

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de
telecomunicación**

TRABAJO FIN DE GRADO

**APLICACIÓN ANDROID PARA PREVENCIÓN DE ACOSO
ESCOLAR**

**María Alejandra González Aranguren
Tutor: Eduardo Boemo Scalvinoni**

JULIO 2020

APLICACIÓN ANDROID PARA PREVENCIÓN DE ACOSO ESCOLAR

AUTOR: María Alejandra González Aranguren
TUTOR: Eduardo Boemo Scalvinoni

DSLabs
Escuela Politécnica Superior
Universidad Autónoma de Madrid
julio de 2020

Resumen (castellano)

Este Trabajo de Fin de Grado surge de la colaboración entre DSLabs y el Departamento de Psicología de la Universidad Complutense con el objetivo de desarrollar una aplicación que sea posteriormente probada y utilizada en colegios e institutos del sur de la Comunidad de Madrid.

El proyecto consiste en el desarrollo de dos aplicaciones para dispositivos móviles con Sistema Operativo Android cuyo objetivo principal es permitir la comunicación entre alumno y profesor para reportar situaciones de acoso escolar y facilitar su detección temprana.

La aplicación del alumno permite al usuario registrarse e informar a sus profesores de situaciones irregulares en el ámbito escolar, para lo cual se le pide rellenar una serie de cuestionarios con datos relevantes para la valoración de su caso y por último se le permite enviar un mensaje explicando la situación. Esta aplicación permite al usuario comunicarse de forma anónima con el equipo docente, para evitar así, que los mismos se rehúsen al uso de la aplicación por miedo a que se revele su identidad si comunican un caso de acoso contra su persona u otro compañero.

Por otro lado, la aplicación del profesor, es la encargada de la recepción de mensajes. Tiene un formato similar al de una bandeja de correos desde la cual el docente tiene acceso a los datos enviados por cada uno de sus alumnos de forma anónima o con sus datos personales, según lo que hayan elegido cada uno de ellos, y con el acceso al detalle, puede revisar todas las respuestas del alumno a las preguntas de carácter emocional y anímico.

Abstract (English)

This Bachelor Thesis starts as a result of cooperation between DSLAbs and the Department of Psychology of Universidad Complutense with the main goal being developing an app that will be tested and used in different high schools in the city of Madrid.

The Thesis consists on the development of two apps for Android devices which main goal is to allow communication between students and teachers to report bullying at schools in an easy way so that it helps rapidly detecting the situation and solving the problem.

The app developed for the students usage allows them to contact their teachers and report abnormal situations. To be able to send a message the student has to answer to a survey first. This survey will provide teachers vital information that will help them evaluate each student's case.

The app developed for teachers is the one in charge of receiving the messages. It has an email inbox kind of style where the teacher is able to see all of the messages from anonymously registered students.

Palabras clave (castellano)

Android, aplicación, acoso escolar, aplicación de mensajería, base de datos

Keywords (inglés)

Android, app, bullying, smartphone, messaging app, database

Agradecimientos

A mis padres,

A mi tutor Eduardo y a Álvaro por confiarme este proyecto,

Y a mis compañeros Diego y Kia.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	2
1.3	Fases del proyecto	2
1.4	Estructura de la memoria.....	3
2	Estado del arte	5
2.1	Aplicaciones similares existentes	5
2.1.1	Stop Bullying: (NO al acoso escolar)	5
2.1.2	Stop Acoso.....	6
2.1.3	Bullying No Mas!	6
2.1.4	Conclusiones.....	6
3	Análisis	9
3.1	Análisis de requisitos.....	9
3.1.1	Requisitos funcionales	9
	Requisitos no funcionales.....	13
3.1.2	Sistemas Operativos compatibles	13
3.1.3	Metodología de desarrollo	13
4	Diseño.....	15
4.1	Objetivos de diseño	15
4.2	Tecnologías empleadas.....	15
4.2.1	Android.....	15
4.2.2	Entorno de desarrollo y lenguaje de programación	16
4.2.3	Firestore.....	16
4.2.4	Versión de Android utilizada para el desarrollo.....	16
4.3	Diagrama de flujo de la aplicación	17
4.4	Diagrama de clases	17
4.4.1	Diagrama de clases de la aplicación del alumno	18
4.4.2	Diagrama de clases de la aplicación de profesores.....	18
4.5	Diseño de vistas	19
4.6	Diseño del logo de la aplicación.....	20
5	Desarrollo	21
5.1	Primeros pasos antes de iniciar el desarrollo.....	21
5.2	Estructura de una aplicación Android	21
5.2.1	Vistas o Views.....	22
5.2.2	Manifest	22
5.2.3	Gradle	22
5.3	Creación del proyecto en Firestore.....	23
5.4	Estructura de datos de la aplicación en Firestore	24
5.4.1	Seguridad de la base de datos de la aplicación	26
5.5	Desarrollo de la aplicación del alumno	26
5.5.1	Registro del alumno.....	26
5.5.2	Autenticación del alumno.....	27
5.5.3	Respuestas a las encuestas y envío del mensaje	27
5.6	Desarrollo de la aplicación del profesor	27
5.6.1	Registro de profesores	28
5.6.2	Autenticación del profesor.....	28
5.6.3	Recogida de datos para el buzón de entrada.....	30

6 Pruebas	31
6.1 Pruebas funcionales	31
6.2 Pruebas de compatibilidad.....	32
7 Conclusiones y trabajo futuro.....	33
7.1 Conclusiones.....	33
7.2 Trabajo futuro	34
Referencias	34
Glosario	35
Anexos.....	I
A Manual de instalación.....	I
B Manual de Usuario	III
	III

INDICE DE FIGURAS

FIGURA 1-1: STOP BULLYING.....	5
FIGURA 1-2: STOP BULLYING.....	5
FIGURA 1-3: STOP BULLYING.....	5
FIGURA 1-4: STOP BULLYING.....	6
FIGURA 1-5: STOP BULLYING.....	6
Figura 4-1: Diagrama de flujo de las aplicaciones.....	17

INDICE DE TABLAS

1 Introducción

En este apartado se explica la motivación por el desarrollo de la aplicación BullyStop como Trabajo de Fin de Grado.

1.1 Motivación

Vivimos en una época marcada por el auge del sector de las telecomunicaciones y de las nuevas tecnologías. El uso de dispositivos electrónicos inteligentes está cada día más extendido y su demanda sigue creciendo de forma exponencial.

Con la entrada al mercado de estos aparatos, aparecieron las primeras aplicaciones, que transformaron radicalmente el concepto que se tenía hasta entonces de teléfono móvil y lo convirtieron en una herramienta muchísimo más potente. Las primeras aplicaciones, tenían un enfoque empresarial; sin embargo, evolucionaron rápidamente y empezaron a estar dirigidas a más sectores conforme se expandió el uso de estos dispositivos, intentando responder a las distintas necesidades y demandas de la sociedad.

La expansión tecnológica ha sido tal, que, en la actualidad, estos dispositivos y aplicaciones se han incorporado a diversos sectores, tales como: sanidad, hostelería y educación.

Intentando responder a una necesidad de este último sector, surge este proyecto. El acoso escolar es un problema que siempre ha existido; sin embargo, en los últimos años ha adquirido una mayor visibilidad, y se debe, en parte, al extendido uso que tienen los *smartphones* y las redes sociales.

Si bien, antes el acoso era un problema que difícilmente salía del colegio, a día de hoy, el hostigamiento puede extenderse más allá de las aulas. Estudios recientes demuestran que la mayoría de niños obtiene su primer teléfono móvil inteligente a edades de entre once y doce años, lo que les da acceso a una enorme cantidad de contenido en la red y les hace partícipes de las populares redes sociales.

En la era del internet, han surgido nuevas formas de acoso *online*, que recientemente han sido denominadas como *cyberbullying*. Este tipo de acoso afecta a personas de distintas edades y es algo prácticamente imposible de controlar en un espacio libre y “sin ley” como la red.

En los últimos años se ha tomado mayor consciencia de este problema social y se han exigido soluciones. Diversos centros escolares en distintos lugares del mundo han tomado cartas en el asunto y han aprovechado la tecnología para desarrollar su propia aplicación de seguimiento a los alumnos. Sin embargo, son pocas las aplicaciones que aporten una solución generalizada a la que se puedan sumar otras organizaciones.

Es por esto que se pretende desarrollar una aplicación de carácter más general que permita a distintos centros escolares hacer un seguimiento de sus alumnos y conocer en detalle la situación para, así, poder actuar y tomar las medidas apropiadas.

Referencia [1]

1.2 Objetivos

El objetivo principal del trabajo es dar una respuesta a la necesidad de los centros escolares de tener algún tipo de herramienta que les sirva de control y seguimiento para posibles casos de acoso escolar, además de servir de módulo de comunicación entre profesorado y alumnado.

Pretende también garantizar el anonimato del alumno que reporta la situación irregular, en caso de que así lo desee, sin que pueda dejar de ser rastreable por el profesor para la resolución del problema.

1.3 Fases del proyecto

Para poder realizar una estimación temporal realista del desarrollo de este Trabajo de Fin de Grado es necesario definir las distintas fases que marcaron su realización de principio a fin.

1. Investigación

Etapa de aprendizaje. Búsqueda de información sobre las herramientas disponibles para la realización del proyecto. Adquirir conocimientos y experiencia en desarrollo de aplicaciones Android. Hacer pruebas pequeñas con el entorno de desarrollo.

2. Análisis

Establecimiento de una serie de requisitos por parte del cliente o, en este caso, departamento que solicita el desarrollo de la aplicación.

3. Diseño

Elegir cómo será la arquitectura de la aplicación para cumplir todos los objetivos y funcionalidades propuestas. Escoger las tecnologías a utilizar en el desarrollo.

4. Desarrollo

Implementación de los módulos del proyecto siguiendo la estructura elegida en el diseño de manera de que cumpla los requisitos y especificaciones y logre la comunicación entre la aplicación del alumno y del profesor.

5. Pruebas

Una vez implementado el código, verificar que todo funciona correctamente y en caso de que algo no funcione como es debido solventar el error de programación.

6. Documentación

Recopilar la documentación generada a lo largo de las distintas fases.

Una vez definidas las seis fases principales de trabajo, se estima la duración total.

Fases	Estimación temporal
<u>Investigación</u>	58 días
-Búsqueda de entorno de desarrollo	1 día
-Instalación de entorno de desarrollo	1 día
-Comprensión del funcionamiento del sistema operativo Android	2 días
-Adquirir conocimientos de desarrollo de aplicaciones Android	30 días
- Adquirir conocimiento sobre tecnologías a utilizar	15 días
-Hacer pequeñas pruebas de desarrollo	10 días
<u>-Análisis</u>	1 día
-Definir requisitos	1 día
<u>Diseño</u>	30 días
-Arquitectura completa de los dos módulos	10 días
-Diseñar la estructura de nodos en base de datos	10 días
-Diseño de pantallas	10 días
<u>Desarrollo</u>	51 días
-Desarrollo de las funcionalidades de la aplicación de alumnos	25 días
-Desarrollo de las funcionalidades de la aplicación de profesores	25 días
-Seguridad de base de datos	1 día
<u>Pruebas</u>	20 días
-Pruebas de casos de uso	15 días
-Solventar pequeños fallos	5 días
<u>Documentación</u>	10 días
Trabajo de Fin de Grado	170 días

1.4 Estructura de la memoria

En este apartado se expondrá la estructura del documento.

- Estado del Arte: en este capítulo se analizarán las aplicaciones que aborden el tema de acoso escolar existentes y disponibles en la tienda de aplicaciones de Google
- Análisis: estudio de los requisitos funcionales y no funcionales de la aplicación.
- Diseño: en este apartado se resolverá la arquitectura global del proyecto en función de sus requisitos.
- Desarrollo: se detallan los pasos seguidos para elaborar la aplicación.

- Pruebas: en este apartado se detallan las distintas pruebas realizadas para garantizar el cumplimiento de todos los casos de uso de la aplicación, así como la compatibilidad de la misma en dispositivos de distintas resoluciones y generaciones.
- Conclusiones: epílogo del documento y valoración del aporte de la aplicación y de los conocimientos adquiridos durante su desarrollo.
- Trabajo futuro: exposición de posibles mejoras de desarrollo y de agregar nuevas funcionalidades

2 Estado del arte

2.1 Aplicaciones similares existentes

Es sabido que el acoso escolar siempre ha existido; sin embargo, en los últimos años se le ha dado una mayor visibilidad a este problema, en gran medida debido a la llegada de las redes sociales a la vida de las personas. A las situaciones irregulares de maltrato físico y psicológico en el ámbito estudiantil se han sumado nuevas formas de hostigamiento como el ciberacoso, también conocido como *cyberbullying*.

Buscando ayudar a responder a este problema, han sido desarrolladas diversas aplicaciones que podemos encontrar en la *Play store* de Google.

Las aplicaciones disponibles presentan distintos enfoques del tema, algunas abordan la parte educativa, con recursos multimedia y textos que explican qué situaciones pueden ser consideradas acoso y deben ser denunciadas, mientras que otras, se enfocan en el seguimiento de este problema mediante encuestas y registros de datos de los alumnos.

2.1.1 Stop Bullying: (NO al acoso escolar)

Disponible en la tienda de aplicaciones de Android tiene carácter educativo. Las primeras pantallas muestran contenido multimedia que sirve de material explicativo para presentar la definición del acoso. Presenta también un chat con horario limitado para pedir ayuda.



Figura 2-1: Stop Bullying



Figura 2-2: Stop Bullying



Figura 2-3: Stop Bullying

2.1.2 Stop Acoso

Otra aplicación disponible, está registrada bajo el nombre **Stop Acoso** y su función principal es el seguimiento y detección de casos de acoso escolar para un centro ubicado en la ciudad de Badajoz. La aplicación es bastante completa, cuenta con un menú principal con varias opciones: acceso para padres, acceso al portal web, y una opción de acceso a reporte. Se adjunta a continuación la captura de pantalla disponible en la tienda.



Figura 2-4: Stop Acoso

2.1.3 Bullying No Mas!

Esta aplicación tiene un carácter didáctico. Presenta un menú principal con distintos apartados que explican qué es el acoso y los tipos que existen.



Figura 2-5: Stop Acoso

2.1.4 Conclusiones

Sin embargo, tras investigar las aplicaciones existentes, se observa que hay pocas opciones disponibles para abordar el reporte de acoso escolar, ya que muchas de ellas son meramente

informativas, y otras están desarrolladas para centros educativos específicos, impidiendo a otros centros sumarse al uso de estas aplicaciones.

3 Análisis

3.1 Análisis de requisitos

Antes de iniciar cualquier desarrollo de software, se debe realizar un análisis para establecer la estructura que tendrá la aplicación en base a las funcionalidades acordadas entre el desarrollador y cliente. Para la toma de decisiones en la elaboración de esta aplicación de reporte de acoso escolar, se establecieron una serie de requisitos funcionales que marcaron el planteamiento del desarrollo y arquitectura de la aplicación.

3.1.1 Requisitos funcionales

Al comprender este proyecto dos aplicaciones separadas, se enunciarán a continuación los requisitos para cada una de ellas.

En primer lugar, para la aplicación orientada al alumno, se estipuló lo enunciado a continuación:

1. Registro de usuarios de tipo alumno:

- La aplicación deberá solicitar un nombre de usuario y contraseña para que el alumno se registre en la aplicación de reporte.
- Dicho nombre de usuario debe tener formato de correo electrónico, aunque no es necesario su autenticidad fuera de la aplicación.
- Se validará que los datos de nombre de usuario cumplan con los requisitos.
- En caso de que la validación sea correcta se guardarán los datos en base de datos y se accederá a la siguiente pantalla.
- En caso de error, se mostrará un mensaje por pantalla.

2. Identificación:

- Para volver a acceder a la aplicación, el alumno debe introducir sus credenciales.
- El sistema valida los datos de autenticación.
- En caso correcto se accede a la aplicación.
- En caso de error se muestra un mensaje por pantalla.

3. Datos de destinatario:

- La aplicación deberá solicitar el email de destino del mensaje, correspondiéndose este con el de su profesor.
- El sistema valida el formato de los datos y se continúa a la siguiente pantalla en caso de éxito.
- En caso de que el alumno no introduzca algún dato o se produzca un error, mostrar un mensaje por pantalla.

4. Cuestionarios de sentimientos, depresión y acoso

- En las sucesivas pantallas posteriores se mostrarán una serie de cuestionarios a los cuales el alumno debe responder marcando una de las opciones, (siendo todas las preguntas obligatorias).
- Se valida la selección de un botón para todas las respuestas presentadas por pantalla.
- En caso de éxito, continuar a la pantalla de envío de mensaje.
- En caso de error, mostrar un mensaje por pantalla.

5. Envío de mensaje

- Por último, el usuario debe rellenar un campo de texto con la explicación detallada de su situación personal.
- Se solicita responder a cuestionarios adicionales complementarios a las preguntas del apartado anterior.
- El sistema valida los datos.
- En caso de éxito se muestra una última pantalla con un mensaje de agradecimiento al alumno y la opción de enviar otro mensaje.

Por otro lado, para la aplicación enfocada a profesores, encargada de recibir los mensajes del alumno, se estipuló:

1. Registro de usuarios de tipo profesor:

- La aplicación deberá solicitar un nombre de usuario y contraseña para que el profesor se registre en la aplicación de reporte.
- Se enviará un enlace de verificación al correo electrónico.

- Se validará que los datos de nombre de usuario cumplan con los requisitos.
- En caso de que la validación sea correcta se guardarán los datos en base de datos y se accederá a la siguiente pantalla.
- En caso de error, se mostrará un mensaje por pantalla.

2. Identificación:

- Para volver a acceder a la aplicación, el profesor debe introducir sus credenciales.
- El sistema valida los datos de autenticación.
- En caso correcto se accede a la aplicación.
- En caso de error se muestra un mensaje por pantalla.

3. Acceso a bandeja de entrada:

- La aplicación deberá mostrar en la vista los mensajes recibidos en una lista con estilo similar al de una bandeja de entrada de correo electrónico.

4. Visualización del mensaje

- Al pulsar en un mensaje se deberá abrir el detalle del mismo.

Requisitos no funcionales

1. Seguridad:

- Es necesario autenticarse para acceder a la aplicación. En el caso de la aplicación del profesor, se debe verificar el email. En el caso de la aplicación del alumno, puede elegir un nombre de usuario aleatorio, pues la comunicación debe poderse realizar de forma anónima.

2. Interfaz:

- La aplicación debe tener un diseño sencillo y fácil de utilizar por el usuario.
- La aplicación debe implementar la totalidad de los cuestionarios facilitados al desarrollador para una recolección de datos completa

3. Compatibilidad:

- Se debe garantizar la compatibilidad con el mayor número de versiones posibles de Android.
- Garantizar el funcionamiento de la aplicación en pantallas de distintas resoluciones

3.1.2 Sistemas Operativos compatibles

En primer lugar, se plantearon las posibilidades de desarrollar la aplicación para Android o iOS. Sin embargo, se optó por la primera debido a su fuerte presencia en el mercado de dispositivos móviles. Android está presente en aproximadamente un 70% de los *smartphones* del mundo, lo que lo posiciona a una amplia distancia de su rival.

Este sistema operativo también presenta varias características que, para este proyecto concreto, lo hacen más atractivo a la hora de desarrollar. Algunas de ellas son: la disponibilidad de entorno de desarrollo gratuito, su característica de código abierto, y el bajo coste que supone adquirir una licencia de desarrollador.

Sin embargo, la razón que tuvo mayor peso al momento de decidir trabajar con Android fue el interés de alcanzar el mayor público posible, ya que el principal objetivo del proyecto es servir de ayuda en la detección precoz de acoso escolar en colegios de la comunidad de Madrid, pudiendo extenderse su uso, en caso de obtener un recibimiento positivo, a otras comunidades. Debido a los precios más asequibles de los dispositivos que trabajan con el sistema operativo seleccionado, es más probable que en colegios e institutos, la mayoría de alumnos posean un dispositivo compatible.

3.1.3 Metodología de desarrollo

A la hora de estructurar un proyecto de desarrollo debe elegirse un marco sobre el que trabajar para controlar la planificación de las distintas fases del proceso. Existen varios modelos:

- En cascada: sigue los pasos establecidos en el análisis de forma secuencial. Las fases se pueden superponer unas con otras y se realiza toda la planificación del proyecto en tiempos desde el inicio del desarrollo.
- Iterativo: este modelo es bastante flexible, pues permite, como su nombre indica, ver las funcionalidades desarrolladas por partes, sin necesidad de tener un resultado cerrado.
- Desarrollo rápido: se basa en el desarrollo iterativo, pero a diferencia de este se centra en la creación de prototipos.
- Existen otras metodologías de desarrollo, como la espiral e incremental, sin embargo, para el proyecto se optó escoger entre una de las tres mencionadas anteriormente.

Tras valorar las distintas metodologías, teniendo en cuenta la necesidad de un desarrollo ágil, se optó por un desarrollo iterativo.

4 Diseño

4.1 Objetivos de diseño

El objetivo principal es diseñar una aplicación que cumpla todos los requisitos anteriormente enunciados y cuyo uso sea sencillo de cara al usuario final.

Gran parte del trabajo de este proyecto se centra en el tratamiento y almacenamiento de los datos facilitados por el alumno para el estudio y resolución de su caso, así como la autenticación de los usuarios. Los usuarios de tipo alumno deben poder acceder de forma anónima con un nombre de usuario no real, o, en caso de que así lo deseen, con su dirección de correo electrónico real. La idea de esto, es evitar que algunos alumnos se nieguen a registrarse en la aplicación y a utilizarla por miedo a contar la situación y que se sepa su identidad. Mientras que los usuarios de tipo profesor deben registrarse con su dirección de correo electrónico y validarlo mediante un enlace enviado a su bandeja de entrada.

4.2 Tecnologías empleadas

En esta categoría se aporta la información de las tecnologías empleadas para el diseño de la arquitectura y vistas de la aplicación.

4.2.1 Android

Para el desarrollo de este proyecto es necesario estudiar y comprender primero la estructura del Sistema Operativo a utilizar.

La arquitectura de Android está dividida en cinco niveles:

La primera capa, analizando la arquitectura en orden ascendente, comprende el *kernel* de Linux. Esta capa es el núcleo del sistema, y en ella se controlan componentes como la memoria, el CPU, y el hardware.

En la siguiente capa, denominada Hardware Abstraction Layer, se encuentran una serie de módulos que se cargan según el componente de hardware al que se invoque.

Prosigue a esta, una capa compartida por las librerías nativas y ART. El entorno que hace posible ejecutar aplicaciones se denomina Android Runtime, esto lo hace mediante la traducción de código intermedio a código máquina y la ejecución de múltiples máquinas virtuales que se asignan a cada uno de los procesos en ejecución. Aparece con la llegada de la versión de Android 5.0. y sustituye a la antigua instancia de máquina virtual exclusiva de Android llamada Dalvik. Las librerías nativas de las que se vale el sistema están escritas en C y C++ y se compilan en código nativo. Todas estas responden a distintas necesidades, tales como, garantizar la seguridad, brindar soporte a los navegadores web, entre otros.

La capa de *Framework*, ofrece a los desarrolladores de aplicaciones un entorno que facilita y agiliza el proceso de diseño de las mismas, ya que proporciona diversos componentes,

como botones y campos de texto, que pueden ser utilizados y personalizados por el desarrollador.

4.2.2 Entorno de desarrollo y lenguaje de programación

Para el diseño y desarrollo de la aplicación se utilizó el entorno de desarrollo oficial de Google, Android Studio. Se trata de un software gratuito cuya construcción de proyectos se basa en la herramienta Gradle, que, a su vez, se basa en Groovy y DSL para la generación del *build* del proyecto. Soporta las tecnologías de Google Cloud y permite integrar a los proyectos los distintos servicios ofrecidos por Firebase. Permite utilizar distintos lenguajes de programación como Java y Kotlin. Este último es un lenguaje novedoso que promete convertirse en el lenguaje predilecto en el desarrollo de aplicaciones Android. Sin embargo, se decidió usar Java.

4.2.3 Firebase

Para el diseño de la estructura y almacenamiento de datos del proyecto se optó por la integración de Firebase, compatible con el entorno de desarrollo y que permite añadir múltiples proyectos de Android Studio a un único proyecto de Firebase; es decir, que da a ambas aplicaciones el acceso a una base de datos única para compartir información entre ellas.

Firebase ofrece distintos servicios, entre los que destacan:

- **Authentication:** proporciona servicios de *backend* de autenticación recogiendo datos para el registro e inicio de sesión de usuarios tales como, direcciones de correo electrónico, números de teléfono, cuentas de Google o Facebook y contraseñas. A cada usuario registrado les asigna un ID único con el que los identifica.
- **Real-time database:** es la base de datos original de Firebase, permite la sincronización de datos con clientes en tiempo real.
- **Cloud storage:** permite compartir contenido multimedia mediante el almacenamiento de ficheros en la nube. Además, se puede utilizar para la implementación del envío y recepción de mensajes de *push* a dispositivos.
- **Hosting:** permite alojar elementos estáticos de las aplicaciones, tales como HTML, CSS y JavaScript.

Para la elaboración de las aplicaciones del alumno y profesor se utilizaron los servicios de **Firestore Authentication** y **real-time database**.

4.2.4 Versión de Android utilizada para el desarrollo

La versión utilizada para la construcción y desarrollo del proyecto es Android 9, con un nivel de API 28. Es una de las últimas versiones estables publicadas. Incluye novedades y grandes cambios con respecto a la forma de implementación de los componentes en comparación con la versión anterior, pero garantiza el funcionamiento durante un mayor período de tiempo de la aplicación, sin necesidad de sacar una actualización, que versiones más antiguas, pues acaban deprecadas tras numerosas actualizaciones publicadas.

4.3 Diagrama de flujo de la aplicación

Como se ha comentado reiteradas veces, el principal objetivo del diseño es construir una aplicación con una interfaz simple e intuitiva que no cause complicaciones a los usuarios al momento de interactuar con ella, que cumpla los requisitos especificados y cuyo uso pueda ser aplicado al ámbito educativo. La estructura global que va a tener el proyecto para que las aplicaciones de alumno y profesor se comuniquen se ilustra a continuación:

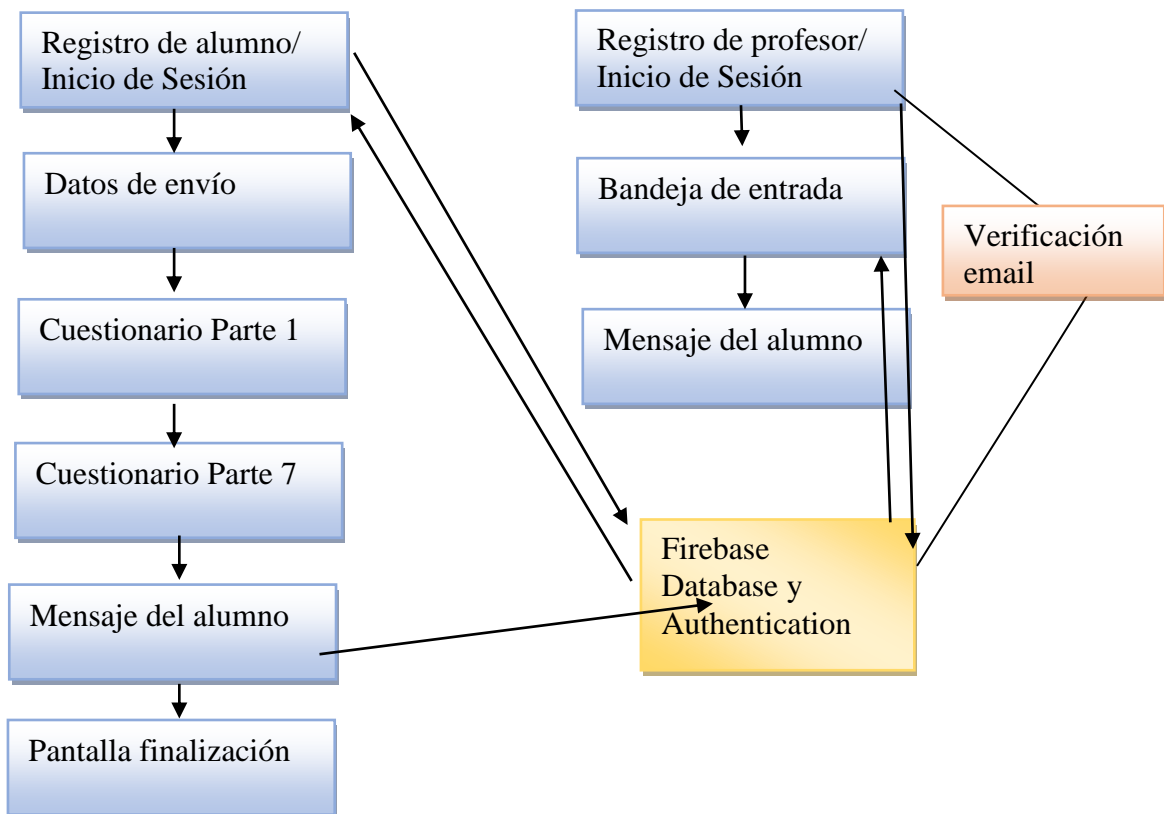
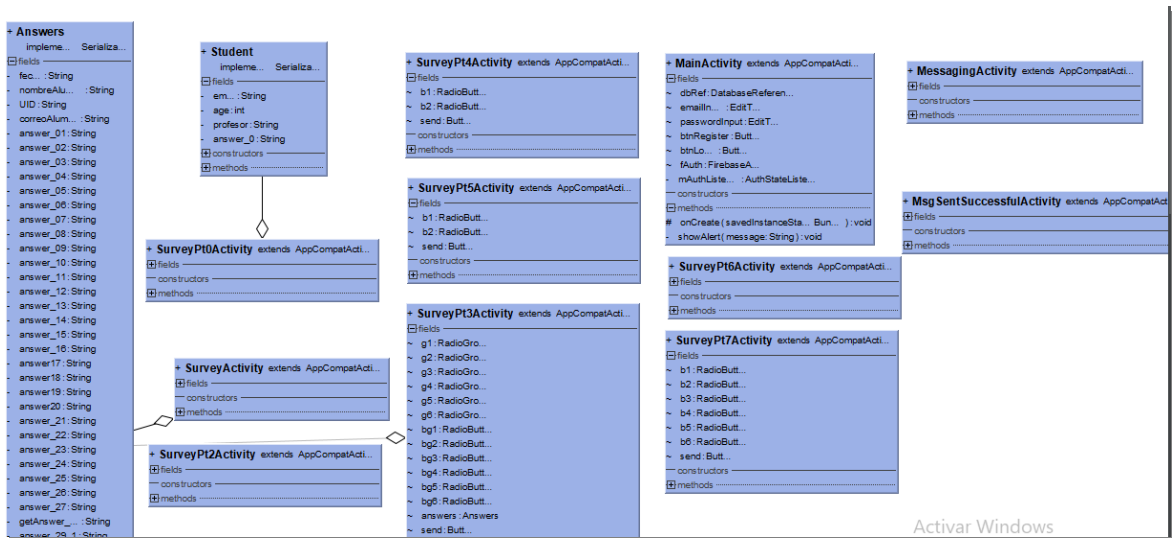


Figura 4-1: Diagrama de flujo de las aplicaciones

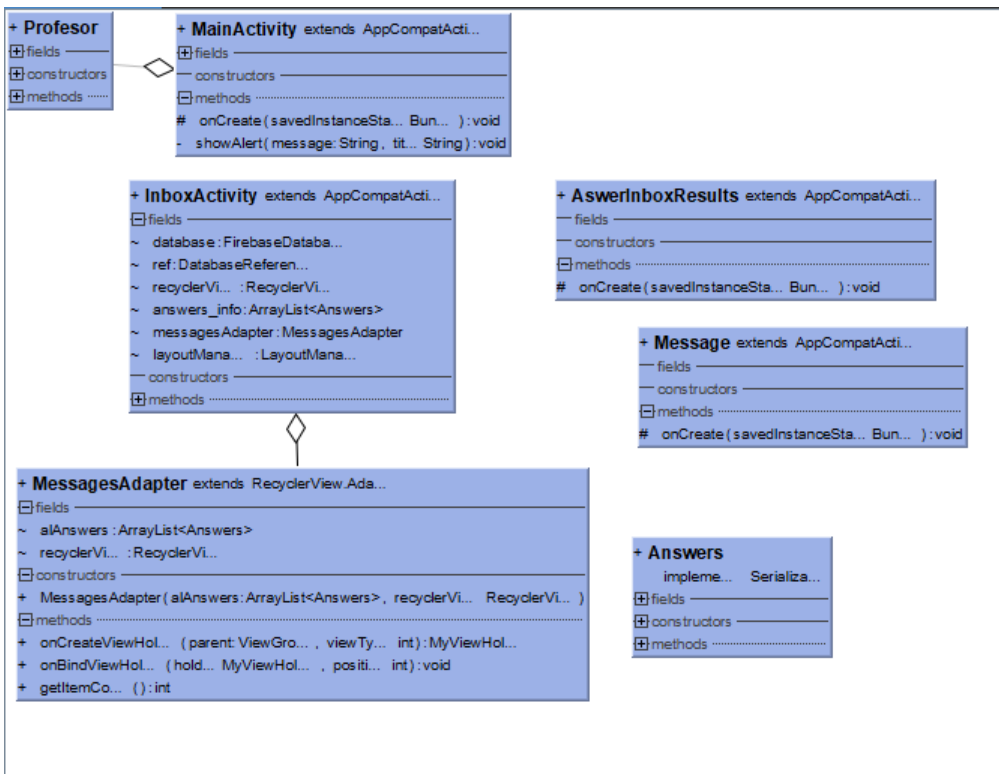
4.4 Diagrama de clases

Con la ayuda de los plugins del entorno de desarrollo se generó un diagrama de clases para cada una de las aplicaciones, en el que se pueden apreciar los métodos de las clases y elementos de las entidades.

4.4.1 Diagrama de clases de la aplicación del alumno



4.4.2 Diagrama de clases de la aplicación de profesores



4.5 Diseño de vistas

La interfaz de usuario tiene un diseño sencillo para que resulte fácil de utilizar, ya que una de las aplicaciones va dirigida a público infantil y adolescente. Las pantallas presentan una información clara al alumno para recopilar tantos datos como sean posibles para que el trabajo de prevención y solución a posibles situaciones de acoso sea más sencillo para el equipo docente.

A continuación, se adjuntan capturas de algunas de las pantallas de **Bully Stop** para los alumnos:

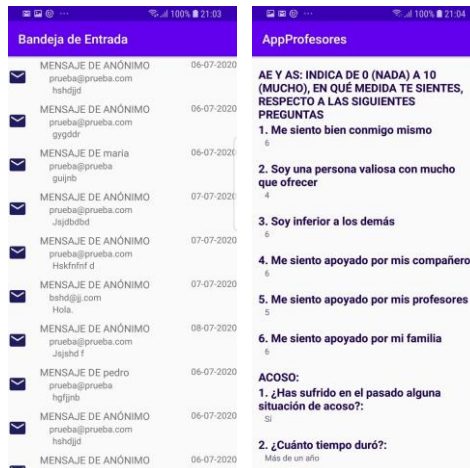
The image displays four screenshots of the Bully Stop app interface for students:

- AppAlumno:** Login screen with fields for Email and Password, and buttons for 'REGISTRARSE' and 'INICIAR SESIÓN'.
- Cuestionario:** Questionnaire form with fields for 'Edad', 'Email del Profesor', and 'Género' (Femenino, Masculino, Otro), followed by a 'SIGUIENTE' button.
- Cuestionario:** Rating scale questionnaire with 11 questions and a scale from 0 to 10.
- Enviar Mensaje:** Screen for sending a message to the teacher, with a text input field for the teacher's name and a list of bullying options to be selected.

Los componentes ofrecidos por Android a utilizar en la aplicación son: campos de texto, *radiobuttons*, *checkbox* y botones.

Para la interfaz del profesor se realizó un diseño con estilo de buzón de correo electrónico, con su correspondiente bandeja de entrada con todos los mensajes dirigidos a su cuenta de Bully Stop y la pantalla de detalle que muestra todas las respuestas marcadas por el alumno en las encuestas además del mensaje explicando la situación ocurrida.

La pantalla de inicio de sesión es igual a la del alumno, aunque funcionalmente realiza otras comprobaciones, por lo que sólo se adjuntan la captura de pantalla del buzón y del correspondiente mensaje seleccionado.



4.6 Diseño del logo de la aplicación

Fueron varios los nombres que se valoraron para la publicación de la aplicación en la tienda de Google. Tras descartar varios, se optó por Bully Stop, nombre corto, fácil de recordar y llamativo.

En función del nombre se realizó un diseño de logotipo que se pudiese identificar y relacionar fácilmente con el tema a tratar en la aplicación.



5 Desarrollo

En este apartado se enuncian y definen los conceptos empleados para desarrollar el proyecto. Este capítulo profundiza en la programación de las aplicaciones desde su creación hasta el resultado final.

5.1 Primeros pasos antes de iniciar el desarrollo

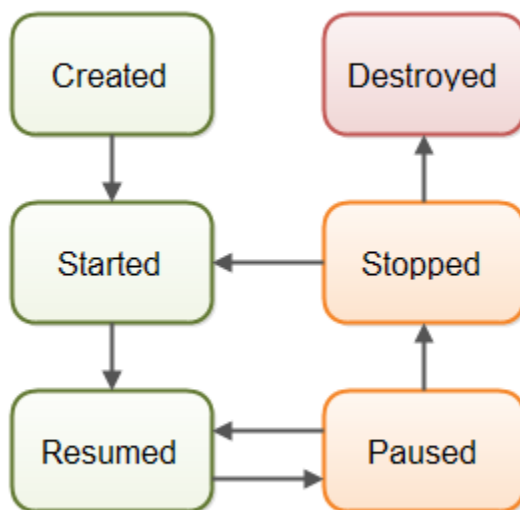
Para el desarrollo de este trabajo, primero, fue necesario adquirir conocimientos de carácter técnico y específico de programación de aplicaciones Android. El proceso de aprendizaje constó de varias fases:

- Breve investigación del funcionamiento del sistema operativo
- Estudio y familiarización con Android y Java
- Estudio del funcionamiento de Firebase
- Prácticas con tutoriales
- Pruebas de desarrollo y de carga y lectura de datos de Firebase

5.2 Estructura de una aplicación Android

Las aplicaciones Android tienen cuatro componentes fundamentales, pero en el desarrollo de esta aplicación se va a utilizar uno de ellos, y es el componente actividad.

- Actividades: cada pantalla de la interfaz de usuario consta de una actividad. En ella, se escribe toda la lógica de la pantalla en el lenguaje elegido, en el caso de esta aplicación, Java, y su ciclo de vida queda definido seis estados.



Los distintos estados de las actividades se guardan con el formato clave, valor en la aplicación y van cambiando en función de la operación que se esté haciendo. No solamente los cambios de actividad o pantalla producen un cambio de estado, sino también las configuraciones de los dispositivos sobre los que se ejecutan. Un ejemplo práctico de cambio de estado de una actividad debido a la configuración, es el que se produce al rotar la pantalla del dispositivo. Las actividades hacen referencia a un xml que es el que contiene los componentes, que deben ser recogido en el *back*, y el estilo de la vista.

Los otros tres componentes fundamentales de una aplicación Android son:

- **Servicios:** permiten realizar operaciones cuando la aplicación está tanto en primer plano, como en segundo. Las primeras son visibles para el usuario, mientras que las últimas no lo son.
- **Proveedores de contenido:** se encarga de realizar la administración de acceso al repositorio con el contenido.
- **Receptor *broadcast*:** permite el envío y recibir mensajes desde el sistema de Android, se utiliza por ejemplo para el desarrollo de chats avanzados. Permite la actualización de datos en segundo plano.

Dichos componentes se mencionan, para que se tenga una visión completa de los elementos que pueden formar una aplicación, aunque en este proyecto su uso no es necesario.

5.2.1 Vistas o Views

La vista simboliza un componente rectangular dentro de la pantalla y contiene todos los componentes a utilizar y recoger en el *back*. Las vistas de Android se estructuran con los llamados *Layouts* que nos permiten establecer el orden de los componentes, su ajuste dentro de la pantalla, márgenes, etc. También definen si una página tiene *scroll* o si tiene fragmentos que permitan cambiar la visualización en la ventana sin la necesidad de cambiar de actividad.

5.2.2 Manifest

Archivo presente en todas las aplicaciones, ubicado en la raíz de proyecto. En él se declaran el nombre de paquete de la aplicación, las actividades que componen nuestro proyecto, así como los permisos especiales que necesite la aplicación para poder acceder a ciertas partes protegidas del sistema. También se declaran las relaciones entre actividades.

5.2.3 Gradle

Sistema de construcción de la aplicación que combina las mejores características de otros sistemas de construcción, proporcionando así un “todo en uno”. Gradle se ejecuta sobre la máquina virtual de Java, sin embargo, no sólo es capaz de construir proyectos de JVM, sino que también da soporte a aplicaciones nativas.

La comprensión de su funcionamiento es fundamental para el desarrollo de cualquier proyecto de Android, ya que aquí se establecen todas las dependencias del mismo, y, si por error falta por agregar alguna, el proyecto no compilaría.

En este apartado, para ambas aplicaciones se incluyeron las librerías de Android X, ya que son necesarias para la versión elegida para el desarrollo.

También fue necesario conectar el proyecto al repositorio de Google

A continuación, se adjunta una captura de parte del **Gradle** que muestra las dependencias del mismo:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.google.firebase:firebase-database:16.0.4'
    implementation 'com.google.firebase:firebase-auth:16.0.5'
    implementation 'com.google.android.material:material:1.0.0'
    implementation 'androidx.annotation:annotation:1.1.0'
    implementation 'androidx.lifecycle:lifecycle-extensions:2.0.0'
    implementation 'androidx.appcompat:appcompat:1.0.0'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    implementation 'androidx.recyclerview:recyclerview-selection:1.1.0-rc01'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
}
```

5.3 Creación del proyecto en Firebase

El primer paso para iniciar el desarrollo, es crear los proyectos de la aplicación del alumno y la aplicación del profesor en Android Studio. Una vez hecho esto, el siguiente paso es la creación del proyecto de Firebase a la que estarán conectadas.

Para conectar nuestro proyecto a Firebase es necesario agregar estas dependencias a ambas aplicaciones en el script de Gradle.

```
dependencies{
    implementation 'com.google.firebase:firebase-database:16.0.4'
    implementation 'com.google.firebase:firebase-auth:16.0.5'
}
```

Con dichas dependencias estamos agregando los servicios de autenticación y acceso a la base de datos a tiempo real del proyecto.

Para que ambos módulos se comuniquen se van a necesitar escribir y leer en la base de datos de Firebase.

El proceso de escritura es sencillo, apuntamos a un nodo, o referencia e introducimos un valor con el método setValue()

```
// Write a message to the database  
FirebaseDatabase database = FirebaseDatabase.getInstance();  
DatabaseReference myRef = database.getReference("message");  
  
myRef.setValue("Este es el Trabajo de Fin de Grado de María");
```

El proceso de escritura es algo más complejo. Al guardar la información mapeada, la estructura de datos es un conjunto de nodos identificados por una clave, cuyos hijos a su vez poseen clave y valor.

A continuación, se adjunta un fragmento de código de ejemplo que más adelante veremos aplicado en el código del módulo del profesor.

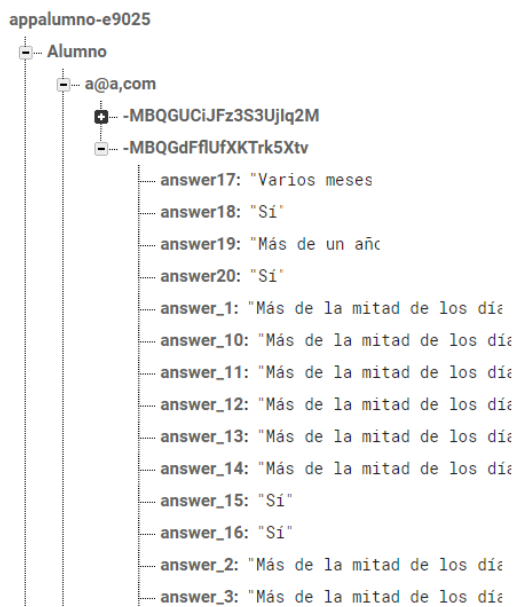
```
// Read from the database  
myRef.addValueEventListener(new ValueEventListener() {  
  @Override  
  public void onDataChange(DataSnapshot dataSnapshot) {  
    // This method is called once with the initial value and again  
    // whenever data at this location is updated.  
    String value = dataSnapshot.getValue(String.class);  
    Log.d(TAG, "Value is: " + value);  
  }  
  
  @Override  
  public void onCancelled(DatabaseError error) {  
    // Failed to read value  
    Log.w(TAG, "Failed to read value.", error.toException());  
  }  
});
```

5.4 Estructura de datos de la aplicación en Firebase

Fueron varias las estructuras de datos que se plantearon como posibles para la resolución de los objetivos establecidos en el análisis. Una de las soluciones que se planteó para estructurar el contenido enviado por el alumno para la lectura por el profesor fue la siguiente:

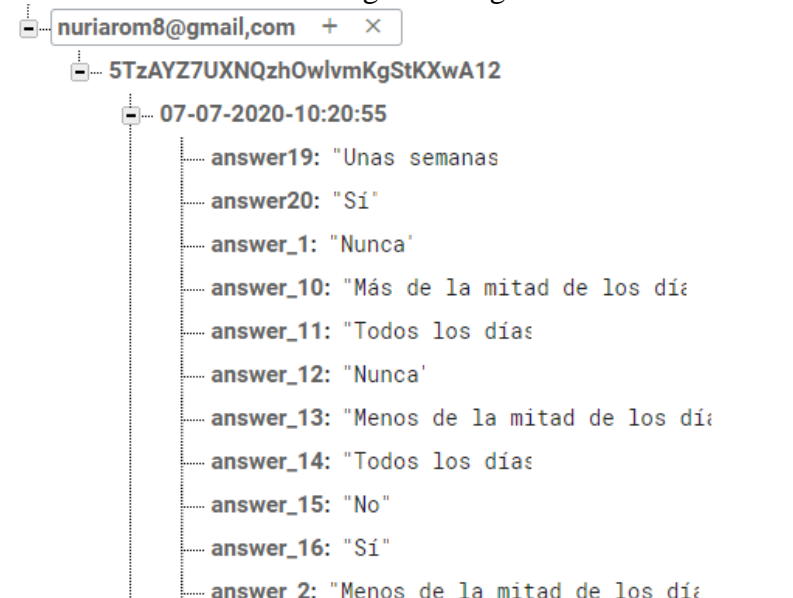
Dentro del nodo Alumno, un nodo padre (profesor) que contuviese como nodos hijos claves únicas generadas por el método push() para almacenar las respuestas y el contenido del mensaje del alumno. Esto permitiría disminuir la complejidad del árbol, ya que sólo sería necesario anidar dos nodos.

En la imagen adjunta a continuación se muestra el nodo bajo el cual el alumno guarda la información correspondiente al profesor a@a.com sin necesidad de almacenar sus credenciales. Esta captura es de una primera versión en la que no se verificaban los dominios permitidos para registrarse como profesor.



A pesar de que por motivos de buenas prácticas de desarrollo se pensó que sería la solución óptima, no fue la definitiva. Al estar dirigido el uso de la aplicación a docentes, el coordinador del proyecto sugirió que la estructura resultaría más sencilla si se añadía un nodo con el usuario del alumno para la posterior clasificación de información.

El resultado final de dicho cambio de código es el siguiente:



También se añadió un nodo con los dominios válidos para el registro de usuarios de tipo profesor y otro nodo para los profesores a los que se permite el acceso tras haber verificado su email.

appalumno-e9025



5.4.1 Seguridad de la base de datos de la aplicación

Una característica fundamental que debe tener toda aplicación es ser segura.

En la actualidad, las personas introducen datos personales constantemente en internet, y el robo o manipulación de los mismos podría resultar muy perjudicial para la víctima; por ello se estudiaron las distintas maneras de asegurar la información almacenada en base de datos. Esta aplicación en concreto, no almacena una gran cantidad de datos confidenciales, pues no se recogen nombres ni documentos del profesorado; sin embargo, sí que se utilizan direcciones de correo electrónico en la ruta de guardado.

El desarrollo se inició usando la base de datos en modo “pruebas”. Dicho modo no garantiza la seguridad de los datos puesto que la escritura y lectura de su contenido no está restringida.

Es decir, si alguien consiguiese entrar a nuestro proyecto podría borrar toda la información ahí almacenada. Sin embargo, tras investigar más, se establecieron reglas de seguridad encontradas entre la documentación de Firebase. Un servicio tan utilizado debe permitir seguridad, y efectivamente es el caso de este sistema de almacenamiento.

A continuación, se adjunta una captura de las reglas para asegurar la restricción de lecturas y escrituras:

También se propuso encriptar los datos que fuesen personales, pero como se menciona anteriormente en el documento, al ser un proyecto elaborado gracias a una colaboración de DSLabs y el departamento de psicología de la Universidad Complutense, se debieron llegar a acuerdos, ya que, es dicho departamento el que dirigirá las pruebas en los centros escolares.

5.5 Desarrollo de la aplicación del alumno

Para el desarrollo de esta parte del proyecto se deben implementar: el registro del alumno, la recolección de información del mismo, y el envío final a la base de datos.

5.5.1 Registro del alumno

Empleando los métodos existentes de autenticación de firebaseAuth, para darse de alta en Bully Stop el alumno sólo debe ingresar los datos que se piden por pantalla, sin necesidad de verificar si el correo electrónico existe o no, permitiendo así el anonimato, por lo que la implementación resultante de este apartado es relativamente sencilla. Se adjunta una captura a continuación:

```
firebaseAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        /*Si el usuario se registra con éxito, Firebase lo marca como un Inicio de Sesión
        * No es necesario llevar a una segunda pantalla de Login*/
        Toast.makeText(context: MainActivity.this, text: "Usuario Registrado", Toast.LENGTH_SHORT).show();
        Intent homeIntent = new Intent(context: MainActivity.this, SurveyPt0Activity.class);
        homeIntent.putExtra(name: "email", email); //Pasamos el parámetro email a la siguiente pantalla
        startActivity(homeIntent); //Vamos a la pantalla de Survey
        finish();
    } else {
        try {
            throw task.getException();
        } catch (FirebaseAuthWeakPasswordException e) { /*Manejo de Excepciones de Autenticación*/
            showAlert("La contraseña debe tener al menos 6 caracteres");
        } catch (FirebaseAuthInvalidCredentialsException e) {
            showAlert("El email introducido no es correcto");
        } catch (FirebaseAuthUserCollisionException e) {
            showAlert("Ya existe un usuario registrado con el email introducido");
        } catch (Exception e) {
            showAlert("No se ha podido registrar el usuario");
        }
    }
});
});
```

5.5.2 Autenticación del alumno

Como se mencionó en apartados anteriores, Firebase Authentication cuenta con un método de autenticación de usuarios que recibe como parámetros el email y la contraseña y permite iniciar sesión en caso de éxito, y devuelve una excepción en caso contrario. Para el manejo de excepciones con mensajes personalizados he implementado mensajes emergentes con la información relacionada con el error.

5.5.3 Respuestas a las encuestas y envío del mensaje

Todas las respuestas seleccionadas por el alumno deben ser guardadas en base de datos para que el profesor destinatario pueda recuperar los mismos. Para ello se creó una entidad de respuestas que almacena la totalidad de las preguntas del cuestionario. El contenido de esta entidad se va completando según se navega por las distintas pantallas, y no es hasta finalizar el envío que se guarda en base de datos, permitiendo al alumno rectificar en respuestas.

5.6 Desarrollo de la aplicación del profesor

Al estar conectadas ambas aplicaciones al mismo proyecto de Firebase, la autenticación de ambos usuarios se realiza en la misma base de datos.

Tras evaluar las posibles soluciones se implantaron dos formas de verificación de acceso para este tipo de usuarios: validar el correo electrónico y verificar que el dominio de la dirección del mismo esté registrado en base de datos como autorizado para profesores.

5.6.1 Registro de profesores

Para registrarse un usuario de tipo profesor, se comprueba que el email introducido tenga un dominio autorizado en base de datos, y en caso de éxito se registra aunque no se le permite el acceso hasta que verifique su cuenta mediante un enlace enviado a su buzón de correo.

El método que permite implementar la verificación de cuenta es `sendEmailVerification()`. En el siguiente fragmento de código se muestra la funcionalidad que impide que el usuario acceda a la aplicación sin haber sido verificada su cuenta previamente.

```
fAuth.createUserWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            /Si el profesor se registra con éxito se le envia un email de
verificación de cuenta/

fAuth.getCurrentUser().sendEmailVerification().addOnCompleteListener(new
OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()) {
            showAlert("Se ha enviado un email de verificación a
su dirección de correo electrónico. Por favor revise su bandeja de entrada", info);
            /Tras enviar el email de verificación limpio los datos
de la pantalla/

            emailInput.setText("");
            passwordInput.setText("");
        }else{
            showAlert("Se ha producido un error al enviar el
email de verificación de usuario", error);
        }
    }
});
```

5.6.2 Autenticación del profesor

Para evitar fallos de seguridad en el acceso a la aplicación de mensajes del profesor se estableció un sistema de autenticación de doble verificación con los recursos con los que se contaba.

Por un lado, Firebase, facilita la implementación de verificación de cuentas de correo electrónico de gran manera, ya que, mediante la invocación a un método en nuestra actividad, es el propio Firebase que se encarga de todo el proceso restante. Para hacer la aplicación aún más segura se estableció una comprobación mediante código de los dominios de las cuentas que intentan registrarse o acceder a esta aplicación, para ello se utilizó Java y expresiones regex.



noreply@appalumno-e9025.firebaseio.com
para mí ▾

mar., 7 jul. 9:29 (hace 2 días) ☆

Hello,

Follow this link to verify your email address.

https://appalumno-e9025.firebaseio.com/_/auth/action?mode=verifyEmail&oobCode=ZBq7-3KXWdjlX5qvZWkHxU5LjMzn7W20b2SfvJgFDyAAAAFzKC20vg&apiKey=AlzaSyCLiI0iMiX-d5nW2XUfl2U-UXi2XOGe_4&lang=en

If you didn't ask to verify this address, you can ignore this email.

Thanks,

En el caso ideal, los usuarios de esta aplicación deberían acceder sólo con correos institucionales, cuyos dominios se grabarían en base de datos para concederles autorización.

A continuación, se adjunta un fragmento de código dónde se valida la dirección del usuario tipo profesor:

```
public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
    for(DataSnapshot ds : dataSnapshot.getChildren()){
        domains.add(ds.getValue().toString());
    }
    boolean found = false;
    for(int i=0; i<domains.size() ; i++){
        String darray [] = domains.get(i).split("\\.");
        String regex = "^([\\w.+\\-]+)";
        String aux="";
        /Para formar el string regex volvemos a concatenar el dominio
separado con \\./
        for(int j=0; j<darray.length;j++){
            aux += darray[j];
            if(j!=darray.length-1){
                aux+="\\";
            }
        }
        regex = regex + aux + "$";
        /Comprueba que el dominio del email introducido por pantalla es
de un dominio válido para el acceso/
        if(email.matches(regex)){
            found=true;
        }
    }
}
```

5.6.3 Recogida de datos para el buzón de entrada

Para recoger los y añadirlos a la vista de la bandeja de entrada del profesor se debe hacer una consulta a base de datos y recorrer los nodos del árbol hasta alcanzar el nivel donde se almacena la información. A continuación se adjunta el código del mismo:

```
ref = database.getInstance().getReference("Alumno");

answers_info = new ArrayList<Answers>();

ref.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for(DataSnapshot ds : dataSnapshot.getChildren()){
            if(ds.getKey().equalsIgnoreCase(cleanTeacherEmail)){
                for(DataSnapshot ds2 : ds.getChildren()) {
                    answers_info.add(ds2.getValue(Answers.class));
                }
            }
        }
        messagesAdapter = new MessagesAdapter(answers_info, recyclerView);
        recyclerView.setAdapter(messagesAdapter);
    }
}
```

6 Pruebas

6.1 Pruebas funcionales

Tras finalizar la codificación de cada uno de los módulos, se realizaron pruebas para comprobar el correcto funcionamiento de la aplicación.

Primeramente, las pruebas se centraron en la comprobación de los métodos cuyo desarrollo resultó más complejo. Como por ejemplo, los métodos de verificación de registro y el almacenamiento y recuperación de datos de *firebase*.

Después, se realizaron pruebas funcionales comprobando uno a uno todos los escenarios que debían devolver algún tipo de mensaje de error por pantalla. A continuación se adjuntarán un par de tablas que recogen todos los casos probados en cada uno de los módulos del proyecto.

Las pruebas realizadas para la interfaz del alumno fueron las siguientes:

Entrada	Salida Esperada
Registro del alumno rellenando Correctamente los campos	Se registra correctamente en base de datos
Intento de registro del alumno sin rellenar ningún campo	Se muestra un mensaje de error indicando que debe completar todos los campos
Intento de registro del alumno Introduciendo datos erróneos	Se muestra un mensaje de error
Intento de inicio de sesión sin rellenar campos	Se muestra un mensaje de error indicando que debe completar todos los campos
Inicio de sesión con datos registrados	Se inicia sesión con éxito y se procede a la siguiente pantalla
Intento de inicio de sesión con datos incorrectos (dirección de correo, contraseña)	Se muestra un mensaje de error específico para el caso de formato incorrecto o datos erróneos
Introducción de un profesor NO registrado como destinatario	Se muestra un mensaje de error relacionado
Intento de continuar a la siguiente pantalla sin haber terminado de responder al cuestionario en el que se encuentra	Se muestra un mensaje específico indicando qué número de pregunta se ha dejado sin contestar
Intento de envío de mensaje sin escribir un texto explicativo	Se muestra un mensaje de error
Envío de mensaje habiendo completado el texto y las pantallas anteriores	Se envía el mensaje con éxito y se registra en base de datos

Las pruebas realizadas para la interfaz del docente fueron las siguientes:

Entrada	Salida Esperada
Registro del docente rellenando Correctamente los campos	Se registra correctamente en base de datos y se envía un enlace de verificación al correo

Intento de registro del docente sin rellenar ningún campo	Se muestra un mensaje de error indicando que debe completar todos los campos
Intento de registro del docente Introduciendo un dominio inválido	Se muestra un mensaje de error indicando que debe utilizar su correo institucional
Intento de inicio de sesión sin rellenar campos	Se muestra un mensaje de error indicando que debe completar todos los campos
Inicio de sesión con datos registrados sin verificar su cuenta previamente	Se muestra un mensaje de error indicando que debe verificar su cuenta
Intento de inicio de sesión con datos incorrectos (dirección de correo, contraseña)	Se muestra un mensaje de error específico para el caso de formato incorrecto o datos erróneos
Inicio de sesión con datos registrados y verificados	Se inicia sesión con éxito y se accede a la bandeja de entrada

Quando se encontraba algún error se procedía a la inspección de código y la solución del mismo.

6.2 Pruebas de compatibilidad

Para intentar reproducir las pruebas de la manera más fiel posible a las que se realizarán cuando se vuelva a las aulas, se instaló la aplicación en distintos *smartphones* y *tablets* con pantallas de distinta resolución y versiones de Android, para asegurar que las actividades se visualizaban correctamente y que los estilos escogidos para los componentes se mantenían.

Los dispositivos que se utilizaron para las pruebas fueron:

Samsung Galaxy S9 (2018)
 Samsung Galaxy A70 (2018)
 Xiaomi Redmi Note 6 Pro (2018)
 Samsung Galaxy Tab (2013)
 Asus MeMO Pad Smart 10 (2013)

En este apartado se encontraron algunas dificultades a lo hora de adaptar ciertos estilos. En específico los radiobuttons con texto en la parte superior no se visualizaban correctamente en versiones de Android más antiguas, inferiores al número 5.0, aunque finalmente se consiguió solventar el error.

7 Conclusiones y trabajo futuro

7.1 Conclusiones

La aplicación Bully Stop aporta una solución tecnológica de valor abordando un problema social real y actual para el cual no existen muchas aplicaciones. Las pocas existentes publicadas, están enfocadas a un sector específico, como un instituto o colegio particular, lo que impide a otros centros que se quieran sumar a la iniciativa de monitorizar *on-line* estos problemas, aprovechar dichas aplicaciones.

Además, esta nueva aplicación aporta un enfoque diferente del tema, ya que a pesar de que en la actualidad existen múltiples plataformas de mensajería instantánea y de correo electrónico, no todas permiten enviar mensajes de forma anónima, y las que sí lo hacen, no se centran en el tema del acoso escolar, sino más bien en la mera comunicación.

El contenido de la aplicación y las preguntas dirigidas al alumno están redactadas por profesionales, asegurando así que se recoge la información realmente necesaria.

Además, mencionar que este trabajo será utilizado en varios colegios de la comunidad de Madrid, lo cual no fue posible este año debido a la situación de alerta sanitaria.

A partir del momento en que empiece a ser empleado en centros escolares, se añadirán colaboradores para la gestión de datos y posibles mejoras o edición de contenido de la aplicación, que estará disponible en la tienda de aplicaciones de Google.

Por último, destacar que el trabajo realizado me ha introducido al mundo de desarrollo de aplicaciones, y me ha aportado numerosos conocimientos de los que antes carecía. Entre ellos se encuentra: el trabajo con Firebase; que, a pesar de que la implementación para el trabajo de fin de grado sólo se vale de dos de sus servicios, espero continuar pronto desarrollando yendo un paso más allá, y trabajando con su servicio de almacenamiento en la nube. También he mejorado mis conocimientos de Java y he ganado experiencia trabajando con la API de Android.

7.2 Trabajo futuro

Ante la motivación de presentar un producto de utilidad a los clientes, se han planteado mejoras a implantar en un futuro próximo.

En función del recibimiento de la aplicación y de las necesidades que puedan surgir entre los usuarios finales, se podrían implantar nuevas funcionalidades como: notificaciones de *push* en la aplicación del profesor implementando el servicio de *Firebase cloud storage*, la posibilidad de adjuntar contenido multimedia por parte de los alumnos y en última instancia implementar un chat con el profesor para tratar los casos más urgentes.

Referencias

- [1] <https://firebase.google.com/docs>
- [2] https://developer.android.com/guide/topics/ui/declaring-layout?hl=es_419
- [3] <https://developer.android.com/guide/components/activities/activity-lifecycle?hl=es>
- [4] <https://www.adslzone.net/reportajes/software/que-es-android/>
- [5] <https://stackoverflow.com/questions/8515936/android-activity-life-cycle-what-are-all-these-methods-for>

Glosario

API	Application Programming Interface
Smartphone:	Teléfono móvil con capacidad para almacenar datos y realizar distintas actividades
Android:	Sistema operativo basado en linux para móviles
Firebase	Base de datos compatible con proyectos Android e iOS
Layout	Estructura de página en Android

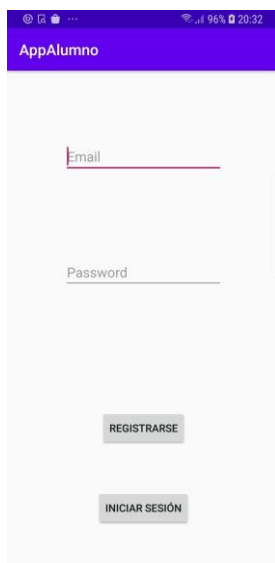
Anexos

A Manual de instalación

Para instalar las aplicaciones desarrolladas en este proyecto sólo es necesario instalarse la apk en un dispositivo de sistema operativo Android. La aplicación será publicada también en la tienda de aplicaciones de Google.

B Manual de Usuario

1. Para acceder a la aplicación del alumno, tras descargar la aplicación en el dispositivo, se nos abre la primera pantalla:



The screenshot shows the 'AppAlumno' registration screen. It features a purple header with the text 'AppAlumno'. Below the header, there are two input fields: 'Email' and 'Password'. At the bottom of the screen, there are two buttons: 'REGISTRARSE' and 'INICIAR SESIÓN'.

2. Una vez registrado el usuario correctamente, la aplicación te redirige a la segunda pantalla, donde se debe introducir el email de un profesor ya registrado, o no se podrá continuar por el resto de pantallas.



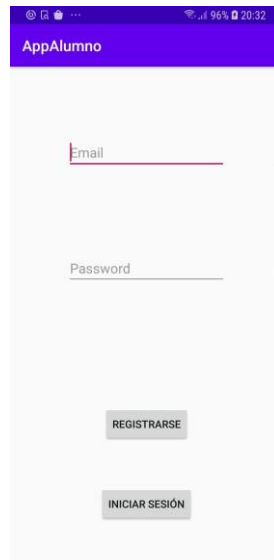
The screenshot shows the 'Cuestionario' screen. It features a purple header with the text 'Cuestionario'. Below the header, there are three input fields: 'Edad', 'Email del Profesor', and 'Género'. The 'Género' field has three radio button options: 'Femenino', 'Masculino', and 'Otro'. Below the input fields, there is a text prompt: 'A continuación deberá responder a varias preguntas'. At the bottom of the screen, there is a button labeled 'SIGUIENTE'.

3. Las siguientes 7 pantallas consisten en una serie de preguntas que deben ser respondidas por el usuario.

4. Finalmente se puede enviar un mensaje de texto y adjuntar tu nombre o actuar como usuario anónimo.

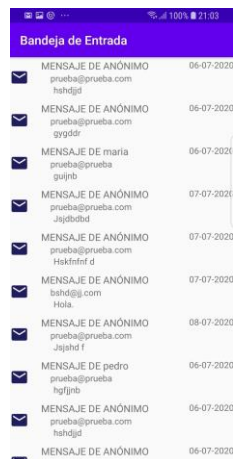
En la aplicación del profesor

1. Registrarse (Se adjunta la pantalla del alumno, pues coinciden) Sólo cambia la cabecera



2. Verificación vía enlace

3. Acceso a la bandeja de entrada



3. Visualización del mensaje

