



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

**FeSPUGT: diseño e implementación de
una aplicación móvil en Android.**

Autora:

Nuria Cancho Diaz



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

Autora:

Nuria Cancho Diaz

Tutor:

Joaquín Adiego Rodríguez

Agradecimientos:

A mis padres por darme la oportunidad de tener un futuro prometedor, y dejarme elegir quién quiero ser.

A mi pareja, por apoyarme en los momentos difíciles y acompañarme en mis locuras.

A mis compañeros y amigos, por los momentos a su lado.

Nuria Cancho Diaz

Contenido

Capítulo 1: Introducción y objetivos.	8
1.1. Motivación del TFG.	8
1.2. Objetivos y alcance.	9
Capítulo 2: Marco teórico.	9
2.1. RSS.	9
2.1.1. ¿Qué es RSS?	9
2.1.2. Para que sirve el RSS.	10
2.1.3. Ventajas de RSS.	10
2.1.4. Desventajas de RSS.	11
2.2. Atom.	12
2.3. XMLPullParser.	13
Capítulo 3: Android.	14
3.1. Introducción.	14
3.2. Breve historia.	15
3.3. ¿Por qué Android?: 8.0 Oreo	16
3.4. Diseño y desarrollo.	17
Capítulo 4: Planificación y Desarrollo del proyecto.	18
4.1. Riesgos y plan de contingencia.	18
4.1.1. Riesgos.	18
4.1.2. Planes de contingencia.	19
4.2. Planificación del proyecto.	20
4.2.1. Fases del proyecto.	20
4.2.2. Presupuestos.	22
Capítulo 5: Análisis.	24
5.1. Actores.	24
5.2. Requisitos.	24
5.2.1. Requisitos funcionales.	24
5.2.2. Requisitos no funcionales.	26
5.2.3. Requisitos de información.	27
5.3. Casos de uso.	28
5.4. Modelo de dominio.	34
5.5. Modelos dinámicos.	35
5.5.1. CU Observar noticias por categoría.	35

5.5.2.	CU Leer noticia elegida.....	36
5.5.3.	CU Activar/desactivar notificaciones.....	36
5.5.4.	CU Búsqueda de una noticia.....	37
5.5.5.	CU Seguimiento en redes sociales.....	37
5.5.6.	CU Consultar Contacto.....	38
5.5.7.	CU Consultar Quienes Somos.....	38
5.5.8.	CU Consultar Aviso Legal.....	39
5.5.9.	CU Consultar hoja de afiliación.....	39
5.6.	Modelo entidad – relación.....	40
Capítulo 6: Diseño.....		40
6.1.	Descripción de la arquitectura.....	40
6.2.1.	MVC.....	40
6.2.2.	Arquitectura global del sistema.....	42
6.2.	Modelo relacional.....	43
6.3.	Patrones usados en el diseño.....	43
6.3.1.	Patrón singleton.....	43
6.3.2.	Iterator.....	44
Capítulo 7: Implementación.....		46
7.1.	Herramientas utilizadas.....	46
7.2.	Entorno de desarrollo.....	46
7.3.	Pruebas.....	47
7.3.1.	Pruebas de caja blanca.....	47
7.3.2.	Pruebas de caja negra.....	47
Capítulo 8: Conclusiones.....		49
8.1.	Objetivos alcanzados.....	49
8.2.	Conocimientos adquiridos.....	49
8.3.	Trabajo futuro.....	50
Bibliografía.....		51
Apéndice 1: Manual del usuario.....		52
Apéndice 2: Glosario.....		60

Contenido de tablas.

Tabla 1. Riesgo: Modificación de los requisitos.	18
Tabla 2. Riesgo: Retraso respecto a la planificación.	18
Tabla 3. Riesgo: Necesidad del estudio de la tecnología a usar.	19
Tabla 4. Riesgo: Fallos al probar la aplicación en distintos dispositivos.	19
Tabla 5. Plan de contingencia: Modificación de requisitos.	19
Tabla 6. Plan de contingencia: Retraso respecto a la planificación.	20
Tabla 7. Plan de contingencia: Necesidad del estudio de la tecnología a usar.	20
Tabla 8. Fases del TFG.	21
Tabla 9. Costes iniciales.	23
Tabla 10. Costes reales.	23
Tabla 11. Actor: Usuario.	24
Tabla 12. Requisito funcional: Mostrar noticias por categoría.	24
Tabla 13. Requisito funcional: Actualización de noticias por categoría.	25
Tabla 14. Requisito funcional: Gestionar notificaciones.	25
Tabla 15. Requisito funcional: Gestionar menú.	25
Tabla 16. Requisito funcional: Mostrar noticia completa.	25
Tabla 17. Requisito no funcional: Sistema operativo.	26
Tabla 18. Requisito no funcional: Conexión a internet.	26
Tabla 19. Requisito no funcional: Usabilidad.	26
Tabla 20. Requisito no funcional: Aprendizaje.	26
Tabla 21. Requisito de información: Noticia.	27
Tabla 22. Caso de uso: Observar noticias por categoría.	29
Tabla 23. Caso de uso: Leer la noticia elegida.	29
Tabla 24. Caso de uso: Activar / desactivar las notificaciones.	30
Tabla 25. Caso de uso: Búsqueda de una noticia.	31
Tabla 26. Caso de uso: Seguimiento en redes sociales.	31
Tabla 27. Caso de uso: Observar contacto.	32
Tabla 28. Caso de uso: Consultar aviso legal.	32
Tabla 29. Caso de uso: Consultar hoja de afiliación.	33
Tabla 30. Caso de uso: Consultar Quienes Somos.	33
Tabla 31. Especificaciones de Hardware.	46
Tabla 32. Especificaciones de Software.	46
Tabla 33. Batería de pruebas: Búsqueda de una noticia.	48
Tabla 34. Batería de pruebas: Seguimiento en redes sociales.	48
Tabla 35. Batería de pruebas: Activar/Desactivar notificaciones.	48

Contenido de ilustraciones.

Ilustración 1. Diagrama de Gantt.....	22
Ilustración 2. Diagrama de casos de uso.	28
Ilustración 3. Modelo de Dominio	34
Ilustración 4. Caso de uso: Observar noticias por categoría.	35
Ilustración 5. Caso de uso: Leer noticia elegida.	36
Ilustración 6. Caso de uso: activar/desactivar notificaciones.	36
Ilustración 7. Caso de uso: Búsqueda de una noticia.	37
Ilustración 8. Caso de uso: Seguimiento en redes sociales.	37
Ilustración 9. Caso de uso: Consultar Contacto.	38
Ilustración 10. Caso de uso: Consultar Quienes Somos.	38
Ilustración 11. Caso de uso: Consultar Aviso Legal.	39
Ilustración 12. Caso de uso: Consultar hoja de afiliación.	39
Ilustración 13. Diagrama Entidad - Relación.	40
Ilustración 14. Modelo - Vista - Controlador.	41
Ilustración 15. Modelo arquitectónico.	42
Ilustración 16. Tabla de la base de datos.....	43
Ilustración 17. Patrón singleton.	44
Ilustración 18. Patrón iterator.	45
Ilustración 19. Pantalla de inicio.....	52
Ilustración 20. Pantalla menú de las categorías.....	53
Ilustración 21. Pantalla con menú de las opciones extras.....	53
Ilustración 22. Pantalla de contactos de FeSP.....	54
Ilustración 23. Pantalla con el documento de afiliación.....	54
Ilustración 24. Pantallas de Quienes Somos y Aviso Legal.....	55
Ilustración 25. Pantalla del menú secundario.....	55
Ilustración 26. Pantalla de búsqueda.	56
Ilustración 27. Pantalla del buscador del cliente.	56
Ilustración 28. Pantalla de las redes sociales.....	57
Ilustración 29. Pantalla de las notificaciones.....	57
Ilustración 30. Pantalla de la categoría E. Pública.	58
Ilustración 31. Pantalla de una noticia detallada.	59

Capítulo 1: Introducción y objetivos.

Este trabajo surge de la solicitud por parte del cliente, y por parte del alumno por querer adquirir nuevos conocimientos dentro de la mención de software. En este capítulo se va a detallar la situación actual y características del trabajo, así como las metas y objetivos que se han decidido.

1.1. Motivación del TFG.

Existe un reto personal por parte del alumno por aprender y realizarse individualmente en el campo de las aplicaciones móviles en Android. En el grado en Ingeniería Informática con mención en Software se imparte de forma optativa la asignatura “Sistemas móviles”, donde se desarrolla el aprendizaje, tanto teórico y práctico, de la creación desde cero de una aplicación móvil en el lenguaje Android.

En mi caso no pude cursar dicha asignatura, y no por falta de interés, y de ahí el reto personal e interesante que plantea este proyecto. Aparte de la gran demanda que existe en el mercado laboral para desarrolladores de aplicaciones móviles y el crecimiento del uso del Smartphone durante la última década.

También cabe destacar la evolución a lo largo de los años, con sus distintas versiones, del sistema operativo en sí. Comparando la primera versión con la última, nadie creería que eso ha podido ser desarrollado por tantas personas con unos conocimientos increíbles, creando un universo paralelo a la vida real, haciéndonos la vida un poco más sencilla.

Por lo tanto, todos los factores relacionados con este TFG intentan seducir a la motivación de crear algo que ayude a interactuar entre el cliente y sus usuarios, creando una comunicación plena y con ganas de estar bien informado sobre lo que puede aportar este salto del ordenador a la pantalla de tu Smartphone o Tablet.

1.2. Objetivos y alcance.

En este trabajo se propone la creación de una aplicación móvil en Android basada en el uso de un feed RSS con la finalidad por parte del alumno mediante un trabajo personal, que se aplique e integren los conocimientos, habilidades y actitudes desarrolladas durante los años de estudio en la titulación. Para ello se dan una serie de objetivos que se deberán llevar a cabo.

Los objetivos del TFG con:

- Estudio del entorno de desarrollo para Android, así como dicho lenguaje de programación.
- Uso de las últimas novedades relacionadas en el desarrollo como los Fragments o Material Design.
- Uso de *XML Pull Parsing* para mostrar el feed RSS.
- Creación de una interfaz de usuario fácil y sencilla de utilizar para los usuarios a los que va enfocado.
- Trabajar en equipo con el tutor asumiendo los distintos roles: participar, liderar, etc.
- Elaborar la memoria correspondiente al TFG: introducción, marco teórico, planificación y desarrollo del TFG, etc.
- Elaborar y defender una presentación pública del TFG.

Capítulo 2: Marco teórico.

2.1. RSS.

2.1.1. ¿Qué es RSS?

RSS es una forma muy sencilla para que puedas recibir, directamente en tu ordenador o en una página web online (a través de un lector RSS) información actualizada sobre tus páginas web favoritas, sin necesidad de que tengas que visitarlas una a una. Esta información se actualiza automáticamente, sin que tengas que hacer nada. Para recibir las noticias RSS la página deberá tener disponible el servicio RSS y deberás tener un lector RSS.

Si existen varias páginas web que te interesan que van actualizando sus contenidos y te gustaría mantenerte informado, un lector RSS te ahorrará mucho tiempo en esta tarea. Gracias al RSS, no tendrás que visitar cada una de las páginas web que te interesan para ver si han añadido o no algún artículo que te pueda interesar. Estas páginas te informarán a ti (a través de tu

lector de RSS). Cuando ingreses a tu Lector RSS (o RSS Reader), estarás automáticamente informado sobre todas las novedades que se han producido en todas las páginas web que has dado de alta.

Los sistemas RSS tienen muchas ventajas que vamos a resumir. Gracias al RSS, tendrás reunidas en un mismo lugar y a un solo clic de distancia, toda la información actualizada de las páginas web (Fuentes o canales RSS) que más te interesan. Si todavía no te han quedado claro lo que es el RSS y lo que te puede aportar, puedes leer información adicional sobre para qué sirve el RSS y algunos ejemplos ilustrativos.

2.1.2. Para que sirve el RSS.

El RSS facilita la gestión y publicación de información y noticia webs. RSS es una forma estandarizada de distribución de la información de las páginas web a los lectores de las páginas. Esta información se distribuye a través de las fuentes RSS o Canales RSS. Gracias al RSS, los lectores pasan a tener una herramienta útil para mantenerse informado sobre las noticias y webs que le resultan de interés, conservando y almacenando toda la información en un solo lugar que se actualiza de manera automática.

Además, el RSS brinda al lector un importante ahorro en tiempo en la lectura de información y noticias, ya que sólo con abrir el lector RSS (ya sea un programa, el navegador o un lector web online), el usuario podrá ver cuáles son las últimas actualizaciones y noticias que han publicado las diferentes páginas web a las que está suscrito. Ahora solo es cuestión de relajarse y leer ya que el resto lo hará el software.

2.1.3. Ventajas de RSS.

Las ventajas que ofrecen los Sistemas RSS son muchas. Se pueden destacar las siguientes:

- Las páginas web y blogs distribuyen a través de los canales RSS las últimas actualizaciones de aquellas páginas web que son de su interés. A través de RSS podrás enterarte de las últimas noticias.
- La decisión está del lado del usuario ya que él es quien elige a qué páginas web suscribirse y cuando darse de baja de estas páginas web.

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

- El RSS supone un importante ahorro en el tiempo de navegación y búsqueda de información. En el lector RSS, el usuario tendrá un resumen de los artículos para poder decidir qué información quiere leer.
- El RSS está libre de SPAM, porque no tienes que dar tu correo electrónico. Esto no ocurre con suscripciones por correo electrónico, en las que además de recibir noticias, podrías recibir también SPAM u otra información no deseada. Cuando estás suscrito a las fuentes RSS de una página web, no recibirás otra información que la que se publique en las páginas web que son de tu interés.
- La cancelación de la suscripción a la página web será rápida y sencilla. En las suscripciones vía correo electrónico, a menudo el suscriptor tiene que especificar las razones por las que quiere darse de baja y luego debe confirmar su petición. En cambio, con el RSS sólo se debe eliminar la página web del lector de RSS. Es así de sencillo.
- Recibir las fuentes o Canales RSS de tus páginas web favoritas es totalmente gratuito. Tanto los contenidos como la mayoría de los programas (lectores RSS) que permiten leer las noticias RSS son totalmente gratuitos.

Todas estas ventajas se resumen en: los sistemas RSS te facilitan enormemente el acceso a aquella información de internet que más te interesa y te permiten estar permanentemente informado. También puedes leer un ejemplo que ilustra las ventajas que tiene contar con un sistema de RSS.

2.1.4. Desventajas de RSS.

Realmente no existen desventajas en el uso de RSS y sí muchas ventajas que ilustramos a través de un ejemplo.

De todos modos, y si bien el RSS no presentan ninguna desventaja, hay algunas cuestiones relacionadas con el RSS que determinados usuarios podrían considerarlas como tales:

- Algunos de los programas o lectores RSS no están en español (aunque los más importantes sí tienen versión en español).
- No todos los lectores RSS muestran las fotos (aunque muchos sí lo hacen).
- Recibir los canales RSS crean mayor tráfico y demanda en el servidor, por lo que puede hacer que la conexión a Internet se vuelva algo más lenta (aunque en realidad, el consumo de banda de los Canales RSS es mínimo),

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

- Dado que se trata de una nueva tecnología, es posible que algunas páginas web de interés no publiquen todavía sus noticias a través de un canal o fuente RSS.
- El vocabulario que se utiliza y la jerga, son algo confusas. Se trata de términos en inglés que a menudo no se traducen al español. Además, y dado que se trata de una tecnología nueva para la mayoría de los usuarios, los términos en español son algo confusos. En esta página, damos un ejemplo ilustrativo para explicar el significado de las palabras que más se usan en español.

2.2. Atom.

El nombre Atom hace referencia a dos estándares relacionados.

- El *Formato de Redifusión Atom* es un fichero en formato XML usado para Redifusión web.
- mientras que el *Protocolo de Publicación Atom* (resumido en inglés *AtomPub* o *APP*) es un protocolo simple basado en HTTP para crear o actualizar recursos en Web.

Las fuentes web permiten que los programas busquen actualizaciones del contenido publicado en un sitio Web. Para crear uno el propietario de un sitio Web puede usar software especializado, como un Sistema de gestión de contenido que publica una lista (o fuente web) de artículos recientes en un formato estándar, legible por máquinas. La fuente web puede ser descargada por sitios web que redifunden el contenido usando la fuente web, o por un agregador que permiten que los lectores en Internet se suscriban y vean los contenidos de la fuente web.

Una fuente web puede contener entradas, que pueden ser encabezados, artículos completos, resúmenes y/o enlaces al contenido de un sitio web.

El formato Atom fue desarrollado como una alternativa a RSS. Ben Trott fue uno de los defensores del nuevo formato que llegó a llamarse Atom. Él notó la incompatibilidad entre algunas versiones del protocolo RSS, ya que pensaba que los protocolos de publicación basados en XML-RPC no eran lo suficientemente interoperables.

Los proponentes del nuevo formato organizaron el grupo de trabajo **IETF Atom Publishing Format and Protocol**. El formato de redifusión ATOM fue publicado como un "estándar propuesto" de la IETF en el RFC 4287, y el protocolo de comunicación se publicó como RFC 5023.

2.3. XMLPullParser.

XML Pull Parser es una interfaz que define la funcionalidad de análisis proporcionada en XMLPULL V1 API.

Existen diferentes tipos de analizador según las características que se establecen:

- Analizador no validador como se define en la especificación XML 1.0 cuando `FEATURE_PROCESS_DOCDECL` se establece en verdadero.
- Validar el analizador tal como se define en la especificación XML 1.0 cuando `FEATURE_VALIDATION` es verdadero (y eso implica que `FEATURE_PROCESS_DOCDECL` es verdadero).
- Cuando `FEATURE_PROCESS_DOCDECL` es falso (esto es predeterminado y si se requiere un valor diferente debe modificarse antes de iniciarse el análisis), el analizador se comporta como un analizador no validador compatible con XML 1.0 bajo la condición de que no haya `DOCDECL` en documentos XML (las entidades internas aún pueden definirse con `defineEntityReplacementText()`). Este modo de operación está diseñado para funcionar en entornos restringidos como J2ME.

La versión de Java de XMLPULL V1 API proporciona:

- Interfaz simple: el analizador consiste en una interfaz, una excepción y una fábrica para crear el analizador.
- Implementación independiente: la clase de fábrica se modela después de JAXP y permite cambiar fácilmente a diferentes implementaciones de API XMLPULL V1 sin modificar el código fuente.
- Facilidad de uso: solo hay un método clave a `next()` que se utiliza para recuperar el próximo evento y solo hay cinco eventos:
 - `START_DOCUMENT`: inicio de documento - el analizador aún no ha leído ninguna entrada.
 - `START_TAG`: el analizador está en la etiqueta de inicio.
 - `TEXT`: analizador está en el contenido del elemento.
 - `END_TAG`: el analizador está en la etiqueta final.
 - `END_DOCUMENT`: documento terminado y no se permite más análisis.
- Versatilidad: es una interfaz genérica para el analizador XML y permite múltiples implementaciones y extensibilidad a través de características y propiedades.

- Rendimiento: la interfaz está diseñada para permitir la implementación de analizadores XML muy rápidos.
Requisitos mínimos : diseñado para ser compatible con J2ME (Java 2 Micro Edition) para trabajar y en dispositivos pequeños y para permitir la creación de analizadores compatibles con XMLPULL de memoria muy pequeña.

Capítulo 3: Android.

3.1. Introducción.

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Android es el sistema operativo móvil más utilizado del mundo, con una cuota de mercado superior al 80% al año 2017, muy por encima de IOS.

El éxito del sistema operativo lo ha convertido en objeto de litigios sobre patentes en el marco de las llamadas guerras de patentes entre las empresas de teléfonos inteligentes. Según los documentos secretos filtrados en 2013 y 2014, el sistema operativo es uno de los objetivos de las agencias de inteligencia internacionales.

La versión básica de Android es conocida como Android Open Source Project (AOSP).

El 25 de junio de 2014 en la Conferencia de Desarrolladores Google I/O, Google mostró una evolución de la marca Android, con el fin de unificar tanto el hardware como el software y ampliar mercados.

El 17 de mayo de 2017, se presentó Android Go. Una versión más ligera del sistema operativo para ayudar a que la mitad del mundo sin smartphone consiga uno en menos de cinco años. Incluye versiones especiales de sus aplicaciones donde el consumo de datos se reduce al máximo.

3.2. Breve historia.

En octubre de 2003, en la localidad de Palo Alto, Andy Rubin, Rich Miner, Chris White y Nick Sears fundan Android Inc. con el objetivo de desarrollar un sistema operativo para móviles basado en Linux.

En julio de 2005, la multinacional Google compra Android Inc. El 5 de noviembre de 2007 se crea la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. El mismo día se anuncia la primera versión del sistema operativo: Android 1.0 Apple Pie. Los terminales con Android no estarían disponibles hasta el año 2008. Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43.6 % en el tercer trimestre. A escala mundial alcanzó una cuota de mercado del 50.9 % durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (iOS de Apple, Inc.)

Tiene una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. A principios de 2018 se superaban ya los dos millones de aplicaciones disponibles en Google Play, la tienda de aplicaciones oficial de Android; a estas habría que añadir las disponibles en otras tiendas no oficiales, como Samsung Apps, de Samsung, SlideME, de Java, y Amazon Appstore. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. La tienda F-Droid es completamente de código abierto, así como sus aplicaciones, una alternativa al software privativo. Los programas están escritos en el lenguaje de programación Java.²¹ No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ello es descargada de sitios de terceros.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución hasta la versión 5.0, luego cambió al entorno Android Runtime (ART).

Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12

millones de líneas de código, incluyendo 3 millones de líneas de XML, 2.8 millones de líneas de lenguaje C, 2.1 millones de líneas de Java y 1.75 millones de líneas de C++.

3.3. ¿Por qué Android?: 8.0 Oreo

Android Oreo es el nombre de la última versión del sistema operativo móvil Android que anunció la firma Google el día 21 de marzo de 2017. Su nombre fue revelado el día 21 de agosto de 2017, el día del eclipse total de Sol en Estados Unidos. Se lanzó por primera vez como una versión previa para desarrolladores en fase alfa el 21 de marzo de 2017, para los móviles de Google (Nexus y Pixel). La segunda versión para desarrolladores se lanzó el 17 de mayo de 2017 y se considera en fase beta. La tercera versión para desarrolladores fue lanzada el 8 de junio de 2017 y contiene las API definitivas.

Las novedades con respecto a las otras versiones son:

- **Optimización.** El sistema limitará más los procesos en segundo plano, ahorrando mucha batería.
- **Notificaciones mejoradas.** Se podrán ordenar según categorías y definir la frecuencia con las que se reciben. También al deslizar la notificación hacia la izquierda aparecerá en la parte de los ajustes para controlar las notificaciones de la aplicación específica.
- **Notificaciones Dots.** Junto a los iconos de las aplicaciones aparecerá un número, dando a entender el número de tareas pendientes o sin leer.
- **Diseño.** Se han modificado varias características tanto visuales como auditivas.
- **Plataforma.** Agregará soporte para redes inalámbricas (NAN) para Wi-Fi basadas en Wi-Fi Aware, gamas de color amplias en aplicaciones, una API para autofillers, multiprocesos y soporte de navegación segura de Google para WebViews, una API que permite al sistema aplicaciones de VoIP, y actividades de lanzamiento en pantallas remotas. Android Runtime (ART) ofrece mejoras de rendimiento. Ofrecerá una distribución adaptada para dispositivos de gama baja conocidos como Android Go, que se utilizará en todos los dispositivos con 1 GB de RAM o menos.

3.4. Diseño y desarrollo.

Android es considerado como uno de los modelos de negocio más exitosos, pues su desarrollo estratégico contempla los factores que más se tienen en cuenta dentro de las herramientas y metodologías desarrollados por expertos en negocios. Este sistema operativo se ha convertido en un modelo a seguir por desarrolladores de tendencias y negocios de alto impacto.

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas todavía no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de poca RAM, velocidad de procesador, etc.

En un principio, Android era eminentemente un sistema operativo pensado para usar con teclado, y gracias a un cursor poder navegar entre las aplicaciones. Desde su comienzo, Android ha sido altamente personalizable. Poco después, antes del lanzamiento del primer teléfono Android, esta filosofía cambió para convertirse en eminentemente táctil, y poder competir contra el iPhone, presentado un año y nueve meses antes.

Capítulo 4: Planificación y Desarrollo del proyecto.

4.1. Riesgos y plan de contingencia.

A continuación, se enumeran y se detallan los principales riesgos que son resultado del estudio de estos, el impacto que conllevan, la probabilidad de producirse y en qué fase del desarrollo del proyecto pueden darse.

4.1.1. Riesgos.

La exposición de los riesgos ha sido calculada utilizando el ISO/IEC 27005:2008 (definido en el glosario).

Nombre del riesgo	Modificación de los requisitos.
Detalle	Tras haber formulado el análisis de la aplicación, puede ser necesario la modificación de los requisitos de la aplicación y revisar el trabajo realizado hasta el momento.
Contexto	Se puede dar en cualquiera de las fases.
Categoría	De proceso.
Impacto	Leve en las fases iniciales y grave en las fases finales.
Consecuencias	Dependerán de la importancia del requisito y la fase en que se produzca la modificación.

Tabla 1. Riesgo: Modificación de los requisitos.

Nombre del riesgo	Retraso respecto a la planificación.
Detalle	A causa de una mala o insuficiente planificación pueden darse retrasos inesperados, o que se necesite más tiempo del planificado en algunas etapas por inexperiencia.
Contexto	Se puede dar en cualquiera de las fases.
Categoría	De proceso, de gestión.
Impacto	Grave
Consecuencias	Retraso en la entrega de los distintos artefactos, así como del producto final.

Tabla 2. Riesgo: Retraso respecto a la planificación.

Nombre del riesgo	Necesidad del estudio de la tecnología a usar.
Detalle	Puede que se necesite utilizar una tecnología que se ajuste al proyecto y que no se domine.
Contexto	Se puede dar en cualquiera de las fases.
Categoría	De proyecto.
Impacto	Crítico.
Consecuencias	Retraso en la entrega de los distintos artefactos, así como del producto final.

Tabla 3. Riesgo: Necesidad del estudio de la tecnología a usar.

Nombre del riesgo	Fallos al probar la aplicación en distintos dispositivos.
Detalle	Puede que una vez desarrollada la aplicación, cambie el aspecto y el funcionamiento de esta en dispositivos Android con características diferentes como la pantalla.
Contexto	Se aplica en la etapa de despliegue.
Categoría	De proceso.
Impacto	Leve
Consecuencias	No adaptación a ciertos dispositivos, y estos serán puestos como no compatibles.

Tabla 4. Riesgo: Fallos al probar la aplicación en distintos dispositivos.

4.1.2. Planes de contingencia.

En este apartado se detallarán los planes de contingencia para los riesgos indicados en el apartado anterior; se expone el escenario en el que puede aparecer el riesgo, los puntos del desarrollo en los que se comprueba la aparición de este y la estrategia seguida para reducir o en el mejor de los casos evitar los efectos perjudiciales que afectan al buen desarrollo del software.

Nombre del riesgo	Modificación de los requisitos.
Escenario	Las consecuencias pueden variar dependiendo de en qué fase del proyecto se produzca la modificación y de la importancia del requisito. Repercutirá en todos los artefactos que guarden relación con dicho requisito.
Punto de comprobación	Se comprobará regularmente a lo largo del proyecto, pero especialmente en las primeras fases.
Estrategia	Protección de riesgo.
Plan de acción	Revisión de los requisitos y el modelo de casos de uso.
Monitorización	Entrevistas y encuentros con el tutor.

Tabla 5. Plan de contingencia: Modificación de requisitos.

Nombre del riesgo	Retraso respecto a la planificación.
Escenario	Se deberá reducir en la medida de lo posible, teniendo en cuenta los recursos disponibles, el tiempo entre la planificación y el desarrollo real del proyecto.
Punto de comprobación	Se comprueba regularmente durante todo el proyecto.
Estrategia	Reducción de riesgo.
Plan de acción	En cada hito incluir los retrasos y documentarlos para evitarlo en casos posteriores y hacer una reasignación de recursos para mitigar el retraso.
Monitorización	En cada uno de los hitos del proyecto.

Tabla 6. Plan de contingencia: Retraso respecto a la planificación.

Nombre del riesgo	Necesidad del estudio de la tecnología a usar.
Escenario	Se carecía de los conocimientos previos de la tecnología a usar en el proyecto, por lo que fue necesario un aumento del tiempo dedicado para el aprendizaje de los conocimientos necesarios.
Punto de comprobación	Se comprueba regularmente durante todo el proyecto,
Estrategia	Reducción de riesgo.
Plan de acción	Aumento del tiempo dedicado para adquirir los conocimientos necesarios.
Monitorización	Entrevista y encuentros con el tutor.

Tabla 7. Plan de contingencia: Necesidad del estudio de la tecnología a usar.

4.2. Planificación del proyecto.

4.2.1. Fases del proyecto.

La planificación está programada de acuerdo con 2 incrementos, se ha decidido así para minimizarlos retrasos por los cambios de requisitos y de los diseños teniendo así para cada incremento todas las fases desde el “análisis” hasta las “pruebas”.

Incremento	Duración (días)	Fases
Planificación	15	Planificación y gestión de riesgos.
Incremento 1	48	Análisis Diseño Implementación Pruebas
Incremento 2	48	Análisis Diseño Implementación Pruebas

Tabla 8. Fases del TFG.

A continuación, se muestra el diagrama de Gantt en el cual hemos de apuntar dos cosas, en primera instancia los fines de semana están dentro de nuestro calendario.

Lo segundo es que no se han incluido los recursos puesto que el proyecto lo ha llevado a cabo una sola persona y por lo tanto en cada uno de los puntos los recursos para esa persona alcanzarán el 100% y no habrá recursos sobre asignados.

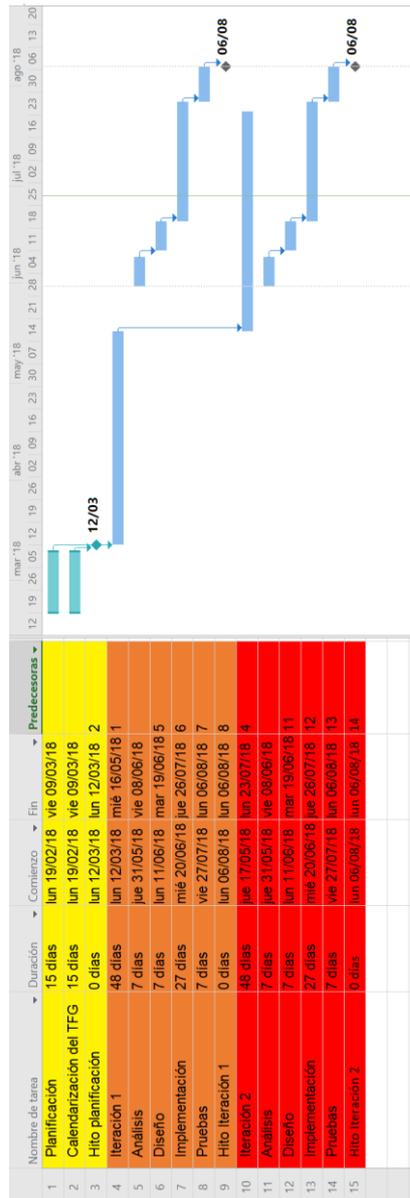


Ilustración 1. Diagrama de Gantt

4.2.2. Presupuestos.

A continuación, se detallan los costes previstos teniendo en cuenta tanto los recursos técnicos como humanos. En el caso de los recursos humanos se ha usado el “Convenio Colectivo de Oficinas y Despachos Valladolid 2015-2018” para un grupo profesional IV de nivel II para obtener la cantidad de euros/horas que gana el trabajador. Además, para disminuir el coste total del proyecto se ha optado por usar software de licencia gratuita u ofrecida por la Universidad de Valladolid. Debido que el coste de este tipo de software es 0 no se ha incluido en la tabla de costes iniciales.

Costes Iniciales:

Se ha optado por hacer un ajuste aplicado como prevención a los posibles riesgos que pudieran aparecer debido a la inexperiencia tanto a la hora de planificar como a un posible incremento en las horas del trabajo a causa de la tecnología usada durante la implementación.

Concepto	Descripción	Importe (€)
Coste del trabajador	85 días x 6h/día* 11,5€/día	5865
Equipamiento	Licencia Microsoft Office 2016	0
	Ordenador portátil Xiaomi Air 13 Notebook	600
	Smartphone Xiaomi Mi5	200
Coste total inicial		6665
Ajuste aplicado	25% del coste del trabajador	1466,25
Coste total		7331,25

Tabla 9. Costes iniciales.

Costes reales:

Aunque se aplicó un ajuste para prevenir el riesgo del aumento de horas que podía existir, no ha sido suficiente y nos hemos encontrado con incremento de 517,5€ por encima de lo presupuestado.

Concepto	Descripción	Importe (€)
Coste total presupuestado		7331,25
Coste real	111 días x 6h/día* 11,5€/día + 800€ de equipamiento	8459
Diferencia de costes	7331,25- 8459	-1127,75

Tabla 10. Costes reales.

Capítulo 5: Análisis.

5.1. Actores.

En esta sección describiremos los distintos actores que interactúan con la aplicación.

Todos los actores pueden interactuar de forma similar, ya que no se necesita que el actor se cree una cuenta y tenga que iniciar y cerrar sesión para tener otros privilegios.

A continuación, se muestran los actores que interactúan con la aplicación:

ACT-001	Usuario
Autor	Nuria Cancho Diaz
Descripción	Cualquier usuario que interactúe con el sistema.

Tabla 11. Actor: Usuario.

5.2. Requisitos.

A continuación, se hallan los distintos tipos de requisitos que se plantean en el proyecto.

5.2.1. Requisitos funcionales.

Un requisito funcional se puede definir como una descripción de lo que un sistema debe hacer. Este tipo de requisito especifica algo que el sistema entregado debe ser capaz de realizar.

Los requisitos funcionales que hemos definido para este sistema son:

FR-001	Mostrar noticias por categoría.
Autores	Nuria Cancho Diaz
Versión	1.0
Dependencias	
Descripción	El sistema deberá mostrar al usuario las últimas noticias publicadas en una categoría.

Tabla 12. Requisito funcional: Mostrar noticias por categoría.

FR-002	Actualización de noticias por categoría.
Autores	Nuria Cancho Diaz
Versión	1.0
Dependencias	[FR-001] Mostrar noticias por categoría.
Descripción	El sistema deberá actualizar las noticias cada cierto tiempo.

Tabla 13. Requisito funcional: Actualización de noticias por categoría.

FR-003	Gestionar notificaciones.
Autores	Nuria Cancho Diaz
Versión	1.0
Dependencias	
Descripción	El sistema deberá dejar gestionar al usuario las notificaciones sobre la aplicación.

Tabla 14. Requisito funcional: Gestionar notificaciones.

FR-004	Gestionar menú.
Autores	Nuria Cancho Diaz
Versión	1.0
Dependencias	
Descripción	El sistema deberá dejar gestionar al usuario las distintas opciones que se encuentran en el menú.

Tabla 15. Requisito funcional: Gestionar menú.

FR-005	Mostrar noticia completa.
Autores	Nuria Cancho Diaz
Versión	1.0
Dependencias	[FR-001] Mostrar noticias por categoría.
Descripción	El sistema deberá mostrar al usuario la noticia completa que ha seleccionado.

Tabla 16. Requisito funcional: Mostrar noticia completa.

5.2.2. Requisitos no funcionales.

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos.

Los requisitos no funcionales que se hallan en el proyecto son los nombrados a continuación:

NFR-001	Sistema operativo.
Autores	Nuria Cancho Diaz
Versión	1.0
Descripción	El sistema deberá funcionar en Smartphone o Tablet con OS Android 5.0 (Lollipop) o superior.

Tabla 17. Requisito no funcional: Sistema operativo.

NFR-002	Conexión a internet.
Autores	Nuria Cancho Diaz
Versión	1.0
Descripción	El sistema deberá tener conexión a internet para poder actualizar y visualizar correctamente el contenido.

Tabla 18. Requisito no funcional: Conexión a internet.

NFR-003	Usabilidad.
Autores	Nuria Cancho Diaz
Versión	1.0
Descripción	El 90% de los usuarios deben ser capaces de utilizar el sistema con una tasa de error del 5%.

Tabla 19. Requisito no funcional: Usabilidad.

NFR-004	Aprendizaje
Autores	Nuria Cancho Diaz
Versión	1.0
Descripción	El sistema deberá permitir que un usuario medio de Android sea capaz de manejar la aplicación menos de 5 minutos, en más del 90 % de los casos.

Tabla 20. Requisito no funcional: Aprendizaje.

5.2.3. Requisitos de información.

Los requisitos de información son una forma especializada de requisitos funcionales que indican el tipo de información que guarda el sistema.

Los requisitos de información que se han obtenido en este proyecto son:

IRQ-001	Noticia
Descripción	El sistema deberá almacenar la información correspondiente a la Noticia.
Datos específicos.	<ul style="list-style-type: none">• Id de la Noticia.• Título de la Noticia.• Descripción de la noticia.• Lista de Noticias.• Url de la imagen.

Tabla 21. Requisito de información: Noticia.

5.3. Casos de uso.

En el diagrama de los casos de uso se muestra las distintas funcionalidades que te proporciona la aplicación, habiendo analizado antes los requisitos y los actores que interactúan con el sistema.

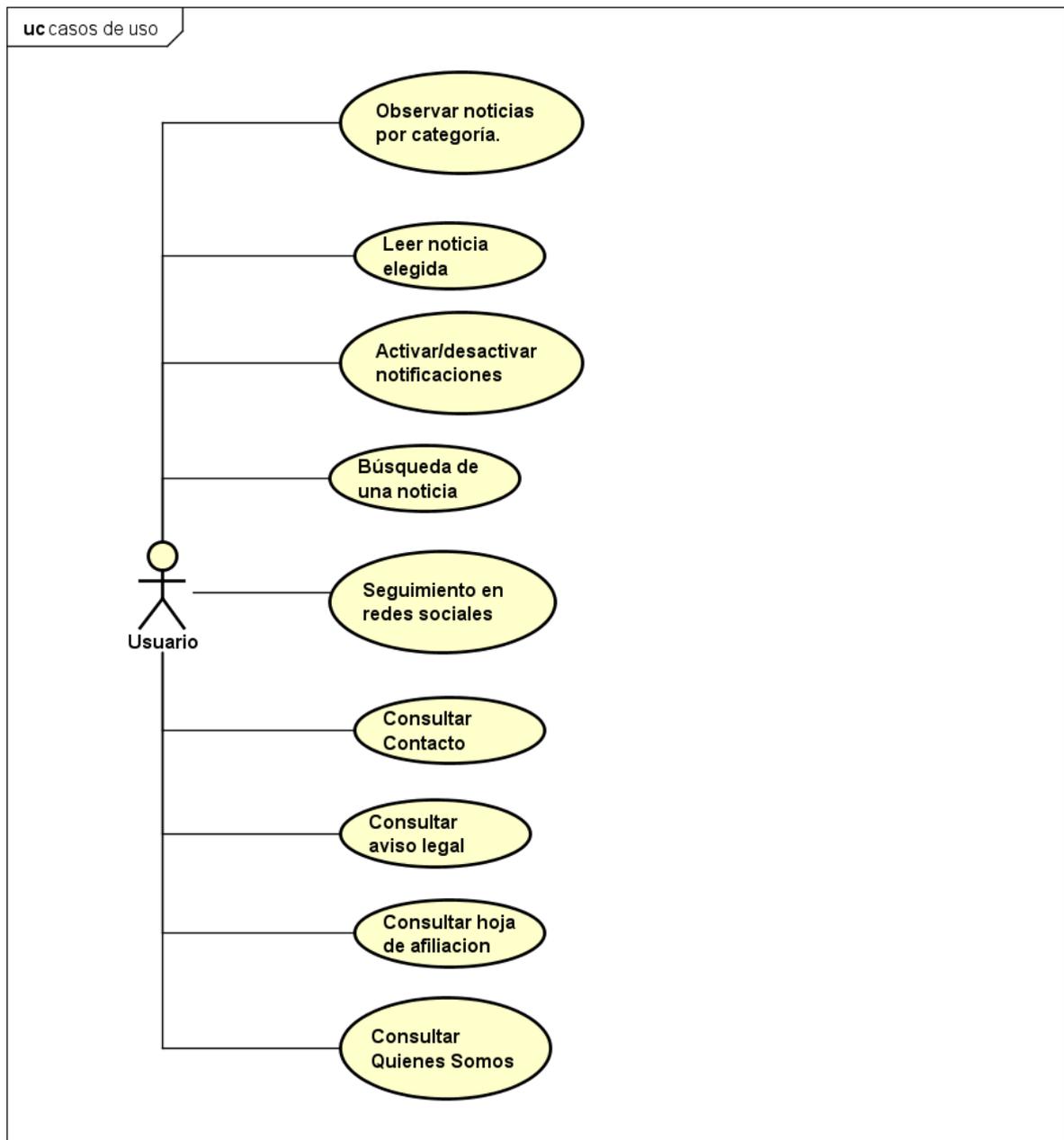


Ilustración 2. Diagrama de casos de uso.

A continuación, se puede ver los casos de uso en detalle, mostrando la funcionalidad de la aplicación.

UC-001		Observar noticias por categoría.
Autores	Nuria Cancho Diaz	
Versión	1.0	
Actor	[ACT-001]	
Precondición	Ninguna.	
Secuencia normal	1.- El actor despliega el menú izquierdo. 2.- El sistema muestra el menú izquierdo con las distintas categorías. 3.- El actor solicita la categoría elegida. 4.- El sistema comprueba la categoría elegida y muestra las noticias de dicha categoría. 5.- El actor observa las noticias de la categoría solicitada.	
Postcondición	El caso se ha creado correctamente y el actor puede visualizar las noticias de la categoría elegida.	
Flujos alternativos	4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso.	

Tabla 22. Caso de uso: Observar noticias por categoría.

UC-002		Leer noticia elegida.
Autores	Nuria Cancho Diaz	
Versión	1.0	
Actor	[ACT-001]	
Precondición	Estar en una categoría elegida previamente.	
Secuencia normal	1.- El actor observa las noticias de la categoría. 2.- El sistema muestra las noticias de la categoría. 3.- El actor solicita la noticia elegida. 4.- El sistema comprueba la noticia elegida y muestra la noticia de dicha categoría. 5.- El actor observa la noticia completa de la categoría solicitada.	
Postcondición	El caso se ha creado correctamente y el actor puede visualizar la noticia completa de la categoría elegida.	
Flujos alternativos	4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso.	

Tabla 23. Caso de uso: Leer la noticia elegida.

UC-003		Activar / desactivar las notificaciones.
Autores	Nuria Cancho Diaz	
Versión	1.0	
Actor	[ACT-001]	
Precondición	Ninguna.	
Secuencia normal	1.- El actor despliega el menú derecho. 2.- El sistema muestra el menú derecho con las distintas posibilidades. 3.- El actor solicita la opción “notificaciones”. 4.- El sistema comprueba el estado de las notificaciones y lo muestra en pantalla. 5.- El actor observa el estado de las notificaciones y decide si modificarlo. 6.- El sistema comprueba si se ha modificado y guarda el nuevo estado.	
Postcondición	El caso se ha creado correctamente y el actor ha podido modificar la opción “notificaciones”.	
Flujos alternativos	4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso. 5.1.- El usuario no realiza ninguna acción y finaliza el caso.	

Tabla 24. Caso de uso: Activar / desactivar las notificaciones.

UC-004		Búsqueda de una noticia.
Autores	Nuria Cancho Diaz	
Versión	1.0	
Actor	[ACT-001]	
Precondición	La existencia de la noticia a buscar.	
Secuencia normal	1.- 1.- El actor despliega el menú derecho. 2.- El sistema muestra el menú derecho con las distintas posibilidades. 3.- El actor solicita la opción “Buscar”. 4.- El sistema muestra la pantalla de buscar. 5.- El actor introduce unas palabras clave y pulsa el botón buscar. 6.- El sistema busca y muestra las coincidencias con las palabras clave. 7.- El actor elige la noticia que buscaba mediante las palabras clave. 8.- El sistema muestra la opción elegida por el actor.	
Postcondición	El caso se ha creado correctamente y el actor puede visualizar el contenido elegido.	

Flujos alternativos	<p>4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso.</p> <p>6.1.- El sistema no encuentra nada y muestra un mensaje informativo.</p> <p>7.1.- El actor no encuentra lo que quería y el caso vuelve al punto 5.</p> <p>7.2.- El usuario no realiza ninguna acción y finaliza el caso.</p>
----------------------------	---

Tabla 25. Caso de uso: Búsqueda de una noticia.

UC-005		Seguimiento en redes sociales.
Autores	Nuria Cancho Diaz	
Versión	1.0	
Actor	[ACT-001]	
Precondición	Ninguna.	
Secuencia normal	<p>1.- El actor despliega el menú derecho.</p> <p>2.- El sistema muestra el menú derecho con las distintas posibilidades.</p> <p>3.- El actor solicita la opción "Síguenos".</p> <p>4.- El sistema muestra en pantalla las distintas redes sociales.</p> <p>5.- El actor observa las opciones de "Síguenos" y elige una.</p> <p>6.- El sistema te redirecciona a la opción elegida.</p> <p>7.- El actor decide "seguir" en la red social elegida.</p>	
Postcondición	El caso se ha creado correctamente y el actor ha podido acceder a las redes sociales del cliente.	
Flujos alternativos	<p>4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso.</p> <p>5.1.- El usuario no realiza ninguna acción y finaliza el caso.</p>	

Tabla 26. Caso de uso: Seguimiento en redes sociales.

UC-006		Consultar contacto.	
Autores		Nuria Cancho Diaz	
Versión		1.0	
Actor		[ACT-001]	
Precondición		Ninguna.	
Secuencia normal		1.- El actor despliega el menú izquierdo. 2.- El sistema muestra el menú izquierdo con las distintas posibilidades. 3.- El actor solicita la opción "Contacto". 4.- El sistema muestra en pantalla los contactos por provincia. 5.- El actor observa las opciones el "Contacto".	
Postcondición		El caso se ha creado correctamente y el actor ha podido acceder a los contactos del cliente.	
Flujos alternativos		4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso.	

Tabla 27. Caso de uso: Observar contacto.

UC-007		Consultar aviso legal.	
Autores		Nuria Cancho Diaz	
Versión		1.0	
Actor		[ACT-001]	
Precondición		Ninguna.	
Secuencia normal		1.- El actor despliega el menú izquierdo. 2.- El sistema muestra el menú izquierdo con las distintas posibilidades. 3.- El actor solicita la opción "Aviso Legal". 4.- El sistema muestra en pantalla la información del aviso legal del cliente. 5.- El actor observa la información del "Aviso Legal".	
Postcondición		El caso se ha creado correctamente y el actor ha podido acceder al aviso legal del cliente.	
Flujos alternativos		4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso.	

Tabla 28. Caso de uso: Consultar aviso legal.

UC-008		Consultar hoja de afiliación.
Autores	Nuria Cancho Diaz	
Versión	1.0	
Actor	[ACT-001]	
Precondición	Ninguna.	
Secuencia normal	1.- El actor despliega el menú izquierdo. 2.- El sistema muestra el menú izquierdo con las distintas posibilidades. 3.- El actor solicita la opción "Afiliate". 4.- El sistema muestra en pantalla la hoja de afiliación. 5.- El actor observa la hoja y puede descargarla en las herramientas de Google.	
Postcondición	El caso se ha creado correctamente y el actor ha podido acceder a la hoja de afiliación.	
Flujos alternativos	4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso.	

Tabla 29. Caso de uso: Consultar hoja de afiliación.

UC-009		Consultar Quienes somos.
Autores	Nuria Cancho Diaz	
Versión	1.0	
Actor	[ACT-001]	
Precondición	Ninguna.	
Secuencia normal	1.- El actor despliega el menú izquierdo. 2.- El sistema muestra el menú izquierdo con las distintas posibilidades. 3.- El actor solicita la opción "Quienes Somos". 4.- El sistema muestra en pantalla la información de quienes somos del cliente. 5.- El actor observa la información del "Quienes Somos".	
Postcondición	El caso se ha creado correctamente y el actor ha podido acceder a la información de quienes somos del cliente.	
Flujos alternativos	4.1.- El sistema muestra un mensaje de error falta de conexión a internet y finaliza el caso de uso.	

Tabla 30. Caso de uso: Consultar Quienes Somos.

5.4. Modelo de dominio.

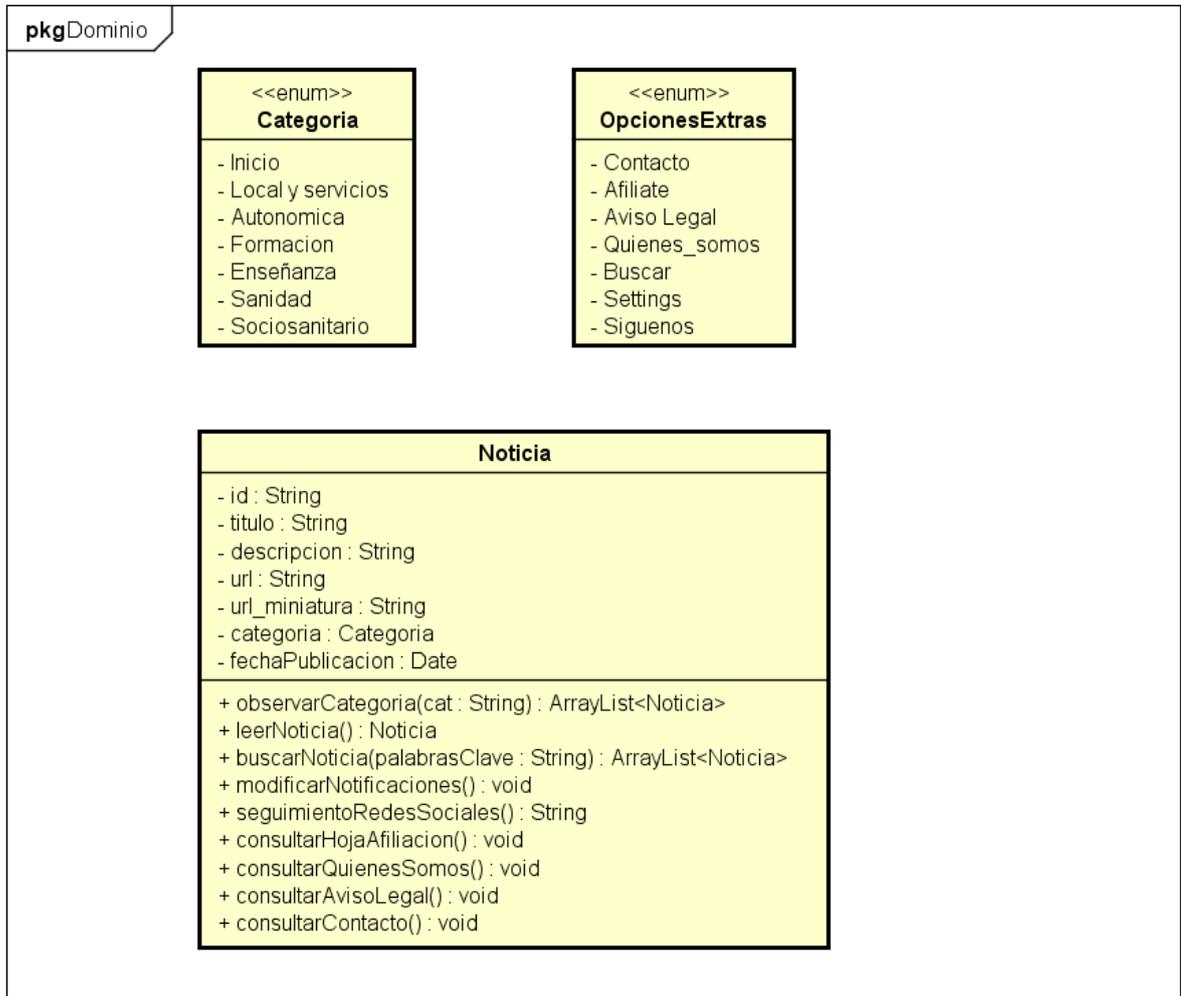


Ilustración 3. Modelo de Dominio

5.5. Modelos dinámicos.

El modelo dinámico se usa para expresar y modelar el comportamiento del sistema a lo largo del tiempo, con el que describe las relaciones temporales entre objetos. Muestran las interacciones entre objetos ocurridas en un escenario del sistema. Los modelos dinámicos se les va a representar con diagramas de secuencia que se pueden observar a continuación:

5.5.1. CU Observar noticias por categoría.

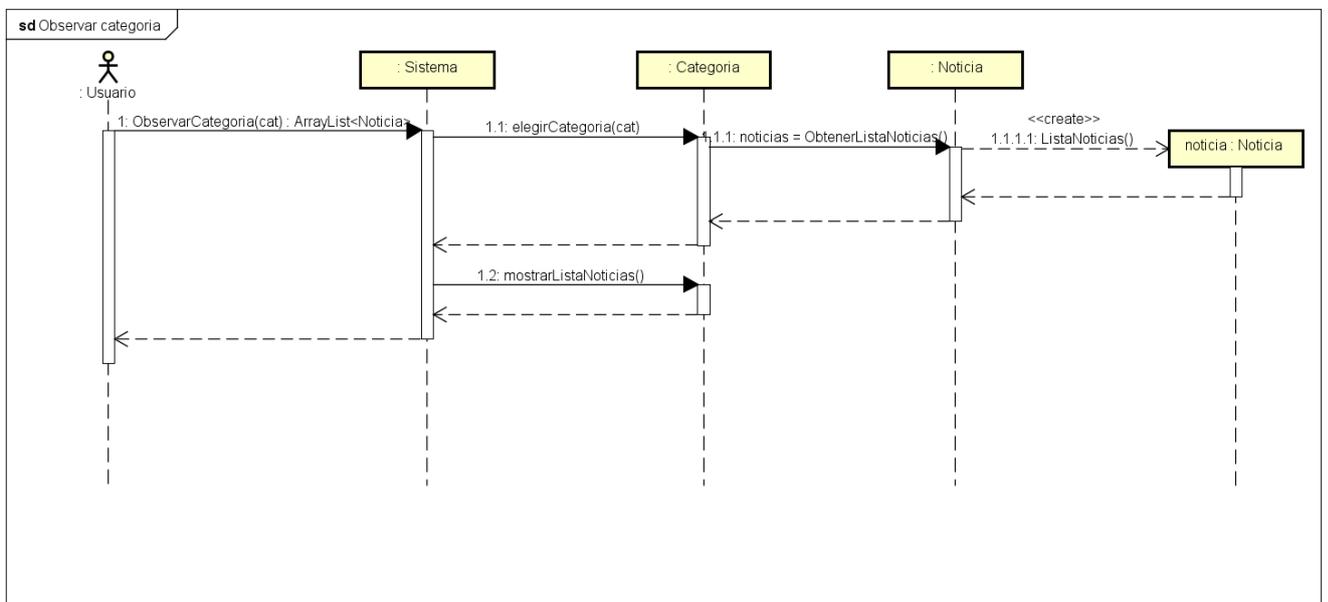


Ilustración 4. Caso de uso: Observar noticias por categoría.

5.5.2. CU Leer noticia elegida.

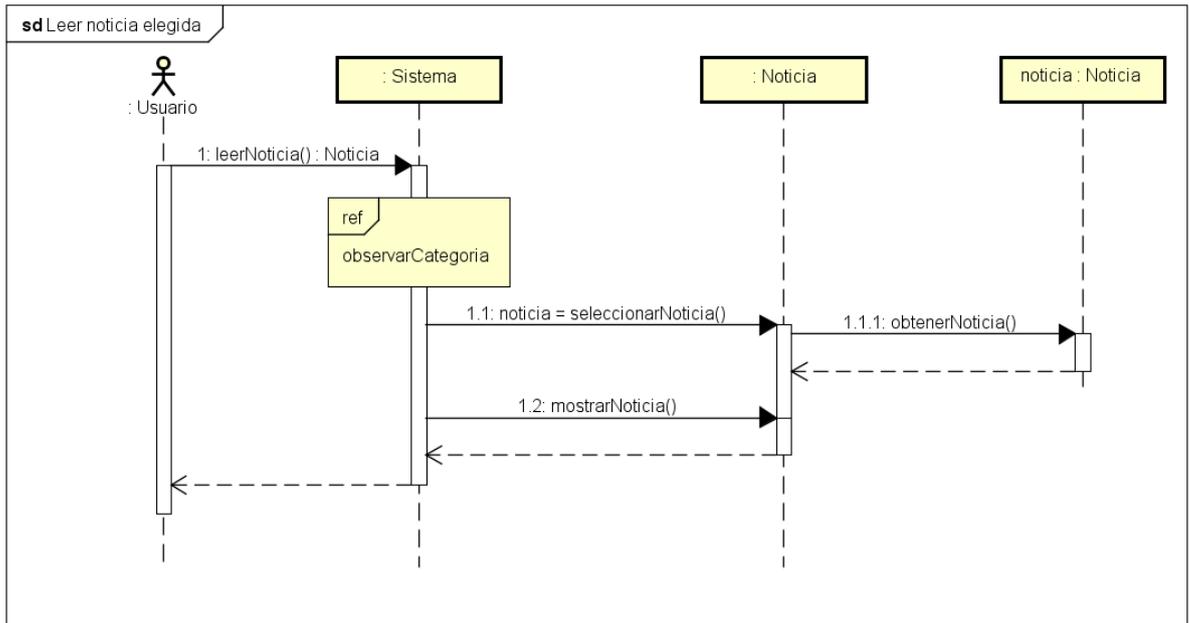


Ilustración 5. Caso de uso: Leer noticia elegida.

5.5.3. CU Activar/desactivar notificaciones.

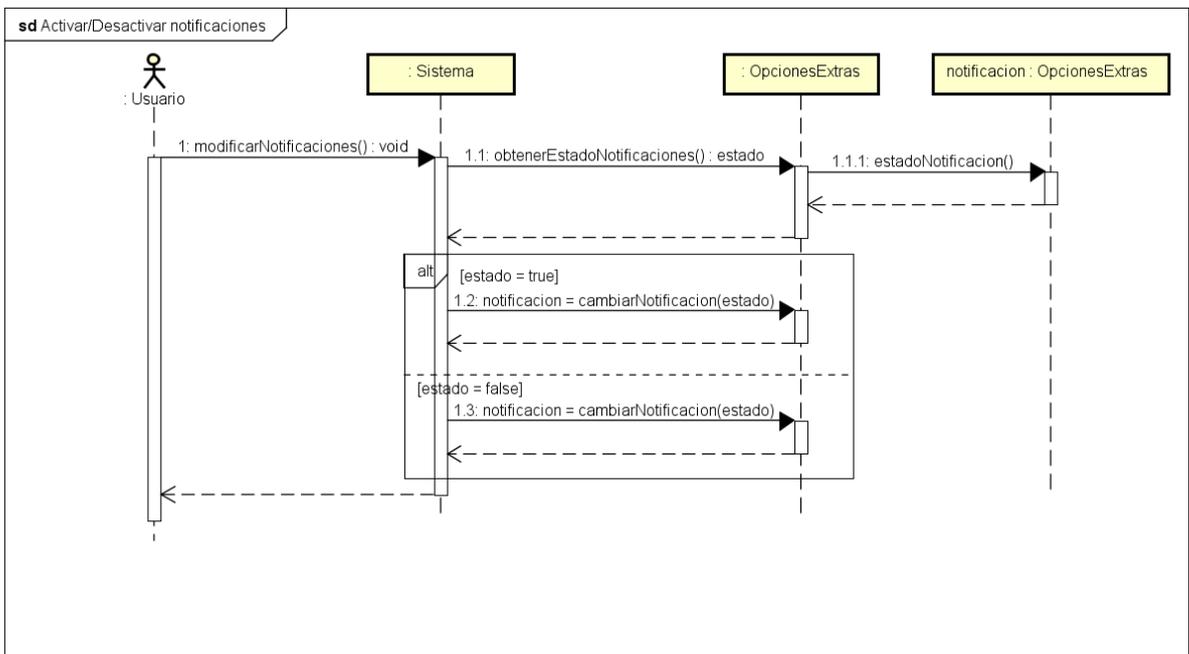


Ilustración 6. Caso de uso: activar/desactivar notificaciones.

5.5.4. CU Búsqueda de una noticia.

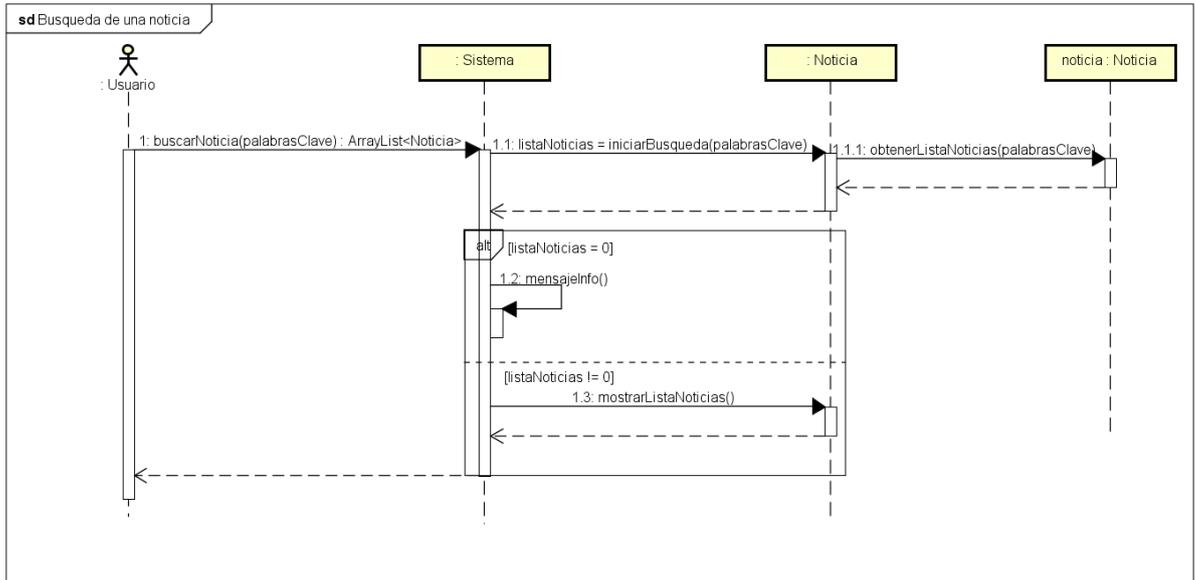


Ilustración 7. Caso de uso: Búsqueda de una noticia.

5.5.5. CU Seguimiento en redes sociales.

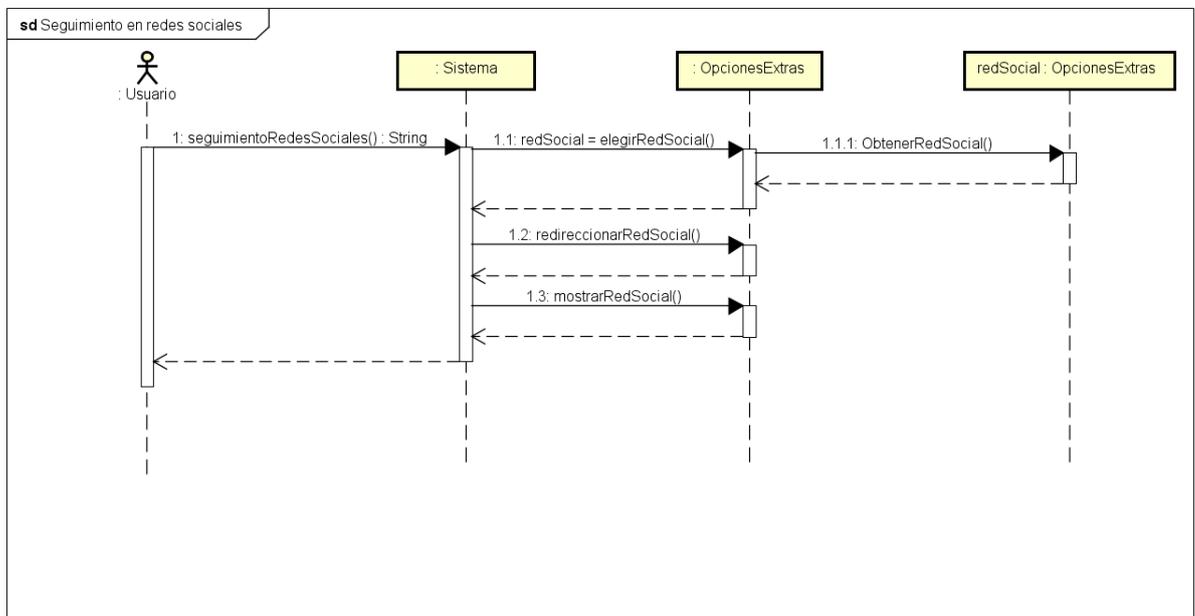


Ilustración 8. Caso de uso: Seguimiento en redes sociales.

5.5.6. CU Consultar Contacto.

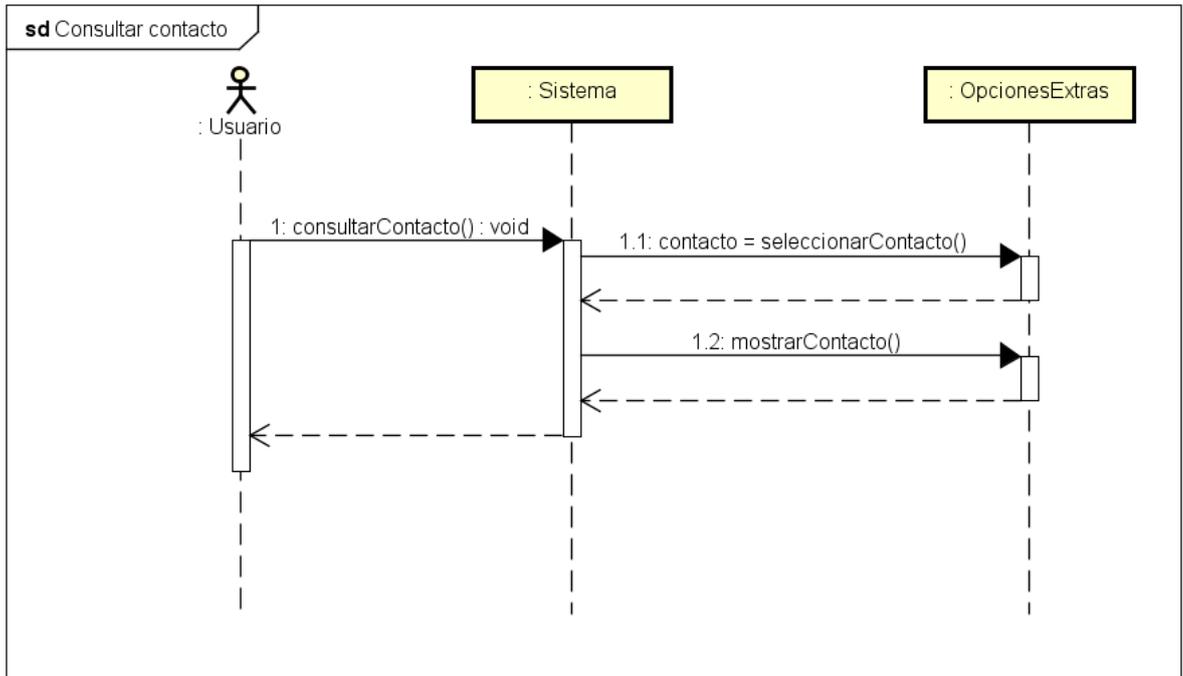


Ilustración 9. Caso de uso: Consultar Contacto.

5.5.7. CU Consultar Quienes Somos.

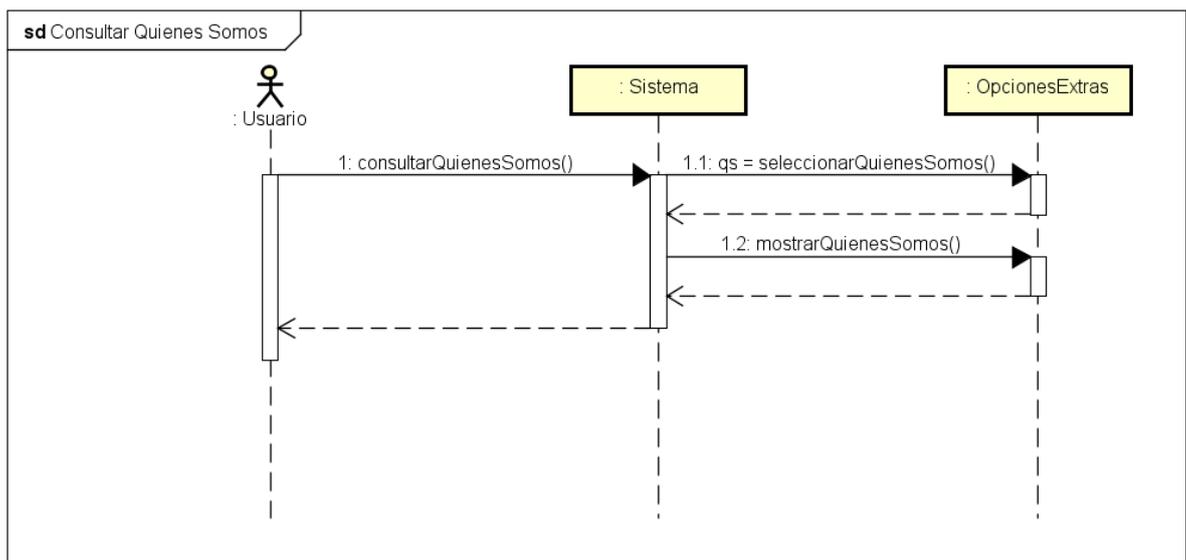


Ilustración 10. Caso de uso: Consultar Quienes Somos.

5.5.8. CU Consultar Aviso Legal.

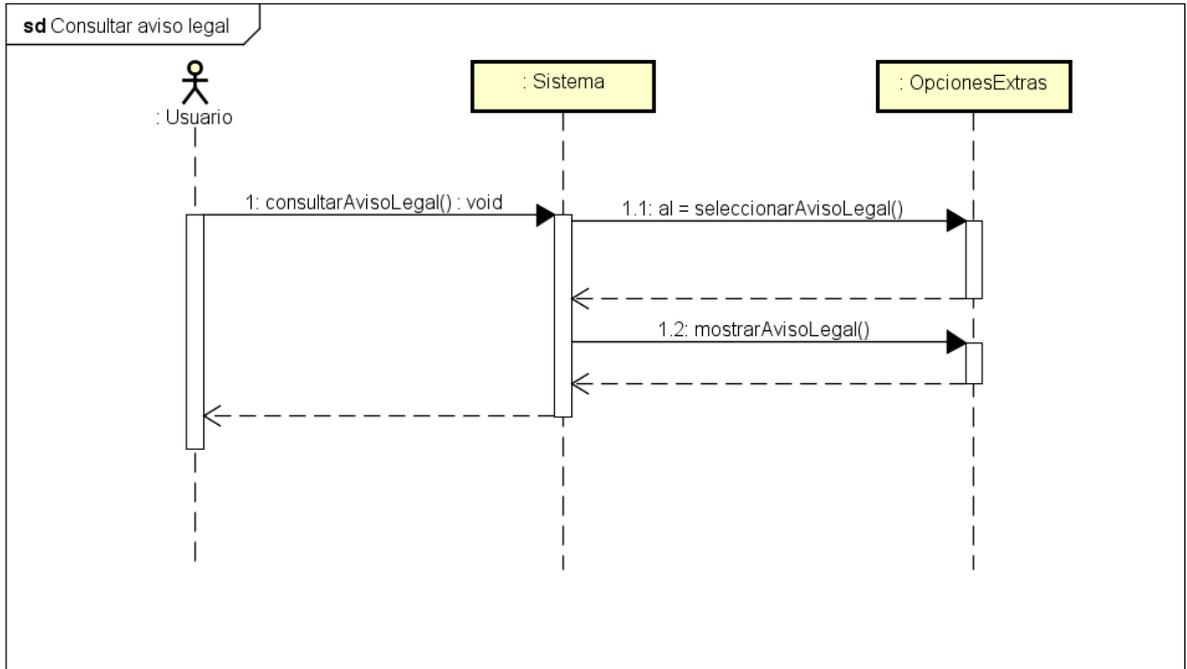


Ilustración 11. Caso de uso: Consultar Aviso Legal.

5.5.9. CU Consultar hoja de afiliación.

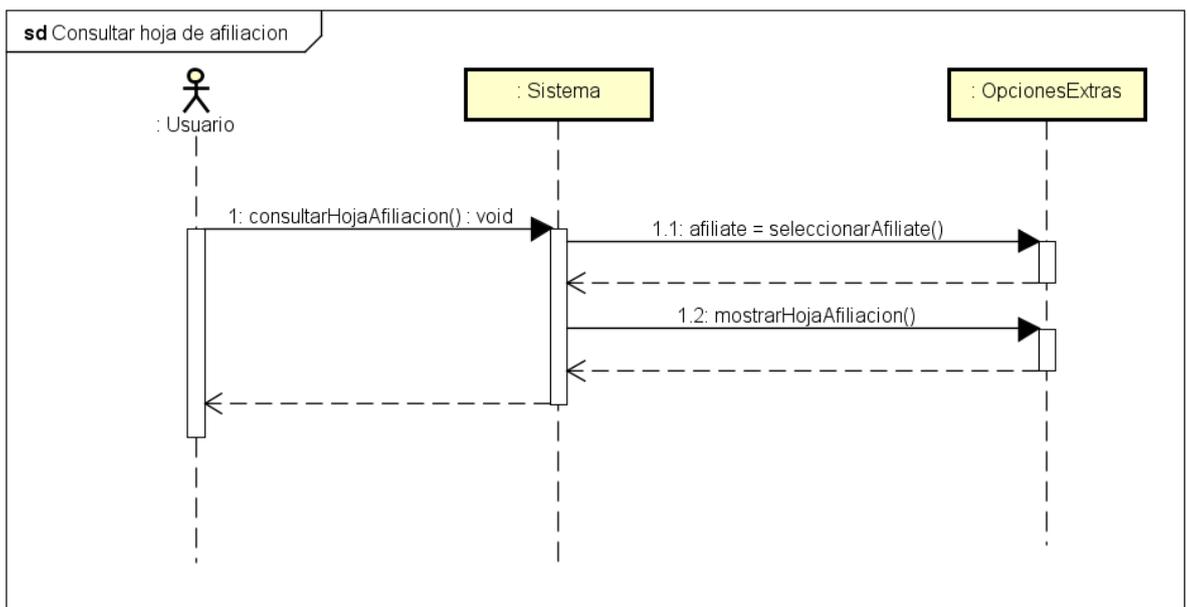


Ilustración 12. Caso de uso: Consultar hoja de afiliación.

5.6. Modelo entidad – relación.

Un diagrama entidad-relación es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. El siguiente modelo entidad – relación representa nuestro sistema de información.

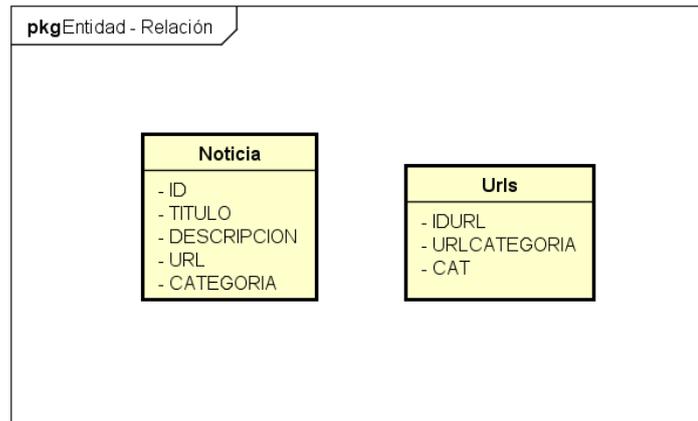


Ilustración 13. Diagrama Entidad - Relación.

Capítulo 6: Diseño.

En este capítulo se detallarán las metodologías de diseño que se tendrán en cuenta durante la etapa de implementación del proyecto. El objetivo principal de este es construir una aplicación móvil de calidad, que sea capaz de adaptarse a futuras mejoras.

6.1. Descripción de la arquitectura.

6.2.1. MVC.

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

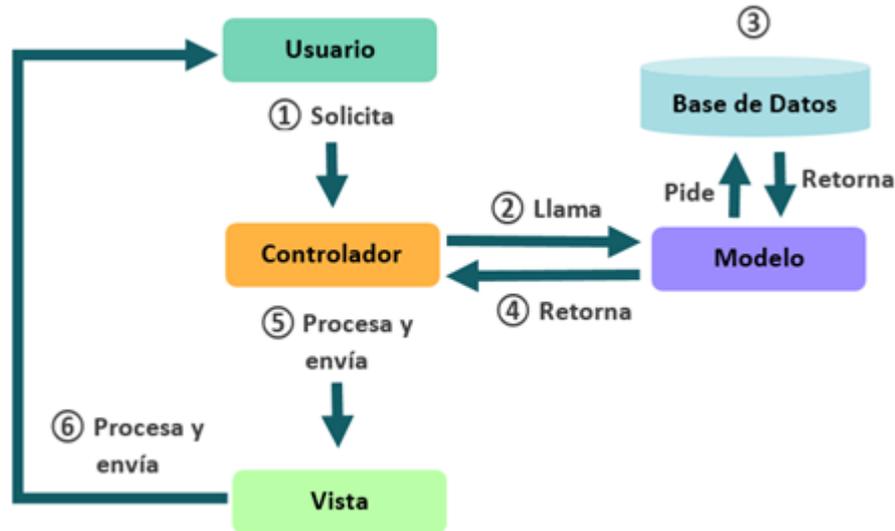


Ilustración 14. Modelo - Vista - Controlador.

(Fuente [7])

Modelo: donde se guarda toda la lógica del negocio de una aplicación, la lógica del negocio puede ser cualquier cosa específica acerca de cómo una aplicación almacena los datos, o utiliza servicios de terceros con el fin de cumplir con sus necesidades. Si la aplicación debe acceder a la información en una base de datos, el código deberá estar guardado en el modelo.

Vista: donde se encontrarán todos los elementos de la interfaz de usuario de una aplicación. Cualquier cosa que el usuario pueda ver, es guardada en la vista, y algunas veces lo que ve el usuario actualmente es la combinación de varias vistas en la misma petición.

Controlador: componente encargado de conectar el modelo con la vista. Los controladores aíslan la lógica del negocio de un modelo de los elementos de la interfaz de usuario de una vista y maneja la forma en la que la aplicación responde a la interacción del usuario en la vista. Los controladores son el primer punto de entrada en estos componentes, ya que la primera solicitud se pasa a un controlador, que luego instancia a los modelos y vistas requeridas para cumplir con una petición a la aplicación.

6.2.2. Arquitectura global del sistema.

La estructura de un proyecto en Android está compuesta por paquetes que contienen actividades, clases java, layouts y recursos en formato xml. Si tomamos las clases java como posibles modelos, las actividades como controladores y los layout como la vista, podríamos entonces aplicar el MVC.

A continuación, se muestra la estructura de paquetes de la aplicación desarrollada siguiendo el MVC.

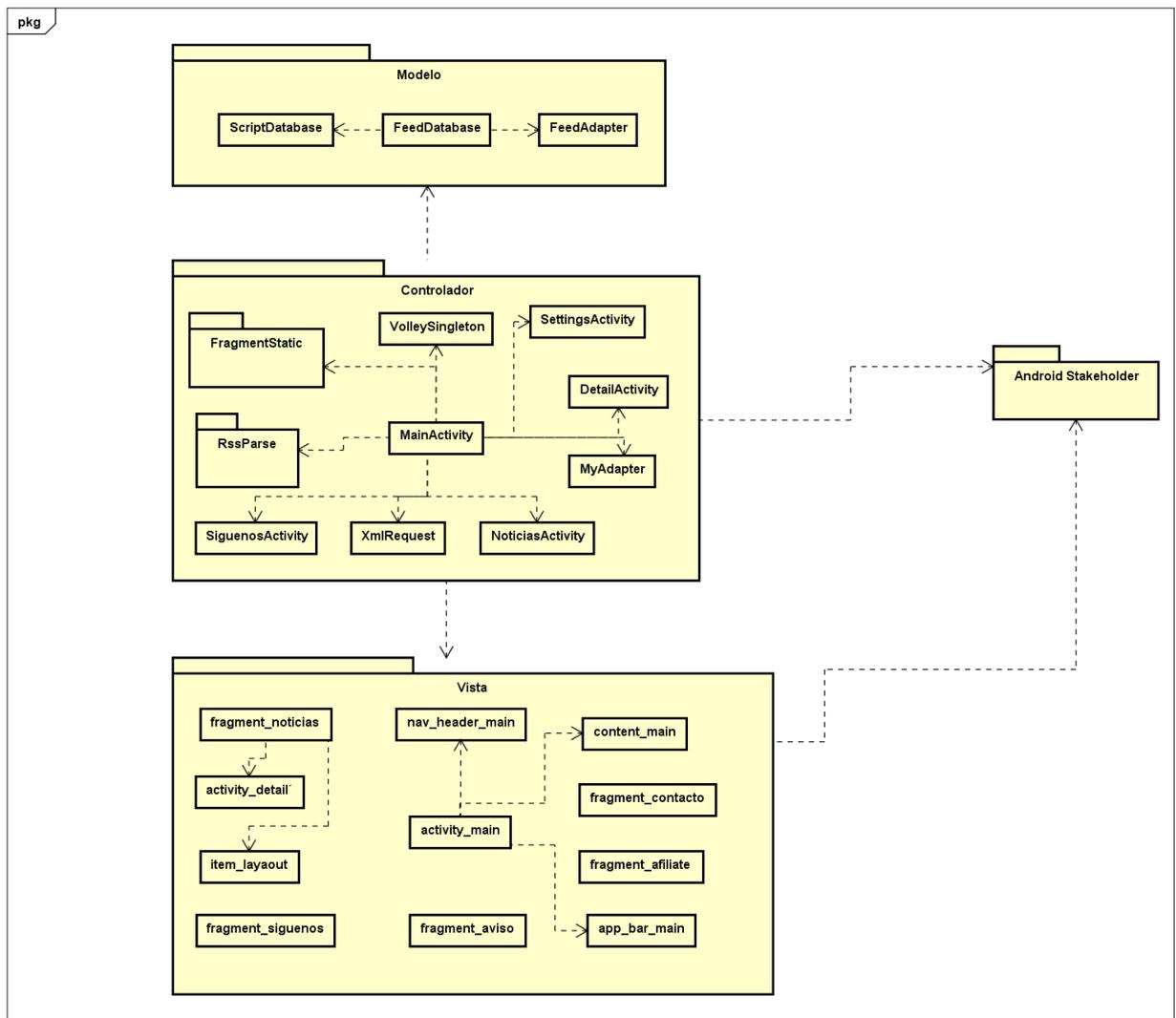


Ilustración 15. Modelo arquitectónico.

6.2. Modelo relacional.

El modelo relacional es muy sencillo y elegante: cada base de datos es un conjunto de relaciones, cada una de las cuales es una tabla con filas y columnas. Esta representación tabular tan sencilla hace que se pueda comprender el contenido de las bases de datos y permite el empleo de lenguajes sencillos de alto nivel para consultar los datos.

El modelo relacional de este proyecto es el representado a continuación:

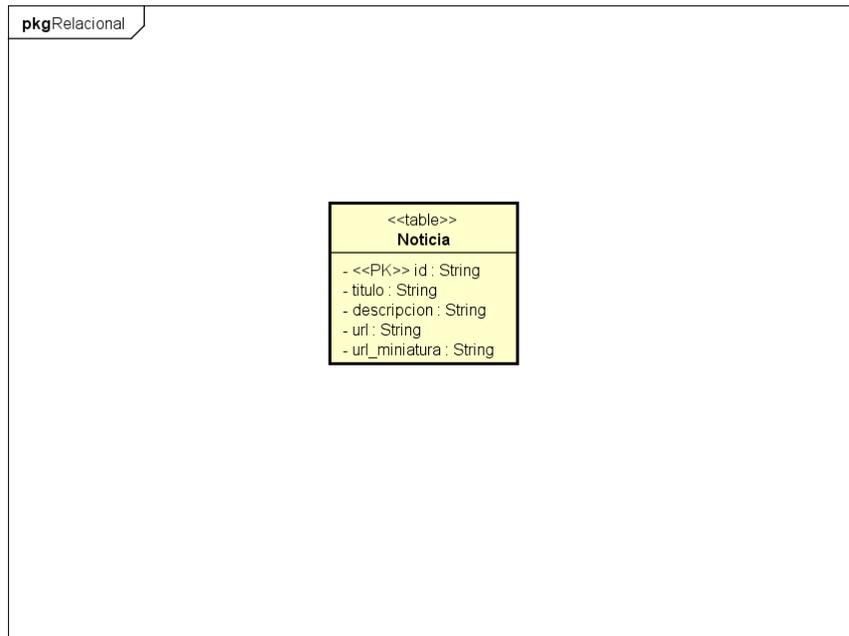


Ilustración 16. Tabla de la base de datos.

6.3. Patrones usados en el diseño.

Un patrón de diseño es una forma reutilizable de resolver un problema común, es decir, son un conjunto de soluciones a problemas comunes en el diseño de una aplicación.

6.3.1. Patrón singleton.

El patrón singleton es un patrón de diseño de software que restringe la creación de instancias de una clase a un objeto. Esto es útil cuando se necesita exactamente un objeto para coordinar acciones en todo el sistema.

El concepto algunas veces se generaliza a sistemas que operan más eficientemente cuando solo existe un objeto, o que restringen la creación de instancias a un cierto número de objetos.

Intención. Asegurarse de que una clase tiene una sola instancia y proporcionar un punto de acceso global a ella.

Motivación. Algunas clases sólo pueden tener una instancia. Una variable global no garantiza que sólo se instancie una vez.

Aplicabilidad. Se utiliza cuando deba haber exactamente una instancia de una clase y deba ser accesible a los clientes desde un punto de acceso conocido.

Consecuencias.

- Acceso controlado a un ejemplar único: controlar cómo y cuando los clientes pueden acceder al ejemplar.
- Espacio de nombres reducido.
- Evita la necesidad de utilizar variables globales.
- Permite refinar las operaciones y la representación.
- Se puede heredar de la clase Singleton para configurar el ejemplar para una aplicación concreta.

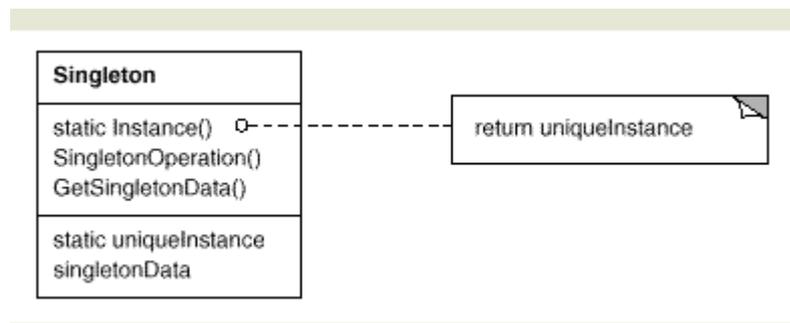


Ilustración 17. Patrón singleton.

Este patrón le podemos encontrar en nuestro proyecto en el connection pool y en la clase VolleySingleton creada para contener las repeticiones realizadas al http de la librería Volley.

6.3.2. Iterator.

Este patrón de diseño nos resultará útil para acceder a los elementos de un array o colección de objetos contenida en otro objeto, es decir, permite acceder secuencialmente a una colección de objetos sin exponer su representación interna.

En este proyecto se utiliza para recorrer la lista de noticias, cuando no sea necesario conocer la estructura interna de dicha clase, o cuando sea necesario diferentes tipos de categorías.

Intención. Proporcionar una forma de acceder secuencialmente a los elementos de un agregado sin revelar su representación interna.

Motivación. Iterar sobre los elementos de una lista sin ampliar la interfaz de la lista y permitiendo recorridos simultáneos.

Aplicabilidad. Se utiliza cuando:

- Acceder al contenido de un objeto agregado sin revelar su representación interna.
- Permitir múltiples tipos de recorridos sobre objetos agregados.
- Proporcionar una interfaz uniforme para recorrer diferentes estructuras de agregación (iteración polimórfica).

Consecuencias.

- Permite variaciones en el recorrido del agregado encapsulando los algoritmos de recorrido en distintas subclases de iteradores.
- Simplifica la interfaz del agregado al no incluir las operaciones de iteración.
- Permite recorridos simultáneos con varios iteradores a la vez.

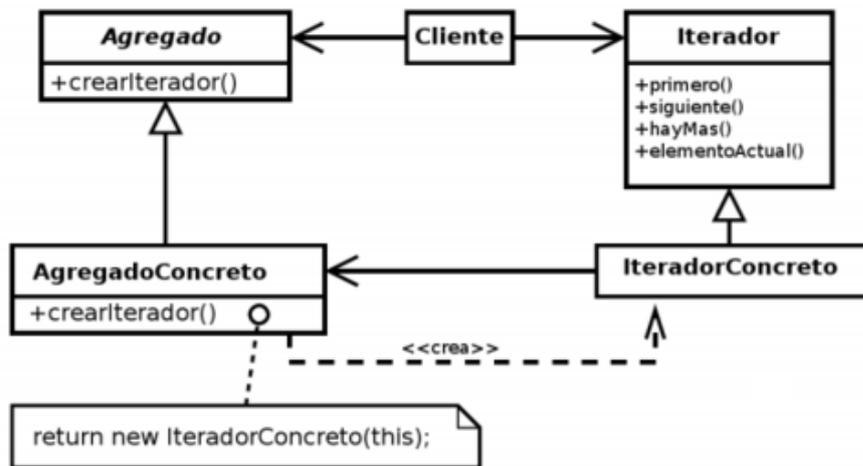


Ilustración 18. Patrón iterador.

Capítulo 7: Implementación.

7.1. Herramientas utilizadas.

Las herramientas necesarias para la implementación del proyecto son las siguientes:

- **Android Studio:** Entorno de desarrollo integrado, basado en IntelliJ IDEA, para el desarrollo de aplicaciones Android.
- **Astah Professional:** Herramienta para el modelado de diagramas UML.
- **Java:** Lenguaje de programación multiplataforma orientado a objetos. Usado en su versión 8.
- **SQLite:** Sistema de gestión de bases de datos relacional que se enlaza con el programa pasando a ser parte integral del mismo, ideal para bases de datos en dispositivos móviles.
- **Proto.io:** Herramienta para el diseño y maquetado de aplicaciones móviles.
- **Dropbox:** Lugar de almacenamiento durante el desarrollo del proyecto en la nube.

7.2. Entorno de desarrollo.

El entorno de desarrollo de este TFG han sido los siguientes:

Hardware	Xiaomi Air 13 Notebook
Procesador	Intel(R) Core(TM) i5-7200U CPU @ 2.50Hz 2.70Hz
Cache	de 3M hasta 3.1 GHz
Memoria	8 GB DDR4
Almacenamiento	256 GB SSD (Samsung PM961 M.2 NVMe)
Tarjeta Gráfica	Intel HD Graphics 620 y NVIDIA GeForce MX150
Pantalla	13,3" IPS FHD (1.920 x 1.080, 166 ppp)

Tabla 31. Especificaciones de Hardware.

Software	
OS	Windows 10 Pro
Resolución de pantalla	1.920 x 1.080
Sistema de Ficheros	NTFS

Tabla 32. Especificaciones de Software.

7.3. Pruebas.

Para comprobar el funcionamiento correcto de la aplicación se van a realizar las pruebas necesarias, teniendo en cuenta los diferentes puntos de vista.

- Las **pruebas de caja blanca** se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. El ingeniero de pruebas escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados.
- Las **pruebas de caja negra** es aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra nos interesará su forma de interactuar con el medio que le rodea entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace. Por tanto, de una caja negra deben estar muy bien definidas sus entradas y salidas, es decir, su interfaz; en cambio, no se precisa definir ni conocer los detalles internos de su funcionamiento.

7.3.1. Pruebas de caja blanca.

Estas pruebas también llamadas estructurales o de caja transparente, se han ido realizando, a medida que se iba desarrollando el código, de forma que cada método y la colaboración entre otras han sido probadas.

Además, al haber tantos métodos no sería óptimo poner las pruebas en de caja blanca en este documento.

7.3.2. Pruebas de caja negra.

Este tipo de pruebas está enfocado a comprobar que los requisitos funcionales se han cumplido. No son una alternativa a las pruebas de caja blanca, sino que son una técnica complementaria que se ha de usar por separado para detectar la mayor cantidad de fallos posibles y así poder corregirlos para un buen desarrollo de la aplicación.

Caso de prueba	Casos cubiertos	Salida
El usuario introduce palabras clave.	Introduce entre 3 y 20 caracteres.	OK
	Introduce menos de 3 caracteres.	Mensaje de aviso
	Introduce caracteres aleatorios	OK
	No se introduce caracteres.	OK

Tabla 33. Batería de pruebas: Búsqueda de una noticia.

Caso de prueba	Casos cubiertos	Salida
El usuario pulsa el botón de una red social.	El botón se encuentra dentro de la pantalla.	OK
	Se pulsa fuera de los botones.	----

Tabla 34. Batería de pruebas: Seguimiento en redes sociales.

Caso de prueba	Casos cubiertos	Salida
El usuario pulsa el botón de las notificaciones	El botón se encuentra dentro de la pantalla.	OK
	Se pulsa fuera del botón.	----

Tabla 35. Batería de pruebas: Activar/Desactivar notificaciones.

El usuario en el resto de casos de uso no debe introducir datos a la aplicación, tan solo se mueve entre las pantallas disponibles obteniendo información. No hay casos en las pruebas de caja negra en la cual el usuario deba rellenar algún formulario y la aplicación deba controlar la entrada de datos, por lo cual si hay algún error lo veremos con las pruebas de caja blanca controlando el flujo de la aplicación.

Capítulo 8: Conclusiones.

8.1. Objetivos alcanzados.

Los siguientes objetivos que se van a nombrar, son los objetivos que se han alcanzado a lo largo de toda la realización del TFG.

- Se ha realizado un estudio para la realización de proyectos Android, además de cómo desarrollar este en el entorno de desarrollo Android Studio usando el SDK de Android.
- Se ha conseguido una interfaz bastante intuitiva y atractiva.
- Se ha realizado el parseo de datos en XML y mostrado en pantalla de forma sencilla y atractiva (Uso de RSS).
- Se ha utilizado dos librerías externas al entorno de desarrollo y de manera gratuita.
- Se permite la visualización de las distintas noticias con información relativa a la una de las categorías planteadas.
- Se permite la visualización de distintos contenidos de más como, por ejemplo, el aviso legal o los contactos, etc.
- Se permite realizar búsquedas por palabras clave y así encontrar una noticia en concreto.
- No se ha necesitado guardar ningún dato personal, consiguiendo la privacidad total de los posibles usuarios.

8.2. Conocimientos adquiridos.

La realización de este TFG me ha permitido adquirir nuevos e interesantes conocimientos al tiempo que iba afianzando los que ya había obtenido durante mi formación como Ingeniero Informático.

En las distintas etapas del desarrollo de un proyecto de software puede que la tecnología inicialmente pensada a usar no sea la esperada o esta no se adapte a las necesidades del proyecto. La capacidad para adaptarme a estos aspectos ha resultado clave para superar estos escollos y llevar el proyecto por el camino adecuado.

Las aplicaciones móviles, en mi opinión, son el presente y el futuro de los sistemas informáticos y están en pleno auge. Podemos encontrar un SO Android en muchos tipos de dispositivos como relojes, móviles o coches. Todo esto unido a que es un lenguaje de programación libre y a su facilidad para comerciar una app en su plataforma PlayStore, fueron factores clave para despertar la idea de realizar el TFG en Android.

En los últimos tiempos se ha convertido en algo muy habitual la programación no nativa usando plataformas de desarrollo que te permiten desplegar un

programa en varios tipos de sistemas. Este tipo de programación favorece la reutilización del código, repercutiendo en un mayor ahorro de coste y un menor tiempo de desarrollo de una app, ya que con un único código podemos lanzarla a varias plataformas. En mi opinión, y sin quitarle importancia a lo anterior dicho, creo que para el desarrollo de aplicaciones de calidad sigue siendo muy importante conocer el lenguaje nativo y programar en este, por eso el TFG se ha desarrollado usando el entorno de desarrollo de Android Studio y el SDK de Android.

8.3. Trabajo futuro.

- Sólo se ha podido realizar en el sistema RSS la categoría de Enseñanza y Formación, por lo que para trabajo futuro habría que implementar las demás categorías.
- Añadir más categorías si el cliente lo demanda.
- Mejorar la interfaz más acorde a posibles nuevas especificaciones por parte del cliente.
- Implementar un buscador propio, cuando se tengan todas las categorías en modo RSS.
- Añadir ajustes más concretos, dependiendo de lo demandado.

Bibliografía.

Sitios Web.

[1] Developer Android: <https://developer.android.com/>

Descripción: Página oficial para desarrolladores de Android

[Última visita: 1-06-2018]

[2] RSS Explicada: <http://www.rss.nom.es/>

Descripción: Página dedicada al RSS

[Última visita: 1-06-2018]

[3] Wikipedia: <https://es.wikipedia.org/wiki/Wikipedia:Portada>

Descripción: Enciclopedia interactiva que se utilizó para ver conceptos y definiciones en las distintas etapas del proyecto.

[Última visita: 1-06-2018]

[4] XMLPullParser: <http://www.xmlpull.org/>

Descripción: Página dedicada al parseo de XML

[Última visita: 1-06-2018]

[5] Android: <https://www.android.com/>

Descripción: Página oficial de Android.

[Última visita: 1-06-2018]

[6] GitHub: <https://github.com/>

Descripción: Página oficial de GitHub.

[Última visita: 1-06-2018]

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

[7] Curso de Programación Android: <http://stackoverflow.com/>

Descripción: Sitio web para la comunidad de desarrolladores informáticos en el cual se ha encontrado soluciones a problemas de implementación.

[Última visita: 1-06-2018]

[7] Blog de Android: <http://www.sgoliver.net/>

Descripción: Sitio web de Salvador Gómez donde plantea y resuelve conceptos de Android.

[Última visita: 1-06-2018]

Apéndice 1: Manual del usuario.

Instalación de Fesp_CyL.

- 1- Ir a la tienda de aplicaciones (PlayStore) del dispositivo Android.
- 2- Introduzca Fesp CyL.
- 3- Presione instalar.
- 4- Acepte los permisos solicitados por la aplicación para un funcionamiento correcto.

Primera ejecución.

La primera vez que se ejecute la aplicación, nos vamos a encontrar con la siguiente pantalla.



Ilustración 19. Pantalla de inicio.

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

Arriba a la izquierda se puede visualizar tres rayas horizontales. Al pulsar encima de ellas, se va a desplegar el menú donde podemos encontrar las diferentes categorías.

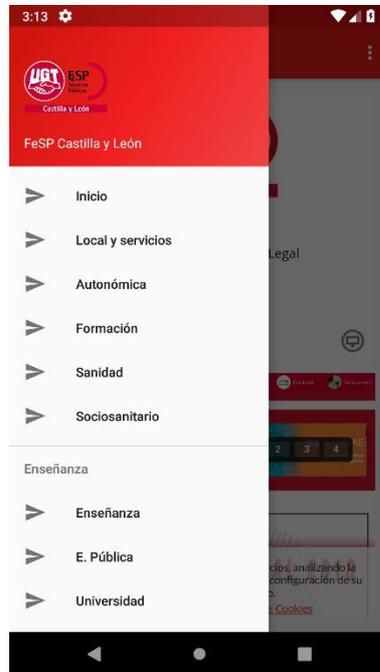


Ilustración 20. Pantalla menú de las categorías.

Y también las opciones extras que son informativas.

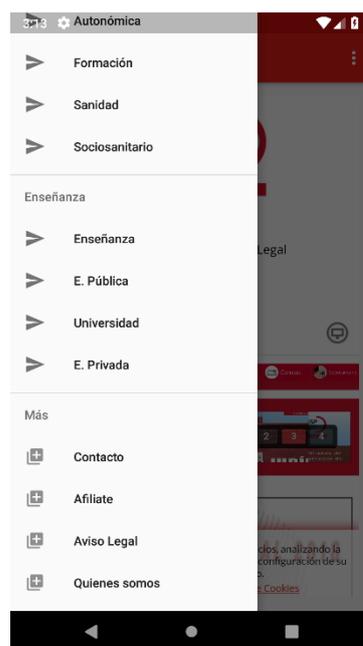


Ilustración 21. Pantalla con menú de las opciones extras.

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

Dentro de las opciones extras encontramos el Contacto, donde se puede visualizar información para contactar con cada sede de FeSP de UGT por provincia.



Ilustración 22. Pantalla de contactos de FeSP.

En la opción de afíliate, se puede ver el documento de inscripción del cliente. Accediendo a tu cuenta de Google, se puede guardar en la nube Drive.

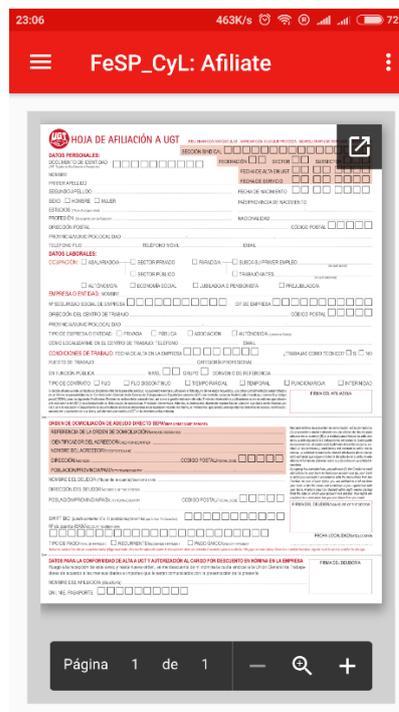


Ilustración 23. Pantalla con el documento de afiliación.

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

Y por último el apéndice legal del cliente y quiénes son y su labor.

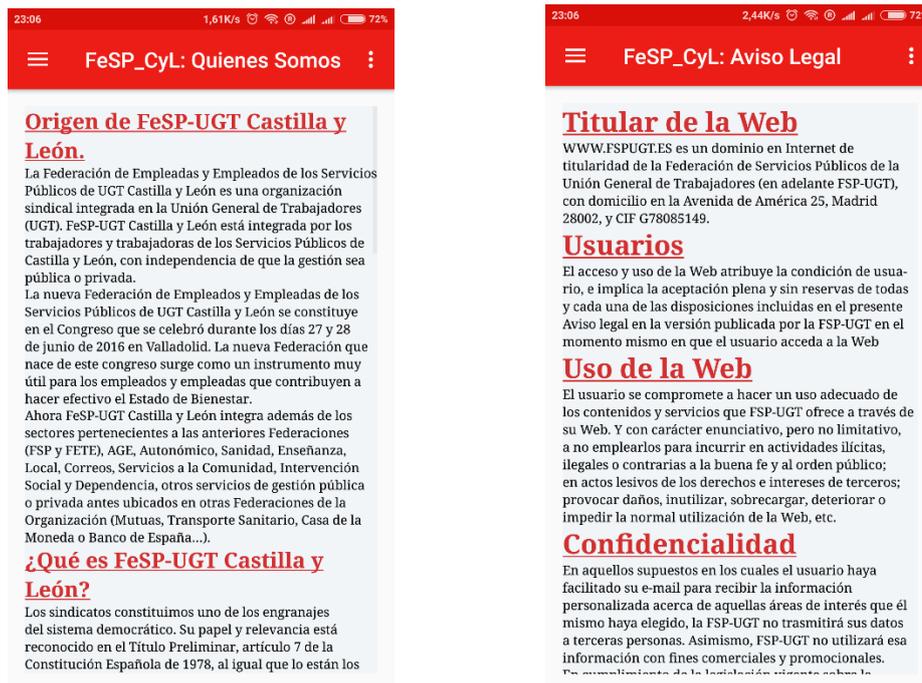


Ilustración 24. Pantallas de Quienes Somos y Aviso Legal.

Al otro lado de la pantalla de inicio se ve tres puntos en vertical, ese es el menú secundario donde se encuentran tres opciones:

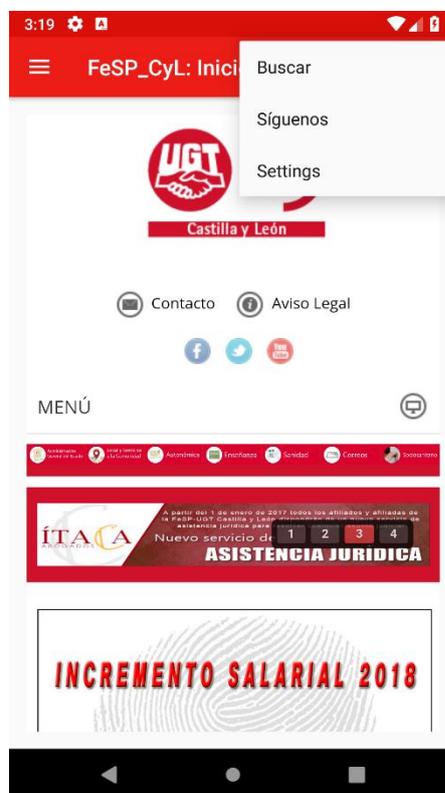


Ilustración 25. Pantalla del menú secundario.

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

En la opción de buscar se puede realizar una búsqueda por palabras clave y se te mostrará el buscador de información del cliente. Pulsando el botón de “Buscar” te redirecciona a la página de búsqueda del cliente.

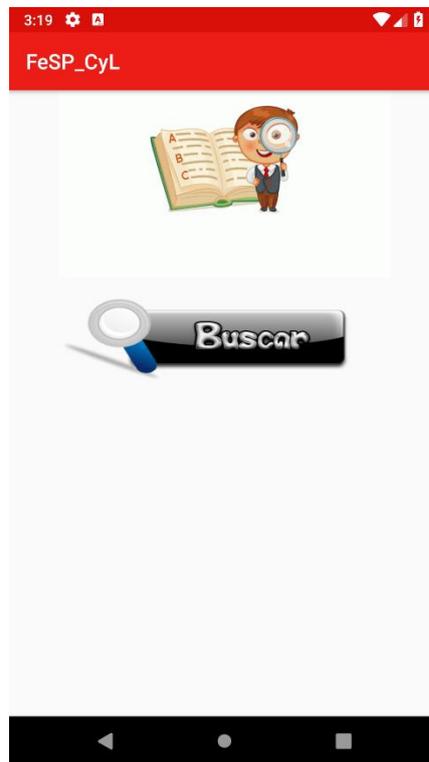


Ilustración 26. Pantalla de búsqueda.



Ilustración 27. Pantalla del buscador del cliente.

FeSPUGT: diseño e implementación de una aplicación móvil en Android.

La siguiente opción es el acceso a las redes sociales del cliente. Se encontrarán 3 botones que te redireccionan a la red social elegida, pudiendo realizar el seguimiento de esta.



Ilustración 28. Pantalla de las redes sociales.

Y por última opción es la posibilidad de recibir notificaciones de las actualizaciones de las distintas categorías.

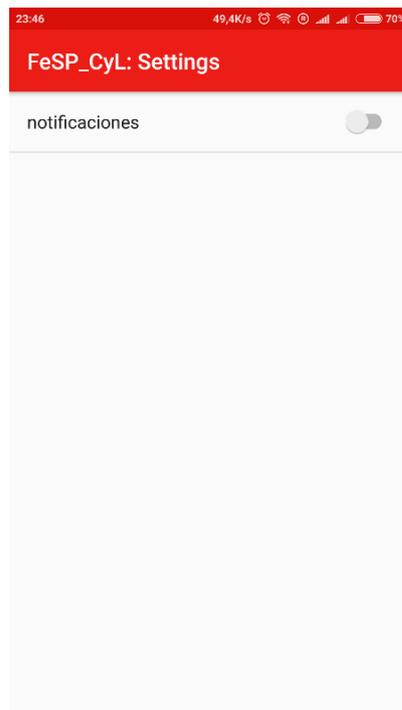


Ilustración 29. Pantalla de las notificaciones.

Y la parte más significativa de la aplicación, es la visualización de las noticias por categoría. Las categorías en las que el cliente todavía no ha implementado el feed RSS, se puede ver la página de la categoría de la página web.

Sin embargo, en las categorías de “Formación”, “Universidad”, “E. Pública” y “E. Privada” se ve las noticias en una lista.

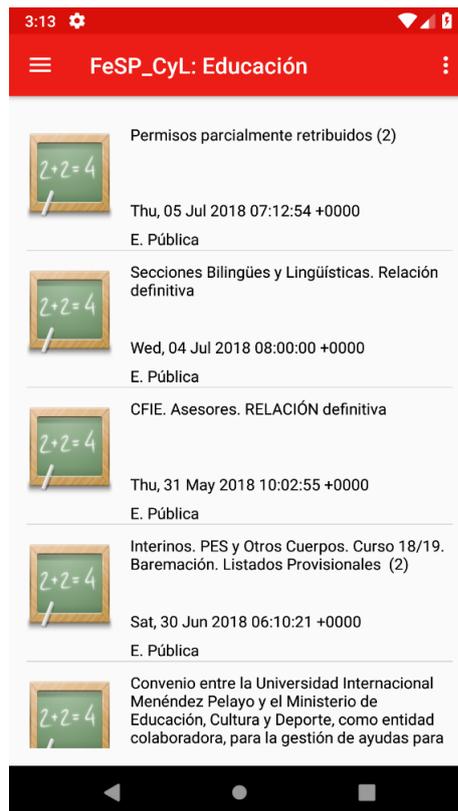


Ilustración 30. Pantalla de la categoría E. Pública.

Y cuando se pincha en cualquiera de una de las noticias de la lista, se puede visualizar el contenido de ella.

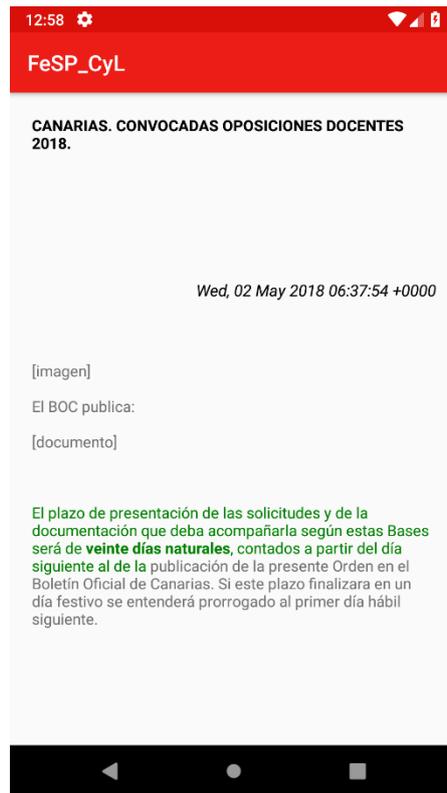


Ilustración 31. Pantalla de una noticia detallada.

Apéndice 2: Glosario.

- TFG: Trabajo de fin de grado.
- MVC: modelo-vista-controlador.
- RSS: Really Simple Syndication.
- HTTP: Hypertext Transfer Protocol.
- XML-RPC: Es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.
- XML: Extensible Markup Language.
- DOM: Document Object Model.