

PROYECTO FIN DE CICLO

AcademyCode

Realizado por: Isip Niculita Samuel



Desarrollo de aplicaciones multiplataforma

Contenido

Contenido	1
1. INTRODUCCIÓN	2
2. DEFINICIÓN DEL PROYECTO.....	3
2.1 Objetivos del proyecto	3
2.2 Contexto y motivación.....	3
2.3 Estado del arte.....	4
3. CALENDARIO DEL PROYECTO	5
3.1 Metodología Ágil	5
3.2 Fases del proyecto	9
4. INFRAESTRUCTURA.....	10
4.1 Elementos Hardware	10
4.2 Elementos Software	10
4.3 Recursos Web	11
5. LENGUAJES INCLUIDOS	11
6. POSIBLES RIESGOS	11
7. DOCUMENTACIÓN	12
7.1 Sistema Operativo.....	13
7.2 Requerimientos Funcionales	14
7.3 Requerimientos No Funcionales	17
7.3.2 Disponibilidad.....	17
7.3.3 Interfaz Gráfica.....	18
7.4 Casos De Uso	18
7.5 Diseño Técnico.....	19
7.5.1 Arquitectura Del Servicio.....	19
7.5.2 Arquitectura Física	19
7.5.3 Arquitectura Lógica.....	20
7.5.3.1Capa De Presentación.....	20
7.5.3.2Capa De Negocio	25
7.5.3.3Capa De Datos	26
7.5.4 Arquitectura De Bases De Datos	27
7.5.5 Diagrama De Clases	28
7.5.6 Prototipo.....	29
7.6 Pruebas	29
8 IMPLEMENTACIÓN	33
9 CONCLUSIONES Y MEJORAS	37
10 BIBLIOGRAFÍA	38

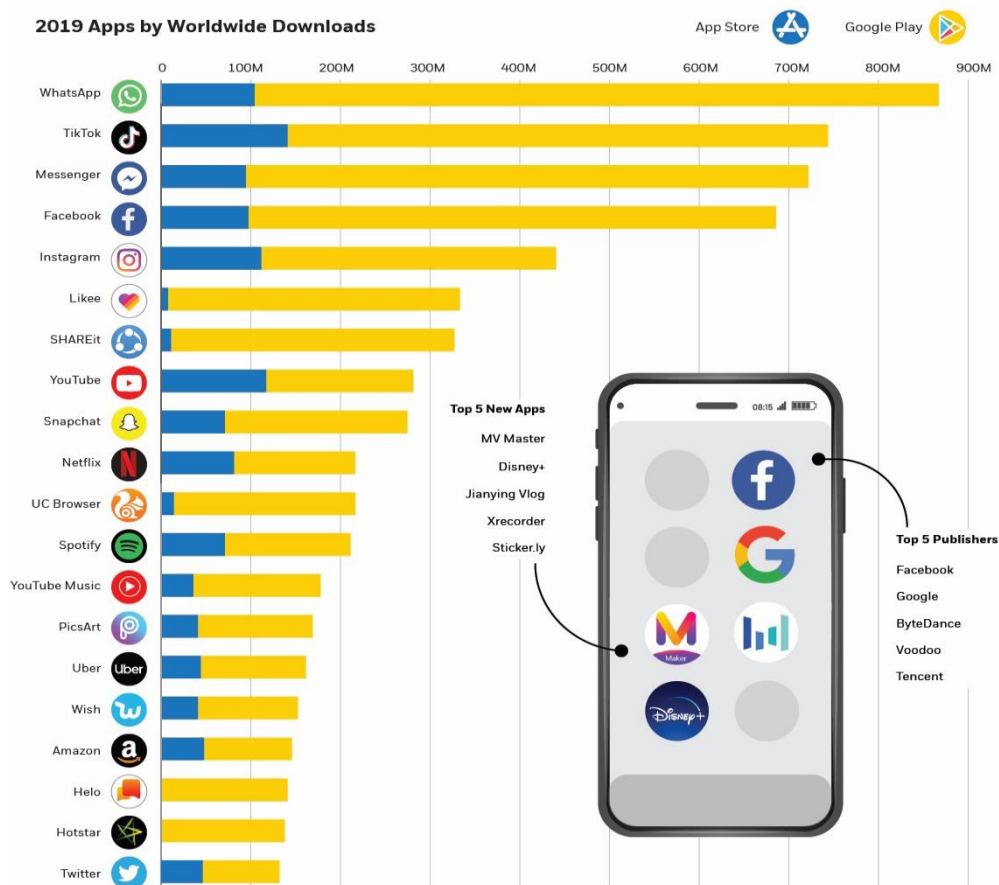
1. INTRODUCCIÓN

Estamos viviendo en un proceso de evolución y desarrollo de las nuevas tecnologías, que comenzó con la *Revolución de la información y las telecomunicaciones* (1985-2000) y el impacto que tuvo sobre la sociedad se sigue viendo incluso 20 años después. Aunque esta revolución quedó en el pasado, vemos como hoy en día el conocimiento y la información son cada vez mayores, lo que facilita mucho más la capacidad de innovación de las tecnologías que ya disponemos.

Con los recursos de hoy en día, la capacidad de aprendizaje cada vez mayor de las personas y la mentalidad de la sociedad, vemos como el mundo entero se está preparando para una segunda revolución tecnológica. Tal vez sea iniciada por “*el internet de las cosas (5G)*” o por otros factores, no se sabe cómo podría comenzar o si ya estamos en sus comienzos, lo que si sabemos seguro es que esta, llegará.

Antes la situación dada, está claro que debemos estar preparados para poder estar al día con las nuevas tecnologías, para saber cómo funcionan técnicamente y saber utilizarlas.

Uno de los sectores tecnológicos más avanzados en la actualidad, son los dispositivos móviles. En toda familia hay por lo menos un dispositivo móvil por persona, pero si pensamos en las aplicaciones que contienen cada uno de estos, llegamos a ver cifras gigantescas del número de usuarios que utilizan determinadas aplicaciones. Hoy en día las aplicaciones más utilizadas cuentan con cientos de millones de usuarios.



Ante tal cantidad de usuarios y de aplicaciones, por desgracia, son pocas las personas que además de utilizarlas, se interesen por saber ¡cómo crearlas! Los programadores son pocos comparados con la gran cantidad de trabajo que conlleva crear y mantener todas esas aplicaciones y sistemas. Aunque cada vez se ve más gente interesada en aprender a programar, por eso cada vez hay más gente que necesite una correcta formación.

Ante esta necesidad, me dispongo a realizar una aplicación que ayude a todos aquellas personas que están comenzando en el mundo de la programación, para facilitarles el trabajo de búsqueda de información y permitirles un aprendizaje más rápido y efectivo.

2. DEFINICIÓN DEL PROYECTO

A continuación, se explicará brevemente el porqué de esta aplicación, cual es el contexto, la motivación y el objetivo de este proyecto. Debo aclarar que esta aplicación forma parte del proyecto de fin de grado del curso de Desarrollo de Aplicaciones Multiplataforma, para el cual también se ha elaborado un plan de empresa (documento adjunto con explicación extensa), en el que se indica que se crea una academia para formación de personas de todas las edades interesadas iniciarse en el mundo de la programación. Esta academia, **AcademyCode**, dispondrá de una aplicación que es la que se desarrolla en este proyecto, pero no estará disponible únicamente para los alumnos de la academia sino para todo el mundo que desee utilizarla, pagando la suscripción que esta incluye.

2.1 Objetivos del proyecto

La aplicación Android a resultar a partir de este proyecto, proporciona a los alumnos de la academia **AcademyCode** un entorno de formación relacionado con los distintos lenguajes de programación más utilizados en la actualidad.

Su objetivo principal es facilitar a los alumnos los recursos necesarios para especializarse en uno o varios lenguajes de programación. Estos recursos son tanto documentos con teoría y ejercicios como material multimedia y un foro por el cual se puede comunicar directamente con los demás participantes de la academia.

2.2 Contexto y motivación

La cantidad de información disponible en internet es innegablemente, grandísima, y más si se trata de las nuevas tecnologías. Hay muchas fuentes para la formación de todo tipo de personas, a través de libros, tutoriales en video, ejercicios en pdf, etc.

Todo esto es una gran ventaja y algo positivo, el único inconveniente es que, ante tal cantidad de información, una persona que se quiere iniciar en la programación, le costará mucho buscar y entender toda esta cantidad de información.

Está comprobado que la mayoría de las personas que quieren iniciarse en este mundo, suelen toparse con tanta variedad de documentación que se desmotivan y pierden el interés por seguir aprendiendo.

De ahí surge la idea de utilizar las tecnologías móviles para solucionar el problema mencionado y poner a disposición del público en general la posibilidad de tener un conjunto de documentación, tutoriales y ejercicios bien estructurados y pensados para una fácil iniciación en el mundo de la programación.

2.3 Estado del arte

En este punto se hará un análisis de las distintas aplicaciones ya existentes en el mercado (Google Play), extrayendo las cualidades que se pueden implementar en esta aplicación y saber que funcionalidades son más utilizadas por los usuarios. También se tendrán en cuenta características técnicas.

Diferentes compañías tecnológicas y creadores de contenido educativo proporcionan formación tanto en Youtube como en páginas web propias, de estas hay una gran variedad y cantidad por todo internet. Aunque las aplicaciones Android de este tipo son escasas en Google Play, algunas de ellas se presentarán a continuación.

Estas aplicaciones se han descargado y probado en la medida de lo posible para ver mejor lo que pueden hacer o no.

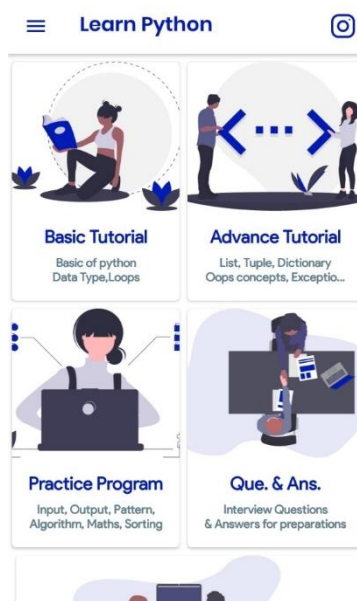
- **Become Coder-Learn**

Libros: Permite ver libros en formato PDF dentro de la misma aplicación, sin descargarlos. También se permite descargar cada PDF. Tiene libros de diferentes lenguajes de programación, aunque no están categorizados correctamente.

Ejercicios/Proyectos: Tiene una sección en la que se pueden ver y descargar ejercicios prácticos e ideas de proyectos enteros.

- **Learn Python**

Teoría: Tiene tutoriales básicos como avanzados, en ambos aparte de la teoría se incluye también código de ejemplo.

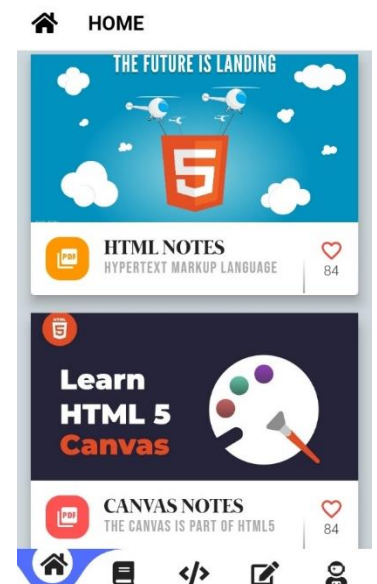


No son pdf y no se pueden descargar. Marca lo que has leído para llevar un mejor orden.

Ejercicios: Los ejercicios son simplemente secciones de código, no son enunciados con problemas a resolver.

Preguntas y respuestas: Tiene una sección de preguntas y respuestas, no se pueden añadir nuevas para que las contesten sino, que simplemente ya hay por defecto.

Área de código: Se puede compilar directamente el código en diferentes lenguajes de programación.



- **SoloLearn**

Esta aplicación es la más avanzada. Solo he recogido las características a las que con mis conocimientos podría llegar a realizar y las que me serían más útiles. Llega al nivel de una red social en la que se puede compartir grandes cantidades de código y ejecutarlas directamente en el programa.

Teoría/Ejercicios: Estos van juntos, tras finalizar la teoría, que no permite descargarse, se debe realizar una prueba, un ejercicio práctico en base a lo aprendido.

Foro: Se puede compartir en él, programas enteros y que los demás usuarios lo compilen, se pueden hacer preguntas y ser respondidas por la comunidad. Cada comentario tiene un sistema de valoración por el cual se organizan.

La documentación está muy bien organizada y se puede encontrar formación tanto para principiantes como para avanzados. Se puede acceder a documentación de prácticamente todos los lenguajes de programación existentes.



Aunque todas tienen un parecido funcional con la aplicación del proyecto, esta última se centra más en un grupo conocido de usuarios (los alumnos) que tendrán el mismo nivel de conocimientos.

Estas aplicaciones no se centran en grupos específicos ni tampoco proporcionan un foro en el que se pueda responder a dudas por parte del profesorado. Entre todas estas, *SoloLearn* es la más avanzada y la única que dispone de foro.

Además, ninguna dispone de tutoriales video que, en mi opinión, es una de las necesidades más importantes para un usuario a la hora de una mejor comprensión.

3. CALENDARIO DEL PROYECTO

El proyecto se dividirá en varias partes, que se irán entregando cada semana a través de la plataforma Delphos Papas. Estas entregas tienen como finalidad comprobar el desarrollo del proyecto.

3.1 Metodología Ágil

Por definición, las **metodologías ágiles** son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo **flexibilidad** e **inmediatez** en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

La **metodología Scrum** es un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos. Es decir, se trata de una **metodología de trabajo ágil** que tiene como finalidad la entrega de valor en períodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación.

¿En qué se basa la metodología Scrum?

Al estar enmarcada dentro de las metodologías agile, Scrum se basa en aspectos como:

- La **flexibilidad** en la adopción de cambios y nuevos requisitos durante un proyecto complejo.
- El factor **humano**.
- La **colaboración** e interacción con el cliente.
- El desarrollo iterativo como forma de asegurar buenos **resultados**.

Ventajas de la metodología Scrum

- Scrum es muy **fácil de aprender**: los roles, hitos y herramientas son claros y tienen un objetivo por lo que es un método muy relacionado con nuestra manera diaria de trabajar.
- El cliente puede comenzar a **usar el producto rápidamente**.
- Se agiliza el proceso, ya que la entrega de valor es muy frecuente.
- Menor probabilidad de sorpresas o imprevistos, porque el cliente está viendo frecuentemente el proyecto.

Desventajas de la metodología Scrum

- Aunque Scrum sea fácil de aprender, es muy **difícil implementarlo**. Esto supone una predisposición y un cambio de cultura de la organización que debe ir desde los altos mandos hasta los clientes.
- La necesidad de tener **equipos multidisciplinarios** puede ser un problema, ya que es difícil encontrar personas que sean capaces de hacer todo el trabajo de un equipo.
- El equipo puede tender a realizar el camino más corto para conseguir el objetivo de un *sprint*, el cual no siempre ofrece resultados de calidad.

En definitiva, **Scrum** es especialmente interesante para proyectos en los que el objetivo es la entrega de valor continua al cliente para poder empezar a ver resultados lo antes posibles. Además, esta metodología permite agilizar procesos, practicar la transparencia y motivar al equipo a través de la autonomía y la independencia.

Sprint review

La review del valor que vamos a entregar al cliente se hace en esta reunión, al final de cada sprint. Su duración es de **4 horas para sprints de un mes**, y es la única reunión de Scrum a la que puede asistir el cliente. En ella el Product Owner **presenta lo desarrollado al cliente** y el equipo de desarrollo **muestra su funcionamiento**. El cliente valida los cambios realizados y además brinda feedback sobre nuevas tareas que el Product Owner tendrá que agregar al Product backlog.

Sprint1

Plan	Diseño	Desarrollo	Test	Implementación	Revisión
Descripción del proyecto	Con el proyecto de empresa busco las cualidades y funcionalidades que serán útiles, sus objetivos y el alcance de la aplicación.	Desarrollo la descripción del proyecto mediante 2 puntos, la idea en si (motivación y utilidad) y el contexto.	-	-	OK
Metodología de desarrollo	Investigo sobre las diferentes metodologías existentes.	Defino porque he elegido esta metodología(scrum).	-	Creo las tablas en word	OK
Estado del Arte	Busco aplicaciones parecidas a la que voy a desarrollar.	Descargo aplicaciones y las pruebo.	Busco características útiles y factibles.	Enuncio las características que pueden ser útiles y una captura.	OK
Diagrama casos de uso	Desarrollo una idea general de las funcionalidades.	Descargo ArgoUML para el diseño. Realizo las los casos y relaciones.	-	Captura del diagrama.	Distribuir mejor.
Diagrama E/R de la BD	Busco app online para diseño E/R.	Creo las tablas de la BD en relación con el diagrama casos de uso.	Prueba de inserción en la BD.	Las importo en phpmyadmin (mysql)	Añadir tabla intermedia.

Sprint2

Plan	Diseño	Desarrollo	Test	Implementación	Revisión
Diagrama de Gantt	Diseño online en Venngage	Según los criterios de evaluación y los sprints	-	Integrar al Proyecto como imagen	OK
Desglosar Casos de Uso (Diseño, Prueba, Implementación)	Capturas de las interfaces, tablas en las que documentar las pruebas.	Desarrollo de código en el backend de las interfaces. Pruebas mediante Debugger	-	Introducir resultados de las pruebas en las tablas del proyecto. Solucionar los errores encontrados en las pruebas.	OK

Sprint3

Plan	Diseño	Desarrollo	Test	Implementación	Revisión
Menú Principal	Realizado a través de xml. Incluyendo CardView, Menú desplegable y pantalla de ajuste.	Con código java se mantiene la sesión activa del usuario, se da funcionalidad a los elementos del diseño, etc.	Abrir Menú y sus distintas funcionalidades, pulsar en las cardview, entrar sin conexión, etc.	El código se implementa en clases java, organizadas en paquetes.	OK

Desarrollo de aplicación móvil Android para formación

Actividad: Teoría	Se diseñan las cardview y su deslizamiento horizontal y la lista de los pdf's.	Mantener sesión activa y sincronización con el servidor, funcionalidad al hacer click en los cardview, etc.	Comprobar si los datos coinciden con la BD del servidor, abrir las cardview y los pdf's.	Las clases .java se organizan dentro de la carpeta del menú principal.	OK
Actividad: Ejercicios	Cardview de menor tamaño en dos columnas, con desplazamiento vertical.	Crear las clases para adaptar la lista de elementos, añadir consulta a php, introducir datos en la bd.	Comprobar si los datos son correctos, si las tablas se relacionan bien (teoría con ejercicios), probar sin conexión.	Se organizan las clases según la estructura de la Teoría, pero en el paquete de Ejercicios.	Intercambiar diseño con Menú.

Sprint4

Plan	Diseño	Desarrollo	Test	Implementación	Revisión
Cambio Diseño Menú e Interfaz Ejercicios	Intercambio de diseño de las card view entre las dos interfaces.	Copiar código de una interfaz a la otra, ajustando las imágenes.	Se prueba que al pulsar cada cardview se acceda a las actividades que les corresponde y se comprueban las imágenes.	Guardar los cambios y subirlos al repositorio de GitHub, cambiar las capturas de las interfaces en la documentación.	OK
Actividad: Cursos Video	Se diseñan las cardview y su deslizamiento horizontal vertical. Se diseña el menú de desplazamiento horizontal.	Se utilizan los recyclerview de las otras interfaces, pero en cada cardview se hace un enlace a los videos de la BD. Se añade un botón que acceda a otra actividad para mostrar pantalla completa.	Comprobar que en cada cardview corresponda el video y el título, y que para cada lenguaje de programación se correspondan los videos del curso.	Añadimos los datos de los videos a la base de datos y tras las comprobaciones subimos los cambios al repositorio. Añadimos la documentación de la actividad.	Añadir video editado con programas de las prácticas

Sprint5

Plan	Diseño	Desarrollo	Test	Implementación	Revisión
Actividad: Foro	Se diseña como se verá cada mensaje, el botón de enviar y el editext para introducir el mensaje.	En el xml se implementan elementos ya utilizados anteriormente. Se ajusta la lista de mensajes para que se vea desde el final.	Se comprueba que los mensajes estén en orden de envío y que la lista de mensajes sea deslizable.	Hacer capturas de pantalla de la actividad y añadir al proyecto.	OK
Usuario / Servidor	Diseñamos la tabla en la BD.	Programamos la parte de acceso a la base de datos del servidor, con PHP.	Probamos que los datos lleguen correctamente a la BD. Se prueba desde otros dispositivos.	Codificamos los métodos necesarios para recoger los mensajes de la BD y mostrarlos en pantalla.	OK
Información Usuario	Diseñamos la interfaz para mostrar la información más importante del usuario.	El xml se desarrolla para desplegarse al hacer click en la imagen del usuario. Incluimos varios botones.	Se comprueba que los datos sean del usuario clickado.	Incorporamos la interfaz en la aplicación y en la documentación a través de capturas de pantalla.	OK

Lista Amigos	Diseñamos una lista para mostrar los usuarios que has añadido como amigos.	El xml es muy simple, parte de lo utilizado hasta ahora en otras actividades.	Comprobamos que los datos de los usuarios sean correctos y que la lista esté completa.	Añadimos la actividad a la parte del menú y a la documentación.	OK
--------------	--	---	--	---	----

3.2 Fases del proyecto

Diagrama de Gantt



4. INFRAESTRUCTURA

Los recursos que se usarán para este proyecto son de 3 tipos, el hardware se tuvo desde el principio del proyecto, pero el software y los recursos web se tuvieron que adquirir durante el desarrollo de este.

4.1 Elementos Hardware

Se ha utilizado un ordenador portátil y un dispositivo móvil.

Componentes	Características técnicas	Tareas
Ordenador Portátil	<ul style="list-style-type: none"> - Asus K540 - Intel Core i3 5005U - 8 GB RAM - 1 TB HDD - IntelHD Graphics 5500 - Windows 10 	Realizar el proyecto con todo lo que implica.
Teléfono móvil	<ul style="list-style-type: none"> - BQ Aquaris U Plus - Qualcomm Snapdragon 430 - 1.4 GHz - 8 núcleos - 4 GB RAM - 5" 	Se harán pruebas de la aplicación, simulando las funciones principales.

4.2 Elementos Software

Se usan varias aplicaciones que permiten realizar las diferentes funciones de la aplicación.

HERRAMIENTAS	TAREAS
ANDROID STUDIO	Entorno de desarrollo para Android
XAMPP	Controlador de servidores
DUC	Servicio que actualiza ip pública
MICROSOFT OFFICE 2016	Suite ofimática para el desarrollo de la documentación
GOOGLE CHROME	Navegador para consultar información
VISUAL STUDIO CODE	Editor de código

POSTMAN	Programa para probar servicios web
VYSOR	Visualizador de pantalla del móvil
7-ZIP FILE MANAGER	Compresor de archivos
JUSTINMIND PROTOTYPER	Aplicación para creación de prototipado

4.3 Recursos Web

Se utilizan otros recursos que son online de compañías con buena reputación.

HERRAMIENTAS	TAREAS
GITHUB	Repositorio y controlador de versiones
PHPMYADMIN	Administrador de BD MySQL
DBDIAGRAM.IO	Página para creación de Diagramas E-R
FLATICON	BD de iconos para aplicaciones
IMGBB	Servicio para guardar imágenes
CREATELY	Servicio para creación de diagramas
NO-IP	Servidor Dynamic DNS

5. LENGUAJES INCLUIDOS

El lenguaje de programación principal y más utilizado en el proyecto es **Java**, que a día de hoy es el segundo más utilizado a nivel mundial según el índice de Tiobe. Es un lenguaje orientado a objetos y utilizado en las aplicaciones Android.

El segundo lenguaje utilizado en el proyecto es **PHP** "Hypertext Preprocessor", uno de los más utilizados en el desarrollo web. Se utiliza para la creación de la API Rest que maneja todas las funciones CRUD, entre otras, de la aplicación.

Para la parte de las interfaces gráficas se utiliza **XML** "Lenguaje de Marcado Extensible", es muy utilizado para el intercambio de información estructurada entre diferentes plataformas. En el caso de Android, la IDE Android Studio lo utiliza a la hora de diseñar las interfaces de cada actividad.

6. POSIBLES RIESGOS

Como en cualquier trabajo, en la realización de un proyecto también pueden surgir imprevistos provenientes de agentes externos o internos al proyecto. Para poder disminuir el efecto de estas eventualidades, se deben determinar, analizar y dar una posible solución a estos riesgos.

En la siguiente tabla se detallarán los riesgos según su importancia:

RIESGO	DETALLE	PROBABILIDAD	IMPACTO	ACCIONES
Mala planificación	No tener claro el objetivo y no seguir las pautas dadas.	Media-Alta	Alto	Analizar bien lo que se pide, seguir las instrucciones dadas por parte del profesorado y utilizar una metodología ágil
Falta de conocimiento	Pueden surgir situaciones en las que se necesite de conocimientos que no se han impartido en el curso.	Medio-Alto	Alto	Investigar todo lo posible, hacer cursos, preguntar a los profesores. Se debe tener una mentalidad abierta en cuanto al continuo aprendizaje.
Fallo hardware o software	Se puede estropear el portátil o el móvil. Errores en el sistema operativo perdiendo todo el proyecto.	Medio-Bajo	Alto	Cuidar el equipo utilizado, realizar copias de seguridad y guardar progreso en la nube o en repositorios. Disponer de otro equipo.
Enfermedad	La salud puede ser comprometida impidiendo así tener el máximo rendimiento o incluso impedir el avance del proyecto.	Bajo	Medio	Adelantar tareas para disponer de más tiempo ante una situación así.

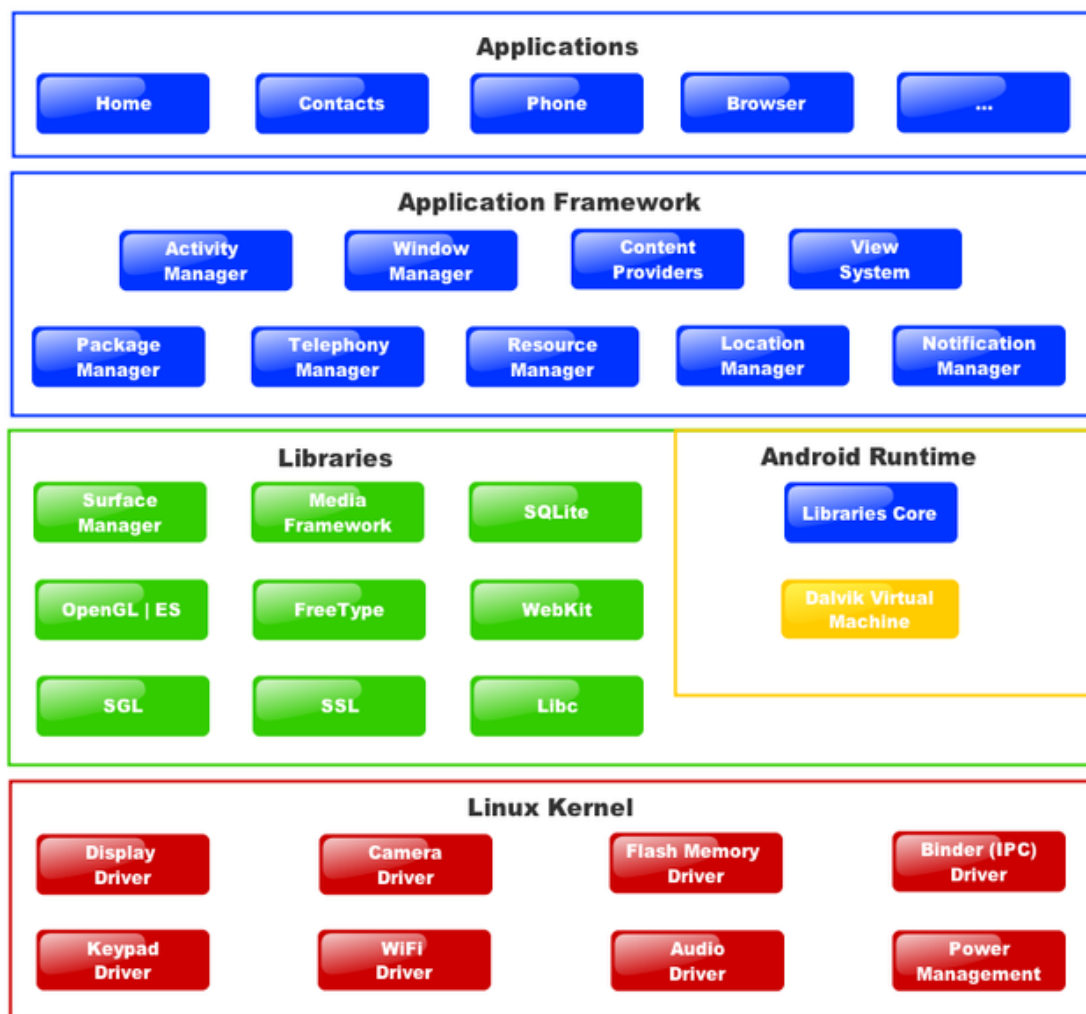
7. DOCUMENTACIÓN

En esta parte del proyecto se desarrollará todo lo necesario para la documentación de la aplicación, se definirán conceptos por escrito y otros a través de gráficos, como los diagramas de caso de uso, de secuencia, entre otros.

7.1 Sistema Operativo

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, **un núcleo de sistema operativo libre, gratuito y multiplataforma.**

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.



Esta sencillez, junto a la existencia de herramientas de programación gratuitas, hacen que una de las cosas más importantes de este sistema operativo sea **la cantidad de aplicaciones disponibles**, que extienden casi sin límites la experiencia del usuario.

Un poco de historia

Android era un sistema operativo para móviles prácticamente desconocido hasta que en 2005 Google lo compró. Hasta noviembre de 2007 sólo hubo

rumores, pero en esa fecha se lanzó la Open Handset Alliance, que agrupaba a muchos fabricantes de teléfonos móviles, chipsets y Google y se proporcionó la primera versión de Android, junto con el SDK para que los programadores empezaran a crear sus aplicaciones para este sistema.

Aunque los inicios fueran un poco lentos, debido a que se lanzó antes el sistema operativo que el primer móvil, rápidamente se ha colocado como el sistema operativo de móviles más vendido del mundo, situación que se alcanzó en el último trimestre de 2010.

En febrero de 2011 se anunció la versión 3.0 de Android, llamada con nombre en clave Honeycomb, que está optimizado para tabletas en lugar de teléfonos móviles. Por tanto, Android ha trascendido los teléfonos móviles para trascender a dispositivos más grandes.

Libertad

Una de las mejores características de este sistema operativo es que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada. Y esto lo hace muy popular entre fabricantes y desarrolladores, ya que los costes para lanzar un teléfono o una aplicación son muy bajos.

Al final podemos resumir que Android es la forma de afrontar la telefonía móvil por parte de Google, que entiende estos dispositivos como una forma de estar conectado constantemente a Internet. Las aplicaciones de Google que vienen preinstaladas en el teléfono, permiten acceder a los servicios de Google de forma muy integrada, aparte de la aplicación Market que permite instalar aplicaciones desarrolladas por terceros de una forma muy sencilla.

7.2 Requerimientos Funcionales

A continuación, se detallarán algunas de las funciones más importantes de la aplicación.

7.2.1 Descripción Básica

-Registrarse como alumno o usuario normal en la app.

Se permite a cualquier usuario crearse una cuenta sean o no alumnos de la academia, aunque los alumnos tendrán a su disposición una mayor cantidad de material didáctico.

-Aprender teoría.

El usuario puede acceder a la documentación del lenguaje de programación que más le interese. Los documentos se verifican que sean aptos para la mejor formación posible del usuario. Se le da la opción de descargar dichos archivos y tenerlos disponibles siempre en su dispositivo móvil para poder aprender de forma offline.

-Realizar ejercicios prácticos.

Al igual que la teoría, los ejercicios también están agrupados por lenguajes de programación y verificados que estén en concordancia con la teoría que se

proporciona. La mayoría de ejercicios cuentan con sus respectivas respuestas. También se pueden descargar para tenerlos offline.

-Visualizar tutoriales.

Los videos proporcionados serán de diferentes canales de YouTube especializados en la tutorización en diferentes lenguajes de programación, para tener acceso a los videos tutorizados por los profesores de la academia se tendrá que estar suscrito a un plan de aprendizaje de esta, o ser un alumno. Los videos solo están disponibles online a excepción de los proporcionados para los alumnos.

-Preguntas o información útil en el foro.

Para proporcionar una formación 24/7 se necesita de un método de comunicación permanente, para esto se dispone de un foro en el que pueden comunicarse todos los usuarios. Muchas de las preguntas serán respondidas por los profesores o por los demás usuarios, formando una comunidad con un nivel de conocimientos similares que pueden ayudarse entre ellos.

-Chatear con otros usuarios.

Si se desea que la comunicación sea individual y privada, se dispone de un chat de solo 2 usuarios.

Todas estas funcionalidades pueden ser accedidas desde el menú principal o de la ventana del menú.

7.2.2 Seguridad

La implementación de una buena seguridad en las aplicaciones que gestionan datos de usuarios es muy recomendada, necesaria e incluso obligatoria según la Ley Orgánica de Protección de Datos (LOPDGDD) *“Las empresas que deben cifrar datos son aquellas que deben implantar medidas de nivel alto por razón de los datos que tratan o el tratamiento que realizan con ellos. Es decir, se deben cifrar: las que traten datos sensibles o especialmente protegidos.”*.



En esta aplicación los datos más sensibles de los usuarios (email, password, mensajes, ip, etc) se cifran en el backend (php). Los métodos de cifrado son:

-El **hash** de contraseñas es una de las consideraciones de seguridad más elementales que se deben llevar a la práctica al diseñar una aplicación que acepte contraseñas de los usuarios.

Si aplicamos un algoritmo hash a las contraseñas antes de almacenarlas en la base de datos, dificultamos al atacante el determinar la contraseña original, pese a que en un futuro podrá comparar el hash resultante con la contraseña original.

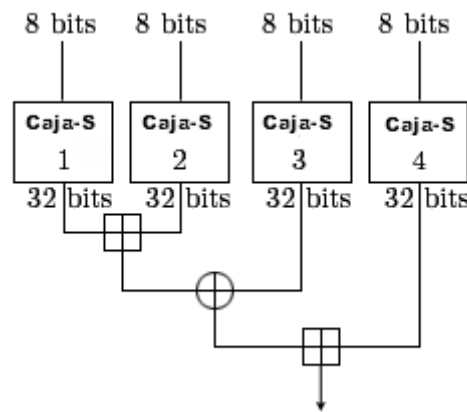
Sin embargo, es importante tener en cuenta que el hecho de aplicar hash a las contraseñas sólo protege de que se vean comprometidas las

contraseñas almacenadas, pero no las protege necesariamente de ser interceptadas por un código malicioso inyectado en la propia aplicación.

¿Por qué las funciones hash más comunes como md5() y sha1() no son adecuadas para las contraseñas?

Los algoritmos hash como MD5, SHA1 o SHA256 están diseñados para ser muy rápidos y eficientes. Con las técnicas y equipos modernos, es algo trivial extraer por fuerza bruta la salida de estos algoritmos, para determinar los datos de entrada originales.

El algoritmo recomendado para el empleo de contraseñas con hash es Blowfish, que es también el predeterminado de la API de hash de contraseñas, que, aunque es significativamente más caro computacionalmente que MD5 o SHA1, sigue siendo escalable.



Al utilizar `password_hash()` o `crypt()`, el valor devuelto incluye la sal como parte del hash generado. Este valor debería almacenarse tal cual en la base de datos, ya que incluye información sobre la función hash que se empleó y así proporcionarla directamente a `password_verify()` o `crypt()` al verificar contraseñas.

El siguiente diagrama muestra el formato de un valor devuelto por `crypt()` o `password_hash()`. Como se puede observar, son autocontenidos, con toda la información del algoritmo y la sal requerida para futuras verificaciones de contraseñas.

`$2y$10$6z7GKa9kpDN7KC3ICW1Hi.f d0/to7Y/x36WUKNP0IndHdkdR9Ae3K`

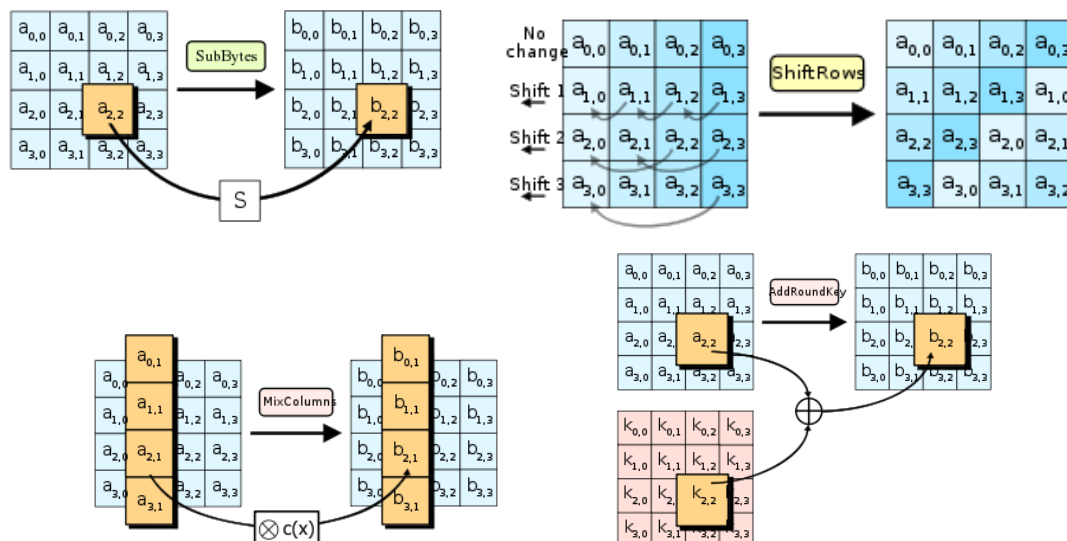
— Salt
— Hashed password
— Algorithm options (eg cost)
— Algorithm

-Advanced Encryption Standard (AES), también conocido como Rijndael, es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica.

Estrictamente hablando, AES no es precisamente Rijndael (aunque en la práctica se los llama de manera indistinta) ya que Rijndael permite un mayor rango de tamaño de bloques y longitud de claves; AES tiene un tamaño de bloque fijo de 128 bits y tamaños de llave de 128, 192 o 256 bits, mientras que Rijndael puede ser especificado por una clave que sea múltiplo de 32 bits, con un mínimo de 128 bits y un máximo de 256 bits.

La mayoría de los cálculos del algoritmo AES se hacen en un campo finito determinado.

AES opera en una matriz de 4×4 bytes, llamada state (algunas versiones de Rijndael con un tamaño de bloque mayor tienen columnas adicionales en el state).



7.3 Requerimientos No Funcionales

En este apartado se detallará la parte no funcional de la aplicación, es decir, la parte que en la que no se tiene en cuenta la información a guardar, ni tampoco funciones a realizar, sino las características de funcionamiento.

7.3.2 Disponibilidad

La aplicación depende de los servicios de varios factores externos, una dependencia es la del servidor, que es la más importante. Si el servidor cae o sufre algún error, toda la aplicación se vería seriamente afectada.

Otra dependencia es la BBDD mysql, que está alojada en el servidor, si se perdiese o por algún motivo ya no hubiera acceso a ella, muchas de las funcionalidades de la aplicación se perderían, la aplicación no está diseñada para funcionar en offline.

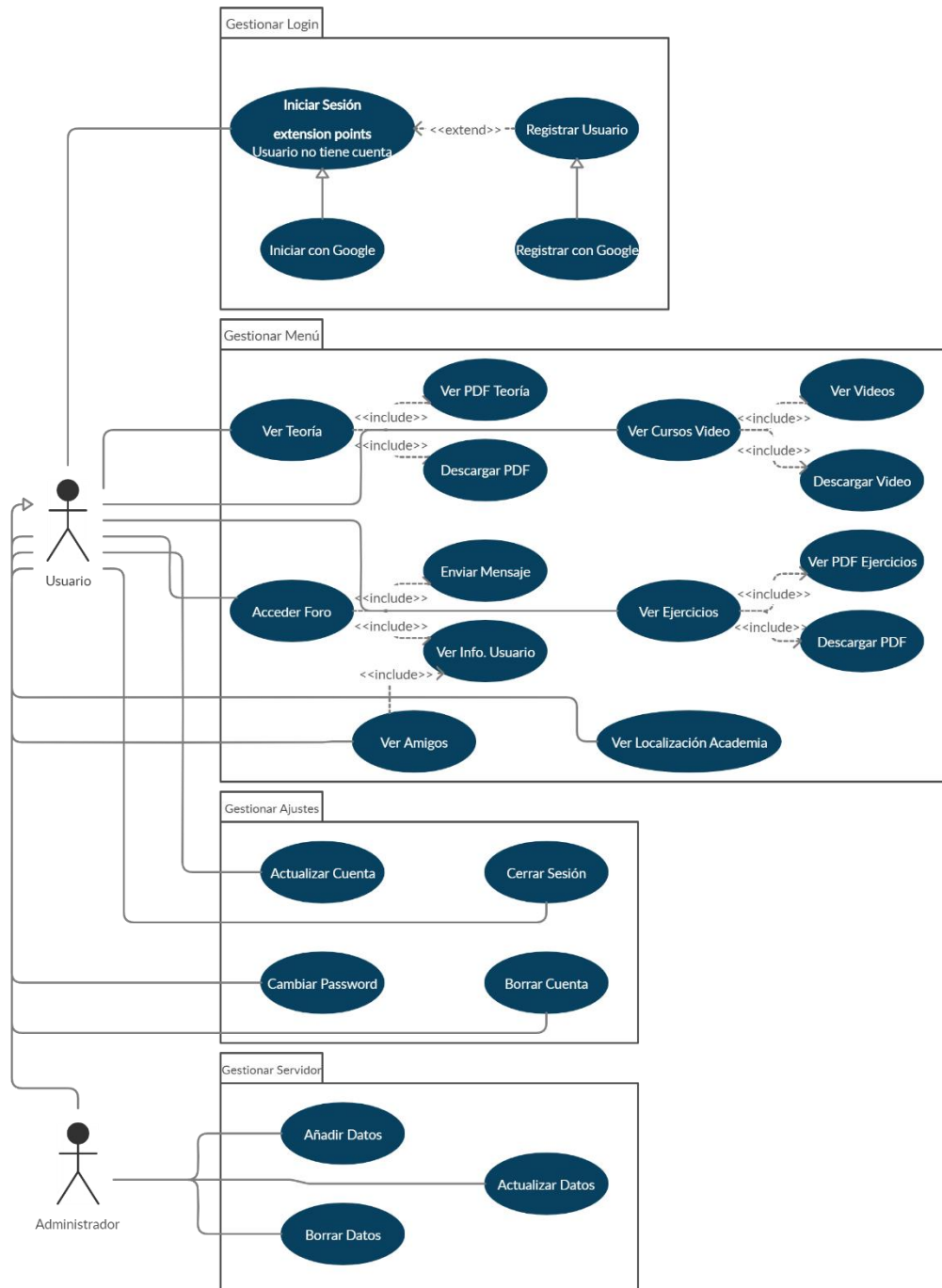
La última dependencia, aunque no tan importante como las anteriores, es la de Google, para el acceso a la aplicación desde la cuenta de este.

7.3.3 Interfaz Gráfica

La interfaz se intenta que sea lo más “responsive” posible, para que se adapte a cualquier dispositivo, aunque en principio la aplicación está pensada y centrada en los smartphones. Se debe poner gran acento sobre el diseño de las interfaces ya que es lo que más influye en el usuario.

El diseño de las interfaces y una mayor explicación de estas se verá en el apartado de “Capa de presentación”.

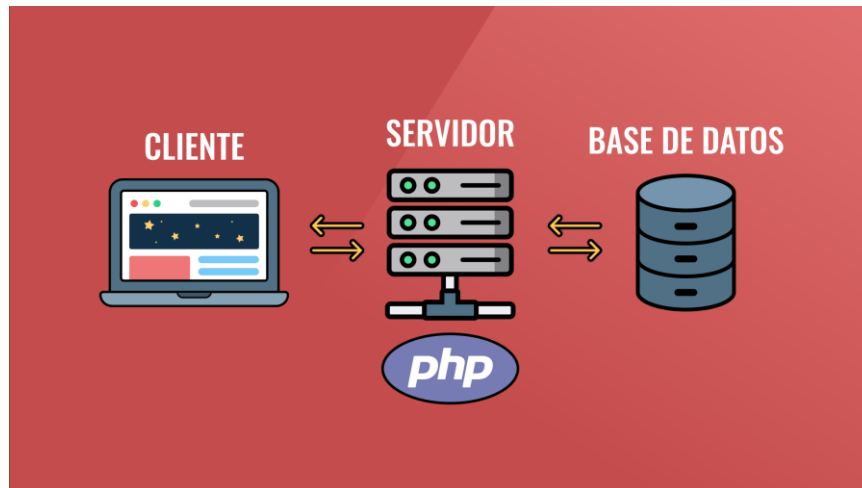
7.4 Casos De Uso



7.5 Diseño Técnico

La arquitectura del proyecto es un modelo cliente-servidor, en la que el cliente es la propia aplicación y el usuario que la utiliza, mientras que el servidor es el servicio que proporciona XAMPP (multiplataforma, Apache, MySQL, PHP, Perl).

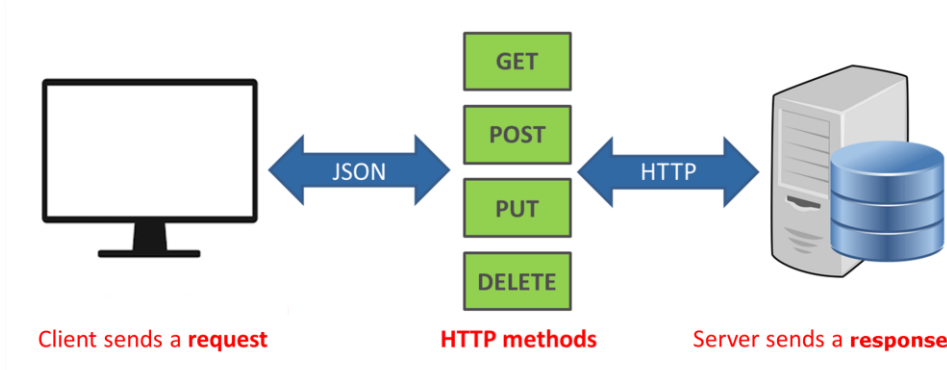
También se cuenta con un hosting de Nolp.



7.5.1 Arquitectura Del Servicio

Apache es el servidor que se utiliza tanto para la API como para la BBDD.

La aplicación se conecta desde internet al servidor Apache, en el que está alojada la API y la BBDD. A través de la API (programada en PHP) se gestionan todas las peticiones hechas al servidor y a mysql. La explicación más sencilla para entender la funcionalidad de la API es compararla con un camarero, esta informa a la BBDD de las peticiones hechas por el cliente y le proporciona a este lo que la BBDD le envíe.



7.5.2 Arquitectura Física

Esta arquitectura es la común para la mayoría de aplicaciones básicas que no necesitan una gran cantidad de recursos.

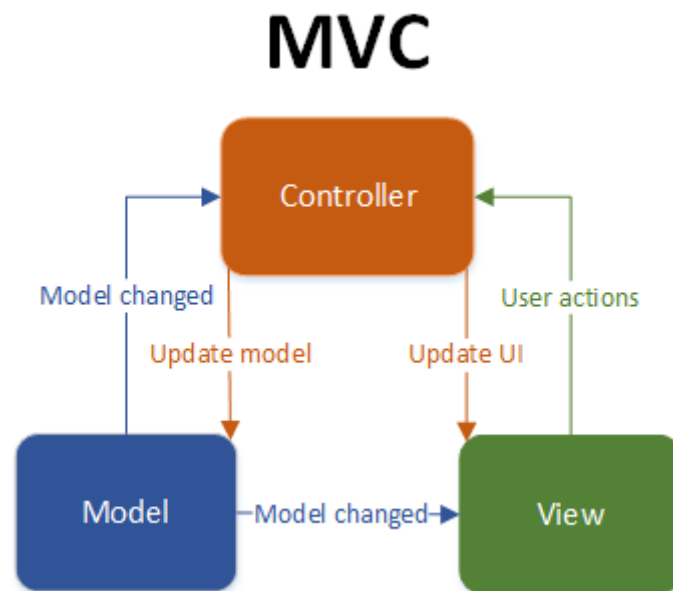
El servidor y la BBDD se puede alojar en el propio portátil ya que la cantidad de información almacenada no necesita de más espacio. Aunque es recomendable que el dispositivo donde estos se alojen tenga suficiente memoria RAM para manejar los datos y la capacidad de estar siempre en funcionamiento.

7.5.3 Arquitectura Lógica

La aplicación tiene una arquitectura por presentación, la capa negocio y la capa de datos. La capa de negocio tiene todas las clases que permiten la ejecución de la aplicación en el dispositivo. La capa de presentación contiene todas las interfaces gráficas, con las que el usuario puede interactuar. Por último, la capa de datos es la que gestiona las peticiones al servidor y a la BBDD, se comunica directamente con la capa modelo.

Utilizar una arquitectura por capas tiene muchas ventajas, algunas de ellas son:

- Posibilita la reutilización de componentes de una sola capa, sin perjudicar al resto de capas. En nuestro caso si se necesitara cambiar el sistema de bases de datos, solo tendría que cambiar la capa de datos y no el resto.
- Permite probar los componentes de forma independiente.



7.5.3.1 Capa De Presentación

Interfaces gráficas

El diseño de las interfaces que forman la aplicación intenta ser lo más parecido a los diseños vistos en las aplicaciones más utilizadas actualmente. Un diseño actual, agradable a la vista y sobre todo intuitivo es lo que se ha intentado implementar en cada interfaz. Estas son muy distintas entre ellas ya que en cada una se desarrolla una actividad diferente, aunque en cuanto al diseño, tienen unas características comunes, como los colores, la forma de los botones, el tipo de letra, etc.

Se describirá el diseño de cada interfaz y las funcionalidades de estas:

Interfaz de Inicio de la aplicación

La pantalla de arranque se muestra cada vez que se abre la aplicación. En la pantalla se muestra una imagen de fondo y sobrepuesto, el logo de la aplicación y de la empresa que representa, además del nombre de esta.

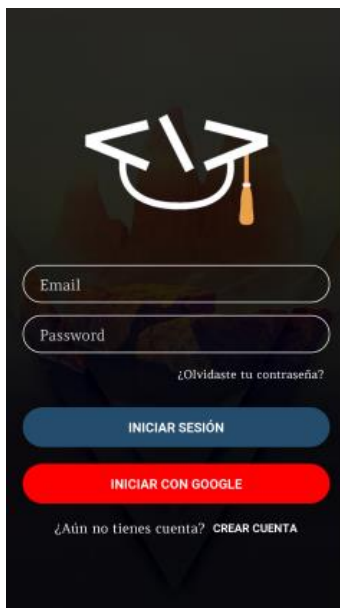
Esta pantalla solo se muestra durante 2 segundos, después desaparece y da paso a otra interfaz.



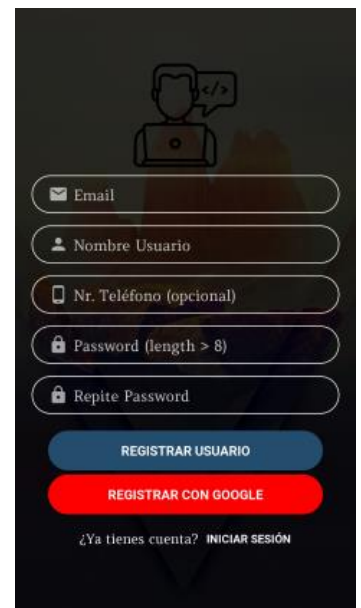
Interfaz Inicio Sesión/Registrar Usuario

La pantalla siguiente a la de arranque es la de Inicio de sesión, que se compone del logo de la aplicación, 2 campos de texto necesarios para poder loguearse (email y contraseña) y 3 botones, un botón para iniciar sesión con cuenta creada exclusivamente en la aplicación, otro iniciar con una

cuenta ya existente (de Google) y el último para acceder a otra pantalla de registro de usuario en caso de que no se disponga de una cuenta.



La pantalla de registro contiene más campos de texto, en los que se deben incluir los datos del usuario, que posteriormente se guardarán en la base de datos (email, nombre de usuario, teléfono y contraseña). Cada campo comprueba el dato introducido que sea correcto según unas condiciones (p.ej: la contraseña tenga más de 8 caracteres, el nombre no exista en la base de datos, el email tenga una estructura válida, etc).



Se incluyen también 3 botones igual que en la pantalla anterior, el primero registra el usuario si cumple todas las condiciones, el segundo introduce los datos de la cuenta de Google seleccionada en los campos de texto para luego poder registrar, el tercer botón vuelve a la pantalla de inicio.

Interfaz Principal/Menú/Ajustes

La pantalla principal es la más gráfica y más compleja en cuanto al desarrollo de su diseño, ya que debe cumplir muchas funcionalidades. Muestra un título y el logo de la aplicación, las “card view” son las que representan otras pantallas en las que se podrá acceder para realizar distintas actividades. Al pulsarlas te dirigen a las pantallas correspondientes.

Actualizar Perfil

Email

Nombre

Teléfono

GUARDAR

Cambiar Password

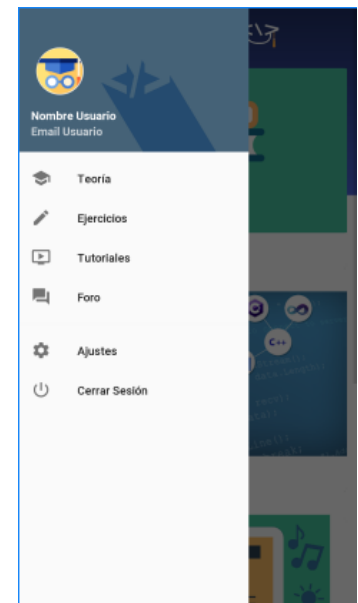
Password Actual

Password Nueva

ACTUALIZAR PASSWORD

BORRAR CUENTA

Deslizando de izquierda a derecha o pulsando el botón de la esquina izquierda, se despliega un menú. En este se mostrará la imagen de perfil del usuario, el nombre y email, además de varios botones (los 4 primeros cumplen la función de las “card view”, redirigen a las mismas pantallas que estas), el botón de ajustes despliega una pantalla, que se mostrará a continuación, y el botón de cerrar sesión permite delogarse y volver a la pantalla de inicio sesión.



La pantalla que se muestra en ajustes se compone de varios bloques, uno para actualizar el perfil (con campos: email, nombre, teléfono y un botón para aplicar cambios), otro para cambiar la contraseña (con campos: contraseña actual y la nueva, y un botón que aplica los cambios si la contraseña actual es correcta) y uno para borrar la cuenta actual, la que está logueada.

Interfaz Teoría

Al entrar desde el menú principal a la card view de teoría, se despliega esta pantalla. Su diseño se basa en el del menú principal, pero cambiando las card view de tamaño, además, los elementos de esta lista ya no se deslizan de forma vertical sino horizontal. Cada card view está diseñada por el logo y el nombre de un lenguaje de programación.

Al pulsar uno de estos card view se despliega una lista de elementos, en este caso son PDF's con teoría del lenguaje de programación pulsado. Estos se cargan desde la base de datos del servidor, se puede actualizar la lista deslizando hacia abajo hasta que aparezca un "swipe".

El diseño de los elementos de la lista son una portada, el título del archivo, autor y la fecha de actualización del archivo.



AcademyCode	
	Ejercicios java 1 Autor: Perez Montés Actualización: 05/05/2020
	Programación estructurada Java Autor: Santiago García Actualización: 06/05/2020
	Fundamentos Java Autor: Gorka Prieto Actualización: 06/05/2020
	Ampliación sistemas Operativos Autor: Rodrigo Santamaría Actualización: 06/05/2020
	Prueba Autor: Prueba Actualización: 06/05/2020
	Prueba Autor: Prueba Actualización: 06/05/2020
	Prueba

Al pulsar en un elemento de la lista, se abre otra ventana, que ya no está alojada en la aplicación sino en el navegador predeterminado del dispositivo móvil. Dependiendo del navegador el diseño podrá variar y las funcionalidades de este también. En el caso de Google Chrome, el diseño es simple y se ajusta el diseño general de la aplicación, las funcionalidades que ofrece son: buscar por palabras en el archivo, descargar el archivo, añadir al Drive, compartir, imprimir, abrir con otra aplicación. Dado que estas funcionalidades ya vienen por defecto en el navegador, ya no se dio la necesidad de implementarlas por código.

Interfaz Ejercicios

En el cardview de los ejercicios, encontramos una pantalla con un diseño diferente en el de teoría, pero también se implementan las cardview con los mismos lenguajes de programación. Cada cardview representa una colección de ejercicios en formato pdf, estos ejercicios están relacionados al lenguaje seleccionado, además en la BD cada libro de teoría puede tener varios de estos ejercicios.

Dentro de cada cardview se comparte el mismo diseño que en el de libros de teoría, es una lista con los ejercicios, mostrando imagen, título, autor y fecha de actualización. Al



seleccionar un elemento se abrirá en el navegador donde se dispondrá de diferentes funcionalidades.

Interfaz Tutoriales Video

Desde el menú se puede acceder a esta interfaz, en la que se encuentran varios videos enlazados de Youtube, estos se agrupan según el lenguaje de programación seleccionado en el menú desplazable de la parte superior de la pantalla.



Cada video se puede ver minimizado o en pantalla completa si se pulsa el botón de “fullscreen”. Los videos están agrupados por cursos, pero mostrados de forma ascendente en cuanto a la dificultad de la teoría que estos desarrollan. Los videos no

pueden descargarse, al igual que no pueden ser descargados desde la plataforma Youtube, pero sí que se pueden compartir los videos o guardarlos en “ver más tarde” en Youtube, solo si estás registrado a la aplicación con la

cuenta de Google.



Interfaz Foro

Al entrar al foro desde el menú, se visualizan directamente todos los mensajes escritos en el foro. Hay una imagen de fondo y los mensajes son cardviews que muestran la foto de perfil, el nombre de usuario, el email, la fecha de envío del mensaje y el mensaje en sí (que tiene 140 caracteres como máximo). El color de fondo de cada mensaje es verde, con una parte más oscura que la otra.

La imagen es la que tenga el usuario y si no tiene se muestra la de por defecto.

El cuadro de texto es de color azul con letras blancas y el botón de enviar es verde. La lista de mensajes se actualiza cada vez que es enviado un nuevo mensaje y siempre se muestra desde el último mensaje enviado, para ver los anteriores se debe hacer scroll hacia arriba.

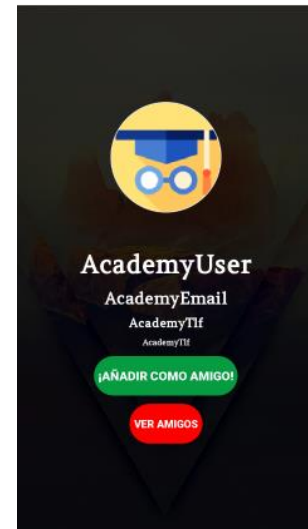


Interfaz Información del usuario

A esta interfaz se puede acceder desde el foro (dando click en la parte superior del mensaje de cada usuario) o desde la interfaz de Amigos.

Su funcionalidad principal es mostrar la información más relevante del usuario seleccionado. También se permite añadir al usuario a tu lista de amigos o ver a todos tus amigos (a través de los botones).

La información que se muestra es: Nombre Usuario, Email, Teléfono, fecha de creación de cuenta e imagen del usuario.

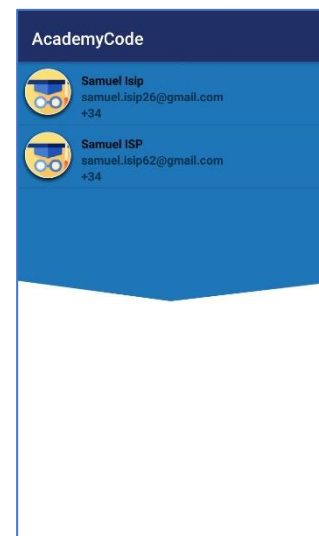


Interfaz Amigos

Se puede acceder a la lista de amigos desde el menú principal o desde la interfaz de información de usuario, con el botón ver amigos.

Se muestran los usuarios agregados a amigos, su nombre de usuario, email, foto perfil y número de teléfono.

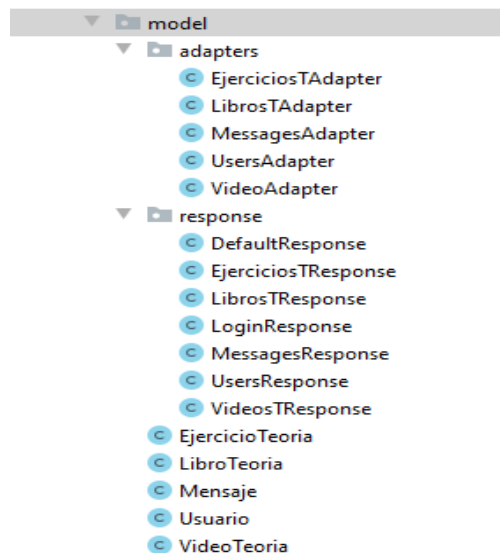
Al pulsar en cada usuario, se abrirá la interfaz en la que se detalla más la información de este.



7.5.3.2Capa De Negocio

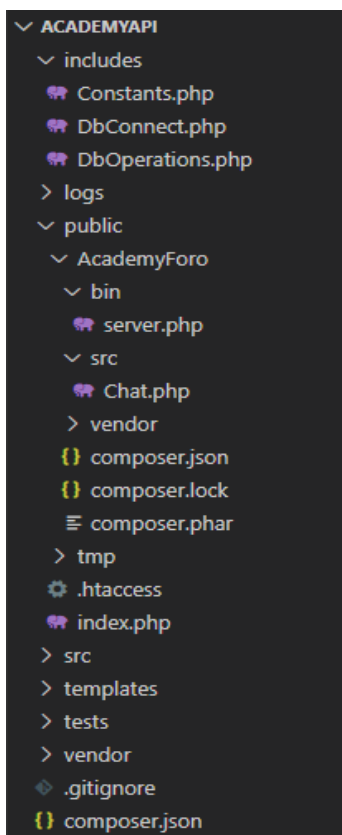
En la capa de negocio tenemos todas las clases que gestionan las acciones propias de la aplicación. Las acciones más básicas como pulsar un botón, desplegar el teclado, guardar la sesión del usuario activa, entre otras, se realizan en esta capa.

Desde esta capa se permite el login a la aplicación a través de la cuenta de Google registrada en el móvil.



7.5.3.3 Capa De Datos

En esta capa gestionamos todo lo relacionado con el servidor y con la BBDD. Uno de los componentes más importantes es la API.



Los archivos que componen a la API son los mostrados en la imagen de la izquierda. A través de estas se manejan todos los datos que pasan desde el cliente al servidor o a la BBDD.

También aquí se hace la conexión para el chat a través de websockets.

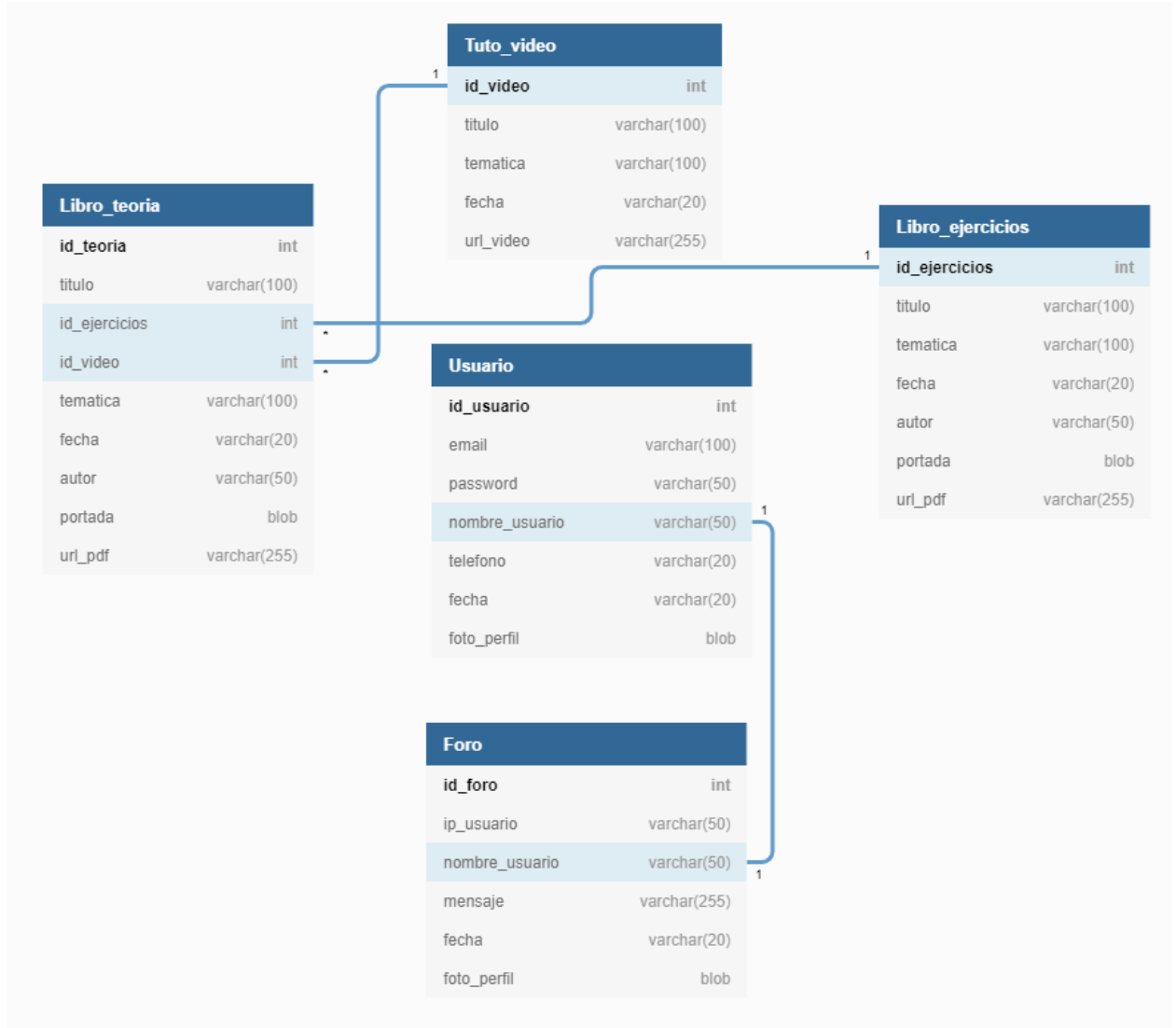
Para que la API funcione necesita que XAMPP esté encendido, tanto el servidor Apache como la Base de Datos mysql.

Las tablas que forman la BBDD, que se manejan a través de phpmyadmin, son estas:

Tabla
<input type="checkbox"/> amigos
<input type="checkbox"/> foro
<input type="checkbox"/> libro_ejercicios
<input type="checkbox"/> libro_teoría
<input type="checkbox"/> tuto_video
<input type="checkbox"/> usuario
6 tablas

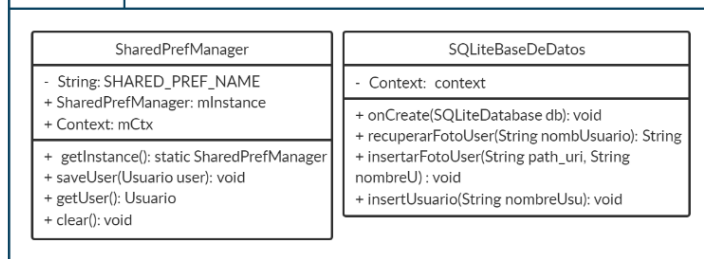
7.5.4 Arquitectura De Bases De Datos

Diagrama E/R de la base de datos/Relacional

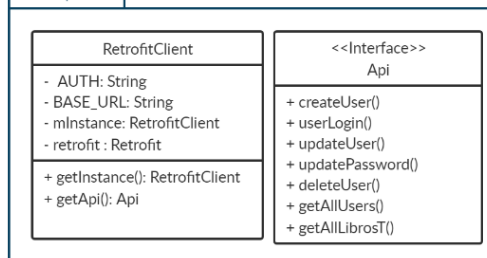


7.5.5 Diagrama De Clases

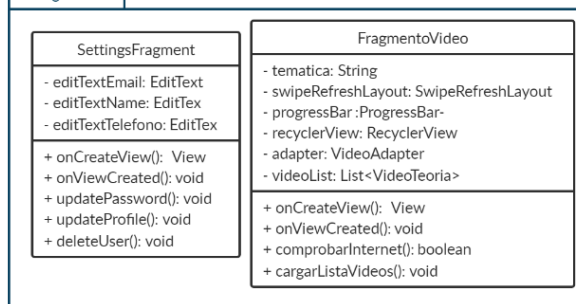
almacenamiento



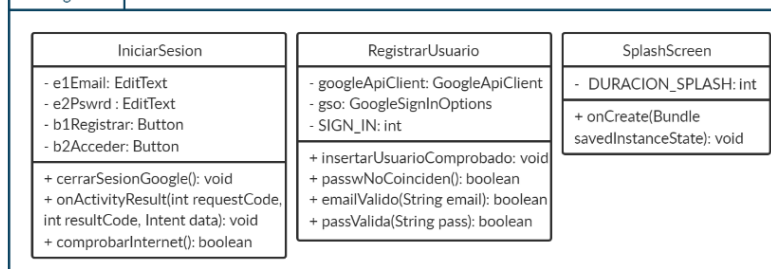
api



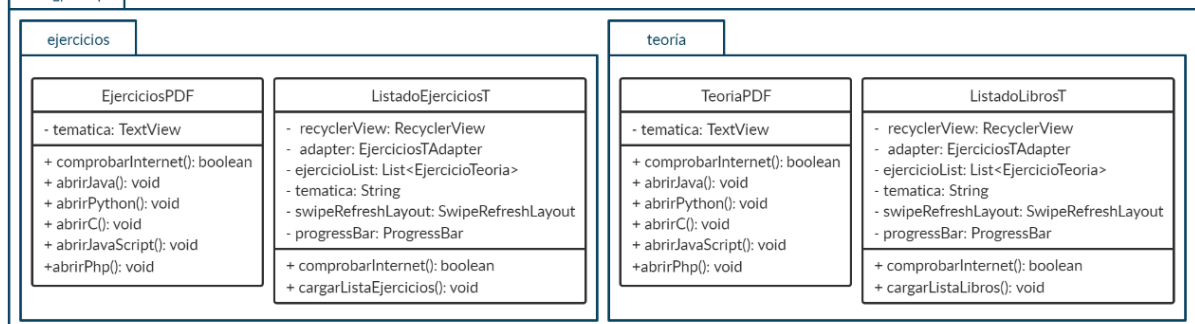
fragments



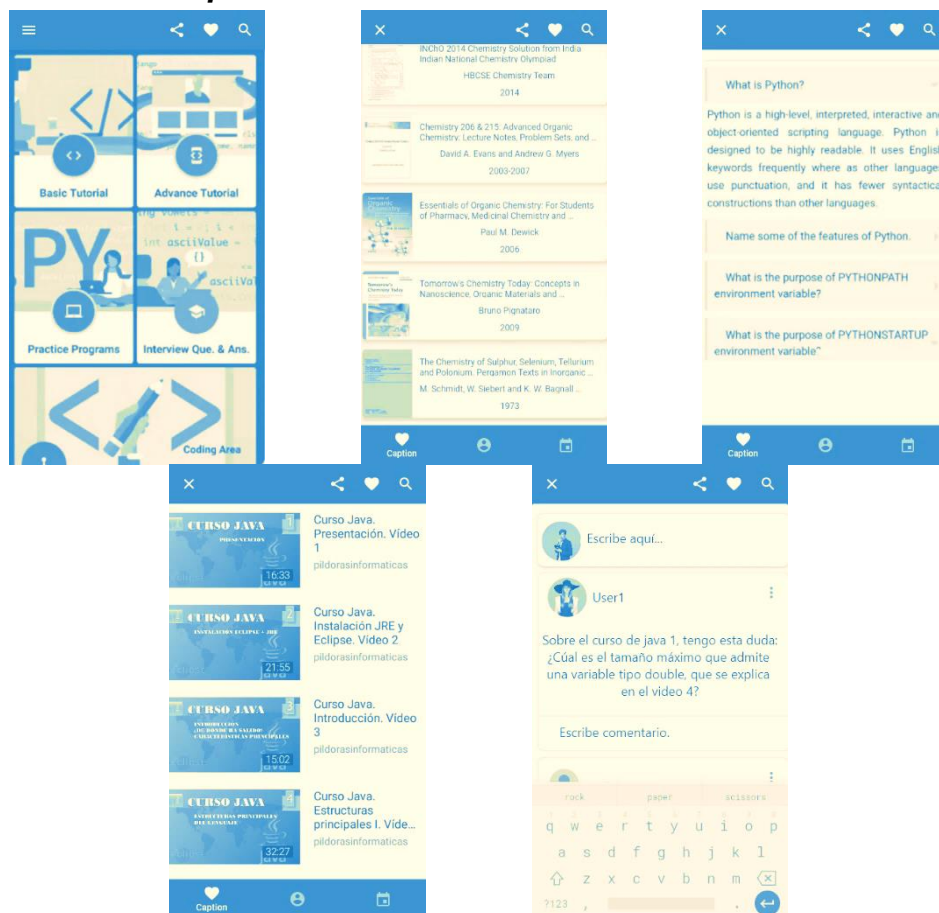
login



menu_principal



7.5.6 Prototipo



7.6 Pruebas

Tras haber diseñado e implementado todas las interfaces de la aplicación y haber codificado las funcionalidades que estas deben proporcionar. El siguiente paso es probar todas las interfaces que interaccionan con el usuario para ver si cumplen su función, si da errores en cuanto a código, en diseño o en integridad de los datos. Para ello se han definido una serie de pruebas que tendrán uno o más requisitos funcionales asociados, y con distintas condiciones iniciales. Las pruebas se han separado en función de los casos de uso a los que están asociados.

Gestión de Login

Prueba	PU01
Descripción	Iniciar sesión sin introducir datos de autenticación
Casos de uso asociados	Iniciar sesión
Condiciones iniciales	-
Resultado obtenido	Mensaje de error en los campos de texto "Email o Password Requerido"

Prueba	PU02
Descripción	Iniciar sesión con datos de login incorrectos
Casos de uso asociados	Iniciar sesión

Condiciones iniciales	Email o contraseña erróneos (Con usuario existente)
Resultado obtenido	Mensaje de error en la parte inferior de la pantalla. -Email incorrecto: “Usuario no existe” -Password incorrecta: “Contraseña incorrecta”

Prueba	PU03
Descripción	Iniciar sesión con Google
Casos de uso asociados	Iniciar sesión
Condiciones iniciales	Tener cuenta de Google
Resultado obtenido	Se introducen los datos de la cuenta en los campos correspondientes

Prueba	PU04
Descripción	Conexión
Casos de uso asociados	Iniciar sesión
Condiciones iniciales	No tener conexión a internet
Resultado obtenido	Mensaje solicitando conexión a internet

Prueba	PU05
Descripción	Registrar usuario con todos los datos introducidos
Casos de uso asociados	Registrar Usuario
Condiciones iniciales	Usuario ya existente
Resultado obtenido	Se muestra un mensaje con Email/Nombre ya existente

Prueba	PU06
Descripción	Conexión Servidor
Casos de uso asociados	Iniciar sesión/ Registrar Usuario
Condiciones iniciales	Servidor desconectado
Resultado obtenido	Se muestra un icono de “cargando” hasta que no haya conexión al servidor

Gestionar Menú Principal

Prueba	PM01
Descripción	Abrir las actividades
Casos de uso asociados	Ver Teoría/Ejercicios/Tutoriales/Foro
Condiciones iniciales	Usuario Logueado y Servidor Activo
Resultado obtenido	Se abren las pantallas correspondientes con los datos cargados

Gestionar Ajustes

Prueba	PA01
Descripción	Cambiar el Email/Nombre/Teléfono
Casos de uso asociados	Actualizar Cuenta
Condiciones iniciales	Usuario Logueado y Servidor Activo

Resultado obtenido	-Si los datos no son válidos, se muestra un mensaje de error correspondiente en cada campo de texto.
---------------------------	--

Prueba	PA02
Descripción	Actualizar la Contraseña
Casos de uso asociados	Cambiar Password
Condiciones iniciales	Usuario Logueado y Servidor Activo
Resultado obtenido	-Si la pass actual es errónea: Se muestra un mensaje de error informativo. -Si la pass nueva es menor de 8 caracteres se muestra un error. -Si las condiciones se cumplen: La pass se modifica correctamente y se muestra mensaje informativo.

Prueba	PA03
Descripción	Eliminar la cuenta actual
Casos de uso asociados	Borrar Cuenta
Condiciones iniciales	Usuario Logueado y Servidor Activo
Resultado obtenido	Se muestra una ventana informando de que la acción es irreversible, si se acepta se elimina la cuenta y se sale de la app.

Gestionar Teoría

Prueba	PT01
Descripción	Abrir listado de libros
Casos de uso asociados	Ver Teoría/ Ver PDF Teoría
Condiciones iniciales	Sin conexión a internet Servidor inactivo
Resultado obtenido	-Se muestra mensaje que debe activarse el internet. Se vuelve a la página Ver Teoría sin cargarse la lista de los pdf's. -Sin servidor activo, pero con internet, los pdf's no se cargan, solo aparece un círculo típico de "cargando..." que seguirá girando indefinidamente hasta que el servidor esté activo.

Prueba	PT02
Descripción	Refrescar lista y abrir documento
Casos de uso asociados	Ver PDF Teoría
Condiciones iniciales	Conexión a internet y servidor activo
Resultado obtenido	La lista se actualiza con el swipe si se han añadido nuevos datos a la BD del servidor. El documento se abre correctamente en el navegador predeterminado del dispositivo. -Si el libro ya no está disponible en la web por la URL, pero sigue en la BD, se mostrará un mensaje diciendo que el libro ya no está disponible.

Gestionar Ejercicios

Prueba	PE01
Descripción	Abrir listado de ejercicios
Casos de uso asociados	Ver Ejercicios/ Ver PDF Ejercicio
Condiciones iniciales	Sin conexión a internet Servidor inactivo
Resultado obtenido	-Se muestra mensaje que debe activarse el internet. Se vuelve a la página Ver Ejercicios sin cargarse la lista de los pdf's. -Sin servidor activo, pero con internet, los pdf's no se cargan, solo aparece un círculo típico de "cargando..." que seguirá girando indefinidamente hasta que el servidor esté activo.

Prueba	PE02
Descripción	Refrescar lista y abrir documento
Casos de uso asociados	Ver PDF Ejercicio
Condiciones iniciales	Conexión a internet y servidor activo
Resultado obtenido	La lista se actualiza con el swipe si se han añadido nuevos datos a la BD del servidor. El documento se abre correctamente en el navegador predeterminado del dispositivo. -Si el libro ya no está disponible en la web por la URL, pero sigue en la BD, se mostrará un mensaje diciendo que el libro ya no está disponible.

Gestionar Tutoriales Video

Prueba	PT01
Descripción	Abrir listado de ejercicios
Casos de uso asociados	Ver Cursos Video
Condiciones iniciales	Sin conexión a internet Servidor inactivo
Resultado obtenido	-Se muestra mensaje que debe activarse el internet. Se vuelve al Menú Principal sin cargarse la lista de los videos. -Sin servidor activo, pero con internet, los videos no se cargan, solo aparece un círculo típico de "cargando..." que seguirá girando indefinidamente hasta que el servidor esté activo.

Prueba	PT02
Descripción	Refrescar lista y abrir documento
Casos de uso asociados	Ver Video
Condiciones iniciales	Enlace caído
Resultado obtenido	Mensaje: "Página web no disponible."

Prueba	PT03
Descripción	Ver video en pantalla completa
Casos de uso asociados	Ver Video
Condiciones iniciales	Pantalla en posición vertical / horizontal
Resultado obtenido	-Vertical: El video se ve en pantalla completa, pero con un gran marco negro, ajustando el video. -Horizontal: El video se ve correctamente en la pantalla completa según el tamaño del dispositivo.

Gestionar Foro

Prueba	PF01
Descripción	Ver / enviar mensaje
Casos de uso asociados	Acceder Foro
Condiciones iniciales	Servidor inactivo / Sin internet
Resultado obtenido	No se cargan los mensajes ni se permite enviar mensaje.

Prueba	PF02
Descripción	Enviar mensaje
Casos de uso asociados	Acceder Foro
Condiciones iniciales	Servidor activo / Con internet
Resultado obtenido	El mensaje se carga en la lista de mensajes con los datos del usuario que lo envía. Solo se permiten como máximo 150 caracteres. La lista de desplazará hacia el último mensaje introducido. Al desplegarse el teclado, toda la lista se desplaza con él hacia arriba, para evitar ocultar los mensajes.

8 IMPLEMENTACIÓN

A continuación, se indican las acciones tomadas durante la fase de implementación y las técnicas que se han realizado.

8.1 IMPLEMENTACIÓN DE LA BBDD

La BBDD tiene 6 tablas que se utilizan para almacenar los datos de la aplicación. A esta se puede acceder desde el servidor de apache, con el hosting proporcionado por noip.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> amigos	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> foro	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> libro_ejercicios	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	37	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> libro_teoría	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	34	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> tuto_video	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	40	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> usuario	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	16.0 KB	-
6 tablas	Número de filas	130	InnoDB	utf8mb4_general_ci	144.0 KB	0 B

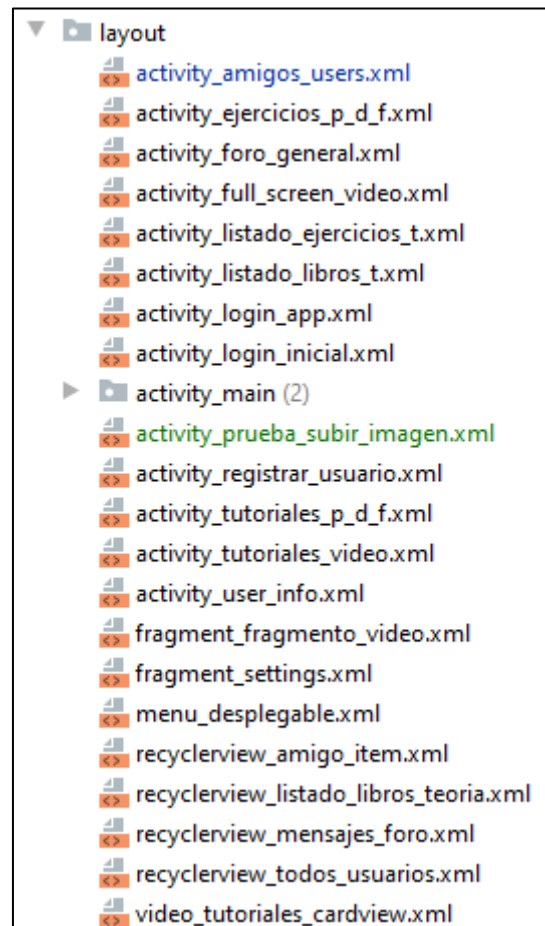
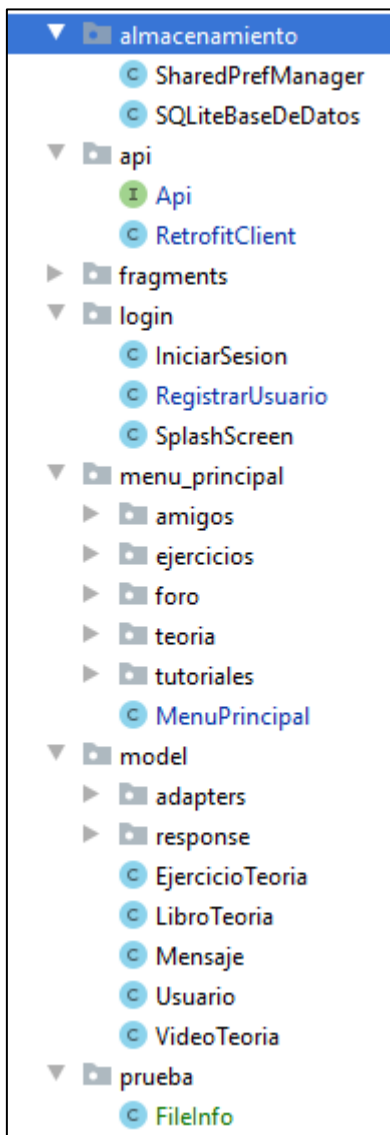
8.2 IMPLEMENTACIÓN DE LA APLICACIÓN MOVIL

Todo el proyecto se desarrolla en base al modelo de desarrollo software por capas, capa presentación, capa de negocio y capa de datos. Se ha intentado que sea lo más optimizado posible, siguiendo los consejos aportados por desarrolladores, en foros y en páginas web.

8.2.1 ARBOL DEL PROYECTO

La estructura del proyecto se divide en dos, una parte del código implementado para la capa de negocio y otra parte para la parte de presentación.

La parte de negocio son archivos .java y la de presentación son .xml:



El archivo AndroidManifest.xml proporciona los permisos de acceso a internet, a la cámara, al almacenamiento, etc, del dispositivo.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.academycode">

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission
        android:name="android.hardware.Camera"
        android:required="true" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

</manifest>
```

8.3 COMUNICACIÓN CON LOS SERVICIOS WEB

Para acceder al servidor y a la BBDD se deben proporcionar unos datos característicos de estos. Para la API REST se proporcionan códigos de respuesta para los métodos que esta implementa.

```
<?php

define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'academycodebd');

define('USER_CREATED', 101);
define('USER_EXISTS', 102);
define('USER_FAILURE', 103);

define('USER_AUTHENTICATED', 201);
define('USER_NOT_FOUND', 202);
define('USER_PASSWORD_DO_NOT_MATCH', 203);

define('PASSWORD_CHANGED', 301);
define('PASSWORD_DO_NOT_MATCH', 302);
define('PASSWORD_NOT_CHANGED', 303);
```

Toda la API se desarrolla en php.

Algunos de los métodos que utiliza la API para gestionar las llamadas y respuestas de la aplicación a los servicios web:

```
//CREAR USUARIO
$app->post('/createuser', function(Request $request, Response $response){ ...
});

//LOGIN DEL USUARIO
$app->post('/userlogin', function(Request $request, Response $response){ ...
});

//Hacer un UPDATE DATOS USUARIO
$app->put('/updateuser/{id_usuario}', function(Request $request, Response $response, array $args){ ...
});

//HACER UPDATE PASSWORD
$app->put('/updatepassword', function(Request $request, Response $response){ ...
});

//COMPROBAR SI EXISTE USUARIO
$app->get('/existuser/{email}', function(Request $request, Response $response, array $args){ ...
});

//BORRAR USUARIO
$app->delete('/deleteuser/{id_usuario}', function(Request $request, Response $response, array $args){ ...
});

//Recoger usuarios
$app->get('/librosteoria/{tematica}', function(Request $request, Response $response, array $args){ ...
});
```

8.4 APLICACIONES DE TERCEROS

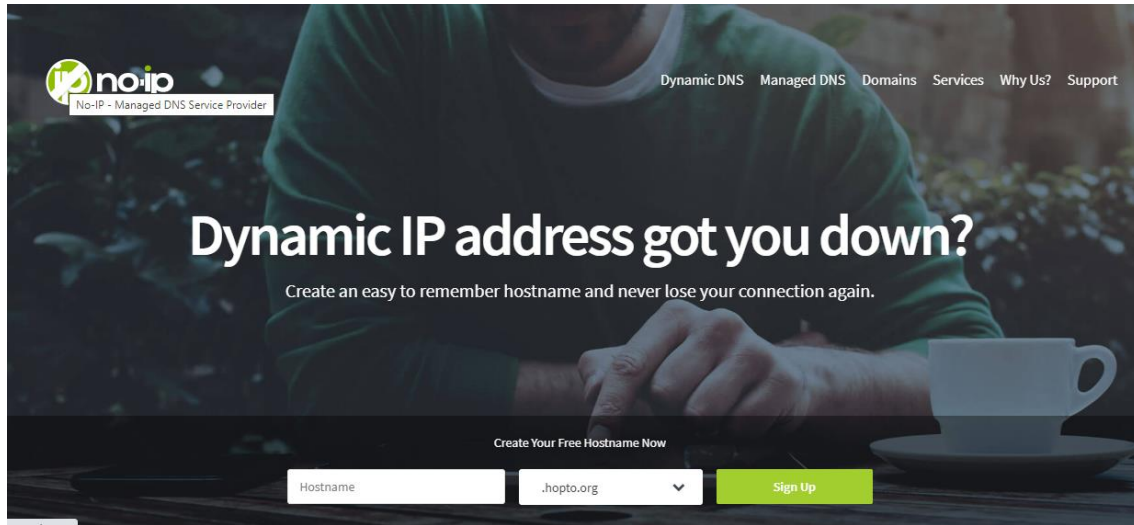
IMGBB

Página web que permite guardar imágenes, en el proyecto se utiliza para las portadas de los libros y ejercicios.



NOIP

Servicio de hosting gratuito, utilizado para acceder al servidor desde dispositivos fuera de la red local.



9 CONCLUSIONES Y MEJORAS

El trabajo realizado en este proyecto ha sido bastante laborioso desde mi punto de vista, ante unos conocimientos básicos he tenido que investigar mucho por mi propia cuenta y aparte de lo que he aprendido durante el curso presencial y las herramientas utilizadas he tenido que aprender y utilizar otras.

Lo bueno de todo esto es que lo aprendido no lo voy a olvidar y que me ha servido para ver que soy capaz de adaptarme a las nuevas tecnologías y utilizarlas según mis necesidades. Además, el proyecto quedará como constancia de mis conocimientos ya que será accesible desde el repositorio de GitHub, pudiendo proporcionarlo a cualquier persona, en especial a las empresas en donde busque que me contraten.

En mi opinión, creo que he llegado a completar todos los objetivos propuestos en la planificación. Hubo requisitos más complicados de cumplir, como los servicios y procesos y otros más sencillos como el diseño de la interfaz.

Las mejoras que quisiera añadir para tener una aplicación más completa son, entre otros, un compilador de los distintos lenguajes de programación que trata la academia, para poder realizar los ejercicios desde el dispositivo móvil sin necesidad de un ordenador. Además, incluiría más funcionalidades en cuanto a la interacción entre usuarios, más libros de teoría y ejercicios y que estos estén categorizados según su nivel de dificultad.

Otras mejoras más básicas que incluiría serían:

- Notificaciones cada vez que se añade un nuevo libro de teoría o de ejercicios.
- Permitir enviar imágenes o documentos por el chat.
- Valoraciones de los cursos.
- Suscripción a cursos o inscripción en la academia.

10 BIBLIOGRAFÍA

<https://stackoverflow.com/>

<https://github.com/>

<https://dbdiagram.io/home>

<https://openwebinars.net/>

<https://www.flaticon.es/>

<https://imgbb.com/>

<https://app.creately.com/manage/project/home>

<https://my.noip.com/#!/dynamic-dns>

<https://www.youtube.com/>

<https://material.io/>

<https://www.simplifiedios.net/php-rest-api-example/>

<https://htmlcolorcodes.com/es/>

<https://docs.oracle.com/javase/7/docs/api/>

<https://developer.android.com/>