyecto, es imprescindible realizar una planificación de las tareas que se van a realizar, para poder obtener, aproximadamente, la fecha de finalización del desarrollo del proyecto.

La planificación es un apartado que recoge la estimación de tiempo y los costes que nos van a suponer los objetivos deseados para la consecución del proyecto.

### 3.2 PLANIFICACIÓN TEMPORAL

La planificación inicial del proyecto se realizó fijando el 12 de septiembre de 2022, fecha en la cual se empezó a realizar la primera iteración del proyecto, compaginado con mi horario laboral de lunes a jueves de 8:00 a 17:00 y los viernes de 8:00 a 14:00, debido a mi empleo como becario en Ingeniería de clientes para el grupo Telefónica de España. Así pues, el horario de trabajo para la realización de este proyecto se ha visto reducido a un horario de 18:30 a 22:30, de lunes a jueves y de 16:00 a 22:00 los viernes. Complementando estos horarios con los fines de semana, pudiendo establecer el horario de 10:00 a 14:00 y de 16:00 a 20:00 los sábados, y los domingos con un horario similar.

El calendario laboral utilizado en este caso ha sido el correspondiente a los últimos meses de 2022 y hasta el mes de junio de 2023, para la Comunidad de Madrid.

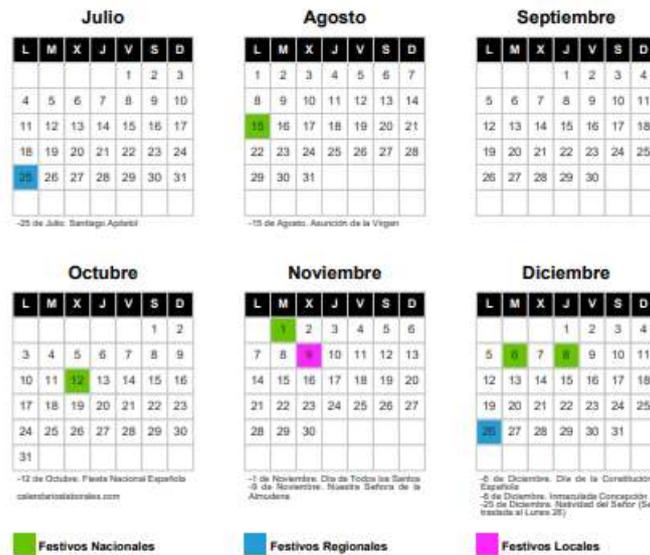


Figura 13: Calendario planificación inicial 2022



Figura 14: Calendario planificación Inicial 2023

Como se describió en la sección “2.2 Metodología”, al aplicar un **modelo iterativo incremental**, es necesario fijar una serie de iteraciones. Se han considerado cuatro iteraciones, donde las primeras fases del proyecto, tendrá un mayor peso en el tiempo empleado en la elicitación de requisitos, en el análisis, diseño y codificación que en el resto de fases del proyecto. En la iteración final se hará más hincapié en la realización de pruebas, tanto de caja negra como de caja blanca, para realizar una depuración del código total y la redacción de la documentación técnica.

Para este tipo de modelo funcional iterativo incremental, donde cada iteración es un incremento en el proyecto, se ha establecido una planificación en la que cada iteración está dividida en seis fases, tal y como podemos observar en la siguiente figura. Las iteraciones que se han considerado son:

- **Primera iteración.** El primer incremento se considera fundamental en la planificación inicial del proyecto. Por lo tanto, se destina un mayor tiempo para las fases de planificación, análisis y diseño. Relegándose a un segundo plano el esfuerzo en implementación esta iteración.

Así pues, la implementación, pruebas y documentación se enfocará en funcionalidades básicas como la gestión de los distintos usuarios de la aplicación, gestión del perfil por parte del propio usuario, la gestión de las publicaciones de productos con la información básica, y la visualización en detalle del producto. Además del desarrollo de la parte *front-end* de las funcionalidades desarrolladas.

- **Segunda iteración.** El segundo incremento comprenderá todo lo relacionado con la gestión de las funcionalidades algo más complejas y que complementan lo desarrollado anteriormente, como la utilización de la propia cámara del dispositivo para añadir imágenes sobre el producto sobre el cual vamos a realizar la publicación, mejora de la accesibilidad de la aplicación modificando el *front-end* desarrollado e incorporando una nueva funcionalidad donde el usuario pueda consultar las publicaciones sobre productos favoritos, cuales tiene actualmente publicados y los productos que ya han sido seleccionados como vendidos. Se estima que este incremento supone

un menor número de días al ser un conjunto de funcionalidades que no son de las más importantes dentro de la aplicación, y además de cara al proyecto, tiene una menor dificultad que las demás características que quedan por abordar.

- **Tercera iteración.** Se estima un esfuerzo mayor para llevar a cabo este incremento, esto se debe a que este incremento se centra en el desarrollo de la funcionalidad relacionada con la mensajería instantánea, donde se desarrollan todas aquellas funcionalidades de gran nivel como la posibilidad de compartir imágenes, tanto tomada al instante, como desde galería, y la integración de un sistema de seguimiento de ubicación en tiempo real que permite compartir entre usuarios, tanto su posición actual como cualquier ubicación. Por lo tanto, se prevé que la implementación será mucho más costosa que en las iteraciones anteriores.
- **Cuarta iteración.** El último incremento estaría relacionado con el desarrollo del sistema de valoración entre usuarios que permita establecer una valoración al usuario que ha realizado dicha publicación. También se añaden distintas funcionalidades de menor dificultad como la posibilidad de editar información de publicaciones ya realizadas previamente, y pequeñas implementaciones de diseño para mejorar la accesibilidad. Además, en esta iteración la documentación tomará un mayor peso, ya que al tratarse del final del proyecto se producirán distintas modificaciones en esta memoria.

### 3.3 PUNTOS DE FUNCIÓN

La estimación por puntos de función trata de realizar el cálculo del coste del proyecto en base a distintos factores que permiten evaluar las diferentes funcionalidades que contendrá la aplicación a desarrollar en este proyecto. Una premisa clara, es que cuánto mayor complejidad tenga la funcionalidad, más tiempo se deberá invertir en su desarrollo, lo que implica que el coste será mayor.

Los pasos que se deben llevar a cabo son los siguientes:

1. Realizar el cálculo de los puntos de función sin ajustar.
2. Calcular los factores de ajuste.
3. Calcular los puntos de función ajustados.

Comenzamos llevando a cabo el primer paso descrito anteriormente, que trata de realizar el cálculo de los puntos de función sin ajustar, los cuales podemos distribuir en las siguientes categorías:

- ❖ Entradas de usuario
- ❖ Salidas de usuario
- ❖ Consultas de usuario
- ❖ Ficheros Lógicos Internos
- ❖ Ficheros de Interfaces Externos

Tras distribuir los diferentes **puntos de función sin ajustar (PFSA)** en las cinco categorías, será imprescindible asignar un nivel de complejidad a cada componente.

- ❖ **Entradas de usuario:** Se trata de aquellos datos que el usuario introduce en el sistema.

ENTRADAS	COMPLEJIDAD
Información y datos de registro del usuario (Nombre, apellidos, password, email )	Media
Formulario registro usuario en la aplicación	Baja
Formulario de inicio de sesión con cuenta de Google	Baja
Información y datos para añadir una nueva publicación de un producto	Media
Formulario modificar perfil ( avatar )	Media
Formulario modificar publicación	Media
Formulario motor de búsqueda de publicaciones	Baja
Información y datos sobre la publicación	Alta
Información y datos del usuario vendedor	Baja
Formulario para valoración hacia un usuario vendedor	Media
Formulario para nuevo mensaje de mensajería instantánea	Baja
Información y datos sobre imágenes enviados mediante mensajería instantánea.	Baja
Formulario motor de búsqueda ubicación	Media
Información y datos sobre la ubicación actual del usuario	Media

*Tabla 5: Puntos de función (Entradas de usuario)*

❖ **Salidas:** Se trata de aquellos datos que el sistema refleja al usuario en la aplicación (pantallas, mensajes de error, mensajes de confirmación).

SALIDAS	COMPLEJIDAD
Pantalla principal DASSEL	Baja
Pantalla principal: inicio de sesión	Baja
Pantalla visualización y modificación de perfil	Media

Pantalla listado de productos publicados por usuarios vendedores	Alta
Pantalla listado de productos publicados por usuarios vendedores filtrados por categoría	Baja
Pantalla listado de productos publicados por usuarios vendedores filtrados por precio	Baja
Pantalla productos publicados por el usuario autenticado	Media
Pantalla publicaciones favoritas	Baja
Pantalla publicaciones marcadas como artículos vendidos	Baja
Pantalla añadir una nueva publicación	Alta
Pantalla para seleccionar una categoría	Baja
Pantalla para seleccionar una subcategoría	Baja
Pantalla sobre los detalles de la publicación	Baja
Pantalla añadir imágenes sobre el producto	Media
Pantalla con aplicación de cámara del dispositivo móvil	Media
Pantalla introducir precio y ubicación de la publicación	Baja
Pantalla información detallada de la publicación	Media
Pantalla modificación de datos de publicación	Baja
Listado de conversaciones abiertas mediante mensajería instantánea	Baja
Pantalla nuevo mensaje mediante teclado	Baja
Listado mensajes recibidos ( mensajería instantánea )	Baja
Listado mensajes enviados (correo aplicación)	Baja
Pantalla compartir elementos con el usuario receptor ( imágenes , ubicación )	Baja
Pantalla selección de ubicación actual	Alta

Pantalla opciones de seguimiento en tiempo real	Baja
Pantalla sistema de valoración de los usuarios vendedores	Baja
Mensajes confirmación de operación realizada con éxito ( toda la aplicación )	Media
Mensajes error validación de datos (toda la aplicación)	Media

Tabla 6: Puntos de función (Salidas)

- ❖ **Consultas de usuario:** Serán aquellas llamadas al sistema que devuelven una respuesta software interactiva al usuario.

CONSULTAS	COMPLEJIDAD
Información acceso a la aplicación	Baja
Información página principal acceso usuario	Baja
Información modificar perfil usuario autenticado	Baja
Información listado de publicaciones por otros usuarios	Alta
Información detalle del producto de la publicación	Baja
Información detalles que se pueden modificar de la publicación	Baja
Información valoraciones de usuarios al usuario vendedor	Baja
Información publicaciones del usuario vendedor	Baja
Información sobre productos publicados	Baja
Información sobre productos guardados como favoritos	Baja
Información sobre productos vendidos	Baja
Información sobre categorías disponibles	Baja
Información sobre subcategorías disponibles	Baja
Información sobre chats de mensajerías instantánea activos	Media
Información chat ( mensajes recibidos y enviados )	Baja

Información imágenes compartidas	Baja
Información ubicación compartida	Media
Información ubicación en tiempo real	Media
Información estado <i>online/offline</i> del usuario	Baja

Tabla 7: Puntos de función (Consultas de usuario)

- ❖ **Ficheros lógicos internos:** Se trata de aquellos ficheros o bases de datos que utiliza internamente el sistema para la salvaguarda de datos.

FICHEROS LOGICOS INTERNOS	COMPLEJIDAD
Base de datos Cloud Firestore DASELL	Alta

Tabla 8: Puntos de función (Ficheros Lógicos Internos)

- ❖ **Ficheros externos:** Se trata de aquellos ficheros externos al sistema y pueden ser vistos por otras aplicaciones

FICHEROS EXTERNOS	COMPLEJIDAD
Manual aplicación DASELL	Baja

Tabla 9: Puntos de función (Ficheros externos)

Tras obtener todos los puntos de función sin ajustar, se debe realizar el sumatorio de todos los **puntos de función sin ajustar (PFSA)**, mediante el cálculo de la ponderación en base del tipo asociado y su factor de complejidad.

Tipo de función	Complejidad	Total x Complej.	Total por tipo	Suma
<b>Ficheros lógicos internos (PFTif)</b>	Baja	0x7	0	15
	Media	0x10	0	
	Alta	1x15	15	
<b>Ficheros externos (PFTef)</b>	Baja	1x5	5	5
	Media	0x7	0	
	Alta	0x10	0	

<b>Entradas de usuario (PFTe)</b>	Baja	6x3	18	52
	Media	7x4	28	
	Alta	1x6	6	
<b>Salidas de usuario (PFTo)</b>	Baja	18x4	72	128
	Media	7x5	35	
	Alta	3x7	21	
<b>Consultas de usuario (PFTq)</b>	Baja	15x3	45	63
	Media	3x4	12	
	Alta	1x6	6	
<b>Total de puntos de función</b>				263

Tabla 10: Sumatorio ponderación Puntos de Función

Tras obtener el sumatorio total de los PFSA = 263, se realiza el cálculo de la variable **Factor de Ajuste (FA)** para proceder al ajuste de estos puntos de función. Para llevarlo a cabo, el cálculo del Factor de Ajuste se basa en 14 factores que miden la funcionalidad general y la complejidad de la aplicación.

Para realizar el cálculo de la complejidad de cada una de las características, se establece una ponderación entre un intervalo del 0 a 5, siendo el 0 la puntuación más baja y el 5 la más alta.

<b>Factores de ajuste</b>	<b>Complejidad</b>
1. Comunicación de datos	4
2. Funciones distribuidas	0
3. Prestaciones	4
4. Gran uso de la configuración	1
5. Velocidad de las transacciones	4
6. Entrada on-line de datos	3
7. Diseño para la eficiencia del usuario final	2
8. Actualización de datos on-line	3
9. Complejidad proceso lógico interno de la aplicación	4
10. Reusabilidad del código	4
11. Facilidad de instalación	4
12. Facilidad de operación	4
13. Localizaciones múltiples	1
14. Facilidad de cambios	3

<b>TOTAL:</b>	<b>41</b>
---------------	-----------

*Tabla 11: Factor de Ajuste*

Tras establecer la tabla con los valores otorgados a los factores de ajuste, que resultan de una **complejidad para este caso de 41**, se aplica la siguiente formula que nos permite obtener el factor de ajuste FA:

$$FA = (0.01 \times \sum FC) + 0.64$$

$$FA = (0.01 \times 41) + 0.64$$

$$FA = 1.05$$

De esta manera, se obtiene un **FA = 1,05**, valor que nos permite obtener los puntos de función ajustados (PFA). Dado que previamente, se ha realizado en cálculo de los PFSA, donde hemos obtenido un valor de **263**, el resultado de los puntos de función ajustados aplicando la siguiente formula es de:

$$PFA = PFSA * FA$$

$$PFA = 263 * 1.05$$

$$PFA = 276,15$$

Concluyendo, tras realizar el cálculo de los Puntos de Función Ajustados (PFA), se puede proceder a realizar el cálculo de la duración del proyecto y del coste final que supondrá.

Se establece, según el histórico, que **un PFA equivale a 4 horas**, por lo que obtenemos:

$$Esfuerzo \text{ en horas} = PFA \times 4 \text{ horas}$$

$$Esfuerzo \text{ en horas} = 276,15 \times 4$$

$$Esfuerzo \text{ en horas} = 1.104,6 \text{ horas}$$

Tras disponer del cálculo del esfuerzo en horas de la aplicación, podemos realizar la planificación inicial, a través del uso de Microsoft Project donde establecemos el siguiente calendario en base a las horas de trabajo especificadas anteriormente y teniendo en cuenta los días festivos tanto nacionales como de la Comunidad de Madrid.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
<b>DaSell</b>	<b>1104 horas</b>	<b>lun 07/11/22</b>	<b>mié 10/05/23</b>	
<b>Primera iteración</b>	<b>432 horas</b>	<b>lun 07/11/22</b>	<b>vie 20/01/23</b>	

Elicitación	6 días	lun 07/11/22	mar 15/11/22	
Análisis	6 días	mar 15/11/22	mié 23/11/22	2
Diseño	8 días	mié 23/11/22	sáb 03/12/22	3
Codificación	31 días	sáb 03/12/22	dom 15/01/23	4
Pruebas	3 días	dom 15/01/23	vie 20/01/23	5
Fin de Iteración 1	0 horas	vie 20/01/23	vie 20/01/23	6
<b>Iteración 2</b>	<b>264 horas</b>	<b>vie 20/01/23</b>	<b>vie 03/03/23</b>	
Elicitación	3 días	vie 20/01/23	lun 23/01/23	7
Análisis	3 días	lun 23/01/23	sáb 28/01/23	9
Diseño	4 días	sáb 28/01/23	mié 01/02/23	10
Codificación	21 días	mié 01/02/23	lun 27/02/23	11
Pruebas	2 días	mar 28/02/23	vie 03/03/23	12
Fin de Iteración 2	0 horas	vie 03/03/23	vie 03/03/23	13
<b>Iteración 3</b>	<b>184 horas</b>	<b>vie 03/03/23</b>	<b>sáb 01/04/23</b>	
Elicitación	2 días	vie 03/03/23	dom 05/03/23	14
Análisis	3 días	dom 05/03/23	jue 09/03/23	16
Diseño	3 días	jue 09/03/23	dom 12/03/23	17
Codificación	12 días	dom 12/03/23	mar 28/03/23	18
Pruebas	3 días	mié 29/03/23	sáb 01/04/23	19
Fin de Iteración 3	0 horas	sáb 01/04/23	sáb 01/04/23	20
<b>Iteración 4</b>	<b>224 horas</b>	<b>dom 02/04/23</b>	<b>mié 10/05/23</b>	
Elicitación	1 día	dom 02/04/23	dom 02/04/23	21
Análisis	2 días	dom 02/04/23	mié 05/04/23	23
Diseño	3 días	mié 05/04/23	lun 10/04/23	24
Codificación	9 días	lun 10/04/23	sáb 22/04/23	25
Pruebas	3 días	sáb 22/04/23	mié 26/04/23	26
Fin de iteración 4	10 días	mié 26/04/23	mié 10/05/23	27

Documentación final	0 días	mié 10/05/23	mié 10/05/23	28
Fin de proyecto	0 días	mié 10/05/23	mié 10/05/23	29

*Tabla 12: Planificación inicial*

En la anterior figura, se refleja la planificación que se va a seguir para obtener el resultado final deseado en el tiempo que se ha establecido. Cabe destacar, que el total de horas de trabajo será de 1104 horas, distribuidas en 138 días laborables.

Mediante la planificación temporal obtenida, la primera iteración es la que más duración comprende, con una duración total de 432 horas. En consecuencia, la segunda iteración consta de 264 horas; la tercera iteración consta de 184 horas; por último, la cuarta iteración comprende un total de 224 horas. Por lo tanto, debido a la obtención los datos obtenidos en dicha planificación, mediante la herramienta de Microsoft Project, se ha podido generar el siguiente diagrama de Gantt

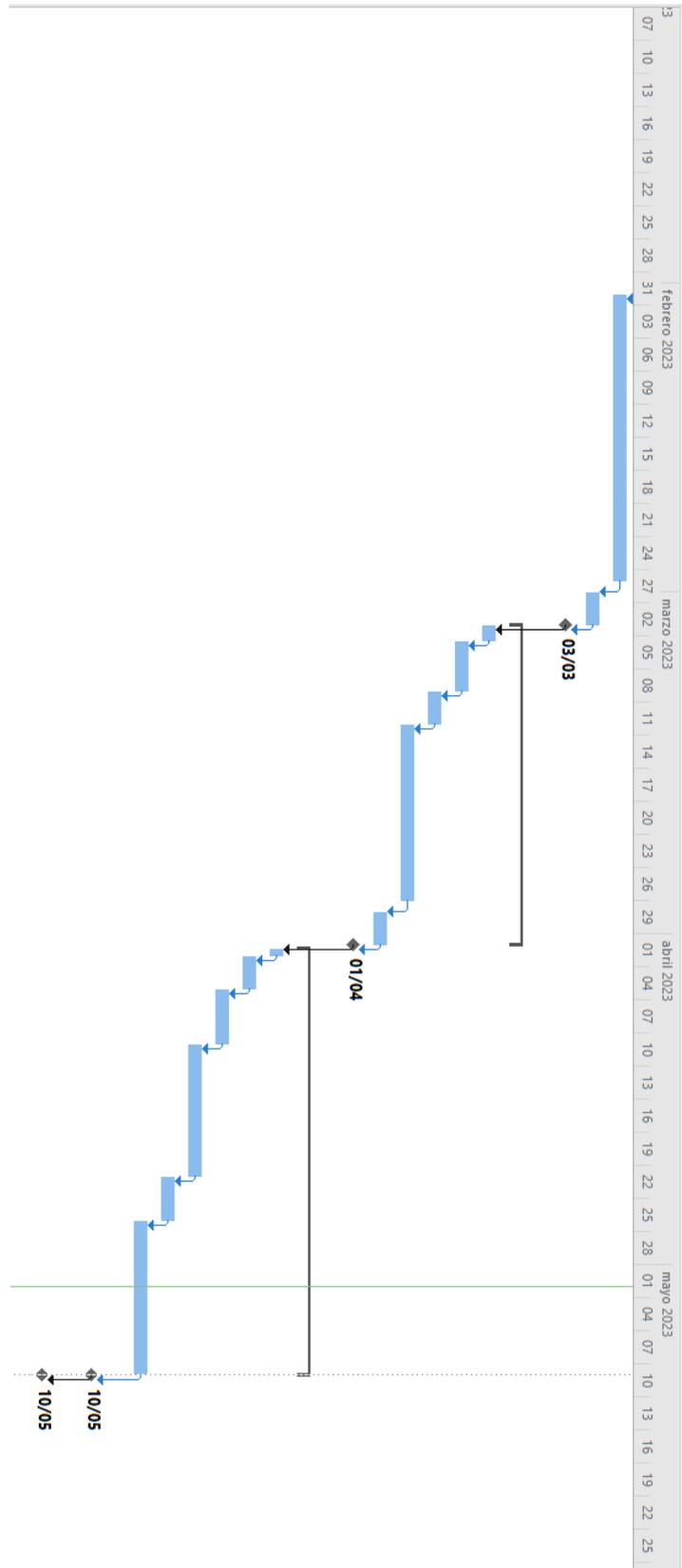


Tabla 13: Diagrama de Gantt - Planificación inicial

El estudio de la planificación anterior que se ha realizado ha dado lugar a una duración total del proyecto en base al tiempo de cada tarea y las horas de trabajo que se realizarán.

La planificación anterior realizada ha dado la duración total del proyecto en base al tiempo de cada tarea y las horas de trabajo que se realizarán. En el siguiente apartado se pretende dar una estimación del coste total del producto software en base a 3 diferentes bloques: Hardware, Software y recursos humanos.

## 3.4 COSTES

En este apartado, se plantea como objetivo realizar una estimación del coste total del producto software en base a 3 diferentes bloques: Recursos humanos, Hardware, Software.

### 3.4.1. RECURSOS HUMANOS

En cuanto al proyecto a desarrollar, se debe tener una casuística como premisa inicial en la que hay que tener en cuenta que el equipo de desarrollo al completo estará formado únicamente por un miembro, el cual realizará los cuatro distintos roles dentro del transcurso del proyecto: analista, desarrollador, arquitecto de software y *tester*<sup>1</sup>.

Podemos repartir las funciones entre los distintos roles del equipo de desarrollo:

- El analista, que realiza un papel fundamental para el proceso de desarrollo, tendrá como tarea encargarse de la elicitación, análisis y diseño de la aplicación.
- El desarrollador, en el que se puede establecer en concreto que se trata de un desarrollador en Flutter bajo lenguaje Dart, con experiencia en el desarrollo de aplicaciones móviles. Tendrá como tarea desarrollar tanto el *front-end* como el *back-end* de la aplicación.
- Arquitecto de software, encargado de realizar la documentación sobre el análisis inicial y el posterior desarrollo de la aplicación
- El *tester*, tiene como finalidad realizar las pruebas de código sobre lo que ya ha sido programado por el desarrollador, verificando que se ha realizado correctamente.

Por el momento, debemos establecer el tiempo que destinará cada uno de los roles anteriormente descrito en el desarrollo del proyecto. Los costes que se establecen por hora de trabajo han sido

---

<sup>1</sup> **Tester:** profesional que se encarga de evaluar y verificar la calidad y el funcionamiento de software, aplicaciones o sistemas. Su objetivo principal es descubrir posibles errores, fallos o deficiencias en el software, para garantizar que cumpla con los requisitos y expectativas establecidos. Los testers utilizan diversas técnicas y herramientas para diseñar y ejecutar pruebas, analizar los resultados y reportar los problemas encontrados. Su trabajo contribuye a mejorar la calidad y la confiabilidad del software antes de su lanzamiento o implementación.

El número de horas que desempeñará cada rol, serán las estimadas en la planificación inicial del proyecto. El coste por hora, se ha obtenido en función del precio actual del mercado.

RECURSOS HUMANOS	TIEMPO (HORAS)	COSTE / €	COSTE
Analista	378	16,26	6.146,26 €
Desarrollador	469	17,80	8.348,20 €
Tester	127	12,10	1.536,70 €
Arquitecto de software	105	11,50	1.207,50 €
		<b>TOTAL</b>	<b>17.238,66 €</b>

Tabla 14: Planificación Inicial - Recursos Humanos

### 3.4.2. HARDWARE

En el caso de los dispositivos hardware implementados, para realizar el cálculo de los costes se ha debido establecer como base el tiempo de utilización que se le dará en base a la planificación realizada previamente, en consecuencia, se describe un listado de los componentes y el porcentaje de uso correspondiente a su vida útil.

- **CPU de sobremesa PcCom Silver**

- ❖ Procesador: AMD Ryzen 5 5500 3.6GHz
- ❖ Placa Base: Gigabyte B550M DS3H
- ❖ Disco duro: 500GB SSD NVMe M.2 2280
- ❖ Memoria: 16GB DDR4 3200 2x8GB CL16
- ❖ Gráfica: AMD Radeon RX 6600 8GB GDDR6
- ❖ Tiempo de vida útil, según las especificaciones y garantía de fabricante → 8 años

Duración del proyecto ≡ 28 semanas → ¿% vida útil?

Vida útil → 8 años ≡ 417 semanas ≡ 100% de vida útil completada

**% vida útil = 6,71%**

- **HP LaserJet M110we Impresora Láser Monocromo WiFi**

- ❖ Velocidad de impresión: 20 ppm
- ❖ Ciclo de trabajo (máximo): 8000 páginas por mes
- ❖ Tiempo de vida útil, según las especificaciones y garantía de fabricante → 7 años

Duración del proyecto ≡ 28 semanas → ¿% vida útil?

Vida útil → 7 años ≡ 365 semanas ≡ 100% de vida útil completada

**% vida útil = 7,6712%**

- **Periféricos: monitor, ratón y teclado**

- ❖ Tiempo de vida útil, según las especificaciones y garantía de fabricante → 5 años

Duración del proyecto ≡ 28 semanas → ¿% vida útil?

Vida útil → 5 años ≡ 260 semanas ≡ 100% de vida útil completada

**% vida útil = 10,77%**

- ❖ **Conectividad a Internet:** Fibra óptica simétrica de 1GB, contratada con la compañía telefónica Movistar. Precio: 43,90 €/mes. Se cobrará el importe relativo a la duración del proyecto, 7 meses.

- ❖ **Material de oficina**

HARDWARE	PRECIO	USO	COSTE TOTAL
Ordenador	1253,67 €	6,71 %	83,99€
Impresora	93,98 €	7,67 %	7,14 €
Pantalla, ratón y teclado	160,00 €	10,77 %	17,12 €
Conectividad Internet	43,90 €	7 meses	307,30 €
Material de oficina	38,00 €	100,00 %	38,00 €
			<b>453,55 €</b>

*Tabla 15: Planificación Inicial - Hardware*

### 3.4.3. SOFTWARE

En el caso del Software, se han implementado diferentes herramientas para el desarrollo de la aplicación. La mayoría son herramientas de software libre que se pueden descargar de manera gratuita. A continuación, se describen las herramientas utilizadas:

- ❖ Software libre:

- Visual Studio Code
- Postman
- Adobe Reader
- Firebase
- StarUML
- Draw.io

- ❖ Pago por uso

- Microsoft Office 2020: Necesario para la utilización de Microsoft Word, Microsoft PowerPoint, Microsoft Project, Visio.
- Windows 10 Pro

SOFTWARE	PRECIO	USO	COSTE TOTAL
Microsoft Office 2016	249,99 €	6,71 %	16,77 €
Windows 10 Pro	48,99 €	6,71 %	3,28 €
<b>TOTAL</b>			<b>20,05 €</b>

Tabla 16: Planificación Inicial - Software

### 3.5. PRESUPUESTO EN BASE A LA PLANIFICACIÓN INICIAL

Por lo tanto, el presupuesto total sumando todos los costes (Hardware, Software y recursos humanos) sería:

PRESUPUESTO TOTAL	COSTE
Software	20,05 €
Hardware	453,55 €
Recursos humanos	9250,20 €
<b>TOTAL</b>	<b>17.238,66 €</b>

Tabla 17: Presupuesto total - Planificación Inicial

### 3.6 SEGUIMIENTO REAL

En el momento en el que se realizó esta planificación inicial, al comienzo del proyecto, se tuvo en cuenta la elaboración del mismo a la vez que se mantenía un horario laboral debido al trabajo en el que me encuentro actualmente. Se despreció la alta carga de trabajo que puede suponer desempeñar el desarrollo de este proyecto de manera individual, haciendo que, debido a complicaciones y retrasos, existan horas poco efectivas en la planificación.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
<b>DaSell</b>	<b>1232 horas</b>	<b>lun 07/11/22</b>	<b>mar 30/05/23</b>	
<b>Primera iteración</b>	<b>448 horas</b>	<b>lun 07/11/22</b>	<b>dom 22/01/23</b>	
Elicitación	6 días	lun 07/11/22	mar 15/11/22	
Análisis	6 días	mar 15/11/22	mié 23/11/22	2
Diseño	8 días	mié 23/11/22	sáb 03/12/22	3
Codificación	33 días	sáb 03/12/22	mié 18/01/23	4
Pruebas	3 días	mié 18/01/23	dom 22/01/23	5
Fin de Iteración 1	0 horas	dom 22/01/23	dom 22/01/23	6
<b>Iteración 2</b>	<b>280 horas</b>	<b>dom 22/01/23</b>	<b>mar 07/03/23</b>	
Elicitación	3 días	dom 22/01/23	jue 26/01/23	7

Análisis	4 días	jue 26/01/23	lun 30/01/23	9
Diseño	4 días	mar 31/01/23	dom 05/02/23	10
Codificación	22 días	dom 05/02/23	dom 05/03/23	11
Pruebas	2 días	dom 05/03/23	mar 07/03/23	12
Fin de Iteración 2	0 horas	mar 07/03/23	mar 07/03/23	13
<b>Iteración 3</b>	<b>240 horas</b>	<b>mar 07/03/23</b>	<b>dom 16/04/23</b>	
Elicitación	2 días	mar 07/03/23	sáb 11/03/23	14
Análisis	3 días	sáb 11/03/23	lun 13/03/23	16
Diseño	5 días	mar 14/03/23	mar 21/03/23	17
Codificación	17 días	mar 21/03/23	jue 13/04/23	18
Pruebas	3 días	jue 13/04/23	dom 16/04/23	19
Fin de Iteración 3	0 horas	dom 16/04/23	dom 16/04/23	20
<b>Iteración 4</b>	<b>264 horas</b>	<b>dom 16/04/23</b>	<b>mar 30/05/23</b>	
Elicitación	2 días	dom 16/04/23	mié 19/04/23	21
Análisis	3 días	jue 20/04/23	dom 23/04/23	23
Diseño	4 días	dom 23/04/23	sáb 29/04/23	24
Codificación	11 días	sáb 29/04/23	dom 14/05/23	25
Pruebas	3 días	dom 14/05/23	jue 18/05/23	26
Fin de iteración 4	10 días	jue 18/05/23	mar 30/05/23	27
Documentación final	0 días	mar 30/05/23	mar 30/05/23	28
Fin de proyecto	0 días	mar 30/05/23	mar 30/05/23	29

Tabla 18: Planificación Seguimiento real

Tras realizar de nuevo una nueva planificación temporal mediante Microsoft Project, introduciendo las fechas reales, es decir, no estimadas del comienzo de casa iteración, obtenemos el Diagrama de Gantt y planificación temporal donde observamos las siguientes características:

- La fecha de comienzo del proyecto es el 7 de noviembre de 2022 al igual que en la planificación inicial.
- La fecha de finalización es el 30 de mayo de 2023, con una diferencia de 20 días respecto a la fecha de finalización estimada, el 10 de mayo del 2023.
- El desarrollo del proyecto ha supuesto una duración de 1232 horas, respecto a las 1104 horas estimadas inicialmente. Esto supone un incremento de 128 horas respecto a la planificación inicial.
- La mayoría de días de retraso respecto en la planificación inicial se ha debido a complicaciones e imprevistos a la hora de las etapas de análisis o codificación, lo que ha supuesto un retraso en las iteraciones.

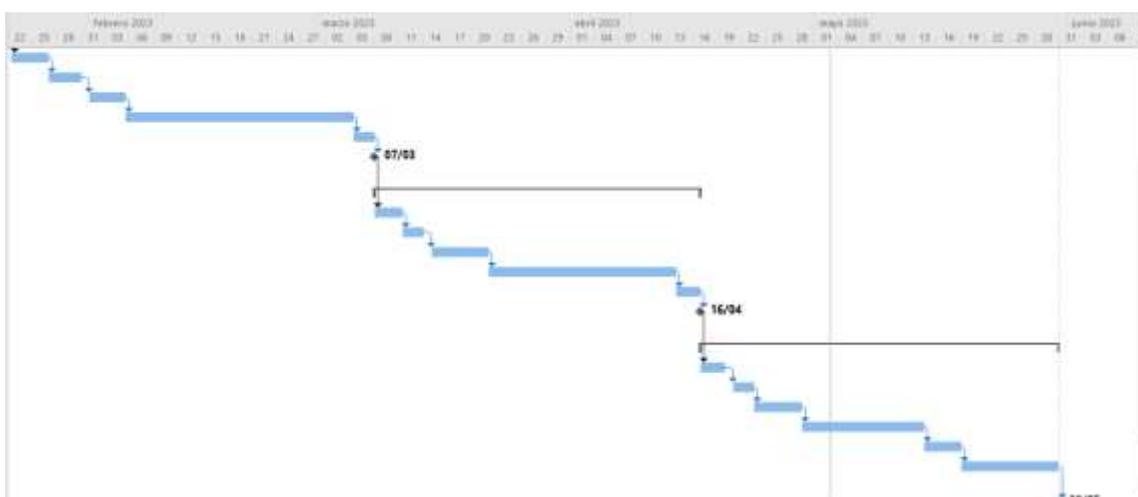


Tabla 19: Diagrama de Gantt - Planificación Seguimiento real

## 3.7 COSTE REAL

### 3.7.1 RECURSOS HUMANOS

En comparación con los recursos estimados previamente, al producirse un retraso en la fecha de entrega del proyecto desarrollado, se debe tener en cuenta que también aumentará el tiempo que destinan los recursos humanos en la elaboración del proyecto.

RECURSOS HUMANOS	TIEMPO (HORAS)	COSTE / €	COSTE
Analista	398	16,26	6.417,48 €
Desarrollador	519	17,80	9.238,20 €
Tester	157	12,10	1.899,70 €
Arquitecto software	148	11,50	1.702,00 €
<b>TOTAL</b>			<b>19.251,38 €</b>

Tabla 20: Coste real - Recursos humanos

### 3.7.2 HARDWARE

En el caso de los dispositivos hardware implementados, se implementan los mismos componentes que los descritos en la planificación inicial. Debido al aumento de la duración del desarrollo de la aplicación, se producirá un aumento en el porcentaje de su uso, lo que implica, un incremento en el precio.

- **CPU de sobremesa PcCom Silver**

Duración del proyecto  $\equiv$  31 semanas  $\rightarrow$  ¿% vida útil?  
 Vida útil  $\rightarrow$  8 años  $\equiv$  417 semanas  $\equiv$  100% de vida útil completada

**% vida útil = 7,43%**

- **HP LaserJet M110we Impresora Láser Monocromo WiFi**

Duración del proyecto  $\equiv$  31 semanas  $\rightarrow$  ¿% vida útil?  
 Vida útil  $\rightarrow$  7 años  $\equiv$  365 semanas  $\equiv$  100% de vida útil completada

**% vida útil = 8,49%**

- **Periféricos: monitor, ratón y teclado**

Duración del proyecto  $\equiv$  31 semanas  $\rightarrow$  ¿% vida útil?  
 Vida útil  $\rightarrow$  5 años  $\equiv$  260 semanas  $\equiv$  100% de vida útil completada

**% vida útil = 11,92%**

- ❖ **Conectividad a Internet:** Fibra óptica simétrica de 1GB, contratada con la compañía telefónica Movistar. Precio: 43,90 €/mes. Se cobrará el importe relativo a la duración del proyecto, 8 meses.

- ❖ **Material de oficina**

HARDWARE	PRECIO	USO	COSTE TOTAL
Ordenador	1253,67 €	7,43 %	93,15€
Impresora	93,98 €	8,49 %	7,98 €
Pantalla, ratón y teclado	160,00 €	11,92 %	19,08 €

<b>Conectividad Internet</b>	43,90 €	8 meses	351,2 €
<b>Material de oficina</b>	38,00 €	100,00 %	38,00 €
<b>TOTAL</b>			<b>509,41 €</b>

Tabla 21: Coste real - Hardware

### 3.7.3 SOFTWARE

En el caso del Software, se han implementado las mismas herramientas para el desarrollo de la aplicación. Por lo que no se ha sufrido variaciones en las implementaciones. A continuación, se describen las herramientas utilizadas:

<b>SOFTWARE</b>	<b>PRECIO</b>	<b>USO</b>	<b>COSTE TOTAL</b>
<b>Microsoft Office 2016</b>	249,99 €	6,71 %	16,77 €
<b>Windows 10 Pro</b>	48,99 €	6,71 %	3,28 €
<b>TOTAL</b>			<b>20,05 €</b>

Tabla 22: Coste real - Software

### 3.7.4 COSTE REAL TOTAL

Por lo tanto, el presupuesto real total sumando todos los costes (Hardware, Software y recursos humanos) sería:

<b>PRESUPUESTO TOTAL</b>	<b>COSTE</b>
<b>Software</b>	20,05 €
<b>Hardware</b>	509,41 €
<b>Recursos humanos</b>	19.251,38 €
<b>TOTAL</b>	<b>19.780,84 €</b>

Tabla 23: Coste real

## 3.8 CONCLUSIONES

Para finalizar este capítulo, se puede extraer tras este análisis de la planificación presupuestaria, podemos decir que, comparando los datos del presupuesto fijado y el coste real que ha supuesto finalmente el proyecto, obtenemos una desviación de un 14%. Las razones principales para que se haya producido esta desviación son que la planificación inicial se realizó de una manera optimista, sin que fueran considerados los inconvenientes que podrían ocurrir durante el desarrollo del proyecto que dan lugar a retrasos en la finalización de cada iteración lo que supone una mayor duración y por consiguiente se han cumplido los hitos más tarde de lo previsto. El tiempo de desarrollo se ha excedido respecto a la duración que debiera tener un Trabajo de fin de Grado, según los créditos que corresponden dentro del plan de estudios de Ingeniería Informática de Servicios y Aplicaciones, principalmente por querer desarrollar el producto lo más completo posible.

El principal aumento de duración se ha debido al aumento de horas de codificación por cada iteración, lo que ha influido en el tiempo de entrega del proyecto y por lo tanto un aumento en los

costes del proyecto, en concreto, si la planificación inicial suponía 17.328,66 € y la planificación real ha supuesto 19.780,4 €, han aumentado los costes en 2.451,74 €.

Por último, cabe reseñar que no se ha utilizado la estimación COCOMO para la elaboración que supondría los costes de desarrollo, ya que los cálculos realizados tienen como objetivo lenguajes de bajo nivel, lo que supone que la estimación para este proyecto software tendría poca fiabilidad, al utilizar diversas librerías en la codificación de la aplicación

---

## 4. ANÁLISIS

---



Este capítulo va a proporcionar al lector el detalle de los diferentes requisitos, reglas y características como resultado del análisis de la aplicación a desarrollar.

## 4.1 ACTORES

Una tarea fundamental del desarrollo de software es obtener de manera correcta los actores que van a interactuar con el sistema. En primer lugar, se va a proceder a describir como se encuentra organizada la jerarquía de actores de la aplicación, donde se describen los distintos roles que desempeña cada usuario.

ACTORES		
ACT-01	Usuario no registrado	Es el usuario del sistema que no está registrado previamente en la aplicación, puede realizar las funcionalidades básicas dentro del sistema.
ACT-02	Usuario registrado y no identificado	Es el usuario que ya está registrado en la aplicación, pero no ha pasado por el proceso de autenticación.
ACT-03	Usuario registrado e identificado	Es el usuario registrado que describe un rol general que agrupa a todos los usuarios que acceden al sistema, tras haberse autenticado previamente.
ACT-04	Vendedor	Es el usuario registrado e autenticado en el sistema, que tiene el rol de “Vendedor”, y por lo tanto únicamente podrán realizar aquellas acciones que le corresponda.
ACT-05	Ciente	Es el usuario registrado e autenticado en el sistema, que tiene el rol de “Cliente”, y por lo tanto únicamente podrán realizar aquellas acciones que le corresponda.
ACT-06	Cloud Firestore	Herramienta o servicio en la nube donde se almacena toda la información en la que se apoya la aplicación.
ACT-07	API Google Maps	La API de Google Maps es un conjunto de herramientas y funcionalidades que permiten a los desarrolladores integrar mapas interactivos, datos de ubicación y servicios relacionados con la geolocalización en sus aplicaciones y sitios web. La API proporciona acceso a funciones como la visualización de mapas, la búsqueda de lugares y direcciones, la obtención de rutas y la incorporación de imágenes de Street View. Esto permite a los desarrolladores crear aplicaciones y servicios personalizados que

		utilizan la información geográfica para mejorar la experiencia del usuario.
<b>ACT-08</b>	<b>API MapBox</b>	Mapbox es una plataforma de mapas y ubicación que ofrece una API (Interfaz de Programación de Aplicaciones) que permite a los desarrolladores integrar mapas personalizados y datos de ubicación en sus aplicaciones y sitios web. La API de Mapbox proporciona acceso a una variedad de herramientas, como el geocodificador, el ruteador de direcciones y el renderizador de mapas, lo que permite a los desarrolladores crear experiencias de usuario personalizadas y visualmente atractivas con mapas y datos de ubicación.

Tabla 24: Descripción actores del sistema

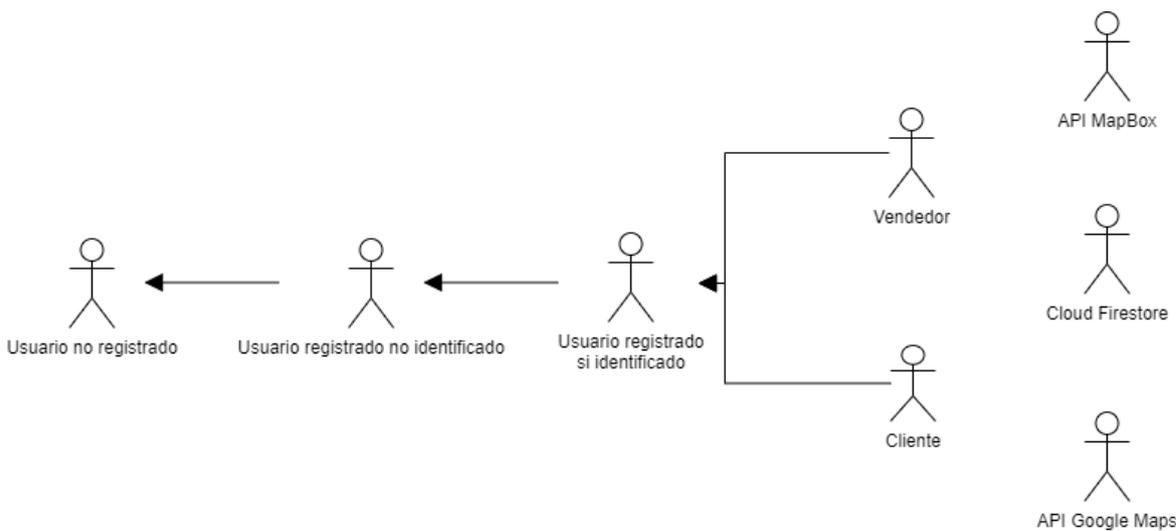


Figura 15: Jerarquía Actores del sistema

## 4.2 REGLAS DE NEGOCIO

Las **reglas de negocio** no son requisitos del software por sí mismas, pero son requisitos que se deben tener en cuenta, ya que toda empresa impone un conjunto de normas, tratados, procedimientos o restricciones de carácter obligatorio. Por lo tanto, estas reglas serán la base y afectan al resto de requisitos que encontramos en nuestro proyecto a desarrollar, ya que establece ciertas limitaciones y están relacionados con todos ellos.

En consecuencia, podemos definir a las **reglas de negocio**, definidas mediante el identificador **RN(-OX)**, como aquellas que establecen todas las regulaciones y políticas de negocio que deberá seguir el desarrollo de acuerdo a lo establecido por las normas vigentes actualmente. El nuevo sistema se debe ajustar

a estas nuevas normativas, por lo que la aplicación deberá satisfacer las diversas normas y restricciones establecidas.

A continuación, se muestra el listado de reglas de negocio definidas mediante un identificador y la descripción correspondiente a cada una de ellas.

<b>REGLAS DE NEGOCIO</b>	
<b>RN-01</b>	El usuario registrado e identificado tendrá la posibilidad tanto de visualizar todas las publicaciones y productos que desee, como de publicar los productos o servicios que desee, por lo que puede actuar a su vez como vendedor y como cliente.
<b>RN-02</b>	El vendedor y el cliente podrán realizar uso del sistema de mensajería instantánea de manera bidireccional.
<b>RN-03</b>	El vendedor podrá realizar tantas publicaciones como quiera, sin necesidad de haber marcado como ventas publicaciones anteriores
<b>RN-04</b>	El vendedor podrá eliminar la publicación cuando así lo desee.
<b>RN-05</b>	La aplicación estará sujeta a 15/1999 Ley de Protección de Datos de Carácter Personal, que garantice la confidencialidad de los datos, por lo que no está permitido su uso por terceros.
<b>RN-06</b>	El cliente podrá realizar una valoración sobre el vendedor una vez haya concluido la transacción del artículo.
<b>RN-07</b>	Tanto cliente como vendedor, podrán compartir su ubicación actual e incluso compartir su ubicación en tiempo real de su dispositivo móvil.
<b>RN-08</b>	En caso de que el precio de una publicación sea de 0, se tomará por defecto que la publicación se considera una donación.
<b>RN-09</b>	No pueden existir dos usuarios diferentes con el mismo email.
<b>RN-10</b>	Será obligatorio, indicar el estado del producto sujeto a la publicación realizada.

*Tabla 25: Reglas de Negocio*

### 4.3 OBJETIVOS DEL SISTEMA

<b>OBJ - 01 GESTIÓN DE LOS USUARIOS</b>	
<b>DESCRIPCIÓN</b>	El sistema deberá gestionar toda la información relativa a los usuarios ya sean vendedores, clientes (Registro, proceso de autenticación, modificación datos perfil, Modificar tema )
<b>IMPORTANCIA</b>	Alta

*Tabla 26: Objetivos del sistema - Gestión de los usuarios*

<b>OBJ - 02 GESTIÓN DE PUBLICACIÓN DE PUBLICACIONES</b>	
<b>DESCRIPCIÓN</b>	El sistema deberá gestionar toda la información relativa a las publicaciones: comprende todo el proceso de publicación por el que transcurre el vendedor (selección de categorías, selección de imágenes, añadir precio, añadir ubicación, añadir estado del producto, etc... ).
<b>IMPORTANCIA</b>	Alta

*Tabla 27: Objetivos del sistema - Gestión publicación de artículos*



podrán realizar en este apartado funcionalidades de configuración como visualización de transacciones del usuario, modificación del tema a oscuro/claro, activar/desactivar notificaciones y cerrar la sesión del usuario activo en el sistema.

### **Gestión de publicación de publicaciones**

Esta característica permite realizar todo el proceso de añadir publicaciones por parte de un usuario vendedor al sistema. Se especificará toda la información al detalle del producto o servicio que se desea publicar, podemos especificar desde sus categorías, título o descripción hasta la localización de la publicación del artículo o imágenes del producto.

### **Gestión de sistema de valoración entre usuarios**

Esta característica permite gestionar el sistema de valoración que permite a los usuarios conocer las valoraciones de otros usuarios sobre los vendedores en transacciones anteriores, información importante para el cliente a la hora de confiar en las buenas practicas del usuario vendedor.

### **Gestión de servicio de mensajería instantánea**

Una de las características más importantes es la del servicio de mensajería instantánea, en formato *chat*, donde permite una comunicación abierta e unidireccional entre el usuario vendedor y el cliente. En ella, se implementan además de funcionalidades básicas de un *chat*, funcionalidades avanzadas como la posibilidad de adjuntar archivos de imagen por medio de la aplicación de la cámara del dispositivo, desde la galería del dispositivo o la posibilidad de compartir la ubicación actual. También, se encuentra integrado en él la característica descrita a continuación, sistema de seguimiento en tiempo real.

### **Gestión de sistema de seguimiento en tiempo real**

Se permite a través de esta característica, compartir entre usuarios del mismo chat la ubicación en tiempo real entre ellos mediante la herramienta de servicio de mensajería instantánea. Se permite introducir una ubicación de destino a partir de una ubicación inicial, posición en la que se encuentra el dispositivo móvil en ese momento. Se incluyen varias funcionalidades que permiten interactuar a los dos usuarios a través del mismo mapa, así como la posibilidad de cancelar este servicio cuando el usuario emisor lo crea conveniente.

### **Gestión de publicaciones**

En este apartado, se engloban todas las funcionalidades relacionadas con publicaciones ya realizadas. Se permite la gestión del listado general de publicaciones del inicio de la aplicación, como buscar la publicación que desea el cliente, realizar filtrados por distintas variables para facilitar encontrar lo que se desea, realizar búsquedas concretas mediante una barra de búsqueda, actualizar el listado de productos publicados por otros usuarios. También, se puede gestionar una publicación, tanto modificar la información, visualizar su información, marcar una publicación como vendida, eliminar una publicación, etc. Por último, se incluye la gestión de los distintos listados de las publicaciones de un usuario donde estarán clasificados por publicaciones guardadas como favoritas, publicaciones marcadas como vendidas y publicaciones ya realizadas.

## 4.5 REQUISITOS DE USUARIO

Tras presentar los actores que tendrán interacción con el sistema, se procede a realizar un listado con cada uno de los requisitos de usuario asociados a los mismos. Estos requisitos de usuarios son necesarios para poder captar las tareas que realizan los usuarios con el sistema. Se puede considerar que se engloban la totalidad de las funcionalidades.

USUARIO NO REGISTRADO	
RU-01	El usuario no registrado podrá consultar la información general de la aplicación.
RU-02	El usuario no registrado podrá registrarse con su usuario de Google.
RU-03	El usuario no registrado podrá registrarse en el sistema.

Tabla 31: Requisitos de usuario - Usuario no registrado

USUARIO REGISTRADO Y NO IDENTIFICADO	
RU-04	El usuario registrado y no identificado podrá autenticarse en la aplicación.
RU-05	El usuario registrado y no identificado podrá iniciar sesión con Google.

Tabla 32: Requisitos de usuario – Usuario registrado y no identificado

USUARIO REGISTRADO E IDENTIFICADO	
RU-06	El usuario registrado e identificado podrá visualizar su perfil en la aplicación.
RU-07	El usuario registrado e identificado podrá modificar su avatar del perfil de usuario.
RU-08	El usuario registrado e identificado podrá acceder a la configuración de su perfil.
RU-09	El usuario registrado e identificado podrá seleccionar el tema oscuro.
RU-10	El usuario registrado e identificado podrá activar/desactivar las notificaciones recibidas.
RU-11	El usuario registrado e identificado podrá acceder a las transacciones relacionadas con su perfil.
RU-12	El usuario registrado e identificado podrá eliminar su cuenta de usuario de la aplicación.
RU-13	El usuario registrado e identificado podrá cerrar sesión en la aplicación.

Tabla 33: Requisitos de usuario – Usuario registrado e identificado

<b>USUARIO VENDEDOR</b>	
<b>GESTIÓN DE PUBLICACIÓN DE ARTÍCULOS</b>	
<b>RU-14</b>	El vendedor podrá realizar una publicación de un producto o servicio.
<b>RU-15</b>	El vendedor podrá visualizar el listado de categorías.
<b>RU-16</b>	El vendedor podrá seleccionar una categoría asociada.
<b>RU-17</b>	El vendedor podrá modificar la categoría seleccionada previamente.
<b>RU-18</b>	El vendedor podrá visualizar el listado de subcategorías.
<b>RU-19</b>	El vendedor podrá seleccionar una subcategoría asociada.
<b>RU-20</b>	El vendedor podrá modificar una subcategoría asociada.
<b>RU-21</b>	El vendedor podrá añadir un título a la publicación.
<b>RU-22</b>	El vendedor podrá añadir una descripción a la publicación.
<b>RU-23</b>	El vendedor podrá activar/desactivar posibilidad de realizar envío.
<b>RU-24</b>	El vendedor podrá añadir el estado del producto.
<b>RU-25</b>	El vendedor podrá añadir imágenes desde la galería del dispositivo.
<b>RU-26</b>	El vendedor podrá añadir imágenes del producto desde la cámara del dispositivo.
<b>RU-27</b>	El vendedor podrá añadir el precio de la publicación.
<b>RU-28</b>	El vendedor podrá activar/desactivar la publicación como un artículo en donación.
<b>RU-29</b>	El vendedor podrá seleccionar la ubicación actual.
<b>RU-30</b>	El vendedor podrá seleccionar una ubicación distinta a la actual.
<b>RU-31</b>	El vendedor podrá visualizar el mapa con la ubicación escogida.
<b>RU-32</b>	El vendedor podrá establecer un marcador con una nueva ubicación.
<b>RU-33</b>	El vendedor podrá confirmar la nueva ubicación al establecer un marcador en el mapa.
<b>RU-34</b>	El vendedor podrá confirmar la información de la publicación.
<b>RU-35</b>	El vendedor podrá cancelar el proceso de publicación.
<b>GESTIÓN DE PUBLICACIONES</b>	
<b>RU-36</b>	El vendedor podrá visualizar el listado completo de publicaciones de la plataforma.
<b>RU-37</b>	El vendedor podrá visualizar la información de una publicación.
<b>RU-38</b>	El vendedor podrá eliminar una publicación.
<b>RU-39</b>	El vendedor podrá confirmar que desea eliminar una publicación.
<b>RU-40</b>	El vendedor podrá marcar una publicación como vendida.
<b>RU-41</b>	El vendedor podrá confirmar que desea marcar una publicación como vendida.
<b>RU-42</b>	El vendedor podrá acceder al mapa con la ubicación de la publicación realizada.
<b>RU-43</b>	El vendedor podrá visualizar el formulario para modificar la información de una publicación.
<b>RU-44</b>	El vendedor podrá modificar el título de una publicación.
<b>RU-45</b>	El vendedor podrá modificar la descripción de una publicación.
<b>RU-46</b>	El vendedor podrá modificar la categoría de una publicación.
<b>RU-47</b>	El vendedor podrá modificar la subcategoría de una publicación.

<b>RU-48</b>	El vendedor podrá modificar el estado de una publicación.
<b>RU-49</b>	El vendedor podrá modificar el envío de una publicación.
<b>RU-50</b>	El vendedor podrá modificar el precio de una publicación.
<b>RU-51</b>	El vendedor podrá acceder a su propio sistema de valoración por los usuarios clientes.
<b>RU-52</b>	El vendedor podrá visualizar sus propias publicaciones en venta.

Tabla 34: Requisitos de usuario - Usuario vendedor

<b>USUARIO CLIENTE</b>	
<b>GESTIÓN DE PUBLICACIONES</b>	
<b>RU-53</b>	El cliente podrá visualizar el listado completo de publicaciones de la plataforma.
<b>RU-54</b>	El cliente podrá visualizar la información de una publicación.
<b>RU-55</b>	El cliente podrá acceder al mapa con la ubicación de la publicación realizada.
<b>RU-56</b>	El cliente podrá seleccionar botón para redirigir a la ubicación actual.
<b>RU-57</b>	El cliente podrá visualizar imágenes de una publicación.
<b>GESTIÓN DE SISTEMA DE VALORACIÓN ENTRE USUARIOS</b>	
<b>RU-58</b>	El cliente podrá acceder al sistema de valoración de un vendedor.
<b>RU-59</b>	El cliente podrá visualizar la información de un vendedor.
<b>RU-60</b>	El cliente podrá visualizar el listado de valoraciones que pertenece al usuario vendedor propietario de la publicación.
<b>RU-61</b>	El cliente podrá visualizar una valoración de otro cliente o una propia valoración, realizada al vendedor propietario de la publicación
<b>RU-62</b>	El cliente podrá añadir una valoración al vendedor propietario de la publicación.
<b>RU-63</b>	El cliente podrá modificar el comentario de una valoración añadida previamente al vendedor propietario de la publicación.
<b>RU-64</b>	El cliente podrá modificar la puntuación de una valoración añadida previamente al vendedor propietario de la publicación.
<b>RU-65</b>	El cliente podrá cancelar una valoración mientras está realizando el proceso de añadirla.
<b>RU-66</b>	El cliente podrá eliminar una valoración realizada previamente por sí mismo.
<b>RU-67</b>	El cliente podrá visualizar el listado de publicaciones en venta del usuario vendedor propietario de la publicación.
<b>GESTIÓN DE SERVICIO DE MENSAJERÍA INSTANTÁNEA</b>	
<b>RU-68</b>	El cliente podrá realizar la apertura del servicio de mensajería instantánea para contactar con el vendedor propietario de la publicación.
<b>RU-69</b>	El cliente podrá acceder al historial de chats en curso actualmente con otros usuarios vendedores.

<b>RU-70</b>	El cliente podrá visualizar el historial de chats que ha abierto con usuarios vendedores.
<b>RU-71</b>	El cliente podrá seleccionar un chat iniciado previamente.
<b>RU-72</b>	El cliente podrá visualizar el listado de mensajes del chat.
<b>RU-73</b>	El cliente podrá escribir un mensaje a través del cuadro de dialogo del chat.
<b>RU-74</b>	El cliente podrá ampliar el selector de emoticonos.
<b>RU-75</b>	El cliente podrá seleccionar un emoticono.
<b>RU-76</b>	El cliente podrá compartir un archivo con el usuario vendedor.
<b>RU-77</b>	El cliente podrá acceder a la cámara del dispositivo.
<b>RU-78</b>	El cliente podrá modificar la configuración de la aplicación de la cámara de su dispositivo.
<b>RU-79</b>	El cliente podrá realizar una fotografía a través de la cámara del dispositivo
<b>RU-80</b>	El cliente podrá confirmar la imagen realizada previamente mediante la cámara del dispositivo.
<b>RU-81</b>	El cliente podrá acceder a la aplicación de galería del dispositivo
<b>RU-82</b>	El cliente podrá visualizar listado de álbumes de la aplicación de la galería del dispositivo.
<b>RU-83</b>	El cliente podrá seleccionar una imagen de la galería del dispositivo.
<b>RU-84</b>	El cliente podrá seleccionar una imagen enviada para ampliarla en el dispositivo.
<b>GESTIÓN SISTEMA DE SEGUIMIENTO EN TIEMPO REAL</b>	
<b>RU-85</b>	El cliente podrá acceder al sistema de gestión de seguimiento en tiempo real, a través del botón "Ubicación".
<b>RU-86</b>	El cliente podrá visualizar mapa con ubicación actual.
<b>RU-87</b>	El cliente podrá ampliar el tamaño del mapa.
<b>RU-88</b>	El cliente podrá reducir el tamaño del mapa.
<b>RU-89</b>	El cliente podrá mostrar la ubicación actual.
<b>RU-90</b>	El cliente podrá rotar el mapa sobre la ubicación actual.
<b>RU-91</b>	El cliente podrá realizar un desplazamiento sobre el mapa.
<b>RU-92</b>	El cliente podrá seleccionar la barra de búsqueda incluida en el mapa.
<b>RU-93</b>	El cliente podrá escribir la ubicación de destino deseada.
<b>RU-94</b>	El cliente podrá realizar una búsqueda de la ubicación de destino.
<b>RU-95</b>	El cliente podrá visualizar el listado de sugerencias de búsqueda.
<b>RU-96</b>	El cliente podrá seleccionar la ubicación de destino deseada.
<b>RU-97</b>	El cliente podrá colocar el marcador con la ubicación de destino manualmente.
<b>RU-98</b>	El cliente podrá confirmar la ubicación de destino seleccionada.
<b>RU-99</b>	El cliente podrá mostrar/no mostrar el recorrido del usuario.
<b>RU-100</b>	El cliente podrá seleccionar el botón que redirija a la ubicación actual del usuario.
<b>RU-101</b>	El cliente podrá compartir ubicación en tiempo real.
<b>RU-102</b>	El cliente podrá iniciar seguimiento de ubicación en tiempo real.
<b>RU-103</b>	El cliente podrá acceder al mapa con la ubicación actual.
<b>RU-104</b>	El cliente podrá visualizar el mapa con el seguimiento en tiempo real.

<b>RU-105</b>	El cliente podrá dejar de compartir la ubicación en tiempo real con el usuario vendedor.
<b>RU-106</b>	El cliente podrá visualizar el listado completo de publicaciones.
<b>RU-107</b>	El cliente podrá activar/desactivar un producto como favorito.
<b>RU-108</b>	El cliente podrá ordenar el listado de publicaciones por proximidad.
<b>RU-109</b>	El cliente podrá ordenar las publicaciones por fecha de publicación más reciente.
<b>RU-110</b>	El cliente podrá realizar filtrado por intervalos de precio.
<b>RU-111</b>	El cliente podrá actualizar el listado completo de publicaciones.
<b>RU-112</b>	El cliente podrá seleccionar la barra de búsqueda.
<b>RU-113</b>	El cliente podrá escribir la publicación que desea buscar.
<b>RU-114</b>	El cliente podrá visualizar el listado coincidente con la búsqueda realizada.
<b>RU-115</b>	El cliente podrá visualizar el listado horizontal completo de categorías disponibles.
<b>RU-116</b>	El cliente podrá seleccionar la categoría deseada.
<b>RU-117</b>	El cliente podrá visualizar el listado coincidente con la categoría seleccionada.
<b>RU-118</b>	El cliente podrá visualizar el listado de publicaciones realizadas por sí mismo.
<b>RU-119</b>	El cliente podrá seleccionar una publicación del listado de publicaciones realizadas por sí mismo
<b>RU-120</b>	El cliente podrá visualizar el listado de publicaciones marcadas como favoritas.
<b>RU-121</b>	El cliente podrá seleccionar una publicación del listado de publicaciones marcadas como favoritas
<b>RU-122</b>	El cliente podrá visualizar el listado de publicaciones marcadas como vendidas por sí mismo.
<b>RU-123</b>	El cliente podrá seleccionar una publicación marcada como vendida.

*Tabla 35: Requisitos de usuario - Usuario cliente*

## 4.6 REQUISITOS FUNCIONALES

Los **requisitos funcionales, identificados como RF-XX**, exponen el comportamiento del sistema cuándo los usuarios interactúan con la propia aplicación. Estos requisitos se pueden obtener por las acciones realizadas durante cada una de las secuencias normales especificadas en el apartado anterior.

Por consiguiente, se mostrará una lista con las actividades que deberá hacer el sistema ante las interacciones de los usuarios

### 4.6.1 GESTIÓN DE USUARIOS

- ❖ **RF-01:** El sistema mostrará un formulario al usuario registrado y no identificado para que pueda autenticarse mediante la introducción de sus credenciales, email y contraseña.
- ❖ **RF-02:** El sistema validará las credenciales comprobando que existe un usuario registrado con ese email y contraseña.

- ❖ **RF-03:** El sistema notificará que no existe un usuario con esas credenciales, o indicará que son inválidas, y lo redirigirá a la página de registro.
- ❖ **RF-04:** El sistema mostrará un mensaje indicado que el usuario no se encuentra almacenado en la base datos.
- ❖ **RF-05:** El sistema mostrará un formulario con los campos de registro al usuario no registrado.
- ❖ **RF-06:** El sistema comprobará que todos los campos requeridos están cumplimentados y cumplen las condiciones mínimas requeridas.
- ❖ **RF-07:** El sistema mostrará mensajes de error en caso de no introducir todos los campos en el formulario o introducir datos erróneos.
- ❖ **RF-08:** El sistema comprobará que no existe un usuario registrado con el mismo email que se ha introducido en el formulario.
- ❖ **RF-09:** El sistema almacenará un nuevo usuario en la base de datos.
- ❖ **RF-10:** El sistema mostrará un mensaje indicando que el registro se ha realizado correctamente.
- ❖ **RF-11:** El sistema mostrará un mensaje indicando que el registro no se ha realizado, correctamente, por lo que el usuario no autenticado deberá revisar los campos del formulario.
- ❖ **RF-12:** El sistema mostrará la opción de iniciar sesión con la cuenta de usuario Google.
- ❖ **RF-13:** El sistema mostrará una ventana emergente para seleccionar la cuenta de Google con la que el usuario se desea identificar.
- ❖ **RF-14:** El sistema creará una nueva sesión para el usuario que se autentica.
- ❖ **RF-15:** El sistema mostrará la información del perfil del usuario que se ha autenticado previamente.
- ❖ **RF-16:** El sistema mostrará la opción de modificar el avatar del usuario, mediante una ventana emergente, donde se dan las opciones de seleccionar imagen de galería, realizar fotografía o eliminar el avatar existente.
- ❖ **RF-17:** El sistema mostrará un mensaje solicitando la confirmación de que el usuario desea realmente darse de baja al seleccionar el botón de Eliminar Cuenta.
- ❖ **RF-18:** El sistema eliminará el de la base de datos el usuario que desea darse de baja de la plataforma.
- ❖ **RF-19:** El sistema mostrará un mensaje indicando que se ha eliminado el usuario de forma correcta.
- ❖ **RF-20:** El sistema eliminará la sesión activa y redirigirá al usuario a la pantalla de Inicio de Sesión.
- ❖ **RF-21:** El sistema ofrecerá la opción de modificar el tema de la aplicación a “Modo oscuro”, teniendo opción a retornar a “Modo claro” si así se desea.
- ❖ **RF-22:** El sistema mostrará la opción de activar/desactivar las notificaciones push de la aplicación para el dispositivo.
- ❖ **RF-23:** El sistema mostrará la opción de acceder a las transacciones del usuario autenticado.
- ❖ **RF-24:** El sistema mostrará un botón para cerrar sesión.
- ❖ **RF-25:** El sistema mostrará un mensaje solicitando la confirmación de que el usuario desea realmente cerrar sesión del usuario autenticado, al seleccionar el botón de Cerrar sesión.
- ❖ **RF-26:** El sistema eliminará la sesión y destruirá la información guardada de la sesión actual.

## 4.6.2 GESTIÓN DE PUBLICACIÓN DE ARTÍCULOS

- ❖ **RF-27:** El sistema mostrará un interfaz con un listado con las categorías disponibles en la plataforma.
- ❖ **RF-28:** El sistema mostrará un interfaz con un listado con las subcategorías disponibles en la plataforma.
- ❖ **RF-29:** El sistema mostrará un formulario donde se podrán especificar la información detallada del producto de la publicación.
- ❖ **RF-30:** El sistema mostrará un botón para continuar a la siguiente pantalla para añadir una publicación.
- ❖ **RF-31:** El sistema mostrará la opción de añadir una imagen de la publicación a través de la cámara del dispositivo
- ❖ **RF-32:** El sistema mostrará la opción de añadir una imagen de la publicación a través de la galería del dispositivo
- ❖ **RF-33:** El sistema mostrará un formulario para añadir el precio de la publicación.
- ❖ **RF-34:** El sistema mostrará un botón para indicar si la publicación es una donación.
- ❖ **RF-35:** El sistema mostrará un mapa para indicar la ubicación de la publicación que se encuentra en el proceso de ser añadida.
- ❖ **RF-36:** El sistema mostrará mensajes de error en casa de que alguno de los parámetros incluidos en el formulario sean incorrectos o no sean válidos.
- ❖ **RF-37:** El sistema mostrará un mensaje de confirmación para solicitar el usuario que desea publicar la información añadida previamente.
- ❖ **RF-38:** El sistema permitirá cancelar el proceso de publicación.
- ❖ **RF-39:** El sistema destruirá toda la información añadida previamente por el usuario
- ❖ **RF-40:** El sistema añadirá un nuevo producto con la información introducida en la base de datos.
- ❖ **RF-41:** El sistema mostrará un mensaje de confirmación de que se ha añadido la publicación con éxito en la base de datos.

## 4.6.3. GESTIÓN DE PUBLICACIONES

- ❖ **RF-42:** El sistema mostrará una interfaz con un listado completo de las publicaciones añadidas por los usuarios registrado e identificados.
- ❖ **RF-43:** El sistema mostrará un mensaje de información, indicando al usuario que no hay ninguna publicación añadida en la plataforma en el caso de que el listado se encuentre vacío.
- ❖ **RF-44:** El sistema mostrará una interfaz de miniatura para cada publicación dentro del listado, donde aparezca la información más relevante de la publicación
- ❖ **RF-45:** El sistema mostrará un botón para activar/desactivar un producto como favorito.
- ❖ **RF-46:** El sistema añadirá/eliminará la publicación del listado de publicaciones favoritas, modificando el estado en la base de datos.
- ❖ **RF-47:** El sistema mostrará la opción de que el usuario ordene el listado de publicaciones por proximidad a su ubicación actual.
- ❖ **RF-48:** El sistema mostrará un nuevo listado de publicaciones ordenado por proximidad en base la ubicación actual del usuario.
- ❖ **RF-49:** El sistema mostrará la opción de que el usuario ordene el listado de publicaciones por fecha de publicación más reciente.

- ❖ **RF-50:** El sistema mostrará un nuevo listado de publicaciones ordenado por fecha de publicación más reciente.
- ❖ **RF-51:** El sistema mostrará la opción de realizar un filtrado por intervalo de precio en las publicaciones sobre el listado de publicaciones que se muestra en la interfaz.
- ❖ **RF-52:** El sistema mostrará en la interfaz un listado de publicaciones que se encuentren dentro del intervalo de precio seleccionado por el usuario previamente.
- ❖ **RF-53:** El sistema mostrará la opción de realizar una búsqueda, mediante la barra de búsquedas, del título de la publicación que el usuario desea visualizar.
- ❖ **RF-54:** El sistema mostrará una barra de búsqueda, donde el usuario podrá escribir la publicación que desea buscar.
- ❖ **RF-55:** El sistema permitirá la opción de cancelar el proceso de búsqueda mediante la barra de búsqueda.
- ❖ **RF-56:** El sistema mostrará en la interfaz un listado de publicaciones que se encuentren dentro de la coincide seleccionado por el usuario previamente.
- ❖ **RF-57:** El sistema mostrará un listado horizontal de todas las categorías disponibles en la plataforma.
- ❖ **RF-58:** El sistema mostrará al usuario la posibilidad de actualizar el listado completo de publicaciones, deslizando el listado hacia abajo en la pantalla.
- ❖ **RF-59:** El sistema mostrará un listado con las publicaciones coincidentes según la categoría seleccionada por el usuario.
- ❖ **RF-60:** El sistema mostrará un mensaje de información indicando al usuario que no hay ninguna publicación coincidente con la categoría seleccionada.
- ❖ **RF-61:** El sistema mostrará un listado completo de las publicaciones actualmente activas que pertenecen al usuario registrado e identificado que mantiene la sesión activa en la plataforma.
- ❖ **RF-62:** En caso de que el usuario no haya realizado ninguna publicación, el sistema mostrará un mensaje de aviso indicando que el usuario no ha añadido todavía ninguna publicación.
- ❖ **RF-63:** El sistema mostrará un listado completo de las publicaciones actualmente marcadas como favoritos por el usuario registrado e identificado que mantiene la sesión activa en la plataforma.
- ❖ **RF-64:** En caso de que el usuario no haya marcado como favorito ninguna publicación, el sistema mostrará un mensaje de aviso indicando que el usuario no ha añadido todavía ninguna publicación como favorita.
- ❖ **RF-65:** El sistema mostrará un listado completo de las publicaciones actualmente marcadas como vendidas por el usuario registrado e identificado que mantiene la sesión activa en la plataforma.
- ❖ **RF-66:** En caso de que el usuario no haya marcado como vendida ninguna publicación, el sistema mostrará un mensaje de aviso indicando que el usuario no ha añadido todavía ninguna publicación como vendido en la plataforma.
- ❖ **RF-67:** El sistema mostrará toda la información relacionada con una publicación, seleccionada previamente por un usuario desde cualquier interfaz que muestre un listado de publicaciones.
- ❖ **RF-68:** El sistema mostrará el carrusel de imágenes relacionadas con la publicación realizada por el usuario vendedor previamente.
- ❖ **RF-69:** El sistema mostrará al usuario vendedor la posibilidad de eliminar la publicación que se muestra en pantalla.

- ❖ **RF-70:** El sistema mostrará un mensaje de confirmación indicando al usuario vendedor si desea realmente eliminar la publicación de la interfaz actual.
- ❖ **RF-71:** El sistema eliminará toda la información relacionada con la publicación en la base de datos.
- ❖ **RF-72:** El sistema mostrará al usuario vendedor la posibilidad de indicar que la publicación que se muestra en pantalla ha sido vendida.
- ❖ **RF-73:** El sistema mostrará un mensaje de confirmación indicando al usuario vendedor si desea realmente marcar como vendido la publicación de la interfaz actual.
- ❖ **RF-74:** El sistema modificará el estado de la publicación como vendido en la base de datos.
- ❖ **RF-75:** El sistema mostrará al usuario vendedor la posibilidad de modificar la información de la publicación que se muestra en pantalla.
- ❖ **RF-76:** El sistema mostrará al usuario una nueva interfaz con un formulario donde se permitirá modificar la información de la publicación actual.
- ❖ **RF-77:** El sistema mostrará al usuario un mensaje de confirmación indicando al usuario vendedor si desea realmente modificar la información editada.
- ❖ **RF-78:** El sistema modificará la información editada por el usuario vendedor en la base de datos.
- ❖ **RF-79:** El sistema mostrará la categoría y subcategoría a la que corresponde la publicación.
- ❖ **RF-93:** El sistema mostrará un mapa con la ubicación asociada a la publicación junto con la dirección.
- ❖ **RF-95:** El sistema ampliará el mapa a pantalla completa, mostrando mediante la API de Google Maps y resaltado mediante un marcador la ubicación actual de la publicación.
- ❖ **RF-96:** El sistema permitirá redirigir a la ubicación actual de la publicación al desplazarse por el mapa y perder el marcador.
- ❖ **RF-97:** El sistema permitirá cerrar a pantalla completa el mapa.
- ❖ **RF-98:** El sistema mostrará un botón que permita abrir una conversación con el usuario vendedor, mediante el servicio de mensajería instantánea, o redirigir a la conversación ya activa previamente.

#### 4.6.4. GESTIÓN DE SISTEMA DE VALORACIÓN DE USUARIOS

- ❖ **RF-80:** El sistema mostrará la información del usuario vendedor, junto con su media de valoración tanto numérica como en categoría por número de estrellas.
- ❖ **RF-81:** El sistema mostrará una nueva interfaz con la información completa del usuario vendedor de la publicación.
- ❖ **RF-82:** El sistema mostrará en una pestaña las publicaciones en venta que mantiene el usuario vendedor en la plataforma actualmente.
- ❖ **RF-83:** El sistema mostrará en una pestaña las valoraciones que ha recibido el usuario vendedor por el resto de usuarios de la plataforma.
- ❖ **RF-84:** El sistema mostrará un botón que permita añadir una nueva valoración al usuario vendedor.
- ❖ **RF-85:** El sistema mostrará una ventana emergente dónde se podrá establecer la valoración numérica, mediante un intervalo de valoración de estrellas, y un comentario de la valoración para el usuario vendedor.
- ❖ **RF-86:** El sistema añadirá la información de una nueva publicación en la base de datos.
- ❖ **RF-87:** El sistema mostrará al usuario propietario de la publicación la posibilidad de editar toda la información añadida.

- ❖ **RF-88:** El sistema mostrará al usuario un mensaje de confirmación indicando al usuario que se ha modificado el contenido de la valoración.
- ❖ **RF-89:** El sistema actualizará la información de la publicación en la base de datos.
- ❖ **RF-90:** El sistema mostrará al usuario propietario de la publicación la posibilidad de eliminarla del sistema.
- ❖ **RF-91:** El sistema mostrará al usuario un mensaje de confirmación indicando al usuario que si desea realmente eliminar por completo la valoración.
- ❖ **RF-92:** El sistema eliminará por completo la valoración asociada en la base de datos.
- ❖ **RF-93:** El sistema dará opción a cancelar una valoración durante el proceso de publicación.

#### 4.6.5 GESTIÓN DE SISTEMA DE MENSAJERÍA INSTANTÁNEA

- ❖ **RF-99:** El sistema mostrará un contador mediante sistema de notificaciones emergente de la cantidad de mensajes no leídos, del global de todos los chats activos, por el usuario autenticado e identificado.
- ❖ **RF-100:** El sistema mostrará un listado completo con todas las conversaciones activas del usuario registrado e identificado.
- ❖ **RF-101:** El sistema mostrará de manera individual la información de cada conversación activa, se mostrará el avatar del usuario con el que se mantiene abierta la conversación, el último mensaje recibido, hace cuánto se envió el último mensaje, y el número de mensajes no leídos que pertenece a esa conversación.
- ❖ **RF-102:** El sistema permitirá seleccionar una conversación, para abrir una nueva interfaz con la conversación seleccionada.
- ❖ **RF-103:** El sistema mostrará un listado con todos los mensajes del chat enviados previamente.
- ❖ **RF-104:** El sistema mostrará el estado conectado/desconectado del usuario con el que se mantiene la conversación, junto con el nombre y su avatar de perfil de usuario.
- ❖ **RF-105:** El sistema mostrará un cuadro de dialogo donde se permitirá escribir un mensaje.
- ❖ **RF-106:** El sistema mostrará una ventana emergente con un amplio listado de emoticonos, al seleccionar el botón correspondiente.
- ❖ **RF-107:** El sistema permitirá seleccionar un emoticono y añadirlo al cuadro de dialogo.
- ❖ **RF-108:** El sistema permitirá enviar el mensaje deseado, escrito mediante el teclado del dispositivo, al seleccionar el botón de confirmación de envío.
- ❖ **RF-109:** El sistema almacenará cada mensaje enviado en la base de datos.
- ❖ **RF-110:** El sistema permitirá adjuntar archivos multimedia al seleccionar el botón correspondiente, a través de una nueva ventana emergente.
- ❖ **RF-111:** El sistema permitirá acceder y activar a la aplicación de la cámara del dispositivo actual.
- ❖ **RF-112:** El sistema permitirá modificar las opciones de configuración de la cámara del dispositivo.
- ❖ **RF-113:** El sistema permitirá realizar una fotografía mediante la aplicación de la cámara del dispositivo.
- ❖ **RF-114:** El sistema solicitará la confirmación de que desea adjuntar la imagen capturada previamente.
- ❖ **RF-115:** El sistema enviará la imagen confirmada mediante una nueva ventana de mensaje.
- ❖ **RF-116:** El sistema almacenará la imagen compartida desde la cámara en la base de datos.

- ❖ **RF-117:** El sistema mostrará un sistema de confirmación de lectura del mensaje por parte del usuario receptor.
- ❖ **RF-118:** El sistema permitirá acceder a la galería con las imágenes almacenadas en el dispositivo del usuario registrado e autenticado.
- ❖ **RF-119:** El sistema permitirá acceder a los distintos álbumes de fotografías del dispositivo.
- ❖ **RF-120:** El sistema permitirá seleccionar cualquier imagen de la aplicación de Galería del dispositivo.
- ❖ **RF-121:** El sistema enviará la imagen seleccionada mediante una nueva ventana de mensaje.
- ❖ **RF-122:** El sistema almacenará la imagen compartida desde galería en la base de datos.
- ❖ **RF-123:** El sistema permitirá seleccionar una imagen enviada previamente mediante el chat para ampliarla a pantalla completa en el dispositivo.
- ❖ **RF-124:** El sistema permitirá ampliar/reducir una imagen a vista de pantalla completa en el dispositivo.
- ❖ **RF-125:** El sistema permitirá acceder al mapa dónde se encuentra de forma ampliada un sistema de seguimiento en tiempo real, para compartir con el usuario con el que se mantendrá la conversación activa.

#### **24.6.6 GESTIÓN DE SISTEMA DE SEGUIMIENTO EN TIEMPO REAL**

- ❖ **RF-126:** El sistema permitirá visualizar el mapa, donde se muestra la ubicación actual del usuario identificado e autenticado en la plataforma, proporcionado por la API de Google Maps y API de MapBox.
- ❖ **RF-127:** El sistema permitirá ampliar/reducir el tamaño del mapa.
- ❖ **RF-128:** El sistema mostrará la ubicación actual del dispositivo.
- ❖ **RF-129:** El sistema permitirá rotar el mapa, independientemente de la ubicación en la que esté el dispositivo.
- ❖ **RF-130:** El sistema permitirá realizar un desplazamiento sobre el mapa.
- ❖ **RF-131:** El sistema mostrará sobre el mapa, una barra de búsqueda que permita indicar la ubicación de destino.
- ❖ **RF-132:** El sistema permitirá acceder a la barra de búsqueda de la ubicación de destino.
- ❖ **RF-133:** El sistema permitirá al usuario escribir mediante el teclado del dispositivo la ubicación de destino que desea encontrar.
- ❖ **RF-134:** El sistema mostrará un listado de ubicaciones de sugerencias de búsqueda.
- ❖ **RF-135:** El sistema permitirá seleccionar la ubicación de destino deseada.
- ❖ **RF-136:** El sistema permitirá seleccionar la ubicación de destino deseada colocando un marcador de forma manual en el mapa.
- ❖ **RF-137:** El sistema almacenará las coordenadas tanto de la ubicación inicial como de la ubicación de destino seleccionadas.
- ❖ **RF-138:** El sistema mostrará un mensaje informativo mientras calcula la ruta seleccionada.
- ❖ **RF-139:** El sistema trazará una ruta sobre el mapa, desde la ubicación actual hasta la ubicación de destino escogida.
- ❖ **RF-140:** El sistema mostrará información asociada a la ruta creada, como la ubicación inicial, la ubicación de destino, el tiempo transcurrido hasta completar la ruta y la distancia que debemos recorrer en la ruta creada.
- ❖ **RF-141:** El sistema mostrará un menú radial con distintas opciones para el usuario, que permite interactuar con el usuario receptor.

- ❖ **RF-142:** El sistema permitirá mostrar/no mostrar el recorrido por la ruta del usuario.
- ❖ **RF-143:** El sistema permitirá seleccionar el botón que redirigirá a la ubicación actual del usuario.
- ❖ **RF-144:** El sistema permitirá iniciar el sistema de seguimiento en tiempo real a través de la ubicación del dispositivo.
- ❖ **RF-145:** El sistema permitirá al usuario compartir la información del sistema de seguimiento al otro usuario con el que mantendrá activa la conversación.
- ❖ **RF-146:** En el caso de que el usuario comparta la ubicación con el otro usuario sin haber indicado una ubicación de destino, se indicará mediante un mensaje de advertencia, indicando que es obligatorio introducir un destino antes de compartir una ubicación.
- ❖ **RF-147:** El sistema notificará al usuario que se está compartiendo la ubicación actual con el otro usuario de la conversación.
- ❖ **RF-148:** El sistema mostrará una miniatura del mapa en el historial de mensajes dentro de la conversación activa.
- ❖ **RF-149:** El sistema permitirá acceder al mapa a pantalla completa para realizar el seguimiento por el mapa de la ubicación en tiempo real del otro usuario.
- ❖ **RF-150:** El sistema actualizará las coordenadas de la ubicación actual del usuario que está compartiendo su ubicación cada dos segundos en la base de datos.
- ❖ **RF-151:** El sistema permitirá al usuario que está compartiendo la ubicación actualmente, dejar de compartir su ubicación cuándo lo desee con el otro usuario.
- ❖ **RF-152:** El sistema mostrara un mensaje de confirmación al usuario para que indique si desea dejar de compartir su ubicación en tiempo real.
- ❖ **RF-153:** El sistema mostrará un mensaje informativo indicando que se ha dejado de compartir la ubicación actual.

## 4.7 REQUISITOS NO FUNCIONALES

Los **requisitos no funcionales (RNF-XX)** engloban todas aquellas características y condiciones establecidas por consenso entre el cliente y el analista asociado al proyecto, para que la aplicación alcance el objetivo deseado. Estos requisitos se determinan por distintas tipologías, donde podemos encontrar los requisitos de interfaz externa, descripciones de las conexiones con otros sistemas; o atributos de calidad, propiedades del rendimiento del sistema.

### 4.7.1 DISPONIBILIDAD

- ❖ **RNF-01:** La aplicación deberá estar disponible los 7 días de la semana, las 24h horas del día y los 365 días del año, salvo por tareas de despliegue o mantenimiento ocasional, que limitará la aplicación a ser posible en horario nocturno.

### 4.7.2. ACCESIBILIDAD

- ❖ **RNF-02:** La aplicación deberá ser accesible desde cualquier sistema operativo móvil (Android, iOS, Windows Phone, etc.) a través de cualquier dispositivo móvil.
- ❖ **RNF-03:** El sistema cumplirá con los estándares de accesibilidad nivel triple-A de W3C<sup>1</sup>

---

<sup>1</sup> W3C ( World Wide Web Consortium ) <https://www.w3.org/WAI/WCAG1AAA-Conformance.html.es>

- ❖ **RNF-04:** La aplicación móvil será adaptable a los diferentes dispositivos (resoluciones altas o bajas, tamaños de pantalla del dispositivo, dispositivos con bajo rendimiento).

### 4.7.3 ESCALABILIDAD

- ❖ **RNF-05:** La aplicación debe ser desarrollada de forma escalable, lo que permitirá incorporar nuevas funcionales a la aplicación sin interferir en el comportamiento de las funcionalidades actuales.

### 4.7.4. USABILIDAD

- ❖ **RNF-06:** La aplicación deberá ser fácil de utilizar e intuitiva para todo tipo de usuarios, sin importar la edad o rango de familiarización con dispositivos móviles.
- ❖ **RNF-07:** La aplicación deberá ser *responsive*, para los diferentes dispositivos móviles con el objetivo de adaptarse al tamaño y resoluciones de sus pantallas.
- ❖ **RNF-08:** La aplicación deberá validar los datos con mensajes de error. En caso de errores producidos por el usuario, el sistema mostrará por pantalla la información errónea o la falta de información no introducida por el usuario.
- ❖ **RNF-09:** La aplicación deberá tener mensajes informativos que permitan al usuario tener la información sobre alguna de las acciones que realiza.
- ❖ **RNF-10:** La plataforma debe ser capaz de utilizarse con la ayuda del manual de usuario proporcionado.
- ❖ **RNF-11:** El sistema cumplirá con el estándar de usabilidad ISO 9241<sup>1</sup>.

### 4.7.5. SEGURIDAD

- ❖ **RNF-12:** Los usuarios se autenticarán en el sistema a través de su email y contraseña.
- ❖ **RNF-13:** La plataforma deberá validar los campos que ingresan desde el teclado del dispositivo, para comprobar que se cumplen con el mínimo de condiciones pertinentes establecidas.
- ❖ **RNF-14:** Las contraseñas deben tener entre 8 y 16 caracteres alfanuméricos, mezclando minúsculas y mayúsculas.
- ❖ **RNF-15:** Las contraseñas deberán estar cifradas por el algoritmo de cifrado “SHA-256” para almacenar las contraseñas cifradas en la base de datos.
- ❖ **RNF-16:** La aplicación solicitará permisos para acceder a distintas funcionalidades del dispositivo físico como ubicación, contactos, acceso a la cámara, acceso a la galería, etc. Debido a su obligatoriedad de dispositivos desarrollados a partir del sistema operativo Android 6.
- ❖ **RNF-17:** Los usuarios no autenticados solo podrán visualizar la parte pública del sistema.
- ❖ **RNF-18:** La aplicación estará sujeta a 15/1999 Ley de Protección de Datos de Carácter Personal, que garantice la confidencialidad de los datos, por lo que no está permitido su uso por terceros.

---

<sup>1</sup> Norma ISO 9241 ( <https://blog.ida.cl/estrategia-digital/iso-92412010-estandar-internacional-industria-digital/> )

- ❖ **RNF-19:** Al cerrar la sesión del usuario, el sistema eliminará los datos del token de la sesión actual.

#### 4.7.6 RESTRICCIONES

- ❖ **RNF-20:** El sistema debe validar que únicamente exista una cuenta con un email único.
- ❖ **RNF-21:** El sistema debe validar que la información introducida por el usuario en los formularios cumple con los requisitos establecidos.
- ❖ **RNF-22:** El sistema debe validar que hay implementaciones o acciones que solo podrá realizar el usuario vendedor sobre sus propias publicaciones.
- ❖ **RNF-23:** La aplicación solo aceptará archivos JPG o PNG para los avatares de los perfiles de usuarios.
- ❖ **RNF-24:** Únicamente se podrá acceder a la aplicación, si se ha autenticado en la aplicación previamente.
- ❖ **RNF-25:** La aplicación solo aceptará archivos JPG o PNG para los avatares de los usuarios y las radiografías de los pacientes.
- ❖ **RNF-26:** La información relacionada con la utilización de la ubicación del dispositivo móvil del usuario deberá ser totalmente confidencial, se prohibirá su uso por terceros.

#### 4.7.7. ATRIBUTOS DE CALIDAD

- ❖ **RNF-27:** El sistema tendrá como idioma, el castellano.
- ❖ **RNF-28:** El sistema utilizará UTF-8 para el formato de caracteres.
- ❖ **RNF-29:** La aplicación integrará la unidad monetaria, euros (€), en todas sus transacciones.

#### 4.7.8 MANTENIBILIDAD

- ❖ **RNF-30:** Ante la aparición de un error, deberá ser solventado en el menor tiempo posible sin interrumpir el servicio.
- ❖ **RNF-31:** En caso de que sea necesario paralizar el servicio, la plataforma deberá ser capaz de estar operativa en el menor tiempo posible.

#### 4.7.9. FIABILIDAD

- ❖ **RNF-32:** Por la complejidad de algunas de las funcionalidades implementadas, el sistema debe ser fiable al realizar distintas funciones.

#### 4.7.10 RENDIMIENTO

- ❖ **RNF-33:** El sistema deberá responder en un tiempo óptimo.

#### 4.7.11 ROBUSTEZ

- ❖ **RNF-34:** La aplicación mostrará mensajes de advertencia, en caso de detección de fallos en la validación por parte del front-end, para así evitar enviar información errónea a la parte del back-end.
- ❖ **RNF-35:** La aplicación mostrará mensajes de aviso, en caso de detectar fallos en el back-end.

- ❖ **RNF-36:** El sistema se deberá mantener activo ante situaciones de caída del servicio.
- ❖ **RNF-37:** El sistema podrá mantenerse estable en situaciones inusuales (corrupción de datos, excesivo tráfico de datos, etc.).

#### 4.7.12 INTEGRIDAD

- ❖ **RNF-38:** Es de gran relevancia que el sistema tenga en cuenta una mayor integridad debido a que una inconsistencia en el sistema puede conducir a riesgos de vulnerabilidad en los datos almacenados.

### 4.8 DIAGRAMA CASOS DE USO

En este apartado, se mostrará la interacción de los actores con el sistema ante un evento que el usuario quiere iniciar mediante los diagramas de caso de uso. Se agrupan los distintos diagramas de casos de uso, para ello se ha estudiado la mejor forma de agrupar los distintos casos de uso, por lo que la decisión ha sido agruparlo mediante las características que disponemos que han sido descritas anteriormente, tanto para facilitar su comprensión ante la magnitud de los diagramas como para mostrarlos con mayor claridad. A su vez, es necesario tener presente la jerarquía de actores descrita en la *Figura 15: Actores del sistema*.

En resumen, se implementarán los diagramas de casos de uso para representar las acciones que podrán realizar los diferentes tipos de actores, de esta forma se podrá ilustrar todos los requisitos del sistema y también como interaccionan los distintos usuarios con los mismos.

En este tipo de diagramas intervienen conceptos como:

- **Actor:** representa a la entidad externa que representa el usuario externo que tiene alguna relación con el sistema y del que se demanda cierta funcionalidad. Cada uno de los diferentes actores externos se relacionan con los diferentes casos de uso que atañen al sistema.
- **Sistema:** engloba todo el sistema e incluye todos los casos de uso y relaciones que tiene el sistema.
- **Requisitos de usuario:** forman los diferentes requisitos de usuario que especifican en el sistema. Se tratará de cada uno de los casos de uso que se desarrollarán en el futuro. También, servicio que se desea que realice el sistema.

Además, las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un actor se comunica con un requisito.
- Un requisito se comunica con uno o varios requisitos.
- Un requisito extiende otro requisito.
- Un requisito incluye otro requisito.
- **Extend:** indican opcionalidad permitiendo extender la funcionalidad.
- **Include:** indican obligación, el elemento al que señalan debe ser incluido obligatoriamente.
- **Asociación:** relacionan un requisito con el usuario al que se le permite invocar el requisito.

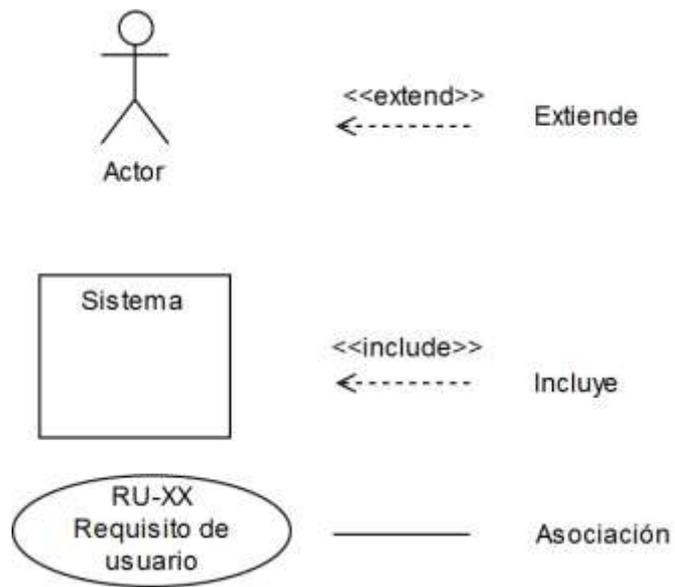


Figura 16: Información Diagramas Casos de uso

## 4.8.1. GESTIÓN DE USUARIOS

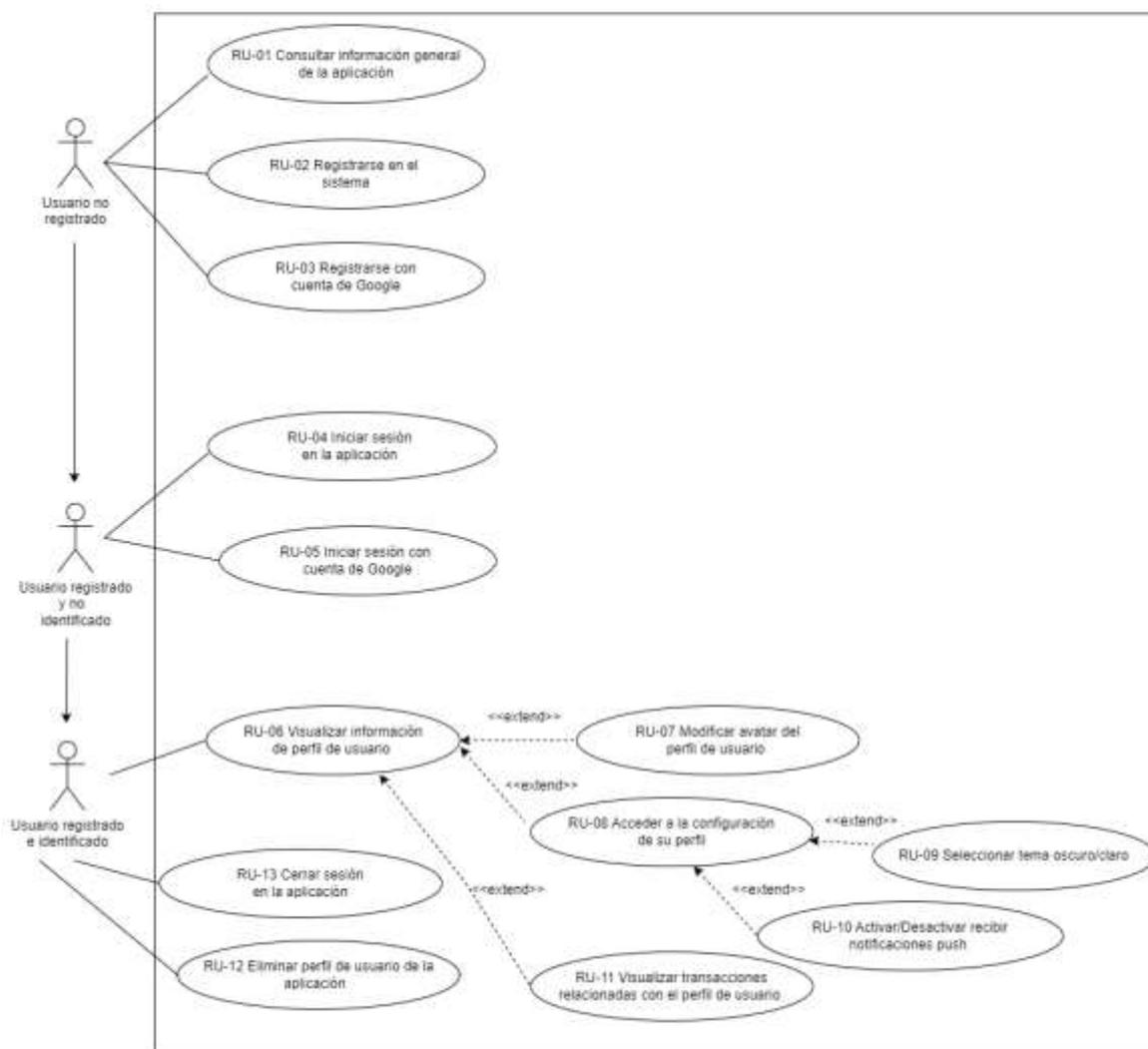


Figura 17: Casos de Uso - Gestión de Usuarios

## 4.8.2. GESTIÓN DE PROCESO DE PUBLICACIÓN

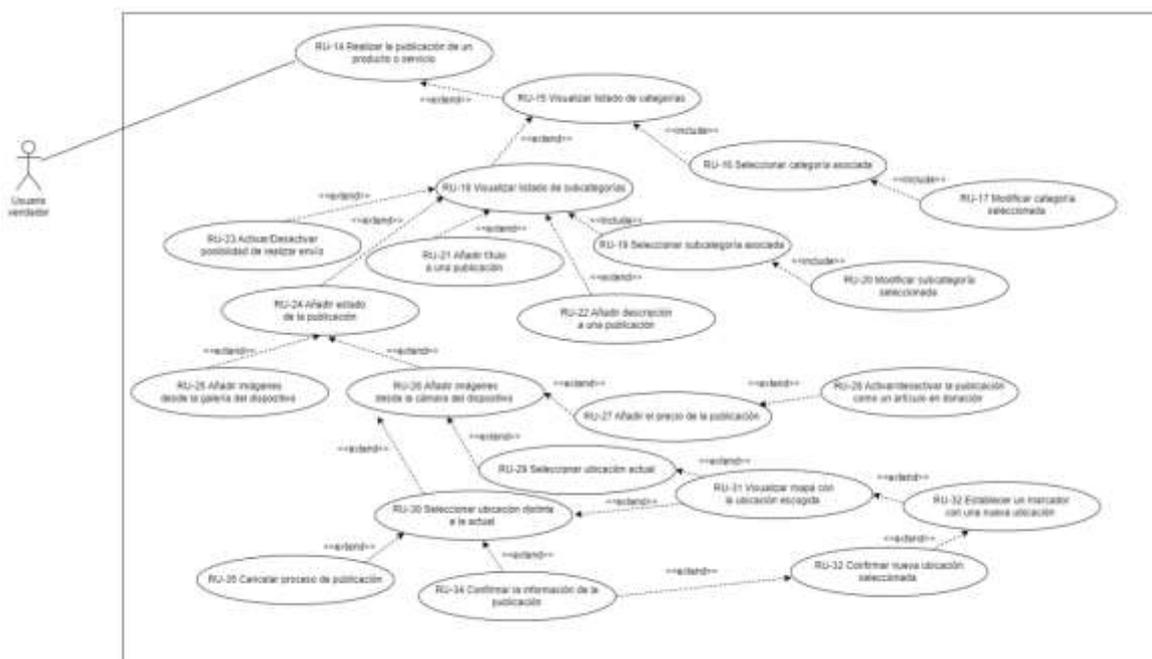


Figura 18: Casos de Uso - Gestión de proceso de publicación - Usuario vendedor

## 4.8.3. GESTIÓN DE PUBLICACIONES

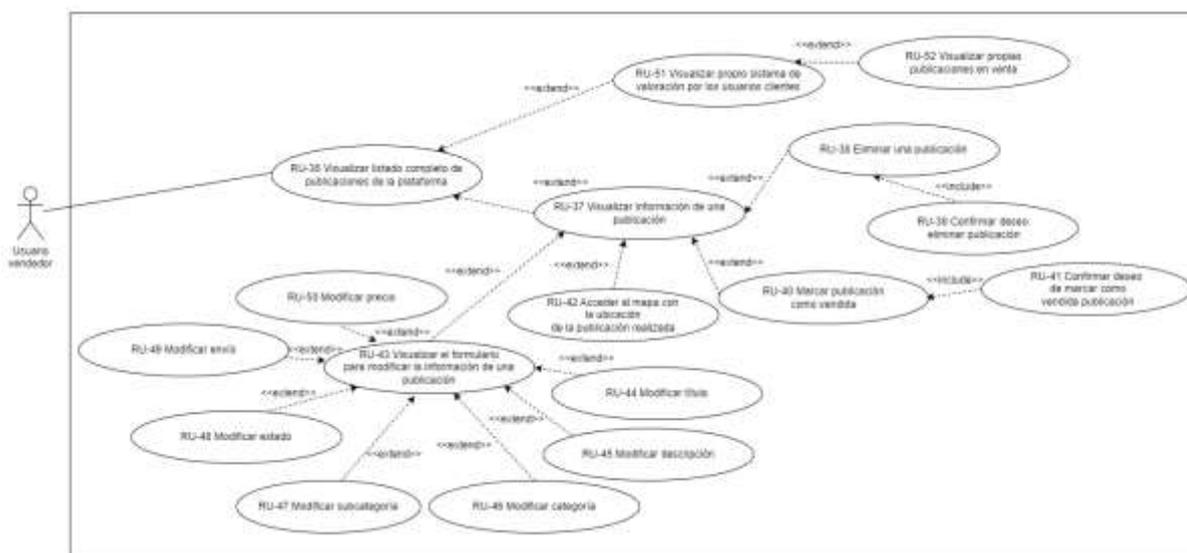


Figura 19: Casos de Uso - Gestión de publicaciones – Usuario vendedor

#### 4.8.4. GESTIÓN DE PUBLICACIONES Y GESTIÓN DE SISTEMA DE VALORACIÓN ENTRE USUARIOS

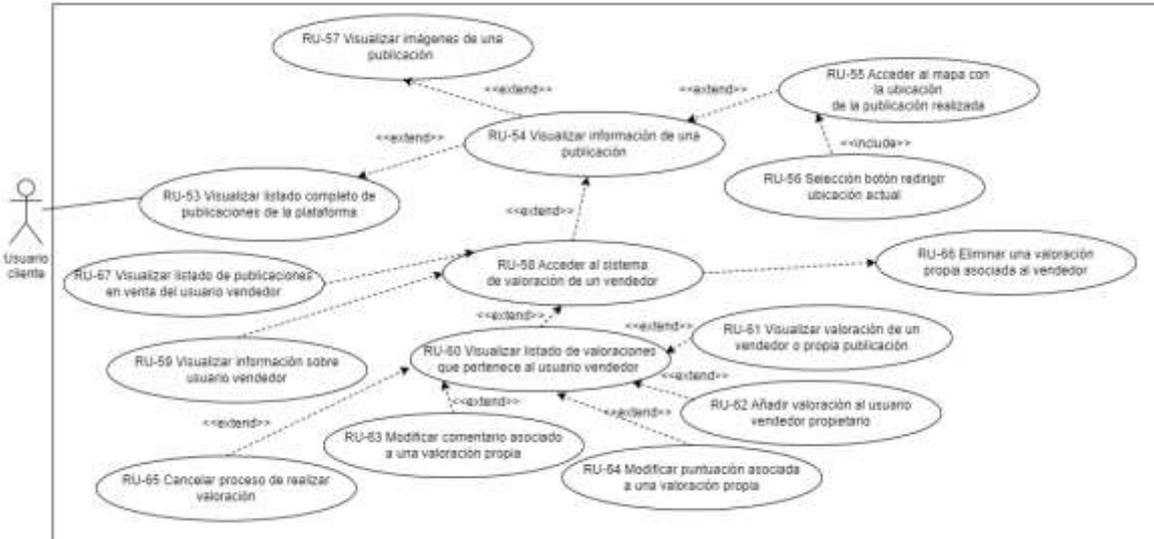


Figura 20: Casos de Uso - Gestión de publicaciones + Gestión de sistema de valoraciones - Usuario cliente

#### 4.8.5. GESTIÓN DE PUBLICACIONES Y GESTIÓN DE SISTEMA DE MENSAJERÍA INSTANTÁNEA

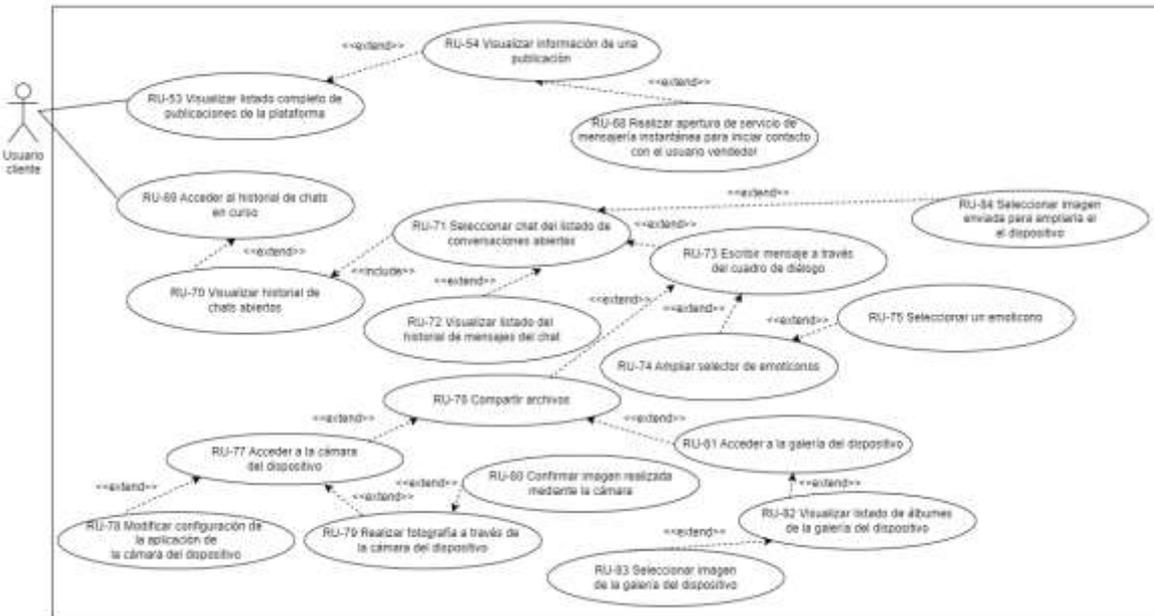


Figura 21: Casos de Uso - Gestión de publicaciones + Gestión de sistema de valoraciones - Usuario cliente

#### 4.8.6. GESTIÓN DE SERVICIO DE MENSAJERÍA INSTANTÁNEA

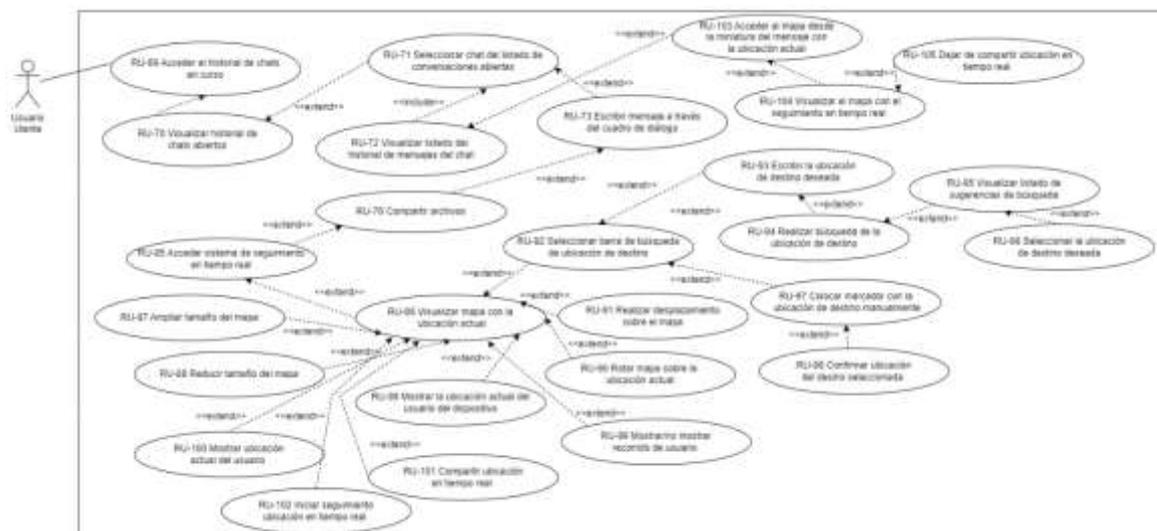


Figura 22: Casos de Uso - Gestión de servicios de mensajería instantánea – Usuario cliente

#### 4.8.7. GESTIÓN DE PUBLICACIONES

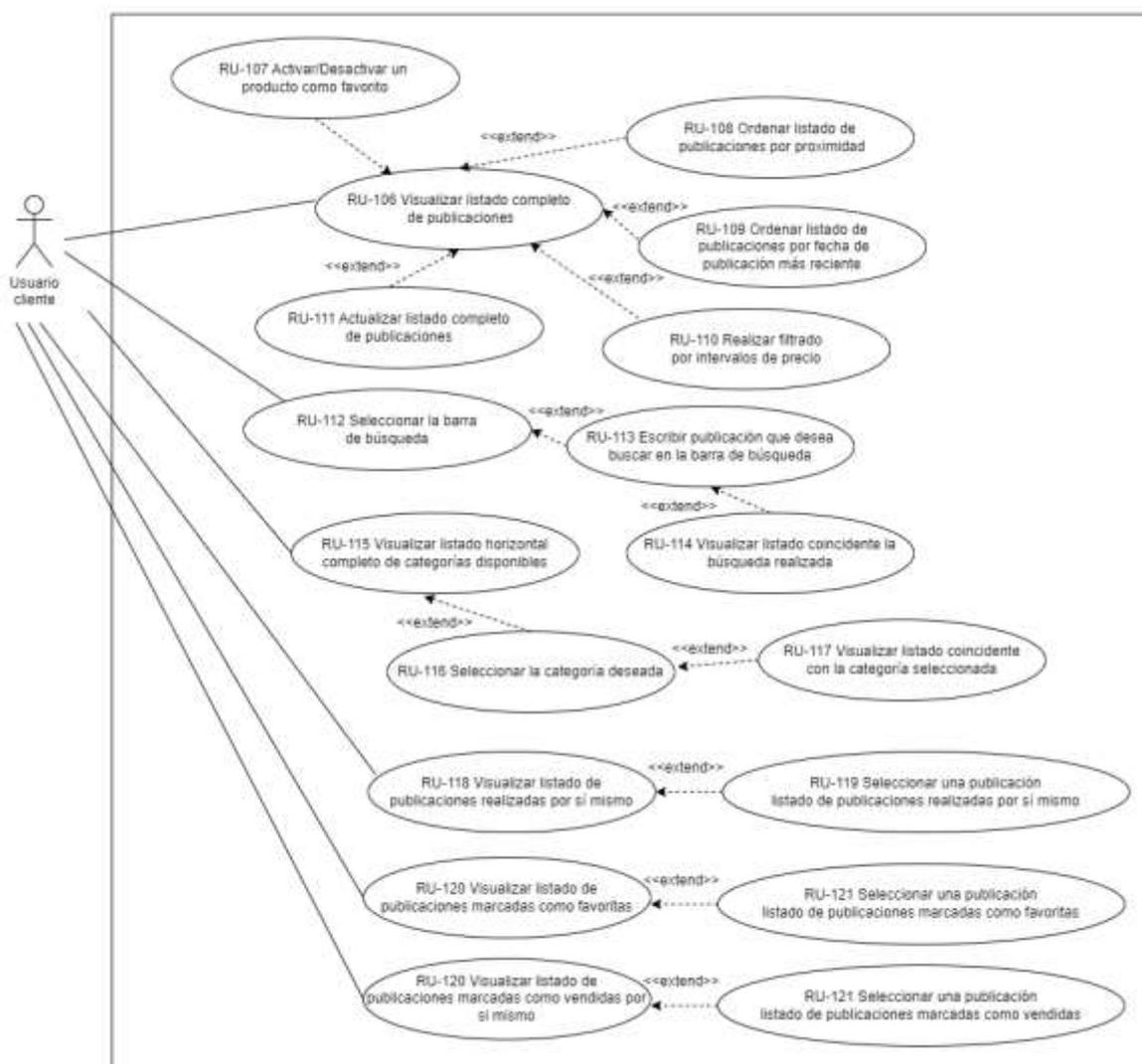


Figura 23: Casos de Uso - Gestión de publicaciones - Usuario cliente

## 4.9 ESPECIFICACIÓN CASOS DE USO

En este apartado, se mostrará la especificación de los Casos de Uso mostrados en el apartado “4.8 Diagrama casos de uso” en las distintas ilustraciones, con los requisitos funcionales que participan en cada caso de uso.

### 4.9.1 CASO DE USO – USUARIO NO REGISTRADO



Figura 24 - Caso de Uso - Usuario no registrado

ID y Nombre	RU-01 CONSULTAR INFORMACIÓN GENERAL DE LA APLICACIÓN
Versión	1.0
Actor principal	Usuario no registrado
Descripción	El usuario podrá consultar la información general de la aplicación accediendo a su página principal
Precondiciones	-
Postcondiciones	-
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario accede a la página principal de la plataforma</li> <li>2. El sistema muestra la información que puede ser visible sin necesidad de estar autenticado</li> </ol>
Excepciones	-
Trazabilidad	✓ RU-01
Prioridad	Alta

Tabla 36: RU-01: Consultar información general de la aplicación

ID y Nombre	RU-02 REGISTRARSE EN EL SISTEMA
Versión	1.0
Actor principal	Usuario no registrado
Descripción	Un usuario no registrado en el sistema podrá registrar una nueva cuenta de usuario en la plataforma
Precondiciones	El usuario no registrado debe contar con conectividad a Internet

<b>Postcondiciones</b>	El sistema almacena la información de un nuevo usuario en la base de datos
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario no registrado solicita crear una nueva cuenta en el sistema.</li> <li>2. El sistema solicita al usuario no registrado los datos correspondientes mediante un formulario para completar el registro en la plataforma.</li> <li>3. El usuario no registrado introduce los datos correspondientes en el formulario y los envía al sistema.</li> <li>4. El sistema comprueba los datos introducidos por el usuario no registrado e insertará u nuevo usuario en la base de datos.</li> <li>5. El sistema registra los datos del usuario no registrado en la plataforma</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>4. El sistema indica que el email introducido ya existe en la plataforma. <ol style="list-style-type: none"> <li>4.1 El sistema notifica del error al usuario no registrado y muestra en pantalla un nuevo registro para el usuario no registrado.</li> <li>4.2 El flujo normal vuelve al punto 2.</li> </ol> </li> <li>4. El sistema detecta que la contraseña es demasiado corta. <ol style="list-style-type: none"> <li>4.1. El sistema notifica al usuario no registrado y muestra en pantalla un nuevo registro para el usuario no registrado.</li> <li>4.2 El flujo normal vuelve al punto 2.</li> </ol> </li> <li>4. El sistema detecta que alguno de los campos obligatorios del formulario no ha sido rellenado para el registro del usuario no registrado. <ol style="list-style-type: none"> <li>4.1 – El sistema notifica al usuario no registrado y muestra en pantalla un nuevo registro para el usuario no registrado.</li> <li>4.2 – El flujo normal vuelve al punto 2.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓RU-02</li> <li>✓RF-05</li> <li>✓RF-06</li> <li>✓RF-08</li> <li>✓RF-09</li> <li>✓RF-10</li> <li>✓RF-11</li> </ul>
<b>Prioridad</b>	Alta

Tabla 37: RU-02 Registrarse en el sistema

<b>ID y Nombre:</b>	<b>RU-03 REGISTRARSE CON USUARIO DE GOOGLE</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario no registrado
<b>Descripción</b>	Un usuario no registrado en el sistema podrá registrar una nueva cuenta mediante su usuario de Google en la plataforma
<b>Precondiciones</b>	Disponer de un perfil de usuario en la plataforma de Google

<b>Postcondiciones</b>	El sistema almacena la información de un nuevo usuario registrado con cuenta de Google en la base de datos
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la página inicial de la plataforma</li> <li>2. El sistema muestra el botón que permite el inicio de sesión con Google</li> <li>3. El sistema muestra una ventana emergente con las cuentas de Google disponibles</li> <li>4. El usuario selecciona la cuenta con la que desea registrarse</li> <li>5. El sistema registra los nuevos datos relacionados con esa cuenta de Google en la plataforma.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-03</li> <li>✓ RF-12</li> <li>✓ RF-13</li> <li>✓ RF-14</li> </ul>
<b>Prioridad</b>	Alta

Tabla 38: RU-03 Registrarse con Usuario de Google

#### 4.9.2. CASO DE USO – USUARIO REGISTRADO Y NO IDENTIFICADO

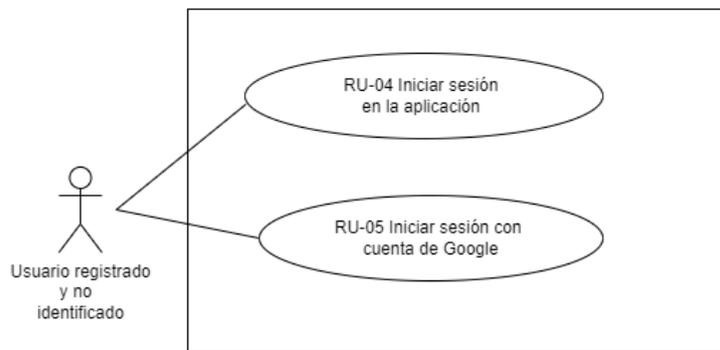


Figura 25: Casos de uso – Usuario registrado y no identificado

ID y Nombre	RU-04 INICIAR SESIÓN EN LA APLICACIÓN
<b>Versión</b>	1.0
<b>Actor principal</b>	Usuario registrado y no identificado
<b>Descripción</b>	Un usuario registrado y no identificado quiere autenticarse en la aplicación mediante su perfil de usuario
<b>Precondiciones</b>	El usuario deberá estar registrado previamente en la plataforma.
<b>Postcondiciones</b>	El usuario registrado y no identificado pasará a ser un usuario registrado e identificado, creándose una nueva sesión del usuario.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado y no identificado solicita iniciar sesión en la aplicación</li> <li>2. El sistema muestra un formulario con los campos que debe introducir el usuario para el proceso de autenticación.</li> </ol>

	<ol style="list-style-type: none"> <li>3. El usuario registrado y no identificado introduce los datos para la autenticación en el formulario, en te caso email y contraseña.</li> <li>4. El sistema valida que los datos introducidos son correctos, y en caso de ser correctos, iniciará sesión.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>4. El sistema valida los campos, y los campos introducidos en el formulario no son correctos o no corresponden con un usuario registrado.             <ol style="list-style-type: none"> <li>4.1 El flujo normal vuelve al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-04</li> <li>✓ RF-01</li> <li>✓ RF-02</li> <li>✓ RF-03</li> <li>✓ RF-04</li> </ul>
<b>Prioridad</b>	Alta

Tabla 39: RU-04: Iniciar sesión en la aplicación

<b>ID y Nombre</b>	
<b>RU-05 INICIAR SESIÓN CON CUENTA DE GOOGLE</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario registrado y no identificado
<b>Descripción</b>	El usuario registrado y no identificado quiere autenticarse en la aplicación mediante su perfil de usuario de Google.
<b>Precondiciones</b>	Haber registrado previamente el usuario mediante la cuenta de Google en la plataforma.
<b>Postcondiciones</b>	El usuario registrado y no identificado ha iniciado sesión mediante su cuenta de Google
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado y no identificado solicita iniciar sesión en la aplicación.</li> <li>2. El sistema muestra una ventana emergente con las cuentas de Google asociadas al dispositivo.</li> <li>3. El usuario selecciona la cuenta de Google con la que desea autenticarse.</li> <li>4. El sistema valida que los datos introducidos son correctos, y en caso de ser correctos, iniciará sesión.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El sistema comprueba que los datos de la cuenta de Google, no son válidos o no se ha registrado previamente.             <ol style="list-style-type: none"> <li>3.1 El flujo retorna al paso 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-05</li> <li>✓ RF-12</li> <li>✓ RF-13</li> <li>✓ RF-14</li> </ul>
<b>Prioridad</b>	Alta

Tabla 40: RU-05: Iniciar sesión con cuenta de Google

### 4.9.3. CASO DE USO – USUARIO REGISTRADO E IDENTIFICADO

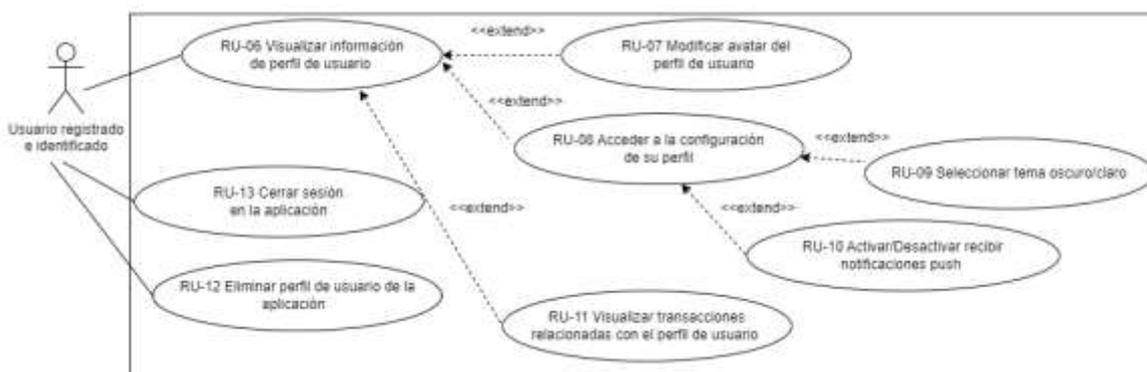


Figura 26: Caso de uso – Usuario registrado e identificado

ID y Nombre	RU-06 VISUALIZAR INFORMACIÓN DE PERFIL DE USUARIO
Versión	1.0
Actor principal	Usuario registrado e identificado
Descripción	El usuario registrado e identificado podrá visualizar la información de su perfil de usuario
Precondiciones	El usuario debe estar registrado e autenticado en la plataforma previamente.
Postcondiciones	El usuario ha visualizado la información asociada a su perfil de usuario
Flujo normal	1. El usuario autenticado solicita visualizar la información almacenada en su perfil de usuario 2. El sistema muestra por pantalla la información del perfil de usuario asociado al usuario identificado en el sistema.
Excepciones	-
Trazabilidad	✓ RU-06 ✓ RF-15
Prioridad	Alta

Tabla 41: RU-06 Visualizar información de perfil de usuario

ID y Nombre	RU-07 MODIFICAR AVATAR DEL PERFIL DE USUARIO
Versión	1.0
Actor principal	Usuario registrado e identificado
Descripción	El usuario registrado e identificado puede modificar el avatar asociado a su perfil de usuario
Precondiciones	El usuario registrado e identificado debe haber seleccionado previamente el RU-06

<b>Postcondiciones</b>	El usuario registrado e identificado retornará al RU-06, pero con el avatar asociado al perfil de usuario modificado.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado quiere modificar el avatar asociado a su perfil de usuario</li> <li>2. El sistema muestra el avatar actual asociado al perfil de usuario, junto con el botón que permite modificar la imagen asociada.</li> <li>3. El usuario registrado e identificado selecciona el botón.</li> <li>4. El sistema muestra un cuadro de dialogo que da la opción de obtener la imagen a través de la cámara del dispositivo, obtener a través de la galería del dispositivo o eliminar la imagen del perfil de usuario.</li> <li>5. El usuario selecciona la opción que desee.</li> <li>6. El sistema actualiza el avatar de usuario asociado a su perfil, en caso de selección de botón "Eliminar", se asociará un avatar genérico.</li> </ol>
<b>Excepciones</b>	<p>5. El usuario registrado e identificado introduce formato de imagen no aceptada en el sistema.</p> <p>5.1 – El sistema muestra un mensaje de error. El flujo retorna al punto 2.</p>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓RU-06</li> <li>✓RU-07</li> <li>✓RF-16</li> </ul>
<b>Prioridad</b>	Alta

Tabla 42: RU-07 Modificar avatar del perfil de usuario

<b>ID y Nombre</b>	<b>RU-08 ACCEDER A CONFIGURACIÓN DEL PERFIL DE USUARIO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario registrado e identificado
<b>Descripción</b>	El usuario registrado e identificado quiere acceder a la configuración de su perfil de usuario.
<b>Precondiciones</b>	1. El usuario registrado e identificado debe haber seleccionado previamente el RU-06
<b>Postcondiciones</b>	1. El usuario registrado e identificado visualizará la configuración asociada a su perfil de usuario.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado selecciona el menú desplegable de la configuración asociada a su perfil</li> <li>2. El sistema muestra las opciones de modificar el tema de la aplicación y de activar/desactivar notificaciones.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-06</li> <li>✓ RU-08</li> </ul>

<b>Prioridad</b>	Alta
------------------	------

Tabla 43: RU-08: Acceder a configuración de perfil de usuario

<b>ID y Nombre</b>	<b>RU-09 SELECCIONAR TEMA OSCURO/TEMA CLARO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario registrado e identificado
<b>Descripción</b>	El usuario registrado e identificado quiere modificar el tema de claro a oscuro o viceversa, de toda la aplicación.
<b>Precondiciones</b>	El usuario registrado e identificado debe haber seleccionado previamente el RU-06 y RU-07.
<b>Postcondiciones</b>	El usuario registrado e identificado habrá modificado el tema de la aplicación a oscuro o claro.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado selecciona la opción de modificar el tema de la aplicación a oscuro o claro.</li> <li>2. El sistema valida la petición, y modifica los colores de la aplicación a tonos oscuros o claros.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-06</li> <li>✓ RU-08</li> <li>✓ RF-21</li> </ul>
<b>Prioridad</b>	Alta

Tabla 44: RU-09 Seleccionar tema oscuro

<b>ID y Nombre</b>	<b>RU-10 ACTIVAR/DESACTIVAR NOTIFICACIONES</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario registrado e identificado
<b>Descripción</b>	El usuario registrado e identificado quiere activar/desactivar las notificaciones push que muestra el dispositivo asociadas a la aplicación.
<b>Precondiciones</b>	El usuario registrado e identificado debe haber seleccionado previamente el RU-06 y RU-07.
<b>Postcondiciones</b>	El usuario registrado e identificado habrá activado o desactivado las notificaciones push de la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado selecciona la opción de activar/desactivar notificaciones push asociadas a la aplicación.</li> <li>2. El sistema valida la petición, y activa o desactiva la aparición de notificaciones en el dispositivo.</li> </ol>
<b>Excepciones</b>	-

<b>Trazabilidad</b>	✓ RU-06 ✓ RU-08 ✓ RF-10 ✓ RF-22
<b>Prioridad</b>	Alta

Tabla 45: RU-10 Activar/desactivar notificaciones

<b>ID y Nombre</b>	<b>RU-11 VISUALIZAR TRANSACCIONES ASOCIADAS</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario registrado e identificado
<b>Descripción</b>	El usuario registrado e identificado podrá visualizar las transacciones asociadas a su perfil de usuario
<b>Precondiciones</b>	El usuario registrado e identificado debe haber seleccionado previamente el RU-06.
<b>Postcondiciones</b>	El sistema muestra la pantalla dónde se encuentran las transacciones y publicaciones asociadas al perfil de usuario.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado selecciona el menú desplegable de configuración</li> <li>2. El sistema muestra la opción de acceder a las transacciones y publicaciones asociadas al perfil de usuario.</li> </ol>
<b>Excepciones</b>	
<b>Trazabilidad</b>	✓ RU-06 ✓ RU-11 ✓ RF-23
<b>Prioridad</b>	Alta

Tabla 46: RU-11 Visualizar transacciones asociadas al perfil de usuario

<b>ID y Nombre</b>	<b>RU-12 ELIMINAR CUENTA DE USUARIO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario registrado e identificado
<b>Descripción</b>	El usuario registrado e identificado desea eliminar toda la información asociada a su perfil de usuario
<b>Precondiciones</b>	El usuario registrado e identificado debe estar registrado previamente en el sistema, y haber accedido al RU-06
<b>Postcondiciones</b>	El sistema eliminará toda la información relacionada con el usuario que mantiene la sesión activa en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado selecciona el botón “Eliminar cuenta”, para eliminar la cuenta asociada a su perfil de usuario.</li> <li>2. El sistema solicita al usuario la confirmación de que realmente desea eliminar su cuenta.</li> <li>3. El usuario confirma que desea eliminar la cuenta de usuario.</li> <li>4. El sistema elimina toda la información del usuario, así como las publicaciones que ha realizado previamente.</li> </ol>

	5. El sistema redirige al usuario a la vista de registro.
<b>Excepciones</b>	3. El usuario cancela la petición de eliminar su cuenta de usuario. 3.1 – El flujo retorna al paso 1.
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-12</li> <li>✓ RF-17</li> <li>✓ RF-18</li> <li>✓ RF-19</li> <li>✓ RF-20</li> </ul>
<b>Prioridad</b>	Alta

Tabla 47: RU-12 Eliminar cuenta de usuario

<b>ID y Nombre</b>	
<b>RU-13 CERRAR SESIÓN</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario registrado e identificado
<b>Descripción</b>	El usuario registrado e identificado podrá cerrar la sesión actual
<b>Precondiciones</b>	El usuario registrado e identificado debe estar autenticado en el sistema previamente, , y haber accedido al RU-06.
<b>Postcondiciones</b>	El usuario registrado e identificado quedará no autenticado en el sistema.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado debe seleccionar el botón “Cerrar sesión”</li> <li>2. El sistema solicita al usuario la confirmación de que desea cerrar sesión en la plataforma.</li> <li>3. El sistema cierra la sesión actual del usuario, y destruye el token de sesión.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-13</li> <li>✓ RF-24</li> <li>✓ RF-25</li> <li>✓ RF-26</li> </ul>
<b>Prioridad</b>	Alta

Tabla 48: RU-13 Cerrar sesión

#### 4.9.4. CASO DE USO – USUARIO VENDEDOR – GESTIÓN DE PUBLICACIÓN DE ARTÍCULOS

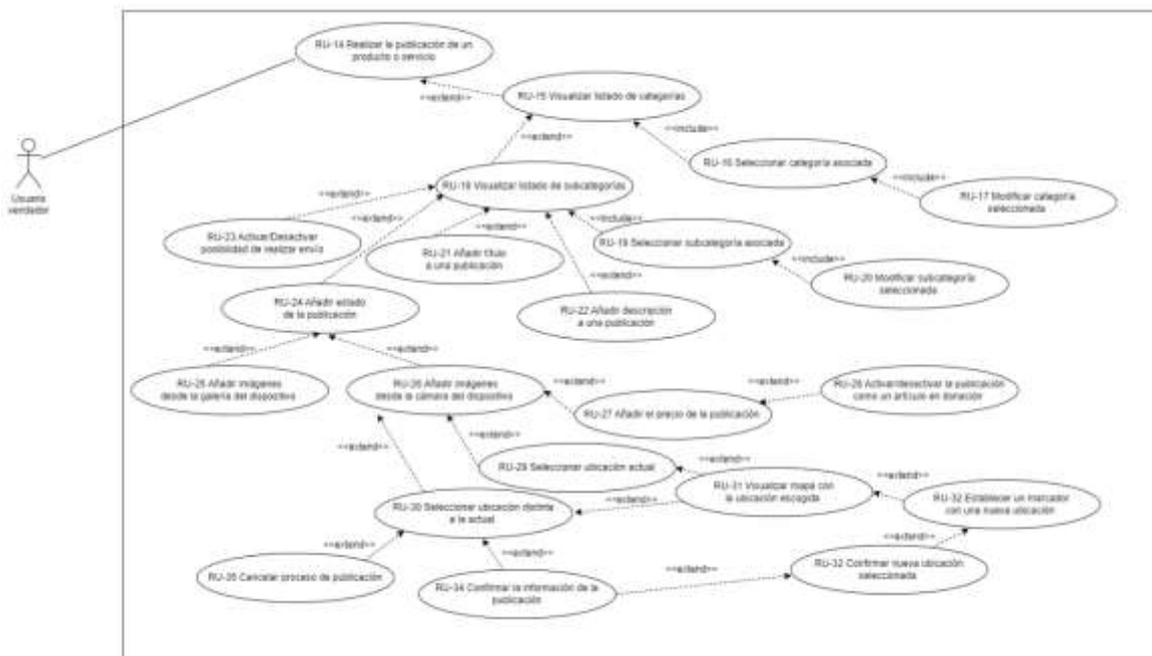


Figura 27: Caso de Uso Usuario Vendedor - Gestión de publicación de artículos

ID y Nombre	RU-14 REALIZAR PUBLICACIÓN DE UN PRODUCTO O SERVICIO
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá realizar una publicación de un producto o servicio desde la vista de “Añadir publicación”.
Precondiciones	El usuario vendedor debe acceder a la vista de “Añadir publicación” desde el menú horizontal de navegabilidad.
Postcondiciones	El usuario vendedor accede al listado completo de categorías
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona desde el menú de navegabilidad horizontal la opción de “Añadir una publicación”, accediendo al proceso de publicación.</li> <li>2. El sistema accede a dicha vista, y muestra el listado completo de categorías disponibles en la plataforma.</li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RF-27</li> <li>✓ RU-34</li> <li>✓ RF-40</li> <li>✓ RF-41</li> </ul>
Prioridad	Alta

Tabla 49: RU-14 Realizar publicación de un producto o servicio

ID y Nombre	RU-15 VISUALIZAR LISTADO DE CATEGORÍAS
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá visualizar el listado completo de categorías que se encuentran disponibles en la plataforma
Precondiciones	El usuario vendedor debe acceder previamente a la vista de “Añadir publicación” desde el menú horizontal de navegabilidad.
Postcondiciones	El usuario vendedor accede al listado completo de categorías
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona desde el menú de navegabilidad horizontal la opción de “Añadir una publicación”</li> <li>2. El sistema accede a dicha vista, y muestra el listado completo de categorías disponibles en la plataforma.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>2. El sistema no contiene ninguna categoría en su base de datos <ol style="list-style-type: none"> <li>2.1 – El sistema muestra un mensaje de error</li> <li>2.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RF-27</li> </ul>
Prioridad	Alta

Tabla 50: RU-15 Visualizar listado de categorías

ID y Nombre	RU-16 SELECCIONAR CATEGORÍA ASOCIADA
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá seleccionar una categoría desde el listado completo de categorías que se encuentran disponibles en la plataforma.
Precondiciones	El usuario vendedor debe acceder previamente a los requisitos RU-14 y RU-15.
Postcondiciones	El usuario vendedor selecciona una categoría del listado completo de categorías.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona una categoría del listado de categorías asociadas disponibles en la plataforma.</li> <li>2. El sistema almacena la categoría seleccionada por el usuario.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>1. El usuario vendedor no selecciona ninguna categoría <ol style="list-style-type: none"> <li>1.1 – El sistema muestra un mensaje de error</li> <li>1.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RF-27</li> </ul>
Prioridad	Alta

Tabla 51: RU-16 Seleccionar categoría asociada

ID y Nombre	RU-17 MODIFICAR CATEGORÍA SELECCIONADA
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá modificar una categoría seleccionada previamente desde el listado completo de categorías que se encuentran disponibles en la plataforma.
Precondiciones	El usuario vendedor debe acceder previamente a los requisitos RU-15 y RU-16.
Postcondiciones	El usuario vendedor modifica una categoría seleccionada previamente.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la categoría seleccionada previamente del listado de categorías asociadas disponibles en la plataforma.</li> <li>2. El sistema almacena la nueva categoría seleccionada por el usuario.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>1. El usuario vendedor no selecciona ninguna categoría <ol style="list-style-type: none"> <li>1.1 – El sistema muestra un mensaje de error</li> <li>1.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RF-27</li> <li>✓ RU-17</li> </ul>
Prioridad	Alta

Tabla 52: RU-17 Modificar categoría seleccionada

ID y Nombre	RU-18 VISUALIZAR LISTADO DE SUBCATEGORÍAS
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá visualizar el listado completo de subcategorías que se encuentran disponibles en la plataforma
Precondiciones	El usuario vendedor debe acceder previamente a la lista de categorías disponibles en la aplicación.
Postcondiciones	El usuario vendedor accede al listado completo de subcategorías
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona previamente una categoría de listado de categorías.</li> <li>2. El sistema accede a dicha vista, y muestra el listado completo de subcategorías disponibles en la plataforma.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>2. El sistema no contiene ninguna subcategoría en su base de datos <ol style="list-style-type: none"> <li>2.1 – El sistema muestra un mensaje de error</li> <li>2.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RF-28</li> <li>✓ RU-18</li> </ul>
Prioridad	Alta

Tabla 53: RU-18 Visualizar listado de subcategorías

ID y Nombre	RU-19 SELECCIONAR SUBCATEGORÍA ASOCIADA
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá seleccionar una subcategoría desde el listado completo de subcategorías que se encuentran disponibles en la plataforma.
Precondiciones	El usuario vendedor debe acceder previamente al requisito RU-18.
Postcondiciones	El usuario vendedor selecciona una subcategoría del listado completo de categorías.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona una subcategoría del listado de categorías asociadas disponibles en la plataforma.</li> <li>2. El sistema almacena la subcategoría seleccionada por el usuario.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>1. El usuario vendedor no selecciona ninguna subcategoría <ol style="list-style-type: none"> <li>1.1 – El sistema muestra un mensaje de error</li> <li>1.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RF-28</li> <li>✓ RU-18</li> <li>✓ RU-19</li> </ul>
Prioridad	Alta

Tabla 54: RU-19 Seleccionar subcategoría asociada

ID y Nombre	RU-20 MODIFICAR SUBCATEGORÍA SELECCIONADA
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá modificar una subcategoría seleccionada previamente desde el listado completo de subcategorías que se encuentran disponibles en la plataforma.
Precondiciones	El usuario vendedor debe acceder previamente al requisito RU-19
Postcondiciones	El usuario vendedor modifica una subcategoría seleccionada previamente.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la subcategoría seleccionada previamente del listado de categorías asociadas disponibles en la plataforma.</li> <li>2. El sistema almacena la nueva subcategoría seleccionada por el usuario.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>1. El usuario vendedor no selecciona ninguna subcategoría <ol style="list-style-type: none"> <li>1.1 – El sistema muestra un mensaje de error</li> <li>1.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RF-28</li> <li>✓ RU-18</li> <li>✓ RU-19</li> <li>✓ RU-20</li> </ul>

<b>Prioridad</b>	Alta
------------------	------

Tabla 55: RU-20 Modificar subcategoría asociada

<b>ID y Nombre</b>	<b>RU-21 AÑADIR TÍTULO A UNA PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá añadir un título durante el proceso de publicación
<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado previamente una subcategoría de la lista de subcategorías disponibles en la aplicación.
<b>Postcondiciones</b>	El usuario vendedor añade un título a la publicación
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un formulario dónde se muestran los campos que debe rellenar para especificar los detalles de la publicación.</li> <li>2. El usuario vendedor selecciona el campo que pertenece a la cualidad de título y rellena el cuadro con el texto deseado.</li> <li>3. El sistema comprueba que la información introducida cumple las restricciones impuestas y lo almacena en memoria.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2. El usuario vendedor introduce un título que no cumple con las restricciones impuestas por la plataforma al ser validado el contenido. <ol style="list-style-type: none"> <li>2.1 – El sistema muestra un mensaje de error</li> <li>2.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RU-18</li> <li>✓ RF-29</li> <li>✓ RU-21</li> <li>✓ RF-30</li> </ul>
<b>Prioridad</b>	Alta

Tabla 56: RU-21 Añadir título a una publicación

<b>ID y Nombre</b>	<b>RU-22 AÑADIR DESCRIPCIÓN A UNA PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá añadir una descripción durante el proceso de publicación
<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado previamente una subcategoría de la lista de subcategorías disponibles en la aplicación.
<b>Postcondiciones</b>	El usuario vendedor añade una descripción a la publicación
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un formulario dónde se muestran los campos que debe rellenar para especificar los detalles de la publicación.</li> </ol>

	<ol style="list-style-type: none"> <li>2. El usuario vendedor selecciona el campo que pertenece a la cualidad de descripción y rellena el cuadro con el texto deseado.</li> <li>3. El sistema comprueba que la información introducida cumple las restricciones impuestas y lo almacena en memoria.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2. El usuario vendedor introduce una descripción que no cumple con las restricciones impuestas por la plataforma al ser validado el contenido. <ol style="list-style-type: none"> <li>2.1 – El sistema muestra un mensaje de error</li> <li>2.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RU-18</li> <li>✓ RF-29</li> <li>✓ RU-22</li> <li>✓ RF-30</li> </ul>
<b>Prioridad</b>	Alta

Tabla 57: RU-22 Añadir descripción a una publicación

<b>ID y Nombre</b>	<b>RU-23 ACTIVAR/DESACTIVAR POSIBILIDAD DE REALIZAR ENVÍO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá indicar si existe la posibilidad de realizar o no envío del producto que contiene la publicación.
<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado previamente una subcategoría de la lista de subcategorías disponibles en la aplicación.
<b>Postcondiciones</b>	El usuario vendedor añade si se admite o no envíos a la publicación
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un formulario dónde se muestran los campos que debe rellenar para especificar los detalles de la publicación.</li> <li>2. El usuario vendedor activa o desactiva la posibilidad de realizar el envío del producto de la publicación.</li> <li>3. El sistema comprueba que la información introducida cumple las restricciones impuestas y lo almacena en memoria.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RU-18</li> <li>✓ RF-29</li> <li>✓ RU-23</li> <li>✓ RF-30</li> </ul>
<b>Prioridad</b>	Alta

Tabla 58: RU-23 Activar/desactivar posibilidad de realizar envío

<b>ID y Nombre</b>		<b>RU-24 AÑADIR ESTADO DE LA PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>	
<b>Actor principal</b>	Usuario vendedor	
<b>Descripción</b>	El usuario vendedor podrá indicar el estado en el que se encuentra el producto que contiene la publicación.	
<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado previamente una subcategoría de la lista de subcategorías disponibles en la aplicación.	
<b>Postcondiciones</b>	El usuario vendedor añade el estado de conversación a la publicación	
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un formulario dónde se muestran los campos que debe rellenar para especificar los detalles de la publicación.</li> <li>2. El usuario vendedor indica dentro del rango establecido, el estado del producto de la publicación.</li> <li>3. El sistema comprueba que la información introducida cumple las restricciones impuestas y lo almacena en memoria.</li> </ol>	
<b>Excepciones</b>	-	
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RU-18</li> <li>✓ RF-29</li> <li>✓ RU-24</li> <li>✓ RF-30</li> </ul>	
<b>Prioridad</b>	Alta	

Tabla 59: RU-24 Añadir estado de publicación

<b>ID y Nombre</b>		<b>RU-25 AÑADIR IMÁGENES DESDE LA GALERÍA DEL DISPOSITIVO</b>
<b>Versión</b>	<b>1.0</b>	
<b>Actor principal</b>	Usuario vendedor	
<b>Descripción</b>	El usuario vendedor podrá añadir imágenes del producto a la aplicación desde la galería del dispositivo a la publicación.	
<b>Precondiciones</b>	El usuario vendedor debe haber completado previamente el formulario de detalles de la publicación, RU-24.	
<b>Postcondiciones</b>	El usuario vendedor añade una/s imagen/es a través de la aplicación de galería del dispositivo a la publicación.	
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un botón que permite acceder a la aplicación de la galería del dispositivo móvil.</li> <li>2. El usuario vendedor selecciona el botón para acceder a la galería del dispositivo.</li> <li>3. El sistema accede a la aplicación de la galería del dispositivo, y muestra los álbumes que se encuentran en el dispositivo.</li> <li>4. El usuario vendedor selecciona la imagen/es que desea adjuntar a la información de la publicación.</li> <li>5. El sistema adjunta al carrusel de imágenes, las imágenes seleccionadas y las almacena en memoria.</li> </ol>	

<b>Excepciones</b>	4. El sistema no muestra ninguna imagen porque la galería del dispositivo no contiene ninguna imagen. 4.1 – El sistema muestra un mensaje de error 4.2 – El flujo de eventos vuelve al punto 1
<b>Trazabilidad</b>	✓ RU-14 ✓ RU-15 ✓ RU-18 ✓ RU-24 ✓ RF-32
<b>Prioridad</b>	Alta

Tabla 60: RU-25 Añadir imágenes desde la galería del dispositivo

<b>ID y Nombre</b>		<b>RU-26 AÑADIR IMÁGENES DESDE LA CÁMARA DEL DISPOSITIVO</b>
<b>Versión</b>		<b>1.0</b>
<b>Actor principal</b>		Usuario vendedor
<b>Descripción</b>		El usuario vendedor podrá añadir imágenes del producto a la aplicación desde la cámara del dispositivo a la publicación.
<b>Precondiciones</b>		El usuario vendedor debe haber completado previamente el formulario de detalles de la publicación, RU-24.
<b>Postcondiciones</b>		El usuario vendedor añade una/s imagen/es a través de la aplicación de galería del dispositivo a la publicación.
<b>Flujo normal</b>		1. El sistema muestra al usuario un botón que permite acceder a la aplicación de la cámara del dispositivo móvil. 2. El usuario vendedor selecciona el botón para acceder a la cámara del dispositivo. 3. El sistema accede a la aplicación de la cámara del dispositivo, y muestra las configuraciones de la cámara que se encuentran en el dispositivo. 4. El usuario vendedor realiza la fotografía/s que desea adjuntar a la información de la publicación, y confirma que desea añadir esa imagen. 5. El sistema adjunta al carrusel de imágenes, las imágenes seleccionadas y las almacena en memoria.
<b>Excepciones</b>		4. El sistema no muestra la aplicación de la cámara del dispositivo por problemas ajenos a la plataforma. 4.1. Se retorna al paso 1.
<b>Trazabilidad</b>		✓ RU-14 ✓ RU-15 ✓ RU-18 ✓ RU-24 ✓ RF-31
<b>Prioridad</b>		Alta

Tabla 61: RU-26 Añadir imágenes desde la cámara del dispositivo

ID y Nombre	RU-27 AÑADIR PRECIO A UNA PUBLICACIÓN
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá añadir un precio durante el proceso de publicación
Precondiciones	El usuario vendedor debe haber adjuntado previamente una/s imagen/es a la publicación, RU-25 y RU-26.
Postcondiciones	El usuario vendedor añade un precio a la publicación
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un formulario dónde se muestran los campos que debe rellenar para especificar los detalles de la publicación.</li> <li>2. El usuario vendedor selecciona el campo que pertenece a la cualidad de precio y rellena el cuadro con la cantidad deseada en euros.</li> <li>3. El sistema comprueba que la información introducida cumple las restricciones impuestas y lo almacena en memoria.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>2. El usuario vendedor introduce un precio que no cumple con las restricciones impuestas por la plataforma al ser validado el contenido. <ol style="list-style-type: none"> <li>2.1 – El sistema muestra un mensaje de error</li> <li>2.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RU-18</li> <li>✓ RU-24</li> <li>✓ RU-26</li> <li>✓ RF-33</li> <li>✓ RU-27</li> </ul>
Prioridad	Alta

Tabla 62: RU-27 Añadir precio a una publicación

ID y Nombre	RU-28 ACTIVAR/DESACTIVAR PUBLICACIÓN COMO DONACIÓN
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá indicar si una publicación se establece como una donación, lo que supondrá un coste de 0€ para el cliente.
Precondiciones	El usuario vendedor debe haber adjuntado previamente una/s imagen/es a la publicación, RU-25 y RU-26.
Postcondiciones	El usuario vendedor indica si se marca como donación o no la publicación.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un formulario dónde se muestran los campos que debe rellenar para especificar los detalles de la publicación.</li> <li>2. El usuario vendedor activa o desactiva si la publicación se considera donación o no.</li> </ol>

	3. El sistema comprueba que la información introducida cumple las restricciones impuestas y lo almacena en memoria.
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-14</li> <li>✓ RU-15</li> <li>✓ RU-18</li> <li>✓ RU-24</li> <li>✓ RU-26</li> <li>✓ RU-27</li> <li>✓ RF-34</li> <li>✓ RU-28</li> </ul>
<b>Prioridad</b>	Alta

Tabla 63: RU-28 Activar/Desactivar publicación como donación

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-29 SELECCIONAR UBICACIÓN ACTUAL</b>
<b>Versión</b>	1.0
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá seleccionar la ubicación actual del dispositivo, como ubicación en la que se va a encontrar la publicación.
<b>Precondiciones</b>	El usuario vendedor debe haber adjuntado previamente una/s imagen/es a la publicación, RU-25 y RU-26.
<b>Postcondiciones</b>	El usuario vendedor indica que la ubicación de la publicación es la ubicación actual del dispositivo.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un formulario dónde se muestran los campos que debe rellenar para especificar los detalles de la publicación.</li> <li>2. El usuario vendedor selecciona el mapa donde se establece la ubicación, y selecciona el botón que indica "Ubicación actual".</li> <li>3. El sistema recibe la solicitud, y a través de la API de Google Maps y la ubicación del dispositivo móvil, obtiene la localización actual y lo añade en memoria.</li> <li>4. El sistema refleja tanto la dirección como la ubicación en el mapa con un marcador en el mapa actualizado.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-24</li> <li>✓ RU-26</li> <li>✓ RF-33</li> <li>✓ RU-26</li> <li>✓ RF-35</li> <li>✓ RU-29</li> </ul>
<b>Prioridad</b>	Alta

Tabla 64: RU-29 Seleccionar ubicación actual

ID y Nombre	RU-30 SELECCIONAR UNA UBICACIÓN DISTINTA A LA ACTUAL
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá seleccionar una ubicación cualquiera, estableciendo un marcador en el mapa, como ubicación en la que se va a encontrar la publicación.
Precondiciones	El usuario vendedor debe haber adjuntado previamente una/s imagen/es a la publicación, RU-25 y RU-26.
Postcondiciones	El usuario vendedor indica que la ubicación de la publicación es la indicada mediante el marcador en el mapa.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un formulario dónde se muestran los campos que debe rellenar para especificar los detalles de la publicación.</li> <li>2. El usuario vendedor selecciona el mapa donde se establece la ubicación, y selecciona el botón que indica "Elegir ubicación".</li> <li>3. El sistema recibe la solicitud, y a través de la API de Google Maps y la ubicación del dispositivo móvil, obtiene la localización actual y muestra la posición del usuario en el mapa.</li> <li>4. El usuario selecciona mediante el marcador, donde desea que se encuentre la ubicación de la publicación.</li> <li>5. El sistema refleja tanto la dirección como la ubicación en el mapa con un marcador en el mapa actualizado.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-24</li> <li>✓ RU-26</li> <li>✓ RF-33</li> <li>✓ RU-26</li> <li>✓ RF-35</li> <li>✓ RU-30</li> </ul>
Prioridad	Alta

Tabla 65: RU-30 Seleccionar una ubicación distinta a la actual

ID y Nombre	RU-31 VISUALIZAR MAPA CON LA UBICACIÓN SELECCIONADA
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá seleccionar el mapa donde se ha establecido la ubicación, para poder ampliarlo a pantalla completa en el dispositivo móvil.
Precondiciones	El usuario vendedor debe haber seleccionado una ubicación previamente, RU-29 y RU-30.
Postcondiciones	El usuario vendedor ha visualizado el mapa que contiene la ubicación que ha indicado.

<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el mapa mediante la API de Google Maps, la ubicación que ha escogido para mostrar en la publicación.</li> <li>2. El usuario vendedor selecciona el mapa donde se establece la ubicación, y lo visualiza a pantalla completa.</li> <li>3. El sistema permite al usuario realizar movimientos sobre el mapa mediante gestos en la pantalla del dispositivo móvil.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-29</li> <li>✓ RU-30</li> <li>✓ RU-31</li> <li>✓ RF-35</li> <li>✓ RU-31</li> </ul>
<b>Prioridad</b>	Alta

Tabla 66: RU-31 Visualizar mapa con la ubicación seleccionada

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-32 ESTABLECER MARCADOR CON NUEVA UBICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá seleccionar mediante el mapa donde se ha establecido la ubicación un nuevo marcador que indique una nueva ubicación en el mapa, modificando así la ubicación anterior.
<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado una ubicación previamente, RU-29 y RU-30 y RU-31.
<b>Postcondiciones</b>	El usuario vendedor ha visualizado el mapa que contiene la ubicación que ha indicado y ha establecido un nuevo marcador.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el mapa mediante la API de Google Maps, la ubicación que ha escogido para mostrar en la publicación.</li> <li>2. El usuario vendedor selecciona el mapa donde se establece la ubicación, y lo visualiza a pantalla completa. Selecciona la nueva ubicación, estableciendo un nuevo marcador.</li> <li>3. El sistema almacena la nueva ubicación indicada.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-29</li> <li>✓ RU-30</li> <li>✓ RU-31</li> <li>✓ RF-35</li> <li>✓ RU-31</li> <li>✓ RU-32</li> </ul>
<b>Prioridad</b>	Alta

Tabla 67: RU-32 Establecer marcador con nueva ubicación

ID y Nombre	RU-33 CONFIRMAR NUEVA UBICACIÓN ESTABLECIDA
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá seleccionar mediante el mapa donde se ha establecido la nueva ubicación, el botón de confirmación de la nueva ubicación seleccionada.
Precondiciones	El usuario vendedor debe haber seleccionado una ubicación previamente, RU-29 y RU-30 y RU-31.
Postcondiciones	El usuario vendedor ha visualizado el mapa que contiene la ubicación que ha indicado y ha confirmado la nueva ubicación.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un botón que permite confirmar la nueva ubicación seleccionada con el marcador previamente.</li> <li>2. El sistema almacena la nueva ubicación indicada.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-29</li> <li>✓ RU-30</li> <li>✓ RU-31</li> <li>✓ RF-35</li> <li>✓ RU-31</li> <li>✓ RU-32</li> <li>✓ RU-33</li> </ul>
Prioridad	Alta

Tabla 68: RU-33 Confirmar nueva ubicación establecida

ID y Nombre	RU-34 CONFIRMAR PROCESO DE PUBLICACIÓN
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá confirmar mediante el botón de confirmación que toda la información detallada previamente de la publicación es correcta y quiere ser publicada.
Precondiciones	1. El usuario vendedor debe haber completado toda la información requerida para realizar la publicación.
Postcondiciones	El usuario vendedor ha realizado una nueva publicación en la plataforma con la información introducida.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un botón que permite confirmar la toda la información añadida previamente, e indicarle a la plataforma que desea realizar dicha publicación.</li> <li>2. El sistema almacena toda la información de la publicación, realizando una nueva inserción en la base de datos.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-30</li> <li>✓ RU-31</li> <li>✓ RF-35</li> <li>✓ RU-31</li> </ul>

	<ul style="list-style-type: none"> <li>✓ RU-32</li> <li>✓ RU-33</li> <li>✓ RF-36</li> <li>✓ RU-34</li> <li>✓ RF-37</li> </ul>
<b>Prioridad</b>	Alta

Tabla 69: RU-34 Confirmar proceso de publicación

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-35 CANCELAR PROCESO DE PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá cancelar mediante el botón de “Cancelar” que desea cancelar todo el proceso de publicación recorrido.
<b>Precondiciones</b>	El usuario vendedor debe haber completado toda la información requerida para realizar la publicación y haber confirmado el deseo de realizar la publicación.
<b>Postcondiciones</b>	El usuario vendedor ha cancelado el proceso de publicación recorrido, retornando al RU-14.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un botón que permite cancelar el proceso de publicación, una vez confirmada la publicación y antes de que se complete la inserción en la base de datos.</li> <li>2. El sistema destruye toda la información de la publicación, almacenada en memoria.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-33</li> <li>✓ RF-36</li> <li>✓ RU-34</li> <li>✓ RF-37</li> <li>✓ RU-35</li> <li>✓ RF-38</li> <li>✓ RF-39</li> </ul>
<b>Prioridad</b>	Alta

Tabla 70: RU-35 Cancelar proceso de publicación

#### 4.9.5 CASO DE USO – USUARIO VENDEDOR – GESTIÓN DE PUBLICACIONES

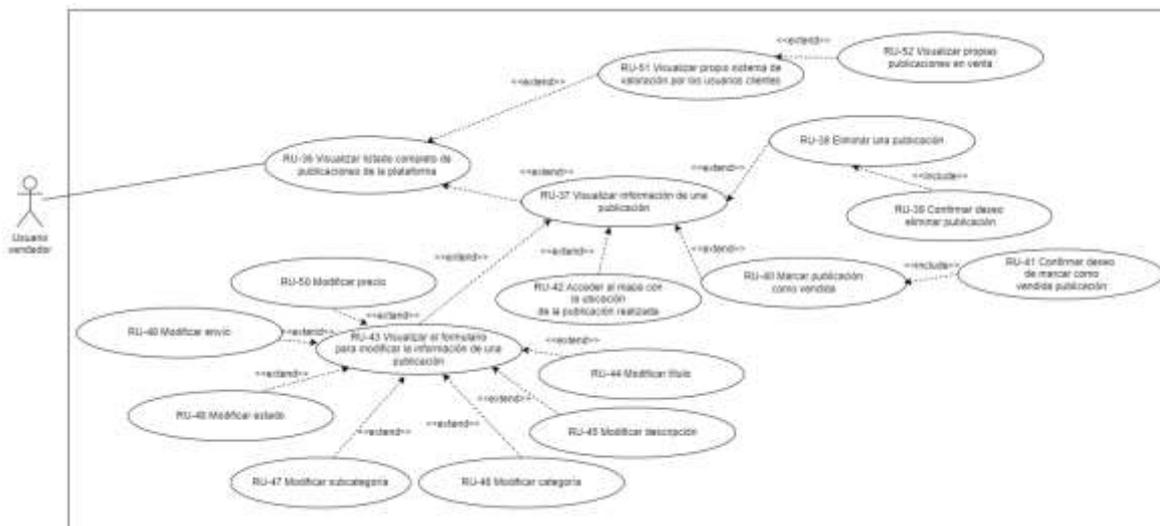


Figura 28: Caso de Uso Usuario Vendedor - Gestión de publicaciones

ID y Nombre	RU-36 VISUALIZAR LISTADO COMPLETO DE PUBLICACIONES
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá visualizar el listado completo de publicaciones realizadas en la plataforma por todos los usuarios activos.
Precondiciones	El usuario vendedor debe haberse registrado y autenticado en la aplicación previamente.
Postcondiciones	El usuario vendedor ha visualizado el listado completo de publicaciones activas en la plataforma por el resto de usuarios.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario se autentica en la aplicación y accede a la página de inicio de la publicación.</li> <li>2. El sistema muestra al usuario un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>2. En caso de que el listado de publicaciones se encuentre vacío, el sistema mostrará un mensaje indicando que no hay publicaciones realizadas.</li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-36</li> <li>✓ RF-42</li> <li>✓ RF-43</li> </ul>
Prioridad	Alta

Tabla 71: RU-36 Visualizar listado completo de publicaciones

<b>RU-37 VISUALIZAR INFORMACIÓN DE UNA PUBLICACIÓN</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá visualizar el contenido al detalle de una publicación realizada al seleccionar la miniatura de la publicación.
<b>Precondiciones</b>	El usuario vendedor debe haber visualizado el listado completo de publicaciones de la plataforma y seleccionado una de las miniaturas, RU-36.
<b>Postcondiciones</b>	El usuario vendedor ha visualizado el detalle de una publicación realizada en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> <li>2. El usuario visualiza el listado completo de publicaciones de la plataforma y escoge una de las miniaturas.</li> <li>3. El sistema redirige al usuario a otra pantalla donde puede visualizar al detalle la publicación.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-36</li> <li>✓ RF-44</li> <li>✓ RU-37</li> </ul>
<b>Prioridad</b>	Alta

Tabla 72: RU-37 Visualizar información de una publicación

<b>RU-38 ELIMINAR UNA PUBLICACIÓN</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá eliminar una publicación, publicada por sí mismo previamente en la plataforma.
<b>Precondiciones</b>	El usuario vendedor debe haber realizado la publicación previamente en la plataforma, y visualizado al detalle RU-37.
<b>Postcondiciones</b>	El usuario vendedor ha solicitado eliminar dicha publicación realizada en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario la información al detalle de la publicación, junto con un menú desplegable en el que se encuentra la opción "Eliminar publicación", que se está visualizando actualmente.</li> <li>2. El usuario selecciona la opción de "Eliminar publicación".</li> <li>3. El sistema solicita al usuario si desea realmente eliminar la publicación que está visualizando actualmente (RU-39).</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-69</li> </ul>
<b>Prioridad</b>	Alta

Tabla 73: RU-38 Eliminar una publicación

ID y Nombre	RU-39 CONFIRMAR ELIMINAR UNA PUBLICACIÓN
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá confirmar si desea eliminar una publicación, publicada por sí mismo previamente en la plataforma.
Precondiciones	El usuario vendedor debe haber realizado la publicación previamente en la plataforma, y visualizado al detalle, RU-37, además de haber solicitado eliminar una publicación ( RU-38).
Postcondiciones	El usuario vendedor ha eliminado por completo una publicación realizada en la plataforma.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema solicita al usuario que confirme, mediante un cuadro de diálogo si realmente desea eliminar la publicación que ha solicitado eliminar.</li> <li>2. El usuario selecciona la opción de “Confirmar”.</li> <li>3. El sistema elimina por completo toda la información relacionada con la publicación de la base de datos.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-69</li> <li>✓ RU-38</li> <li>✓ RF-70</li> <li>✓ RF-71</li> </ul>
Prioridad	Alta

Tabla 74: RU-39 Confirmar eliminar una publicación

ID y Nombre	RU-40 MARCAR UNA PUBLICACIÓN COMO VENDIDA
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá marcar una publicación realizada previamente como vendida.
Precondiciones	El usuario vendedor debe haber realizado la publicación previamente en la plataforma, y visualizado al detalle, RU-37.
Postcondiciones	El usuario vendedor ha solicitado marcar como vendida una publicación realizada en la plataforma.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario la información al detalle de la publicación, junto con un menú desplegable en el que se encuentra la opción “Marcar publicación como vendida”, que se está visualizando actualmente.</li> <li>2. El usuario selecciona la opción de “Marcar publicación como vendida”.</li> <li>3. El sistema solicita al usuario si desea realmente marcar la publicación como vendida que está visualizando actualmente (RU-39).</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-69</li> <li>✓ RU-40</li> </ul>

	✓ RF-72
<b>Prioridad</b>	Alta

Tabla 75: RU-40 Marcar una publicación como vendida

<b>ID y Nombre</b>	<b>RU-41 CONFIRMAR ELIMINAR UNA PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá confirmar si desea marcar como vendida una publicación, publicada por sí mismo previamente en la plataforma.
<b>Precondiciones</b>	El usuario vendedor debe haber realizado la publicación previamente en la plataforma, y visualizado al detalle, RU-37, además de haber solicitado marcar como vendida una publicación ( RU-39).
<b>Postcondiciones</b>	El usuario vendedor ha marcado como vendida una publicación realizada en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema solicita al usuario que confirme, mediante un cuadro de diálogo si realmente desea marcar como vendida la publicación que ha solicitado.</li> <li>2. El usuario selecciona la opción de “Confirmar”.</li> <li>3. El sistema modifica el estado y marca la publicación como vendida en la base de datos.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-69</li> <li>✓ RU-40</li> <li>✓ RF-72</li> <li>✓ RF-73</li> <li>✓ RU-41</li> <li>✓ RF-74</li> </ul>
<b>Prioridad</b>	Alta

Tabla 76: RU-41 Confirmar eliminar una publicación

<b>ID y Nombre</b>	<b>RU-42 ACCEDER AL MAPA CON LA UBICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá acceder al mapa con la ubicación de la publicación.
<b>Precondiciones</b>	El usuario vendedor debe haber realizado la publicación previamente en la plataforma, y visualizado al detalle, RU-37.
<b>Postcondiciones</b>	El usuario vendedor ha visualizado el mapa con la ubicación relacionada con la publicación realizada en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el mapa con la ubicación relacionada con la publicación, mediante un marcador y el mapa de la API de Google Maps.</li> </ol>

	<ol style="list-style-type: none"> <li>El usuario selecciona el mapa para ampliarlo a pantalla completa en el dispositivo.</li> <li>El sistema muestra el mapa a pantalla completa junto la ubicación actual del usuario y la ubicación de la publicación.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> <li>✓ RU-42</li> <li>✓ RF-94</li> </ul>
<b>Prioridad</b>	Alta

Tabla 77: RU-42 Acceder al mapa con la ubicación

<b>ID y Nombre</b>	<b>RU-43 VISUALIZAR FORMULARIO PARA MODIFICAR PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá acceder al formulario donde se muestra la información de la publicación seleccionada que se puede modificar.
<b>Precondiciones</b>	El usuario vendedor debe haber realizado la publicación previamente en la plataforma, y visualizado al detalle, RU-37, además de seleccionar el botón que permite modificar la publicación.
<b>Postcondiciones</b>	El usuario vendedor ha accedido al formulario donde se encuentra la información a modificar por la publicación.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>El sistema muestra al usuario el botón que permite seleccionar al usuario para acceder al formulario.</li> <li>El usuario selecciona el botón, con esta acción accede a la página del formulario y visualiza la información que puede modificar.</li> <li>El sistema muestra los datos relacionados con la publicación.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> <li>✓ RU-43</li> <li>✓ RF-76</li> </ul>
<b>Prioridad</b>	Alta

Tabla 78: RU-43 Visualizar formulario para modificar publicación

<b>ID y Nombre</b>	<b>RU-44 MODIFICAR TÍTULO DE PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá modificar el título asociado a la publicación desde el formulario al que ha accedido en RU-43.

<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado el botón de modificar publicación, y visualizar el formulario con el contenido de la publicación que podemos modificar.
<b>Postcondiciones</b>	El usuario vendedor ha accedido al formulario donde se encuentra la información a modificar por la publicación y ha modificado el título asociado.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el formulario que contiene la información de la publicación que se puede modificar.</li> <li>2. El usuario selecciona el campo que desea modificar, en este caso el título, y escribe el título que desee.</li> <li>3. El sistema muestra el nuevo contenido escrito y valida que la información introducida por teclado por el usuario cumple las restricciones impuestas.</li> <li>4. El usuario selecciona el botón de confirmar.</li> <li>5. El sistema actualiza la información modificada asociada a esa publicación en la base de datos.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El sistema valida la información introducida y no es válido. 3.1 Retorna al punto 1.</li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> <li>✓ RU-43</li> <li>✓ RF-76</li> <li>✓ RF-77</li> <li>✓ RF-78</li> <li>✓ RU-44</li> </ul>
<b>Prioridad</b>	Alta

Tabla 79: RU-44 Modificar título de la publicación

<b>ID y Nombre</b>	<b>RU-45 MODIFICAR DESCRIPCIÓN DE PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá modificar la descripción asociada a la publicación desde el formulario al que ha accedido en RU-43.
<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado el botón de modificar publicación, y visualizar el formulario con el contenido de la publicación que podemos modificar.
<b>Postcondiciones</b>	El usuario vendedor ha accedido al formulario donde se encuentra la información a modificar por la publicación y ha modificado la descripción asociada.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el formulario que contiene la información de la publicación que se puede modificar.</li> <li>2. El usuario selecciona el campo que desea modificar, en este caso la descripción, y escribe la descripción que desee.</li> </ol>

	<ol style="list-style-type: none"> <li>3. El sistema muestra el nuevo contenido escrito y valida que la información introducida por teclado por el usuario cumple las restricciones impuestas.</li> <li>4. El usuario selecciona el botón de confirmar.</li> <li>5. El sistema actualiza la información modificada asociada a esa publicación en la base de datos.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El sistema valida la información introducida y no es válido.               <ol style="list-style-type: none"> <li>3.1 Retorna al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> <li>✓ RU-43</li> <li>✓ RF-76</li> <li>✓ RF-77</li> <li>✓ RF-78</li> <li>✓ RU-45</li> </ul>
<b>Prioridad</b>	Alta

Tabla 80: RU-45 Modificar descripción de publicación

<b>ID y Nombre</b>	<b>RU-46 MODIFICAR CATEGORÍA DE PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá modificar la categoría asociada a la publicación desde el formulario al que ha accedido en RU-43.
<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado el botón de modificar publicación, y visualizar el formulario con el contenido de la publicación que podemos modificar.
<b>Postcondiciones</b>	El usuario vendedor ha accedido al formulario donde se encuentra la información a modificar por la publicación y ha modificado la categoría asociada.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el formulario que contiene la información de la publicación que se puede modificar.</li> <li>2. El usuario selecciona el campo que desea modificar, en este caso la categoría, y selecciona la nueva categoría dentro del menú desplegable.</li> <li>3. El sistema muestra la nueva categoría seleccionada y valida que la categoría corresponde a una categoría incluida en la plataforma.</li> <li>4. El usuario selecciona el botón de confirmar.</li> <li>5. El sistema actualiza la información modificada asociada a esa publicación en la base de datos.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El sistema valida la información introducida y no es válido.               <ol style="list-style-type: none"> <li>3.1 Retorna al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> <li>✓ RU-43</li> </ul>

	<ul style="list-style-type: none"> <li>✓ RF-76</li> <li>✓ RF-77</li> <li>✓ RF-78</li> <li>✓ RU-46</li> </ul>
<b>Prioridad</b>	Alta

Tabla 81: RU-46 Modificar categoría de una publicación

<b>ID y Nombre</b>		<b>RU-47 MODIFICAR SUBCATEGORÍA DE PUBLICACIÓN</b>
<b>Versión</b>		<b>1.0</b>
<b>Actor principal</b>		Usuario vendedor
<b>Descripción</b>		El usuario vendedor podrá modificar la subcategoría asociada a la publicación desde el formulario al que ha accedido en RU-43.
<b>Precondiciones</b>		El usuario vendedor debe haber seleccionado el botón de modificar publicación, y visualizar el formulario con el contenido de la publicación que podemos modificar.
<b>Postcondiciones</b>		El usuario vendedor ha accedido al formulario donde se encuentra la información a modificar por la publicación y ha modificado la subcategoría asociada.
<b>Flujo normal</b>		<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el formulario que contiene la información de la publicación que se puede modificar.</li> <li>2. El usuario selecciona el campo que desea modificar, en este caso la subcategoría, y selecciona la nueva subcategoría dentro del menú desplegable.</li> <li>3. El sistema muestra la nueva subcategoría seleccionada y valida que la subcategoría corresponde a una subcategoría incluida en la plataforma.</li> <li>4. El usuario selecciona el botón de confirmar.</li> <li>5. El sistema actualiza la información modificada asociada a esa publicación en la base de datos.</li> </ol>
<b>Excepciones</b>		<ol style="list-style-type: none"> <li>3. El sistema valida la información introducida y no es válido.               <ol style="list-style-type: none"> <li>3.1 Retorna al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>		<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> <li>✓ RU-43</li> <li>✓ RF-76</li> <li>✓ RF-77</li> <li>✓ RF-78</li> <li>✓ RU-47</li> </ul>
<b>Prioridad</b>		Alta

Tabla 82: RU-47 Modificar subcategoría de publicación

ID y Nombre	RU-48 MODIFICAR ESTADO DE PUBLICACIÓN
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá modificar el estado asociado a la publicación desde el formulario al que ha accedido en RU-43.
Precondiciones	El usuario vendedor debe haber seleccionado el botón de modificar publicación, y visualizar el formulario con el contenido de la publicación que podemos modificar.
Postcondiciones	El usuario vendedor ha accedido al formulario donde se encuentra la información a modificar por la publicación y ha modificado el estado asociado.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el formulario que contiene la información de la publicación que se puede modificar.</li> <li>2. El usuario selecciona el campo que desea modificar, en este caso el estado, y desliza la barra escogiendo el nuevo estado que desee.</li> <li>3. El sistema muestra el nuevo estado seleccionado y valida que el estado corresponde a los de la plataforma.</li> <li>4. El usuario selecciona el botón de confirmar.</li> <li>5. El sistema actualiza la información modificada asociada a esa publicación en la base de datos.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>3. El sistema valida la información introducida y no es válido.               <ol style="list-style-type: none"> <li>3.1 Retorna al punto 1.</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> <li>✓ RU-43</li> <li>✓ RF-76</li> <li>✓ RF-77</li> <li>✓ RF-78</li> <li>✓ RU-48</li> </ul>
Prioridad	Alta

Tabla 83: RU-48 Modificar estado de publicación

ID y Nombre	RU-49 MODIFICAR LA POSIBILIDAD DE ENVÍO DE PUBLICACIÓN
Versión	1.0
Actor principal	Usuario vendedor
Descripción	El usuario vendedor podrá modificar la posibilidad de envío asociada a la publicación desde el formulario al que ha accedido en RU-43.
Precondiciones	1. El usuario vendedor debe haber seleccionado el botón de modificar publicación, y visualizar el formulario con el contenido de la publicación que podemos modificar.
Postcondiciones	1. El usuario vendedor ha accedido al formulario donde se encuentra la información a modificar por la publicación y ha modificado la posibilidad en envío asociada.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el formulario que contiene la información de la publicación que se puede modificar.</li> </ol>

	<ol style="list-style-type: none"> <li>2. El usuario selecciona el campo que desea modificar, en este caso la posibilidad de envío, y cambia el estado de la variable.</li> <li>3. El sistema muestra la posibilidad de envío seleccionada y valida que la posibilidad de envío cumple las restricciones impuestas.</li> <li>4. El usuario selecciona el botón de confirmar.</li> <li>5. El sistema actualiza la información modificada asociada a esa publicación en la base de datos.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El sistema valida la información introducida y no es válido.               <ol style="list-style-type: none"> <li>3.1 Retorna al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> <li>✓ RU-43</li> <li>✓ RF-76</li> <li>✓ RF-77</li> <li>✓ RF-78</li> <li>✓ RU-49</li> </ul>
<b>Prioridad</b>	Alta

Tabla 84: RU-49 Modificar la posibilidad de envío de la publicación

ID y Nombre	
<b>ID y Nombre</b>	<b>RU-50 MODIFICAR PRECIO DE PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá modificar el precio asociado a la publicación desde el formulario al que ha accedido en RU-43.
<b>Precondiciones</b>	El usuario vendedor debe haber seleccionado el botón de modificar publicación, y visualizar el formulario con el contenido de la publicación que podemos modificar.
<b>Postcondiciones</b>	El usuario vendedor ha accedido al formulario donde se encuentra la información a modificar por la publicación y ha modificado el precio asociado.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el formulario que contiene la información de la publicación que se puede modificar.</li> <li>2. El usuario selecciona el campo que desea modificar, en este caso el precio, y escribe la cantidad en euros que desee.</li> <li>3. El sistema muestra la nueva cantidad y valida que la información introducida por teclado por el usuario cumple las restricciones impuestas.</li> <li>4. El usuario selecciona el botón de confirmar.</li> <li>5. El sistema actualiza la información modificada asociada a esa publicación en la base de datos.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El sistema valida la información introducida y no es válido.               <ol style="list-style-type: none"> <li>3.1 Retorna al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-93</li> </ul>

	<ul style="list-style-type: none"> <li>✓ RU-43</li> <li>✓ RF-76</li> <li>✓ RF-77</li> <li>✓ RF-78</li> <li>✓ RU-50</li> </ul>
<b>Prioridad</b>	Alta

Tabla 85: RU-50 Modificar precio de publicación

<b>ID y Nombre</b>	<b>RU-51 VISUALIZAR SISTEMA DE VALORACIÓN PERSONAL</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá visualizar su propio sistema de valoración donde se encuentran las valoraciones del resto de los usuarios clientes.
<b>Precondiciones</b>	El usuario vendedor debe haber visualizado previamente el detalle de una publicación ( RU-37) , y haber seleccionado el cuadro dónde aparece la información sobre su propia valoración.
<b>Postcondiciones</b>	El usuario vendedor ha visualizado su propio sistema de valoración personal junto con las opiniones individuales publicadas por los usuarios clientes.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el cuadro informativo con su información sobre las valoraciones de otros usuarios, dentro de la información detallada de la publicación.</li> <li>2. El sistema muestra su propio sistema de valoración al usuario vendedor.</li> <li>3. El usuario visualiza todas las valoraciones recibidas por el resto de usuarios clientes, junto con su valoración media y su información de usuario.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-80</li> <li>✓ RU-51</li> </ul>
<b>Prioridad</b>	Alta

Tabla 86: RU-51 Visualizar sistema de valoración personal

<b>ID y Nombre</b>	<b>RU-52 VISUALIZAR PUBLICACIONES EN VENTA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario vendedor
<b>Descripción</b>	El usuario vendedor podrá visualizar sus propias publicaciones en venta, desde su propio sistema de valoración donde se encuentran las valoraciones del resto de los usuarios clientes.
<b>Precondiciones</b>	El usuario vendedor debe haber visualizado su propio sistema de valoración previamente ( RU-51) , y haber seleccionado la pestaña de publicaciones en venta.

<b>Postcondiciones</b>	El usuario vendedor ha visualizado sus propias publicaciones a la venta desde su propio sistema de valoración.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la pestaña que indica “En venta”, para mostrar la ventana con publicaciones en venta actualmente.</li> <li>2. El sistema muestra el listado de miniaturas sobre todas las publicaciones en venta del usuario vendedor.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-37</li> <li>✓ RF-80</li> <li>✓ RU-51</li> <li>✓ RF-81</li> <li>✓ RU-52</li> <li>✓ RF-82</li> </ul>
<b>Prioridad</b>	Alta

Tabla 87: RU-52 Visualizar publicaciones en venta

#### 4.9.6 CASO DE USO - USUARIO CLIENTE – GESTIÓN DE PUBLICACIONES

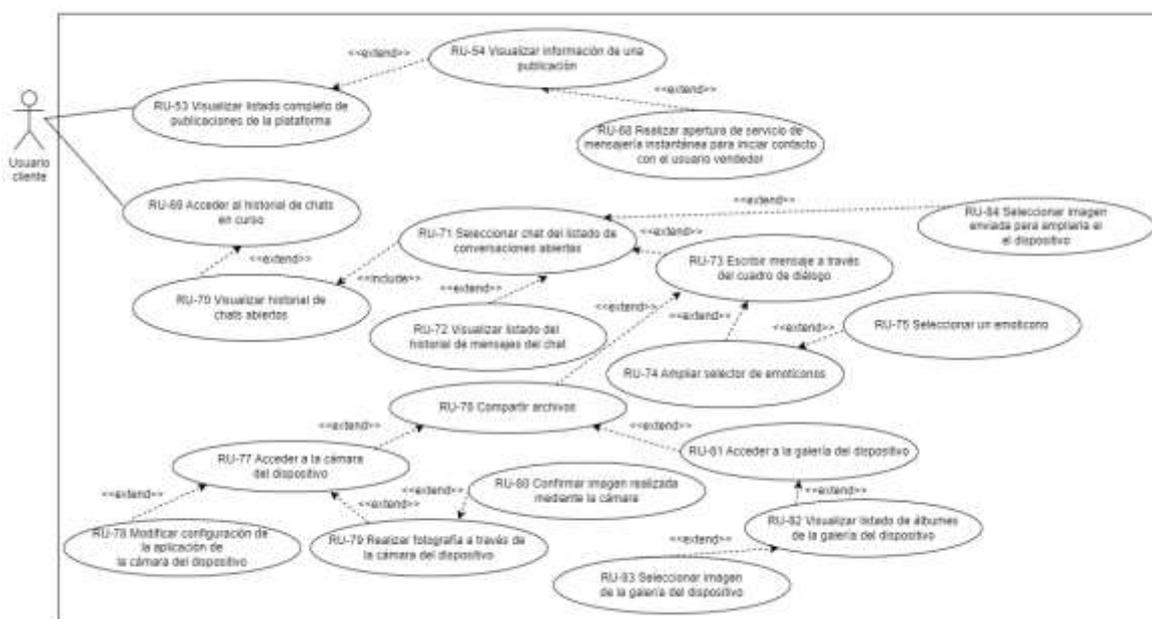


Figura 29: Caso de uso - Usuario cliente – Gestión de publicaciones

<b>ID y Nombre</b>	<b>RU-53 VISUALIZAR LISTADO COMPLETO DE PUBLICACIONES</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado completo de publicaciones realizadas en la plataforma por todos los usuarios vendedores.

<b>Precondiciones</b>	El usuario cliente debe haberse registrado y autenticado en la aplicación previamente.
<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado completo de publicaciones activas en la plataforma por el resto de usuarios.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario se autentica en la aplicación y accede a la página de inicio de la publicación.</li> <li>2. El sistema muestra al usuario un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2. En caso de que el listado de publicaciones se encuentre vacío, el sistema mostrará un mensaje indicando que no hay publicaciones realizadas.</li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-53</li> <li>✓ RF-42</li> <li>✓ RF-43</li> </ul>
<b>Prioridad</b>	Alta

Tabla 88: RU-53 Visualizar listado completo de publicaciones

<b>RU-54 VISUALIZAR INFORMACIÓN DE UNA PUBLICACIÓN</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el contenido al detalle de una publicación realizada al seleccionar la miniatura de la publicación.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones de la plataforma y seleccionado una de las miniaturas, RU-53.
<b>Postcondiciones</b>	El usuario cliente ha visualizado el detalle de una publicación realizada en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> <li>2. El usuario visualiza el listado completo de publicaciones de la plataforma y escoge una de las miniaturas.</li> <li>3. El sistema redirige al usuario a otra pantalla donde puede visualizar al detalle la publicación.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-53</li> <li>✓ RF-42</li> <li>✓ RF-43</li> <li>✓ RU-54</li> <li>✓ RF-44</li> </ul>
<b>Prioridad</b>	Alta

Tabla 89: RU-54 Visualizar información de una publicación

ID y Nombre	RU-55 ACCEDER AL MAPA CON LA UBICACIÓN
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá acceder al mapa con la ubicación de la publicación.
Precondiciones	El usuario cliente debe haber realizado la publicación previamente en la plataforma, y visualizado al detalle, RU-54.
Postcondiciones	El usuario cliente ha visualizado el mapa con la ubicación relacionada con la publicación realizada en la plataforma.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el mapa con la ubicación relacionada con la publicación, mediante un marcador y el mapa de la API de Google Maps.</li> <li>2. El usuario selecciona el mapa para ampliarlo a pantalla completa en el dispositivo.</li> <li>3. El sistema muestra el mapa a pantalla completa junto la ubicación actual del usuario y la ubicación de la publicación.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-54</li> <li>✓ RF-93</li> <li>✓ RU-55</li> <li>✓ RF-94</li> </ul>
Prioridad	Alta

Tabla 90: RU-55 Acceder al mapa con la ubicación

ID y Nombre	RU-56 SELECCIONAR BOTÓN PARA REDIRIGIR A UBICACIÓN ACTUAL
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá redirigir a la posición actual del usuario en el mapa con la ubicación de la publicación.
Precondiciones	El usuario cliente debe haber visualizado previamente el mapa con la ubicación actual, RU-55.
Postcondiciones	El usuario cliente ha redirigido en el mapa a la ubicación actual del usuario activo mediante su dispositivo móvil.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el mapa con la ubicación relacionada con la publicación, mediante un marcador y el mapa de la API de Google Maps.</li> <li>2. El usuario selecciona el botón que permite redirigir la pantalla a la ubicación actual del usuario que está visualizando el mapa.</li> <li>3. El sistema muestra el mapa a pantalla completa junto la ubicación actual del usuario activo.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-54</li> <li>✓ RF-93</li> <li>✓ RU-55</li> </ul>

	<ul style="list-style-type: none"> <li>✓ RF-94</li> <li>✓ RF-95</li> <li>✓ RF-96</li> <li>✓ RF-97</li> </ul>
<b>Prioridad</b>	Alta

Tabla 91: RU-56 Seleccionar botón para redirigir a ubicación actual

<b>ID y Nombre</b>	<b>RU-57 VISUALIZAR IMÁGENES DE UNA PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar las imágenes insertadas en la publicación que permiten una información visual del producto.
<b>Precondiciones</b>	El usuario cliente debe haber realizado la publicación previamente en la plataforma, y visualizado al detalle, RU-54.
<b>Postcondiciones</b>	El usuario cliente ha visualizado el carrusel de imágenes asociadas a la publicación.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario toda la información al detalle de la publicación seleccionada previamente en el listado, RU-54.</li> <li>2. El usuario visualiza el carrusel con las imágenes asociadas a la publicación e interactúa con él.</li> <li>3. El sistema muestra al usuario el carrusel con todas las imágenes relacionadas con la publicación.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El sistema no muestra ninguna imagen dado que no hay ninguna imagen relacionada con la publicación               <ol style="list-style-type: none"> <li>3.1 El sistema muestra un mensaje informativo</li> <li>3.2 Retorno al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-54</li> <li>✓ RF-93</li> <li>✓ RU-58</li> <li>✓ RF-68</li> </ul>
<b>Prioridad</b>	Alta

Tabla 92: RU-57 Visualizar imágenes de una publicación

<b>ID y Nombre</b>	<b>RU-58 ACCEDER AL SISTEMA DE VALORACIÓN ENTRE USUARIOS</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá acceder al sistema de valoración del usuario vendedor propietario de la publicación.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado la información de una publicación previamente.
<b>Postcondiciones</b>	El usuario cliente ha accedido al sistema de valoración entre usuarios del propietario de la publicación que se encontraba visualizando.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario toda la información al detalle de la publicación seleccionada previamente en el listado, RU-54.</li> </ol>

	<ol style="list-style-type: none"> <li>El usuario visualiza la información del usuario vendedor y lo selecciona.</li> <li>El sistema muestra al usuario cliente toda la información sobre las valoraciones correspondientes al propietario de la publicación.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-54</li> <li>✓ RF-93</li> <li>✓ RU-58</li> <li>✓ RF-80</li> </ul>
<b>Prioridad</b>	Alta

Tabla 93: RU-58 Acceder al sistema de valoración entre usuarios

<b>ID y Nombre</b>	<b>RU-59 VISUALIZAR INFORMACIÓN DE UN VENDEDOR</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar la información del usuario vendedor propietario de la publicación.
<b>Precondiciones</b>	El usuario cliente debe haber accedido al sistema de valoración del usuario vendedor previamente. (RU-58)
<b>Postcondiciones</b>	El usuario cliente ha visualizado la información del usuario vendedor en la vista de su sistema de valoración
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>El sistema muestra al usuario cliente toda la información del sistema de valoración del usuario vendedor propietario de la publicación.</li> <li>El usuario visualiza la información del usuario vendedor, junto con su nombre, valoración y media aritmética de todas las valoraciones.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-54</li> <li>✓ RF-93</li> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-59</li> <li>✓ RF-81</li> </ul>
<b>Prioridad</b>	Alta

Tabla 94: RU-59 Visualizar información de un vendedor

<b>ID y Nombre</b>	<b>RU-60 VISUALIZAR LISTADO DE VALORACIONES DEL USUARIO VENDEDOR</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado de valoraciones realizadas por los usuarios clientes al usuario vendedor propietario de la publicación.
<b>Precondiciones</b>	El usuario cliente debe haber accedido al sistema de valoración del usuario vendedor previamente. (RU-58)
<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado de valoraciones realizadas al usuario vendedor propietario de la publicación

<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la pestaña “Opiniones” desde el sistema de valoración correspondiente.</li> <li>2. El sistema muestra al usuario cliente el listado de publicaciones realizadas hacía el usuario vendedor propietario de la publicación.</li> <li>3. El usuario visualiza el listado completo de todas las valoraciones realizadas hacía el usuario vendedor propietario de la publicación.</li> </ol>
<b>Excepciones</b>	3. El sistema muestra un mensaje informativo en caso de que no existan publicaciones realizadas asociadas con ese usuario vendedor.
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-59</li> <li>✓ RF-81</li> <li>✓ RU-60</li> <li>✓ RF-83</li> </ul>
<b>Prioridad</b>	Alta

Tabla 95: RU-60 Visualizar listado de valoraciones del usuario vendedor

<b>ID y Nombre</b>	<b>RU-61 VISUALIZAR VALORACIÓN A UN USUARIO VENDEDOR</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar una valoración realizada por los usuarios clientes al usuario vendedor propietario de la publicación.
<b>Precondiciones</b>	El usuario cliente debe haber accedido al sistema de valoración del usuario vendedor previamente (RU-58) y visualizado el listado completo de valoraciones (RU-60).
<b>Postcondiciones</b>	El usuario cliente ha visualizado el contenido de una valoración donde se muestra el nombre del usuario cliente que ha publicado la valoración, la valoración dada y la fecha de la valoración.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado completo de todas las valoraciones realizadas hacía el usuario vendedor propietario de la publicación.</li> <li>2. El sistema devuelve la información de una valoración.</li> <li>3. El usuario puede visualizar la información de una valoración realizada previamente.</li> </ol>
<b>Excepciones</b>	1. El sistema muestra un mensaje informativo en caso de que no existan publicaciones realizadas asociadas con ese usuario vendedor.
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-59</li> <li>✓ RF-81</li> <li>✓ RU-61</li> <li>✓ RF-83</li> </ul>
<b>Prioridad</b>	Alta

Tabla 96: RU-61 Visualizar valoración a un usuario vendedor

<b>ID y Nombre</b>		<b>RU-62 AÑADIR VALORACIÓN A UN USUARIO VENDEDOR</b>
<b>Versión</b>	<b>1.0</b>	
<b>Actor principal</b>	Usuario cliente	
<b>Descripción</b>	El usuario cliente podrá añadir una valoración al usuario vendedor propietario de la publicación.	
<b>Precondiciones</b>	El usuario cliente debe haber accedido al sistema de valoración del usuario vendedor previamente (RU-58) y visualizado el listado completo de valoraciones (RU-60).	
<b>Postcondiciones</b>	El usuario cliente ha añadido una valoración al usuario vendedor propietario de la publicación dentro del sistema de valoración del usuario vendedor.	
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado completo de todas las valoraciones realizadas hacia el usuario vendedor propietario de la publicación.</li> <li>2. El sistema muestra el botón flotante que permite añadir una valoración.</li> <li>3. El usuario selecciona el botón de “Añadir valoración”.</li> <li>4. El sistema muestra al usuario el cuadro de diálogo junto con el formulario que permite añadir la información asociada a una valoración.</li> <li>5. El sistema obtiene los datos introducidos por el usuario y los valida.</li> <li>6. El usuario pulsa el botón de “Enviar”.</li> <li>7. El sistema almacena la información asociada a la valoración en la base de datos.</li> </ol>	
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2. El sistema valida los datos introducidos, y estos no cumplen las restricciones impuestas. <ol style="list-style-type: none"> <li>2.1 El sistema muestra un mensaje de error</li> <li>2.2 El flujo retorna al punto 2.</li> </ol> </li> </ol>	
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-62</li> <li>✓ RF-84</li> </ul>	
<b>Prioridad</b>	Alta	

*Tabla 97: RU-62 Añadir valoración a un usuario vendedor*

<b>ID y Nombre</b>		<b>RU-63 MODIFICAR COMENTARIO DE UNA VALORACIÓN</b>
<b>Versión</b>	<b>1.0</b>	
<b>Actor principal</b>	Usuario cliente	
<b>Descripción</b>	El usuario cliente podrá modificar el comentario de una valoración realizada previamente al usuario vendedor propietario de la publicación.	
<b>Precondiciones</b>	El usuario cliente debe haber accedido al sistema de valoración del usuario vendedor previamente (RU-58) y añadido una valoración al usuario vendedor (RU-62).	

<b>Postcondiciones</b>	El usuario cliente ha modificado el comentario asociado a una valoración realizada al usuario vendedor propietario de la publicación.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado completo de todas las valoraciones realizadas hacia el usuario vendedor propietario de la publicación.</li> <li>2. El sistema muestra el botón que permite desplegar el menú donde dispone de la opción de modificar una valoración.</li> <li>3. El usuario selecciona el botón de “Editar opinión”.</li> <li>4. El sistema muestra el formulario en una nueva ventana emergente con la información de la valoración, y ofrece a escribir el nuevo comentario que se desee.</li> <li>5. El usuario escribe el nuevo comentario y selecciona el botón de enviar.</li> <li>6. El sistema valida la información introducida y almacena la nueva información en la base de datos.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>6. El sistema valida los datos introducidos, y estos no cumplen las restricciones impuestas. <ol style="list-style-type: none"> <li>a. El sistema muestra un mensaje de error</li> <li>b. El flujo retorna al punto 2.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-63</li> <li>✓ RF-87</li> <li>✓ RF-88</li> <li>✓ RF-89</li> </ul>
<b>Prioridad</b>	Alta

Tabla 98: RU-63 Modificar comentario de una valoración

<b>ID y Nombre</b>	<b>RU-64 MODIFICAR PUNTUACIÓN DE UNA VALORACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá modificar la puntuación de una valoración realizada previamente al usuario vendedor propietario de la publicación.
<b>Precondiciones</b>	El usuario cliente debe haber accedido al sistema de valoración del usuario vendedor previamente (RU-58) y añadido una valoración al usuario vendedor (RU-62).
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente ha modificado la puntuación asociada a una valoración realizada al usuario vendedor propietario de la publicación.</li> </ol>

<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado completo de todas las valoraciones realizadas hacía el usuario vendedor propietario de la publicación.</li> <li>2. El sistema muestra el botón que permite desplegar el menú donde dispone de la opción de modificar una valoración.</li> <li>3. El usuario selecciona el botón de “Editar opinión”.</li> <li>4. El sistema muestra el formulario en una nueva ventana emergente con la información de la valoración, y ofrece a escribir la nueva puntuación que se desee.</li> <li>5. El usuario escribe la nueva puntuación y selecciona el botón de enviar.</li> <li>6. El sistema valida la información introducida y almacena la nueva información en la base de datos.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>7. El sistema valida los datos introducidos, y estos no cumplen las restricciones impuestas. <ol style="list-style-type: none"> <li>a. El sistema muestra un mensaje de error</li> <li>b. El flujo retorna al punto 2.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-64</li> <li>✓ RF-87</li> <li>✓ RF-88</li> <li>✓ RF-89</li> </ul>
<b>Prioridad</b>	Alta

Tabla 99: RU-64 Modificar puntuación de valoración

<b>ID y Nombre</b>	<b>RU-65 CANCELAR PROCESO DE AÑADIR UNA VALORACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá cancelar el proceso de añadir una valoración al usuario vendedor propietario de la publicación durante el proceso.
<b>Precondiciones</b>	El usuario cliente debe haber accedido al sistema de valoración del usuario vendedor previamente (RU-58) y encontrarse en el proceso de añadir valoración al usuario vendedor (RU-62).
<b>Postcondiciones</b>	El usuario cliente ha modificado ha cancelado el proceso de “añadir una valoración” y la información almacenada en memoria ha sido eliminada.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el botón flotante de “Añadir una valoración” y realiza la apertura del formulario para insertar una nueva valoración.</li> <li>2. El sistema muestra el cuadro de dialogo con el formulario donde se debe introducir la información de la valoración.</li> <li>3. El usuario selecciona fuera del cuadro de dialogo.</li> <li>4. El sistema detecta que el proceso de añadir una valoración ha sido cancelado, y elimina toda la información de la valoración guardada en memoria.</li> </ol>

<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-65</li> <li>✓ RF-93</li> </ul>
<b>Prioridad</b>	Alta

Tabla 100: RU-65 Cancelar proceso de añadir una valoración

<b>ID y Nombre</b>		<b>RU-66 ELIMINAR VALORACIÓN REALIZADA</b>
<b>Versión</b>		<b>1.0</b>
<b>Actor principal</b>		Usuario cliente
<b>Descripción</b>		El usuario cliente podrá eliminar una valoración realizada previamente al usuario vendedor propietario de la publicación.
<b>Precondiciones</b>		El usuario cliente debe haber accedido al sistema de valoración del usuario vendedor previamente (RU-58) y añadido una valoración al usuario vendedor (RU-62).
<b>Postcondiciones</b>		El usuario cliente ha eliminado una valoración realizada al usuario vendedor propietario de la publicación.
<b>Flujo normal</b>		<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado completo de todas las valoraciones realizadas hacía el usuario vendedor propietario de la publicación.</li> <li>2. El sistema muestra el botón que permite desplegar el menú donde dispone de la opción de modificar una valoración.</li> <li>3. El usuario selecciona el botón de "Eliminar opinión".</li> <li>4. El sistema solicita al usuario mediante un cuadro de dialogo si está seguro de que desea eliminar la valoración.</li> <li>5. El usuario selecciona el botón de "Confirmar".</li> <li>6. El sistema valida de que valoración se trata y elimina toda la información relacionada con la valoración en la base de datos.</li> </ol>
<b>Excepciones</b>		<ol style="list-style-type: none"> <li>5. El usuario selecciona el botón de "Cancelar"</li> <li>5.1 El flujo retorna al punto 1.</li> </ol>
<b>Trazabilidad</b>		<ul style="list-style-type: none"> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-66</li> <li>✓ RF-90</li> <li>✓ RF-91</li> <li>✓ RF-92</li> </ul>
<b>Prioridad</b>		Alta

Tabla 101: RU-66 Eliminar valoración realizada

<b>ID y Nombre</b>		<b>RU-67 VISUALIZAR LISTADO PUBLICACIONES EN VENTA</b>
<b>Versión</b>		<b>1.0</b>
<b>Actor principal</b>		Usuario cliente

<b>Descripción</b>	El usuario cliente podrá visualizar sus propias publicaciones en venta, desde un sistema de valoración donde se encuentran las valoraciones del resto de los usuarios clientes.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado un sistema de valoración previamente ( RU-51 ) , y haber seleccionado la pestaña de publicaciones en venta.
<b>Postcondiciones</b>	El usuario cliente ha visualizado las publicaciones a la venta desde el sistema de valoración del usuario vendedor.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la pestaña que indica “En venta”, para mostrar la ventana con publicaciones en venta actualmente.</li> <li>2. El sistema muestra el listado de miniaturas sobre todas las publicaciones en venta del usuario vendedor.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-58</li> <li>✓ RF-80</li> <li>✓ RU-67</li> <li>✓ RF-82</li> </ul>
<b>Prioridad</b>	Alta

Tabla 102: RU-67 Visualizar listado de publicaciones en venta

#### 4.9.7. CASO DE USO - USUARIO CLIENTE – GESTIÓN DE SERVICIO DE MENSAJERÍA INSTANTÁNEA

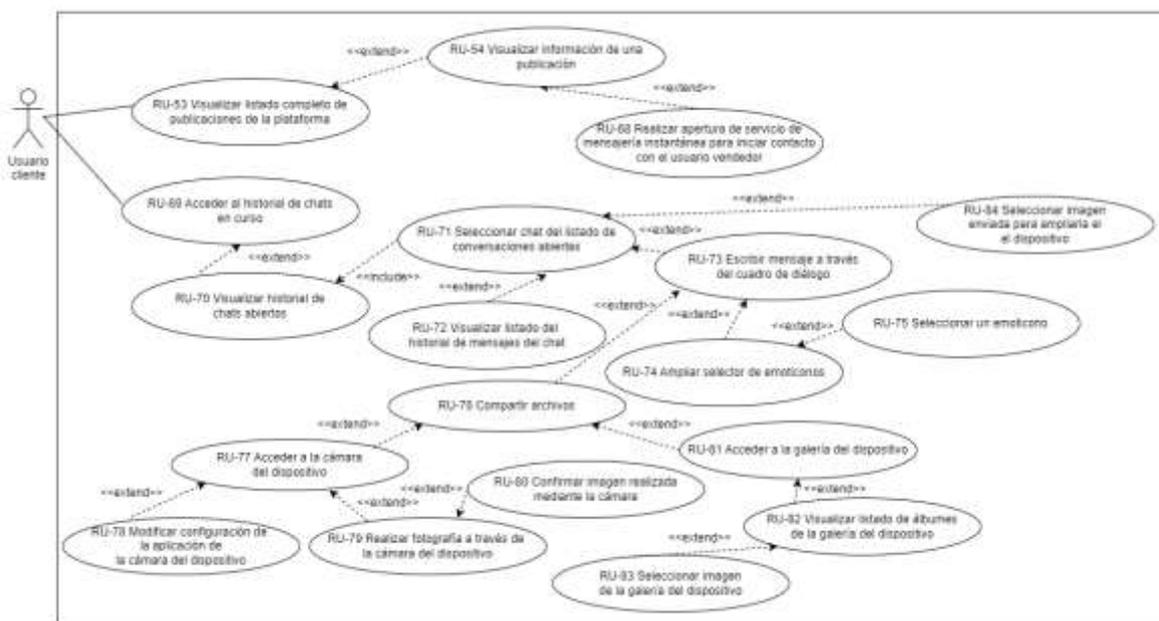


Figura 30: Caso de Uso - Usuario cliente - Gestión de servicio de mensajería instantánea

ID y Nombre	RU-68 ACCEDER A SERVICIO DE MENSAJERÍA INSTANTÁNEA
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá realizar la apertura de una nueva conversación con el usuario vendedor propietario de la publicación.
Precondiciones	El usuario cliente debe haber visualizado la información de una publicación previamente (RU-54).
Postcondiciones	El usuario cliente ha realizado la apertura de una nueva conversación o ha accedido a la conversación activa con el usuario vendedor.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario toda la información al detalle de la publicación seleccionada previamente en el listado, RU-54.</li> <li>2. El usuario visualiza la información al detalle de la publicación, y selecciona el botón flotante que permite realizar la apertura de una conversación mediante el sistema de mensajería instantánea o acceder a la conversación activa con el usuario vendedor.</li> <li>3. El sistema muestra al usuario la conversación del chat con el usuario vendedor propietario de la publicación.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-54</li> <li>✓ RF-42</li> <li>✓ RU-68</li> <li>✓ RF-98</li> </ul>
Prioridad	Alta

Tabla 103: RU-68 Acceder a servicio de mensajería instantánea

ID y Nombre	RU-69 ACCEDER A HISTORIAL DE MENSAJERÍA INSTANTÁNEA
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá acceder al historial de mensajería instantánea, seleccionando la opción “Chats” desde el menú de opciones del inicio de la plataforma.
Precondiciones	El usuario cliente debe estar registrado e autenticado previamente.
Postcondiciones	El usuario cliente ha accedido al historial del servicio de mensajería instantánea de sus propias conversaciones activas.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario la opción “chats” desde el menú de inicio de la aplicación.</li> <li>2. El usuario selecciona la opción “Chats” desde el menú de inicio de la aplicación.</li> <li>3. El sistema muestra al usuario el listado de chats activos actualmente mediante el servicio de mensajería instantánea.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-54</li> <li>✓ RF-42</li> </ul>

	<ul style="list-style-type: none"> <li>✓ RU-68</li> <li>✓ RF-99</li> <li>✓ RF-100</li> </ul>
<b>Prioridad</b>	Alta

Tabla 104: RU-69 Acceder al historial de mensajería instantánea

<b>ID y Nombre</b>	<b>RU-70 VISUALIZAR HISTORIAL DE CHATS ABIERTOS</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá acceder al historial de mensajería instantánea, seleccionando la opción “Chats” desde el menú de opciones del inicio de la plataforma.
<b>Precondiciones</b>	El usuario cliente debe estar registrado e autenticado previamente.
<b>Postcondiciones</b>	El usuario cliente ha accedido al historial del servicio de mensajería instantánea de sus propias conversaciones activas.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario la opción “chats” desde el menú de inicio de la aplicación.</li> <li>2. El usuario selecciona la opción “Chats” desde el menú de inicio de la aplicación.</li> <li>3. El sistema muestra al usuario el listado de chats activos actualmente mediante el servicio de mensajería instantánea.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-54</li> <li>✓ RF-42</li> <li>✓ RU-69</li> <li>✓ RF-99</li> <li>✓ RF-100</li> <li>✓ RF-101</li> </ul>
<b>Prioridad</b>	Alta

Tabla 105: RU-70 Visualizar historial de chats abiertos

<b>ID y Nombre</b>	<b>RU-71 SELECCIONAR CONVERSACIÓN DEL CHAT</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar un chat desde el listado de conversaciones de mensajería instantánea que ha activado previamente.
<b>Precondiciones</b>	El usuario cliente debe acceder al historial de mensajería instantánea, seleccionando la opción “Chats” desde el menú de opciones del inicio de la plataforma.
<b>Postcondiciones</b>	El usuario cliente ha seleccionado una conversación del historial del servicio de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra al usuario el listado de chats activos actualmente mediante el servicio de mensajería instantánea.</li> </ol>

	<ol style="list-style-type: none"> <li>El usuario visualiza la conversación que desea seleccionar y lo selecciona.</li> <li>El sistema abre la conversación de chat a pantalla completa.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-69</li> <li>✓ RF-99</li> <li>✓ RF-100</li> <li>✓ RF-101</li> <li>✓ RF-102</li> </ul>
<b>Prioridad</b>	Alta

Tabla 106: RU-71 Seleccionar conversación de chat

<b>ID y Nombre</b>	<b>RU-72 VISUALIZAR LISTADO DE MENSAJES DEL CHAT</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado de mensajes que componen el historial de la conversación en el chat.
<b>Precondiciones</b>	El usuario cliente debe acceder al historial de mensajería instantánea, seleccionando la opción “Chats” desde el menú de opciones del inicio de la plataforma.
<b>Postcondiciones</b>	El usuario cliente ha visualizado los mensajes de una conversación del historial del servicio de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>El sistema muestra al usuario el listado de chats activos actualmente mediante el servicio de mensajería instantánea.</li> <li>El usuario selecciona la conversación que desea abrir.</li> <li>El sistema abre la conversación de chat a pantalla completa y muestra el listado de mensajes del historial del chat.</li> <li>El usuario visualiza el listado de mensajes del chat.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-69</li> <li>✓ RU-71</li> <li>✓ RF-100</li> <li>✓ RF-101</li> <li>✓ RF-102</li> <li>✓ RF-103</li> <li>✓ RF-104</li> </ul>
<b>Prioridad</b>	Alta

Tabla 107: RU-72 Visualizar listado de mensajes del chat

ID y Nombre	RU-73 ESCRIBIR MENSAJE
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá escribir un mensaje en el cuadro de dialogo del chat desde el historial de la conversación en el chat.
Precondiciones	El usuario cliente debe visualizar el historial de mensajes del chat previamente (RU-72).
Postcondiciones	El usuario cliente ha enviado un mensaje a través del cuadro de diálogo del chat al otro usuario con el que mantiene abierta la conversación.
Flujo normal	<ol style="list-style-type: none"> <li>1. El sistema abre la conversación de chat a pantalla completa y muestra el listado de mensajes del historial del chat.</li> <li>2. El usuario visualiza el listado de mensajes del chat y selecciona el cuadro de dialogo de la parte inferior de la vista.</li> <li>3. El sistema muestra el teclado del dispositivo móvil.</li> <li>4. El usuario escribe el contenido en texto que desea enviar u otras funcionalidades.</li> <li>5. El sistema valida la información introducida y almacena el contenido del mensaje en la base de datos.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>5. El sistema valida la información introducida y no cumple las restricciones impuestas. <ol style="list-style-type: none"> <li>5.1 El flujo retorna al punto 3</li> </ol> </li> </ol>
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-71</li> <li>✓ RF-103</li> <li>✓ RF-104</li> <li>✓ RU-73</li> <li>✓ RF-105</li> </ul>
Prioridad	Alta

Tabla 108: RU-73 Escribir mensaje

ID y Nombre	RU-74 AMPLIAR SELECTOR DE EMOTICONOS
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá ampliar el selector de emoticonos incluidos en el cuadro del dialogo del chat.
Precondiciones	El usuario cliente debe visualizar el historial de mensajes del chat previamente (RU-72).
Postcondiciones	El usuario cliente ha realizado la apertura del selector de emoticonos, a través del cuadro de diálogo del chat.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado de mensajes del chat y selecciona el cuadro de dialogo de la parte inferior de la vista.</li> <li>2. El sistema muestra el teclado del dispositivo móvil.</li> <li>3. El usuario selecciona el selector de emoticonos.</li> </ol>

	4. El sistema muestra en nueva ventana el contenido del selector de emoticonos.
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-103</li> <li>✓ RU-73</li> <li>✓ RF-104</li> <li>✓ RF-74</li> <li>✓ RF-106</li> </ul>
<b>Prioridad</b>	Alta

Tabla 109: RU-74 Ampliar selector de emoticonos

<b>ID y Nombre</b>	<b>RU-75 SELECCIONAR UN EMOTICONO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar un emoticono del selector de emoticonos incluidos en el cuadro del dialogo del chat.
<b>Precondiciones</b>	El usuario cliente debe visualizar el selector de emoticonos previamente (RU-74).
<b>Postcondiciones</b>	El usuario cliente ha seleccionado un emoticono del selector de emoticonos, a través del cuadro de diálogo del chat.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado de emoticonos del selector.</li> <li>2. El sistema muestra todos los emoticonos disponibles, organizados por categorías.</li> <li>3. El usuario selecciona el emoticono que desee enviar.</li> <li>4. El sistema muestra en el cuadro de dialogo el emoticono seleccionado.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-103</li> <li>✓ RU-73</li> <li>✓ RF-104</li> <li>✓ RU-74</li> <li>✓ RF-106</li> <li>✓ RF-107</li> </ul>
<b>Prioridad</b>	Alta

Tabla 110: RU-75 Seleccionar un emoticono

<b>ID y Nombre</b>	<b>RU-76 COMPARTIR ARCHIVO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá compartir un archivo a través del cuadro de diálogo del chat.

<b>Precondiciones</b>	El usuario cliente debe visualizar el historial de mensajes del chat previamente (RU-73).
<b>Postcondiciones</b>	El usuario cliente ha realizado la apertura del selector de tipo de archivo que desea compartir con el usuario vendedor, a través del cuadro de diálogo del chat.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el cuadro de diálogo del chat.</li> <li>2. El sistema muestra el teclado del dispositivo móvil, junto con el botón que permite seleccionar la opción de “Compartir archivo”.</li> <li>3. El usuario selecciona el botón “Compartir archivo”.</li> <li>4. El sistema muestra en nueva ventana el tipo de archivos que puede compartir el usuario, imágenes desde la galería del dispositivo, imágenes desde la cámara del dispositivo e ubicación.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-103</li> <li>✓ RU-73</li> <li>✓ RF-104</li> <li>✓ RU-76</li> <li>✓ RF-110</li> </ul>
<b>Prioridad</b>	Alta

Tabla 111: RU-76 Compartir archivo

<b>ID y Nombre</b>	<b>RU-77 ACCEDER A LA CÁMARA DEL DISPOSITIVO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá acceder a la aplicación de la cámara del dispositivo móvil en el que está implementando la plataforma.
<b>Precondiciones</b>	El usuario cliente debe visualizar previamente la ventana donde aparecen los tipos de archivos a compartir (RU-75).
<b>Postcondiciones</b>	El usuario cliente ha realizado la apertura de la aplicación de la cámara del dispositivo.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el botón “Compartir archivo”.</li> <li>2. El sistema muestra en nueva ventana el tipo de archivos que puede compartir el usuario, imágenes desde la galería del dispositivo, imágenes desde la cámara del dispositivo e ubicación.</li> <li>3. El usuario selecciona la opción “Cámara”.</li> <li>4. El sistema muestra la aplicación que gestiona la cámara del dispositivo móvil para realizar la fotografía que desee el usuario.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>4. El sistema no puede mostrar la aplicación de la cámara del dispositivo por fallo en la misma. <ul style="list-style-type: none"> <li>4.1 Se retorna al punto 1.</li> </ul> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-76</li> <li>✓ RF-109</li> <li>✓ RU-77</li> <li>✓ RF-111</li> </ul>
<b>Prioridad</b>	Alta

Tabla 112: RU-77 Acceder a la cámara del dispositivo

ID y Nombre	RU-78 MODIFICAR CONFIGURACIÓN DE LA CÁMARA DEL DISPOSITIVO
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá modificar la configuración de la aplicación de la cámara del dispositivo móvil en el que está implementando la plataforma.
Precondiciones	El usuario cliente debe realizar la apertura de la aplicación de la cámara previamente (RU-77).
Postcondiciones	El usuario cliente ha escogido la configuración deseada en la aplicación de la cámara del dispositivo.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario cliente ha realizado la apertura de la aplicación de la cámara del dispositivo.</li> <li>2. El usuario selecciona las distintas opciones que ofrece la aplicación de la cámara del dispositivo.</li> <li>3. El sistema muestra las modificaciones que ha escogido el usuario sobre la configuración que desea antes de tomar la fotografía con el dispositivo.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-76</li> <li>✓ RF-109</li> <li>✓ RU-77</li> <li>✓ RF-111</li> <li>✓ RF-112</li> </ul>
Prioridad	Alta

Tabla 113 RU-78 Modificar configuración de la cámara del dispositivo

ID y Nombre	RU-79 REALIZAR FOTOGRAFÍA DE LA CÁMARA DEL DISPOSITIVO
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá realizar una fotografía a través de la aplicación de la cámara del dispositivo móvil en el que está implementando la plataforma.
Precondiciones	El usuario cliente debe realizar la apertura de la aplicación de la cámara previamente (RU-77) y escogido la configuración (RU-78).
Postcondiciones	El usuario cliente ha escogido la configuración deseada en la aplicación de la cámara del dispositivo.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario cliente ha realizado la apertura de la aplicación de la cámara del dispositivo.</li> <li>2. El sistema muestra lo que visualiza la cámara del dispositivo, ya sea tanto la cámara trasera como delantera.</li> <li>3. El usuario selecciona el botón para realizar la fotografía.</li> <li>4. El sistema muestra al usuario la fotografía realizada.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-76</li> <li>✓ RF-109</li> <li>✓ RU-77</li> </ul>

	<ul style="list-style-type: none"> <li>✓ RF-111</li> <li>✓ RF-112</li> <li>✓ RF-113</li> </ul>
<b>Prioridad</b>	Alta

Tabla 114: RU-79 Realizar fotografía de la cámara del dispositivo

<b>ID y Nombre</b>	<b>RU-80 CONFIRMAR IMAGEN REALIZADA MEDIANTE LA CÁMARA DEL DISPOSITIVO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá confirmar una fotografía realizada a través de la aplicación de la cámara del dispositivo móvil en el que está implementando la plataforma.
<b>Precondiciones</b>	El usuario cliente debe realizar la apertura de la aplicación de la cámara previamente (RU-77) y realizado la fotografía (RU-79).
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente ha confirmado y compartido la fotografía realizada la aplicación de la cámara del dispositivo.</li> <li>2. El usuario cliente añade una/s imagen/es a través de la aplicación de cámara del dispositivo a la publicación.</li> </ol>
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el botón para realizar la fotografía.</li> <li>2. El sistema muestra al usuario la fotografía realizada.</li> <li>3. El usuario confirma mediante el botón confirmar que desea compartir la imagen capturada mediante la cámara.</li> <li>4. El sistema envía al otro usuario la imagen realizada y la almacena en la base de datos.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-76</li> <li>✓ RF-109</li> <li>✓ RU-77</li> <li>✓ RF-111</li> <li>✓ RU-78</li> <li>✓ RF-112</li> <li>✓ RU-79</li> <li>✓ RF-113</li> <li>✓ RU-80</li> <li>✓ RF-114</li> <li>✓ RF-115</li> <li>✓ RF-116</li> </ul>
<b>Prioridad</b>	Alta

Tabla 115: RU-80 Confirmar imagen realizada mediante la cámara del dispositivo

ID y Nombre	RU-81 ACCEDER A LA GALERÍA DEL DISPOSITIVO
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá acceder a la aplicación de la galería del dispositivo móvil en el que está implementando la plataforma.
Precondiciones	El usuario cliente debe visualizar previamente la ventana donde aparecen los tipos de archivos a compartir (RU-75).
Postcondiciones	El usuario cliente ha realizado la apertura de la aplicación de la galería del dispositivo.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona el botón “Compartir archivo”.</li> <li>2. El sistema muestra en nueva ventana el tipo de archivos que puede compartir el usuario, imágenes desde la galería del dispositivo, imágenes desde la cámara del dispositivo e ubicación.</li> <li>3. El usuario selecciona la opción “Galería”.</li> <li>4. El sistema muestra la aplicación que gestiona la galería del dispositivo móvil para seleccionar la imagen/es que desee el usuario.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>4. El sistema no muestra ninguna imagen porque la galería del dispositivo no contiene ninguna imagen. <ol style="list-style-type: none"> <li>4.1 – El sistema muestra un mensaje de error</li> <li>4.2 – El flujo de eventos vuelve al punto 1</li> </ol> </li> </ol>
	<ul style="list-style-type: none"> <li>✓ RU-76</li> <li>✓ RF-109</li> <li>✓ RU-81</li> <li>✓ RF-118</li> </ul>
Prioridad	Alta

Tabla 116: RU-81 Acceder a la galería del dispositivo

ID y Nombre	RU-82 VISUALIZAR LISTADO DE ALBÚMES DE LA GALERÍA DEL DISPOSITIVO
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá visualizar los álbumes de la aplicación de la galería del dispositivo móvil en el que está implementando la plataforma.
Precondiciones	El usuario cliente debe seleccionar previamente la opción donde se accede a la galería del dispositivo (RU-82).
Postcondiciones	El usuario cliente ha realizado la apertura de la aplicación de la galería del dispositivo y ha visualizado los álbumes.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Galería”.</li> <li>2. El sistema muestra la aplicación que gestiona la galería del dispositivo móvil para seleccionar la imagen/es que desee el usuario, categorizadas por álbumes que tenga creados el usuario previamente.</li> </ol>

	3. El usuario selecciona la/s imagen/es desde los álbumes que desea enviar.
<b>Excepciones</b>	4. El sistema no muestra ninguna imagen porque la galería del dispositivo no contiene ninguna imagen. 4.1 – El sistema muestra un mensaje de error 4.2 – El flujo de eventos vuelve al punto 1
<b>Trazabilidad</b>	✓ RU-76 ✓ RF-109 ✓ RU-81 ✓ RF-118 ✓ RU-82 ✓ RF-119
<b>Prioridad</b>	Alta

Tabla 117: RU-82 Visualizar listado de álbumes de la galería del dispositivo

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-83 SELECCIONAR IMAGEN DE LA GALERÍA DEL DISPOSITIVO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar una imagen desde la galería del dispositivo móvil en el que está implementando la plataforma.
<b>Precondiciones</b>	El usuario cliente debe seleccionar previamente la opción donde se accede a la galería del dispositivo (RU-82) y a visualizar los álbumes (RU-82).
<b>Postcondiciones</b>	El usuario cliente ha realizado la apertura de la aplicación de la galería del dispositivo y ha seleccionado una/ imagen/es.
<b>Flujo normal</b>	1. El usuario selecciona la opción “Galería”. 2. El sistema muestra la aplicación que gestiona la galería del dispositivo móvil para seleccionar la imagen/es que desee el usuario, categorizadas por álbumes que tenga creados el usuario previamente. 3. El usuario selecciona la/s imagen/es desde los álbumes que desea enviar.
<b>Excepciones</b>	4. El sistema no muestra ninguna imagen porque la galería del dispositivo no contiene ninguna imagen. a. – El sistema muestra un mensaje de error 4.2 – El flujo de eventos vuelve al punto 1
<b>Trazabilidad</b>	✓ RU-76 ✓ RF-109 ✓ RU-81 ✓ RF-118 ✓ RU-82 ✓ RF-119 ✓ RF-120 ✓ RF-121
<b>Prioridad</b>	Alta

Tabla 118 RU-83 Seleccionar imagen de la galería del dispositivo

<b>ID y Nombre</b>	<b>RU-84 SELECCIONAR IMAGEN ENVIADA PARA AMPLIARLA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar una imagen enviada desde el historial de mensajes del chat para ampliarla a pantalla completa en el dispositivo móvil.
<b>Precondiciones</b>	El usuario cliente debe visualizar el historial de mensajes del chat previamente (RU-73).
<b>Postcondiciones</b>	El usuario cliente ha realizado la apertura de la imagen seleccionada para poder visualizarla a pantalla completa en el dispositivo.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona un mensaje cuyo contenido sea una imagen enviada previamente.</li> <li>2. El sistema muestra la imagen seleccionada a pantalla completa, ofreciendo al usuario visualizar la imagen con más detalle e incluso realizar gestos, como zoom para facilitar su visión.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU- 81</li> <li>✓ RU-71</li> <li>✓ RF-100</li> <li>✓ RF-101</li> <li>✓ RF-102</li> <li>✓ RF-123</li> <li>✓ RF-124</li> </ul>
<b>Prioridad</b>	Alta

*Tabla 119: RU-84 Seleccionar imagen enviada para ampliarla*

#### 4.9.8. CASO DE USO - USUARIO CLIENTE – GESTIÓN SISTEMA DE SEGUIMIENTO EN TIEMPO REAL

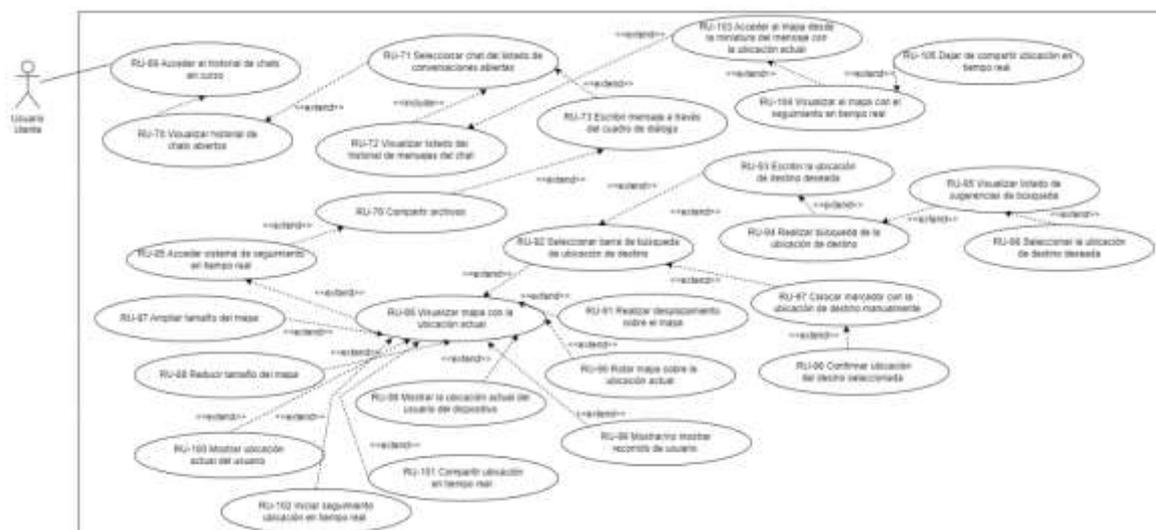


Figura 31: Caso de Uso - Usuario cliente - Gestión de sistema de seguimiento en tiempo real

ID y Nombre	RU-85 ACCEDER A SISTEMA DE SEGUIMIENTO EN TIEMPO REAL
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá acceder al sistema de seguimiento en tiempo real, desde la opción “Compartir archivos” integrada en el servicio de mensajería instantánea.
Precondiciones	El usuario cliente debe visualizar previamente la ventana donde aparecen los tipos de archivos a compartir (RU-76).
Postcondiciones	El usuario cliente ha realizado la apertura del sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona el botón “Compartir archivo”.</li> <li>2. El sistema muestra en nueva ventana el tipo de archivos que puede compartir el usuario, imágenes desde la galería del dispositivo, imágenes desde la cámara del dispositivo e ubicación.</li> <li>3. El usuario selecciona la opción “Ubicación”.</li> <li>4. El sistema muestra el sistema de seguimiento en tiempo real, a través del mapa integrado mediante las APIs de Google Maps y MapBox al usuario cliente.</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>4. El sistema no muestra ningún mapa integrado a través de la API de Google Maps. <ol style="list-style-type: none"> <li>4.1 – El sistema muestra un mensaje de error</li> <li>4.2 – El flujo de eventos vuelve al punto 1.</li> </ol> </li> </ol>

<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-76</li> <li>✓ RF-109</li> <li>✓ RU-85</li> <li>✓ RF-125</li> </ul>
<b>Prioridad</b>	Alta

Figura 120: RU-85 Acceder a sistema de seguimiento en tiempo real

<b>ID y Nombre</b>		<b>RU-86 VISUALIZAR MAPA CON UBICACIÓN ACTUAL</b>
<b>Versión</b>		<b>1.0</b>
<b>Actor principal</b>		Usuario cliente
<b>Descripción</b>		El usuario cliente podrá visualizar el mapa con la ubicación actual, al seleccionar la opción “Ubicación” desde el cuadro de dialogo de compartir archivos.
<b>Precondiciones</b>		El usuario cliente debe acceder previamente al sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-85).
<b>Postcondiciones</b>		El usuario cliente ha visualizado el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
<b>Flujo normal</b>		<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Ubicación”.</li> <li>2. El sistema muestra el sistema de seguimiento en tiempo real, a través del mapa integrado mediante las APIs de Google Maps y MapBox al usuario cliente.</li> <li>3. El usuario visualiza el mapa con la ubicación actual del usuario del dispositivo móvil.</li> </ol>
<b>Excepciones</b>		<ol style="list-style-type: none"> <li>4. El sistema no muestra ningún mapa integrado a través de la API de Google Maps. <ul style="list-style-type: none"> <li>4.1– El sistema muestra un mensaje de error</li> <li>4.2 – El flujo de eventos vuelve al punto 1</li> </ul> </li> </ol>
<b>Trazabilidad</b>		<ul style="list-style-type: none"> <li>✓ RU-76</li> <li>✓ RF-109</li> <li>✓ RU-85</li> <li>✓ RF-125</li> <li>✓ RU-86</li> <li>✓ RF-126</li> </ul>
<b>Prioridad</b>		Alta

Tabla 121: RU-86 Visualizar mapa con ubicación actual

<b>ID y Nombre</b>		<b>RU-87 AMPLIAR TAMAÑO DEL MAPA</b>
<b>Versión</b>		<b>1.0</b>
<b>Actor principal</b>		Usuario cliente
<b>Descripción</b>		El usuario cliente podrá ampliar el tamaño del mapa con la ubicación actual, realizando el gesto de ampliar en la pantalla del dispositivo móvil.

<b>Precondiciones</b>	El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).
<b>Postcondiciones</b>	El usuario cliente ha ampliado el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el mapa con la ubicación actual del usuario del dispositivo móvil.</li> <li>2. El sistema muestra el mapa y permite al usuario desplazarse sobre el mapa.</li> <li>3. El usuario realiza el gesto de ampliar el tamaño del mapa.</li> <li>4. El sistema muestra la imagen ampliada del mapa seleccionado.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-85</li> <li>✓ RF-125</li> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-87</li> <li>✓ RF-127</li> </ul>
<b>Prioridad</b>	Alta

Tabla 122: RU-87 Ampliar tamaño del mapa

<b>ID y Nombre</b>	<b>RU-88 REDUCIR TAMAÑO DEL MAPA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá reducir el tamaño del mapa con la ubicación actual, realizando el gesto de reducir en la pantalla del dispositivo móvil.
<b>Precondiciones</b>	El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).
<b>Postcondiciones</b>	El usuario cliente ha reducido el tamaño del mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el mapa con la ubicación actual del usuario del dispositivo móvil.</li> <li>2. El sistema muestra el mapa y permite al usuario desplazarse sobre el mapa.</li> <li>3. El usuario realiza el gesto de reducir el tamaño del mapa.</li> <li>4. El sistema muestra la imagen reducida del mapa seleccionado.</li> </ol>
<b>Excepciones</b>	-

<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-85</li> <li>✓ RF-125</li> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-88</li> <li>✓ RF-127</li> </ul>
<b>Prioridad</b>	Alta

Tabla 123: RU-88 Reducir tamaño del mapa

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-89 MOSTRAR UBICACIÓN ACTUAL EN EL MAPA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar la ubicación actual del usuario en el mapa que pertenece al sistema de seguimiento en tiempo real.
<b>Precondiciones</b>	El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).
<b>Postcondiciones</b>	El usuario cliente ha visualizado el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el mapa con la ubicación actual del usuario del dispositivo móvil.</li> <li>2. El sistema muestra el mapa y permite al usuario desplazarse sobre el mapa.</li> <li>3. El usuario selecciona el botón que permite visualizar su ubicación actual en el mapa.</li> <li>4. El sistema muestra la imagen de la ubicación actual en el mapa.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-85</li> <li>✓ RF-125</li> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-89</li> <li>✓ RF-128</li> </ul>
<b>Prioridad</b>	Alta

Tabla 124: RU-89 Mostrar ubicación actual en el mapa

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-90 ROTAR EL MAPA SOBRE LA UBICACIÓN ACTUAL</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá rotar el mapa sobre la posición actual en la que se encuentra, realizando el gesto de rotar en la pantalla del dispositivo móvil.

<b>Precondiciones</b>	El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).
<b>Postcondiciones</b>	El usuario cliente ha rotado la posición del mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el mapa con la ubicación actual del usuario del dispositivo móvil.</li> <li>2. El sistema muestra el mapa y permite al usuario desplazarse sobre el mapa.</li> <li>3. El usuario realiza el gesto de rotar el mapa sobre la posición en la que se encuentra actualmente.</li> <li>4. El sistema muestra la imagen rotada de posición del mapa seleccionado.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-85</li> <li>✓ RF-125</li> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-90</li> <li>✓ RF-129</li> </ul>
<b>Prioridad</b>	Alta

Tabla 125: RU-90 Rotar mapa sobre ubicación actual

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-91 REALIZAR UN DESPLAZAMIENTO SOBRE EL MAPA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá realizar un desplazamiento sobre el mapa con la ubicación actual, realizando el gesto de desplazarse en la pantalla del dispositivo móvil.
<b>Precondiciones</b>	El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).
<b>Postcondiciones</b>	El usuario cliente se ha desplazado sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el mapa con la ubicación actual del usuario del dispositivo móvil.</li> <li>2. El sistema muestra el mapa y permite al usuario desplazarse sobre el mapa.</li> <li>3. El usuario realiza el gesto de desplazarse sobre la pantalla del dispositivo.</li> <li>4. El sistema muestra una nuevo fragmento del mapa seleccionado.</li> </ol>
<b>Excepciones</b>	-

<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-85</li> <li>✓ RF-125</li> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-91</li> <li>✓ RF-131</li> </ul>
<b>Prioridad</b>	Alta

Tabla 126: RU-91 Realizar un desplazamiento sobre el mapa

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-92 SELECCIONAR BARRA DE BÚSQUEDA INCLUIDA EN EL MAPA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar la barra de búsqueda incluida para seleccionar la ubicación de destino en el mapa con la ubicación actual.
<b>Precondiciones</b>	El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).
<b>Postcondiciones</b>	El usuario cliente ha seleccionado la barra de búsqueda que permite realizar la búsqueda de ubicación de destino deseada sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el mapa con la ubicación actual del usuario del dispositivo móvil.</li> <li>2. El sistema muestra el mapa junto con la barra de búsqueda que permite buscar la ubicación de destino.</li> <li>3. El usuario selecciona la barra de búsqueda.</li> <li>4. El sistema despliega una nueva vista, donde se visualiza la barra de búsqueda junto con la opción de colocar ubicación manualmente.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-85</li> <li>✓ RF-125</li> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-92</li> <li>✓ RF-131</li> <li>✓ RF-132</li> </ul>
<b>Prioridad</b>	Alta

Tabla 127: RU-92 Seleccionar barra de búsqueda incluida en el mapa

ID y Nombre	RU-93 ESCRIBIR UBICACIÓN DE DESTINO EN LA BARRA DE BÚSQUEDA
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá escribir en el cuadro de texto de la barra de búsqueda incluida para seleccionar la ubicación de destino en el mapa con la ubicación actual.
Precondiciones	El usuario cliente debe seleccionar la barra de búsqueda que permite realizar la búsqueda de ubicación de destino deseada sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea. (RU-92)
Postcondiciones	El usuario cliente ha escrito en la barra de búsqueda la ubicación de destino deseada.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la barra de búsqueda y escribe la ubicación de destino deseada, a través del teclado del dispositivo.</li> <li>2. El sistema muestra al usuario el contenido que ha escrito previamente.</li> <li>3. El usuario confirma el texto escrito como ubicación de destino.</li> </ol>
Excepciones	-
Trazabilidad	<ul style="list-style-type: none"> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-92</li> <li>✓ RF-131</li> <li>✓ RF-132</li> <li>✓ RU-93</li> <li>✓ RF-133</li> </ul>
Prioridad	Alta

Tabla 128: RU-93 Escribir ubicación de destino en la barra de búsqueda

ID y Nombre	RU-94 REALIZAR BÚSQUEDA DE LA UBICACIÓN DE DESTINO EN LA BARRA DE BÚSQUEDA.
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá realizar la búsqueda que ha escrito en el cuadro de texto de la barra de búsqueda incluida para seleccionar la ubicación de destino en el mapa con la ubicación actual.
Precondiciones	<ol style="list-style-type: none"> <li>1. El usuario cliente debe seleccionar la barra de búsqueda que permite realizar la búsqueda de ubicación de destino deseada sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea. (RU-92)</li> <li>2. El usuario debe haber escrito la ubicación previamente en la barra de búsqueda (RU-93).</li> </ol>
Postcondiciones	El usuario cliente ha realizado la búsqueda sobre la ubicación destino escrita en la barra de búsqueda.

<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la barra de búsqueda y escribe la ubicación de destino deseada, a través del teclado del dispositivo.</li> <li>2. El sistema muestra al usuario el contenido que ha escrito previamente.</li> <li>3. El usuario confirma el texto escrito como ubicación de destino.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-92</li> <li>✓ RF-131</li> <li>✓ RF-132</li> <li>✓ RU-93</li> <li>✓ RF-133</li> <li>✓ RU-94</li> </ul>
<b>Prioridad</b>	Alta

Tabla 129: RU-94 Realizar búsqueda de la ubicación de destino en la barra de búsqueda.

<b>ID y Nombre</b>	<b>RU-95 VISUALIZAR LISTADO DE SUGERENCIAS EN LA BARRA DE BÚSQUEDA.</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado de sugerencias sobre la ubicación de destino que ofrece sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe seleccionar la barra de búsqueda que permite realizar la búsqueda de ubicación de destino deseada sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea. (RU-92).</li> <li>2. El usuario debe haber realizado la búsqueda de la ubicación previamente en la barra de búsqueda (RU-94).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado de sugerencias sobre la ubicación destino escrita en la barra de búsqueda.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la barra de búsqueda y escribe la ubicación de destino deseada, a través del teclado del dispositivo.</li> <li>2. El sistema muestra al usuario el contenido que ha escrito previamente.</li> <li>3. El usuario confirma el texto escrito como ubicación de destino.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-92</li> <li>✓ RF-131</li> <li>✓ RF-132</li> <li>✓ RU-93</li> <li>✓ RF-133</li> <li>✓ RU-94</li> </ul>

	✓ RF-134 ✓ RU-95
<b>Prioridad</b>	Alta

Tabla 130: RU-95 Visualizar listado de sugerencias en la barra de búsqueda

<b>ID y Nombre</b>	<b>RU-96 SELECCIONAR UBICACIÓN DE DESTINO DESEADA.</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar la ubicación de destino deseada a partir del listado de sugerencias que muestra la barra de búsqueda.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe seleccionar la barra de búsqueda que permite realizar la búsqueda de ubicación de destino deseada sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea. (RU-92).</li> <li>2. El usuario debe haber visualizado el listado de sugerencias que ofrece la barra de búsqueda previamente (RU-95).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha seleccionado una ubicación del listado de sugerencias sobre la ubicación destino escrita en la barra de búsqueda.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la barra de búsqueda y escribe la ubicación de destino deseada, a través del teclado del dispositivo.</li> <li>2. El sistema muestra al usuario el contenido que ha escrito previamente.</li> <li>3. El usuario confirma el texto escrito como ubicación de destino.</li> <li>4. El sistema muestra un listado de sugerencias según el texto escrito por el usuario.</li> <li>5. El usuario selecciona la ubicación de destino deseada.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-93</li> <li>✓ RF-133</li> <li>✓ RU-94</li> <li>✓ RF-134</li> <li>✓ RU-95</li> <li>✓ RF-135</li> <li>✓ RU-96</li> </ul>
<b>Prioridad</b>	Alta

Tabla 131: RU-96 Seleccionar ubicación de destino deseada

<b>ID y Nombre</b>	<b>RU-97 COLOCAR MARCADOR CON LA UBICACIÓN DE DESTINO MANUALMENTE</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá situar un marcador de forma manual que indique la ubicación de destino, sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea.

<b>Precondiciones</b>	1. El usuario cliente debe seleccionar la barra de búsqueda que permite realizar la búsqueda de ubicación de destino deseada sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea. (RU-92).
<b>Postcondiciones</b>	El usuario cliente ha situado un marcador sobre el mapa, indicando la ubicación de destino deseada.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la barra de búsqueda, y visualiza la vista de la barra de búsqueda al completo.</li> <li>2. El sistema muestra la opción de “Colocar la ubicación manualmente”.</li> <li>3. El usuario selecciona la opción de “Colocar ubicación manualmente”.</li> <li>4. El sistema muestra el mapa junto con el marcador que nos permite colocar donde deseemos que se encuentre nuestra ubicación de destino.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-92</li> <li>✓ RF-131</li> <li>✓ RF-132</li> <li>✓ RU-97</li> <li>✓ RF-136</li> </ul>
<b>Prioridad</b>	Alta

Tabla 132: RU-97 Colocar marcador con la ubicación de destino manualmente

<b>ID y Nombre</b>	<b>RU-98 CONFIRMAR UBICACIÓN DE DESTINO SELECCIONADA MEDIANTE EL MARCADOR</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá confirmar la ubicación seleccionada mediante el marcador situado previamente sobre el mapa con la ubicación actual.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe seleccionar la barra de búsqueda que permite realizar la búsqueda de ubicación de destino deseada sobre el mapa con la ubicación actual que pertenece al sistema de seguimiento en tiempo real integrado en el sistema de mensajería instantánea. (RU-92).</li> <li>2. El usuario debe haber colocado un marcador sobre el mapa que indique la ubicación de destino deseada (RU-97).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha confirmado el marcador que ha sido situado previamente sobre el mapa, indicando la ubicación de destino deseada.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de “Colocar ubicación manualmente”.</li> <li>2. El sistema muestra el mapa junto con el marcador que nos permite colocar donde deseemos que se encuentre nuestra ubicación de destino.</li> <li>3. El usuario selecciona el botón de “Confirmar destino” para indicar que la ubicación escogida mediante el marcador es la deseada.</li> </ol>

	4. El sistema realiza el cálculo de la ruta mediante la ubicación inicial y de destino, y traza la ruta entre ambos puntos y muestra diversa información asociada a la ruta sobre el mapa.
<b>Excepciones</b>	-
	<ul style="list-style-type: none"> <li>✓ RU-92</li> <li>✓ RF-131</li> <li>✓ RF-132</li> <li>✓ RU-97</li> <li>✓ RF-136</li> <li>✓ RU-98</li> <li>✓ RF-137</li> <li>✓ RF-138</li> <li>✓ RF-139</li> <li>✓ RF-140</li> </ul>
<b>Prioridad</b>	Alta

Tabla 133: RU-98 Confirmar ubicación de destino seleccionada mediante el marcador

<b>ID y Nombre</b>	<b>RU-99 MOSTRAR/NO MOSTRAR RECORRIDO TRAZADO EN LA RUTA POR EL USUARIO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá escoger si mostrar/no mostrar el recorrido que sigue el usuario sobre la ruta traza previamente entre la ubicación inicial y destino seleccionadas anteriormente.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).</li> <li>2. El usuario debe haber escogido una ubicación de destino mediante la barra de búsqueda, ya sea por búsqueda mediante la barra de búsqueda (RU-96) o colocado manualmente (RU-98) y el sistema haber trazado la ruta.</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha escogido entre mostrar o no mostrar la ruta trazada entre ambas ubicaciones y que muestra el recorrido seguido por el usuario.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el menú radial que aparece sobre el mapa que muestra la ruta trazada entre ambas ubicaciones.</li> <li>2. El sistema muestra las opciones disponibles, entre ellas la opción de “mostrar/no mostrar el recorrido del usuario”.</li> <li>3. El usuario selecciona la opción “mostrar/no mostrar trazado”, dependiendo de la opción que se encuentre activa actualmente.</li> <li>4. El sistema muestra el trazado o no en el mapa, dependiendo de si se muestra o no actualmente, y lo renderiza en el mapa.</li> </ol>
<b>Excepciones</b>	-

<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-99</li> <li>✓ RF-141</li> <li>✓ RF-142</li> </ul>
<b>Prioridad</b>	Alta

Tabla 134: RU-99 Mostrar/no mostrar recorrido trazado en la ruta por el usuario

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-100 SELECCIONAR BOTÓN QUE REDIRIJA A LA UBICACIÓN ACTUAL</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar el botón que redirige a la ubicación actual donde se encuentra el usuario autenticado en el dispositivo móvil.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).</li> <li>2. El usuario debe haber escogido una ubicación de destino mediante la barra de búsqueda, ya sea por búsqueda mediante la barra de búsqueda (RU-96) o colocado manualmente (RU-98) y el sistema haber trazado la ruta.</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha seleccionado el botón que redirige a la posición actual del usuario autenticado, y se muestra la ubicación en el mapa.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el menú radial que aparece sobre el mapa que muestra la ruta trazada entre ambas ubicaciones.</li> <li>2. El sistema muestra las opciones disponibles, entre ellas la opción de “Redirigir a la ubicación actual del usuario”.</li> <li>3. El usuario selecciona la opción “Redirigir a la ubicación actual del usuario”.</li> <li>4. El sistema muestra la ubicación actual del usuario mediante la ubicación del dispositivo en el mapa y centra la posición de forma central en la pantalla.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-99</li> <li>✓ RF-141</li> <li>✓ RF-143</li> </ul>
<b>Prioridad</b>	Alta

Tabla 135: RU-100 Seleccionar botón que redirija a la ubicación actual

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-101 COMPARTIR UBICACIÓN EN TIEMPO REAL</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente

<b>Descripción</b>	El usuario cliente podrá compartir la ubicación en tiempo real mediante el sistema de seguimiento al usuario sobre el mapa, donde se muestra el recorrido que ha realizado el usuario.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).</li> <li>2. El usuario debe haber escogido una ubicación de destino mediante la barra de búsqueda, ya sea por búsqueda mediante la barra de búsqueda (RU-96) o colocado manualmente (RU-98) y el sistema haber trazado la ruta.</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha compartido su ubicación en tiempo real con el usuario mediante la conversación abierta en el sistema de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el menú radial que aparece sobre el mapa que muestra la ruta trazada entre ambas ubicaciones.</li> <li>2. El sistema muestra las opciones disponibles, entre ellas la opción de “Compartir ubicación”.</li> <li>3. El usuario selecciona la opción “Compartir ubicación”.</li> <li>4. El sistema comparte al otro usuario la localización del usuario en tiempo real, así como su recorrido, su ubicación inicial y su ubicación de destino mediante la conversación activa en el servicio de mensajería instantánea.</li> <li>5. El usuario visualiza la información de la ubicación compartida a través del mapa en miniatura en el chat.</li> <li>6. El sistema almacena las coordenadas en la base de datos, actualizándolas cada periodo de 2 segundos.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-99</li> <li>✓ RF-141</li> <li>✓ RF-145</li> <li>✓ RF-146</li> <li>✓ RF-147</li> </ul>
<b>Prioridad</b>	Alta

Tabla 136: RU-101 Compartir ubicación actual

<b>ID y Nombre</b>	<b>RU-102 INICIAR SEGUIMIENTO EN TIEMPO REAL</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá iniciar el seguimiento en tiempo real de su propia ubicación para que el mapa que muestra su ubicación actual se desplace según va recorriendo la ruta trazada.

<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe visualizar previamente el mapa con la ubicación actual que pertenece a sistema de seguimiento en tiempo real, a través del servicio de mensajería instantánea (RU-86).</li> <li>2. El usuario debe haber escogido una ubicación de destino mediante la barra de búsqueda, ya sea por búsqueda mediante la barra de búsqueda (RU-96) o colocado manualmente (RU-98) y el sistema haber trazado la ruta.</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha compartido su ubicación en tiempo real con el usuario mediante la conversación abierta en el sistema de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el menú radial que aparece sobre el mapa que muestra la ruta trazada entre ambas ubicaciones.</li> <li>2. El sistema muestra las opciones disponibles, entre ellas la opción de “Iniciar seguimiento en tiempo real”.</li> <li>3. El usuario selecciona la opción “Iniciar seguimiento en tiempo real”.</li> <li>4. El sistema inicia el seguimiento del usuario según se desplaza por el recorrido marcado en el trazado del mapa.</li> <li>5. El usuario visualiza el seguimiento que se está realizando según se desplaza con su dispositivo móvil.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-86</li> <li>✓ RF-126</li> <li>✓ RU-99</li> <li>✓ RF-141</li> <li>✓ RF-144</li> </ul>
<b>Prioridad</b>	Alta

Tabla 137: RU-102 Iniciar seguimiento en tiempo real

<b>ID y Nombre</b>	<b>RU-103 ACCEDER A MINIATURA MAPA CON LA UBICACIÓN ACTUAL</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá acceder al mapa donde el otro usuario o él mismo, ha compartido la ubicación actual y se realiza el seguimiento en tiempo real de la ubicación, donde se muestra el trazado que recorre el usuario.
<b>Precondiciones</b>	El usuario cliente debe visualizar el historial de mensajes del chat previamente (RU-73).
<b>Postcondiciones</b>	El usuario cliente ha accedido al mapa que contiene la ubicación compartida por el otro usuario o por sí mismo por medio del sistema de seguimiento en tiempo real.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualizado el listado con el historial de chats de la conversación abierta.</li> <li>2. El sistema muestra ventanas por cada mensaje junto con su contenido enviado.</li> <li>3. El usuario selecciona el mensaje que contiene la miniatura del mapa.</li> </ol>

	4. El sistema muestra el mapa a pantalla completa en el dispositivo móvil, junto con la ubicación del usuario que ha compartido la ubicación y toda la información asociada.
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-72</li> <li>✓ RF-100</li> <li>✓ RF-101</li> <li>✓ RF-102</li> <li>✓ RF-103</li> <li>✓ RF-104</li> <li>✓ RU-103</li> <li>✓ RF-148</li> </ul>
<b>Prioridad</b>	Alta

Tabla 138: RU-103 Acceder a mapa con la ubicación actual

<b>ID y Nombre</b>	<b>RU-104 VISUALIZAR EL MAPA CON EL SEGUIMIENTO EN TIEMPO REAL</b>
<b>Versión</b>	1.0
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el mapa al completo junto con sus funcionalidades asociadas, donde el otro usuario o él mismo, ha compartido la ubicación actual y se realiza el seguimiento en tiempo real de la ubicación, donde se muestra el trazado que recorre el usuario.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe visualizar el historial de mensajes del chat previamente (RU-73).</li> <li>2. El usuario debe haber accedido al mapa con la ubicación actual compartida (RU-103).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha visualizado el mapa que contiene la ubicación compartida por el otro usuario o por sí mismo por medio del sistema de seguimiento en tiempo real.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el mensaje que contiene la miniatura del mapa compartido entre los usuarios de la conversación activa del chat.</li> <li>2. El sistema muestra el mapa a pantalla completa en el dispositivo móvil, junto con la ubicación del usuario que ha compartido la ubicación y toda la información asociada.</li> <li>3. El usuario interactúa con el mapa como desee, realizando los movimientos y funcionalidades implementadas.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-72</li> <li>✓ RF-100</li> <li>✓ RF-101</li> <li>✓ RF-102</li> <li>✓ RF-103</li> </ul>

	<ul style="list-style-type: none"> <li>✓ RF-104</li> <li>✓ RU-103</li> <li>✓ RF-148</li> <li>✓ RF-149</li> <li>✓ RF-150</li> </ul>
<b>Prioridad</b>	Alta

Tabla 139: RU-104 Visualizar el mapa con el seguimiento en tiempo real

<b>RU-105 DEJAR DE COMPARTIR UBICACIÓN EN TIEMPO REAL</b>	
<b>ID y Nombre</b>	<b>RU-105 DEJAR DE COMPARTIR UBICACIÓN EN TIEMPO REAL</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá dejar de compartir la ubicación compartida en tiempo real con el otro usuario con el que mantiene abierto el chat al que ha accedido.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe visualizar el historial de mensajes del chat previamente (RU-73).</li> <li>2. El usuario debe haber accedido al mapa con la ubicación actual compartida (RU-104).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha dejado de compartir la ubicación en tiempo real con el otro usuario con el que mantiene activo el chat mediante el servicio de mensajería instantánea.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el mensaje que contiene la miniatura del mapa.</li> <li>2. El sistema muestra el mapa a pantalla completa en el dispositivo móvil, junto con la ubicación del usuario que ha compartido la ubicación y toda la información asociada.</li> <li>3. El usuario visualiza el botón que indica “Cancelar seguimiento”, en el caso de que la miniatura que está visualizando se trate de una ubicación que ha compartido él.</li> <li>4. El sistema recibe la orden del usuario, y pregunta mediante una nueva ventana emergente si realmente quiere dejar de compartir esa ubicación en tiempo real.</li> <li>5. El usuario deja de compartir la ubicación, por lo que el otro usuario no puede visualizar su ubicación actual.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-103</li> <li>✓ RF-148</li> <li>✓ RF-149</li> <li>✓ RF-151</li> <li>✓ RF-152</li> <li>✓ RF-153</li> </ul>
<b>Prioridad</b>	Alta

Tabla 140: RU-105 Dejar de compartir ubicación en tiempo real

#### 4.9.9 CASO DE USO - USUARIO CLIENTE - GESTIÓN DE PUBLICACIONES

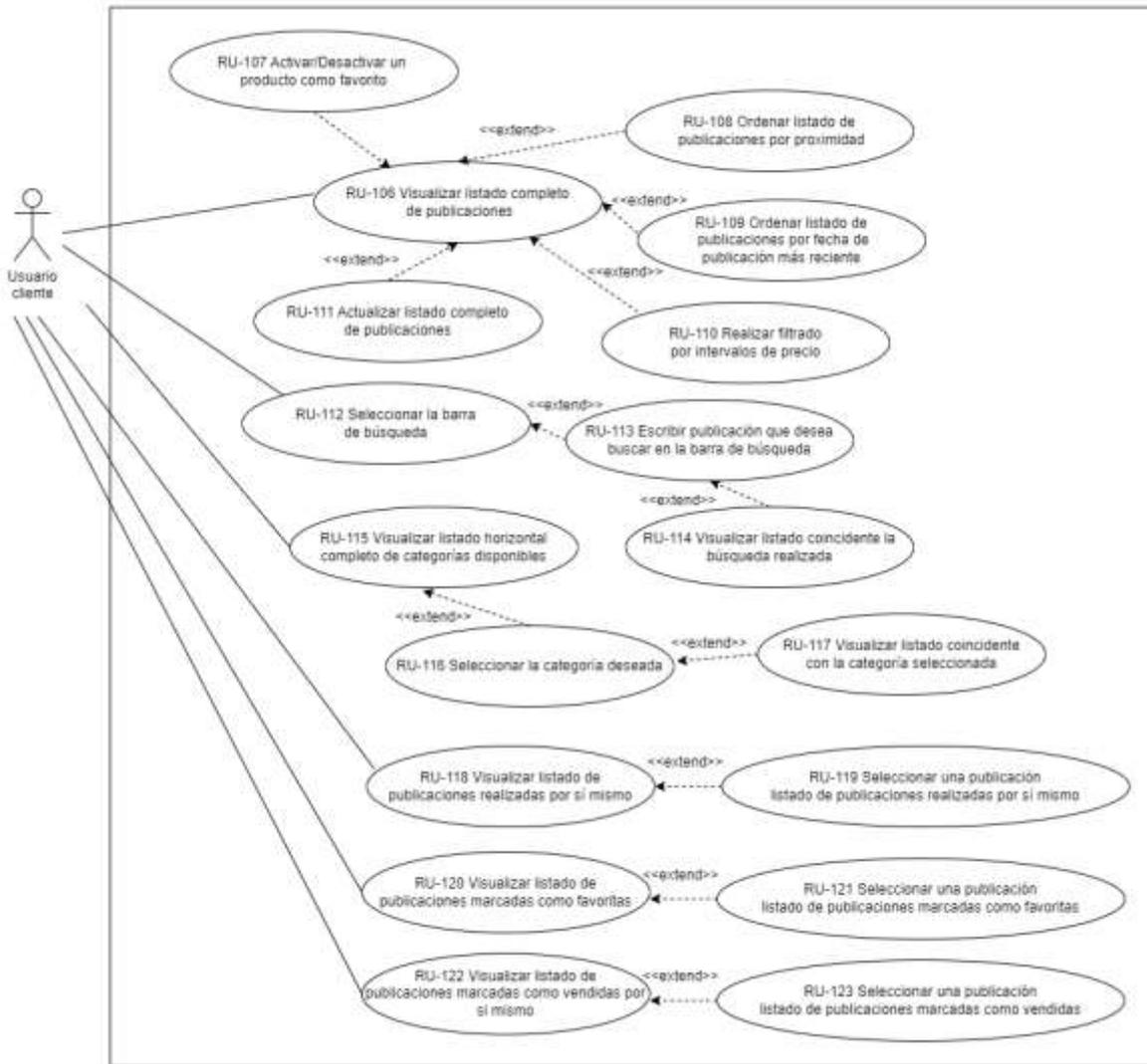


Figura 32: Caso de Uso - Usuario cliente - Gestión de publicaciones

ID y Nombre	RU-106 VISUALIZAR LISTADO COMPLETO DE PUBLICACIONES
Versión	1.0
Actor principal	Usuario cliente
Descripción	El usuario cliente podrá visualizar el listado completo de publicaciones realizadas en la plataforma por todos los usuarios activos.
Precondiciones	El usuario cliente debe haberse registrado y autenticado en la aplicación previamente.
Postcondiciones	El usuario cliente ha visualizado el listado completo de publicaciones activas en la plataforma por el resto de usuarios.
Flujo normal	1. El usuario se autentica en la aplicación y accede a la página de inicio de la publicación.

	2. El sistema muestra al usuario un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.
<b>Excepciones</b>	En caso de que el listado de publicaciones se encuentre vacío, el sistema mostrará un mensaje indicando que no hay publicaciones realizadas.
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-106</li> <li>✓ RF-42</li> <li>✓ RF-43</li> </ul>
<b>Prioridad</b>	Alta

Tabla 141: RU-106 Visualizar listado completo de publicaciones

<b>ID y Nombre</b>	<b>RU-107 ACTIVAR/DESACTIVAR PUBLICACIÓN COMO FAVORITA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá activar o desactivar una publicación como favorita.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario vendedor ha marcado o desactivado una publicación como favorita.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> <li>2. El sistema muestra en cada miniatura correspondiente a una publicación, un botón que permite que el usuario indique si es una publicación favorita o no.</li> <li>3. El usuario selecciona el botón para activar o desactivar la publicación marcada como favorita, dependiendo del estado actual en el que se encuentre.</li> <li>4. El sistema modifica el estado de marcado como favorito o no de la base de datos.</li> </ol>
<b>Excepciones</b>	En caso de que el listado de publicaciones se encuentre vacío, el sistema mostrará un mensaje indicando que no hay publicaciones realizadas.
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-106</li> <li>✓ RF-42</li> <li>✓ RU-107</li> <li>✓ RF-43</li> <li>✓ RF-45</li> <li>✓ RF-46</li> </ul>
<b>Prioridad</b>	Alta

Tabla 142: RU-107 Activar/Desactivar publicación como favorita

<b>ID y Nombre</b>	<b>RU-108 ORDENAR LISTADO DE PUBLICACIONES POR PROXIMIDAD</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá ordenar el listado completo de publicaciones según su proximidad respecto a la ubicación actual del usuario autenticado en la plataforma.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario vendedor ha ordenado el listado completo de publicaciones activas en la plataforma por el resto de usuarios según su proximidad a su ubicación actual.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> <li>2. El sistema muestra un botón en la parte superior de la interfaz que permite al usuario seleccionar.</li> <li>3. El sistema muestra una ventana con dos opciones, una de ellas ordenar por “más cercano”.</li> <li>4. El usuario selecciona la opción “más cercano”.</li> <li>5. El sistema ordena el listado de publicaciones por proximidad según su ubicación actual.</li> </ol>
<b>Excepciones</b>	En caso de que el listado de publicaciones se encuentre vacío, el sistema mostrará un mensaje indicando que no hay publicaciones realizadas.
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-106</li> <li>✓ RF-42</li> <li>✓ RU-108</li> <li>✓ RF-43</li> <li>✓ RF-47</li> <li>✓ RF-48</li> </ul>
<b>Prioridad</b>	Alta

Tabla 143: RU-108 Ordenar listado de publicaciones por proximidad

<b>ID y Nombre</b>	<b>RU-109 ORDENAR LISTADO DE PUBLICACIONES POR FECHA DE PUBLICACIÓN</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá ordenar el listado completo de publicaciones según su fecha de publicación más reciente.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario vendedor ha ordenado el listado completo de publicaciones activas en la plataforma por el resto de usuarios según su fecha de publicación más reciente.

<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> <li>2. El sistema muestra un botón en la parte superior de la interfaz que permite al usuario seleccionar.</li> <li>3. El sistema muestra una ventana con dos opciones, una de ellas ordenar por “más recientes”.</li> <li>4. El usuario selecciona la opción “más recientes”.</li> <li>5. El sistema ordena el listado de publicaciones por fecha de publicación más reciente.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. En caso de que el listado de publicaciones se encuentre vacío, el sistema mostrará un mensaje indicando que no hay publicaciones realizadas.</li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-106</li> <li>✓ RF-42</li> <li>✓ RU-109</li> <li>✓ RF-49</li> <li>✓ RF-50</li> </ul>
<b>Prioridad</b>	Alta

Tabla 144: RU-109 Ordenar listado de publicaciones por fecha de publicación

<b>ID y Nombre</b>	<b>RU-110 REALIZAR FILTRADO POR INTERVALO DE PRECIO</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá realizar un filtrado por un intervalo de precio que desee al listado completo de publicaciones.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario vendedor ha filtrado por intervalo de precio el listado completo de publicaciones activas en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> <li>2. El sistema muestra un botón en la parte superior de la interfaz que permite al usuario seleccionar la opción de filtrado de intervalo de precio.</li> <li>3. El usuario selecciona el intervalo de precio por el que desea realizar la búsqueda.</li> <li>4. El sistema muestra el listado de publicaciones que comprende el intervalo de precio que ha seleccionado el usuario.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. En caso de que el listado de publicaciones se encuentre vacío, el sistema mostrará un mensaje indicando que no hay publicaciones realizadas.</li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-106</li> <li>✓ RF-42</li> </ul>

	<ul style="list-style-type: none"> <li>✓ RU-110</li> <li>✓ RF-51</li> <li>✓ RF-52</li> </ul>
<b>Prioridad</b>	Alta

Tabla 145: RU-110 Realizar filtrado por intervalo de precio

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-111 ACTUALIZAR LISTADO COMPLETO DE PUBLICACIONES</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá actualizar el listado completo de publicaciones, realizando el gesto de deslizar hacia abajo el widget correspondiente al listado.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario vendedor ha actualizado el listado completo de publicaciones activas en la plataforma, realizando el gesto de deslizar hacia abajo.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> <li>2. El sistema muestra un botón que permita actualizar mediante un gesto de deslizar la pantalla el listado completo de publicaciones.</li> <li>3. El usuario visualiza el listado completo de publicaciones actualizado, con la posibilidad de visualizar las publicaciones más recientes.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2. En caso de que el listado de publicaciones se encuentre vacío, el sistema mostrará un mensaje indicando que no hay publicaciones realizadas.</li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RU-106</li> <li>✓ RF-42</li> <li>✓ RU-111</li> <li>✓ RF-58</li> </ul>
<b>Prioridad</b>	Alta

Tabla 146: RU-111 Actualizar listado completo de publicaciones

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-112 SELECCIONAR BARRA DE BÚSQUEDA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar la barra de búsqueda incluida para realizar la búsqueda por título de publicación dentro de toda la plataforma.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario cliente ha seleccionado la barra de búsqueda que permite realizar la búsqueda de la publicación deseada, a partir del título que tiene asignado la publicación.

<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza un listado completo con todas las publicaciones realizadas por el resto de usuarios en la plataforma, mostrando cada publicación en una miniatura.</li> <li>2. El sistema muestra en la parte superior un botón que al seleccionarlo muestra una barra de búsqueda.</li> <li>3. El usuario selecciona el botón para acceder a la barra de búsqueda.</li> <li>4. El sistema muestra la vista de la barra de búsqueda.</li> <li>5. El usuario escribe el tipo de publicación que desea buscar.</li> <li>6. El sistema realiza la búsqueda en la base de datos por el título de publicación y muestra el listado coincidente con la búsqueda realizada.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-112</li> <li>✓ RF-53</li> <li>✓ RF-54</li> </ul>
<b>Prioridad</b>	Alta

Tabla 147: RU-112 Seleccionar barra de búsqueda

<b>ID y Nombre</b>	<b>RU-113 ESCRIBIR PUBLICACIÓN QUE SE DESEA BUSCAR</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá escribir en la barra de búsqueda la publicación que desea buscar.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).</li> <li>2. El usuario cliente debe haber seleccionado la barra de búsqueda previamente (RU-112).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha escrito en la barra de búsqueda el título de publicación que desea encontrar.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra en la parte superior un botón que al seleccionarlo muestra una barra de búsqueda.</li> <li>2. El usuario selecciona el botón para acceder a la barra de búsqueda.</li> <li>3. El sistema muestra la vista de la barra de búsqueda.</li> <li>4. El usuario escribe el tipo de publicación que desea buscar.</li> <li>5. El sistema realiza la búsqueda en la base de datos por el título de publicación y muestra el listado coincidente con la búsqueda realizada.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-112</li> <li>✓ RF-53</li> <li>✓ RF-54</li> <li>✓ RU-113</li> <li>✓ RF-54</li> <li>✓ RF-55</li> </ul>
<b>Prioridad</b>	Alta

Tabla 148: RU-113 Escribir publicación que se desea buscar

<b>ID y Nombre</b>	
<b>RU-114 VISUALIZAR LISTADO COINCIDENTE CON LA BÚSQUDA REALIZADA</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá escribir en la barra de búsqueda la publicación que desea buscar.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).</li> <li>2. El usuario cliente debe haber seleccionado la barra de búsqueda previamente (RU-112) y escrito el contenido que desea encontrar (RU-113).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado coincidente con la búsqueda realizada previamente.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario escribe el tipo de publicación que desea buscar.</li> <li>2. El sistema realiza la búsqueda en la base de datos por el título de publicación y muestra el listado coincidente con la búsqueda realizada.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-112</li> <li>✓ RF-53</li> <li>✓ RF-54</li> <li>✓ RU-113</li> <li>✓ RF-54</li> <li>✓ RF-55</li> <li>✓ RF-56</li> </ul>
<b>Prioridad</b>	Alta

Tabla 149: RU-114 Visualizar listado coincidente con la búsqueda realizada

<b>ID y Nombre</b>	
<b>RU-115 VISUALIZAR LISTADO HORIZONTAL CATEGORÍAS DISPONIBLES</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado horizontal completo de las categorías disponibles en la plataforma, que nos permite filtrar por la categoría seleccionada.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado horizontal completo de categorías disponibles.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza un listado completo con todas las categorías disponibles en la plataforma.</li> <li>2. El sistema muestra el listado de categorías que puede seleccionar el usuario en la parte superior de la pantalla del dispositivo.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-115</li> </ul>

	✓ RF-57
<b>Prioridad</b>	Alta

Tabla 150: RU-115 Visualizar listado horizontal de categorías disponibles

<b>ID y Nombre</b>	<b>RU-116 SELECCIONAR CATEGORÍA DESEADA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar una categoría del listado horizontal completo de las categorías disponibles en la plataforma, que nos permite filtrar por la categoría seleccionada.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).</li> <li>2. El usuario cliente debe haber visualizado el listado completo de categorías disponibles de la plataforma (RU-115).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha seleccionado una categoría del listado horizontal completo de categorías disponibles.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza un listado completo con todas las categorías disponibles en la plataforma.</li> <li>2. El sistema muestra el listado de categorías que puede seleccionar el usuario en la parte superior de la pantalla del dispositivo.</li> <li>3. El usuario selecciona la categoría por la que desee realizar el filtrado de las publicaciones.</li> </ol>
<b>Excepciones</b>	-
	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-115</li> <li>✓ RF-57</li> <li>✓ RU-116</li> <li>✓ RF-59</li> </ul>
<b>Prioridad</b>	Alta

Tabla 151: RU-116 Seleccionar categoría deseada

<b>ID y Nombre</b>	<b>RU-117 VISUALIZAR LISTADO COINCIDENTE CON LA CATEGORÍA SELECCIONADA</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado coincidente con una categoría del listado horizontal completo de las categorías disponibles en la plataforma.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).</li> <li>2. El usuario cliente debe haber seleccionado una categoría del listado completo de categorías disponibles de la plataforma (RU-116).</li> </ol>

<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado coincidente con una categoría del listado horizontal completo de categorías disponibles.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra el listado de categorías que puede seleccionar el usuario en la parte superior de la pantalla del dispositivo.</li> <li>2. El usuario selecciona la categoría por la que desee realizar el filtrado de las publicaciones.</li> <li>3. El sistema realiza el filtrado de publicaciones por la categoría seleccionada.</li> <li>4. El usuario visualiza el listado de categorías coincidentes con la categoría seleccionada.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-115</li> <li>✓ RF-57</li> <li>✓ RU-116</li> <li>✓ RF-59</li> <li>✓ RU-117</li> <li>✓ RF-60</li> </ul>
<b>Prioridad</b>	Alta

Tabla 152: RU-117 Visualizar listado coincidente con la categoría seleccionada

<b>ID y Nombre</b>	
<b>ID y Nombre</b>	<b>RU-118 VISUALIZAR LISTADO DE PUBLICACIONES PROPIAS</b>
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado de las publicaciones que han sido publicadas por el propio usuario en la plataforma.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado de publicaciones realizadas por sí mismo en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción correspondiente a “Mis publicaciones” desde el menú que aparece en la parte inferior en el inicio.</li> <li>2. El sistema muestra la pestaña de publicaciones publicadas por el usuario que está autenticado actualmente.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2. El listado de las publicaciones propias publicadas por el propio usuario autenticado aparece vacío. <ol style="list-style-type: none"> <li>2.1 El sistema muestra un mensaje informativo.</li> <li>2.2 El flujo retorna al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-118</li> <li>✓ RF-61</li> <li>✓ RF-62</li> </ul>
<b>Prioridad</b>	Alta

Tabla 153: RU-118 Visualizar listado de publicaciones propias

<b>ID y Nombre</b>	
<b>RU-119 SELECCIONAR PUBLICACIÓN DEL LISTADO DE PUBLICACIONES REALIZADAS POR EL USUARIO</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar una publicación del listado de publicaciones realizadas por el propio usuario.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).</li> <li>2. El usuario cliente debe haber visualizado el listado completo de publicaciones realizadas por sí mismo (RU-118).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha seleccionado una categoría del listado horizontal completo de categorías disponibles.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado de publicaciones realizadas por sí mismo desde el menú “Mis publicaciones”.</li> <li>2. El sistema muestra el listado de publicaciones realizadas por el usuario.</li> <li>3. El usuario selecciona una publicación del listado de publicaciones realizadas por sí mismo.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-118</li> <li>✓ RF-61</li> <li>✓ RU-119</li> <li>✓ RF-62</li> <li>✓ RF-67</li> </ul>
<b>Prioridad</b>	Alta

Tabla 154: RU-119 Seleccionar publicación del listado de publicaciones realizadas por el usuario

<b>ID y Nombre</b>	
<b>RU-120 VISUALIZAR EL LISTADO DE PUBLICACIONES MARCADAS COMO FAVORITAS POR EL USUARIO</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado de las publicaciones que han sido marcadas como favoritas por el usuario autenticado.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado de publicaciones marcadas como favoritas por sí mismo.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción correspondiente a “Mis publicaciones” desde el menú que aparece en la parte inferior en el inicio.</li> <li>2. El sistema muestra la pestaña de publicaciones marcadas como favoritas por el usuario que está autenticado actualmente.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El listado de las publicaciones marcadas como favoritas por el propio usuario autenticado aparece vacío. <ol style="list-style-type: none"> <li>3.1 El sistema muestra un mensaje informativo.</li> <li>3.2 El flujo retorna al punto 1.</li> </ol> </li> </ol>

<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-120</li> <li>✓ RF-63</li> </ul>
<b>Prioridad</b>	Alta

Tabla 155: RU-120 Visualizar el listado de publicaciones marcadas como favoritas por el usuario

<b>ID y Nombre</b>	
<b>RU-121 SELECCIONAR PUBLICACIÓN DEL LISTADO DE PUBLICACIONES MARCADAS COMO FAVORITAS</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar una publicación del listado de publicaciones marcadas como favoritas.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).</li> <li>2. El usuario cliente debe haber visualizado el listado completo de publicaciones marcadas como favoritas (RU-120).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha seleccionado una publicación del listado de publicaciones marcadas como favoritas.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado de publicaciones marcadas como favoritas desde el menú “Mis publicaciones”.</li> <li>2. El sistema muestra el listado de publicaciones marcadas como favoritas por el usuario.</li> <li>3. El usuario selecciona una publicación del listado de publicaciones marcadas como favoritas por sí mismo.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-120</li> <li>✓ RF-63</li> <li>✓ RU-121</li> <li>✓ RF-64</li> <li>✓ RF-67</li> </ul>
<b>Prioridad</b>	Alta

Tabla 156: RU-121 Seleccionar publicación del listado de publicaciones marcadas como favoritas

<b>ID y Nombre</b>	
<b>RU-122 VISUALIZAR EL LISTADO DE PUBLICACIONES MARCADAS COMO VENDIDAS</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá visualizar el listado de las publicaciones que han sido marcadas como vendidas por el usuario autenticado.
<b>Precondiciones</b>	El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).
<b>Postcondiciones</b>	El usuario cliente ha visualizado el listado de publicaciones marcadas como vendidas por sí mismo.

<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción correspondiente a “Mis publicaciones” desde el menú que aparece en la parte inferior en el inicio.</li> <li>2. El sistema muestra la pestaña de publicaciones marcadas como vendidas por el usuario que está autenticado actualmente.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. El listado de las publicaciones marcadas como vendidas por el propio usuario autenticado aparece vacío. <ol style="list-style-type: none"> <li>3.1 El sistema muestra un mensaje informativo.</li> <li>3.2 El flujo retorna al punto 1.</li> </ol> </li> </ol>
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-122</li> <li>✓ RF-65</li> </ul>
<b>Prioridad</b>	Alta

Tabla 157: RU-122 Visualizar el listado de publicaciones marcadas como vendidas

<b>ID y Nombre</b>	
<b>RU-123 SELECCIONAR PUBLICACIÓN DEL LISTADO DE PUBLICACIONES MARCADAS COMO VENDIDAS</b>	
<b>Versión</b>	<b>1.0</b>
<b>Actor principal</b>	Usuario cliente
<b>Descripción</b>	El usuario cliente podrá seleccionar una publicación del listado de publicaciones marcadas como vendidas.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario cliente debe haber visualizado el listado completo de publicaciones (RU-106).</li> <li>2. El usuario cliente debe haber visualizado el listado completo de publicaciones marcadas como vendidas (RU-122).</li> </ol>
<b>Postcondiciones</b>	El usuario cliente ha seleccionado una publicación del listado de publicaciones marcadas como vendidas.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado de publicaciones marcadas como vendidas desde el menú “Mis publicaciones”.</li> <li>2. El sistema muestra el listado de publicaciones marcadas como vendidas por el usuario.</li> <li>3. El usuario selecciona una publicación del listado de publicaciones marcadas como vendidas por sí mismo.</li> </ol>
<b>Excepciones</b>	-
<b>Trazabilidad</b>	<ul style="list-style-type: none"> <li>✓ RF-42</li> <li>✓ RU-122</li> <li>✓ RF-65</li> <li>✓ RU-123</li> <li>✓ RF-66</li> <li>✓ RF-67</li> </ul>
<b>Prioridad</b>	Alta

Tabla 158: RU-123 Seleccionar publicación del listado de publicaciones marcadas como vendidas

## 4.10 REQUISITOS DE INFORMACIÓN

A continuación, se contempla un listado donde se muestran los **Requisitos de Información (RI-XX)** de la plataforma, con el objetivo de almacenar toda la información necesaria para un buen funcionamiento del sistema y salvaguarda de la información. Por lo que, se deberá almacenar la información relativa de los usuarios que se den de alta en la aplicación y todas las funcionalidades disponibles en la aplicación.

En los siguientes apartados, se tratará más en profundidad todos ellos, representándolos en un modelo entidad-relación y especificando los atributos que forman cada uno de ellos.

- ❖ **RI-01:** El sistema deberá almacenar toda la información referida a todos los usuarios que se registren en la plataforma.
  - Nombre y apellidos
  - Email
  - Contraseña
  - Avatar de perfil de usuario
  - Media de puntuación sobre valoraciones.
  - Valoraciones recibidas por el resto de usuarios.
  - Token de sesión
  - UID
- ❖ **RI-02:** El sistema deberá almacenar toda la información referida a las valoraciones realizadas mediante el sistema de valoración entre usuarios.
  - Comentario.
  - ID del Comentario.
  - Fecha de publicación de la valoración.
  - Puntuación.
  - Nombre del usuario que realiza la valoración.
  - Avatar de perfil del usuario propietario de la valoración.
  - ID del usuario propietario de la valoración.
- ❖ **RI-03:** El sistema deberá almacenar toda la información referida a las publicaciones.
  - ID publicación.
  - ID del propietario de la publicación.
  - Título.
  - Descripción.
  - Precio.
  - Fecha de creación de la publicación.
  - Ubicación de la publicación.
  - Coordenadas físicas de la publicación.
  - Estado del producto relacionado con la publicación.
  - Estado de admisión de envío del producto o no
  - Imágenes de la publicación.
  - Estado de publicación favorita o no favorita.
  - Estado de publicación vendida o no vendida.
  - Categoría asociada.
  - Subcategoría asociada.

- ❖ **RI-04:** El sistema deberá almacenar toda la información referida a las categorías que jerarquizan las publicaciones añadidas.
  - Categorías asociada
  - Subcategoría asociada
  
- ❖ **RI-05:** El sistema deberá almacenar toda la información referida a la información sobre los chats del servicio de mensajería instantánea.
  - ID del chat
  - Contenido del último mensaje.
  - Fecha de la última modificación.
  - ID del último usuario que ha enviado un mensaje
  - Fecha del último mensaje enviado.
  - ID del usuario emisor.
  - ID del usuario receptor.
  
- ❖ **RI-06:** El sistema deberá almacenar toda la información referida a los mensajes enviado a través del servicio de mensajería instantánea.
  - URL de imagen
  - Leído/No leído
  - Contenido mensaje.
  - ID usuario emisor.
  - ID usuario receptor.
  - Fecha de envío de mensaje.
  
- ❖ **RI-07:** El sistema deberá almacenar toda la información referida a la ubicación de las publicaciones.
  - Coordenadas latitud ubicación de origen.
  - Coordenadas longitud ubicación de origen.
  - Coordenadas longitud ubicación de destino.
  - Coordenadas latitud ubicación de destino
  
- ❖ **RI-09:** El sistema almacenará toda la información referida a la posición obtenida del usuario mediante el sistema de seguimiento en tiempo real, donde la base de datos actualizará la información cada 2 segundos.
  - Coordenadas latitud.
  - Coordenadas longitud.
  
- ❖ **RI-10:** El sistema almacenará toda la información en las cookies de la aplicación almacenadas en la memoria del dispositivo móvil sobre las funcionalidades referidas al sistema de seguimiento en tiempo real, es decir, información relativa a la ruta.
  - Puntos de trazado
  - Ubicación inicial
  - Ubicación de destino
  - Duración
  - Distancia
  - Nombre ubicación
  - Descripción ubicación.

## 4.11 MODELO LÓGICO DE DATOS (DIAGRAMA ENTIDAD – RELACIÓN)

Tras el análisis de los requisitos de información, contenido que se deberá almacenar en la base de datos, se mostrará una serie de entidades conectadas entre sí mediante relaciones para satisfacer los requisitos.

El modelo Entidad-Relación, es una herramienta usada para el modelado de datos que permite representar de una forma clara y simple las entidades que se pueden encontrar en un sistema de información, así como los requisitos de información, y las relaciones que se establecen entre ellas. Es el principal modelo utilizado y diseñado para representar toda la información recogida en una base de datos.

El modelo E-R está formado por un diagrama donde se realiza una representación gráfica de los datos del sistema de información y es complementada por un diccionario de datos, donde se especifican los tipos de datos que podemos encontrar en el diagrama.

En este modelo, podemos encontrar los siguientes elementos:

- **Entidad:** se define como un objeto, cosa o concepto que podemos encontrar en nuestro entorno del que se recoge información. En el diagrama, se presenta con un rectángulo y se deben representar con sustantivos, además el nombre de una entidad no se puede repetir en el diagrama.
- **Atributos:** características que definen a una entidad o relación. En el diagrama se representan mediante un círculo o elipse, que se encuentra unida por una línea a la entidad o relación a la que pertenece.
- **Relación:** es la asociación entre dos o más entidades. Se deben identificar con verbos, generalmente en presente verbal, y en el diagrama se representan mediante rombos

En el modelo E-R, podemos encontrar otro factor, representar la cardinalidad con la que una entidad participa en una relación. Las cardinalidades máximas y mínimas de las entidades que participan en una relación son el número máximo y mínimo de objetos de una entidad, que pueden relacionarse con uno o varios objetos de la otra entidad que participa en la relación. Dichas cardinalidades son representadas en el modelo en la línea que establece la conexión de la entidad y la relación.

Las entidades representadas pueden encontrarse de dos tipos distintos:

- **Fuerte:** son aquellas cuya existencia no depende de ninguna otra entidad.
- **Débil:** son aquellas entidades cuya existencia depende de otra. Se representan en el diagrama con un rectángulo con doble borde. Pueden ser de dos tipos; de existencia o de identificación. Ambos casos comparten que al eliminar la entidad fuerte se pierde la entidad débil, pero en el caso de identificación la entidad débil necesita a la fuerte para poder identificarse.

Otro aspecto a tener en cuenta que debemos destacar es la generalización y especialización. Se define cuando de una entidad, superclase, se pueden generar más entidades, subclasses que heredan los atributos y relaciones de esta. La forma de representarlo en el modelo es mediante un triángulo.

Ambas, Generalización y Especialización, tienen dos restricciones:

- **Restricción de participación**, la cual puede ser de dos tipos:

- **Obligatoria:** cuando todos los objetos que se pueden encontrar en la superclase tienen que pertenecer a una subclase. Se representa en el diagrama mediante un círculo entre la línea que une la superclase con el triángulo que representa la generalización.
- **Optativa:** se da cuando no es necesario que un objeto de la superclase pertenezca a una subclase. Se representa en el diagrama mediante una línea que forma la unión de la superclase con el triángulo.
  
- **Restricción de disyunción**, la cual puede ser de dos tipos:
  - **Disjunta:** se da cuando un objeto de la superclase solo puede pertenecer a una subclase. Se representa en el diagrama mediante una línea de forma curva que pasa por encima de las líneas que unen el triángulo con las subclases.
  - **No disjunta:** se da en caso contrario, cuando un objeto de la superclase puede pertenecer a más de una subclase. Se representa en el diagrama mediante líneas que unen el triángulo con las subclases.

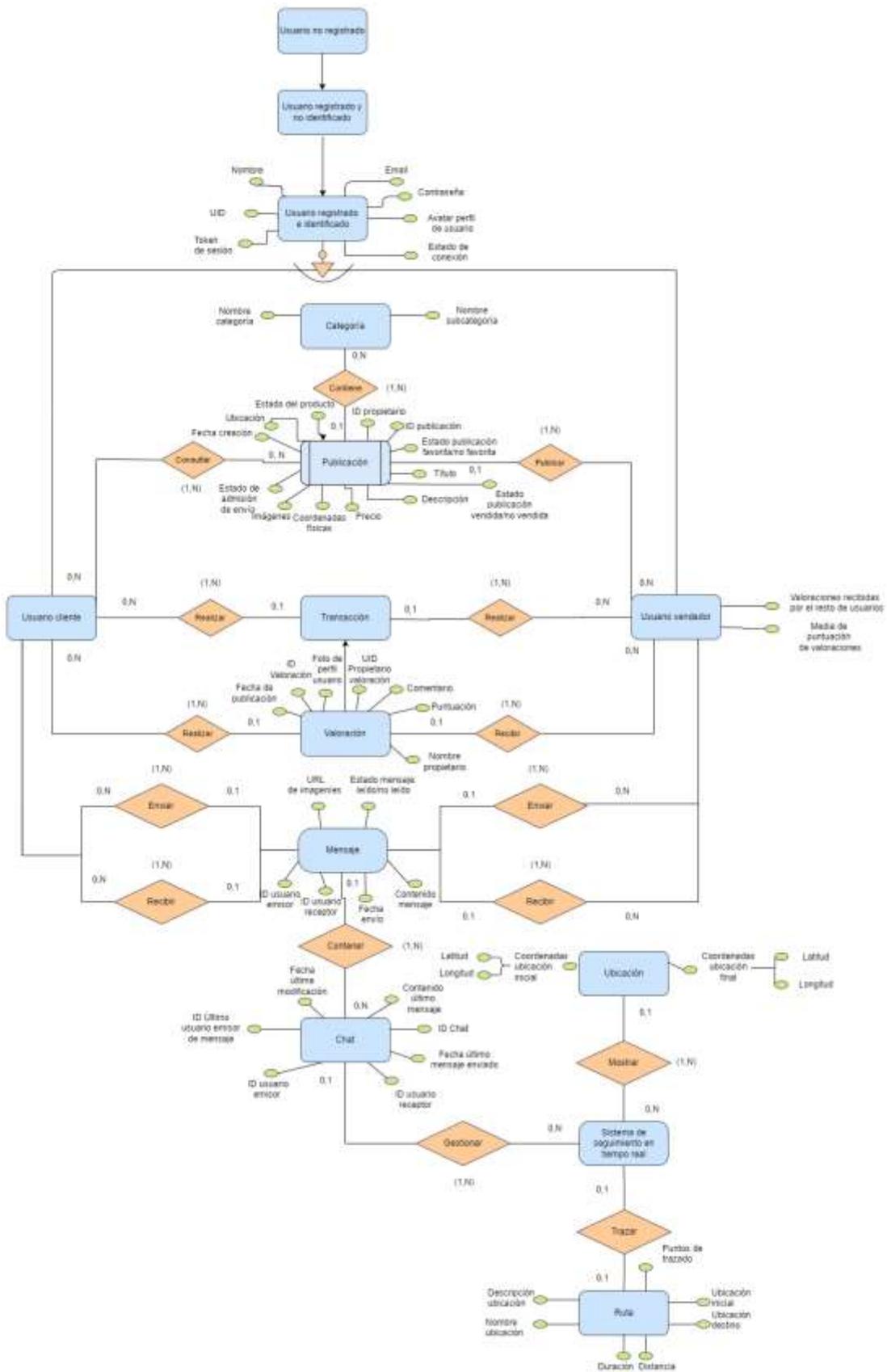


Figura 33: Modelo Entidad-Relación



---

---

## 5. DISEÑO

---

---



## 5.1 ARQUITECTURA LÓGICA

A continuación, se mostrará mediante su arquitectura lógica, la estructura mediante la cual se relacionan los diferentes componentes lógicos de la aplicación, que dan forma al sistema junto con las relaciones de los componentes entre sí mismos, es decir, se define cómo interactúan entre sí mismos en la aplicación.

En el caso de la aplicación, se ha escogido una arquitectura basada en el modelo *Clean Architecture*, conjunto de principios y patrones de diseño que deben facilitar el proceso de construcción del software, así como su mantenimiento y escalabilidad.

La mayor particularidad de esta manera de estructurar la arquitectura de una aplicación en Flutter, dado que se establece una base de estructura de código en capas o *layers* por jerarquías, donde cada capa solo se comunica con sus capas contiguas. Así se consigue el objetivo principal del modelo, donde cada capa tiene sus funcionalidades y dependencias específicas dentro de la plataforma.

Se pueden destacar las siguientes ventajas respecto a *Clean Architecture*:

- ❖ **Separación de componentes:** promueve la separación de componentes, lo que permite crear una estructura de código más organizada y modularizada. Cada capa tiene una responsabilidad específica, y las capas están poco acopladas entre sí, lo que hace que el código sea más fácil de mantener y realizar pruebas.
- ❖ **Pruebas:** Facilita la realización de pruebas automatizadas, ya que las distintas capas están aisladas y pueden probarse de forma independiente. Esto mejora la calidad del software y reduce el riesgo de introducir errores.
- ❖ **Flexibilidad:** al disponer de un diseño modular que permite a los desarrolladores intercambiar componentes o cambiar los detalles de implementación sin afectar al resto del sistema. Esto facilita la adaptación a requisitos que se deben modificar durante el desarrollo o la incorporación de nuevas funcionalidades.
- ❖ **Independencia de frameworks:** Permite cambiar *frameworks* o librerías sin afectar al resto del sistema.
- ❖ **Mantenimiento:** Ayuda a desarrollar un código más fácil de comprender y mantener. Al separar los componentes y utilizar interfaces claras entre las capas, los desarrolladores pueden trabajar en diferentes funcionalidades de la plataforma sin afectar a las restantes.

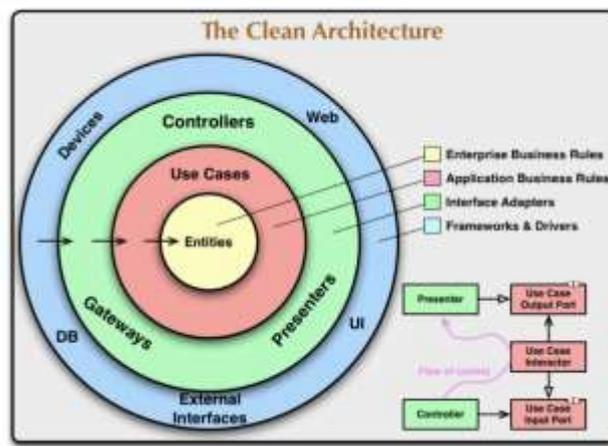


Figura 34: Estructura Clean Architecture

A continuación, podemos observar en la siguiente figura la arquitectura lógica correspondiente a la plataforma:

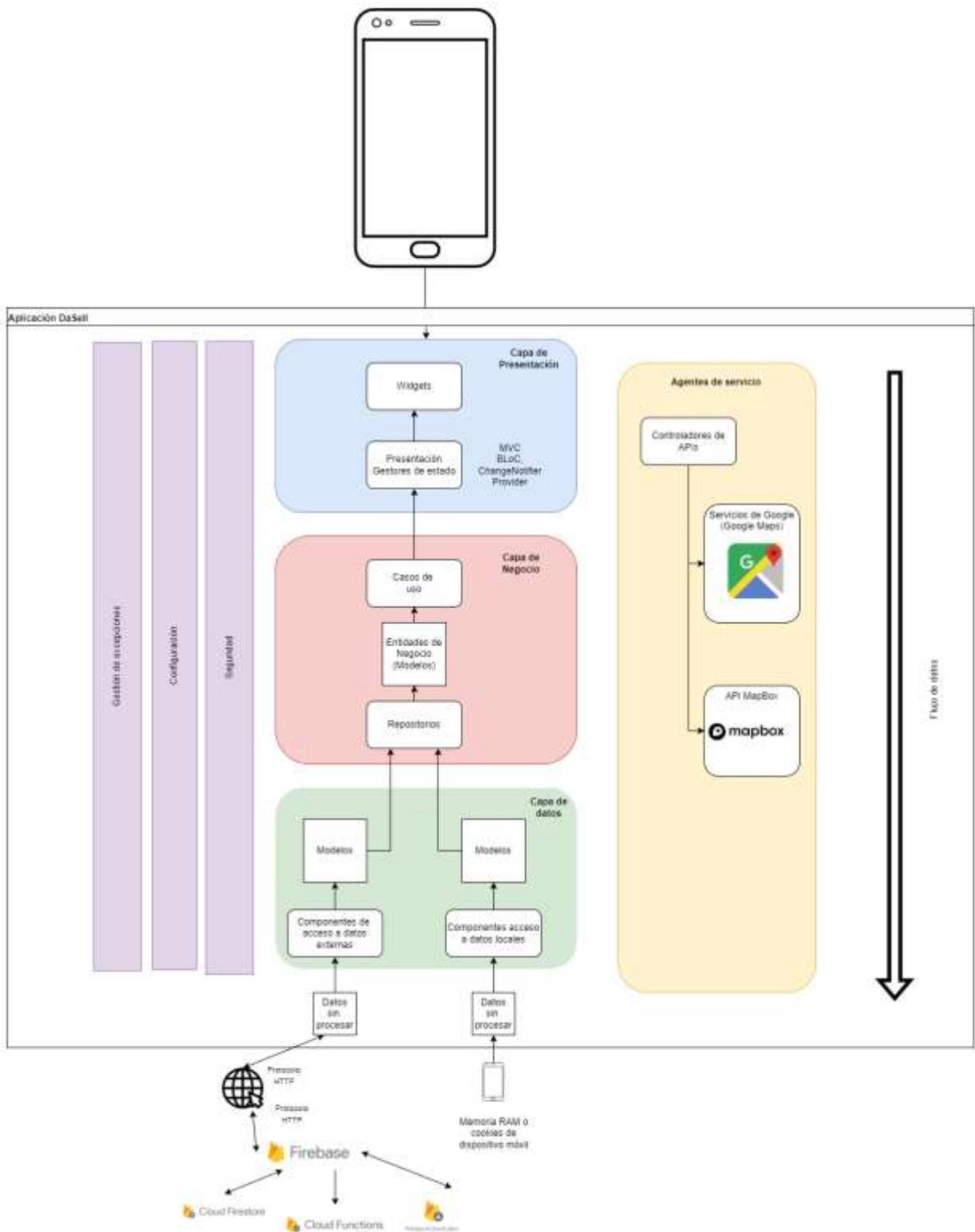


Figura 35: Arquitectura lógica – Aplicación Flutter

Como se puede observar representado en la *Figura 35: Arquitectura lógica – Aplicación Flutter*, la arquitectura lógica se identifica con un modelo multicapa, en este caso podemos diferenciar tres capas:

capa de presentación, capa de negocio y capa de datos. Completando a estas capas, se añade una capa que identifica a los agentes de servicio que componen la aplicación, y que interactúan con las tres capas principales, suministrando la información necesaria para el funcionamiento de la plataforma. De manera adicional, se incluyen otras tres capas transversales que identifican toda la funcionalidad relacionada con la seguridad, la configuración y el manejo de excepciones.

Una capa es un conjunto de clases, paquetes o ficheros que tienen unas responsabilidades relacionadas dentro del sistema. Estas capas están organizadas de forma jerárquica unas encima de otras y las dependencias siempre van hacia abajo. Es decir, que una capa dependerá solamente de las capas inferiores, pero nunca de las superiores.

A continuación, se realiza una breve descripción sobre cada capa que compone la arquitectura lógica de la aplicación:

- **Capa de presentación:** engloba el conjunto de vistas con las que el usuario realiza un intercambio de información con la aplicación, por lo que se encarga de la interfaz de usuario y la interacción con el usuario. A su vez, podemos encontrar los controladores de interfaz, cuyo objetivo es recuperar los datos de las vistas, tareas de procesamiento y presentan los resultados al usuario del dispositivo, a través de las vistas.

En esta capa podemos encontrar los *widgets* de Flutter, como pantallas, componentes de navegación y elementos de diseño. Así como los patrones de diseño implementados. Dichos componentes serán descritos con detalle en la siguiente sección, referida a los detalles de la implementación.

- **Capa de negocio:** se encuentra la implementación de la lógica de negocio del código de la aplicación. Dicha lógica es necesario que sea interpretada y realice operaciones sobre la información de los datos. Se ocupa de procesar los datos, aplicar reglas de negocio y coordinar las operaciones entre la capa de presentación y la capa de datos. Aquí se definen los servicios y los repositorios que se utilizan para acceder a los datos y realizar operaciones.
- **Capa de acceso a datos:** se encarga de recibir peticiones por parte de la capa de negocio y los controladores de los *widgets*, para interactuar con el almacenamiento, permitiendo que se puedan aislar los datos del resto de capas, favoreciendo su acceso y manipulación. En esta capa es donde se integra Firebase y se utilizan los servicios proporcionados por Firebase SDK para interactuar con la base de datos en tiempo real, almacenamiento en la nube, autenticación, etc. La capa de acceso a datos abstrae los detalles de implementación de Firebase y proporciona métodos y funciones coherentes para la capa de lógica de negocio.
- **Agentes de servicios:** servicios que interactúan con el sistema mediante llamadas a sus propias APIs. Esta capa se maneja mediante servicios, que implementan controladores como en el caso del resto de capas, lo que permite implementarlo en cualquier de esas capas. En este caso implementamos APIs de servicios de Google y MapBox. Estas clases pueden incluir anotaciones de serialización/deserialización para facilitar la integración con Firebase.

A lo largo de la arquitectura, es común utilizar patrones de diseño como inyección de dependencias para gestionar las dependencias entre las capas y facilitar las pruebas unitarias. Además, también se pueden implementar patrones como BLOC y Provider para gestionar el estado de la aplicación y los *widgets* que la componen para facilitar la gestión de la reactividad.

Podemos observar que la arquitectura se asemeja bastante al patrón de diseño Modelo-Vista-Controlador. En la siguiente figura, podemos definir la arquitectura del patrón Modelo-Vista-Controlador (MVC).

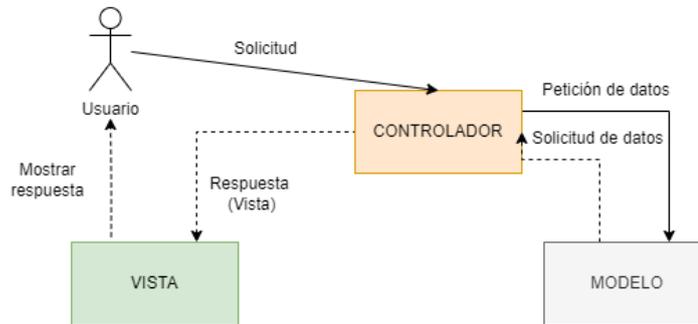


Figura 36: Patrón Modelo-Vista-Controlador

Existen dos reglas fundamentales de este patrón:

- Cada vista vendrá siempre acompañada por su modelo de vista.
- La vista sólo conoce a su modelo de vista y este al modelo, la vista no conoce al modelo.

## 5.2 ARQUITECTURA FÍSICA

Tras establecer la arquitectura lógica de la aplicación, es fundamental realizar el diseño de la arquitectura física donde se establece la estructura de los componentes físicos que intervienen en la aplicación. Se encarga de representar la forma en la que se distribuye la aplicación hacia los usuarios finales, en la que se resaltan los actores y el dispositivo final por el cual se hace llegar la aplicación al usuario final.

Por lo que, la arquitectura física trata de describir la estructura en la que se encuentra desplegada la tecnología implementada. Se definen los componentes físicos que requiere la arquitectura lógica para lograr cumplir con sus objetivos, con el fin de satisfacer los requisitos principales de seguridad, disponibilidad y escalabilidad definidos en la sección “4.7 Requisitos no funcionales”, como se puede observar en la siguiente figura:

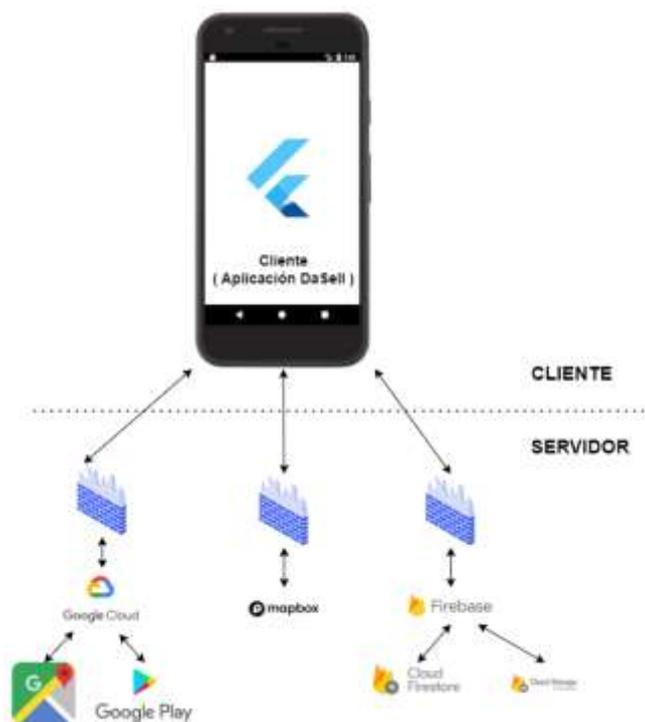


Figura 37: Arquitectura física

Como podemos observar en la figura, se ha definido la arquitectura física mediante un modelo de dos capas, estableciendo el paradigma Cliente-Servidor. El cliente está compuesto por un dispositivo móvil, ya tenga implementado el sistema operativo Android o iOS, indistintamente, al tratarse de una aplicación multiplataforma.

Por otro lado, la capa servidor podemos distribuirla en tres módulos distintos:

- **Google Services:** servicios de Google, los cuales se implementan o se accede mediante las APIs de Google Maps o Play Store.
- **MapBox:** la plataforma proporciona su propia API y podemos acceder a través de ella.
- **Firebase:** sus servicios son implementados a través de la API de Firebase. En el caso de esta aplicación, se utilizarán los siguientes servicios:
  - ❖ **Cloud Firestore**, que nos permite gestionar toda la información almacenada y relacionada con la plataforma.
  - ❖ **Firebase Authentication**, que nos permite la gestión de usuarios tanto por registro en la aplicación como por registro mediante cuenta de Google
  - ❖ **Storage**, para el almacenamiento de contenido audiovisual.

Por último, destacar que las conexiones y comunicaciones entre el módulo cliente y módulo servidor, se encuentra protegidas a través de *firewalls*, los cuales permiten disponer, en la capa de persistencia, de seguridad y filtrado de las peticiones realizadas y las respuestas realizadas hacia el servidor.

### 5.3 MÓDELO LÓGICO DE DATOS

Tras definir las arquitecturas, tanto lógica como física de la aplicación desarrollada, necesitamos establecer la estructura y el diseño de la base de datos. Dentro de esta sección, se trata de explicar los conceptos del modelo lógico de datos junto con las diferentes entidades que intervienen.

En primer lugar, la tipología de base de datos implementada es una base de datos NoSQL, es decir, una base de datos no relacional. Dicha tipología se puede definir como un mecanismo para el almacenamiento y recuperación de datos que se distingue, en aspectos muy relevantes, a la tipología de bases de datos SQL. Estas tratan de ser modelos de relaciones entre entidades mediante tablas, siendo su aspecto más destacado el no usar SQL como lenguaje para la realización de consultas.

Las razones principales por la utilización de esta tipología de base de datos, es la posibilidad de manejar un esquema de datos no homogéneo, y la capacidad de escalabilidad que nos permite, ya que esta tipología es capaz de soportar alta demanda del sistema por una gran cantidad de usuarios en momentos determinados. Para reforzar estos motivos de elección, se definen las características principales que se destacan en esta tipología:

- ❖ Se almacena la información como un documento.
- ❖ Se utilizan estructuras de distinto tipo como puede ser JSON, XML, BSON.
- ❖ Se permite realizar búsquedas mediante la clave-valor y la realización de consultas avanzadas sobre el contenido de los documentos.
- ❖ Debido a su flexibilidad, estructura y jerarquía, ofrece capacidad de evolución según las necesidades de la aplicación.

Antes de realizar el desarrollo de conceptos, se define a alto nivel, el proveedor de este servicio implementado para la base de datos. La aplicación está soportada en la parte de *backend* por Firebase, una plataforma que proporciona a los usuarios la posibilidad de implementar una gran variedad de servicios destinados al desarrollo de aplicaciones tanto para plataformas web como móviles, dónde en este último tipo incluimos a DaSell. Para este proyecto, uno de los servicios implementados para ofrecer de un servicio de base de datos a la aplicación, es el servicio denominado Cloud Firestore, junto con el servicio Cloud Storage, que nos permite almacenar archivos multimedia.

El servicio Cloud Firestore destaca por tratarse de una base de datos no relacional, es decir, NoSQL, al contrario que las bases de datos relacionales, es decir, SQL. Existen diversas diferencias entre las dos tipologías de servicios de bases de datos, pero cabe destacar que se debe analizar concretamente la estructura relacionada y el modelado de los datos que se realiza. En la siguiente tabla, se muestra una comparativa de ambas tipologías.

Base de datos relacionales (SQL)	Cloud Firestore (NoSQL)
Destaca por una estricta integridad referencial, entre las tablas, ya que debe disponer de relación entre tablas e índices.	<b>Permite flexibilidad en el modelado de datos</b> → los documentos están formados por diferentes tipos de datos.
<b>La información se almacena por tablas</b> , distribuidas en filas y columnas.	La información se almacena en colecciones de documentos.

Como podemos observar en la siguiente figura, se representan las principales colecciones de documentos donde están referenciadas las distintas entidades que componen y por las que se rige la base de datos, quedando definida la estructura de la siguiente forma.



Figura 38: Estructura documentos, base de datos Cloud Firestore

Si se accede a cada uno de los documentos, profundizando en los siguientes niveles de la estructura, se puede observar cómo están definidos mediante un identificador único y pares clave-valor, que permiten soportar multitud de tipologías de información para almacenar los datos, contando con un número heterogéneo de documentos.

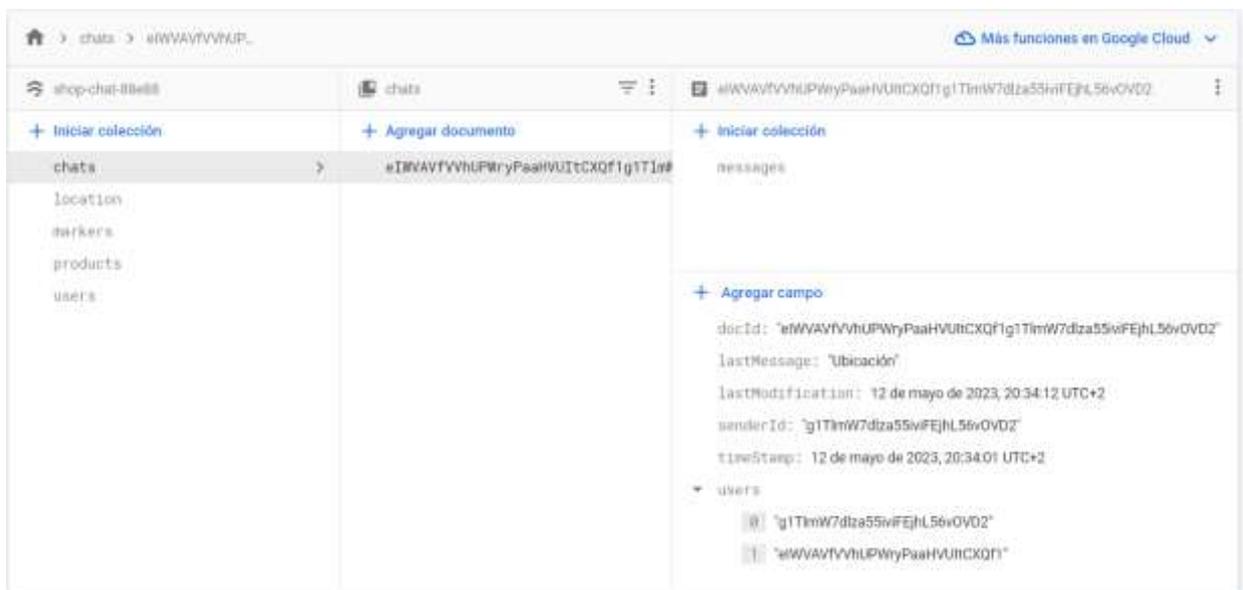


Figura 39: Estructura de un documento en Cloud Firestore

En consecuencia, los documentos pueden disponer en su estructura de subcolecciones de documentos, como podemos observar en la siguiente figura, se proporciona la estructura flexible y jerarquizada que se ha mencionada anteriormente.

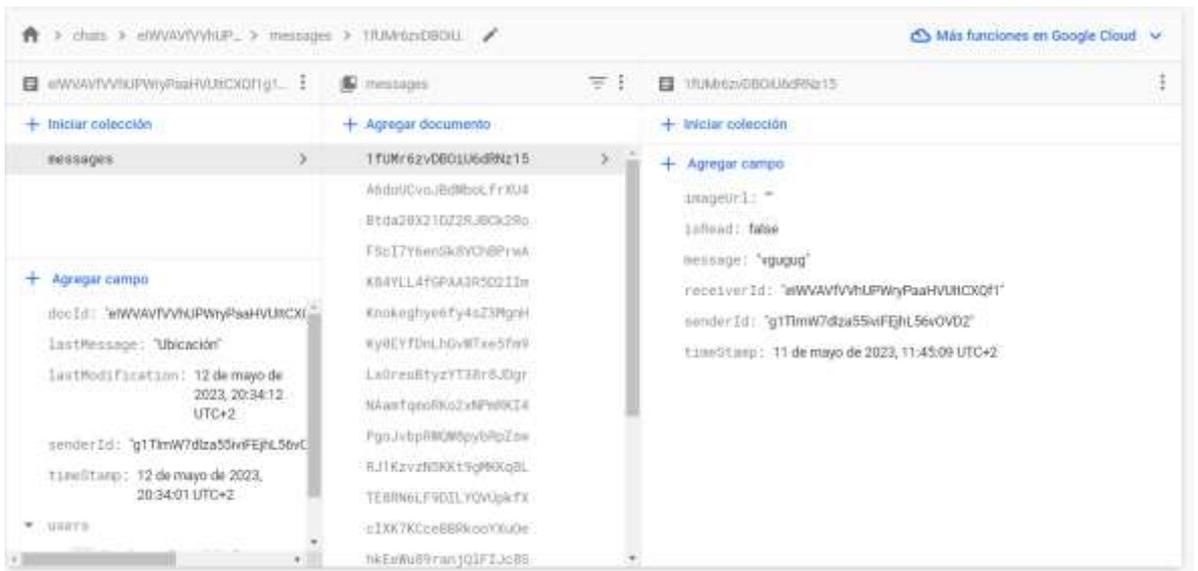


Figura 40: Subcolecciones de Cloud Firestore

En la *Figura 40: Subcolecciones de Cloud Firestore*, se puede observar una subcolección, denominada Messages, junto con todos sus valores clave-valor, la cual se encuentra dentro de la colección chats.

Con el objetivo de mejorar y simplificar la complicada labor de conciliar los modelos de datos empleados en el código fuente con los modelos de documentos, Cloud Firestore ofrece la capacidad de interpretar y traducir los modelos de código para que se ajusten a sus propios modelos de documento, y viceversa. Esto contribuye a mejorar el almacenamiento, procesamiento y consulta de datos para los desarrolladores, brindándoles una mayor facilidad en estas tareas.

Se utilizan los modelos de datos, mediante definición de objetos, para tratar los datos de forma estándar y estructurada, con el fin de evitar las complicaciones generadas por la flexibilidad a la hora de definir un documento. Así, cada objeto que forma una colección, debe estar definido mediante un objeto en el código de la aplicación. A continuación, podemos ver la estructura de un modelo en el código junto con su documento correspondiente en Cloud Firestore.

```
class ChatRoomVo {
  bool? isRead;
  String? lastMessage;
  String? senderId;
  Timestamp? timeStamp;
  String docId;
}
```

Figura 39: Modelo de datos (Flutter)



Figura 40: Relación de modelo de datos (Cloud Firestore)

### 5.3.1 DISEÑO BASE DE DATOS

En este apartado, se muestra un modelo lógico de datos, aunque se implemente una base de datos no relacional, es decir, NoSQL. Este modelo nos permite visualizar las relaciones existentes entre entidades, atributos, identificados del diseño de la base de datos.

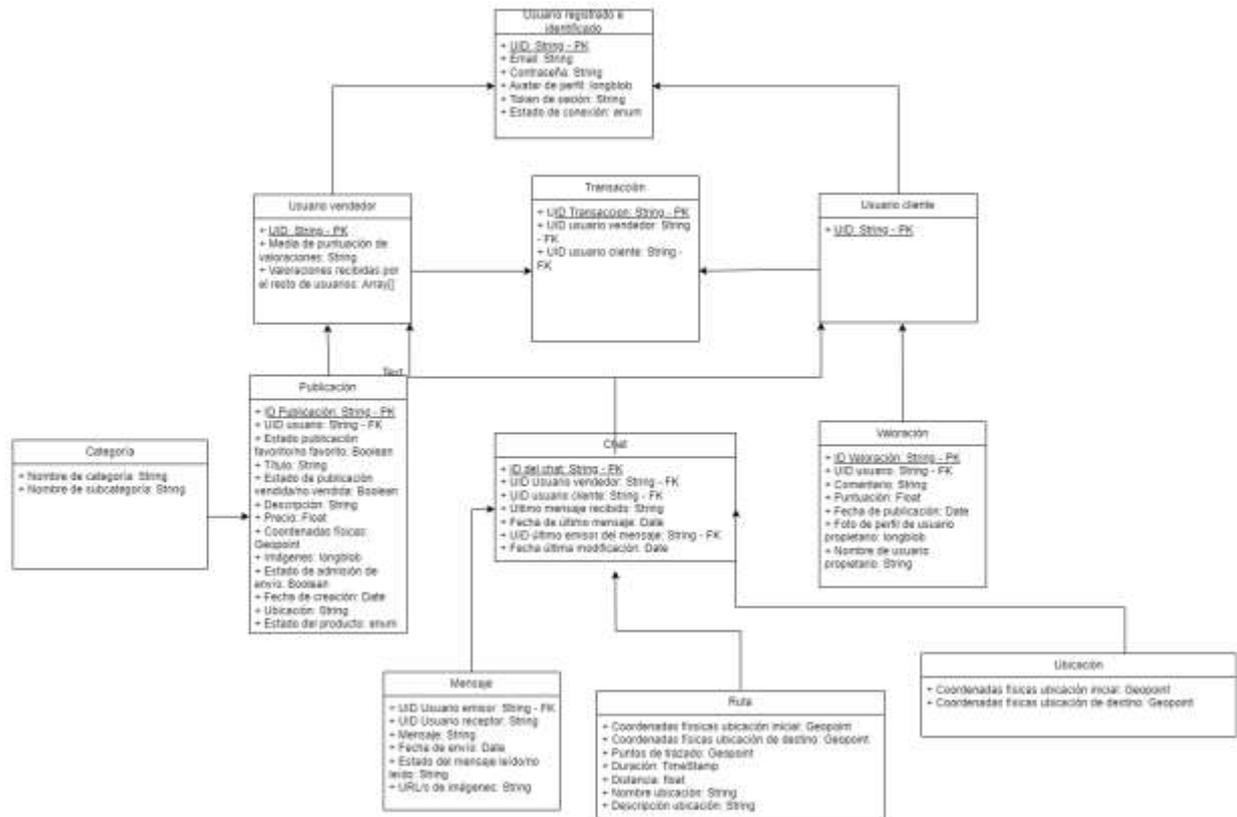


Figura 41: Modelo lógico de datos - DaSell

### 5.4 DICCIONARIO DE DATOS

En este apartado, se realiza el análisis de las entidades y relaciones representadas en el modelo lógico de datos, modelo Entidad-Relación. Cada entidad se encuentra definida por atributos, que se representan detalladamente junto a su entidad correspondiente para analizarlos e identificarlos.

#### 5.4.1. USUARIO REGISTRADO E IDENTIFICADO

USUARIO REGISTRADO E IDENTIFICADO						
Entidad que representa a todos los usuarios que pueden interactuar en la aplicación.						
ID	Nombre	Tipo	Nulo	Unique	Descripción	Comentario
RI-EURI-01	UID	Cadena de caracteres	NO	SÍ	Identificador del usuario	PRIMARY KEY
RI-EURI-02	Nombre	Cadena de caracteres	NO	NO	Nombre del usuario	
RI-EURI-03	Email	Cadena de caracteres	NO	SÍ	Dirección de correo electrónico del usuario	user@dominio.com

<b>RI-EURI-04</b>	Contraseña	Cadena de caracteres	NO	NO	Contraseña de inicio de sesión en la aplicación	
<b>RI-EURI-05</b>	Avatar de perfil	longblob	SÍ	NO	Imagen de perfil del usuario	
<b>RI-EURI-06</b>	Token de sesión	Cadena de caracteres	NO	SÍ	Clave generada tras el inicio de sesión del usuario	
<b>RI-EURI-07</b>	Estado de conexión	Enum	NO	NO	Estado de conexión en la plataforma	{Conectado, desconectado}

Tabla 159: Diccionario de datos – Usuario registrado e identificado

#### 5.4.2. USUARIO VENDEDOR

<b>USUARIO VENDEDOR</b>						
Entidad que representa a todos los usuarios vendedores que pueden interactuar en la aplicación.						
ID	Nombre	Tipo	Nulo	Unique	Descripción	Comentario
<b>RI-EUV-01</b>	Media puntuación de valoraciones	float	SÍ	NO	Medía de la puntuación, calculada a partir de las valoraciones recibidas por el resto de usuarios de la plataforma	
<b>RI-EUV-02</b>	Valoraciones recibidas por el resto de usuarios	Array[N]	SÍ	NO	Conjunto de información todas las valoraciones recibidas por parte del resto de usuarios	Array

Tabla 160: Diccionario de datos – Usuario vendedor

#### 5.4.3. CATEGORÍA

<b>CATEGORÍA</b>						
Entidad que representa a la categoría y su subcategoría asociada perteneciente a una publicación.						
ID	Nombre	Tipo	Nulo	Unique	Descripción	Comentario
<b>RI-EC-01</b>	Nombre de categoría	Cadena de caracteres	SÍ	NO	Nombre de la categoría relacionada con una publicación	
<b>RI-EC-02</b>	Nombre de subcategoría	Cadena de caracteres	SÍ	NO	Nombre de la subcategoría relacionada con una publicación	

Tabla 161: Diccionario de datos - Categoría

#### 5.4.4. PUBLICACIÓN

PUBLICACIÓN						
Entidad que representa a cada publicación realizada por el usuario vendedor en la plataforma.						
ID	Nombre	Tipo	Nulo	Unique	Descripción	Comentario
RI-EP-01	UID Usuario propietario	Cadena de caracteres	NO	SÍ	Identificador del usuario propietario de la publicación	FOREIGN KEY
RI-EP-02	ID publicación	Cadena de caracteres	NO	SÍ	Identificador asociado a la publicación	PRIMARY KEY
RI-EP-03	Estado publicación favorito/no favorito	Boolean	SÍ	NO	Indica si la publicación está marcada o no marcada como favorito por un usuario cliente	
RI-EP-04	Título	Cadena de caracteres	SÍ	NO	Título asociado a la publicación	
RI-EP-05	Estado de publicación vendida/no vendida	Boolean	NO	NO	Indica si la publicación está marcada o no marcada como vendida por un usuario vendedor	
RI-EP-06	Descripción	Cadena de caracteres	SÍ	NO	Descripción asociada a la publicación	
RI-EP-07	Precio	Float	SÍ	NO	Precio asociado a la publicación	Valor = 0, en el caso de tratarse de una publicación marcada como donación
RI-EP-08	Coordenadas físicas	Geopoint	NO	NO	Coordenadas físicas correspondientes a la latitud y longitud de la ubicación de la publicación	Par de valores: {Lat, Lng}
RI-EP-09	Imágenes	longblob	SÍ	NO	Imágenes asociadas a la publicación	
RI-EP-10	Estado de admisión de envío	Boolean	NO	NO	Indica si el usuario vendedor admite envíos o no del producto publicado	
RI-EP-11	Fecha creación	Date	NO	NO	Fecha de creación de la publicación	Formato: dd/MM/yyyy HH:MM:SS
RI-EP-12	Ubicación	Cadena de caracteres	NO	NO	Indica la dirección asociada a la ubicación de la publicación	

RI-EP-13	Estado del producto	Enum	SÍ	NO	Estado del producto asociado a la publicación	{'0': 'Lo ha dado todo', '25': 'En condiciones aceptables', '50': 'En buen estado', '75': 'Como nuevo', '100': 'Nuevo',}
----------	---------------------	------	----	----	---	--

Tabla 162: Diccionario de datos - Publicación

### 5.4.5. VALORACIÓN

VALORACIÓN						
Entidad que representa a todas las valoraciones realizadas a los usuarios vendedores.						
ID	Nombre	Tipo	Nulo	Unique	Descripción	Comentario
RI-EV-01	UID Propietario valoración	Cadena de caracteres	NO	SÍ	Identificador del usuario propietario de valoración	
RI-EV-02	ID Valoración	Cadena de caracteres	NO	SÍ	Identificador de la valoración	PRIMARY KEY
RI-EV-03	Comentario	Cadena de caracteres	SÍ	NO	Contenido de valoración realizada por el usuario	
RI-EV-04	Puntuación	Float	NO	NO	Puntuación asignada por el usuario propietario de la valoración al usuario vendedor	
RI-EV-05	Fecha de publicación de valoración	Date	SÍ	NO	Fecha de realización de la valoración al usuario vendedor	Formato: dd/MM/yyyy HH:MM:SS
RI-EV-06	Foto de perfil del usuario propietario de la valoración	longblob	NO	NO	Foto asignada al perfil de usuario del propietario de la valoración	
RI-EV-07	Nombre del usuario propietario de la valoración	Cadena de caracteres	NO	NO	Nombre asignado al perfil de usuario del propietario de la valoración	

Tabla 163: Diccionario de datos - Valoración

### 5.4.6. TRANSACCIÓN

TRANSACCIÓN						
Entidad que representa a todas las transacciones realizadas entre un usuario vendedor y un usuario cliente.						
ID	Nombre	Tipo	Nulo	Unique	Descripción	Comentario
RI-ET-01	UID transacción	Cadena de caracteres	NO	SÍ	Identificador de la transacción	

### 5.4.7. MENSAJE

MENSAJE						
Entidad que representa a cada mensaje enviado a través del servicio de mensajería instantánea entre un usuario cliente y un usuario vendedor						
ID	Nombre	Tipo	Nulo	Único	Descripción	Comentario
RI-EM-01	UID Usuario emisor	Cadena de caracteres	NO	SÍ	Identificador del usuario emisor del mensaje	
RI-EM-02	UID Usuario receptor	Cadena de caracteres	NO	SÍ	Identificador del usuario receptor del mensaje	
RI-EM-03	Mensaje	Cadena de caracteres	SÍ	NO	Contenido del mensaje enviado por el usuario emisor	
RI-EM-04	Fecha de envío	Date	NO	NO	Fecha de envío del mensaje	Formato: dd/MM/yyyy HH:MM:SS
RI-EM-05	Estado de mensaje: leído/no leído	Boolean	NO	NO	Estado del mensaje, respecto a si ha leído o no ha leído el mensaje el usuario receptor	
RI-EM-06	URL/s de imágenes	Cadena de caracteres	NO	NO	URL de contenido multimedia enviado mediante un mensaje	

Tabla 165: Diccionario de datos - Mensaje

### 5.4.8. CHAT

CHAT						
Entidad que representa a cada una de las conversaciones activas mediante el servicio de mensajería instantánea.						
ID	Nombre	Tipo	Nulo	Único	Descripción	Comentario
RI-ECH-01	ID del Chat	Cadena de caracteres	NO	SÍ	Identificador del chat abierto entre los dos usuarios	Se trata de la concatenación de los UID de ambos usuarios PRIMARY KEY
RI-ECH-02	UID Usuario vendedor	Cadena de caracteres	NO	SÍ	Identificador del usuario vendedor	
RI-ECH-03	UID Usuario cliente	Cadena de caracteres	NO	SÍ	Identificador del usuario cliente	
RI-ECH-04	Último mensaje recibido	Cadena de caracteres	NO	NO	Contenido del último mensaje enviado o recibido en el chat	

<b>RI-ECH-05</b>	Fecha del último mensaje	Date	SÍ	NO	Fecha de envío del último mensaje disponible en el historial del chat	Formato: dd/MM/yyyy HH:MM:SS
<b>RI-ECH-06</b>	UID último emisor de mensaje	Cadena de caracteres	NO	SÍ	Identificador del usuario que ha enviado el último mensaje	
<b>RI-ECH-07</b>	Fecha última modificación	Cadena de caracteres	NO	NO	Fecha de la última modificación que se ha realizado en el chat	Formato: dd/MM/yyyy HH:MM:SS

Tabla 166: Diccionario de datos - Chat

#### 5.4.9. UBICACIÓN

<b>UBICACIÓN</b>						
Entidad que representa la ubicación actual del usuario que se encuentra compartiendo o utilizando el sistema se seguimiento en tiempo real.						
ID	Nombre	Tipo	Nulo	Único	Descripción	Comentario
<b>RI-EUB-01</b>	Coordenadas físicas ubicación inicial	Geopoint	NO	NO	Coordenadas físicas correspondientes a la latitud y longitud de la ubicación inicial del usuario	
<b>RI-EUB-02</b>	Coordenadas físicas ubicación de destino	Geopoint	NO	NO	Coordenadas físicas correspondientes a la latitud y longitud de la ubicación de destino del usuario	

Tabla 167: Diccionario de datos - Ubicación

#### 5.4.10. RUTA

<b>VALORACIÓN</b>						
Entidad que representa la ruta trazada por el sistema de seguimiento en tiempo real entre la ubicación inicial y la ubicación de destino escogida por el usuario.						
ID	Nombre	Tipo	Nulo	Único	Descripción	Comentario
<b>RI-ER-01</b>	Coordenadas físicas ubicación inicial	Geopoint	NO	NO	Coordenadas físicas correspondientes a la latitud y longitud de la ubicación inicial del usuario	
<b>RI-ER-02</b>	Coordenadas físicas ubicación de destino	Geopoint	NO	NO	Coordenadas físicas correspondientes a la latitud y longitud de la ubicación de destino del usuario	
<b>RI-ER-03</b>	Puntos de trazado	Geopoint	NO	NO	Coordenadas físicas intermedias que permiten realizar el	

					trazado al unir dichas coordenadas	
<b>RI-ER-04</b>	Duración	TimeStamp	NO	NO	Tiempo que tarda el usuario en recorrer el trazado generado	
<b>RI-EV-05</b>	Distancia	Float	SÍ	NO	Distancia entre la ubicación inicial y la ubicación de destino	
<b>RI-EV-06</b>	Nombre ubicación	Cadena de caracteres	NO	NO	Nombre de la ubicación mostrada	
<b>RI-EV-07</b>	Descripción ubicación	Cadena de caracteres	NO	NO	Descripción con datos de interés sobre la ubicación	

Tabla 168: Diccionario de datos - Ruta

## 5.5 DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia son una representación gráfica encargada de representar las transacciones de iteraciones que se producen entre los actores y los distintos objetos que intervienen en un escenario concreto. Permiten describir la secuencia de mensajes que intercambian estos objetos realizados en cada acción.

La representación se realiza mediante un rectángulo y una línea temporal vertical por cada uno de los actores que intervienen en el escenario. Tras ello se establece un orden y cada iteración de transacción se representa por líneas horizontales que unen cada una de las líneas temporales de los objetos implicados. En este caso, se ha decidido representar tres diagramas de secuencia como referencia al funcionamiento del resto de ellos dentro de la plataforma.

### 5.5.1. RU-02: REGISTRARSE EN EL SISTEMA

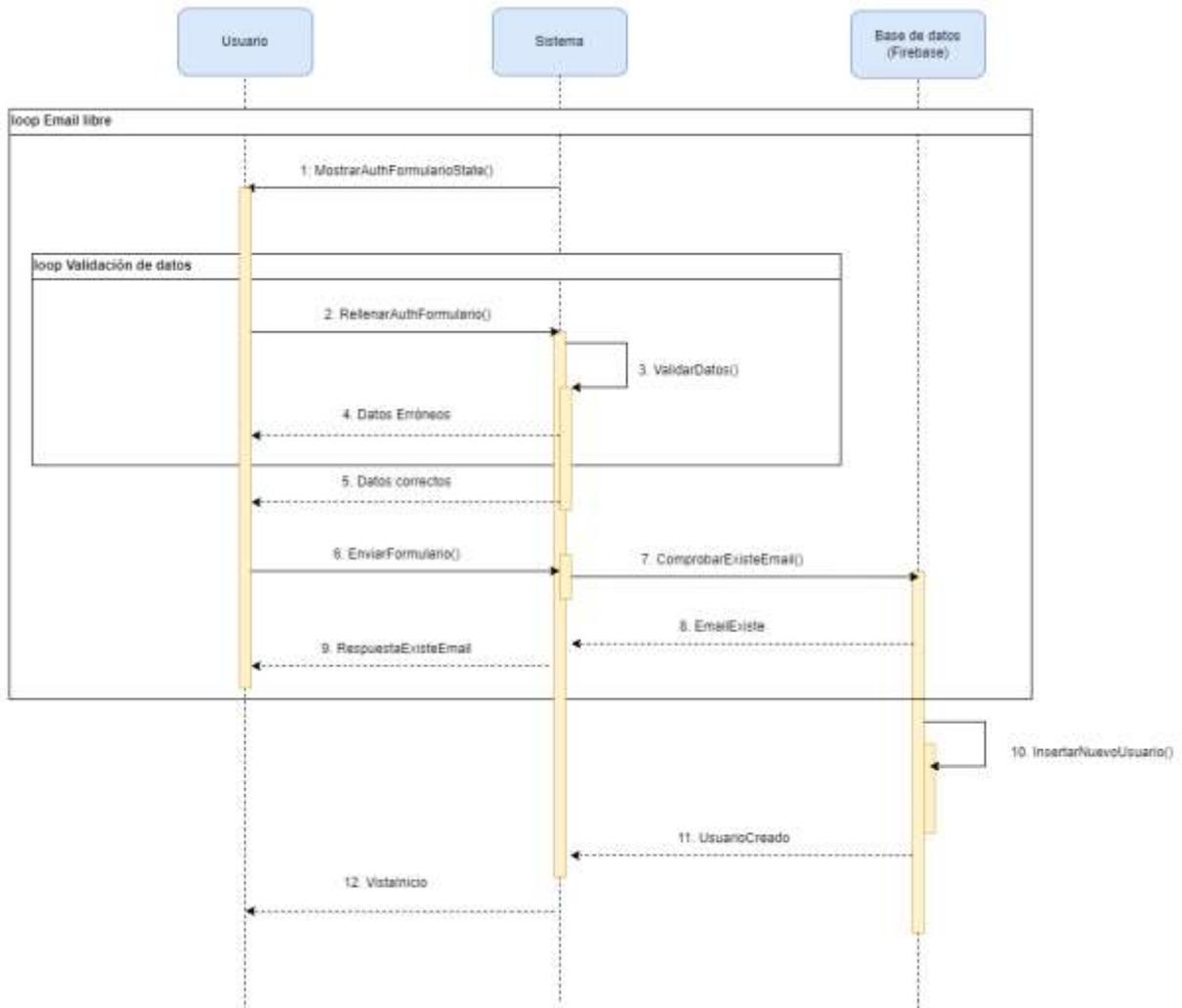


Figura 42: Diagrama de secuencia - RU-02 Registrarse en el sistema

## 5.5.2. RU-62: AÑADIR VALORACIÓN

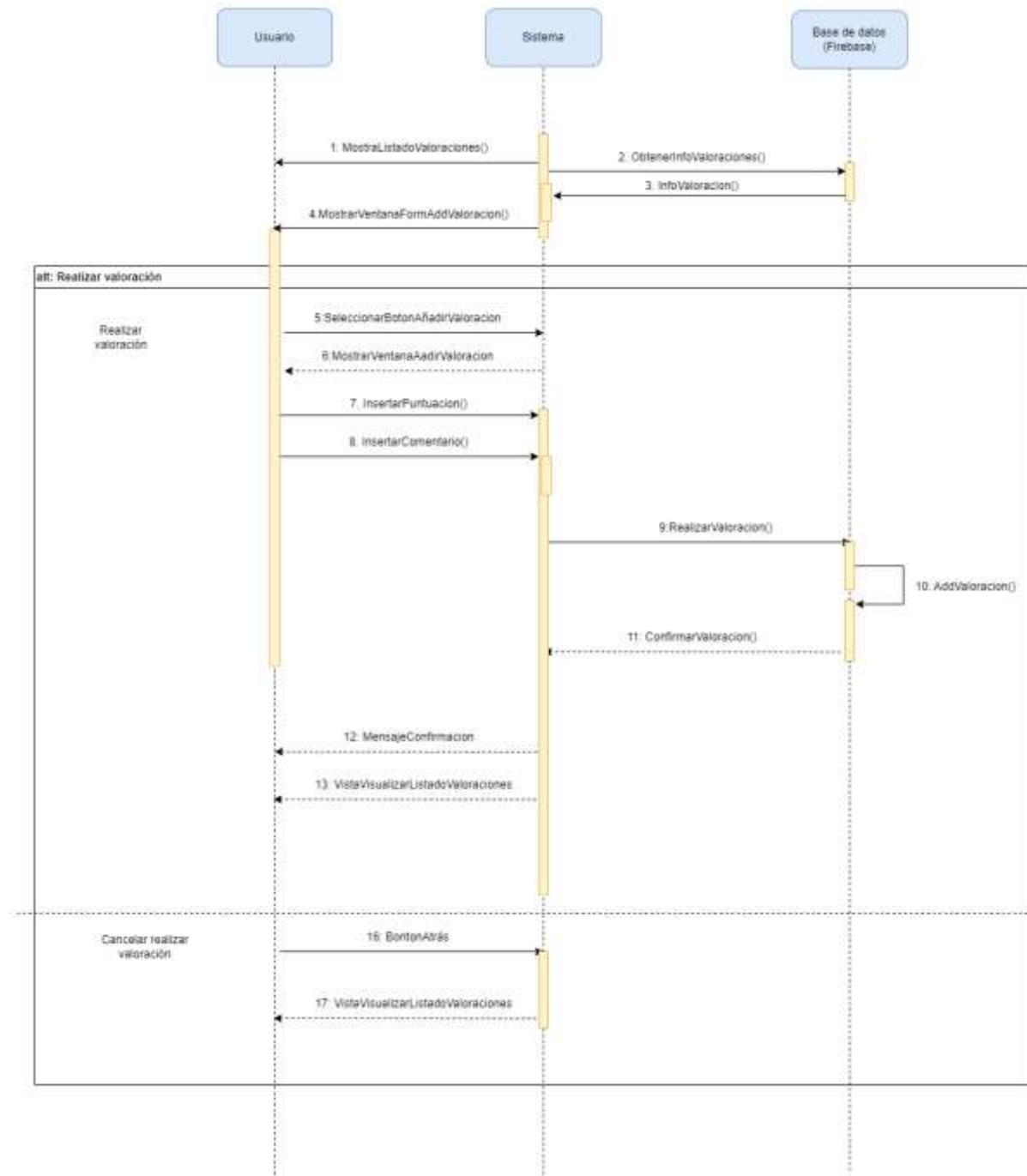


Figura 43: Diagrama de secuencia - Añadir valoración

### 5.5.3. RU-43: MODIFICAR PUBLICACIÓN

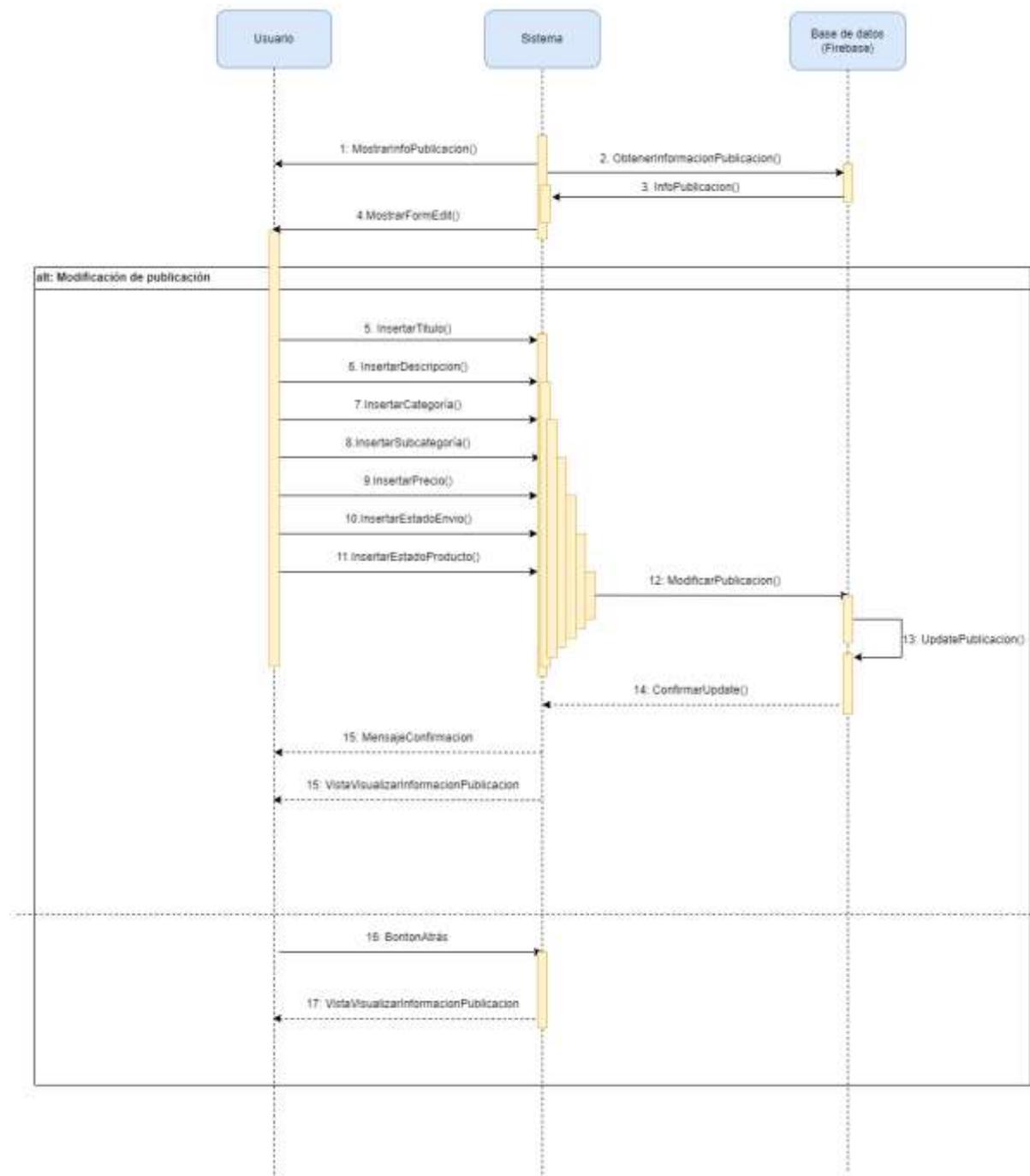


Figura 44: Diagrama de secuencia- RU:43 Modificar publicación

### 5.5.4 RU-85: ACCEDER A SISTEMA DE SEGUIMIENTO EN TIEMPO REAL

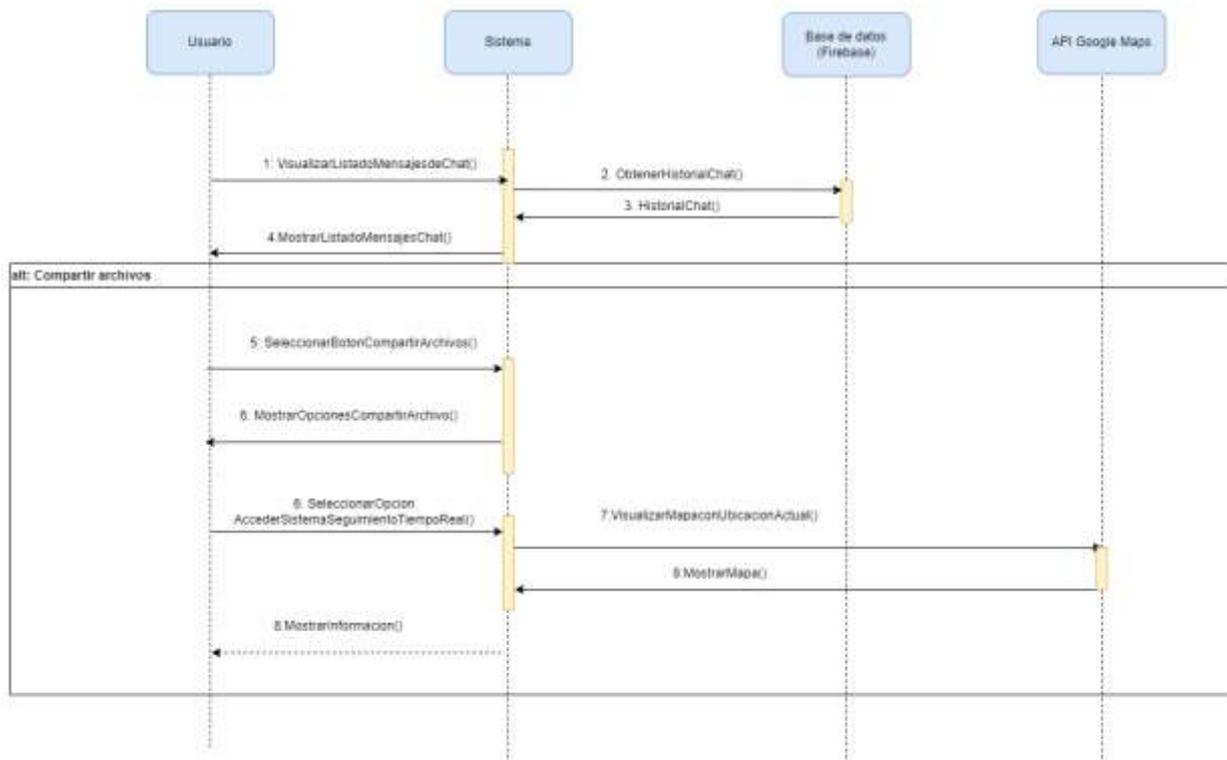


Figura 45: Diagrama de secuencia - Acceder a sistema de seguimiento en tiempo real

## 5.6 DISEÑO DE LA INTERFAZ DE USUARIO

En esta sección, se ha considerado que, debido a la gran extensión de esta memoria, y por no aumentar aún más su tamaño, no incluir ningún boceto de la interfaz de usuario, dado que disponemos ya de una representación gráfica de la interfaz de usuario final en la *sección 8.2. Manual de usuario*. También podemos encontrar información sobre el diseño e implementación de la interfaz de usuario en base a los widgets y su implementación en *Flutter*, en la *sección 6.5 Detalles de implementación*.

---

## 6. IMPLEMENTACIÓN

---

En primer lugar, se va a detallar los aspectos de implementación más reseñables de la aplicación desarrollada. Además, se describirá la estructura interna de directorios en el proyecto.

## 6.1 REQUISITOS MÍNIMOS

En este apartado, se describirán los requisitos mínimos exigibles para la instalación o despliegue de la aplicación desarrollada para dispositivos móviles, tanto con sistema operativo Android o como con iOS.

REQUISITOS MÍNIMOS		
<b>SOFTWARE</b>	<b>Sistema operativo</b>	Android 5.0 o iOS 11.0
<b>HARDWARE</b>	<b>Acceso a funcionalidades propias del dispositivo</b>	Ubicación, Cámara, Galería, etc.
	<b>Otros</b>	Compatibilidad con servicios de Google
	<b>Memoria RAM</b>	2GB o superior
	<b>CPU</b>	Quad-Core 1,2GHz o superior
	<b>Memoria interna</b>	16GB
	<b>Conectividad</b>	Conectividad por red WiFi o red LTE 4G

Tabla 169: Requisitos mínimos Hardware y Software

## 6.2 HERRAMIENTAS UTILIZADAS

A continuación, se realiza la descripción de las diferentes herramientas utilizadas para el desarrollo del proyecto, además de las tecnologías utilizadas enumeradas anteriormente, en la *Sección "1.4 Entorno tecnológico"*.

### 6.2.1 HERRAMIENTAS DE DESARROLLO SOFTWARE MÓVIL

- ❖ **Visual Studio Code:** Entorno de desarrollo oficial (*IDE, Intergrated Development Enviroment*), que permite la elaboración de software o aplicación para distintas plataformas. Se trata de un editor de código fuente desarrollado por Microsoft. Se caracteriza por ser un editor ligero, altamente personalizable y de código abierto. Está diseñado para ser multiplataforma y es compatible con Windows, macOS y Linux. Destaca por ser un editor de código fuente potente y versátil, que se ha vuelto muy popular entre los desarrolladores debido a su flexibilidad, rendimiento y amplia gama de características.

VS Code ofrece una amplia gama de funciones y extensiones que lo hacen adecuado para diferentes lenguajes de programación y entornos de desarrollo. Se caracteriza por:

- Resaltado de sintaxis, autocompletado inteligente, depuración integrada, terminal integrada e interfaz de usuario intuitiva.
- Sistema de compilación basado en Gradle

- Emuladores configurables, lo que permite al desarrollador, disponer de una gran capacidad para realizar pruebas de software.
  - Integración con sistemas de control de versiones, especialmente GitHub, facilitando su control, implementación y compilación de funciones.
  - Compatibilidad con Google Services mediante Google Cloud Platform, lo que permite la integración de sus servicios en las aplicaciones desarrolladas.
  - Capacidad de admitir una gran variedad de lenguajes y *frameworks* populares.
  - Instalación de extensiones desarrolladas por la comunidad, lo que amplía aún más su funcionalidad y lo hace adaptable a las necesidades de cada desarrollador.
- ❖ **Firestore:** se trata de una plataforma de desarrollo de aplicaciones móviles y web ofrecida por Google. Proporciona una amplia gama de herramientas y servicios que permiten a los desarrolladores crear, mejorar y escalar sus aplicaciones de manera rápida y eficiente. Sus características más destacadas son:
- Almacenamiento en la nube, autenticación de usuarios, bases de datos en tiempo real, notificaciones push, hosting de aplicaciones, análisis de datos, pruebas.

En el caso de esta aplicación, se han implementado los siguientes servicios:

- **Firestore Authentication:** servicio de autenticación proporcionado por *Firestore*, que proporciona una integración de funcionalidades de autenticación a la aplicación, permitiendo a los usuarios realizar acciones de registro, autenticación dinámica y administración de credenciales de forma segura.

Este servicio admite diferentes métodos de autenticación, incluidos el inicio de sesión con correo electrónico y contraseña, el inicio de sesión con proveedores de identidad populares como Google, Facebook, Twitter y GitHub, así como la autenticación mediante SMS y autenticación anónima.

Una vez que los usuarios se autentican en una aplicación utilizando *Firestore Authentication*, se generan *tokens* de acceso seguros que se pueden utilizar para autorizar el acceso a los recursos y datos de la aplicación. Esto facilita la implementación de funciones como la protección de rutas o páginas específicas, la personalización de la experiencia del usuario y la gestión de permisos y roles.

- **Firestore Cloud Firestore:** servicio *cloud* que se encarga de proporcionar una base de datos NoSQL, escalable y flexible, diseñada para almacenar y sincronizar datos en tiempo real en aplicaciones móviles y servidores.

Implementa una estructura de colecciones y documentos para organizar los datos. Los documentos son contenedores de información y se organizan en colecciones. Cada documento contiene campos con valores asociados, lo que permite almacenar datos estructurados.

Una de las principales características de *Firestore* es su capacidad para sincronizar datos en tiempo real entre diferentes dispositivos y plataformas. Esto significa que los cambios realizados en un dispositivo se reflejan automáticamente en otros dispositivos conectados a la misma base de datos.

Además, *Firestore* ofrece una amplia gama de características adicionales, como consultas flexibles para buscar y filtrar datos, reglas de seguridad para proteger la base de datos y la integración con otras herramientas de Firebase, como la autenticación de usuarios y el almacenamiento de archivos.

- **Firestore Cloud Functions:** servicio de computación en la nube sin servidor. Las *Firestore Cloud Functions* permiten escribir y desplegar pequeñas piezas de código, es decir, funciones, en la nube que se ejecutan automáticamente en respuesta a eventos específicos. Estos eventos pueden ser acciones realizadas en otras partes de la plataforma *Firestore*, como cambios en la base de datos en tiempo real, autenticación de usuarios, almacenamiento de archivos o solicitudes HTTP.

La idea principal detrás de *las Firestore Cloud Functions* es permitir la ejecución de lógica personalizada en la nube, sin tener que preocuparse por la infraestructura subyacente. Las funciones se pueden escribir en JavaScript, TypeScript, o incluso en otros lenguajes a través del uso de entornos de ejecución personalizados.

Estas funciones se escalan automáticamente bajo la demanda, lo que significa que *Firestore* se encarga de administrar la capacidad de cómputo necesaria para ejecutar las funciones de manera eficiente. Además, *las Firestore Cloud Functions* están estrechamente integradas con otros servicios de *Firestore*, lo que permite una fácil interacción con otros componentes de la plataforma y simplifica el desarrollo de aplicaciones completas.

- **Firestore Cloud Messaging (FCM):** servicio de mensajería en la nube, que permite enviar mensajes y notificaciones a dispositivos móviles y aplicaciones web de manera confiable y escalable.

Da la posibilidad de enviar mensajes a dispositivos móviles y aplicaciones web en tiempo real, incluso cuando la aplicación no está en primer plano o está cerrada. Esto permite la entrega de notificaciones *push*, mensajes personalizados y actualizaciones en tiempo real a los usuarios de las aplicaciones.

FCM es compatible con plataformas como Android, iOS y aplicaciones web, y ofrece múltiples opciones de mensajería, incluyendo notificaciones *push*, mensajes de datos y mensajes con estado. Los desarrolladores pueden utilizar la consola de *Firestore* o la API de FCM para enviar mensajes a usuarios individuales o grupos de usuarios.

❖ **Maps SDK for Android/iOS:** biblioteca que contiene un conjunto de herramientas y funciones, proporcionadas por *Google Cloud Platform*, destinadas a la integración de mapas interactivos y servicios de ubicación en aplicaciones móviles. Permite mostrar mapas, mostrar marcadores, dibujar rutas, realizar búsquedas de lugares y utilizar funciones de geolocalización. El SDK utiliza la API de *Google Maps* para obtener datos y servicios de mapas, lo que permite crear aplicaciones personalizadas con características basadas en la ubicación.

- **Geocoding API:** servicio externo que tiene la función de traducir coordenadas geográficas en direcciones físicas, y el proceso inverso.

- **Geolocation API:** servicio externo que permite realizar la búsqueda de una localización utilizando las torres de comunicación y los puntos de acceso Wifi cercanos.
  - **Places API:** servicio externo que se encarga de obtener información sobre los puntos de interés almacenados en la propia base de datos de Google.
- ❖ **API MapBox SDK:** plataforma de servicios potente y versátil de mapas en línea que proporciona herramientas y APIs con el fin de integrar mapas interactivos en aplicaciones y sitios web. La API de *Mapbox* permite acceder a una amplia gama de funciones relacionadas con mapas, como la visualización de mapas, la geocodificación (conversión de direcciones en coordenadas geográficas) y la generación de direcciones y rutas. La API también ofrece capacidades avanzadas, como la personalización de estilos de mapas, la superposición de datos geoespaciales y la creación de experiencias de realidad aumentada.

## 6.2.2 HERRAMIENTAS DE SOPORTE

- ❖ **Microsoft Word:** herramienta desarrollada por Microsoft para el procesamiento de textos. Es parte de la *suite* de productividad de Microsoft Office y se utiliza para crear, editar, formatear y compartir documentos de texto. Word ofrece una amplia gama de herramientas y funciones para el procesamiento de texto, como opciones de formato, estilos, tablas, gráficos, autocorrección, revisión de ortografía y gramática, entre otras. En DaSell ha sido utilizado para el desarrollo de esta memoria.
- ❖ **Microsoft Project:** aplicación de gestión y planificación de proyectos desarrollada por Microsoft. Es un software ampliamente utilizado para planificar, administrar y controlar proyectos de diversa índole. Proporciona herramientas para la programación de tareas, asignación de recursos, seguimiento del progreso, establecimiento de dependencias entre actividades y generación de informes. Microsoft Project permite a los usuarios visualizar el cronograma del proyecto, asignar recursos a tareas específicas, establecer hitos y realizar un seguimiento detallado del progreso del proyecto. Se ha implementado para el desarrollo de la planificación y el seguimiento del proyecto.
- ❖ **Adobe Acrobat Reader:** software desarrollado por Adobe Systems, que permite crear, editar y visualizar archivos en formato PDF. En este proyecto su cometido ha sido la creación, lectura y control de archivos PDF.
- ❖ **GitHub:** plataforma de desarrollo colaborativo basada en la web que utiliza el sistema de control de versiones Git. Proporciona herramientas para gestionar y colaborar en proyectos de software de manera eficiente. GitHub permite alojar, revisar y controlar los cambios en el código fuente, así como colaborar con otros miembros del equipo a través de solicitudes de extracción y comentarios. También ofrece funciones adicionales, como seguimiento de problemas, administración de proyectos y despliegue automatizado, lo que lo convierte en una plataforma integral para el desarrollo de software en equipo.
- ❖ **Google Drive:** servicio en línea proporcionado por Google que permite a los usuarios almacenar, sincronizar y compartir archivos en la nube. Es una plataforma de almacenamiento en la nube que ofrece espacio gratuito y opciones de almacenamiento adicional mediante planes de pago. Los usuarios pueden acceder a sus archivos en Google Drive desde cualquier dispositivo con conexión a internet, lo que les permite organizar y gestionar sus documentos, fotos, videos y otros archivos de manera conveniente. Además, Google Drive permite la colaboración en tiempo real, lo que facilita el trabajo en equipo y la edición compartida de documentos. Este potencial ha permitido en este desarrollo

realiza *backups* de la memoria y toda la documentación generada, así como almacenar todos los diagramas realizados.

- ❖ **Draw.io:** herramienta *online* que permite crear diagramas y gráficos de manera fácil y rápida. Es una aplicación basada en la web que ofrece una interfaz intuitiva y una amplia gama de elementos gráficos predefinidos para ayudar a los usuarios a crear diagramas de flujo, organigramas, diagramas de red, diagramas de procesos y muchos otros tipos de diagramas. Destaca por ser gratuita, de código abierto y compatible con diferentes plataformas, lo que permite a los usuarios crear y colaborar en diagramas desde cualquier lugar utilizando su navegador web. Además, ofrece opciones para exportar y compartir los diagramas creados en varios formatos, lo que facilita su integración en otros documentos o herramientas de trabajo.
- ❖ **Postman:** herramienta de desarrollo de API que permite probar, documentar y colaborar en el proceso de construcción de API. Proporciona una interfaz de usuario intuitiva que permite enviar solicitudes HTTP a diferentes endpoints y recibir respuestas para probar y depurar las API. Además, Postman ofrece características avanzadas como la capacidad de crear y guardar solicitudes, automatizar flujos de trabajo, simular servidores y colaborar con otros desarrolladores en el diseño y desarrollo de API.
- ❖ **DartPad:** entorno de desarrollo en línea (IDE) basado en la web para el lenguaje de programación Dart. Proporciona un espacio para escribir, ejecutar y depurar código Dart sin necesidad de configurar un entorno de desarrollo local. *DartPad* ofrece una interfaz de usuario sencilla y cuenta con funciones útiles como resaltado de sintaxis, autocompletado de código, historial de cambios y la capacidad de compartir y colaborar en código con otros desarrolladores en tiempo real. Es una herramienta útil para experimentar con el lenguaje Dart, prototipar rápidamente ideas y aprender la sintaxis y las características de Dart.

### 6.3. TECNOLOGÍAS IMPLEMENTADAS

- ❖ **Flutter:** framework de código abierto desarrollado por Google para crear aplicaciones multiplataforma de alta calidad. Utiliza el lenguaje de programación Dart y ofrece un enfoque de desarrollo rápido y eficiente. Flutter se destaca por su capacidad para construir interfaces de usuario atractivas y fluidas, con una amplia gama de *widgets* personalizables. Además, permite compilar aplicaciones nativas tanto para dispositivos iOS como Android, lo que significa que una sola base de código puede utilizarse para crear aplicaciones para diferentes plataformas. Flutter también ofrece un rendimiento excepcional gracias a su motor de renderizado propio, lo que contribuye a una experiencia de usuario fluida y receptiva.
- ❖ **Dart:** lenguaje de programación desarrollado por Google que se utiliza principalmente para construir aplicaciones móviles, de escritorio y web. Es un lenguaje de programación de alto nivel, orientado a objetos y con una sintaxis similar a otros lenguajes como C, Java y JavaScript. Dart fue diseñado para ser fácil de aprender y productivo, y está diseñado para funcionar tanto en el lado del cliente como en el lado del servidor. Además, Dart viene con un conjunto de herramientas y bibliotecas estándar que facilitan el desarrollo de aplicaciones en diferentes plataformas. Una de las características destacadas de Dart es su compilador *just-in-time (JIT)* que permite una ejecución rápida y eficiente del código, así como su capacidad para ser compilado previamente (AOT) para lograr un rendimiento aún mayor. Dart se utiliza en el *framework* Flutter, que es una herramienta popular para la construcción de aplicaciones móviles multiplataforma.

- ❖ **JSON:** formato ligero de intercambio de datos que se utiliza ampliamente en la comunicación entre sistemas. Se basa en una estructura de texto legible por humanos que representa objetos y conjuntos de datos organizados en pares clave-valor. Cada par clave-valor está separado por comas, y los objetos se delimitan con llaves { }. Los valores pueden ser cadenas de texto, números, booleanos, arrays, otros objetos JSON o el valor nulo. JSON es ampliamente utilizado en aplicaciones web y en servicios de API para enviar y recibir datos estructurados de manera eficiente. Es un formato independiente del lenguaje y es fácilmente legible y escribible por humanos y máquinas.

## 6.4. LIBRERIAS Y DEPENDENCIAS

Además de las herramientas y tecnologías descritas anteriormente, junto con las librerías que permiten mantener estos servicios operativos, se dispone de un conjunto de paquetes que permiten incluir funcionalidades. La mayoría de estos paquetes están desarrollados por terceros y se encuentran en el repositorio oficial de paquetes para aplicaciones Dart y Flutter<sup>10</sup>.

Junto a las librerías necesarias para contar con los servicios operativos de Firebase, PayPal, Maps y Gmail en la aplicación, también se cuenta con:

- ❖ dependencies:
  - flutter: sdk: flutter
  - gap
  - cupertino\_icons
  - cloud\_firestore
  - firebase\_auth
  - flutter\_svg
  - provider
  - image\_picker
  - path\_provider
  - path
  - location
  - geolocator
  - multi\_image\_picker2
  - firebase\_storage
  - carousel\_slider
  - flutter\_signin\_button
  - google\_sign\_in
  - intl
  - theme\_provider
  - shared\_preferences
  - firebase\_messaging
  - get\_it

---

<sup>10</sup> Repositorio oficial de librerías y dependencias para Dart y Flutter ( <https://pub.dev/> )

- `firebase_core`
- `http`
- `google_maps_flutter`
- `cached_network_image`
- `font_awesome_flutter`
- `pull_to_refresh`
- `google_fonts`
- `animate_do`
- `bloc`
- `dio`
- `equatable`
- `flutter_bloc`
- `google_polyline_algorithm`
- `permission_handler`
- `emoji_picker_flutter`
- `flutters_toast`
- `grouped_list`
- `auto_size_text`
- `geocoder2`
- `lottie`
- `image_cropper`
- `photo_view`
- `flutter_app_badger`
- `flutter_local_notifications`
- `timeago`
- `flutter_rating_bar`
- `flutter_launcher_icons`

## 6.5. DETALLES DE IMPLEMENTACIÓN

En este apartado, para llevar a cabo de la tarea de implementación de la aplicación, hay que tener en cuenta, en primer lugar, los objetivos y requisitos definidos previamente, cuestiones y acciones previas que indican cómo se debe realizar el proyecto correctamente. Estos requisitos y objetivos se encuentran definidos los capítulos anteriores (véase Capítulos 1 y 4).

Las principales funcionalidades que se destacan en la aplicación son: el sistema de gestión de realización de publicaciones, el uso del servicio de mensajería instantánea y la implementación del sistema de seguimiento en tiempo real, adjunto al servicio de mensajería instantánea.

Por lo tanto, la aplicación debe ser capaz de cumplir todas las funcionalidades y características planteadas. Por esta razón, será primordial disponer de un *backend* robusto sobre el que se apoye el proyecto. Para ello, se contará con la API de Firebase, mediante el servicio *Firebase Auth*, el cual permite realizar la gestión de usuarios de la plataforma, tanto en funcionalidades de autenticación mediante la aplicación, autenticación mediante la plataforma Google o registro de usuarios.

A su vez, el proyecto debe contar con una base de datos que permita acceder a toda la información necesaria para realizar el proceso de publicaciones en tiempo real. Este servicio será proporcionado por el API de Firebase, mediante el servicio de base de datos *Cloud Firestore*.

También, durante este proceso de publicación se debe contar una plataforma que nos permita acceder e integrar mapas interactivos donde poder indicar la ubicación o poder visualizar la ubicación asignada a una publicación. Esta característica se implementa mediante la API de Google Maps, en concreto con *Maps SDK for Android/iOS*, integrada en Google Cloud Platform. Este sistema nos permite recoger y procesar dicha información y volcarla al servicio de base de datos *Cloud Firestore*, asegurando al usuario seguridad y control en el tratamiento de estos datos.

Los servicios API descritos anteriormente también se deben consumir en la implementación de la funcionalidad de Sistema de Seguimiento en tiempo real, que también se apoya en la API de MapBox, que nos permite obtener toda la información relativa al trazado de rutas e ubicación del usuario.

Las demás funcionalidades que deben satisfacer el resto de objetivos y requisitos relacionados, quedan respaldadas mediante las APIs de Firebase: *Firebase Auth*, *Cloud Firestore*, *Firebase Cloud Messaging* y *Firebase Cloud Functions*. A continuación, se realizará una explicación más detallada sobre su puesta en marcha e implementación en el proyecto.

## 6.5.1. DESCRIPCIÓN DEL SERVICIO DE BASE DE DATOS

### 6.5.1.1. FIREBASE

#### ¿QUÉ ES FIREBASE?



Figura 46: Logo Plataforma Firebase

*Firebase* es una plataforma de desarrollo de aplicaciones en la nube, integrada con *Google Cloud Platform*, que proporciona una amplia gama de herramientas y servicios para ayudar a los desarrolladores a crear aplicaciones web y móviles de alta calidad. Fue adquirida por Google en 2014 y desde entonces ha experimentado un crecimiento significativo, adquiriendo gran relevancia en el mercado.

La plataforma se basa en un modelo de distribución de tipo SaaS (*Software as a Service*) debido a que su soporte lógico y los datos se encuentran bajo la infraestructura de Google.

Los servicios y soluciones que incluye buscan satisfacer las siguientes necesidades, y aportan una serie de ventajas al usar la plataforma:

- Sincronizar de manera sencilla los datos del proyecto, sin necesidad de tener que administrar conexiones o escribir lógica de sincronización compleja.
- Implementa un conjunto de herramientas multiplataforma con fácil integración en la plataforma móvil.

- Se distribuye sobre la infraestructura de Google, y permite escalabilidad independientemente del tamaño y potencia que tenga la aplicación desarrollada.
- Permite crear el proyecto sin necesidad de la obligación de desplegarla mediante un servidor, las herramientas se incluyen en los SDK para los dispositivos móviles de los usuarios, por lo que no es necesario la creación de un servidor para el proyecto.

Para cumplir estas características, los servicios suministrado se clasifican en distintos fundamentos:

1. **Crecimiento:** herramientas destinadas a la comunicación y captación de usuarios, además de herramientas que son capaces de pronosticar comportamientos o situaciones futuras.
2. **Desarrollo:** herramientas *software* implementadas en el desarrollo y fundamentos de la aplicación, encargada de dar soporte de almacenamiento de datos, procesos de autenticación, hosting y demás configuraciones.
3. **Analítica:** La herramienta disponible en la plataforma, *Google Analytics*, monitoriza y genera informes del comportamiento y estadísticas de la aplicación.
4. **Calidad:** herramientas centradas en mantenimiento, realización de pruebas y distribución de la plataforma.



Figura 47: Módulos de Firebase

Firebase ofrece una variedad de servicios que abarcan diferentes áreas del desarrollo de aplicaciones. Estos servicios incluyen:

1. **Firebase Authentication:** Proporciona un sistema de autenticación seguro y fácil de usar que permite a los desarrolladores autenticar a los usuarios de sus aplicaciones mediante métodos como correo electrónico/contraseña, autenticación mediante redes sociales (por ejemplo, Google, Facebook, Twitter) y autenticación mediante proveedores de identidad federados (por ejemplo, OAuth).

**2. Firebase Realtime Database:** Es una base de datos en tiempo real en la nube que permite a los desarrolladores almacenar y sincronizar datos en tiempo real entre los clientes y el servidor. Los cambios en los datos se propagan automáticamente a todos los clientes conectados, lo que permite construir aplicaciones colaborativas y en tiempo real.

**3. Firebase Cloud Firestore:** Es una base de datos NoSQL flexible y escalable que permite a los desarrolladores almacenar, sincronizar y consultar datos estructurados en documentos, colecciones y subcolecciones. Firestore ofrece una sincronización en tiempo real similar a la de la Realtime Database, pero con una estructura de datos más avanzada y consultas más potentes.

**4. Firebase Storage:** Proporciona un sistema de almacenamiento en la nube para guardar y servir archivos estáticos como imágenes, videos y otros archivos multimedia. Los desarrolladores pueden cargar y descargar archivos de forma segura y fácil, y también pueden controlar los permisos de acceso a los archivos.

**5. Firebase Cloud Messaging:** Es un servicio de mensajería en la nube que permite a los desarrolladores enviar notificaciones push a los usuarios de sus aplicaciones en dispositivos móviles y web. Es útil para enviar mensajes de alerta, actualizaciones y promociones a los usuarios de manera efectiva.

**6. Firebase Hosting:** Permite a los desarrolladores alojar y servir sus aplicaciones web estáticas, así como contenido dinámico generado por el lado del servidor. Proporciona una forma sencilla de implementar y gestionar aplicaciones web en producción con una configuración mínima.

Estos son solo algunos de los servicios más destacados que Firebase ofrece. La plataforma también incluye otras herramientas, como *Firebase Functions* (para ejecutar código en la nube), *Firebase Crashlytics* (para informes de errores y análisis de estabilidad de la aplicación) y *Firebase Analytics* (para rastrear el comportamiento de los usuarios y medir el rendimiento de la aplicación). *Firebase* se ha convertido en una opción popular para desarrolladores de aplicaciones que buscan una solución integral para el desarrollo, la gestión y el crecimiento de sus aplicaciones en la nube.

### 6.5.1.2. MOTIVOS ELECCIÓN DE FIREBASE EN EL DESARROLLO

Para concluir, dado que uno de los principales problemas que se producen en el desarrollo de aplicaciones, tanto móviles como web, es la escalabilidad, debido a la falta de adaptación al aumento de número de usuarios que se pueden producir a lo largo de la vida útil de una aplicación. En el caso de DaSell, la aplicación no está tan solo enfocada a un comercio local, si no que el proyecto está destinado a un comercio nacional, por lo que debe estar considerado que el proyecto debe soportar un número de usuarios ampo, lo que supone un aumento en el número de dispositivos y plataformas que deben asegurar el correcto funcionamiento de la aplicación.

A su vez, uno de los conceptos más importantes que debe gestionar la plataforma es suministrar, al sistema, la información y datos en tiempo real a todos los usuarios. Dado que uno de los objetivos principales a cubrir es la realización de publicaciones de manera rápida por parte del usuario, y la obtención de datos sobre ubicación en tiempo real, lo que requiere rapidez en la obtención de datos y que estos se encuentren accesibles de forma actualizada, dado que si no esto supondría un riesgo en la aplicación.

En definitiva, por estos diversos motivos se ha realizado la elección de la implementación de Firebase y los servicios que ofrece. Dado que Firebase, a su vez, está disponible para plataformas desarrolladas mediante programación web, lo que solventa el inconveniente de la escalabilidad. También, hay que destacar las

posibilidades que ofrece el servicio de base de datos Cloud Firestore, que es el encargado de suministrar la información en tiempo real a los usuarios, potenciando la escalabilidad y sincronización, lo que permite resolver los principales problemas.

### 6.5.1.3. CLOUD FIRESTORE



Figura 48: Logo Cloud Firestore

Cloud Firestore se trata de un servicio de base de datos *cloud*, que se caracteriza principalmente por manejar un base de datos no relacional, es decir, NoSQL, flexible y escalable proporcionado por Firebase, que destaca por permitir una buena sincronización de datos en tiempo real con las plataformas desarrolladas, al contar con agentes de servicio de escucha de eventos en tiempo real, con la capacidad de notificar cualquiera de estos eventos sucedidos en la base de datos.

En este servicio, los datos se encuentran almacenados en **Documentos**, compuestos por campos que se jerarquizan mediante campo-valor, destacando que la tipología de datos admitida es muy amplia. Estos documentos se agrupan en **Colecciones** que permiten mayor claridad en la estructura de los datos y su organización. Estos documentos admiten a su vez subcolecciones dentro de las colecciones descritas previamente, junto con estructuras jerárquicas de datos que mejora la escalabilidad de la plataforma.

### 6.5.1.4. MOTIVOS ELECCIÓN CLOUD FIRESTORE

La plataforma tiene como objetivo principal ofrecer a los usuarios una aplicación que permita la realización de publicaciones y de transacciones entre usuarios, apoyada por funcionalidades como el servicio de mensajería instantánea integrado, sistema de seguimiento en tiempo real o sistema de valoraciones entre usuarios.

Todas estas funcionalidades requieren que los datos estén disponibles y sean persistentes para los usuarios continuamente, para que el usuario final de la aplicación sea capaz de acceder a los mismos en cualquier instante y ubicación en la que se encuentre con su dispositivo móvil. Dada esta casuística, las bases de datos relacionales no son recomendables o conllevan un esfuerzo exagerado para manejar las consultas, inserciones, modificaciones de datos. Teniendo en cuenta que la información de debe mostrar siempre actualizada en todo momento.

Por concluir, tras comprobar las distintas posibilidades para cubrir los requerimientos y condiciones que se impone, se ha escogido *Cloud Firestore*, como servicio de base de datos, y que proporciona una solución completa a lo descrito previamente y su dada su óptima integración con aplicaciones móviles.

Podemos describir las siguientes características principales de *Cloud Firestore*:

- **Flexibilidad:** Su modelo de datos admite estructuras de datos flexibles y jerárquicas. La información se almacena en documentos, organizados en colecciones. Dichos documentos pueden contener subcolecciones de forma anidada como datos complejos.

- **Actualizaciones en tiempo real:** *Cloud Firestore* usa la sincronización de datos para la actualización de datos en dispositivos con conectividad. Aun así, está planteado para ejecutar consultas de recuperación únicas y sencillas de manera eficiente.
- **Consultas expresivas:** se permite la implementación de las consultas para recuperar documentos específicos o para recuperar los documentos de una colección, que coincidan con los parámetros de la consulta. Dichas consultas pueden incluir distintos filtros en cadena y combinar los filtros con criterios de orden. Otra funcionalidad que se permite es la posibilidad de indexar de forma predeterminadas consultas, lo que provoca que el rendimiento de las consultas sea proporcional al tamaño del conjunto de resultados, no del conjunto de datos.
- **Asistencia sin conexión:** permite el almacenamiento en caché de los datos que usa la plataforma desplegada de forma activa, por lo que la aplicación puede escribir, leer, escuchar y consultar datos, aunque no se disponga de conectividad en el dispositivo móvil. Cuando dicho dispositivo se conecte a la red, *Cloud Firestore* realiza la sincronización de las modificaciones almacenadas de forma local y se vuelcan en el servicio *Cloud Firestore*.

## 6.5.2. IMPLEMENTACIÓN DEL SERVICIO DE BASE DE DATOS

Como se ha descrito anteriormente, la implementación del modelo de base de datos es esencial para el buen rendimiento de la aplicación, la cual se trata de una base de datos no relacional, NoSQL, bajo el servicio de Cloud Firestore, integrado en la plataforma de Firebase.

A continuación, se especifica el procedimiento seguido para la implementación del servicio de base de datos para que la configuración sea eficiente y con un buen rendimiento.

### 6.5.2.1. CREACIÓN DEL PROYECTO DENTRO DE FIREBASE

1. Se accede a la dirección [Plataforma Firebase](#), a través de un navegador web.
2. Seleccionamos la opción “Ir a la consola”, que se encuentra en la barra superior de la página de inicio.
3. Se realiza la autenticación en la plataforma mediante las credenciales del usuario.
4. Al acceder a la plataforma, se muestra la siguiente pantalla, véase *Figura 49: Proyectos asociados a la cuenta de desarrollo*, dónde se muestran los proyectos ya generados por el usuario dentro de la plataforma Firebase.

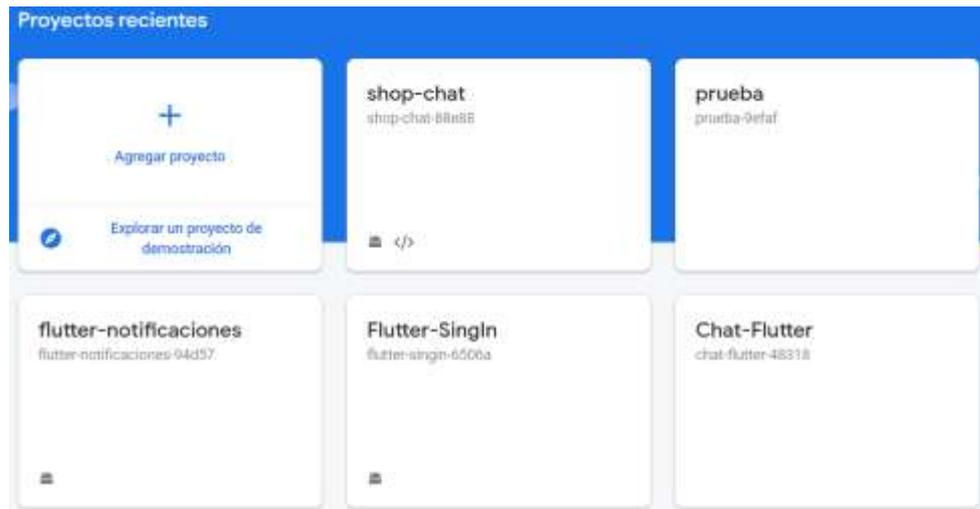


Figura 49: Proyectos asociados a la cuenta de desarrollo

- Para añadir o crear un nuevo proyecto en *Firebase*, se selecciona la opción “Añadir proyecto”, dónde a continuación se debe ir realizando distintos pasos y cumplimentado con la configuración general que deseamos asignar al proyecto ya la integración con servicios, véase Figuras 50 y 51.



Figura 50: Creación proyecto Firebase



Figura 51: Implementación Google Analytics

- Tras crear y configurar el proyecto, accedemos a la consola de control y gestión de los servicios de *Firebase*, los cuales aparecen en el menú lateral. Dependiendo del servicio seleccionado podemos acceder a sus respectivas configuraciones y nos muestra el contenido ofrecido por la plataforma.
- En el menú lateral, se muestran todos los servicios que ofrece la plataforma, los cuales se estructuran de la siguiente manera:
  - ❖ **Compilación:** En este apartado se encuentran los servicios más potentes y relativos al desarrollo de aplicaciones. Encontramos aquí los servicios implementados en DaSell de *Authentication, Cloud Firestore, Storage, Cloud Functions*.
  - ❖ **Lanzamiento y supervisión:** engloba los servicios cuyo cometido es el mantenimiento del software, con herramientas como reporte de informe de errores, pruebas de rendimiento y distribución dentro de la aplicación.

- ❖ **Analytics:** se incluyen los servicios destinados a la monitorización del *software* y mediciones de interacción de la herramienta con los usuarios.
- ❖ **Participación:** servicios enfocados a la comunicación con los usuarios de la aplicación, predicciones de comportamiento del sistema.

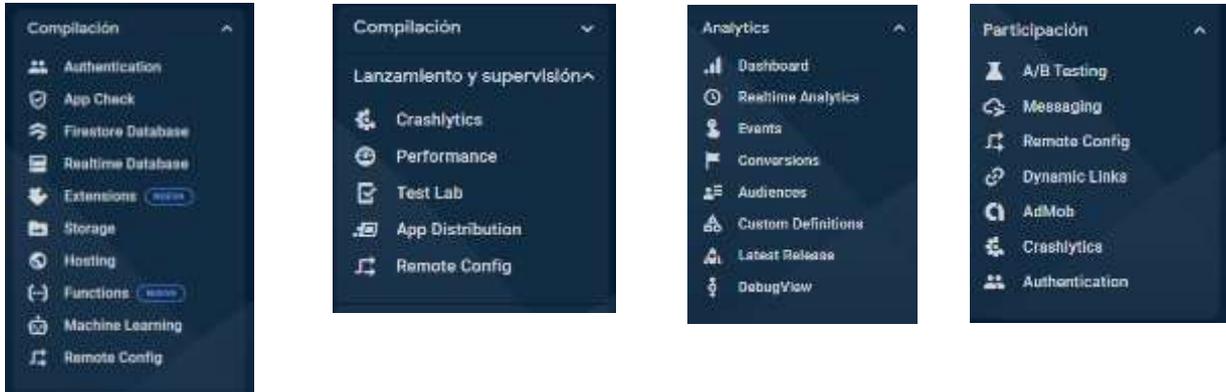


Figura 52: Organización herramientas Firebase

### 6.5.2.2. INTEGRACIÓN DEL SDK DE CLOUD FIRESTORE EN LA APLICACIÓN

A continuación, se enumeran los diferentes pasos a realizar para la integración del SDK:

1. El siguiente paso que se debe seguir en el procedimiento es integrar el servicio *Firebase* dentro de nuestro proyecto o software desarrollado. Para ello, seleccionamos el botón que permite integrarlo en un entorno para dispositivos móviles para desarrollos en *Flutter*, véase *Figura 53: Integrar Cloud Firestore en aplicación móvil*.

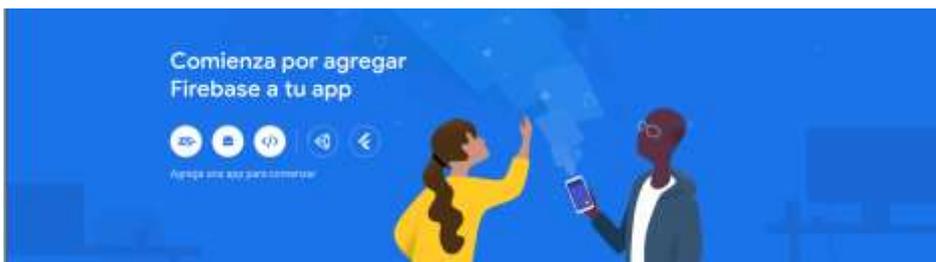


Figura 53: Integrar Cloud Firestore en aplicación móvil

2. A continuación, se introducen una serie de valores que se solicitan. El primer valor solicitado, es el nombre del paquete que tenemos asignado a nuestro proyecto en fase de desarrollo, un alias asignado y la firma digital que debemos generar mediante el protocolo SHA-1. A su vez se comprueba que disponemos del entorno *Firebase CLI* en el editor, siguiendo los pasos que se muestran en *la Figura 54: Configuración entorno*.

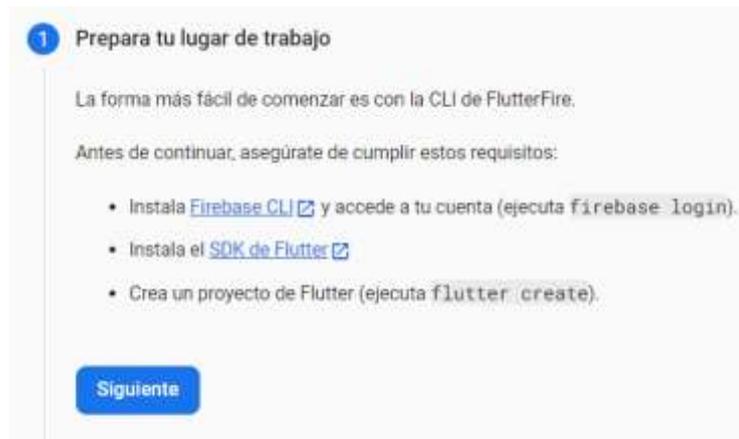


Figura 54: Configuración entorno

Se descarga el archivo "google-service.json", archivo en formato JSON generado y que necesitamos incorporar en el entorno del proyecto mediante *Visual Studio Code*, en el directorio *root* de la aplicación.

*FlutterFire* es una colección de complementos (*plugins*) para el desarrollo de aplicaciones móviles utilizando el *framework Flutter*. Estos complementos permiten aprovechar las funcionalidades de *Firebase*. Con *FlutterFire*, se puede integrar fácilmente características como autenticación de usuarios, almacenamiento en la nube, bases de datos en tiempo real, mensajería en la nube y análisis de datos en sus aplicaciones *Flutter*. Los complementos de *FlutterFire* proporcionan una capa de abstracción que facilita la comunicación entre la aplicación *Flutter* y los servicios de *Firebase*, permitiendo un desarrollo rápido y eficiente.



Figura 55: Instalación y ejecución de CLI de FlutterFire

3. Para finalizar dicha integración del SDK en el entorno, se deben activar *Google Services* incluyendo las dependencias relacionadas en los archivos `build.gradle`. Siguiendo el procedimiento descrito, véase *Figura 56: Inicialización Firebase y agregar complementos*.

### 3 Inicializa Firebase y agrega complementos

Para inicializar Firebase, llama a `Firebase.initializeApp` desde el paquete `firebase_core` con la configuración de tu nuevo archivo `firebase_options.dart`:

```
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

// ...

await Firebase.initializeApp(
  options: DefaultFirebaseOptions.currentPlatform,
);
```



Luego, agrega y comienza a usar los [complementos de Flutter](#) de los productos de Firebase que quieras usar.

Nota: Si usas Analytics o Performance Monitoring, es posible que debas seguir algunos pasos de configuración adicionales.

Figura 56: Inicialización Firebase y agregar complementos

### 6.5.2.3. PROCEDIMIENTO DE INSERCCIÓN Y RECUPERACIÓN DE DATOS

Tras realizar la integración del SDK de *Firebase* en el entorno tecnológico de la aplicación, se realiza la implementación de los métodos desarrollados en *Flutter* con Dart que van a manejar la lógica que interaccione con la base de datos *Cloud Firestore* implementada, utilizando la notación adecuada para que pueda ser correctamente interpretada por *Firebase*, mediante el mapeo de los JSON generados o la creación de valores de tipo objeto. Estos tipos de datos serán convertidos en estructuras de datos compatibles por parte de *Cloud Firestore*, que será capaz de estructurar esta información en documentos, mediante los cuales se organiza el servicio de base de datos.

Para poder realizar estas operaciones con la base de datos, en primer lugar, debemos declarar una variable que nos permita instanciar nuestra base de datos.

```
final FirebaseFirestore _firestore = FirebaseFirestore.instance;
```

Figura 57: Crear instancia de FirebaseFirestore

Una vez declarada la instancia de la base de datos de *Firebase, Firestore*, se puede acceder a ella desde cualquier parte del código dónde se ha instanciado.

A continuación, se muestran métodos que se encuentran de forma global en el proyecto, como ejemplo de las consultas de inserción, modificación, eliminación y obtención de datos.

En el caso de la inserción de datos, podemos mostrar el siguiente método. En este caso, podemos observar cómo se realiza la inserción de los datos en la colección "chats", y a su vez se estructura la información dentro de la subcolección "messages", mediante una función asíncrona, que será la ubicación donde se almacenará la información asociada.

```

Future<void> sendMessage(
    String message, {
        required String docId,
        required String senderId,
        required String receiverId,
    }) async {
    final ts = Timestamp.now();

    final messageData = {
        'message': message,
        'imageUrl': '',
        'senderId': senderId,
        'receiverId': receiverId,
        'timeStamp': ts,
        'isRead': false,
    };

    await firestore
        .collection('chats')
        .doc(docId)
        .collection('messages')
        .add(messageData);

    final chatData = {
        'docId': docId,
        'lastMessage': message,
        'senderId': senderId,
        'timeStamp': ts,
        'lastModification': FieldValue.serverTimestamp(),
        'users': [senderId, receiverId]
    };

    await firestore.collection('chats').doc(docId).set(chatData);
}

```

Figura 58: Método Insertar mensaje de chat en Cloud Firestore

En caso de la obtención de datos, podemos observar el siguiente método como ejemplo dentro del código de la aplicación. Donde la variable definida localmente, recibe toda la información de las publicaciones realizadas que se encuentran almacenadas en la base de datos mediante el método `from()`, el cual se encarga de obtener la información de la colección "products", definida en el servicio que realizada la llamada al método `getProducts()`.

```

Future<void> _loadData() async {
    currentProducts.clear();
    _allProducts.clear();
    isLoading = true;
    update();
    final products = await _service.getProducts();
    if (products == null) {
        trace('Error cargando productos.');
```

Figura 59: Método de obtención de colección de datos de publicaciones

El método que podemos visualizar en la Figura 60: Método de recuperación de datos de la colección "products", `getProducts()`, obtiene la información desde la colección "products", donde se aplica la función adicional `orderBy()`, que permite ordenar los elementos de la colección según la directriz que indicamos por parámetro, en este caso por orden descendente; y por último, se obtiene la información mediante `get()`.

```

//// get products.
Future<List<ResponseProductVo?>> getProducts({bool descending = true}) async {
  final res = await firestore
    .collection('products')
    .orderBy('createdAt', descending: descending)
    .get();
  final list = res.docs.toList();
  return list.map((e) => ResponseProductVo.fromJson(e.data())).toList();
}

```

Figura 60: Método de recuperación de datos de la colección "products"

#### 6.5.2.4. PROCEDIMIENTO DE MODIFICACIÓN Y ELIMINACIÓN

Continuando con el procedimiento de implementación de los métodos de modificación, estos mantienen la misma estructura que la utilizada en los métodos de inserción descritos anteriormente. Como ejemplo, mostramos un par de métodos de modificación de distintos datos, almacenando su información en sus correspondientes documentos.

El primer método, véase Figura 61, ejemplifica un método asíncrono en el que se modifica el estado de publicación, marcada como vendida o no, en nuestro servicio *Cloud Firestore*. En dicho método, se accede a la colección de productos, realizando la búsqueda por el ID, `docId`, asociado a la publicación que enviamos por parámetro en el método `doc()`, y mediante el método `update()` modificamos el estado de la variable *boolean*, marcándola en *true* o *false* según su estado actual.

```

Future<void> markAsSold(int docId) async {
  await firestore
    .collection('products')
    .doc('$docId')
    .update({'isSold': true});
}

```

Figura 61: Método para modificar estado de publicación vendida o no

En el caso del segundo método que se muestra como ejemplo, es un método asíncrono similar al anterior ejemplo, en el que se modifica el estado de conexión del usuario en la plataforma, al realizar el usuario el proceso de autenticación en el sistema se dispara dicho método y modifica el estado de "desconectado" a estado "conectado". Al realizar el cierre de sesión se produce el proceso inverso.

```

void setStatus( String status ) async{
  await _firestore.collection('users').doc( uid ).update({
    'status': status,
  });
}

```

Figura 62: Método modificación de estado de conexión

Por último, se muestra un procedimiento básico de consultas, en este caso para realizar el borrado de una publicación. Se trata de un método asíncrono que mantiene una estructura algorítmica similar a los métodos de modificación.

La única diferencia que encontramos en el método respecto al procedimiento llevado a cabo en los métodos de modificación, es la llamada al método `delete()`, que permite eliminar la información asociada a la colección referencia mediante el método `collection()`, en este caso `'products'`, y realiza la búsqueda mediante el `"docID"` referenciado en el método `doc()`.

```
void deleteAd(BuildContext ctx) async {
  Navigator.of(ctx).pop();

  await FirebaseFirestore.instance
    .collection('products')
    .doc(
      docId.toString(),
    )
    .delete();
}
```

Figura 63: Método de borrado de una publicación

### 6.5.2.5. CONFIGURACIÓN DE BASE DE DATOS

Dentro de la base de datos de Cloud Firestore, se permite al administrador del entorno Firebase incorporar todas las restricciones necesarias sobre el acceso y modificación de los datos. Mediante esta configuración en el lado del servidor de Firebase, ofrecemos un nivel de seguridad alto y robustez en el sistema.

### 6.5.3. ADMINISTRACIÓN DE USUARIOS

Una de las funcionalidades más relevantes de la aplicación es la importancia de realizar una excelente gestión y control de las sesiones y almacenamiento de datos relacionados con los usuarios finales de la plataforma. Se destacan tres aspectos fundamentales que son: el registro de nuevos usuarios en el sistema, la autenticación de nuevos usuarios y la gestión de la sesión del usuario.

El procedimiento comienza con la declaración de una instancia de `FirebaseAuth`, que permite obtener una instancia del servicio `FirebaeAuth`, mediante la cual instanciamos el usuario que está autenticado en ese instante y obtenemos su información relacionada.

```
final uid = FirebaseAuth.instance.currentUser!.uid;
```

Figura 64: Crear instancia de Firebase Authentication

#### 6.5.3.1 REGISTRO DE USUARIOS

Para realizar el registro de usuarios en el sistema a través de sus propias credenciales, en este caso email y contraseña, como podemos observar en la *Figura 65: Método registro de usuarios en el sistema mediante credenciales*, se implementará el método de `FirebaseAuth`, `createUserWithEmailAndPassword()`, implementado dentro del método `submitFunction()`. Este método genera un nuevo usuario en la plataforma a través de las credenciales introducidas por el usuario al sistema.

Tras este proceso se genera una nueva instancia del objeto usuario y se almacena la información en colección "users" dentro de Cloud Firestore con la información indicada por el usuario. Se implementa un flujo mediante una función try-catch que permita validar que la información introducida por el usuario cumpla con las restricciones impuestas o en el caso de que se produzca un error en el sistema, controlando así problemas en la inserción de datos.

```
void submitFunction(  
  String? userName,  
  String? email,  
  String? password,  
  bool isLogin,  
  BuildContext context,  
) async {  
  try {  
  
    setState(() {  
      isLoading = true;  
    });  
  
    UserCredential authResult;  
  
    if (!isLogin) {  
  
      authResult = await FirebaseAuth.instance.createUserWithEmailAndPassword(  
        email: email!,  
        password: password!,  
      );  
  
      FirebaseFirestore.instance  
        .collection('users')  
        .doc(authResult.user!.uid)  
        .set({  
          'email': email,  
          'name': userName,  
          'uid': authResult.user!.uid,  
          'profilePicture': '',  
          'token': '',  
          'status': 'Unavalible',  
        });  
  
    } else {  
  
      authResult = await FirebaseAuth.instance.signInWithEmailAndPassword(  
        email: email!,  
        password: password!,  
      );  
    }  
  } on PlatformException catch (error) {  
    setState(() {  
      isLoading = false;  
    });  
  
    ScaffoldMessenger.of(context).showSnackBar(  
      SnackBar(  
        content: Text('Ha ocurrido un error. Por favor, comprueba tus credenciales'),  
        backgroundColor: Theme.of(context).errorColor,  
      ), // SnackBar  
    );  
  }  
}
```

Figura 65: Método registro de usuarios en el sistema mediante credenciales

### 6.5.3.2. AUTENTICACIÓN DE USUARIOS

Tras realizar el registro del usuario en la aplicación, se podrá realizar la autenticación mediante las credenciales, *email* y contraseña, mediante las que se ha registrado previamente. En el caso de autenticación mediante estas credenciales, se implementará el método de *FirebaseAuth*, *signInWithEmailAndPassword()*, véase Figura 66.

```
Future<UserCredential?> signInWithGoogle() async {  
  
  // Trigger the authentication flow  
  final GoogleSignInAccount? googleUser = await GoogleSignIn().signIn();  
  
  // Obtain the auth details from the request  
  final GoogleSignInAuthentication googleAuth = await googleUser!.authentication;  
  
  // Create a new credential  
  final AuthCredential credential = GoogleAuthProvider.credential(  
    accessToken: googleAuth.accessToken,  
    idToken: googleAuth.idToken,  
  );  
  
  // Once signed in, return the UserCredential  
  authResult = await FirebaseAuth.instance.signInWithCredential(credential);  
  
  FirebaseFirestore.instance  
    .collection('users')  
    .doc( authResult!.user!.uid )  
    .update({  
      'email': authResult!.user!.email,  
      'name': authResult!.user!.displayName,  
      'uid': authResult!.user!.uid,  
      'profilePicture': authResult!.user!.photoURL,  
      'token': '',  
      //'reviews': [],  
      //'averageReview': 0.0  
    });  
  
  return authResult;  
}
```

Figura 66: Método para autenticación de usuarios mediante credenciales

En el caso de que el usuario desee autenticarse mediante su cuenta de usuario asociada a los servicios de Google, se realizará el mismo procedimiento que en caso de la autenticación por credenciales, para ello se declara un objeto de tipo *GoogleSignInAccount* que realiza una llamada al método *GoogleSignIn()*, y dispara el método interno *signIn()*. En la siguiente figura, podemos observar el método al completo.

```

Future<UserCredential?> signInWithGoogle() async {

  // Trigger the authentication flow
  final GoogleSignInAccount? googleUser = await GoogleSignIn().signIn();

  // Obtain the auth details from the request
  final GoogleSignInAuthentication googleAuth = await googleUser!.authentication;

  // Create a new credential
  final AuthCredential credential = GoogleAuthProvider.credential(
    accessToken: googleAuth.accessToken,
    idToken: googleAuth.idToken,
  );

  // Once signed in, return the UserCredential
  authResult = await FirebaseAuth.instance.signInWithCredential(credential);

  FirebaseFirestore.instance
    .collection('users')
    .doc( authResult!.user!.uid )
    .update({
      'email': authResult!.user!.email,
      'name': authResult!.user!.displayName,
      'uid': authResult!.user!.uid,
      'profilePicture': authResult!.user!.photoURL,
      'token': '',
      //'reviews': [],
      //'averageReview': 0.0
    });

  return authResult;
}

```

Figura 67: Método para autenticación de usuarios mediante los servicios de Google

En el caso de que el proceso se realice correctamente, se genera el directorio “Shared Preferences”, donde se almacena la información necesaria del usuario para el uso de la plataforma.

### 6.5.3.3. CIERRE DE SESIÓN EN EL SISTEMA

Finalmente, en el caso de que el usuario autenticado en el sistema desee cerrar su sesión en la aplicación, se implementa el método que se muestra en la Figura 68.

```
void signOut( int option ) async {  
  
    if( option == 1 ) {  
  
        setStatus('Desconectado');  
  
        await GoogleSignIn().signOut();  
        await FirebaseAuth.instance.signOut();  
  
    }  
}
```

Figura 68: Método de cierre de sesión en la plataforma

En este caso, se elimina la información del usuario almacenada en el archivo *Shared Preferences* previamente al cierre de sesión definitivo para que no quede registro alguno del usuario una vez finalizado el proceso.

#### 6.5.4. IMPLEMENTACIÓN TECNOLOGÍAS *FRONT-END*

El contexto actual del mercado para los desarrolladores de *software*, lleva a la necesidad de que, si quieres que tu plataforma sea un éxito, no basta tan solo con desarrollar una plataforma para un sistema operativo concreto, sino que debe funcionar en diversos sistemas operativos. Como es el caso de DaSell, plataforma la cual se puede distribuir en dos sistemas operativos móviles, Android e iOS.

Para obtener este resultado, antes de la aparición de frameworks como *Flutter* o lenguajes de programación, como *Dart*, había que adaptar y compilar el código mediante distintos entornos para cada una de estas plataformas. En ocasiones, hacían falta modificaciones más importantes para tener en cuenta las particularidades de cada sistema operativo móvil, e incluso implementar el diseño y las funcionalidades de la interfaz de usuario la aplicación.

Mediante la implementación de *Flutter*, simplificamos el desarrollo para varias plataformas ya que se usa la misma base de código fuente para todas ellas.

### 6.5.4.1. ¿QUÉ ES FLUTTER?



Figura 69: Implementación - Logo Flutter

*Flutter* es un *framework* de código abierto, creado por Google, publicado por primera vez como proyecto de código abierto a finales de 2018, que se utiliza para desarrollar aplicaciones móviles nativas o híbridas y de alta calidad para múltiples plataformas, web o de escritorio, para distintos sistemas operativos. Se utiliza actualmente en el desarrollo de aplicaciones para Android, iOS, Windows, Mac, Linux, Google Fuchsia y web a partir de un único código base. Este *framework*, escrito en C, C++ y Dart, también desarrollado por Google, proporciona un conjunto completo de herramientas y bibliotecas para facilitar el desarrollo de interfaces de usuario atractivas y fluidas.

Una de las características principales de Flutter es su **enfoque en la creación de interfaces de usuario mediante *widgets***, que son elementos de la interfaz de usuario como botones, campos de texto, imágenes, etc. Flutter proporciona un conjunto variado de *widgets* personalizables y predefinidos, que se pueden combinar y anidar para crear interfaces de usuario complejas y dinámicas.

Gracias a su rendimiento de alta velocidad, *Flutter* permite crear aplicaciones con interfaces de usuario fluidas y *responsive*. También proporciona acceso a APIs nativas y servicios del sistema operativo a través de *plugins*, lo que permite integrar funcionalidades específicas de la plataforma en las aplicaciones.

Los paquetes oficiales y todos aquellos desarrollados por la comunidad, tanto para *Flutter* como para *Dart*, están disponibles en [pub.dev](https://pub.dev), junto con sus dependencias correspondientes. A su vez, la página web oficial con la documentación del *framework* es [flutter.dev](https://flutter.dev).

Es decir, *Flutter* es un *framework* versátil y poderoso para el desarrollo de aplicaciones multiplataforma, que se destaca por su enfoque en la creación de interfaces de usuario atractivas y su rendimiento de alta velocidad.

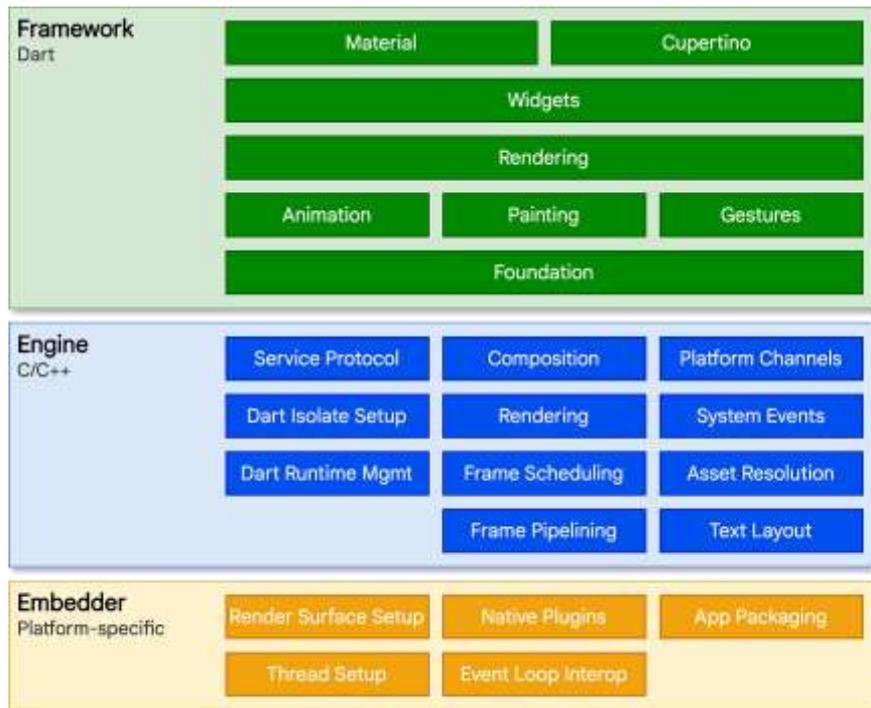


Figura 70: Arquitectura lógica a alto nivel aplicación Flutter

Este *framework*, de estilo reactivo está compuesto por un motor de renderizado 2D, *widgets* listos para poder ser implementados e herramientas de desarrollo. La arquitectura de *Flutter* se encuentra estructurada mediante los siguientes componentes:

- **Plataforma Dart:** para dispositivos móviles se realiza compilación anticipada (AOT, ahead of-time) que permite que Flutter tenga un rendimiento elevado. Además, permite *hot reload* durante el desarrollo, es decir, permite realizar cambios y ver su comportamiento sin necesidad de volver a compilar toda la aplicación.
- **Motor de Flutter:** escrito en C++ y con soporte de renderizado de bajo nivel gracias a las librerías de gráficos de *Skia* de Google. Interactúa con el SDK específico de cada plataforma para implementar gráficos, entrada y salida de ficheros y de red, así como el tiempo de ejecución de *Dart*.
- **Widgets específicos para cada plataforma,** concretamente *widgets* de *Material Design* que implementan el lenguaje de diseño de Google y *widgets Cupertino* que implementan las guías de interfaz humana de iOS de Apple. Además, existe la posibilidad de crear tus propios *widgets* ya que en *Flutter* todo es un *widget*.

## Flutter Approach

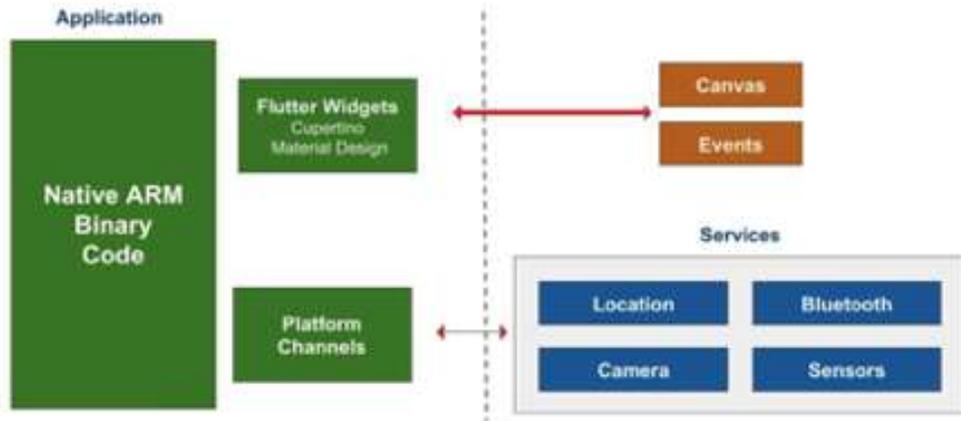


Figura 71: Arquitectura Flutter en un terminal

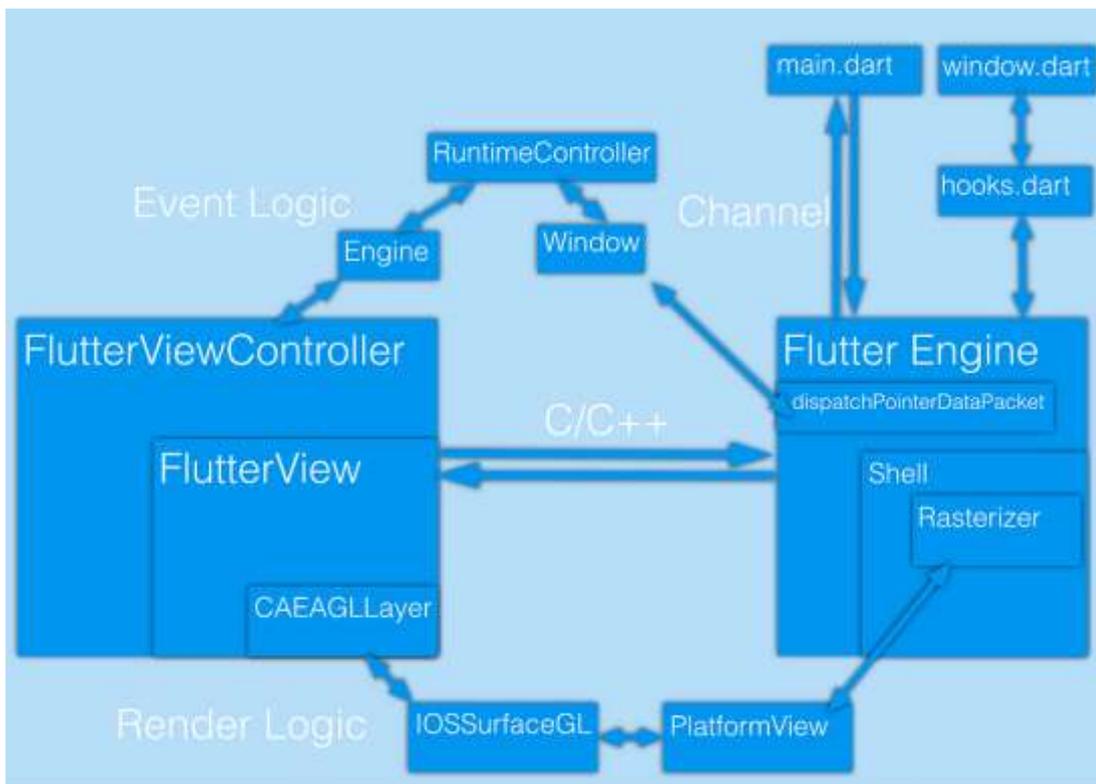


Figura 72: Lógica de ejecución de Flutter en iOS

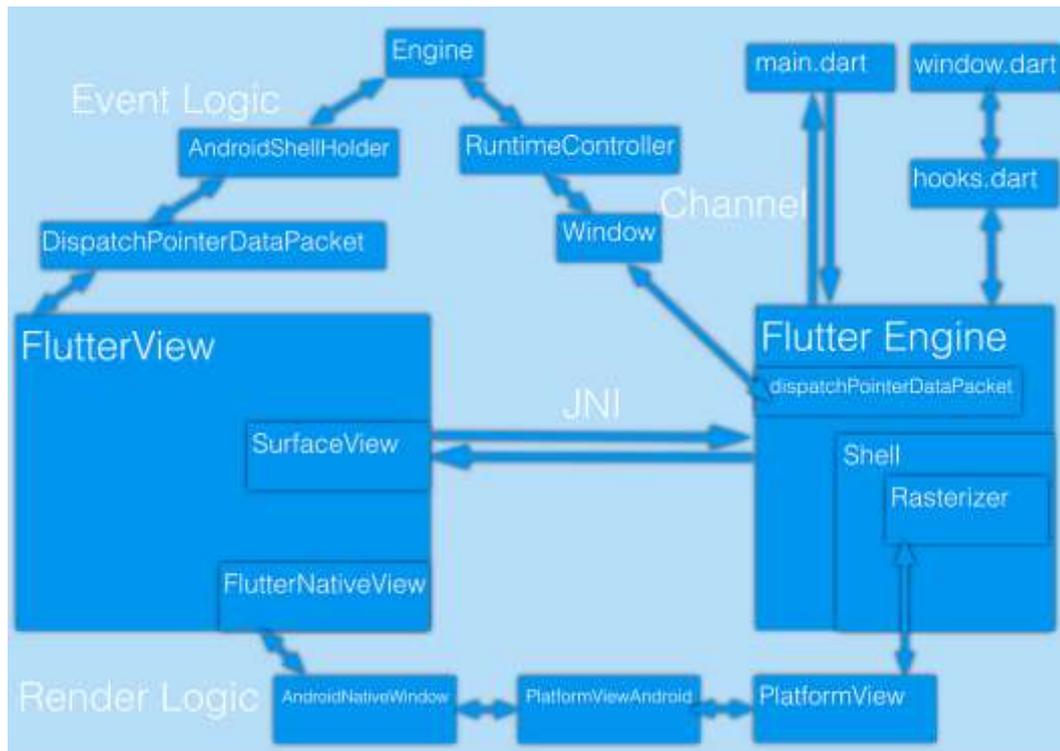


Figura 73: Lógica de ejecución de Flutter en Android

Este framework se puede utilizar actualmente desde los editores *Android Studio*, *IntelliJ*, mediante la instalación de los *plugins* de *Flutter* y de *Dart*; mientras que en el segundo mediante la instalación del complemento para *Flutter*.

### 6.5.4.1.1 CARACTERÍSTICAS DE FLUTTER

Las características principales que se pueden destacar del *framework Flutter* son las siguientes:

1. **Lenguaje de programación Dart:** utiliza el lenguaje de programación *Dart*, desarrollado por Google. *Dart* es un lenguaje moderno y fácil de aprender, con características como tipado estático opcional, recolección de basura y soporte para programación orientada a objetos.
2. **Interfaz de usuario declarativa:** utiliza una arquitectura de interfaz de usuario declarativa, lo que significa que describes cómo debería ser la interfaz de usuario en lugar de cómo se debe crear. Esto permite que *Flutter* realice un "dibujo" eficiente de la interfaz de usuario y proporcione un rendimiento suave y rápido.
3. **Widgets personalizables:** ofrece una amplia gama de *widgets* personalizables y preconstruidos que facilitan la creación de interfaces de usuario atractivas. Los *widgets* en *Flutter* son elementos básicos de la interfaz de usuario, como botones, cajas de texto, imágenes, diseños y más. Puedes combinar y personalizar estos *widgets* para crear una interfaz de usuario única.
4. **Hot Reload:** una de las características más destacadas de *Flutter* es su característica denominada *Hot Reload*. Con *Hot Reload*, se pueden realizar modificaciones en el código de la aplicación y visualizar los resultados en tiempo real, de manera instantánea en la aplicación en

ejecución sin necesidad de realizar un reinicio completo, lo que agiliza el proceso de desarrollo y facilita la depuración de errores.

5. **Plataforma única:** se pueden desarrollar aplicaciones para múltiples plataformas utilizando una sola base de código. Esto significa que puedes crear aplicaciones para iOS y Android sin tener que escribir y mantener dos conjuntos de códigos separados. Flutter también ofrece una apariencia y un rendimiento nativo en ambas plataformas.
6. **Rendimiento nativo:** *Flutter* utiliza el motor de renderizado *Skia* para crear una interfaz de usuario de alto rendimiento y con aspecto nativo. El código de *Flutter* se compila en código nativo para cada plataforma, lo que permite un rendimiento rápido y una apariencia nativa en todos los dispositivos compatibles.
7. **Acceso a APIs nativas:** Si necesitas acceder a características o API específicas de la plataforma, *Flutter* proporciona un mecanismo para interactuar con código nativo a través de canales de plataforma. Esto te permite aprovechar las funcionalidades adicionales de iOS y Android mientras sigues manteniendo una gran parte de tu código compartido.

## ¿Para qué se usa Flutter?

*Flutter* se implementa para desarrollar aplicaciones de Android e iOS, sin necesidad de escribir un código base propio para cada uno de estos sistemas, completamente diferentes entre sí. En este contexto, las aplicaciones móviles se ejecutan como auténticas aplicaciones nativas en los dispositivos. Antes de su publicación, se compilan para la plataforma correspondiente, de manera que no necesitan un *módulo runtime*<sup>11</sup> ni un navegador. Sobre la misma base de código, se pueden crear aplicaciones web para navegadores y programas nativos para Windows, Linux y macOS.

Google utiliza *Flutter* en diversos módulos de *Google Assistant* y en la interfaz de usuario de *Google Home Hub*. Asimismo, varios proveedores de comercio electrónico, como *eBay*, *Groupon* y el *Alibaba Group*, emplean *Flutter* para lograr una apariencia y estilo uniformes en sus aplicaciones web y móviles.

### 6.5.4.1.2 COMPONENTES DE FLUTTER

Las arquitecturas de las aplicaciones *Flutter* están formadas por componentes, los cuales consisten principalmente en, y podemos distribuir de la siguiente manera:

- **Widgets:** Los *widgets* son el componente principal de cualquier aplicación de *Flutter*. Por medio del cual, actúa como una interfaz de usuario para que el usuario interactúe con la aplicación. Cualquier aplicación desarrollada en *Flutter* es, en sí misma, un *widget* que se compone de una combinación de *widgets*. En una aplicación estándar, la raíz define la estructura de la aplicación seguida de un *widget* `MaterialApp` que básicamente mantiene sus componentes internos en su lugar.

---

<sup>11</sup> **Módulo runtime:** se refiere a un componente o conjunto de bibliotecas que proporcionan las funcionalidades esenciales necesarias para ejecutar un programa o una aplicación. Este módulo puede incluir funciones y servicios relacionados con la gestión de memoria, el manejo de excepciones, la administración de hilos o procesos, y otras tareas fundamentales para el correcto funcionamiento del programa en tiempo de ejecución.

Un *widget* puede ser algo tan simple como un botón o un texto, o puede ser algo más complejo, como una lista desplegable o una tarjeta de perfil. Los *widgets* pueden contener otros *widgets* y formar una jerarquía, lo que permite construir interfaces de usuario complejas combinando *widgets* más simples, lo que se denomina como árbol de *widgets*, véase apartado **6.5.4.3. Árbol de Widgets**.

En *Flutter*, hay dos tipos principales de *widgets*: los *widgets* sin estado, ***StatelessWidget*** y los *widgets* con estado, ***StatefulWidget***.

- ❖ ***StatelessWidget***: Es un *widget* que no tiene estado interno y su apariencia se determina únicamente por los parámetros proporcionados en su construcción. No puede cambiar su apariencia una vez creado. Es un *widget* inmutable cuyo contenido y aspecto no cambian una vez que se construye. Un *StatelessWidget* no tiene un objeto *State* asociado y, por lo tanto, no puede realizar cambios en su estado interno.

Un *StatelessWidget* no puede cambiar su contenido o aspecto durante la ejecución de la aplicación. Esto significa que cada vez que se necesita actualizar la interfaz de usuario, se debe crear una nueva instancia del *widget*.

En *Flutter*, *StatelessWidget* es una clase base que se utiliza para construir *widgets* que no cambian su estado interno. Un *StatelessWidget* es inmutable, lo que significa que una vez que se crea y se muestra en la interfaz de usuario, no puede cambiar su apariencia ni su comportamiento.

Un *StatelessWidget*, se construye mediante la implementación del método *build()*, como podemos observar en la Figura 74, que devuelve un árbol de *widgets* que representa la interfaz de usuario que se va a mostrar. Este árbol de *widgets* se construye a partir de otros *widgets* existentes, y puede contener cualquier combinación de *widgets* predefinidos o personalizados.

La principal característica de un *StatelessWidget* es que no tiene un estado interno mutable. Esto significa que no puede almacenar ni modificar datos de forma directa. En su lugar, un *StatelessWidget* se basa en los datos que se le proporcionan a través de sus propiedades, llamadas "props" o "parámetros", al momento de su creación.

Cuando los datos proporcionados a un *StatelessWidget* cambian, se debe crear una nueva instancia del mismo para reflejar esos cambios. *Flutter* se encarga automáticamente de comparar las instancias antiguas y nuevas, y realizar la actualización eficientemente solo en los *widgets* que realmente han cambiado.

Concluyendo, un *StatelessWidget* en *Flutter* es un *widget* inmutable que se basa en datos proporcionados externamente y se utiliza para construir interfaces de usuario que no cambian su estado interno. Es útil cuando se necesita crear *widgets* que solo dependan de los datos de entrada y no requieran realizar cambios en su comportamiento o apariencia a lo largo del tiempo. A continuación, se muestra en la Figura 74, un ejemplo de un *StatelessWidget* desarrollado dentro de la plataforma DaSell.

```

class ManualMarker extends StatelessWidget {
  const ManualMarker({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return BlocBuilder<SearchBloc, SearchState>(
      builder: (context, state) {
        return state.displayManualMarker
          ? const _ManualMarkerBody()
          : kEmptyWidget;
      },
    ); // BlocBuilder
  }
}

```

Figura 74: Ejemplo código StatelessWidget

- ❖ **StatefulWidget:** Es un *widget* que puede tener un estado interno mutable, lo que significa que puede cambiar su apariencia o comportamiento en respuesta a interacciones del usuario, eventos o cambios en los datos. Los StatelessWidget son útiles cuando se necesita mantener información y actualizar la interfaz de usuario en función de los cambios en esa información.

Un StatefulWidget consta de dos clases principales: la clase *StatefulWidget* y la clase *State*. La clase *StatefulWidget*, es una representación inmutable del *widget* que contiene la lógica y los datos necesarios para construir el *widget*. La clase *State*, es una representación mutable del estado del *widget*, que puede cambiar a medida que la aplicación se ejecuta.

Cuando se crea un StatefulWidget, Flutter crea una instancia de la clase StatefulWidget y una instancia de la clase State asociada. El objeto State se utiliza para almacenar y actualizar el estado del widget. Cuando el estado cambia, Flutter vuelve a llamar al método build() del widget para reconstruir la interfaz de usuario y reflejar los cambios.

El uso de un StatefulWidget es útil cuando se necesita mantener y actualizar un estado en respuesta a interacciones del usuario, cambios en los datos o eventos en la aplicación. Algunos ejemplos comunes de StatefulWidget son los formularios, los contadores y las listas que pueden cambiar dinámicamente.

Para facilitar su comprensión, se muestra en la Figura 75: Ejemplo código StatefulWidget, un ejemplo básico de cómo se puede definir un StatefulWidget en Flutter:

```

class MyStatefulWidget extends StatefulWidget {
  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Stateful Widget Example'),
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text('Counter:', style: TextStyle(fontSize: 24)),
            Text('$ _counter', style: TextStyle(fontSize: 48)),
          ], // <Widget>[]
        ), // Column
      ), // Center
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        child: Icon(Icons.add),
      ), // FloatingActionButton
    ); // Scaffold
  }
}

```

Figura 75: Ejemplo básico StatefulWidget en Flutter

En este ejemplo, se define un StatefulWidget llamado **MyStatefulWidget**. La clase **\_MyStatefulWidgetState** es la clase State asociada. El estado del widget se almacena en la variable **\_counter**, que se actualiza en respuesta al botón de incremento. La llamada al método **setState()** notifica a Flutter que el estado ha cambiado y debe volver a construir la interfaz de usuario.

Los widgets son los bloques de construcción fundamentales para crear aplicaciones y se utilizan para definir tanto la estructura como el aspecto visual de la interfaz de usuario. Al combinar y anidar widgets, puedes crear interfaces de usuario complejas y dinámicas en Flutter.

- **Capas:** se pueden utilizar diferentes capas para organizar y estructurar la aplicación. A continuación, se realiza una breve descripción de las capas más comunes en Flutter:

- ❖ **Capa de interfaz de usuario (UI Layer):** La capa de interfaz de usuario en *Flutter* está compuesta por *widgets*, que son los bloques de construcción fundamentales de la interfaz de usuario. *Flutter* ofrece una amplia gama de *widgets* predefinidos que se pueden utilizar para definir la interfaz de usuario de tu aplicación. Estos *widgets* se agrupan en *widgets* básicos, como *Container*, *Text*, *Image*, *Button*, así como *widgets* de diseño más complejos, como *ListView*, *GridView* y *Stack*.
- ❖ **Capa de lógica de negocio (Business Logic Layer):** La capa de lógica de negocio es responsable de manejar la funcionalidad principal de la aplicación. Aquí es donde se puede implementar la lógica y las reglas específicas de la aplicación. Se crean clases y métodos para manejar tareas como la recuperación de datos, el procesamiento de datos, la validación de formularios y la comunicación con servicios externos.
- ❖ **Capa de acceso a datos (Data Access Layer):** La capa de acceso a datos se ocupa de interactuar con diferentes fuentes de datos, como bases de datos locales, servicios web o almacenamiento en la nube. Aquí se pueden definir clases y métodos para realizar operaciones de lectura y escritura de datos, consultas y transformaciones de datos.
- ❖ **Capa de servicios (Services Layer):** La capa de servicios puede considerarse como una extensión de la capa de acceso a datos. Aquí se puede encapsular la lógica de acceso a servicios externos, como servicios web, APIs o servicios de notificación. Se pueden utilizar bibliotecas de manejo de estado o paquetes específicos para interactuar con estos servicios.
- ❖ **Capa de presentación de datos (Presentation Layer):** La capa de presentación de datos se utiliza para representar y mostrar los datos de la aplicación de manera visual. Aquí es donde se puede dar formato a los datos y presentarlos de forma adecuada al usuario. Se pueden utilizar *widgets* para mostrar listas, gráficos, tablas y otros elementos visuales para representar los datos de manera comprensible y atractiva.

Es importante tener en cuenta que estas capas no son estrictamente separadas y pueden tener interacciones entre sí. La arquitectura y organización específica de las capas dependerá de la estructura y complejidad de tu aplicación. *Flutter* no impone una arquitectura específica, pero existen patrones comunes como BLoC (*Business Logic Component*), *Provider*, *MobX*, *Redux*, entre otros, que puedes utilizar para estructurar un código de manera ordenada y escalable.

- **Gestos:** se utilizan para detectar y responder a las interacciones del usuario, como toques, deslizamientos y gestos de arrastre. *Flutter* proporciona una amplia gama de *widgets* y clases que permiten implementar gestos en la aplicación. Algunos de estos ejemplos, implementado en la aplicación de DaSell son:
  - ❖ **GestureDetector:** Este *widget* se utiliza para detectar varios gestos en un área determinada. Puedes envolver cualquier otro *widget* con *GestureDetector* y utilizar sus propiedades de devolución de llamada para detectar eventos de gestos como *onTap*, *onDoubleTap*, *onLongPress*, etc.

```

GestureDetector(
  onTap: () {
    // Lógica a ejecutar cuando se toca el widget
  },
  onDoubleTap: () {
    // Lógica a ejecutar cuando se toca dos veces el widget
  },
  onLongPress: () {
    // Lógica a ejecutar cuando se mantiene presionado el widget
  },
  child: Container(
    // Widget envuelto por GestureDetector
  ), // Container
); // GestureDetector

```

Figura 76: Ejemplo métodos GestureDetector

- ❖ **InkWell:** Este *widget* es similar a GestureDetector, pero proporciona una respuesta visual cuando se realiza un gesto táctil. InkWell es útil cuando deseas mostrar una animación de salpicadura al seleccionar un *widget*.

```

InkWell(
  onTap: () {
    // Lógica a ejecutar cuando se toca el widget
  },
  child: Container(
    // Widget envuelto por InkWell
  ), // Container
); // InkWell

```

Figura 77: Ejemplo método InkWell

- ❖ **DragGestureRecognizer:** Esta clase se utiliza para detectar gestos de arrastre en un *widget*. Se puede envolver un *widget* con GestureDetector y utilizar un DragGestureRecognizer para detectar gestos de arrastre en ese *widget*.

```

GestureDetector(
  onVerticalDragDown: (details) {
    // Lógica cuando se inicia el gesto de arrastre vertical
  },
  onVerticalDragUpdate: (details) {
    // Lógica cuando se actualiza el gesto de arrastre vertical
  },
  onVerticalDragEnd: (details) {
    // Lógica cuando se finaliza el gesto de arrastre vertical
  },
  child: Container(
    // Widget envuelto por GestureDetector
  ), // Container
); // GestureDetector

```

Figura 78: Ejemplo métodos DragGestureRecognizer

- **Gestión de estado:** se refiere al manejo y control de los datos y la información que se utiliza en una aplicación *Flutter*. En *Flutter*, hay diferentes enfoques para gestionar el estado, dependiendo de la complejidad de la aplicación y de las necesidades específicas.
  1. **Estado local:** Para componentes simples o pequeñas secciones de la interfaz de usuario, se utiliza el estado local. El estado local se gestiona dentro del propio *widget* y se actualiza utilizando el método `setState()`. Este enfoque es útil cuando el estado solo afecta a un *widget* y no necesita ser compartido con otros *widgets*.
  2. **Estado compartido:** Para compartir y administrar el estado entre múltiples *widgets*, se pueden utilizar diferentes opciones, en este caso se ha implementado, *InheritedWidget* o *Provider*. Estas son bibliotecas y patrones de gestión de estado de terceros que ayudan a simplificar la comunicación entre *widgets* y a mantener el estado de la aplicación de manera eficiente.
    - ❖ **InheritedWidget:** Es una clase base que permite compartir datos a través de la jerarquía de *widgets*. Se puede crear una clase que extienda *InheritedWidget* y definir el estado que se desea compartir. Los *widgets* descendientes pueden acceder a ese estado utilizando `InheritedWidget.of(context)`.
    - ❖ **Provider:** Es una biblioteca popular que implementa patrones de gestión de estado como *ChangeNotifier* y *Provider*. *ChangeNotifier* es una clase base que notifica a los *widgets* cuando el estado cambia. *Provider* facilita el acceso a ese estado compartido en toda la aplicación.
  3. **Estado global:** Para el estado que necesita ser accesible desde cualquier parte de la aplicación, se puede utilizar un estado global. En la plataforma, se ha implementado una de las opciones más comunes para gestionar el estado global como es el patrón BLoC (*Business Logic Component*).
    - ❖ **BLoC:** Es un patrón de gestión de estado que utiliza flujos (*streams*) para enviar y recibir eventos y datos. Puedes utilizar las clases de los paquetes `flutter_bloc` o `rxdart` para implementar el patrón BLoC.

### 6.5.4.1.3 WIDGETS IMPLEMENTADOS

En este apartado se describen los componentes *widgets* genéricos más comunes que han sido implementados en el desarrollo. En *Flutter*, una aplicación se construye utilizando varios componentes y *widgets* que definen la interfaz de usuario y la lógica de la aplicación. Aquí hay algunos componentes clave en *Flutter*:

1. **MaterialApp:** *widget* que configura una aplicación de *Flutter* con el estilo y comportamiento típicos de una aplicación móvil. Proporciona características como el manejo de rutas, el tema de la aplicación y mucho más.
2. **Scaffold:** *widget* que proporciona una estructura básica para la interfaz de usuario de la aplicación. Proporciona una barra de aplicación, un área de contenido principal y áreas opcionales como el cajón de navegación (*drawer*) y la parte inferior de la aplicación (*bottom navigation bar*).

3. **Container:** *widget* que permite la personalización del diseño y la apariencia de otros *widgets*. Puede definir el tamaño, el fondo, los bordes y el espaciado de un *widget* contenido dentro de él.
4. **Text:** muestra un texto en la interfaz de usuario. Puede personalizar el estilo del texto, como el tamaño de fuente, el color y la alineación.
5. **Image:** muestra una imagen en la interfaz de usuario. Puede cargar imágenes desde la red, desde el sistema de archivos o desde recursos locales.
6. **ListView:** *widget* que muestra una lista de elementos desplazables en la pantalla. Puede contener cualquier otro *widget* como elementos de lista.
7. **GestureDetector:** *widget* que permite detectar gestos de entrada del usuario, como toques, deslizamientos y arrastrar y soltar.
8. **AppBar:** *widget* que muestra una barra de aplicación en la parte superior de la pantalla. Puede contener un título, acciones y otros *widgets*.
9. **RaisedButton y FlatButton:** Son *widgets* que representan botones en la interfaz de usuario. `RaisedButton` muestra un botón en relieve, mientras que `FlatButton` muestra un botón plano. Ambos *widgets*, pueden tener un manejador de eventos para responder a las interacciones del usuario.

Estos son solo algunos de los muchos componentes disponibles en *Flutter*. La combinación y personalización de estos componentes permiten construir interfaces de usuario potentes y dinámicas en aplicaciones *Flutter*.

#### 6.5.4.1.4. ÁRBOL DE WIDGETS

La estrategia que sigue *Flutter* cumple un principio básico, **todo es un widget**. Esta estrategia sigue las bases de la programación orientada a objetos hasta la interfaz de usuario: la interfaz del programa consta de diferentes *widgets* que pueden estar anidados entre ellos. Cada botón y texto mostrado es un *widget*.

Estos *widgets* cuentan con diferentes propiedades o atributos que se pueden modificar, los cuales pueden interactuar entre sí y reaccionar a cambios de estado externos mediante sus funciones integradas. Todos los elementos importantes de la interfaz de usuario incluyen *widgets* que se corresponden con los diseños de Android y iOS o las aplicaciones web convencionales. Si se desea, estos *widgets* se pueden ampliar con funciones adicionales o se pueden crear *widgets* propios que se pueden combinar fácilmente con los ya existentes.

En comparación con otros SDK, los *widgets* ofrecen una mayor flexibilidad, pero cuentan con la desventaja de que forman parte del código fuente del programa, por lo que el código queda muy anidado y puede resultar confuso.

Un aspecto destacado de *Flutter* es la organización de su estructura mediante un árbol de *widgets*. Un **árbol de widgets** es una representación jerárquica de *widgets* que se utiliza para construir la interfaz de usuario de una aplicación. Los *widgets* son componentes de la interfaz de usuario en

Flutter, que pueden ser desde elementos básicos, como texto y botones, hasta elementos más complejos, como listas y formularios.

El árbol de *widjets* sigue el principio de composición, donde los *widjets* se anidan unos dentro de otros para construir la interfaz de usuario. Cada *widjet* tiene un propósito y una funcionalidad específica y se puede personalizar mediante la configuración de sus propiedades.

Para facilitar su comprensión, se muestra un ejemplo básico de un árbol de *widjets* en Flutter:

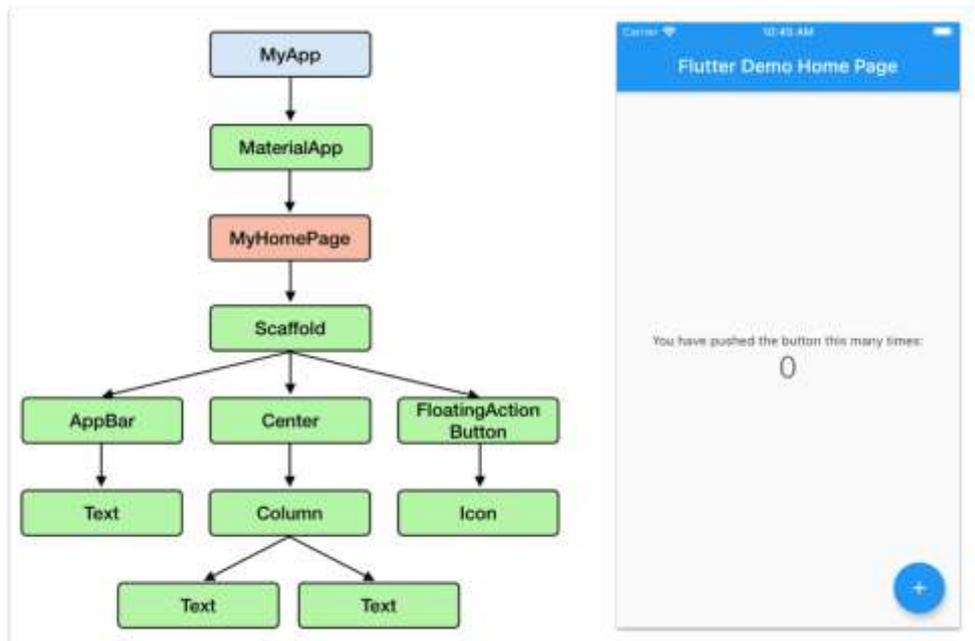


Figura 79: Árbol de Widgets genérico

En la *Figura 79: Árbol de Widgets genérico*, se representa un árbol de widgets que comienza con el *widjet* `MaterialApp`, que es el *widjet* principal que configura la aplicación de Flutter. Dentro de `MaterialApp` se instancia la clase `MyHomePage` que contiene en el estado global de todos los *widjets* que se encuentran bajo `MaterialApp`. Dentro de esta clase encontramos un *widjet* `Scaffold` que proporciona una estructura básica de la pantalla de la aplicación, incluyendo una barra de aplicaciones (`AppBar`) y un cuerpo (`body`). Dentro del cuerpo, hay un *widjet* `Center` que coloca su hijo, en este caso un *widjet* `Body`, en el centro de la pantalla. Por último, un *widjet* `FloatingActionButton` que permite establecer un botón flotante sobre la pantalla del dispositivo.

A continuación, se muestra el código de la representación de árbol de *widjets* descrita anteriormente.

```

1 import 'package:flutter/material.dart';
2
3 // Entrada de la aplicación
4 void main() => runApp(MyApp());
5
6 class MyApp extends StatelessWidget {
7   @override
8   Widget build(BuildContext context) {
9     return MaterialApp(
10      title: 'Flutter Demo',
11      theme: ThemeData(
12        // tema azul
13        primarySwatch: Colors.blue,
14      ), // ThemeData
15      // Enrutamiento de la página de inicio de la aplicación
16      home: MyHomePage(title: 'Flutter Demo Home Page'),
17    ); // MaterialApp
18  }
19 }
20
21 class MyHomePage extends StatefulWidget {
22   MyHomePage({Key? key, required this.title}) : super(key: key);
23
24   final String title;
25   @override
26   _MyHomePageState createState() => _MyHomePageState();
27 }
28
29 class _MyHomePageState extends State<MyHomePage> {
30   int _counter = 0;
31
32   void _incrementCounter() {
33     setState(() {
34       _counter++;
35     });
36   }
37
38   @override
39   Widget build(BuildContext context) {
40     return Scaffold(
41       appBar: AppBar(
42         title: Text(widget.title),
43       ), // AppBar
44       body: Center(
45
46         child: Column(
47           mainAxisAlignment: MainAxisAlignment.center,
48           children: <Widget>[
49             Text(
50               'You have pushed the button this many times:',
51             ), // Text
52             Text(
53               '$_counter',
54               style: Theme.of(context).textTheme.display1,
55             ), // Text
56           ], // <Widget>[]
57         ), // Column
58       ), // Center
59       floatingActionButton: FloatingActionButton(
60         onPressed: _incrementCounter,
61         tooltip: 'Increment',
62         child: Icon(Icons.add),
63       ), // FloatingActionButton
64     ); // Scaffold
65   }
66 }

```

Figura 80: Ejemplo código Árbol de Widgets

Este es solo un ejemplo básico, pero en *Flutter* puedes construir árboles de *widgets* mucho más complejos y anidar *widgets* dentro de otros *widgets* para crear interfaces de usuario personalizadas y dinámicas.

Por último, se muestra, una representación gráfica a alto nivel del árbol de *widgets* de forma global sobre cómo está estructurado en la aplicación *DaSell*,

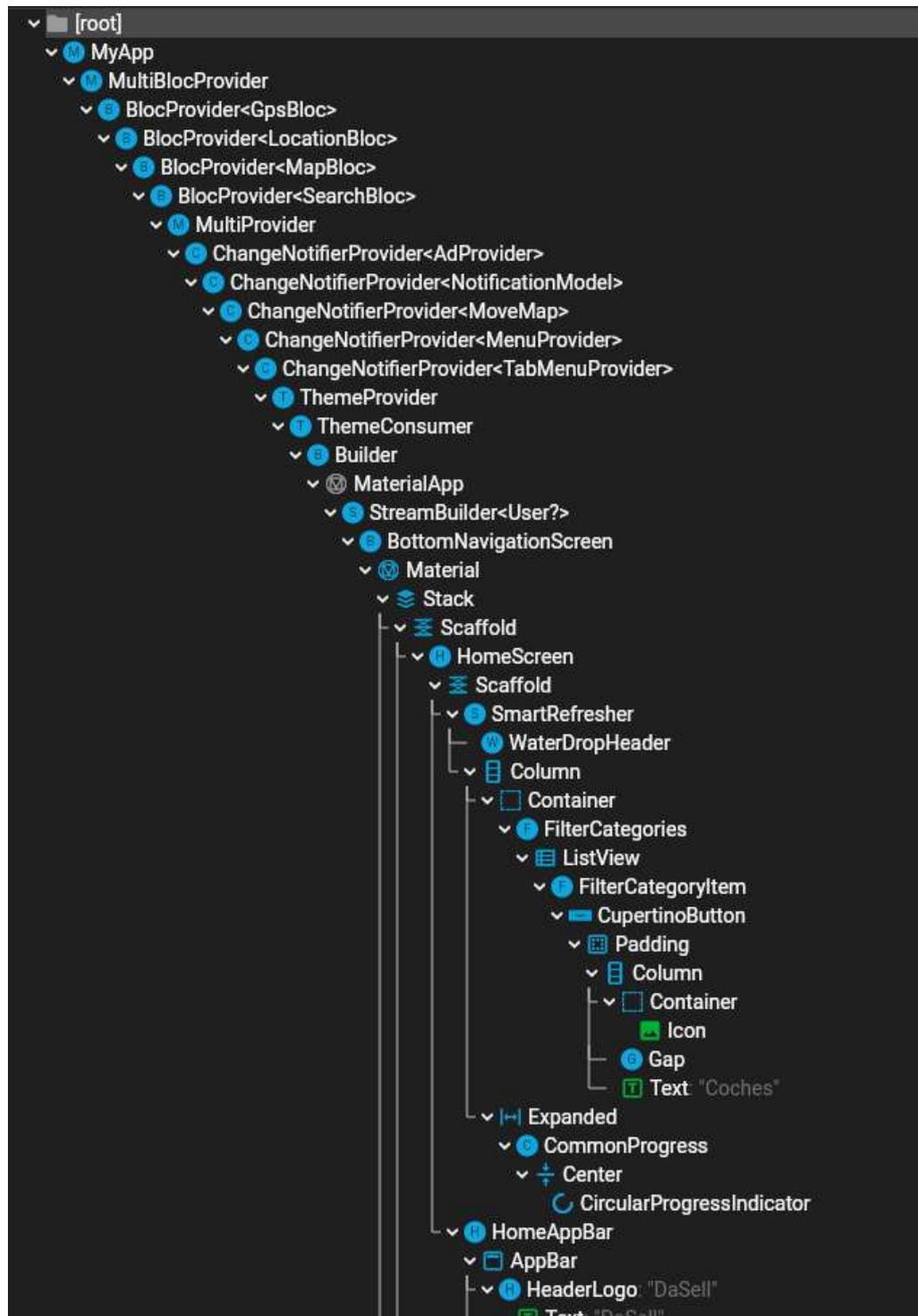


Figura 81: Árbol de Widgets aplicación *DaSell*

## 6.5.4.2. LENGUAJE DART



Figura 82: Implementación - Logo Dart

Dart es un lenguaje de programación de código abierto desarrollado por Google. Fue presentado inicialmente en 2011 y ha ganado popularidad principalmente como el lenguaje de programación principal para el desarrollo de aplicaciones móviles utilizando el framework *Flutter*. *Dart* es un estándar ECMA<sup>12</sup>, la organización europea para la estandarización de sistemas informáticos, de comunicación y productos electrónicos. Se ha postulado como una de las alternativas más interesantes a la utilización de JavaScript en el desarrollo móvil que se realiza en la actualidad.

En el caso de la programación web, dado que los navegadores no pueden trabajar mediante este lenguaje de forma genérica, en contraposición a JavaScript que, si puede ejecutarse en todos los navegadores web existentes, existe el compilador *Dart2js*, que permite su implementación.

A continuación, se realiza una descripción de las características clave del lenguaje de programación Dart:

1. **Tipado estático y dinámico:** *Dart* es un lenguaje de tipado estático, lo que significa que las variables deben ser declaradas con un tipo específico y no pueden cambiar de tipo una vez que se les asigna un valor. Sin embargo, *Dart* también es capaz de realizar inferencia de tipos, lo que permite omitir las anotaciones de tipo en algunas ocasiones y dejar que el compilador deduzca el tipo.
2. **Orientado a objetos:** *Dart* es un lenguaje orientado a objetos en el que todo es un objeto, incluyendo los números, funciones y nulos. Admite características tradicionales de la programación orientada a objetos como la herencia, interfaces, clases abstractas y *mixins*.
3. **Sintaxis familiar:** La sintaxis de *Dart* se asemeja a la de otros lenguajes de programación populares, como C, C++ y Java, lo que facilita el aprendizaje y la adopción por parte de los desarrolladores que ya están familiarizados con estos lenguajes.
4. **Soporte para asincronía:** *Dart* proporciona un modelo de programación asíncrona nativo para facilitar el desarrollo de aplicaciones que necesitan realizar operaciones no bloqueantes, como llamadas a API, lectura y escritura de archivos, entre otros. Utiliza palabras clave como `async` y `await` para manejar la asincronía de manera más sencilla.

---

<sup>12</sup> **Estándar ECMA (European Computer Manufacturers Association):** organización internacional de estándares que se dedica a la creación y promoción de estándares para tecnologías de la información y la comunicación. Establece normas técnicas y especificaciones para diversos ámbitos, incluyendo el lenguaje de programación, los sistemas operativos, las interfaces de usuario y otros componentes relacionados con la informática. ( Fuente: [Estándar ECMA](#) ).

5. **Colecciones y estructuras de datos:** *Dart* ofrece una variedad de colecciones y estructuras de datos integradas, como listas, conjuntos y mapas. También proporciona métodos y operadores para trabajar con estas estructuras de datos de manera eficiente.
6. **Compatibilidad multiplataforma:** Una de las principales fortalezas de *Dart* es su capacidad para compilar a código nativo, lo que permite ejecutar aplicaciones *Dart* en múltiples plataformas. En particular, *Dart* se utiliza ampliamente en el desarrollo de aplicaciones móviles con *Flutter*, pero también puede ser utilizado para crear aplicaciones de línea de comandos o aplicaciones web.
7. **Herramientas y ecosistema:** *Dart* cuenta con una serie de herramientas y bibliotecas útiles para el desarrollo de software. El compilador de *Dart* puede generar código nativo rápido y eficiente, y la biblioteca estándar proporciona una amplia gama de funcionalidades. Además, el ecosistema de *Dart* ha crecido rápidamente con la comunidad de desarrolladores, lo que ha dado lugar a una gran cantidad de paquetes y extensiones de terceros.

Para finalizar, *Dart* es un lenguaje de programación versátil que se ha vuelto muy popular para el desarrollo de aplicaciones móviles y ofrece una sintaxis familiar, soporte para asincronía, tipado estático, orientación a objetos y un ecosistema en crecimiento.

### 6.5.4.3 GESTOR DE ESTADOS: PATRÓN PROVIDER

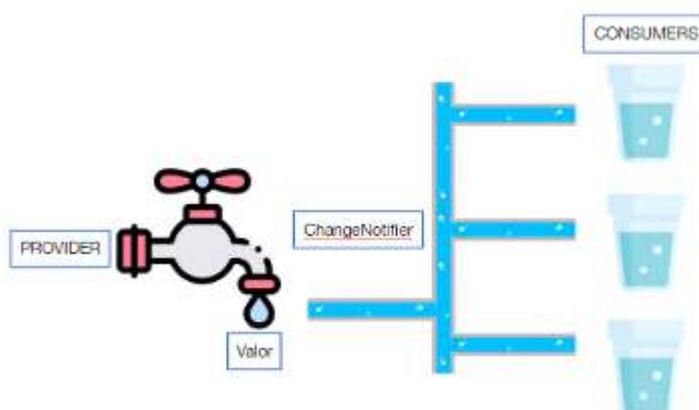


Figura 83: Gestor de estados: Patrón Provider

En el contexto de *Flutter*, el patrón `Provider` se refiere a un patrón de gestión de estado que se utiliza para compartir datos entre diferentes `widgets` de una aplicación *Flutter*. Es un patrón de diseño que se basa en el principio de inversión de dependencia y se implementa utilizando la biblioteca `Flutter Provider`.

En esencia, *Provider* proporciona una forma de mantener y compartir el estado de la aplicación de manera eficiente, entre varios `widgets`, sin necesidad de pasar explícitamente los datos a través de los árboles de `widgets`.

El patrón `Provider` se basa en los fundamentos:

1. **Provider:** Es una clase que contiene el estado de la aplicación y proporciona métodos para acceder a dicho estado. El *Provider* se coloca en la parte superior del árbol de *widgets* y se puede acceder desde cualquier *widget* descendiente.
2. **Consumer:** Es un *widget* que se utiliza para consumir el estado proporcionado por un *Provider*. Un *Consumer* se suscribe al *Provider* y se reconstruye automáticamente cada vez que cambia el estado.
3. **ChangeNotifier:** Es una clase que extiende la clase base **ChangeNotifier** de *Flutter*. Representa el estado mutable de la aplicación y notifica a los consumidores cuando el estado cambia. Los *Providers* a menudo utilizan clases que implementan *ChangeNotifier* para administrar su estado.
4. **Selector:** Es un *widget* que se utiliza para suscribirse a una parte específica del estado proporcionado por un *Provider*. Esto permite evitar reconstrucciones innecesarias de *widgets* que no dependen de la parte del estado que ha cambiado.

El patrón *Provider* se utiliza ampliamente en *Flutter* debido a su simplicidad y eficiencia para administrar el estado de la aplicación. Proporciona una forma ordenada de manejar la propagación de cambios de estado y evita el acoplamiento directo entre los *widgets*.

#### 6.5.4.3.1. IMPLEMENTACIÓN DEL PATRÓN PROVIDER

En este apartado, se describe el proceso que se ha seguido para implementar el patrón *Provider* en la aplicación *DaSell*, mediante un ejemplo genérico que facilite su comprensión. Los pasos que se han llevado a cabo son:

1. Se agrega la dependencia de `provider` en el archivo `pubspec.yaml`.

```
dependencies:
flutter:
  sdk: flutter
provider: ^6.0.5
```

Figura 84: Implementación *Provider* – paquete *provider*

2. Se importa el paquete `provider` en el archivo *Dart*, donde se realizará la implementación:

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
```

Figura 85. Implementación *Provider* - paquete *Provider*

3. Se crea una clase para almacenar el estado global de la aplicación. Esta clase debe extender de *ChangeNotifier* y contener los datos que se desea compartir:

```

class MiEstado extends ChangeNotifier {
  String _miDato = 'Hola';

  String get miDato => _miDato;

  void actualizarDato(String nuevoDato) {
    _miDato = nuevoDato;
    notifyListeners();
  }
}

```

Figura 86: Implementación Provider - ChangeNotifier

4. En la clase principal, generalmente `main.dart`, se envuelve el `MaterialApp` con `ChangeNotifierProvider` y se proporciona una instancia de la clase `MiEstado`:

```

void main() {
  runApp(
    ChangeNotifierProvider(
      create: (context) => MiEstado(),
      child: MyApp(),
    ), // ChangeNotifierProvider
  );
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Mi Aplicación',
      home: MiPagina(),
    ); // MaterialApp
  }
}

```

Figura 87: Implementación Provider - ChangeNotifierProvider

5. En los `widgets`, se puede acceder al estado global utilizando `Provider.of<T>(context)` o utilizando la extensión de contexto `context.read<T>()` y `context.watch<T>()`. A continuación, se muestra un ejemplo como representativo al resto de `widgets` implementados. En este ejemplo, se muestra el valor de `_miDato` en un `Text` y se actualiza al presionar el botón.

```

class MiPagina extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final miEstado = context.watch<MiEstado>();

    return Scaffold(
      appBar: AppBar(
        title: Text('Ejemplo Provider'),
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(miEstado.miDato),
            ElevatedButton(
              onPressed: () {
                miEstado.actualizarDato('¡Hola actualizado!');
              },
              child: Text('Actualizar'),
            ), // ElevatedButton
          ],
        ), // Column
      ), // Center
    ); // Scaffold
  }
}

```

Figura 88: Implementación Provider - Gestión estado mediante Provider

#### 6.5.4.4 GESTOR DE ESTADOS: PATRÓN BLOC

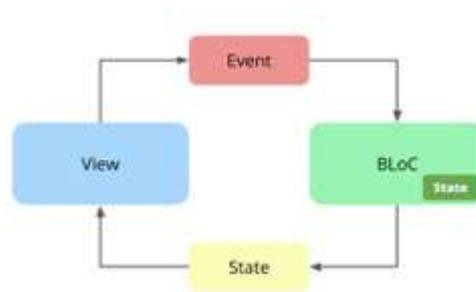


Figura 89: Patrón BLoC Flutter

Un patrón BLoC (**Business Logic Component**) en Flutter es una arquitectura de diseño utilizada para separar la lógica empresarial de la interfaz de usuario en una aplicación. El patrón BLoC se basa en la programación reactiva y se utiliza para mantener un estado predecible y gestionar los flujos de datos en una aplicación Flutter.

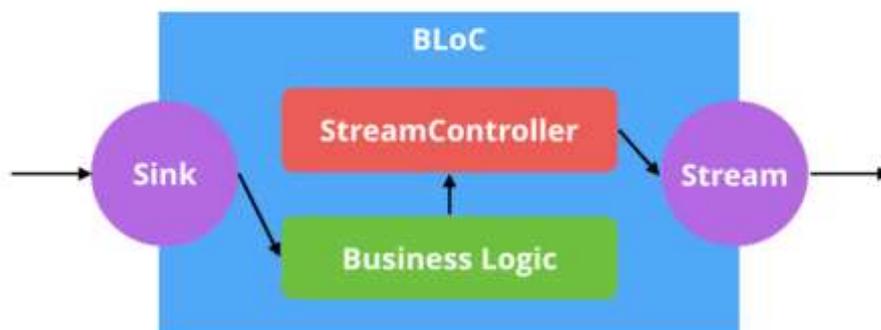


Figura 90: Patrón BLoC - Flujo de información

El patrón BLoC consta de los siguientes componentes principales:

- **Sink:** se refiere a una entrada o receptor de eventos en un flujo de datos. Un *sink* se utiliza para recibir y consumir eventos en un flujo de datos, generalmente utilizado en la capa de presentación para enviar datos a la capa de lógica de negocios. En el contexto de *Flutter* y el patrón BLoC, un *sink* se implementa como un `StreamSink`. Un `StreamSink` es una clase que implementa tanto la interfaz `Stream` como la interfaz `Sink`. Proporciona un método llamado `add()` que se utiliza para enviar eventos o datos a través de un flujo de datos, y estos eventos o datos son consumidos por el BLoC correspondiente.

Por ejemplo, en un escenario típico de uso del patrón BLoC, se puede tener un `StreamSink` en la capa de presentación que actúa como un *sink*. Este *sink* se utiliza para enviar eventos como acciones del usuario (como un botón presionado) o cambios en los datos de entrada a través del flujo de datos hacia el BLoC. El BLoC procesa estos eventos y puede emitir nuevas salidas a través de un `Stream` que es consumido por la capa de presentación para actualizar la interfaz de usuario.

En concreto, un *sink* es una forma de enviar eventos o datos desde la capa de presentación a la capa de lógica de negocios a través de un flujo de datos controlado por un BLoC.

- **Business Logic:** La lógica empresarial se refiere a las operaciones y el procesamiento de datos que ocurren en la aplicación. Esto puede incluir consultas a una base de datos, llamadas a una API, cálculos complejos, etc.
- **Events:** Los eventos son las acciones o sucesos que ocurren en la interfaz de usuario y que desencadenan un cambio en el estado de la aplicación. Por ejemplo, un botón presionado, un campo de texto modificado, etc.
- **State:** El estado representa el estado actual de la aplicación en un momento dado. Puede incluir datos cargados desde una fuente externa, resultados de operaciones, valores de campos de entrada, etc.
- **BLoC:** componente central del patrón BLoC. Actúa como un intermediario entre la interfaz de usuario y la lógica empresarial. Recibe eventos de la interfaz de usuario, procesa la lógica empresarial correspondiente y emite nuevos estados como resultado.
- **Streams:** Los *streams* son flujos de datos reactivos que se utilizan para transmitir eventos y estados dentro del Bloc. Permiten una comunicación bidireccional y en tiempo real entre los diferentes componentes de la aplicación.

- **StreamController:** clase fundamental que se utiliza para controlar y administrar flujos de datos en forma de *Stream*. Un *StreamController* actúa como una fuente de datos que emite eventos a medida que se producen cambios en el estado de la aplicación.

En términos generales, un *StreamController* es responsable de crear y gestionar un *Stream*, así como de enviar eventos a ese *Stream* cuando sea necesario. Proporciona métodos para añadir eventos al *Stream* y notificar a los suscriptores cuando se produce un nuevo evento.

Un *StreamController* puede utilizarse en conjunto con el patrón BLoC para separar la lógica de negocio de la interfaz de usuario. El BLoC actúa como un intermediario entre los componentes de la interfaz de usuario y los datos, y utiliza un *StreamController* para emitir eventos a medida que los datos cambian.

Algunas de las **funciones principales** de un *StreamController* son:

- Crear un *Stream* asociado que permita la transmisión de eventos.
- Añadir eventos al *Stream* mediante el método `add()`.
- Notificar a los suscriptores del *Stream* sobre nuevos eventos utilizando el método *stream*.
- Controlar el flujo de eventos con métodos como `sink` y `close`.

Se puede concluir que el *StreamController* es una clase que ayuda a administrar y controlar la transmisión de eventos a través de un *Stream*, facilitando la separación de la lógica de negocio y la interfaz de usuario.

En definitiva, el patrón BLoC en Flutter permite separar la lógica de negocio y la interfaz de usuario de una manera ordenada y gestionable. Proporciona un flujo de datos reactivo y predecible, lo que facilita la gestión de estados complejos y la actualización de la interfaz de usuario en respuesta a eventos específicos.

### 6.5.4.3.1. IMPLEMENTACIÓN DEL PATRÓN BLOC

Al igual que para el patrón Provider, se describe el proceso que se ha seguido para implementar el patrón BLoC en la aplicación DaSell, mediante un ejemplo genérico que facilite su comprensión. Los pasos que se han llevado a cabo son:

1. **Se agregan las dependencias necesarias**, mediante el archivo `pubspec.yaml` y nos debemos asegurar de tener las dependencias para el patrón BLoC.

```
dependencies:
  flutter_bloc: ^7.0.0
  equatable: ^2.0.0
```

Figura 91: Implementación BLoC - Dependencias BLoC

2. **Definición de estado**, para ello se define una clase que represente el estado de la aplicación. Esta clase debe extender de `Equatable` para facilitar la comparación de objetos. Por ejemplo:

```

class CounterState extends Equatable {
  final int count;

  CounterState(this.count);

  @override
  List<Object> get props => [count];
}

```

Figura 92: Implementación BLOC – Definición estados

3. **Definición de eventos**, se debe crear una clase para cada evento que pueda afectar el estado de la aplicación. Estas clases también deben extender de `Equatable`. Por ejemplo:

```

abstract class CounterEvent extends Equatable {
  const CounterEvent();

  @override
  List<Object> get props => [];
}

class IncrementEvent extends CounterEvent {}

class DecrementEvent extends CounterEvent {}

```

Figura 93: Implementación Patrón BLOC - Definición de eventos

4. **Creación del BLOC**, se define una clase que extienda de `BLOC` y especifique el estado inicial. Implementa el método `mapEventToState()` para manejar los eventos y emitir nuevos estados. Por ejemplo:

```

class CounterBloc extends Bloc<CounterEvent, CounterState> {
  CounterBloc() : super(CounterState(0));

  @override
  Stream<CounterState> mapEventToState(CounterEvent event) async* {
    if (event is IncrementEvent) {
      yield CounterState(state.count + 1);
    } else if (event is DecrementEvent) {
      yield CounterState(state.count - 1);
    }
  }
}

```

Figura 94: Implementación de BLOC - Creación de BLOC

5. **Implementación del BLOC en la interfaz de usuario**: en el *widget*, se puede utilizar el BLOC para interactuar con el estado y los eventos. Se puede utilizar el *widget* `BlocBuilder` para escuchar los cambios de estado y actualizar la interfaz de usuario en consecuencia. Por ejemplo:

```

class CounterPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Bloc Counter')),
      body: BlocBuilder<CounterBloc, CounterState>(
        builder: (context, state) {
          return Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Text('Count:  $\{state.count\}$ '),
                SizedBox(height: 16),
                RaisedButton(
                  child: Text('Increment'),
                  onPressed: () {
                    BlocProvider.of<CounterBloc>(context).add(IncrementEvent());
                  },
                ), // RaisedButton
                RaisedButton(
                  child: Text('Decrement'),
                  onPressed: () {
                    BlocProvider.of<CounterBloc>(context).add(DecrementEvent());
                  },
                ), // RaisedButton
              ],
            ), // Column
          ); // Center
        },
      ), // BlocBuilder
    ); // Scaffold
  }
}

```

Figura 95: Implementación BLOC - Implementación en widget

#### 6.5.4.5 COMPARATIVA: PATRÓN BLOC VS PROVIDER

El patrón Provider y el patrón BloC son dos enfoques diferentes para administrar el estado en aplicaciones de *Flutter*. A continuación, se proporciona una comparativa entre ambos:

##### OBJETIVO

- ❖ **Provider:** se utiliza para administrar el estado y la inyección de dependencias en una aplicación *Flutter*. Permite compartir los datos entre widgets de manera eficiente y manejar la reactividad de los mismos.
- ❖ **BloC:** El patrón *BLOC* (*Business Logic Component*) se utiliza para separar la lógica de negocio de la interfaz de usuario en una aplicación *Flutter*. Ayuda a gestionar los eventos y estados de una manera predecible y mantenible.

## CONCEPTO

- ❖ **Provider:** se utiliza un árbol de *widgets* para proporcionar y consumir objetos en toda la jerarquía de la aplicación. Los *widgets* pueden suscribirse a los cambios en los datos y ser notificados automáticamente cuando ocurren actualizaciones.
- ❖ **BloC:** se utiliza una combinación de *Streams* (flujos) y *Sinks* (sumideros) para comunicar eventos y estados entre componentes. Los Blocs actúan como intermediarios que procesan los eventos y emiten nuevos estados.

## RESPONSABILIDADES

- ❖ **Provider:** los *widgets* son responsables de consumir los datos proporcionados y actualizar su interfaz de usuario en consecuencia. También pueden modificar el estado global al emitir eventos.
- ❖ **BloC:** los *Blocs* son responsables de recibir eventos, procesarlos y emitir nuevos estados. Los *widgets* escuchan los cambios de estado y actualizan su interfaz de usuario en consecuencia.

## FLEXIBILIDAD

- ❖ **Provider:** proporciona una forma flexible de compartir datos y administrar el estado en una aplicación *Flutter*. Es especialmente útil cuando se trabaja con una cantidad pequeña o moderada de datos y no se necesita una lógica de negocio compleja.
- ❖ **BloC:** proporciona una mayor separación entre la lógica de negocio y la interfaz de usuario. Es más adecuado para aplicaciones más grandes o con lógica de negocio compleja, ya que permite una gestión más estructurada de eventos y estados.

Para concluir, el patrón *Provider* se centra más en la administración del estado y la inyección de dependencias, mientras que el patrón *Bloc* se centra en la separación de la lógica de negocio y la interfaz de usuario. Ambos patrones tienen sus propias ventajas y se pueden utilizar en diferentes contextos según las necesidades de la aplicación.

### 6.4.5.1 VENTAJAS IMPLEMENTACIÓN CONJUNTA PATRÓN BLOC Y PATRÓN PROVIDER

En el caso de la plataforma DaSell se ha realizado la implementación conjunta de los patrones Provider y BloC, donde esta puede ser beneficiosa en proyectos de desarrollo de aplicaciones complejas, como es este caso. Dado el escenario actual, se puede implementar el patrón Provider para gestionar la inyección de dependencias y proporcionar instancias de los BloCs a los diferentes componentes que lo necesiten.

Se realiza la implementación de un Provider centralizado que se encargue de crear y administrar los distintos Blocs de la aplicación. Los componentes de la interfaz de usuario pueden solicitar los Blocs necesarios al Provider y utilizarlos para gestionar la lógica de presentación y comunicación la lógica de negocio.

En una misma aplicación de *Flutter*, se puede implementar *Provider* para administrar el estado global de la aplicación y compartir datos entre los diferentes componentes. Se definen proveedores o servicios para diferentes aspectos de la aplicación, como datos de usuario, configuración, etc. Con ello, los widgets pueden consumir estos datos utilizando el widget *Consumer* o utilizando el *mixín Provider.of<T>(context)*.

El patrón Provider puede encapsular la lógica para crear y mantener las instancias de los BloCs, y los BloCs a su vez pueden utilizar el Provider para acceder a otros servicios o dependencias necesarios para su funcionamiento.

En definitiva, la implementación conjunta de los patrones Provider y Bloc puede ayudar a mantener una arquitectura modular, facilitar la gestión del estado y la inyección de dependencias, y separar claramente la lógica de presentación de la lógica de negocio en la aplicación.

## 6.6 ESTRUCTURA INTERNA DEL PROYECTO

En esta sección, se describe la organización en directorios que se ha realizado en la estructura interna del proyecto de DaSell. La organización de directorios se establece según las funcionalidades implementadas en el sistema, y por archivos en directorios y paquetes. Esto permite tener un mayor rendimiento en el desarrollo y fluidez a la hora de generar el código o actualizaciones en la aplicación.

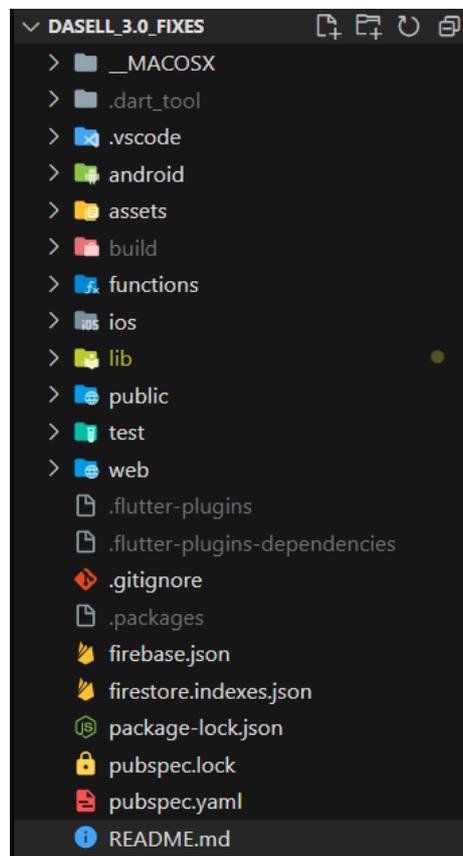


Figura 96: Estructura interna del proyecto

La organización se encuentra estructurada de la siguiente manera:

- ❖ **../.vscode:** carpeta específica que contiene archivos de configuración utilizados por Visual Studio Code (VSCode) para proporcionar un entorno de desarrollo integrado (IDE) optimizado para el desarrollo de aplicaciones Flutter. Dentro del directorio `vscode`, puedes encontrar varios archivos, como:

- **launch.json:** Este archivo define la configuración de depuración para ejecutar y depurar una aplicación *Flutter* en VSCode. Puedes personalizar las configuraciones de depuración según tus necesidades.
- **settings.json:** permite configurar preferencias específicas de VSCode para un proyecto *Flutter*. Se pueden ajustar aspectos como la sangría, el tamaño de la fuente, los temas de color, entre otros.
- **extensions.json:** enumera las extensiones recomendadas para el desarrollo de *Flutter* en VSCode.

La presencia del directorio `vscode` y sus archivos en un proyecto *Flutter* no es obligatoria. Sin embargo, proporciona una configuración optimizada para el desarrollo en VSCode y puede ayudar a aprovechar al máximo las características y herramientas específicas de *Flutter* en este entorno.

❖ **../android:** se encuentran los archivos relacionados con la configuración del proyecto y archivos relacionados para la plataforma Android. Este directorio contiene todos los archivos necesarios para la compilación y ejecución de la plataforma *Flutter* en un entorno Android. Contiene varios archivos y directorios relevantes como:

- **app/:** contiene los archivos específicos de la aplicación Android, como el archivo `build.gradle`, que configura la compilación de la aplicación y sus dependencias.
- **gradle/:** contiene los archivos y scripts relacionados con el sistema de compilación *Gradle*, utilizado para compilar y construir la aplicación Android.
- **Gradlew y gradlew.bat:** Estos son scripts ejecutables que permiten compilar y construir la aplicación utilizando *Gradle* sin necesidad de tener *Gradle* instalado globalmente en el sistema.
- **AndroidManifest.xml:** archivo de configuración importante para la aplicación Android. Contiene información sobre la aplicación, como el nombre del paquete, los permisos requeridos, las actividades principales, etc.
- **res/:** contiene los recursos de la aplicación Android, como archivos de diseño XML, imágenes, iconos, etc.

❖ **../assets:** carpeta donde se almacenan los archivos estáticos que se utilizarán en la aplicación, como imágenes, fuentes de texto, archivos de configuración, etc. para indicar a *Flutter* que un archivo debe ser considerado como un recurso estático, se debe agregar una referencia a ese archivo en el archivo `pubspec.yaml` del proyecto.

❖ **../build:** permite almacenar los archivos generados durante el proceso de compilación y construcción de una aplicación. Cuando ejecutas un proyecto de *Flutter*, se generan archivos temporales y compilados que se almacenan en el directorio `build`. Puede contener varios subdirectorios, dependiendo del tipo de compilación y plataforma objetivo. Algunos de los subdirectorios comunes dentro de `build` son:

- **build/ios:** contiene los archivos generados para la plataforma iOS cuando se compila la aplicación para dispositivos Apple.
- **build/android:** almacena los archivos generados para la plataforma Android cuando se compila la aplicación para dispositivos Android.

- **build/web:** Contiene los archivos generados cuando se compila la aplicación para la web.

Estos subdirectorios y sus contenidos son generados automáticamente por *Flutter* durante el proceso de compilación y no se deben editar manualmente. *Flutter* utiliza los archivos almacenados en el directorio *build* para ejecutar la aplicación en el dispositivo o plataforma seleccionada.

- ❖ **../functions:** contiene las *cloud functions* definidas en formato *JavaScript* o *TypeScript* que gestiona la API de *Cloud Functions de Firebase*.
- ❖ **../ios:** se refiere a la ubicación donde se encuentran los archivos y recursos relacionados con la plataforma iOS. Este directorio es generado automáticamente cuando se crea un nuevo proyecto de *Flutter* y contiene todo lo necesario para compilar y ejecutar la aplicación en dispositivos iOS.

El directorio `../ios` contiene principalmente dos subdirectorios importantes:

- **Runner:** Este subdirectorio contiene el archivo `AppDelegate.swift`, que es el punto de entrada de la aplicación en iOS. Aquí también se encuentran otros archivos relacionados con la configuración de la aplicación, como el `Info.plist` que contiene información sobre la aplicación, como su nombre, iconos, permisos, etc.
- **Assets:** Este subdirectorio es utilizado para almacenar los recursos estáticos de la aplicación, como imágenes, fuentes, archivos de configuración, etc., que serán utilizados en la aplicación iOS. Se pueden colocar los archivos en este directorio y acceder a ellos en tu código *Flutter* utilizando rutas relativas.

Es importante tener en cuenta que el directorio `ios` es específico de la plataforma iOS y solo se utiliza cuando se compila la aplicación para dispositivos iOS. Para compilar y ejecutar la aplicación en otras plataformas, como Android, se utiliza el directorio `android`.

- ❖ **../lib:** directorio donde se almacena la mayor parte del código fuente de la aplicación. Este directorio contiene los archivos de código fuente que componen la biblioteca o conjunto de funciones de la aplicación.

Dentro del directorio `lib`, se encuentran los archivos de *Dart* que contienen la lógica de la aplicación, como la creación de interfaces de usuario, la gestión de datos, la lógica de negocios, etc. Estos archivos se organizan en diferentes carpetas según la estructura de tu proyecto y tus preferencias de organización.

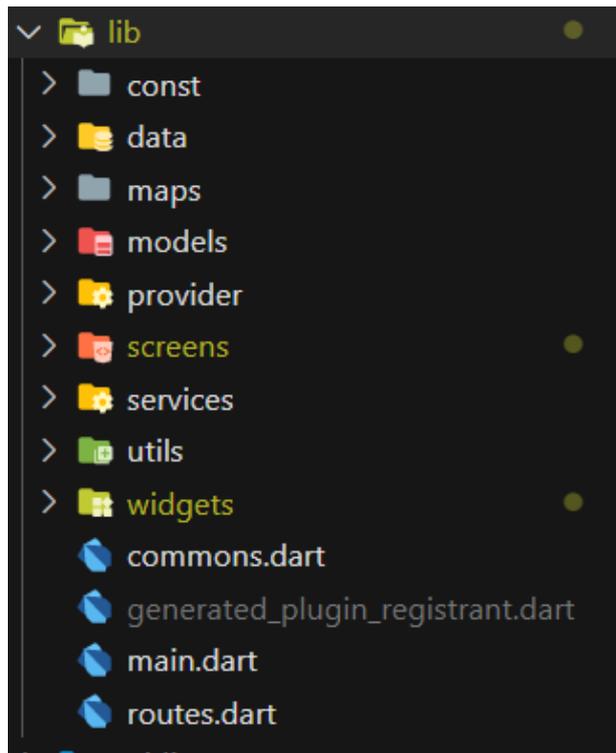


Figura 97: Estructura interna - directorio lib

- **/const:** directorio que almacena las clases que almacenan los valores constantes. Las constantes se refieren a valores que se declaran una vez y no cambian durante la ejecución de la aplicación. Las constantes se definen utilizando la palabra clave `const` en Dart y pueden ser utilizadas para almacenar valores como números, cadenas de texto, booleanos, listas o mapas inmutables. Ejemplos: `colors.dart`, `styles.dart`, etc.
- **/data:** directorio que almacena los datos de manera estática, generalmente en formato JSON.
- **/maps:** directorio que almacena los *widgets*, que generan la interfaz de usuarios. Los *screens*, que contienen los archivos de código para cada pantalla que pertenece a la funcionalidad, directorios para la gestión de estado mediante el patrón BLoC, servicios para obtener los datos tanto de manera externa mediante APIs, y la lógica de negocio para la gestión de todos los componentes definidos.

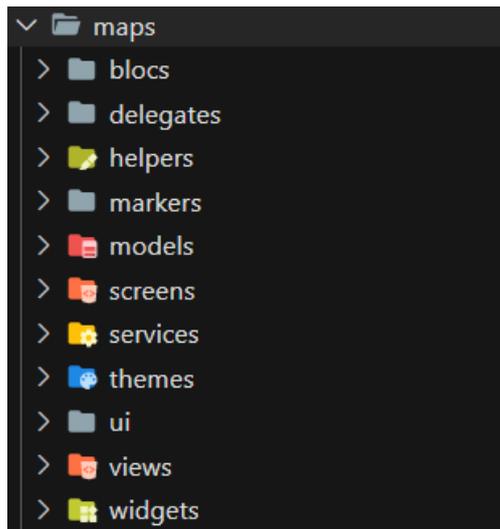


Figura 98: Directorios Maps - Sistema de seguimiento en tiempo real

- **/models:** se utiliza para almacenar las clases o estructuras de datos que representan el modelo de negocio o la lógica de la aplicación. Estas clases suelen contener propiedades y métodos que definen cómo se manejan los datos dentro de la aplicación y cómo se comunican con otras partes del código, como servicios o controllers.
- **/provider:** se definen los directorios que contienen los providers que permiten la administración y gestión eficiente del estado en toda la aplicación y facilita el intercambio de datos entre widgets. En términos generales, Provider se basa en el concepto de Inyección de Dependencias para proporcionar una forma eficiente y sencilla de traspasar y acceder a datos o estados entre *widgets*.
- **/screens:** en este directorio se organizan y almacenan las diferentes pantallas de la aplicación. Proporciona una forma organizada de estructurar y acceder a las diferentes pantallas o vistas que componen la interfaz de usuario de la aplicación.

Al organizar las pantallas en un directorio específico como `screens`, se puede mantener una estructura de proyecto limpia y modular, lo que facilita la navegación y el mantenimiento del código. Además, esto permite separar claramente la lógica de presentación y la lógica de negocio de la aplicación.

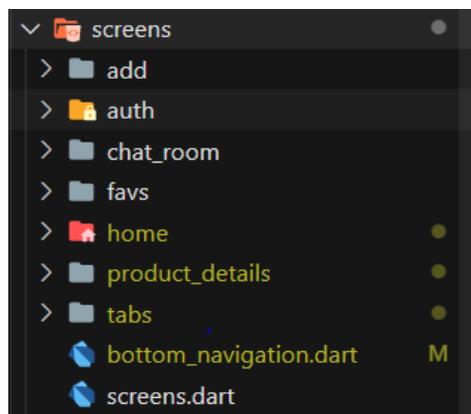


Figura 99: Estructura interna - Screens

- **/services:** clases o componentes que proporcionan funcionalidades específicas a lo largo de la aplicación. Los servicios son una forma de encapsular la lógica de negocio y la lógica de acceso a datos para que puedan ser reutilizados en diferentes partes de la aplicación. Se implementan como clases y se pueden utilizar para realizar diversas tareas, como comunicarse con las APIs, manejar el almacenamiento local, administrar la autenticación de usuarios, etc. Estos servicios se crean generalmente como clases independientes que se pueden inyectar en otros componentes de la aplicación, como widgets o bloques de estado.

La ventaja de utilizar servicios en un proyecto Flutter es que promueven la modularidad y el reutilización del código. Al encapsular la lógica en servicios separados, se facilita la realización de pruebas y el mantenimiento de la aplicación, ya que cada servicio puede tener su propia responsabilidad y ser probado de manera aislada.

- **/utils:** directorio o un paquete que contiene clases y funciones de utilidades. Estas clases y funciones son herramientas genéricas y reutilizables que brindan funcionalidades comunes y facilitan el desarrollo del proyecto.

El propósito de tener una carpeta o un paquete `utils` es mantener el código modularizado y facilitar la reutilización. Al agrupar las funciones y herramientas comunes en un solo lugar, se mejora la legibilidad del código y se evita la duplicación. Además, si una función o clase de utilidad necesita cambios o actualizaciones, solo se necesita modificar un archivo en lugar de buscar y modificar múltiples ubicaciones en el proyecto.

- **/widgets:** conjunto de directorios que albergan los widgets globales que se usan en el código fuente.
  - **commons.dart:** conjunto de paquetes importados globalmente que se implementan y se trata de código compartido y reutilizable en diferentes partes de la aplicación. Es una convención utilizada para agrupar funcionalidad
  - **main.dart:** punto de entrada principal de la aplicación. Contiene el método `main()`, que se ejecuta cuando se inicia la aplicación. Aquí se define la estructura básica de la aplicación y se configuran las rutas, los temas, los widgets y otros componentes esenciales.
  - **routes.dart:** archivo de configuración que se utiliza para definir y gestionar las rutas (pantallas) de la aplicación
- ❖ **../public:** carpeta dentro del proyecto donde se encuentran los archivos y recursos que estarán disponibles públicamente en la aplicación. Esta carpeta se utiliza para almacenar archivos estáticos como imágenes, fuentes, archivos HTML, archivos de configuración y otros recursos que deben ser accesibles desde el código Flutter.

Una vez se han colocado los archivos en el directorio `public` o en una subcarpeta dentro de él, podrás acceder a ellos utilizando el paquete `Flutter flutter/services`. Puedes utilizar `rootBundle.load()` para cargar un archivo específico o `rootBundle.loadString()` para cargar el contenido de un archivo como una cadena de texto.

- ❖ **../test:** se utiliza para almacenar todos los archivos de prueba relacionados con el código de la aplicación. El directorio `test` es un componente importante en el desarrollo de aplicaciones

porque proporciona una estructura organizada para escribir y ejecutar pruebas automatizadas. Aquí es donde se pueden crear archivos de prueba para probar diversas funcionalidades de tu aplicación, como la lógica de negocios, la interfaz de usuario y la integración con servicios externos.

Dentro del directorio `test`, se pueden tener subdirectorios adicionales para organizar las pruebas de la manera deseada. Los archivos de prueba en *Flutter* suelen tener la extensión `.dart` y utilizan un *framework* de pruebas como `flutter_test` para realizar las pruebas de manera eficiente.

Al ejecutar las pruebas, *Flutter* buscará automáticamente los archivos de prueba dentro del directorio `test` y ejecutará las pruebas definidas en ellos. Esto permite verificar si el código de la aplicación funciona correctamente y detectar posibles errores o comportamientos inesperados.

- ❖ **../web:** se refiere a una ubicación dentro de la estructura de archivos del proyecto donde se almacenan los archivos y recursos relacionados con el lado del cliente de una aplicación web desarrollada con *Flutter*.

Dado que *Flutter* es principalmente un *framework* para el desarrollo de aplicaciones móviles, no tiene un enfoque nativo en el desarrollo web. Sin embargo, *Flutter* permite crear aplicaciones web utilizando *Flutter for Web*, una función que permite compilar el código *Flutter* en JavaScript y ejecutarlo en un navegador web.

Cuando se crea un proyecto *Flutter* para la web, se crea un directorio llamado `web` en la raíz del proyecto. Este directorio contiene los archivos HTML, CSS, JavaScript y otros recursos necesarios para que la aplicación *Flutter* funcione en un entorno web.

El directorio `web` puede contener los siguientes archivos y carpetas:

- **index.html:** archivo HTML principal que se carga en el navegador y sirve como punto de entrada para la aplicación web *Flutter*.
  - **main.dart.js:** archivo JavaScript se genera automáticamente al compilar el código *Flutter* y contiene la lógica de la aplicación convertida a JavaScript.
  - **main.dart:** archivo Dart principal de la aplicación *Flutter*. Contiene el código fuente de la aplicación que se ejecuta en el lado del cliente.
  - **assets/:** Esta carpeta puede contener recursos estáticos, como imágenes, fuentes o archivos de configuración, que se utilizan en la aplicación web.
- ❖ **../.gitignore:** archivo especial que se utiliza para especificar los archivos y directorios que no deben ser rastreados por Git, el sistema de control de versiones. Git utiliza el archivo `gitignore` para determinar qué archivos y carpetas debe ignorar al realizar operaciones como `git push`, `git clone`, `git pull`, etc.

El objetivo principal de tener un archivo `gitignore` es evitar que se rastreen y comprometan archivos y directorios que no son necesarios en el control de versiones y que

pueden cambiar de una instancia a otra, como archivos de compilación, dependencias, archivos generados automáticamente, archivos sensibles, etc.

❖ **../firebase.json:** archivo de configuración que se utiliza para definir la configuración y las reglas de implementación de *Firebase Hosting*, así como para especificar las configuraciones adicionales para las funciones y los despliegues. Los campos más comunes que se pueden encontrar en un archivo *firebase.json* son:

- **Hosting:** Este campo se utiliza para configurar *Firebase Hosting*, que permite alojar y distribuir tu aplicación *Flutter*. Se pueden especificar propiedades como el directorio de implementación (`public`), las reescrituras de URL (`rewrites`), las redirecciones (`redirects`), el caché (`headers`), etc.
- **functions:** Este campo se utiliza para configurar las *Cloud Functions de Firebase*, que te permiten ejecutar código en el servidor de *Firebase*. Se puede especificar la configuración de las funciones, como la región en la que se implementan, el *runtime* (por ejemplo, Node.js), la cantidad de memoria asignada, etc.
- **database, firestore, storage:** Estos campos se utilizan para configurar las bases de datos y el almacenamiento de *Firebase*. Se pueden especificar las reglas de seguridad, la configuración regional, el límite de uso y otras configuraciones específicas de cada servicio.
- **emulators:** Este campo se utiliza para configurar los emuladores de *Firebase*, que son herramientas que te permiten simular el comportamiento de los servicios de *Firebase* en tu entorno local durante el desarrollo y las pruebas.

❖ **../firebase.indexes.json:** archivo de configuración, que permite definir los índices de consultas en la base de datos de *Firebase*. Los índices son utilizados para optimizar las consultas y mejorar el rendimiento de la aplicación al acceder a los datos almacenados en *Firebase*.

Cuando se realiza una consulta en *Firebase*, la base de datos utiliza estos índices para buscar y recuperar los datos de manera eficiente. Sin un índice adecuado, las consultas pueden ser más lentas y consumir más recursos.

En este archivo, se especifica la estructura de los índices necesarios para las consultas en tu base de datos *Firebase*. Puedes definir múltiples índices para diferentes consultas o combinaciones de campos. Cada índice está compuesto por una lista de campos y un tipo de ordenamiento (ascendente o descendente) para cada campo.

❖ **../package-lock.json:** archivo de bloqueo de versiones que se utiliza para asegurar que todas las dependencias del proyecto se instalen en las versiones específicas indicadas.

Cuando se utiliza *Flutter* para desarrollar una aplicación, es común que el proyecto tenga varias dependencias, como paquetes de terceros. Estas dependencias pueden tener sus propias dependencias y, a su vez, sus propias versiones. El archivo *package-lock.json* se utiliza para garantizar que se instalen las mismas versiones de las dependencias en todos los entornos.

Cuando se ejecuta el comando `flutter pub get` o `flutter pub upgrade` en un proyecto *Flutter*, se analiza el archivo `pubspec.yaml` para determinar las dependencias necesarias y luego se verifica el archivo `package-lock.json` para garantizar que todas las dependencias se instalen en las versiones específicas mencionadas en el archivo.

El archivo `package-lock.json` contiene información detallada sobre cada dependencia, incluyendo su nombre, versión, ubicación del repositorio y otras dependencias requeridas. Esta información se utiliza para resolver las dependencias y asegurar que el proyecto se construya de manera consistente en diferentes entornos.

Es importante incluir el archivo `package-lock.json` en el control de versiones de tu proyecto, ya que garantiza que todos los desarrolladores que trabajen en el proyecto tengan las mismas versiones de las dependencias. Esto evita posibles conflictos o problemas de compatibilidad que podrían surgir al utilizar diferentes versiones de las dependencias.

- ❖ **../pubspec.lock:** archivo generado automáticamente por el administrador de paquetes de *Dart* llamado `pub`. Este archivo registra las versiones exactas de las dependencias utilizadas en tu proyecto de *Flutter*, incluyendo las dependencias directas e indirectas.

El propósito principal del archivo `pubspec.lock` es asegurarse de que un proyecto *Flutter* siempre utilice las mismas versiones específicas de las dependencias. Esto ayuda a mantener la consistencia del entorno de desarrollo y garantiza que todos los miembros del equipo estén utilizando las mismas versiones de los paquetes.

Cuando se ejecuta el comando `flutter pub get` o `flutter pub upgrade`, el archivo `pubspec.lock` se actualiza automáticamente con las versiones más recientes y compatibles de las dependencias especificadas en el archivo `pubspec.yaml`. Estas versiones se resuelven en función de las restricciones definidas en el archivo `pubspec.yaml`, como las versiones mínimas y máximas permitidas.

- ❖ **../pubspec.yaml:** archivo de configuración de la aplicación, que permite administrar y definir las dependencias utilizadas en el proyecto, así como recursos adicionales y las configuraciones de la aplicación, como paquetes desarrollados por terceros, de manera rápida e eficiente. Las distintas secciones que podemos encontrar son:

- **name:** permite definir el nombre del proyecto. Debe ser único y normalmente sigue una convención de nomenclatura en minúsculas y sin espacios.
- **description:** proporciona una descripción breve del proyecto.
- **version:** especifica la versión actual del proyecto.
- **environment:** define las restricciones de versión del SDK de *Flutter* y *Dart* que admite el proyecto.
- **dependencies:** enumera las dependencias del proyecto. Se pueden agregar bibliotecas y paquetes de terceros necesarios, junto con sus respectivas versiones.
- **dev\_dependencies:** similar a las dependencias, esta sección enumera las dependencias específicas para el desarrollo, como herramientas de prueba, bibliotecas de análisis estático, etc.
- **flutter:** contiene configuraciones específicas de *Flutter* para el proyecto, como los `assets` (recursos estáticos), `fonts` (fuentes), y otros ajustes.

Al utilizar el comando `flutter pub get`, Flutter leerá el archivo `pubspec.yaml`, descargará e instalará las dependencias especificadas en el proyecto. Además, cuando se agrega o se actualiza una dependencia, es necesario ejecutar `flutter pub get` nuevamente para que los cambios surtan efecto.

- ❖ **README.md:** archivo de texto con formato *Markdown* que se utiliza para proporcionar información, instrucciones y detalles importantes sobre el proyecto. La extensión `.md` indica que está escrito en *Markdown*, un lenguaje de marcado ligero que permite dar formato al texto de manera sencilla.

---

## 7. PRUEBAS

---



En este capítulo, se englobará la realización de una serie de pruebas con el objetivo de verificar el correcto y óptimo funcionamiento de la aplicación. Así, en caso de que no sea correcto el funcionamiento o no se obtenga el resultado esperado, resolver, y poder corregir el comportamiento anómalo detectado.

Se pueden distinguir dos tipos de pruebas entre las realizadas en el sistema:

- ❖ Pruebas de caja blanca
- ❖ Pruebas de caja negra

## 7.1 PRUEBAS CAJA BLANCA

Las pruebas de caja blanca son llevadas a cabo durante la etapa de implementación del desarrollo, dado que dichas pruebas deben ser realizadas sobre el código fuente que se encuentra en proceso de desarrollo con el objetivo de evaluar y verificar la funcionalidad interna de la lógica de datos de cada uno de los módulos correspondientes, y comprobar que el flujo de información se mueve dentro de la aplicación. Por lo que, el procedimiento de estas pruebas es realizado por parte de los desarrolladores o el miembro del equipo destinado a testear este tipo de trabajos, y comprobar si existen vulnerabilidades.

Estas pruebas deben garantizar:

- Se ejecute al menos una vez cada instrucción de código en el programa.
- Se usen las estructuras definidas internamente.
- Se garantice que la toma de decisiones es correcta, teniendo la posibilidad de considerarse como verdaderas o falsas.
- Se realice la ejecución de los bucles, verificando el flujo general y los casos extremos.

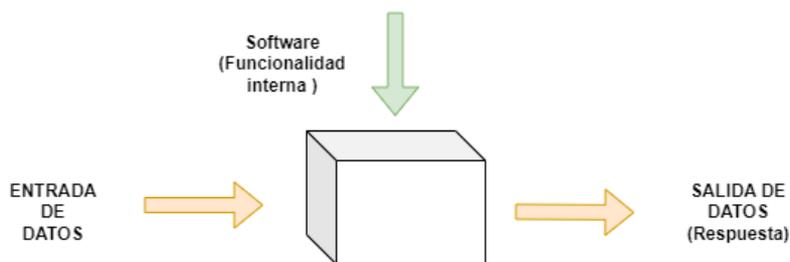


Figura 100: Pruebas de caja blanca

Teniendo en cuenta que el proyecto desarrollado es bastante amplio, implica que el número de pruebas de caja blanca a realizar es muy elevado. Por lo que las pruebas realizadas se han llevado a cabo según se desarrollaban las nuevas funcionalidades de la plataforma, verificando que todos los caminos posibles pueden cumplir su objetivo. Por este motivo, se especifica un resumen con los aspectos más destacados que se han considerado:

- Verificación de gestión de usuarios y roles asignados de manera correcta (usuario no registrado y no identificado, usuario registrado y no identificado, usuario registrado e identificado).
- Verificación de control y manejo de sesiones de forma correcta.
- Verificación de funcionalidades asignadas a usuario según su rol asignado.
- Verificación de ejecución del sistema sin conectividad.

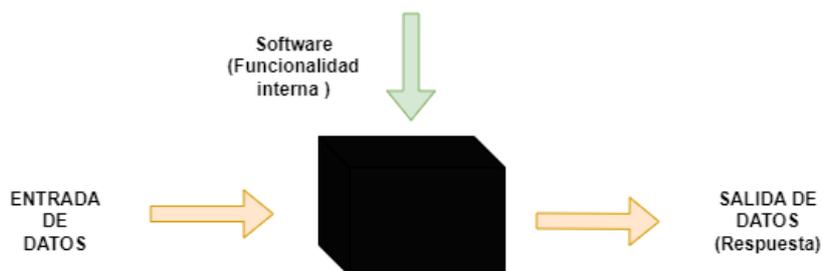
- Verificación de realización de consultas (inserción, obtención, modificación, eliminación) de manera correcta, retornando la información correcta desde el servicio de base datos Cloud Firestore en tiempo real.
- Verificación del funcionamiento y comunicación con los agentes externos, APIs, con los que interactúa la aplicación.
- Gestión del funcionamiento de métodos asíncronos en las peticiones realizadas hacia la base de datos.
- Comprobación de que los datos de la base de datos, Cloud Firestore, corresponde con los definidos en el código fuente de la aplicación mediante los modelos de datos definidos.
- Verificaciones y validaciones realizadas sobre los formularios diseñados de entrada de datos en el sistema.
- Verificación de la correcta obtención de la ubicación del dispositivo móvil mediante el que se conecta el usuario a la plataforma.

## 7.2 PRUEBAS CAJA NEGRA

En el caso de las pruebas de caja negra, tienen como objetivo verificar la funcionalidad por el motivo por el que se ha desarrollado la aplicación, sin tener en cuenta el código desarrollado.

Se centran en las entradas y salidas del sistema, sin necesidad de conocer la estructura y proceso que se ha llevado a cabo. Para obtener el detalle de cuáles deben ser esas entradas y salidas, nos basamos en los requerimientos de software y especificaciones funcionales.

Al igual que en el caso de las pruebas de caja blanca, dado que el tamaño del proyecto es elevado, el número de pruebas de esta tipología realizadas es bastante grande. Por lo que, por este motivo solo se muestran algunas de las pruebas realizadas como ejemplo a nivel de funcionalidad básica que debe satisfacer el sistema.



*Figura 101: Pruebas caja negra*

A continuación, se muestra el detalle de algunas de las pruebas realizadas:

ID y Nombre	PCN-01 REGISTRO DE USUARIOS
<b>Objetivo</b>	Verificar que el registro de un nuevo usuario desde la aplicación móvil se ha realizado correctamente.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. No exista una cuenta de usuario en el sistema con las mismas credenciales.</li> <li>2. Email introducido, debe tener un formato válido con el patrón definido y único.</li> <li>3. Contraseñas con longitud entre 8 y 16 caracteres, combinando caracteres alfanuméricos.</li> <li>4. El nombre y apellidos deben cumplir las restricciones impuestas.</li> </ol>
<b>Datos de entrada</b>	Nombre: Daniel Apellidos: Martín Cubero Email: <a href="mailto:test1@test.com">test1@test.com</a> Contraseña: abc123DaSell Dirección: Vía Roma (Segovia)
<b>Acción esperada</b>	Al seleccionar el botón “Registrarse”, el sistema debe almacenar la información del nuevo usuario, mostrando en pantalla la confirmación de la creación de este.
<b>Resultado</b>	Correcto

Tabla 170: PCN-01 Registro de usuarios

ID y Nombre	PCN-02 ACCEDER AL SISTEMA
<b>Objetivo</b>	Verificar el acceso a la plataforma de un usuarios mediante sus propias credenciales.
<b>Precondiciones</b>	El usuario registrado y no identificado debe haberse registrado previamente en el sistema
<b>Datos de entrada</b>	Email: <a href="mailto:test1@test.com">test1@test.com</a> Contraseña: abc123DaSell
<b>Acción esperada</b>	Al seleccionar el botón “Iniciar sesión”, el sistema debe redirrecionar al usuario a la página principal y se realiza la activación de funcionalidades asignadas al perfil de usuario autenticado.
<b>Resultado</b>	Correcto

Tabla 171: PCN-02 Acceder al sistema

ID y Nombre	PCN-03 LISTADO DE PUBLICACIONES
<b>Objetivo</b>	Verificar que el listado completo de publicaciones realizadas por todos los usuarios se visualiza correctamente.
<b>Precondiciones</b>	El usuario registrado e identificado debe haberse autenticado previamente en el sistema.
<b>Datos de entrada</b>	Ninguno
<b>Acción esperada</b>	Al acceder a la página de inicio de la aplicación se visualiza correctamente en listado completo de publicaciones de la plataforma.
<b>Resultado</b>	Correcto

Tabla 172: PCN-03 Listado de publicaciones

<b>ID y Nombre</b>		<b>PCN-04 INFORMACIÓN DE LA PUBLICACIÓN</b>
<b>Objetivo</b>	Verificar que el sistema valida toda la información requerida de las publicaciones para continuar con el proceso de publicación	
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado debe haberse autenticado previamente en el sistema.</li> <li>2. El listado completo de publicaciones debe contener al menos una publicación añadida.</li> <li>3. Seleccionar una miniatura de publicación desde el listado completo de publicaciones.</li> </ol>	
<b>Datos de entrada</b>	Ninguno	
<b>Acción esperada</b>	Al acceder a la miniatura de la publicación, visualizar toda la información relacionada con dicha publicación.	
<b>Resultado</b>	Correcto	

*Tabla 173: PCN-04 Información de la publicación*

<b>ID y Nombre</b>		<b>PCN-05 INFORMACIÓN DE LA PUBLICACIÓN</b>
<b>Objetivo</b>	Verificar que el sistema valida toda la información requerida de las publicaciones para continuar con el proceso de publicación	
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado debe haberse autenticado previamente en el sistema.</li> <li>2. El listado completo de publicaciones debe contener al menos una publicación añadida.</li> <li>3. Seleccionar una miniatura de publicación desde el listado completo de publicaciones.</li> </ol>	
<b>Datos de entrada</b>	Ninguno	
<b>Acción esperada</b>	Al acceder a la miniatura de la publicación, visualizar toda la información relacionada con dicha publicación.	
<b>Resultado</b>	Correcto	

*Tabla 174: PCN-05 Información de la publicación*

<b>ID y Nombre</b>		<b>PCN-06 BUSCADOR DE PUBLICACIONES</b>
<b>Objetivo</b>	Verificar que el sistema de búsqueda de publicaciones funciona correctamente.	
<b>Precondiciones</b>	El usuario registrado e identificado debe haberse autenticado previamente en el sistema.	
<b>Datos de entrada</b>	Datos de búsqueda	
<b>Acción esperada</b>	El sistema actualiza el listado mostrando las publicaciones que coinciden mediante el criterio de búsqueda realizado por el usuario.	
<b>Resultado</b>	Correcto	

*Tabla 175: PCN-06 Buscador de publicaciones*

ID y Nombre	PCN-07 MODIFICAR AVATAR DE PERFIL DE USUARIO
<b>Objetivo</b>	Verificar si el sistema permite al usuario actualizar su imagen asociada al avatar del perfil de usuario.
<b>Precondiciones</b>	El usuario registrado e identificado debe haberse autenticado previamente en el sistema.
<b>Datos de entrada</b>	Nueva imagen de avatar del perfil de usuario desde la cámara o galería del dispositivo.
<b>Acción esperada</b>	El sistema actualiza la imagen asociada al perfil del usuario autenticado, tras seleccionar el botón "Actualizar".
<b>Resultado</b>	Correcto

Tabla 176: PCN-07 Modificar avatar de perfil de usuario

ID y Nombre	PCN-08 REALIZAR PUBLICACIÓN
<b>Objetivo</b>	Verificar si el sistema permite al usuario realizar una nueva publicación en la plataforma tras la información introducida previamente.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado debe haberse autenticado previamente en el sistema.</li> <li>2. El usuario debe haber introducido toda la información obligatoria que solicita el sistema a la hora de realizar una publicación.</li> </ol>
<b>Datos de entrada</b>	Categoría: Informática y Electrónica Subcategoría: Ordenadores Título: Venta ordenador portátil HP. Descripción: Nvidia GeForce 1050 GTX, Intel Core i7 Estado de artículo asociado a la publicación: Como nuevo Estado de admisión de envío: Hago envíos Imágenes de la publicación: 1 imagen desde galería, 3 imágenes realizadas con cámara del dispositivo. Precio: 750€ Ubicación: Avd. Vía Roma, 40, Segovia.
<b>Acción esperada</b>	El sistema añade una nueva publicación en la base de datos con la información insertada, verificando su validez previamente.
<b>Resultado</b>	Correcto

Tabla 177: PCN-08 Realizar publicación

ID y Nombre	PCN-09 AÑADIR UNA NUEVA VALORACIÓN
<b>Objetivo</b>	Verificar si el sistema permite al usuario cliente realizar una nueva valoración a un usuario vendedor al realizar una transacción.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado debe haberse autenticado previamente en el sistema.</li> <li>2. El usuario debe haber realizado una transacción con el usuario vendedor previamente.</li> <li>3. El usuario debe acceder al sistema de valoración entre usuarios desde la publicación realizada por el usuario vendedor.</li> </ol>
<b>Datos de entrada</b>	Comentario: Todo perfecto, vendedor recomendado. Valoración: 4 estrellas

<b>Acción esperada</b>	El sistema añade una nueva valoración asociada a un usuario vendedor propietario de la publicación.
<b>Resultado</b>	Correcto

Tabla 178: PCN-09 Añadir una nueva valoración

<b>ID y Nombre</b>	<b>PCN-10 ELIMINAR UNA PUBLICACIÓN</b>
<b>Objetivo</b>	Verificar si el sistema permite al usuario vendedor eliminar una publicación realizada por sí mismo.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado debe haberse autenticado previamente en el sistema.</li> <li>2. El usuario debe haber realizado una publicación previamente en la plataforma.</li> <li>3. Seleccionar una miniatura de publicación desde el listado completo de publicaciones.</li> </ol>
<b>Datos de entrada</b>	Ninguno
<b>Acción esperada</b>	Al seleccionar el botón “Eliminar publicación”, y tras confirmar por el usuario su deseo de eliminar la publicación, la publicación seleccionada se elimina de la plataforma.
<b>Resultado</b>	Correcto

Tabla 179: PCN-10 Eliminar una publicación

<b>ID y Nombre</b>	<b>PCN-11 ACCESO A SISTEMA DE SEGUIMIENTO EN TIEMPO REAL</b>
<b>Objetivo</b>	Verificar si el sistema permite acceder al usuario al sistema se seguimiento en tiempo real.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario registrado e identificado debe haberse autenticado previamente en el sistema.</li> <li>2. El usuario debe haber accedido a la ventana “Compartir archivos” dentro del cuadro de mensaje de uno de los chats activos dentro del servicio de mensajería instantánea.</li> </ol>
<b>Datos de entrada</b>	Ninguno
<b>Acción esperada</b>	Al seleccionar el botón “Ubicación”, el sistema muestra el mapa que muestra la ubicación actual del usuario junto el sistema de seguimiento en tiempo real.
<b>Resultado</b>	Correcto

Tabla 180: PCN-11 Acceso a sistema de seguimiento en tiempo real

<b>ID y Nombre</b>	<b>PCN-12 CERRAR SESIÓN</b>
<b>Objetivo</b>	Verificar si el sistema permite a cualquier usuario autenticado cerrar sesión en la aplicación.
<b>Precondiciones</b>	El usuario registrado e identificado debe haberse autenticado previamente en el sistema.
<b>Datos de entrada</b>	Ninguno.
<b>Acción esperada</b>	El sistema muestra la pantalla de inicio de sesión de la plataforma.
<b>Resultado</b>	Correcto

Tabla 181: PCN-12 Cerrar sesión



---

## 8. MANUALES

---



## 8.2.1 INICIO DE SESIÓN

Al ejecutar la aplicación, se muestra la pantalla de inicio, donde se indica al usuario un formulario donde debe realizar el inicio de sesión en la plataforma.

Para iniciar sesión en la aplicación y contar con las funcionalidades proporcionadas por estas a los usuarios previamente registrados, el usuario deberá seguir los siguientes pasos:

1. Un requisito indispensable es que el usuario no disponga de una sesión activa en la aplicación, por lo que previamente se debe haber destruido el token de sesión si se ha iniciado sesión anteriormente.
2. Seleccionar el formulario de inicio de sesión.
3. Completar el formulario con las credenciales del usuario.
4. Seleccionar el botón "Iniciar sesión".

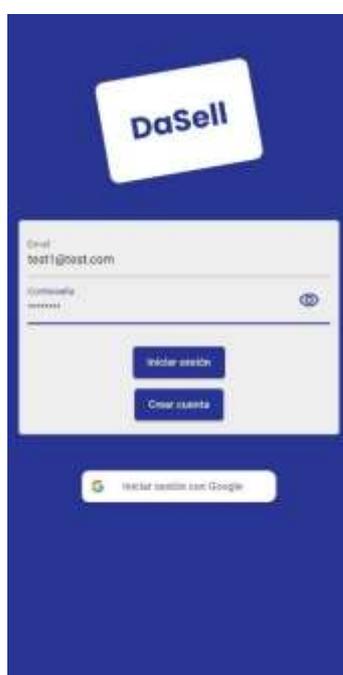


Figura 102: Manual de usuario - Inicio de sesión

## 8.1 MANUAL DE INSTALACIÓN

En esta sección se comentará el proceso de instalación del *software* necesario para la puesta en marcha de la aplicación. Esta aplicación desarrollada, al consumir el *backend* como un servicio API Rest ( Representational State Transfer), tan solo se requiere la instalación en el dispositivo para comenzar a utilizarla.

Los pasos a seguir para realizar la instalación de la aplicación móvil DaSell en el dispositivo son los siguientes:

1. En primer lugar, dado que el origen de la aplicación se considerará como de origen desconocido, Android e iOS indican que las aplicaciones que no provienen desde plataforma oficiales de *Google Play Store* o *Apple Store*, respectivamente. Se impone como requisito tener habilitado desde la configuración del dispositivo la instalación de aplicaciones con origen desconocido o de terceros.

2. Descargar el archivo con la extensión `.apk`<sup>1</sup> de la aplicación, directamente desde el dispositivo móvil o incorporar el mismo a través de un cable USB o una tarjeta SD.
3. Tras disponer el archivo de instalación `.apk`, se debe ejecutar en el sistema operativo del dispositivo para realizar el proceso de instalación.
4. Dado que la aplicación implementa las API de Google y API de MapBox, el dispositivo requerirá que los servicios de Google, si por algún motivo se encuentran desactualizados, el sistema solicita al usuario la actualización de estos.
5. Tras finalizar la instalación, durante la ejecución de la aplicación, el sistema solicitará al usuario permisos para habilitar o utilizar funcionalidades propias del dispositivo que son requeridas para el correcto funcionamiento de la aplicación. Por ejemplo, la ubicación del dispositivo, acceso a la cámara del dispositivo, etc. Estas funcionalidades solo podrán ser utilizadas por la aplicación si el usuario da el consentimiento para ello.
6. Tras este proceso, la aplicación está instalada al completo en el dispositivo con todas las funcionalidades disponibles para ser utilizadas en cualquier momento.

## 8.2 MANUAL DE USUARIO

En esta sección, tras haber realizado el análisis de todos los aspectos técnicos referentes a las funcionalidades, mediante los requisitos definidos, el diseño realizado previamente, y la implementación realizada. Tras este proceso, se puede explicar mediante un manual de usuario, el funcionamiento y manejo que se debe realizar de la aplicación desarrollada.

El manual de usuario tiene distintas instrucciones que debe seguir el usuario para un correcto manejo de la aplicación. Este manual puede suponer un punto de iniciación para todos los usuarios que deseen aprender, y obtener el máximo potencial a la plataforma. Además de proporcionar a los usuarios un soporte donde poder consultar cualquier tipo de cuestión que se tenga sobre el manejo en todo momento.

Para ello, se comenzará explicando la gestión de autenticación de usuario, parte esencial de la plataforma, ya que es necesario que el usuario se autentique previamente para poder usar las funcionalidades de las que dispone la aplicación.

### 8.2.2 INICIO DE SESIÓN MEDIANTE CUENTA DE GOOGLE

Al ejecutar la aplicación, se muestra la pantalla de inicio, donde se indica al usuario un formulario donde debe realizar el inicio de sesión en la plataforma.

Para iniciar sesión en la aplicación mediante su cuenta de Google y contar con las funcionalidades proporcionadas por estas a los usuarios previamente registrados, el usuario deberá seguir los siguientes pasos:

1. Un requisito indispensable es que el usuario no disponga de una sesión activa en la aplicación, por lo que previamente se debe haber destruido el token de sesión si se ha iniciado sesión anteriormente.

---

<sup>1</sup> **.apk**: archivo que contiene una aplicación Android. "APK" es el acrónimo de *Android Package*, lo que significa que es un paquete de instalación para dispositivos Android. Un archivo APK contiene todos los elementos necesarios para instalar y ejecutar una aplicación en un dispositivo Android, como el código de la aplicación, los recursos, las bibliotecas y el archivo manifest.

2. Seleccionar el botón de “Iniciar sesión con Google”.
3. Seleccionar dese la ventana emergente ofrecida por los servicios de Google, la cuenta de usuario de Google con la cual se desea autenticarse en la aplicación.



Figura 103: Manual de usuario - Inicio de sesión con Google

### 8.2.3 REGISTRO DE UN NUEVO USUARIO

Para crear una nueva cuenta en la plataforma y acceder a los servicios y funcionalidades adicionales proporcionadas a los usuarios registrados, el usuario deberá seguir los siguientes pasos:

1. Un requisito indispensable es que el usuario no disponga de una sesión activa en la aplicación, por lo que previamente se debe haber destruido el token de sesión si se ha iniciado sesión anteriormente.
2. Seleccionar el formulario de inicio de sesión.
3. Seleccionar el botón “Crear Cuenta”
4. Completar el formulario con las credenciales y la información del usuario solicitada por el sistema.
5. Seleccionar el botón “Registrarse”.
6. En caso de que el usuario ya se encuentre registrado, seleccionar el botón “Ya estoy registrado” para dirigirse a la pantalla de inicio de sesión.



Figura 104: Manual de usuario - Registro de un nuevo usuario

## 8.2.4 CONSULTAR INFORMACIÓN DE PERFIL DE USUARIO

Para acceder a la información personal de la cuenta del perfil de usuario del usuario autenticado, se deberán seguir los siguientes pasos:

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. Seleccionar el botón “Perfil”, situado en la parte derecha de la barra de navegación, situada en la parte inferior de la pantalla.
3. Acceder a la pantalla “Mi Perfil” donde se muestra la información del perfil de usuario autenticado. Desde esta pantalla se pueden realizar las siguientes acciones:
  - a. **Modificar avatar de perfil de usuario:** seleccionar el botón “Editar avatar de perfil de usuario” para realizar una de las siguientes acciones:
    - a. Modificar el avatar asociado al perfil de usuario con una imagen obtenida mediante la cámara del dispositivo móvil.
    - b. Modificar el avatar asociado al perfil de usuario seleccionando una imagen desde la galería del dispositivo móvil.
    - c. Eliminar la imagen asociada al avatar de perfil de usuario.
  - b. **Cerrar sesión:** seleccionar el botón “Cerrar sesión”, para cerrar la sesión actual del usuario autenticado y destruir toda su información no almacenada en base de datos.

- c. **Eliminar cuenta:** seleccionar el botón “Eliminar Cuenta”, para eliminar el usuario asociado al perfil de usuario autenticado actualmente en el sistema., junto con toda la información relativa a las publicaciones realizadas.
- d. **Configuración:** acceder a la configuración global de la aplicación
  1. **Tema oscuro:** Seleccionar botón “Tema oscuro”, para activar/desactivar el tema oscuro.
  2. **Notificaciones:** Seleccionar botón “Notificaciones” para activar/desactivar las notificaciones push que se muestran en el dispositivo.
  3. **Transacciones:** Seleccionar el botón “Transacciones”, para acceder a la pantalla de “Mis publicaciones”.



Figura 105: Manual de usuario - Consultar información de perfil de usuario

## 8.2.5 GESTIÓN DE MIS PUBLICACIONES

Para realizar la gestión de las publicaciones por parte del usuario, se puede acceder bien desde el botón “Transacciones” desde la pantalla de “Mi Perfil” o desde el menú de navegación de la parte inferior de aplicación.

En esta funcionalidad, se pueden visualizar los listados completos de publicaciones marcadas como favoritas, publicaciones marcadas como vendidas o publicaciones publicadas por el usuario autenticado actualmente.

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. Seleccionar el botón “Mis publicaciones”, segundo botón situado en la parte izquierda de la barra de navegación, situada en la parte inferior de la pantalla.
3. Acceder a la pantalla “Mis publicaciones” donde se muestra un menú con tres opciones:

- a. **Visualizar listado de publicaciones realizadas:** si se selecciona la opción “Publicadas”, se muestra el listado completo de publicaciones publicadas por el usuario autenticado, en formato miniatura.
- ❖ En el caso que la opción seleccionada “Publicadas”, muestre el listado de publicaciones sin contenido, se muestra un mensaje informativo, para publicar un nuevo producto se debe seleccionar el botón “Subir producto”.



Figura 106: Manual de usuario - Gestión de mis publicaciones - Publicadas

- b. **Visualizar listado de publicaciones marcadas como favoritas:** si se selecciona la opción “Favoritas”, se muestra el listado completo de publicaciones marcadas como favoritas por el usuario autenticado, en formato miniatura.
- ❖ En el caso que la opción seleccionada “Favoritas”, muestre el listado de publicaciones sin contenido, se muestra un mensaje informativo, para realizar la búsqueda de productos que se puedan activar/desactivar como favoritas se debe seleccionar el botón “Buscar producto”, el cual redirige a la página de inicio de la aplicación.



Figura 107: Manual de usuario - Gestión de mis publicaciones - Favoritas

- c. **Visualizar listado de publicaciones marcadas como vendidas:** si se selecciona la opción “Vendidas”, se muestra el listado completo de publicaciones marcadas como vendidas por el usuario autenticado, en formato miniatura.
  - ❖ En el caso que la opción seleccionada “Vendidas”, muestre el listado de publicaciones sin contenido, se muestra un mensaje informativo, indicando al usuario que puede realizar publicaciones en la aplicación.

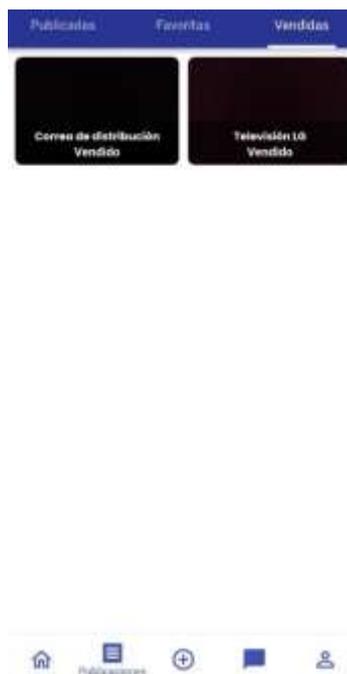


Figura 108 Manual de usuario - Gestión de mis publicaciones - Vendidas

4. Para visualizar la información completa de una publicación, se debe seleccionar la miniatura de una publicación desde el listado completo que se esté consultando actualmente, donde se mostrará a pantalla completa toda la información correspondiente a la publicación seleccionada.

## 8.2.6 PÁGINA DE INICIO

Para acceder a la página de inicio, el usuario comprador/vendedor deberá seguir los siguientes pasos:

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. Seleccionar el botón “Inicio”, situado en la parte izquierda de la barra de navegación, situada en la parte inferior de la pantalla.
3. Acceder a la pantalla “Inicio” donde se puede visualizar el listado completo de publicaciones realizadas en toda la plataforma. Desde esta pantalla se pueden realizar las siguientes acciones:



Figura 109: Página de Inicio – Listado completo de publicaciones

- a. **Filtrado por intervalo de precio:** si se desea realizar un filtrado por intervalo de precio, seleccionar el icono situado en la parte superior, e indicar mediante la barra de desplazamiento el intervalo deseado.
  - ❖ Seleccionar el botón “Aplicar” para establecer el intervalo deseado.

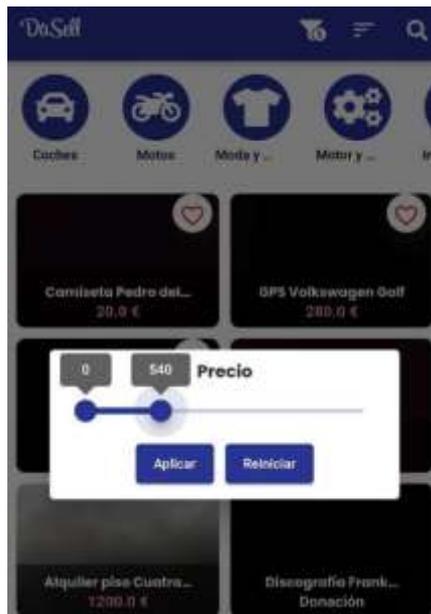


Figura 110: Página de Inicio - Filtrado por intervalo de precio

- b. **Ordenar listado por cercanía:** si se desea ordenar el listado por cercanía, seleccionar el botón “Más cercano”, el cual reordena el listado mostrando en primer lugar la miniatura de la publicación, más cercano a la posición actual del usuario autenticado.
- c. **Ordenar listado por fecha de publicación más reciente:** si se desea ordenar por fecha de publicación más reciente, seleccionar el botón “Más reciente”, el cual reordena el listado mostrando en primer lugar la miniatura de la publicación, con la fecha de publicación más reciente respecto al fecha actual.



Figura 111: Página de Inicio - Ordenar por más cercano o más recientes

4. **Gestión de barra de búsqueda:** para realizar una búsqueda de una publicación mediante los criterios deseados, en primer lugar, se debe seleccionar la barra de búsqueda, mediante el botón situado en la esquina superior derecha de la pantalla, que nos permite acceder a la vista general de la barra de búsqueda.
  - a. Seleccionar el cuadro de texto de la barra de búsqueda.
  - b. Escribir el contenido que se desea encontrar, donde se mostrará los títulos y descripciones coincidentes respecto al criterio de búsqueda deseado.
  - c. Seleccionar la publicación que se desea visualizar desde el listado de miniaturas mostrado respecto a los criterios de búsqueda introducidos.



Figura 112: Página de Inicio - Gestor de búsqueda

5. **Filtrado por categoría:** para mostrar todas las publicaciones coincidentes con una misma categoría seleccionada.
  - a. Seleccionar el listado horizontal de todas las categorías disponibles en la plataforma, situado en la parte superior de la pantalla del dispositivo.
  - b. Seleccionar una categoría del listado sobre la que deseamos realizar el filtrado.
  - c. Visualizar y seleccionar miniatura de publicación del listado de categorías coincidentes con la categoría seleccionada previamente.



Figura 113: Página de Inicio - Filtrado por categoría

6. **Actualizar listado de publicaciones mostrado:** deslizar con el dedo desde la parte superior de la pantalla, bajo la barra de inicio, hacia abajo. Produciéndose una actualización del listado de publicaciones mostrado en ese instante.

## 8.2.7 GESTIÓN DE PUBLICACIÓN

Para acceder a la página que muestra la información detallada de una publicación concreta, el usuario comprador/vendedor deberá seguir los siguientes pasos:

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. Seleccionar la miniatura de la publicación la cual se desea visualizar su información al completo, esta acción se puede realizar desde cualquier lugar de la aplicación en la que nos encontremos donde se muestre un listado de miniaturas de las publicaciones disponibles en la plataforma.
3. Acceder a la pantalla donde se puede visualizar la información detallada de una publicación realizada previamente. Desde esta pantalla se pueden realizar las siguientes acciones:
  - a. **Retroceder a la página anterior:** seleccionar el botón incorporado en la barra superior, situado en la parte superior izquierda de la pantalla del dispositivo móvil.
  - b. **Modificar una publicación:** seleccionar el botón “Modificar publicación”, incorporado en la barra superior, situado en la parte superior derecha de la pantalla del dispositivo móvil. Esta funcionalidad se detallará al completo en la sección **8.2.8. Modificar una publicación.**
  - c. **Menú de opciones para el usuario vendedor sobre su publicación:** seleccionar el botón que permite acceder al menú desplegable, incorporado en la barra superior, situado en la parte superior derecha de la pantalla del dispositivo móvil.
    1. **Eliminar publicación:** seleccionar el botón con la opción “Eliminar publicación”.

- ❖ Confirmar que se desea eliminar la publicación de la plataforma mediante el botón “Confirmar”.
  - ❖ Cancelar el proceso de eliminar publicación, seleccionando el botón “Salir”.
2. **Marcar publicación como vendida:** seleccionar el botón con la opción “Marcar publicación como vendida”.
- ❖ Confirmar que se desea marcar como vendida una publicación de la plataforma mediante el botón “Confirmar”.
  - ❖ Cancelar el proceso de marcar una publicación como vendida, seleccionando el botón “Cancelar”.
- d. **Visualizar imágenes asociadas a la publicación:** deslizar el dedo por el carrusel de imágenes para visualizar la/s imagen/es asociadas a la publicación.

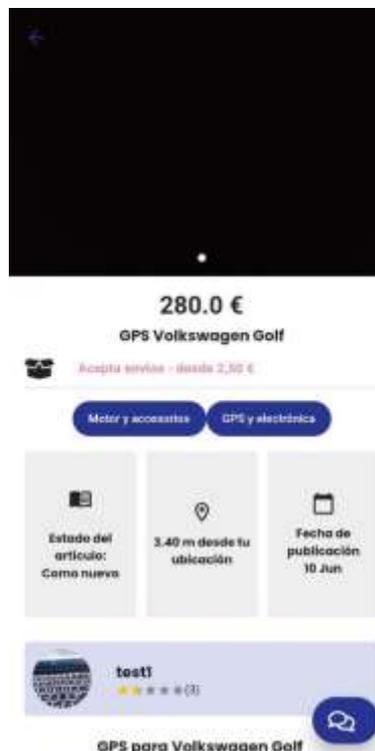


Figura 114: Gestión de una publicación - Información general

- e. **Visualizar información del usuario vendedor propietario de la publicación:** seleccionar el cuadro, situado bajo la información general de la publicación, donde se muestra el avatar asociado al perfil del usuario vendedor propietario de la publicación, su nombre, la media de valoración en formato de estrellas y el número de valoraciones recibidas.
- ❖ Para acceder al sistema de valoración entre usuarios correspondiente al usuario vendedor del que se está visualizando la información, seleccionar sobre el cuadro de información. El funcionamiento de dicha funcionalidad, se detallará en la sección **8.2.9 Gestión de sistema de valoración entre usuarios**.



Figura 115: Gestión de publicación - Acceso sistema de valoración usuario vendedor

- f. **Visualizar mapa con la ubicación asociada a la publicación:** seleccionar el mapa que muestra la ubicación asociada a la publicación, situado en la parte inferior de la pantalla que muestra la información de la publicación.
- ❖ Seleccionar mapa para visualizarlo a pantalla completa en el dispositivo móvil.
  - ❖ Seleccionar el botón situado en la parte inferior derecha de la pantalla del dispositivo para redirigir la ubicación asociada a la publicación, en caso de desplazarse por el mapa.
  - ❖ Para reducir el mapa a tamaño completo en la pantalla del dispositivo, seleccionar el botón de la parte superior de la izquierda.



Figura 116: Gestión de publicación - Mapa con ubicación actual

- g. **Acceder al servicio de mensajería instantánea:** seleccionar el botón flotante de “chat”, que se encuentra en la parte inferior derecha de la pantalla del dispositivo, para abrir una nueva conversación con el usuario vendedor de la publicación que está siendo visualizada en este momento, o abrir el chat en activo que se mantiene con ese usuario vendedor.



Figura 117: Gestión de Publicación - Acceder a servicio de mensajería instantánea

## 8.2.8 MODIFICAR UNA PUBLICACIÓN

Para acceder a la vista para modificar una publicación, se deberán seguir los siguientes pasos:

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. Acceder al detalle de la publicación, a través de su miniatura correspondiente, y ser el usuario vendedor propietario de la publicación.
3. Seleccionar el botón “Modificar publicación”, situado en la esquina superior derecha de la pantalla del dispositivo.
4. Acceder a la pantalla “Modificar publicación” donde se muestra la información de la publicación seleccionada. Desde esta pantalla se pueden realizar las siguientes acciones:
  - ❖ Tras pulsar el icono, barra superior derecha, se despliega el formulario.
  - ❖ Modificar los campos sobre los que se desean realizar cambios.
  - ❖ Seleccionar el botón “Actualizar publicación”, para actualizar la información del usuario.



Figura 118: Gestión de publicación - Modificar una publicación

## 8.2.9 GESTIÓN DE SISTEMA DE VALORACIONES ENTRE USUARIOS

Para acceder al sistema de valoraciones entre usuarios, el usuario comprador/vendedor deberá seguir los siguientes pasos:

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. Acceder al detalle de la publicación, a través de su miniatura correspondiente, y ser el usuario vendedor propietario de la publicación.
3. Seleccionar el cuadro, situado bajo la información general de la publicación, donde se muestra el avatar asociado al perfil del usuario vendedor propietario de la publicación, su nombre, la media de valoración en formato de estrellas y el número de valoraciones recibidas.
4. Acceder a la pantalla donde se muestra la información del perfil del usuario vendedor propietario de la publicación. Desde esta pantalla se pueden realizar las siguientes acciones:
  - a. **Visualizar información del usuario vendedor propietario de la publicación:** visualizar la información del usuario vendedor, en la parte superior de la pantalla, donde se muestra su nombre, media de puntuación de valoraciones, número total de valoraciones recibidas y su avatar de perfil de usuario.
  - b. **Visualizar listado de publicaciones en venta:** seleccionar la pestaña “En venta” para visualizar el listado completo de las miniaturas de las publicaciones marcadas “en venta” por el usuario vendedor.



Figura 119: Gestión de sistema de valoración entre usuarios - Listado publicaciones en venta

- c. **Visualizar listado completo de valoraciones:** seleccionar la pestaña “Opiniones” para visualizar el listado completo de las opiniones que ha recibido el usuario vendedor.
  - ❖ **Visualizar información de valoración:** deslizar el listado completo de opiniones, realizando el gesto de mover la pantalla hacia arriba y hacia abajo para visualizar todas las valoraciones realizadas. Se puede visualizar el avatar de perfil usuario asociado al usuario propietario de la valoración, su nombre, la puntuación dada, y la fecha de publicación.



Figura 120: Gestión de sistema de valoración entre usuarios - Listado de valoraciones recibidas

❖ **Menú opciones sobre valoración:** para realizar distintas acciones sobre una valoración, seleccionar el icono situado en la parte superior derecha de cada valoración, donde se despliega un menú en el que se puede realizar:

1. **Modificar valoración:** seleccionar la primera opción del menú desplegable mostrado en la parte inferior de la pantalla del dispositivo.
  - a. **Modificar puntuación añadida previamente en la valoración:** seleccionar el número de estrellas que se desea añadir de nuevo.
  - b. **Modificar comentario añadido previamente en la valoración:** introducir en el cuadro de texto el nuevo comentario que se desea añadir.
  - c. **Confirmar información modificada:** seleccionar el botón “Enviar” para modificar la información de la valoración.
  
2. **Eliminar valoración:** seleccionar la segunda opción del menú desplegable mostrado en la parte inferior de la pantalla del dispositivo.
  - a. **Confirmar eliminar valoración:** seleccionar el botón “Confirmar” mostrado en el cuadro de dialogo.
  - b. **Cancelar proceso de eliminar valoración:** seleccionar el botón “Cancelar” para cancelar el proceso de eliminación.

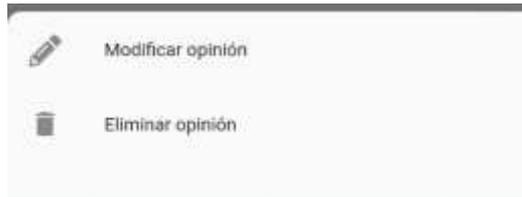


Figura 121: Menú de opciones - Modificar valoración y eliminar valoración

- ❖ **Añadir una nueva valoración:** seleccionar el botón flotante que se muestra en la esquina inferior derecha de la pantalla del dispositivo móvil.
  - a. **Añadir puntuación asociada a la valoración:** añadir puntuación de la valoración mediante el formato de puntuación en estrellas mostrado, seleccionando el número de estrellas deseado.
  - b. **Añadir un comentario asociado a la valoración:** seleccionar el cuadro de dialogo e introducir el comentario deseado que se quiere asignar a la valoración.
  - c. **Añadir valoración:** seleccionar el botón “Enviar” en caso de que se desee añadir una nueva valoración al usuario vendedor con la información introducida previamente.

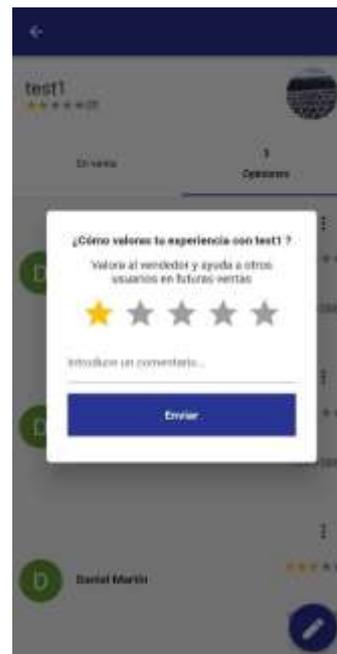


Figura 122: Añadir una valoración

## 8.2.10 GESTIÓN DE PROCESO DE PUBLICACIÓN

Para acceder a la página que permite realizar el proceso de publicación, el usuario vendedor deberá seguir los siguientes pasos:

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. **Seleccionar el botón “Añadir”**, situado en la parte central de la barra de navegación, situada en la parte inferior de la pantalla.
3. **Acceder a la pantalla “Añadir”**, donde se puede visualizar el listado completo de categorías disponibles en toda la plataforma, primer paso para comenzar el proceso de publicación de una nueva publicación. A partir de esta pantalla se pueden realizar las siguientes acciones de forma consecutiva, siendo un proceso en el que se debe seguir el orden de acciones establecido:
  - a. **Seleccionar la categoría deseada:** se visualiza el listado completo de categorías que se encuentran en la aplicación, y se selecciona la categoría en la que se desea añadir la publicación.



Figura 123: Gestión de proceso de publicación - Seleccionar categoría

- b. **Seleccionar la subcategoría deseada:** se visualiza el listado completo de subcategorías que se encuentran en la aplicación, y se selecciona la subcategoría, incluida en la categoría previamente seleccionada, en la que se desea añadir la publicación.



Figura 124: Gestión de proceso de publicación - Seleccionar subcategoría

4. **Acceder a la página “Detalles del producto”**, para ello, se debe seleccionar previamente la categoría y subcategoría deseada. En esta página, se pueden realizar las siguientes acciones:
  - a. **Añadir título:** añadir en el cuadro de texto correspondiente, la información que se desea añadir como título.
  - b. **Añadir descripción:** añadir en el cuadro de texto correspondiente, la información que se desea añadir como descripción.
  - c. **Modificar categoría seleccionada:** en el caso que se desee modificar la categoría seleccionada previamente, seleccionar el botón que indica “Categoría”, lo que permite volver a seleccionar una categoría distinta.
  - d. **Modificar subcategoría seleccionada:** en el caso que se desee modificar la subcategoría seleccionada previamente, seleccionar el botón que indica la “Subcategoría”, lo que permite volver a seleccionar una subcategoría distinta.
  - e. **Indicar si se admite o no envíos del producto asociado a la publicación:** indicar mediante el selector “Hago envíos”, activando o desactivando el selector si se admiten o no envíos.
  - f. **Indicar estado del producto asociado a la publicación:** indicar mediante la barra deslizante, el estado del producto asociado a la publicación, entre los estados definidos dentro de la plataforma.
  - g. **Seleccionar el botón “Siguiente”**, mostrado en la parte inferior de la pantalla del dispositivo, para continuar a la siguiente vista.



Figura 125: Gestión del proceso de publicación - Detalles del producto

5. **Acceder a la página “Añadir imágenes del producto”**, para ello, se debe haber introducido la información solicitada anteriormente. En esta página, se pueden realizar las siguientes acciones:
  - a. **Añadir imagen mediante la cámara del dispositivo móvil:** seleccionar el botón “Cámara”, para acceder a la aplicación de la cámara del dispositivo móvil y realizar la fotografía que deseamos añadir a la publicación. Para ello podemos ajustar la configuración de la aplicación de cámara del dispositivo.
  - b. **Añadir imagen mediante la galería del dispositivo móvil:** seleccionar el botón “Galería” para acceder a la aplicación de galería del dispositivo, y seleccionar la imagen que se desea añadir.
  - c. **Visualizar carrusel de imágenes añadidas a la publicación:** tras añadir al menos una imagen, se puede visualizar las imágenes añadidas, mediante el carrusel de imágenes disponible. En caso de disponer de más de una imagen, deslizar con el dedo hacia los laterales de la pantalla del dispositivo para visualizar el resto de imágenes.
  - d. **Seleccionar el botón “Siguiente”**, para confirmar que se desean añadir las imágenes insertadas previamente y continuar con el proceso de publicación.

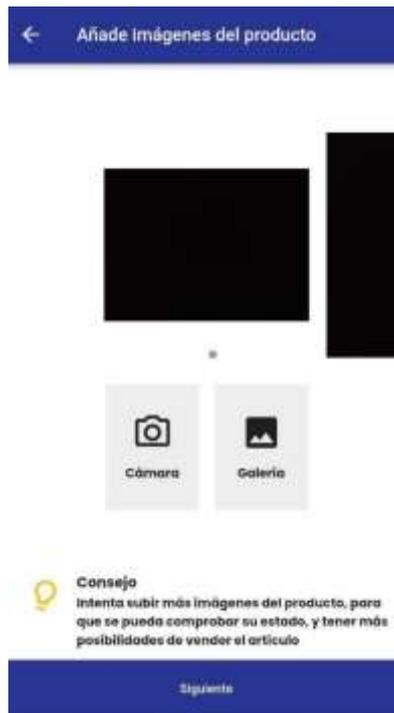


Figura 126: Gestión de proceso de publicación - Añadir imágenes del producto

6. **Acceder a la página “Introduzca precio y localización”**, para ello, se debe haber introducido la información solicitada anteriormente. En esta página, se pueden realizar las siguientes acciones:
- a. **Introducir precio en la publicación:** introducir la cantidad, en euros, en el cuadro de texto “Precio”, de la cantidad que se desea añadir a la publicación del artículo.
  - b. **Indicar que la publicación es una donación:** seleccionar el icono “quiero donar este artículo” para marcar la publicación en ese estado.
  - c. **Seleccionar una ubicación:** para añadir una ubicación a la publicación, se debe seleccionar una de estas opciones:
    1. **Ubicación actual:** seleccionar el botón “Ubicación actual”, situado bajo el cuadro que muestra el mapa, para escoger la localización en la que se encuentra actualmente el usuario con el dispositivo móvil.
    2. **Elegir ubicación:** seleccionar el botón “Elegir ubicación”, situado bajo el cuadro que muestra el mapa. Al seleccionarlo, se ampliará el mapa en la pantalla del dispositivo, mostrando la ubicación actual del usuario. Para seleccionar una ubicación distinta a la actual, seleccionar con el dedo en el lugar del mapa en el que se desea establecer la ubicación, y se situara en el mapa un marcador indicando las coordenadas seleccionadas.
      - a. **Redirigir a ubicación actual:** En cualquier lugar en el que nos encontramos, seleccionar el botón situado en la esquina derecha inferior de la pantalla, para redireccionar a la ubicación actual.

Figura 127: Gestión de proceso de publicación - Introducir precio y localización

7. **Confirmar publicación:** Seleccionar el botón “Publicar”, para confirmar que se desea añadir la publicación con la información insertada.

### 8.2.11 GESTIÓN DE SERVICIO DE MENSAJERÍA INSTANTÁNEA

Para acceder a la página que permite utilizar la funcionalidad del servicio de mensajería instantánea, el usuario vendedor deberá seguir los siguientes pasos:

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. **Seleccionar el botón “Chats”**, situado en la parte derecha de la barra de navegación, situada en la parte inferior de la pantalla.
3. **Acceder a la pantalla “Chats”**, donde se puede visualizar el listado completo de conversaciones activas mediante el servicio de mensajería instantánea con usuarios vendedores. Por cada conversación activa de chat, se puede observar el avatar de perfil asociado al otro usuario con el que se mantiene activa la conversación, el último mensaje enviado, y la fecha de publicación del último mensaje enviado.
4. **Seleccionar chat:** seleccionar el chat al que se desea acceder desde el listado de conversaciones activas.



Figura 128: Gestión de servicio de mensajería instantánea - Listado de chats

5. **Visualizar chat:** se visualiza el historial completo de mensajes enviados previamente, realizando el gesto de deslizar hacia arriba o hacia abajo, donde se pueden realizar las siguientes acciones:

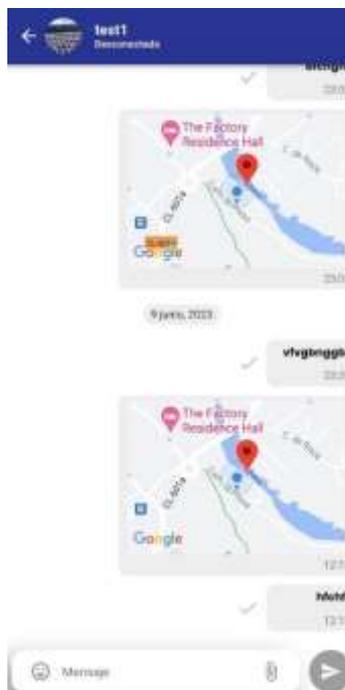


Figura 129: Gestión de servicio de mensajería instantánea – Visualizar chat

- a. **Escribir mensaje:** seleccionar el cuadro de texto, situado en la parte inferior de la pantalla del dispositivo, donde se permite introducir el texto que se desea enviar mediante el teclado del dispositivo móvil.



- e. **Compartir imagen desde la cámara del dispositivo:** seleccionar el botón situado en primer lugar, en la ventana emergente donde se muestran los distintos tipos de archivos a compartir.
1. **Acceder a la cámara del dispositivo,** donde se debe realizar la captura de la imagen deseada, para modificar la configuración de la cámara del dispositivo, ajustar las opciones disponibles en la aplicación.
  2. **Confirmar imagen capturada:** en caso de desear añadir al chat la imagen realizada, confirmar mediante el icono de confirmación, situado en la parte derecha de la pantalla del dispositivo.
  3. **Cancelar imagen capturada:** en caso de desear realizar de nuevo la fotografía, seleccionar el botón situado en la parte izquierda de la pantalla del dispositivo.
- f. **Compartir imagen desde la galería del dispositivo:** seleccionar el botón situado en segundo lugar, en la ventana emergente donde se muestran los distintos tipos de archivos a compartir.
1. **Acceder a la galería del dispositivo,** acceder a la aplicación de la galería del dispositivo móvil, y seleccionar la imagen deseada dentro de los álbumes disponibles.
  2. **Confirmar imagen seleccionada:** en caso de desear añadir al chat la imagen seleccionada, confirmar mediante el icono de confirmación, situado en la parte derecha de la pantalla del dispositivo.
- g. **Ubicación:** seleccionar el botón situado en tercer lugar, en la ventana emergente donde se muestran los distintos tipos de archivos a compartir. Al seleccionar esta opción se accederá al sistema de seguimiento en tiempo real, que será detallado en la sección **8.2.12 Sistema de seguimiento en tiempo real.**
- h. **Visualizar imagen del historial de mensajes del chat:** seleccionar la imagen que se muestra en miniatura en el mensaje enviado previamente por el chat. Esta se amplía a pantalla completa, donde se puede ampliar o reducir el tamaño de la imagen que se está visualizando.
- i. **Visualizar miniatura de mapa compartido mediante el sistema de seguimiento en tiempo real:** seleccionar la miniatura del mapa que se muestra en el mensaje para ampliar el mapa a pantalla completa, donde se pueden realizar las siguientes acciones:
1. **Realizar acciones sobre el mapa:** sobre el mapa se pueden realizar los siguientes movimientos:
    - a. **Rotar sobre el mapa,** realizando el gesto de giro en la pantalla del dispositivo.
    - b. **Reducir tamaño del mapa,** realizando el gesto de reducir el tamaño de pantalla del dispositivo.
    - c. **Ampliar tamaño del mapa,** realizando el gesto de ampliar el tamaño de pantalla del dispositivo.
    - d. **Mostrar ubicación actual,** seleccionar el botón que redirige a la ubicación actual.
    - e. **Visualizar información de la ruta trazada,** distancia y tiempo en ser recorrida.
    - f. Visualizar información de ubicación de origen y ubicación de destino.

- g. **Seleccionar el botón “Cancelar seguimiento”,** mostrado en la parte inferior de la pantalla, para dejar de compartir el seguimiento en tiempo real con el otro usuario.



Figura 133: Gestión de servicio de mensajería instantánea - mapa con ubicación actual

## 8.2.12 GESTIÓN DE SISTEMA DE SEGUIMIENTO EN TIEMPO REAL

Para acceder al sistema de seguimiento en tiempo real, el usuario comprador/vendedor deberá seguir los siguientes pasos:

1. Es un requisito indispensable que el usuario se encuentre registrado y autenticado en la aplicación.
2. Acceder a la página de “Chats”, mediante el botón “Chats”, situado en la parte derecha de la barra de navegación, situada en la parte inferior de la pantalla.
3. **Acceder a la pantalla “Chats”,** donde se puede visualizar el listado completo de conversaciones activas mediante el servicio de mensajería instantánea con usuarios vendedores.
4. **Seleccionar chat:** seleccionar el chat al que se desea acceder desde el listado de conversaciones activas.
5. **Visualizar chat:** se visualiza el historial completo de mensajes enviados previamente, realizando el gesto de deslizar hacia arriba o hacia abajo, donde se pueden realizar las siguientes acciones:
  - a. **Compartir archivos:** seleccionar el icono situado dentro del cuadro de texto, que representa el botón “Compartir archivos”, se realizará la apertura de una ventana donde se mostraran las tipologías de archivos que se pueden compartir mediante el servicio de mensajería instantánea.
  - b. **Ubicación:** seleccionar el botón situado en tercer lugar, en la ventana emergente donde se muestran los distintos tipos de archivos a compartir. Al seleccionar esta opción se

accederá al sistema de seguimiento en tiempo real, donde se podrá realizar las siguientes acciones:

**1. Realizar búsqueda de ubicación de destino:** seleccionar la barra de búsqueda, situada en la parte superior de la pantalla del dispositivo, para acceder a la vista de la barra de búsqueda completa, donde se podrán realizar estas acciones:

**a. Realizar búsqueda de ubicación de destino mediante la barra de búsqueda:** seleccionar la barra de búsqueda, y escribir el lugar que se quiere establecer como ubicación de destino, mediante el teclado del dispositivo. Tras ello, confirmar el texto introducido por teclado.

**1. Seleccionar ubicación del listado de sugerencias:** tras introducir la ubicación, seleccionar del listado de sugerencias, la ubicación deseada.



Figura 134: Gestión de sistema de seguimiento en tiempo real - Seleccionar ubicación

**2. Confirmación ubicación de destino deseada:** tras seleccionar la ubicación del listado de sugerencias mostrado, confirmar ubicación, donde el sistema traza la ruta que se debe recorrer, y muestra información correspondiente a la ruta.

**b. Colocar ubicación manualmente:** seleccionar el botón “Colocar ubicación manualmente”, situado bajo la barra de búsqueda. Tras ello, se muestra una nueva vista donde se pueden realizar las siguientes acciones:

**1. Confirmar destino:** situar el marcador, mostrado en el centro de la pantalla del dispositivo, en la ubicación que deseamos seleccionar como ubicación de destino. Para confirmar esta ubicación, seleccionar el botón “Confirmar destino”,

2. **Salir de la búsqueda de ubicación de destino:** seleccionar el botón “Atrás”, situado en la parte superior izquierda de la pantalla del dispositivo.

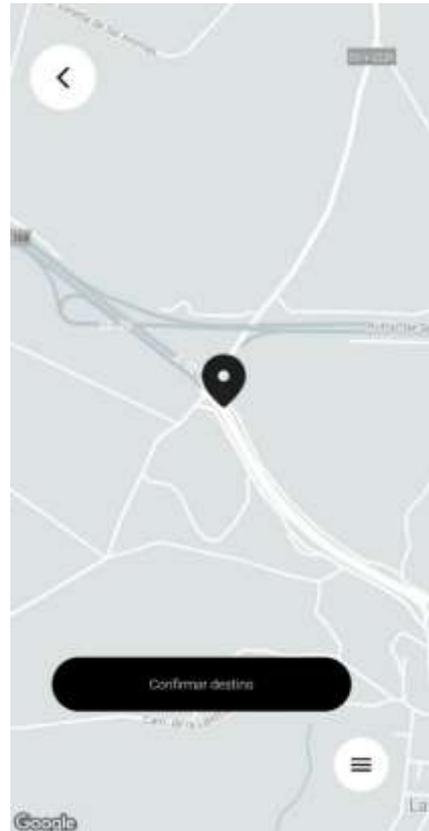


Figura 135: Gestión de sistema de seguimiento en tiempo real - Confirmar ubicación seleccionada

2. **Menú de opciones:** seleccionar el botón de menú, situado en la esquina inferior derecha de la pantalla del dispositivo, donde se despliega un menú radial donde se pueden realizar las siguientes acciones:
  - a. **Mostrar/no mostrar el recorrido de la ruta del usuario:** seleccionar el primer botón mostrado en el menú, situado en la parte superior derecha del menú radial, para activar o desactivar el trazado de la ruta que recorre el usuario en tiempo real.
  - b. **Redirigir a la ubicación actual:** seleccionar el segundo botón mostrado en el menú, situado en la parte superior derecha del menú radial, para redirigir a la ubicación actual del usuario.
  - c. **Iniciar sistema de seguimiento en tiempo real:** seleccionar el tercer botón mostrado en el menú, situado en la parte central del menú radial, para iniciar el seguimiento en tiempo real de la ubicación del usuario mediante el mapa mostrado.
  - d. **Compartir ubicación actual mediante el servicio de mensajería instantánea:** seleccionar el cuarto botón mostrado en el menú, situado en la parte inferior izquierda del menú radial, para compartir la ubicación actual del usuario, junto con el trazado y su información obtenida.

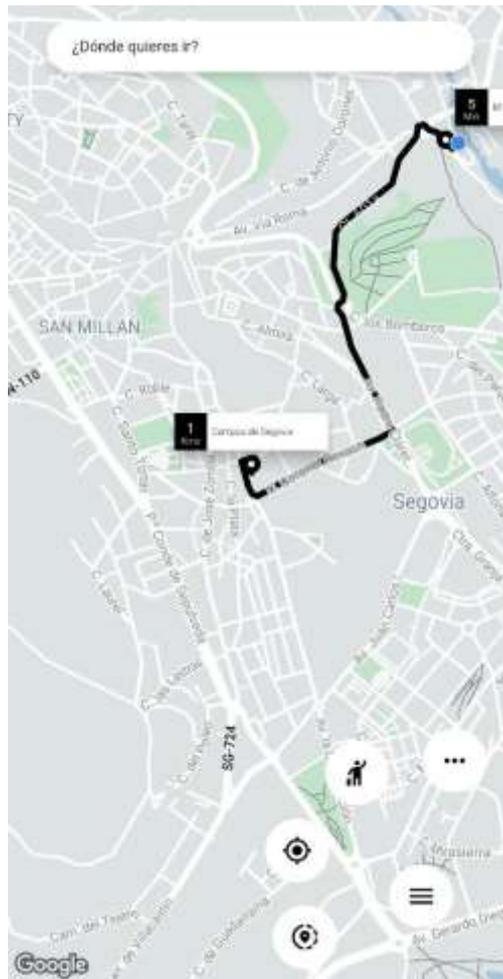


Figura 136: Gestión de sistema de seguimiento en tiempo real - Menú de opciones



---

## 9. CONCLUSIONES

---

## 9.1 CONCLUSIONES

El Trabajo de Fin de Grado (TFG), en mi opinión, es una gran oportunidad para demostrar los conocimientos conseguidos a lo largo del grado. No es fácil, y mucho menos, en el caso de este proyecto, donde se implementa un lenguaje y un entorno, totalmente distinto a los conocimientos sobre este tipo de tecnologías que se imparten en el grado. Uno de los aspectos más importantes, según mi consideración, es la falta de experiencia previa en la gestión y desarrollo en un proyecto de tal magnitud. Para mí, ha supuesto mi primer desarrollo completo de una aplicación, donde he tenido que ser capaz de establecer mis propias pautas o controles, ya que anteriormente siempre he tenido alguien que dirigiera el proceso durante cualquier proyecto académico.

Esto ha generado incertidumbre desde un primer momento, ya que según avanzaba el proyecto, los requisitos y propuestas iniciales crecían de manera exponencial y en poco tiempo, por lo que mi capacidad de análisis y de toma de decisiones, se ha visto reforzada. Igualmente, también se ha reforzado la capacidad de adaptación al cambio, según avanzaba el desarrollo de la aplicación.

A su vez, he tenido cierto grado de dificultad en el desarrollo, especialmente en la implementación de bases de datos, NoSQL, ya que no disponía de ninguna experiencia previa en su utilización. Esto ha supuesto un aumento en el tiempo dedicado a la investigación, estudio y desarrollo para la parte de la persistencia de datos. Este esfuerzo se ha traducido en un refuerzo en el conocimiento de las bases de datos, y lo cual valoro mucho, pues en el ámbito profesional, cada día son más demandados profesionales con experiencia en este tipo de bases de datos. En relación, con ello, se implementó Firebase, donde también se ha debido realizar un gran proceso de estudio para establecer cuál de sus servicios de bases de datos, permiten una implementación más óptima dentro del desarrollo. El manejo, entendimiento, adaptación e integración de este entorno en la aplicación era requisito indispensable para el devenir de este proyecto y, por este motivo, estas tareas han sido duras e intensas.

Por otro lado, ante la necesidad de aprender un nuevo lenguaje como Dart, en un entorno Flutter, ha sido necesario mucho tiempo de investigación, consulta de tutoriales, realización de cursos, lectura de manuales, así como, muchos ensayos prueba-error, para conseguir añadir al proyecto las distintas funcionalidades. También he encontrado dificultad, a la hora de utilizar servicios y librerías de terceros.

El desarrollo de este proyecto, comenzó como una idea que surge de realizar una aplicación del ámbito de *e-commerce*. En el contexto actual, tenemos la responsabilidad de hacer de este mundo, un mundo más sostenible, y para lo cual puede resultar de interés, dar una segunda vida a aquellos objetos que no utilizamos o la posibilidad de vender objetos nuevos, que por una razón u otra deseamos vender, así como ofrecer o contratar distintos tipos de servicios sin necesidad de intermediarios. Estas casuísticas, me dieron pie a reflexionar de la necesidad de disponer de una plataforma que facilite alcanzar este fin.

En conclusión, este desarrollo de proyecto ha supuesto un reto personal, ya que empezó siendo algo que parecía más simple y ha terminado llegando a unos niveles de complejidad bastante altos. Esto ha supuesto que, en muchos de los momentos del desarrollo, no se llegara a avanzar como se deseaba desde un planteamiento inicial, por lo que no se veía el final del desarrollo, tras el tiempo invertido.

Este proyecto ha conseguido despertar en mí, el interés para seguir aprendiendo más en cuanto al desarrollo de aplicaciones móviles, para poder aplicar los conocimientos adquiridos durante la resolución de este proyecto a mi ámbito laboral o profesional. También, este proyecto ha supuesto

darme cuenta de la necesidad de poner en valor la disponibilidad de un buen equipo de trabajo y desarrollo, que sirva de apoyo en proyectos de tal magnitud. Por todo ello considero que el objetivo perseguido en la realización de este TFG se ha visto cumplido a nivel personal.

## 9.2 FUTURAS MEJORAS

Al igual que se ha ido definiendo a lo largo de este documento, durante el desarrollo del proyecto y conforme se iba adquiriendo un mayor manejo del entorno de su lenguaje, han surgido propuestas y sugerencias que se han tenido que descartar, dado que se alejaban de los objetivos establecidos para una primera versión de la aplicación, y suponían una gran desviación de la planificación temporal establecida.

Las mejoras que se podrán implementar en un futuro, forman parte de una amplia serie de ideas, donde la aplicación puede extenderse en futuros desarrollos. Las funcionalidades son:

- ❖ **Pasarela de pago:** posibilidad de realizar la transacción y el pago por medio de la aplicación, con la implementación de una pasarela de pago mediante APIs que permitan realizar pagos de forma segura.
- ❖ **Notificaciones:** mayor número de notificaciones para que el usuario tenga un mayor conocimiento de lo que ocurre en la plataforma.
- ❖ **Verificación de cuentas de usuario:** comprobación mediante DNI ú otro sistema de identificación, para comprobar la indetidad veraz del usuario vendedor y minimizar así, el riesgo de ocurrencia de estafa.
- ❖ **Autenticación en dos pasos:** sistema de autenticación en dos pasos, mediante huella dactilar o confirmación por email, que nos permita dotar de una mayor seguridad y robustez al sistema.
- ❖ **Obtener confirmaciones de acciones realizadas:** confirmación mediante una plataforma de webmail, de confirmaciones de registro, confirmaciones de compra o cambio de contraseña del perfil de usuario.
- ❖ **Sistema de alertas:** permite comunicar al usuario que se ha realizado una publicación sobre algún producto o servicio deseado, o se han producido modificaciones en una publicación en la que está interesado.



---

# REFERENCIAS

---

Para finalizar, en esta sección se van a representar las distintas fuentes que han ido consultando durante el desarrollo del proyecto. Estas fuentes, se van a ordenar en dos tipologías: bibliografía, los cuales incluyen los libros de consulta; y en webgrafía, donde se incluyen los artículos de consulta, foro de desarrolladores y manuales online.

## 10.1 BIBLIOGRAFÍA

- ❖ Julien Trillard (2021). *“Desarrolle sus aplicaciones móviles multiplataforma con Dart”* [1ª Edición], Editorial ENI.
- ❖ Vicente Carbonell (2018). *“Firebase: trabajar en la nube”* [2ª Edición], Editorial S.A. MARCOMBO.
- ❖ Simone Alessandria (2021). *“Flutter Cookbook: with Flutter 2.2 and Dart”* [3ª Edición], Editorial Packt Publishing

## 10.2 WEBGRAFÍA

- ❖ **Google Cloud – Cloud Computing Services** Disponible en: <https://cloud.google.com/> ,\_Último acceso: 23/04/2023
- ❖ **Cloud Firestore | Firebase Google** Disponible en: <https://firebase.google.com/docs/firestore/> ,\_Último acceso: 12/05/2023
- ❖ **Firestore Authentication | Firebase Google** ,\_ Disponible en: <https://firebase.google.com/docs/auth/> ,\_ Ultimo acceso: 23/05/2023
- ❖ **Android Developers – Android Developers** Disponible en: <https://developer.android.com/> ,\_Último acceso: 25/05/2023
- ❖ **Stack Overflow - We <3 people who code** Disponible en: <https://stackoverflow.com/> ,\_ Último acceso: 28/05/2023
- ❖ **Github - Where the world builds software.** Disponible en: <https://github.com/> ,\_ Último acceso: 01/06/2023
- ❖ **entreDesarrolladores - Recibe ayuda de expertos** Disponible en: <https://entredesarrolladores.com/> ,\_Último acceso: 27/05/2023
- ❖ **Codexpedia – Android** Disponible en: <https://www.codexpedia.com/category/Android/> ,\_ Último acceso: 23/04/2023
- ❖ **Medium - Mi primer app con Firebase y Android** ,\_ Disponible en: <https://medium.com/@ykro/mi-primer-app-con-firebase-yandroid-parte-1-fd0b7d717e0b> ,\_ Último acceso: 26/04/2023
- ❖ **AWS Amazon - ¿Qué es NoSQL? Bases de datos no relacionales con excelente rendimiento y modelos de datos flexibles** Disponible en: <https://aws.amazon.com/es/nosql/> ;\_ Último acceso: 26/03/2023

- ❖ **Graph Everywhere - Bases de Datos NoSQL | Qué son, marcas, tipos y ventajas**, Disponible en: <https://www.grapheverywhere.com/bases-de-datos-nosql-marcastipos-ventajas/>, Último acceso: 30/04/2023
- ❖ **MongoDB - Fundamentos de las bases de datos NoSQL** Disponible en: <https://www.mongodb.com/es/nosql-explained>, Último acceso: 03/05/2023
- ❖ **Flutter - Introducción al patrón BLoC**, Disponible en: <https://xurxodev.com/introduccion-al-patron-bloc/>, Último acceso: 02/06/2023
- ❖ **Chema Rubio - Flutter everywhere**, Disponible en: <https://www.bbvanexttechnologies.com/blogs/flutter-everywhere/>, Último acceso: 27/05/2023
- ❖ **Digital Guide Ionos - Flutter: Introducción al framework multiplataforma**, Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-flutter/>, Último acceso: 03/06/2023
- ❖ **Pub dev - The official package repository for Dart and Flutter apps**, Disponible en: <https://pub.dev/>, Último acceso: 29/05/2023
- ❖ **GitHub – repositorios GitHub**, Disponible en: <https://github.com/>, Último acceso: 08/06/2023
- ❖ **FlutterDev - Build apps for any screen**, Disponible en: <https://flutter.dev/>, Último acceso: 04/06/2023
- ❖ **AbdulMuaz Aqeel - Flutter Clean Architecture Series**, Disponible en: <https://devmuaz.medium.com/flutter-clean-architecture-series-part-1-d2d4c2e75c47#:~:text=The%20Clean%20Architecture%20is%20the,that%20communicate%20through%20defined%20interfaces.>, Último acceso: 11/05/2023
- ❖ **Alberto Chamorro - Arquitecturas y patrones de diseño en Flutter**, Disponible en: <https://medium.com/flutter-madrid/arquitecturas-y-patrones-de-dise%C3%B1o-en-flutter-cap%C3%ADtulo-1-mvvm-591ad3d63bb>, Último acceso: 03/06/2023

## 10.3 CONTENIDO AUDIOVISUAL

En esta sección, se describen todos aquellos contenidos audiovisuales, tutoriales y cursos online consultados para la realización del proyecto.

- ❖ Udemy | Fernando Herrera - **“Flutter: Tu guía completa para IOS y Android”**, Disponible en: <https://www.udemy.com/course/flutter-ios-android-fernando-herrera/>, Último acceso: 13/05/2023
- ❖ Udemy | Fernando Herrera - **“Flutter avanzado: Lleva tu conocimiento al siguiente nivel”**, Disponible en: <https://www.udemy.com/course/flutter-avanzado-fernando-herrera/>, Último acceso: 28/05/2023
- ❖ Udemy | Fernando Herrera - **“Flutter: Una introducción al SDK de Flutter”**, Disponible en: <https://www.udemy.com/course/flutter-avanzado-fernando-herrera/>, Último acceso: 03/05/2023
- ❖ Udemy | Fernando Herrera - **“Flutter Intermedio: Diseños profesionales y animaciones”**, Disponible en: <https://www.udemy.com/course/flutter-disenos-y-animaciones/>, Último acceso: 09/05/2023
- ❖ YouTube | Fernando Herrera- **Canal Fernando Herrera - @DevTalles**, Disponible en: <https://www.youtube.com/@DevTalles>, Último acceso: 11/05/2023
- ❖ Youtube -| Brais Moure - **Canal MoureDev by Brais Moure**, Disponible en: <https://www.youtube.com/c/MouredevApps/playlists>, Último acceso: 14/05/2023
- ❖ Youtube |Diegoveloper – **Canal Diegoveloper.** Disponible en: <https://www.youtube.com/@diegoveloper>, Último acceso: 10/05/2023

