

SURVY. APLICACIÓN EDUCATIVA DE ENCUESTAS Y PREGUNTAS ONLINE

Trabajo de Fin de Grado

Grado en Ingeniería Informática



**VNiVERSiDAD
D SALAMANCA**

Septiembre de 2022

Autor

Daniel Velerdas Sedano

Tutor/a

ANDRÉ FILIPE SALES MENDES

GABRIEL VILLARRUBIA GONZÁLEZ

JUAN FRANCISCO DE PAZ SANTANA

CERTIFICADO DE LOS TUTORES

D. ANDRÉ FILIPE SALES MENDES, D. GABRIEL VILLARRUBIA GONZÁLEZ y
D. JUAN FRANCISCO DE PAZ SANTANA, profesores/as del Departamento de
INFORMÁTICA Y AUTOMÁTICA de la Universidad de Salamanca

CERTIFICAN:

Que el trabajo titulado “SURVY. APLICACIÓN EDUCATIVA DE ENCUESTAS Y
PREGUNTAS ONLINE” ha sido realizado por D. DANIEL VELERDAS SEDANO, con
DNI 45690728D y constituye la memoria del trabajo realizado para la superación de la
asignatura Trabajo de Fin de Grado de la Titulación INGENIERÍA INFORMÁTICA de
esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a _____ de _____ de 20__

D ANDRÉ FILIPE SALES
MENDES
Dpto. INFORMÁTICA Y
AUTOMÁTICA
Universidad de Salamanca

D GABRIEL
VILLARRUBIA
GONZÁLEZ
Dpto. INFORMÁTICA Y
AUTOMÁTICA
Universidad de Salamanca

JUAN FRANCISCO DE
PAZ SANTANA
Dpto. INFORMÁTICA Y
AUTOMÁTICA
Universidad de Salamanca

RESUMEN

Este Trabajo de Fin de Grado detalla el proceso completo de desarrollo de una aplicación móvil destinada a la creación, resolución y evaluación de encuestas y preguntas online en el ámbito educativo.

Orientado al entorno del aula, *Survvy* es concebida como una herramienta que permita a los profesores generar diferentes encuestas para las asignaturas que imparten, de manera que estas sean resueltas por sus alumnos, con el fin de obtener una visión general del nivel de conocimiento adquirido por los alumnos en determinadas materias, facilitando así a los profesores la evaluación del seguimiento de las clases.

La aplicación ha sido desarrollada en el lenguaje de programación Kotlin, siguiendo un patrón Modelo-Vista-Controlador (o MVC), utilizando el SDK ofrecido por Android Studio. Para el almacenamiento de los datos de la aplicación y la autenticación de los usuarios en la misma, se ha hecho uso de diferentes librerías de Firebase.

A lo largo de todo el desarrollo de la aplicación, se han empleado técnicas de ingeniería del software siguiendo el Proceso Unificado. Todas estas técnicas se encuentran reflejadas en la memoria de este trabajo y son detalladas en cada uno de los seis anexos que la acompañan.

Palabras clave: aplicación, Android, educación, Firebase, encuestas, Kotlin, Proceso Unificado.

SUMMARY

This Final Degree Project details the complete process of developing a mobile application for the creation, resolution and evaluation of online surveys and questions in the educational field.

Oriented to the classroom environment, Survy is conceived as a tool that allows teachers to generate different surveys for the subjects they teach so that they can be answered by their students, in order to obtain an overview of the level of knowledge acquired by the students in certain subjects, making it easier for teachers to evaluate the follow-up of the classes.

The application has been developed in the Kotlin programming language, following Model-View-Controller (or MVC) pattern, using the SDK offered by Android Studio. For the storage of the application data and the authentication of the users in it, different Firebase libraries have been used.

Throughout the development of the application, software engineering techniques have been used following the Unified Process. All these techniques are reflected in the memory of this work and are detailed in each of the six annexes that come along with it.

Keywords: application, Android, education, Firebase, surveys, Kotlin, Unified Process.

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN	1
2. OBJETIVOS	3
2.1 Objetivos del sistema	3
2.2 Objetivos personales	4
3. CONCEPTOS TEÓRICOS	5
3.1 Desarrollo móvil	5
3.2 Android	5
3.3 SDK (Software Development Kit)	5
3.4 Librerías o bibliotecas	5
3.5 Base de datos NoSQL	6
3.6 CRUD.....	6
4. TÉCNICAS Y HERRAMIENTAS UTILIZADAS	7
4.1 Entorno	7
4.1.1 Android SDK.....	7
4.2 Librerías	7
4.2.1 Firebase	7
4.2.2 Zxing.....	8
4.2.3 CircleImageView	8
4.2.4 Picasso.....	8
4.2.5 Dokka	8
4.3 Lenguajes	9
4.3.1 Kotlin	9
4.3.2 XML.....	9
4.4 Herramienta de control de versiones	9
4.4.1 Github y Github Desktop	9
4.5 Herramientas CASE.....	9
4.5.1 Visual Paradigm y Visual Paradigm Online.....	9
4.5.2 Diagrams.net.....	9
4.5.3 EZEstimate	9
4.5.4 Microsoft Project	10
4.6 Herramientas de diseño gráfico	10
4.6.1 Looka	10
4.6.2 FontAwesome	10
4.6.3 IconFinder	10
5. ASPECTOS RELEVANTES DEL TRABAJO	11

5.1 Marco de trabajo	11
5.2 Estimación de coste y esfuerzo	12
5.2.1 UUCP	12
5.2.2 TCF y ECF	15
5.2.3 Resultados	16
5.3 Planificación temporal	16
5.3.1 Inicio	17
5.3.2 Elaboración.....	17
5.3.3 Construcción.....	17
5.3.4 Transición.....	17
5.4 Especificación de requisitos	18
5.4.1 Objetivos del sistema	18
5.4.2 Requisitos de información.....	19
5.4.3 Requisitos de restricción de información	19
5.4.4 Definición de actores	19
5.4.5 Casos de uso del sistema	20
5.4.6 Requisitos no funcionales.....	22
5.5 Análisis del sistema.....	23
5.5.1 Modelo del dominio	23
5.5.2 Paquetes de análisis.....	24
5.5.3 Realización de casos de uso	25
5.6 Diseño del sistema	26
5.6.1 Modelo de diseño.....	26
5.6.2 Subsistemas de diseño.....	27
5.6.3 Clases de diseño.....	27
5.6.4 Realización de casos de uso	31
5.6.5 Modelo de despliegue	32
6. IMPLEMENTACIÓN.....	33
6.1 Configuración de la base de datos.....	33
6.2 Inclusión de librerías	36
6.3 Desarrollo de la aplicación	37
7. CONCLUSIONES.....	48
8. LÍNEAS DE TRABAJO FUTURAS.....	48
9. BIBLIOGRAFÍA.....	50

ÍNDICE DE TABLAS

Tabla 1. Complejidad de los actores del sistema.....	13
Tabla 2. Complejidad de los casos de uso del sistema.....	14
Tabla 3. OBJ-001. Gestión de Usuarios.....	18
Tabla 4. ACT-001. Actor Usuario	20
Tabla 5. UC-016. Nueva Encuesta.....	22

ÍNDICE DE FIGURAS

Figura 1. Resultados estimación de coste y esfuerzo EZEestimate.....	16
Figura 2. Diagrama de Gantt - Inicio, iteración 1	17
Figura 3. Diagrama de Actores.....	19
Figura 4. Diagrama de Casos de Uso – Gestión de Encuestas y Preguntas	21
Figura 5. Diagrama de Clases.....	23
Figura 6. Diagrama de Paquetes de Análisis	24
Figura 7. Diagrama de Paquetes de Análisis – Gestión de Usuarios.....	24
Figura 8. Diagrama de Secuencia UC-016. Nueva Encuesta	25
Figura 9. Patrón Modelo-Vista-Controlador	26
Figura 10. Patrón Adaptador	26
Figura 11. Diagrama de Subsistemas.....	27
Figura 12. Diagrama Modelo del subsistema de la Aplicación Móvil	28
Figura 13. Diagrama Vista del subsistema de la Aplicación Móvil	29
Figura 14. Diagrama Controlador del subsistema de la Aplicación Móvil.....	30
Figura 15. Diagrama de Secuencia de diseño UC-016. Nueva Encuesta.....	31
Figura 16. Diagrama de Despliegue.....	32
Figura 17. Diagrama de la base de datos del sistema	33
Figura 18. Base de datos de Cloud Firestore desde la consola de Firebase	34
Figura 20. Implementación de las librerías en el fichero build.gradle del módulo app..	36
Figura 20. Formularios de inicio de sesión y registro.....	38
Figura 21. Menús desplegables del menú principal de alumno y profesor	39
Figura 22. Mis Alumnos	40
Figura 23. Vista de detalle de asignatura para matricular un alumno.....	41
Figura 24. Vistas de listado y detallada de asignaturas del alumno	42
Figura 25. Vista de listado de encuestas del alumno	43
Figura 26. Resolver encuesta.....	44

Figura 27. Mis Resultados Alumno	45
Figura 28. Ver resultados de una encuesta Profesor	46
Figura 29. Mi perfil Alumno y Profesor	47

1. INTRODUCCIÓN

La pandemia provocada durante el año 2020 por el COVID-19 provocó una inevitable transición dentro de muchos campos, como el empleo, la investigación o la educación hacia los medios digitales. Este cambio, en particular dentro del campo de la educación, dejó patente que ni las herramientas ni los conocimientos de los profesionales eran suficientes como para aprovechar plenamente esta tecnología.

Dos años después del comienzo de la pandemia y volviendo a ser la norma la educación totalmente presencial, podemos ver que ciertas herramientas informáticas empleadas de manera excepcional durante la pandemia han pasado a formar parte de la mayoría de los planes educativos. Podríamos destacar la grabación de las clases en forma de píldoras formativas, la extensión de la utilización de plataformas virtuales tipo Moodle, la normalización de tutorías en remoto a través de videollamadas, entre otros.

En este contexto, nace la aplicación desarrollada durante este Trabajo de Fin de Grado de Ingeniería Informática: *Survvy*, una aplicación educativa de encuestas y preguntas online. El objetivo principal de este proyecto consiste en ofrecer a los docentes una herramienta más que incorporar fácilmente en sus planes didácticos con el fin de poder obtener de una manera sencilla información acerca de la eficacia con la que los alumnos están adquiriendo los conocimientos impartidos en las aulas, así como darles también una herramienta alternativa de evaluación, simple, interactiva y basada en la gamificación.

En esta memoria, se detallará todo el proceso de desarrollo de la aplicación. Este documento estará compuesto por los siguientes apartados:

- **Introducción.** Consta del apartado en el que nos encontramos actualmente. En él, se aporta una contextualización del trabajo, además de una esquematización de la estructura de la memoria y los anexos.
- **Objetivos.** En este apartado describiremos brevemente los objetivos a cumplir por nuestro sistema, así como los objetivos personales que han servido como motivación a lo largo del mismo.
- **Conceptos teóricos.** En este punto, explicaremos algunos de los conceptos que serán mencionados en numerosas ocasiones a lo largo de la memoria, para facilitar su comprensión al lector.
- **Técnicas y herramientas utilizadas.** En este apartado detallaremos aquellas herramientas y técnicas que han sido utilizadas a lo largo del desarrollo del trabajo.
- **Aspectos relevantes del trabajo.** Este apartado servirá como resumen de aquellos apartados más importantes del desarrollo de nuestro sistema software, los cuales se encontrarán detallados más en profundidad a lo largo de los anexos.
- **Implementación.** En esta etapa, trataremos la configuración del entorno en el que se desarrollará nuestro sistema, así como detallaremos el proceso de desarrollo del mismo.
- **Conclusiones y líneas de trabajo futuro.** En estos puntos, detallaremos tanto las conclusiones derivadas de la realización de este trabajo como las posibles modificaciones y mejoras que se podrían realizar sobre nuestro sistema de cara a un futuro.
- **Bibliografía.** Finalmente, recogeremos las referencias y citas a las fuentes de las que nos hemos servido para el desarrollo de esta memoria.

Además de la memoria, este trabajo cuenta con la documentación técnica del desarrollo del proyecto, compuesta por los siguientes anexos:

- **Anexo I – Planificación temporal.** Este anexo abarca tanto la estimación de esfuerzo de nuestro desarrollo como su planificación temporal.
- **Anexo II – Especificación de requisitos.** Siguiendo la metodología de Durán y Bernárdez, realizaremos una recogida y especificación de los objetivos y requisitos del sistema.
- **Anexo III – Análisis de requisitos.** Partiendo del anexo anterior, realizaremos un análisis de todos los requisitos obtenidos, además de la realización de los casos de uso.
- **Anexo IV – Especificación de diseño.** Contiene la documentación del diseño de nuestro sistema, basándonos en los dos anexos anteriores, lo que sentará las bases de la posterior implementación del sistema.
- **Anexo V – Documentación técnica de programación.** Este anexo ofrecerá una visión general del código desarrollado durante la implementación del sistema, con el fin de facilitar a cualquier persona interesada la comprensión del mismo.
- **Anexo VI – Manual del usuario.** En este anexo se detallarán todas y cada una de las funcionalidades implementadas en nuestra aplicación, así como la navegación entre las mismas.

Además de estos documentos, se adjunta a este Trabajo de Fin de Grado diversos ficheros generados durante la realización del mismo, para su consulta por parte del lector.

2. OBJETIVOS

En este apartado, expondremos los objetivos principales que deberemos cumplir a lo largo del desarrollo del sistema, además de los objetivos personales que motivan este trabajo.

2.1 Objetivos del sistema

Todos los objetivos que mencionaremos a continuación parten del objetivo principal de diseñar y desarrollar una aplicación móvil orientada al mundo educativo, para facilitar el seguimiento por parte del profesor de los conocimientos del alumno a lo largo del desarrollo de una asignatura.

- **Gestión de Usuarios**

La aplicación debe permitir a cada usuario registrar una cuenta nueva única, que solamente sea accesible por el propio usuario mediante la introducción de sus credenciales. Este registro, creará un perfil para cada usuario del sistema, que el propio usuario podrá modificar en cualquier momento. Además, deberá contemplar cambios y recuperaciones de las credenciales del usuario, además de diferenciar a los dos tipos de usuarios que interactuarán con el sistema: alumnos y profesores.

- **Gestión de Asignaturas**

El sistema deberá permitir la gestión de asignaturas por parte de los usuarios, ofreciendo a los usuarios la posibilidad de crear, modificar y eliminar las mismas. También deberá gestionar cómo interactúan tanto profesores como alumnos sobre los datos relacionados con cada asignatura.

- **Gestión de Encuestas y Preguntas**

Dentro de cada asignatura, la aplicación debe ocuparse también de la gestión de encuestas y preguntas, siguiendo otra vez el modelo CRUD. Igual que en el caso anterior, habrá que gestionar la interacción de alumnos y profesores con las diferentes encuestas y preguntas almacenadas en nuestra base de datos.

- **Gestión de Resultados**

Una vez cubiertos los otros objetivos, la aplicación debe ser capaz de generar diferentes resultados para las diferentes encuestas y preguntas, cuando estas sean resueltas por los alumnos. En este caso, los resultados se tratarán de una forma especial, dado que serán generados automáticamente cuando un alumno resuelva una encuesta y solamente podrá ser eliminado por un profesor, pero no podrá editarlo.

- **Usabilidad**

Otro objetivo importante, es garantizar la facilidad de uso de la aplicación, con el objetivo de que esta se convierta en una herramienta de uso habitual en el entorno del aula.

- **Responsividad**

Debido a que existen una gran variedad de dispositivos Android, el diseño de la aplicación debe contar con soporte para el mayor tipo de dispositivos posibles, para que la aplicación sea plenamente funcional en cualquiera de ellos.

- **Mantenibilidad**

El sistema debe ser capaz de actualizarse y adaptarse a los cambios y versiones posteriores del sistema operativo Android.

- **Concurrencia**

El diseño del sistema contempla la posibilidad de que ocurran varias interacciones simultáneas con la aplicación y el sistema por parte de diferentes usuarios, por lo que habrá que comprobar que este comportamiento no genere fallos ni problemas.

2.2 Objetivos personales

En cuanto a mis objetivos personales, me propuse como meta aprender una tecnología que en la que no hemos llegado a profundizar a lo largo del Grado y que cada día está más presente en nuestras vidas: las aplicaciones móviles.

Mi objetivo principal ha sido obtener los conocimientos necesarios para desarrollar una aplicación plenamente funcional, que perfectamente pudiera ser publicada, comercializada y utilizada como una herramienta de apoyo a la educación.

Además, el hecho de desarrollar una aplicación de este tipo me servirá como introducción al campo del desarrollo móvil, una rama de la informática que ahora mismo se encuentra en auge en el mercado laboral.

3. CONCEPTOS TEÓRICOS

3.1 Desarrollo móvil

El desarrollo móvil es la disciplina dentro del mundo de la programación y el desarrollo que agrupa los diferentes procesos y procedimientos que se llevan a cabo con el objetivo de desarrollar software para dispositivos móviles. Dentro de este grupo, podemos diferenciar entre múltiples dispositivos, como teléfonos inteligentes, tablets, relojes inteligentes, televisiones inteligentes y muchos otros.

Dentro de este tipo de desarrollo de software, podemos diferenciar entre dos grandes sistemas operativos predominantes: Android e iOS.

3.2 Android

Actualmente, el sistema operativo móvil de mayor popularidad llegando a representar el 84% de los dispositivos móviles vendidos en 2020 (Mena Roa, 2021), basado en el núcleo de Linux y hoy en día, propiedad de la empresa tecnológica Google.

Su diferencia principal con el otro sistema operativo móvil por excelencia, iOS, es que Android puede ser ejecutado en una infinidad de dispositivos, de diferentes fabricantes. En el caso de su competidor, el sistema operativo iOS solamente puede ser ejecutado y utilizado en dispositivos móviles fabricados por la compañía Apple.

A día de hoy, Android puede utilizarse no solamente en móviles, sino también en relojes inteligentes, tabletas, televisores o incluso en automóviles.

3.3 SDK (Software Development Kit)

Un SDK o Kit de Desarrollo Software es un conjunto de herramientas creadas para el desarrollo de hardware y/o software.

Permiten a los desarrolladores agrupar en un mismo kit, las diferentes herramientas para crear productos software para una plataforma concreta. Entre estas herramientas, podemos encontrar software como editores, bibliotecas, compilador, depurador y diferentes APIs, entre otros (Red Hat, 2020).

En este trabajo, emplearemos el Android SDK para el desarrollo de nuestra aplicación.

3.4 Librerías o bibliotecas

Una librería o biblioteca es un conjunto de recursos, escritos en un lenguaje concreto de programación que ofrecen diferentes funcionalidades al desarrollo software (Wikipedia, 2021a).

Su principal diferencia con los programas ejecutables es que una librería no está pensada para ser utilizada de manera independiente, es decir, sus funcionalidades siempre tendrán que ser invocadas desde otro programa para ser utilizadas.

Muchas de estas bibliotecas, son generadas bajo la licencia de código abierto, de manera que cualquier usuario puede utilizarlas de manera gratuita, reduciendo en gran medida la carga del desarrollo software.

3.5 Base de datos NoSQL

Una base de datos NoSQL (Not Only SQL), constituye una estructura de datos destinada al almacenamiento de información que no sigue el modelo Entidad-Relación de las bases de datos (Amazon Web Services, s. f.).

Las principales ventajas de estas bases de datos con respecto a las bases de datos tradicionales son la facilidad en el desarrollo e implementación, su alto rendimiento y su alta funcionalidad.

Debido a que nuestro proyecto no necesitará una estructura compleja de base de datos, hemos elegido utilizar una base de datos NoSQL, concretamente, Firebase.

3.6 CRUD

Acrónimo utilizado para nombrar a las operaciones básicas a realizar sobre los datos almacenados en una base de datos: crear, leer, actualizar y eliminar (Create Read Update Delete, en inglés).

4. TÉCNICAS Y HERRAMIENTAS UTILIZADAS

4.1 Entorno

4.1.1 Android SDK

Android Studio constituye el entorno de desarrollo oficial para Android, creado por la empresa Google, propietaria del sistema operativo móvil. A través de este IDE, podremos tener acceso a las diferentes herramientas que conforman el Android SDK, entre ellas el editor de texto, el compilador, así como diferentes emuladores basados en QEMU de una amplia variedad de dispositivos capaces de correr el sistema operativo Android (Wikipedia, 2022).

Android Studio puede ser descargado en sistemas operativos Windows a partir de Windows XP, MAC OS X 10.5.8 o posterior, y GNU/Linux. Esto permite que el desarrollo de aplicaciones Android sea mucho más accesible que en el caso de iOS, cuyo SDK oficial solamente puede ser ejecutado desde equipos fabricados por la propia Apple.

4.2 Librerías

4.2.1 Firebase

Firebase es una plataforma en la nube oficial de Google que ofrece diferentes herramientas con el objetivo de facilitar el desarrollo web y móvil (Digital55, 2020). Estas herramientas, se dividen en cuatro grandes grupos:

- **Desarrollo.** En este grupo, se incluyen todos aquellos servicios de Firebase centrados en la etapa de desarrollo de la aplicación móvil o web. Entre estas herramientas encontraremos desde bases de datos en tiempo real, almacenamiento en la nube, reportes de crash, entre otros.
- **Crecimiento.** Las herramientas de esta etapa están centradas en el incremento del número de usuarios de la aplicación, así como de la gestión de los usuarios actuales. Algunas de estas herramientas son gestores de notificaciones, de publicidad o de invitaciones, entre otros.
- **Monetización.** Este grupo conforma aquellas herramientas destinadas a la integración de publicidad en la aplicación como medio de monetización de la misma.
- **Analítica.** Agrupan las diferentes herramientas relacionadas con el análisis de datos generados por nuestra aplicación, con el objetivo de crear estadísticas a partir de las cuales tomar diferentes decisiones sobre nuestro producto.

Para nuestro proyecto, utilizaremos solo herramientas propias del grupo de desarrollo, concretamente: Firebase Authentication, Cloud Firestore y Cloud Storage.

4.2.1.1 Firebase Authentication

Esta librería proporciona servicios de autenticación backend, que permiten gestionar el acceso de los usuarios a la aplicación (Firebase, s. f.-c).

Entre las diferentes formas de autenticación, destacan la autenticación basada en correo electrónico y contraseña, la integración con proveedores de identidad federada, la autenticación con número de teléfono, la integración de sistemas de autenticación personalizados o la autenticación anónima.

Además, permite a los desarrolladores implementar de manera sencilla diferentes herramientas complementarias a la autenticación, como la verificación de registro mediante correo electrónico, la recuperación de contraseña o la modificación de la misma.

4.2.1.2 Cloud Firestore

Permite al usuario crear y gestionar una base de datos NoSQL flexible y escalable alojada en la nube (Firebase, s. f.-a). Al igual que cualquier base de datos, permite al usuario realizar las diferentes operaciones CRUD sobre los datos almacenados, además de permitir al administrador gestionar las diferentes reglas de acceso a la base de datos.

La unidad de almacenamiento de esta base de datos es el documento, que está compuesto por pares campo-valor, que permiten representar la información. Estos documentos se agrupan formando diferentes colecciones, de manera que el usuario pueda organizar todos sus documentos en ellas. También existe la posibilidad de crear subcolecciones.

4.2.1.3 Cloud Storage

Servicio de almacenamiento de objetos en la nube que permite la carga y descarga de contenidos como imágenes, audios y vídeos, entre otros (Firebase, s. f.-b).

Este sistema permite crear diferentes carpetas donde organizar los contenidos del usuario. Este tipo de almacenamiento es perfecto para archivos sencillos de los que no es necesario almacenar mucha información.

4.2.2 Zxing

Librería de código abierto que nos permite escanear y generar códigos de barras y códigos QR, entre otros (JourneyApps, 2012/2022).

4.2.3 CircleImageView

Permite crear Image Views circulares para los diferentes Activites de nuestra aplicación. Esta librería nos permite modificar aspectos como el color de fondo de la imagen, así como el color y el ancho del borde, entre otros (Dodenhof, 2014/2022).

4.2.4 Picasso

Permite cargar y mostrar imágenes subidas en la red desde nuestra aplicación Android para mostrarlas dentro de nuestros Activities (square, 2013/2022).

4.2.5 Dokka

Dokka es un motor de documentación en Kotlin, análogo a Javadoc para Java. Nos permite generar una documentación del código de nuestra aplicación en formato HTML a partir de comentarios escritos en diferentes formatos en el propio código (Kotlin, 2014/2022).

4.3 Lenguajes

4.3.1 Kotlin

Kotlin es un lenguaje de programación de tipado estático que se ejecuta sobre la máquina virtual de Java y desarrollado por la empresa JetBrains (Wikipedia, 2021).

Es un lenguaje muy similar a Java, pero con algunas importantes mejoras con respecto a él, como por ejemplo, la falta de necesidad de puntos y coma al final de cada línea de código, la inferencia de tipos y el manejo de nulos para evitar excepciones, entre otros.

Junto a Java, son los dos lenguajes utilizados para desarrollar aplicaciones Android en el IDE Android Studio, siendo Kotlin cada vez más popular.

4.3.2 XML

XML, Extensible Markup Language o Lenguaje de Marcado Extensible es un lenguaje de marcado de propósito general (Mozilla, s. f.).

En el campo del desarrollo Android, XML es el lenguaje utilizado para generar las diferentes interfaces de usuario de las Activities que conforman cada aplicación.

4.4 Herramienta de control de versiones

4.4.1 Github y Github Desktop

GitHub es una plataforma de desarrollo colaborativo, diseñado para el control de versiones utilizando el sistema GIT. Es uno de los clientes de Git GUI más populares. Al igual que GIT, permite a los usuarios crear y gestionar repositorios de código.

Su versión de escritorio, Github Desktop, permite realizar modificaciones en el repositorio sin necesidad de utilizar comandos de consola.

4.5 Herramientas CASE

4.5.1 Visual Paradigm y Visual Paradigm Online

Herramienta de gestión y diseño que permite crear diferentes esquemas, diagramas y modelos UML destinados al desarrollo de software y sistemas de TI (Visual Paradigm, s. f.).

Además de su versión de escritorio, cuenta con una versión web, Visual Paradigm Online, que dependiendo del caso concreto, puede resultar más útil o sencillo de utilizar.

4.5.2 Diagrams.net

Aplicación web de diseño de diagramas que permite, de una forma sencilla, generar distintos tipos de diagramas, incluyendo numerosas herramientas UML (diagrams.net, s. f.).

Cuenta con soporte para diferentes espacios de almacenamiento, como Google Drive, One Drive, Dropbox, entre otros.

4.5.3 EZEstimate

Herramienta que permite, a partir de los actores, los casos de uso y los factores de complejidad de un proyecto software, estimar la duración del mismo, expresada en horas/UCP (Use Case Points).

4.5.4 Microsoft Project

Software propiedad de Microsoft que ofrece diferentes funcionalidades para planificar y gestionar el proceso de desarrollo de un producto software (Microsoft, s. f.).

Permite dividir el proyecto en diferentes tareas, asignando a cada una de ellas una duración, prioridad y número de recursos con el objetivo de optimizar la planificación del proyecto.

4.6 Herramientas de diseño gráfico

4.6.1 Looka

Plataforma web destinada al diseño automatizado de logos. Cuenta con algoritmos de inteligencia artificial, que generan diferentes opciones de diseños en función de los patrones indicados por el usuario (Looka, s. f.).

4.6.2 FontAwesome

Es una librería de iconos online, con numerosos estilos y diseños. Cuenta con una amplia variedad de iconos, tanto gratuitos como de pago, destinados al diseño web y de aplicaciones (Fontawesome, s. f.).

4.6.3 IconFinder

Librería de iconos e ilustraciones online. Al igual que FontAwesome, cuenta con diferentes opciones, algunas gratis y otras de pago. Sin embargo, cuenta con mayor variedad de estilos, además de permitir la descarga de los iconos en diferentes tamaños y formatos (Iconfinder, s. f.).

5. ASPECTOS RELEVANTES DEL TRABAJO

En este apartado se detallarán los aspectos más destacables a lo largo del desarrollo del proyecto, diferenciando entre las diferentes fases del mismo.

5.1 Marco de trabajo

Durante el desarrollo del proyecto se ha seguido la metodología del Proceso Unificado, una metodología utilizada en desarrollo software que utiliza el lenguaje UML para representar en forma de gráficos y diagramas los diversos conceptos de un proyecto software. Las características principales del proceso unificado son:

- **Es iterativo e incremental.** A la hora de desarrollar un sistema, se divide este desarrollo en las siguientes etapas: inicio, elaboración, construcción y transición. Cada una de estas etapas se divide a su vez en un número determinado de iteraciones, cada una de estas compuesta por diversas disciplinas, las cuales pueden ser: modelado de negocio, requisitos, análisis, diseño, implementación y pruebas. Al finalizar cada una de las iteraciones, se alcanza un hito, que marca el final de esa fase del desarrollo del proyecto.
- **Está centrado en la arquitectura.** Para representar un sistema, el Proceso Unificado emplea múltiples modelos y vistas para intentar cubrir todos los aspectos del sistema software a desarrollar.
- **Está dirigido por los casos de uso.** Los casos de uso representan las diferentes acciones que los actores del sistema pueden realizar sobre el mismo, las cuales pueden generar una serie de respuestas por parte del sistema. El conjunto de los casos de uso de un sistema constituye la base del Proceso Unificado.

Como hemos mencionado anteriormente, el Proceso Unificado divide el desarrollo de un sistema en 4 etapas bien diferenciadas:

- **Inicio.** En esta fase inicial se define la amplitud del proyecto, además de los objetivos, metas y complejidad del mismo.
- **Elaboración.** Durante esta etapa, se especifican los aspectos funcionales a implementar en el sistema. El resultado de la finalización de esta etapa será la obtención de una visión refinada del proyecto.
- **Construcción.** Esta es la etapa más larga del Proceso Unificado, que abarca desde la implementación de las diferentes funcionalidades del sistema hasta la generación de un producto listo y que incluya, al menos, los requisitos mínimos.
- **Transición.** En esta etapa final, se realizarán las diferentes pruebas y labores de mantenimiento del producto, generando al final un producto completo, preparado para ser utilizado por el cliente.

5.2 Estimación de coste y esfuerzo

En este apartado detallaremos el proceso de estimación de coste y esfuerzo del proyecto, mediante el cual obtendremos una dimensión aproximada de la dimensión del sistema y del esfuerzo requerido para completar el desarrollo del mismo.

Para esta estimación se ha empleado el método de cálculo de puntos de casos de uso del sistema (UCP). El resultado de este nos dará una aproximación general del esfuerzo requerido para el desarrollo de nuestro sistema en horas de trabajo por persona. Esta métrica se obtendrá a partir de la siguiente fórmula:

$$UCP = UUCP \cdot TCF \cdot ECF$$

A continuación, detallaremos la definición de cada una de las variables, así como su obtención.

5.2.1 UUCP

Los UUCP o puntos de caso de uso sin ajustar se calculan como la suma del peso de los casos de uso sin ajustar o UUCW más el peso de los actores sin ajustar o UAW.

$$UUCP = UUCW + UAW$$

El UUCW se calcula siguiendo la siguiente fórmula:

$$UUCP = \sum peso\ i$$

donde *peso i* será el peso de cada uno de los diferentes casos de uso en función de su complejidad. Esta complejidad la definirá el número de transiciones que existan en cada caso de uso, siguiendo la siguiente clasificación:

- Caso de uso **simple**. Si su secuencia consta de 3 transacciones o menos.
- Caso de uso **medio**. Si su secuencia consta de entre 4 y 7 transacciones.
- Caso de uso **complejo**. Si su secuencia consta de más de 7 transacciones.

En cuanto al peso de los actores sin ajustar o UAW, se calculará mediante la siguiente fórmula:

$$UAW = \sum peso\ i$$

donde *peso i* será el peso de cada uno de los diferentes actores del sistema en función de su complejidad, definida en la siguiente clasificación:

- Actor **simple**. El actor es un sistema y la aplicación se comunica con él a través de una API.
- Actor **medio**. El actor es un sistema y la aplicación utiliza un protocolo para comunicarse con él.
- Actor **complejo**. El actor es una persona que utiliza interfaces gráficas para comunicarse con la aplicación.

Los UUCP y UAW de nuestro sistema, serán representados en la *Tabla 1* y en la *Tabla 2*, respectivamente.

Actor	Complejidad
Usuario	Complejo
Alumno	Complejo
Profesor	Complejo

Tabla 1. Complejidad de los actores del sistema

Caso de Uso	Complejidad
UC-001. Iniciar Sesión	Simple
UC-002. Registrar	Medio
UC-003. Recuperar Contraseña	Medio
UC-004. Ver Perfil	Simple
UC-005. Modificar Perfil	Medio
UC-006. Cerrar Sesión	Medio
UC-007. Ver Asignatura Alumno	Simple
UC-008. Ver Asignatura Profesor	Simple
UC-009. Nueva Asignatura Alumno	Simple
UC-010. Nueva Asignatura Profesor	Simple
UC-011. Editar Asignatura	Medio
UC-012. Eliminar Asignatura	Medio
UC-013. Desmatricular Alumno	Medio
UC-014. Ver Encuesta Alumno	Simple
UC-015. Ver Encuesta Profesor	Simple
UC-016. Nueva Encuesta	Simple
UC-017. Editar Encuesta	Medio
UC-018. Eliminar Encuesta	Medio
UC-019. Ver Preguntas	Simple
UC-020. Nueva Pregunta	Simple
UC-021. Editar Encuesta	Medio
UC-022. Eliminar Pregunta	Medio
UC-023. Resolver Encuesta	Medio
UC-024. Ver Resultados Alumno	Simple
UC-025. Ver Resultados Profesor	Simple
UC-026. Eliminar Resultado	Medio

Tabla 2. Complejidad de los casos de uso del sistema

5.2.2 TCF y ECF

Los TCF o factores de complejidad técnica son 13 factores diferentes que agrupan diferentes aspectos del proyecto, asignando a cada uno de ellos un peso y una complejidad. Se calculan mediante la siguiente fórmula:

$$TCF = C_1 + C_2 \sum_{i=1}^{13} W_i F_i$$

donde W_i es el peso del factor de complejidad i y F_i , la complejidad de dicho factor. Esta complejidad puede oscilar entre 0, considerando esta como una complejidad irrelevante y 5, siendo esta complejidad de alta relevancia. C_1 y C_2 son dos constantes. En nuestro caso, los valores de W_i vienen definidos dentro del modelo de la aplicación de escritorio que utilizaremos para la estimación de coste y esfuerzo: EZEstimate.

Los ECF o factores de complejidad de entorno son 8 factores diferentes que agrupan diferentes características del entorno de desarrollo del proyecto. Se calculan de manera análoga a los TCF, siguiendo la siguiente fórmula.

$$ECF = C_1 + C_2 \sum_{i=1}^8 W_i F_i$$

donde W_i es el peso del factor de complejidad i y F_i , la complejidad de dicho factor. Igual que en los TCF, puede oscilar entre 0 y 5, C_1 y C_2 son dos constantes y los valores de W_i vienen definidos dentro del modelo de EZEstimate.

En la Tabla 3 y la Tabla 4, se representarán los TCF y los ECF de nuestro sistema.

5.2.3 Resultados

Tras introducir todos los datos mostrados en el apartado anterior en la herramienta *EZEstimate*, obtendremos los resultados mostrados en la *Figura 1*.

Id	Module	Type	Name	complexity
1	Default	Actor	Usuario	Complex
10	Default	Usecase	UC-007. Ver Asi...	Simple
11	Default	Usecase	UC-008. Ver Asi...	Simple
12	Default	Usecase	UC-009. Nueva...	Simple
13	Default	Usecase	UC-010. Nueva...	Simple
14	Default	Usecase	UC-011. Editar ...	Average
15	Default	Usecase	UC-012. Elimina...	Average
16	Default	Usecase	UC-013. Desma...	Average
17	Default	Usecase	UC-014. Ver En...	Simple
18	Default	Usecase	UC-015. Ver En...	Simple
19	Default	Usecase	UC-016. Nueva...	Simple
2	Default	Actor	Alumno	Complex
20	Default	Usecase	UC-017. Editar ...	Average
21	Default	Usecase	UC-018. Elimina...	Average
22	Default	Usecase	UC-019. Ver Pr...	Simple
23	Default	Usecase	UC-020. Nueva...	Simple
24	Default	Usecase	UC-021. Editar ...	Average
25	Default	Usecase	UC-022. Filmina	Average

Figura 1. Resultados estimación de coste y esfuerzo EZEstimate

Como podemos ver en ella, la estimación del coste nos dará como resultado 1208,088, que representarían las horas por persona que llevaría desarrollar la totalidad del proyecto.

5.3 Planificación temporal

A continuación, detallaremos el proceso de planificación temporal del desarrollo de nuestro proyecto, que consiste en dividir este proceso en diferentes tareas y subtareas, asignando a cada una de ellas tanto su duración como el número de personas que se encargarán de llevar a cabo las mismas en cada momento del desarrollo.

Para realizar esta planificación, tomaremos el modelo del diagrama de Gantt, utilizando la herramienta *Microsoft Project*, que nos permite realizar todas funciones descritas anteriormente.

Como mencionamos anteriormente en el apartado *5.1 Marco de trabajo*, la planificación temporal de nuestro trabajo tendrá en cuenta que el desarrollo se dividirá en 4 fases principales, que detallaremos una por una a continuación.

5.3.1 Inicio

En esta primera etapa, dividida en dos iteraciones, sirve para aportar una visión general de la dimensión del proyecto, así como también de captar los diferentes objetivos, actores y requisitos derivados del enunciado.

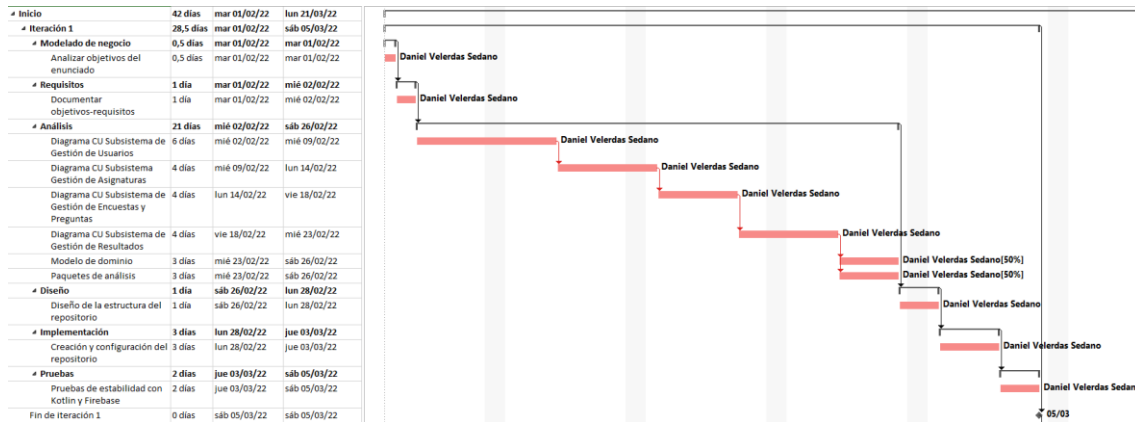


Figura 2. Diagrama de Gantt - Inicio, iteración 1

5.3.2 Elaboración

En esta fase, dividida en tres iteraciones, abarcaremos la parte de análisis de los requisitos de nuestra aplicación, dejando todo preparado para llevar a cabo la siguiente fase de desarrollo.

5.3.3 Construcción

Constituye la fase más extensa del desarrollo, y no es para menos, puesto que esta etapa abarca el diseño y la implementación de cada uno de los diferentes paquetes en los que se divide nuestro proyecto, así como también de refinar los diferentes diagramas generados del análisis llevado a cabo en la etapa anterior.

5.3.4 Transición

La fase de transición constituirá la fase final de nuestra planificación, donde realizaremos una revisión general del proyecto, refinando todas las etapas anteriores, así como realizando pruebas para detectar y subsanar posibles errores en la implementación. En esta fase también se contemplará la introducción de nuevas funcionalidades al código.

5.4 Especificación de requisitos

En este apartado, recogeremos la especificación de los requisitos de nuestro sistema. Esta especificación incluirá la definición de los objetivos, así como la definición de actores, requisitos funcionales y no funcionales de nuestro proyecto.

Todos estos requisitos se obtendrán siguiendo la metodología de Durán y Bernárdez, como aparece desarrollado en el *Anexo II. Especificación de Requisitos Software*.

5.4.1 Objetivos del sistema

Como bien se menciona en el apartado *2.1 Objetivos del sistema*, los objetivos a cumplir por nuestro sistema software se dividen principalmente en cuatro:

- Gestión de usuarios
- Gestión de asignaturas
- Gestión de encuestas y preguntas
- Gestión de resultados

En el *Anexo II. Especificación de Requisitos Software* se representa cada uno de estos siguiendo la estructura detallada en la *Tabla 3*.

OBJ-001	Gestión de Usuarios
Versión	1.0
Autores	Daniel Velerdas Sedano
Descripción	El sistema deberá permitir que los usuarios se registren, inicien y cierren sesión (siendo esta independiente para cada usuario) y modifiquen sus datos.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 3. OBJ-001. Gestión de Usuarios

5.4.2 Requisitos de información

Son aquellos que reflejan los datos que debe almacenar nuestro sistema para su correcto funcionamiento.

Entre ellos encontramos:

- Datos de usuario
- Datos de asignatura
- Datos de encuesta
- Datos de pregunta
- Datos de resultados

5.4.3 Requisitos de restricción de información

Representan aquellas restricciones especiales que deben cumplirse en nuestro sistema para que el buen funcionamiento del mismo.

En nuestro caso, solamente hemos identificado uno:

- Usuarios registrados

5.4.4 Definición de actores

Representan a todos los agentes que interactuarán con nuestro sistema.

En nuestro caso, diferenciamos entre tres:

- Usuario
- Profesor
- Alumno

Estos actores, se relacionan entre sí siguiendo la jerarquía mostrada en la *Figura 3* y su presentación se realiza mediante diferentes tablas análogas a la mostrada en la *Tabla 4*.

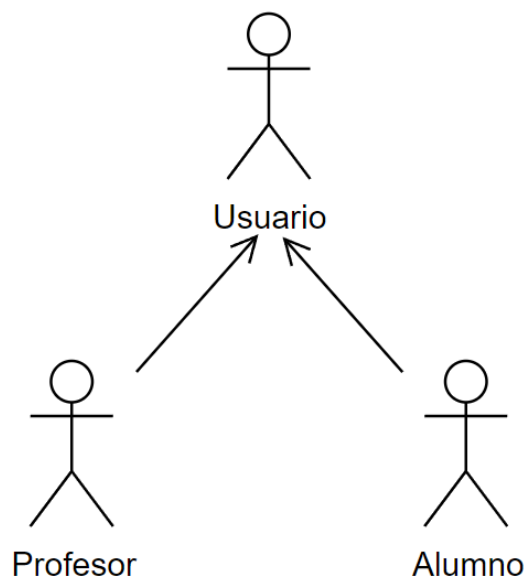


Figura 3. Diagrama de Actores

ACT-001	Usuario
Versión	1.0
Autores	Daniel Velerdas Sedano
Descripción	Este actor representa los usuarios que aún no se han autenticado en el sistema.
Comentarios	Ninguno

Tabla 4. ACT-001. Actor Usuario

5.4.5 Casos de uso del sistema

Se ocupan de representar la funcionalidad o servicios que ofrece nuestro sistema. Los casos de uso de nuestro sistema se encuentran divididos entre los diferentes paquetes que componen nuestro proyecto.

Para reflejar las relaciones entre actores y casos de uso dentro de cada uno de los paquetes, emplearemos diagramas como el mostrado en la *Figura 4*.

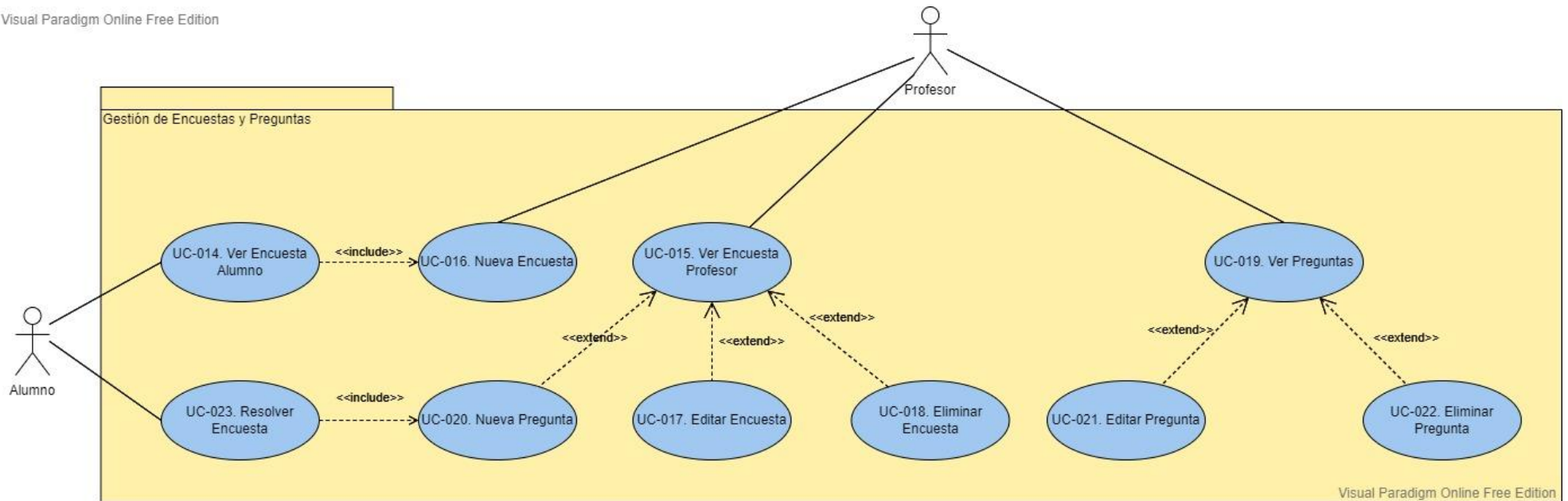


Figura 4. Diagrama de Casos de Uso – Gestión de Encuestas y Preguntas

Para la especificación de cada uno de los casos de uso del sistema, se empleará la estructura mostrada en la *Tabla 5*.

UC-016	Nueva Encuesta	
Versión	1.0	
Autores	Daniel Velerdas Sedano	
Dependencias	[OBJ-003] Gestión de Encuestas y Preguntas [IRQ-003] Datos de Encuesta	
Descripción	El profesor creará una encuesta nueva.	
Precondición		
Secuencia normal	Paso	Acción
	1	El actor ACT-002 Profesor solicitará crear una encuesta.
	2	El profesor introducirá los datos de la nueva encuesta.
	3	La encuesta será creada.
Postcondición		
Excepciones	Paso	Acción
	2	Si alguno de los datos introducidos no es válido, se notificará al usuario para que vuelva a intentarlo.
Importancia	Vital	
Urgencia	Alta	
Estado	Validado	
Comentarios	Ninguno	

Tabla 5. UC-016. Nueva Encuesta

5.4.6 Requisitos no funcionales

Estos se ocupan de mostrar diversas limitaciones y características propias de nuestro sistema software.

5.5 Análisis del sistema

En esta etapa, se realizará un análisis de los requisitos obtenidos en la etapa anterior. Al finalizar, tendremos una idea general de la arquitectura de nuestro sistema, que será revisada y perfeccionada en la fase de diseño.

Esta fase se encuentra desarrollada en el *Anexo III. Análisis de Requisitos* y será dividida en cuatro apartados, los cuales detallaremos a continuación.

5.5.1 Modelo del dominio

Representa conceptos del mundo real que son significativos para el dominio del problema de nuestro sistema, detallando además los atributos más relevantes de cada uno de ellos y cómo estos conceptos se relacionan entre sí. Para realizar esta representación se ha empleado el diagrama de clases, que podemos ver en la *Figura 5*.

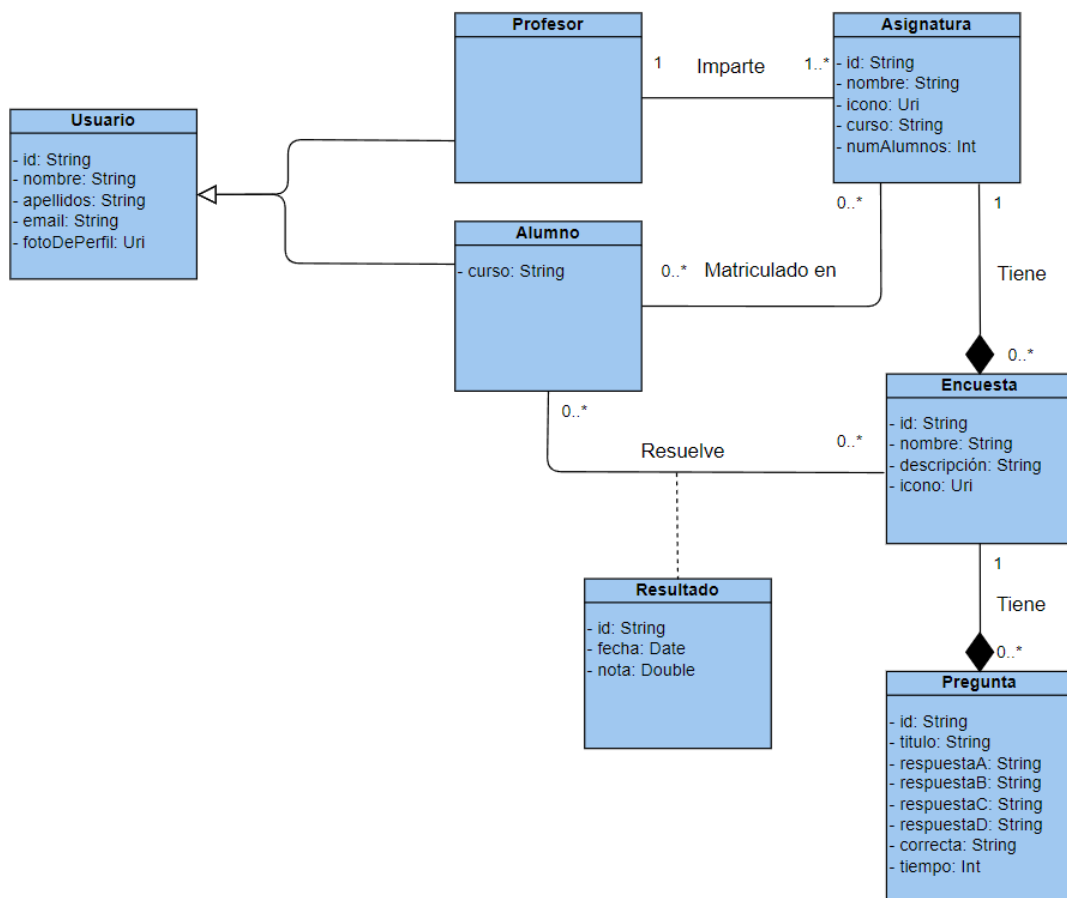


Figura 5. Diagrama de Clases

5.5.2 Paquetes de análisis

Para este apartado, utilizamos el diagrama de paquetes de análisis, en el que se refleja la división de la estructura interna del sistema en los diferentes paquetes, representando a su vez, cada uno de estos por separado. En la *Figura 6*, veremos el diagrama general de paquetes y en la *Figura 7*, un ejemplo de diagrama de paquete detallado.

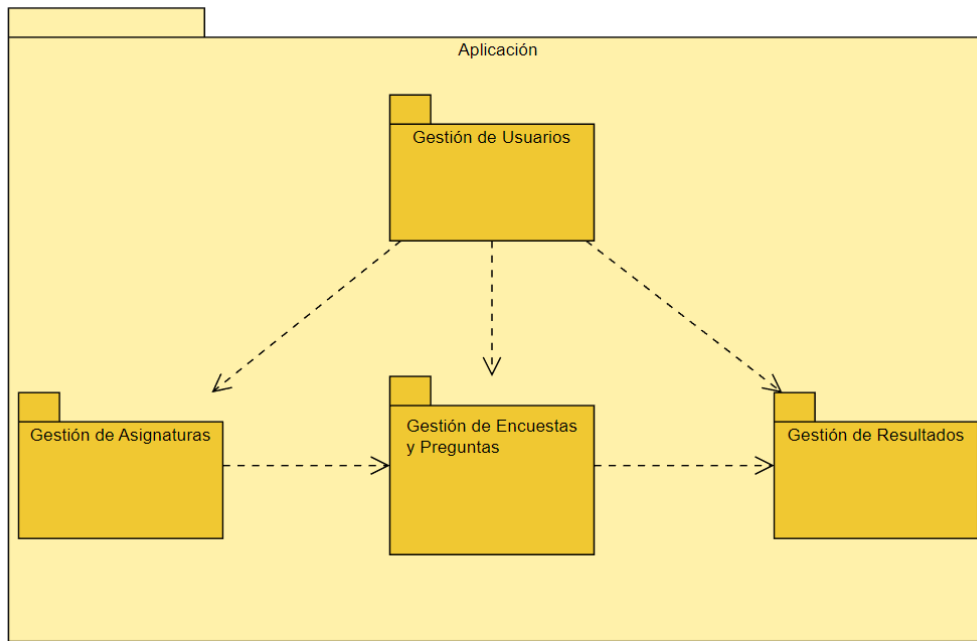


Figura 6. Diagrama de Paquetes de Análisis

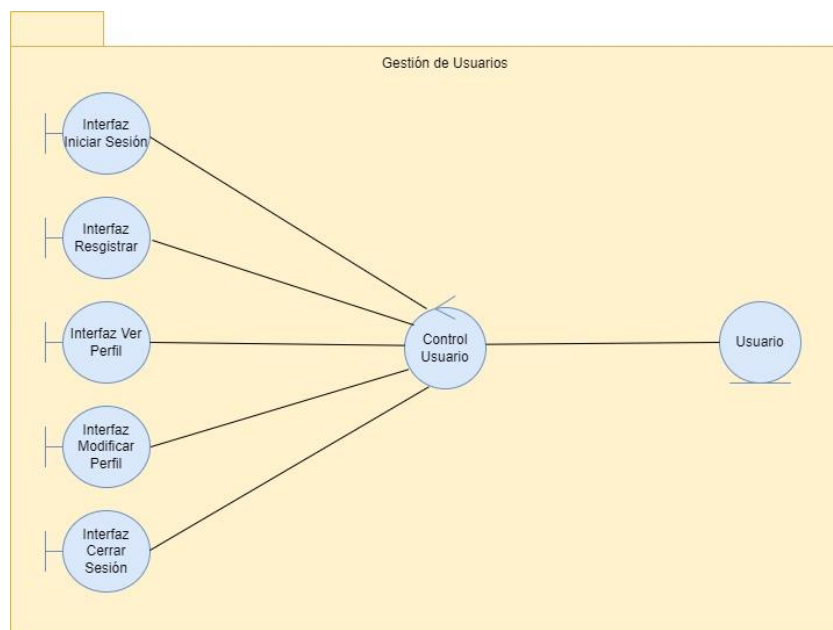


Figura 7. Diagrama de Paquetes de Análisis – Gestión de Usuarios

5.5.3 Realización de casos de uso

Partiendo de los casos de uso definidos en el apartado 5.4.5 *Casos de uso del sistema* y el *Anexo II. Especificación de Requisitos*, llevaremos a cabo su realización mediante el uso de diagramas de secuencia como el mostrado en la *Figura 8*.

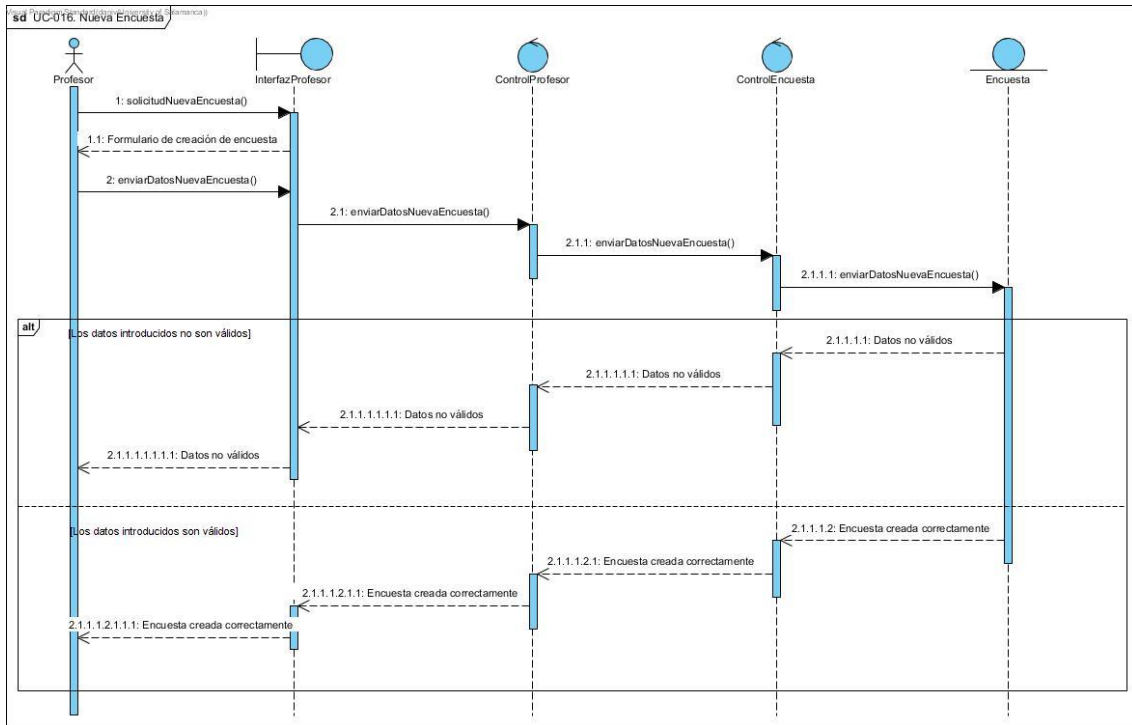


Figura 8. Diagrama de Secuencia UC-016. Nueva Encuesta

5.6 Diseño del sistema

Esta etapa comprende el comienzo del modelo de la solución de nuestro desarrollo. Este proceso está detallado en el *Anexo IV. Especificación de Diseño*.

5.6.1 Modelo de diseño

En este apartado se definirán los diferentes patrones y arquitecturas que facilitarán la posterior implementación del sistema.

5.6.1.1 Patrones arquitectónicos

- **Modelo-Vista-Controlador.** Este patrón se caracteriza por dividir los componentes software del sistema en tres grupos: el modelo, que constituye una representación de los datos del sistema, la vista, que sirve como comunicación entre la aplicación y el usuario, y el controlador, que será el intermediario entre modelo y vista, además de realizar los cambios y adaptaciones necesarias a los distintos datos.

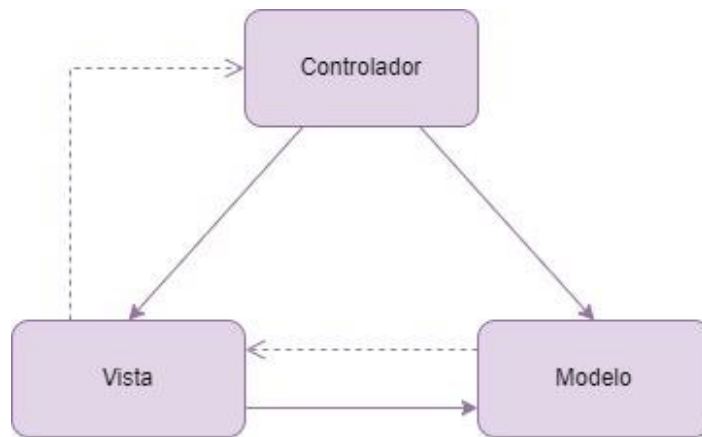


Figura 9. Patrón Modelo-Vista-Controlador

- **Adaptador.** Se emplea para transformar una interfaz antigua en otra nueva para que esta pueda ser utilizada por otra clase que no podía utilizar la primera interfaz. En nuestra aplicación, hacemos uso de este patrón de diseño para representar en la vista datos provenientes del Modelo.

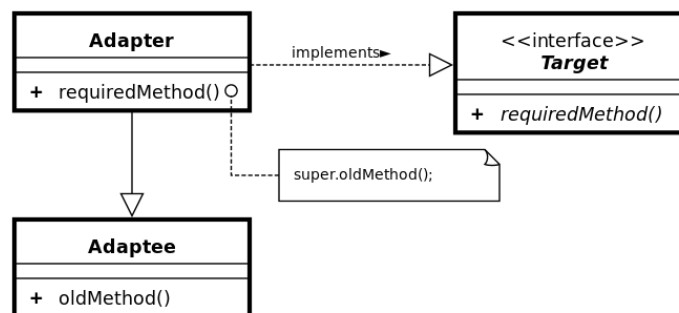


Figura 10. Patrón Adaptador

5.6.2 Subsistemas de diseño

Dividiremos el sistema de nuestro sistema en diferentes subsistemas. Esta estructura se representará mediante el diagrama de subsistemas, como puede verse en la *Figura 11*.

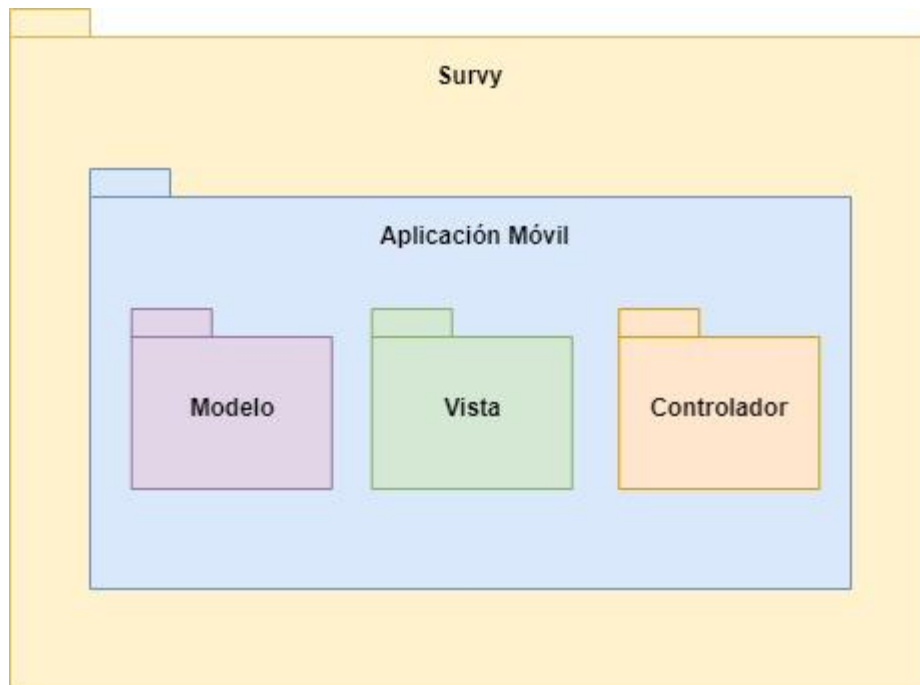


Figura 11. Diagrama de Subsistemas

5.6.3 Clases de diseño

Dentro de nuestra aplicación Android y como vemos en el diagrama de subsistemas de la *Figura 11*, dividiremos las diferentes clases siguiendo el patrón Modelo-Vista-Controlador explicado en el anterior apartado *5.6.1.1 Patrones arquitectónicos*.

Esta división está detallada en los diferentes diagramas de modelo de cada subsistema los cuales pueden verse en las *Figura 12*, *Figura 13* y *Figura 14*.

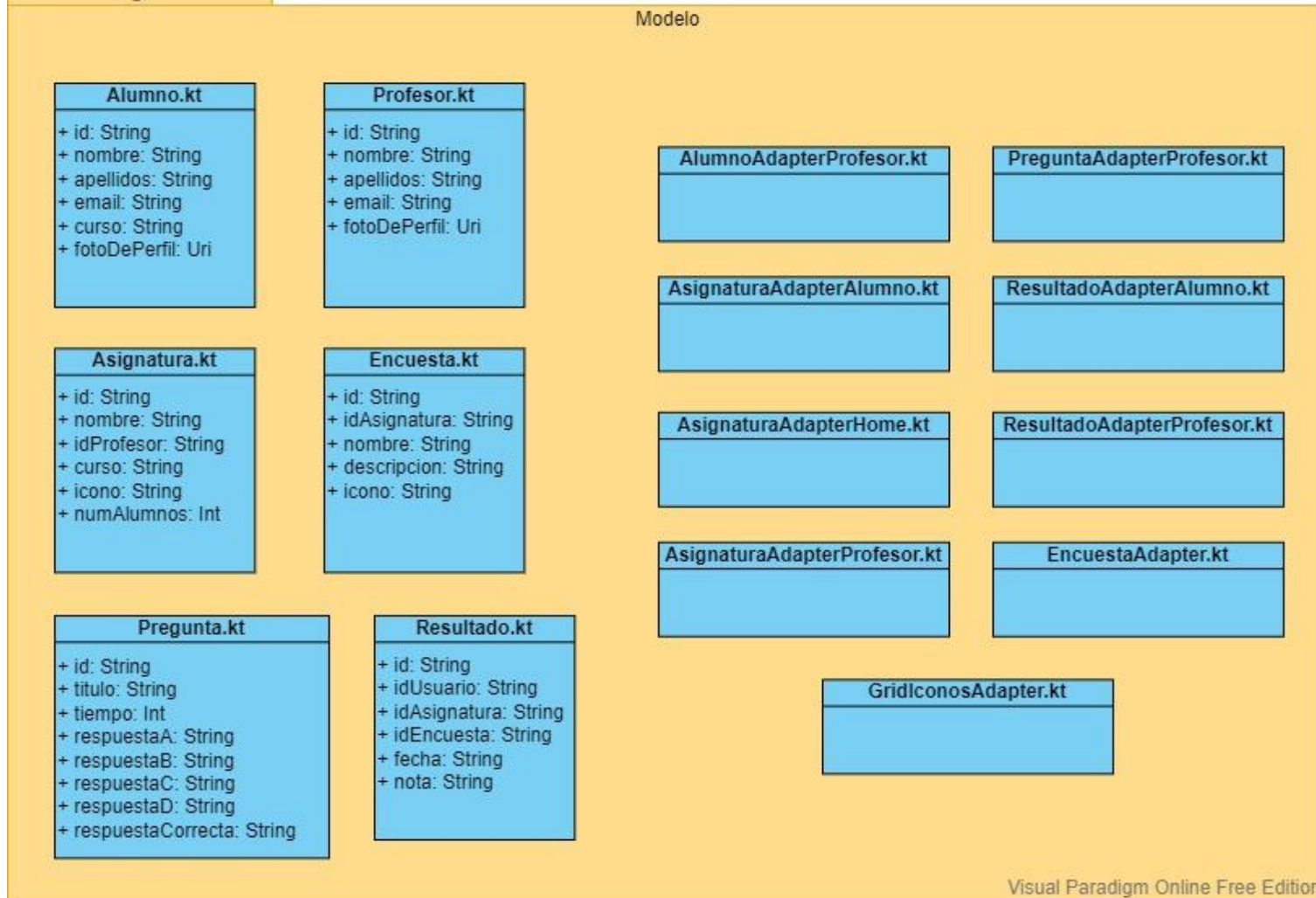


Figura 12. Diagrama Modelo del subsistema de la Aplicación Móvil

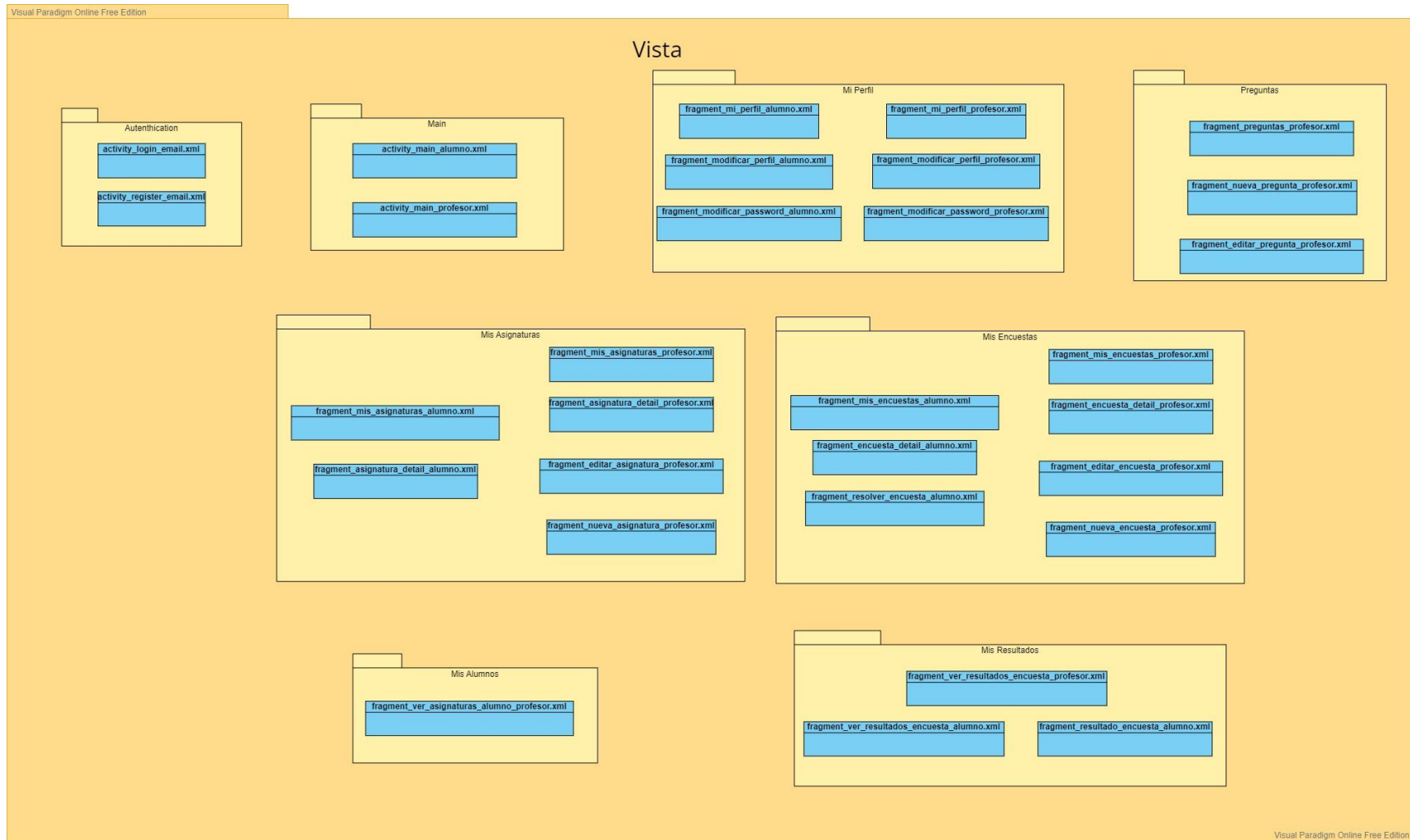


Figura 13. Diagrama Vista del subsistema de la Aplicación Móvil

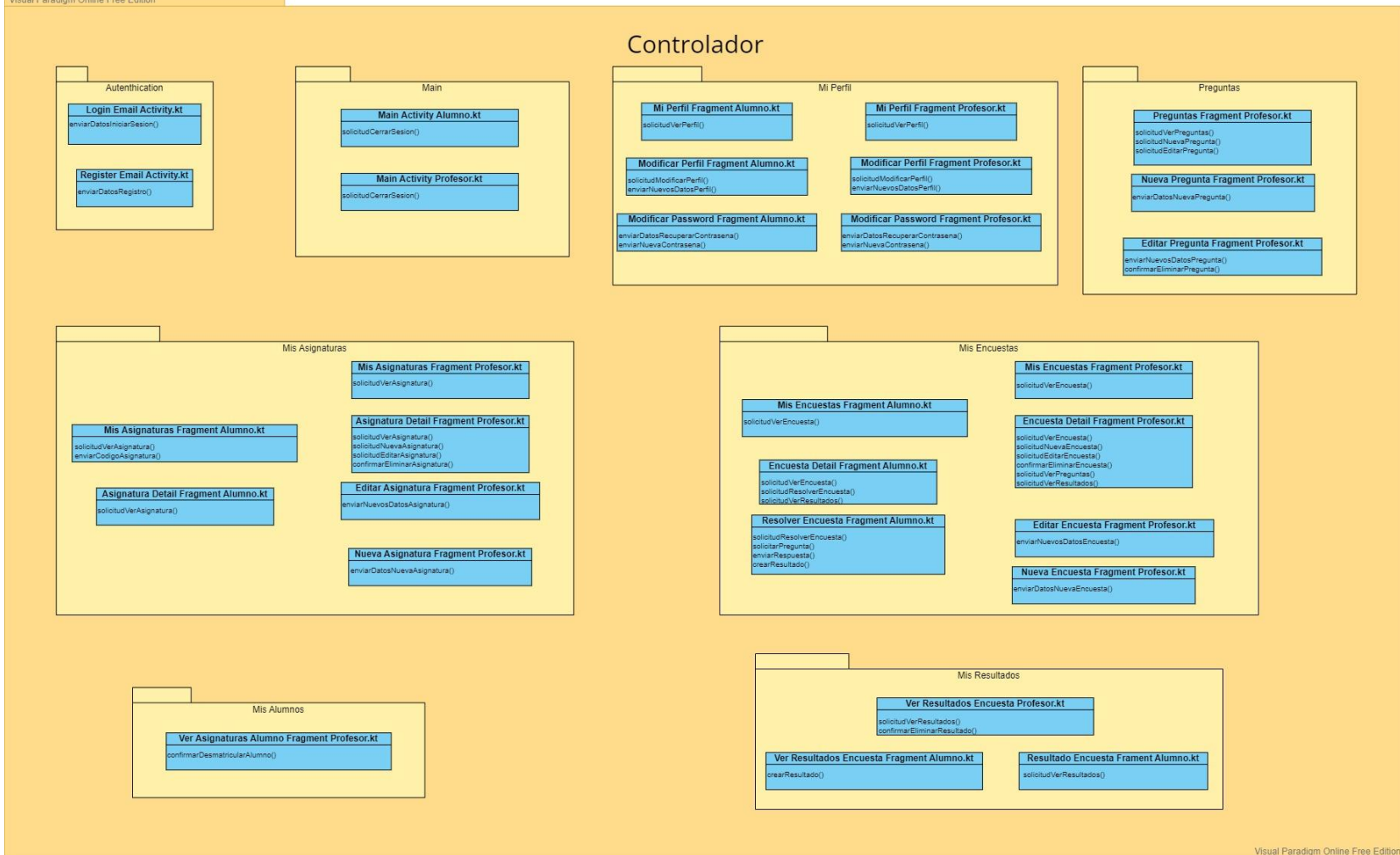


Figura 14. Diagrama Controlador del subsistema de la Aplicación Móvil

5.6.4 Realización de casos de uso

A partir de los diagramas de secuencia del apartado anterior 5.5.3 *Realización de casos de uso* detallados en el *Anexo III. Análisis de Requisitos*, compondremos los diagramas de secuencia de diseño, que tendrán una estructura similar al mostrado en la *Figura 15*. Todos estos diagramas aparecerán de manera detallada en el *Anexo IV. Especificación de Diseño*.

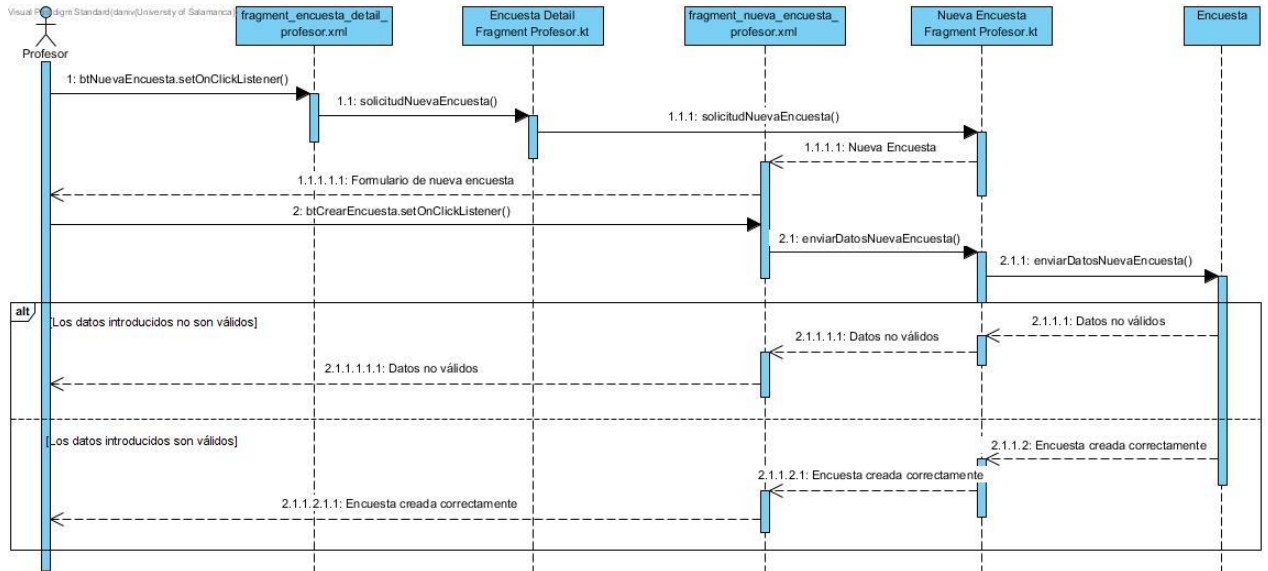


Figura 15. Diagrama de Secuencia de diseño UC-016. Nueva Encuesta

5.6.5 Modelo de despliegue

Finalmente, mediante el diagrama de despliegue, mostraremos cómo se han distribuido los diferentes artefactos del sistema, además de la relación entre ellos.

Estos aspectos pueden verse en la *Figura 16*.

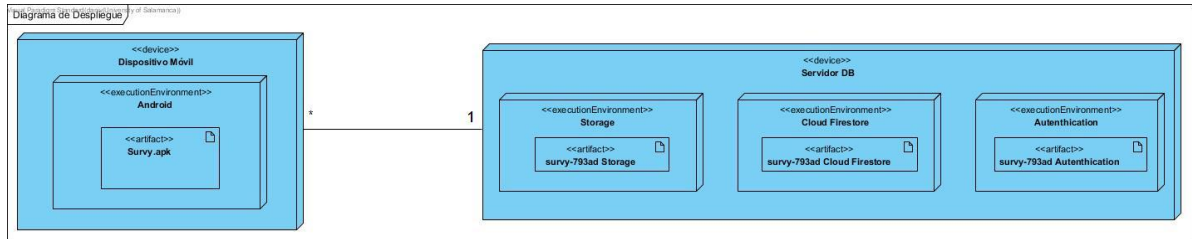


Figura 16. Diagrama de Despliegue

6. IMPLEMENTACIÓN

Una vez finalizadas las etapas de análisis y diseño y partiendo de estas, realizaremos la implementación de la aplicación. Como pudimos ver en el apartado 5.3 *Planificación temporal*, la implementación de nuestra aplicación corresponde a la etapa de construcción, que es la más larga de todo el proceso de desarrollo.

6.1 Configuración de la base de datos

Para nuestro proyecto hemos optado por la utilización de una base de datos NoSQL, concretamente la librería de Firebase Cloud Firestore, para el almacenamiento de los datos de nuestro sistema.

La estructura de nuestra base de datos fue diseñada utilizando la herramienta diagrams.net y tiene el aspecto definido en la *Figura 17*.

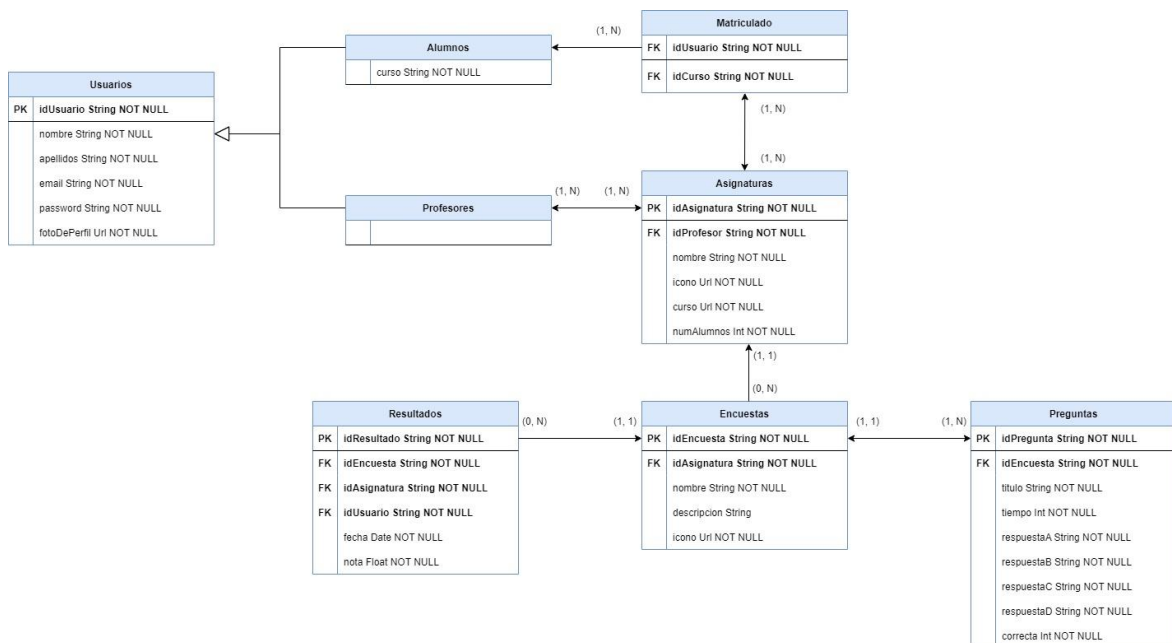


Figura 17. Diagrama de la base de datos del sistema

El aspecto de nuestra base de datos en Cloud Firestore visto desde la consola de Firebase es el mostrado en la *Figura 18*.

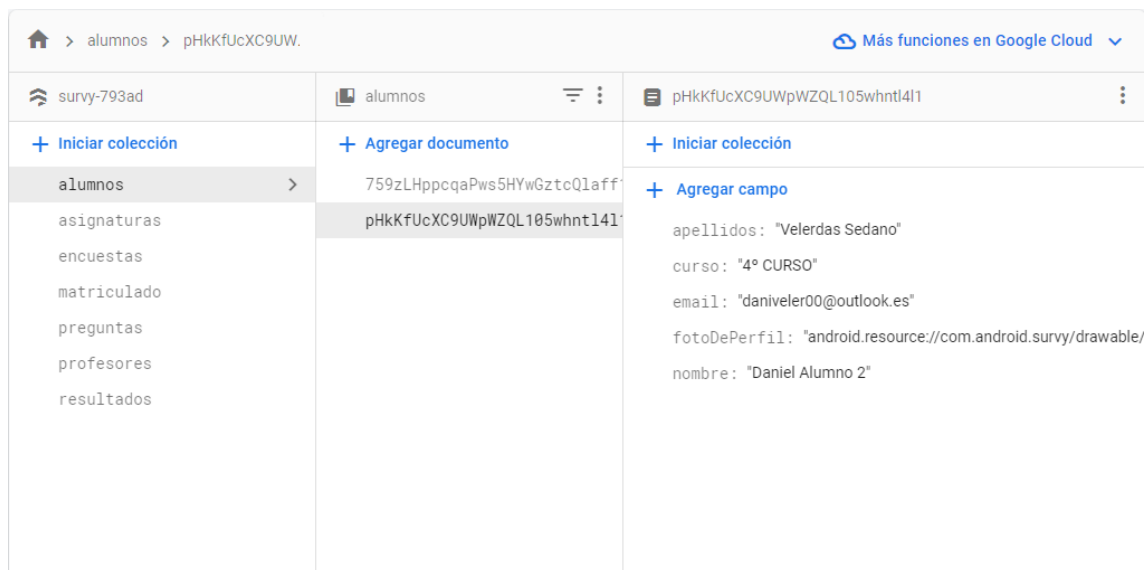


Figura 18. Base de datos de Cloud Firestore desde la consola de Firebase

Como mencionamos en el apartado *4.2.1.2 Cloud Firestore*, la base de datos de Cloud Firestore se dividen en colecciones, cada una de las cuales está compuesta por un número determinado de documentos.

Cada uno de los documentos almacenados en las colecciones, poseerá un identificador único, generado automáticamente por la propia librería.

En nuestra base de datos, contamos con las siguientes colecciones:

- **Alumnos.** Se encargará de almacenar los datos del perfil de los alumnos que se registren en la aplicación. En este caso, su identificador no es generado igual que los demás, sino que se extrae de la librería mencionada en el apartado *4.2.1.1 Firebase Authentication*, la cual genera automáticamente un UID a cada usuario cuando se registra.
- **Asignaturas.** Almacenará los datos de las diferentes asignaturas que sean creadas por los profesores registrados en el sistema. Además, almacenará el identificador del profesor que la creó, a modo de clave foránea.
- **Encuestas.** Guardará las encuestas creadas de una asignatura, guardando el identificador de dicha asignatura como clave foránea.
- **Matriculado.** Almacenará como claves foráneas los identificadores de los alumnos y los de las asignaturas a las que estos están matriculados.
- **Preguntas.** Contendrá los datos de las preguntas propias de una encuesta, de la que almacenará el id como clave foránea.
- **Profesores.** Esta colección se comportará de manera análoga a la de Alumno, pero para el rol de profesores.
- **Resultados.** Estos documentos serán generados cuando un alumno resuelva una encuesta y almacenará los datos del resultado obtenido, almacenando como claves foráneas los id de la encuesta, de la asignatura a la que pertenezca y del alumno.

Además, esta librería de Firebase nos permitirá definir unas reglas de lectura y escritura sobre nuestra base de datos, para limitar el acceso a la misma y así garantizar la seguridad de los datos almacenados.

Las reglas de nuestra base de datos están definidas en el apartado de Reglas dentro de la consola de Firebase. En nuestro caso, mediante la inclusión de reglas, hemos añadido una capa extra de seguridad a nuestra base de datos, permitiendo tan solo leer o modificar los datos de la misma a aquellos usuarios que se encuentren autenticados en nuestra aplicación, mediante el módulo de Firebase Authentication.

En el caso de nuestra aplicación, no sería necesario incluir una regla en la consola para que la cuenta del usuario que intente leer o escribir esté verificada, dado que por la lógica de nuestra aplicación, solamente podrán iniciar sesión y por tanto, autenticarse aquellos usuarios que ya hayan verificado su registro.

6.2 Inclusión de librerías

Las librerías incluidas en nuestro proyecto son aquellas mencionadas en el apartado 4.2 *Librerías*. Estas serán incluidas a nuestro proyecto en Android Studio dentro del documento build.gradle del módulo app.

El aspecto de estas implementaciones será el mostrado en la *Figura 20*.

```
39 dependencies {
40
41     implementation 'androidx.core:core-ktx:1.7.0'
42     implementation 'androidx.appcompat:appcompat:1.4.1'
43     implementation 'com.google.android.material:material:1.5.0'
44     implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
45     implementation('org.jetbrains.kotlin:kotlinx-coroutines-android:1.3.9')
46
47     // PICASSO
48     implementation 'com.squareup.picasso:picasso:2.71828'
49
50     // ZXING (CÓDIGOS QR)
51     implementation 'com.journeyapps:zxing-android-embedded:4.1.0'
52
53     // FIREBASE
54     implementation platform('com.google.firebase:firebase-bom:30.0.2')
55     implementation 'com.google.firebase:firebase-analytics-ktx'
56     implementation 'com.google.firebase:firebase-auth-ktx:21.0.3'
57     implementation 'com.google.firebase:firebase-auth:19.2.0'
58     implementation 'com.google.firebase:firebase-firestore-ktx'
59     implementation 'com.google.firebase:firebase-storage-ktx'
60
61
62     implementation 'com.google.android.gms:play-services-auth:20.1.0'
63     implementation 'com.facebook.android:facebook-login:latest.release'
64
65     implementation 'de.hdodenhof:circleimageview:3.1.0'
66     implementation 'androidx.legacy:legacy-support-v4:1.0.0'
67
68     testImplementation 'junit:junit:4.13.2'
69     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
70     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
```

Figura 19. Implementación de las librerías en el fichero build.gradle del módulo app

Una vez realizadas estas implementaciones, las funciones propias de cada librería serán utilizables a lo largo de nuestro desarrollo.

6.3 Desarrollo de la aplicación

En este apartado y con ayuda del *Anexo V. Documentación Técnica de Programación*, describiremos brevemente el proceso de desarrollo llevado a cabo durante la implementación de nuestro sistema.

Una vez realizada la configuración de la base de datos y la implementación de las librerías, detallados en los apartados anteriores, organizaremos nuestro código en diferentes directorios. En nuestro caso, contamos con los siguientes:

- Adapters
- Authentication
- Clases
- Fragments
- Main
- res/layout
- drawable, font, menú y values

Durante el desarrollo del código de nuestra aplicación, se empleó la librería Dokka, que nos permite generar una documentación del código en formato HTML. La documentación de nuestro proyecto puede consultarse ejecutando el archivo *index.html*, en el directorio *dokka*, dentro de los archivos de nuestra aplicación.

6.4 Funcionalidad de la aplicación

Finalmente, explicaremos el resultado de la implementación de nuestra aplicación, detallando sus diferentes funcionalidades y la navegación dentro de la misma. Toda esta información, se encuentra desarrollada en el *Anexo VI. Manual de Usuario*.

6.4.1 Autenticación

Este apartado de la aplicación comprenderá tanto el registro de un nuevo usuario como el inicio de sesión de un usuario ya registrado. Ambas funcionalidades permitirán al usuario identificarse como alumnos o profesores.

En la *Figura 20* veremos tanto el formulario de inicio de sesión como el de registro.

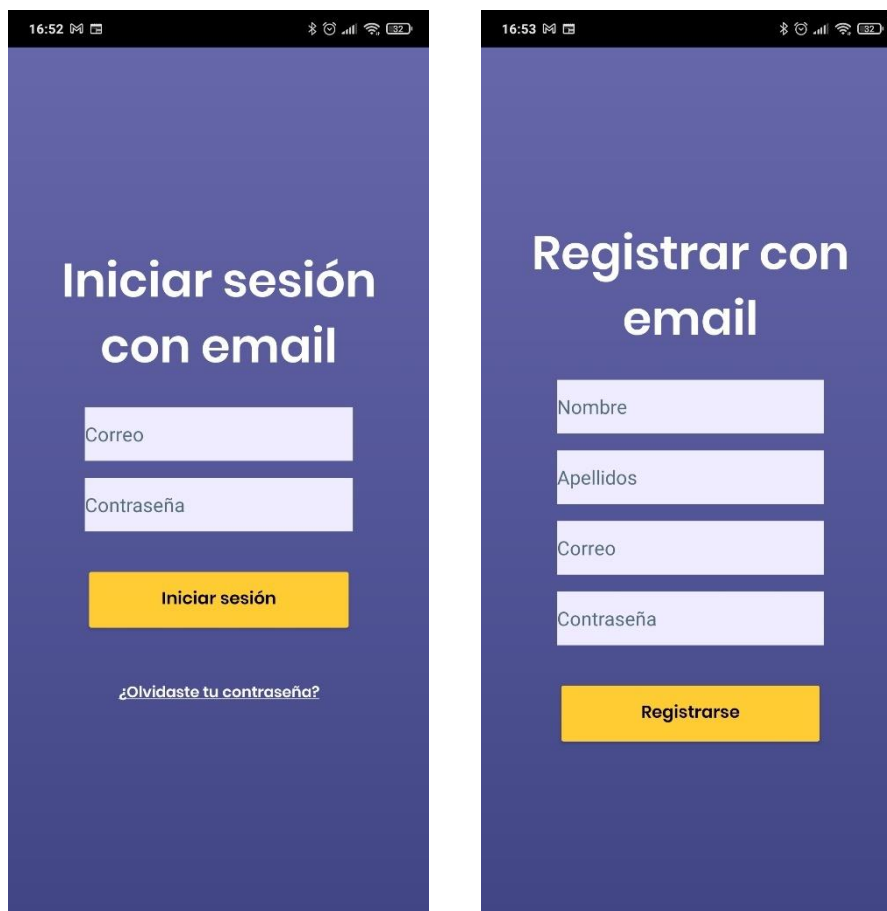


Figura 20. Formularios de inicio de sesión y registro

6.4.2 Menú principal

En nuestra aplicación, existirá un menú principal diferente en función de si el usuario autenticado es alumno o profesor, permitiéndole acceder a las funcionalidades propias de uno u otro rol.

El menú principal contará además con un menú desplegable, para poder navegar entre las diferentes funcionalidades de la aplicación. En la *Figura 21* podemos ver el aspecto de los dos menús desplegables, el de alumno y el de profesor.

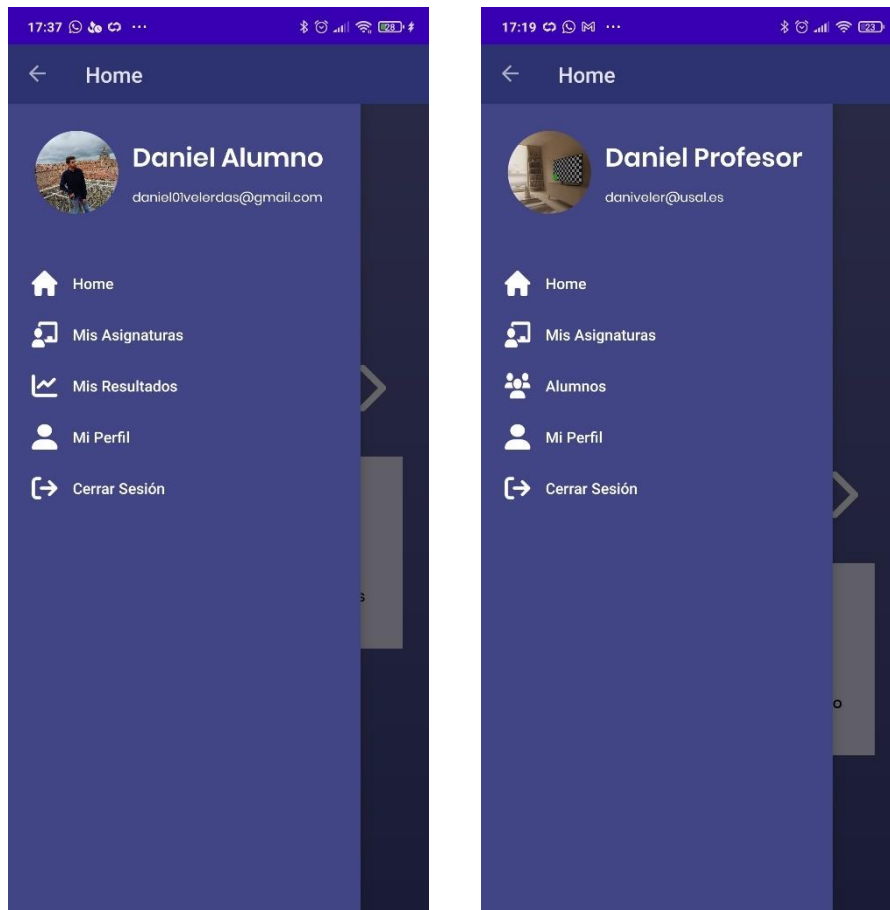


Figura 21. Menús desplegables del menú principal de alumno y profesor

6.4.3 Alumnos

Esta funcionalidad es propia de los profesores. Servirá para observar un listado de los alumnos registrados en la aplicación, como podemos ver en la *Figura 22*, desde el que se podrá acceder además a las diferentes asignaturas en las que cada alumno se encuentra matriculado, permitiendo desmatricularles.

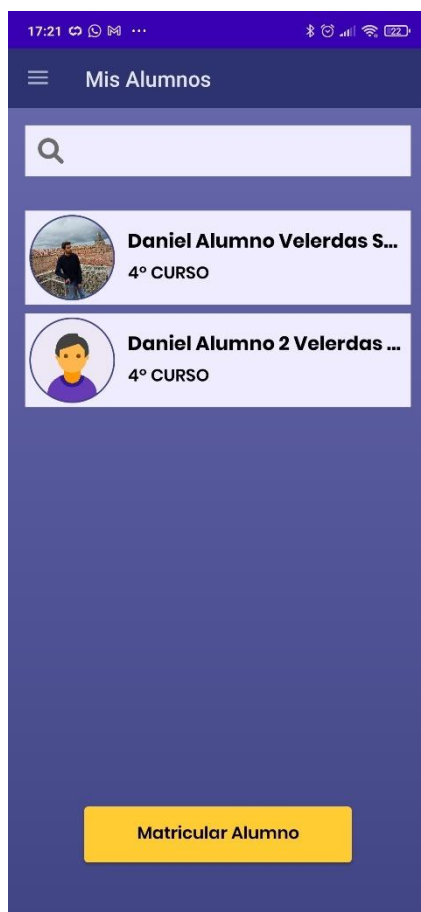


Figura 22. Mis Alumnos

Además, se nos permitirá matricular a un nuevo alumno en una de las asignaturas del profesor. Para ello, la aplicación generará un código QR como el que se puede ver en la *Figura 23* para que el alumno lo escanee desde su aplicación.



Figura 23. Vista de detalle de asignatura para matricular un alumno

6.4.4 Asignaturas

Este apartado puede ser accedido tanto por alumnos como por profesores.

Los alumnos, verán en este apartado las asignaturas en las que se encuentran matriculados en forma de lista, permitiendo acceder a cada una de ellas de forma individual, para ver tanto las encuestas como los resultados vinculados a la asignatura, además de matricular al alumno en una asignatura nueva, escaneando el QR de la asignatura deseada. La vista de listado y la de detalles pueden verse en la *Figura 24*.

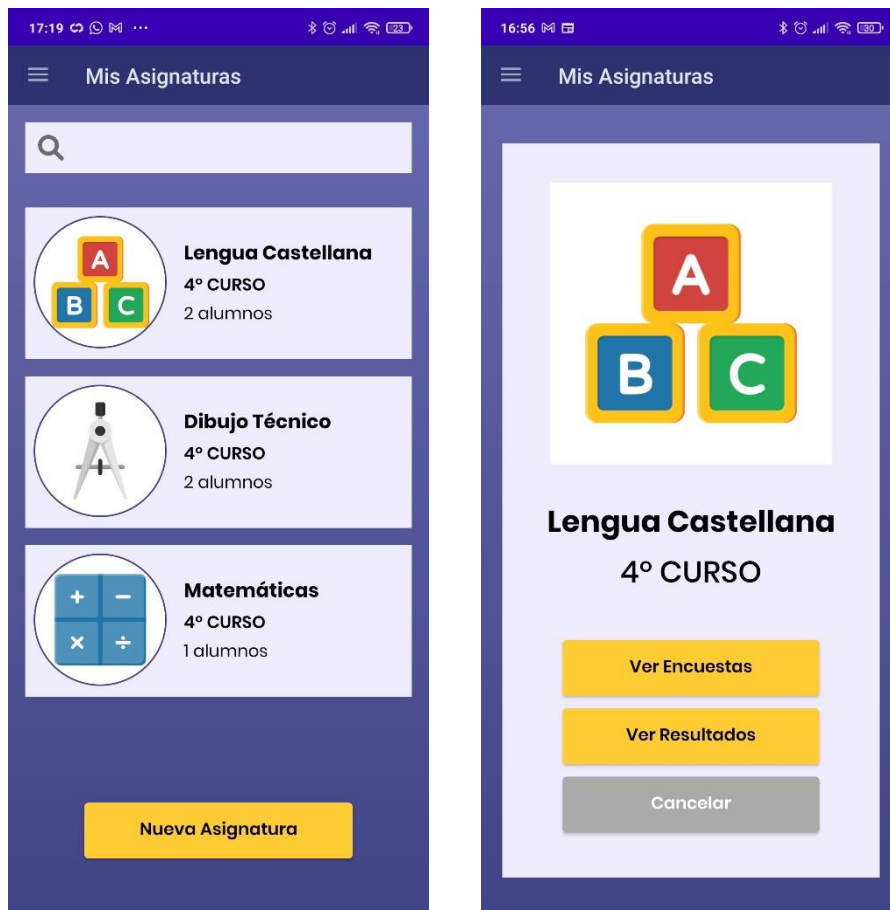


Figura 24. Vistas de listado y detallada de asignaturas del alumno

En el caso del profesor, en esta vista se incluirán las distintas asignaturas impartidas por él en forma de listado, permitiendo consultar cada una de ellas para editarlas, eliminarlas o ver las encuestas que contiene. Además, se permitirá al profesor crear nuevas asignaturas.

6.4.5 Encuestas y preguntas

De manera análoga a las asignaturas, ofrece una vista de listado de las encuestas, como el mostrado en la *Figura 25*.



Figura 25. Vista de listado de encuestas del alumno

El profesor podrá crear encuestas nuevas, modificar o eliminar las existentes, además de incluir diferentes preguntas dentro de cada encuesta. Estas preguntas también podrán ser consultadas, modificadas y eliminadas.

El alumno, por su parte, podrá consultar las diferentes encuestas creadas para una asignatura en la que se encuentre matriculado, además de poder resolverlas, como se puede ver en la *Figura 26*.

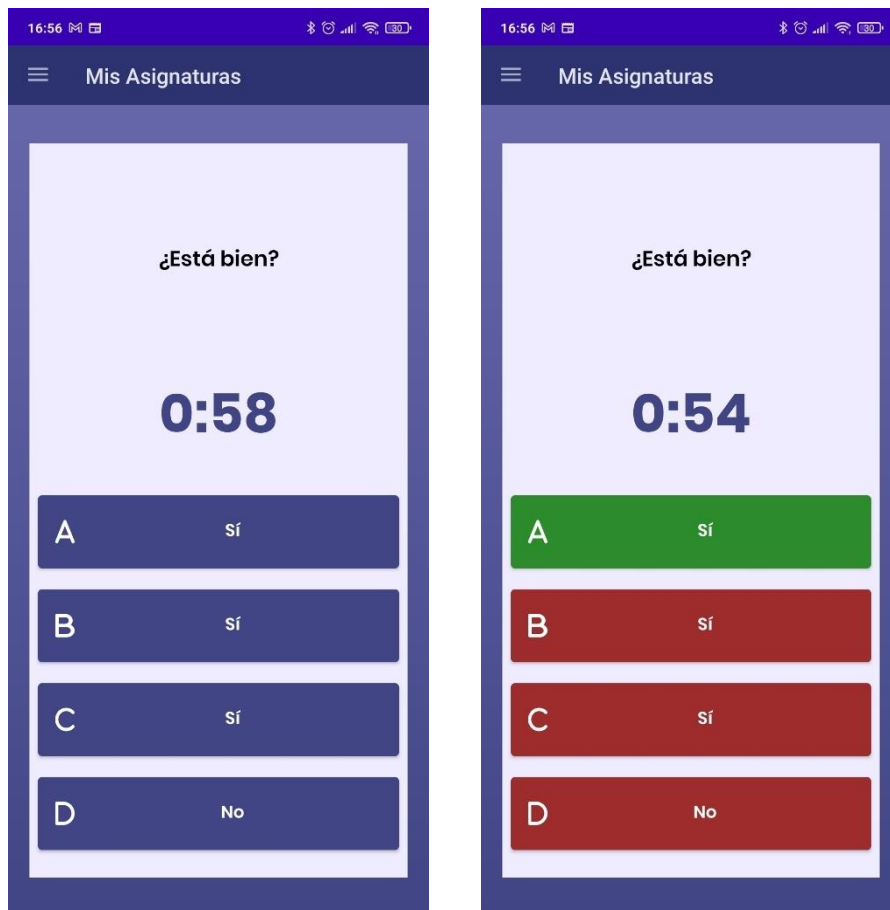


Figura 26. Resolver encuesta

6.4.6 Resultados

Esta vista permitirá a los alumnos ver los resultados obtenidos por ellos al resolver las encuestas. El alumno no podrá modificar ni eliminar ninguno de estos. Esta información se mostrará al alumno en forma de listado, como en la *Figura 27*.

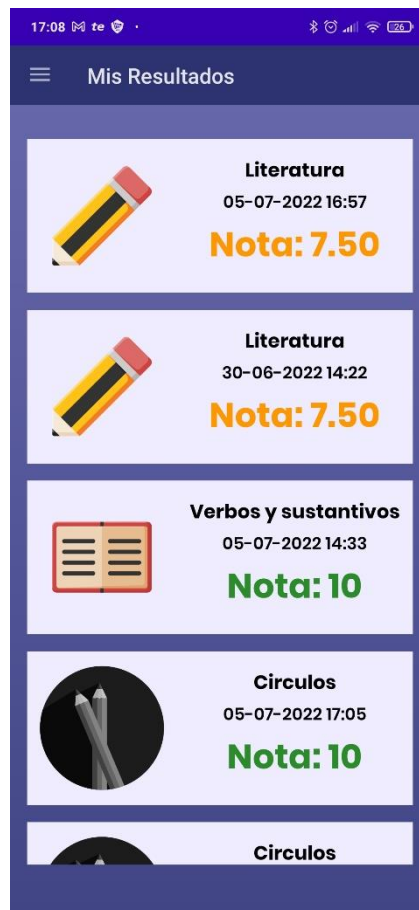


Figura 27. Mis Resultados Alumno

En el caso de los profesores, se mostrarán los resultados obtenidos por los alumnos que hayan resuelto una encuesta en una de sus asignaturas. Los profesores tendrán la opción de eliminar un resultado. Estos resultados serán mostrados de manera similar a como ocurría con los alumnos, como podemos ver en la *Figura 28*.



Figura 28. Ver resultados de una encuesta Profesor

6.4.7 Mi Perfil

Por último, cada usuario podrá tanto consultar como modificar sus datos personales, incluida su contraseña de acceso a la aplicación. Podemos ver estos datos en la *Figura 29*.

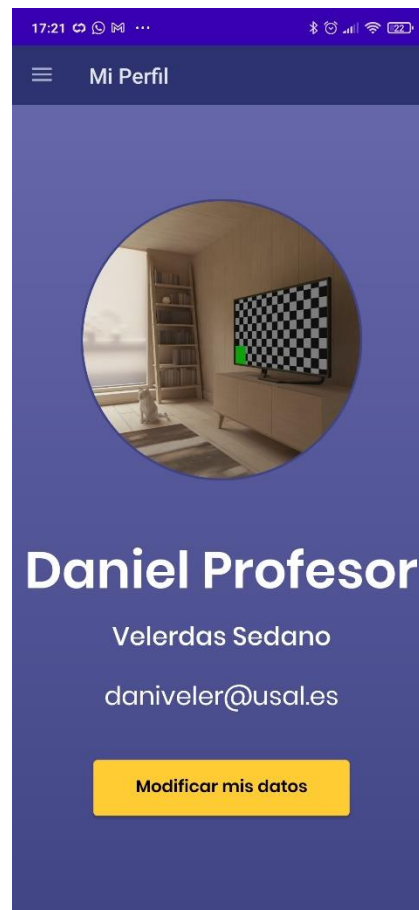


Figura 29. Mi perfil Alumno y Profesor

7. CONCLUSIONES

En este apartado, evaluaremos el cumplimiento de los objetivos marcados en el apartado 2. *Objetivos*, así como las conclusiones obtenidas tras el desarrollo de este Trabajo de Fin de Grado.

- La aplicación cuenta con una correcta gestión de usuarios, que permite a los mismos crear cuentas nuevas o acceder a las ya existentes, diferenciando entre los dos roles: alumno y profesor, cada uno con sus funcionalidades propias. Los usuarios pueden ver y modificar su perfil, tanto su nombre y apellidos como su curso o foto de perfil, que puede ser elegida desde la galería del dispositivo.
- El sistema es capaz de permitir la creación y gestión de diferentes asignaturas por parte de los profesores, así como de regular la correcta interacción entre estas y los usuarios. Los usuarios autorizados, en este caso los profesores, pueden modificar los datos de cada una de estas asignaturas, como el nombre, el curso o el icono.
- Nuestro sistema permite de igual manera, la creación y gestión de encuestas y preguntas por parte de los profesores, moderando también la interacción de estas con los usuarios, así como la generación de resultados a partir de la resolución de las mismas por parte de los alumnos y la gestión de estos por parte de los profesores. Los profesores también podrán modificar los datos de las encuestas, como su nombre o las preguntas que contienen, así como también podrán eliminar resultados generados por los alumnos.
- La aplicación resultado de este trabajo, es una aplicación fácil de usar, responsiva en función del dispositivo empleado, fácilmente mantenible a lo largo del tiempo y que permite la concurrencia de usuarios sin ningún tipo de problema. La base de datos es consistente, rápida y eficiente, por lo que la información se almacena de manera segura y los accesos a la misma son casi inmediatos.

En cuanto a mis objetivos personales, gracias a la realización de este proyecto, he obtenido una base bastante sólida sobre el ámbito del desarrollo móvil, así como también he profundizado en la aplicación del Proceso Unificado durante el desarrollo de software. Todo lo trabajado para la realización de este trabajo ha servido como complemento a la formación adquirida a lo largo del Grado, ampliando aún más mis conocimientos en este campo.

Una vez finalizado, hemos cumplido los objetivos que nos propusimos al inicio de este Trabajo de Fin de Grado y gracias a las herramientas utilizadas y al desarrollo siguiendo la metodología del Proceso Unificado, hemos obtenido como resultado una aplicación plenamente funcional, escalable y eficaz que perfectamente podría ser utilizada en el entorno del aula por alumnos y profesores como una herramienta más en el desarrollo de las actividades lectivas, el cual ha sido nuestro principal objetivo a lo largo de todo el desarrollo.

8. LÍNEAS DE TRABAJO FUTURAS

Partiendo del sistema resultado del proceso de desarrollo llevado a cabo en este Trabajo de Fin de Grado, podríamos pensar en diferentes posibilidades que permitirían mejorar el mismo, de cara a que este fuera convertido en una aplicación real.

- **Realizar el desarrollo en diferentes plataformas.** La primera y más evidente mejora, podría ser la implementación de nuestra aplicación en otras plataformas aparte de Android, como puede ser el sistema operativo iOS o su adaptación como aplicación web, accesible desde cualquier dispositivo. De esta manera, el número de posibles usuarios aumentaría notablemente.
- **Añadir nuevas funcionalidades.** En este apartado, mencionaremos diversas funcionalidades que pueden aportar más posibilidades a nuestra aplicación.
 - Adición de nuevos tipos de preguntas en cada encuesta, como por ejemplo, preguntas a desarrollar, respuestas de elección múltiple, cuestiones relacionadas con imágenes.
 - Opción de añadir diferentes tipos de evaluación a cada encuesta, pudiendo seleccionar si cada respuesta incorrecta en una pregunta conllevara una penalización y de cuántos puntos sería la misma.
 - Inclusión de un sistema de notificaciones, que avise a los usuarios cuando un profesor publique una nueva encuesta o cuando un alumno genere un resultado, entre otros.

Todas estas mejoras, dotarían al sistema de una mayor funcionalidad, preparando a la aplicación para una posible publicación como producto real.

9. BIBLIOGRAFÍA

- Amazon Web Services. (s. f.). *Bases de datos no relacionales / Bases de datos de gráficos / AWS*. Amazon Web Services, Inc. Recuperado 23 de julio de 2022, de <https://aws.amazon.com/es/nosql/>
- diagrams.net. (s. f.). *Diagram Software and Flowchart Maker*. Recuperado 27 de julio de 2022, de <https://www.diagrams.net/>
- Digital55. (2020, mayo 17). Qué es Firebase: Funcionalidades, ventajas y conclusiones. *DIGITAL55*. <https://d55.lextrendlabs.com/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- Dodenhof, H. (2022). *CircleImageView* [Java]. <https://github.com/hdodenhof/CircleImageView> (Original work published 2014)
- Firebase. (s. f.-a). *Cloud Firestore | Firebase Documentation*. Firebase. Recuperado 26 de julio de 2022, de <https://firebase.google.com/docs/firestore?hl=es-419>
- Firebase. (s. f.-b). *Cloud Storage para Firebase | Firebase Documentation*. Firebase. Recuperado 26 de julio de 2022, de <https://firebase.google.com/docs/storage?hl=es-419>
- Firebase. (s. f.-c). *Firebase Authentication | Firebase Documentation*. Recuperado 26 de julio de 2022, de <https://firebase.google.com/docs/auth?hl=es-419>
- Fontawesome. (s. f.). *Font Awesome*. Recuperado 27 de julio de 2022, de <https://fontawesome.com>
- Iconfinder. (s. f.). *6,625,000+ free and premium vector icons, illustrations and 3D illustrations*. Iconfinder. Recuperado 27 de julio de 2022, de <https://www.iconfinder.com/>
- JourneyApps. (2022). *ZXing Android Embedded* [Java]. JourneyApps. <https://github.com/journeyapps/zxing-android-embedded> (Original work published 2012)
- Kotlin. (2022). *Dokka* [Kotlin]. Kotlin. <https://github.com/Kotlin/dokka> (Original work published 2014)
- Looka. (s. f.). *Free Logo Maker & Intelligent Brand Designer*. Looka. Recuperado 27 de julio de 2022, de <https://looka.com/>
- Mena Roa, M. (2021, agosto 30). *Infografía: Android e iOS dominan el mercado de los smartphones*. Statista Infografías. <https://es.statista.com/grafico/18920/cuota-de-mercado-mundial-de-smartphones-por-sistema-operativo/>
- Microsoft. (s. f.). *Software de administración de proyectos | Microsoft Project*. Recuperado 27 de julio de 2022, de <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>
- Mozilla. (s. f.). *Introducción a XML - XML: Extensible Markup Language | MDN*. Recuperado 27 de julio de 2022, de https://developer.mozilla.org/es/docs/Web/XML/XML_introduction

- Red Hat. (2020, junio 10). *¿Qué es un SDK?* Red Hat. <https://www.redhat.com/es/topics/cloud-native-apps/what-is-SDK>
- square. (2022). *Picasso* [Kotlin]. Square. <https://github.com/square/picasso> (Original work published 2013)
- Visual Paradigm. (s. f.). *Visual Paradigm Product Overview*. Recuperado 27 de julio de 2022, de https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html
- Wikipedia. (2021a). Biblioteca (informática). En *Wikipedia, la enciclopedia libre*. [https://es.wikipedia.org/w/index.php?title=Biblioteca_\(inform%C3%A1tica\)&oldid=138652726](https://es.wikipedia.org/w/index.php?title=Biblioteca_(inform%C3%A1tica)&oldid=138652726)
- Wikipedia. (2021b). Kotlin (lenguaje de programación). En *Wikipedia, la enciclopedia libre*. [https://es.wikipedia.org/w/index.php?title=Kotlin_\(lenguaje_de_programaci%C3%B3n\)&oldid=140469342](https://es.wikipedia.org/w/index.php?title=Kotlin_(lenguaje_de_programaci%C3%B3n)&oldid=140469342)
- Wikipedia. (2022). Android SDK. En *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Android_SDK&oldid=144839864