

Escuela Superior de Ingeniería y Tecnología

CURSO ACADÉMICO 2017/2018

MEMORIA DEL TRABAJO DE FIN DE GRADO

SISTEMA DE APRENDIZAJE M-LEARNING PARA **PRÁCTICAS DE CONDUCIR**

M-LEARNING SYSTEM FOR DRIVING PRACTICES



La Laguna, 3 de Septiembre de 2018

D. **Cándido Caballero Gil**, con N.I.F. 42.201.070-A profesor Ayudante Doctor de Universidad adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna, como tutor

D. **Carina Soledad González González**, con N.I.F. 54.064.251-Z profesora Catedrática de Universidad adscrita al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna, como cotutora

C E R T I F I C A (N)

Que la presente memoria titulada:

“Sistema de aprendizaje m-learning para prácticas de conducir”

ha sido realizada bajo su dirección por D. **Edna Liliana Galiano Camacho**, con N.I.F. x6936169j.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 3 de septiembre de 2018

Agradecimientos

Como dice Winston Churchill “El éxito es la capacidad de ir de fracaso en fracaso sin perder el entusiasmo”. Este ha sido un largo camino, con victorias y derrotas, pero siempre con un objetivo en mente, luchar por aquello que realmente quiero. Por supuesto, esto no habría sido posible sin personas que han influido en mi vida y han sido una inspiración para mí, algunas de ellas, sin ni siquiera saberlo.

Para empezar, agradezco a mis padres, por todo su apoyo, por estar a mi lado en las buenas y no tan buenas. A aquellos amigos y amigas que han estado ahí con palabras de ánimo y consejos, nunca han caído en saco roto. En especial a Orne y Vero, por tantas enseñanzas y por ser dos ejemplos de esfuerzo y dedicación, mujeres valientes. He aprendido que cuando uno hace lo que le gusta y apasiona, sean cuales sean los resultados, ya ha ganado. Así que me llevo no solo una experiencia de crecimiento a nivel profesional sino también a nivel personal.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

Resumen

A día de hoy, no se puede negar la importancia que han adquirido los dispositivos móviles en nuestro día a día. Se hace uso de ellos para todo tipo de actividad cotidiana, por simple que sea, existe una amplia gama de aplicaciones a nuestra disposición para facilitar la labor. Debido a ello, se ha producido un crecimiento del número de herramientas diseñadas para estos dispositivos.

Android es un sistema operativo que está basado en un núcleo Linux. A principios de 2017, se presenta liderando la cuota de mercado con más del 81,7%, a nivel mundial. Debido al crecimiento de este sector, ya no son solo las empresas las que deciden apostar por estas nuevas tecnologías sino también el sistema de enseñanza, que evoluciona con ellas. M-Learning es un sistema de aprendizaje vía móvil, que busca el desarrollo del conocimiento y habilidades de forma automática, por medio de un dispositivo del móvil, opción que es cada vez más demandada por miembros de la comunidad educativa para llevar a cabo la dinámica enseñanza - aprendizaje.

El objetivo de este trabajo de fin de grado es el desarrollo de una aplicación que implemente un sistema de aprendizaje m-learning para prácticas de conducir, intentando potenciar y mejorar el rendimiento en las mismas, basándose en la recogida y gestión de los datos generados durante cada sesión. *DriverMoves* se pondrá a disposición del usuario para permitir el acceso a todos los datos de interés recogidos en cada una de sus prácticas o exámenes de conducir, teniendo la posibilidad de acceder a ellos en cualquier momento y lugar.

Palabras Clave: Aprendizaje, Enseñanza, M-Learning, Prácticas, Conducir, Android, Móvil, Aplicación, Geolocalización, Coordenadas, Ruta, Examen, Autoescuela, DGT

Abstract

Nowadays, the importance that mobile devices have acquired day in day out you can not be denied. It is used for all kinds of daily activities, no matter how simple it is, there are a wide range of applications at our disposal to facilitate the work. Due to this, there are an increasing number of tools designed for these devices.

Android is an operating system based on the Linux kernel. At the beginning of 2017, it is leading the market share with more than 81.7%, worldwide. Due to the growth of this sector, it is not only the companies that decide to bet on these new technologies but also the teaching system, which evolves with them. M-Learning is a mobile learning system, which seeks the development of knowledge and skills in an automatic way, through a mobile device, an option that is increasingly demanded by members of the educational community to carry out the dynamics teaching - learning.

The objective of this end-of-degree project is the development of an application that implements an m-learning system for driving practices, trying to enhance and improve performance in them, based on the collection and management of data generated during each session. DriverMoves will be available to the user to allow access to all the data of interest collected in each of their practices or driving tests, having the possibility of accessing them at any time and place.

Keys Words: Learning, Teaching, M-Learning, Practices, Driving, Android, Mobile, Application, Geolocation, Coordinates, Route, Exam, Driving school, DGT

INDICE TRABAJO

Capítulo 1. Introducción	10
1.1 Objetivos Generales	11
1.3 Fases de desarrollo	13
1.4. Estructura	14
Capítulo 2. Contexto y Antecedentes	15
Capítulo 3. Estado del Arte	15
3.1 Android	16
3.1.1 Arquitectura	16
3.1.2 Versiones	19
3.2 Enfoques	20
Capítulo 4. Estudio previo	22
4.1. Requerimientos	23
4.1.1 Requerimientos Funcionales	23
4.1.1.1 Gestion de Usuarios	23
4.1.1.2 Aplicación	24
4.1.2 Requerimientos No Funcionales	26
4.2. Metodología de desarrollo	27
4.2.1 Prototipos	29
4.3. Base de datos	29
4.4 API's	29
4.4 Seguridad	31
Capítulo 5. Diseño	31
5.1. Arquitectura	31
5.2.1 Android Studio	32
5.2.2 Firebase	33
5.2.3 Ninja Mock	34
5.2.4 Java	35
5.2.5 XML	35
5.2.6 Json	36
5.2.7 GPX	36
5.2.8 Google Maps	36
5.2.9 GPS	37

5.2.10 Emulador Android	37
5.3. Diseño de Pantallas	38
A continuación, se exponen por medio de dos diagramas el diseño de pantallas según el rol del usuario. Están representados los cambios de las pantallas y los recursos que se usan en cada momento.	38
5.3.1 Diseño de Pantalla del Profesor	38
5.8.2 Diseño de Pantalla del Alumno	39
5.9. Diseño de la Interfaz	40
5.10. Diseño de la Base de Datos	45
Capítulo 6. Implementación	48
6.1 Implementación de la app	48
6.1.1 Gestión de usuario	48
6.1.2 Gestión de Prácticas y Exámenes	50
6.1.2 Gestión de una Sesión	52
6.2 Implementación de la interfaz	54
6.2.2 Pantalla principal	55
6.2.2.1 Inicio de sesión	55
6.2.2.2 Prácticas y Exámenes	56
Estas dos pantallas están disponibles para todos los usuarios , la única diferencia es que el alumno no ve los Fragments de “INICIO” y “BUSCAR”.	56
6.2.2.3 Búsqueda	57
6.2.3 Pantalla de datos de una sesión	58
6.2.4 Pantalla de Mapa	58
Capítulo 7. Pruebas	59
7.1 Pruebas en simulación	59
7.2 Prueba en caso real	60
Capítulo 8. Conclusiones	61
Capítulo 9. Trabajos futuros	63
Capítulo 10. Presupuesto	63
Bibliografía	64

ÍNDICE FIGURAS

<i>Figura 1: Gráfica de uso de Sistemas Operativos</i>	12
<i>Figura 2: Arquitectura Android</i>	17
<i>Figura 3: App Android para Test de Conducir</i>	21
<i>Figura 4: App Android para aprender señales de tráfico</i>	22
<i>Figura 5: App simulador de conducir</i>	22
<i>Figura 6: app para aprender a aparcar</i>	23
<i>Figura 7: Fases de desarrollo</i>	29
<i>Figura 8: Captura de Ninja Mock</i>	35
<i>Figura 9: Coordenadas Google Maps</i>	37
<i>Figura 10: Diagrama Pantalla Profesor</i>	39
<i>Figura 11: Diagrama Pantalla Alumno</i>	40
<i>Figura 12: Diseño Pantalla Login</i>	41
<i>Figura 13: Diseño Pantalla Registro</i>	41
<i>Figura 14: Diseño Pantalla Inicio</i>	42
<i>Figura 15: Diseño Pantalla de Sesión</i>	42
<i>Figura 16: Diseño Pantalla Mapa</i>	42
<i>Figura 17: Diseño Pantalla de Búsqueda</i>	42
<i>Figura 18: Diseño Pantalla Prácticas Profesor</i>	43
<i>Figura 19: Diseño Pantalla Exámenes Profesor</i>	43
<i>Figura 20: Diseño Pantalla Prácticas Alumno</i>	44
<i>Figura 21: Diseño Pantalla Exámenes Alumno</i>	44
<i>Figura 22: Diseño Pantalla Datos de una Sesión</i>	44
<i>Figura 23: Menú Lateral</i>	45
<i>Figura 24: Cuenta Usuario</i>	45
<i>Figura 25: Ajustes</i>	45
<i>Figura 26: Estructura JSON Prácticas</i>	46
<i>Figura 27: Estructura JSON Exámenes</i>	47
<i>Figura 28: Estructura JSON Usuarios</i>	47
<i>Figura 29: Código de Comprobación de Email y Contraseña</i>	49
<i>Figura 30: Código de Registro de usuario en la Base de datos</i>	49
<i>Figura 31: Código de Comprobación de credenciales</i>	49
<i>Figura 32: Código de Cierre de Sesión</i>	50
<i>Figura 33: Código para las Shared Preferences</i>	50
<i>Figura 34: Traer el Listado de Usuarios de La BDD</i>	50
<i>Figura 35: Traer el Información de una Sesión de La BDD</i>	51
<i>Figura 36: Traer el Información de un Mapa de la BDD</i>	51
<i>Figura 37: Despliegue del Mapa y Creación de Marcadores</i>	52
<i>Figura 38: GPS no activado</i>	52
<i>Figura 39: Mensaje de Alerta para activar GPS</i>	53

<i>Figura 40: Ejemplo de creación de un Marcador.....</i>	<i>54</i>
<i>Figura 41: Función que toma las coordenadas de la posición actual, una vez ésta cambie.....</i>	<i>54</i>
<i>Figura 42: Pantalla de Login.....</i>	<i>55</i>
<i>Figura 43: Pantalla de Registro.....</i>	<i>55</i>
<i>Figura 44: Pantalla de Inicio Profesor.....</i>	<i>56</i>
<i>Figura 45: Listado Prácticas.....</i>	<i>57</i>
<i>Figura 46: Pantalla Listado Exámenes.....</i>	<i>57</i>
<i>Figura 47: Búsqueda Usuarios.....</i>	<i>58</i>
<i>Figura 48: Pantalla de un Examen.....</i>	<i>58</i>
<i>Figura 49: Pantalla de una Práctica.....</i>	<i>58</i>
<i>Figura 50: Ruta de un Examen.....</i>	<i>59</i>
<i>Figura 51: Ruta de una Práctica.....</i>	<i>59</i>
<i>Figura 52: Datos Prueba Simulación.....</i>	<i>60</i>
<i>Figura 53: Mapa Prueba Simulación.....</i>	<i>60</i>
<i>Figura 54: Datos Prueba Real.....</i>	<i>61</i>
<i>Figura 55: Mapa Prueba Real.....</i>	<i>61</i>

ÍNDICE TABLAS

<i>Tabla 1: Versiones Android.....</i>	<i>20</i>
--	-----------

Capítulo 1. *Introducción*

No cabe duda de que, con el paso de los años, los dispositivos móviles, de todo tipo, han ido ganando terreno, ya no solo como una herramienta destinada al hobby sino también como un instrumento de trabajo. Su crecimiento ha ido de la mano del avance y desarrollo de nuevas tecnologías y herramientas, que se ponen a disposición del usuario para potenciar, de forma conjunta, su experiencia y productividad.

Actualmente para obtener el carnet de conducir es necesario aprobar dos exámenes, el teórico y el práctico, y como es de esperar, han surgido numerosas aplicaciones que buscan facilitar el aprendizaje y asimilación de los conceptos, así como conocimientos y destrezas, aplicadas a la conducción. Dentro del mercado de aplicaciones de este tipo, se encuentra una amplia gama de opciones; aplicaciones para la preparación de los test de forma online, aplicaciones para memorizar las señales de tráfico, aplicaciones que involucran la gamificación en su metodología, aplicada a la simulación del estacionamiento, etc. Sin embargo, ninguna de las aplicaciones existentes contempla ofrecer al usuario datos que han sido generados por él mismo y en tiempo real para las prácticas de conducir, es decir, cada vez que se da por finalizada una de ellas, el alumno no tiene ninguna herramienta o material a su disposición que le permita llevar un seguimiento de sus avances en cada una de ellas, salvo la experiencia adquirida.

1.1 Objetivos Generales

El objetivo general de este trabajo ha sido el diseñar e implementar una aplicación llamada *DriverMoves*, que dé lugar a una nueva forma de desarrollar el conocimiento y las capacidades en base a la experiencia, y aplicado el campo de la conducción. La idea surge debido a la inexistencia de una aplicación que contemple estas características, de este modo, se plantea hacer uso de las distintas tecnologías de las que se dispone a día de hoy para facilitar la preparación para el examen práctico de conducir, se aprovecha el campo de acción tan amplio y las numerosas ventajas que pueden brindar las aplicaciones móviles que llegan con fuerza para transformar el sistema de aprendizaje.

Por un lado, se ha llevado a cabo un estudio que presenta el estado del uso del dispositivo móvil. Desde agosto de 2017, hay más de 3.500 millones de usuarios de Internet móvil. Además, los usuarios destinan en promedio el 69% de su tiempo en smartphones, dando lugar a generar el 80% del uso global de Internet. El 50% del tiempo que destinan los usuarios a los medios digitales se emplea en aplicaciones móviles.

Por otro lado, el siguiente estudio refleja el porcentaje de cada Sistema operativo. Android pasa a tener un 37,93 % en el mercado, superando así al sistema operativo de Windows, con un 37,91% y, en segundo lugar, para el S.O. de móviles, se encuentra iOS que presenta un 13,09%.

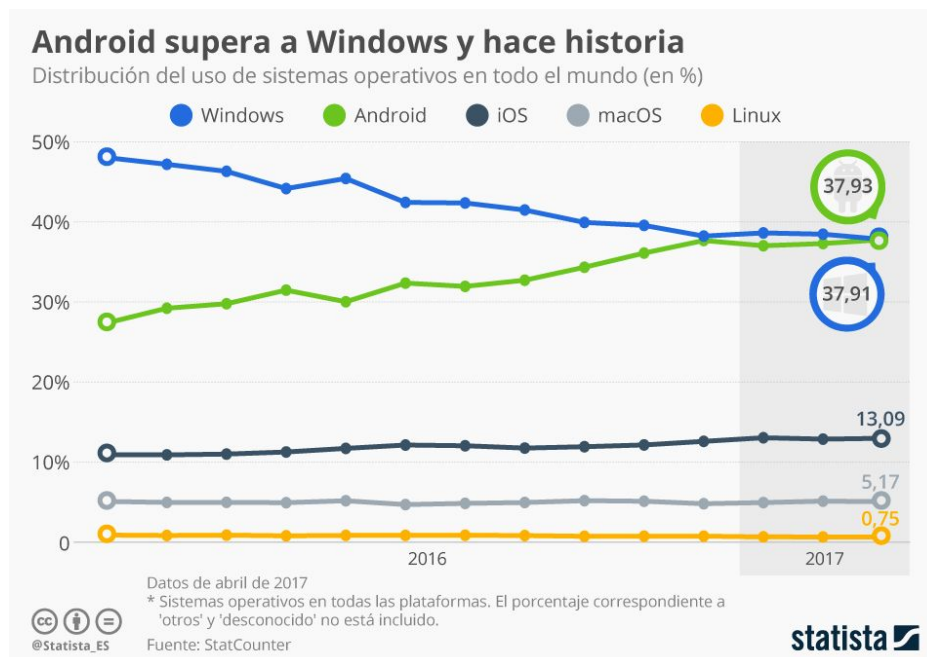


Figura 1: Gráfica de uso de Sistemas Operativos

Teniendo en cuenta lo comentado anteriormente, junto con la gran demanda que presenta Android en el mercado, es seleccionado para llevar a cabo el desarrollo de **DriverMoves**.

1.2 Objetivos del Proyecto

El objetivo de este proyecto, es desarrollar una aplicación móvil que propone un sistema de aprendizaje m-learning, esta permitirá a cada usuario establecer un seguimiento de su actividad, llevando a cabo un análisis posterior. También se busca un desarrollo de conocimiento y evaluación de forma automática, es decir, tendrán al alcance de su mano un sistema de seguimiento de los errores cometidos a lo largo de sus propias prácticas, la ruta seguida y más puntos de interés.

El desarrollo debe dar lugar a una aplicación con una interfaz sencilla pero atractiva al mismo tiempo, a la que el usuario se adapte con facilidad y por supuesto, cumpla con los requisitos funcionales marcados tras el análisis previo. Deben quedar bien diferenciados los roles de los usuarios, ya que la interfaz del profesor presenta ciertas diferencias con respecto a la del alumno.

Este será un proyecto cuyo desarrollo se llevará a cabo desde cero, tanto el análisis como la planificación de cada una de sus fases hasta llegar al producto final. De esta manera se pondrá en práctica todos los conocimientos y aptitudes adquiridas durante el transcurso de la carrera.

1.3 Fases de desarrollo

A continuación, se describe brevemente las fases a seguir en el desarrollo de *DriverMoves*.

- Descripción de requisitos y análisis: Se lleva a cabo un análisis de los requisitos necesarios para cumplir los objetivos que busca alcanzar la aplicación.
- Estudio de las tecnologías implicadas: Se trata de una investigación sobre las tecnologías que mejor se adapten a las exigencias y necesidades del proyecto.
- Estudio de la plataforma de desarrollo: Se busca una plataforma que brinde un desarrollo eficaz e intuitivo, en pos de facilitar todo el proceso de la implementación y aproveche al máximo los recursos presentados.
- Diseño del proyecto (Base de datos - App): Una vez bien definidos los objetivos anteriores, se procede primero con el diseño de la app. Una vez finalizado, se pasa al diseño de la BBDD.

- Creación de la parte de aplicación Android: Desarrollo de la aplicación en Android Studio.
- Creación de la Base de Datos: Implementación de la BBDD en Firebase.
- Conexión entre Base de datos y App: Integración de la BBDD de Firebase con la aplicación de desarrollada en Android Studio.
- Comprobación de requisitos, mejoras y testeo: Una vez terminada la parte de la implementación, se pasa a las distintas pruebas para comprobar el correcto funcionamiento de la aplicación. Se analizan y plantean mejoras.
- Preparación de la memoria y presentación: Preparación de todo el material pertinente para la documentación de *DriverMoves*.

1.4. Estructura

A lo largo de esta memoria se abordarán con detalle, todas las fases del desarrollo comentadas anteriormente, junto con las tecnologías y herramientas que intervienen en la creación de *DriverMoves*.

En el **capítulo 2 - Estado del Arte**, se hará una breve introducción al sistema operativo Android y sus principales características.

A continuación, en el **capítulo 3 - Contexto / Antecedentes**, Se pondrá en conocimiento del lector el contexto actual que envuelve este proyecto, junto con una comparativa de aplicaciones similares que existan actualmente en el mercado.

En el **capítulo 4 - Estudio previo**, partiendo del estudio que se ha realizado previamente, se comentarán los requerimientos funcionales y no funcionales, así como la metodología implementada para el desarrollo de *DriverMoves*.

Posteriormente, En el **capítulo 5 - Diseño** se tocará el diseño tanto de la aplicación como de la base de datos de la misma; la arquitectura utilizada y las tecnologías implementadas.

Más adelante, **El capítulo 6 - Implementación**, desglosa las fases de implementación tanto de la aplicación como de su base de datos, usando para ello las tecnologías comentadas anteriormente.

Seguidamente, en el **capítulo 7 - Pruebas**, se expone la descripción de las características obtenidas de los resultados. También las funcionalidades que presenta *DriverMoves*.

El capítulo 8 - Conclusiones, presenta las conclusiones obtenidas una vez concluido el proyecto.

En el Capítulo 9 - trabajos futuros, se contemplan las propuestas de mejora de cara al futuro para este proyecto.

Capítulo 2. *Contexto y Antecedentes*

Las nuevas tecnologías y sus continuos avances han ido revolucionando el método de enseñanza que conocíamos hasta ahora. Desde hace unos años, un nuevo sistema ha ido cogiendo fuerza, el m-learning, que consiste en el aprendizaje electrónico móvil, como su nombre indica, hace uso de cualquier dispositivo móvil que cuente con conectividad, de esta forma, se consigue el desarrollo de diversas habilidades y la construcción del conocimiento de forma automática. El usuario aprende a desenvolverse en distintas destrezas y en la resolución de problemas, así como también trabaja las capacidades comunicativas. Una de las ventajas que aporta este sistema, es la posibilidad de brindar una experiencia personalizada, ya que puede llevarse a cabo en cualquier momento o lugar, aportando flexibilidad y comodidad, sin perder el carácter colaborativo.

Así como ha crecido este concepto y evolucionado el modo de aprendizaje con la integración de las TIC, también lo han hecho las diversas herramientas que se tienen a disposición para llevarlo a cabo. Cada vez existe un mayor número de aplicaciones en el mercado que proporcionan este tipo de sistema de aprendizaje. Son muy comunes las aplicaciones que implementan una base de datos a la que el usuario accede en busca de información como los horarios del transporte público, información de la actividad cultural y artística de una zona o programas de entrenamiento físico; aplicaciones que entrenan las diferentes aptitudes por medio de juegos para mejorar la atención, la coordinación, la capacidad de deducción, entre otros.

Capítulo 3. *Estado del Arte*



Android es un sistema operativo basado en el núcleo de Linux. Su diseño se centra principalmente en dispositivos móviles con pantalla táctil, tales como smartphones, tablets, smartwatches, televisores y coches. En un principio fue desarrollado por Android Inc. Ésta empresa fue respaldada por Google, quien posteriormente, en 2005, la compra. En 2007, Android fue presentado junto la fundación del Open Handset Alliance, que confirma un consorcio de compañías de

hardware, software y telecomunicaciones, para progresar en los estándares abiertos de los dispositivos móviles. En octubre de 2008, se vendió el primer móvil con el sistema operativo Android, el HTC Dream.

Dentro de las características que distinguen a este sistema operativo, destaca una interfaz de usuario intuitiva y de fácil manejo, que se aprovecha sobre todo en smartphones. Android es una plataforma abierta, lo que significa que los fabricantes y operadoras pueden partir del sistema operativo y realizar modificaciones pensando en sus usuarios.

3.1 Android

3.1.1 Arquitectura

Android presenta una estructura en capas, como se puede ver en la figura 2.

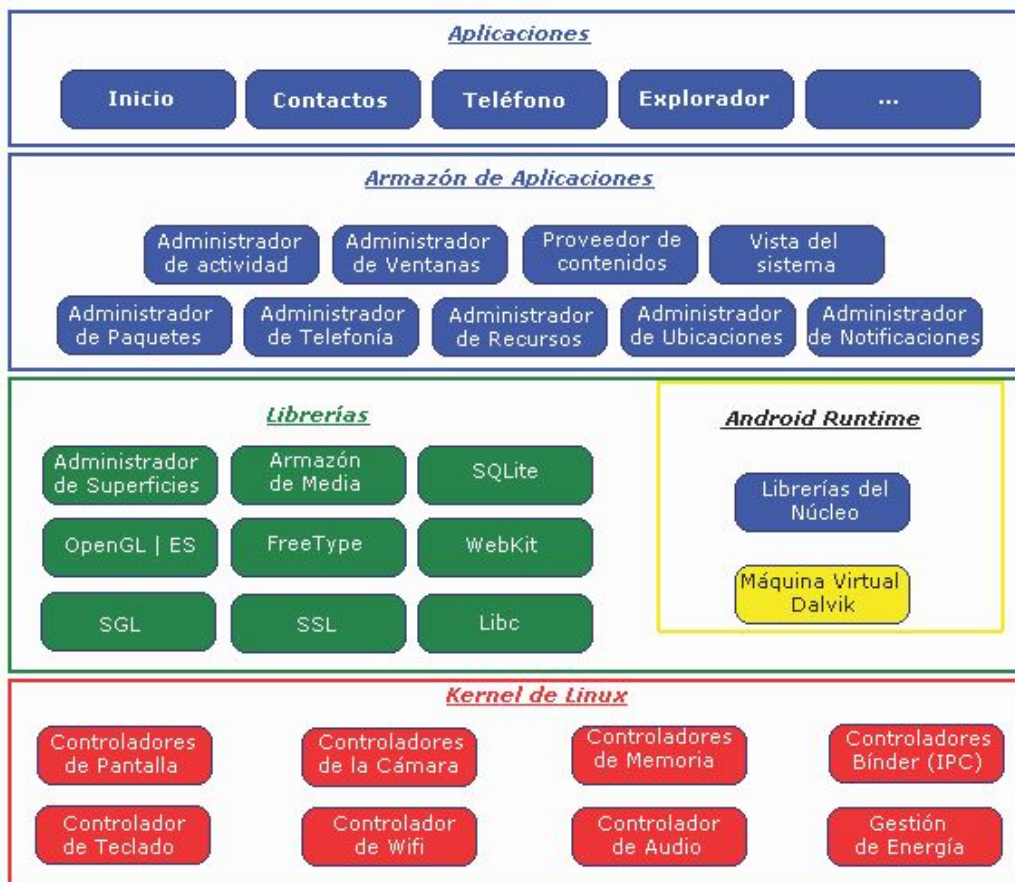


Figura 2: Arquitectura Android

- **Aplicaciones:** En esta capa, se encuentran tanto las aplicaciones incluidas por Android como aquellas que ha ido instalando el usuario posteriormente, ya sean desarrolladas por él mismo u obtenidas por terceros. Estas aplicaciones hacen uso de las librerías, los servicios y API de los niveles anteriores. Un ejemplo de estas aplicaciones puede ser Google Maps, el navegador web, el marcador de teléfono, entre otras.
- **Framework de Aplicaciones:** Es el conjunto de herramientas de desarrollo de cualquier aplicación. Para toda aplicación de Android, se utiliza el mismo conjunto de API y el mismo "framework", representado por este nivel, independientemente de si es desarrollada por Google o terceros. Los más importantes son:

View System: Conjunto de elementos que permiten construir la interfaz de usuario (GUI) (mosaicos, botones, listas, control de interfaces mediante teclado, "check-boxes", etc.)

Location Manager: brinda servicios de localización a aplicaciones que hacen uso de ella.

Activity Manager: Conjunto de API que gestiona el ciclo de vida de las aplicaciones en Android.

Notification Manager: Comunicación al usuario de los eventos que ocurren durante una ejecución: mensajes, llamadas, conexiones, ubicación, etc. Si llevan asociada alguna acción, en Android denominada Intent, ésta se activa mediante un simple clic.

Content Provider: Permite compartir los datos de una aplicación con las demás aplicaciones de Android.

Telephone Manager: Incluye todas las API vinculadas a las funcionalidades propias del teléfono.

XMPP Service: Conjunto de API para utilizar este protocolo de intercambio de mensajes basado en XML.

- **Librerías:** La siguiente capa es la de las librerías utilizadas por Android. Como se comentaba anteriormente, Android presenta un kernel basado en Linux y junto con las librerías, escritas en C/C++, conforman el corazón de Android. Entre las librerías más importantes destacan:

Librería libc: Cabeceras y funciones según el estándar C.

Librería Surface Manager: Gestiona las ventanas de aplicaciones activas y compone los elementos de navegación de pantalla.

Librería Media Libraries: Permite acceder a los códecs para el contenido multimedia soportado en Android.

Librería WebKit: Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

Open/SL y SGL: Representan las librerías gráficas que ayudan a sustentar la parte gráfica de Android, permitiendo así, manejar gráficos 3D y utilizar, en el caso de que esté disponible, el hardware destinado para ello. SGL brinda gráficos en 2D, lo que la convierte en la librería más utilizada por la mayoría de aplicaciones. Android permite hacer uso de gráficos tanto 3D como 2D de forma combinada.

Librería SQLite: Creación y gestión de bases de datos relacionales.

FreeType: Ésta librería permite trabajar de forma rápida y sencilla con diferentes tipos de fuentes.

Librería SSL: Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.

- Tiempo de ejecución de Android: Se presenta en el entorno de ejecución, de la misma forma que las librerías, y constituye las Core Libraries (librerías con clases Java y la máquina virtual Dalvik.)
- Núcleo Linux: Android utiliza el núcleo de Linux como una capa de abstracción para el hardware disponible en los dispositivos móviles. En ella, se encuentran los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Cada vez que un fabricante realiza una actualización en su hardware añadiendo nuevos elementos, se deben crear las librerías de control o drivers dentro de este kernel de Linux embebido en el propio Android.

3.1.2 Versiones

En noviembre de 2007 se produce el primer lanzamiento de Android beta. En 2008, se lanza la primera versión comercial (de prueba), Android 1.0. A partir de entonces, este sistema operativo ha visto numerosas actualizaciones en el sistema operativo original.

En estas actualizaciones se encuentran correcciones a los fallos de programa y nuevas funcionalidades. Desde abril de 2009, las versiones de Android han sido desarrolladas bajo un nombre en clave y sus nombres siguen un orden alfabético: Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow, Nougat lanzado en agosto de 2016 y el ya anunciado "Android Oreo". En la siguiente tabla se ven con más detalle.

Nombre código	úmero de versión	Fecha de lanzamiento	ivel de API
Conocido internamente como "Petit Four"	1.1	9 de febrero de 2009	2
(Sin nombre código)	1.0	3 de septiembre de 2008	1
Cupcake	1.5	25 de abril de 2009	3
Donut	1.6	5 de septiembre de 2009	4
Eclair	2.0–2.1	26 de octubre de 2009	5–7
Froyo	2.2–2.2.3	20 de mayo de 2010	8
Gingerbread	2.3–2.3.7	6 de diciembre de 2010	9–10
Honeycomb	3.0–3.2.6	22 de febrero de 2011	11–13
Ice Cream Sandwich	4.0–4.0.5	18 de octubre de 2011	14–15
Jelly Bean	4.1–4.3.1	9 de julio de 2012	16–18
KitKat	4.4–4.4.4, 4.4W–4.4W.1	31 de octubre de 2013	19–20
Lollipop	5.0–5.1.1	2 de noviembre de 2014	21–22
Marshmallow	6.0–6.0.1	5 de octubre de 2015	23
Nougat	7.0 - 7.1 - 7.1.1 - 7.1.2	15 de junio de 2016	24-25
Oreo	8.0 - 8.1	21 de agosto de 2017	26-27
P	9.0	Agosto de 2018	28

Tabla 1: Versiones Android

3.2 Enfoques

Cuando hablamos de aplicaciones que presenten un sistema de aprendizaje m-learning para la obtención del carnet de conducir, se presentan dos enfoques. Por un lado, están aquellas apps que destinadas al entrenamiento para el examen teórico

(tipo test), y por otro, aquellas que se encargan de reforzar el aprendizaje para el examen práctico.

Dentro del primer grupo, destacan aplicaciones como *Todotest*, que ofrece hasta 1.200 tests distintos para todo tipo de permiso de conducir, recibe actualizaciones de forma periódica y se encuentra disponible para IOS y Android.

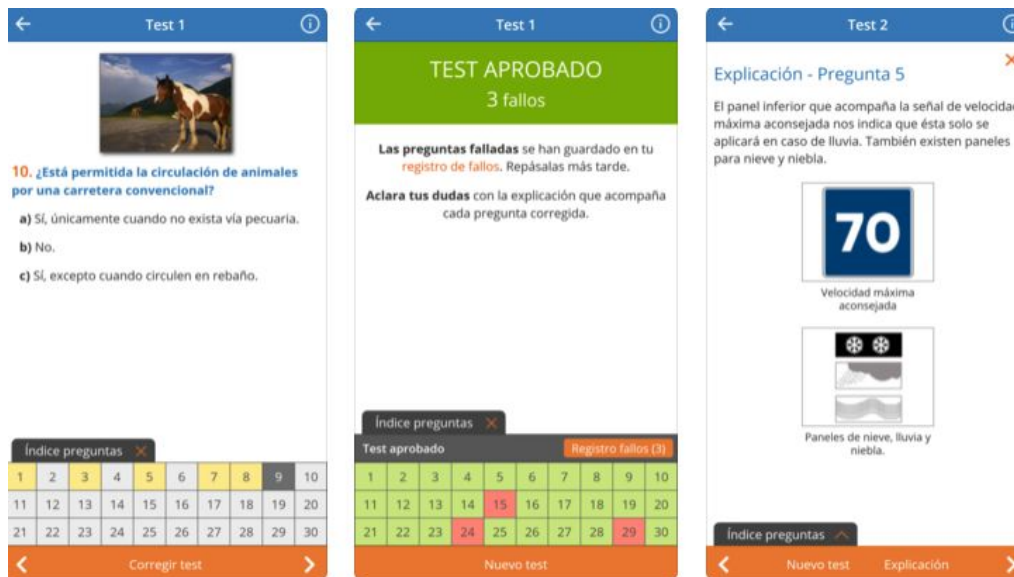


Figura 3: App Android para Test de Conducir

Otro ejemplo de este tipo de aplicaciones es *Señales de Tráfico*, a diferencia de las aplicaciones de m-learning que ofrecen al usuario la oportunidad de acceder a numerosos test, esta app tiene como fin enseñar las señales de tráfico, usando imágenes y códigos de referencia clasificados con su respectiva categoría. también está disponible para IOS y Android.



Figura 4: App Android para aprender señales de tráfico

Las dos siguientes aplicaciones están destinadas al aprendizaje de la parte práctica del carnet y lo hacen a través de la simulación. La primera de ellas es **Dr. Driving**, aplicación que permite familiarizar al usuario con el entorno de conducción incluso en el momento previo a coger el coche. Brinda una perspectiva lo más realista posible. Disponible para IOS y Android.



Figura 5: App simulador de conducir.

La segunda app es **Ultra 3D car parking**, el objetivo de esta aplicación es ofrecer un entorno de simulación para aprender a aparcar, buscando que el usuario se habitúe a las maniobras necesarias para llevar a cabo el proceso de forma correcta.



Figura 6: app para aprender a aparcar.

Capítulo 4. *Estudio previo*

Este capítulo describe el estudio realizado, previo a la implementación de DriverMoves. Se pasa a definir los requerimientos necesarios para llevar a cabo un desarrollo de la forma más óptima posible, distinguiendo entre requerimientos funcionales y no funcionales. Se expondrá la metodología utilizada y el por qué ha sido seleccionada para este proyecto, que en este caso es el desarrollo de prototipo junto con las API's utilizadas.

También se comentarán de forma breve los aspectos generales de la Base de datos, ya que más adelante se toca en profundidad, y la seguridad que proporciona firebase a toda la información almacenada en la misma.

La arquitectura que se establece es la de Modelo-Vista-Controlador para mejorar la separación de componentes y optimizar el mantenimiento de sí misma.

Se debe tener en cuenta la importancia de los siguientes pasos, ya que sentarán la base sobre la que se iniciará la implementación y tendrán gran peso en el resultado obtenido tras la fase de desarrollo.

4.1. Requerimientos

4.1.1 Requerimientos Funcionales

4.1.1.1 Gestión de Usuarios

Login: La aplicación pondrá a disposición del usuario un formulario en el que deberá rellenar los campos de correo electrónico y contraseña. Los datos son enviados a Firebase para llevar a cabo la autenticación de usuario. En caso de que los datos introducidos no sean correctos, aparecerá un mensaje en pantalla (poner mensaje). Si la validación es correcta, el Usuario podrá acceder al contenido de la aplicación.

SharedPreference: En la pantalla de Login, DriverMoves presenta un interruptor de que le brinda la opción de mantener guardados sus datos, de esta forma, una vez el usuario cierra sesión, no debe introducir de nuevo las credenciales.

Logout: Estando dentro de la aplicación, el usuario dispone de la opción de Logout en el menú desplegable de la parte izquierda de la pantalla.

Registro: En caso de que el usuario entre por primera vez a la aplicación, tiene en la parte inferior de la pantalla, una opción que lo lleva a la pantalla de registro. En ella, se presentan diferentes campos a rellenar; correo, contraseña, dirección, password y rol, una vez completados correctamente, se volcarán en la base de datos y a continuación, se redirige al usuario a la pantalla de login, donde posteriormente deberá introducir sus credenciales para poder acceder al contenido de la App.

Perfil Usuario: Cada usuario tiene a su disposición un perfil en el que se ve reflejada la información proporcionada por él mismo en el momento en el que se registró.

Control de acceso: Dependiendo si el usuario se haya registrado como alumno o profesor, en el momento en el que entre en la aplicación, tendrá acceso a determinadas funciones y contenido.

Búsqueda usuarios: Este es un requerimiento que estará disponible únicamente para el profesor. Podrá buscar cualquier usuario dado de alta en la ampliación y acceder su perfil.

4.1.1.2 Aplicación

Pantalla principal

Alumno:

- La pantalla principal posee dos pestañas. En la primera el alumno verá la lista de prácticas realizadas.
- En la segunda pestaña tendrán la opción de desplazarse a la derecha y ver el listado de exámenes.

Profesor:

- En el caso del profesor, dispone de cuatro pestañas. Por un lado, la pestaña de inicio, en ella se ve un campo para introducir el correo del alumno al que se le va a impartir la práctica, además de un botón flotante que al desplegarse permite seleccionar si lo que se va a llevar a cabo es una práctica de conducir o un examen.
- Por otro lado, al igual que el alumno, el profesor podrá ver una pestaña en la que se encuentra el listado de prácticas impartidas.
- La tercera pestaña, presenta la lista de exámenes.
- Por último, en la cuarta pestaña se encuentra con la búsqueda de usuarios.

Menú lateral

El menú lateral, es el mismo tanto para el alumno como para el profesor y dispone de las siguientes opciones:

- Inicio: Pantalla principal

- Perfil: Datos del usuario
- Ajustes: El usuario tiene la opción de silenciar las notificaciones (No funcional)
- Cerrar sesión: Cierra sesión del usuario y lo lleva al usuario a la pantalla de Login

Acceso a los datos de Prácticas

El profesor y el alumno tienen a su disposición el listado de prácticas que han impartido y recibido, respectivamente. Si se selecciona cualquiera de ellas, el usuario es llevado a una pantalla donde se proporciona toda la información relacionada con la misma.

Acceso a los datos de Exámenes

En este caso ocurre lo mismo que con las prácticas, tanto el profesor como el alumno tienen a su disposición el listado exámenes a los que ha asistido o exámenes realizados, el listado de exámenes en los que ha asistido con un alumno.

Acceso al Mapa de Ruta

Una vez se ha seleccionado una práctica o examen, en la parte inferior de la pantalla habrá un botón que lleva al usuario a un mapa. En él, que se verá marcada la ruta seguida a lo largo de la sesión (En rojo, para un examen, y en azul, para una práctica), junto con sus puntos de interés (Punto de partida, tipos de errores de conducción cometidos, lugar de finalización, etc.).

Pantalla de Registro de actividad

Ésta es una pantalla a la que únicamente tendrá acceso el profesor.

- Se despliega una botonera con los tipos de errores de conducción (saltarse un ceda el paso, sobrepasar el límite de velocidad, no respetar un semáforo, no llevar el alumbrado correcto, etc.).
- En esta pantalla, también hay un cronómetro con sus respectivos botones de “Play” y “Pause”, dándole al profesor la opción de realizar una pausa, en el caso de que fuera necesario, en medio de una sesión.
- Por último, en la parte inferior se encuentra el botón de “Fin de práctica”, una vez pulsado, el cronometro se para y se lleva al usuario a la siguiente pantalla; Un mapa con el resumen de todos los datos recogidos durante la sesión.

4.1.2 Requerimientos No Funcionales

Requisitos complementarios o atributos de calidad. Especifican criterios que juzgan operaciones del sistema en lugar de su comportamiento.

Interfaz y usabilidad

- La aplicación presentará al usuario una interfaz sencilla e intuitiva, procurando así, que la curva de aprendizaje sea la mínima posible.
- Además, se buscará ofrecer un diseño atractivo y que inste al usuario a tener una mejor experiencia e interacción con la aplicación.

Seguridad

- Se accede por medio de la autenticación al contenido de la ampliación.
- En el caso de que el usuario haya seleccionado la opción de “Recordar” a la hora de loguearse, su información se almacenará en las Shared Preferences, de esta forma, cuando cierre sesión, no hará falta que escriba nuevamente las credenciales, solo deberá pulsar “Login” para entrar.

- Su sesión se mantendrá abierta hasta que decida cerrarla, de esta forma, sólo debe autenticarse una sola vez

Rendimiento

- El acceso a la información de la base de datos se hace de forma rápida y sin suponer gran esfuerzo para el dispositivo móvil. Éste tipo de operaciones tiene respuestas de milésimas de segundo.

- En cuanto a la escritura en la base de datos, distinguimos los dos siguientes modos; Información personal del usuario, no consume muchos recursos y se hace de forma rápida, y los datos recopilados durante las prácticas, en este caso, en el momento en el que se va realizando cada sesión, los datos generados por el usuario, son volcados en la base de datos en tiempo real, con lo cual, este proceso lleva consigo un alto consumo de recursos.

4.2. Metodología de desarrollo

Una Metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo.¹ Con el paso de los años, han sido muchos los métodos desarrollados diferenciándose por su fortaleza y debilidad. Una metodología de desarrollo de software consiste en un framework (entorno o marco de trabajo) que consiste en Herramientas, modelos y métodos para asistir el proceso de desarrollo de software.

Dentro del marco de metodologías de desarrollo se encuentran distintos enfoques que dependen de las características del proyecto que se vaya a llevar a cabo. En este caso, DriverMoves sigue un desarrollo de prototipos, que pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.²

Las fases de este modelo son las siguientes:

¹ Metodología de desarrollo. okhosting.com/blog/metodologias-del-desarrollo-de-software/

² Modelo de prototipo. Mayo 11, 2018. wikipedia.org/wiki/Modelo_de_prototipos

- Comunicación
- Plan rápido
- Modelado, diseño rápido
- Construcción del Prototipo
- Desarrollo, entrega y retroalimentación
- Entrega del desarrollo final

Este modelo se caracteriza por ser útil cuando el cliente conoce los objetivos que desea alcanzar de forma general, sin embargo, no tiene claro los requisitos de entrada, procesamiento o salida.

En la siguiente figura, se refleja el modelo de prototipo aplicado a este proyecto.



Figura 7: Fases de desarrollo

4.2.1 Prototipos

A continuación, se describe brevemente las características de cada uno de los prototipos desarrollados.

En el primer prototipo se crea la interfaz de usuario. Teniendo en cuenta que existen dos roles, se implementan todas las pantallas (las que son comunes y las que están disponibles para un determinado rol) y sus funcionalidades.

El segundo prototipo consta de la base de datos totalmente funcional. También presenta una actualización de la interfaz, mejorando su diseño y mostrando información perteneciente a cada usuario.

En el tercer y último prototipo, se lleva a cabo una mejora de la interfaz y se añaden más detalles al mapa, de esta manera no solo se verán los errores cometidos por el alumno sino también marcadores que indiquen el punto de partida, de pausa y de continuación de cada sesión (examen o práctica).

4.3. Base de datos

En la base de datos se almacenará toda la información de cada usuario, así como los datos referentes a cada practica o examen (Coordenadas de rutas, marcadores, etc.).

De cara a los requerimientos de la aplicación, interesa tener acceso a una base de datos en la que se pueda escribir en tiempo real, contemplando la posible pérdida de conexión durante la escritura en la misma. También es importante que se pueda acceder a esta información en cualquier momento y lugar, ya que esto potencia la característica de este sistema de aprendizaje que hace uso del dispositivo móvil.

4.4 API's

En el caso de *DriverMoves*, se hace uso del complemento **Authentication**, que permite autenticar usuarios de forma sencilla y da la posibilidad de hacerlo, no solo con el correo electrónico, como se hace en este caso, sino con otros proveedores (Google, Teléfono, Facebook, Twitter, GitHub, Anónimo).

Realtime Database, proporciona un acceso seguro a la base de datos desde al código del cliente. Esta API, está diseñada para permitir solo operaciones que puedan ser ejecutadas rápidamente, de esta forma, mejora la experiencia del almacenamiento y sincronización datos de la app en tiempo real. Utiliza una base de datos **NoSQL** alojada en la nube, al ser no relacional, presenta ciertas ventajas en comparación con una base de datos relacional, ya que pueden manejar enormes cantidades de datos y poseen un escalamiento sencillo, entre otras. También contempla la pérdida de conexión, haciendo que los datos persistan en el disco a la espera de que ésta sea restablecida. Éstas característica son importantes, ya que son las que nos permiten ir almacenando las coordenadas conforme va cambiando la ubicación del usuario, incluso si se pierde la conexión durante el proceso. Al integrar Realtime Database con

Authentication se puede definir quién tiene acceso a los datos y cómo se accede a ellos. Éstos datos son almacenados en formato **JSON**,³ acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos.

Otra de las tecnologías que se incorpora a *DriverMoves* es la API de **Google Maps**. Es una herramienta que nos permite oficialmente la creación de mapas y su manipulación de prácticamente cualquier aspecto de la interfaz original.

Debido a la complejidad de las herramientas explicadas anteriormente, es necesario, antes de llevar a cabo la implementación e integración de las mismas, entender su funcionamiento a través del estudio de cada una de ellas. Para Firebase, se emplea tiempo en el análisis de la forma más óptima para estructurar la BBDD no relacional para la app. Una vez definida la estructura, se procede al estudio para la implementación de la misma. En el caso de Android Studio, se dedica tiempo al análisis y planificación de la implementación tanto para la parte de la funcionalidad como la parte del diseño, ya que tras el estudio realizado, se busca presentar una interfaz sencilla, que no conlleve una gran curva de aprendizaje para el usuario y sea intuitiva. Todo el estudio que se refuerza con la información que ofrecen las páginas oficiales de estas herramientas y es complementada dos cursos: *Programación de Android desde Cero +35 horas Curso COMPLETO* y *Curso Firebase para Android: Aprende a construir tu backend*, de Udemy, una plataforma de aprendizaje Online.

4.4 Seguridad

Firebase Realtime Database proporciona un lenguaje flexible de reglas basadas en expresiones con una sintaxis similar a la de JavaScript que te permite definir cómo se deben estructurar los datos, cómo se deben indexar y cuándo se pueden leer y escribir, de esta manera se tiene un control de acceso a los datos alojados en la base de datos. En combinación con los servicios de autenticación, se puede definir quiénes pueden acceder a determinados datos y proteger la información personal de tus usuarios contra el acceso no autorizado.

³ JSON. Junio 20, 2018. json.org/json-es.html

En DriverMoves, en la configuración predeterminada, las reglas de la base de datos necesitan Firebase Authentication y proporcionan permisos totales de lectura y escritura solo a los usuarios autenticados de la app, pero no para todo el mundo.

Capítulo 5. *Diseño*

A continuación, se da paso a la descripción detallada del diseño de DriverMoves. En este apartado se abordará la arquitectura implementada, las tecnologías utilizadas para su desarrollo, el diseño de pantallas, de interfaz y de la base de datos. Se justificarán las decisiones tomadas en el apartado anterior **4. Estudio previo** y se tratarán los cambios producidos, durante el proceso de desarrollo, con respecto a dicho estudio.

5.1. Arquitectura

DriverMoves se basará en Arquitectura Model-View-Presenter (MVP), de ahora en adelante MVP. Este patrón pretende abstraer nuestra interfaz de usuario de lo que es la lógica de negocio y optimizar el mantenimiento de sí misma, de esta forma los desarrolladores pueden llevar a cabo una modificación modular, sin que una parte afecte a la otra. La interacción es sencilla: el presentador, el modelo y la vista. En MVC la vista nos abstrae de la interfaz de usuario que se le presenta. Contiene elementos como textos, formularios de entrada.

El presentador escucha los eventos de la vista y comunica a la vista, ciertos cambios que puedan ocurrir en ella. Así mismo, lanza los casos de uso que actúan sobre el modelo de la lógica de negocio, es decir, administra la comunicación entre la vista y el modelo.

Y el modelo son las entidades del negocio junto con los casos de uso que en él se han implementado.

En este patrón, la vista es totalmente pasiva. El usuario realiza una serie de eventos sobre ella que son comunicados de forma inmediata al presentador y es el presentador el que decide qué acción se ha de realizar en ese momento, ya sea avanzar un caso de uso en concreto u otro.

5.2 Tecnologías

En primer lugar, se lleva a cabo un análisis previo teniendo en cuenta dos objetivos principales: contar con una app que permita guardar datos en tiempo real y, posteriormente, permita gestionarlos. Estos datos serán las coordenadas, partiendo de ellas, se traza una ruta y se señalan los puntos de interés, como errores cometidos durante cada sesión, lugar de inicio, entre otros. Sin embargo, no todos los datos serán recogidos del mismo modo, ya que, la app debe proporcionar, a cada usuario, un perfil donde se muestra información personal previamente almacenada. Con ello en mente, se lleva a cabo el desarrollo de *DriverMoves* de forma nativa, así, se aprovecha al máximo las capacidades hardware y software del dispositivo.

Tecnologías que se utilizan en el desarrollo de *DriverMoves*:

5.2.1 Android Studio

Android Studio, que es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ.⁴ Dentro de las múltiples funcionalidades que ofrece, consta de una Interfaz de usuario bastante intuitiva junto con la compleción de código que a el proceso de desarrollo de *DriverMoves*. Android Studio también hace uso de Herramientas Lint y otras inspecciones de IDE para detectar problemas de compatibilidad de versiones, de rendimiento y usabilidad entre otros, lo que ayuda enormemente a la localización y corrección rápida de errores. Se utiliza el lenguaje del sistema operativo de la plataforma, **Java**. Éste código, se usará en toda operación de comunicación con el sistema operativo o en la que se trabaje de forma directa con el hardware. Con él definimos el comportamiento de nuestra aplicación, mientras que con el lenguaje **XML**, la presentación, de esta forma conseguimos trabajar estos aspectos de forma separada. Otro punto a destacar es la administración de dependencias, el propio IDE se encarga de configurar debidamente los proyectos para el uso de las mismas.

⁴ Conoce Android Studio. Abril 25, 2018. developer.android.com

5.2.2 Firebase

Firebase llega de la mano de Google como una nueva y mejorada plataforma de desarrollo móvil en la nube, disponible para Android, iOS y web, de esta forma, queda a disposición de los usuarios una potente herramienta que facilita el desarrollo del Back-End ahorrando tiempo de desarrollo entre otras ventajas.

Dentro de la amplia gama de servicios que ofrece Firebase, se hace un breve resumen de cada uno de ellos:

Base de datos Realtime: Base de datos en tiempo real, como se comentaba en el apartado 4.4 API's. Los datos se almacenan en formato JSON y se pueden agregar reglas para permitir requests con token o solo desde una URL por ejemplo.

Autenticación: También comentado en el apartado 4.4 API's. simplifica el inicio de sesión y la gestión de la misma en nuestra aplicación.

Almacenamiento: Ideal para aplicaciones que requieren guardar datos de usuarios. Permite almacenar contenido multimedia y acceder en cualquier momento a este.

Hosting: Permite subir una aplicación y automáticamente contar con SSL y HTTP2.

Cloud Functions(BETA): El objetivo de esta herramienta es transformar el código backend en varias funciones (Dividir en piezas) creadas en NodeJS. Se genera una URL a la que se llega desde AJAX para la ejecución del código necesario.

Laboratorio de tests para Android: Pone a disponibilidad del desarrollador la posibilidad de probar los tests una aplicación en entornos previamente configurados como el dispositivo, la versión del OS, entre otros.

Informes sobre fallos: Permite poner en conocimiento los errores presentes en las aplicaciones Android e iOS y corregirlos. De cara al futuro, dará lugar a Crashlytics de Fabric.

Monitoreo de rendimiento: Mide el rendimiento de la aplicación y el tiempo que

transcurre durante los diferentes procesos para los usuarios, de tal forma, que se puedan detectar los problemas de tiempo de carga y corregirlos.

Notificaciones: Permite gestionar las notificaciones a los usuarios, teniendo la opción de programarlas según determinados parámetros.

Configuración remota: Brinda la posibilidad de modificar la aplicación sin implementar una versión nueva.

Enlaces dinámicos: Permite hacer uso de vínculos directos con atribución para impulsar el crecimiento y dar una mejor experiencia de instalación al usuario.

Admob: Dirige el tráfico de búsqueda a la aplicación. Son muchas las aplicaciones que llevan implementado este servicio.

Firestore presenta compatibilidad con los siguientes lenguajes: JavaScript (tanto en el Frontend como con Node), iOS (Swift y Objective C), Android (Java y Kotlin), Python y aplicaciones Java, C++ y Unity.

5.2.3 Ninja Mock

Ninja Mock es una herramienta online de prototipado donde los elementos presentan un acabado a mano alzada y permite crear de manera rápida maquetas y estructuras mediante bocetos de aplicaciones móviles y diseño de páginas web.

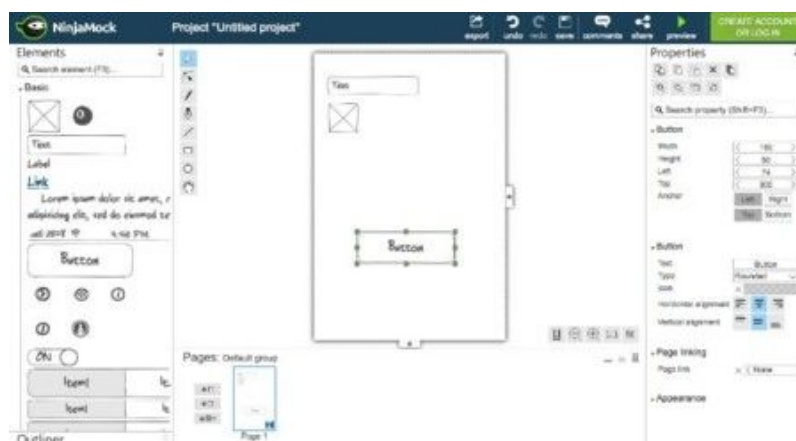


Figura 8: Captura de Ninja Mock

5.2.4 Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.⁵ Es el lenguaje nativo que utiliza Android, de esta forma, permite que una aplicación pueda comunicarse directamente con el hardware del dispositivo.

5.2.5 XML

XML es un metalenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.⁶

5.2.6 Json

JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.⁷

5.2.7 GPX

GPX (GPS eXchange Format - Formato de Intercambio GPS) es un formato utilizado por terminales GPS. Es utilizado para almacenar puntos, recorridos y rutas,

⁵ ¿Qué es el lenguaje de programación JAVA?. cs.ictea.com

⁶ Extensible Markup Language. *Febrero 20, 2018.* wikipedia.org/wiki/Extensible_Markup_Language

⁷ JSON. json.org/json-es.html

referenciado al Datum WGS84, que permite que un Sistema de Coordenadas Geográficas represente fielmente la superficie de la Tierra.

Se encuentra en texto plano, por lo que su contenido es fácil de visualizar y editar con cualquier editor de texto. Presenta un esquema de almacenamiento en XML, con etiquetas. A continuación vemos un ejemplo Real, creado a partir de una ruta trazada en Google Maps.

```
<trk>
  <name>TF-5 & Autopista del Nte. - Calle Montijos</name>
  <desc>6.2 km, 0:14</desc>
  <trkseg>
    <trkpt lat="28.39716" lon="-16.54136"></trkpt>
    <trkpt lat="28.39713" lon="-16.54128"></trkpt>
    <trkpt lat="28.39684" lon="-16.54049"></trkpt>
    <trkpt lat="28.39672" lon="-16.54014"></trkpt>
    <trkpt lat="28.39658" lon="-16.53978"></trkpt>
    <trkpt lat="28.39658" lon="-16.53978"></trkpt>
    <trkpt lat="28.39655" lon="-16.53977"></trkpt>
    <trkpt lat="28.39652" lon="-16.53976"></trkpt>
    <trkpt lat="28.39649" lon="-16.53975"></trkpt>
    <trkpt lat="28.39647" lon="-16.53973"></trkpt>
  </trkseg>
</trk>
```

Figura 9: Coordenadas Google Maps.

5.2.8 Google Maps

Es una herramienta muy conocida y utilizada en un sinfín de aplicaciones, así mismo, es una de las principales herramientas de las que hace uso *DriverMoves*. Pone a disposición del usuario un servidor de aplicaciones de mapas en Web, donde se pueden encontrar imágenes desplegadas, fotos vía satélite a nivel mundial, ubicaciones e información de las mismas, trazar rutas, ver sus características, recorrer las calles gracias al Google Street View y un gran número de servicios más. Desde Google Maps, se sacan las rutas que se han utilizado para hacer las pruebas en la simulación de *DriverMoves*.

5.2.9 GPS

Es el Sistema de Posicionamiento Global (GPS), un sistema de navegación por satélite. Está compuesto por una red de 24 satélites colocados en órbita por el Departamento de Defensa de Estados Unidos. Es un sistema que funciona en cualquier lugar del mundo y a cualquier hora.

5.2.10 Emulador Android

Android Emulator, tecnología que permite emular un dispositivo, creando un prototipo con el fin de hacer pruebas sin hacer uso de un dispositivo hardware. Brinda rapidez y potencia, acelerando así, el proceso de implementación, de esta forma, el desarrollador tiene la posibilidad de elegir el tipo de dispositivo, la versión del Sistema Operativo, entre otras características.

Una de las funciones más interesantes de cara al desarrollo de *DriverMoves*, es la de la ubicación. Es capaz de hacer una simulación sobre la ubicación actual del dispositivo y hay varias formas de hacer uso de ésta función, pero la que más se adapta a las necesidades de *DriverMoves*, es aquella que utiliza los datos geográficos a partir de un formato de intercambio de GPS. El formato de los ficheros es GPX, explicado anteriormente.

5.3. Diseño de Pantallas

A continuación, se exponen por medio de dos diagramas el diseño de pantallas según el rol del usuario. Están representados los cambios de las pantallas y los recursos que se usan en cada momento.

5.3.1 Diseño de Pantalla del Profesor

En el diagrama se puede ver cómo está diseñada la aplicación para el profesor. Tiene a su disposición el Login y la pantalla de registro que es la que permite acceso al contenido de la aplicación. En el Login se utiliza la API de Authentication y en ambas, la base de datos de Firebase. Se muestran las distintas opciones disponibles en la pantalla de inicio; Menú lateral, empezar una sesión (práctica o examen), ver el listado de prácticas y exámenes impartidos por el profesor y la búsqueda de un usuario registrado en la aplicación, cada una de estas presenta interacción con la Base de datos de Firebase para volcar los datos en la aplicación. También se ven las pantallas que tienen interacción con la API de Google Maps, en ellas se despliega el mapa con la información almacenada en la base de datos, rutas, marcadores y demás.

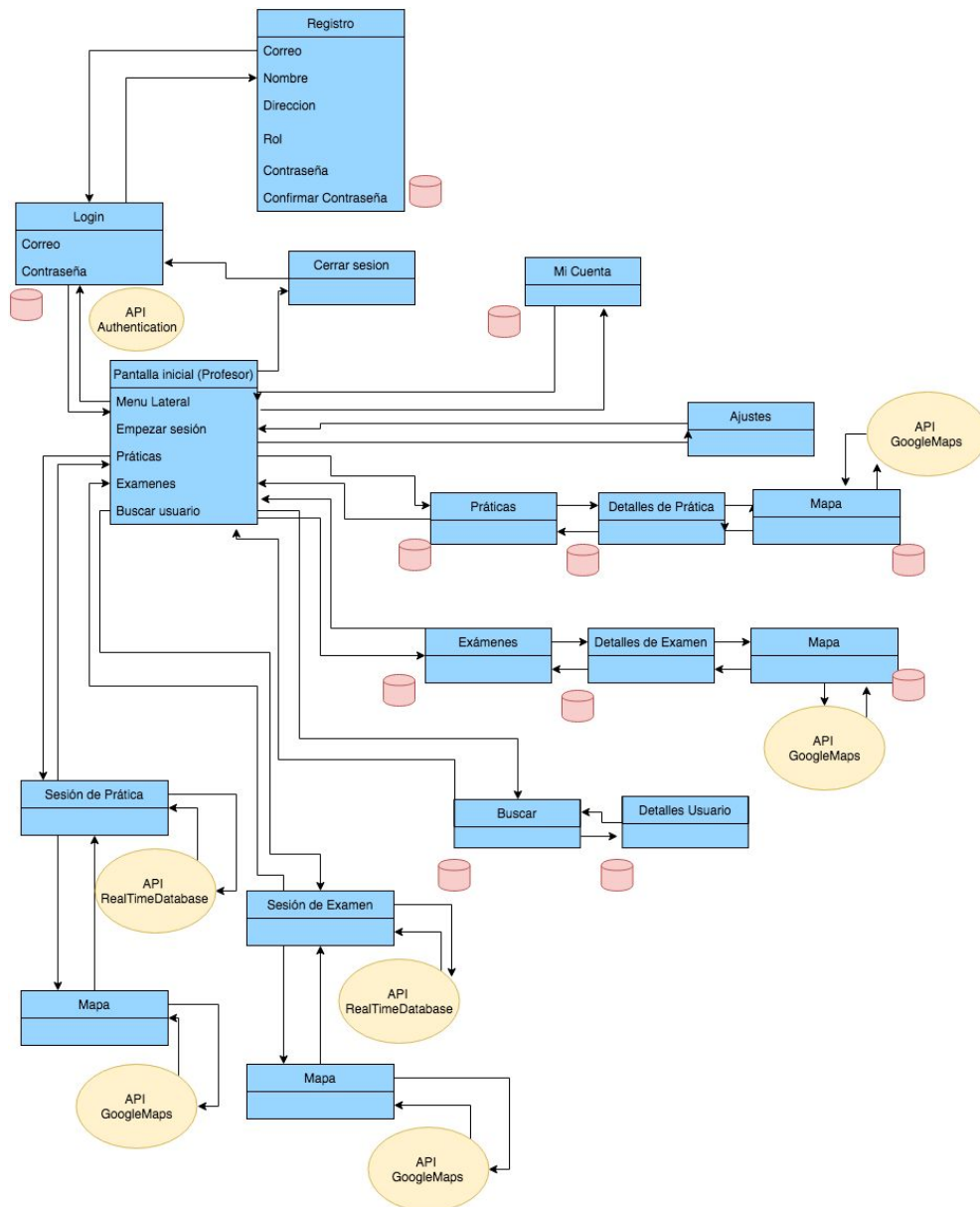


Figura 10: Diagrama Pantalla Profesor.

5.8.2 Diseño de Pantalla del Alumno

En el caso del alumno se presentan menos pantallas, ya que este no tiene la opción de impartir una sesión. El alumno puede ver su progreso por medio de las prácticas y exámenes que ha realizado, no solo tiene acceso a los detalles de cada sesión como la fecha, la duración, quien le dirige la práctica y demás, sino que también tiene acceso al mapa, por medio de la a API de Google Maps.

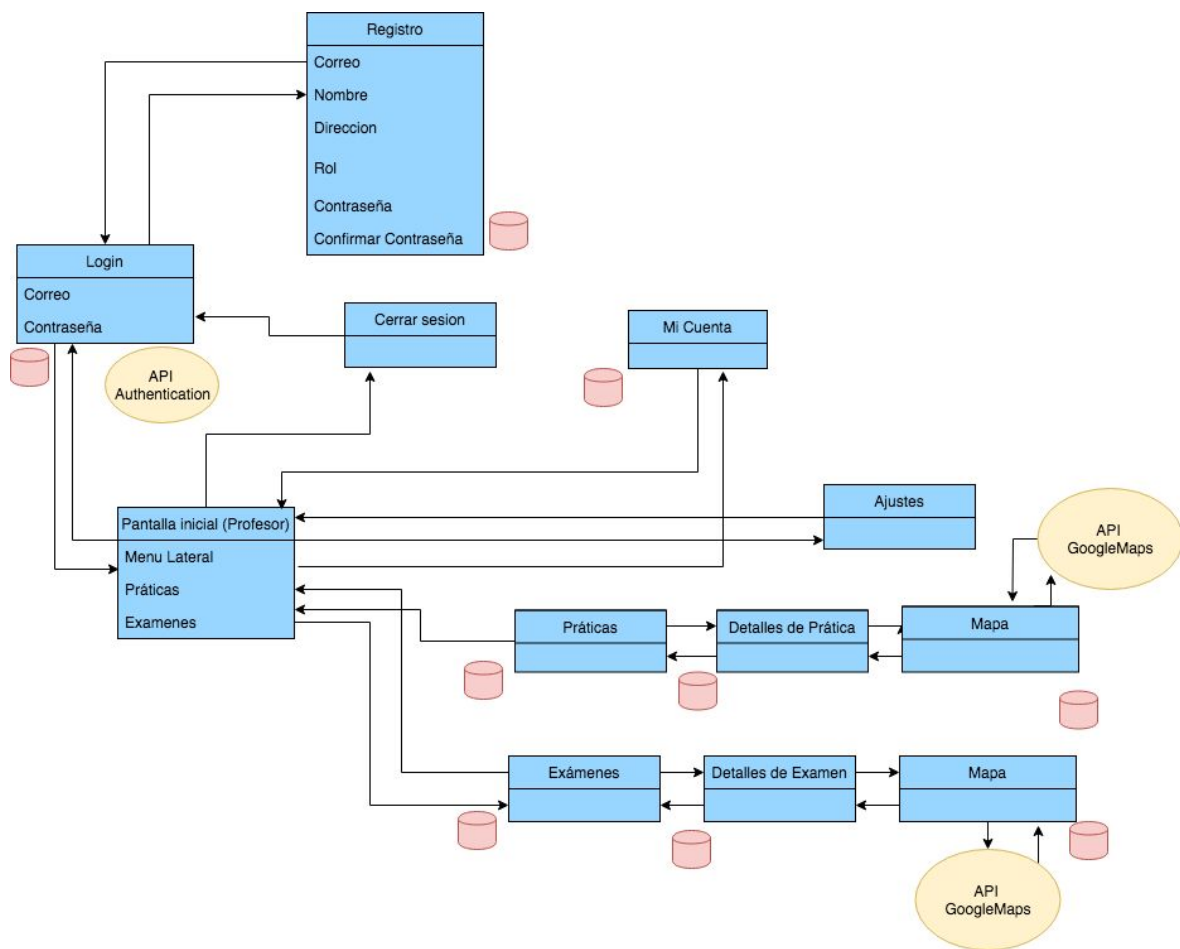


Figura 11: Diagrama Pantalla Alumno.

5.9. Diseño de la Interfaz

El diseño de la interfaz se ha llevado a cabo teniendo en mente como objetivos presentar un diseño sencillo e intuitivo al usuario, siguiendo los patrones de diseño. Para preparar el diseño, se hizo uso de la herramienta NinjaMock, anteriormente descrita.

En primer lugar, se encuentran las pantallas de Login y de Registro. Esta interfaz es la misma tanto para profesores como para alumnos. En el caso de que el usuario no esté registrado en **DriverMoves**, tiene la opción de hacerlo y luego volver a la pantalla de Login a introducir sus credenciales.

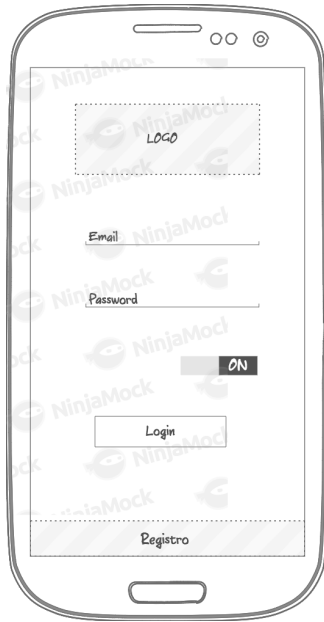


Figura 12: Diseño Pantalla Login.



Figura 13: Diseño Pantalla Registro

Una vez se ha hecho Login, tenemos dos versiones de pantalla inicial. En primer lugar, en la versión del profesor muestra directamente la opción en pantalla de introducir el nombre del alumno al que se va a evaluar o a impartir una práctica (*Figura 14*). En la parte inferior se encuentra un botón desplegable que nos permite diferenciar entre esas dos opciones. En el caso del alumno, la pantalla inicial muestra directamente el listado de prácticas que ha realizado, ya que este no cuenta con las opciones de impartir sesiones o buscar usuarios.

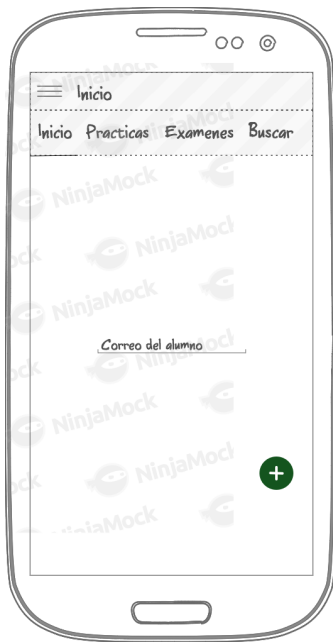


Figura 14: Diseño Pantalla Inicio

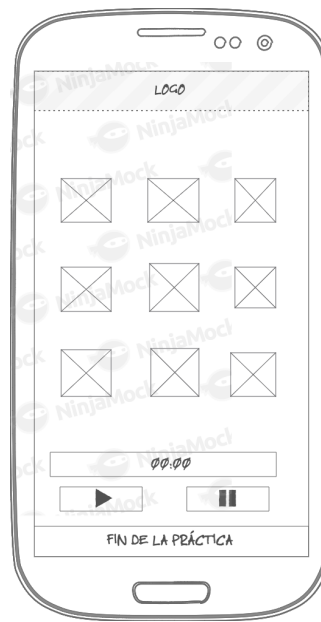


Figura 15: Diseño Pantalla de Sesión

Cuando se introduce el correo del alumno se pasa la pantalla de los botones. (Figura 15) (es la misma para evaluar una práctica o examen.) Una vez en ella, da comienzo la sesión, y el profesor que la esté impartiendo es el encargado de indicar, a lo largo del trayecto, los errores cometidos por el alumno. Cuando esta termina, al presionar el botón de “Fin de la práctica”, se despliega un mapa en el que se encuentra un resumen de todos los datos generados. (Figura 16).

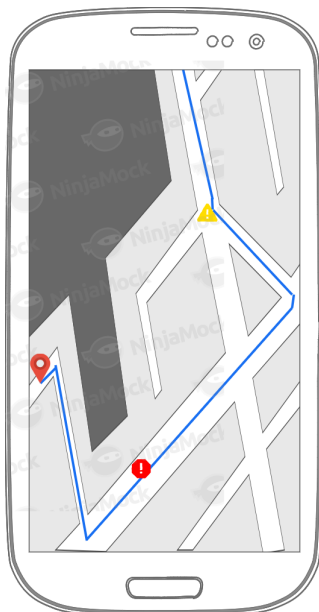


Figura 16: Diseño Pantalla Mapa

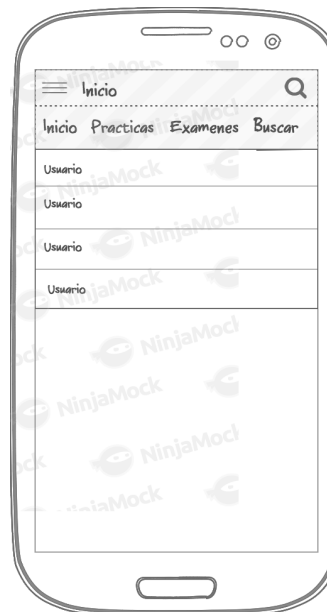


Figura 17: Diseño Pantalla de Búsqueda

Otra de las pantallas que pertenece únicamente al perfil del profesor es la de Buscar, en ella se puede acceder al perfil de cualquier usuario registrado en la aplicación. (Figura 17)

Las siguientes pantallas son las del listado de prácticas y exámenes pertenecientes al usuario logueado. Como se comentaba anteriormente, el listado de prácticas es la pantalla de inicio del alumno (Figura 20)

Vista desde el perfil del profesor:

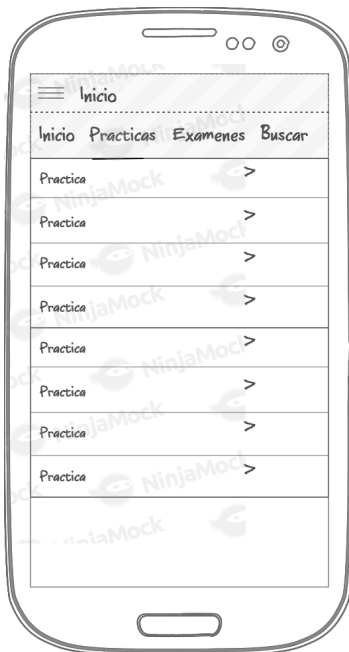


Figura 18: Diseño Pantalla Prácticas Profesor

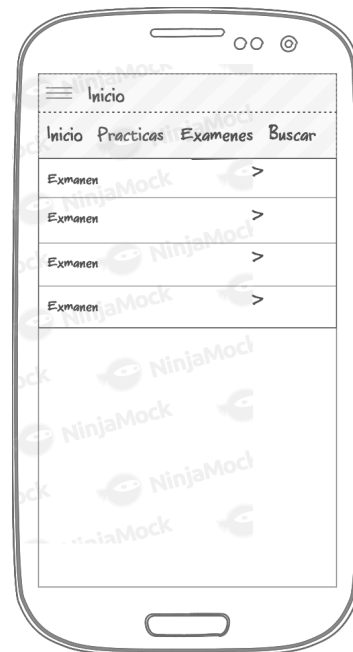


Figura 19: Diseño Pantalla Exámenes Profesor

Vista desde el perfil del alumno:

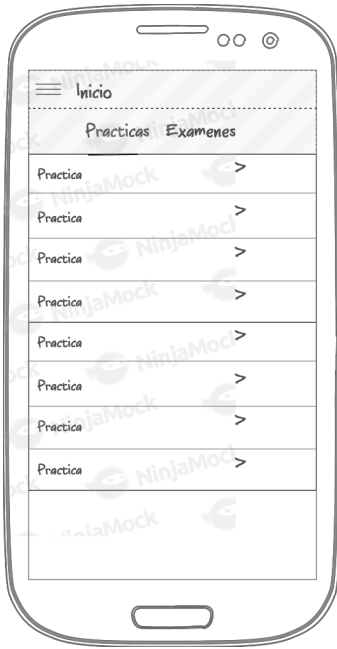


Figura 20: Diseño Pantalla Prácticas Alumno

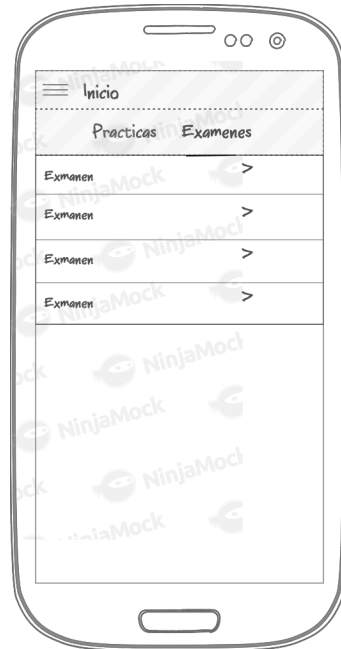


Figura 21: Diseño Pantalla Exámenes Alumno

Una vez se selecciona una práctica o examen la pantalla de detalles es la misma para ambos roles, siguiendo la misma estructura y diferenciando si se trata de un tipo u otro. En la zona inferior de esta pantalla se tiene la opción de desplegar el mapa que muestra el recorrido y demás datos de la sesión. (Figura 22)

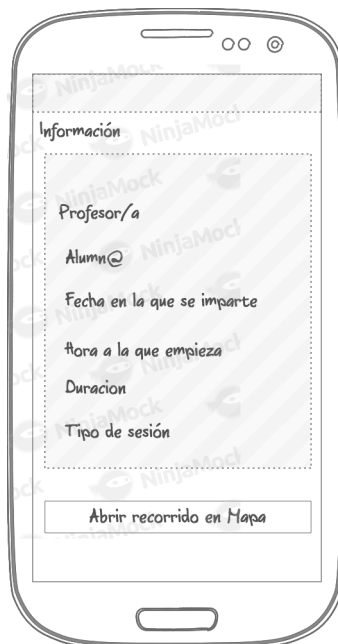


Figura 22: Diseño Pantalla Datos de una Sesión

A continuación, se expondrán las opciones de Menú lateral, que en este caso también es igual para ambos perfiles. Como se puede ver en la figura (Figura 22), el menú lateral consta de cuatro opciones; la primera de ellas es la de inicio, que lleva a la pantalla principal dependiendo del tipo de rol, la segunda es “Mi cuenta”, en ella se ve la información del usuario logueado (Figura 23), la tercera opción es “Ajustes” (Figura 24), y por último se tiene la opción de “Cerrar sesión”, una vez seleccionada, el usuario es llevado a la pantalla de Login.

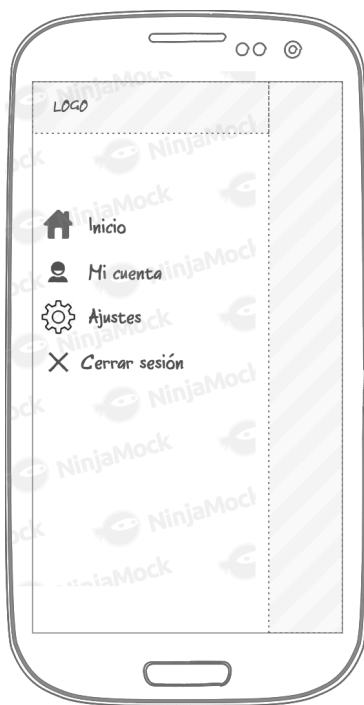


Figura 23: Menú Lateral

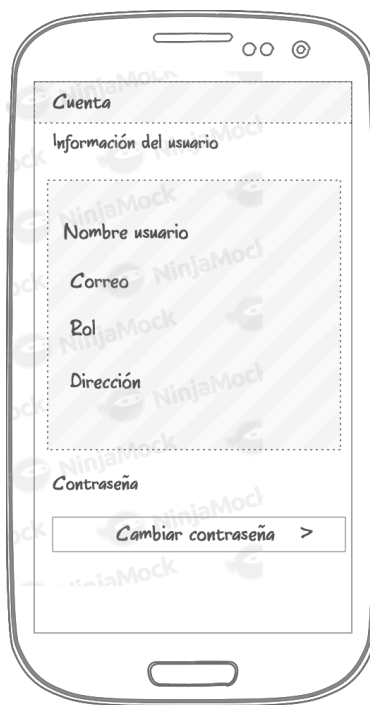


Figura 24: Cuenta Usuario



Figura 25: Ajustes

5.10. Diseño de la Base de Datos

El siguiente apartado tratará el tema del diseño de la base de datos de DriverMoves, que como se ha comentado anteriormente, viene de la mano de Firebase. Esta herramienta trabaja con una arquitectura de datos cuya estructura se basa en JSON. En este caso, no hay tablas ni registros, a diferencia de una base de datos de SQL. Los

datos son almacenados como objetos JSON y se convierten en un nodo con clave asociada de la estructura del árbol.

A continuación, se explicará la estructura diseñada para las prácticas en la base de datos. Como se puede ver en la *figura 25*, cada práctica se guardará en la base de datos con un ID único, estructurado de la siguiente manera: “practica_correoDelAlumno_fecha”. De esta forma, no habrá otra práctica igual.

Dentro de cada práctica se encuentran los datos de la misma; la duración, la fecha en la que se imparte, la hora, el alumno que la recibe, el profesor que la imparte, el tipo (práctica en este caso) y los datos más relevantes: las coordenadas y los marcadores. En el caso de las coordenadas se van creando conforme los nuevos datos son enviados a la base de datos, de tal forma que se van organizando según llegan. Cada una de estas coordenadas contiene una longitud y una latitud. En el caso de los marcadores se sigue un recuento parecido, según se generan, va aumentando el contador, sin embargo, dentro de ellos encontramos una coordenada, compuesta por longitud y latitud y también un tipo, este atributo indica que tipo de marcador es (stop, ceda el paso...).

```
"Practices" : {
  "practica_correoAlumno_fecha" : {
    "coordenadas" : {
      "coordenada_0000000000" : "Lat, Long",
      "coordenada_0000000001" : "Lat, Long",
      "coordenada_0000000002" : "Lat, Long",
      "coordenada_0000000003" : "Lat, Long",
      "coordenada_0000000004" : "Lat, Long",
      "coordenada_0000000005" : "Lat, Long"
    },
    "duracion" : "",
    "fecha" : "",
    "hora" : "",
    "marcadores" : {
      "marcador_0000000000" : {
        "coordenadas" : "Lat, Long",
        "tipo" : ""
      },
      "marcador_0000000001" : {
        "coordenadas" : "Lat, Long",
        "tipo" : ""
      },
      "marcador_0000000002" : {
        "coordenadas" : "Lat, Long",
        "tipo" : ""
      }
    },
    "nomAlumno" : "",
    "nomProfesor" : "",
    "tipo" : ""
  }
},
```

Figura 26: Estructura JSON Prácticas

En el caso de los exámenes la estructura es exactamente igual, salvo por incluir en los ID la palabra “examen” como se puede ver en la figura (Figura 26).

```
{
  "Exams": {
    "examen_correoAlumno_fecha": {
      "coordenadas": {
        "coordenada_0000000000" : "Lat, Long",
        "coordenada_0000000001" : "Lat, Long",
        "coordenada_0000000002" : "Lat, Long",
        "coordenada_0000000003" : "Lat, Long",
        "coordenada_0000000004" : "Lat, Long",
        "coordenada_0000000005" : "Lat, Long"
      },
      "duracion": "",
      "fecha": "",
      "hora": "",
      "marcadores": {
        "marcador_0000000000" :{
          "coordenadas": "Lat, Long",
          "tipo": ""
        },
        "marcador_0000000001" :{
          "coordenadas": "Lat, Long",
          "tipo": ""
        },
        "marcador_0000000002" :{
          "coordenadas": "Lat, Long",
          "tipo": ""
        }
      },
      "nomAlumno": "",
      "nomProfesor": "",
      "tipo": ""
    }
  }
}
```

Figura 27: Estructura JSON Exámenes

Por último, en la figura 27 se presenta la estructura diseñada para cada usuario. El correo el alumno será el ID único para cada uno de ellos y contará con los atributos que son su información personal; dirección, nombre, rol, y correo.

```
66   "Users" : {
67     "Correo del usuario" : {
68       "address" : "",
69       "email" : "",
70       "name" : "",
71       "rolUsuario" : ""
72     }
73   }
74 }
```

Figura 28: Estructura JSON Usuarios

Capítulo 6. *Implementación*

En el presente capítulo se hará un breve desglose de los aspectos más importante de la implementación de *DriverMoves*. Para llevarla a cabo se hace uso del emulador integrado en Android Studio AVD (Android Virtual Device) que como su nombre indica, permite virtualizar un dispositivo móvil y ofrece la libertad de elegir ciertas características de dicho dispositivo como se ha comentado anteriormente. Durante el proceso de implementación se toma como referencia los puntos definidos durante la fase del diseño.

En primer lugar, se comentará la implementación de aplicación en sí, se presentará como se han desarrollado sus diferentes funcionalidades y cómo es integrada con la base de datos de Firebase, cuya estructura ha sido previamente establecida. Después de esto, se hablará de la implementación de su interfaz.

6.1 Implementación de la app

6.1.1 Gestión de usuario

Dentro de la gestión de usuario, por un lado, DriverMoves cuenta con las opciones de Login, de Registro y Cierre de sesión. Por otro lado, cuenta con el acceso a la información personal de cada usuario registrado en la aplicación. A estos datos solo pueden acceder aquellos cuyo rol sea el de profesor.

En primer lugar, en el registro, el usuario debe introducir todos los datos que se especifican para que estos sean volcados en la base de datos, teniendo en cuenta una serie de requisitos que deben cumplir, como el formato del correo o el tamaño de la contraseña, la cual debe ser introducida dos veces para comprobar que coincidan.


```

//Comprobamos si el Email introducido es valido
private boolean emailValido (String email){
    //Aquí debemos buscar si el usuario existe, en el caso de que así sea, devolver true
    //return CheckUser(email, password);
    return !TextUtils.isEmpty(email) && Patterns.EMAIL_ADDRESS.matcher(email).matches();
}

//Comprobamos si el Password introducido es valido
private boolean passwordValido (String password){
    return password.length() >= 4;
}

```

Figura 29: Código de Comprobación de Email y Contraseña

```

 mAuth.createUserWithEmailAndPassword(emailWith, password)
    .addOnCompleteListener( activity: RegistroActivity.this, (task) -> {
        if (task.isSuccessful()) {
            DatabaseReference mDatabase = FirebaseDatabase.getInstance().getReference().child("Users");
            DatabaseReference currentUserDB = mDatabase.child(emailKey);
            currentUserDB.child("name").setValue(name);
            currentUserDB.child("email").setValue(emailWith);
            currentUserDB.child("address").setValue(address);
            currentUserDB.child("rolUsuario").setValue(rol);
            Toast.makeText( context: RegistroActivity.this, text: "Tu usuario ha sido registrado con éxito!", Toast.LENGTH_SHORT)
                .show();
            goToLogin();
        } else {
            Toast.makeText( context: RegistroActivity.this, text: "Error en el registro, compruebe que la contraseña tenga más
        }
    });

```

Figura 30: Código de Registro de usuario en la Base de datos

Una vez la operación finaliza correctamente vuelve a la pantalla de Login para introducir el correo y contraseña. A continuación, se hace una llamada a la base de datos. Aquí se utiliza del API de Firebase, Authentication, que nos proporciona la confirmación de las credenciales.

```

private void login (String email, String password){
    mAuth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener( activity: this, (task) -> {
            if (task.isSuccessful()) {
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG, msg: "signInWithEmail:success");
                FirebaseUser user = mAuth.getCurrentUser();
                updateUI(user);
            } else {
                // If sign in fails, display a message to the user.
                Log.w(TAG, msg: "signInWithEmail:failure", task.getException());
                Toast.makeText( context: LoginActivity.this, text: "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
                updateUI( user: null);
            }
        });
}

```

Figura 31: Código de Comprobación de credenciales

Otra de las opciones implementadas es la de Cierre de sesión. En el caso de que el usuario lo desee, desplegará el menú lateral donde se encuentra esta opción. Una vez seleccionada, es dirigido a la pantalla de Login.

```

private void logOut (){
    //borrarPref();
    USER_DATA_EXAMENES.clear();
    USER_DATA_PRACTICAS.clear();
    Intent intent = new Intent( packageContext: MainActivity.this, LoginActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(intent);
}

```

Figura 32: Código de Cierre de Sesión

Una de las características que se han añadido es la opción de “Recordar”, de esta forma antes de hacer Login, si es accionada, al cerrar sesión, los datos del usuario se guardarán, de tal forma que si desea acceder nuevamente a la aplicación solo tendrá que presionar el botón de Login, sin necesidad de escribir sus credenciales. Si por el contrario no selecciona esta opción, una vez cierre sesión, si desea volver a entrar, debe introducir sus datos de nuevo.

```

private void guardPref (String email, String password){
    if(switchRemember.isChecked()){
        SharedPreferences.Editor editor = sharedPref.edit();
        editor.putString("email", email);
        editor.putString("password", password);
        editor.apply();
    }else{
        sharedPref.edit().clear().apply();
    }
}

```

Figura 33: Código para las Shared Preferences

Lo siguiente que se tratará, es el acceso al perfil de cada usuario. En este caso, dentro de la opción de buscar, se hace una llamada a la base de datos para desplegar el listado con todos los correos de los usuarios registrados en DriverMoves, tanto alumnos como profesores.

```

DatabaseReference UsersRef = FirebaseDatabase.getInstance()
    .getReference( s: "Users").child(USER_DATA_ID_MAIN.get(i));
UsersRef.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        User usuairo = dataSnapshot.getValue(User.class);
        if(usuairo != null){
            USER_DATA_EMAIL_MAIN.add(new String(dataSnapshot.child("email").getValue().toString()));
        }
    }
    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
}

```

Figura 34: Traer el Listado de Usuarios de La BDD

En el caso de que uno sea seleccionado, se vuelve a hacer una llamada y se traen los datos.

6.1.2 Gestión de Prácticas y Exámenes

Una vez dentro de la aplicación el usuario tiene acceso a las prácticas y exámenes en los que han intervenido. En el caso del alumno, accederá a aquellas sesiones que le han sido impartidas y en el caso del profesor, a aquellas prácticas que ha impartido y exámenes en los que haya estado presente.

Cuando un usuario selecciona una de las prácticas que se ven es su lista, se hace una llamada a la base de datos para traer toda la información relacionada con ella, como se ve en la *figura 35*.

```
DatabaseReference UsersRef = FirebaseDatabase.getInstance()
    .getReference( s: "Practices").child(key);
UsersRef.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        User usuario = dataSnapshot.getValue(User.class);
        if (usuario != null) {
            nomAlumno.setText(dataSnapshot.child("nomAlumno").getValue().toString());
            nomProfesor.setText(dataSnapshot.child("nomProfesor").getValue().toString());
            fecha.setText(dataSnapshot.child("fecha").getValue().toString());
            hora.setText(dataSnapshot.child("hora").getValue().toString());
            duracion.setText(dataSnapshot.child("duracion").getValue().toString());
            tipo.setText(dataSnapshot.child("tipo").getValue().toString());
        }
    }
    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
});
```

Figura 35: Traer el Información de una Sesión de La BDD

En esta misma pantalla, en la zona inferior, el usuario tiene la opción de desplegar el mapa, en tal caso, se vuelve a hacer una llamada a la base de datos pidiendo los datos (*Figura 36*).

```
DatabaseReference databaseReference = FirebaseDatabase.getInstance()
    .getReference( s: "Practices");
Query myTopPostsQuery = databaseReference.child(practicaId)
    .child("coordenadas")
    .orderByKey();
myTopPostsQuery.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        if (COORDENADAS_DATA.isEmpty()) {
            for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                //añadimos un try/catch por si hay elementos erroneos
                try {
                    COORDENADAS_DATA.add(StringToLatLng(snapshot.getValue().toString()));
                } catch (DatabaseException ignored) {
                }
            }
            Toast.makeText( context: DatosPracticaActivity.this, text: "Tamaño vector coordena
        }
    }
    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
});
```

Figura 36: Traer el Información de un Mapa de la BDD

Esta información se pasa al `MapsActivity.java`, en el que se cargarán la ruta junto con los marcadores, punto de partida, punto de pausa y punto de finalización.

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    EntradaMap = true;

    // Add a marker and move the camera
    CameraUpdate zoom = CameraUpdateFactory.zoomTo( v: 12);

    //Creamos los Marker de los errores
    if(listadoMar != null ) {
        for (int i = 0; i < listadoMar.size(); i++) {
            Marker mMarker = mMap.addMarker(listadoMar.get(i));
            mMap.moveCamera(CameraUpdateFactory.newLatLng(new LatLng(listadoMar.get(i).lat, listadoMar.get(i).lon)));
            mMap.animateCamera(zoom);
        }
    }

    //Geocoder recoge la informacion. Será capaz de darnos info dependiendo de
    // la lat y long que le pasemos
    geocoder = new Geocoder(getApplicationContext(), Locale.getDefault());
}
```

Figura 37: Despliegue del Mapa y Creación de Marcadores

En el caso de los exámenes se sigue la misma estructura, ya que las llamadas a la base de datos y el despliegue del mapa son iguales, salvo que se especifica que es un examen los que el usuario ve.

6.1.2 Gestión de una Sesión

En este caso, únicamente un usuario con el rol de profesor puede empezar una sesión. Lo primero que debe hacer es escribir el correo del alumno en el campo correspondiente. Es importante que el GPS esté activado, en caso de que no lo esté, salta un mensaje en pantalla (Figura 37) y lo lleva directamente a la ventana correspondiente para activarlo (Figura 38).

```
private void GPSyNextExamen (){
    if (!locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
        showAlert();
    }
}
```

Figura 38: GPS no activado

```

private void ShowAlert(){
    new AlertDialog.Builder(getContext())
        .setTitle("GPS Signal")
        .setMessage("GPS no activado, desea activarlo?")
        .setPositiveButton( text: "OK", (dialog, which) -> {
            // GPS no activado. Entonces el siguiente
            // intent nos lleva a una ventana de las settings de android
            // para poder activar la señal GPS
            Intent intentGps = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            startActivity(intentGps);
        })
        .setNegativeButton( text: "CANCEL", listener: null)
        .show();
}

```

Figura 39: Mensaje de Alerta para activar GPS

Una vez es activado el GPS e introducido el correo del alumno correctamente, se pasa a la pantalla de sesión. Una de las opciones que tiene DriverMoves, es que el alumno que se introduzca no tiene que estar dado de alta en la base de datos necesariamente, es decir, el profesor puede asociar una sesión a un correo de un alumno, ésta se guardará en la base de datos con su respectiva información y en el momento en el que el alumno decida darse de alta, ya contará con estos datos a su disposición.

Cuando ya el profesor tiene delante la pantalla con los botones que indican los distintos tipos de errores, la aplicación ya ha empezado a recoger datos; La hora en la que empezó, las coordenadas de la posición en la que se encuentra y así continúa recogiendo información hasta que la sesión se dé por finalizada. Toda esta información es enviada en tiempo real a la Base de datos de Firebase. En caso de perder conexión, los datos se siguen almacenando hasta que puedan ser enviados.

Cada vez que se pulsa uno de los botones que se encuentran en la pantalla, se sigue la misma estructura; se especifica el tipo, se crea el marcador con todas sus características, se añade a la lista de marcadores y luego es enviado a la base de datos.

```

imgBtnsemaforo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String type = "semaforo";

        if(currentLocation == null) {
            Toast.makeText(context: PracticaActivity.this, text: "Todavía no tenemos ubicación");
        } else {
            Toast.makeText(context: PracticaActivity.this, text: "Error, Se ha saltado la ubicación");
            BitmapDrawable bitmapdraw = (BitmapDrawable) getResources().getDrawable(R.drawable.semaforo);
            Bitmap b = bitmapdraw.getBitmap();
            final Bitmap smallMarker = Bitmap.createScaledBitmap(b, width, height, true);

            //Se crea al marcador
            MarkerOptions mar = new MarkerOptions()
                .position(new LatLng(currentLocation.getLatitude(), currentLocation.getLongitude()))
                .icon(BitmapDescriptorFactory.fromBitmap(smallMarker));

            //Se añade a la lista de marcadores
            listadoMar.add(mar);
            //Subimos el marcador a la BBDD
            WriteMarker(mar, type);
        }
    }
});

```

Figura 40: Ejemplo de creación de un Marcador

En el caso de las coordenadas, cada vez que estas son actualizadas, son añadidas a un array tipo LatLng y a un array que convierte esa variable LatLng en String, un array es utilizado para enviar directamente la información al mapa que se despliega después con un resumen de la sesión y el otro, para ser volcado en la base de datos. Una vez hecho esto, el contador aumenta una unidad. Este contador es necesario ya que es utilizado para que las coordenadas se introduzcan en el orden que se desea, así a la hora de traer la información de Firebase, ya está organizada.

```

@Override
public void onLocationChanged(Location location) {
    currentLocation = location;
    LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());
    //Guardamos en array de LatLng
    coordenadasRuta.add(latLng);
    //Guardamos en array de String
    coordenadasRutaString.add(LatLngToString(latLng));
    //Subimos a BBDD
    String aux = LatLngToString(latLng);
    WriteCoodenada(aux);
    NumeroCoodenadas++;
}

```

Figura 41: Función que toma las coordenadas de la posición actual, una vez ésta cambie.

Cuando se selecciona el fin de la sesión, toda la información generada es enviada al MapsActivity.java, de la misma forma que se explicó en el apartado 6.1.2 Gestión de Prácticas y Exámenes, cuando el mapa es desplegado.

6.2 Implementación de la interfaz

En este apartado, se tratará las partes más relevantes de la implementación llevada a cabo en la interfaz de DriverMoves.

6.2.1 Login y Registro

La pantalla de Login (*Figura 42*) es implementada de tal forma que se presenta ante el usuario una interfaz sencilla e intuitiva. En ella se encuentran dos campos en los que se introducen correo y contraseña. También se implementa un Switch que es el que permite activar las Shared Preferences, de tal forma que se recuerden las credenciales del usuario Logueado. Por último, está el botón de Login y la opción que lleva a la pantalla de registro en caso de que el usuario no esté dado de alta en la base de datos.

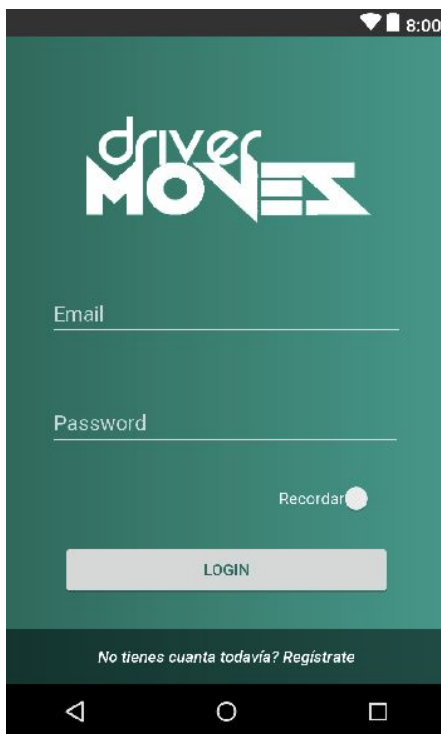


Figura 42: Pantalla de Login

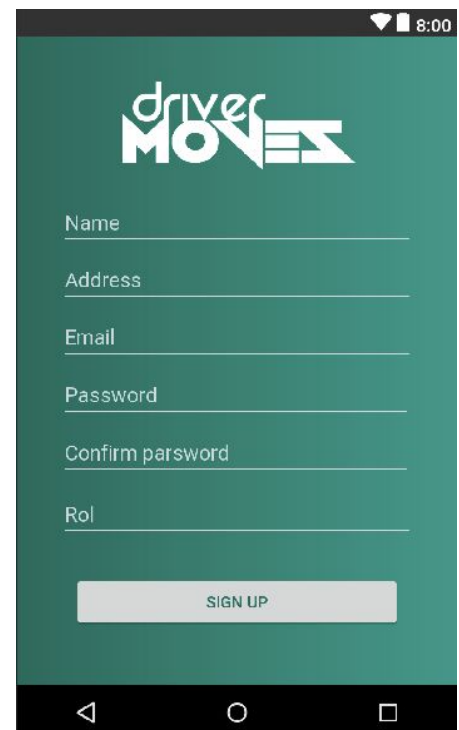


Figura 43: Pantalla de Registro

En la pantalla de registro (*Figura 43*), el usuario se encuentra con una serie de campos que ha de rellenar para dar por completado este proceso. Una vez se haya logueado correctamente, es devuelto a la pantalla de inicio para poder seguir con el Login.

6.2.2 Pantalla principal

La pantalla principal está compuesta por un conjunto de Fragments en los que se implementan las distintas funcionalidades. Como se ha comentado en capítulos anteriores, la pantalla principal puede variar según el rol del usuario, pero en ambos casos que siguen el mismo patrón de diseño.

6.2.2.1 Inicio de sesión

Esta pantalla está solo disponible para el usuario que tenga el rol de profesor. Como se puede ver en la imagen (*Figura 44*), se presenta un campo en el que debe introducir el correo del alumno y a continuación en la zona inferior derecha, se implementa un Floating Button que despliega las dos opciones disponibles, Práctica y Examen.

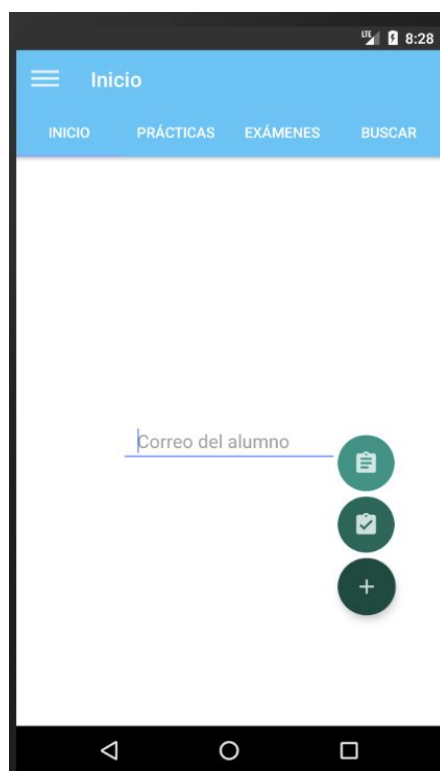


Figura 44: Pantalla de Inicio Profesor

6.2.2.2 Prácticas y Exámenes

Estas dos pantallas están disponibles para todos los usuarios, la única diferencia es que el alumno no ve los Fragments de “INICIO” y “BUSCAR”.

Como se puede apreciar en ambas imágenes, se listan todas las sesiones identificadas por el tipo, correo del usuario y fecha en la que se realizó.

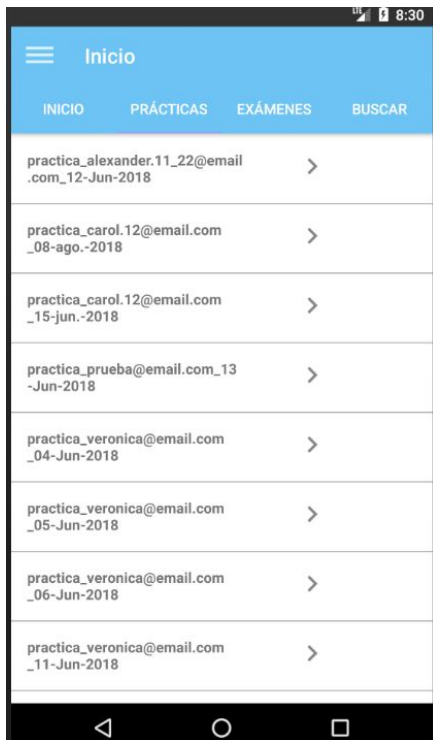


Figura 45: Listado Prácticas

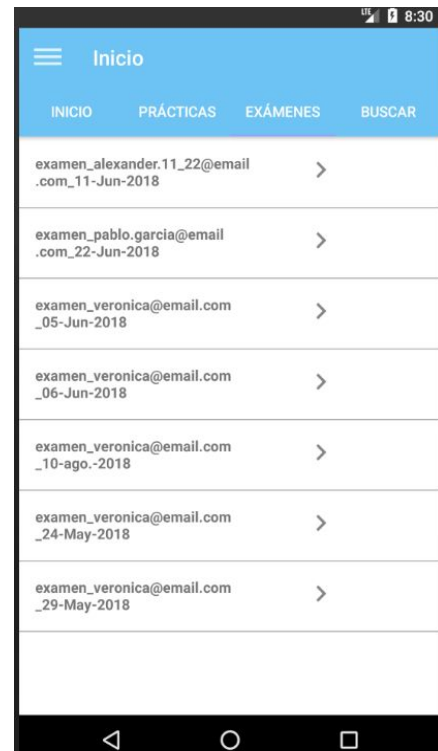


Figura 46: Pantalla Listado Exámenes

6.2.2.3 Búsqueda

En este caso se implementa la búsqueda sobre un listado de usuarios que previamente se trae de la base de datos de Firebase.

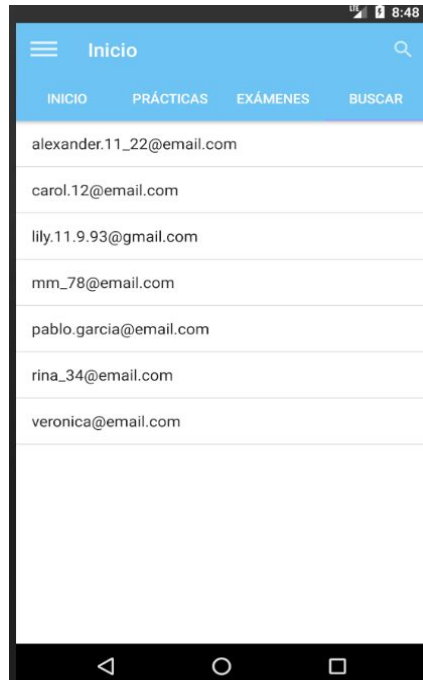


Figura 47: Búsqueda Usuarios

6.2.3 Pantalla de datos de una sesión

En la pantalla de una sesión, ya sea práctica o examen, se implementan los botones que indican los distintos tipos de errores. También se implementa el marcador en la zona inferior su respectivo botón de “Play” y “Pause”. Por último, el botón que da por finalizada la sesión y da paso al mapa con el resumen.

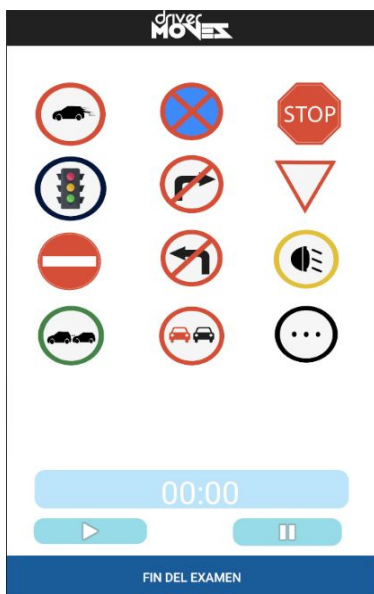


Figura 48: Pantalla de un Examen

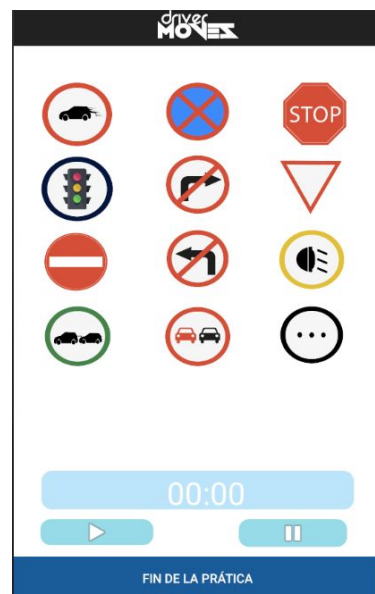


Figura 49: Pantalla de una Práctica

6.2.4 Pantalla de Mapa

La pantalla de mapa cuenta con toda la información relacionada a cada sesión, como se puede ver en las imágenes (*Figura 50 y Figura 51*), la polyline indica el recorrido que se ha hecho, el marcador azul representa el punto inicial, el gris, el lugar donde se ha hecho una parada o pausa y el rojo, el lugar donde finaliza la práctica. A lo largo de la ruta también se ven los distintos tipos de marcadores y los errores que cada uno representa.

En este caso, se hace una distinción entre una práctica y un examen. Para las prácticas, la polyline es de color azul y para los exámenes, de color rojo, con esto se busca identificar de manera más rápida y clara el tipo de sesión.

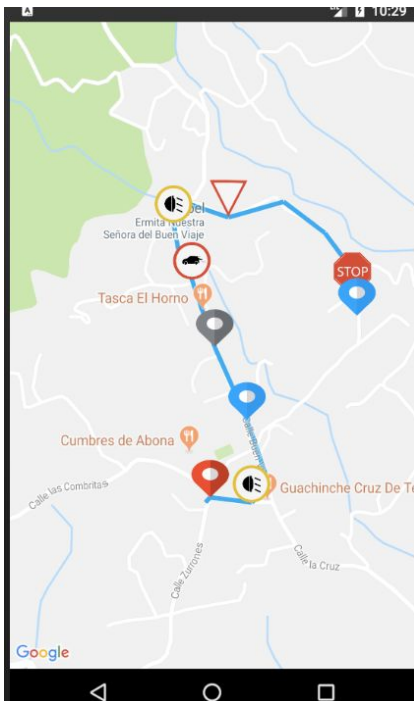


Figura 50: Ruta de un Examen

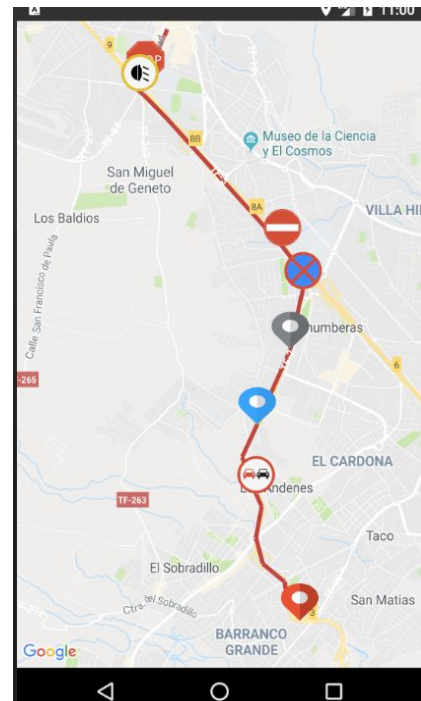


Figura 51: Ruta de una Práctica

Capítulo 7. *Pruebas*

Se expondrán dos pruebas realizadas con la aplicación, la primera de ellas es una prueba llevada a cabo en el simulador que pone Android a disposición, y la segunda se realiza con un dispositivo real, concretamente el Huawei P8.

7.1 Pruebas en simulación

La prueba realizada con el simulador consta de los siguientes datos, en orden; profesora, alumno, fecha en la que se hizo, hora en la que da comienzo, duración y tipo (*figura 52*). En este caso se trata de una práctica, por esta razón la polyline es de color azul.

Los resultados obtenidos en el mapa tras finalizar la prueba son los siguientes:

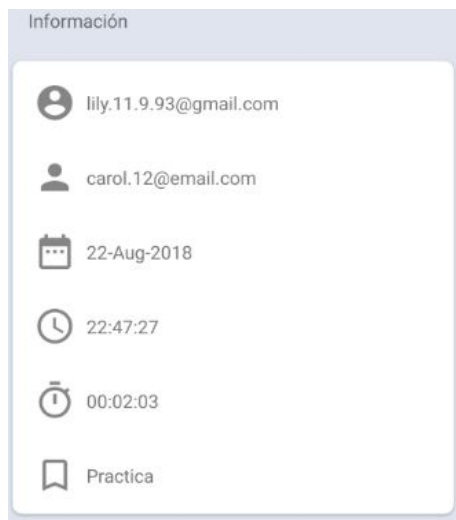


Figura 52: Datos Prueba Simulación

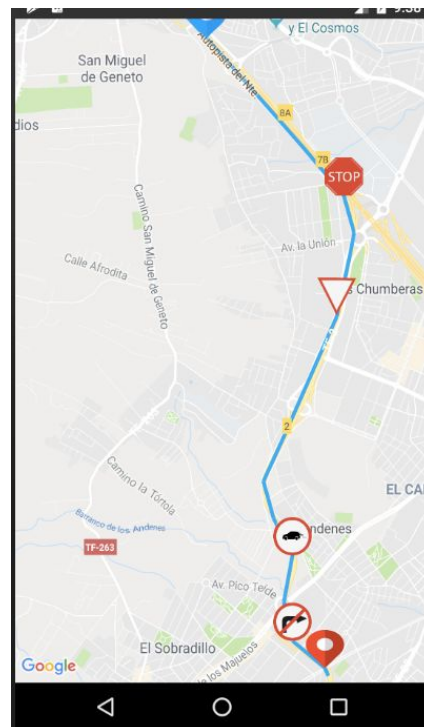


Figura 53: Mapa Prueba Simulación

7.2 Prueba en caso real

La prueba realizada con el dispositivo Huawei consta de los siguientes datos, en orden; profesora, alumno, fecha en la que se hizo, hora en la que da comienzo, duración y tipo. En este caso, se trata de un examen.

Los resultados obtenidos en el mapa tras finalizar la prueba son los siguientes:

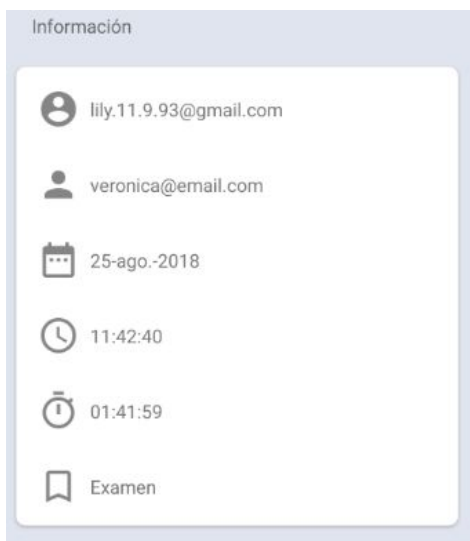


Figura 54: Datos Prueba Real

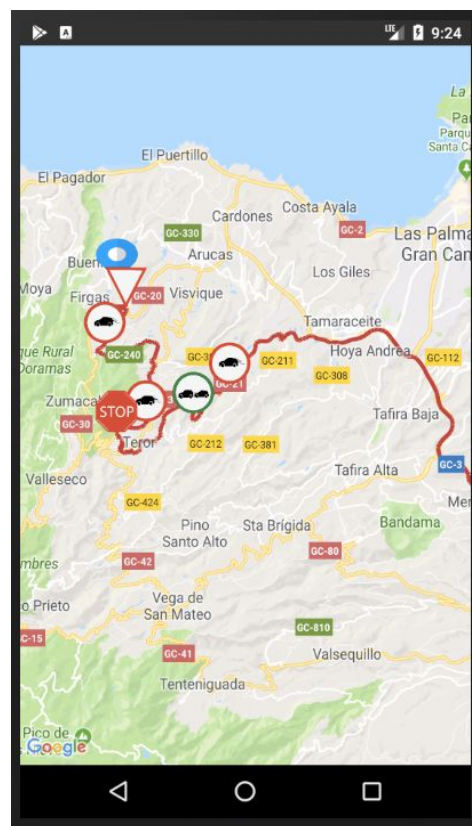


Figura 55: Mapa Prueba Real

Capítulo 8. Conclusiones

En el siguiente capítulo se exponen las conclusiones finales del proyecto, siendo estas la suma de todos los procesos llevados a cabo para la implementación de DriverMoves.

Se han alcanzado todos los requisitos y objetivos definidos al principio de este proyecto, tanto funcionales como no funcionales, obteniendo como resultado una aplicación que ofrece un sistema de aprendizaje m-learning para prácticas de conducir, con el fin de desarrollar el conocimiento y habilidades de forma automática, por medio de un dispositivo del móvil, basándose en los datos generados por el propio usuario. Es capaz de guardar la información generada durante una práctica o examen en tiempo real en la base de datos, contemplando la posibilidad de la pérdida de conexión durante el transcurso de una sesión.

Cabe destacar la importancia de las fases de análisis y diseño ya que sentaron las bases del camino a seguir en el desarrollo, el cual se presentó como un reto, ya que suponía el cumplimiento de todos los objetivos planteados y el uso de tecnologías con las que no se había trabajado anteriormente, a esto se le suma la curva de aprendizaje de cada una de ellas.

DriverMoves se presenta como una app flexible y funcional, con una interfaz de usuario agradable y fácil de utilizar para el usuario. Especialmente desarrollada para cumplir con requerimientos de usabilidad dado que el profesor debe interactuar con la aplicación en tiempo de conducción.

Finalmente, gracias al desarrollo de esta aplicación, se ha aprendido a gestionar todas las fases de un proyecto de este tipo, diseñando y desarrollando todos sus módulos y adquiriendo nuevos conocimientos en el transcurso del mismo. Tecnologías como Android, Java, XML, JSON, Firebase, simulador de android, geolocalización, entre otras.

Conclusions

In this chapter, the final conclusions of the project are exposed, these are the sum of all the processes carried out for the implementation of DriverMoves.

All the requirements and objectives defined at the beginning of this project, functional and non-functional, have been developed, resulting in an application that offers an m-learning system for driving practices, in order to develop the knowledge and skills of automatic form, by using a mobile device, based on the data generated by the user. It is able to save the information generated during a practice or exam in real time in the database, contemplating the possibility of loss of connection during a session.

DriverMoves is presented as a flexible and functional app, with a user-friendly and easy to use interface, having into account the usability requirements that this app has.

It is important to highlight the importance of the analysis and design of the different phases since they laid the foundations for the development path, which was presented as a challenge, since it meant compliance with all the objectives set and the use of technologies with which it had not been worked previously, due to this fact, the learning curve of each new technology learn has been added.

Finally, we have managed to carry out a project of this type, developing all its stages and acquiring new knowledge in technologies such as Android, Java, XML, JSON, Firebase, android simulator, geolocation, among others.

Capítulo 9. Trabajos futuros

Planteando trabajos futuros para DriverMoves surgen varias ideas que podrían potenciar sus funciones y mejorar la accesibilidad de cara a los usuarios.

Por un lado, se podría implementar un chat entre los alumnos y profesores para que se dé entre ellos una comunicación directa, por medio de la app.

Podría implementarse un calendario en el que ambos puedan definir horarios y fechas de las prácticas y exámenes, y un sistema de recordatorio para notificar al usuario cuando se acerque la fecha señalada.

Otro de los posibles trabajos es añadir para el profesor la opción de crear notas o comentarios en los marcadores que se ven en el mapa.

Por otro lado, en cuanto a la interfaz, se pondría a disposición de los usuarios la posibilidad de personalizar la interfaz, lo cual podría resultar más atractivo.

Capítulo 10. Presupuesto

A la hora de plantear un presupuesto para DriverMoves se tienen en cuenta una serie de características presentes en la aplicación, así como el tiempo invertido en el desarrollo y aprendizaje de nuevas tecnologías para conseguir el mejor resultado posible. Dentro de estas características destacan, por un lado, el diseño de una interfaz sencilla e intuitiva, personalizada para cada usuario, ya que según el perfil se verá determinada información u otra, teniendo en cuenta los dos tipos de Roles disponibles en la App. Cada usuario tiene su propio perfil en el que se refleja su información personal.

Por otro lado, DriverMoves es desarrollada para el sistema operativo Android de forma nativa, de esta forma se le saca mayor partido a las características del dispositivo. Posee un sistema de geolocalización con el que se trabaja a lo largo de cada sesión y se generan los datos de interés para el alumno en prácticas, y cuenta con la posibilidad de recuperar los datos en caso de pérdida de conexión, aspecto muy importante ya que se pueden dar condiciones de poca cobertura durante una sesión. También dispone de un sistema de autenticación por medio del Login y la posibilidad de registrarse en caso de ser nuevo en la aplicación.

DriverMoves es una aplicación que propone explorar una nueva forma de aprendizaje dentro del campo del m-learning para las prácticas de conducir, por esta razón y las expuestas anteriormente se propone un presupuesto de 10.000 €.

Bibliografía

- About Español: ¿Qué es una app y cómo descargarla?. Recuperado de <https://www.aboutespanol.com>. Consultado el 20/02/2018.
- Developer Android: Conoce Android Studio. Recuperado de <https://developer.android.com>. Consultado el 20/02/2018.
- Firebase : Productos. Recuperado de <https://firebase.google.com/>. Consultado el 21/02/2018.
- YeePLY Recursos: Lenguajes de programación que debes saber si quieres ser desarrollador Android. Recuperado de <https://www.yeePLY.com/blog/lenguajes-basicos-desarrollador-android/>. Consultado el 09/03/2018.
- Google Maps Platform: Google Maps APIs. Recuperado de <https://enterprise.google.com/intl/es-419/maps/products/mapsapi.html>. Consultado el 21/03/2018.
- Ayuda Google Maps: Buscar o introducir la latitud y la longitud. Recuperado de https://support.google.com/maps/answer/18539?hl=es&topic=1687353&ctx=topic&visit_id=636711538354624655-4123735668&rd=1. Consultado el 12/03/2018.
- Microsiervos: Cómo localizar coordenadas con Google Maps . Recuperado de <https://www.microsiervos.com/archivo/tecnologia/coordenadas-en-google-maps.html>. Consultado el 12/04/2018.
- firebase.google. Firebase Documentación: Primeros pasos con las reglas de bases de datos. Recuperado de <https://firebase.google.com/docs/database/security/quickstart?hl=es-419>. Consultado el 14/04/2018.
- milagrosrp.wordpress. La Educación al Alcance de tu Móvil : Mobile Learning. Recuperado de <https://milagrosrp.wordpress.com/tag/m-learning/>. Consultado el 20/04/2018.
- AristaSur: Qué es el datum de las coordenadas geográficas y su uso en el gps. Recuperado de

<https://www.aristasur.com/contenido/que-es-el-datum-de-las-coordenadas-geograficas-y-su-uso-en-el-gps>. Consultado el 23/04/2018.

- abc: Los 5 grandes cambios para sacarse el carné de conducir en 2018. Recuperado de http://www.abc.es/sociedad/abci-5-grandes-cambios-para-sacarse-carne-c-onducir-2018-201802141411_noticia.html. Consultado el 28/04/2018.
- MultiAtlas. comercios-electronicos: Estadísticas uso móvil 2018 - cómo afectará esto a tu negocio y qué debes saber sobre las app para el futuro de tus ventas. Recuperado de <https://www.comercios-electronicos.com/estadisticas-uso-movil-2018-como-afectara-a-tu-negocio-y-lo-que-debes-saber-sobre-las-app-para-el-futuro-tus-ventas/>. Consultado el 05/05/2018.
- Deustoformacion: Lenguajes para programar aplicaciones en Android. Recuperado de <https://www.deustoformacion.com/blog/desarrollo-apps/lenguajes-para-programar-aplicaciones-android>. Consultado el 10/05/2018.
- Xataka Móvil: Siete aplicaciones de test para aprobar el examen del carnet de conducir. Recuperado de <https://www.xatakamovil.com/aplicaciones/siete-aplicaciones-de-test-para-aprobar-el-examen-del-carnet-de-conducir>. Consultado el 14/05/2018.
- Firebase: Firebase Realtime Database. Recuperado de <https://firebase.google.com/docs/database/?hl=es-419>. Consultado el 26/05/2018.
- Json: Introducing JSON. Recuperado de <http://www.json.org/>. Consultado el 26/05/2018.
- Abril 25, 2018. Developer Android: Ejecutar apps en el emulador de Android. Recuperado <https://developer.android.com/studio/run/emulator?hl=es-419>. Consultado el 26/05/2018.
- luna : Android. Recuperado de <https://www.luna.ovh/planeta/es/Android>
- Blog Historia de la informática. Android : Introducción - ¿Qué es Android?. Recuperado de <https://histinf.blogs.upv.es/2012/12/14/android/>. Consultado el 26/05/2018.

- Xataka Android: Historia y evolución de Android: cómo un sistema operativo para cámaras digitales acabó conquistando los móviles. Recuperado de <https://www.xatakandroid.com/sistema-operativo/historia-y-evolucion-de-android-como-un-sistema-operativo-para-camaras-digitales-acabo-conquistando-los-moviles>. Consultado el 01/06/2018.
- Megapractical : Top 5 Metodologías de Desarrollo de Software. Recuperado de <https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/metodologias-de-desarrollo-de-software>. Consultado el 01/06/2018.
- Calvaza Servicios Informáticos : Qué es gps (sistema de posicionamiento global) <https://www.carvalza.es/que-es-un-gps>. Consultado el 01/06/2018.
- Okhosting: Metodologías del desarrollo Software. Recuperado de <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>. Consultado el 01/06/2018.
- Software de Comunicaciones: Arquitectura Android. Recuperado de <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>. Consultado el 01/06/2018.
- Arpen Technologies: ¿Qué es firebase y qué nos aporta?. Recuperado de <https://arpentechnologies.com/es/blog/aplicaciones-movil/que-es-firebase-y-que-nos-aporta/>. Consultado el 11/06/2018.
- AristaSur: Tipos de formatos para almacenar tracks del GPS. Recuperado de <https://www.aristasur.com/contenido/tipos-de-formatos-para-almacenar-tracks-del-gps>. Consultado el 11/06/2018.
- Topografix: The GPS Exchange Format. Recuperado de <http://www.topografix.com/gpx.asp>. Consultado el 11/06/2018.
- Wikipedia. Anexo: Historial de versiones de Android. Recuperado de https://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android. Consultado el 14/06/2018.
- Certimedica: 5 Apps para aprender a conducir: Teoría y Práctica. Recuperado de

<https://certimedic.es/5-apps-para-aprender-conducir-teoria-y-practica/>.

Consultado el 28/06/2018.