

Proyecto de Fin de Grado



Sergio Rodríguez Geniz.
I.E.S. Velázquez.
CFGS Desarrollo de Aplicaciones Multiplataforma.
8 de diciembre de 2023.

Índice

Lista de Cambios	2
1. Introducción	3
2. Estudio de Viabilidad	4
2.1. Descripción del Sistema Actual	4
2.2. Descripción del Sistema Nuevo	4
2.3. Identificación de Requisitos del Sistema	5
2.3.1. Requisitos de información	5
2.3.2. Requisitos funcionales	8
2.3.3. Otros requisitos	8
2.4. Descripción de la Solución	9
2.5. Planificación del Proyecto	9
2.5.1. Equipo de Trabajo	9
2.5.2. Planificación Temporal	9
2.6. Estudio del Coste del Proyecto	10
3. Análisis del Sistema de Información	10
3.1. Identificación del Entorno Tecnológico	10
3.2. Modelado de Datos	11
3.2.1. Modelo Entidad-Relación	11
3.2.2. Esquema de la Base de Datos	11
3.2.3. Datos de Prueba	11
3.3. Identificación de los Usuarios Participantes y Finales	12
3.4. Identificación de Subsistemas de Análisis	12
3.5. Establecimiento de Requisitos	13
3.6. Diagrama de Análisis	13
3.7. Definición de Interfaces de Usuario	14
3.7.1. Especificación de principios generales de interfaz	14
3.7.2. Especificación de formatos individuales de la interfaz de pantalla	14
3.7.3. Identificación de perfiles de usuario	29
3.7.4. Especificación de formatos de impresión	29
3.7.5. Especificación de la navegabilidad entre pantallas	29
4. Construcción del Sistema	30
5. Conclusión	31
6. Glosario de términos	32
7. Bibliografía	34
Apéndices	34

Listado de Cambios

Tabla 1.

Versión del documento y fechas de cambios.

Fecha	Versión	Descripción breve
29 de septiembre de 2023	1.0.0	Creación del repositorio, creación de la Base de datos.
16 de octubre de 2023	1.1.0	Realizada el LogIn y la conexión con la API.
23 de octubre de 2023	1.2.0	Pantalla de SignUp realizada junto a su API.
30 de octubre de 2023	1.3.0	Realizada la pantalla de cursos gestores para mostrar los cursos a los profesores.
13 de noviembre de 2023	1.4.0	Realizada la pantalla de editor cursos para poder modificar los test de los cursos.
20 de noviembre de 2023	1.5.0	Revisión y finalización de retoques de todas las pantallas realizadas junto a la lógica pertinente de las últimas pantallas.
24 de noviembre	1.5.1	Realización de pruebas con los diferentes usuarios posibles.

1. Introducción

Este proyecto se basa en el desarrollo de una aplicación para la gestión de cursos online, que servirá de ayuda a las personas para evaluar sus conocimientos de informática realizando cursos y tests. Además, podrán gestionar su perfil, descargar los cursos a los que están suscritos y ver su historial de puntuaciones. Los administradores podrán gestionar los cursos y subir o eliminar documentos, además de gestionar los tests. La aplicación a desarrollar tendrá el nombre de “**StudyHub**” y constará de una interfaz sencilla e intuitiva para que el usuario se sienta cómodo en la navegación por la misma.

2. Estudio de Viabilidad

Viabilidad técnica: El alumno que va a desarrollar este proyecto posee los conocimientos y recursos técnicos necesarios para cumplir con los requisitos.

Viabilidad económica: Tanto los costes materiales como los beneficios de este proyecto son nulos. Los únicos costes de este proyecto son los costes de desarrollo.

Viabilidad temporal: Este proyecto necesitará alrededor de dos meses y medio para el correcto desarrollo de cada componente del mismo, además de la resolución de errores.

2.1. Descripción del Sistema Actual

La aplicación utilizada actualmente por el centro educativo Instituto de Educación Secundaria Velázquez para impartir cursos y asignar actividades entregables es “Moodle Centros Sevilla”. Esta aplicación permite a los alumnos realizar tests propuestos por los profesores. Sin embargo, presenta ciertas limitaciones, ya que los alumnos dependen de la publicación de los tests por parte de los profesores y deben realizarlos en el momento en el que estos lo determinen.

Por otro lado, “StudyHub” permite a los alumnos la flexibilidad de realizar los tests de los cursos las veces que lo deseen y en el momento en el que lo consideren oportuno, lo que facilita a los alumnos llevar a cabo una autoevaluación para así medir sus conocimientos.

2.2. Descripción del Sistema Nuevo

La aplicación desarrollada para la gestión de cursos y la evaluación de los conocimientos adquiridos sobre el curso realizado por los alumnos será “StudyHub”, que permite a los alumnos conocer y estudiar la teoría sobre un curso y, seguidamente, realizar un examen tipo test para poder evaluar sus conocimientos, mientras que los profesores (o gestores) pueden modificar, subir nuevos cursos o incluso borrar aquellos que ya no pertenezcan al temario correspondiente.

2.3. Identificación de Requisitos del Sistema

2.3.1. Requisitos de información

La información que nuestro sistema necesitará almacenar será:

- Respecto a los usuarios:
 - DNI
 - Documento nacional de identificación, por el que podremos localizar a los usuarios.
 - Es de tipo String (cadena de caracteres).
 - El volumen de información es de nueve caracteres (ocho dígitos y una letra).
 - El dni no puede ser nulo y debe ser único.
 - Nombre
 - Palabra o conjunto de palabras que se le asignan a una persona en su nacimiento.
 - Es de tipo String.
 - El volumen de información estimado para este campo es de 50 caracteres de longitud.
 - El nombre no puede ser nulo.
 - Apellido1
 - Palabra que se le asigna a una persona en su nacimiento, para identificar a su familia paterna.
 - Es de tipo String.
 - El volumen de información estimado para este campo es de 50 caracteres de longitud.
 - El apellido 1 no puede ser nulo.
 - Apellido2
 - Palabra que se le asigna a una persona en su nacimiento, para identificar a su familia materna.
 - Es de tipo String.
 - El volumen de información estimado para este campo es de 50 caracteres de longitud.
 - El apellido 2 puede ser nulo.
 - Teléfono
 - Secuencia de dígitos utilizada para identificar una telefonía celular dentro de una red telefónica.
 - Es de tipo String.
 - El volumen de información estimado para este campo es de 9 caracteres de longitud.
 - El teléfono no puede ser nulo y debe ser único para cada persona.
 - Email
 - Secuencia de caracteres utilizada para transmitir información de forma digital.

- Es de tipo String.
- El volumen de información estimado para este campo es de 50 caracteres de longitud.
- El email no puede ser nulo y debe ser único para cada persona.
- Usuario
 - Secuencia de caracteres generada por el usuario.
 - Es de tipo String.
 - El volumen de información es de 15 caracteres.
 - El usuario no puede ser nulo.
- Contraseña
 - Da seguridad a la base de datos, puesto que sólo los usuarios correctamente registrados podrán acceder a la app y a los datos administrados por la base de datos.
 - Es de tipo MD5.
 - El volumen de información es de 32 caracteres, el número de caracteres usado por el algoritmo de hashing de MD5.
 - La contraseña no puede ser nula.
- Perfil
 - Permite saber si un usuario es alumno o gestor.
 - Es de tipo String.
 - El volumen de información es de 1 carácter.
 - El perfil no puede ser nulo ya que si no no sabríamos si es alumno o gestor.
- Respecto a los cursos:
 - Id
 - Código único por el que podremos localizar el curso.
 - Es de tipo Integer.
 - El volumen de información estimado para este campo es de 2 caracteres de longitud.
 - El identificador es único y no puede ser nulo, para prevenir posibles errores será el propio sistema el encargado de asignar un identificador.
 - Nombre
 - Secuencia de caracteres que muestra una breve descripción sobre el curso.
 - Es de tipo String.
 - El volumen de información estimado para este campo es de 50 caracteres de longitud.
 - El nombre no puede ser nulo.
 - Ruta_pdf
 - Secuencia de caracteres que indica la ruta en la que se encuentra el curso.
 - Es de tipo String.

- El volumen de información estimado para este campo es de 255 caracteres de longitud.
- La ruta no puede ser nulo.
- Respecto a las preguntas:
 - Id
 - Código único por el que podremos localizar las preguntas.
 - Es de tipo Integer.
 - El volumen de información estimado para este campo es de 3 caracteres de longitud.
 - El identificador es único y no puede ser nulo, para prevenir posibles errores será el propio sistema el encargado de asignar un identificador.
 - Enunciado
 - Secuencia de caracteres que indica el enunciado de la pregunta a responder.
 - Es de tipo String.
 - El volumen de información estimado para este campo es de 255 caracteres de longitud.
 - El enunciado no puede ser nulo.
 - Id_curso
 - Código que indica el curso al que pertenece la pregunta.
 - Es de tipo Integer.
 - El volumen de información estimado para este campo es de 2 caracteres de longitud.
 - El identificador de curso no puede ser nulo.
- Respecto a la respuesta Test:
 - Id
 - Código único de cada pregunta.
 - Es de tipo Integer.
 - El volumen de información estimado para este campo es de 5 caracteres de longitud.
 - El identificador es único y no puede ser nulo.
 - Id_pregunta
 - Identificador de la pregunta a la que pertenece.
 - Es de tipo Integer.
 - El volumen de información estimado para este campo es de 3 caracteres de longitud.
 - El identificador de la pregunta no puede ser nulo.
 - Id_curso
 - Código del curso al que pertenece la opción.
 - Es de tipo Integer.
 - El volumen de información estimado para este campo es de 2 caracteres de longitud.
 - El identificador del curso no puede ser nulo.

- Opción
 - Secuencia de caracteres que indica la opción a seleccionar como correcta.
 - Es de tipo String.
 - El volumen de información estimado para este campo es de 255 caracteres de longitud.
 - La opción no puede ser nula.
- esCorrecta
 - Dígito que indica si la opción indicada es correcta o no.
 - Es de tipo Integer.
 - El volumen de información estimado para este campo es de 1 carácter de longitud.
 - El campo “esCorrecta” no puede ser nulo y el valor debe ser 1 (si la opción es correcta) o 0 (si la opción no es correcta).

2.3.2. Requisitos funcionales

Los requisitos funcionales son:

1. Realizar LogIn si se ha creado una cuenta anteriormente.
2. Si el perfil del usuario es “alumno”:
 - a. Elegir un curso a realizar.
 - b. Descargar un documento PDF con toda la teoría del mismo.
 - c. Realizar una prueba de evaluación sobre el conocimiento obtenido sobre el mismo.
 - d. Ver las puntuaciones obtenidas en cada realización de las pruebas de evaluación junto a la fecha del mismo.
3. Si el perfil del usuario es “gestor”:
 - a. Escoger un curso para modificar su prueba de evaluación.
 - b. Eliminar un curso no actualizado.
 - c. Añadir un nuevo curso subiendo el PDF con su teoría.
 - d. Gestionar el perfil de los alumnos.
4. Ver el perfil del usuario y modificarlo si fuera necesario.
5. Crear cuenta para poder realizar cursos o gestionarlos.

2.3.3. Otros requisitos

El dispositivo Android ® en el que se instale la aplicación deberá tener una versión Interfaz de Programación de Aplicaciones (API) posterior a 29 (Quince Tart) e inferior a la versión API 32 (Snow Cone), si la versión del dispositivo no está en el rango de versiones incluidas, no será posible ejecutar la aplicación.

2.4. Descripción de la Solución

El problema de la utilización de la página web “Moodle Centros” es la necesidad de tener un usuario creado por la Junta de Andalucía el cual al terminar el periodo educativo del alumno no podrá volver a acceder a los cursos realizados ni ampliar su conocimiento.

Para ello, el centro educativo ha decidido crear una aplicación móvil que permita formar tanto a aquellos alumnos matriculados como a otras personas que deseen obtener conocimiento sobre algún tema y autoevaluarse para saber el conocimiento obtenido sin la necesidad de haber estado matriculado en el centro años anteriores.

Los cambios sobre la aplicación moodle centros sevilla son tales como:

- Creación de una nueva cuenta personal.
- Posibilidad de la realización de cursos oficiales subidos por los docentes.
- Posibilidad de ver la puntuación obtenida en los diferentes cursos activos realizados.
- Posibilidad de la realización de cursos en el momento que el alumno crea oportuno y las veces que lo deseé.

2.5. Planificación del Proyecto

2.5.1. Equipo de Trabajo

El equipo está formado por Sergio Rodríguez Geniz, alumno del segundo curso del Módulo de Formación Profesional “Desarrollo de Aplicaciones Multiplataforma” del Centro Educativo Instituto de Educación Secundaria Velázquez.

2.5.2. Planificación Temporal

La planificación de del tiempo de desarrollo de la aplicación “StudyHub” es la siguiente:

- 25 al 29 de Septiembre: creación del repositorio de github, desarrollo base de datos y realización de la api para la obtención del perfil del usuario.
- 2 al 6 de Octubre: Realización de la pantalla de LogIn y conexión con la API.
- 9 al 13 de Octubre: Realización de las pantallas de ayuda, información y SignUp.
- 16 al 20 de Octubre: Conexión de la pantalla SignUp con la API y realización de comprobaciones de datos insertados.
- 23 al 27 de Octubre: Realización de la API para la obtención de 1 o todos los cursos y finalización de la pantalla de los cursos de los alumnos.
- 30 de Octubre al 3 de Noviembre: Realización de la pantalla de los cursos de los gestores y conexión con la API.
- 6 al 10 de Noviembre: Realización de la pantalla de subida de cursos junto al desarrollo de la API para la inserción de cursos.
- 13 al 17 de Noviembre: Desarrollo de la pantalla de perfil junto a su correspondiente API para la modificación de datos.
- 20 al 24 de Noviembre: Realización de la pantalla para realización y modificación de test.

- 27 de Noviembre al 1 de Diciembre: Pulido de todas las funcionalidades.
- 4 al 8 de Diciembre: Realización de las pruebas pertinentes.

2.6. Estudio del Coste del Proyecto

Suponiendo que trabajamos como desarrolladores software junior y que el salario bruto medio anual en Accenture es de 19.000€.

Obtendremos los siguientes datos:

Tabla 2.

Datos del coste del proyecto.

Retención de IRPF/mes	13.73%
Paga extra neta	1.365,94€
Total del IRPF/año	2.608,7€
Total de cuotas a la S.S.	1.339,5€
Sueldo neto anual	15.051,8€
Sueldo neto mensual	1.254,34€

$$\text{Sueldo diario} = \frac{\text{Sueldo neto anual}}{\text{Número total de días laborales}} = \frac{15051,8}{252} = 59,73\text{€}$$

Tras estos datos, se obtendrían 59,73€ por jornada de ocho horas, teniendo en cuenta que el proyecto comenzará el 25 de septiembre y la fecha de fin es el 8 de diciembre, por lo que se tendría que trabajar unas 55 jornadas, un total de 3.285,15€.

3. Análisis del Sistema de Información

3.1. Identificación del Entorno Tecnológico

El entorno de explotación del sistema utilizado será:

- PC:
 - SO: WIndows 11
 - Procesador: Intel Core i5 1155G7
 - RAM: 16 Gb
 - Almacenamiento: Disco principal SSD NVME de 512 Gb y un disco secundario SSD de 1 TB.

El entorno de pruebas utilizado será:

- Dispositivo Android:
 - SO: MIUI Versión 14.0.3
 - CPU: Octa-core, Max 2.32 GHz

- RAM: 6 Gb(por defecto) + 2 Gb (Ampliados)
- Almacenamiento: 128 Gb
- Android Versión: 12 SKQ1.211019.001

El entorno de desarrollo que se utilizará para la realización del proyecto será:

- Visual Studio Code para el desarrollo de API REST.
- Postman para la comprobación de consultas a la base de datos.
- Base de datos MySQL alojada en el servidor Xampp.
- Android Studio con lenguaje Java para el desarrollo de la aplicación móvil.

3.2. Modelado de Datos

3.2.1. Modelo Entidad-Relación

El modelo Entidad Relación está realizado con la aplicación Dia.

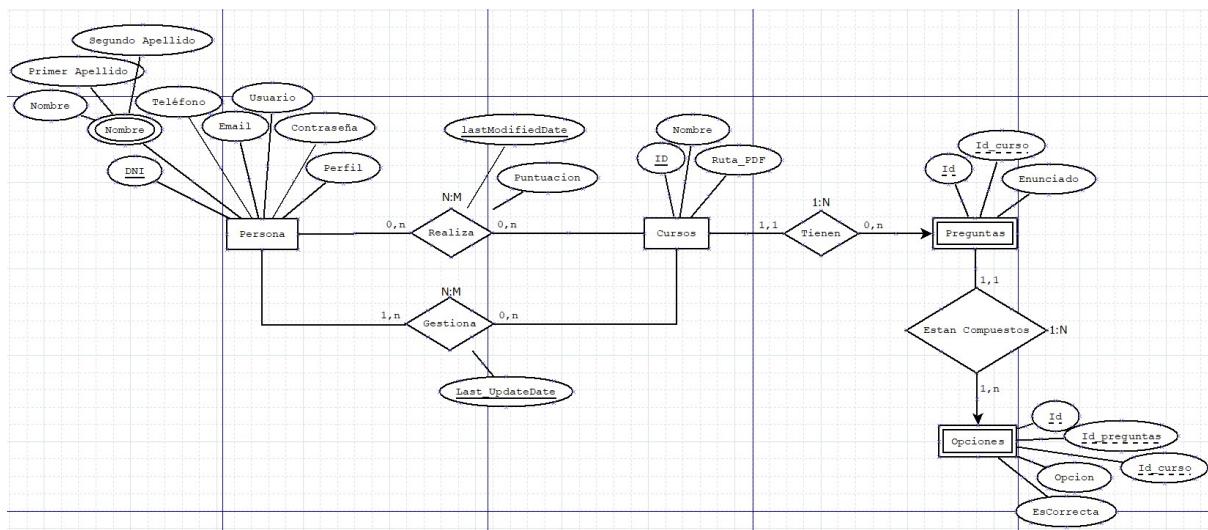


Figura 1. Modelo Entidad-Relación.

3.2.2. Esquema de la Base de Datos

El esquema de la base de datos irá incluido en la entrega.

Archivo: DDL.

3.2.3. Datos de Prueba

El archivo con los datos de prueba irá incluido en la entrega.

Archivo: DML.

3.3. Identificación de los Usuarios Participantes y Finales

La aplicación la podrán utilizar tanto los profesores (gestores) como los alumnos. Los profesores serán los encargados de la subida de cursos, creación de test y la modificación de estos así como la gestión de los perfiles de los alumnos registrados. Mientras que los alumnos, podrán descargarse los cursos que deseen y realizar un cuestionario una vez vista la teoría para poder comprobar el conocimiento adquirido sobre dicho tema.

3.4. Identificación de Subsistemas de Análisis

- **Subsistema de Autenticación:** Este subsistema maneja el registro de usuarios, el inicio de sesión y la redirección a la pantalla correcta dependiendo del perfil del usuario (gestor o alumno).
- **Subsistema de Gestión de Cursos (para Profesores):** Este subsistema permite a los profesores gestionar las cuentas de usuario de los alumnos y los cursos, es decir, subir nuevos cursos, modificar los anteriores y eliminar otros obsoletos.
- **Subsistema de Realización de Cursos (para Alumnos):** Este subsistema permite a los alumnos realizar los cursos. Los alumnos podrán realizar los cursos que deseen y comprobar el conocimiento adquirido durante la realización del mismo.
- **Subsistema de Gestión de Datos:** Este subsistema se encarga de la gestión de datos utilizando una base de datos MySQL y una API desarrollada en PHP para la conexión con la base de datos.

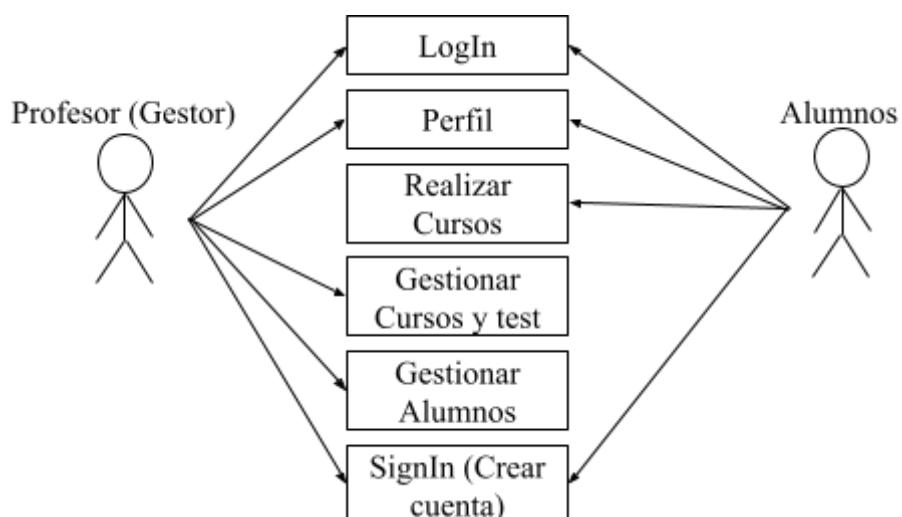


Figura 2. Diagrama de subsistemas.

3.5. Establecimiento de Requisitos

- **Subsistema de Autenticación:** Este subsistema maneja el registro de usuarios, el inicio de sesión, la creación de nuevas cuentas y la redirección a la pantalla correcta dependiendo el perfil del usuario.
- **Subsistema de Gestión de Cursos:** El usuario que es redirigido a este subsistema son los que tienen perfil gestor (profesores) los cuales pueden gestionar los cursos, subiendo nuevos cursos, borrando o modificando los antiguos y gestionar las cuentas de usuario de los alumnos, modificando su perfil o eliminando su cuenta.
- **Subsistema de Realización de Cursos:** Los usuarios redirigidos a este subsistema son los alumnos, los cuales pueden descargar cursos y pueden realizar cuestionarios para evaluar sus conocimientos adquiridos.
- **Subsistema de Gestión de Perfiles:** Este subsistema es accesible desde ambos perfiles, el cual te permite modificar tus datos personales.

3.6. Diagrama de Análisis

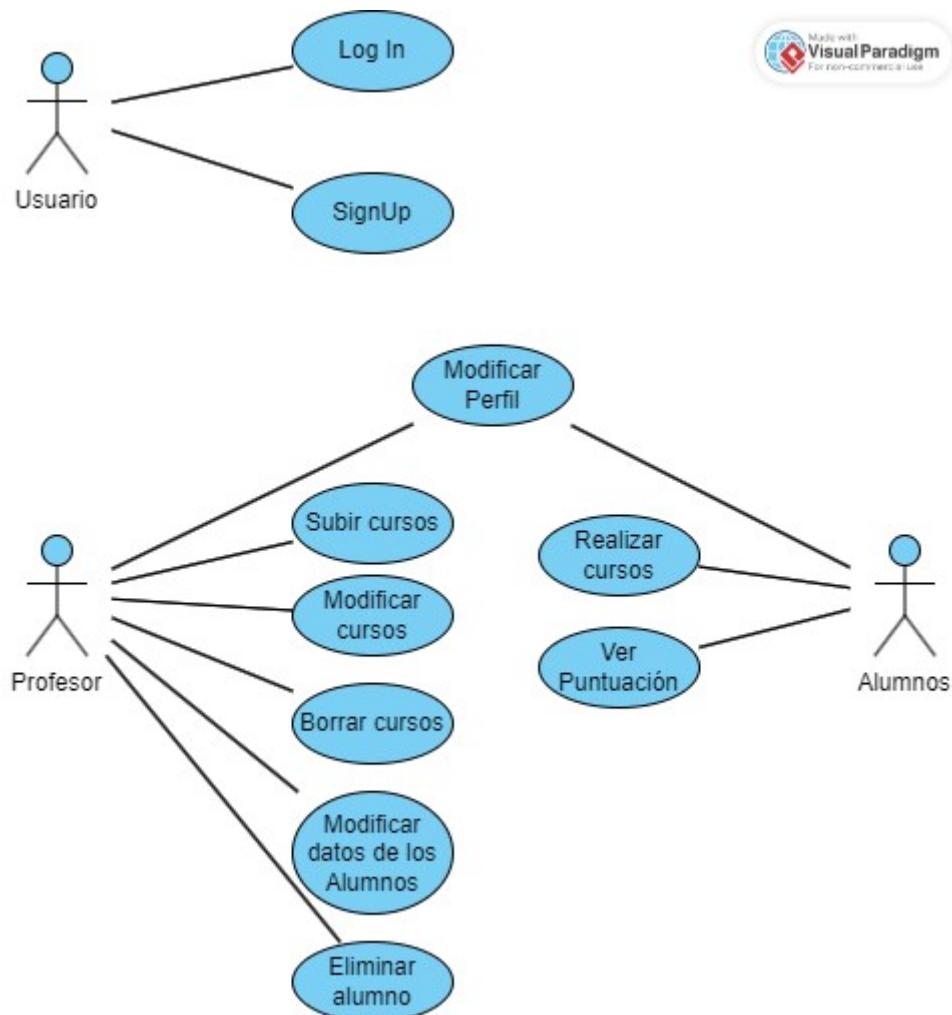


Figura 3. Diagrama de casos de usos.

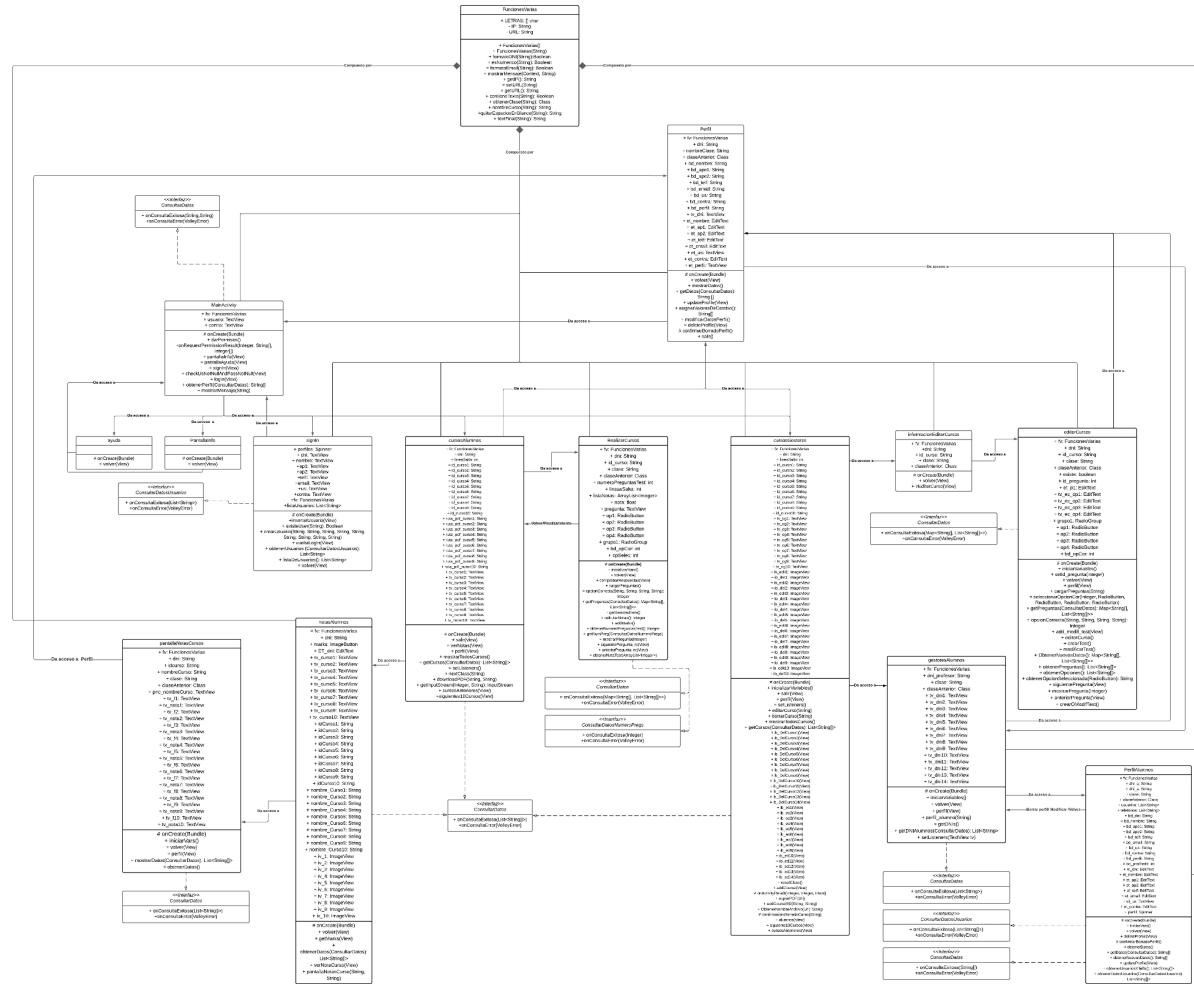


Figura 4. Diagrama de clases.

Para una mejor visualización del diagrama de clases pueden entrar en el siguiente [enlace](#) o abrir el archivo adjunto en la carpeta “otrosDocumentos” llamado “DiagramaDeClases”.

3.7. Definición de Interfaces de Usuario

3.7.1. Especificación de principios generales de interfaz

Cada interfaz tendrá el mismo formato, en la parte superior encontraremos la información a mostrar de los cursos, notas o los datos de cada usuario, mientras que en la parte inferior podremos encontrar una botonera de acción, todos nivelados a la misma altura.

3.7.2. Especificación de formatos individuales de la interfaz de pantalla

La aplicación consta de un total de catorce pantallas, de las cuales cinco de ellas se puede acceder desde ambos perfiles, cinco de ellas serán únicamente para los gestores/profesores y cuatro serán de acceso exclusivo para los usuarios con perfil de alumno.

Las diferentes interfaces para ambos usuarios son las siguientes:

- **activity_main:** Esta es la pantalla que se muestra al iniciar la aplicación, que consta de un TextView indicando el nombre de la aplicación a modo de bienvenida y dos EditText que permiten al usuario introducir su usuario y contraseña y poder acceder a la aplicación. Debajo de estos podemos observar otros dos botones con el texto de “LOGIN” el cual permite acceder a la aplicación y otro con el texto de “SIGN IN” que permite crear una cuenta para poder realizar los diferentes cursos.

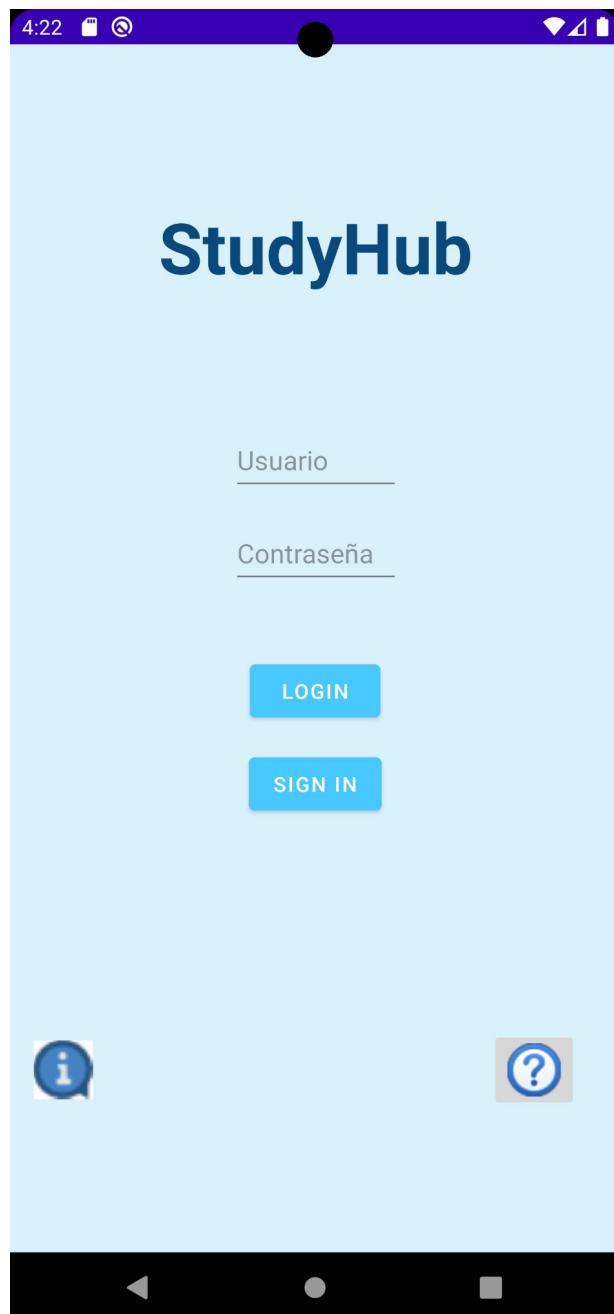


Imagen 1. Pantalla activity_main.

- `activity_sign_in`: Esta interfaz permite a aquellos nuevos usuarios crear una nueva cuenta para poder realizar cursos o gestionarlos. Esta pantalla está compuesta por ocho `EditText` en los cuales el usuario deberá introducir sus datos personales para la creación de la cuenta, un `Spinner` (desplegable) en el cual el usuario deberá elegir su rol y varios botones de acción que permiten al usuario navegar hacia la pantalla anterior o crear su cuenta.

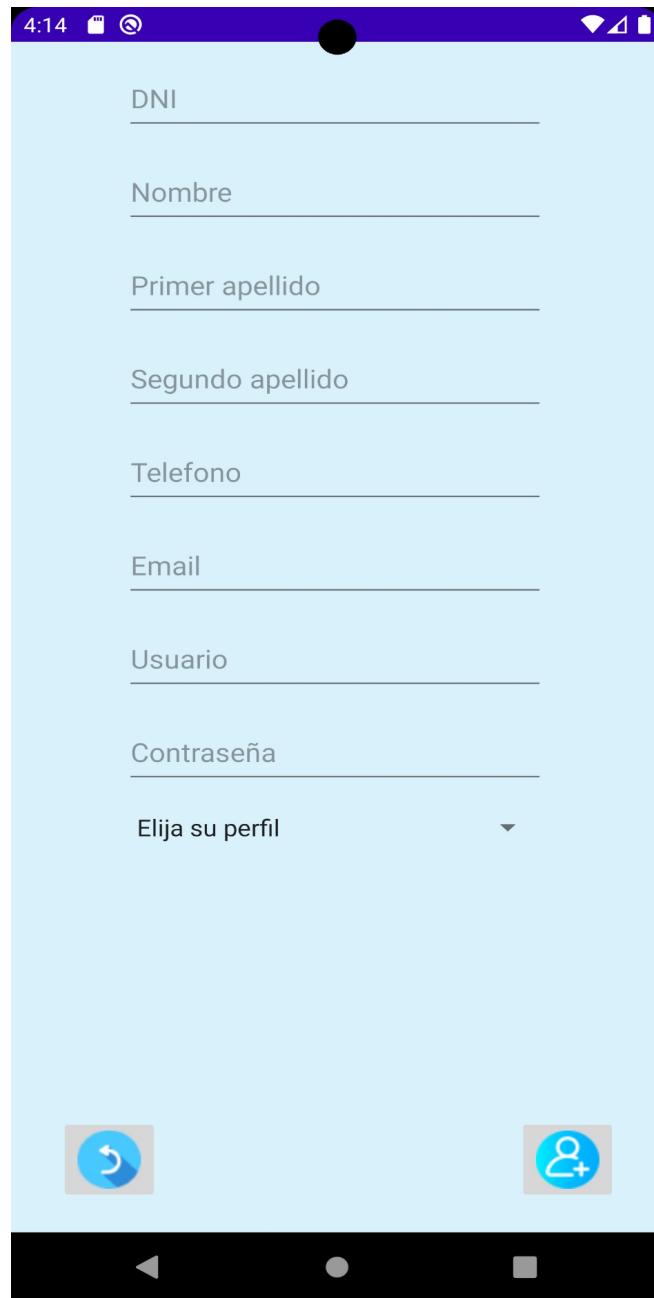


Imagen 2. Pantalla `activity_sign_in`.

- `activity_ayuda`: En esta pantalla podemos observar un cuadro de texto (`TextView`) el cual da información sobre cómo iniciar sesión o crear una cuenta. En la parte inferior de la pantalla observamos un botón (`ImageButton`) que permite volver a la pantalla anterior.

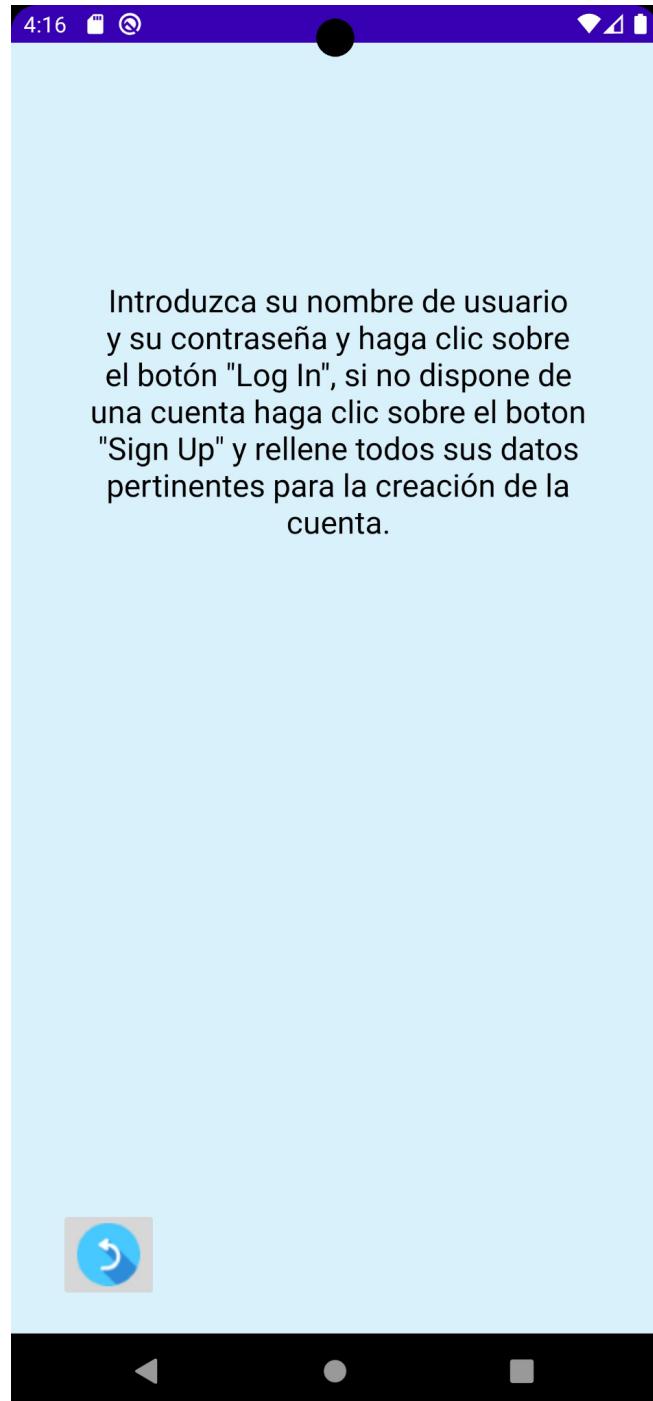


Imagen 3. Pantalla `activity_ayuda`.

- **activity_pantalla_info:** En esta interfaz podemos observar dos TextView, en el primero observamos el nombre de la aplicación “StudyHub” mientras que en el otro observamos el nombre del creador de la aplicación. En la parte inferior de la pantalla observamos un botón (ImageButton) que permite volver a la pantalla anterior.

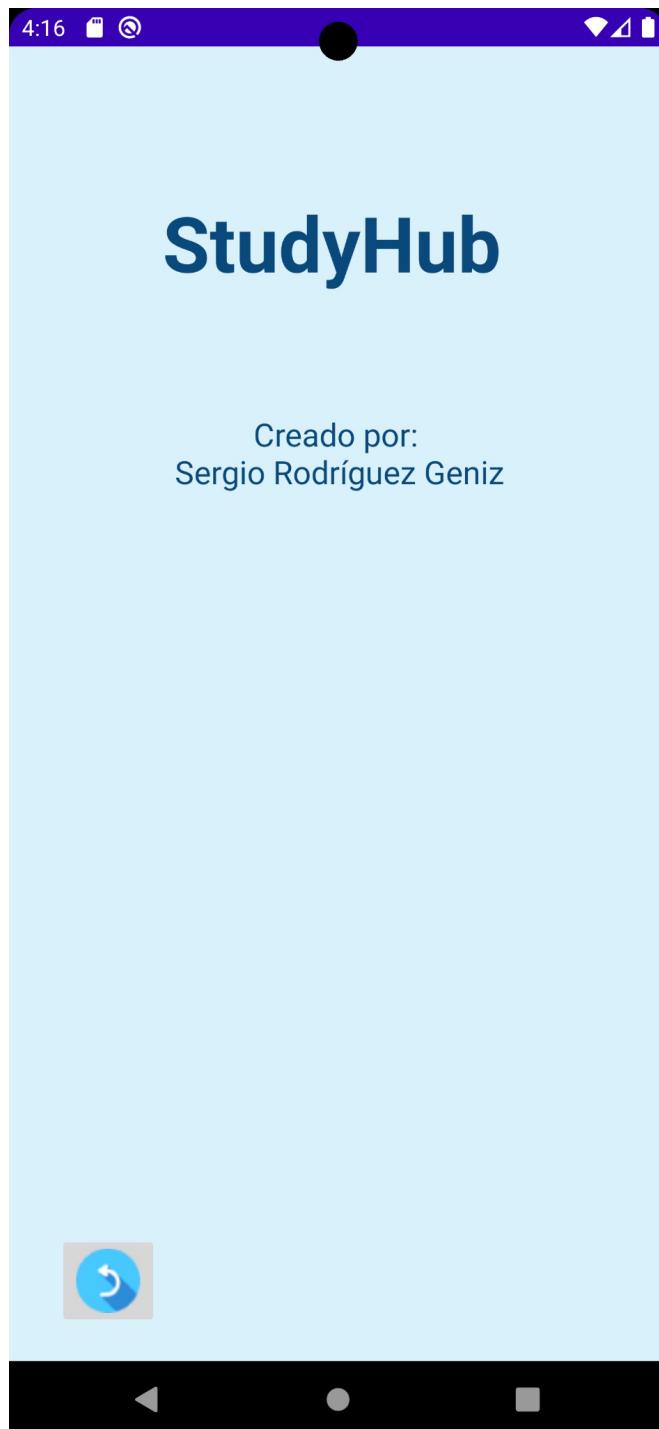


Imagen 4. Pantalla activity_pantalla_info.

- **activity_perfil:** En esta pantalla podemos observar un TextView que muestra el DNI del usuario, seguidamente podremos observar dos columnas, la de la izquierda indica el nombre del campo, es decir, “Nombre”, “Primer apellido”, “Segundo apellido”, etc... mientras que en la columna de la derecha podemos observar el valor del campo, de los cuales es posible cambiar el valor para poder actualizar sus datos. En la parte inferior de la pantalla podemos observar una botonera, comenzando por la izquierda, tenemos el botón de vuelta a la pantalla anterior, en el centro podemos observar el botón para eliminar cuentas, el cual elimina todos sus datos de la base de datos y, por último, el botón de la derecha que modifica los datos del usuario.



Imagen 5. Pantalla activity_perfil.

Pantallas para los gestores:

- **activity_cursos_gestores:** Esta pantalla podemos observar los diferentes cursos que hay en la base de datos, los gestores podrán editarlos haciendo click en el nombre del curso o pinchando en el ícono del lápiz que aparece a la derecha del mismo o eliminarlos haciendo click en la imagen de la papelera roja que sale en la parte derecha junto al lápiz de edición. Seguidamente, podemos observar otros dos botones que permiten mostrar más cursos al pinchar en el botón “MOSTRAR MÁS CURSOS” o ver los diez cursos anteriores al pinchar en “VER CURSOS ANTERIORES”. En la parte inferior de la pantalla, podemos observar cuatro ImageButtons que permiten volver a la clase anterior (Imagen de la izquierda), podremos añadir nuevos cursos a la aplicación seleccionando archivos de tipo PDF que tengamos en nuestro dispositivo (ícono del centro izquierda), podremos dirigirnos a la pantalla para ver los alumnos que hay registrados (ícono del centro derecha) y, por último, nos podemos dirigir al perfil del usuario con el que hayamos iniciado sesión.

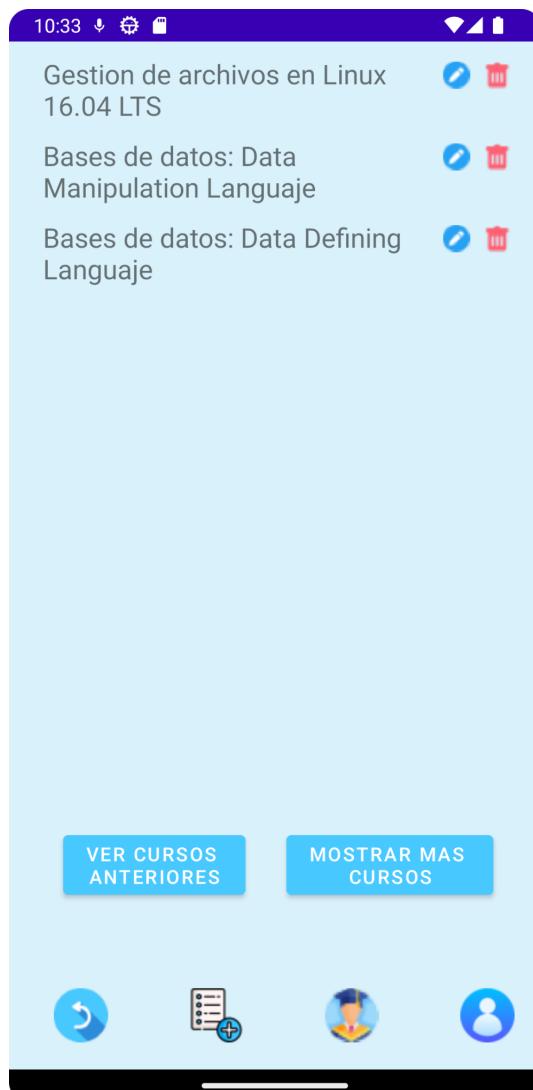


Imagen 6. Pantalla activity_cursos_gestores.

- **activity_informacion_editar_cursos:** En esta interfaz podemos observar un TextView en el centro, el cual informa al docente de la correcta realización o modificación de los tests de evaluación de los cursos. En la parte inferior, podemos observar un único botón que nos permite dirigirnos a la pantalla de realización de los tests (activity_editor_cursos).

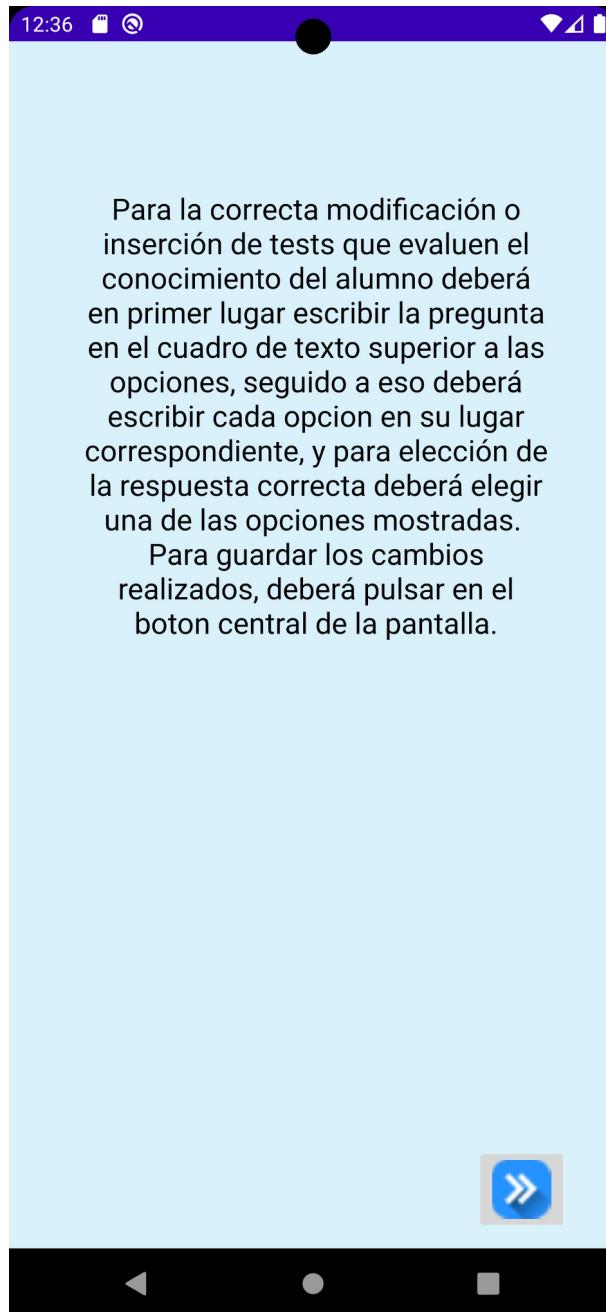


Imagen 7. Pantalla activity_informacion_editar_cursos.

- **activity_editor_cursos:** Esta pantalla permite a los usuarios con perfil de “gestor” realizar test para nuevos cursos añadidos o modificar aquellos test de cursos ya existentes. Para cada pregunta podemos observar cinco EditText, uno de ellos para escribir el enunciado de la pregunta, y los otros cuatro para cada opción de respuesta, podemos observar también cuatro imágenes indicando cual es la pregunta uno, dos, tres y cuatro, a parte podemos observar un grupo de RadioButtons con el texto de “Opción 1”, “Opción 2”, “Opción 3” y “Opción 4”, de los cuales habrá que seleccionar uno de ellos para indicar cuál de las opciones introducidas anteriormente es la correcta. Seguidamente, podemos observar dos botones, que permiten pasar a la siguiente pregunta para editar o volver a la anterior. En la parte inferior de la pantalla observamos tres ImageButtons, el de la izquierda nos dirigirá a la pantalla anterior (activity_cursos_gestores), el del medio guardará el test en la base de datos y redirigirá al usuario a la pantalla en la que se muestran todos los cursos existentes y por último, el botón de la derecha, permitirá al usuario ir a su perfil.

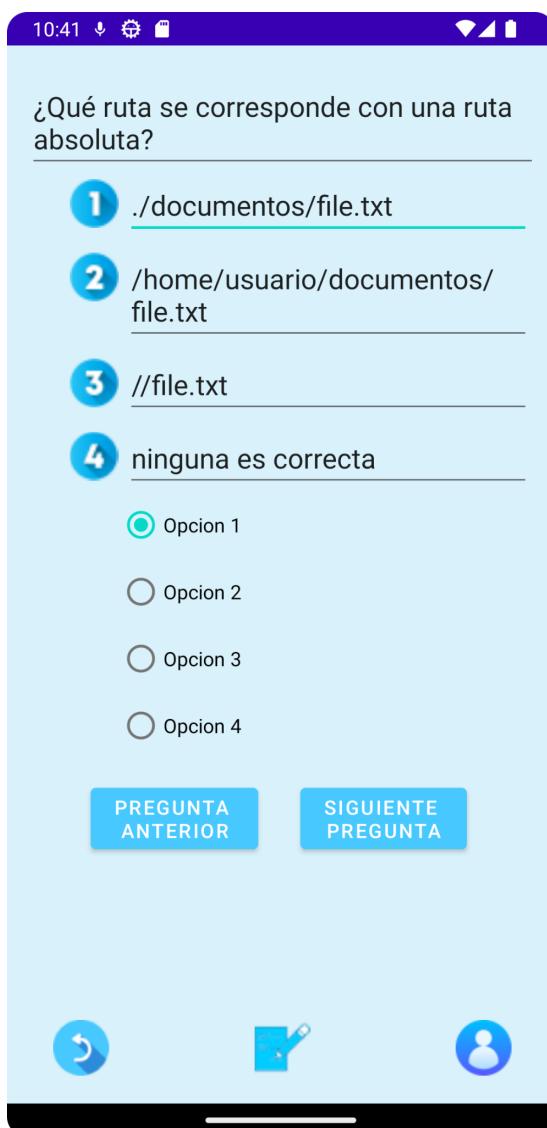


Imagen 8. Pantalla activity_editor_cursos.

- `activity_gestores_alumnos`: Esta pantalla permite a los profesores ver los alumnos registrados en la aplicación y elegir al que deseé en el caso de que necesite modificar su perfil. En la parte inferior de la pantalla podemos observar una botonera que permite volver a la pantalla anterior (Pantalla `activity_cursos_gestores`) al pulsar el botón de la izquierda o dirigirnos a la pantalla de edición de perfil (pantalla `activity_perfil`) al pulsar el botón de la derecha.

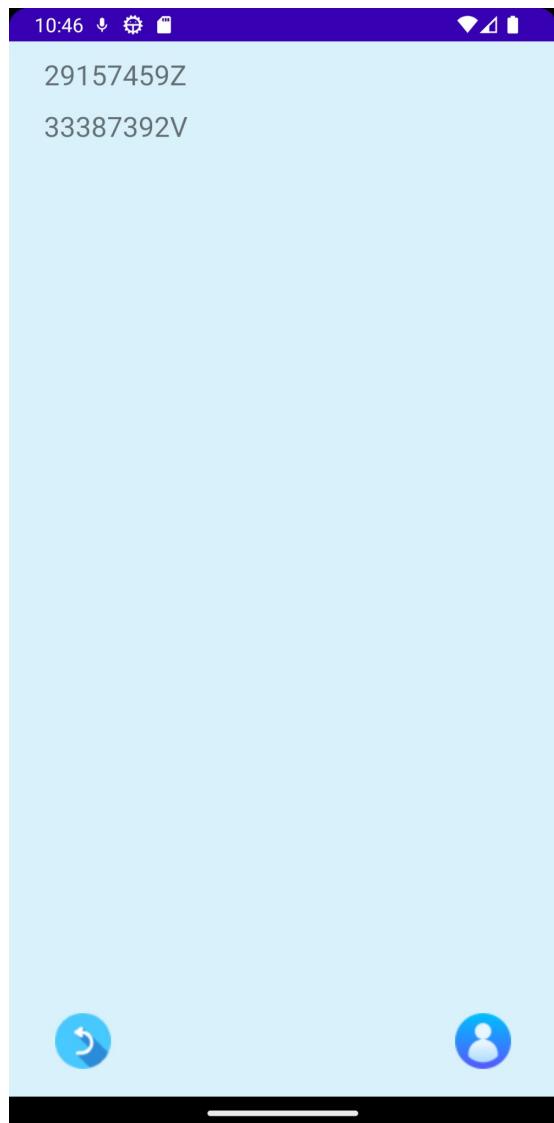


Imagen 9. Pantalla `activity_gestores_alumnos`.

- `activity_perfil_alumnos`: En esta pantalla se pueden observar dos columnas, la primera está compuesta por nueve `TextViews` los cuales indican el campo que vamos a editar, y la segunda columna está compuesta por ocho `EditText` y un `Spinner` que muestran los datos actuales del alumno y permiten editar sus datos para actualizar el perfil del alumno. En la parte inferior de la pantalla podemos observar tres `ImageButtons` que permiten regresar a la pantalla anterior (pantalla `activity_gestores_alumnos`), eliminar el perfil del alumno (botón central) o modificar los datos del alumno.



Imagen 10. Pantalla `activity_perfil_alumnos`.

Las pantallas para aquellos usuarios con perfil de alumno serán las siguientes:

- `activity_cursos_alumnos`: En esta pantalla podemos observar los diferentes cursos existentes en la base de datos. Está compuesta por varios `TextViews` los cuales mostrarán el nombre de los diferentes cursos que están disponibles para su realización. Para ello, deberán pinchar en el nombre de un curso y automáticamente se les descargará un documento en formato PDF con la teoría sobre el curso elegido en la carpeta “cursos” la cual se creará automáticamente en la carpeta “Downloads” del dispositivo, y le redirigirá a la pantalla de realizar cursos (`activity_realizar_cursos`). Seguidamente, podremos observar dos botones los cuales permiten mostrar más o menos cursos dependiendo del botón pulsado. En la parte inferior de la pantalla podemos observar tres `ImageButtons` los cuales permitirán ir a la pantalla anterior (botón de la izquierda), ir a la pantalla de notas del alumno (botón central) e ir a la pantalla de perfil para ver los datos personales del alumno.



Imagen 11. Pantalla `activity_cursos_alumnos`.

- **activity_notas_alumnos:** Esta pantalla permite al usuario ver los diferentes cursos realizados. Esta interfaz está compuesta por un total de trece TextViews, el de la parte superior muestra el DNI del usuario, justo debajo observamos dos columnas, las dos primera filas indican el nombre del campo mostrado, es decir, “Nombre del curso” y “Ver las notas del curso” mientras que los siguientes indican el propio nombre del curso y un botón el cual redirige a la siguiente pantalla (pantalla activity_pantalla_notas_cursos). En la parte inferior, podemos observar dos ImageButtons, que permiten volver a la pantalla anterior (ImageButton de la izquierda) y refrescar la pantalla de notas (ImageButton de la derecha).

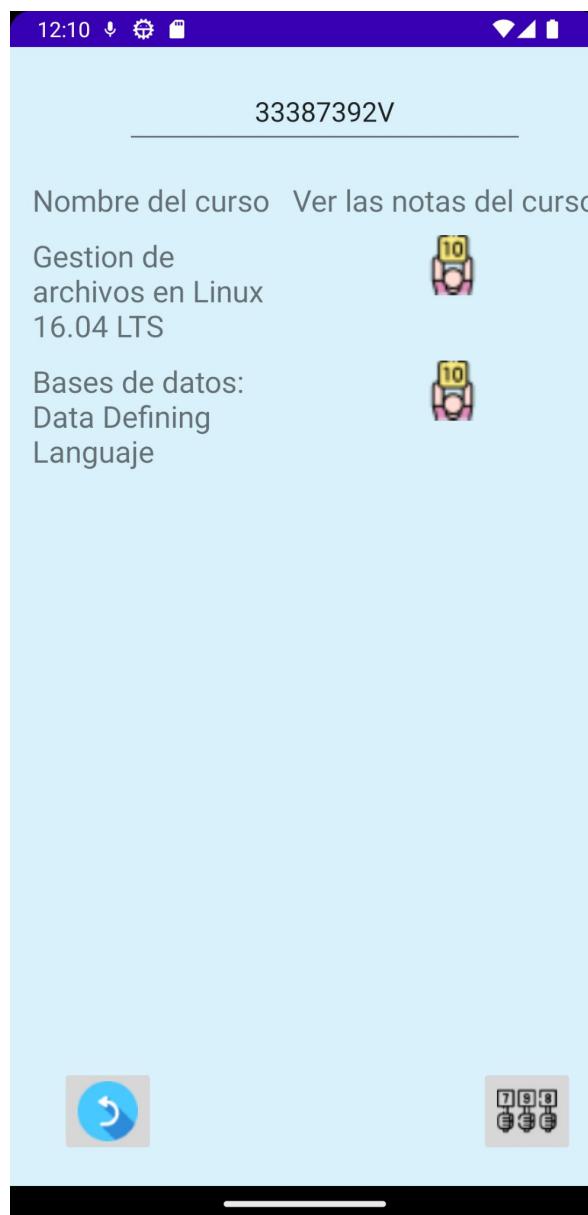


Imagen 12. Pantalla activity_notas_alumnos.

- **activity_pantalla_notas_cursos:** En esta pantalla el alumno podrá ver la puntuación obtenida en el curso seleccionado junto a la fecha de realización del mismo. Está compuesta por un TextView principal que mostrará el nombre del curso, seguidamente observamos dos columnas, la primera muestra la fecha de realización del curso mientras que la segunda muestra la puntuación obtenida. En la parte inferior de la pantalla podemos observar dos botones que permiten ir a la pantalla anterior (pantalla activity_notas_alumnos) o dirigir hacia la pantalla de perfil del alumno (pantalla activity_perfil).



Imagen 13. Pantalla activity_pantalla_notas_cursos.

- **activity_realizar_cursos:** Esta interfaz permite a los alumnos realizar los test de los cursos y ver la nota obtenida mediante un mensaje que se mostrará en la pantalla una vez realizado el test. Cuando el alumno haya completado la prueba se redirigido a la pantalla de todos los cursos (activity_cursos_alumnos). Esta pantalla está compuesta por un TextView, el cual muestra el enunciado de la pregunta, y un grupo de cuatro RadioButtons los cuales muestran las opciones a elegir por pregunta. Seguidamente de las opciones, podemos observar dos botones los cuales permiten ir a la siguiente pregunta o a la anterior. En la parte inferior podemos observar dos ImageButtons los cuales permiten desplazar a la pantalla anterior (ImageButton de la izquierda) y finalizar el intento de test guardando la puntuación en la base de datos (ImageButton de la derecha).

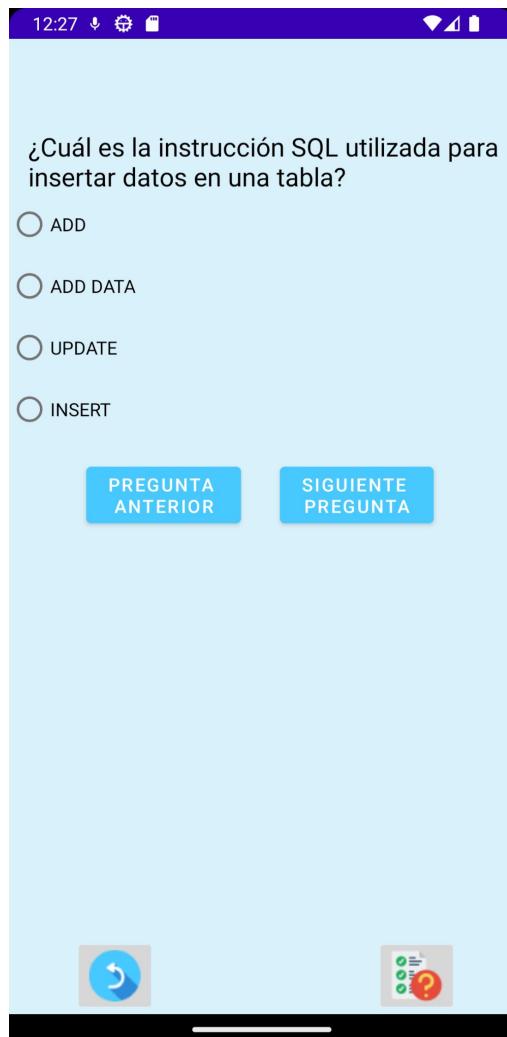


Imagen 14. Pantalla activity_realizar_cursos.

3.7.3. Identificación de perfiles de usuario

Los perfiles existentes en la aplicación son:

- “Gestor” o “profesor” que son los encargados de la subida de cursos, creación y modificación de test para la evaluación de los cursos y la eliminación de estos, así como la propia gestión del perfil de los “alumnos”.
- “Alumnos” que pueden descargar los cursos que deseen y autoevaluar el conocimiento en dicho tema además de ver las notas obtenidas en cada curso realizado.

3.7.4. Especificación de formatos de impresión

La aplicación desarrollada permitirá a los alumnos la descarga de archivos en formato PDF, el cual es uno de los más utilizados debido a su compatibilidad universal. Sin embargo, la aplicación no permitirá a los usuarios imprimir el citado curso mientras que el propio documento PDF con el contenido teórico del curso si podrá ser impreso.

3.7.5. Especificación de la navegabilidad entre pantallas

Para el perfil docente sería la siguiente:

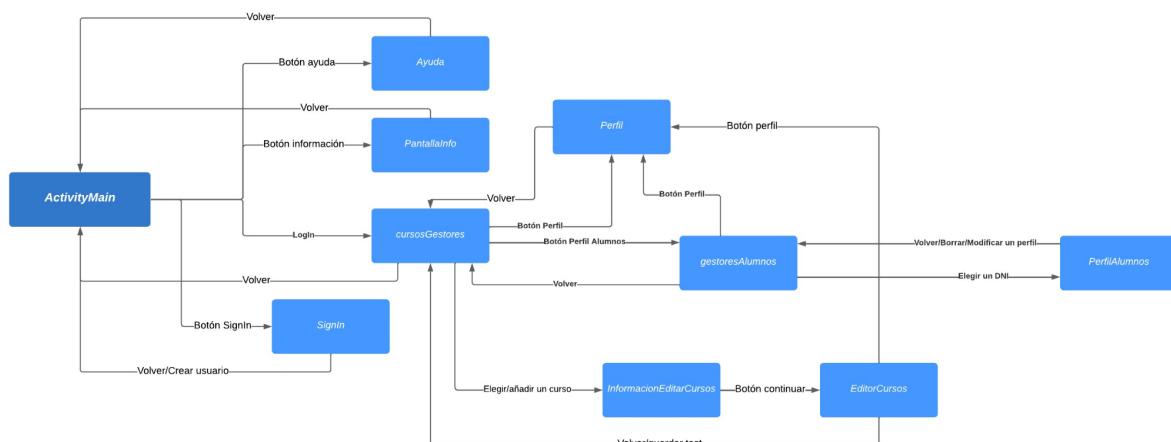


Figura 5. Diagrama de navegabilidad para gestores.

[Enlace](#)

Para el perfil de alumnos sería el siguiente:

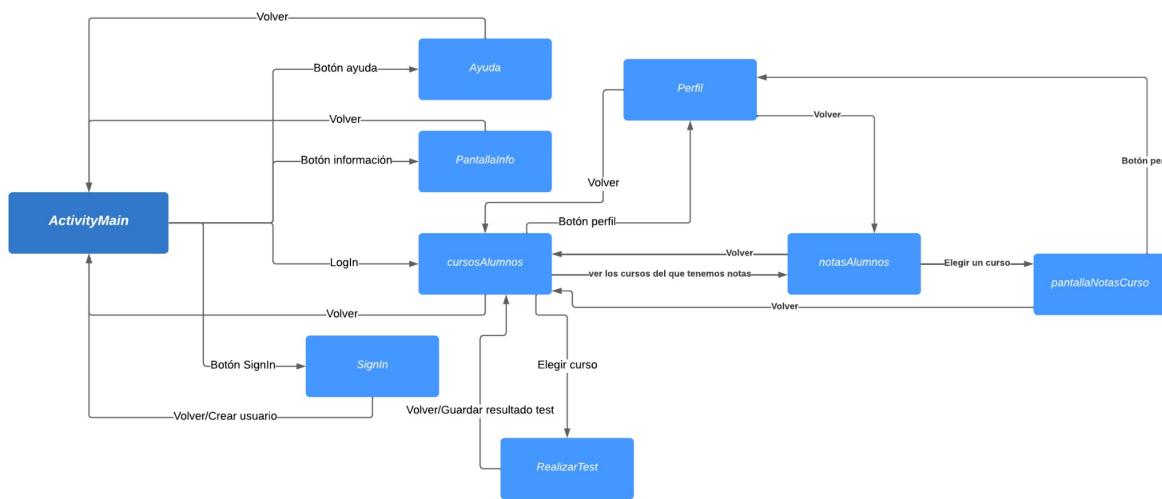


Figura 6. Diagrama de navegabilidad para alumnos.

[Enlace](#)

4. Construcción del Sistema

Para la puesta en funcionamiento de la aplicación se necesitará una base de datos MySQL alojada en un servidor de XAMPP. Para la creación de las tablas e inserción de datos se hará uso del lenguaje Structure Query Language o más comúnmente conocido como “SQL”. Para la conexión entre la aplicación y la base de datos utilizaremos una API escrita en lenguaje PHP para la creación de todas las sentencias CRUD (Create, Read, Update, Delete) necesarias. La API realizará peticiones sobre la base de datos y obtendrá la información solicitada. Para la realización de pruebas sobre la API hemos utilizado la herramienta Postman que permite realizar peticiones sobre cualquier API y mostrar el resultado devuelto por la misma.

Para llevar a cabo un control sobre las diferentes versiones del proyecto y los cambios realizados en cada fecha se hará uso de GIT, una herramienta que permite el control de las diferentes versiones del proyecto y la cual permite subir los cambios realizados a un repositorio en [GITHUB](#) y poder tener una copia de seguridad del mismo.

Una vez completado lo anterior, lo único que faltaría sería el desarrollo de aplicación, es decir, la creación de las interfaces necesarias para cada usuario, la lógica necesaria para el correcto funcionamiento entre ellas y la correcta realización de peticiones HTTP.

5. Conclusión

Durante la realización de este proyecto he adquirido ciertos conocimientos de los cuales cabe destacar:

- La conexión entre dos lenguajes de programación (Java y PHP) mediante el envío de datos en formato JSON.
- La conexión entre PHP y la base de datos alojada en MySQL.
- He aprendido a conectar mi dispositivo móvil con una base de datos alojada en un servidor Xampp.
- He aprendido a usar una librería de la cual no tenía conocimiento alguno antes de comenzar el proyecto.

Durante la realización de este proyecto me he enfrentado a diversos problemas, por ejemplo:

- Para la conexión de la aplicación con la API al hacer una petición de tipo DELETE no se llega a realizar correctamente, por lo que para su solución se realizará una petición de tipo POST en su lugar. La sentencia SQL que elimina el registro se realiza correctamente por lo que no hay ningún problema al utilizar dicho tipo de petición.
- A la hora de pedir confirmación al usuario para el borrado de datos al utilizar un ImageView y un ImageButton a la hora de crear el AlertDialog.Builder no se llegaba a mostrar la alerta hasta el segundo click en el botón ya que estaba creando por segunda vez el Listener al hacer click (onClickListener). Para resolver dicho problema he borrado el clickOnListener que abarcaba desde antes del AlertDialog.Builder hasta el fin de este.
- Cuando intentamos añadir un archivo obteniendo su URI con el método getData(), nos devuelve su URL indicando si es content, es decir, con este formato content:Downloads%2cursos%2id_del_archivo, por lo que para la obtención del nombre cree un método llamado obtenerNombreArchivo(Uri uri) al cual le pasamos por parámetro la ruta obtenida del archivo, seguidamente comprobamos que el esquema es content, seguidamente creamos un cursor para obtener los resultados de la consulta realizada con el método “getContentResolver()”, seguidamente comprobamos que el cursor tenga valores y se pueda mover a la primera fila obtenemos el índice de la columna que contiene el nombre del archivo, seguidamente se comprueba si el índice es -1, es decir, no ha recopilado ningún valor, si el índice es diferente a -1 obtenemos el valor del nombre con el método “getString(indice)” del cursor. Finalmente, devolvemos el valor del nombre.

6. Glosario de términos

TextView: Componente que podemos incluir en las vistas de la aplicación para mostrar texto al usuario.

EditText: Es una variante del TextView que permite al usuario introducir texto.

API: Pieza de código que permite a diferentes aplicaciones comunicarse entre sí y compartir información y funcionalidades. Una API es un intermediario entre dos sistemas, que permite que una aplicación se comunique con otras y pida datos o acciones específicas.

StringRequest: Clase proporcionada por la librería Volley que permite realizar solicitudes HTTP en formato de texto plano.

Volley: Volley es una biblioteca HTTP que facilita y agiliza el uso de redes en apps para Android.

Map<Key, Value>: Los Objetos MAP son una lista que asocia una clave a un valor o lista de valores. La longitud de los mapas es indeterminada, es decir, puede variar.

List<Obj>: Los objetos List son listas de valores, la longitud de las listas son indeterminadas.

ImageButton: Variante del componente Button que permite mostrar una imagen como botón.

ScrollView: Vista especial que permite al usuario deslizar la pantalla.

LinearLayout: Grupo de vistas que permite alinear todos los elementos secundarios en una única dirección.

AlertDialog.Builder: Ventana emergente que le indica al usuario que debe elegir una opción o ingresar texto.

RadioGroup: Subclase de LinearLayout que permite agrupar varios radioButtons.

RadioButton: Es un control de selección que permite a los usuarios elegir una opción dentro de un conjunto.

CRUD: Es el acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar). Este concepto se utiliza para describir las cuatro operaciones básicas que se pueden realizar en una base de datos.

SELECT: Cláusula que permite obtener los campos deseados de una tabla o vista. Debe usarse con la cláusula FROM.

FROM: Cláusula que permite especificar las tablas de las que obtendremos los campos a mostrar obtenidos en la cláusula SELECT.

WHERE: Cláusula que permite filtrar los datos obtenidos en la cláusula SELECT

CREATE TABLE: Declaración utilizada para la creación de tablas.

DROP TABLE: Declaración utilizada para borrar tablas existentes.

CREATE USER: Declaración utilizada para la creación de usuarios.

DROP USER: Declaración utilizada para borrar usuarios.

CREATE SCHEMA: Declaración que permite crear un esquema en el que guardar la base de datos.

DROP SCHEMA: Declaración que permite borrar un esquema.

INSERT INTO table VALUES (val1,): Declaración que permite la inserción de datos en una tabla.

UPDATE tabla SET campo1 = val1 [WHERE campo2 = val2]: Declaración que permite la actualización de datos de una tabla o de un registro de la misma definido por la cláusula “WHERE”.

7. Bibliografía

[Android Studio](#)

[Android API](#)

[API Java](#)

[Visual Studio Code](#)

[Postman](#)

[API PHP](#)

[Dia](#) → aplicación utilizada para la creación del modelo Entidad-Relación.

[Visual Paradigm](#) → Aplicación utilizada para la realización del diagrama de casos de usos.

[GIT](#)

[LucidChart](#) → Aplicación utilizada para la creación de los diagramas de navegabilidad y diagrama de clase.

[Documentación SQL](#)

Apéndices

README.txt → Fichero de explicación para la correcta instalación y configuración del servidor y la Aplicación.

Documentación del proyecto → StudyHub.pdf

Colección de Potsman para la realización de pruebas sobre la API:
tfg_sergio.postman_collection.json

Diagrama de navegabilidad:

- Gestores: DiagramaDeNavegabilidad_Gestor.png
- Alumnos: DiagramaDeNavegabilidad_Alumno.png

Modelo Entidad-Relación (Archivo original Dia): ModeloEntidad-Relacion.dia

Diagrama de casos de uso: Diagrama de casos de uso.jpg

Diagrama de subsistemas: Diagrama de Subsistemas.jpg

Diagrama de clases: DiagramaDeClases.png

Las capturas de pantalla de las interfaces se encuentran en la carpeta “imagenesPantallas” dentro de la carpeta “OtrosDocumentos”.