



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍA INFORMÁTICA DE
SEGOVIA

Grado en Ingeniería Informática de Servicios y
Aplicaciones

AVANTIS

Aplicación móvil para
la compra de billetes y
reserva de viajes

Autor: Pablo Jambrina Sanz

Tutor: Aníbal Bregón Bregón

Tutor: Miguel Ángel Martínez Prieto

Agradecimientos

A toda mi familia, principalmente a mis padres que han estado siempre en primera línea recorriendo este camino conmigo, apoyándome y ayudándome a sacar fuerzas de donde no las tenía en muchos momentos. A mi hermana porque siempre que lo he necesitado me ha facilitado buenos consejos y opiniones además de estar conmigo en los peores momentos del proyecto. A mis amigos y compañeros durante esta etapa de mi vida que han sido un gran desahogo cuando más presión tenía.

Y por último a mis profesores de la titulación, en especial a Aníbal y Miguel Ángel por aceptar sin dudarlo ser mis tutores de proyecto, resolviendo todos y cada uno de mis problemas y sugerencias y entendiendo en todo momento la situación en la que me encontraba.

Resumen

El objetivo principal de este Trabajo de Fin de Grado se basa en la creación de una aplicación móvil dirigida a plataformas Android que permita **la gestión y la compra de billetes y abonos de transporte** por parte de los usuarios en cualquier momento y en cualquier lugar.

Por este motivo, se busca que el usuario cuente con una herramienta intuitiva y simple que facilite todo lo posible este proceso de adquisición y posterior gestión.

Además, también se pretende que la información más importante y relevante esté siempre accesible a los usuarios, y por este motivo también contará con un foro de discusión, un tablón con las últimas noticias y promociones y un buscador de estaciones.

En resumen, la finalidad del proyecto es mejorar la experiencia del usuario a la hora de efectuar reservas, adquirir abonos, gestionar su contenido y acceder a la información más relevante a la hora de viajar.

Palabras clave

Android, aplicación móvil, billete, abono, autobús, transporte, Firebase, Cloud Firestore

Abstract

The main objective of this Final Degree Project is based on the creation of a mobile application aimed at Android platforms that allows users to manage and purchase tickets and season tickets at any time and in any place.

For this reason, the aim is for the user to have an intuitive and simple tool that makes this acquisition and subsequent management process as easy as possible.

In addition, it is also intended that the most important and relevant information is always accessible to users, and for this reason it will also have a discussion forum, a board with the latest news and promotions and a station search engine.

In summary, the purpose of the project is to improve the user experience when making reservations, purchasing tickets, managing their content and accessing the most relevant information when traveling.

Keywords

Android, mobile application, ticket, transport pass, bus, transport, Firebase, Cloud Firestore

Índice de contenidos

Capítulo 1: INTRODUCCIÓN	15
1.1 Motivación	16
1.2 Objetivos	17
1.3 Alcance	17
1.4 Estructura del documento.....	21
Capítulo 2: ESTADO DEL ARTE	22
2.1 Propuesta RENFE.....	23
Descripción general.....	23
Detalles de aplicación móvil.....	24
2.2 Propuesta ALSA	25
Descripción General	25
Detalles de aplicación móvil.....	25
2.3 Propuesta TRAINLINE	26
Descripción General	26
Detalles de aplicación móvil.....	26
2.4 Propuesta RYANAIR.....	27
Descripción General	27
Detalles de aplicación móvil.....	27
2.5 Propuesta IBERIA.....	28
Descripción General	28
Detalles de aplicación móvil.....	29
2.6 Comparativa de propuestas de mercado vs. Avantis.....	29
Capítulo 3: PLANIFICACIÓN Y PRESUPUESTOS	32
3.1 Modelo de desarrollo	33
3.2 Planificación inicial de proyecto.....	34
3.2.1 Gestión de conflictos temporales	34
3.2.2 Aspectos generales de la planificación inicial	35
3.3 Costes iniciales del proyecto	39
3.3.1 Costes de personal	39
3.3.2 Costes hardware.....	39
3.3.3 Costes software	40
3.3.4 Otros costes.....	40
3.4 Estimaciones iniciales.....	40
3.4.1 Estimación por puntos de función (Método Albretch)	41

3.4.2 Estimación por COCOMO	45
3.4.3 Estimación por casos de uso	47
3.4.4 Comparativa de estimaciones	54
3.5 Presupuestos	54
3.5.1 Presupuesto inicial de proyecto	54
3.5.2 Presupuesto final de proyecto	58
Capítulo 4: ANÁLISIS DEL SISTEMA	61
4.1 Actores del sistema	62
4.2 Requisitos de Usuario	63
4.2.1 Propios de USUARIOS ANÓNIMOS	63
4.2.2 Comunes a usuarios ANÓNIMOS y usuarios AUTENTICADOS	63
4.2.3 Propios de los USUARIOS AUTENTICADOS	64
4.2.4 Propios del usuario ADMINISTRADOR	65
4.2.5 Modelo de Casos de Uso	67
4.2.6 Especificación de Casos de Uso	71
4.3 Reglas de negocio	80
4.4 Requisitos Funcionales	82
4.5 Requisitos No Funcionales	89
4.5.1 Usabilidad	89
4.5.2 Eficiencia	90
4.5.3 Seguridad	90
4.5.4 Disponibilidad	91
4.5.5 Interfaces de comunicación	91
4.5.6 Implementación	92
4.6 Requisitos de Información	92
4.6.1 Modelo Entidad-Relación	93
4.6.2 Diccionario de datos	95
Capítulo 5: DISEÑO DEL SISTEMA	107
5.1 Arquitectura	108
5.1.1 Arquitectura Lógica	108
5.1.2 Arquitectura Física	110
5.2 Modelos de diseño	110
5.2.1 Diagrama de Actividades	111
5.2.2 Diagramas de Secuencia	113
5.2.3 Modelo Lógico de Datos	118

5.2.4 Diagrama de Clases	121
5.3 Diseño de Interfaces.....	121
5.3.1 Tecnologías y elementos utilizados.....	121
5.3.2 Interfaces.....	123
5.3.2.1 Menús.....	123
5.3.2.2 Destinadas a funcionalidad básica	126
5.3.2.3 Destinadas a los sistemas de compra.....	137
5.3.2.4 Destinados a la gestión de información de usuarios	149
5.3.2.5 Propias de administración.....	157
Capítulo 6: IMPLEMENTACIÓN	161
6.1 Requerimientos Hardware y Software	162
6.2 Herramientas utilizadas	162
6.2.1 Herramientas para el desarrollo móvil.....	162
6.2.2 Herramientas de soporte	163
6.3 Tecnologías utilizadas.....	164
6.4 Librerías y dependencias.....	164
6.5 Detalles de implementación	165
6.5.1 Pasos seguidos en la implementación	166
Paso 1: Creación del proyecto dentro de la plataforma Firebase.....	166
Paso 2: Integración del SDK de Cloud Firestore en la aplicación Android	169
Paso 3: Procedimientos de inserción y recuperación de datos	171
Paso 4: Procedimientos de modificación y eliminación de datos	174
Paso 5: Configuración de reglas de seguridad en Cloud Firestore.....	177
6.5.2 Administración de usuarios en Firebase	177
6.5.2.1 Registro de usuarios	177
6.5.2.2 Autenticación de usuarios.....	178
6.5.2.3 Cerrar sesión	179
6.6 Estructura interna de proyecto	179
Capítulo 7: PRUEBAS	183
7.1 Pruebas de Caja Blanca	184
7.2 Pruebas de Caja Negra	185
Capítulo 8: MANUALES.....	194
8.1 Manual de Despliegue.....	195
8.2 Manual de Usuario	195
8.2.1 ¿Empezamos?.....	195

8.2.2 Compra de billetes estándar	196
8.2.3 Compra de abonos	197
8.2.4 Compra de billetes utilizando un abono asociado	197
8.2.5 Preguntas frecuentes	198
8.2.6 Consulta de estaciones.....	198
8.2.7 Consultar información personal.....	199
8.2.8 Método de pago (Tarjeta de crédito).....	199
8.2.9 Consultar mis viajes.....	200
8.2.10 Consultar mis abonos	200
8.2.11 Iniciar sesión.....	201
8.2.12 Registro de nuevo usuario.....	201
8.2.13 Acceso a redes sociales (Twitter)	201
Capítulo 9: CONCLUSIONES Y FUTURAS AMPLIACIONES.....	202
9.1 Conclusiones.....	203
9.2 Futuras Ampliaciones	204
Capítulo 10: BIBLIOGRAFÍA Y REFERENCIAS	206
10.1 Bibliografía	207
10.2 Webgrafía.....	207
10.3 Complementos audiovisuales	209
Anexo I: Firebase, PayPal y Cloud Firestore.....	210
Firebase	211
Descripción General	211
¿Por qué la elección de Firebase en el proyecto?.....	212
Servicios de Firebase utilizados en el proyecto.....	213
PayPal	213
Descripción General	213
Beneficios del uso de PayPal	214
¿Por qué la elección de PayPal en el proyecto?.....	214
Cloud Firestore	215
Descripción General	215
¿Por qué se integra Cloud Firestore al proyecto?.....	215
Anexo II: Casos de Uso	217

Índice de tablas

Tabla 1 Objetivos.....	17
Tabla 2 Comparativa de alternativas a Avantis.....	31
Tabla 3 Etapas de planificación inicial de proyecto	37
Tabla 4 Componentes hardware	39
Tabla 5 Componentes software	40
Tabla 6 Especificación de otros gastos.....	40
Tabla 7 Entradas de usuario	41
Tabla 8 Salidas de usuario	42
Tabla 9 Consultas externas de usuario.....	42
Tabla 10 Ficheros lógicos internos	43
Tabla 11 Ficheros lógicos externos	43
Tabla 12 Correspondencias y puntos de función sin ajustar (PFNA)	43
Tabla 13 Factores de complejidad y puntuación	44
Tabla 14 Relación líneas de código - lenguaje de programación	45
Tabla 15 Aspectos y ponderación de factores de esfuerzo.....	46
Tabla 16 Calificación factores de esfuerzo.....	47
Tabla 17 Propuestas de cálculo por COCOMO	47
Tabla 18 Ponderaciones UAW.....	48
Tabla 19 UAW totales.....	48
Tabla 20 Ponderaciones para el cálculo de UUAW basado en transacciones	48
Tabla 21 Ponderaciones para el cálculo de UUAW basado en clases de análisis	49
Tabla 22 UUAW totales	51
Tabla 23 Factores de complejidad técnica (TCF).....	52
Tabla 24 Factores ambientales (EF)	52
Tabla 25 Relación horas/persona - factor de productividad (FP)	53
Tabla 26 Modelo de esfuerzo total de proyecto.....	53
Tabla 27 Comparativa de estimaciones 1	54
Tabla 28 Comparativa de estimaciones definitiva	54
Tabla 29 Total inicial costes de personal.....	55
Tabla 30 Total inicial costes hardware	56
Tabla 31 Total inicial costes software	56
Tabla 32 Total inicial de otros gastos.....	57
Tabla 33 Presupuesto total inicial	57
Tabla 34 Total real del coste de personal	59
Tabla 35 Total real de los costes hardware.....	59
Tabla 36 Total real de los costes software	60
Tabla 37 Total real de otros gastos	60
Tabla 38 Presupuesto total real (a la finalización de proyecto).....	60
Tabla 39 Descripción de los actores del sistema.....	62
Tabla 40 Requisitos de usuario propios de los usuarios ANÓNIMOS	63
Tabla 41 Requisitos de usuario comunes a usuarios ANÓNIMOS y AUTENTICADOS	64
Tabla 42 Requisitos de usuario propios de los usuarios AUTENTICADOS.....	65
Tabla 43 Requisitos de usuario propios del usuario ADMINISTRADOR	67
Tabla 44 Caso de uso "Iniciar Sesión"	71
Tabla 45 Caso de uso "Registrarse".....	72
Tabla 46 Caso de uso "Consultar información de las estaciones"	73

Tabla 47 Caso de uso "Consultar Horarios Viajes"	74
Tabla 48 Caso de uso "Comprar Billetes"	75
Tabla 49 Caso de uso "Comprar Billetes con Abono"	77
Tabla 50 Caso de uso "Comprar Abono"	78
Tabla 51 Caso de uso "Añadir tema de discusión"	78
Tabla 52 Caso de uso "Añadir Respuesta"	79
Tabla 53 Reglas de negocio	82
Tabla 54 Clasificación requisitos funcionales	89
Tabla 55 Requisitos no Funcionales – Usabilidad	90
Tabla 56 Requisitos no Funcionales – Eficiencia	90
Tabla 57 Requisitos no Funcionales – Seguridad	91
Tabla 58 Requisitos no Funcionales – Disponibilidad	91
Tabla 59 Requisitos no Funcionales - Interfaces de Comunicación	91
Tabla 60 Requisitos no Funcionales - Implementación.....	92
Tabla 61 Requisitos de información	93
Tabla 62 Entidad "Usuario"	95
Tabla 63 Entidad "Usuario Autenticado"	96
Tabla 64 Entidad "Tarjeta de Crédito"	96
Tabla 65 Entidad "Estación"	97
Tabla 66 Entidad "Precio"	98
Tabla 67 Entidad "Trayecto"	98
Tabla 68 Entidad "Reserva"	99
Tabla 69 Entidad "Noticia"	99
Tabla 70 Entidad "Promoción"	100
Tabla 71 Entidad "Tipo de Abono"	101
Tabla 72 Entidad "Abono"	101
Tabla 73 Entidad "Tema de Discusión"	102
Tabla 74 Entidad "Respuesta"	103
Tabla 75 Entidad "Horario"	103
Tabla 76 Entidad "Vehículo"	103
Tabla 77 Requisitos mínimos.....	162
Tabla 78 Servicios ofertados por Firebase para el proyecto	168
Tabla 79 Prueba de caja negra "Registrar Usuario"	185
Tabla 80 Prueba de caja negra "Iniciar Sesión"	186
Tabla 81 Prueba de caja negra "Seleccionar parámetros de viaje"	186
Tabla 82 Prueba de caja negra "Información del Pasajero"	187
Tabla 83 Prueba de caja negra "Selección de plaza"	187
Tabla 84 Prueba de caja negra "selección de trayecto (compra de abono)"	187
Tabla 85 Prueba de caja negra "Selección parámetros de compra (compra de billetes con abono)"	188
Tabla 86 Prueba de caja negra "Buscador de estaciones"	188
Tabla 87 Prueba de caja negra "Modificar perfil de usuario"	188
Tabla 88 Prueba de caja negra "Agregar nueva tarjeta de crédito"	189
Tabla 89 Prueba de caja negra "Eliminar billete"	189
Tabla 90 Prueba de caja negra "Descargar billete"	190
Tabla 91 Prueba de caja negra "Crear nuevo tema de discusión "	190
Tabla 92 Prueba de caja negra "Crear nueva respuesta"	190

Tabla 93 Prueba de caja negra "Acceso a redes sociales"	191
Tabla 94 Prueba de caja negra "Crear tipo de abono"	191
Tabla 95 Prueba de caja negra "Eliminar Noticia"	192
Tabla 96 Prueba de caja negra "Modificar promoción"	192
Tabla 97 Prueba de caja negra "Añadir vehículo"	192
Tabla 98 Prueba de caja negra "Cerrar Sesión"	193
Tabla 99 Características principales de Cloud Firestore	216
Tabla 100 Caso de uso "Consultar Noticias"	218
Tabla 101 Caso de uso "Mostrar detalles de noticia"	219
Tabla 102 Caso de uso "Consultar promociones"	220
Tabla 103 Caso de uso "Mostrar detalles de promoción"	220
Tabla 104 Caso de uso "Buscar estación"	221
Tabla 105 Caso de uso "Mostrar detalles de estación"	222
Tabla 106 Caso de uso "Acceso a redes sociales"	222
Tabla 107 Caso de uso "Consultar foro de sugerencias"	223
Tabla 108 Caso de uso "Mostrar detalles de tema de discusión"	224
Tabla 109 Caso de uso "Modificar plaza"	225
Tabla 110 Caso de uso "Realizar Pago"	226
Tabla 111 Caso de uso "Descargar billete en PDF"	226
Tabla 112 Caso de uso "Enviar billete por correo"	227
Tabla 113 Caso de uso "Mostrar manual de usuario"	228
Tabla 114 Caso de uso "Cerrar sesión"	229
Tabla 115 Caso de uso "Consultar perfil de usuario"	229
Tabla 116 Caso de uso "Modificar perfil de usuario"	230
Tabla 117 Caso de uso "Darse de baja"	231
Tabla 118 Caso de uso "Consultar tarjeta de crédito asociada"	231
Tabla 119 Caso de uso "Añadir tarjeta de crédito"	232
Tabla 120 Caso de uso "Eliminar tarjeta de crédito"	233
Tabla 121 Caso de uso "Consultar billetes"	234
Tabla 122 Caso de uso "Mostrar detalles de billete"	235
Tabla 123 Caso de uso "Anular billete"	236
Tabla 124 Caso de uso "Filtrar Reservas"	236
Tabla 125 Caso de uso "Consultar abonos"	237
Tabla 126 Caso de uso "Mostrar detalles de abono"	238
Tabla 127 Caso de uso "Mostrar estaciones"	239
Tabla 128 Caso de uso "Mostrar precios de trayectos"	240
Tabla 129 Caso de uso "Mostrar viajes/trayectos"	241
Tabla 130 Caso de uso "Mostrar tipos de abono"	242
Tabla 131 Caso de uso "Mostrar noticias"	242
Tabla 132 Caso de uso "Mostrar promociones"	243
Tabla 133 Caso de uso "Mostrar reservas"	244
Tabla 134 Caso de uso "Mostrar horarios"	245
Tabla 135 Caso de uso "Mostrar vehículos"	246
Tabla 136 Caso de uso "Crear estación"	247
Tabla 137 Caso de uso "Crear precio de trayecto"	247
Tabla 138 Caso de uso "Crear viaje/trayecto"	248
Tabla 139 Caso de uso "Crear tipo de abono"	249

Tabla 140 Caso de uso "Crear noticia"	250
Tabla 141 Caso de uso "Crear promoción"	250
Tabla 142 Caso de uso "Crear horario"	251
Tabla 143 Caso de uso "Crear vehículo"	252
Tabla 144 Caso de uso "Modificar estación"	253
Tabla 145 Caso de uso "Modificar precio de trayecto"	253
Tabla 146 Caso de uso "Modificar viaje/trayecto"	254
Tabla 147 Caso de uso "Modificar tipo de abono"	255
Tabla 148 Caso de uso "Modificar noticia"	256
Tabla 149 Caso de uso "Modificar promoción"	257
Tabla 150 Caso de uso "Modificar horario"	257
Tabla 151 Caso de uso "Modificar vehículo"	258
Tabla 152 Caso de uso "Eliminar estación"	259
Tabla 153 Caso de uso "Eliminar precio de trayecto"	260
Tabla 154 Caso de uso "Eliminar viaje/trayecto"	260
Tabla 155 Caso de uso "Eliminar tipo de abono"	261
Tabla 156 Caso de uso "Eliminar noticia"	262
Tabla 157 Caso de uso "Eliminar promoción"	262
Tabla 158 Caso de uso "Eliminar horario"	263
Tabla 159 Caso de uso "Eliminar vehículo"	264

Índice de figuras

Figura 1	Árbol de Características.....	19
Figura 2	Logotipo Renfe	23
Figura 3	Logotipo Alsa	25
Figura 4	Logotipo TrainLine.....	26
Figura 5	Logotipo RyanAir	27
Figura 6	Logotipo Iberia	28
Figura 7	Modelo de desarrollo iterativo-incremental.....	33
Figura 8	Festividades en el calendario de planificación.....	36
Figura 9	Planificación inicial de proyecto (Diagrama de Gantt).....	38
Figura 10	Actores del sistema	62
Figura 11	Diagrama de casos de uso de usuario ANÓNIMO	68
Figura 12	Diagrama de casos de uso de usuario AUTENTICADO	69
Figura 13	Diagrama de casos de uso del usuario ADMINISTRADOR	70
Figura 14	Modelo entidad-relación.....	94
Figura 15	Arquitectura lógica	108
Figura 16	Patrón MVC (Modelo-Vista-Controlador)	109
Figura 17	Arquitectura física	110
Figura 18	Diagrama de actividades "Compra de Billetes sin Abono"	111
Figura 19	Diagrama de actividades "Compra de Billetes con Abono"	112
Figura 20	Diagrama de actividades "Compra de Abonos"	113
Figura 21	Diagrama de secuencia "Iniciar Sesión"	114
Figura 22	Diagrama de secuencia "Registro de usuario"	115
Figura 23	Diagrama de secuencia "Consultar Información de Estación"	115
Figura 24	Diagrama de secuencia "Crear nuevo tema de discusión"	116
Figura 25	Diagrama de secuencia "Crear nueva respuesta.....	117
Figura 26	Estructura jerárquica de datos en Cloud Firestore	118
Figura 27	Estructura de documento en Cloud Firestore	119
Figura 28	Sub-colecciones en Cloud Firestore	119
Figura 29	Relación modelos de datos (código) - modelos de datos (Cloud Firestore)	119
Figura 30	Modelo lógico de datos 1	120
Figura 31	Modelo lógico de datos 2	121
Figura 32	Interfaz "Pantalla Principal"	123
Figura 33	Interfaz "Menú lateral (usuario no autenticado)"	124
Figura 34	Interfaz "Menú lateral (Administrador)"	124
Figura 35	Interfaz "Menú lateral (usuario autenticado)"	124
Figura 36	Interfaz "Gestión de Información Personal"	125
Figura 37	Interfaz "Gestión de Sistema de Compra"	126
Figura 38	Interfaz "Inicio de Sesión"	127
Figura 39	Interfaz "Registro de usuario"	128
Figura 40	Interfaz "Home / Pantalla principal"	129
Figura 41	Interfaz "Acerca de Avantis"	129
Figura 42	Interfaz "Foro de discusión"	130
Figura 43	Interfaz "Crear nuevo tema de discusión"	131
Figura 44	Interfaz "Detalles de tema de discusión"	132
Figura 45	Interfaz "Crear nueva respuesta"	133
Figura 46	Interfaz "Consulta de estaciones"	134

Figura 47 Interfaz "Detalles de estación"	135
Figura 48 Interfaz "Mensajes de solicitud de confirmación"	135
Figura 49 Interfaz "Mensaje de información"	136
Figura 50 Interfaz "Manual de usuario 1"	137
Figura 51 Interfaz "Manual de usuario 2"	137
Figura 52 Interfaz "Selector parámetros de viaje (Compra sin Abono)"	138
Figura 53 Interfaz "Resultado Búsqueda de Viaje"	139
Figura 54 Interfaz "Información de los Pasajeros (Compra sin Abono)"	140
Figura 55 Interfaz "Servicios Adicionales de Viaje"	141
Figura 56 Interfaz "Selección de Asiento"	142
Figura 57 Interfaz "Resumen de Viaje"	143
Figura 58 Interfaz "Finalización Compra Billetes"	144
Figura 59 Interfaz "Selector tipo de abono"	145
Figura 60 Interfaz "Selector parámetros de Abono"	146
Figura 61 Interfaz "Finalización Compra Abono"	147
Figura 62 Interfaz "Selector parámetros de viaje (Compra con Abono)"	148
Figura 63 Interfaz "Selector sentido de viaje (Compra con Abono)"	149
Figura 64 Interfaz "Mi Perfil"	150
Figura 65 Interfaz "Modificar Perfil de Usuario"	151
Figura 66 Interfaz "No existe Tarjeta de Crédito asociada"	152
Figura 67 Interfaz "Ver tarjeta de crédito asociada"	152
Figura 68 Interfaz "Añadir Tarjeta de Crédito"	153
Figura 69 Interfaz "Mis Viajes"	154
Figura 70 Interfaz "Detalles Billete"	155
Figura 71 Interfaz "Mis Abonos"	156
Figura 72 Interfaz "Detalles Abono"	157
Figura 73 Interfaz "Consultar contenido (Vehículos)"	158
Figura 74 Interfaz "Consultar contenido (Reservas)"	158
Figura 75 Interfaz "Consultar contenido (Precios)"	158
Figura 76 Proyectos asignados a la cuenta de desarrollo	167
Figura 77 Creación de proyecto Firebase 1	167
Figura 78 Creación de proyecto Firebase 2	167
Figura 79 Integración Cloud Firestore a proyecto paso 1	169
Figura 80 Integración Cloud Firestore a proyecto paso 2	169
Figura 81 Integración Cloud Firestore a proyecto paso 3	170
Figura 82 Integración Cloud Firestore a proyecto paso 4.1	170
Figura 83 Integración Cloud Firestore a proyecto paso 4.2	171
Figura 84 Crear instancia Cloud Firestore (código)	171
Figura 85 instancia Objeto "DatosEstación"	172
Figura 86 Método de inserción de contenido en Cloud Firestore (código)	172
Figura 87 Método de obtención de datos Cloud Firestore 1	173
Figura 88 Método de obtención de datos Cloud Firestore 2	173
Figura 89 Método de obtención de datos Cloud Firestore 3	173
Figura 90 Método modificación completa en Cloud Firestore (código)	175
Figura 91 Método modificación parcial en Cloud Firestore (código)	175
Figura 92 Método de eliminación de documento en Cloud Firestore (código)	176
Figura 93 Crear instancia Firebase Authentication (código)	177

Figura 94 Método de registro de usuarios en Firebase Auth (código).....	177
Figura 95 Método de inicio de sesión a través de Firebase Auth (código)	178
Figura 96 Método de logout en Firebase Auth (código)	179
Figura 97 Pruebas de caja blanca	184
Figura 98 Pruebas de caja negra	185
Figura 99 Logotipo Firebase	211
Figura 100 Servicios proporcionados por Firebase	212
Figura 101 Logotipo PayPal	213
Figura 102 Logotipo Cloud Firestore	215



Capítulo 1: INTRODUCCIÓN



1.1 Motivación

Segovia se encuentra en una posición estratégica dentro del marco geográfico de la Península Ibérica. Al norte, a unos escasos 100 km nos encontramos con la ciudad de Valladolid, motor económico y capital de la comunidad autónoma de Castilla y León. Dirección sur se encuentra uno de los factores más importantes para tener en cuenta en este desarrollo: Madrid.

Debido a la oferta económica menos competitiva en la provincia de Segovia, muchas personas deciden buscar empleo y desarrollar su actividad laboral fuera del territorio, aunque mantengan sus hogares y servicios, convirtiendo a Segovia en una ciudad dormitorio para muchos de sus habitantes.

Por otro lado, la comunidad autónoma de Castilla y León cuenta con una población rural muy elevada, a lo que se suman unos niveles de envejecimiento altos. Esto provoca que muchas personas que viven en las zonas rurales se encuentren aisladas o no encuentren facilidades a la hora de acceder a servicios de transporte que les permitan cumplir con sus necesidades. Un ejemplo claro de esto es acudir a ciertos servicios que se desarrollan en Segovia capital como puede ser acudir al hospital, realizar gestiones personales en servicios públicos o visitar a sus familiares. A esto se suma la incapacidad para muchas zonas rurales de reservar viajes con antelación, puesto que en muchos no existe una estación donde adquirir los mismos, teniendo que realizar esto en el propio medio de transporte el mismo día del viaje.

Otro factor importante que se tiene en cuenta es la gran riqueza patrimonial y oferta gastronómica que ofrece la provincia de Segovia, lo que provoca que el sector turístico sea uno de los principales motores económicos con una gran afluencia de turistas a lo largo de todo el año, a los cuales les sería de gran utilidad contar con una herramienta capaz de facilitarles la planificación de sus rutas.

Muchas de estas compañías y servicios de transporte público que funcionan en nuestra provincia cuentan con una plataforma web que permite realizar la compra de billetes o consultar información, pero muchas de ellas no cuentan con una aplicación móvil que permita realizar este proceso en cualquier momento y en cualquier lugar con la única condición de contar con un smartphone con conexión a Internet. Esto también provoca que el número de plataformas aumente y genere cierto grado de conflicto entre las mismas.

Por estos motivos descritos anteriormente se ha propuesto la idea de crear una aplicación móvil que permite centralizar todos los servicios de transporte público de la provincia y que facilite a cualquier persona que lo desee la posibilidad de moverse y disfrutar de Segovia y alrededores (principalmente Valladolid y Madrid) reservando los viajes con antelación, sin necesidad de recurrir a un vehículo privado, aprovechando al máximo las ofertas y promociones que se encuentran disponibles junto con la reducción de los niveles de contaminación gracias al aumento y fomento del transporte público.

1.2 Objetivos

El **objetivo principal** que se quiere lograr en este proyecto es el desarrollo una aplicación Android que facilite y agilice a los usuarios la consulta de información y horarios referentes a un viaje y posteriormente, si el usuario lo solicita, reservar plazas en ese viaje.

Una vez planteado el objetivo principal que debe cumplir la aplicación, esta también cuenta con una serie de objetivos derivados que complementan el objetivo principal y proporcionan una riqueza y una funcionalidad muy interesante para el usuario final. Estos objetivos son:

OBJ-02	Reducir la necesidad de las personas de acudir al punto de venta físico para adquirir sus billetes/abonos.
OBJ-03	Centralizar el proceso de compra de billetes/reservas propuestas por las distintas compañías de transporte.
OBJ-04	Facilitar y reducir el tiempo necesario para la gestión de los abonos a los usuarios.
OBJ-05	Agilizar el acceso a promociones y noticias a los usuarios de la aplicación.
OBJ-06	Mejorar la comunicación entre los usuarios de la aplicación y las compañías de servicios de transporte público.
OBJ-07	Fomentar y aumentar el uso de transporte público en detrimento del vehículo privado.
OBJ-08	Fomentar y aumentar el turismo dentro de la provincia de Segovia.
OBJ-09	Evitar el aislamiento de la población rural.
OBJ-10	Reducir el consumo de celulosa (papel, cartulina) en la extensión de billetes/reservas.

Tabla 1 Objetivos

Por otro lado, la aplicación cuenta con una parte destinada a la administración del sistema, cuyo objetivo principal es otorgar al administrador la capacidad de gestionar toda la información y contenido ofertado a los usuarios finales con el fin de contar con un sistema actualizado y completamente funcional en cualquier momento.

1.3 Alcance

El proyecto está pensado para abarcar a todos los usuarios que realizan un viaje esporádico o que viajan con frecuencia y les gustaría poder gestionar estos viajes o estos abonos desde una plataforma capaz de funcionar en cualquier dispositivo móvil Android que cuente con una conexión a Internet sin necesidad de tener que acudir a una oficina o punto de venta físico para llevar a cabo cualquier tipo de consulta, modificación o anulación de estos.

También está pensada para que todos los usuarios puedan compartir sus inquietudes, dudas o sugerencias dentro de la comunidad de usuarios de Avantis, puesto que queda demostrado que no hay mejor retroalimentación que la ofrecida por los propios usuarios de la plataforma y es una herramienta que permite un aprendizaje rápido e intuitivo.

Por último, la plataforma cuenta con perfiles en redes sociales, dirigidos a ese tipo de usuarios jóvenes y no tan jóvenes influenciados en los últimos años por estas nuevas tecnologías, otorgando así a la plataforma un vehículo de comunicación e información excepcional en este siglo XXI.

Este proyecto está pensado, en un principio, para funcionar en cualquier dispositivo que funcione con el sistema operativo Android y se encargará de cumplir con los servicios y funcionalidades previamente descritas en los puntos anteriores. Por supuesto la idea del proyecto va mucho más allá, y en un futuro la aplicación podría soportar la planificación de viajes con transbordos, la localización a tiempo real de los vehículos en un trayecto, información a tiempo real de las estaciones o sugerencias de viajes.

Aun así, por el momento, el sistema va a contar con una serie de características principales que se representan a través del siguiente diagrama:

Como se puede observar en la figura 1, se distinguen 2 escenarios distintos. En la parte superior se encuentran definidas las características principales relacionadas con los usuarios comunes de la aplicación agrupadas en los siguientes módulos:

- **“Gestión de Usuarios”**: engloba las características y funcionalidades referentes a los usuarios y la gestión de los perfiles de estos dentro de la plataforma.
- **“Sistema de Compra Billetes”**: una de las características más importantes de la plataforma que se encarga de englobar las funcionalidades y flujos relacionados con la expedición de nuevas reservas/billetes por parte de los usuarios.
- **“Sistema de Compra Abonos”**: cuenta con una importancia similar a la anterior y es el encargado de englobar la funcionalidad y características necesarias para la expedición de abonos por parte de los usuarios.
- **“Gestión de Foro de la comunidad”**: engloba las características y funcionalidades referentes a la participación y retroalimentación de los usuarios dentro de la plataforma, junto con la comunicación a través de redes sociales.
- **“Explotación”**: engloba las funcionalidades y características referentes a la consulta y sus operaciones derivadas sobre la información dentro de la plataforma.

Por otro lado, se definen todas las características básicas destinadas a la parte administrativa de la aplicación, de las cuales se encarga el administrador del sistema. Se encuentran agrupadas principalmente en los siguientes módulos:

- **“Gestión de Estaciones”**: engloba la funcionalidad y las características necesarias para la visualización, creación, modificación y eliminación de las estaciones.
- **“Gestión de Precios”**: engloba la funcionalidad y las características necesarias para la visualización, creación, modificación y eliminación de los precios de billete para un trayecto determinado.
- **“Gestión de Trayectos”**: engloba la funcionalidad y las características necesarias para la visualización, creación, modificación y eliminación de los viajes disponibles en la plataforma.
- **“Gestión de Tipos de Abono”**: engloba la funcionalidad y las características necesarias para la visualización, creación, modificación y eliminación de los tipos de abono disponibles para adquirir por parte de los usuarios de la plataforma.
- **“Gestión de Promociones”**: engloba la funcionalidad y las características necesarias para la visualización, creación, modificación y eliminación de las promociones disponibles para los usuarios de la plataforma.
- **“Gestión de Noticias”**: engloba la funcionalidad y las características necesarias para la visualización, creación, modificación y eliminación de noticias y contenido relevante para el usuario de la plataforma.
- **“Gestión de Horarios”**: engloba la funcionalidad y las características necesarias para la visualización, creación, modificación y eliminación de horarios.
- **“Gestión de Vehículos”**: engloba la funcionalidad y las características necesarias para la visualización, creación, modificación y eliminación de vehículos.

1.4 Estructura del documento

Este epígrafe estará destinado a describir detalladamente la estructura principal de este documento. La memoria estará compuesta por 9 capítulos, referencias utilizadas en el desarrollo del proyecto y 2 anexos. Para comprender mejor el contenido de estas, a continuación, se resume su contenido:

- **Capítulo 1- Introducción:** Capítulo donde se presentan los aspectos generales del proyecto como son los objetivos que se buscan cumplir, el alcance, la motivación y las características principales de la herramienta.
- **Capítulo 2 - Estado del Arte:** Capítulo destinado 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.
- **Capítulo 3 - Planificación y presupuestos:** Capítulo destinado al análisis de la metodología de trabajo utilizada en el desarrollo, sumado a la planificación temporal y estimación de costes requeridos por el proyecto.
- **Capítulo 4 - Análisis de Sistema:** Capítulo destinado a describir detalladamente los requisitos propuestos por el usuario correspondientes a la funcionalidad de la herramienta, junto con la descripción detallada de los actores participantes de la misma.
- **Capítulo 5 - Diseño de Sistema:** Capítulo destinado a la descripción de la arquitectura física y lógica necesaria para la correcta puesta en marcha de la herramienta junto con el análisis de la experiencia de usuario e interfaces de usuario utilizadas.
- **Capítulo 6 - Implementación:** Capítulo destinado a analizar aspectos técnicos del proyecto, como son las herramientas, librerías externas y software de terceros utilizados en el desarrollo de la herramienta.
- **Capítulo 7 - Pruebas:** Capítulo destinado a recoger y analizar las pruebas y trabajos de “testeo” realizados sobre la herramienta.
- **Capítulo 8 - Manuales:** Capítulo donde se exponen las guías y manuales de usuario necesarios para la instalación y el aprendizaje y posterior manejo de todos los aspectos esenciales de la herramienta desarrollada.
- **Capítulo 9 - Conclusiones y posibles mejoras:** Capítulo donde se recoge la valoración general final del desarrollo completo de la herramienta, 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 de documentación externas utilizadas en el desarrollo de la herramienta (referencias físicas y referencias web).
- **Anexo I:** Epígrafe destinado a explicar detalladamente servicios utilizados en el desarrollo del proyecto (Firebase, PayPal y Cloud Firestore).
- **Anexo II:** Epígrafe destinado a completar la especificación de los casos de uso.



Capítulo 2: ESTADO DEL ARTE



Antes de desarrollar la aplicación móvil es necesario conocer y analizar las propuestas similares de herramientas ya existentes en el mercado actual, las cuales sirven para inspeccionar y marcar en cierto modo el dominio de nuestro proyecto, es decir, analizar los aspectos relevantes y características esenciales que no pueden faltar en nuestro proyecto, junto con aspectos nuevos que marquen las diferencias de nuestra herramienta frente a las demás propuestas.

Para comenzar, se ha analizado el entorno de Segovia en busca de aplicaciones que permitan la reserva de viajes y compra de abonos, es decir, que pertenezcan a un marco similar al proyecto propuesto. Después de investigar posibles soluciones, el resultado ha sido bastante pobre, ya que la única empresa que cuenta con una aplicación móvil capaz de cumplir con esta funcionalidad es Renfe. Por otro lado, Grupo Avanza no cuenta con una aplicación móvil operativa, pero a su vez sí que cuenta con un portal web donde se pueden realizar compras y gestiones de billetes y bonos virtuales. Por último, se encuentra Linecar, la cual no cuentan con ningún tipo de aplicación móvil y su portal web solo es una mera plataforma de información que no permite realizar gestiones ni compras de ningún tipo.

Por este motivo se ha tenido que recurrir a investigar otras plataformas además de Renfe que, aunque no funcionen en el ámbito de Segovia, ofrecen servicios y funcionalidades muy semejantes a las propuestas para este proyecto.

2.1 Propuesta RENFE



Figura 2 Logotipo Renfe

Descripción general

RENFE, más concretamente Renfe Operadora es la principal compañía ferroviaria de España y la que se proporciona, de manera unánime, servicios de transporte de viajeros por toda la red ferroviaria del país. Además, también es una de las compañías que se encargan de gestionar y proporcionar el servicio de transporte de mercancías por la misma red.

Fue fundada en 1941, aunque en 2005 la desaparición de esta dio lugar a la división de los servicios en dos empresas: ADIF “Administrador de Infraestructuras Ferroviarias” y la presente Renfe Operadora, la cual se dirige a la gestión de los servicios ferroviarios en

particular. En los años siguientes se fueron incorporando poco a poco los servicios de transporte de mercancías, viajeros y la infraestructura de vía estrecha gracias a la desaparición de la empresa propietaria de estos (FEVE). En 2014, y gracias a la decisión tomada por el Ministerio de Interior, el grupo empresarial Renfe es el encargado de gestionar todos estos servicios previamente expuestos a través de cuatro ramas/sociedades anónimas como son: Renfe Viajeros, Renfe Mercancías, Renfe Integria y Renfe Alquiler de Material Ferroviario.

Por último, cabe destacar los principales servicios ofertados a los viajeros, los cuales se clasifican en: Larga Distancia, Media Distancia (o “Regionales”), Cercanías y Trenes Turísticos de Lujó.

Detalles de aplicación móvil

Esta aplicación cuenta con una interfaz bastante intuitiva en un primer vistazo. Cuenta con un menú contextual de ayuda muy útil para que el usuario en los principales escenarios que se van presentando. Es posible utilizar la misma tanto como un usuario anónimo como estando autenticado. Permite también el registro de nuevos usuarios. Como funcionalidad principal, la aplicación de Renfe cumple los siguientes aspectos:

- **Todos los usuarios**
 - Compra de Billetes / Consulta Horarios.
 - Planificación de Viajes (Esta funcionalidad redirecciona al usuario al portal web de la compañía, la funcionalidad no se desarrolla en la aplicación móvil).
 - Puntualidad de los trenes a través del identificador de la estación, del billete o del propio tren.
 - Configuración (selección de idioma, modo offline, activación del menú contextual de ayuda...).
- **Usuario Autenticado**
 - Cambio de Billetes.
 - Anulación de Billetes.
 - Asociar un billete físico.
 - Asociar Tarjeta de Crédito / Débito.
 - Consultar Mis Viajes.
 - Consultar Mis Abonos.
 - Asociar abono físico.
 - Compra de Billetes con Abono.
 - Gestión Tarjeta +Renfe (consulta de puntos acumulados).

La compra de billetes permite además la selección de tarifas para los distintos viajes y pasajeros, el billete es personalizado (introducción de los datos de pasajeros obligatoria), posibilidad de modificar las plazas del viaje y permite el pago a través de tarjeta de crédito o PayPal.

2.2 Propuesta ALSA



Figura 3 Logotipo Alsa

Descripción General

ALSA Grupo S.L.U es una multinacional española fundada en 1923 en Asturias que se encarga de proporcionar servicios de transporte de pasajeros en carretera. En 2005 fue adquirida por el grupo empresarial británico *National Express*.

Los servicios de transporte ofrecidos por esta compañía se pueden clasificar en: Autobuses Urbanos y Metropolitanos, Autobuses Regionales, Nacionales, Internacionales (servicios que operan en los propios países donde la compañía tiene concesiones) y servicios de Ferrocarril (como colaborador).

Detalles de aplicación móvil

Esta aplicación cuenta con una interfaz bastante intuitiva en un primer vistazo. Es posible utilizar la misma tanto como un usuario anónimo como estando autenticado. Permite también el registro de nuevos usuarios. Como funcionalidad principal, la aplicación de Alsa cumple los siguientes aspectos:

- **Todos los Usuarios**
 - Compra Billetes / Consulta de Horarios.
 - Consulta de Pasajeros Frecuentes.
 - Consultar Mis Viajes (previamente asociando un billete físico a la aplicación).
 - Consultar información de contacto y sobre la compañía: Esta funcionalidad redirecciona al usuario al portal web de la compañía, la funcionalidad no se desarrolla en la app móvil.
 - Localización de autobús: Esta funcionalidad redirecciona al usuario al portal web de la compañía, la funcionalidad no se desarrolla en la aplicación móvil.
- **Usuario Autenticado**
 - Gestión de Perfil de Usuario
 - Modificación datos del titular, datos de contacto, información personal y preferencias de viaje.

- Consulta de documentaciones, políticas y reglamentos de Alsa.
- Consultar Mis Viajes (billetes físicos asociados y digitales).
- Consultar Mis Bonos: Esta funcionalidad redirige al usuario al portal web de la compañía, la funcionalidad no se desarrolla en la aplicación móvil.

La compra de billetes permite filtrar la búsqueda de viajes, visualizar detalles del viaje seleccionado (número de paradas y recorrido en mapa), seleccionar tarifas para los distintos viajes y pasajeros, el billete es personalizado (introducción de los datos de pasajeros obligatoria), posibilidad de modificar las plazas del viaje, posibilidad de incorporar suplementos adicionales (mascotas, material deportivo, seguro de viaje), permite el pago a través de tarjeta de crédito, PayPal o puntos acumulados por el usuario y el envío de billete a través del correo electrónico o un SMS (mensaje de texto).

2.3 Propuesta TRAINLINE



Figura 4 Logotipo TrainLine

Descripción General

Trainline Europe es el principal comparador de precios de billetes de tren y autobús de Europa. Además, es líder en el sector de compra y venta de billetes gracias a su plataforma móvil, portal web y su amplia variedad de cupones y tarjetas de fidelización que proporciona a los compradores.

Originalmente este servicio era proporcionado por la empresa *Captain Train*, pero fue adquirida por la compañía Trainline en 2016.

Actualmente cuenta con la capacidad de vender billetes para más de 180 empresas ferroviarias y de transporte por carretera (autobús).

Detalles de aplicación móvil

Esta aplicación cuenta con una interfaz bastante intuitiva en un primer vistazo. Es posible utilizar la misma tanto como un usuario anónimo como estando autenticado. Permite también el registro de nuevos usuarios. Como funcionalidad principal, la aplicación de Trainline cumple los siguientes aspectos:

- **Todos los Usuarios**
 - Compra Billetes / Consulta de Horarios.
 - Consultar datos en tiempo real (solo trenes).

- Paneles de salidas/llegadas de la estación.
- Identificador de tren / billete.
- Consultar Mis Billetes.
- Modificar Billeto.
- Anular Billeto.
- Configuración de la aplicación: consulta de políticas, condiciones, información de contacto / preguntas frecuentes, selección de divisa.
- **Usuario Autenticado**
 - Gestión de Perfil de Usuario: solo permite la administración de los métodos de pago de la cuenta de usuario.

La compra de billetes permite visualizar detalles de los viajes (localización de estaciones/paradas en mapa), seleccionar tarifas para los distintos viajes y pasajeros, el billete es personalizado (introducción de los datos de pasajeros obligatoria), posibilidad de seleccionar la preferencia de asientos del viaje (pasillo o ventanilla), permite el pago a través de tarjeta de crédito o PayPal y el envío de billete a través del correo electrónico.

2.4 Propuesta RYANAIR



Figura 5 Logotipo RyanAir

Descripción General

Ryanair Plc. es una de las principales aerolíneas de bajo coste a nivel mundial. Fue fundada en 1987 por Tony Ryan (empresario irlandés).

A pesar de las numerosas controversias y problemas legislativos que esta aerolínea ha generado en nuestro país, esta cuenta con más de 2000 rutas y 84 bases repartidas por todo el mundo, aunque principalmente se encuentran repartidas por Europa.

Detalles de aplicación móvil

Esta aplicación cuenta con una interfaz bastante compleja en un primer vistazo debido a que hay múltiples formas de acceder a las mismas funcionalidades, lo que complica al usuario el uso de la herramienta. Es posible utilizar la misma tanto como un usuario anónimo como estando autenticado. Permite también el registro de nuevos usuarios. Como funcionalidad principal, la aplicación de Ryanair cumple los siguientes aspectos:

- **Todos los Usuarios**
 - Compra Billetes / Consulta de Horarios.
 - Consultar / Asociar Mis Viajes.

- Realizar Facturación de la reserva.
- Consultar Tarjetas de Embarque.
- Consultar productos a bordo de los vuelos.
- Consultar información de un vuelo.
- Consultar políticas, condiciones y centro de ayuda: Esta funcionalidad redirecciona al usuario al portal web de la compañía, la funcionalidad no se desarrolla en la aplicación móvil.
- Aparcamiento en Aeropuerto, alquiler de coches y hoteles: Esta funcionalidad redirecciona al usuario al portal web de la compañía, la funcionalidad no se desarrolla en la aplicación móvil.
- **Usuario Autenticado**
 - Gestión de Perfil de Usuario:
 - Añadir / Modificar datos personales.
 - Gestionar métodos de pago.
 - Gestionar compañeros de viaje.

La compra de billetes permite visualizar detalles de los vuelos, seleccionar tarifas y descuentos para los distintos vuelos y pasajeros, el billete es personalizado (introducción de los datos de pasajeros obligatoria), posibilidad de seleccionar los asientos del viaje, posibilidad de incorporar servicios adicionales (tipo de equipaje de pasajero, seguro de viaje, coches de alquiler, hoteles) y el pago a través de tarjeta de crédito o PayPal.

Al no realizar el pago de ninguna reserva a través de esta aplicación, no se ha podido comprobar si el usuario tiene la posibilidad de contar con sus reservas para descargar o enviar por correo electrónico (siendo estos los dos sistemas más utilizados por otras herramientas utilizadas).

2.5 Propuesta IBERIA



Figura 6 Logotipo Iberia

Descripción General

Iberia, actualmente conocida como Iberia Líneas Aéreas de España, S. A. Operadora Unipersonal, fundada en 1927, es la principal aerolínea española y una de las más importantes y antiguas del mundo.

En 2011 se fusionó con la compañía *British Airways*, dando así lugar a un “holding” empresarial conocido como IAG “*International Airlines Group*”, convirtiendo a esta fusión en una de las compañías más fuertes del mercado mundial.

Actualmente el grupo Iberia opera 47 países distintos, a los cuales hay que sumar otros 50 países más que cuentan con código compartido con otras compañías aéreas.

Detalles de aplicación móvil

Esta aplicación cuenta con una interfaz bastante sencilla e intuitiva en un primer vistazo para el usuario. Es posible utilizar la misma tanto como un usuario anónimo como estando autenticado. Permite también el registro de nuevos usuarios. Como funcionalidad principal, la aplicación de Iberia cumple los siguientes aspectos:

- **Todos los Usuarios**
 - Compra Billetes / Consulta de Horarios.
 - Consultar / Asociar Mis Viajes.
 - Modificar Reserva.
 - Anular Reserva.
 - Realizar *Check-in* de la reserva / Consultar Tarjetas de Embarque.
 - Consultar información de un vuelo.
 - Acceso a prensa digital: solo se puede acceder a este servicio a través de reservas que cuenten con este servicio.
 - *IBot*: asistente virtual para ayudar al usuario a realizar cualquier operación o consulta dentro de la aplicación.
 - Hoteles, coches, inspire, *bag on board* (facturación de equipaje): Esta funcionalidad redirecciona al usuario al portal web de la compañía, la funcionalidad no se desarrolla en la app móvil.
- **Usuario Autenticado**
 - Gestión de Perfil de Usuario:
 - Visualización Tarjeta de Usuario (+ código QR de esta).
 - Añadir / Modificar datos de usuario.
 - Gestionar documentos de viaje: documentación adicional utilizada para viajar.
 - Gestionar métodos de pago.
 - Gestionar pasajeros frecuentes.
 - Gestionar suscripciones y servicios de la cuenta.
 - Información de contacto y atención al cliente.

La compra de billetes permite visualizar detalles de los vuelos, seleccionar tarifas y descuentos para los distintos vuelos y pasajeros, el billete es personalizado (introducción de los datos de pasajeros obligatoria), posibilidad de seleccionar los asientos del viaje, posibilidad de incorporar servicios adicionales (tipo de equipaje de pasajero, seguro de viaje) y el pago a través de tarjeta de crédito.

Al no realizar el pago de ninguna reserva a través de esta aplicación, no se ha podido comprobar si el usuario tiene la posibilidad de contar con sus reservas para descargar o enviar por correo electrónico (siendo estos los dos sistemas más utilizados por otras herramientas utilizadas).

2.6 Comparativa de propuestas de mercado vs. Avantis

Por último y una vez analizadas y estudiadas las funcionalidades que estas aplicaciones proporcionan a los usuarios/compradores, es necesario valorar y comparar cuántas de

estas estarán presentes y totalmente operativas para los usuarios en esta primera versión del proyecto. En la siguiente tabla se puede observar esta comparativa:

	Renfe	Alsa	Trainline	Ryanair	Iberia	AVANTIS
Compra Billetes	✓	✓	✓	✓	✓	✓
Selector Tarifas y Promociones	✓	✓	✓	✓	✓	✓
Selector Asiento	✓	✓	✓	✓	✓	✓
Selector Servicios Adicionales	✗	✓	✗	✓	✓	✗
Plataforma/s de Pago	✓	✓	✓	✓	✓	✓
Billete electrónico (email, PDF...)	✓	✓	✓	✓	✓	✓
Pasajeros Frecuentes	✗	✓	✗	✓	✓	✗
Compra Abonos	✗	✗	✗	✗	✗	✓
Compra Billete asociado a Abono	✓	✗	✗	✗	✗	✓
Consultar Reserva	✓	✓	✓	✓	✓	✓
Anular Reserva	✓	✓	✓	✓	✓	✓
Modificar Datos Reserva	✓	✓	✓	✓	✓	✗
Gestión Perfil Usuario	✓	✓	✗	✓	✓	✓
Gestión Métodos de Pago	✓	✓	✓	✓	✓	✓
Preguntas Frecuentes	✗	✗	✓	✗	✓	✓
Acceso a Redes Sociales (Twitter, Facebook...)	✓	✗	✗	✗	✗	✓

Últimas Noticias	×	×	×	✓	×	✓
Geolocalización Estación / Parada	×	✓	×	×	×	✓
Localización en tiempo real de vehículo	✓	✓	✓	✓	✓	×
Recordar Reserva	×	✓	×	✓	✓	×

Tabla 2 Comparativa de alternativas a Avantis



Capítulo 3: PLANIFICACIÓN Y PRESUPUESTOS



En este apartado se procede al análisis del modelo o metodología de desarrollo escogida para la realización del proyecto junto con la planificación, costes y estimaciones iniciales de este. Por último, se detalla el coste final o real en base a todos los condicionantes que han afectado en mayor o menor medida al desarrollo del sistema.

3.1 Modelo de desarrollo

Una vez analizadas todas las metodologías y paradigmas utilizados actualmente en el desarrollo de proyectos software, para este proyecto en concreto se ha decidido utilizar el modelo de desarrollo o metodología iterativa - incremental, el cual combina características y ventajas tanto del modelo en cascada como el modelo basado en la construcción de prototipos.

Este modelo de carácter evolutivo se basa en la construcción de la herramienta planificando el desarrollo en varios bloques de trabajo (conocidos como iteraciones o etapas) sobre las cuales se va ampliando y mejorando la funcionalidad y características del producto (Véase figura 7).

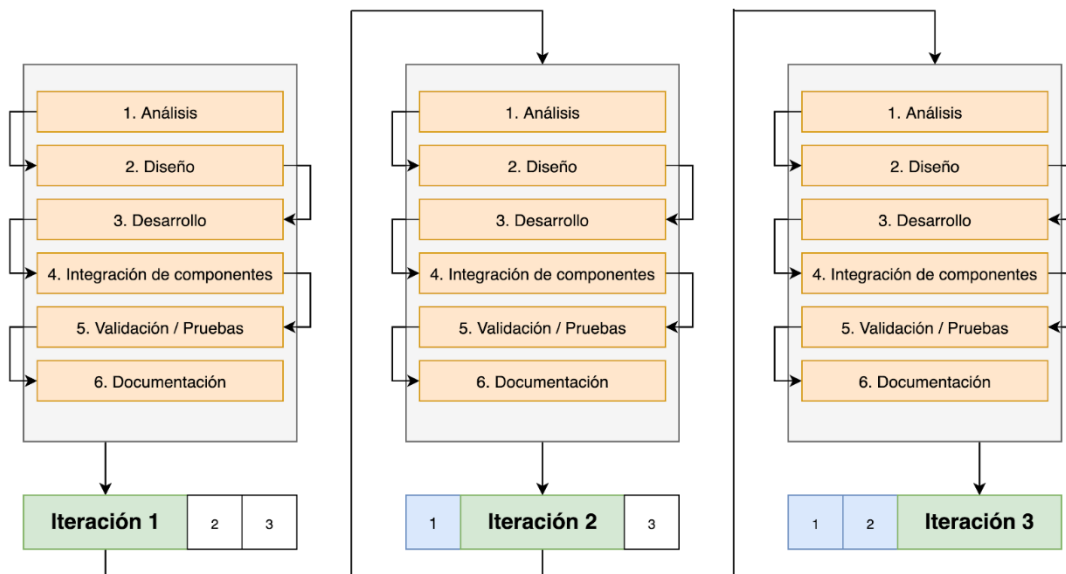


Figura 7 Modelo de desarrollo iterativo-incremental

Cada una de esas etapas o iteraciones dan lugar a un software totalmente operativo sobre el que se cimenta la nueva iteración, y es precisamente por este motivo que se requiere un desarrollo completo del requisito o característica que se ha propuesto para esa iteración, incluyendo una serie de pruebas y una documentación completa que permita a los desarrolladores contar con una base firme para comenzar el desarrollo de nuevos requerimientos y características que formarán parte de las futuras iteraciones. Esto genera como consecuencia una serie de beneficios en el desarrollo, aunque también puede acarrear ciertos inconvenientes. A continuación, se muestran, de manera simplificada, estas consecuencias:

Beneficios:

- Permite contar con un software totalmente operativo al finalizar cada una de las iteraciones.

- Reducir los riesgos en el desarrollo del software, debido a la realización de pruebas para cada una de las iteraciones.
- El desarrollo del proyecto se simplifica, debido a la división de este en bloques de trabajo (etapas).
- Proporciona un alto grado de flexibilidad a la hora de incorporar nuevos requisitos y funcionalidades al desarrollo.
- Mejora el aprendizaje y la experiencia, debido a que con cada una de las iteraciones el equipo de trabajo aumenta su conocimiento del software.

Inconvenientes:

- Requiere de un cliente comprometido a lo largo de todo el desarrollo del proyecto.
- Las distintas fases que conforman cada iteración son rígidas, hasta que no termina una de ellas no comienza la siguiente.

Como se ha explicado previamente, cada etapa o iteración está formada por una serie de fases que se deben completar individualmente al 100% cada una de ellas para poder dar paso a una nueva iteración o para poner punto final al desarrollo del proyecto y entregar un producto completo. Por ello, estas fases serán 6: **análisis, diseño, desarrollo, integración de componentes, validación y pruebas y documentación.**

Por último, para el desarrollo de este proyecto se han considerado un total de 9 etapas o iteraciones como se muestra a continuación:

- 1 iteración inicial para realizar una primera tarea de análisis de requisitos completa.
- 1 iteración para la gestión de los usuarios y conexión con redes sociales.
- 2 iteraciones para el sistema de compra de billetes (compra de billetes ocasional y compra de billetes con abono).
- 1 iteración para el sistema de compra de abonos.
- 1 iteración para la consulta de información de estaciones y gestión de billetes y abonos.
- 1 iteración para el desarrollo del foro comunitario integrado.
- 2 iteraciones para la parte administrativa del sistema.

3.2 Planificación inicial de proyecto

3.2.1 Gestión de conflictos temporales

Dentro de las distintas etapas en las que se encuentra dividido el proyecto pueden surgir distintos problemas que afecten considerablemente a la planificación inicial programada. Uno de estos problemas es la gestión de retrasos o avances temporales, el cual provoca un incremento o decremento en la duración del proyecto.

Este tipo de problemas se originan debido al tiempo destinado por el equipo de trabajo (más concretamente al desarrollador) al aprendizaje y formación requerido en las distintas herramientas y lenguajes utilizados en el desarrollo del proyecto. Es por ello que el tiempo

destinado a las fases de implementación e integración de componentes puede sufrir ciertas modificaciones.

Para intentar paliar este problema, la solución que se ha aplicado es la de aumentar la duración de estas tareas para cada una de las etapas, que consigue a su vez establecer y mantener controlada la fecha de finalización de proyecto, lo que proporciona una estimación más precisa en el caso de que este tipo de problema se encontrara presente.

Si este periodo de formación y aprendizaje no fuera necesario por parte del equipo de trabajo, la duración del proyecto se vería reducida, estableciendo la fecha de finalización de este antes de lo previsto.

Por otro lado, puede darse la situación de que el aumento de tiempo destinado a estas etapas no sea suficiente para cubrir dichas necesidades de formación y que, a pesar de contar con esa ampliación de tiempo, la duración total del proyecto aumente y la fecha de finalización se retrase a lo estimado. Para solucionar este supuesto y que la duración y finalización del proyecto no se vean modificadas, los miembros del equipo de trabajo que se vean afectados deberán aumentar (de forma extra) el número de horas dedicadas a la tarea en cuestión, para así evitar el retraso que se originaría en las etapas y tareas posteriores del desarrollo.

Una vez establecida esta condición, para este proyecto se ha seguido el criterio de aumentar el tiempo destinado a la realización de las tareas de implementación e integración de componentes, debido a que la formación y el aprendizaje básico del desarrollador no formará parte del proyecto al considerar que este cuenta con experiencia previa. Aun así, y de forma preventiva, se ha tenido en cuenta en la estimación temporal.

3.2.2 Aspectos generales de la planificación inicial

Para la planificación de este proyecto se han tenido en consideración una serie de aspectos que repercuten de forma directa con la duración y estimación temporal del mismo como son:

- La planificación del proyecto comienza el 1 de octubre de 2019 debido a motivos laborales previos que no han permitido comenzar el proyecto con anterioridad.
- Se han considerado los fines de semana (sábado y domingo) como días no laborables. Además de esto, el equipo de trabajo solo puede trabajar por las tardes, lo que significa que las semanas laborables quedan definidas de lunes a viernes con un horario de 16:00 a 20:00 (jornada laboral de 4 horas).
- Se han considerado como periodos no laborables para el equipo de trabajo los días festivos y las semanas de vacaciones estipuladas en los calendarios laborales a nivel nacional, autonómico y provincial (véase Figura 8).

	Nombre	Comienzo	Fin
1	Día de la Hispanidad	12/10/2019	12/10/2019
2	Festividad de San Frutos	25/10/2019	25/10/2019
3	Festividad Todos los Santos	01/11/2019	01/11/2019
4	Festividad de la Constitución	06/12/2019	06/12/2019
5	Festividad Inmaculada Concepción	08/12/2019	08/12/2019
6	Vacaciones de Navidad	20/12/2019	08/01/2020
7	Vacaciones de Semana Santa	03/04/2020	13/04/2020
8	Festividad Castilla y León	23/04/2020	23/04/2020
9	Festividad Día del Trabajo	01/05/2020	01/05/2020
10	Fiesta Nacional	12/10/2020	12/10/2020
11	San Frutos	25/10/2020	25/10/2020
12	Todos los santos	02/11/2020	02/11/2020
13	Día de la Constitución	07/12/2020	07/12/2020
14	Inmaculada Concepción	08/12/2020	08/12/2020

Figura 8 Festividades en el calendario de planificación inicial

- Como se ha expuesto anteriormente, la formación y familiarización con las herramientas y tecnologías utilizadas en el desarrollo del proyecto por parte de los miembros del equipo de trabajo no se ha considerado como tal en la estimación temporal. La base de este aprendizaje se considera previa al proyecto, mientras que la formación y experiencia adquirida queda recogida implícitamente en cada una de las etapas del desarrollo.
- En las primeras etapas del proyecto, la duración estimada de las tareas de análisis y diseño es mayor. Esto deriva de la metodología utilizada en el desarrollo del proyecto, la cual requiere al inicio de un análisis exhaustivo de los requisitos, objetivos y características esenciales de la herramienta necesarios para crear los primeros productos que cumplirán con las necesidades básicas del cliente. A todo esto, hay que sumar que es en estas primeras etapas donde el equipo de trabajo se familiariza en mayor medida con el proyecto y las herramientas utilizadas.
- Por el contrario, es en las etapas centrales y finales donde las tareas de implementación, integración y pruebas cobran mayor importancia. Esto deriva del aumento en la complejidad de desarrollo de las nuevas funcionalidades, dando lugar a productos mucho más completos cuya integración con los demás componentes de la herramienta también se complica. Estos productos a su vez requieren de unas pruebas más intensas destinadas a comprobar que se satisfacen todos los requerimientos propuestos por el cliente.

Teniendo en cuenta estos supuestos, la duración estimada del proyecto ha sido de 264 días, con fecha inicial el 1 de octubre de 2019 y con fecha de finalización estimada el 11 de noviembre de 2020. El proyecto, siguiendo con la metodología de trabajo previamente descrita, se encuentra dividido en un total de 9 etapas con una duración aproximada de 30 días cada una junto con una etapa final de revisión final de proyecto con una duración estimada de 5 días. A continuación, en la tabla 3 se detallan estas etapas, las fechas de comienzo y finalización y los días estimados para cada una de ellas.

Fecha Inicio Proyecto: 01/10/2019				
ID	Descripción	Fecha Inicio	Fecha Fin	Días Totales
Etapa 1	Analítica completa	01/10/2019	20/11/2019	35
Etapa 2	Gestión Usuarios e Integración con redes sociales	21/11/2019	20/01/2020	28
Etapa 3	Sistema de Compra de Billetes (Sin Abono)	21/01/2020	27/02/2020	28
Etapa 4	Sistema de Compra de Billetes (Con Abono)	28/02/2020	16/04/2020	28
Etapa 5	Sistema de Compra de Abonos	17/04/2020	27/05/2020	27
Etapa 6	Gestión de contenido y Sistema de consulta de estaciones	28/05/2020	07/07/2020	29
Etapa 7	Foro de Discusión	08/07/2020	17/08/2020	29
Etapa 8	Parte Administrativa 1	18/08/2020	25/09/2020	29
Etapa 8	Parte Administrativa 2	28/09/2020	04/11/2020	26
Revisión final de proyecto		05/11/2020	11/11/2020	5
Fecha Estimada Finalización de Proyecto: 11/11/2020				

Tabla 3 Etapas de planificación inicial de proyecto

Por último, en la figura 9 se muestra un resumen completo de la planificación inicial junto con el diagrama de Gantt y los recursos utilizados para el desempeño de cada una de las tareas planificadas.

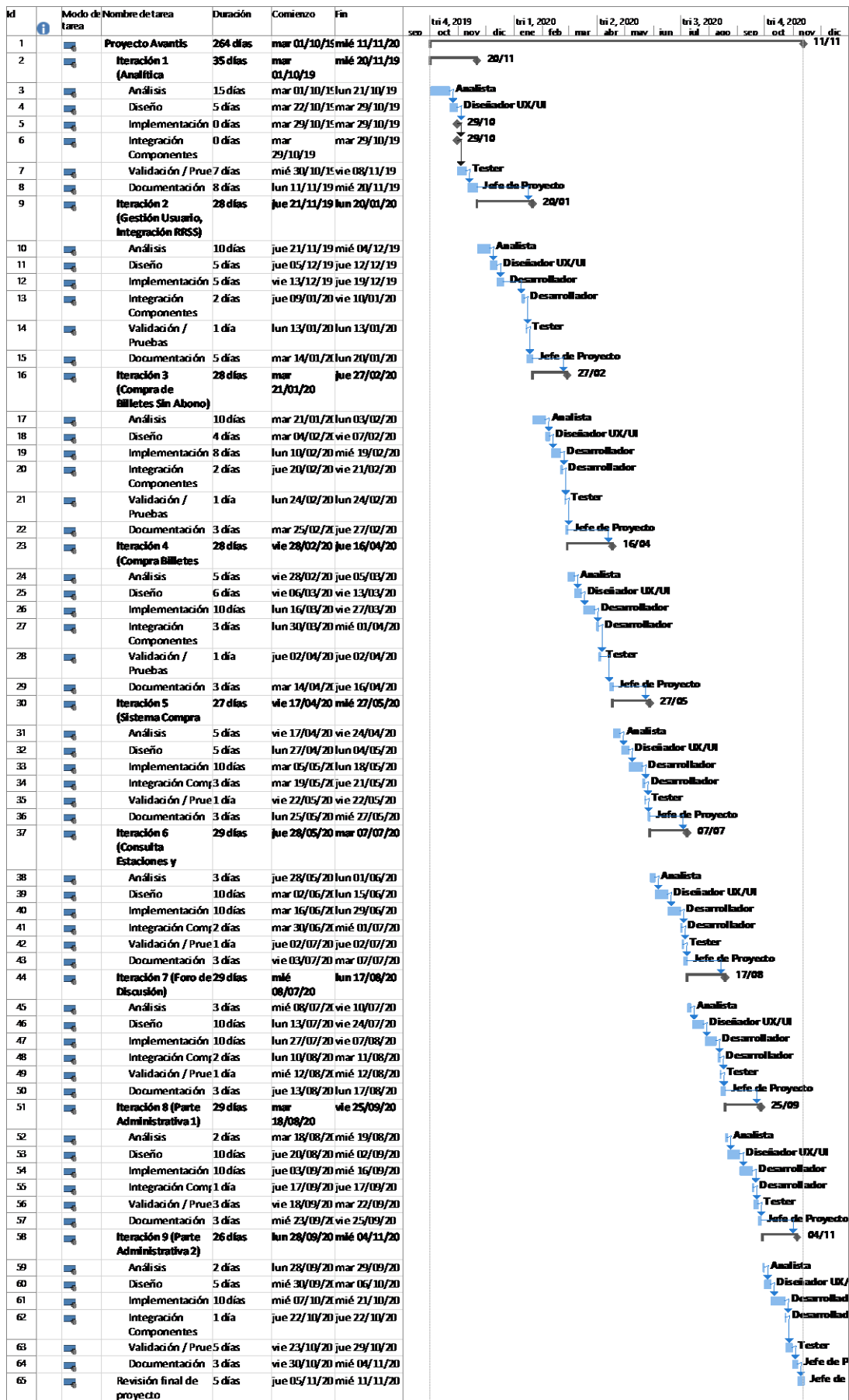


Figura 9 Planificación inicial de proyecto (Diagrama de Gantt)

3.3 Costes iniciales del proyecto

3.3.1 Costes de personal

Estos costes son los relacionados con todos los recursos humanos que intervienen en las distintas fases de planificación inicial establecidas en el diagrama de Gantt previamente expuesto.

Para este proyecto solo se cuenta con un recurso humano que participa activamente en todas las fases del proyecto: el autor. Este autor va a asumir diferentes roles dependiendo de la fase y de la tarea que se encuentre desempeñando, para así, proporcionar una estimación más precisa y realista de este tipo de costes. Los roles asumidos en el proyecto son los siguientes:

- **Jefe de Proyecto:** Persona encargada de gestionar tanto al equipo de trabajo y sus tareas desempeñadas como el control del cumplimiento de objetivos y características del proyecto. También será el encargado de realizar las tareas de revisión y redacción de la documentación.
- **Analista:** Persona encargada de realizar las tareas de analítica y exploración de requisitos que debe cumplir el proyecto. También se encargará de gestionar la arquitectura del sistema.
- **Diseñador UX / UI:** Persona encargada de analizar y diseñar todos los aspectos relacionados con la experiencia e interfaces de usuario.
- **Desarrollador / Programador:** Persona encargada de llevar a cabo las labores de implementación del proyecto.
- **Tester:** Persona encargada de comprobar que se cumplen todos los objetivos y requisitos establecidos para el producto (tanto el producto completo como los distintos productos que se van generando a lo largo de las fases del proyecto).

3.3.2 Costes hardware

Son los costes derivados de la utilización de soportes físicos en el desarrollo del proyecto. En la siguiente tabla (véase tabla 4) aparece reflejado coste de adquisición de cada componente junto con su tiempo de vida útil medio:

Componente (Hardware)	Precio (€)	Vida Útil media
Ordenador Portátil	1299 €	6 años
Ratón	19,90 €	3 años
Pantalla	115,90 €	6 años
Disco duro Externo	49,90 €	5 años
Dispositivo Móvil 1	750 €	3 años
Dispositivo Móvil 2	149 €	3 años

Tabla 4 Componentes hardware

3.3.3 Costes software

Son los costes derivados de todas los programas y herramientas software utilizadas en el desarrollo del proyecto. En la siguiente tabla (véase tabla 5) aparece reflejado el resultado de este análisis:

Componente (Software)	Precio (€)	Vida Útil media
Windows 10 Home	120 €	4 años
Microsoft Project	849,00 €	Ilimitada
Word (Office 365)	0,00 €	
Adobe Reader DC	0,00 €	
Adobe XD	0,00 €	
Android Studio	0,00 €	
Google Drive	0,00 €	
Draw.io	0,00 €	
Notepad++	0,00 €	

Tabla 5 Componentes software

3.3.4 Otros costes

Son costes que intervienen en el desarrollo del proyecto, pero no se pueden clasificar en ninguna de las anteriores tipologías recogidas. En la siguiente tabla (véase tabla 6) aparece reflejado el resultado de este análisis:

Otros gastos	Coste unidad (€) / mes
Servicios básicos (electricidad, agua...)	30,00 €
Conexión a Internet (Fibra óptica 600 MB) + tarifa dispositivo móvil	45,90 €
Papel	7,50 €

Tabla 6 Especificación de otros gastos

3.4 Estimaciones iniciales

En este punto se va a llevar a cabo la estimación inicial de los costes y duración del proyecto. Para ello se recurre a tres de los principales métodos de estimación utilizados en el desarrollo del software: estimación por casos de uso, método de Albrecht con puntos de función y el método COCOMO.

De estos tres procedimientos, el más preciso de los tres es la estimación por casos de uso teniendo en cuenta que para realizar esta estimación de forma correcta se debe contar con

unos casos de uso bien definidos y analizados previamente. Por otro lado, y de forma complementaria se realizarán los procedimientos de Albretch y COCOMO, aunque estos no llegan a ser tan precisos como el anterior.

Por último, se explica y comparan los resultados obtenidos en todos los procedimientos junto con la planificación inicial propuesta para obtener así la visión temporal y presupuestaria más acorde con el proyecto.

3.4.1 Estimación por puntos de función (Método Albretch)

Basada en la visión por parte del usuario del tamaño del sistema, manteniendo al margen todas las especificaciones y detalles técnicos relacionados con la implementación y el código. Debido a que es una estimación más propia de los usuarios, estos tendrán en consideración los siguientes dominios de información a la hora de definir el sistema:

- **Entradas de usuario:** toda aquella información que el usuario introduce en el sistema.
- **Salidas de usuario:** toda aquella información que el sistema presenta al usuario una vez procesada la petición de este.
- **Consultas:** todas aquellas peticiones o entradas que requieren de una respuesta por parte del sistema.
- **Ficheros lógicos internos:** ficheros utilizados exclusivamente por el sistema de forma local.
- **Ficheros lógicos externos:** ficheros externos, los cuales el sistema utiliza para el procesamiento de datos.

Establecidos los dominios de información, es necesario establecer la complejidad para cada una de las funcionalidades del sistema. Para llevar a cabo este procedimiento se utilizarán las siguientes tablas:

Entradas de Usuario	
Descripción	Complejidad
- Alta de nuevo usuario (registro)	Baja
- Modificación perfil de usuario	Baja
- Autenticar usuario (inicio de sesión)	Baja
- Alta tema de discusión	Baja
- Alta respuesta (sobre tema de discusión)	Baja
- Parámetros búsqueda de viaje (compra estándar)	Baja
- Información de Pasajeros	Baja
- Selección plaza	Baja
- Alta método de pago (tarjeta de crédito)	Baja
- Dar de baja método de pago (tarjeta de crédito)	Baja
- Parámetros compra de abono	Baja
- Parámetros búsqueda de viaje (compra con abono)	Baja
- Alta de contenido en gestión administrativa	Baja
- Modificación de contenido en gestión administrativa	Baja

Tabla 7 Entradas de usuario

Salidas Externas	
Descripción	Complejidad
- Notificación (mensaje) en proceso de compra de billetes	Baja
- Notificación (mensaje) en proceso de compra de abonos	Baja
- Notificación (mensaje) en gestión de perfil de usuario	Baja
- Notificación (mensaje) en gestión de método de pago	Baja
- Notificación (mensaje) en anulación de billete	Baja
- Notificación (mensaje) en descarga de billete	Baja
- Resumen compra de billetes	Baja
- Billete en formato PDF	Baja
- Correo electrónico (información de billete)	Baja
- Código QR (billete)	Baja
- Notificación (mensaje) en resultado de gestión administrativa	Baja

Tabla 8 Salidas de usuario

Consultas Externas	
Descripción	Complejidad
- Consultar listado temas de discusión	Baja
- Detalles tema de discusión (individual)	Baja
- Detalles ultimas noticias (individual)	Baja
- Detalles promociones disponibles (individual)	Baja
- Consulta listado de viajes disponibles	Media
- Consulta plazas disponibles	Media
- Consulta listado de estaciones	Baja
- Detalles estación (individual)	Baja
- Consulta listado de abonos disponibles	Baja
- Consulta abonos disponibles (compra con abono)	Baja
- Detalles perfil de usuario	Baja
- Detalles método de pago asociado (tarjeta de crédito)	Baja
- Consultar listado de billetes	Baja
- Detalles billete (individual)	Baja
- Consultar listado de abonos	Baja
- Detalles abono (individual)	Baja
- Consultar perfil de empresa (redes sociales)	Baja
- API de PayPal	Baja
- Consultar listados en gestión administrativa	Baja
- Eliminación de contenido en gestión administrativa	Baja

Tabla 9 Consultas externas de usuario

Ficheros Lógicos Internos	
Descripción	Complejidad
Configuración e información de usuario	Baja

Tabla 10 Ficheros lógicos internos

Ficheros Lógicos Externos	
Descripción	Complejidad
Cloud Firestore (Firebase)	Alta
Authentication (Firebase)	Alta

Tabla 11 Ficheros lógicos externos

A continuación, después de establecer todas las complejidades requeridas por parte del sistema, es momento de calcular los puntos de función sin ajustar (PFNA) en base a la tabla de la figura 12:

Dominio	Complejidad	Peso por Complejidad	Número Total de Funciones	Total
Entradas de Usuario	Baja	x3	14	42
	Media	x4	0	0
	Alta	x6	0	0
Salidas de Usuario	Baja	x4	11	44
	Media	x5	0	0
	Alta	x7	0	0
Consultas de Usuario	Baja	x3	16	48
	Media	x4	2	8
	Alta	x6	0	0
Ficheros Lógicos Internos	Baja	x7	1	7
	Media	x10	0	0
	Alta	x15	0	0
Ficheros Lógicos Externos	Baja	x5	0	0
	Media	x7	0	0
	Alta	x10	2	20
Puntos de Función Sin Ajustar (PFNA)				169

Tabla 12 Correspondencias y puntos de función sin ajustar (PFNA)

Ya definidos los puntos de función sin ajustar (PFNA) es momento de asignar el grado de complejidad de los 14 factores de ajuste que caracterizan la complejidad y la funcionalidad del sistema. Para ello se cuenta con una escala que va desde el 0 al 5, puntuando los factores de complejidad descritos en la tabla 13:

Factor de Ajuste	Complejidad
Comunicación de Datos	5
Funciones Distribuidas	2
Prestaciones	3
Gran uso de la configuración	1
Velocidad de las transacciones	4
Entrada on-line de datos	4
Diseño para la eficiencia del usuario final	3
Actualización de datos on-line	2
Complejidad de los procesos lógicos de la aplicación	3
Reusabilidad del código	1
Facilidad de instalación	1
Facilidad de operación	1
Localizaciones Múltiples	1
Facilidad de cambios	2
TOTAL	33

Tabla 13 Factores de complejidad y puntuación

Para calcular el factor de ajuste (FA) se aplica la siguiente fórmula basada en la suma total de los factores de complejidad (FC) obtenidos previamente:

$$\mathbf{FA} = 0,65 + (0,01 * FC) = 0,65 + (0,01 * 33) = \mathbf{0,98}$$

Una vez obtenido el factor de ajuste, se calculan los puntos de función ajustados (PFA):

$$\mathbf{PFA} = PFNA * FA = 169 * 0,98 = \mathbf{165,62}$$

Para finalizar la estimación es necesario calcular la duración del proyecto junto con el esfuerzo personal en horas. Para ello es necesario establecer la equivalencia puntos de horas por puntos de función contando con que 1 mes de esfuerzo (21 días laborables) equivale a 13 puntos de función.

$$\mathbf{Duración\ de\ proyecto\ estimada} = 165,62\ PFA / 13 = \mathbf{12,74\ meses}$$

3.4.2 Estimación por COCOMO

Este tipo de estimación se basan en el tamaño en líneas de código obtenido por medio de los valores obtenidos en la estimación por puntos de función realizada en el apartado anterior *3.4.1 Estimación por puntos de función (Método Albretch)*. Las mediciones obtenidas representan el tiempo y el esfuerzo dedicados en el desarrollo del sistema.

Antes de comenzar, es necesario conocer las distintas tipologías de COCOMO que se pueden aplicar en base a las características del proyecto que se desea realizar.

- **Orgánico:** Utilizado en proyectos sencillos, pequeños, con un bajo número de desarrolladores y con unos requisitos bastante flexibles.
- **Empotrado:** Utilizado en proyectos complejos, con un número muy elevado de requisitos caracterizados por su gran rigidez.
- **Semi-empotrado:** Punto intermedio entre los dos anteriores.

Una vez definidos y observadas las características principales del proyecto y de los requisitos que se deben tener en cuenta, la opción escogida es la estimación a través de un modelo semi-empotrado.

El siguiente paso es establecer el número de líneas de código estimadas necesarias para la realización de un punto de función. Para ello se utiliza la siguiente tabla donde aparecen reflejado un estándar de medición por cada lenguaje de programación.

Líneas de código / punto de función			
Lenguaje	Número Líneas	Lenguaje	Número Líneas
Ensamblador	320	Basic ANSI/Quick/Turbo	64
Macroensamblador	213	Java	53
C	150	Visual C++	34
Fortran	106	Foxpro 2,5	34
Cobol	106	Visual Basic	32
Pascal	91	Delphi	29
Cobol ANSI 85	91	C++	29
Basic	91	Visual Cobol	20
RPG	80	Clipper	19
PL / I	80	Power builder	16
Ada	71	Hoja de Cálculo	6

Tabla 14 Relación líneas de código - lenguaje de programación

En la tabla no aparece reflejado el lenguaje Android, pero como este está basado en el lenguaje Java en gran parte de sus características se utiliza el valor correspondiente al lenguaje de programación Java.

$$\text{LDC} = \text{PFA} * (\text{Líneas} / \text{PF}) = 165,62 * 53 = \mathbf{8777,86}$$

Una vez calculadas las líneas de código y valorando que el desarrollo de la aplicación se lleva a cabo en un entorno estable y se estima por debajo de las 50 KLDC, se procede a utilizar el modelo orgánico de estimación. Para llevar a cabo este procedimiento es necesario aplicar un factor de esfuerzo basado en 15 características para tener en cuenta.

Factores	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
Fiabilidad requerida	0.75	0.88	1.00	1.15	1.4	
Tamaño de la base de datos		0.94	1.00	1.08	1.16	
Complejidad del software	0.7	0.85	1.00	1.15	1.30	1.65
Restricciones de tiempo de ejecución			1.00	1.11	1.30	1.66
Restricciones de memoria			1.00	1.06	1.21	1.56
Volatilidad del hardware		0.87	1.00	1.15	1.30	
Restricciones de tiempo de respuesta		0.87	1.00	1.07		
Calidad de los analistas	1.46	1.19	1.00	0.86	0.71	
Experiencia con el tipo de aplicación	1.29	1.13	1.00	0.91	0.82	
Experiencia con el hardware	1.21	1.10	1.00	0.90		
Experiencia con el lenguaje de programación	1.14	1.07	1.00	0.95		
Calidad de los programadores	1.42	1.17	1.00	0.86	0.70	
Técnicas modernas de programación	1.24	1.10	1.00	0.91	0.82	
Empleo de herramientas	1.24	1.10	1.00	0.91	0.83	
Restricciones a la duración del proyecto	1.23	1.08	1.00	1.04	1.10	

Tabla 15 Aspectos y ponderación de factores de esfuerzo

El factor aplicado para cada una de ellas aparece reflejado en la tabla 16:

Fiabilidad Requerida	Alto	1,15
Tamaño de la base de datos	Medio	1
Complejidad del software	Medio	1
Restricciones en tiempo de ejecución	Alto	1,11
Restricciones de memoria	Medio	1
Volatilidad del Hardware	Bajo	0,87
Restricciones de tiempo de respuesta	Alto	1,07
Calidad de los analistas	Medio	1
Experiencia con el tipo de aplicación	Medio	1
Experiencia con el hardware	Alto	0,9
Experiencia con el lenguaje de programación	Alto	0,95
Calidad de los programadores	Medio	1

Técnicas modernas de programación	Medio	1
Empleo de herramientas	Alto	0,91
Restricciones a la duración del proyecto	Bajo	1,08

Tabla 16 Calificación factores de esfuerzo

A continuación, se multiplican los valores de la tabla entre sí para obtener el valor m_x utilizado para el posterior cálculo del esfuerzo del proyecto (E) y el tiempo de desarrollo de este (T_{DEV}). Para obtener estos valores, es necesario utilizar las constantes propuestas en la tabla 17 en función del modelo de COCOMO elegido:

Modelo	a	b	C	d
Orgánico	2,40	1,05	2,50	0,38
Semi- Empotrado	3,00	1,12	2,50	0,35
Empotrado	3,6	1,20	2,50	0,32

Tabla 17 Propuestas de cálculo por COCOMO

$$m_x = 1,15 * 1,11 * 0,87 * 1,07 * 0,9 * 0,95 * 0,91 * 1,08 = \mathbf{0,9985}$$

$$E = a * (KLDC)^b * m_x = 3 * (8,77786)^{1,12} * 0,9985 = \mathbf{34,124 \text{ persona-mes}}$$

$$T_{DEV} = c * E^d = 2,5 * (34,124)^{0,35} = \mathbf{8,6 \text{ meses}}$$

Para terminar, se calcula el esfuerzo nominal (N) necesario para la realización del proyecto:

$$N = E / T_{DEV} = 34,124 / 8,6 = \mathbf{3,97}$$

Conclusión: Se necesitan 4 personas a tiempo completo para llevar a cabo el desarrollo del proyecto en 8,6 meses

3.4.3 Estimación por casos de uso

Para realizar este tipo de estimación es necesario tener en cuenta los casos de uso definidos para el proyecto, junto con los actores que intervienen en este. El resultado es la obtención del esfuerzo necesario para completar el proyecto. Para conseguir esto, el modelo de estimación constará de 4 etapas:

1. Obtener el factor de peso de los actores sin ajustar (UAW) basándose en el grado de complejidad de los actores que intervienen en el sistema utilizando para ello las ponderaciones de la tabla 18:

Actor	Descripción	Factor
Simple	Sistema externo que interviene e interactúa con la herramienta a través de una API	1

Medio	Sistema externo que interviene e interactúa con la herramienta a través de un protocolo u otro actor	2
Complejo	Persona que interactúa con la herramienta a través de una interfaz gráfica	3

Tabla 18 Ponderaciones UAW

Los valores obtenidos son:

Actor	Tipo de Actor	Factor
Usuario Anónimo	Complejo	3
Usuario Autenticado	Complejo	3
Administrador	Complejo	3
Plataforma de Pago	Simple	1
Redes Sociales (Twitter)	Simple	1
Cloud Firestore	Medio	2
UAW		13

Tabla 19 UAW totales

2. Obtener el factor de peso de los casos de uso sin ajustar (UUAW) que describen el grado de complejidad de los casos de uso del proyecto. Este procedimiento se puede realizar de dos formas distintas:
 - a) Basado en transacciones: analiza el número de transacciones que realiza un caso de uso. Utiliza las ponderaciones reflejadas en la tabla 20:

Tipo CU	Descripción	Factor
Simple	número transacciones < 4	5
Medio	4 < número transacciones < 7	10
Alto	número transacciones > 7	15

Tabla 20 Ponderaciones para el cálculo de UUAW basado en transacciones

- b) Basado en clases de análisis: analiza el número de clases que intervienen en un caso de uso. Utilizan las ponderaciones reflejadas en la tabla 21:

Tipo CU	Descripción	Factor
Simple	número clases < 5	5
Medio	5 < número clases < 10	10
Alto	número clases > 10	15

Tabla 21 Ponderaciones para el cálculo de UUAW basado en clases de análisis

Al no contar con el desarrollo de la aplicación finalizado se utiliza el procedimiento basado en transacciones, por lo tanto, los factores de peso de los casos de uso sin ajustar (UUAW) para el proyecto son los siguientes:

Actor	Caso de Uso	Transacciones	UUCW
Usuario Anónimo	CU-01 Iniciar Sesión	2	5
	CU-02 Registrarse	2	5
Usuario Anónimo y Usuario Autenticado	CU-Genérico Consultar Noticias y Promociones	2	5
	CU-Genérico Mostrar detalles noticia y promoción	1	5
	CU-07 Consultar estaciones	2	5
	CU-08 Buscar estación	2	5
	CU-09 Mostrar detalles estación	2	5
	CU-10 Acceder a redes sociales	2	5
	CU-11 Consultar foro de sugerencias	2	5
	CU-12 Mostrar detalles tema de discusión	1	5
	CU-13 Consultar horarios viajes	3	5
	CU-14 Comprar billetes	6-7	10

	CU-15 Modificar plaza	3	5
	CU-16 Realizar Pago	3	5
	CU-17 Descargar billete en PDF	1	5
	CU-18 Enviar billete por correo	1	5
	CU-19 Mostrar manual de usuario	1	5
Usuario Autenticado	CU-20 Cerrar sesión	1	5
	CU-21 Consultar perfil usuario	2	5
	CU-22 Modificar perfil usuario	2	5
	CU-23 Darse de baja	2	5
	CU-24 Consultar tarjeta de crédito asociada	2	5
	CU-25 Añadir tarjeta de crédito	2	5
	CU-26 Eliminar tarjeta de crédito	2	5
	CU-Genérico Consultar billetes y abonos	3	5
	CU-28 Mostrar detalles de billete	1	5
	CU-29 Anular billete	2	5
	CU-30 Filtrar reservas	2	5
	CU-32 Mostrar detalles de abono	1	5
	CU-33 Comprar abono	4	10
	CU-34 Comprar billetes con abono	6	10

	CU-Genérico Añadir tema de discusión y respuesta	2	5
Administrador	CU-Genérico Mostrar contenido	2	5
	CU-Genérico Crear contenido	3	5
	CU-Genérico Modificar contenido	3	5
	CU-Genérico Eliminar contenido	3	5
UUCW			195

Tabla 22 UUAW totales

3. Obtener los puntos de caso de uso ajustados (UCP). Dicho valor se obtiene de la multiplicación de los puntos de caso de uso sin ajustar (UAW + UUCW) con los factores técnicos (TCF) y los factores ambientales (EF).

a) Puntos de caso de uso sin ajustar (UUCP)

$$\text{UUCP} = \text{UAW} + \text{UUCW} = 13 + 195 = \mathbf{208}$$

- b) Factor de complejidad técnica (TCF).** Se obtienen de puntuar 13 características basadas en la complejidad del sistema y ponderar estas con sus pesos correspondientes. Las puntuaciones irán de 0 a 5, siendo de 0 a 2 poco relevantes, de 3 a 4 relevantes y 5 en el caso de ser muy relevantes.

Factor	Descripción	Peso	Valor	Resultado
T1	Sistema Distribuido	2	1	2
T2	Tiempos de respuesta	1	4	4
T3	Eficiencia usuario final	1	4	4
T4	Procesamiento interno complejo	1	5	5
T5	Código reutilizable	1	3	3
T6	Facilidad de instalación	0,5	5	2,5
T7	Facilidad de uso	0,5	5	2,5
T8	Portabilidad	2	2	4
T9	Facilidad de cambio	1	0	0

T10	Concurrencia	1	5	5
T11	Objetivos especiales de seguridad	1	2	2
T12	Acceso directo a terceras partes	1	1	1
T13	Formación de los usuarios	1	4	4
Resultado total				39

Tabla 23 Factores de complejidad técnica (TCF)

Una vez puntuados, se calcula el factor de complejidad técnica (TCF):

$$\text{TCF} = 0,06 + (0,01 * \text{Resultado Total}) = 0,06 + (0,01 * 39) = \mathbf{0,45}$$

- c) **Factor ambiental (EF).** Se obtienen a partir de puntuar 8 características basadas en la experiencia, conocimientos y habilidades de todo el personal implicado en el proyecto con sus pesos correspondientes. Las puntuaciones se establecen de la misma forma que las puntuaciones de los factores de complejidad técnica (TCF).

Factor	Descripción	Peso	Valor	Resultado
E1	Familiaridad con el modelo de proyecto	1,5	3	4,5
E2	Experiencia en la aplicación	0,5	3	1,5
E3	Experiencia en orientación a objetos	1	5	5
E4	Capacidad del analista líder	0,5	3	1,5
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	4	8
E7	Personal a tiempo parcial	-1	0	0
E8	Dificultad del lenguaje de programación	-1	2	-2
Resultado total				22,5

Tabla 24 Factores ambientales (EF)

Una vez puntuados, se calcula el factor ambiental (EF):

$$\text{EF} = 1,4 + (-0,03 * \text{Resultado Total}) = 1,4 + (-0,03 * 22,5) = \mathbf{0,725}$$

Una vez obtenidos todos estos valores, se calculan los puntos de función ajustados (UCP):

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF} = 208 * 0,45 * 0,725 = \mathbf{67,86}$$

4. Obtener el esfuerzo horas-persona (E). Dicho valor se obtiene de la multiplicación entre los puntos de caso de uso ajustados (UCP) y el número de horas-persona (CF).

a) Número de horas-persona (CF):

1. **Factor de productividad (FP).** Este valor se obtiene de la suma en primer lugar de los factores ambientales (EF) comprendidos entre el E1 y E6 (ambos inclusive) con un valor inferior a 3 y los factores ambientales E7 y E8 con un valor superior a 3

$$FP = 2 + 0 = 2$$

Con el factor de productividad obtenido se deben ajustar el número de horas-persona en base a este factor utilizando la siguiente tabla de correspondencias:

Horas-Persona (CF)	Descripción
20	FP <= 2
28	FP <= 4
36	FP >= 5

Tabla 25 Relación horas/persona - factor de productividad (FP)

Para esta estimación, **el número de horas-persona (CF) es 20.**

Una vez obtenidos estos valores, se calcula el esfuerzo (E):

$$E = UCP * CF = 67,86 * 20 = 1618,2 \text{ horas-persona}$$

El resultado obtenido se corresponde con el esfuerzo realizado en la fase de codificación del proyecto (representa un 40% del total del proyecto). Por lo tanto, el último paso es calcular el resto de las horas del proyecto utilizando el siguiente modelo:

Fase de desarrollo	% del proyecto	Horas-Persona
Análisis	10%	339,25
Diseño	20%	678,5
Programación	40%	1357
Pruebas	15%	508,875
Sobrecarga	15%	508,875
Resultado Total		3392,5

Tabla 26 Modelo de esfuerzo total de proyecto

Al final, al estimar que solo se cuenta con **1 trabajador** para englobar todo el desarrollo del proyecto, la duración total del mismo asciende a **20,2 meses**.

3.4.4 Comparativa de estimaciones

La información obtenida después de realizar las estimaciones oportunas es la siguiente:

Modelo estimación	Personas	Tiempo
Puntos de función	1	12,74 => 13 meses
COCOMO	4	8,6 meses
Casos de Uso	1	20,2 => 20 meses

Tabla 27 Comparativa de estimaciones 1

Hay que tener en consideración que el proyecto solo será desarrollado por una única persona y que, por ese motivo, la traducción final de los datos obtenidos es la siguiente:

Modelo estimación	Personas	Tiempo
Puntos de función	1	13 meses
COCOMO	1	34,4 meses
Casos de Uso	1	20 meses

Tabla 28 Comparativa de estimaciones definitiva

Observando los resultados y comparándolo con la primera planificación temporal realizada en el apartado 3.2.2 *Aspectos generales de la planificación inicial*, la estimación más precisa y la que más se ajusta se corresponde con **la estimación de Puntos de Función (Método Albretch)**.

3.5 Presupuestos

Una vez realizados todos los análisis y estimaciones iniciales del proyecto, es el momento de realizar el cálculo de los presupuestos previos al desarrollo del proyecto y posteriores al desarrollo de este.

3.5.1 Presupuesto inicial de proyecto

Una vez calculada la estimación más precisa para el proyecto, en este apartado se lleva a cabo el cálculo de los presupuestos en base a la planificación inicial y a la estimación por puntos de función (Método de Albretch).

Costes de personal

Para realizar el cálculo derivado de la planificación inicial se tiene en consideración el número de horas que ha dedicado cada miembro del equipo de trabajo

por separado, aplicándole su correspondiente tasa por cada hora de trabajo. En la tabla 29 se pueden reflejar los resultados obtenidos:

Rol	Tiempo (Horas)	Salario (€)	salario (€) / hora	Coste en Proyecto (€)
Jefe de Proyecto	156	4000 € / mes	23,80 €/h	3712,80 €
Analista	240	3400 € / mes	20,25 €/h	4860,00 €
Diseñador UX/UI	260	1500 € /mes	8,90 €/h	2314,00 €
Desarrollador	376	2100 € / mes	12,50 €/h	4700,00 €
Tester	104	1600 € / mes	9,50 €/h	988,00 €
Total	1136			16574,80 €

Tabla 29 Total inicial costes de personal

Por otro lado, para el cálculo de estos costes para la estimación, al no conocer el porcentaje de trabajo realizado por cada uno de los roles que interviene, se ha decidido utilizar una tasa de horas de trabajo correspondiente a la media de salario entre todos los miembros de trabajo previamente identificados (véase tabla 29).

$$\text{Salario Medio} = 4000 + 3400 + 1500 + 2100 + 1600 = 2520 \text{ € / mes} = 15 \text{ € / hora}$$

Por lo tanto, el coste de personal para la estimación asciende a:

$$12,75 \text{ meses} = 1072 \text{ horas aproximadamente}$$

$$\text{Coste Personal} = 1072 * 15 = 16080,00 \text{ €}$$

Costes hardware

Para realizar el cálculo se ha tenido en cuenta la amortización en base al tiempo de vida útil estimado para cada uno de ellos, junto con su coste de adquisición. Con este procedimiento se obtiene la cuantía real del gasto que supone la utilización de estos componentes en el proyecto. Al ser prácticamente semejantes en duración la planificación y la estimación se ha redondeado a 13 meses el periodo de amortización para ambos cálculos. En la tabla 30 se pueden reflejar los resultados obtenidos:

Componente (Hardware)	Precio (€)	Vida Útil media	Porcentaje (%) de Uso	Coste real en proyecto (€)
Ordenador Portátil	1299 €	6 años	18 %	233,82 €
Ratón	19,90 €	3 años	36 %	7,15 €

Pantalla	115,90 €	6 años	18 %	20,90 €
Disco duro Externo	49,90 €	5 años	21,5 %	10,70 €
Dispositivo Móvil 1	750 €	3 años	36 %	270,00 €
Dispositivo Móvil 2	149 €	3 años	36 %	53,65 €
TOTAL				596,22 €

Tabla 30 Total inicial costes hardware

Costes software

Para realizar el cálculo se ha tenido en cuenta la amortización en base al tiempo de vida útil estimado para cada uno de ellos, junto con su coste de adquisición. Con este procedimiento se obtiene la cuantía real del gasto que supone la utilización de estos componentes en el proyecto. Al ser prácticamente semejantes en duración la planificación y la estimación se ha redondeado a 13 meses el periodo de amortización para ambos cálculos. Además, se ha considerado utilizar todo el software gratuito posible para abaratar los costes. En la tabla 31 se pueden reflejar los resultados obtenidos:

Componente (Software)	Precio (€)	Vida Útil media	Porcentaje (%) de Uso	Coste real en proyecto (€)
Windows 10 Home	120 €	4 años	18,75 %	27,00 €
Microsoft Project	849,00 €	Ilimitada	0 %	0,00 €
Word (Office 365)	0,00 €			0,00 €
Adobe Reader DC	0,00 €			0,00 €
Adobe XD	0,00 €			0,00 €
Android Studio	0,00 €			0,00 €
Google Drive	0,00 €			0,00 €
Draw.io	0,00 €			0,00 €
Notepad++	0,00 €			0,00 €
TOTAL				27,00 €

Tabla 31 Total inicial costes software

Otros costes

Para realizar el cálculo y al considerar semejantes en duración a la planificación y a la estimación se ha redondeado a 13 meses la duración inicial de proyecto. En la tabla 32 se pueden reflejar los resultados obtenidos:

Otros gastos	Coste unidad (€) / mes	Coste real en proyecto (€)
Servicios básicos (electricidad, agua...)	30,00 €	390,00 €
Conexión a Internet (Fibra óptica 600 MB) + tarifa dispositivo móvil	45,90 €	596,70 €
Papel	7,50 €	97,5 €
TOTAL		1084,20 €

Tabla 32 Total inicial de otros gastos

Resultado Final

En la tabla 33 se pueden reflejar los resultados una vez aplicados todos los costes:

Coste real de proyecto (fase inicial)			
Modelo	Tiempo de Vida	Costes (€)	Coste final (€)
Planificación Inicial	13,33 meses	-Coste personal: +16574,80€ -Coste hardware: +596,22€ -Coste software: +27,00€ -Otros gastos: +1084,20€	18282,22 €
Estimación Puntos de Función (Albrecht)	12,75 meses	-Coste personal: +16080,00 € -Coste hardware: +596,22€ -Coste software: +27,00€ -Otros gastos: +1084,20€	17787,42 €

Tabla 33 Presupuesto total inicial

3.5.2 Presupuesto final de proyecto

Una vez concluido el desarrollo del proyecto se procede a calcular el costo real de este en función de los condicionantes que se han ido produciendo a lo largo del tiempo y que han afectado de forma directa a la planificación inicial de este.

La fecha de finalización real del proyecto se ha retrasado al 15 de enero de 2021, es decir, 46 días laborables por encima de la fecha marcada en la planificación inicial. Esto principalmente ha sido provocado por los siguientes motivos:

- Falta de experiencia en la planificación de proyectos.
- Retrasos debido a la pandemia mundial provocada por el coronavirus (COVID-19) que durante el mes de abril de 2020 tuvo completamente parado el trabajo y que, además, durante los meses de verano redujo enormemente los días destinados a la realización del proyecto.
- Incremento de días destinados a la integración, validación y realización de pruebas sobre las etapas destinadas a la implementación de los sistemas de compra que afectan al desarrollador y al tester.
 - +3 días de incremento que afectan a la etapa 3.
 - +3 días de incremento que afectan a la etapa 4.
 - +2 días de incremento que afectan a la etapa 5.
- Incremento de horas de trabajo destinadas a la revisión de la documentación por parte del equipo de trabajo completo.
 - +3 días de incremento.

Después de justificar los principales motivos que han derivado en el retraso del proyecto y en consecuencia de estos, se aplican sanciones económicas. Esta sanción se calcula de la siguiente forma:

$$\text{Sanción diaria} = (0,5 * \text{tasa media hora trabajada}) * \text{horas jornada laboral}$$

$$\text{Sanción diaria} = (0,5 * 15,00) * 4 = 30 \text{ €}$$

. Por lo tanto, la sanción total impuesta al proyecto asciende a:

$$\text{Valor Sanción Total} = \text{días de retraso} * \text{sanción diaria} = 46 * 30,00 = 1380,00 \text{ €}$$

Coste Personal

Rol	Tiempo (Horas)	Salario (€)	salario (€) / hora	Coste en Proyecto (€)
Jefe de Proyecto	156 + 12	4000 € / mes	23,80 €/h	4236,40 €

Analista	240 + 12	3400 € / mes	20,25 €/h	5103,00 €
Diseñador UX/UI	260 + 12	1500 € /mes	8,90 €/h	2420,80 €
Desarrollador	376 + 32	2100 € / mes	12,50 €/h	5100,00 €
Tester	104 + 24	1600 € / mes	9,50 €/h	1216,00 €
Total	1228			18076,20 €

Tabla 34 Total real del coste de personal

Coste hardware

Componente (Hardware)	Precio (€)	Vida Útil media	Porcentaje (%) de Uso	Coste real en proyecto (€)
Ordenador Portátil	1299 €	6 años	20,8 %	270,20 €
Ratón	19,90 €	3 años	41,6 %	8,28 €
Pantalla	115,90 €	6 años	20,8 %	24,11 €
Disco duro Externo	49,90 €	5 años	25 %	12,50 €
Dispositivo Móvil 1	750 €	3 años	41,6 %	312,00 €
Dispositivo Móvil 2	149 €	3 años	41,6 %	62,00 €
TOTAL				689,10 €

Tabla 35 Total real de los costes hardware

Coste software

Componente (Software)	Precio (€)	Vida Útil media	Porcentaje (%) de Uso	Coste real en proyecto (€)
Windows 10 Home	120 €	4 años	31,25 %	37,50 €
Microsoft Project	849,00 €	Ilimitada	0 %	0,00 €
Word (Office 365)	0,00 €			0,00 €

Adobe Reader DC	0,00 €	0,00 €
Adobe XD	0,00 €	0,00 €
Android Studio	0,00 €	0,00 €
Google Drive	0,00 €	0,00 €
Draw.io	0,00 €	0,00 €
Notepad++	0,00 €	0,00 €
TOTAL		37,50 €

Tabla 36 Total real de los costes software

Otros gastos

Otros gastos	Coste unidad (€) / mes	Coste real en proyecto (€)
Servicios básicos (electricidad, agua...)	30,00 €	450,00 €
Conexión a Internet (Fibra óptica 600 MB) + tarifa dispositivo móvil	45,90 €	688,50 €
Papel	7,50 €	112,50 €
TOTAL		1251,00 €

Tabla 37 Total real de otros gastos

Resultado Final

En la tabla 38 se pueden reflejar los resultados una vez aplicados todos los costes:

Coste real de proyecto (fase final)		
Tiempo de Vida	Costes (€)	Coste final (€)
15 meses	-Coste personal: +18076,20€ -Coste hardware: +689,10€ -Coste software: +37,50€ -Otros gastos: +1251,00€ -Sanción por retraso: +1380,00€	21433,80€

Tabla 38 Presupuesto total real (a la finalización de proyecto)



Capítulo 4: ANÁLISIS DEL SISTEMA



En este bloque se expone detalladamente el análisis llevado a cabo a lo largo de este proyecto compuesto principalmente por la descripción minuciosa de todos los requisitos a tener en cuenta en todo el proceso de desarrollo del producto junto con los diagramas necesarios para su mejor comprensión. Estos requisitos se encuentran organizados en plantillas (tablas) para normalizar en todo momento su análisis junto con el diagrama de casos de uso y el modelo entidad-relación, cuyas características se describen a lo largo del bloque.

4.1 Actores del sistema

Los actores son todos esos elementos que desempeñan un rol concreto o interaccionan en el sistema, tanto personas como otros sistemas software externos. En la tabla 39 se pueden observar las características principales de cada uno de estos:

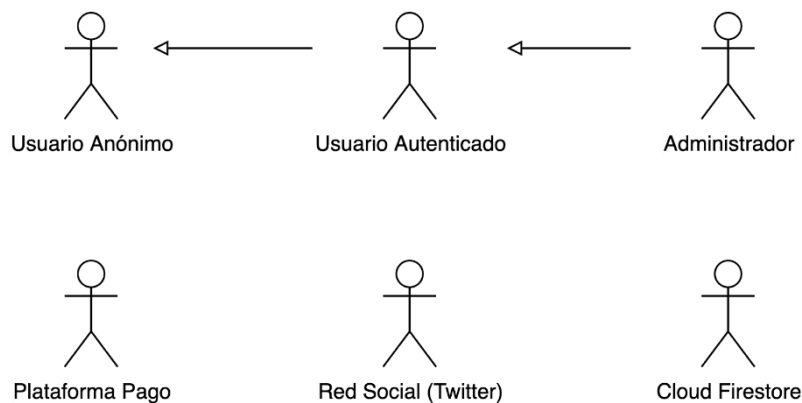


Figura 10 Actores del sistema

ID	Nombre	Descripción
ACT-01	Usuario Anónimo	Usuario que accede al sistema sin iniciar sesión o sin previo registro en la aplicación.
ACT-02	Usuario Autenticado	Usuario registrado y que posteriormente ha iniciado sesión en la aplicación.
ACT-03	Administrador	Usuario encargado de gestionar toda la información de la base de datos.
ACT-04	Plataforma Pago	Herramienta que otorga al sistema la capacidad de soportar pagos en línea.
ACT-05	Redes Sociales (Twitter)	Herramienta que otorga al sistema la capacidad de consultar y explotar los servicios de la red social especificada (Twitter).
ACT-06	Cloud Firestore	Herramienta o servicio en la nube donde se almacena toda la información en la que se apoya la aplicación.

Tabla 39 Descripción de los actores del sistema

El concepto de jerarquía es necesario explicarlo para entender por un lado la funcionalidad propia de un actor, y por otro la funcionalidad que hereda de otro. En este caso y como se puede observar en la figura X, los usuarios autenticados realizarán tanto su funcionalidad propia como toda la que puede realizar un usuario anónimo.

4.2 Requisitos de Usuario

Antes de continuar con su análisis es necesario entender el significado de un requisito de usuario. Un requisito de usuario es toda aquella necesidad propia de un usuario que el sistema (en nuestro caso aplicación móvil) deberá satisfacer.

En este apartado se procede al análisis completo de todos los requisitos de usuario. Para ello es importante identificar primero los requisitos en función del actor/usuario que interviene en la aplicación para poder discernir cuales de estos pertenecen a cada uno de los actores y presentarlos de una forma organizada.

Para llevar a cabo esa presentación organizada de los mismos se utiliza el diagrama de caso de uso, que al comienzo del bloque se ha nombrado. Este diagrama permite representar de forma sencilla e intuitiva las necesidades que debe cumplir el sistema utilizando un lenguaje más próximo al usuario, evitando cualquier tipo de tecnicismo. Es muy importante tener en cuenta la jerarquía de actores previamente descrita en el apartado 4.1 *Actores del sistema* puesto que así se evita duplicar ciertos requisitos que afectan al usuario anónimo y al usuario registrado.

A continuación, se muestra un listado con todos los requisitos de usuario encontrados organizados en función de los usuarios/actores:

4.2.1 Propios de USUARIOS ANÓNIMOS

ID	Descripción
RU-01	El usuario anónimo puede iniciar sesión en la aplicación.
RU-02	El usuario anónimo puede registrarse en el sistema.

Tabla 40 Requisitos de usuario propios de los usuarios ANÓNIMOS

4.2.2 Comunes a usuarios ANÓNIMOS y usuarios AUTENTICADOS

ID	Descripción
RU-03	El usuario anónimo puede visualizar las noticias que se encuentren publicadas en la plataforma.
RU-04	El usuario anónimo puede seleccionar y visualizar los detalles de una noticia publicada.
RU-05	El usuario anónimo puede visualizar las promociones que se encuentren publicadas en la plataforma.

RU-06	El usuario anónimo puede seleccionar y visualizar los detalles de una promoción publicada y en vigor.
RU-07	El usuario anónimo puede consultar y visualizar la información referente a las estaciones que se encuentren en el sistema.
RU-08	El usuario anónimo puede buscar estaciones a través de un buscador (por nombre de estación/municipio).
RU-09	El usuario anónimo puede seleccionar y visualizar la información detallada de una estación.
RU-10	El usuario anónimo puede acceder a los perfiles en redes sociales de la empresa desde la aplicación.
RU-11	El usuario anónimo puede visualizar los temas tratados en el foro de sugerencias.
RU-12	El usuario anónimo puede visualizar los detalles y respuestas propuestas para cada uno de los temas de discusión propuestos en el foro.
RU-13	El usuario anónimo puede consultar y visualizar los horarios de los viajes.
RU-14	El usuario anónimo puede realizar compras de billetes para los viajes disponibles.
RU-15	El usuario anónimo puede seleccionar las plazas que quieran ocupar en un viaje
RU-16	Todos ellos deben completar el pago de cualquier compra en la aplicación con éxito.
RU-17	El usuario anónimo puede descargar su billete en formato PDF una vez finalizada la compra y el pago de este.
RU-18	El usuario anónimo puede enviar su billete a su correo electrónico una vez finalizada la compra y el pago de este.
RU-19	El usuario anónimo puede desplegar un menú de ayuda dentro de la plataforma para facilitar el aprendizaje y entendimiento de la aplicación.

Tabla 41 Requisitos de usuario comunes a usuarios ANÓNIMOS y AUTENTICADOS

4.2.3 Propios de los USUARIOS AUTENTICADOS

ID	Descripción
RU-20	El usuario autenticado puede cerrar sesión en la aplicación.
RU-21	El usuario autenticado puede consultar la información de su cuenta.
RU-22	El usuario autenticado puede modificar su perfil de usuario.

RU-23	El usuario autenticado puede darse de baja de la aplicación.
RU-24	El usuario autenticado puede consultar la información de su tarjeta de crédito/débito (método de pago) asociada.
RU-25	El usuario autenticado puede asociar una tarjeta de crédito a su perfil para facilitar los pagos realizados a través de la aplicación.
RU-26	El usuario autenticado puede eliminar la tarjeta de crédito que tengan asociada a su perfil.
RU-27	El usuario autenticado puede consultar y visualizar todos los billetes asociados a su perfil.
RU-28	El usuario autenticado puede seleccionar y visualizar la información detallada del billete junto con el código QR de este.
RU-29	El usuario autenticado puede anular cualquiera de sus billetes.
RU-30	El usuario autenticado puede buscar sus reservas a través de un buscador (por fecha).
RU-31	El usuario autenticado puede consultar y visualizar todos los abonos asociados a su perfil.
RU-32	El usuario autenticado puede seleccionar y visualizar la información detallada del abono.
RU-33	El usuario autenticado puede realizar compras de abonos.
RU-34	El usuario autenticado puede utilizar un abono que se encuentre en vigor para obtener billetes.
RU-35	El usuario autenticado puede crear nuevos temas de discusión en el foro de sugerencias.
RU-36	El usuario autenticado puede enviar respuestas a un tema de discusión abierto en el foro de sugerencias.

Tabla 42 Requisitos de usuario propios de los usuarios AUTENTICADOS

4.2.4 Propios del usuario ADMINISTRADOR

ID	Descripción
RU-37	El administrador puede visualizar todas las estaciones existentes en la plataforma.
RU-38	El administrador puede visualizar todos los precios de trayectos entre estaciones.
RU-39	El administrador puede visualizar todos los trayectos ofertados en la plataforma.

RU-40	El administrador puede visualizar todos los tipos de abonos ofertados en la plataforma.
RU-41	El administrador puede visualizar todas las noticias publicadas en la plataforma.
RU-42	El administrador puede visualizar todas las promociones publicadas en la plataforma.
RU-43	El administrador puede visualizar todas las reservas almacenadas en la plataforma.
RU-44	El administrador puede visualizar todos los horarios almacenados en la plataforma.
RU-45	El administrador puede visualizar todos los vehículos almacenados en la plataforma.
RU-46	El administrador puede añadir una nueva estación.
RU-47	El administrador puede crear un nuevo precio para un trayecto concreto.
RU-48	El administrador puede crear un trayecto.
RU-49	El administrador puede crear un nuevo tipo de abono.
RU-50	El administrador puede crear una nueva noticia.
RU-51	El administrador puede crear una nueva promoción.
RU-52	El administrador puede crear un nuevo horario.
RU-53	El administrador puede crear un nuevo vehículo.
RU-54	El administrador puede modificar los datos de una estación.
RU-55	El administrador puede modificar el precio para un trayecto.
RU-56	El administrador puede modificar los datos de un trayecto.
RU-57	El administrador puede modificar los datos de un tipo de abono.
RU-58	El administrador puede modificar los datos de una noticia.
RU-59	El administrador puede modificar los datos de una promoción.
RU-60	El administrador puede modificar los datos de un horario.
RU-61	El administrador puede modificar los datos de un vehículo.
RU-62	El administrador puede eliminar una estación.
RU-63	El administrador puede eliminar el precio para un trayecto.

RU-64	El administrador puede eliminar un trayecto.
RU-65	El administrador puede eliminar un tipo de abono.
RU-66	El administrador puede eliminar una noticia.
RU-67	El administrador puede eliminar una promoción.
RU-68	El administrador puede eliminar un horario.
RU-68	El administrador puede eliminar un vehículo.
RU-70	El administrador puede eliminar una reserva existente.
RU-71	El administrador puede buscar sus reservas a través de un buscador (por localizador de reserva).

Tabla 43 Requisitos de usuario propios del usuario ADMINISTRADOR

4.2.5 Modelo de Casos de Uso

El caso de uso se utiliza para representar las interacciones y los procesos de intercambio de información entre el sistema y los usuarios. Es en este apartado donde los requisitos de usuario previamente descritos se representarán visualmente organizados con la ayuda de los diagramas de casos de uso.

Normalmente el diagrama de casos de uso es único, pero en este caso se ha optado por dividir el mismo en función de los actores para una mejor comprensión.



Figura 11 Diagrama de casos de uso de usuario ANÓNIMO



Figura 12 Diagrama de casos de uso de usuario AUTENTICADO

Como se puede observar en el diagrama de casos de uso del usuario autenticado (figura 12) aparecen resaltados cuatro requisitos de usuario. Esto es debido a que también aparecen previamente en el diagrama de casos de uso del usuario anónimo (figura 11) pero son necesarios para entender ciertas funcionalidades que dependen de ellos en el caso del usuario autenticado.

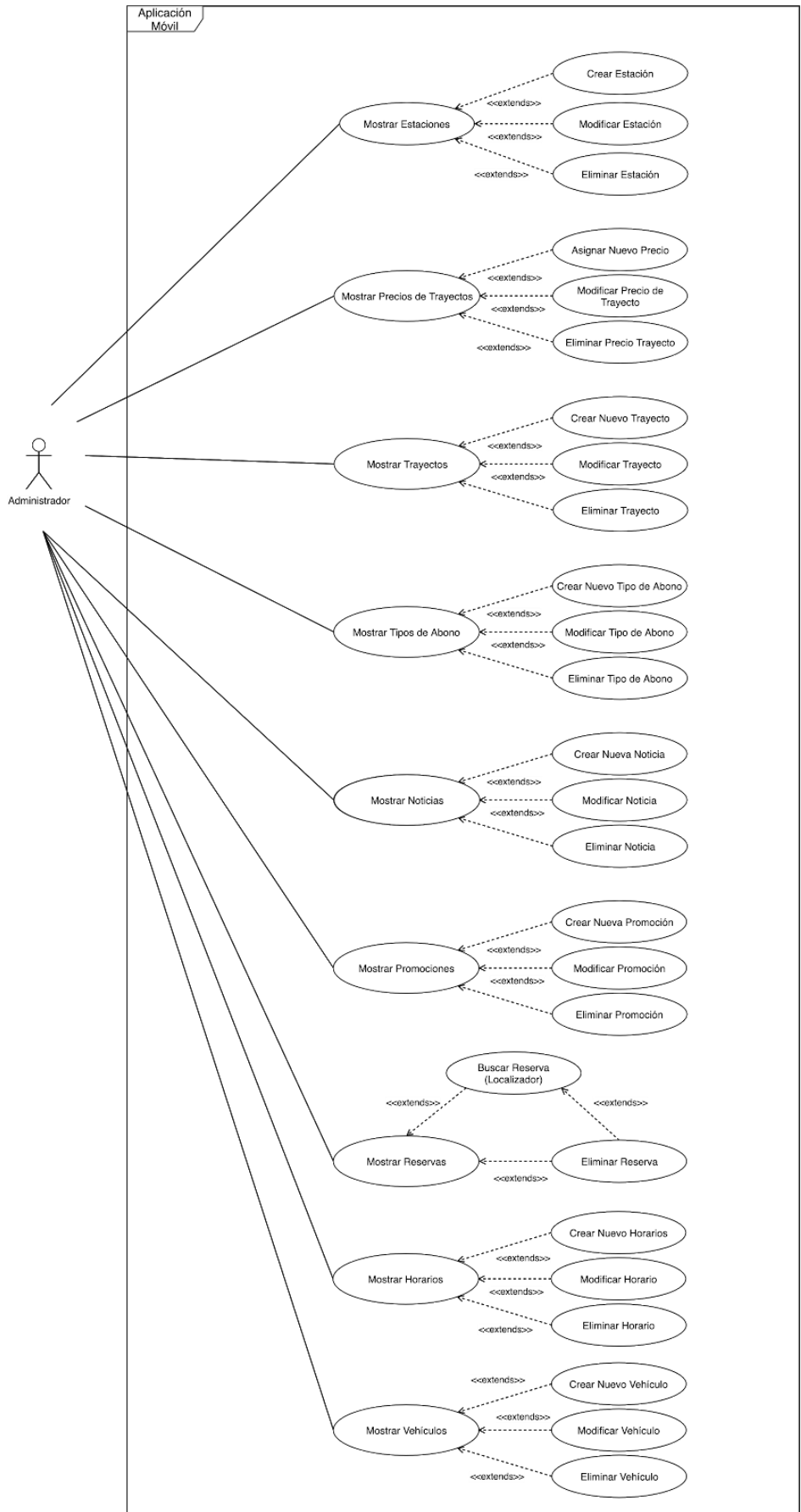


Figura 13 Diagrama de casos de uso del usuario ADMINISTRADOR

4.2.6 Especificación de Casos de Uso

Una vez presentados los casos de uso en sus diagramas correspondientes, es momento de analizar detalladamente cada uno de ellos con el fin de comprender con precisión la funcionalidad completa del sistema, las relaciones existentes entre requisitos y la influencia de estos. Debido al gran número de casos de uso, en este apartado solo se encuentran especificados aquellos más representativos e importantes en el sistema. El resto de ellos se pueden consultar en el *Anexo II Casos de Uso* del documento.

Identificador	CU-01 - Iniciar Sesión		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El usuario se acredita en el sistema a través de su correo electrónico y su contraseña.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse registrado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita acceder a su cuenta. 2. El sistema muestra al usuario los datos que debe rellenar (correo electrónico y contraseña). 3. El usuario completa los campos y los envía al sistema. 4. El sistema comprueba que los datos existan y sean correctos. 5. El sistema redirige al usuario a la página principal de la aplicación. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Los datos de acceso son incorrectos, devolviendo al usuario al paso 2.		
Postcondiciones	El usuario accede a su cuenta de usuario en la aplicación.		
Importancia	Alta	Frecuencia	Alta

Tabla 44 Caso de uso "Iniciar Sesión"

Identificador	CU-02 – Registrarse		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El usuario crea una nueva cuenta de usuario para acceder a la plataforma.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita crear una nueva cuenta. 2. El sistema muestra al usuario los campos que debe completar. 3. El usuario completa los campos y los envía al sistema. 4. El sistema comprueba que los datos sean correctos. 5. El sistema genera una nueva cuenta de usuario. 		
Secuencia Alternativa			
Excepciones en la secuencia	<p>4.1. Ya existe una cuenta de usuario similar, devolviendo al usuario al paso 2.</p> <p>4.2. Los datos de acceso son erróneos, devolviendo al usuario al paso 2.</p>		
Postcondiciones	El sistema almacena la información del nuevo usuario en la base de datos.		
Importancia	Alta	Frecuencia	Alta

Tabla 45 Caso de uso "Registrarse"

Identificador	CU-07 - Consultar información de las estaciones		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra el listado de todas las estaciones que se encuentran registradas.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		

Secuencia Normal	2. La información de la estación debe de encontrarse en estado “VISIBLE” para los usuarios.		
	<ol style="list-style-type: none"> 1. El usuario solicita visualizar el listado de estaciones. 2. El sistema mostrará la lista de todas las estaciones disponibles. 3. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere buscar una estación en concreto, se llevaría a cabo el caso de uso “CU-08 - Buscar estación”. 2. Si el usuario selecciona una estación, se llevaría a cabo el caso de uso “CU-09 - Mostrar detalles de estación”. 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. No hay estaciones disponibles, se notifica la incidencia al usuario.		
Postcondiciones	El usuario visualiza el listado de todas las estaciones que se encuentran registradas y disponibles para consulta.		
Importancia	Media	Frecuencia	Media

Tabla 46 Caso de uso "Consultar información de las estaciones"

Identificador	CU-13 - Consultar Horarios Viajes		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Cliente	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra al usuario todos los posibles viajes y su disponibilidad para los parámetros seleccionados (origen, destino, fecha del viaje y número de pasajeros).		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita acceder al sistema de compras. 2. El usuario solicita acceder a la consulta de horarios/compra billetes. 3. El sistema muestra al usuario los campos que debe completar. 		

Secuencia Alternativa	<ol style="list-style-type: none"> 4. El usuario completa los campos y envía la información al sistema. 5. El sistema comprueba la información. 6. El sistema muestra al usuario la lista con los viajes y la disponibilidad de estos. 7. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario selecciona uno de los viajes, se llevaría a cabo el caso de uso “CU-14 - Comprar Billetes”. 		
Excepciones en la secuencia	<ol style="list-style-type: none"> 5.1. Información de campos errónea, se notifica la incidencia al usuario y se devuelve al paso 3. 6.1. No hay ningún viaje que cumpla los criterios de búsqueda, se notifica la incidencia al usuario. 		
Postcondiciones	El usuario visualiza todos los viajes que cumplen sus criterios de búsqueda, ordenados cronológicamente junto con la disponibilidad de estos.		
Importancia	Alta	Frecuencia	Alta

Tabla 47 Caso de uso "Consultar Horarios Viajes"

Identificador	CU-14 - Comprar Billetes		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema registra una o varias reservas para el viaje o los viajes seleccionados.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario selecciona el viaje que quiere reservar. 2. El sistema asigna automáticamente una plaza vacante del viaje a la solicitud. 3. El sistema genera una reserva “PROVISIONAL” con los datos recogidos. 4. El sistema muestra al usuario los campos requeridos para la “Información del Pasajero”. 		

	<ol style="list-style-type: none"> 5. El usuario completa los campos y envía la información al sistema. 6. El sistema comprueba la información. 7. El sistema actualiza la o las reservas con la “Información del Pasajero” recibida. 8. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere modificar la o las plazas asignadas para su viaje, se llevaría a cabo el caso de uso “CU-15 - Modificar Plaza”. 9. El sistema muestra un resumen con los datos de la o las reservas y solicita al usuario el “Método de Pago” y la confirmación de compra. 10. El usuario selecciona el “Método de Pago” y confirma la compra. 11. El sistema comprueba la información. 12. Se lleva a cabo el caso de uso “CU-16 - Realizar Pago”. 13. El sistema muestra al usuario la información de las reservas realizadas. 14. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere descargar el PDF con la información de la reserva en su dispositivo, se llevaría a cabo el caso de uso “CU-17 - Descargar Billete en PDF”. 2. Si el usuario quiere enviar por correo electrónico la información de la reserva, se llevaría a cabo el caso de uso “CU-18 - Enviar Billete por Correo”.
Secuencia Alternativa	
Excepciones en la secuencia	<p>6.1. Información de campos errónea, se notifica la incidencia al usuario y se devuelve al paso 4.</p> <p>11.1. Información de campos errónea, se notifica la incidencia al usuario y se devuelve al paso 9.</p> <p>12.1. Transacción errónea, se notifica la incidencia al usuario y se devuelve al paso 9.</p>
Postcondiciones	El sistema registra la o las reservas adquiridas en la plataforma.
Importancia	Alta Frecuencia Alta
Comentarios	El paso 1, 2, 3 y 7 de “Secuencia Normal” se llevarán a cabo un número determinado de veces dependiendo de si se quiere comprar un billete “Solo Ida”, “Ida y Vuelta” y del número de pasajeros seleccionado.

Tabla 48 Caso de uso "Comprar Billetes"

Identificador	CU-34 - Comprar Billetes con abono		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema registra una reserva asociada al abono seleccionado.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 3. El usuario debe contar con abonos en vigor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la fecha del viaje. 2. El sistema selecciona los abonos que se encuentran en vigor para la fecha seleccionada. 3. El usuario selecciona uno de esos abonos. 4. El usuario selecciona el trayecto que desea realizar. 5. El sistema muestra al usuario la lista con los viajes y la disponibilidad de estos. 6. El usuario selecciona uno de los viajes disponibles. 7. El sistema asigna automáticamente una plaza vacante del viaje a la solicitud. 8. El sistema genera una reserva “PROVISIONAL” con los datos recogidos. 9. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere modificar la o las plazas asignadas para su viaje, se llevaría a cabo el caso de uso “CU-15 - Modificar Plaza”. 10. El usuario confirma la compra. 11. El sistema muestra al usuario la información de las reservas realizadas. 12. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere descargar el PDF con la información de la reserva en su dispositivo, se llevaría a cabo el caso de uso “CU-17 - Descargar Billete en PDF”. 2. Si el usuario quiere enviar por correo electrónico la información de la reserva, se llevaría a cabo el caso de uso “CU-18 - Enviar Billete por Correo”. 		

Secuencia Alternativa			
Excepciones en la secuencia	2.1. No hay disponibles abonos en vigor para la fecha seleccionada, se notifica la incidencia al usuario y se devuelve al paso 1.		
Postcondiciones	<ol style="list-style-type: none"> 1. La reserva se registra en la plataforma. 2. Los viajes disponibles del abono seleccionado decrecientan en 1. 		
Importancia	Alta	Frecuencia	Alta

Tabla 49 Caso de uso "Comprar Billetes con Abono"

Identificador	CU-33 -Comprar abono		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore, Plataforma de Pago
Obj. Asociados		Req. Asociados	
Descripción	El sistema registra un nuevo abono que se vincula a la cuenta del usuario.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita comprar un nuevo abono. 2. El sistema muestra una lista con los posibles abonos que se pueden adquirir. 3. El usuario selecciona el tipo de abono que desea. 4. El sistema muestra al usuario los campos que debe completar. 5. El usuario completa los campos. 6. El sistema simultáneamente al paso 5 comprueba los campos y muestra al usuario el precio de compra. 7. El usuario confirma la compra. 8. Se realiza el caso de uso "CU-16 - Realizar Pago". 9. El sistema muestra al usuario la información del abono adquirido. 		
Secuencia Alternativa			
Excepciones en la secuencia	6.1. Campos erróneos, se notifica al usuario y se devuelve al paso 4.		

Postcondiciones	El nuevo abono queda registrado en la plataforma y asociado a su comprador/usuario.		
Importancia	Alta	Frecuencia	Alta

Tabla 50 Caso de uso "Comprar Abono"

Identificador	CU-35 - Añadir tema de discusión		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El usuario, a través de un formulario, registra un nuevo tema de discusión en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita añadir un nuevo tema de discusión en el foro de sugerencias. 2. El sistema muestra al usuario los campos que debe completar. 3. El usuario completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra el nuevo tema de discusión y notifica al usuario el resultado de la inserción. 6. El sistema redirige al usuario al foro de sugerencias 		
Secuencia Alternativa			
Excepciones en la secuencia	<p>3.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.</p> <p>5.1 Inserción errónea, se notifica la incidencia al usuario y se devuelve al paso 2.</p>		
Postcondiciones	El sistema registra un nuevo tema de discusión en la plataforma.		
Importancia	Media	Frecuencia	Media

Tabla 51 Caso de uso "Añadir tema de discusión"

Identificador	CU-36 - Añadir respuesta		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El usuario, a través de un formulario, registra una nueva respuesta para el tema de discusión seleccionado.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 3. El usuario debe de seleccionar un tema de discusión registrado en la plataforma. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita añadir una nueva respuesta. 2. El sistema muestra al usuario los campos que debe completar. 3. El usuario completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra la nueva respuesta y notifica al usuario el resultado de la inserción. 6. El sistema redirige al usuario al foro de sugerencias 		
Secuencia Alternativa			
Excepciones en la secuencia	<p>3.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.</p> <p>5.1 Inserción errónea, se notifica la incidencia al usuario y se devuelve al paso 2.</p>		
Postcondiciones	El sistema registra una nueva respuesta asociada al tema de discusión seleccionado en la plataforma.		
Importancia	Media	Frecuencia	Media

Tabla 52 Caso de uso "Añadir Respuesta"

4.3 Reglas de negocio

Este tipo de reglas son especialmente importantes para el proyecto debido a que se encargan de definir las distintas políticas, restricciones, regulaciones y consideraciones esenciales de una compañía y que, por lo tanto, marcan una serie de pautas y especificaciones concretas en el desarrollo del proyecto. Una vez estudiadas y analizadas, las reglas de negocio que se van a considerar las siguientes:

ID	Descripción
RN-01	Avantis deberá cumplir con los estándares definidos en la Ley Orgánica de Protección de Datos (LOPD).
RN-02	El administrador será el único autorizado para gestionar y actualizar la información de los contenidos ofertados por la aplicación.
RN-03	Solo puede existir una cuenta de usuario por correo electrónico en la aplicación.
RN-04	Las contraseñas se almacenan cifradas en la base de datos (SHA256).
RN-05	Las contraseñas contarán obligatoriamente con las siguientes características: <ul style="list-style-type: none"> • Longitud mínima y máxima: 8 - 16 caracteres. • Mínimo una letra MAYÚSCULA y una MINÚSCULA. • Mínimo un caracter especial (?,!,@,-,_, etc.).
RN-06	Los usuarios accederán a su área privada en la aplicación a través de sus credenciales correo electrónico y contraseña.
RN-07	El usuario solo podrá tener asociado un método de pago (tarjeta de crédito/débito) en su cuenta.
RN-08	Los métodos de pago (tarjeta de crédito/débito) admitidos son MasterCard y Visa.
RN-09	Los abonos de 10 viajes se caracterizan por: <ul style="list-style-type: none"> • El número de viajes máximo es 10. • Periodo de validez máximo de 15 días desde su expedición. • Descuento del 10% al total del precio del trayecto seleccionado.
RN-10	Los abonos de 20 viajes se caracterizan por: <ul style="list-style-type: none"> • El número de viajes máximo es 20. • Periodo de validez máximo de 30 días desde su expedición. • Descuento del 20% al total del precio del trayecto seleccionado.
RN-11	Los abonos de 30 viajes se caracterizan por: <ul style="list-style-type: none"> • El número de viajes máximo es 30. • Periodo de validez máximo de 45 días desde su expedición. • Descuento del 25% al total del precio del trayecto seleccionado.

RN-12	Solo los usuarios registrados pueden adquirir abonos.
RN-13	Solo los usuarios registrados pueden adquirir billetes utilizando sus abonos.
RN-14	El abono adquirido por el usuario será personal e intransferible.
RN-15	Los abonos no están sujetos a devolución una vez adquiridos por el usuario.
RN-16	Los viajes no canjeados al final del periodo de validez del abono se perderán sin opción a reembolso.
RN-17	En la compra de un abono, el usuario no puede aplicar ningún tipo de descuento ni promoción.
RN-18	La tarifa/promoción para personas de más de 65 años aplica un descuento del 60% sobre el precio de compra del billete.
RN-19	La tarifa/promoción para jóvenes (menores de 26 años) aplica un descuento del 15% sobre el precio de compra del billete.
RN-20	La tarifa/promoción para jóvenes en posesión del carnet joven acreditativo aplican un descuento del 20% sobre el precio de compra del billete.
RN-21	La tarifa/promoción para familia numerosa con acreditación aplica un descuento del 50% sobre el precio de compra del billete.
RN-22	La tarifa/promoción para discapacitados con acreditación aplica un descuento del 50% sobre el precio de compra del billete.
RN-23	Solo se podrá aplicar una promoción por pasajero en la compra de billetes.
RN-24	El número máximo de pasajeros por proceso de compra de billetes es 4.
RN-25	Toda reserva deberá contener los siguientes datos del comprador: nombre completo, DNI y correo electrónico para identificar al pasajero en todo momento.
RN-26	El número máximo de plazas correspondientes a cada viaje es 55 (capacidad máxima de un autobús tamaño estándar).
RN-27	La reserva es personal e intransferible.
RN-28	No se admiten modificaciones de reserva en ningún supuesto.
RN-29	Solo El usuario autenticado puede participar en el foro de la aplicación, proponiendo temas de discusión y respondiendo cuestiones.
RN-30	El usuario puede anular la reserva siempre que esta petición se realice una fecha anterior a la de la propia reserva.

RN-31	La aplicación solo mostrará la ubicación en Google Maps de la estación si la localización de esta se encuentra definida previamente.
RN-32	La búsqueda de estaciones solo se puede realizar a través del nombre de la estación.
RN-33	Cualquier anulación de reserva será notificada al usuario por correo electrónico a la dirección de correo indicada en la reserva.
RN-34	El formato de descarga del billete en el dispositivo es PDF.
RN-35	La divisa aceptada para la realización de cualquier tipo de pago es el EURO (€).

Tabla 53 Reglas de negocio

4.4 Requisitos Funcionales

Este tipo de requisitos describen todas aquellas características o aspectos fundamentales para el correcto desempeño de la funcionalidad del sistema, es decir, representan toda la funcionalidad que es necesaria desarrollar.

En este proyecto el número de requisitos funcionales es tan grande que se ha procurado sintetizar lo máximo posible estos con el fin de que su comprensión sea más sencilla. Para mejorar esto último también se han organizado en función del caso de uso al que hacen referencia.

Para todos los casos de uso que requieran de un formulario en cualquier parte de su secuencia de ejecución, contarán con los siguientes requisitos:

RF-01: El sistema presentará los campos necesarios para la introducción de información.

RF-02: El sistema será capaz de obtener la respuesta de los valores introducidos en los campos de inicio de sesión.

RF-03: El sistema será capaz de validar los campos del formulario.

RF-04: El sistema será capaz de devolver *feedback* al usuario en función del estado de validación de los campos.

Para todos los casos de uso que requieran de un listado en cualquier parte de su secuencia de ejecución, contarán con los siguientes requisitos:

RF-05: El sistema será capaz de recuperar la información de la base de datos y mostrar al usuario dicha información ordenada

Para todos los casos de uso que requieran de una inserción de información en la base de datos de Firebase en cualquier parte de su secuencia de ejecución, contarán con los siguientes requisitos:

RF-06: El sistema será capaz de crear un nuevo registro en base de datos con la información incorporada por el usuario en la secuencia de ejecución.

RF-07: El sistema será capaz de devolver *feedback* al usuario en función del estado de inserción en base de datos.

Para todos los casos de uso que requieran de una modificación de información en la base de datos de Firebase en cualquier parte de su secuencia de ejecución, contarán con los siguientes requisitos:

RF-08: El sistema será capaz de actualizar el registro seleccionado en base de datos con la información incorporada por el usuario en la secuencia de ejecución.

RF-09: El sistema será capaz de devolver *feedback* al usuario en función del estado de modificación en base de datos.

Para todos los casos de uso que requieran de una eliminación de información en la base de datos de Firebase en cualquier parte de su secuencia de ejecución, contarán con los siguientes requisitos:

RF-10: El sistema será capaz de eliminar el registro seleccionado en base de datos.

RF-11: El sistema será capaz de devolver *feedback* al usuario en función del estado de eliminación en base de datos.

Una vez establecidos estos requisitos comunes a gran parte de las funcionalidades del sistema, a continuación, se muestran todos los requisitos organizados según el caso de uso al que pertenecen:

CU-01	RF-01, RF-02, RF-03, RF-04
	RF-12: El sistema permite almacenar la sesión del usuario autenticado.
CU-02	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-03	RF-05
CU-04	RF-13: El sistema muestra por pantalla la información referente a la noticia seleccionada.
CU-05	RF-05
CU-06	RF-14: El sistema muestra por pantalla la información referente a la promoción seleccionada.
CU-07	RF-05
CU-08	RF-15: El sistema permite aplicar un filtro sobre el total de estaciones y mostrar los resultados.

CU-09	RF-16: El sistema muestra por pantalla la información referente a la estación seleccionada.
	RF-17: El sistema solicita y muestra por pantalla el mapa relativo a la localización de la estación seleccionada.
CU-10	RF-18: El sistema conecta con el perfil social de la empresa en la red social solicitada (Twitter).
CU-11	RF-05
CU-12	RF-19: El sistema muestra por pantalla la información referente al tema de discusión seleccionado.
	RF-20: El sistema solicita y muestra por pantalla el listado de respuestas del tema de discusión seleccionado.
CU-13	RF-01, RF-02, RF-03, RF-04, RF-05
	RF-21: El sistema calcula y muestra la disponibilidad de plazas para cada viaje.
CU-14	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07
	RF-22: El sistema comprueba si la compra la realiza un usuario anónimo o un cliente (usuario autenticado).
	RF-23: El sistema solicita la información de los usuarios en función del número de pasajeros seleccionado.
	RF-24: El sistema busca y asigna aleatoriamente una plaza disponible a cada usuario para cada viaje que se haya seleccionado (Ida, Vuelta).
	RF-25: El sistema permite seleccionar el tipo de tarifa de forma individual a cada pasajero.
	RF-26: El sistema permite solicitar el cambio de plaza a los usuarios para cada viaje (Ida, Vuelta).
	RF-27: El sistema calcula el precio total de la compra de billetes.
	RF-28: El sistema permite escoger el método de pago de las reservas al usuario.
	RF-29: El sistema comprueba si el comprador tiene una tarjeta de crédito asignada para la realización de los pagos.
	RF-30: El sistema muestra por pantalla la información referente a las reservas adquiridas por el comprador.
	RF-31: El sistema permite descargar en PDF la reserva seleccionada.

	RF-32: El sistema permite enviar por correo electrónico la reserva seleccionada.
CU-15	RF-08, RF-09
	RF-33: El sistema muestra por pantalla los asientos que se encuentran libres, ocupados y seleccionados.
	RF-34: El sistema permite al usuario escoger cualquier asiento que se encuentre libre.
	RF-35: El sistema comprueba que los asientos seleccionados por el usuario no han sido ocupados por otra persona mientras se realizaba el proceso.
CU-16	RF-36: El sistema conecta con la plataforma de pago y envía el importe del pago a realizar.
	RF-37: El sistema recibe la respuesta por parte de la plataforma de pago y notifica al usuario dicha respuesta.
CU-17	RF-38: El sistema recoge la información de la reserva y genera un PDF de esta.
	RF-39: El sistema almacena el PDF generado con la información de la reserva en la memoria interna del dispositivo.
	RF-40: El sistema comprobará si el usuario tiene permisos para acceder a la memoria interna del dispositivo. En caso negativo, le solicitará al usuario que los conceda.
	RF-41: El sistema notificará al usuario el estado de la descarga.
CU-18	RF-42: El sistema recoge la información de la reserva y envía un correo electrónico con la misma.
	RF-43: El sistema notificará al usuario el estado del envío.
CU-19	RF-44: El sistema muestra la información de ayuda al usuario.
	RF-45: El sistema permite al usuario cerrar el panel de ayuda.
CU-20	RF-46: El sistema permitirá al usuario autenticado la posibilidad de cerrar sesión.
	RF-47: El sistema cerrará la sesión del usuario autenticado.
	RF-48: El sistema será capaz de devolver <i>feedback</i> al usuario autenticado después de cerrar la sesión.

CU-21	RF-05
CU-22	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09
CU-23	RF-10, RF-11
	RF-49: El sistema solicita al usuario la confirmación de la orden “Darse de baja”.
CU-24	RF-05
CU-25	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-26	RF-10, RF-11
	RF-50: El sistema solicita al usuario la confirmación de la orden de eliminación del método de pago (tarjeta asociada).
CU-27	RF-05
CU-28	RF-51: El sistema muestra por pantalla la información referente al billete/reserva seleccionado.
CU-29	RF-10, RF-11
	RF-52: El sistema solicita al usuario la confirmación de la orden de eliminación de billete/reserva.
CU-30	RF-53: El sistema permite aplicar un filtro sobre las reservas y mostrar los resultados.
CU-31	RF-05
CU-32	RF-54: El sistema muestra por pantalla la información referente al abono seleccionado.
CU-33	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07
CU-34	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-24, RF-26
CU-35	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-36	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-37	RF-05

CU-38	RF-05
CU-39	RF-05
CU-40	RF-05
CU-41	RF-05
CU-42	RF-05
CU-43	RF-05
CU-44	RF-05
CU-45	RF-05
CU-46	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-47	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-48	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-49	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-50	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-51	RF-01, RF-02, RF-03, RF-04, RF-06, RF-07
CU-52	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-08, RF-09, RF-10, RF-11
	RF-55: El sistema permite mostrar al usuario el listado de paradas asignadas al horario.
	RF-56: El sistema permite asignar una nueva parada al horario.
	RF-57: El sistema permite eliminar la parada asignada al horario.
CU-53	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-08, RF-09, RF-10, RF-11
	RF-58: El sistema permite mostrar al usuario el listado de horarios asignados al vehículo.

	RF-59: El sistema permite asignar un nuevo horario al vehículo.
	RF-60: El sistema permite eliminar el horario asignado al vehículo.
CU-54	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09
CU-55	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09
CU-56	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09
CU-57	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09
CU-58	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09
CU-59	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09
CU-60	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09, RF-55, RF-56, RF-57, RF-58
CU-61	RF-01, RF-02, RF-03, RF-04, RF-08, RF-09, RF-58, RF-59, RF-60
CU-62	RF-10, RF-11
	RF-61: El sistema solicita al usuario la confirmación de la orden de eliminación de estación.
CU-63	RF-10, RF-11
	RF-62: El sistema solicita al usuario la confirmación de la orden de eliminación de precio de trayecto.
CU-64	RF-08, RF-09, RF-10, RF-11
	RF-63: El sistema solicita al usuario la confirmación de la orden de eliminación de trayecto.
CU-65	RF-10, RF-11
	RF-64: El sistema solicita al usuario la confirmación de la orden de eliminación del tipo de abono.
CU-66	RF-10, RF-11
	RF-65: El sistema solicita al usuario la confirmación de la orden de eliminación de noticia.

CU-67	RF-10, RF-11
	RF-66: El sistema solicita al usuario la confirmación de la orden de eliminación de promoción.
CU-68	RF-10, RF-11
	RF-67: El sistema solicita al usuario la confirmación de la orden de eliminación de horario.
CU-69	RF-10, RF-11
	RF-68: El sistema solicita al usuario la confirmación de la orden de eliminación de vehículo.

Tabla 54 Clasificación requisitos funcionales

4.5 Requisitos No Funcionales

En este apartado se tienen en cuenta y analizan ciertos requerimientos, también denominados atributos de calidad, que completan aspectos concretos de la funcionalidad de la aplicación tales como comportamientos, restricciones o condicionantes para su correcto funcionamiento impuestos por el cliente. Estos se organizan según su tipología como aparecen a continuación.

4.5.1 Usabilidad

Encargados de establecer los aspectos más importantes para conseguir que un sistema sea fácil de aprender, fácil de utilizar y estéticamente correcto.

ID	Descripción
RNF-USA-01	El tiempo requerido de aprendizaje por parte de un usuario del funcionamiento del sistema (una vez revisado el manual de usuario) no será superior a las 2 horas.
RNF-USA-02	La aplicación tiene que ofrecer un acceso sencillo al manual de usuario en cualquiera de las funciones estimadas para la misma.
RNF-USA-03	La tasa de errores cometidos por el usuario final debe de ser inferior al 0'5% del total.
RNF-USA-04	La aplicación móvil debe caracterizarse por contar con un diseño "Responsive" para su correcta visualización y ejecución en los distintos dispositivos donde se despliegue.

RNF-USA-05	Los procesos de compra y los distintos pasos que se deben seguir se deben estructurar de forma ordenada e intuitiva.
RNF-USA-06	El sistema debe presentar al usuario final mensajes de error que informen concretamente del motivo, sin provocar ningún tipo de duda.
RNF-USA-07	La aplicación se desarrolla para dispositivos Android 5.0 y superiores.

Tabla 55 Requisitos no Funcionales – Usabilidad

4.5.2 Eficiencia

Encargados de valorar y establecer las métricas referentes a las respuestas del sistema frente a acciones de los usuarios.

ID	Descripción
RNF-EFI-01	El sistema debe soportar una tasa de sesiones activas concurrentes de 5000.
RNF-EFI-02	Toda respuesta a cualquier transacción realizada por el usuario final o por el sistema no debe exceder un tiempo de respuesta de 10 segundos.
RNF-EFI-03	La probabilidad de fallo de la aplicación debe ser inferior al 1%.

Tabla 56 Requisitos no Funcionales – Eficiencia

4.5.3 Seguridad

Encargados de establecer los estándares, criterios y acciones a tener en cuenta para prevenir y proteger nuestro sistema frente a brechas de seguridad o ataques.

ID	Descripción
RNF-SEG-01	Los permisos de acceso a datos son gestionados exclusivamente por los administradores del sistema.
RNF-SEG-02	En caso de detectar una brecha de seguridad, el sistema se detiene hasta que el administrador de seguridad solucione el problema.
RNF-SEG-03	Las transacciones realizadas con servidor son siempre cifradas y realizadas bajo el protocolo seguro HTTPS.
RNF-SEG-04	No pueden existir dos usuarios registrados con el mismo correo electrónico.
RNF-SEG-05	Las contraseñas se cifran mediante el algoritmo SHA256 en base 64.

RNF-SEG-06	Las contraseñas contarán obligatoriamente con las siguientes características: <ul style="list-style-type: none"> • Longitud mínima: 8 caracteres. • Longitud máxima: 16 caracteres. • Mínimo una letra MAYÚSCULA. • Mínimo una letra MINÚSCULA. • Mínimo un caracter especial (?,!,@,-,_, etc.).
RNF-SEG-07	La autenticación o inicio de sesión en la aplicación por parte de los usuarios se realiza a través de las credenciales correo electrónico-contraseña.
RNF-SEG-08	Todos los componentes del sistema deben de cumplir los siguientes estándares de seguridad: Ley Orgánica de Protección de Datos (LOPD), ISO 27001, ISO 27017, ISO 27018, SOC1, SOC2, SOC3.

Tabla 57 Requisitos no Funcionales – Seguridad

4.5.4 Disponibilidad

Encargados de establecer y medir el tiempo operativo de un sistema.

ID	Descripción
RNF-DSP-01	El sistema debe tener una disponibilidad 24/7 (24 horas al día los 7 días de la semana).
RNF-DSP-02	La disponibilidad del sistema cada vez que el usuario final intente acceder al mismo será del 99%.

Tabla 58 Requisitos no Funcionales – Disponibilidad

4.5.5 Interfaces de comunicación

Encargados de definir y establecer los protocolos y los estándares en la comunicación del sistema.

ID	Descripción
RNF-ICM-01	El sistema se comunica con el servidor a través del protocolo HTTP utilizando servicios REST.
RNF-ICM-02	El sistema se comunica con la plataforma de pago (PayPal) a través de servicios REST.

Tabla 59 Requisitos no Funcionales - Interfaces de Comunicación

4.5.6 Implementación

Encargados de indicar las tecnologías y lenguajes que se utilizan en el proceso de desarrollo de software.

ID	Descripción
RNF-IMP-01	El desarrollo de la aplicación se realizará utilizando el entorno de desarrollo Android Studio 3.3 y como lenguaje de programación Java.
RNF-IMP-02	El diseño de la interfaz de usuario será desarrollado con el lenguaje XML.

Tabla 60 Requisitos no Funcionales - Implementación

4.6 Requisitos de Información

Este tipo de requisitos se encargan de describir todas aquellas características relacionadas con los datos y la gestión de esos datos por parte del sistema (creación, modificación, borrado). En la tabla 61 se definen los datos requeridos por la aplicación.

ID	Descripción
RI-01	El sistema se encargará de almacenar la información de todos los usuarios que se registren en la plataforma.
RI-02	El sistema se encargará de almacenar la información de las tarjetas de crédito asignadas a los usuarios en la plataforma.
RI-03	El sistema se encargará de almacenar la información de las estaciones/paradas registradas en la plataforma.
RI-04	El sistema se encargará de almacenar la información de los precios de billete para cada trayecto registrado en la plataforma.
RI-05	El sistema se encargará de almacenar la información de los trayectos disponibles en la plataforma.
RI-06	El sistema se encargará de almacenar la información de las reservas adquiridas por los usuarios para los viajes disponibles en la plataforma.
RI-07	El sistema se encargará de almacenar la información de las noticias disponibles en la plataforma.
RI-08	El sistema se encargará de almacenar la información de las promociones disponibles en la plataforma.
RI-09	El sistema se encargará de almacenar la información de los tipos de abono disponibles para compra en la plataforma.

RI-10	El sistema se encargará de almacenar la información de los abonos adquiridos por los usuarios en la plataforma.
RI-11	El sistema se encargará de almacenar la información de los temas de discusión en la plataforma.
RI-12	El sistema se encargará de almacenar la información de las respuestas propuestas para los temas de discusión en la plataforma.
RI-13	El sistema se encargará de almacenar la información de los horarios de trayecto en la plataforma.
RI-14	El sistema se encargará de almacenar la información de los vehículos disponibles en la plataforma.

Tabla 61 Requisitos de información

4.6.1 Modelo Entidad-Relación

Después de indicar todos los requisitos de información del sistema en el apartado anterior, a continuación, se representan a través de un diagrama entidad-relación. Este diagrama recoge los requisitos y les muestra de forma organizada indicando las principales entidades y relaciones que los caracterizan.

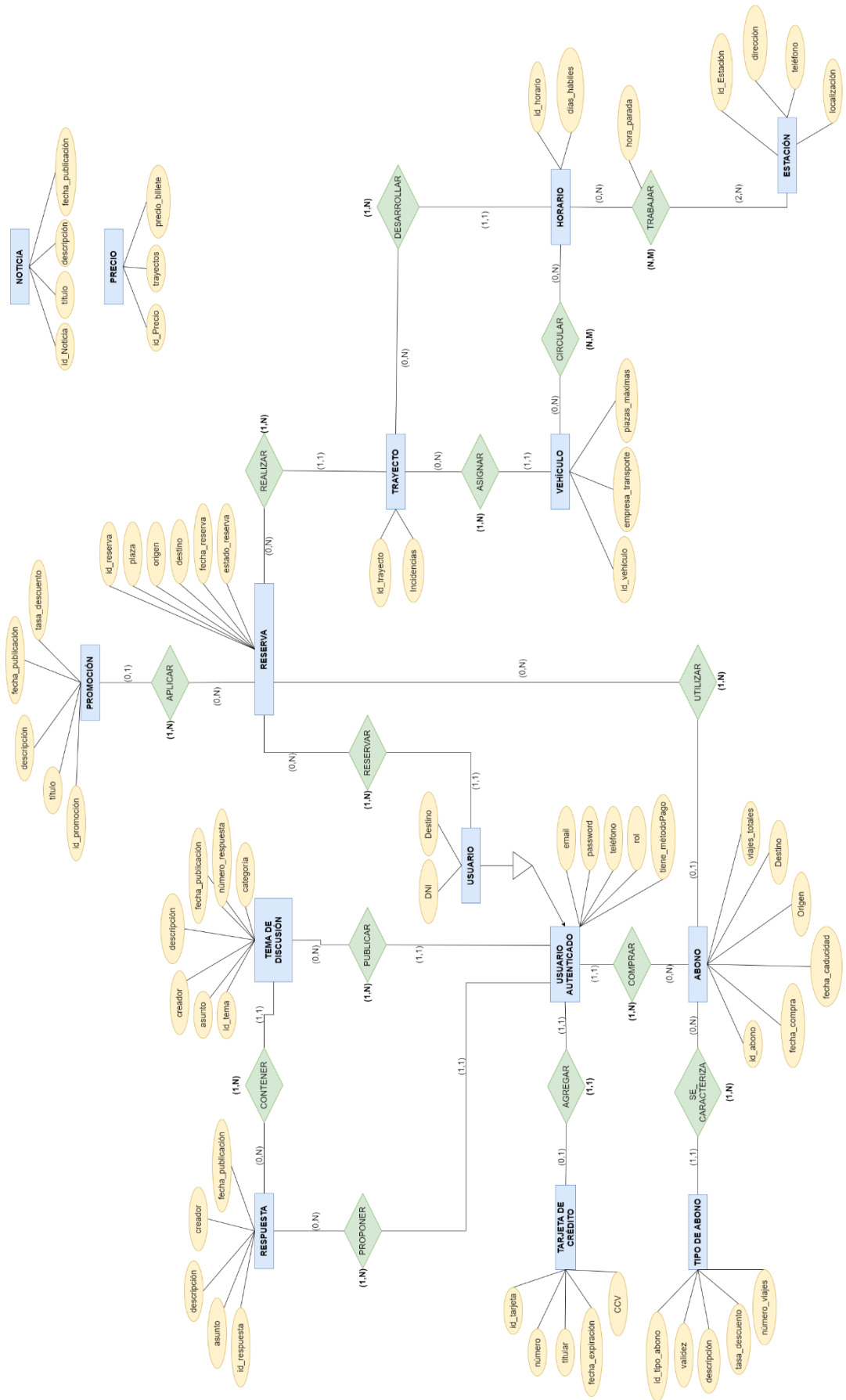


Figura 14 Modelo entidad-relación

4.6.2 Diccionario de datos

Una vez definidos y representados, el último paso es analizar detalladamente estos requisitos en base al modelo entidad-relación. Cada entidad está definida por una serie de atributos que, en este apartado, son identificados, analizados y referenciados a la propia entidad de manera única.

Entidad: Usuario						
Representa a todos los usuarios que pueden interactuar con la aplicación.						
ID	Campo	Tipo	Nulo	Descripción	Unique	Restricción
RI-EU-01	DNI	Cadena de caracteres	NO	Clave identificativa del usuario	SI	Primary key
RI-EU-02	nombre	Cadena de caracteres	NO	Representa el nombre completo del usuario	NO	

Tabla 62 Entidad "Usuario"

Entidad: Usuario Autenticado						
Representa al usuario que ha iniciado sesión en la aplicación						
ID	Campo	Tipo	Nulo	Descripción	Unique	Restricción
RI-EUA-01	email	Cadena de caracteres	NO	Dirección de correo electrónico asociada al usuario	SI	Primary key formato: xxx@yyy.com
RI-EUA-02	password	Cadena de caracteres	NO	Contraseña de inicio de sesión en la aplicación	NO	8 <= Long <=16 Min. 1 mayúscula Min. 1 especial (?,@,-,....)
RI-EUA-03	teléfono	Cadena de caracteres	NO	Teléfono asociado al usuario	NO	Longitud = 9

RI-EUA-04	rol	Cadena de caracteres	NO	Rol del usuario dentro del sistema	NO	“Usuario” o “Administrador”
RI-EUA-05	tiene método Pago	BOOLEAN	NO	Indica si el usuario tiene una tarjeta de crédito asignada o no	NO	por defecto = false

Tabla 63 Entidad "Usuario Autenticado"

Entidad: Tarjeta de Crédito

Representa el método de pago asociado a un usuario.

ID	Campo	Tipo	Nulo	Descripción	Único	Restricción
RI-ETC-01	id tarjeta	Cadena de caracteres	NO	Clave identificativa de una tarjeta de crédito	SI	Primary key
RI-ETC-02	número	Cadena de caracteres	NO	Número de tarjeta de crédito	NO	Longitud = 16
RI-ETC-03	titular	Cadena de caracteres	NO	Nombre del titular de la tarjeta de crédito	NO	
RI-ETC-04	fecha expiración	Cadena de caracteres	NO	Fecha de caducidad de la tarjeta de crédito	NO	Formato: MM/YY
RI-ETC-05	CCV	Cadena de caracteres	NO	Código de seguridad de la tarjeta de crédito	NO	Longitud = 3

Tabla 64 Entidad "Tarjeta de Crédito"

Entidad: Estación

Representa toda la información sobre las estaciones vinculadas a la plataforma.

ID	Campo	Tipo	Nulo	Descripción	Uniqu e	Restricción
RI-EE-01	id estación	Cadena de caracteres	NO	Clave identificativa de una estación	SI	Primary key
RI-EE-02	dirección	Cadena de caracteres	NO	Representa la dirección física de la estación	NO	
RI-EE-03	Teléfono	Cadena de caracteres	NO	Teléfono de contacto de la estación	NO	Longitud = 9
RI-EE-04	email	Cadena de caracteres	NO	Correo electrónico de contacto de la estación	NO	formato: xxx@yyy.com
RI-EE-05	Localización	Geopoint	NO	Representa en coordenadas de la localización la estación.	NO	Latitud, Longitud

Tabla 65 Entidad "Estación"

Entidad: Precio

Representa al precio de billete asociado para un trayecto concreto.

ID	Campo	Tipo	Nulo	Descripción	Unique	Restricción
RI-EP-01	id precio	Cadena de caracteres	NO	Clave identificativa de un precio	SI	Primary key
RI-EP-02	precio billete	Real	NO	Precio base establecido para un trayecto determinado	NO	

RI-EP-03	trayectos	Array [2]	NO		NO	{ origen-destino, destino-origen}
-----------------	-----------	-----------	----	--	----	-----------------------------------

Tabla 66 Entidad "Precio"

Entidad: Trayecto						
Representa toda la información sobre un trayecto / viaje.						
ID	Campo	Tipo	Nulo	Descripción	Unique	Restricción
RI-ET-01	id trayecto	Cadena de caracteres	NO	Clave identificativa de un trayecto	SI	Primary key
RI-ET-02	incidencia	Cadena de caracteres	NO	Detalles de las incidencias de un trayecto	NO	

Tabla 67 Entidad "Trayecto"

Entidad: Reserva						
Representa toda la información sobre una reserva.						
ID	Campo	Tipo	Nulo	Descripción	Unique	Restricción
RI-ER-01	id reserva	Cadena de caracteres	NO	Clave identificativa de una reserva	SI	Primary key
RI-ER-02	origen	Cadena de caracteres	NO	Representa la estación donde se inicia el trayecto	NO	
RI-ER-03	destino	Cadena de caracteres	NO	Representa la estación donde se finaliza el trayecto	NO	
RI-ER-04	fecha reserva	Cadena de caracteres	NO	Representa la fecha precisa de realización del trayecto	NO	dd/MM/yyyy HH:mm

RI-ER-05	estado reserva	Cadena de caracteres	NO	Cualquier incidencia que afecte a la reserva	NO	Por defecto: OK
RI-ER-06	plaza	Cadena de caracteres	NO	Representa la plaza asignada al comprador para el trayecto	NO	

Tabla 68 Entidad "Reserva"

Entidad: Noticia

Representa cualquier noticia publicada en la aplicación.

ID	Campo	Tipo	Nulo	Descripción	Único	Restricción
RI-EN-01	id noticia	Cadena de caracteres	NO	Clave identificativa de una noticia	SI	Primary key
RI-ETD-02	título	Cadena de caracteres	NO	Título de la noticia	NO	
RI-ETD-03	descripción	Cadena de caracteres	NO	Descripción /detalles de la noticia	NO	
RI-ETD-04	fecha publicación	DATE	NO	Representa la fecha de creación de la noticia	NO	formato: dd/MM/yyyy HH:mm:ss

Tabla 69 Entidad "Noticia"

Entidad: Promoción

Representa las promociones que se pueden aplicar en el proceso de compra de billetes.

ID	Campo	Tipo	Nulo	Descripción	Único	Restricción
RI-EPr-01	id promoción	Cadena de caracteres	NO	Clave identificativa de una promoción	SI	Primary key
RI-EPr-02	título	Cadena de caracteres	NO	Nombre de la promoción	NO	

RI-EPr-03	descripción	Cadena de caracteres	NO	Descripción/ detalles de la promoción	NO	
RI-EPr-04	fecha publicación	DATE	NO	Fecha publicación de promoción	NO	formato: dd/MM/yyyy HH:mm:ss
RI-EPr-05	tasa descuento	Entero	NO	Representa el % descuento aplicable sobre el precio de compra de un billete	NO	Valor > 0

Tabla 70 Entidad "Promoción"

Entidad: Tipo de Abono

Representa las características de las tipologías de abono que el usuario puede adquirir.

ID	Campo	Tipo	Nulo	Descripción	Unique	Restricción
RI-ETA-01	id tipo abono	Cadena de caracteres	NO	Clave identificativa de un tipo de abono	SI	Primary key
RI-ETA-02	validez	Entero	NO	Representa la duración máxima de validez del tipo de abono indicado	NO	Valor > 0
RI-ETA-03	descripción	Cadena de caracteres	NO	Descripción del tipo de abono	NO	
RI-ETA-04	tasa descuento	Entero	NO	Representa el % de descuento aplicado sobre el precio total para un trayecto	NO	Valor > 0
RI-ETA-05	número viajes	Entero	NO	Número de trayectos máximo que se pueden adquirir para un tipo de	NO	Valor > 0

				abono determinado		
--	--	--	--	-------------------	--	--

Tabla 71 Entidad "Tipo de Abono"

Entidad: Abono						
Representa el abono adquirido por un usuario.						
ID	Campo	Tipo	Nulo	Descripción	Único	Restricción
RI-EA-01	id abono	Cadena de caracteres	NO	Clave identificativa de un abono	SI	Primary key
RI-EA-02	fecha compra	Cadena de caracteres	NO	Fecha de adquisición de abono	NO	Formato: dd/MM/yyyy
RI-EA-03	fecha caducidad	Cadena de caracteres	NO	Fecha de caducidad del abono	NO	Formato: dd/MM/yyyy
RI-EA-04	origen	Cadena de caracteres	NO	Representa la estación de origen del abono	NO	
RI-EA-05	destino	Cadena de caracteres	NO	Representa la estación final del abono	NO	
RI-EA-06	número viajes	Entero	NO	Número de trayectos máximo que se pueden adquirir con el abono	NO	Valor > 0

Tabla 72 Entidad "Abono"

Entidad: Tema de Discusión						
Representa al tema de discusión propuesto en el foro de la aplicación.						
ID	Campo	Tipo	Nulo	Descripción	Único	Restricción
RI-ETD-01	id tema	Cadena de caracteres	NO	Clave identificativa de un tema	SI	Primary key

RI-ETD-02	asunto	Cadena de caracteres	NO	Título del tema	NO	
RI-ETD-03	descripción	Cadena de caracteres	NO	Descripción /detalles del tema	NO	
RI-ETD-04	creador	Cadena de caracteres	NO	Indicar el propietario/creador de un tema	SI	
RI-ETD-05	categoría	Cadena de caracteres	NO	Indica la categoría establecida para un tema.	SI	
RI-ETD-06	fecha publicación	DATE	NO	Fecha de creación del tema	NO	Formato: dd/MM/yyyy HH:mm:ss
RI-ETD-07	número respuestas	Entero	NO	Representa el número de respuestas asociadas a un tema	NO	

Tabla 73 Entidad "Tema de Discusión"

Entidad: Respuesta

Representa a la respuesta asociada a un tema de discusión propuesto en el foro de la aplicación.

ID	Campo	Tipo	Nulo	Descripción	Unique	Restricción
RI-ER-01	id respuesta	Cadena de caracteres	NO	Clave identificativa de una respuesta	SI	Primary key
RI-ETD-02	asunto	Cadena de caracteres	NO	Título de la respuesta	NO	
RI-ETD-03	descripción	Cadena de caracteres	NO	Descripción /detalles de la respuesta	NO	
RI-ETD-04	creador	Cadena de caracteres	NO	Indicar el propietario/creador de una respuesta	NO	
RI-ETD-05	fecha publicación	DATE	NO	Fecha de creación de respuesta	NO	Formato: dd/MM/yyyy

						HH:mm:ss
--	--	--	--	--	--	----------

Tabla 74 Entidad "Respuesta"

Entidad: Horario						
Representa toda la información sobre los horarios de trayecto.						
ID	Campo	Tipo	Nulo	Descripción	Único	Restricción
RI-EH-01	id horario	Cadena de caracteres	NO	Identificador de horario	SI	Primary key
RI-EE-02	días hábiles	Array []	NO	Días de circulación del horario	NO	Mín. 1 día de la semana

Tabla 75 Entidad "Horario"

Entidad: Vehículo						
Representa toda la información sobre los vehículos disponibles para realizar trayectos.						
ID	Campo	Tipo	Nulo	Descripción	Único	Restricción
RI-EV-01	id vehículo	Cadena de caracteres	NO	Identificador de horario	SI	Primary key
RI-EE-01	empresa transporte	Cadena de caracteres	NO	Identificador de la empresa a la que pertenece el vehículo	NO	
RI-EE-01	plazas máximas	Entero	NO	Número de plazas disponibles del vehículo	NO	Valor > 0

Tabla 76 Entidad "Vehículo"

Por otro lado, se encuentran las relaciones entre las distintas entidades que se definen a continuación:

Relación "Agregar"

Es el resultado de la relación entre "Usuario Autenticado" y "Tarjeta de Crédito" y define la tarjeta que se encuentra asociada a cada usuario autenticado.

Cardinalidad Global: (1,1) → Debido a que el usuario puede tener ninguna o 1 tarjeta asignada (0,1) y la tarjeta solo puede estar asociada a un único usuario (1,1).

Relación “se caracteriza”

Es el resultado de la relación entre “Abono” y “Tipo de Abono” y define las características de un abono en función del tipo de abono adquirido.

Cardinalidad Global: (1,N) → Debido a que el abono solo puede tomar características de 1 tipo de abono (1,1) pero pueden existir ninguno o varios abonos con las características del mismo tipo de abono (0,N).

Relación “comprar”

Es el resultado de la relación entre “Usuario Autenticado” y “Abono” y define el abono que se encuentra asociado a cada usuario autenticado.

Cardinalidad Global: (1,N) → Debido a que el usuario puede tener ninguno o varios abonos adquiridos (0,N) y el abono solo puede estar asociada a un único usuario (1,1).

Relación “publicar”

Es el resultado de la relación entre “Usuario Autenticado” y “Tema de Discusión” y define el tema de discusión que se encuentra asociado a cada usuario autenticado.

Cardinalidad Global: (1,N) → Debido a que el usuario puede tener ninguno o varios temas de discusión publicados (0,N) y el tema de discusión solo puede estar asociado a un único usuario (1,1).

Relación “proponer”

Es el resultado de la relación entre “Usuario Autenticado” y “Respuesta” y define la respuesta asociada a cada usuario autenticado.

Cardinalidad Global: (1,N) → Debido a que el usuario puede tener ninguna o varias respuestas publicadas (0,N) y la respuesta solo puede estar asociada a un único usuario (1,1).

Relación “contener”

Es el resultado de la relación entre “Tema de Discusión” y “Respuesta” y define la respuesta asociada a cada tema de discusión.

Cardinalidad Global: (1,N) → Debido a que el tema de discusión puede tener ninguna o varias respuestas publicadas (0,N) y la respuesta solo puede estar asociada a un único tema de discusión (1,1).

Relación “reservar”

Es el resultado de la relación entre “Usuario” y “Reserva” y define la reserva asociada a cada usuario.

Cardinalidad Global: (1,N) → Debido a que el usuario puede tener ninguna o varias reservas asociadas (0,N) y la reserva solo puede estar asociada a un único usuario (1,1).

Relación “utilizar”

Es el resultado de la relación entre “Abono” y “Reserva” y define el abono asociado a la reserva.

Cardinalidad Global: (1,N) → Debido a que el abono puede tener ninguna o varias reservas asociadas (0,N) y la reserva solo puede estar asociada o no a un único abono (0,1).

Relación “aplicar”

Es el resultado de la relación entre “Promoción” y “Reserva” y define la promoción asociada a la reserva.

Cardinalidad Global: (1,N) → Debido a que la promoción puede tener ninguna o varias reservas donde se aplica (0,N) y la reserva puede aplicar o no una única promoción (0,1).

Relación “realizar”

Es el resultado de la relación entre “Trayecto” y “Reserva” y define el trayecto al que está asociada a la reserva.

Cardinalidad Global: (1,N) → Debido a que el trayecto puede estar asociado a ninguna o varias reservas (0,N) y la reserva está vinculada a un único trayecto (1,1).

Relación “asignar”

Es el resultado de la relación entre “Trayecto” y “Vehículo” y define el vehículo al que está asociado el trayecto.

Cardinalidad Global: (1,N) → Debido a que el vehículo puede estar asociado a ninguno o varios trayectos (0,N) y el trayecto es realizado por un único vehículo (1,1).

Relación “desarrollar”

Es el resultado de la relación entre “Trayecto” y “Horario” y define el horario en el que transcurre el trayecto.

Cardinalidad Global: (1,N) → Debido a que el horario puede estar asociado a ninguno o varios trayectos (0,N) y el trayecto es realizado durante un solo horario (1,1).

Relación “circular”

Es el resultado de la relación entre “Vehículo” y “Horario” y define los horarios en los que circula el vehículo.

Cardinalidad Global: (N,M) → Debido a que el horario puede estar asociado a ninguno o varios vehículos (0,N) y el vehículo puede trabajar en ninguno o en varios horarios (0,N).

Relación “trabajar”

Es el resultado de la relación entre “Estación” y “Horario” y define las paradas que realiza un horario.

Cardinalidad Global: (N,M) → Debido a que la estación/parada puede estar asociada a ninguno o varios horarios (0,N) y el horario puede establecer como mínimo 2 paradas/estaciones (origen y destino) o más a una determinada hora cada una de ellas (0,N).

También es necesario definir las relaciones de especialización utilizadas para aplicar la herencia de atributos y propiedades entre entidades.

“Usuario → Usuario Autenticado”

La entidad “Usuario Autenticado” hereda los atributos y relaciones propios de la entidad “Usuario” debido a que la primera requiere de otra serie de atributos y relaciones propias.

Por último, y no por ello menos importante, hay que destacar las 2 entidades que no tienen relación con ninguna otra: “Precio” y “Noticia”. Esto es debido principalmente a que su peso en el sistema no es muy alto. También hay que destacar todas las relaciones no representadas en el anterior diagrama de entidad-relación de creación, modificación y borrado propias del usuario “Administrador” sobre las siguientes entidades:

- Estación, Precio, Trayecto, Noticia, Promoción, Tipo de Abono, Horario y Vehículo.
- Sobre la entidad “Reserva” el usuario “Administrador” solo puede desarrollar tareas de borrado.
- El resto de las entidades (Usuario, Tema de Discusión, Respuesta y Abono) quedan ajenas al usuario “Administrador”.



Capítulo 5: DISEÑO DEL SISTEMA



5.1 Arquitectura

5.1.1 Arquitectura Lógica

La arquitectura lógica se define como la representación de los componentes lógicos que dan forma a un sistema junto con la relación de estos entre sí, es decir, cómo interactúan unos con otros dentro de la aplicación. La arquitectura lógica correspondiente a este sistema se encuentra modelada en la figura 15:

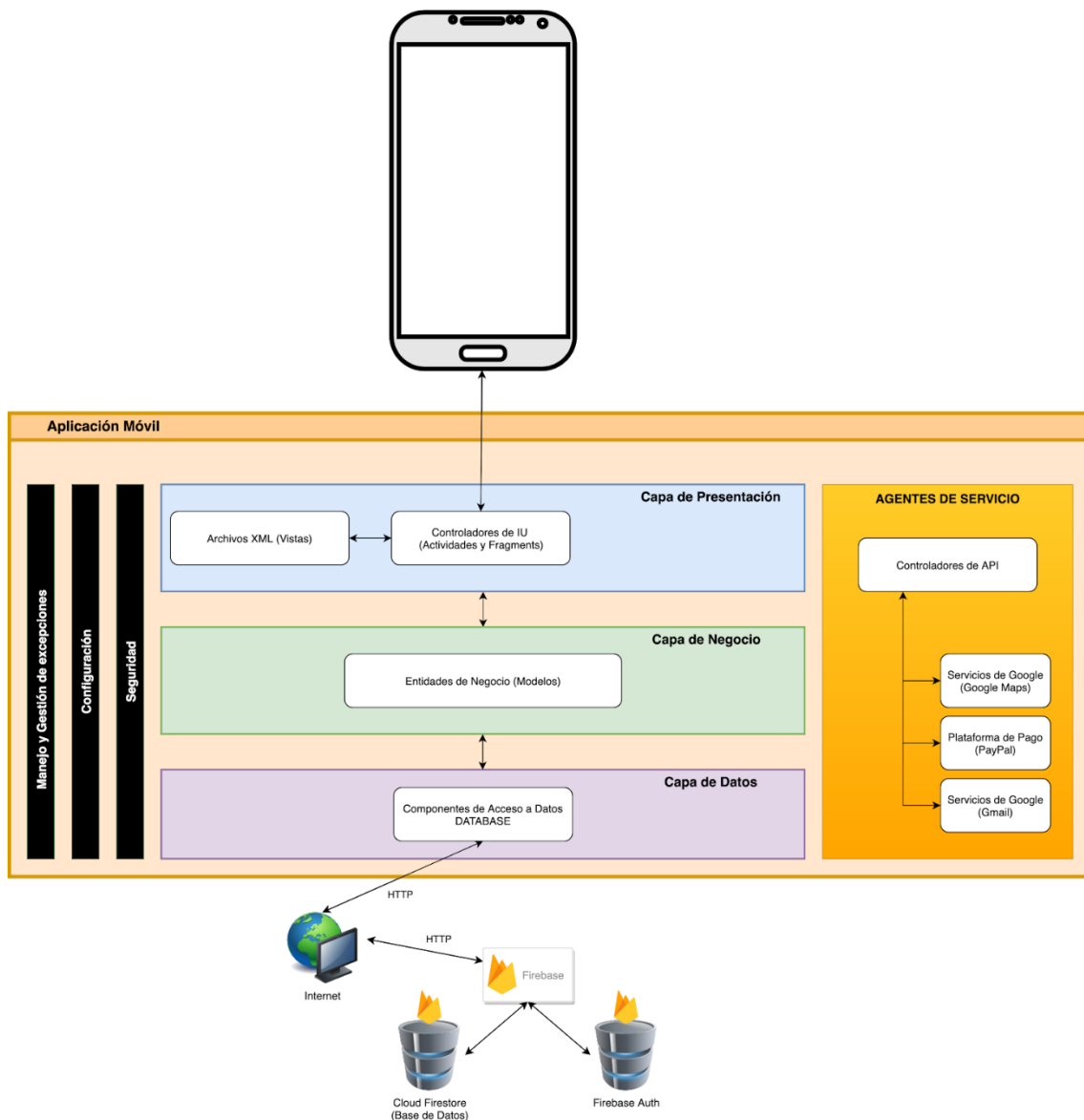


Figura 15 Arquitectura lógica

Como se puede observar, la arquitectura lógica se identifica con el modelo de tres capas (capa de presentación, capa de negocio y capa de datos) junto con una capa que identifica a los agentes de servicio que forman parte de la aplicación y que, a su vez, interactúan y se nutren de las tres capas principales. Para completar aparecen otras tres capas transversales que identifican toda la funcionalidad relacionada con la seguridad, la configuración y el manejo de excepciones.

- **Capa de Presentación:** Alberga las vistas con las que el usuario intercambia información con la aplicación. A su vez cuenta con los controladores de interfaz, cuya funcionalidad es la recuperación de datos desde las vistas, las posteriores tareas de procesamiento y posterior presentación de los resultados a los usuarios, de nuevo a través de las vistas. Estos componentes controladores también sirven como puente de esta capa de presentación con el resto de las componentes de la arquitectura.
- **Capa de Negocio:** en ella se encuentra implementada toda la lógica de negocio de la aplicación. Es necesaria para interpretar y realizar operaciones sobre los datos que se envían desde la capa de presentación y se recuperan de la capa de datos.
- **Capa de Datos:** Es la encargada de recibir peticiones por parte de la capa de negocio y los controladores e interactuar con los sistemas de almacenamiento, favoreciendo así el aislamiento de los datos a cualquier otra capa. En el caso de esta aplicación quedará definida por la instancia generada para acceder a la base de datos.
- **Agentes de Servicios:** representa todos los servicios que interactúan de cualquier forma con el sistema. En este caso, estos servicios trabajan junto con las tres capas principales, aunque siempre serán los controladores de interfaz los encargados de solicitar estos servicios.

Una vez realizado este análisis, podemos identificar una arquitectura que se asemeja en gran medida a la arquitectura modelo-vista-controlador (MVP). En la figura 16 se pueden observar sus componentes:

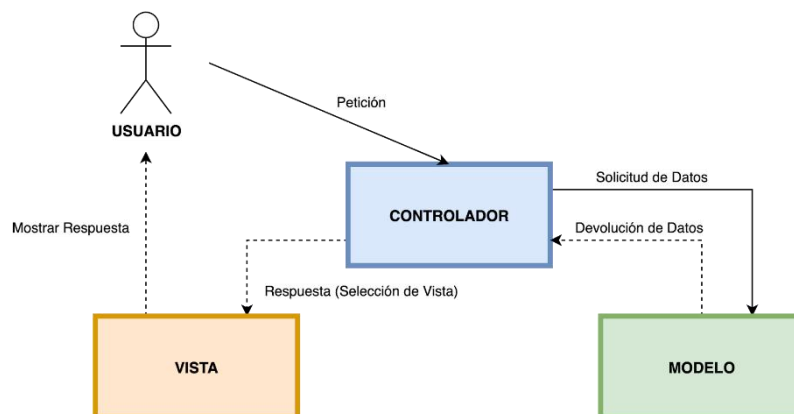


Figura 16 Patrón MVC (Modelo-Vista-Controlador)

- “Vista” representa a las vistas XML de la aplicación.
- “Controlador” representa a los controladores encargados de realizar el procesamiento de datos (actividades y fragmentos) e intermediarios entre los datos y la presentación de estos (vistas).
- “Modelo” representa a las capas de negocio y datos de la aplicación.

5.1.2 Arquitectura Física

La arquitectura física trata de describir la estructura o la forma en la que se encuentra implementada la tecnología utilizada, es decir, define los elementos físicos que requiere la arquitectura lógica para lograr cumplir con sus objetivos como aparece reflejado en la figura 17:

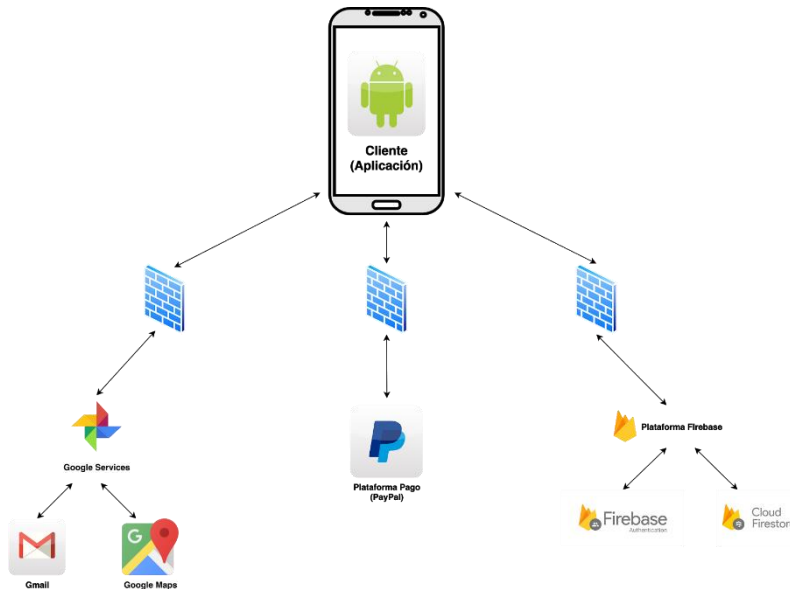


Figura 17 Arquitectura física

En esta aplicación, la arquitectura física queda definida por un patrón cliente-servidor. El cliente estará definido por el dispositivo móvil utilizado por el usuario. El lado del servidor en cambio se encuentra definido por tres componentes distintos:

- **Google Services:** a los cuales se accederá a través de las propias API de los servicios de Gmail y Maps.
- **PayPal:** se accede a través de la API propia de la plataforma.
- **Firebase:** se accede a ella (y a sus servicios) a través de las API de Firebase, Cloud Firestore y Firebase Auth. Es en esta plataforma donde se almacenan y gestionan todos los datos relacionados con las aplicaciones y los usuarios registrados en el sistema.

Como se puede observar, todas las conexiones y comunicaciones entre el lado cliente y servidor quedan protegidas a través de un cortafuegos o “firewall”, que permite dotar de seguridad y filtrado a las peticiones entrantes y salientes por parte de la aplicación.

5.2 Modelos de diseño

Para conocer en profundidad el diseño del sistema se utilizarán un conjunto de herramientas representativas denominadas modelos que se encuentran definidos en los siguientes apartados.

5.2.1 Diagrama de Actividades

Este tipo de diagrama está compuesto por los distintos procesos que se desarrollan a lo largo de un procedimiento (denominados actividades) y su propósito es representar la secuencia en la que se llevan a cabo estas actividades para conseguir desempeñar una funcionalidad de forma adecuada. Para ello, se representa el paso entre estas actividades distinguiendo cuales de estas se realizan secuencialmente o en paralelo para definir la lógica de la funcionalidad que representan. A continuación, se muestran los diagramas referentes a las funciones más representativas de la aplicación como son la compra de billetes normal, la adquisición de billetes utilizando un abono asociado y la compra de abonos.

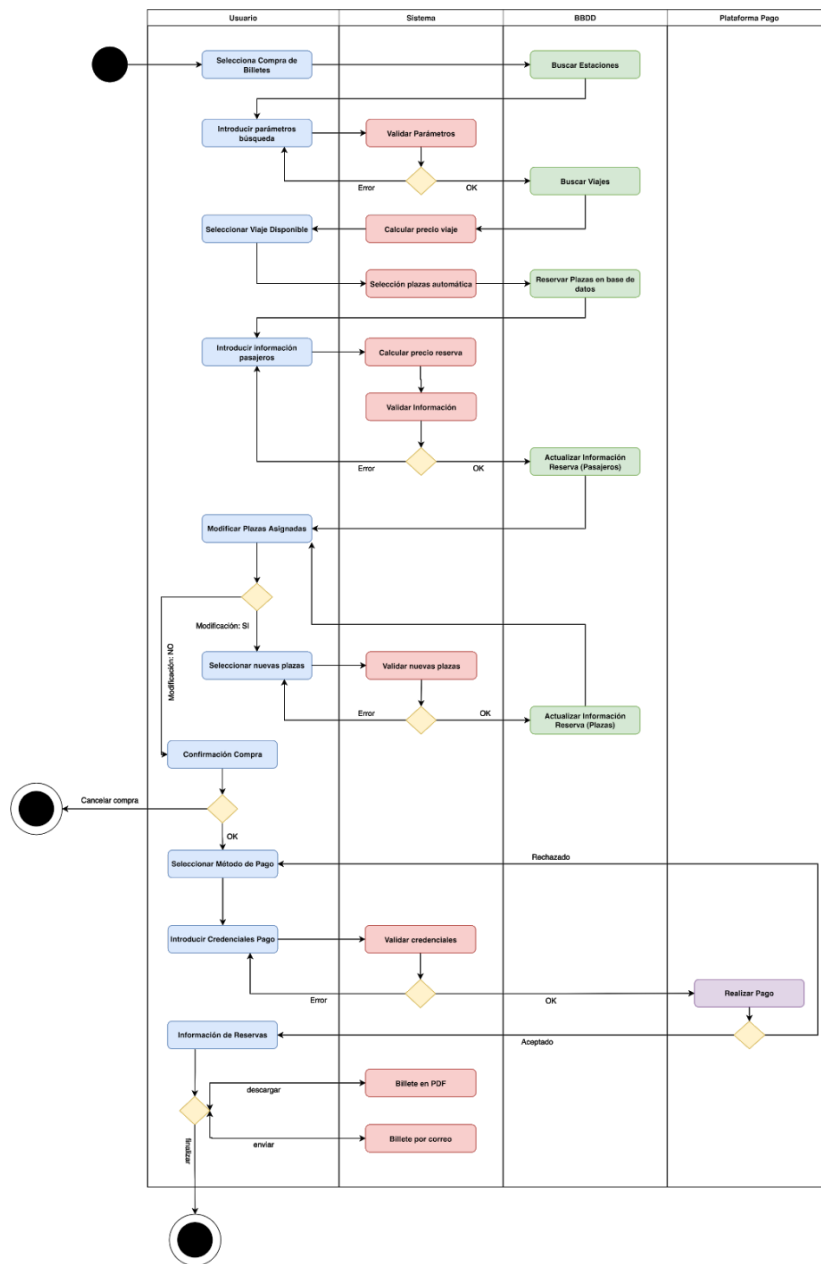


Figura 18 Diagrama de actividades "Compra de Billetes sin Abono"

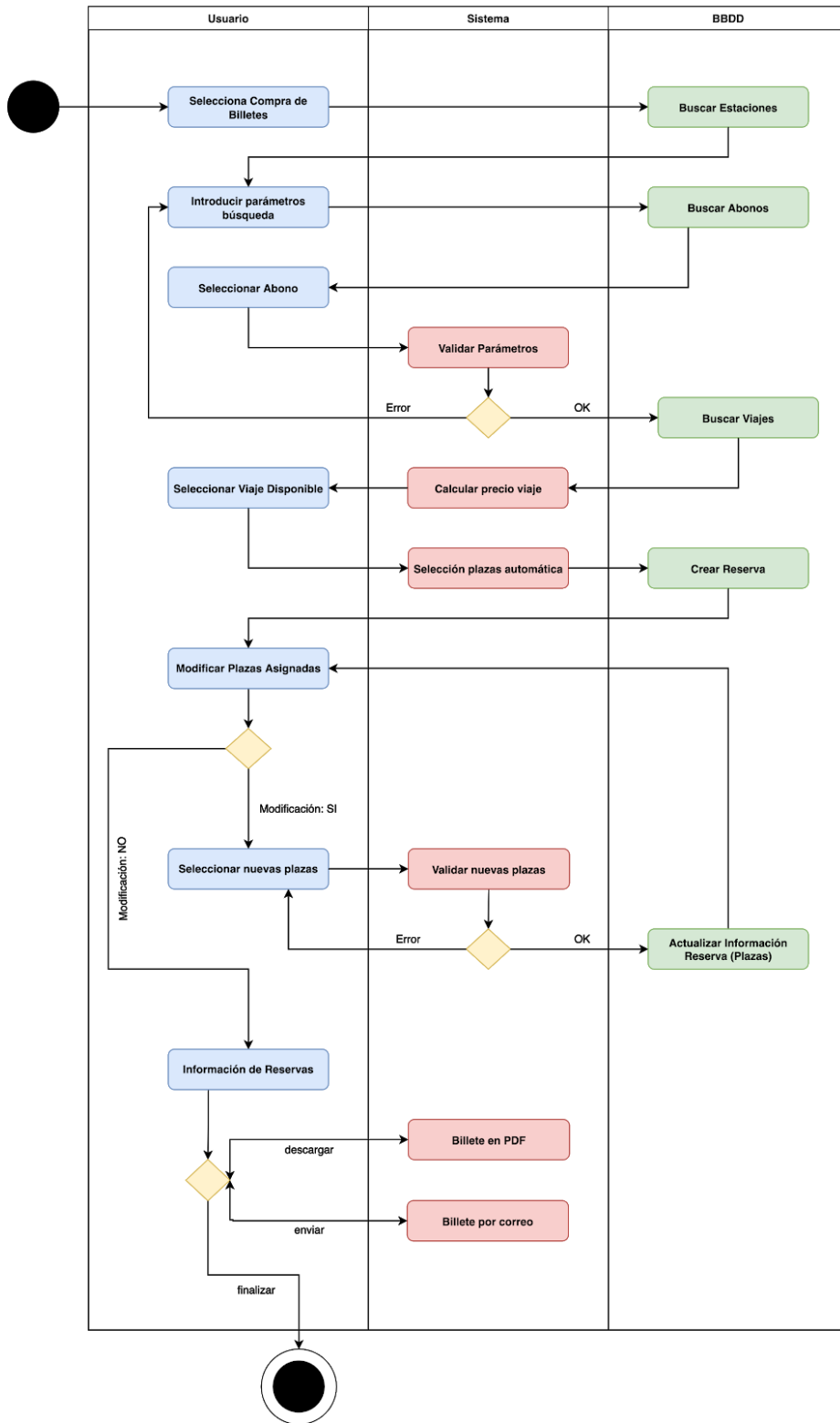


Figura 19 Diagrama de actividades "Compra de Billetes con Abono"

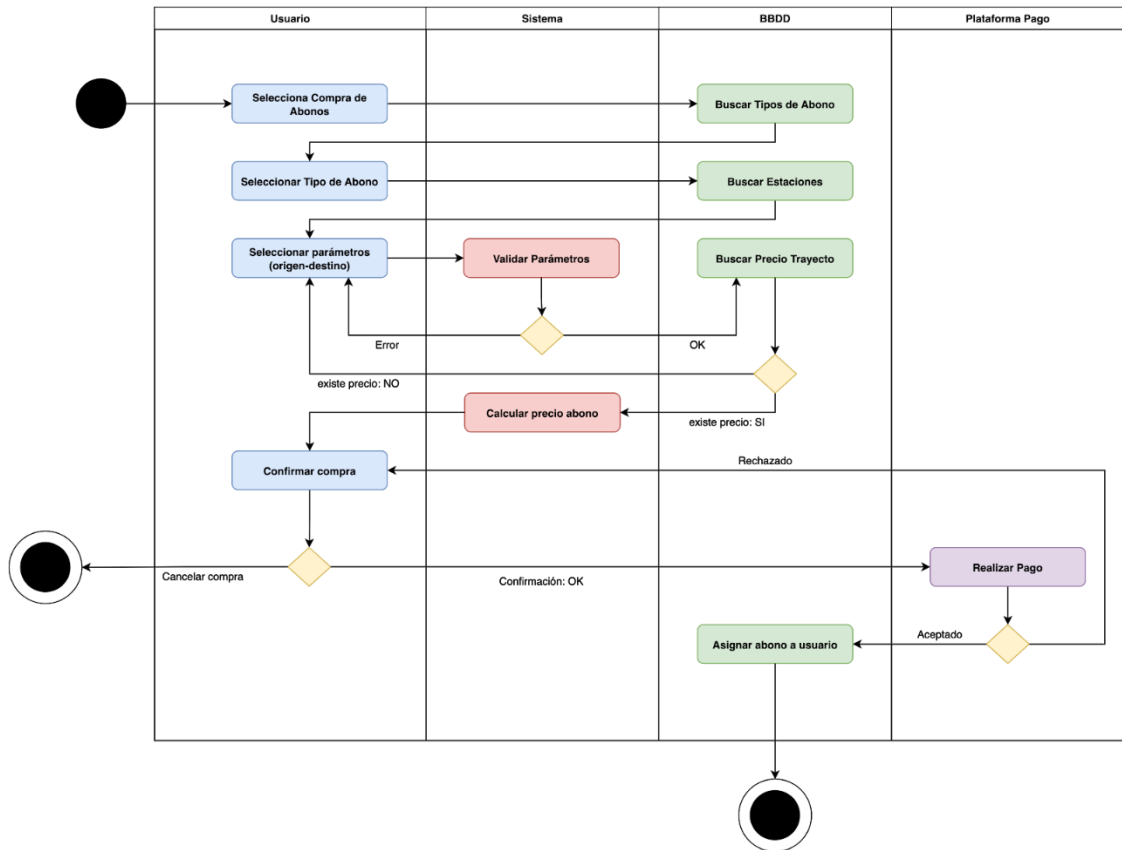


Figura 20 Diagrama de actividades "Compra de Abonos"

5.2.2 Diagramas de Secuencia

Este tipo de diagrama se encarga de representar las transacciones de mensajes entre los distintos objetos que intervienen en un escenario o conjunto de escenarios concretos. La representación de estos objetos se lleva a cabo mediante un rectángulo y una línea temporal vertical por cada uno de ellos. Una vez definidos se ordenan horizontalmente y las transacciones son representadas por líneas horizontales que unen cada una de las líneas temporales de los objetos afectados.

En este caso se ha decidido incluir una serie de diagramas que intentan englobar y representar toda la funcionalidad desarrollada para la aplicación y que no se han descrito en el apartado anterior.

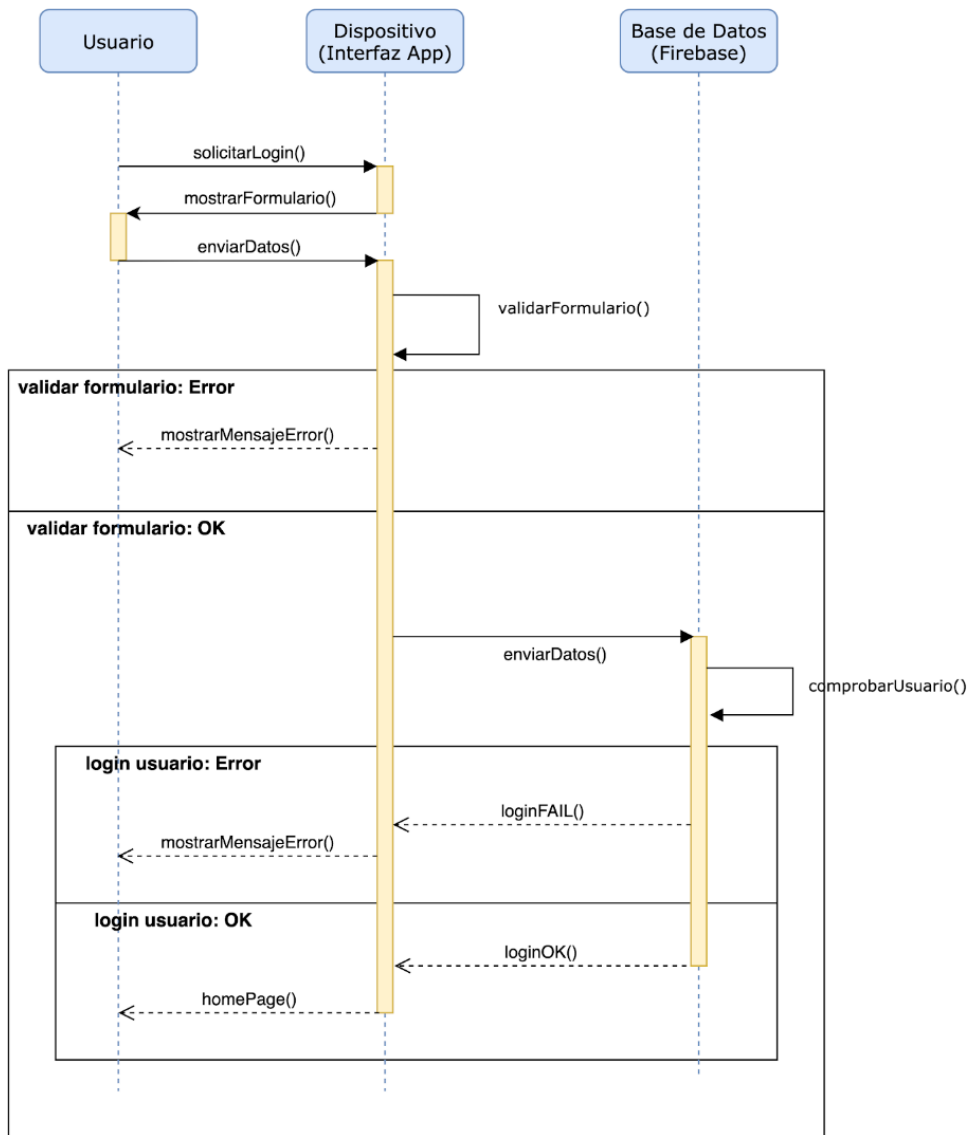


Figura 21 Diagrama de secuencia "Iniciar Sesión"

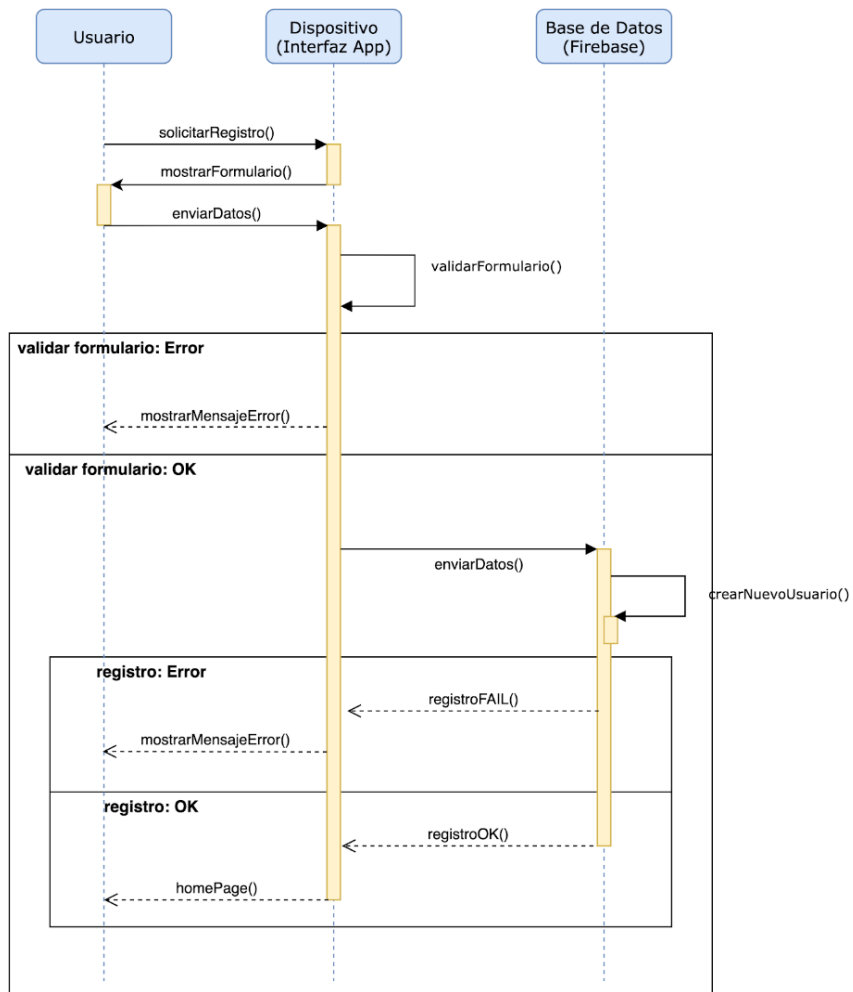


Figura 22 Diagrama de secuencia "Registro de usuario"

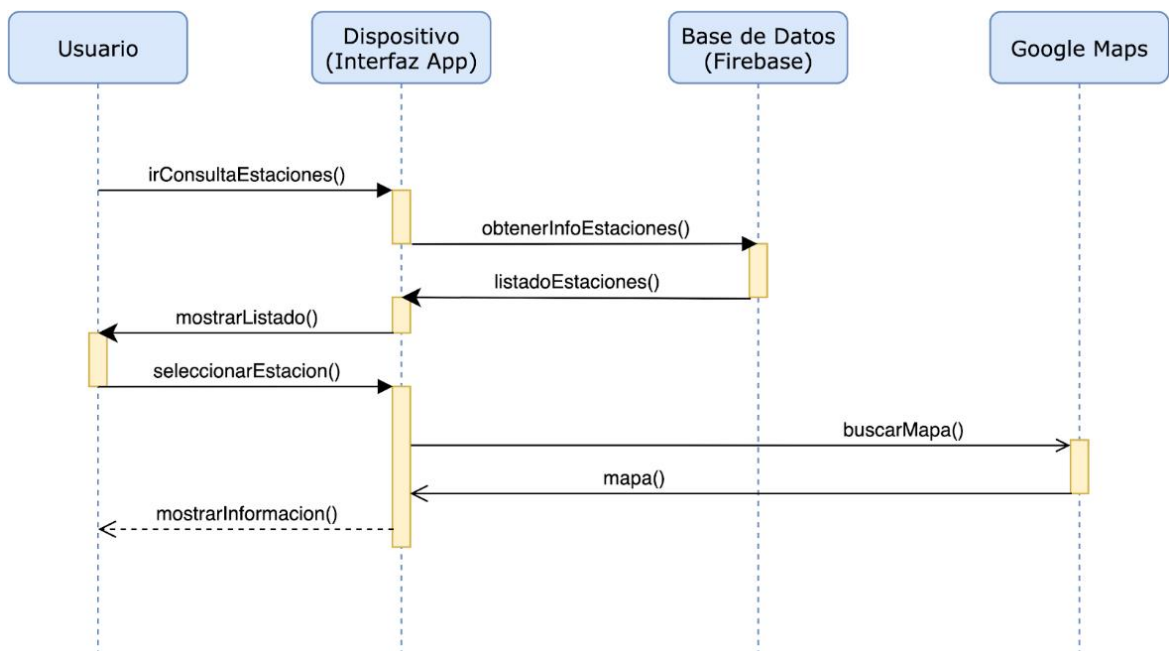


Figura 23 Diagrama de secuencia "Consultar Información de Estación"

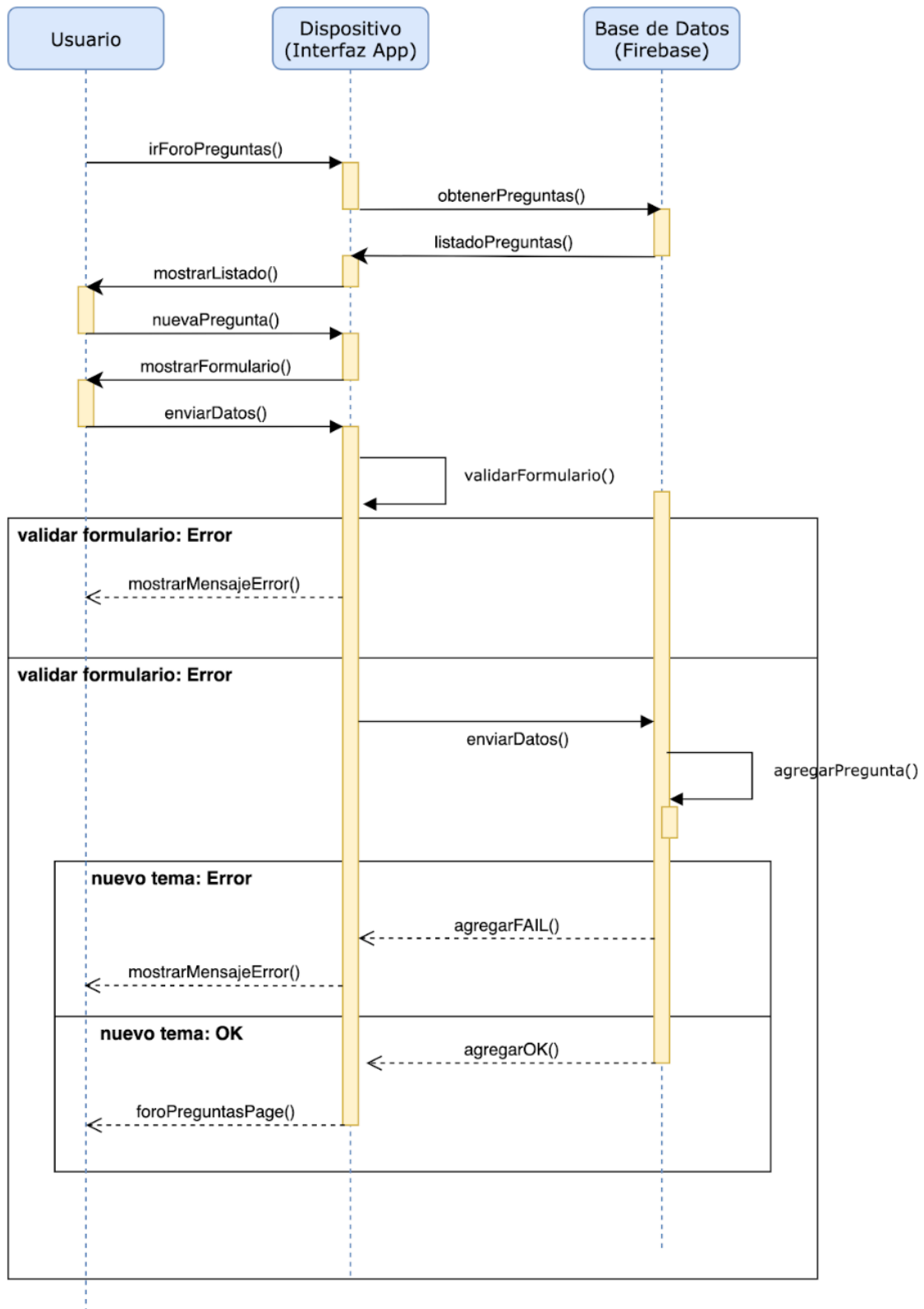


Figura 24 Diagrama de secuencia "Crear nuevo tema de discusión"

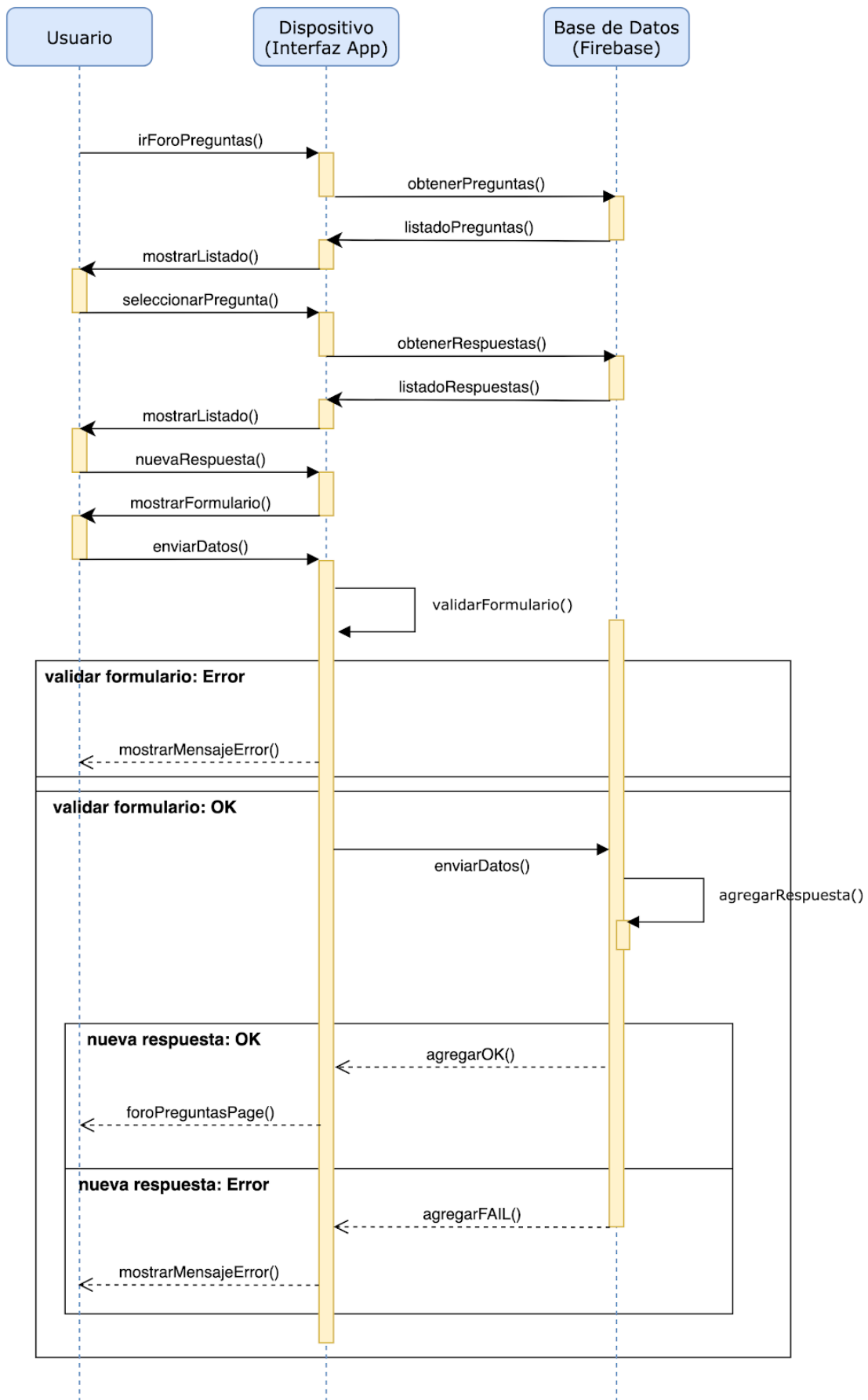


Figura 25 Diagrama de secuencia "Crear nueva respuesta"

5.2.3 Modelo Lógico de Datos

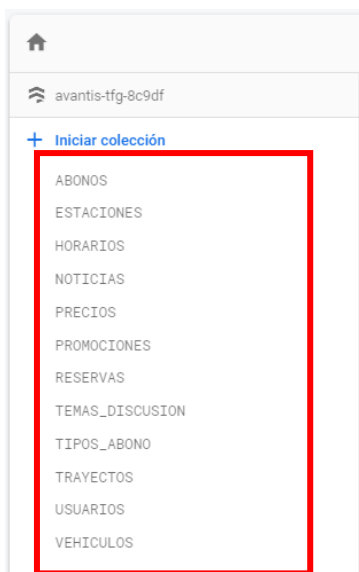
Para entender la estructura y el diseño de la base de datos, en este apartado se explicarán detalladamente estos conceptos junto con las diversas entidades que intervienen, haciendo hincapié en la tipología de base de datos que se ha utilizado: NoSQL.

Antes de desarrollar estos conceptos es necesario definir, a grandes rasgos, el proveedor de este servicio. El sistema se apoya en Firebase, una plataforma digital que proporciona a los usuarios de esta una serie de servicios destinados al desarrollo de aplicaciones (en este caso, aplicaciones móviles Android). En este caso, el servicio utilizado para otorgar a la aplicación de un servicio de base de datos se denomina Cloud Firestore. Para ampliar la información relativa a Firebase y Cloud Firestore, consultar *Anexo I* del documento.

Cloud Firestore se define por ser una base de datos no relacional (NoSQL) en contraposición a las bases de datos relacionales (o SQL). Las diferencias son muy significativas entre ambas, pero en este apartado, la que se analiza más concretamente es la relacionada con la estructura y el modelado de los datos como se puede observar en la siguiente comparativa:

Cloud Firestore (NoSQL)	Bases de datos relacionales (SQL)
Almacenamiento de la información en COLECCIONES DE DOCUMENTOS.	Almacenamiento de la información en TABLAS (filas y columnas).
Flexibilidad en el modelado de los datos (documentos compuestos por diferentes tipologías de datos)	Estricta integridad referencial entre tablas, relaciones entre tablas e índices.

En consecuencia, la estructura de los datos va a quedar definida de la siguiente forma como se muestra en la figura 26:



En ella aparecen representadas las principales colecciones de documentos que hacen referencia a las diferentes entidades que componen y nutren la base de datos.

Figura 26 Estructura jerárquica de datos en Cloud Firestore

Explorando los documentos de forma individual se puede observar cómo estos se encuentran definidos por un identificador único y campos o pares clave-valor que soportan multitud de tipologías de datos para almacenar la información.

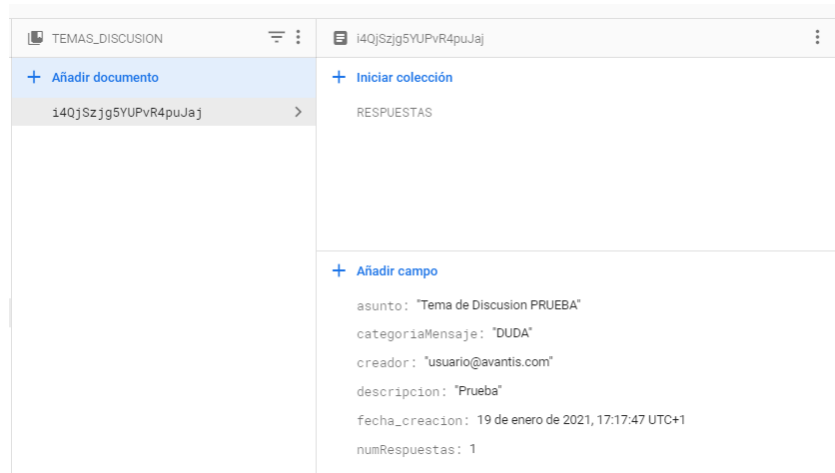


Figura 27 Estructura de documento en Cloud Firestore

Además, los documentos pueden a su vez albergar sub-colecciones de documentos (como se muestra en la figura 28) que proporcionan al sistema una estructura flexible y jerarquizada.

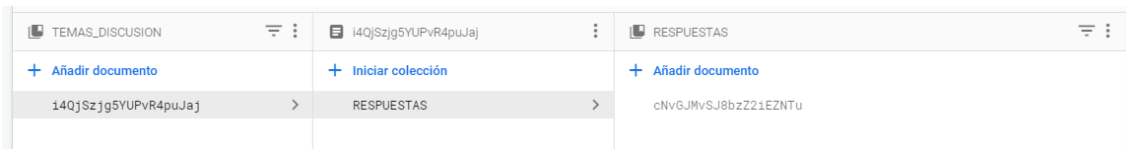


Figura 28 Sub-colecciones en Cloud Firestore

Para facilitar aún más la tediosa tarea de compaginar los modelos de datos utilizados en el código fuente con los modelos de documentos, Cloud Firestore es capaz de interpretar y traducir los modelos de código para adaptarlos a sus modelos de documento y viceversa. Esto mejora el almacenamiento, tratamiento y la consulta de los datos por parte de los desarrolladores.

Se utilizan los modelos de datos (objetos personalizados) del código para trabajar con los datos de forma “estándar” y organizada y así evitar entrar en la problemática generada por la flexibilidad a la hora de definir un documento. Es decir, todos los documentos que componen una colección vienen definidos por su objeto personalizado en el código.

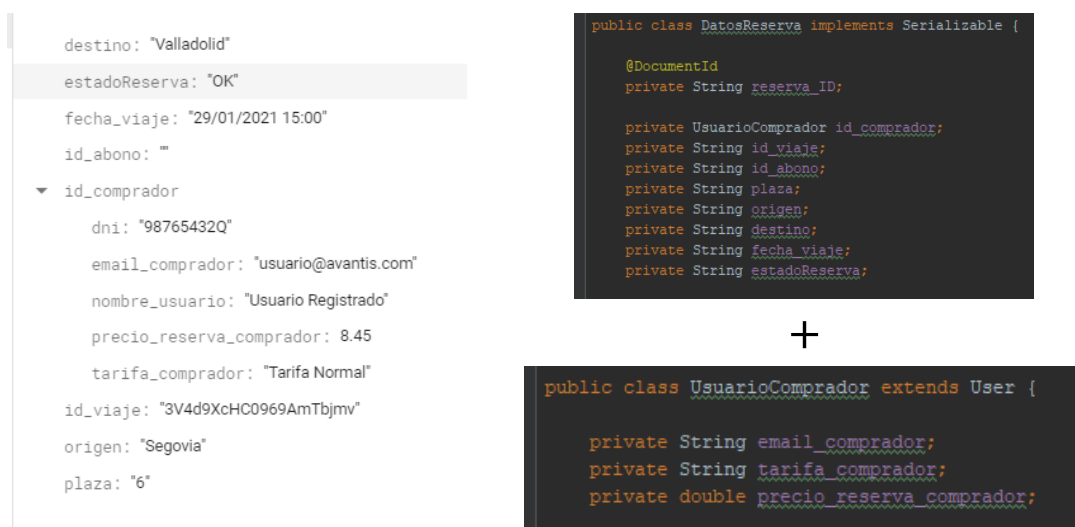


Figura 29 Relación modelos de datos (código) - modelos de datos (Cloud Firestore)

Una vez aclarados estos principios básicos de Cloud Firestore y las bases de datos no relacionales (NoSQL) y a pesar de no trabajar con bases de datos relacionales, se va a incorporar el modelo lógico de datos al proyecto. Este modelo no específico de una base de datos permite entender con más claridad las relaciones existentes entre las entidades, sus atributos, sus identificadores y ciertos aspectos para tener en cuenta.

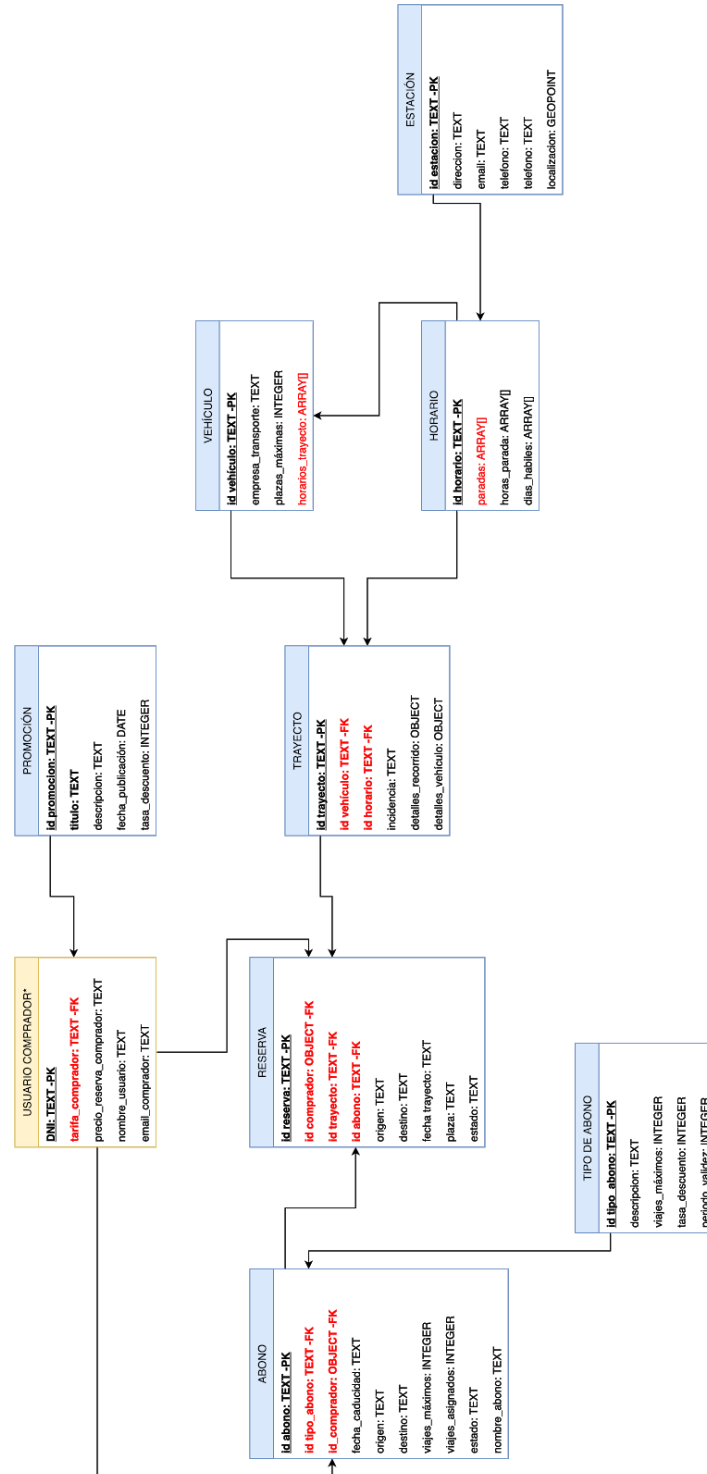


Figura 30 Modelo lógico de datos 1

Como se puede observar en la figura 30, aparece resaltada en amarillo una entidad denominada “Usuario Comprador”. Esta entidad nunca definida con anterioridad se utiliza para que el proceso de compra sea uniforme, es decir, se siga el mismo procedimiento tanto para la compra por parte de los usuarios autenticados como para la compra por parte de los usuarios anónimos.

Por otro lado, en la figura 31 aparece representado el modelo relacional propio del foro de la aplicación y las relaciones entre las entidades participantes.

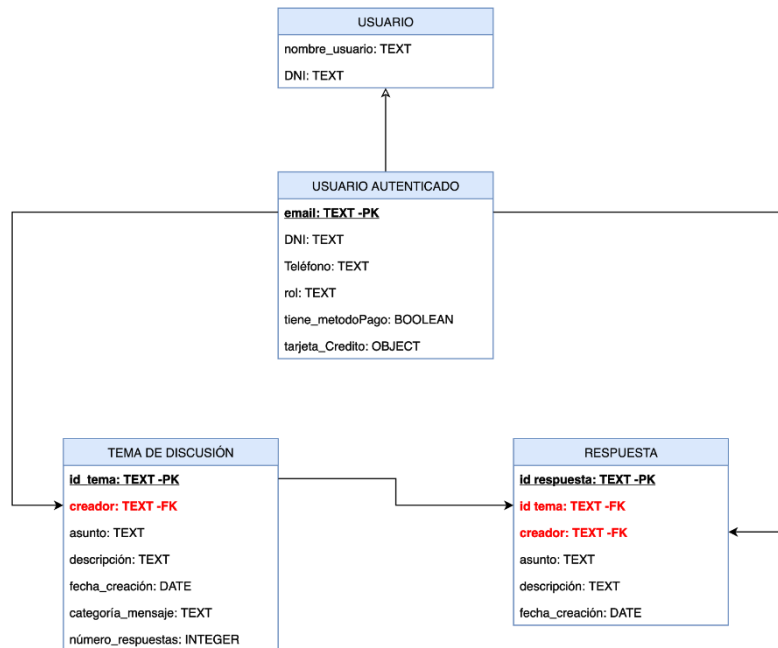


Figura 31 Modelo lógico de datos 2

5.2.4 Diagrama de Clases

Este tipo de diagrama se utiliza para la representación del dominio (todas las clases, atributos y relaciones estáticas del sistema), es decir, la estructura completa de la aplicación. El tamaño de este diagrama es tan abrumador y extenso que para poder observarlo correctamente se adjuntará al proyecto final aparte de este documento.

5.3 Diseño de Interfaces

5.3.1 Tecnologías y elementos utilizados

Material Design

Normativa de diseño compuesta por una serie de guías destinadas a la utilización de buenas prácticas en el desarrollo de aplicaciones. Fue presentada en el evento Google I/O de 2014 para dotar de estos patrones de diseño a las nuevas aplicaciones Android, aunque también es utilizado en otras plataformas como por ejemplo IOS o web. Estos patrones se basan en objetos que ocupan un espacio y un tiempo determinados, lo que se conoce como “objetos materiales”, los cuales analizan cuidadosamente la profundidad, las

superficies, los colores, las sombras y los bordes para generar diseños lo más próximos a la realidad a pesar de trabajar en un entorno digital. También se tiene en consideración el uso de animaciones y superposiciones, siempre y cuando se adapten a la lógica basada en las propias leyes de la física. Además de esto, *Material Design* son patrones de diseño multiplataforma, lo que significa que son capaces de adaptarse a todo tipo de plataformas y soluciones que se encuentran en el mercado, siendo así esta transversalidad uno de sus atractivos más valorados.

Para más información consultar:

<https://material.io/develop/android>

Constraint Layouts

Fueron incorporados en el evento de Google I/O de 2016 y posteriormente empezaron a desarrollar y a potenciar su utilización en el año 2017. Es una herramienta que ofrece un gran rendimiento en el diseño debido a que permite desarrollar estructuras complejas tratando a estas como jerarquías planas.

Esto simplifica en gran medida las tareas de dibujo de la vista por parte de Android, puesto que utilizando este tipo de *layout* no se requiere el anidamiento de vistas y elementos *viewgroups*, los cuales obligaban a recorrer el árbol completo de la vista para crear los elementos, consumiendo recursos y tiempo por parte del sistema.

Para más información consultar:

<https://developer.android.com/training/constraint-layout?hl=es>

Navigation Drawer

Patrón de navegación, convertido prácticamente en un “standard” en los últimos años destinado a simplificar la navegación entre los distintos módulos de una aplicación. En este caso estará compuesto por un menú deslizante ubicado en la parte izquierda del dispositivo que contendrá las principales rutas de navegación (funcionalidades) operativas en la aplicación. Para acceder y desplegar el mismo, el usuario deberá deslizar el dedo hacia la derecha o pulsar el icono ubicado en la esquina superior izquierda dentro de la aplicación. Una vez seleccionada una ruta de navegación, este se cierra automáticamente. Por otro lado, si se quiere cerrar una vez desplegado, es tan sencillo como pulsar el botón de retroceso del dispositivo.

Para más información consultar:

<https://developer.android.com/guide/navigation/navigation-ui?hl=es>

Recycler View


Soporte (*viewgroup*) que se encarga de procesar y mostrar los datos en forma de lista dentro de la aplicación. Su uso en el desarrollo de aplicaciones ha aumentado considerablemente debido a que la forma con la que se trabaja con los datos se ha mejorado y ofrece una mayor flexibilidad que su competidor *ListView*. Se utiliza principalmente para la visualización de grandes cantidades de datos de forma fluida ya que gestiona y limita de forma autónoma la cantidad de vistas que son vistas por parte del usuario, evitando así la sobrecarga innecesaria de vistas y datos que el usuario, por motivos físicos del dispositivo utilizado, no puede visualizar simultáneamente.

Cuenta con una librería de soporte que facilita la implementación de interfaces dinámicas que simplifiquen estas tareas previamente descritas.

Para más información consultar:

<https://developer.android.com/guide/topics/ui/layout/recyclerview?hl=es-419>

5.3.2 Interfaces

Pantalla Inicial	
Descripción	Pantalla que se muestra al iniciar la ejecución de la aplicación
Activación	<ul style="list-style-type: none"> Al iniciar la ejecución de la aplicación
Eventos	
Diseño / Interfaz de Usuario	 <p><i>Figura 32 Interfaz "Pantalla Principal"</i></p>

5.3.2.1 Menús

Menú lateral	
Descripción	Pantalla desde la cual el usuario puede acceder a toda la funcionalidad proporcionada por la aplicación
Activación	<ul style="list-style-type: none"> Pulsando el icono (esquina superior izquierda) presente en la amplia mayoría de pantallas Desplegando el menú deslizando hacia la derecha
Eventos	<ul style="list-style-type: none"> Acceso a la funcionalidad principal (pantallas) de la aplicación

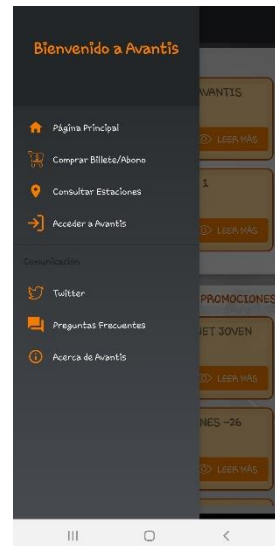



Figura 33 Interfaz "Menú lateral (usuario no autenticado)"




Figura 34 Interfaz "Menú lateral (Administrador)"



Figura 35 Interfaz "Menú lateral (usuario autenticado)"

Gestión de Información Personal	
Descripción	Pantalla que permite al usuario autenticado acceder a los distintos aspectos propios de su cuenta de usuario propia
Activación	Desde el menú lateral (esquina superior izquierda o desplegable)
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Acceso a la pantalla “Mi Perfil” • Acceso a la pantalla “Tarjeta de Crédito” • Acceso a la pantalla “Mis Viajes” • Acceso a la pantalla “Mis Abonos” • Acceso a panel de ayuda (esquina superior derecha)
Diseño / Interfaz de Usuario	 <p><i>Figura 36 Interfaz "Gestión de Información Personal"</i></p>

Gestión Sistema de Compra	
Descripción	Pantalla que permite al usuario autenticado acceder a los distintos métodos de compra disponibles en la aplicación
Activación	Desde el menú lateral (esquina superior derecha o desplegable)
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Acceso a la pantalla “Selector sentido de viaje (Compra sin Abono)”

	<ul style="list-style-type: none"> • Acceso a la pantalla “Selector sentido de viaje (Compra con Abono)” • Acceso a la pantalla “Selector tipo de abono” • Acceso a panel de ayuda (esquina superior derecha)
Diseño / Interfaz de Usuario	 <p style="text-align: center;"><i>Figura 37 Interfaz "Gestión de Sistema de Compra"</i></p>

5.3.2.2 Destinadas a funcionalidad básica

Inicio de Sesión	
Descripción	<p>Pantalla que permite a cualquier usuario acceder a su cuenta personal previamente creada en la plataforma. También permite acceder a la pantalla de registro para crear una nueva cuenta de usuario si se requiere.</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • Campos EditText para la inserción de credenciales • Button para la confirmación y envío de credenciales • Etiqueta TextView para el acceso a la pantalla “Registro de usuario”
Activación	<p>Desde el menú lateral (esquina superior izquierda o desplegable) en el caso de que previamente el usuario no cuente con una sesión activa en la plataforma.</p>
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para el envío de credenciales • Validaciones sobre campos EditText • Pulsación en TextView para el acceso a la pantalla de registro

**Diseño /
Interfaz de
Usuario**



Figura 38 Interfaz "Inicio de Sesión"

Registro de Usuario

<p>Descripción</p>	<p>Pantalla que permite crear una nueva cuenta de usuario dentro de la plataforma. También permite acceder a la pantalla de inicio de sesión en el caso de que el usuario previamente posea una cuenta de usuario. Compuesta por:</p> <ul style="list-style-type: none"> • Campos EditText para inserción de información de usuario • Button para la confirmación y envío de información • Etiqueta TextView para el acceso a la pantalla "Inicio de sesión"
<p>Activación</p>	<p>Desde la pantalla "Inicio de sesión"</p>
<p>Eventos</p>	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para el envío de información de usuario • Validaciones sobre campos EditText • Pulsación en TextView para el acceso a la pantalla "Inicio de sesión" • Mensaje informativo sobre los requerimientos de contraseña. • Mensaje informativo sobre el resultado de la creación de un nuevo usuario

**Diseño /
Interfaz de
Usuario**



Figura 39 Interfaz "Registro de usuario"

Home / Pantalla Principal

<p>Descripción</p>	<p>Pantalla que permite visualizar las últimas noticias y promociones disponibles para los usuarios de la plataforma Compuesta por:</p> <ul style="list-style-type: none"> • RecyclerView (ítem + Button) para la visualización de las últimas noticias • RecyclerView (ítem + Button) para la visualización de las promociones disponibles
<p>Activación</p>	<p>Pantalla inicial de la aplicación</p>
<p>Eventos</p>	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para acceder a la información de una noticia (individual) • Pulsación sobre Button para acceder a la información de una promoción (individual) • Información de noticia (individual) • Acceso a panel de ayuda (pulsación en esquina superior derecha)

Diseño / Interfaz de Usuario



Figura 40 Interfaz "Home / Pantalla principal"

Acerca de Avantis

Descripción	AlertDialog que muestra a los usuarios información de carácter general de la aplicación.
Activación	Desde el menú lateral (esquina superior izquierda o desplegable)
Eventos	<ul style="list-style-type: none"> • Pulsación para cerrar la ventana de información

Diseño / Interfaz de Usuario

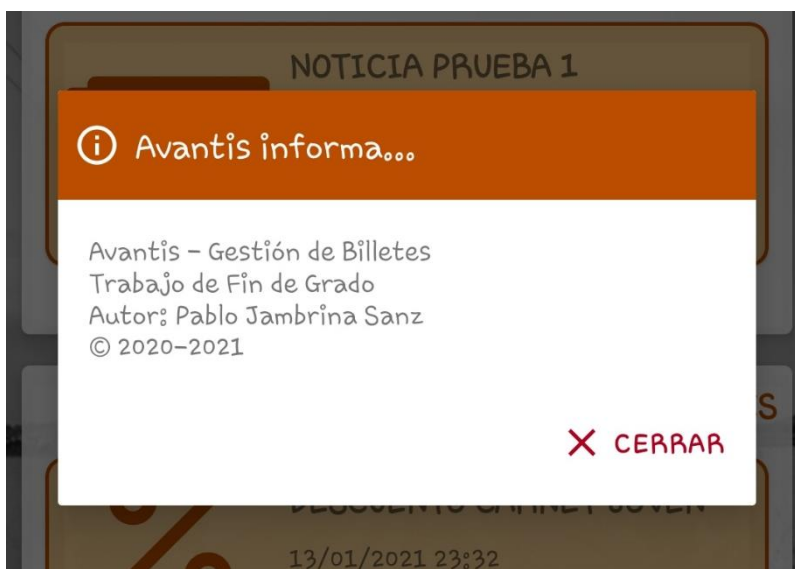




Figura 41 Interfaz "Acerca de Avantis"

Foro de discusión	
Descripción	<p>Pantalla que muestra a los usuarios los distintos temas de discusión propuestos en el foro de la plataforma.</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • RecyclerView (ítem + Button) para la visualización de los temas de discusión propuestos
Activación	Desde el menú lateral (esquina superior izquierda o desplegable)
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para acceder a los detalles y respuestas de un tema de discusión (individual) • Acceso a panel de ayuda (pulsación en esquina superior derecha) • Acceso a la pantalla “Crear nuevo tema de discusión” (pulsar icono en barra superior derecha)
Diseño / Interfaz de Usuario	<div style="text-align: center;">  </div> <p style="text-align: center;"><i>Figura 42 Interfaz "Foro de discusión"</i></p>

Crear nuevo tema de discusión	
Descripción	<p>Pantalla que muestra al usuario el formulario que debe completar para la creación de un nuevo tema de discusión en el foro.</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • Campos EditText de formulario • Button para la confirmación y envío de información
Activación	Desde la pantalla de “Preguntas Frecuentes”

Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para el envío e inserción de tema • Validaciones sobre campos EditText
Diseño / Interfaz de Usuario	 <p data-bbox="730 949 1230 976"><i>Figura 43 Interfaz "Crear nuevo tema de discusión"</i></p>

Detalles Tema de Discusión	
Descripción	<p>Pantalla que muestra la información relativa a un tema de discusión (descripción y respuestas), junto con la posibilidad de crear nuevas respuestas para dicho tema</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • CardView con los detalles del tema de discusión • RecyclerView para la visualización de las respuestas publicadas • Button para acceder a pantalla "Crear nueva respuesta"
Activación	Desde la pantalla de "Preguntas Frecuentes"
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre "Nueva Respuesta" para acceder a la pantalla "Crear nueva respuesta" • Acceso a panel de ayuda (pulsación en esquina superior derecha)

Diseño / Interfaz de Usuario

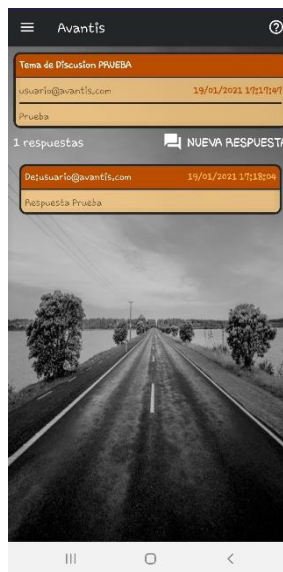


Figura 44 Interfaz "Detalles de tema de discusión"

Crear nueva respuesta	
Descripción	<p>Pantalla que muestra al usuario el formulario que debe completar para la creación de una nueva respuesta para el tema propuesto en el foro. Compuesta por:</p> <ul style="list-style-type: none"> • Campos EditText de formulario • Button para la confirmación y envío de información
Activación	Desde la pantalla de "Detalles Tema de Discusión"
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para el envío e inserción de tema • Validaciones sobre campos EditText

Diseño / Interfaz de Usuario



Figura 45 Interfaz "Crear nueva respuesta"

Consulta Estaciones

Descripción	<p>Pantalla que muestra a los usuarios un listado con las posibles estaciones que se pueden consultar. Compuesta por:</p> <ul style="list-style-type: none">• RecyclerView para la visualización de las estaciones disponibles
Activación	<p>Desde el menú lateral (esquina superior izquierda o desplegable)</p>
Eventos	<ul style="list-style-type: none">• Acceso al menú lateral (esquina superior izquierda o desplegable)• Pulsación sobre ítems para acceder a la pantalla "Detalles Estación"• Acceso a filtro de búsqueda de estación (pulsación en esquina superior derecha)

Diseño / Interfaz de Usuario



Figura 46 Interfaz "Consulta de estaciones"

Detalles Estación

Descripción	<p>Pantalla que muestra la información relativa a una estación, junto a su ubicación a través del servicio de mapas de Google.</p> <p>Compuesta por:</p> <ul style="list-style-type: none">• CardView para la visualización de la información de la estación
Activación	Desde la pantalla de "Consulta Estaciones"
Eventos	<ul style="list-style-type: none">• Acceso al menú lateral (esquina superior izquierda o desplegable)• Gestión del posicionamiento del mapa (zoom, ubicación, ...)

Diseño / Interfaz de Usuario

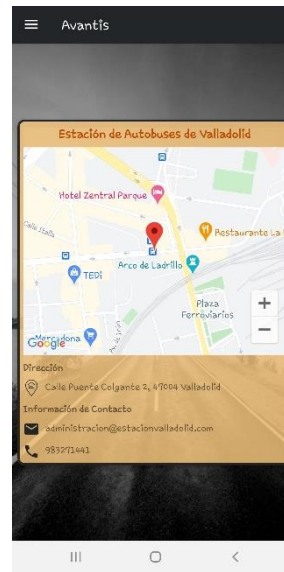


Figura 47 Interfaz "Detalles de estación"

Mensajes de Solicitud de Confirmación

Descripción	AlertDialog que muestra a los usuarios una confirmación de solicitud
Activación	Cada vez que el sistema requiera de la confirmación de una orden por parte del usuario
Eventos	<ul style="list-style-type: none"> • Pulsación para cerrar la ventana de información • Pulsación para confirmar la orden y cerrar la ventana de información

Diseño / Interfaz de Usuario

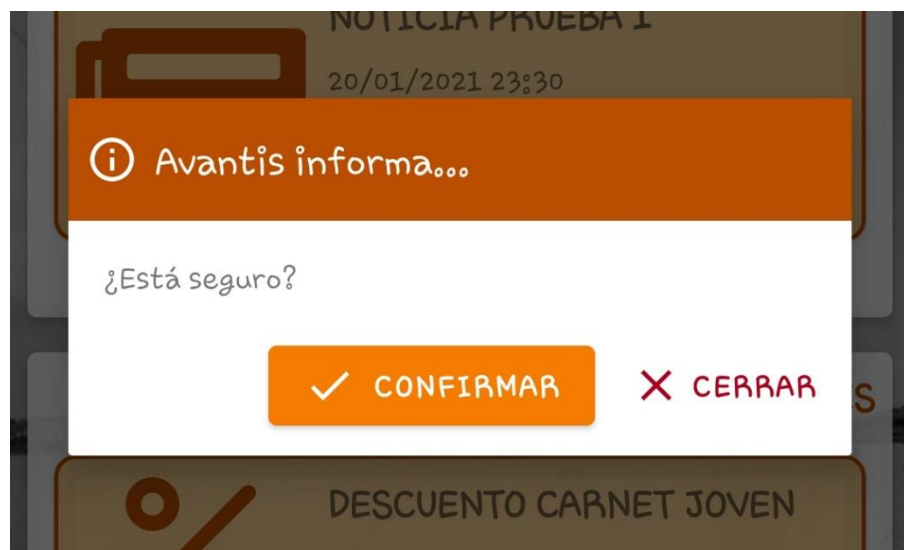
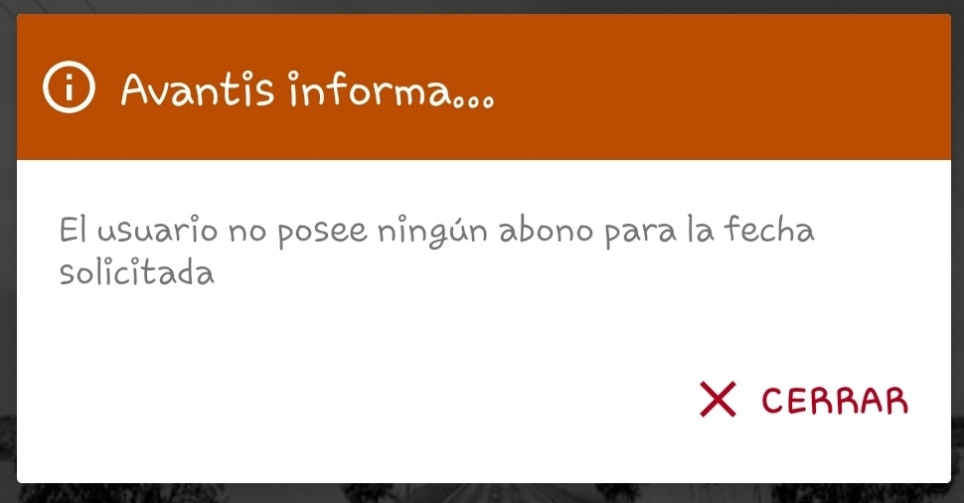


Figura 48 Interfaz "Mensajes de solicitud de confirmación"

Mensajes de Información	
Descripción	AlertDialog que muestra a los usuarios información relevante sobre algún aspecto de funcionalidad.
Activación	Cada vez que el sistema lo requiera
Eventos	<ul style="list-style-type: none"> • Pulsación para cerrar la ventana de información
Diseño / Interfaz de Usuario	 <p style="text-align: center;"><i>Figura 49 Interfaz "Mensaje de información"</i></p>

Manual de Usuario	
Descripción	AlertDialog que muestra a los usuarios un panel de ayuda con información relevante de la funcionalidad en la que se encuentran
Activación	Desde la barra superior (esquina derecha)
Eventos	<ul style="list-style-type: none"> • Pulsación para cerrar el panel de ayuda

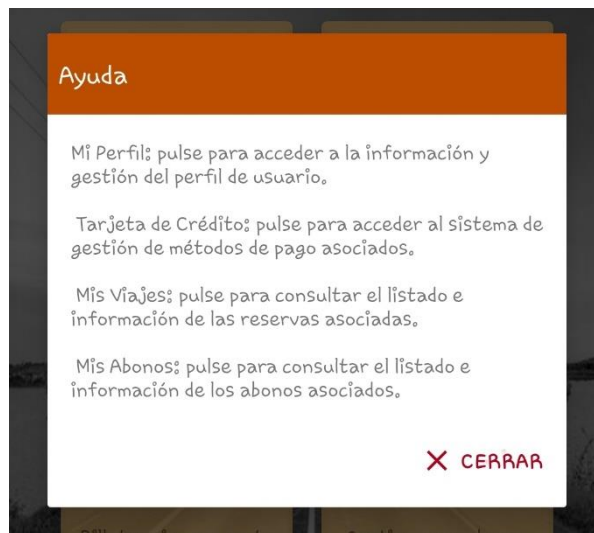


Figura 50 Interfaz "Manual de usuario 1"



Figura 51 Interfaz "Manual de usuario 2"

5.3.2.3 Destinadas a los sistemas de compra


Selector parámetros de viaje (Compra sin Abono)

Descripción

Pantalla que permite a los usuarios la selección de los parámetros de consulta de un viaje.

Compuesta por:

- Campos EditText para selección de origen y destino
- Checkbox para la selección del tipo de viaje (Ida / Ida y Vuelta)
- EditText "Ida y Vuelta" para la selección de fechas de viaje
- 2 Button para la selección del número de pasajeros
- Button "Búsqueda" para enviar los parámetros de búsqueda

Activación	Desde el menú “Gestión Sistema de Compra”
Eventos	<ul style="list-style-type: none"> • Volver (pulsar icono esquina superior izquierda) • Acceso a panel de ayuda (esquina superior derecha) • Pulsación sobre Button “Búsqueda” para solicitar la petición • Validaciones sobre campos EditText • Pulsación sobre Checkbox para mostrar/ocultar el selector de fecha de vuelta • Pulsación sobre los campos EditText “Ida y Vuelta” para desplegar los calendarios selectores de fechas (DatePicker)
Diseño / Interfaz de Usuario	 <p><i>Figura 52 Interfaz "Selector parámetros de viaje (Compra sin Abono)"</i></p>

Resultado Búsqueda de Viaje	
Descripción	<p>Pantalla que muestra a los usuarios un listado con los posibles viajes disponibles para los parámetros especificados. Compuesta por:</p> <ul style="list-style-type: none"> • CardView con los parámetros de viaje seleccionados • RecyclerView (item + Button) para la visualización de los viajes disponibles
Activación	A continuación de pulsar el botón “Búsqueda” en la pantalla “Selector parámetros de viaje” y validar la petición de búsqueda
Eventos	<ul style="list-style-type: none"> • Volver al paso anterior (pulsar icono esquina superior izquierda) • Acceso a panel de ayuda (esquina superior derecha) • Pulsación sobre Button para seleccionar el viaje

Diseño / Interfaz de Usuario



Figura 53 Interfaz "Resultado Búsqueda de Viaje"

Información de los Pasajeros (Compra sin Abono)

<p>Descripción</p>	<p>Pantalla que muestra a los usuarios un formulario para introducir los datos de los pasajeros en la compra. Compuesta por:</p> <ul style="list-style-type: none"> • CardView con los parámetros de viaje seleccionados • RecyclerView (item + formulario (EditText, Spinner)) para visualizar e incorporar los datos de los pasajeros • Button para confirmar los datos de los pasajeros
<p>Activación</p>	<p>A continuación de seleccionar el viaje en la pantalla “Resultados Búsqueda de viaje”</p>
<p>Eventos</p>	<ul style="list-style-type: none"> • Volver al paso anterior (pulsar icono esquina superior izquierda) • Acceso a panel de ayuda (esquina superior derecha) • Pulsación sobre Button para confirmar la información de los pasajeros • Validaciones de los campos EditText • Calcular precio en función de la tarifa/promoción seleccionada para cada pasajero

Diseño / Interfaz de Usuario

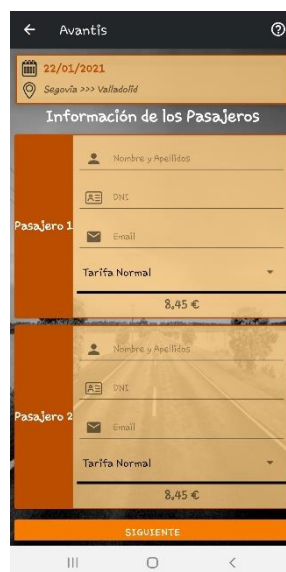


Figura 54 Interfaz "Información de los Pasajeros (Compra sin Abono)"

Servicios Adicionales de Viaje

<p>Descripción</p>	<p>Pantalla que otorga a los usuarios la posibilidad de modificar aspectos relativos a su viaje (selección de asiento). Compuesta por:</p> <ul style="list-style-type: none"> • CardView con los parámetros de viaje seleccionados • CardView + Button para modificar los asientos del viaje de ida • CardView + Button para modificar los asientos del viaje de vuelta • Button “Continuar” para avanzar con el proceso de compra de billetes
<p>Activación</p>	<p>A continuación de agregar la información de los pasajeros en la pantalla “Información de Pasajeros”</p>
<p>Eventos</p>	<ul style="list-style-type: none"> • Volver al paso anterior (pulsar icono esquina superior izquierda) • Acceso a panel de ayuda (esquina superior derecha) • Pulsación sobre Button “Continuar” para avanzar en el proceso de compra • Pulsación sobre cualquiera de los Button de CardView para solicitar la selección de asiento

Diseño / Interfaz de Usuario

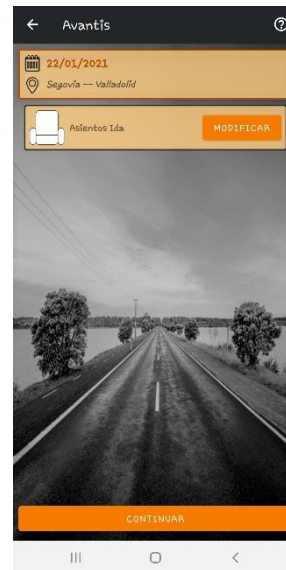
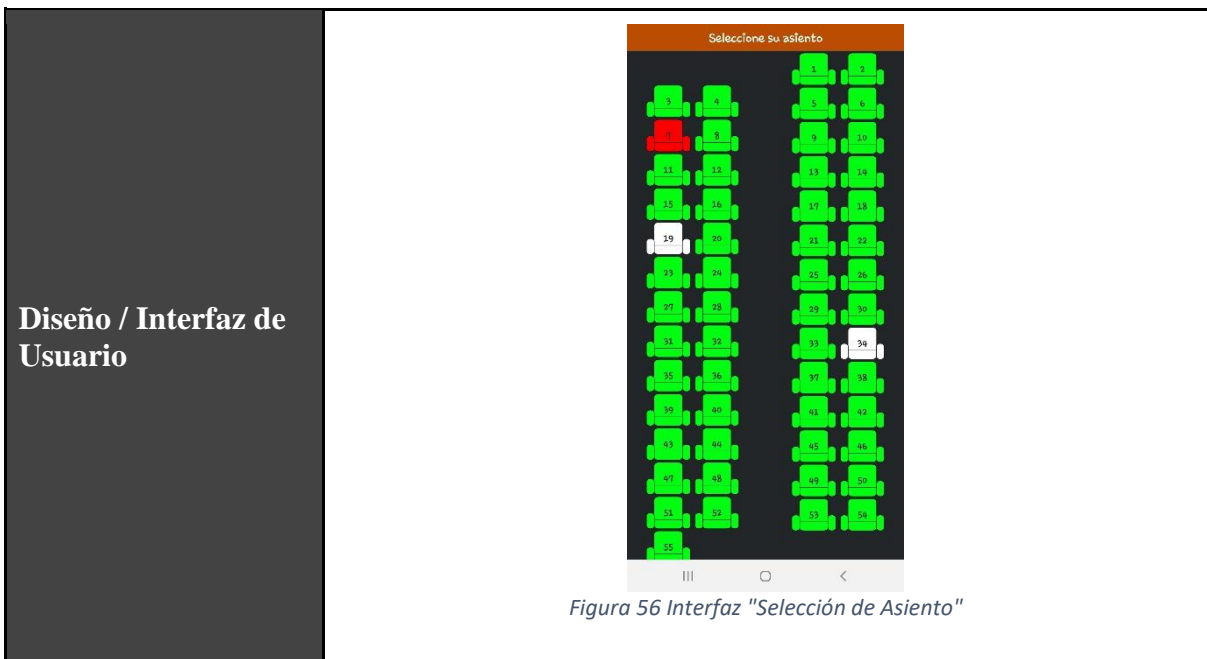



Figura 55 Interfaz "Servicios Adicionales de Viaje"

Selección de Asiento


Descripción	<p>Pantalla que posibilita a los pasajeros la selección personalizada del asiento para su viaje. Compuesta por:</p> <ul style="list-style-type: none">• Conjunto de Button que representan los asientos disponibles y ocupados para el viaje• Button "Confirmar" para la confirmación de operación
Activación	<p>Posteriormente a la solicitud de selección de asiento en la pantalla "Servicios Adicionales de Viaje"</p>
Eventos	<ul style="list-style-type: none">• Pulsación sobre cualquiera de los asientos disponibles (Button) para seleccionar el asiento• Pulsación sobre Button "Confirmar" para actualizar la información de compra



<p align="center">Resumen de Viaje</p>	
<p>Descripción</p>	<p>Pantalla muestra a los usuarios un resumen de la compra, junto con un selector del método de pago que se utilizará.</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • Layout desplegable para la visualización de la información de compra • RadioButton para la selección del método de pago • Formulario (EditText) para introducir los datos de la tarjeta de crédito • Button “Realizar Pago” para completar el proceso de compra • Button “Cancelar Compra” para cancelar el proceso de compra
<p>Activación</p>	<p>Posteriormente después de la pantalla “Servicios Adicionales de Viaje”</p>
<p>Eventos</p>	<ul style="list-style-type: none"> • Volver al paso anterior (pulsar icono esquina superior izquierda) • Acceso a panel de ayuda (esquina superior derecha) • Pulsación sobre Layout para visualizar el resumen detallado de la compra • Pulsación sobre RadioButton para seleccionar método de pago • Pulsación sobre “Realizar Pago” para continuar con el proceso de compra • Pulsación sobre “Cancelar Compra” para salir del proceso de compra

	<ul style="list-style-type: none"> • Formulario (EditText) para introducir datos de tarjeta de crédito • Validación de campos EditText
Diseño / Interfaz de Usuario	 <p style="text-align: center;"><i>Figura 57 Interfaz "Resumen de Viaje"</i></p>

Finalización Compra Billetes	
Descripción	<p>Pantalla muestra a los usuarios el resultado final del proceso de compra</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • Layout desplegable para la visualización de la información billete (individual) • Button (individual) para descargar el billete comprado en formato PDF • Button (individual) para enviar el billete al correo electrónico del pasajero • Button “Menu Principal” para volver al menú principal de la aplicación
Activación	<p>Posteriormente a la realización del pago (a través de PayPal)</p>
Eventos	<ul style="list-style-type: none"> • Pulsación sobre Layout para visualizar el resumen detallado del billete • Pulsación sobre Button para descargar el billete en formato PDF • Pulsación sobre Button para enviar el billete por correo electrónico • Pulsación sobre Button “Menu Principal” para acceder al menú principal

	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable)
<p>Diseño / Interfaz de Usuario</p>	 <p><i>Figura 58 Interfaz "Finalización Compra Billetes"</i></p>

Selector tipo de abono	
<p>Descripción</p>	<p>Pantalla que permite seleccionar el tipo de abono que el usuario desea adquirir</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • RecyclerView (Item + Button) con los distintos tipos de abono disponibles
<p>Activación</p>	<p>Desde el menú "Gestión Sistema de Compra"</p>
<p>Eventos</p>	<ul style="list-style-type: none"> • Pulsación sobre item (Button) para seleccionar el tipo de abono • Acceso al menú lateral (esquina superior izquierda o desplegable) • Acceso a panel de ayuda (esquina superior derecha)

Diseño / Interfaz de Usuario



Figura 59 Interfaz "Selector tipo de abono"

Selector parámetros de Abono	
Descripción	<p>Pantalla que permite seleccionar el trayecto en el proceso de compra de abono</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • Campos EditText para selección de origen y destino (trayecto) • TextView para mostrar el precio del abono • Button "Confirmar Compra" para continuar con el proceso de compra
Activación	<p>Posteriormente a seleccionar el tipo de abono que se desea adquirir en la pantalla "Selector tipo de abono"</p>
Eventos	<ul style="list-style-type: none"> • Pulsación sobre Button "Confirmar Compra" continuar con el proceso de compra de abono • Validaciones sobre campos EditText • Actualización del campo TextView con la información del precio de abono

Diseño / Interfaz de Usuario



Figura 60 Interfaz "Selector parámetros de Abono"

Finalización Compra Abono

Descripción	<p>Pantalla muestra a los usuarios el resultado final del proceso de compra de abono Compuesta por:</p> <ul style="list-style-type: none">• Button “Mis Abonos” para acceder a la colección de abonos adquiridos por el usuario previamente• Button “Menu Principal” para volver al menú principal de la aplicación
Activación	<p>Posteriormente a la realización del pago (a través de PayPal)</p>
Eventos	<ul style="list-style-type: none">• Pulsación sobre Button “Menu Principal” para acceder al menú principal• Pulsación sobre Button “Mis abonos” para acceder a la colección de abonos del usuario• Acceso al menú lateral (esquina superior izquierda o desplegable)

Diseño / Interfaz de Usuario



Figura 61 Interfaz "Finalización Compra Abono"

Selector parámetros de viaje (Compra con Abono)	
Descripción	<p>Pantalla que permite a los usuarios la selección de los parámetros de consulta de un viaje sujeto a un abono activo</p> <p>Compuesta por:</p> <ul style="list-style-type: none">• RecyclerView (Item + Button) para seleccionar el abono que se desea utilizar en la asignación de billete• EditText "Ida" para la selección de la fecha de viaje
Activación	Desde el menú "Gestión Sistema de Compra"
Eventos	<ul style="list-style-type: none">• Volver (pulsar icono esquina superior izquierda)• Acceso a panel de ayuda (esquina superior derecha)• Pulsación sobre item (Button) para solicitar la petición de búsqueda de viajes• Validaciones• Actualización del listado de abonos disponibles en función de la fecha elegida• Pulsación sobre el campo EditText "Ida" para desplegar el calendario selector de fecha (DatePicker)

Diseño / Interfaz de Usuario



Figura 62 Interfaz "Selector parámetros de viaje (Compra con Abono)"

Selector sentido de viaje (Compra con Abono)

Descripción	<p>Pantalla que permite a los usuarios seleccionar el sentido del trayecto en función del abono utilizado</p> <p>Compuesta por:</p> <ul style="list-style-type: none">• RadioButtons para la selección de sentido• Button "Siguiente" para continuar con el proceso de asignación
Activación	<p>A continuación de escoger el abono que se desea utilizar en la pantalla "Selector parámetros de viaje (Compra con Abono)"</p>
Eventos	<ul style="list-style-type: none">• Volver (pulsar icono esquina superior izquierda)• Pulsación sobre cualquiera del RadioButton para seleccionar el sentido del viaje• Pulsación sobre Button "Siguiente" para continuar el proceso de asignación

Diseño / Interfaz de Usuario

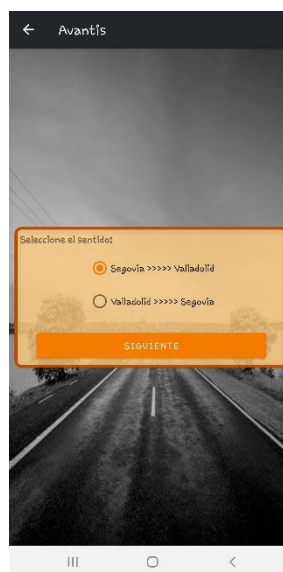


Figura 63 Interfaz "Selector sentido de viaje (Compra con Abono)"

5.3.2.4 Destinados a la gestión de información de usuarios

Mi Perfil	
Descripción	<p>Pantalla que permite visualizar la información personal del usuario autenticado en la aplicación</p> <p>Compuesta por:</p> <ul style="list-style-type: none">• CardView (TextViews) con la información personal del usuario
Activación	Desde el menú "Gestión Información Personal"
Eventos	<ul style="list-style-type: none">• Acceso al menú lateral (esquina superior izquierda o desplegable)• Acceso a panel de ayuda (esquina superior derecha)• Acceso a la pantalla "Modificar Perfil de Usuario" (pulsar icono en barra superior)

Diseño / Interfaz de Usuario



Figura 64 Interfaz "Mi Perfil"

Modificar Perfil de Usuario

Descripción

Pantalla que permite visualizar y modificar la información personal del usuario autenticado en la aplicación
Compuesta por:

- CardView (EditText) con la información personal del usuario
- Button "Actualizar Perfil" para confirmar la actualización del perfil del usuario

Activación

Desde la pantalla "Mi Perfil"

Eventos

- Acceso al menú lateral (esquina superior izquierda o desplegable)
- Pulsación sobre Button "Actualizar Perfil" para ejecutar la petición de actualización
- Validaciones sobre campos EditText

Diseño / Interfaz de Usuario

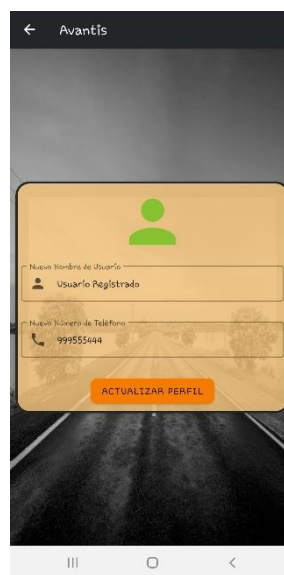


Figura 65 Interfaz "Modificar Perfil de Usuario"

Tarjeta de Crédito

Descripción	Pantalla que permite visualizar la información relativa al método de pago (tarjeta de crédito) asignada al usuario Compuesta por: <ul style="list-style-type: none"> • CardView (TextViews) con la información de la tarjeta de crédito asignada
Activación	Desde el menú "Gestión Información Personal"
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Acceso a panel de ayuda (esquina superior derecha) • En caso de SI existir previamente, Acceso a AlertDialog "Eliminar Tarjeta de Crédito" (pulsar icono en barra superior) con posterior solicitud de confirmación • En caso de NO existir previamente, Acceso a la pantalla "Añadir Tarjeta de Crédito" (pulsar icono en barra superior)

Diseño / Interfaz de Usuario



Figura 66 Interfaz "No existe Tarjeta de Crédito asociada"

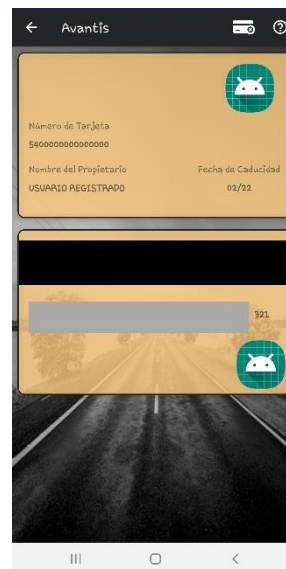


Figura 67 Interfaz "Ver tarjeta de crédito asociada"

Añadir Tarjeta de Crédito

Descripción

Pantalla que muestra al usuario el formulario que debe completar para asociar una tarjeta de crédito a su cuenta personal.

Compuesta por:

- Campos EditText de formulario
- Button para la confirmación y envío de información

Activación

Desde la pantalla "Mi Tarjeta de Crédito"

Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para el envío e inserción de tema • Validaciones sobre campos EditText
Diseño / Interfaz de Usuario	<div style="text-align: center;">  </div> <p style="text-align: center;"><i>Figura 68 Interfaz "Añadir Tarjeta de Crédito"</i></p>

Mis Viajes	
Descripción	<p>Pantalla que muestra los billetes adquiridos por el usuario</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • EditText superior (buscador) • RecyclerView (item + 3 Button por cada billete) <ul style="list-style-type: none"> ○ Button “Descarga” ○ Button “Ver Billete” ○ Button “Eliminar Billete”
Activación	<p>Desde el menú “Gestión Información Personal”</p>
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Acceso a panel de ayuda (esquina superior derecha) • Pulsación sobre Button “Descarga” para descargar el billete en formato PDF en el dispositivo • Pulsación sobre Button “Ver Billete” para acceder a la pantalla “Detalles Billete” • Pulsación sobre Button “Eliminar Billete” para eliminar el billete adquirido • Validaciones y actualizaciones destinadas a la eliminación de un billete

Diseño / Interfaz de Usuario



Figura 69 Interfaz "Mis Viajes"

Detalles Billete	
Descripción	<p>Pantalla que permite visualizar la información de un billete adquirido por parte del usuario</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • CardView (TextViews) con la información referente al viaje • Código QR de billete
Activación	<ul style="list-style-type: none"> • Desde la pantalla "Mis Billetes" • Desde la pantalla "Detalles Abono" al seleccionar uno de los billetes asociados al abono
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable)

Diseño / Interfaz de Usuario



Figura 70 Interfaz "Detalles Billete"

Mis Abonos

Descripción	<p>Pantalla que muestra los abonos adquiridos por el usuario</p> <p>Compuesta por:</p> <ul style="list-style-type: none">• RecyclerView (item + Button "Consulta" por cada abono)
Activación	<p>Desde el menú "Gestión Información Personal"</p>
Eventos	<ul style="list-style-type: none">• Acceso al menú lateral (esquina superior izquierda o desplegable)• Acceso a panel de ayuda (esquina superior derecha)• Pulsación sobre Button "Consulta" para acceder a la pantalla "Detalles Abono"

Diseño / Interfaz de Usuario



Figura 71 Interfaz "Mis Abonos"

Detalles Abono

Descripción	<p>Pantalla que permite visualizar la información referente a un abono adquirido por parte del usuario</p> <p>Compuesta por:</p> <ul style="list-style-type: none">• CardView (TextViews) con la información del abono• Estructura y funcionalidad similar a la pantalla "Mis Billetes" para mostrar los billetes asociados al abono
Activación	<ul style="list-style-type: none">• Desde la pantalla "Mis Abonos"
Eventos	<ul style="list-style-type: none">• Acceso al menú lateral (esquina superior izquierda o desplegable)• Acceso a panel de ayuda (esquina superior derecha)• Funcionalidad similar a la pantalla "Mis Billetes"

Diseño / Interfaz de Usuario



Figura 72 Interfaz "Detalles Abono"

5.3.2.5 Propias de administración

El contenido que puede gestionar el administrador es muy variado, pero las pantallas desarrolladas para esta gestión son muy semejantes tanto en estilo como en tareas y procedimientos. Por este motivo se ha decidido sintetizar el análisis de cada una de estas interfaces en las siguientes:

Consulta Contenido	
Descripción	<p>Pantalla que muestra al administrador un listado con todos los datos que conforman cada uno de los contenidos que este puede gestionar en el sistema</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • RecyclerView (item + 2 Button) para la visualización de datos <ul style="list-style-type: none"> ○ Button "Modificar" ○ Button "Eliminar"
Activación	Desde el menú lateral (esquina superior izquierda o desplegable)
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button "Modificar" para acceder a la pantalla "Modificación Contenido" • Pulsación sobre Button "Eliminar" para eliminar cierto contenido • Pulsación sobre icono (ActionBar superior) para acceder a la pantalla "Agregar Contenido"

Diseño / Interfaz de Usuario

Las interfaces presentan un diseño semejante. Por ese motivo se representan solo 3 de ellas (vea figuras 73, 74, 75).



Figura 73 Interfaz "Consultar contenido (Vehículos)"



Figura 74 Interfaz "Consultar contenido (Reservas)"



Figura 75 Interfaz "Consultar contenido (Precios)"

Agregar Contenido	
Descripción	<p>Pantalla que muestra al usuario el formulario que debe completar para la creación de un nuevo contenido en el sistema. Compuesta por:</p> <ul style="list-style-type: none"> • Campos EditText, Checkbox, Spinner, TimePicker, ... para formulario • Button para la confirmación y envío de información
Activación	Desde la pantalla de "Consulta Contenido" (seleccionando agregar nuevo contenido)
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para el envío e inserción de tema • Validaciones sobre campos
Diseño / Interfaz de Usuario	Vea figuras desde 76 hasta 83 (ambas inclusive).

Modificación Contenido	
Descripción	<p>Pantalla que permite visualizar y modificar la información del contenido seleccionado por parte del administrador.</p> <p>Compuesta por:</p> <ul style="list-style-type: none"> • Campos EditText, Checkbox, Spinner, TimePicker, ... para formulario • Button para la confirmación y envío de información para proceder a la actualización
Activación	Desde la pantalla "Consulta Contenido" (seleccionando la modificación de contenido)
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral (esquina superior izquierda o desplegable) • Pulsación sobre Button para el envío e inserción de tema • Validaciones sobre campos
Diseño / Interfaz de Usuario	Las interfaces de modificación presentan una estructura semejante a las interfaces de creación de contenido (vea figuras desde 76 a 83).



Figura 76 "Añadir/Modificar vehículo"



Figura 77 "Añadir/Modificar trayecto"



Figura 78 Interfaz "Añadir/Modificar estación"

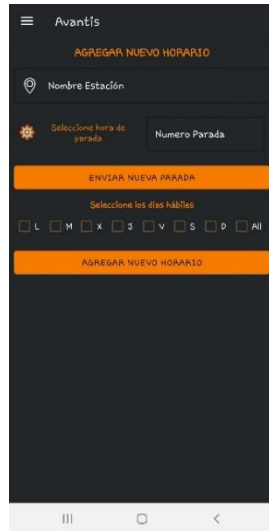


Figura 79 Interfaz "Añadir/Modificar horario"

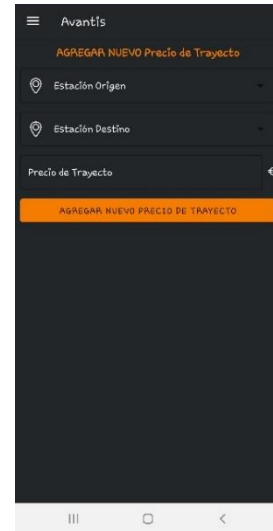


Figura 80 Interfaz "Añadir/Modificar precio"



Figura 81 Interfaz "Añadir/Modificar tipo de abono"

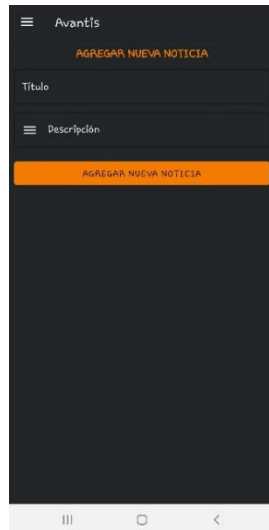


Figura 82 Interfaz "Añadir/Modificar noticia"

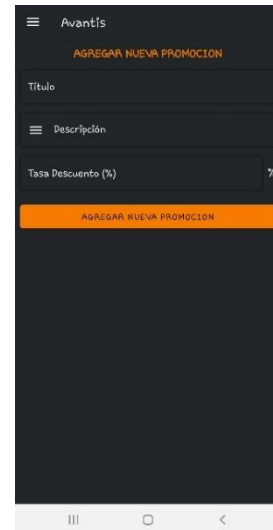


Figura 83 Interfaz "Añadir/Modificar promoción"



Capítulo 6: IMPLEMENTACIÓN



6.1 Requerimientos Hardware y Software

Para poder ejecutar la aplicación sin ningún tipo de error y con la mayor fluidez posible, los dispositivos Android deberán contar con los siguientes requisitos mínimos:

Requisitos Mínimos		
SOFTWARE	Sistema Operativo	Android 5 o superior
	Otros	Gestor de contenido PDF (ej: Adobe Acrobat)
HARDWARE	RAM	1,5 GB o superior
	CPU	Quad-Core 1,2GHz o superior
	Memoria Interna	8GB o superior
	Conectividad	Wifi o 4G (servicio de datos) para contar con una conexión a Internet estable.
	Otros	Compatibilidad con los servicios de Google

Tabla 77 Requisitos mínimos

6.2 Herramientas utilizadas

6.2.1 Herramientas para el desarrollo móvil

- **Android Studio (1):** Entorno de desarrollo oficial (IDE) para la realización de cualquier software o aplicación destinada a trabajar en plataformas Android. Se basa en la tecnología IntelliJ IDEA (2) y cuenta con potentes herramientas de desarrollo integradas, tales como:
 - Editor de código.
 - Sistema de compilación basado en Gradle.
 - Emuladores rápida y ampliamente configurables capaces de soportar y otorgar al usuario una gran capacidad de prueba de software.
 - Integración con GitHub (3), facilitando el control de versiones y la implementación y compilación de funciones.
 - Compatibilidad con Google Cloud Platform (4), lo que facilita la integración de sus servicios en las aplicaciones.
- **Firebase:** plataforma de servicios destinados a facilitar y mejorar el desarrollo de aplicaciones, tanto aplicaciones web como desarrollos móviles.
 - **Firebase Authentication:** servicio que se encarga de facilitar la integración dinámica de usuarios en nuestro sistema/aplicación. Permite además la integración de servicios de autenticación con redes sociales.

- **Firestore:** servicio que se encarga de proporcionar a nuestro sistema/ aplicación una base de datos que trabaja en tiempo real. Esta base de datos NoSQL es completamente escalable y flexible con la información gestionada.
- **Maps SDK for Android:** conjunto de aplicaciones y herramientas proporcionadas por Google Cloud Platform destinadas a la obtención y visualización de mapas a través de la web.
 - **Geocoding API:** servicio externo que se encarga de traducir coordenadas geográficas en direcciones físicas y viceversa.
 - **Geolocation API:** servicio externo que se encarga de buscar 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.
- **Gmail API:** servicio externo encargado de gestionar todo el contenido de una cuenta de correo electrónico (buzones, mensajes, etiquetas, carpetas...).
- **PayPal SDK:** conjunto de aplicaciones y herramientas proporcionadas por PayPal destinadas a otorgar a la aplicación la posibilidad de realizar pagos en línea entre usuarios que dispongan de una cuenta registrada en la plataforma de PayPal.

6.2.2 Herramientas de soporte

- **Microsoft Word:** herramienta creada por Microsoft utilizada para el procesamiento de textos. En el proyecto Avantis su utilidad ha sido el desarrollo de la memoria de este.
- **Microsoft Project:** herramienta creada por Microsoft cuya funcionalidad principal es la gestión y organización de proyectos. En Avantis se ha utilizado en la planificación y el seguimiento del proyecto (Diagrama de Gantt).
- **Adobe Reader DC:** herramienta desarrollada por Adobe Systems utilizada para la gestión y el manejo de archivos en formato PDF. En el proyecto Avantis su utilidad ha sido la creación, lectura y manejo de estos archivos.
- **Adobe XD:** herramienta desarrollada por Adobe Systems cuya funcionalidad es facilitar al desarrollador la tarea de diseño y prototipado de la experiencia de usuario (UX) de la aplicación. En el proyecto Avantis su utilidad ha sido la realización de pruebas de diseño y el diseño de parte de la iconografía de la aplicación.

- **Google Drive:** servicio proporcionado por Google cuya utilidad es otorgar al usuario un repositorio de contenidos. En el proyecto Avantis se ha utilizado para generar y almacenar los diagramas requeridos.
- **Draw.io:** herramienta de código abierto utilizada para la realización de diagramas en línea. Permite su integración en distintas plataformas, aunque la que se ha llevado a cabo en este proyecto es la integración en Google Drive.
- **Online Java Compiler IDE:** entorno de desarrollo (IDE) en línea utilizado para la realización de validaciones sobre bloques de código Java.
- **Notepad++ :** editor de textos y de código libre con soporte para Microsoft Windows capaz de soportar una gran variedad de lenguajes de programación.

6.3 Tecnologías utilizadas

- **Java:** es uno de los lenguajes de programación orientados a objetos cuyo objetivo consiste en lograr que el software desarrollado sea capaz de ejecutarse en la mayoría de las plataformas y contextos propuestos (Multiplataforma, portabilidad). Utilizado para el desarrollo del contexto funcional de la aplicación Avantis.
- **XML:** *Extensible Markup Language*. Es un lenguaje de marcado que se encarga de describir el comportamiento o la estructura de los datos de la aplicación. En este proyecto, los archivos XML se utilizan para definir las vistas de la aplicación.

6.4 Librerías y dependencias

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

- **IText PDF ‘com.itextpdf:itextg:5.5.10’:** Librería que permite la gestión de contenido en formato PDF y que se utiliza en la creación de archivos PDF con la información del billete.
- **QRGenerator ‘androidmads.library.qrgenerator:QRGenerator:1.0.3’:** Librería que permite la creación y gestión de códigos QR a partir de una entrada de datos. Utilizada para generar el código QR con la información del billete.
- **TapTargetView ‘com.getkeepsafe.taptargetview:taptargetview:1.11.0’:** Librería que permite crear y gestionar contenido de ayuda dentro de la aplicación.

- **Commons Text ‘org.apache.commons:commons-text:1.8’:** Librería que permite gestionar y formatear cadenas de caracteres. Utilizada para normalizar la información introducida y previamente registrada en la aplicación.
- **Gmail Sender Library ‘com.github.tntkhang:gmail-sender-library:1.3.0’:** Librería que permite automatizar y simplificar el envío de correos electrónicos a través de Gmail. Utilizada para enviar la información de un billete por correo electrónico (Gmail).

6.5 Detalles de implementación

Para llevar a cabo la tarea de implementación de la aplicación, hay que tener muy en cuenta los objetivos y requisitos que esta debe desempeñar correctamente y que se han descrito previamente en capítulos anteriores (véase *Bloques 1 y 4*).

Para comenzar, la tarea principal que debe desempeñar la aplicación es el sistema de compra de billetes (diferenciando entre la compra de billetes ocasionales y la compra de estos utilizando un abono) y la compra de abonos. Para ambas tareas será esencial contar en un primer lugar con el API de Firebase “Firebase Auth”, el cual permitirá distinguir qué tipo de usuario se encuentra utilizando el sistema en el momento de realizar la compra para distinguir los sistemas de compra a los que puede tener acceso.

El siguiente paso es contar con una base de datos que nos permita acceder a los datos necesarios para realizar la compra (como es la información de los viajes, disponibilidad, abonos asociados al usuario, tipos de abono que se pueden adquirir...) en tiempo real. Esta será proporcionada por el API de Firebase “Cloud Firestore”, el servicio de base de datos que posteriormente se describirá con más profundidad.

Como último paso de este proceso de compra, se debe contar con una plataforma que sea capaz de recoger y procesar el pago de dicha compra. Aquí se contará con el API de PayPal, servicio que permitirá al usuario realizar el pago de forma segura y controlará en todo momento el estado de este.

De forma adicional, en el sistema de compra de billetes también se dispone de la posibilidad de enviar los mismos por correo electrónico a los compradores, y para ello se cuenta con el API de Google “servicio de mensajería Gmail”, que funcionará en background y de manera automática para simplificar esta tarea de envío.

Por otro lado, la aplicación cuenta con la posibilidad de consultar la información de una estación o parada, lo que incluye la localización de esta. Para ello se requiere de la API de Google “servicio de Google Maps”, que se encargará de crear un mapa visible para los usuarios donde se representará la localización real de esta.

Los demás objetivos y requisitos relacionados con la funcionalidad de la aplicación quedarán todos ellos cubiertos con las API de Firebase “Firebase Auth” y “Cloud Firestore”, siendo estas las más importantes en el desarrollo de la aplicación y en las que posteriormente se profundizará en su puesta en marcha e implementación en el proyecto.

Hay que tener en cuenta y entender que el software desarrollado no es simplemente una herramienta con la que el usuario final tenga en su mano la posibilidad de administrar sus viajes, gestionar la compra y anulación tanto de billetes como de abonos y tener disponibilidad total a otro gran número de funcionales con las que cuenta la aplicación.

El sistema, entendido como conjunto no sería un producto completo si no se tuviera en consideración la parte administrativa del mismo. Esta última parte se considera esencial puesto que es la que permite hacer crecer y mantener nuestro sistema en todo momento y en cualquier circunstancia. Por este motivo, se consideró totalmente necesario desarrollar tanto la parte destinada al usuario final como la parte encargada de la administración de todo el sistema (Ambas especificadas en capítulos anteriores).

A partir de este momento es cuando sí que podemos considerar a nuestra aplicación/sistema un producto completo.

6.5.1 Pasos seguidos en la implementación

Parte importante del desarrollo de esta aplicación radica en la elección del modelo de base de datos utilizado, una base de datos albergada en la nube (Firebase *Platform*) NoSQL denominada Cloud Firestore.

En el anexo I del documento se definen las características básicas y se explican los motivos principales por los cuales se ha elegido este tipo de base de datos para desarrollar todo el sistema.

Como consecuencia de la elección de este aspecto fundamental en el desarrollo, se llevan a cabo una serie de pasos esenciales para que la implementación y configuración sea correcta y eficiente:

PASO 1	Creación del proyecto dentro de la plataforma Firebase.
PASO 2	Integración del SDK de Cloud Firestore en la aplicación Android.
PASO 3	Procedimientos de inserción y recuperación de datos.
PASO 4	Procedimientos de modificación y eliminación de datos.
PASO 5	Configuración de reglas de seguridad en Cloud Firestore.

Paso 1: Creación del proyecto dentro de la plataforma Firebase

1. En cualquier navegador accedemos a <https://firebase.google.com/?hl=es>.
2. Una vez allí, pulsaremos en “Ir a la consola” (esquina superior derecha).
3. Iniciaremos sesión con nuestras credenciales.
4. Una vez dentro nos encontraremos con la siguiente pantalla (figura)Aquí podremos observar como ya contamos con un proyecto asignado a nuestra cuenta.

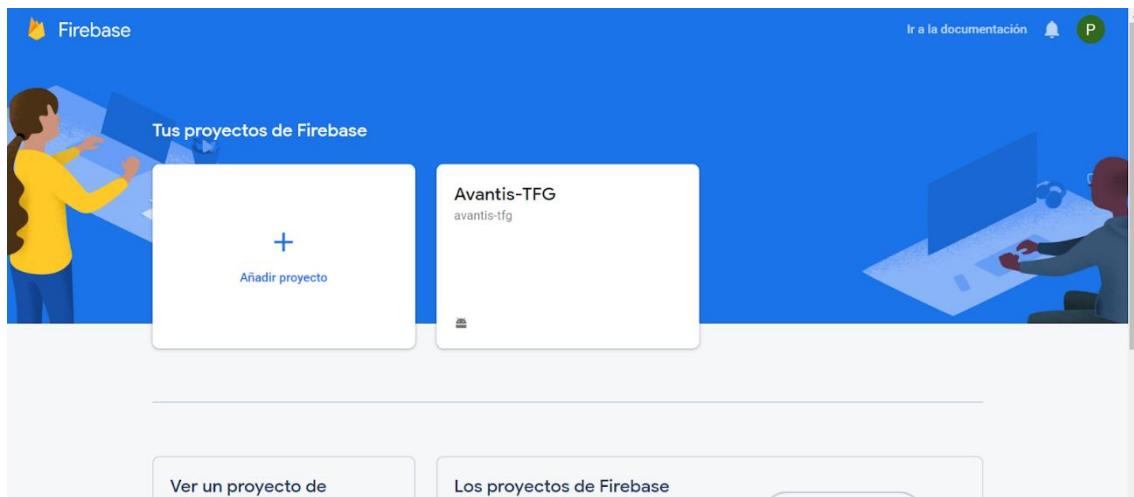


Figura 76 Proyectos asignados a la cuenta de desarrollo

- Si se quiere crear un nuevo proyecto, pulsaremos en “Añadir proyecto”. Nos aparecerán una serie de pantallas donde podremos configurar aspectos generales de nuestro nuevo proyecto como son el nombre, integración de Google Analytics... (véase figuras 85 y 86)

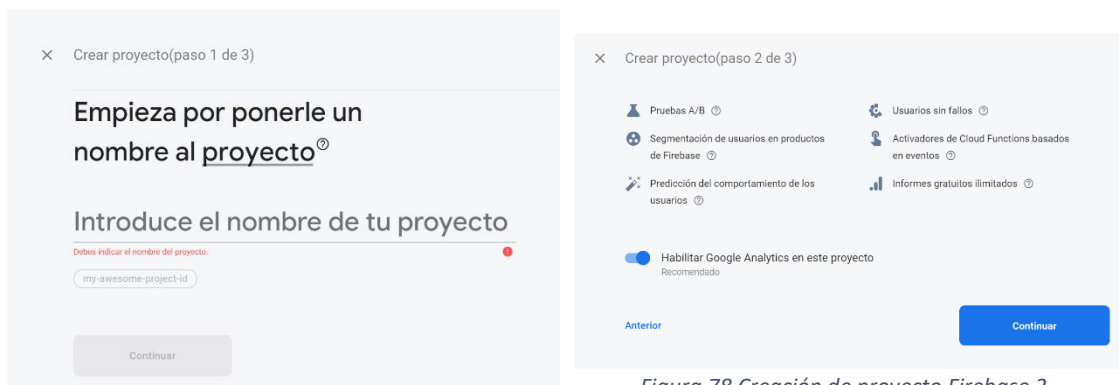


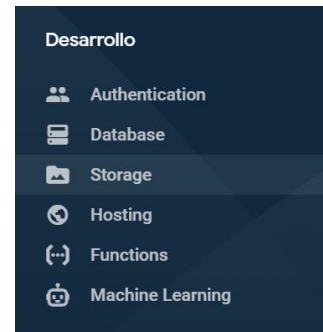
Figura 77 Creación de proyecto Firebase 1

Figura 78 Creación de proyecto Firebase 2

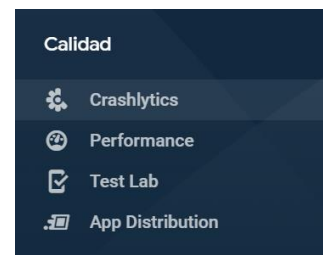
- Una vez creado y ya dentro del proyecto Firebase, nos fijamos en que se distinguen 2 partes diferenciadas: un menú lateral donde encontramos todos los servicios que nos proporciona la plataforma y una pantalla principal donde se muestra la información dependiendo de nuestra selección lateral.

7. En el menú lateral se encuentran todos los servicios que Firebase pone a nuestra disposición, distribuidos en:

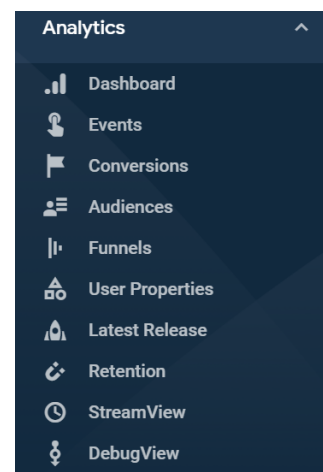
Desarrollo: en nuestro caso es la más importante puesto que es aquí donde se alojan los servicios utilizados más frecuentemente en el desarrollo de aplicaciones, tales como *Authentication*, *Cloud Firestore*, *Storage*, *Functions*,...



Calidad: Servicios destinados al mantenimiento del software, tales como informes de errores, laboratorio de *testing*, pruebas de rendimiento y distribución de la aplicación.



Analytics: Servicio destinado a la monitorización del software y mediciones de interacción de la herramienta con los usuarios.



Crecimiento: Servicios destinados a la comunicación con los usuarios, predicciones de comportamiento,...

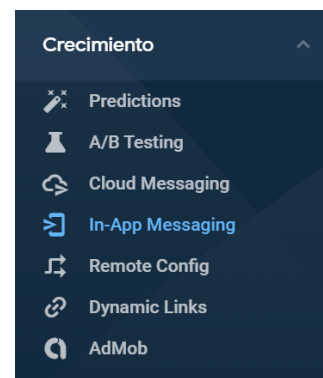


Tabla 78 Servicios ofertados por Firebase para el proyecto

Paso 2: Integración del SDK de Cloud Firestore en la aplicación Android

1. Retomando el proceso en el punto 6 del Paso 1, tendremos que añadir Firebase a nuestra aplicación. Pulsaremos en el icono de Android para empezar a enlazar (figura 87).



Figura 79 Integración Cloud Firestore a proyecto paso 1

2. Nos aparecerá una pantalla en la cual debemos de seguir una serie de pasos. El primero de todos ellos es introducir el nombre del paquete de Android de nuestra aplicación, un apodo para nuestra aplicación y la firma digital (de manera opcional y requerida sólo en casos muy especiales para ciertos servicios). (figura 88).

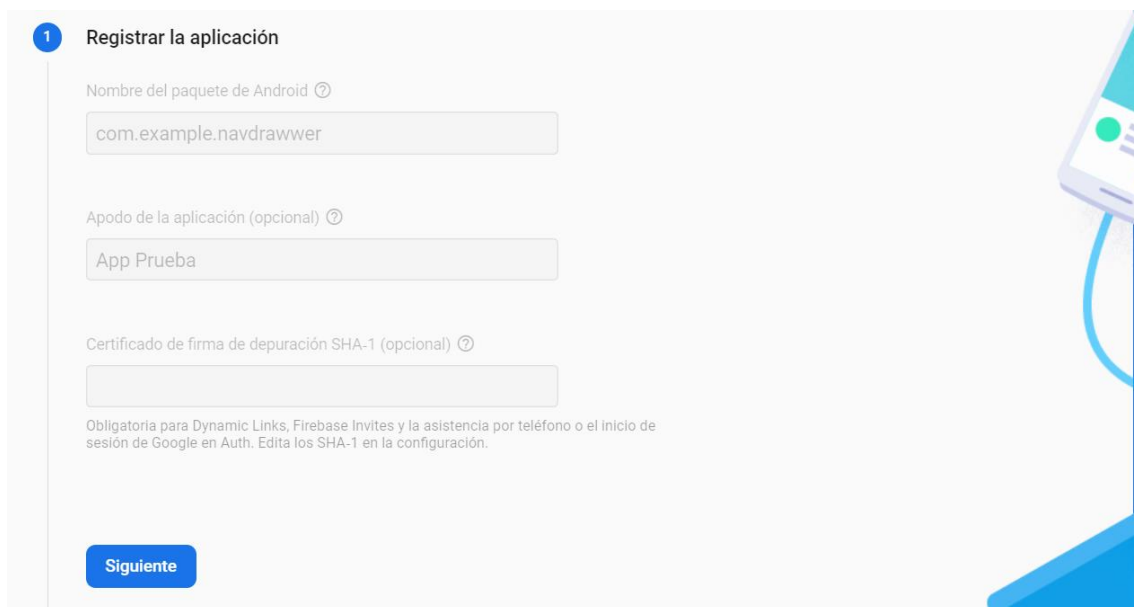


Figura 80 Integración Cloud Firestore a proyecto paso 2

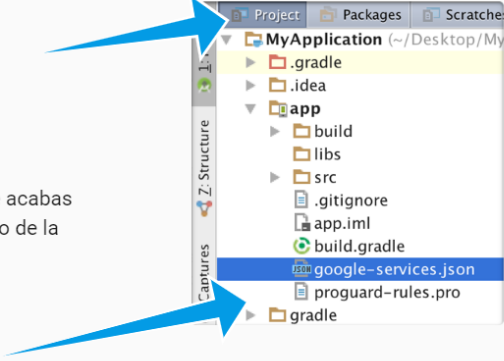
3. A continuación, debemos descargar un archivo JSON que tenemos que incorporar a nuestro proyecto en Android Studio. (figura 89)

2 Descargar el archivo de configuración Instrucciones para Android Studio más abajo | [Unity](#) [C++](#)

[Descargar google-services.json](#)

Cambia a la vista **Proyecto** de Android Studio para ver el directorio "root" de tu proyecto.

Mueve el archivo `google-services.json` que acabas de descargar al directorio "root" del módulo de la aplicación de Android.



Anterior [Siguiente](#)

Figura 81 Integración Cloud Firestore a proyecto paso 3

4. La incorporación del archivo “`google-services.json`” se debe de realizar en el root de la aplicación en Android Studio.
5. Para terminar con la integración del SDK, será necesario activar los servicios de Google (Google *Services*) e incorporar las dependencias que nos indican en los archivos “`build.gradle`” como se puede observar en las figuras 90 y 91.

3 Añadir el SDK de Firebase Instrucciones para Gradle | [Unity](#) [C++](#)

El complemento de los servicios de Google para [Gradle](#) carga el archivo `google-services.json` que acabas de descargar. Para utilizar dicho complemento, debes modificar tus archivos `build.gradle`.

Build.gradle de nivel de proyecto (<proyecto>/build.gradle):

```

buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.3'
    }
}

allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        ...
    }
}

```

Figura 82 Integración Cloud Firestore a proyecto paso 4.1



Figura 83 Integración Cloud Firestore a proyecto paso 4.2

Paso 3: Procedimientos de inserción y recuperación de datos

Una vez terminado el proceso de integración del SDK de Firebase a nuestra aplicación ya podemos empezar a implementar los métodos principales que van a interactuar con nuestra base de datos en la nube, Cloud Firestore.

Para comenzar a implementar estos métodos es importante explicar y entender que, pese a que Cloud Firestore permite utilizar objetos del tipo *Map* para el manejo de bloques de datos y representación de documentos, no es la opción recomendable.

Es mucho más cómodo e intuitivo la creación de objetos personalizados, siempre y cuando la notación utilizada en los mismos sea la adecuada y pueda ser interpretada correctamente por Firestore. De ser así, este mismo se encargará de convertir los objetos personalizados en estructuras de datos compatibles que darán lugar a documentos (unidad de almacenamiento). Trabajar con estos objetos personalizados será entonces la manera de proceder en el desarrollo de nuestra aplicación.

El siguiente paso es declarar una instancia de nuestra base de datos para poder trabajar con ella. Esto se realizará en todos los métodos *onCreate()* y *onCreateView()* en los que sea necesario realizar algún tipo de consulta a la base de datos. Una vez instanciada la base de datos, ya se puede interactuar con ésta en cualquier momento.

```
private FirebaseFirestore database = FirebaseFirestore.getInstance();
```

Figura 84 Crear instancia Cloud Firestore (código)

Los métodos que se van a mostrar a partir de ahora son métodos generales para la realización de consultas básicas (inserción, modificación, eliminación y obtención de datos), aun así, todos ellos son una simplificación de los métodos utilizados en la aplicación para agilizar la comprensión de estos.

Comenzamos con el método general para insertar datos (en este caso, una nueva estación):

```
String id_estacion = "LOCALIZADOR ESTACION";

DatosEstacion estacion = new DatosEstacion( stationName: "Nombre Estacion",
    direction: "Direccion",
    stationEmail: "estacion@gmail.com",
    stationPhone: "987987987",
    new ArrayList<>(),
    new GeoPoint( latitude: 40.1234, longitude: -8.987),
    publicado: true);
```

Figura 85 instancia Objeto "DatosEstación"

```
private void agregarEstacion(DatosEstacion estacion) {
    database.collection( collectionPath: "STATIONS") CollectionReference
        .add(estacion) Task<DocumentReference>
        .addOnSuccessListener( documentReference -> {

            Log.d( tag: "RESULTADO", msg: "ÉXITO");
            //CÓDIGO SI LA INSERCIÓN HA SIDO EXITOSA

        }) Task<DocumentReference>
        .addOnFailureListener(e -> {

            Log.d( tag: "RESULTADO", msg: "FALLO EN PETICIÓN");
            //CÓDIGO EN FALLO DE INSERCIÓN

        });
}
```

Figura 86 Método de inserción de contenido en Cloud Firestore (código)

Como podemos observar, la estación que se quiere incorporar queda definida como un objeto del tipo “*DatosEstacion*” que previamente hemos definido en la parte superior (Figura X).

A continuación, aparece la instancia de la base de datos a la que indicaremos que apunte a la colección “STATIONS”, que será el lugar donde se crea un nuevo documento con la información de la estación. Por último, se incorporan 2 escuchadores:

- ***addOnSuccessListener()***: se ejecutará cuando la consulta realizada a base de datos se haya completado y además se haya completado con éxito.
- ***addOnFailureListener()***: se ejecutará cuando la consulta realizada a base de datos haya fallado.

En ambos casos se crea un Log para notificar y almacenar el resultado de la operación.

Para terminar este paso, se muestran 3 ejemplos de métodos destinados a la obtención de datos:

```
private void listarEstaciones() {
    database.collection( collectionPath: "STATIONS") CollectionReference
        .get() Task<QuerySnapshot>
        .addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                for (QueryDocumentSnapshot document : task.getResult()) {

                    Log.d( tag: "RESULTADO", msg: "ÉXITO");
                    //CÓDIGO PARA CADA RESULTADO DEVUELTO EN CONSULTA

                }
            }
        }) Task<QuerySnapshot>
        .addOnFailureListener(e -> {

            Log.d( tag: "RESULTADO", msg: "FALLO EN PETICIÓN");
            //CÓDIGO EN CASO DE FALLO EN CONSULTA A BASE DE DATOS

        });
};
```

Figura 87 Método de obtención de datos Cloud Firestore 1

```
private void listarEstaciones() {

    String valorContenido = "valor";

    database.collection( collectionPath: "STATIONS") CollectionReference
        .whereArrayContains( field: "campoArray", valorContenido) Query
        .whereEqualTo( field: "valorEncontrado", valorContenido) Query
        .get() Task<QuerySnapshot>
        .addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                for (QueryDocumentSnapshot document : task.getResult()) {

                    Log.d( tag: "RESULTADO", msg: "ÉXITO");
                    //CÓDIGO PARA CADA RESULTADO DEVUELTO EN CONSULTA

                }
            }
        }) Task<QuerySnapshot>
        .addOnFailureListener(e -> {

            Log.d( tag: "RESULTADO", msg: "FALLO EN PETICIÓN");
            //CÓDIGO EN CASO DE FALLO EN PETICIÓN A BASE DE DATOS

        });
};
```

Figura 88 Método de obtención de datos Cloud Firestore 2

```
private void listarEstaciones() {

    String id_estacion = "LOCALIZADOR ESTACION";

    database.collection( collectionPath: "STATIONS") CollectionReference
        .document(id_estacion) DocumentReference
        .get() Task<DocumentSnapshot>
        .addOnSuccessListener(documentSnapshot -> {

            Log.d( tag: "RESULTADO", msg: "ÉXITO");
            //CÓDIGO PARA EL RESULTADO DEVUELTO EN CONSULTA

        }) Task<DocumentSnapshot>
        .addOnFailureListener(e -> {

            Log.d( tag: "RESULTADO", msg: "FALLO EN PETICIÓN");
            //CÓDIGO EN CASO DE FALLO EN CONSULTA A BASE DE DATOS

        });
};
```

Figura 89 Método de obtención de datos Cloud Firestore 3

El primero de ellos (figura 95) está destinado a obtener todos los documentos de la colección “STATIONS”. En este caso se han incorporado también 2 escuchadores de eventos, pero con algunas diferencias respecto al método de inserción:

- ***addOnCompleteListener()***: se ejecutará cuando la consulta realizada a base de datos se haya completado.
 - ***if (task.isSuccessful())***: se ejecutará en el caso de que la consulta haya tenido éxito.
 - ***bucle for***: código que se ejecuta para cada uno de los documentos (QueryDocumentSnapshot) que se han obtenido.
- ***addOnFailureListener()***: se ejecutará cuando la consulta realizada a base de datos haya fallado.

El segundo método (figura 96) está destinado a obtener todos los documentos de la colección “STATIONS” que cumplen con unas determinadas condiciones.

- ***whereArrayContains()***: se utiliza para filtrar todos los documentos que en su campo del tipo array “campoArray” contengan el valor “valorContenido”.
- ***whereEqualTo()***: se utiliza para filtrar todos los documentos que en su campo “valorEncontrado” tengan el “valorContenido”.
- Cuando se aplican 2 filtros como en este caso, se traducen por una consulta AND, es decir, solo devolverá los documentos que cumplan las 2 condiciones.

En este caso se han incorporado también 2 escuchadores de eventos idénticos a los incorporados en el primer método.

En el tercero de ellos (figura 97) también existe la posibilidad de recuperar un documento concreto de la base de datos. Para ello simplemente habría que apuntar a la colección donde se encuentre registrado (en nuestro caso “STATIONS”) y buscarlo a través de su identificador (“id_estacion”). Contará con los mismos escuchadores ya explicados previamente.

Todos ellos también incorporan la creación de un Log para notificar el resultado de la operación de búsqueda.

Paso 4: Procedimientos de modificación y eliminación de datos

Para la implementación del método de modificación se seguirá un modelo muy similar al utilizado en la inserción. A pesar de guardar estas semejanzas, se han propuesto 2 procedimientos distintos para realizar esta tarea.

```

private void modificarEstacionCompleta(String id_estacion, DatosEstacion estacion) {
    database.collection( collectionPath: "STATIONS") CollectionReference
        .document(id_estacion) DocumentReference
        .set(estacion) Task<Void>
        .addOnSuccessListener(documentReference -> {

            Log.d( tag: "RESULTADO", msg: "ÉXITO");
            //CÓDIGO SI LA MODIFICACIÓN HA SIDO EXITOSA

        }) Task<Void>
        .addOnFailureListener(e -> {

            Log.d( tag: "RESULTADO", msg: "FALLO EN PETICIÓN");
            //CÓDIGO EN FALLO DE MODIFICACIÓN

        });
}

```

Figura 90 Método modificación completa en Cloud Firestore (código)

```

private void modificarEstacionParcial(String id_estacion) {
    database.collection( collectionPath: "STATIONS") CollectionReference
        .document(id_estacion) DocumentReference
        .update( field: "atributo1", value: "valor1",
            ...moreFieldsAndValues: "atributo2", "valor2") Task<Void>
        .addOnSuccessListener(documentReference -> {

            Log.d( tag: "RESULTADO", msg: "ÉXITO");
            //CÓDIGO SI LA MODIFICACIÓN HA SIDO EXITOSA

        }) Task<Void>
        .addOnFailureListener(e -> {

            Log.d( tag: "RESULTADO", msg: "FALLO EN PETICIÓN");
            //CÓDIGO EN FALLO DE MODIFICACIÓN

        });
}

```

Figura 91 Método modificación parcial en Cloud Firestore (código)

El primero de ellos (figura 98) se utiliza para modificar un documento completo de Firebase. Para ello contamos con la instancia de la base de datos apuntando al documento que se quiere modificar (en nuestro caso es una estación que se encuentra en la colección “STATIONS” y cuyo identificador es “id_estacion”).

Una vez establecida la referencia del documento y a través del método `set(estación)`, se lleva a cabo la modificación de este. Este procedimiento, más que actuar como una modificación, realiza una sustitución completa del documento almacenado por el nuevo objeto “estación” pero manteniendo su identificador.

El segundo (figura 99) se utiliza principalmente para la modificación parcial del documento, es decir, para actualizar atributos de forma aislada. Para ello es necesario obtener la referencia del documento de la misma forma que se ha obtenido en el anterior procedimiento. Una vez obtenida, se utiliza el método `update()`, el cual recibe como

parámetros los campos del documento que se quieren modificar junto a los nuevos valores.

Ambos procedimientos cuentan con 2 escuchadores para capturar la respuesta de la modificación junto con la creación de un Log para notificar y almacenar el resultado de la operación:

- ***addOnSuccessListener()***: se ejecutará cuando la consulta realizada a base de datos se haya completado y además se haya completado con éxito.
- ***addOnFailureListener()***: se ejecutará cuando la consulta realizada a base de datos haya fallado.

Para terminar con los procedimientos básicos de Cloud Firestore, contamos con la eliminación de un documento. Para este caso, la estructura del algoritmo será similar a la utilizada en los métodos de modificación.

La diferencia más significativa que existe es que, una vez obtenida la referencia del documento que se desea eliminar, se hará uso de la sentencia `delete()` para realizar la tarea (figura 100).

```
private void eliminarEstacion(String id_estacion) {
    database.collection( collectionPath: "STATIONS") CollectionReference
        .document(id_estacion) DocumentReference
        .delete() Task<Void>
        .addOnSuccessListener(documentReference -> {

            Log.d( tag: "RESULTADO", msg: "ÉXITO");
            //CÓDIGO SI LA ELIMINACIÓN HA SIDO EXITOSA

        }) Task<Void>
        .addOnFailureListener(e -> {

            Log.d( tag: "RESULTADO", msg: "FALLO EN PETICIÓN");
            //CÓDIGO EN FALLO DE ELIMINACIÓN

        });
}
```

Figura 92 Método de eliminación de documento en Cloud Firestore (código)

Como consideración, hay que tener en cuenta que, a pesar de borrar un documento, las subcolecciones que este pueda albergar seguirán estando accesibles desde la consola de Firebase a través de su referencia. Por este motivo, si se requiere de una eliminación completa, el proceso se realizará de forma manual.

Paso 5: Configuración de reglas de seguridad en Cloud Firestore.

La base de datos de Cloud Firestore permite al administrador incorporar restricciones al acceso y manipulación, de tal manera que, con este mecanismo, se consigue dotar a la misma de un nivel óptimo de seguridad. La configuración se debe realizar desde la consola de Firebase, en el panel de reglas de Firestore.

6.5.2 Administración de usuarios en Firebase

Otro aspecto importante de la aplicación es la forma de controlar y gestionar a los usuarios que utilizarán la aplicación. En este apartado se analizarán los tres mecanismos más importantes referentes al usuario como son el registro de nuevos usuarios, el inicio de sesión y el cierre de sesión.

El primer paso es declarar una instancia de Firebase *Auth* para poder trabajar con este servicio. Esto se realizará en todos los métodos *onCreate()* y *onCreateView()* en los que sea necesario utilizar la información de un usuario.

```
private FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
```

Figura 93 Crear instancia Firebase Authentication (código)

6.5.2.1 Registro de usuarios

Para la creación de nuevos usuarios en la aplicación se utilizará el método de Firebase *Auth* “*createUserWithEmailAndPassword()*”. Este se encargará de generar un nuevo usuario en la plataforma a través del par de credenciales correo electrónico y contraseña (figura 102):

```
private void registrarUsuario(String email, String password, UsuarioRegistrado infoUsuario) {
    firebaseAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener( activity, this, task -> {
            if (task.isSuccessful()) {
                //CÓDIGO SI EL REGISTRO HA SIDO EXITOSO
                database.collection( collectionPath: "USERS") CollectionReference
                    .add(infoUsuario) Task<DocumentReference>
                    .addOnSuccessListener(documentReference -> {
                        //CÓDIGO SI LA CREACIÓN DE INFORMACIÓN DE USUARIO HA SIDO EXITOSA
                    }) Task<DocumentReference>
                    .addOnFailureListener(e -> {
                        //CÓDIGO EN FALLO DE CREACIÓN DE INFORMACIÓN DE USUARIO
                    });
            } else {
                if (task.getException() instanceof FirebaseAuthUserCollisionException) {
                    Toast.makeText( context: MainActivity.this,
                        text: "El usuario actual ya existe",
                        Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText( context: MainActivity.this,
                        text: "Error al REGISTRAR USUARIO",
                        Toast.LENGTH_LONG).show();
                }
            }
        });
}
```

Figura 94 Método de registro de usuarios en Firebase Auth (código)

Como se puede observar, se ha incorporado un escuchador a nuestro evento de registro “*addOnCompleteListener()*” para comprobar en todo momento el estado de la creación.

1. Si la inserción se ha completado con éxito, se procede a crear un registro en la base de datos Cloud Firestore con la información adicional del usuario registrado (número de teléfono, DNI, dirección postal...). Este procedimiento también cuenta con 2 escuchadores para notificar al usuario en todo momento el estado de la inserción.
2. Si la inserción ha fallado se controlarán dos posibles motivos:
 1. Ya existe un usuario con el mismo correo electrónico registrado en la aplicación.
 2. Fallo genérico por causas ajenas.

6.5.2.2 Autenticación de usuarios

Una vez el usuario se encuentre registrado en la aplicación, podrá iniciar sesión en la misma con el par correo electrónico/contraseña utilizado en el registro. Para llevar a cabo el proceso de autenticación se utilizará el método de Firebase *Auth* “*signInWithEmailAndPassword*” (figura 103):

```
private void autenticarse(String email, String password) {
    firebaseAuth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener( activity: this, task -> {
            if (task.isSuccessful()) {

                //CÓDIGO SI LA AUTENTICACIÓN HA SIDO EXITOSA

                //CREAR SHARED PREFERENCES

            } else {

                Toast.makeText( context: MainActivity.this,
                    text: "Error al iniciar sesión",
                    Toast.LENGTH_LONG).show();

            }

        });
}
```

Figura 95 Método de inicio de sesión a través de Firebase Auth (código)

En este caso también se contará con dos escuchadores para controlar el proceso de inicio de sesión. En el caso de que el proceso haya sido exitoso, cabe destacar que se creará el archivo “*Shared Preferences*” donde se aloja la información del usuario necesaria para la realización de múltiples tareas una vez se encuentre autenticado.

6.5.2.3 Cerrar sesión

Si el usuario autenticado quiere cerrar su sesión en la aplicación, el método utilizado es el siguiente (figura 104):

```
private void cerrarSesion() {  
    //ELIMINAR SHARED PREFERENCES  
    //CERRAR  
    FirebaseAuth.getInstance().signOut();  
}
```

Figura 96 Método de logout en Firebase Auth (código)

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.

6.6 Estructura interna de proyecto

En este apartado se analiza la estructura interna del proyecto. Se entiende como estructura interna a la organización de los archivos en directorios y paquetes, de tal manera que se permita a los desarrolladores un mayor rendimiento y fluidez a la hora de desarrollar nueva funcionalidad o realizar tareas de mantenimiento en la aplicación.

A pesar de contar con una estructura bastante compleja, el código principal de la aplicación se encuentra en el directorio:

- */Avantis_TFG/app/src*

Este directorio a su vez se encuentra organizado en una serie de subdirectorios como son:

- */Avantis_TFG/app/src/androidTest* y */Avantis_TFG/app/src/test*: compuesto por una serie de archivos destinados a la definición y ejecución de tareas de “testing”.
- */Avantis_TFG/app/src/main*: compuesto por todos los archivos que contienen el código fuente y los recursos que darán forma a la aplicación. Este directorio también se encuentra dividido en subdirectorios:
 - */Avantis_TFG/app/src/main/res*: compuesto por todos los archivos encargados de definir las vistas de la aplicación (archivos XML) y recursos utilizados (textos, paleta de colores, temas, imágenes...).
 - **values**: carpeta donde se alojan los archivos encargados de definir los temas, los colores, las cadenas de caracteres, las dimensiones e identificadores de la aplicación.

- **menu:** directorio donde se alojan las vistas utilizadas por los menús de la aplicación.
 - **mipmap:** directorio donde se aloja el icono principal de la aplicación. Dependiendo de la densidad de píxeles del dispositivo que utilice la aplicación, este selecciona el icono adecuado.
 - **layout:** directorio que contiene todos los archivos xml referentes a las vistas de la aplicación.
 - **drawable:** directorio que contiene todos los recursos (imágenes, iconos, definiciones de estilos para elementos de la vista...) de la aplicación.
- */Avantis_TFG/app/src/main/java/com.example.avantis_tfg:* compuesto por todos los archivos encargados de definir y ejecutar la parte lógica y funcional de la aplicación (archivos Java).
 - */Avantis_TFG/app/src/main/java/AndroidManifest.xml:* archivo que se encarga de definir las características principales de la aplicación, como son el nombre, el icono principal, el estilo que utiliza, las actividades y servicios que la componen, los permisos que requiere e información esencial en forma de metadatos.

Para entender mejor cómo se encuentra distribuido el código fuente de la aplicación, se define la estructura del directorio */src/main/java/com.example.avantis_tfg/* y la funcionalidad que desempeñan los archivos que componen cada paquete:

- */src/main/java/com.example.avantis_tfg/Adapters:* paquete compuesto por todos los archivos encargados de generar las vistas contenidas de listas a partir de la interpretación de los datos.
- */src/main/java/com.example.avantis_tfg/Administrador:* paquete compuesto por todos los archivos encargados de gestionar toda la funcionalidad propia del administrador. Está compuesto por:
 - */src/main/java/com.example.avantis_tfg/Administrador/AdminAdapters*
 - */src/main/java/com.example.avantis_tfg/Administrador/HorariosManage*
 - */src/main/java/com.example.avantis_tfg/Administrador/NewsPromoManage*
 - */src/main/java/com.example.avantis_tfg/Administrador/PricesManage*
 - */src/main/java/com.example.avantis_tfg/Administrador/StationManage*
 - */src/main/java/com.example.avantis_tfg/Administrador/TPTypesManage*
 - */src/main/java/com.example.avantis_tfg/Administrador/TrayectosManage*
 - */src/main/java/com.example.avantis_tfg/Administrador/VehiculosManage*
 - */src/main/java/com.example.avantis_tfg/Administrador/ReservasManageFragment*

- */src/main/java/com.example.avantis_tfg/ForoPreguntas:* paquete compuesto por todos los archivos encargados de gestionar la funcionalidad del foro de sugerencias/temas de discusión.
- */src/main/java/com.example.avantis_tfg/InfoEstaciones:* paquete compuesto por todos los archivos encargados de gestionar la consulta de estaciones.
- */src/main/java/com.example.avantis_tfg/Login:* paquete compuesto el archivo LoginFragment.java encargado de gestionar la autenticación del usuario en la aplicación.
- */src/main/java/com.example.avantis_tfg/Main:* paquete compuesto por:
 - *MainActivity.java:* es el *activity* principal y el encargado de contener y gestionar todos los *fragments* (parte lógica y vistas) de la aplicación.
 - *BienvenidaFragment.java*
 - *InformacionPersonalFragment.java*
 - *SistemaComprasFragment.java*
- */src/main/java/com.example.avantis_tfg/MiPerfil:* paquete compuesto por todos los archivos encargados de gestionar la información relativa al usuario autenticado.
- */src/main/java/com.example.avantis_tfg/MisAbonos:* paquete compuesto por todos los archivos encargados de gestionar la consulta de abonos asociados al usuario autenticado.
- */src/main/java/com.example.avantis_tfg/MisBilletes:* paquete compuesto por todos los archivos encargados de gestionar la consulta de los billetes asociados al usuario autenticado.
- */src/main/java/com.example.avantis_tfg/Model:* paquete compuesto por todos los archivos que se encargan de definir las estructuras de datos (objetos personalizados) que se utilizan en la aplicación.
- *src/main/java/com.example.avantis_tfg/Registro:* paquete compuesto el archivo RegistroFragment.java encargado de gestionar el registro de un nuevo usuario en la aplicación.
- */src/main/java/com.example.avantis_tfg/SistemaCompraAbonos:* paquete compuesto por todos los archivos encargados de gestionar la compra de abonos.
- */src/main/java/com.example.avantis_tfg/SistemaCompraBilletesABONO:* paquete compuesto por todos los archivos encargados de gestionar la compra de billetes utilizando los abonos asociados al usuario autenticado.
- */src/main/java/com.example.avantis_tfg/SistemaCompraBilletesNOABONO:* paquete compuesto por todos los archivos encargados de gestionar la compra de billetes por parte de los usuarios.

- */src/main/java/com.example.avantis_tfg/SplashScreen:* paquete compuesto el archivo SplashScreen.java que se ejecuta al iniciar la ejecución de la aplicación y comprueba si el usuario tiene una sesión activa en la aplicación.
- */src/main/java/com.example.avantis_tfg/Utilidades:* paquete compuesto por una serie de archivos que albergan funciones útiles utilizadas en distintos aspectos de la aplicación (calendarios, barras de progreso, selector de asientos...)y evitan en gran medida replicar el código.



Capítulo 7: PRUEBAS



7.1 Pruebas de Caja Blanca

Este tipo de pruebas son llevadas a cabo en la etapa de implementación, puesto que se realizan sobre el código fuente con el fin de evaluar y comprobar la funcionalidad interna (posibles caminos lógicos) de cada uno de los módulos que componen la aplicación. Para ello, este procedimiento es realizado por los propios desarrolladores o a través del personal cualificado para esta tarea.

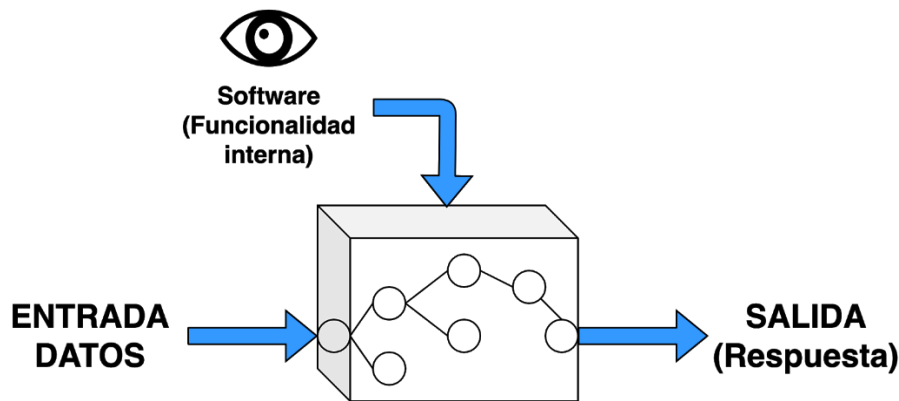


Figura 97 Pruebas de caja blanca

Debido a que el tamaño de este proyecto es bastante considerable, el número de pruebas que se han realizado también es bastante alto. Por este motivo, en este documento solo se presenta un resumen con todos los aspectos más relevantes sometidos a este tipo de pruebas:

- Comprobación de reconocimiento y distinción de roles (usuario no registrado, usuario autenticado, administrador).
- Comprobación de adaptación de interfaz a los distintos roles presentes en la plataforma.
- Comprobación de funcionalidad asociada a cada uno de los roles.
- Comprobación del control de sesión.
- Comprobación de la ejecución (comportamiento) del sistema sin conexión.
- Comprobación individual del funcionamiento y comunicación de cada uno de los servicios externos utilizados por la aplicación.
- Comprobación de peticiones a base de datos (Cloud Firestore) en tiempo real.
- Gestión de la asincronía propia de la base de datos en todo tipo de peticiones.
- Comprobación de correspondencia de datos entre la base de datos (Cloud Firestore) y la aplicación móvil (clases y objetos Java).
- Comprobaciones realizadas sobre los formularios de entrada de datos (validación de campos, tareas de autocompletado, filtrado de datos...).
- Comprobación de distinción entre los posibles sistemas de compra.

7.2 Pruebas de Caja Negra

Este tipo de pruebas se encargan de evaluar si la aplicación cumple con todos los requisitos y funcionalidades establecidas para la aplicación. Para ello, el encargado de realizar estas pruebas no requiere de ningún conocimiento previo sobre la estructura y los procesos internos utilizados por el sistema, puesto que simplemente analiza y estudia el resultado de las respuestas en base a las entradas recibidas.

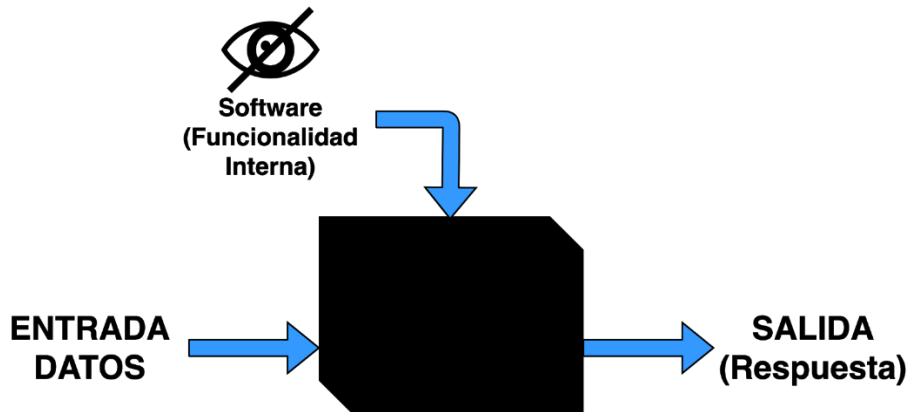


Figura 98 Pruebas de caja negra

Debido a que el tamaño de este proyecto es bastante considerable, el número de pruebas que se han realizado también es bastante alto. Por este motivo, en este documento solo se presentan algunas de las pruebas más importantes a nivel de funcionalidad básica que debe de satisfacer el sistema.

PCN-01: Registrar usuario	
Objetivo de la prueba	Evaluar la creación de nuevos usuarios desde la aplicación móvil.
Precondiciones	No exista una cuenta de usuario en el sistema con las mismas credenciales.
Datos de entrada	Nombre de Usuario: Prueba DNI: 12345678F Teléfono: 111111111 Email: prueba@avantis.com Contraseña: Prueba_1 Repetir Contraseña: Prueba_1
Respuesta esperada	La cuenta de usuario se cree correctamente en el sistema.
Resultado final	CORRECTO

Tabla 79 Prueba de caja negra "Registrar Usuario"

PCN-02: Iniciar sesión	
Objetivo de la prueba	Evaluar el acceso a la cuenta personal de un usuario a través de sus credenciales
Precondiciones	El usuario se haya registrado previamente en el sistema
Datos de entrada	Email: prueba@avantis.com Contraseña: Prueba_1
Respuesta esperada	Al pulsar “Entrar” el sistema redirecciona al usuario a la página principal y activa las funciones propias de un usuario autenticado.
Resultado final	CORRECTO

Tabla 80 Prueba de caja negra "Iniciar Sesión"

PCN-03: Selección parámetros de viaje	
Objetivo de la prueba	Comprobar que el sistema valida todos los campos necesarios para la búsqueda de un viaje.
Precondiciones	Ninguna.
Datos de entrada	Origen: Valladolid Destino: Segovia Fecha de Ida: Número de Pasajeros: 1
Respuesta esperada	Al pulsar “Buscar Viajes” el sistema notifica al usuario que el campo “Fecha de Ida” es obligatorio.
Resultado final	CORRECTO

Tabla 81 Prueba de caja negra "Seleccionar parámetros de viaje"

PCN-04: Información del pasajero	
Objetivo de la prueba	Comprobar que el sistema valida toda la información requerida de los pasajeros de viaje para continuar con el proceso de compra.
Precondiciones	<ul style="list-style-type: none"> • Seleccionar los parámetros de búsqueda del viaje. • Seleccionar uno de los viajes disponibles.
Datos de entrada	Nombre y Apellidos: Pablo Jambrina Sanz DNI: 12345678F Email: prueba@avantis.com Tarifa: Tarifa Normal

Respuesta esperada	Al pulsar “Siguiete” el sistema redirecciona al comprador a la siguiente pantalla.
Resultado final	CORRECTO

Tabla 82 Prueba de caja negra "Información del Pasajero"

PCN-05: Selección de plaza	
Objetivo de la prueba	Comprobar que el sistema permite al usuario modificar las plazas seleccionadas para un viaje en el proceso de compra.
Precondiciones	<ul style="list-style-type: none"> • Seleccionar los parámetros de búsqueda del viaje. • Seleccionar uno de los viajes disponibles. • Completar correctamente la información de los pasajeros
Datos de entrada	Ninguno.
Respuesta esperada	Al pulsar “Confirmar” la plaza asignada para el billete se actualiza a la seleccionada por el usuario.
Resultado final	CORRECTO

Tabla 83 Prueba de caja negra "Selección de plaza"

PCN-06: Selección de trayecto en la compra de abono	
Objetivo de la prueba	Comprobar que el sistema calcula correctamente la fecha de caducidad y el precio del abono en función de la selección del trayecto (origen-destino) por parte del usuario.
Precondiciones	<ul style="list-style-type: none"> • El usuario se encuentre autenticado en el sistema. • Seleccionar uno de los tipos de abono disponibles.
Datos de entrada	Origen: Valladolid Destino: Segovia
Respuesta esperada	El sistema muestra la fecha de caducidad y el precio del abono para el trayecto seleccionado, activando además el botón de “Confirmar Compra”.
Resultado final	CORRECTO

Tabla 84 Prueba de caja negra "selección de trayecto (compra de abono)"

PCN-07: Selección de parámetros en compra a través de abono	
Objetivo de la prueba	<ul style="list-style-type: none"> • Comprobar que el sistema valida la selección de fecha por parte del usuario. • Comprobar la disponibilidad de abonos para la fecha seleccionada.
Precondiciones	El usuario se encuentre autenticado en el sistema.
Datos de entrada	Fecha del Viaje: 31/12/2020
Respuesta esperada	El sistema notifica al usuario la ausencia de abonos activos para la fecha seleccionada.
Resultado final	CORRECTO

Tabla 85 Prueba de caja negra "Selección parámetros de compra (compra de billetes con abono)"

PCN-08: Buscador de estaciones	
Objetivo de la prueba	Comprobar que el filtro de búsqueda de estaciones funciona correctamente.
Precondiciones	Ninguna.
Datos de entrada	Buscador: Valladolid
Respuesta esperada	El sistema actualiza el listado mostrando las estaciones que coinciden con el criterio utilizado en el buscador.
Resultado final	CORRECTO

Tabla 86 Prueba de caja negra "Buscador de estaciones"

PCN-09: Modificar perfil de usuario	
Objetivo de la prueba	Evaluar si el sistema permite al usuario actualizar su información en cualquier momento.
Precondiciones	El usuario se encuentre previamente autenticado en la aplicación.
Datos de entrada	Nuevo nombre de usuario: Prueba Nuevo Teléfono: 222222222
Respuesta esperada	Al pulsar "Actualizar Perfil" el sistema actualiza la información del usuario.
Resultado final	CORRECTO

Tabla 87 Prueba de caja negra "Modificar perfil de usuario"

PCN-10: Agregar nueva tarjeta de crédito	
Objetivo de la prueba	Comprobar si el sistema permite asociar una tarjeta de crédito al usuario.
Precondiciones	<ul style="list-style-type: none"> • El usuario se encuentre previamente autenticado en la aplicación. • El usuario no cuente con una tarjeta de crédito asociada previamente.
Datos de entrada	Número de Tarjeta: 5432 0000 9999 8888 Nombre del Titular: PABLO JAMBRINA SANZ Fecha Caducidad: 09/23 CCV: 212
Respuesta esperada	Al pulsar “Guardar Tarjeta” la tarjeta de crédito es agregada al sistema y asociada al usuario.
Resultado final	CORRECTO

Tabla 88 Prueba de caja negra "Agregar nueva tarjeta de crédito"

PCN-11: Eliminar Billete	
Objetivo de la prueba	Comprobar si el sistema permite eliminar un billete ya adquirido por parte del usuario.
Precondiciones	<ul style="list-style-type: none"> • El usuario se encuentre previamente autenticado en la aplicación. • La fecha del viaje sea posterior a la fecha de solicitud de eliminación.
Datos de entrada	Confirmación de eliminación.
Respuesta esperada	El billete se elimina correctamente, actualizando el listado de billetes del usuario.
Resultado final	CORRECTO

Tabla 89 Prueba de caja negra "Eliminar billete"

PCN-12: Descargar Billete	
Objetivo de la prueba	Comprobar si el sistema permite la descarga del billete en formato PDF en el propio dispositivo móvil.
Precondiciones	<ul style="list-style-type: none"> • El usuario se encuentre previamente autenticado en la aplicación.

Datos de entrada	Ninguno.
Respuesta esperada	El billete se descarga correctamente y se almacena en la memoria del dispositivo móvil.
Resultado final	CORRECTO

Tabla 90 Prueba de caja negra "Descargar billete"

PCN-13: Crear nuevo tema de discusión	
Objetivo de la prueba	Comprobar si el sistema permite crear un nuevo tema de discusión en la plataforma.
Precondiciones	Ninguna.
Datos de entrada	Email de creador: prueba@avantis.com Asunto: Sugerencia de Prueba Descripción: Esta es la descripción detallada sobre la sugerencia de prueba
Respuesta esperada	Al pulsar "Publicar" la consulta/sugerencia queda registrada en el sistema.
Resultado final	CORRECTO

Tabla 91 Prueba de caja negra "Crear nuevo tema de discusión "

PCN-14: Crear nueva respuesta	
Objetivo de la prueba	Comprobar si el sistema permite crear una nueva respuesta asociada a un tema de discusión de la plataforma.
Precondiciones	<ul style="list-style-type: none"> Se debe seleccionar un tema de discusión.
Datos de entrada	Email de creador: prueba@avantis.com Descripción: Esta es la descripción detallada de la respuesta
Respuesta esperada	Al pulsar "Publicar" la respuesta queda registrada en el sistema y asociada al tema de discusión seleccionado.
Resultado final	CORRECTO

Tabla 92 Prueba de caja negra "Crear nueva respuesta"

PCN-15: Acceso a redes sociales (Twitter)	
Objetivo de la prueba	Comprobar si el sistema permite acceder al perfil público de la empresa en la red social Twitter.
Precondiciones	Ninguna.
Datos de entrada	Ninguno.
Respuesta esperada	La aplicación redirecciona al usuario a su perfil de Twitter a través de la aplicación de Twitter o navegador web del dispositivo.
Resultado final	CORRECTO

Tabla 93 Prueba de caja negra "Acceso a redes sociales"

PCN-16: Crear tipo de abono	
Objetivo de la prueba	Comprobar si el sistema permite crear un nuevo tipo de abono al administrador.
Precondiciones	El administrador debe de estar autenticado en la aplicación móvil.
Datos de entrada	Nombre de Abono: PLUS 30 Descripción: Descripción abono PLUS 30 Tasa Descuento: 30% Número de Viajes: 30 Periodo Validez: 60 días
Respuesta esperada	Al pulsar "Agregar nuevo tipo de abono" este queda registrado en el sistema
Resultado final	CORRECTO

Tabla 94 Prueba de caja negra "Crear tipo de abono"

PCN-17: Eliminar Noticia	
Objetivo de la prueba	Comprobar si el sistema permite eliminar una noticia al administrador.
Precondiciones	<ul style="list-style-type: none"> El administrador debe de estar autenticado en la aplicación móvil. La plataforma requiere de noticias registradas previamente.
Datos de entrada	Ninguno.

Respuesta esperada	Al pulsar sobre el botón de eliminación, la noticia se elimina de la plataforma actualizando el listado de las mismas.
Resultado final	CORRECTO

Tabla 95 Prueba de caja negra "Eliminar Noticia"

PCN-18: Modificar Promoción	
Objetivo de la prueba	Comprobar si el sistema permite actualizar la información relativa a una promoción al administrador.
Precondiciones	<ul style="list-style-type: none"> • El administrador debe de estar autenticado en la aplicación móvil. • La plataforma requiere de promociones registradas previamente.
Datos de entrada	Título: PROMOCION 1 Descripción: Esta es la descripción de la promoción 1 Tasa Descuento: 25%.
Respuesta esperada	Al pulsar sobre "Modificar" se actualiza la información de la promoción seleccionada.
Resultado final	CORRECTO

Tabla 96 Prueba de caja negra "Modificar promoción"

PCN-19: Añadir vehículo	
Objetivo de la prueba	Comprobar si el sistema crear un nuevo vehículo al administrador.
Precondiciones	<ul style="list-style-type: none"> • El administrador debe de estar autenticado en la aplicación móvil. • La plataforma requiere de horarios registrados previamente.
Datos de entrada	Identificador: L-120 Empresa de transportes: AVANTIS TRANS. Número de plazas: 0 Horarios de circulación {horario1, horario2}
Respuesta esperada	Al pulsar sobre "Añadir" el sistema muestra una notificación al administrador notificando que el número de plazas debe de ser superior a 0 y el vehículo no se registra en la base de datos.
Resultado final	CORRECTO

Tabla 97 Prueba de caja negra "Añadir vehículo"

PCN-20: Cerrar sesión	
Objetivo de la prueba	Comprobar si el sistema permite a cualquier usuario autenticado cerrar sesión en la aplicación.
Precondiciones	El usuario se encuentre autenticado.
Datos de entrada	Ninguno.
Respuesta esperada	El sistema muestra la pantalla del menú principal.
Resultado final	CORRECTO

Tabla 98 Prueba de caja negra "Cerrar Sesión"



Capítulo 8: MANUALES



8.1 Manual de Despliegue

La aplicación desarrollada, al consumir el *backend* como un servicio, no requiere nada más que la instalación en el dispositivo para empezar a utilizarla. Para realizar la instalación será necesario:

1. Es requisito esencial tener habilitada la opción en el dispositivo que nos permite la instalación de aplicaciones con un origen desconocido (aplicaciones que no provienen de plataformas oficiales como Google Play).
2. Descargar el archivo *.apk* de la aplicación directamente en el dispositivo o incorporar el mismo a través de un cable USB o tarjeta SD.
3. Una vez se cuente con el archivo de instalación, solo se debe de ejecutar para comenzar el proceso.
4. Si la aplicación utiliza alguno de los servicios de Google y por cualquier motivo estos se encuentran desactualizados, el sistema solicita al usuario la actualización de estos.
5. Una vez finalizada la instalación, la aplicación requiere de una serie de permisos para llevar a cabo ciertas funcionalidades, como por ejemplo la descarga de un billete en el dispositivo.
 - a. El sistema solicitará al usuario que de su consentimiento para estos permisos.

Con todo esto, la aplicación queda instalada 100% en el dispositivo con todas sus funciones y permisos activados y disponibles en cualquier momento.

8.2 Manual de Usuario

En este apartado y después de haber analizado todos los aspectos técnicos referentes a la funcionalidad, requisitos y diseño, es momento de exponer y explicar (desde un punto de vista más coloquial) el funcionamiento básico de la aplicación desarrollada.

Esto será un gran punto de partida para todo tipo de personas que quieran aprender y sacar el máximo partido a esta herramienta, además de proporcionar a los usuarios un soporte donde poder consultar cualquier tipo de duda referente al manejo en todo momento.

8.2.1 ¿Empezamos?

Nada más acceder a la aplicación se muestra la pantalla principal. En esta pantalla el usuario podrá consultar las últimas noticias y promociones disponibles pulsando el botón “Leer Más” (presente en cada una de ellas).

Por otro lado, pulsando el icono de la esquina superior izquierda o deslizando la pantalla hacia la derecha se puede acceder al menú lateral para acceder a toda la funcionalidad propuesta por la aplicación.

8.2.2 Compra de billetes estándar

Para acceder a la compra de billetes estándar, el usuario/comprador deberá seguir los siguientes pasos:

1. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
2. Pulsar “Compra Billeto / Abono”.
3. Pulsar “Consulta Horario / Compra Billeto”.
4. Seleccionar los criterios de búsqueda del viaje y pulsar “Buscar Viajes”.
5. Seleccionar el viaje pulsando el botón donde aparece el precio de este.
 - a. Si se selecciona un **viaje ida y vuelta**, primero se selecciona el viaje de ida y a continuación se selecciona el de vuelta antes de avanzar al siguiente paso.
6. Completar la información de TODOS los pasajeros
 - a. **Si el comprador se encuentra autenticado**, se completará automáticamente la información del pasajero autenticado teniendo que completar manualmente la información del resto de pasajeros.
 - b. **En caso contrario** será el comprador el que complete la información manualmente.
7. Seleccionar la tarifa o promoción que se aplicará a cada pasajero y pulsar “Siguiente”.
8. El comprador cuenta con la posibilidad de modificar las plazas asignadas para cada uno de los viajes solicitados.
 - a. En caso afirmativo:
 - i. El comprador pulsa el botón “Modificar” para el viaje objeto del cambio.
 - ii. Se mostrará un panel donde el comprador selecciona la nueva plaza en función de las plazas vacantes para el viaje seleccionado.
 - iii. El comprador pulsará el botón “Confirmar” para ejecutar el cambio de plaza.
 - iv. Por último, el comprador pulsará el botón “Siguiente” para continuar con el proceso de compra.
 - b. Si no se desea modificar las plazas, el usuario pulsa el botón “Siguiente” para continuar con el proceso de compra.
9. El sistema muestra un resumen con la información de compra.
10. El comprador selecciona el método de pago:
 - a. Tarjeta de Crédito:
 - i. En caso de que el comprador se encuentre autenticado y cuente con una tarjeta de crédito asociada previamente, el sistema obtiene automáticamente los datos de esta.
 - ii. En cualquier otro caso, el comprador deberá introducir los datos de la tarjeta de crédito de forma manual.
 - b. PayPal
11. El comprador pulsa el botón “Realizar Pago” para confirmar la compra.
12. El sistema redirige a la plataforma de PayPal, donde el comprador introduce sus claves para la realización del pago.
13. Por último, el sistema mostrará al comprador el/los billetes adquiridos junto con la posibilidad de:
 - a. Descargar el billete en formato PDF en el dispositivo.
 - b. Enviar el billete a través del correo electrónico.

8.2.3 Compra de abonos

Para acceder a la compra de billetes, el usuario/comprador deberá seguir los siguientes pasos:

1. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
2. Pulsar “Compra Billete / Abono”.
3. Pulsar “Comprar Abono”.
4. El comprador selecciona el tipo de abono que desea comprar pulsando el botón “Comprar Abono”
5. El comprador selecciona el trayecto (Origen-Destino) para el que desea el abono.
 - a. Si el trayecto no es válido, el sistema no permite la compra de ese abono al comprador notificando dicha incidencia.
6. El comprador pulsa el botón “Confirmar Compra” para continuar con el proceso de compra.
7. El sistema redirige a la plataforma de PayPal, donde el comprador introduce sus claves para la realización del pago.
8. El sistema confirma la operación y la compra se efectúa con éxito.

8.2.4 Compra de billetes utilizando un abono asociado

Para acceder a la compra de billetes, el usuario/comprador deberá seguir los siguientes pasos:

1. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
2. Pulsar “Compra Billete / Abono”.
3. Pulsar “Utilizar Abono”.
4. El comprador selecciona la fecha del viaje y el abono disponible.
 - a. En caso de no contar con abonos disponibles para la fecha seleccionada, el sistema notificará la incidencia al comprador.
5. El comprador selecciona el sentido del trayecto.
6. Seleccionar el viaje pulsando el botón donde aparece el precio de este.
7. El comprador cuenta con la posibilidad de modificar la plaza asignada para el viaje escogido.
 - a. En caso afirmativo:
 - i. El comprador pulsa el botón “Modificar”.
 - ii. Se mostrará un panel donde el comprador selecciona la nueva plaza en función de las plazas vacantes en el viaje.
 - iii. El comprador pulsará el botón “Confirmar” para ejecutar el cambio de plaza.
 - iv. Por último, el comprador pulsará el botón “Finalizar” para obtener su billete asociado al abono.
 - b. Si no se desea modificar las plazas, el usuario pulsa el botón “Siguiente” para continuar con el proceso de compra.
8. Por último, el sistema mostrará al comprador el/los billetes adquiridos junto con la posibilidad de:
 - a. Descargar el billete en formato PDF en el dispositivo.
 - b. Enviar el billete a través del correo electrónico.

8.2.5 Preguntas frecuentes

Para acceder al foro de discusión de la aplicación, el usuario deberá seguir los siguientes pasos:

1. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
2. Pulsar “Preguntas Frecuentes”.

Una vez llegado a este punto, el usuario podrá observar un listado con todos los temas de discusión, sugerencias, quejas y dudas propuestas por parte de los usuarios de la aplicación. Desde aquí, el usuario podrá realizar las siguientes acciones:

- **Crear nuevo tema de discusión:** para ello el usuario debe realizar los siguientes pasos:
 - Pulsar sobre el icono (barra superior derecha) para desplegar el formulario.
 - Completar el formulario con la información requerida.
 - Pulsar el botón “ ” para publicar el nuevo tema de discusión en la plataforma.
- **Consultar información sobre un tema de discusión:** para ello el usuario debe realizar los siguientes pasos:
 - Pulsar sobre el botón “ X RESPUESTAS”.
- **Crear una nueva respuesta para un tema de discusión:** para ello el usuario debe realizar los siguientes pasos:
 - Pulsar sobre el botón “ X RESPUESTAS”.
 - Una vez el usuario se encuentre en la pantalla de consulta información de un tema de discusión, se pulsa sobre el botón “NUEVA RESPUESTA” para desplegar el formulario.
 - Completar el formulario con la información requerida.
 - Pulsar el botón “ ” para publicar la respuesta en la plataforma.

8.2.6 Consulta de estaciones

Para acceder a la consulta de estaciones, el usuario deberá seguir los siguientes pasos:

1. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
2. Pulsar “Consulta Estación”.

Una vez llegado a este punto, el usuario podrá observar un listado con todas las estaciones disponibles en la plataforma. Para consultar la información de una estación en particular, el usuario cuenta con los siguientes métodos:

- **Utilizando el buscador de estaciones:** para ello el usuario debe realizar los siguientes pasos:
 - Pulsar sobre el icono (esquina superior derecha) para desplegar el buscador.
 - Introducir el nombre de la estación que se desea encontrar
 - El sistema mostrará los resultados de la búsqueda.
 - Pulsar sobre la estación para desplegar su información.
- **De forma manual:** para ello el usuario debe realizar los siguientes pasos:

- Desplazarse sobre el listado de estaciones.
- Pulsar sobre la estación para desplegar su información.

8.2.7 Consultar información personal

Para acceder a la información personal de la cuenta, el usuario deberá seguir los siguientes pasos:

1. Es requisito indispensable que el usuario se encuentre AUTENTICADO en la aplicación.
2. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
3. Pulsar “Información Personal”.
4. Pulsar “Mi Perfil”.

Una vez llegado a este punto, el usuario podrá observar los datos personales de su cuenta. Desde aquí, el usuario podrá realizar las siguientes acciones:

- **Modificar perfil:** para ello el usuario debe realizar los siguientes pasos:
 - Pulsar sobre el icono (barra superior derecha) para desplegar el formulario.
 - Modificar los campos con la nueva información personal.
 - Pulsar el botón “ACTUALIZAR PERFIL ” para actualizar la información del usuario.

8.2.8 Método de pago (Tarjeta de crédito)

Para acceder al método de pago asociado y configurado para la cuenta, el usuario deberá seguir los siguientes pasos:

1. Es requisito indispensable que el usuario se encuentre AUTENTICADO en la aplicación.
2. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
3. Pulsar “Información Personal”.
4. Pulsar “Tarjeta de Crédito”.

Una vez llegado a este punto, el usuario podrá observar:

- En caso de contar con una tarjeta asociada previamente, la información de la tarjeta de crédito asociada.
- En caso contrario, el sistema mostrará al usuario un mensaje notificando la falta de una tarjeta asociada.

Desde aquí, el usuario podrá realizar las siguientes acciones:

- **Asociar tarjeta de crédito:** para ello el usuario debe realizar los siguientes pasos:
 - *CONDICIÓN* El usuario no debe de contar con una tarjeta asociada previamente.
 - Pulsar sobre el icono (barra superior derecha) para desplegar el formulario.
 - Completar el formulario con la información requerida.
 - Pulsar el botón “ ” para asociar la tarjeta al perfil del usuario.
- **Desvincular tarjeta de crédito:** para ello el usuario debe realizar los siguientes pasos:
 - *CONDICIÓN* El usuario debe de contar con una tarjeta asociada previamente.

- Pulsar sobre el icono (barra superior derecha).
- El usuario confirma la petición de desvinculación.
- La tarjeta de crédito queda desvinculada del perfil del usuario.

8.2.9 Consultar mis viajes

Para acceder a los billetes adquiridos por parte del usuario, este deberá seguir los siguientes pasos:

1. Es requisito indispensable que el usuario se encuentre AUTENTICADO en la aplicación.
2. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
3. Pulsar “Información Personal”.
4. Pulsar “Mis Viajes”.

Una vez llegado a este punto, el usuario podrá observar todos los billetes ordenados por la fecha del viaje. Desde aquí, el usuario podrá realizar las siguientes acciones sobre cada billete de forma individual:

- **Descargar billete:** para ello el usuario debe realizar los siguientes pasos:
 - Pulsar sobre el botón de descarga.
 - El billete en formato PDF se almacenará en el dispositivo móvil.
- **Visualizar información de billete:** para ello el usuario debe realizar los siguientes pasos:
 - Pulsar sobre el botón de visualización.
 - El sistema muestra la información del billete junto a su código QR.
- **Eliminar billete:** para ello el usuario debe realizar los siguientes pasos:
 - *CONDICIÓN* La fecha del viaje sea posterior al momento de solicitar la eliminación de este.
 - Pulsar sobre el botón de eliminación.
 - El usuario confirma la petición de anulación de billete.
 - El sistema eliminará y actualizará el listado de billetes automáticamente.

8.2.10 Consultar mis abonos

Para acceder a la información de los abonos adquiridos por parte del usuario, este deberá seguir los siguientes pasos:

1. Es requisito indispensable que el usuario se encuentre AUTENTICADO en la aplicación.
2. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
3. Pulsar “Información Personal”.
4. Pulsar “Mis Abonos”.

Una vez llegado a este punto, el usuario podrá observar todos los abonos con los que cuenta el usuario. Desde aquí, el usuario podrá realizar las siguientes acciones sobre cada abono de forma individual:

- **Consultar detalles de abono:** para ello el usuario debe realizar los siguientes pasos:

- Pulsar sobre el botón “CONSULTA”.
- El sistema muestra los detalles del abono seleccionado junto a los billetes que tiene asociados, pudiendo el usuario realizar las mismas operaciones sobre los mismos definidas previamente en “CONSULTAR MIS VIAJES”.

8.2.11 Iniciar sesión

Para iniciar sesión en la aplicación y contar con la funcionalidad adicional proporcionada por ésta a los usuarios previamente registrados, el usuario deberá seguir los siguientes pasos:

1. Es requisito indispensable que el usuario no cuente con una sesión activa previa en la aplicación.
2. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
3. Pulsar “Acceder a Avantis”.
4. Completar el formulario con las credenciales del usuario.
5. Pulsar el botón “ENTRAR”.

8.2.12 Registro de nuevo usuario

Para crear una nueva cuenta en la plataforma y acceder a los servicios y funcionalidad adicional proporcionada a los usuarios registrados, el usuario deberá seguir los siguientes pasos:

1. Es requisito indispensable que el usuario no cuente con una sesión activa previa en la aplicación.
2. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
3. Pulsar “Acceder a Avantis”.
4. Pulsar en la etiqueta (parte inferior del formulario) “¿Aun no es usted miembro? Regístrese”.
5. Completar el formulario con la información solicitada por el sistema.
6. Pulsar el botón “CONFIRMAR REGISTRO” para completar el registro.

8.2.13 Acceso a redes sociales (Twitter)

Para consultar el perfil público de la compañía en la red social Twitter, el usuario deberá seguir los siguientes pasos:

1. Acceder al “Menú Lateral” (esquina superior izquierda o desplegar).
2. Pulsar “Twitter”.
3. El sistema redirige al usuario a la cuenta pública de Avantis en Twitter.



Capítulo 9: CONCLUSIONES Y FUTURAS AMPLIACIONES



9.1 Conclusiones

El primer aspecto reseñable una vez finalizado el proyecto, que para mí es el más importante de todos, es la falta de experiencia previa en la gestión y desarrollo de proyectos de esta envergadura. Probablemente este sea uno de los primeros desarrollos de mi vida que no está sujeto a las pautas y controles programados para cualquier otro proyecto académico realizado con anterioridad. Esto ha supuesto grandes retos a la hora de fijar objetivos, analizar los requisitos necesarios para satisfacer los mismos, asumir diferentes roles dentro del proyecto y planificar todas estas tareas, de tal manera que el proyecto se ajuste a la realidad lo máximo posible. Todo esto ha generado incertidumbre debido a que a medida que avanzaba el proyecto, los requisitos y las propuestas crecían a un ritmo muy alto despertando en mí el miedo de dejarme cosas en el tintero, aunque también despertando una capacidad de análisis selectivo de la cual estoy muy orgulloso.

Por otro lado, me he encontrado con cierto grado de dificultad a la hora de trabajar con bases de datos NoSQL. Al carecer de experiencia previa en este aspecto, las horas dedicadas a la investigación, el estudio y la adaptación a los conocimientos con los que anteriormente contaba en bases de datos SQL ha sido intenso y gratificante a partes iguales. Derivado de esta situación aparece Firebase, más concretamente su servicio de base de datos en tiempo real *Cloud Firestore*. 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.

Además de todos estos aspectos se suma la ampliación de conocimientos en la programación Android. A pesar de contar con una gran base, han sido necesarias muchas horas de dura investigación, consulta de tutoriales, manuales y ensayo prueba-error en solitario para conseguir incorporar a este proyecto distintos aspectos y funcionalidades que parecían muy lejanas o imposibles de implementar al principio de esta aventura. Reseñable también ha sido la dificultad de implementar y adaptar servicios y librerías externas al proyecto como son *PayPal*, *Google Play Services*, *Google Maps* o *Gmail*.

En definitiva, este proyecto ha sido un reto personal para mí en todos los sentidos. Empezó siendo algo que parecía ser simple y ha terminado con unos niveles de complejidad bastante altos. Esto ha provocado que en muchos momentos la sensación de no avanzar, de no ver el final, de tantas horas de investigación, de pruebas fallidas, de modificaciones, ampliaciones y soledad casi haya podido con mis fuerzas.

Pero al final no, este proyecto ha conseguido que me haga mucho más fuerte. Ha conseguido despertar en mí un interés por seguir aprendiendo más y más en este mundo del desarrollo de aplicaciones y poder aplicar ese conocimiento en mis propios proyectos (este en particular). También me he dado cuenta de la importancia de contar con un buen equipo de trabajo que ayude en las distintas etapas de desarrollo y sirvan de apoyo y desahogo para no decaer en proyectos de tal envergadura. Pero a pesar de todo, son muchos los nuevos conocimientos adquiridos y experiencias vividas que me servirán de aquí en adelante en futuros desafíos. Además, hay muchos aspectos en los que me gustaría profundizar más y estoy seguro de que con un poco de tiempo y estas ganas que tengo podré llevarlo a cabo sin ningún problema en un futuro próximo.

9.2 Futuras Ampliaciones

Como he descrito en el apartado anterior, a lo largo del desarrollo del proyecto y motivadas por la ganancia de experiencia a medida que se avanzaba, han surgido propuestas y sugerencias que se han tenido que desestimar puesto que se alejaban de los objetivos establecidos en un principio para esta primera versión y además provocaban una gran variación en la planificación temporal establecida para el mismo.

A pesar de estos condicionantes, esta serie de ideas siguen formando parte del amplio abanico de posibilidades de mejora con los que cuenta la aplicación en futuros desarrollos.

- **Unificación de cuentas para autenticación de usuarios:** esta mejora consiste en proporcionar a la aplicación la posibilidad de iniciar sesión con las credenciales utilizadas por el usuario en sus cuentas de Facebook o Google, por ejemplo.
- **Aumento de funcionalidad del foro de discusión:** esta mejora abarca diferentes aspectos como son:
 - Cerrar tema de discusión por parte del creador.
 - Buscador de temas de discusión
 - Organizar temas de discusión (por fecha de publicación, estado, número de respuestas...).
 - Notificar nueva respuesta al creador del tema de discusión en tiempo real.
 - “Me gusta” para temas de discusión y respuestas.
 - “Compartir” para temas de discusión y respuestas.
 - Eliminar respuestas o temas de discusión por parte de los administradores.
- **Bot asistente virtual:** esta mejora implica la creación de un asistente virtual a través de inteligencia artificial para ayudar a los usuarios de la aplicación.
- **Notificación al usuario “Viaje Próximo”:** esta mejora consiste en notificar y prevenir al usuario que su viaje está cerca de realizarse.
- **Billetes en más formatos:** como por ejemplo en *Passbook* o enviar el billete en PDF a través del correo electrónico.
- **Digitalización de billetes físicos:** esta mejora implica poder visualizar el billete físico en la aplicación a través del localizador del billete y el DNI del pasajero.
- **Notificación de nuevas promociones:** esta mejora consiste en proporcionar a los usuarios información en tiempo real de estos descuentos.
- **Información en tiempo real de los paneles de las estaciones:** esta mejora implica obtener la información de las llegadas y salidas de cada estación para que el usuario pueda consultarla en cualquier momento.
- **Sugerencias de viajes:** información sobre los posibles trayectos interesantes para el usuario.

- **Filtro de visualización en “Mis Viajes”:** esta mejora proporciona al usuario la posibilidad de seleccionar los billetes que se muestran por pantalla (todos, viajes ya realizados, viajes pendientes).



Capítulo 10: BIBLIOGRAFÍA Y REFERENCIAS



Para finalizar este documento se presentan las distintas fuentes que se han consultado a lo largo del desarrollo del proyecto. Estas se van a separar por un lado en libros de consulta, por otro en páginas web (principalmente artículos de consulta, foros de desarrolladores y manuales en línea) y por último en contenido audiovisual. Para evitar que el tamaño de este apartado se incremente en exceso y debido a que se han consultado muchos contenidos en diferentes posts del mismo portal web, solo se incluyen las URL principales de los distintos sitios web.

Una vez aclarada esta premisa y para terminar esta introducción se especifica que la fecha de consulta de todos los sitios web corresponde al día 15/01/2021, día en el cual se procedió a la elaboración de este epígrafe.

10.1 Bibliografía

- **El Gran Libro de Android.** Editorial S.A. MARCOMBO. Jesús Tomás Gironés. 2020. ISBN: 978-8426731937
- **El Gran Libro de Android Avanzado.** Editorial S.A. MARCOMBO. Jesús Tomás Gironés. 2018. ISBN: 978-8426726650
- **Firestore: trabajar en la nube.** Editorial S.A. MARCOMBO. Vicente Carbonell. 2018. ISBN: 978-8426726605
- **Desarrollo de Aplicaciones con Android.** Editorial RA-MA. Jorge Santiago Nolasco Valenzuela. 2019. ISBN: 978-8499648101
- **Desarrollo de Aplicaciones para Android.** Editorial ANAYA MULTIMEDIA. Joan Ribas Lequerica. 2017. ISBN: 978-8441538924

10.2 Webgrafía

- **Google Cloud – Cloud Computing Services**
Disponible en: <https://cloud.google.com/>
- **Cloud Firestore | Firebase Google**
Disponible en: <https://firebase.google.com/docs/firestore>
- **Firebase Authentication | Firebase Google**
Disponible en: <https://firebase.google.com/docs/auth>
- **Paypal Developer - Test in the sandbox**
Disponible en: <https://developer.paypal.com/docs/integration/paypal-here/sandbox-testing/using-sandbox-accounts/>
- **Material Design Android - Build beautiful, usable products using Material Components for Android.**
Disponible en: <https://material.io/develop/android>
- **Android Developers – Android Developers**
Disponible en: <https://developer.android.com/>
- **Stack Overflow - We <3 people who code**
Disponible en: <https://stackoverflow.com/>
- **Github - Where the world builds software.**
Disponible en: <https://github.com>

- **The Android Arsenal - *Android developer portal with tools, libraries, and apps***
Disponible en: <https://android-arsenal.com/>
- **Coding in Flow – *Android Tutorials***
Disponible en: <https://codinginflow.com/>
- **Tutlane – *Android Tutorial***
Disponible en: <https://www.tutlane.com/tutorial/android>
- **entreDesarrolladores - *Recibe ayuda de expertos***
Disponible en: <https://entredesarrolladores.com/>
- **Desarrollador Android - *Aprende a desarrollar apps Android.***
Disponible en: <https://desarrollador-Android.com/>
- **SGOliver – *Programación en Android, .Net y Java.***
Disponible en: <https://www.sgoliver.net/>
- **Ready Android - *Pursuing Perfection Passionately***
Disponible en: <https://readyandroid.wordpress.com/>
- **Codexpedia – *Android***
Disponible en: <https://www.codexpedia.com/category/Android/>
- **La Web del Programador - *Códigos Fuente de Android***
Disponible en:
<https://www.lawebdelprogramador.com/codigo/Android/index1.html>
- **danielme.com - *Programación Android, Spring y JEE – Android***
Disponible en: <https://danielme.com/android/>
- **Medium - *Using of ConstraintLayout to build out CardView with Material Design***
Disponible en: <https://medium.com/@eugenebrusov/using-of-constraintlayout-to-build-out-cardview-with-material-design-e111e64575c2>
- **Medium - *Mi primer app con Firebase y Android***
Disponible en: <https://medium.com/@ykro/mi-primer-app-con-firebase-y-android-parte-1-fd0b7d717e0b>
- **Medium - *Android: onBackPressed() for Fragments***
Disponible en: <https://medium.com/@Wingnut/onBackPressed-for-fragments-357b2bf1ce8e>
- **C-Sharp Corner - *How To Generate QR Code In Android***
Disponible en: <https://www.c-sharpcorner.com/article/how-to-generate-qr-code-in-android/>
- **Android Learner Site - *Fragment lifecycle during Fragment transaction***
Disponible en: <https://androidlearnersite.wordpress.com/2017/02/27/fragment-lifecycle-during-fragment-transaction/>
- **Protocoders Point - *How to check internet connection in android using Broadcast Receiver***
Disponible en: <https://protocoderspoint.com/how-to-check-internet-connection-in-android/>
- **Raywenderlich - *TapTargetView for Android Tutorial***
Disponible en: : <https://www.raywenderlich.com/5194-taptargetview-for-android-tutorial>

- **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/>
- **Graph Everywhere - *Bases de Datos NoSQL | Qué son, marcas, tipos y ventajas***
Disponible en: <https://www.graph everywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>
- **MongoDB - *Fundamentos de las bases de datos NoSQL***
Disponible en: <https://www.mongodb.com/es/nosql-explained>

10.3 Complementos audiovisuales

En este apartado se nombran todos aquellos contenidos audiovisuales, tutoriales y cursos online consultados para la realización del proyecto.

- **Youtube | PildorasInformáticas – *Curso de Android con Android Studio***
Disponible en: <https://www.youtube.com/watch?v=pdYkmCcQFd8&list=PLU8oAIHdN5Bkn-KS1sRFISEnXXcAtAJ9P>
- **Youtube - canal MoureDev by Brais Moure**
Disponible en: <https://www.youtube.com/c/MouredevApps/playlists>
- **Youtube | CodigoFacilito – *Curso Programación Android***
Disponible en: <https://www.youtube.com/watch?v=sS3oDIcHNFo&list=PLpOqH6AE0tNjZdEeWgHvxkXoeoPCvDGPL>
- **Youtube – Cristian Henao – *Curso Android desde Cero***
Disponible en: <https://www.youtube.com/watch?v=2b7bXOiOA38&list=PLAg6Lv5BbjdvIcLQdVg4ROZnfuuQcqXB>



Anexo I: Firebase, PayPal y Cloud Firestore



Firebase



Firebase

Figura 99 Logotipo Firebase

Descripción General

Firebase se define como una plataforma digital compuesta por una serie de herramientas y servicios destinados al desarrollo de aplicaciones móviles o web. Fue lanzado al mercado en un principio en 2011, aunque fue a partir de 2014 con la compra de esta por parte de Google cuando el crecimiento y potenciación de las funcionalidades empezaron a adquirir una gran importancia en el mercado.

El modelo de distribución de esta plataforma responde al paradigma SaaS (*Software as a Service*) debido a que su soporte lógico y los datos se encuentran en servidores pertenecientes a empresas y compañías tecnológicas externas. Por este motivo, para acceder a estos servicios será necesario contar con una conexión a Internet por parte de los usuarios/desarrolladores.

Las soluciones y servicios que proporciona buscan satisfacer:

1. Desarrollo rápido de productos software, aislando a todos los usuarios de la infraestructura interna.
2. Escalabilidad automática proporcionada por aplicaciones que cuentan con el respaldo y confianza de Google.
3. Trabajo en conjunto, destacando que todos los servicios individualmente trabajan sin ningún problema, pero buscando siempre la utilización conjunta de los mismos que genera aplicaciones robustas y de muy alta calidad.

Para satisfacer estos objetivos, dichos servicios se pueden clasificar en 4 aspectos fundamentales:

1. **Desarrollo:** herramientas utilizadas en la creación y construcción de la aplicación, principalmente encargadas de dar soporte de almacenamiento de datos, autenticación, hosting, configuración y funcionalidad.
2. **Crecimiento:** herramientas centradas en la comunicación y captación de usuarios, junto con herramientas capaces de pronosticar comportamientos o situaciones futuras que ayudarán en las actualizaciones de la aplicación.
3. **Calidad:** herramientas centradas en el mantenimiento, pruebas y distribución de la aplicación.
4. **Analítica:** Google Analytics es la herramienta encargada de monitorizar y generar informes en todo momento el comportamiento y estadísticas de la aplicación.

Plataforma completa de desarrollo de aplicaciones

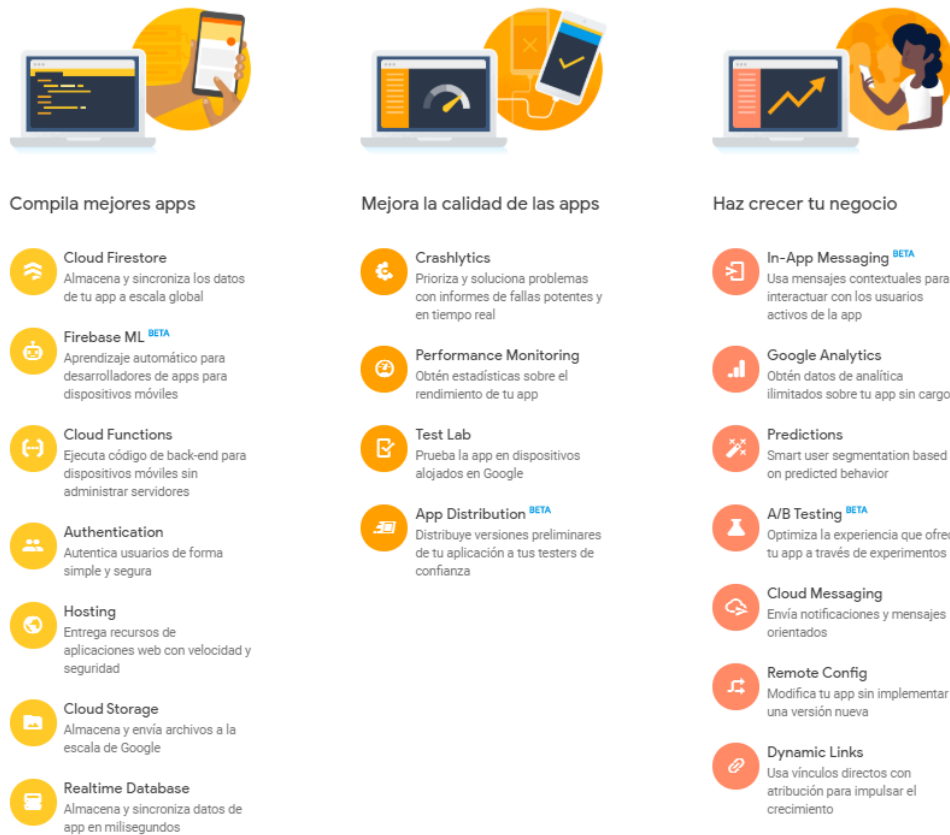


Figura 100 Servicios proporcionados por Firebase

¿Por qué la elección de Firebase en el proyecto?

En la actualidad uno de los problemas más frecuentes que se presentan en el desarrollo de aplicaciones móviles es la escalabilidad. Este problema surge de la falta de adaptabilidad a raíz del aumento del número de usuarios de la herramienta. En este proyecto se tiene en especial consideración puesto que, a pesar de estar concebido en un principio para un número no muy elevado de usuarios (habitantes de la provincia de Segovia mayoritariamente), la idea es que en un futuro abarque un número de estos mucho mayor, lo que supone también un aumento en el número de dispositivos y plataformas que deben soportar correctamente la funcionalidad de la aplicación. Si este riesgo no se controla debidamente, las esperanzas puestas en el futuro de la aplicación se terminarán antes de empezar.

Por otro lado, y en particular un aspecto esencial que se debe gestionar en nuestra aplicación es garantizar la información a tiempo real para todos los usuarios. El objetivo principal es la compra de billetes rápida y sencilla por parte del usuario, y para cumplir con ello es necesario que la información de los viajes y la disponibilidad de la información requerida por el usuario se encuentre actualizada y sea accesible en tiempo real, sin esperas engorrosas que pondrían en serio peligro también el futuro del sistema en el mercado.

Por estos dos motivos se ha optado por la propuesta y utilización de Firebase y algunos de sus potentes servicios. Por un lado, Firebase no solo está disponible para desarrollos basados en Android, si no para otra serie de plataformas como son IOS, Unity, C++ o desarrollos web, lo que nos solventa enormemente los problemas de escalabilidad. Además, a todo esto se añade la capacidad añadida por el servicio de base de datos de Cloud Firestore, que permite a los usuarios disponer de la información en tiempo real, mejorando la escalabilidad y la sincronización, con lo que los dos principales inconvenientes quedan totalmente solventados y controlados por parte de los administradores y desarrolladores, que además cuentan con el respaldo y la confianza aportada por Google y su servicio técnico, que como anteriormente se ha explicado, es el encargado de gestionar toda la infraestructura que soporta este entramado de servicios.

Servicios de Firebase utilizados en el proyecto

Para este proyecto, en un principio se ha contado con el uso de los siguientes servicios:

1. **Cloud Firestore:** Esencial para nuestro proyecto y encargado de proporcionar a la aplicación una base de datos en la nube, totalmente escalable, flexible y configurable que permite acceder a los usuarios a la información albergada en tiempo real. También cuenta con la capacidad de establecer reglas para restringir la accesibilidad a la información. Los datos se almacenan en estructuras denominadas “colecciones”, las cuales a su vez contendrán los registros conocidos como “documentos”.
2. **Firestore Authentication:** servicio cuya función principal es la gestión de los usuarios de la aplicación.

PayPal



Figura 101 Logotipo PayPal

Descripción General

PayPal se conoce como una plataforma gratuita creada para la realización de pagos en línea en cualquier momento y en cualquier lugar. Requiere que el usuario cuente con una cuenta PayPal con sus tarjetas de crédito o cuenta bancaria asociada.

Una vez se cuente con ello, la realización del pago por parte del usuario será tan sencillo como identificarse con sus credenciales y seleccionar el método con el que desea realizar dicha transacción.

El usuario también tiene la posibilidad de gestionar y controlar la información de su cuenta y transacciones en todo momento.

Beneficios del uso de PayPal

Las ventajas de realizar los pagos a través de esta plataforma son numerosas, tales como:

1. **Rapidez:** solo será necesario acceder con el usuario y contraseña y seleccionar el método de pago para disfrutar del servicio.
2. **Facilidad:** ya no será necesario recordar los datos de tu tarjeta de crédito o cuenta bancaria a la hora de realizar un pago.
3. **Seguridad:** los datos del comprador siempre estarán protegidos. Además, se contará con la asistencia total de PayPal en todos los pagos que cumplan con los requisitos de la Protección del Comprador de PayPal.
4. **Es gratuito,** tanto para la creación de una nueva cuenta como en la realización de pagos.

¿Por qué la elección de PayPal en el proyecto?

La aplicación, en un principio, no requiere de la realización de pagos “reales”, sino que lo que se pretende es dotar a la misma de una plataforma capaz de simular estos pagos sin necesidad de involucrar dinero real.

Después de investigar múltiples alternativas de mercado, se han encontrado problemas en muchas de ellas principalmente provocados por los siguientes motivos:

- No contaban con la funcionalidad de simulación y requerían de la realización de micro pagos que involucran dinero real, y que por lo tanto son transacciones reales.
- Dificultad de integración en el desarrollo de la aplicación Android.

A raíz de esta investigación se encontró la posibilidad de integrar y utilizar en la aplicación el entorno de pruebas de PayPal, conocido comúnmente como “*Sandbox*”. Esta opción permite la creación de una cuenta de negocio y personal ficticias y totalmente personalizables por parte de un usuario (desarrollador) que cuente previamente con una cuenta PayPal (ejercerá de administrador de las cuentas ficticias).

El pago se realiza utilizando las credenciales ficticias de la cuenta personal sin involucrar dinero real, apareciendo el mismo en la cuenta de negocio. Además, iniciando sesión con estas credenciales ficticias, se pueden controlar todos los pagos realizados y tener un control total del estado de estos.

Además de proporcionar todo lo necesario que requiere el proyecto, la integración de esta plataforma requiere de una serie de pasos muy sencillos, lo que reduce en gran medida el tiempo de desarrollo del sistema de compras, funcionalidad esencial de la aplicación.

Cloud Firestore



Figura 102 Logotipo Cloud Firestore

Descripción General

Cloud Firestore es un servicio de base de datos en la nube, NoSQL, flexible y escalable proporcionado por Firebase, que permite la sincronización de datos en tiempo real entre las aplicaciones cliente al contar con agentes de escucha de eventos en tiempo real capaces de notificar cualquier evento sucedido en la base de datos. Se utiliza principalmente en la programación de servidores, aplicaciones móviles y aplicaciones web.

En Firestore, la información/datos se encuentra almacenada en “Documentos” compuestos por campos que siguen el patrón clave-valor. Los tipos de datos admitidos son numerosos y abarcan desde cadenas de caracteres hasta objetos complejos.

Los documentos se pueden agrupar en “Colecciones” que facilitan las tareas de organización de los datos y la creación de consultas para mejorar el acceso a los datos.

Los documentos admiten sub-colecciones dentro de ellos y estructuras jerárquicas de datos que facilitan gestionar la escalabilidad de la aplicación.

¿Por qué se integra Cloud Firestore al proyecto?

La idea principal de la aplicación es proporcionar a los usuarios una plataforma que permita la consulta y compra de billetes para un viaje determinado, pero que además permita la compra de abonos, consulta de estaciones, consulta de noticias, participación y consulta en el foro de la comunidad y la consulta y gestión de información referente al usuario (información personal, mis reservas, mis abonos).

Todo esto requiere que los datos se encuentren disponibles 24/7 y sean lo más persistentes posibles para que el usuario final sea capaz de acceder a los mismos en cualquier momento y en cualquier lugar. Como consecuencia de este requerimiento se descartó la utilización de una base de datos local en el propio dispositivo.

También es importante destacar que un usuario cualquiera requiere de una conexión continua a Internet para ejecutar todas las tareas que permite desempeñar la aplicación como son las consultas, inserciones, modificaciones o eliminaciones de datos.

Sumado a todo esto se tiene en cuenta que la aplicación debe presentar al usuario datos actualizados en todo momento, como por ejemplo ocurre en la compra de billetes, donde se requiere un conocimiento inmediato de la disponibilidad del viaje.

Por este motivo y después de investigar alternativas que dieran solución a esta serie de condicionantes, se recurrió a este servicio de base de datos caracterizado por los siguientes aspectos y que, como se puede observar, proporciona una solución completa a todo lo expuesto anteriormente y su integración con aplicaciones Android es completa:

Flexibilidad	El modelo de datos de Cloud Firestore admite estructuras de datos flexibles y jerárquicas. Almacena tus datos en documentos, organizados en colecciones. Los documentos pueden contener objetos anidados complejos, además de sub-colecciones.
Consultas Expresivas	En Cloud Firestore, puedes usar consultas para recuperar documentos individuales específicos o para recuperar todos los documentos de una colección que coinciden con los parámetros de la consulta. Tus consultas pueden incluir varios filtros en cadena y combinar los filtros con criterios de orden. También se indexan de forma predeterminada, por lo que el rendimiento de las consultas es proporcional al tamaño de tu conjunto de resultados, no del conjunto de datos.
Actualizaciones en tiempo real	Al igual que <i>Realtime Database</i> , Cloud Firestore usa la sincronización de datos para actualizar los datos de cualquier dispositivo conectado. Sin embargo, también está diseñado para ejecutar consultas de recuperación únicas y sencillas de manera eficiente.
Asistencia sin conexión	Cloud Firestore almacena en caché datos que usa tu app de forma activa, por lo que la app puede escribir, leer, escuchar y consultar datos, aunque el dispositivo se encuentre sin conexión. Cuando el dispositivo vuelve a estar en línea, Cloud Firestore sincroniza todos los cambios locales de vuelta a Cloud Firestore.
Diseño para ajustarse a escala	Cloud Firestore te ofrece lo mejor de la poderosa infraestructura de <i>Google Cloud Platform</i> : replicación automática de datos multirregión, garantías de coherencia sólida, operaciones atómicas por lotes y asistencia real sobre transacciones. Diseñamos Cloud Firestore para controlar las cargas de trabajo de las bases de datos más complejas de las aplicaciones más grandes del mundo.

Tabla 99 Características principales de Cloud Firestore



Anexo II: Casos de Uso



Identificador	CU-03 - Consultar noticias		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra el listado de las noticias que se encuentran registradas.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita consultar las noticias. 2. El sistema mostrará la lista con todas las noticias. 3. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario selecciona una noticia, se llevaría a cabo el caso de uso “CU-04 - Mostrar detalles de noticia”. 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. No hay ninguna noticia registrada, se notifica la incidencia al usuario.		
Postcondiciones	El usuario visualiza el listado con todas las noticias que se encuentran registradas ordenadas cronológicamente.		
Importancia	Media	Frecuencia	Alta

Tabla 100 Caso de uso "Consultar Noticias"

Identificador	CU-04 - Mostrar detalles de noticia		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra la información detallada de una noticia.		

Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	1. El usuario selecciona una de las noticias. 2. El sistema mostrará la información detallada de la noticia.		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El usuario visualiza la información detallada de la noticia seleccionada.		
Importancia	Media	Frecuencia	Alta

Tabla 101 Caso de uso "Mostrar detalles de noticia"

Identificador	CU-05 - Consultar promociones		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra el listado de las promociones que se encuentran registradas.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	1. El usuario solicita consultar las promociones. 2. El sistema mostrará la lista con todas las promociones. 3. El usuario tiene la posibilidad de realizar las siguientes acciones: 1. Si el usuario selecciona una promoción, se llevaría a cabo el caso de uso " CU-06 - Mostrar detalles de promoción ".		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. No hay ninguna promoción registrada, se notifica la incidencia al usuario.		
Postcondiciones	El usuario visualiza el listado con todas las promociones que se encuentran registradas ordenadas cronológicamente.		

Importancia	Media	Frecuencia	Alta
--------------------	-------	-------------------	------

Tabla 102 Caso de uso "Consultar promociones"

Identificador	CU-06 - Mostrar detalles de promoción		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra la información detallada de una promoción.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	1. El usuario selecciona una de las promociones. 2. El sistema mostrará la información detallada de la promoción.		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El usuario visualiza la información detallada de la promoción seleccionada.		
Importancia	Media	Frecuencia	Alta

Tabla 103 Caso de uso "Mostrar detalles de promoción"

Identificador	CU-08 - Buscar estación		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El usuario utiliza un filtro para facilitar la búsqueda de la estación deseada.		

Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 2. La información de la estación debe de encontrarse en estado "VISIBLE" para los usuarios. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario introduce valores en el campo de búsqueda. 2. El sistema aplica el filtro sobre el listado de estaciones disponibles. 3. El sistema mostrará la estación que concuerde con los criterios de búsqueda introducidos. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El usuario visualiza la estación que cumple con los criterios de búsqueda utilizados en el filtro.		
Importancia	Media	Frecuencia	Media

Tabla 104 Caso de uso "Buscar estación"

Identificador	CU-09 - Mostrar detalles de estación		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra la información detallada de una estación.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 2. La información de la estación debe de encontrarse en estado "VISIBLE" para los usuarios. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario selecciona una de las estaciones. 2. El sistema mostrará la información detallada de la estación. 		
Secuencia Alternativa			

Excepciones en la secuencia			
Postcondiciones	El usuario visualiza la información detallada de la estación seleccionada.		
Importancia	Media	Frecuencia	Media

Tabla 105 Caso de uso "Mostrar detalles de estación"

Identificador	CU-10 - Acceso a redes sociales		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	Red Social (Twitter)
Obj. Asociados		Req. Asociados	
Descripción	El usuario puede acceder a los perfiles de la compañía en las distintas redes sociales que se encuentran disponibles (Twitter, Facebook).		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario selecciona una de las posibles redes sociales disponibles (Twitter, Facebook). 2. El sistema mostrará el perfil social de la compañía a los usuarios a través de la red social escogida. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El usuario será redireccionado a la red social seleccionada, mostrando el contenido publicado por la compañía en dicho espacio.		
Importancia	Media	Frecuencia	Alta

Tabla 106 Caso de uso "Acceso a redes sociales"

Identificador	CU-11 - Consultar foro de sugerencias		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra el listado de todos los temas de discusión que se encuentran registrados.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita consultar el foro de sugerencias. 2. El sistema mostrará la lista con todos los temas de discusión disponibles. 3. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere añadir un nuevo tema de discusión, se llevaría a cabo el caso de uso “CU-35 - Añadir tema de discusión”. 2. Si el usuario selecciona un tema de discusión, se llevaría a cabo el caso de uso “CU-12 - Mostrar detalles del tema de discusión”. 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. No hay ningún tema de discusión registrado, se notifica la incidencia al usuario.		
Postcondiciones	El usuario visualiza el listado con todos los temas de discusión que se encuentran registrados y disponibles para consulta.		
Importancia	Media	Frecuencia	Media

Tabla 107 Caso de uso "Consultar foro de sugerencias"

Identificador	CU-12 - Mostrar detalles del tema de discusión		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	

Descripción	El sistema muestra la información detallada de un tema de discusión.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	1. El usuario selecciona uno de los temas de discusión. 2. El sistema mostrará la información detallada del tema de discusión. 3. El sistema mostrará la información detallada de las respuestas propuestas para ese tema. 4. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere añadir una respuesta, se llevaría a cabo el caso de uso “CU-36 - Añadir Respuesta”. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El usuario visualiza la información detallada del tema de discusión seleccionado.		
Importancia	Media	Frecuencia	Media

Tabla 108 Caso de uso "Mostrar detalles de tema de discusión"

Identificador	CU-15 - Modificar plaza		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Cliente	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El usuario puede modificar la plaza asignada para su viaje.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 2. El usuario ha seleccionado los viajes (Sólo Ida/Ida y Vuelta) y completado la “Información de los Pasajeros”.		
Secuencia Normal	1. El usuario selecciona la opción de modificar los asientos 2. El sistema mostrará los asientos disponibles y ocupados del viaje.		

	<ol style="list-style-type: none"> 3. El usuario selecciona una plaza y confirma la modificación. 4. El sistema comprueba la modificación. 5. El sistema actualiza el asiento escogido en la reserva y notifica al usuario la actualización. 6. Se vuelve al mismo punto del proceso de compra. 		
Secuencia Alternativa			
Excepciones en la secuencia	<p>2.1 Todos los asientos ocupados, no hay posibilidad de modificar la plaza.</p> <p>4.1 Nueva plaza ocupada mientras se seleccionaba, se notifica al usuario y se vuelve al proceso de compra.</p> <p>5.1. Actualización errónea, se notifica al usuario y se vuelve al proceso de compra.</p>		
Postcondiciones	Plazas asignadas a las reservas actualizadas.		
Importancia	Media	Frecuencia	Media
Comentarios	La modificación de plazas depende del tipo de viaje (Ida/Ida y Vuelta) y el número de pasajeros para cada viaje.		

Tabla 109 Caso de uso "Modificar plaza"

Identificador	CU-16 - Realizar Pago		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	Plataforma Pago
Obj. Asociados		Req. Asociados	
Descripción	El sistema de pagos procesa la transacción para mandar una respuesta al proceso de compra.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 2. Se debe haber confirmado la compra por parte del usuario. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema envía la cuantía del pago al sistema de pagos (Paypal). 2. El sistema de pago muestra al usuario un formulario para confirmar el pago. 3. El usuario completa el formulario y envía la información al sistema de pago. 		

	4. El sistema de pago procesa la transacción y envía una respuesta al proceso de compra de la plataforma.		
Secuencia Alternativa			
Excepciones en la secuencia	4.1 Error al procesar la transacción, se notifica al usuario y se paraliza el proceso de compra. 4.2 Campos de formulario erróneos, se notifica al usuario y se paraliza el proceso de compra.		
Postcondiciones	El usuario ha realizado el pago con éxito y se pueden finalizar los procesos de compra por parte de la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 110 Caso de uso "Realizar Pago"

Identificador	CU-17 - Descargar Billete en PDF		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al usuario descargar el billete con la reserva en su dispositivo en formato PDF.		
Precondiciones	1. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
Secuencia Normal	1. El usuario solicita descargar la reserva en PDF. 2. El sistema genera un PDF con la información de la reserva. 3. El sistema almacena el archivo PDF en la memoria interna del dispositivo.		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 Los permisos para el almacenamiento interno no están concedidos, se le solicita al usuario que los active.		
Postcondiciones	El billete en formato PDF queda almacenado en la memoria interna del dispositivo.		
Importancia	Alta	Frecuencia	Alta

Tabla 111 Caso de uso "Descargar billete en PDF"

Identificador	CU-18 - Enviar billete por correo		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al usuario enviar la información de su reserva a través del correo electrónico.		
Precondiciones	El usuario debe de contar con una conexión de Internet para conectarse al servidor. El usuario requiere de una cuenta de correo Gmail donde enviar su billete.		
Secuencia Normal	El usuario solicita enviar el billete por correo electrónico. El sistema genera un correo electrónico con la información de la reserva y se envía al usuario. El sistema notifica al usuario que el envío se ha realizado correctamente.		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 Fallo en el envío, se notifica al usuario.		
Postcondiciones	El correo electrónico con la información de la reserva se ha enviado con éxito.		
Importancia	Alta	Frecuencia	Alta

Tabla 112 Caso de uso "Enviar billete por correo"

Identificador	CU-19 - Mostrar manual de usuario		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Anónimo, Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	

Descripción	El sistema permite al usuario acceder al panel de ayuda de la aplicación.		
Precondiciones			
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita mostrar el panel de ayuda. 2. El sistema muestra la información de ayuda. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	La información de ayuda es mostrada al usuario que la solicita.		
Importancia	Alta	Frecuencia	Alta

Tabla 113 Caso de uso "Mostrar manual de usuario"

Identificador	CU-20 - Cerrar sesión		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El usuario autenticado podrá cerrar sesión en el sistema.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario autenticado solicita cerrar sesión en el sistema. 2. El sistema solicita al usuario la confirmación de orden. 3. El usuario confirma la petición. 4. El sistema cierra la sesión. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	La sesión se cerrará, el usuario deja de estar autenticado en el sistema.		
Importancia	Alta	Frecuencia	Alta

Comentarios

Tabla 114 Caso de uso "Cerrar sesión"

Identificador	CU-21 - Consultar perfil de usuario		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al usuario visualizar su información personal.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita consultar su información personal. 2. El sistema muestra los datos referentes al perfil del usuario. 3. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere modificar su información personal, se llevaría a cabo el caso de uso "CU-22 - Modificar perfil de usuario". 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	La información personal del usuario se muestra por pantalla.		
Importancia	Alta	Frecuencia	Media

Tabla 115 Caso de uso "Consultar perfil de usuario"

Identificador	CU-22 - Modificar perfil de usuario		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	

Descripción	El sistema permite al usuario modificar y actualizar su información personal.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita modificar la información relativa a su perfil de usuario. 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El usuario modifica su información y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 5. El sistema actualiza la información personal del usuario. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
Postcondiciones	La información personal del usuario queda modificada/actualizada.		
Importancia	Alta	Frecuencia	Media

Tabla 116 Caso de uso "Modificar perfil de usuario"

Identificador	CU-23 - Darse de baja		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al usuario eliminar su cuenta de usuario del sistema.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita darse de baja en la aplicación. 2. El sistema solicita al usuario la confirmación de esta orden. 3. El usuario confirma la orden de eliminación. 4. El sistema elimina los datos asociados al usuario. 		

Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	La cuenta referente al usuario queda dada de baja en el sistema.		
Importancia	Alta	Frecuencia	Media

Tabla 117 Caso de uso "Darse de baja"

Identificador	CU-24 - Consultar tarjeta de crédito asociada		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra la información relativa a su tarjeta de crédito asociada		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita consultar su información personal. 2. El sistema muestra los datos referentes a su tarjeta de crédito. 3. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario no tiene una tarjeta asociada, se podría seleccionar el caso de uso "CU-25 - Añadir tarjeta de crédito". 2. Si el usuario tiene una tarjeta asociada, se podría seleccionar el caso de uso "CU-26 - Eliminar tarjeta de crédito". 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	Se muestra por pantalla la información relativa a la tarjeta de crédito vinculada al usuario.		
Importancia	Alta	Frecuencia	Media

Tabla 118 Caso de uso "Consultar tarjeta de crédito asociada"

Identificador	CU-25 - Añadir tarjeta de crédito		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al usuario vincular a su cuenta una nueva tarjeta de crédito.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 3. El usuario no puede tener vinculada una tarjeta de crédito. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita vincular una nueva tarjeta de crédito. 2. El sistema muestra al usuario los campos que debe completar. 3. El usuario completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra la nueva tarjeta de crédito y notifica al usuario el resultado de la vinculación. 		
Secuencia Alternativa			
Excepciones en la secuencia	<p>3.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.</p> <p>5.1 Inserción errónea, se notifica la incidencia al usuario y se devuelve al paso 2.</p>		
Postcondiciones	La nueva tarjeta de crédito queda vinculada a la cuenta del usuario.		
Importancia	Alta	Frecuencia	Media

Tabla 119 Caso de uso "Añadir tarjeta de crédito"

Identificador	CU-26 - Eliminar tarjeta de crédito		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	

Descripción	El sistema permitirá al usuario eliminar la tarjeta de crédito que tenga vinculada.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 3. El usuario debe de tener asignada una tarjeta de crédito. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar la tarjeta de crédito que tiene vinculada. 2. El sistema solicita al usuario la confirmación de esta orden. 3. El usuario confirma la orden de eliminación. 4. El sistema elimina los datos de la tarjeta asociada al usuario. 		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	El usuario deja de tener tarjeta de crédito asociada a su cuenta.		
Importancia	Alta	Frecuencia	Media

Tabla 120 Caso de uso "Eliminar tarjeta de crédito"

Identificador	CU-27 - Consultar Billetes		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al usuario mostrar un listado con los billetes asociados a su cuenta.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita consultar sus billetes. 2. El sistema muestra al usuario la lista con los billetes disponibles. 3. El usuario tiene la posibilidad de realizar las siguientes acciones: 		

	<ol style="list-style-type: none"> 1. Si el usuario quiere consultar los detalles de un billete, se llevará a cabo el caso de uso “CU-28 - Mostrar detalles de Billete”. 2. Si el usuario quiere anular un billete, se llevará a cabo el caso de uso “CU-29 - Anular billete”. 3. Si el usuario quiere descargar el billete en PDF, se llevará a cabo el caso de uso “CU-17 -Descargar Billete en PDF”. 4. Si el usuario quiere buscar una reserva en concreto, se llevaría a cabo el caso de uso “CU-30 - Filtrar reservas”. 		
Secuencia Alternativa	<p>A. El usuario lleva a cabo el caso de uso “CU-32 - Mostrar detalles de Abono”.</p> <p>B. El sistema automáticamente muestra los billetes registrados por el usuario que a su vez se encuentran asociados al abono seleccionado.</p> <p>C. El usuario tiene la posibilidad de realizar las siguientes acciones:</p> <ol style="list-style-type: none"> a. Si el usuario quiere consultar los detalles de un billete, se llevará a cabo el caso de uso “CU-28 -Mostrar detalles de Billete”. b. Si el usuario quiere anular un billete, se llevará a cabo el caso de uso “CU-29 - Anular billete”. c. Si el usuario quiere descargar el billete en PDF, se llevará a cabo el caso de uso “CU-17 -Descargar Billete en PDF”. 		
Excepciones en la secuencia	2.1 y B.1 El usuario no posee billetes registrados, se notifica la incidencia al usuario.		
Postcondiciones	El sistema muestra el listado con los billetes registrados por el usuario.		
Importancia	Alta	Frecuencia	Alta

Tabla 121 Caso de uso "Consultar billetes"

Identificador	CU-28 - Mostrar detalles de billete		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra al usuario la información relativa del billete seleccionado.		
Precondiciones	1. El usuario debe encontrarse autenticado en el sistema.		

Secuencia Normal	2. El usuario debe de contar con una conexión de Internet para conectarse al servidor.		
	3. El usuario debe seleccionar uno de sus billetes.		
Secuencia Alternativa	1. El usuario solicita consultar la información detallada del billete seleccionado.		
	2. El sistema muestra los datos referentes al billete.		
Excepciones en la secuencia			
Postcondiciones	Se muestra por pantalla la información relativa del billete seleccionado.		
Importancia	Alta	Frecuencia	Alta

Tabla 122 Caso de uso "Mostrar detalles de billete"

Identificador	CU-29 - Anular billete		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado, Administrador		Actores Implicados
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al usuario anular una reserva.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 3. La fecha actual es anterior a la fecha de la reserva/billete. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita anular la reserva seleccionada. 2. El sistema solicita al usuario la confirmación de esta orden. 3. El usuario confirma la orden de eliminación. 4. El sistema comprueba que: <ol style="list-style-type: none"> 1. Si la reserva pertenece a un abono, la fecha actual debe de ser anterior a la fecha de caducidad del abono. 5. El sistema elimina la reserva de la plataforma. <ol style="list-style-type: none"> 1. Si la reserva pertenece a un abono, este se actualiza para disponer de nuevo de una reserva más. 		

Secuencia Alternativa	6. El sistema notifica a través del correo electrónico la anulación de la reserva.		
	3.1 El usuario deniega la petición, no se produce la eliminación.		
	4.1 La reserva no cumple con las condiciones de fecha, se notifica al usuario y no se produce la eliminación.		
Postcondiciones	<ol style="list-style-type: none"> 1. La reserva queda eliminada de la plataforma. 2. Si la reserva estaba asociada a un abono, este recupera un viaje disponible más para canjear. 		
Importancia	Alta	Frecuencia	Alta

Tabla 123 Caso de uso "Anular billete"

Identificador	CU-30 - Filtrar reservas		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado, Administrador	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El usuario utiliza un filtro para facilitar la búsqueda de la reserva deseada.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario introduce valores en el campo de búsqueda. 2. El sistema aplica el filtro sobre el listado de reserva. 3. El sistema mostrará las reservas que concuerden con los criterios de búsqueda introducidos. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El usuario visualiza las reservas que cumplen con los criterios de búsqueda utilizados en el filtro.		
Importancia	Media	Frecuencia	Media

Tabla 124 Caso de uso "Filtrar Reservas"

Identificador	CU-31 - Consultar abonos		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al usuario mostrar un listado con los abonos asociados a su cuenta.		
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita consultar sus abonos. 2. El sistema muestra al usuario la lista con los abonos disponibles. 3. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere consultar los detalles de un abono, se llevará a cabo el caso de uso “CU-32 - Mostrar detalles de Abono”. 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1 El usuario no posee abonos registrados, se notifica la incidencia al usuario.		
Postcondiciones	El sistema muestra el listado con los abonos registrados por el usuario.		
Importancia	Alta	Frecuencia	Alta

Tabla 125 Caso de uso "Consultar abonos"

Identificador	CU-32 - Mostrar detalles de abono		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Usuario Autenticado	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema muestra al usuario la información relativa del abono seleccionado.		

Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse autenticado en el sistema. 2. El usuario debe de contar con una conexión de Internet para conectarse al servidor. 3. El usuario debe seleccionar uno de sus abonos. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita consultar la información detallada del abono seleccionado. 2. El sistema muestra los datos referentes al abono. 3. El sistema realiza automáticamente el caso de uso “CU-27 - Consultar Billetes”, mostrando al usuario el listado de billetes asociados al abono. 4. El usuario tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el usuario quiere consultar los detalles de un billete, se llevará a cabo el caso de uso “CU-28 - Mostrar detalles de Billete”. 2. Si el usuario quiere anular un billete, se llevará a cabo el caso de uso “CU-29 - Anular billete”. 3. Si el usuario quiere descargar el billete en PDF, se llevará a cabo el caso de uso “CU-17 -Descargar Billete en PDF”. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	Se muestra por pantalla la información relativa al abono seleccionado junto a sus billetes asociados.		
Importancia	Alta	Frecuencia	Alta

Tabla 126 Caso de uso "Mostrar detalles de abono"

Identificador	CU-37 - Mostrar estaciones		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todas las estaciones registradas en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		

Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de estaciones. 2. El sistema muestra al administrador la lista con todas las estaciones registradas. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el administrador quiere registrar en la plataforma una nueva estación, se llevará a cabo el caso de uso “CU-46 - Crear estación”. 2. Si el administrador quiere modificar la información de una estación, se llevará a cabo el caso de uso “CU-54 - Modificar estación”. 3. Si el usuario quiere eliminar una estación de la plataforma, se llevará a cabo el caso de uso “CU-62 - Eliminar estación”. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El sistema muestra el listado con todas las estaciones registradas en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 127 Caso de uso "Mostrar estaciones"

Identificador	CU-38 - Mostrar precios de trayectos		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todos los precios registrados en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de precios. 2. El sistema muestra al administrador la lista con todos los precios registrados. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: 		

	<ol style="list-style-type: none"> 1. Si el administrador quiere registrar en la plataforma un nuevo precio para un trayecto, se llevará a cabo el caso de uso “CU-47 - Crear precio de trayecto”. 2. Si el administrador quiere modificar el precio de un trayecto, se llevará a cabo el caso de uso “CU-55 - Modificar precio de trayecto”. 3. Si el usuario quiere eliminar el precio de un trayecto, se llevará a cabo el caso de uso “CU-63 - Eliminar precio de trayecto”. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El sistema muestra el listado con todos los precios registrados en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 128 Caso de uso "Mostrar precios de trayectos"

Identificador	CU-39 - Mostrar Viajes		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todos los viajes registrados en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de viajes. 2. El sistema muestra al administrador la lista con todos los viajes registrados. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el administrador quiere registrar en la plataforma un nuevo viaje, se llevará a cabo el caso de uso “CU-48 - Crear viaje”. 2. Si el administrador quiere modificar la información de un viaje, se llevará a cabo el caso de uso “CU-56 - Modificar viaje”. 		

Secuencia Alternativa	3. Si el usuario quiere eliminar un viaje de la plataforma, se llevará a cabo el caso de uso “ CU-64 - Eliminar viaje ”.		
	Excepciones en la secuencia		
	Postcondiciones		
	El sistema muestra el listado con todos los viajes registrados en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 129 Caso de uso "Mostrar viajes/trayectos"

Identificador	CU-40 - Mostrar tipos de abono		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todos los tipos de abono registrados en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de tipos de abono. 2. El sistema muestra al administrador la lista con todos los tipos de abono registrados. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el administrador quiere registrar en la plataforma un nuevo tipo de abono, se llevará a cabo el caso de uso “CU-49 - Crear tipo de abono”. 2. Si el administrador quiere modificar la información de un tipo de abono, se llevará a cabo el caso de uso “CU-57 - Modificar tipo de abono”. 3. Si el usuario quiere eliminar un tipo de abono de la plataforma, se llevará a cabo el caso de uso “CU-65 - Eliminar tipo de abono”. 		
Secuencia Alternativa			

Excepciones en la secuencia			
Postcondiciones	El sistema muestra el listado con todos los tipos de abono registrados en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 130 Caso de uso "Mostrar tipos de abono"

Identificador	CU-41 - Mostrar noticias		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todas las noticias registradas en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de noticias. 2. El sistema muestra al administrador la lista con todas las noticias registradas. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el administrador quiere registrar en la plataforma una nueva noticia, se llevará a cabo el caso de uso "CU-50 - Crear noticia". 2. Si el administrador quiere modificar la información de una noticia, se llevará a cabo el caso de uso "CU-58 - Modificar noticia". 3. Si el usuario quiere eliminar una noticia de la plataforma, se llevará a cabo el caso de uso "CU-66 - Eliminar noticia". 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El sistema muestra el listado con todas las noticias registradas en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 131 Caso de uso "Mostrar noticias"

Identificador	CU-42 - Mostrar promociones		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todas las promociones registradas en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de promociones. 2. El sistema muestra al administrador la lista con todas las promociones registradas. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el administrador quiere registrar en la plataforma una nueva promoción, se llevará a cabo el caso de uso “CU-51 - Crear promoción”. 2. Si el administrador quiere modificar la información de una promoción, se llevará a cabo el caso de uso “CU-59 - Modificar promoción”. 3. Si el usuario quiere eliminar una promoción de la plataforma, se llevará a cabo el caso de uso “CU-67 - Eliminar promoción”. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El sistema muestra el listado con todas las promociones registradas en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 132 Caso de uso "Mostrar promociones"

Identificador	CU-43 - Mostrar reservas		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore

Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todas las reservas registradas en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de reservas. 2. El sistema muestra al administrador la lista con todas las reservas registradas. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el administrador quiere eliminar una reserva de la plataforma, se llevará a cabo el caso de uso “CU-29 - Anular reserva”. 2. Si el administrador quiere buscar una reserva en concreto, se llevaría a cabo el caso de uso “CU-30 - Filtrar reservas”. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El sistema muestra el listado con todas las reservas registradas en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 133 Caso de uso "Mostrar reservas"

Identificador	CU-44 - Mostrar horarios		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todos los horarios registrados en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		

Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de horarios. 2. El sistema muestra al administrador la lista con todos los horarios registrados. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: <ol style="list-style-type: none"> 1. Si el administrador quiere registrar en la plataforma un nuevo horario, se llevará a cabo el caso de uso “CU-52 – Crear horario”. 2. Si el administrador quiere modificar la información de un horario, se llevará a cabo el caso de uso “CU-60 – Modificar horario”. 3. Si el usuario quiere eliminar un horario de la plataforma, se llevará a cabo el caso de uso “CU-68 – Eliminar horario”. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El sistema muestra el listado con todos los horarios registrados en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 134 Caso de uso "Mostrar horarios"

Identificador	CU-45 - Mostrar vehículos		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador mostrar un listado con todos los vehículos registrados en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita el listado de horarios. 2. El sistema muestra al administrador la lista con todos los horarios registrados. 3. El administrador tiene la posibilidad de realizar las siguientes acciones: 		

	<ol style="list-style-type: none"> 1. Si el administrador quiere registrar en la plataforma un nuevo vehículo, se llevará a cabo el caso de uso “CU-53 – Crear vehículo”. 2. Si el administrador quiere modificar la información de un vehículo, se llevará a cabo el caso de uso “CU-61 – Modificar vehículo”. 3. Si el usuario quiere eliminar un vehículo de la plataforma, se llevará a cabo el caso de uso “CU-69 – Eliminar vehículo”. 		
Secuencia Alternativa			
Excepciones en la secuencia			
Postcondiciones	El sistema muestra el listado con todos los vehículos registrados en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 135 Caso de uso "Mostrar vehículos"

Identificador	CU-46 - Crear estación		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El administrador, a través de un formulario, registra una nueva estación en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita añadir una nueva estación. 2. El sistema muestra al administrador los campos que debe completar. 3. El administrador completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra la nueva estación y notifica al usuario el resultado de la inserción. 		
Secuencia Alternativa			

Excepciones en la secuencia	4.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.		
Postcondiciones	El sistema registra una nueva estación en la plataforma.		
Importancia	Alta	Frecuencia	Media

Tabla 136 Caso de uso "Crear estación"

Identificador	CU-47 - Crear precio de trayecto		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El administrador, a través de un formulario, registra un nuevo precio en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita añadir un nuevo precio. 2. El sistema muestra al administrador los campos que debe completar. 3. El administrador completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra el nuevo precio para el trayecto y notifica al usuario el resultado de la inserción. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.		
Postcondiciones	El sistema registra un nuevo precio en la plataforma.		
Importancia	Alta	Frecuencia	Media

Tabla 137 Caso de uso "Crear precio de trayecto"

Identificador	CU-48 - Crear viaje		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El administrador, a través de un formulario, registra un nuevo viaje en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita añadir un nuevo viaje. 2. El sistema muestra al administrador los campos que debe completar. 3. El administrador completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra el nuevo viaje y notifica al usuario el resultado de la inserción. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.		
Postcondiciones	El sistema registra un nuevo viaje en la plataforma.		
Importancia	Alta	Frecuencia	Media

Tabla 138 Caso de uso "Crear viaje/trayecto"

Identificador	CU-49 - Crear tipo de abono		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El administrador, a través de un formulario, registra un nuevo tipo de abono en la plataforma.		

Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita añadir un nuevo tipo de abono. 2. El sistema muestra al administrador los campos que debe completar. 3. El administrador completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra el nuevo tipo de abono y notifica al usuario el resultado de la inserción. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.		
Postcondiciones	El sistema registra un nuevo tipo de abono en la plataforma.		
Importancia	Alta	Frecuencia	Media

Tabla 139 Caso de uso "Crear tipo de abono"

Identificador	CU-50 - Crear noticia		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El administrador, a través de un formulario, registra una nueva noticia en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita añadir una nueva noticia. 2. El sistema muestra al administrador los campos que debe completar. 3. El administrador completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 		

	5. El sistema registra la nueva noticia y notifica al usuario el resultado de la inserción.		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.		
Postcondiciones	El sistema registra una nueva noticia en la plataforma.		
Importancia	Alta	Frecuencia	Alta

Tabla 140 Caso de uso "Crear noticia"

Identificador	CU-51 - Crear promoción		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El administrador, a través de un formulario, registra una nueva promoción en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita añadir una nueva promoción. 2. El sistema muestra al administrador los campos que debe completar. 3. El administrador completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra la nueva promoción y notifica al usuario el resultado de la inserción. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.		
Postcondiciones	El sistema registra una nueva promoción en la plataforma.		
Importancia	Alta	Frecuencia	Media

Tabla 141 Caso de uso "Crear promoción"

Identificador	CU-52 - Crear horario		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El administrador, a través de un formulario, registra un nuevo horario en la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita añadir un nuevo horario. 2. El sistema muestra al administrador los campos que debe completar. 3. El administrador completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra el nuevo horario y notifica al usuario el resultado de la inserción. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.		
Postcondiciones	El sistema registra un nuevo horario en la plataforma.		
Importancia	Alta	Frecuencia	Media

Tabla 142 Caso de uso "Crear horario"

Identificador	CU-53 - Crear vehículo		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El administrador, a través de un formulario, registra un nuevo vehículo en la plataforma.		

Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita añadir un nuevo vehículo. 2. El sistema muestra al administrador los campos que debe completar. 3. El administrador completa los campos y envía la información al sistema. 4. El sistema comprueba la información. 5. El sistema registra el nuevo vehículo y notifica al usuario el resultado de la inserción. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Información de campo errónea, se notifica la incidencia al usuario y se devuelve al paso 2.		
Postcondiciones	El sistema registra un nuevo vehículo en la plataforma.		
Importancia	Alta	Frecuencia	Media

Tabla 143 Caso de uso "Crear vehículo"

Identificador	CU-54 - Modificar estación		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al administrador modificar y actualizar la estación seleccionada		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita modificar la información de la estación seleccionada. 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El administrador modifica los campos y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 		

Secuencia Alternativa	5. El sistema actualiza la información relativa a la estación seleccionada.		
	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
	La información de la estación seleccionada queda modificada/actualizada.		
Importancia	Media	Frecuencia	Baja

Tabla 144 Caso de uso "Modificar estación"

Identificador	CU-55 - Modificar precio de trayecto		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al administrador modificar y actualizar el precio seleccionado		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita modificar el precio seleccionado. 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El administrador modifica los campos y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 5. El sistema actualiza la información relativa al precio del trayecto seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
Postcondiciones	El precio del trayecto seleccionado queda modificado/actualizado.		
Importancia	Media	Frecuencia	Baja

Tabla 145 Caso de uso "Modificar precio de trayecto"

Identificador	CU-56 - Modificar viaje		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al administrador modificar y actualizar el viaje seleccionado.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita modificar la información del viaje seleccionado. 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El administrador modifica los campos y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 5. El sistema actualiza la información relativa al viaje seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
Postcondiciones	La información del viaje seleccionado queda modificada/actualizada.		
Importancia	Media	Frecuencia	Baja

Tabla 146 Caso de uso "Modificar viaje/trayecto"

Identificador	CU-57 - Modificar tipo de abono		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	

Descripción	El sistema permite al administrador modificar y actualizar el tipo de abono seleccionado.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita modificar la información del tipo de abono seleccionado. 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El administrador modifica los campos y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 5. El sistema actualiza la información relativa al tipo de abono seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
Postcondiciones	La información del tipo de abono seleccionado queda modificada/actualizada.		
Importancia	Media	Frecuencia	Baja

Tabla 147 Caso de uso "Modificar tipo de abono"

Identificador	CU-58 - Modificar noticia		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al administrador modificar y actualizar la noticia seleccionada.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita modificar la información de la noticia seleccionada. 		

	<ol style="list-style-type: none"> 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El administrador modifica los campos y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 5. El sistema actualiza la información relativa a la noticia seleccionada. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
Postcondiciones	La información de la noticia seleccionada queda modificada/actualizada.		
Importancia	Media	Frecuencia	Baja

Tabla 148 Caso de uso "Modificar noticia"

Identificador	CU-59 - Modificar promoción		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al administrador modificar y actualizar la promoción seleccionada		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita modificar la información de la promoción seleccionada. 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El administrador modifica los campos y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 5. El sistema actualiza la información relativa a la promoción seleccionada. 		
Secuencia Alternativa			

Excepciones en la secuencia	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
Postcondiciones	La información de la promoción seleccionada queda modificada/actualizada.		
Importancia	Media	Frecuencia	Baja

Tabla 149 Caso de uso "Modificar promoción"

Identificador	CU-60 - Modificar horario		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al administrador modificar y actualizar el horario seleccionado.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita modificar la información del horario seleccionado. 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El administrador modifica los campos y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 5. El sistema actualiza la información relativa al horario seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
Postcondiciones	La información del horario seleccionado queda modificada/actualizada.		
Importancia	Media	Frecuencia	Baja

Tabla 150 Caso de uso "Modificar horario"

Identificador	CU-61 - Modificar vehículo		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permite al administrador modificar y actualizar el vehículo seleccionado.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita modificar la información del vehículo seleccionado. 2. El sistema muestra al usuario los campos que pueden estar sometidos a modificación. 3. El administrador modifica los campos y confirma el envío con las modificaciones. 4. El sistema comprueba la información. 5. El sistema actualiza la información relativa al vehículo seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1 Campos de formulario erróneos, se notifica al usuario y se devuelve al paso 2.		
Postcondiciones	La información del vehículo seleccionado queda modificada/actualizada.		
Importancia	Media	Frecuencia	Baja

Tabla 151 Caso de uso "Modificar vehículo"

Identificador	CU-62 - Eliminar estación		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador eliminar la estación seleccionada de la plataforma.		

Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar la estación seleccionada. 2. El sistema solicita al administrador la confirmación de esta orden. 3. El administrador confirma la orden de eliminación. 4. El sistema elimina la estación seleccionada. 		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	La estación seleccionada es eliminada de la plataforma.		
Importancia	Media	Frecuencia	Baja

Tabla 152 Caso de uso "Eliminar estación"

Identificador	CU-63 - Eliminar precio de trayecto		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador eliminar el precio seleccionado de la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar el precio de trayecto seleccionado. 2. El sistema solicita al administrador la confirmación de esta orden. 3. El administrador confirma la orden de eliminación. 4. El sistema elimina el precio seleccionado. 		
Secuencia Alternativa			

Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	El precio de trayecto seleccionado es eliminado de la plataforma.		
Importancia	Media	Frecuencia	Baja

Tabla 153 Caso de uso "Eliminar precio de trayecto"

Identificador	CU-64 - Eliminar viaje		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador eliminar el viaje seleccionado de la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar el viaje seleccionado. 2. El sistema solicita al administrador la confirmación de esta orden. 3. El administrador confirma la orden de eliminación. 4. El sistema elimina el viaje seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	El viaje seleccionado es eliminado de la plataforma.		
Importancia	Media	Frecuencia	Baja

Tabla 154 Caso de uso "Eliminar viaje/trayecto"

Identificador	CU-65 - Eliminar tipo de abono		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore

Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador eliminar el tipo de abono seleccionado de la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar el tipo de abono seleccionado. 2. El sistema solicita al administrador la confirmación de esta orden. 3. El administrador confirma la orden de eliminación. 4. El sistema elimina el tipo de abono seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	El tipo de abono seleccionado es eliminado de la plataforma.		
Importancia	Media	Frecuencia	Baja

Tabla 155 Caso de uso "Eliminar tipo de abono"

Identificador	CU-66 - Eliminar noticia		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador eliminar la noticia seleccionada de la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar la noticia seleccionada. 2. El sistema solicita al administrador la confirmación de esta orden. 3. El administrador confirma la orden de eliminación. 4. El sistema elimina la noticia seleccionada. 		

Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	La noticia seleccionada es eliminada de la plataforma.		
Importancia	Media	Frecuencia	Baja

Tabla 156 Caso de uso "Eliminar noticia"

Identificador	CU-67 - Eliminar promoción		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador eliminar la promoción seleccionada de la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar la promoción seleccionada. 2. El sistema solicita al administrador la confirmación de esta orden. 3. El administrador confirma la orden de eliminación. 4. El sistema elimina la promoción seleccionada. 		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	La promoción seleccionada es eliminada de la plataforma.		
Importancia	Media	Frecuencia	Baja

Tabla 157 Caso de uso "Eliminar promoción"

Identificador	CU-68 - Eliminar horario		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador eliminar el horario seleccionado de la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar el horario seleccionado. 2. El sistema solicita al administrador la confirmación de esta orden. 3. El administrador confirma la orden de eliminación. 4. El sistema elimina el horario seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	El horario seleccionado es eliminado de la plataforma.		
Importancia	Media	Frecuencia	Baja

Tabla 158 Caso de uso "Eliminar horario"

Identificador	CU-69 - Eliminar vehículo		
Versión Actual	1.0	Autor	Pablo Jambrina Sanz
Actor Principal	Administrador	Actores Implicados	Cloud Firestore
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador eliminar el vehículo seleccionado de la plataforma.		
Precondiciones	<ol style="list-style-type: none"> 1. El administrador debe encontrarse autenticado en el sistema. 2. El administrador debe de contar con una conexión de Internet para conectarse al servidor. 		

Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar el vehículo seleccionado. 2. El sistema solicita al administrador la confirmación de esta orden. 3. El administrador confirma la orden de eliminación. 4. El sistema elimina el vehículo seleccionado. 		
Secuencia Alternativa			
Excepciones en la secuencia	3.1 El usuario deniega la petición, no se produce la eliminación.		
Postcondiciones	El vehículo seleccionado es eliminado de la plataforma.		
Importancia	Media	Frecuencia	Baja

Tabla 159 Caso de uso "Eliminar vehículo"