Título del proyecto: PassLive

Autor: Javier García Pérez

Profesor: Mario Gago

Semestre: 2S2223

Dirección de entrega:

https://drive.google.com/drive/folders/1AtBZIcmy-rBsTSrJ9XY2yDreaiUCfY54?usp=share_link

Índice

Índice de contenidos

1.Introducción

1.1 Motivación

La seguridad en línea es un tema cada vez más relevante y urgente en la actualidad. Con el aumento de la cantidad de información personal y financiera que se almacena en línea, la necesidad de protegerla se ha vuelto crítica. La gran mayoría de los usuarios crea contraseñas simples y fáciles de recordar para sus cuentas en línea, pero esto los hace vulnerables a ataques de hackers, quienes pueden descifrarlas fácilmente y acceder a la información privada.

Además, la cantidad de sitios web y aplicaciones que requieren contraseñas sigue aumentando, lo que resulta en una dificultad para recordar todas las contraseñas para múltiples cuentas. Esto aumenta el riesgo incluso de violaciones de seguridad, ya que los usuarios pueden optar por utilizar la misma contraseña para varias cuentas o escribirlas en lugares poco seguros, como una nota en su escritorio.

A pesar de que ya existen aplicaciones de administración de contraseñas, muchas de ellas pueden resultar complicadas para los usuarios menos experimentados. Las aplicaciones más avanzadas también pueden requerir conocimientos técnicos o habilidades de seguridad cibernética para configurar y utilizar adecuadamente, lo que puede disuadir a los usuarios de adoptar una aplicación de administración de contraseñas.

Por otro lado, la necesidad de acceder a información financiera en línea también es cada vez mayor. Con la popularidad de las compras en línea y la gestión de finanzas personales, los usuarios necesitan una forma segura de acceder a su información bancaria y de tarjetas de crédito en línea.

En este contexto, el desarrollo de una aplicación móvil en Android de administración de contraseñas fácil de usar, segura y que permita a los usuarios acceder a su información financiera puede ayudar a abordar estos problemas. Al

proporcionar una forma segura y fácil de almacenar y acceder a contraseñas y datos financieros, los usuarios pueden aumentar su seguridad y comodidad, a la vez de proteger su información privada. Además, una aplicación de este tipo también puede ayudar a disminuir el riesgo de violaciones de seguridad y ataques de hackers.

1.2 Abstract

Online security and password management have become increasingly important topics in today's digital age. With the amount of personal and financial information being stored online, the need to protect this information from cyber threats has become more critical than ever. Unfortunately, users often have difficulty remembering multiple complex passwords, which can lead to the use of weak or repeated passwords that are easy for hackers to guess or crack. This can result in compromised accounts and stolen personal information.

To address this problem, the proposed project aims to develop a mobile password management application that is easy to use and allows users to store different passwords for each application. By using a password manager, users can generate strong and unique passwords for each account and keep them safe in a secure location. This approach can significantly improve the security of online accounts and reduce the risk of cyber threats.

In addition to password management, the app will also allow users to store and access their financial information, such as bank account and credit card details. This will provide users with a convenient and secure way to manage their finances online, without the need to remember multiple login credentials or repeatedly enter sensitive information. The application will be developed using the Android Studio environment, which is a popular and widely used tool for mobile application development.

To ensure the security of user data, the application relies on advanced algorithms and other security measures. The user interface will also be designed

to be intuitive and easy to use, with a focus on simplicity and ease of use. This will make it easy for users of all skill levels to use the app and manage their passwords and financial information securely.

Overall, the proposed project aims to provide a comprehensive solution that addresses the growing need for online security and password management. By developing a mobile app that is easy to use and securely store and access passwords and financial information, the project can help users protect their personal information and reduce the risk of cyberthreats.

1.3. Objetivos propuestos (generales y específicos)

- **Objetivos generales:**

- Desarrollar una aplicación móvil para la gestión de contraseñas y el almacenamiento de información financiera que sea fácil de usar, segura y amigable para el usuario. El proyecto también se centrará en desarrollar una aplicación que sea altamente personalizable para satisfacer las necesidades específicas de cada usuario. Los usuarios podrán ajustar sus preferencias y personalizar la aplicación según sus propias necesidades.

- **Objetivos específicos:**

- Realice un análisis exhaustivo de las aplicaciones existentes en el mercado para la gestión de contraseñas y el almacenamiento de información financiera, identificando las deficiencias y limitaciones en términos de seguridad y usabilidad.
- Diseñar y desarrollar una aplicación móvil utilizando el entorno de Android Studio que aborde las deficiencias y limitaciones identificadas y proporciones una solución segura y fácil de usar para la gestión de contraseñas y el almacenamiento de información financiera.
- Diseñar una interfaz intuitiva y amigable para el usuario que sea fácil de navegar y proporcione una experiencia de usuario fluida.

- Desarrollar un sistema de encriptación de la base de datos que sea lo más seguro posible ante posible vulneraciones y ataques.
- Realizar pruebas de depuración exhaustivas para garantizar la estabilidad y confianza de la aplicación.
- Proporcionar mantenimiento, soporte y actualizaciones continuas para garantizar la usabilidad y seguridad continua de la aplicación.

Al lograr estos objetivos, el proyecto tiene como objetivo proporcionar una solución que aborde la creciente necesidad de seguridad en línea y gestión de contraseñas, y proporción de una manera.

2.Metodología empleada

Para el desarrollo de la aplicación, se ha optado por utilizar el modelo en cascada con retroalimentación. Es un modelo secuencial y lineal que se utiliza para el desarrollo de software, donde cada fase depende del éxito de la anterior y los resultados de una fase se entregan a la siguiente fase como entrada. Este modelo es adecuado para proyectos donde los requisitos son claros y estables, ya que es difícil realizar cambios importantes una vez que se ha completado una fase.

Por otro lado, la retroalimentación es un proceso que se utiliza para mejorar el rendimiento de un proyecto. En el modelo en cascada con retroalimentación, se integra una fase de retroalimentación después de la implementación y las pruebas. Esto permite identificar problemas en la aplicación y ajustar las especificaciones del proyecto en caso de ser necesario.

Ciclo de vida del proyecto:

1. **Definición de requisitos:**

En esta fase se identificarán los requisitos del proyecto, lo que incluye sus necesidades y expectativas Se debe definir el alcance del proyecto y los objetivos que se quieren lograr.

2. **Análisis y diseño**

En la fase de diseño, se llevará a cabo la definición de la arquitectura de la aplicación y la definición de las funcionalidades que se implementarán. El diseño de la aplicación utiliza la definición de la interfaz de usuario, la arquitectura de la aplicación, la definición de la base de datos y la definición de la seguridad de la aplicación.

3. **Implementación y prueba de unidades**

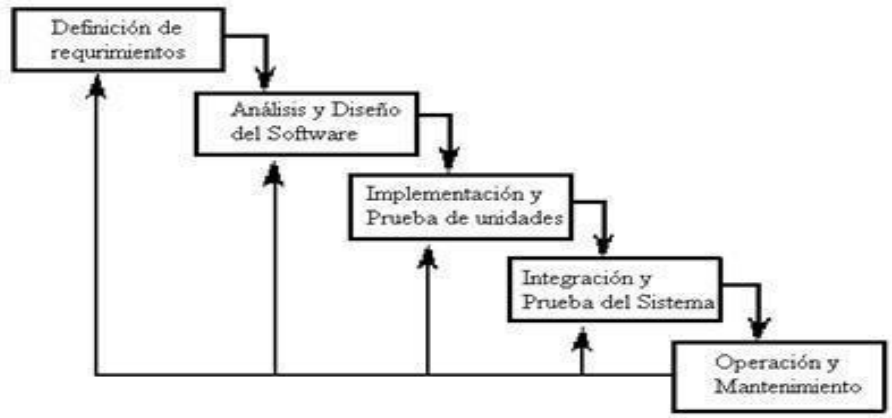
La fase de implementación es donde se codifica la aplicación utilizando las especificaciones definidas en la fase de diseño. Esta fase resultó ser la codificación de los módulos de software, la creación de la base de datos, la integración de los diferentes componentes de la aplicación y la implementación de los diferentes algoritmos y lógica de la aplicación.

4. Integración y pruebas del sistema

La fase de pruebas es donde se evalúa la aplicación para asegurarse de que cumple con los requisitos establecidos. Esta fase demostró pruebas de funcionalidad, pruebas de rendimiento y pruebas de seguridad. Es importante que se realicen pruebas exhaustivas para identificar posibles errores o problemas en la aplicación.

5. Operación y mantenimiento

La fase de mantenimiento es donde se realizan las correcciones de errores y se realizan mejoras adicionales en la aplicación. Esta fase se llevará a cabo después del lanzamiento de la aplicación y durante todo el ciclo de vida de la misma. Es importante asegurarse de que la aplicación esté actualizada y sea segura para los usuarios.



Después de la fase de implementación, se llevará a cabo una evaluación de la aplicación para identificar posibles problemas y errores. En caso de que se identifiquen

problemas, se retroalimentará el proceso, regresando a la fase de diseño para ajustar las especificaciones y volver a la fase de implementación. En resumen, el proyecto de final de desarrollo de aplicaciones multiplataforma basado en una aplicación desarrollada para el sistema operativo de Android en la que su función principal es recordar contraseñas y datos importantes para la identificación en los registros de aplicaciones o páginas web, de cuentas bancarias y tarjetas bancarias, se desarrollará utilizando el modelo en cascada con retroalimentación, y constará de las fases de requisitos, análisis y diseño, implementación, integración y mantenimiento.

Fases del diseño:

En la fase de diseño se llevarán a cabo las siguientes fases:

- Diseño de la interfaz de usuario: Se definirá la apariencia visual de la aplicación y la forma en que el usuario interactuará con ella consiguiendo un buen nivel de usabilidad con la interfaz desarrollando un diseño estético a la vez que minimalista.
- Diseño de la arquitectura: Se definirá la estructura y los componentes de la aplicación.
- Diseño de la base de datos: Se definirá la estructura de la base de datos que se utilizará para almacenar las contraseñas y la información de los usuarios.
- Diseño de la seguridad: Se definirán las medidas de seguridad que se utilizarán para proteger la información de los usuarios, tomando especial atención en la encriptación de la base de datos con información muy sensible.
- Diseño de las funcionalidades: Se definirán las funcionalidades de la aplicación, como la posibilidad de almacenar contraseñas, acceder a cuentas de aplicaciones y páginas web, cuentas bancarias y tarjetas de crédito, y generar contraseñas seguras.

3.Tecnologías y herramientas utilizadas en el proyecto

En este proyecto se han utilizado tecnologías y herramientas de vanguardia para garantizar la seguridad y el rendimiento de la aplicación. En este contexto, es importante destacar la importancia de una metodología de desarrollo adecuada para garantizar el éxito del proyecto y satisfacer las necesidades del usuario final.

Entre las tecnologías y herramientas utilizadas se encuentran:

- **Java** como lenguaje de programación: Java es un lenguaje de programación multiplataforma orientado a objetos, es compilado aún lenguaje intermedio y después interpretado, puede ser utilizado para desarrollar aplicaciones para Android. Java es una elección popular entre los desarrolladores de Android debido a su facilidad de uso, seguridad y eficiencia.

- **Android Studio** como IDE: Android Studio es el IDE oficial para el desarrollo de aplicaciones Android. Es una herramienta poderosa que proporciona un ambiente de desarrollo integrado completo para diseñar, desarrollar, depurar y probar aplicaciones Android.

- **Sistema de creación de diseño nativo de Android (XML):** El sistema de diseño nativo de Android es una herramienta de diseño que permite a los desarrolladores interfaces de usuario para sus aplicaciones utilizando un archivo XML.

- **GitHub:** GitHub es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso.

- **SQLite:** Es una biblioteca de software que proporciona un sistema de gestión de base de datos relacionales local en el dispositivo del usuario para almacenar y acceder a la información. También podremos realizar configuraciones de usuario, registros de actividad, contenido sin conexión que es ideal para desarrollar el proyecto, ya que ofrece una solución de bases de datos liviana, fácil de implementar y administrar.

- **SQLCipher** y **Android Keystore** como sistemas de encriptación de datos: SQLCipher es una biblioteca de cifrado de base de datos que proporciona seguridad adicional a los datos almacenados en la base de datos. Android Keystore es una herramienta de seguridad que se utiliza para almacenar y proteger las claves de cifrado utilizadas por la aplicación.

- **LottieFiles** para las animaciones: Es un formato de archivo de animación basado en JSON que le permite enviar animaciones a cualquier plataforma de forma tan sencilla como enviar activos estáticos. Son archivos pequeños que funcionan en cualquier dispositivo y pueden agrandarse o reducirse sin

pixelación. LottieFiles permite crear, editar, probar, colaborar y enviar un Lottie de la forma más sencilla posible.

- **Miro** para la creación de esquemas y diseños de maquetas: Miro es una herramienta de colaboración en línea que se utiliza para crear y compartir

esquemas y diseños de maquetas. Es una herramienta útil para la comunicación y colaboración entre los miembros del equipo de desarrollo.

- **Lucidchart** es una plataforma de diagramación que permite trabajar en un documento con otros usuarios al mismo tiempo, en pocas palabras, un espacio digital para crear en equipo sin importar el lugar donde se encuentren.
  A diferencia de los tableros virtuales, esta plataforma cuenta con elementos especiales para la creación de organigramas, mapas mentales, diagramas de flujo, entre otros formatos para ver cómo se relacionan y conectan los conceptos.

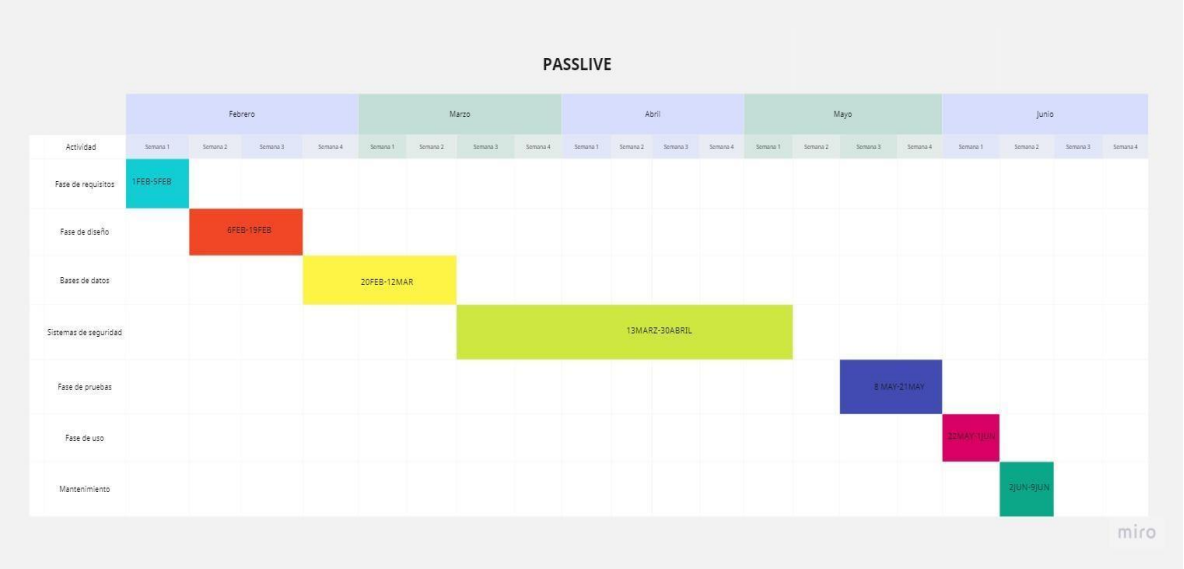
En conclusión, la selección de estas tecnologías y herramientas se basó en su potencia, eficacia y facilidad de uso. Cada una de estas herramientas y tecnologías ha sido ampliamente utilizada en el desarrollo de aplicaciones móviles y ha demostrado ser confiable y eficiente. Al utilizar estas tecnologías y herramientas, se logró crear una aplicación de alta calidad que cumple con los requisitos del usuario final.

4.Estimación de recursos y planificación

El proyecto se ha dividido en seis fases principales, cada una con sus tareas específicas. La duración total estimada del proyecto es de 17 semanas, aunque el tiempo real empleado ha sido de 18 semanas.

En la fase de requisitos, se ha dedicado una semana para definir los objetivos y especificaciones del proyecto. En la fase de diseño, se han dedicado dos semanas para diseñar las vistas de la aplicación, la base de datos y los sistemas de seguridad. La fase de implementación ha sido la más larga, con una duración estimada de diez semanas y una duración real de once semanas, y ha incluido la codificación, las pruebas unitarias y la integración de todos los componentes del sistema.

La fase de pruebas ha durado dos semanas y se ha centrado en probar el rendimiento y la seguridad de la aplicación. La fase de uso ha durado una semana y ha incluido la creación de paquetes de instalación. Por último, se ha dedicado una semana a la fase de mantenimiento para garantizar el correcto funcionamiento de la aplicación y solucionar cualquier problema que surja.



A pesar de que el tiempo real empleado ha sido superior al estimado, la aplicación ha sido desarrollada con éxito y cumple con los objetivos y especificaciones definidas

al inicio del proyecto. El uso del diagrama de Gantt ha permitido una planificación y seguimiento eficaz del proyecto, lo que ha contribuido a su éxito.

5.Análisis

La aplicación tiene como objetivo ofrecer a los usuarios una forma fácil de almacenar y recordar sus contraseñas de forma segura, sin la necesidad de memorizarlas todas. Además, la aplicación debe garantizar la seguridad de los datos almacenados, lo que implica la encriptación de los datos y la prevención de accesos no autorizados.

En términos de necesidades, la aplicación debe ser fácil de usar, permitiendo a los usuarios acceder y administrar sus contraseñas de forma rápida y sencilla. Debe ser compatible con múltiples dispositivos, incluyendo dispositivos móviles y tablet. También debe permitir a los usuarios generar contraseñas seguras y únicas para cada servicio y aplicación.

Además, con la creciente amenaza de hackers y ciberdelincuentes, es fundamental que las personas utilicen contraseñas seguras y únicas para cada servicio y aplicación en línea. La tarea de recordar y administrar estas contraseñas puede ser abrumadora y, en muchos casos, resulta en la reutilización de contraseñas inseguras o en la utilización de contraseñas demasiado simples.

Por esta razón, una aplicación que ofrece una forma segura y conveniente de almacenar y recordar contraseñas se vuelve esencial para la seguridad en línea. Además, la aplicación debe garantizar la privacidad y seguridad de los datos almacenados, lo que significa que la encriptación y el uso de medidas de seguridad adicionales, como el uso de Android Keystore y SQLCipher, son fundamentales.

En cuanto a la facilidad de uso, la aplicación debe ser intuitiva y fácil de usar para los usuarios, con una interfaz limpia y organizada. Debe permitir a los usuarios acceder y administrar sus contraseñas de forma rápida y sencilla, lo que implica la capacidad de buscar y filtrar las contraseñas almacenadas.

En resumen, la aplicación para recordar contraseñas se vuelve esencial en la era digital para mejorar la seguridad en línea de las personas y simplificar la tarea de administrar contraseñas múltiples. La seguridad y privacidad de los datos almacenados, así como la facilidad de uso y la capacidad de generar contraseñas seguras, son fundamentales para el éxito de la aplicación.

Los requisitos son una parte fundamental en el proceso de desarrollo de cualquier aplicación, ya que establecen las características y funcionalidades que deben ser incluidas en la misma. En el caso de nuestra aplicación, existen requisitos funcionales y no funcionales que deben ser considerados para garantizar la seguridad, usabilidad y eficiencia de la aplicación. Los requisitos funcionales se enfocan en las funciones específicas que la aplicación debe proporcionar, mientras que los requisitos no funcionales se enfocan en las características que la aplicación debe tener para garantizar su calidad y rendimiento. En este sentido, es importante definir y analizar cuidadosamente los requisitos para asegurar que la aplicación cumpla con las expectativas y necesidades de los usuarios.

- ## Requisitos Funcionales:

- **Registro de usuario**: La aplicación debe permitir a los usuarios registrarse en la plataforma y crear una cuenta personalizada.
- **Almacenamiento de contraseñas**: La aplicación debe permitir a los usuarios almacenar y organizar sus contraseñas de forma segura y encriptada.
- **Búsqueda y filtrado**: La aplicación debe permitir a los usuarios buscar y filtrar sus contraseñas guardadas para encontrar rápidamente la información que necesitan mediante filtrado por categorías, por título ascendente-descendente, por fecha de más reciente y más antigua.
- **Exportación e importación de los registros**: La aplicación en formato puede tanto exportar como importar registros en formato CSV.
- **Re-direccionar a páginas y aplicaciones web**. La aplicación dispondrá de un botón para enlazar con la aplicación o página web deseada, para así acceder de manera más práctica.
- **Tomar fotografías**. La aplicación podrá tomar fotografías y almacenarlas junto a su registro de entrada.
- **Restablecer contraseña**. La aplicación tendrá la función para cambiar la contraseña para acceder a ella.

- **Acceso a la aplicación rápido**. La aplicación dispondrá del acceso por huella dactilar para mejorar y garantizar la seguridad al acceder a ella.
- **Evitar las capturas de pantalla**. La aplicación tendrá la habilitación para evitar las capturas de pantalla por motivos de seguridad.

- ## Requisitos no funcionales:

- **Seguridad:** La aplicación debe ser segura y encriptada para proteger la información personal del usuario.
- **Usabilidad**: La aplicación debe ser fácil de usar y navegar para cualquier tipo de usuario, independientemente de sus habilidades tecnológicas.
- **Rendimiento**: La aplicación debe ser rápida y eficiente para proporcionar una experiencia de usuario fluida y sin interrupciones.
- **Compatibilidad**: La aplicación debe ser compatible con una variedad de dispositivos Android y versiones del sistema operativo.
- **Mantenimiento**: La aplicación debe ser fácil de mantener y actualizar para corregir errores y agregar nuevas características.

Diagramas de Entidad-Relación

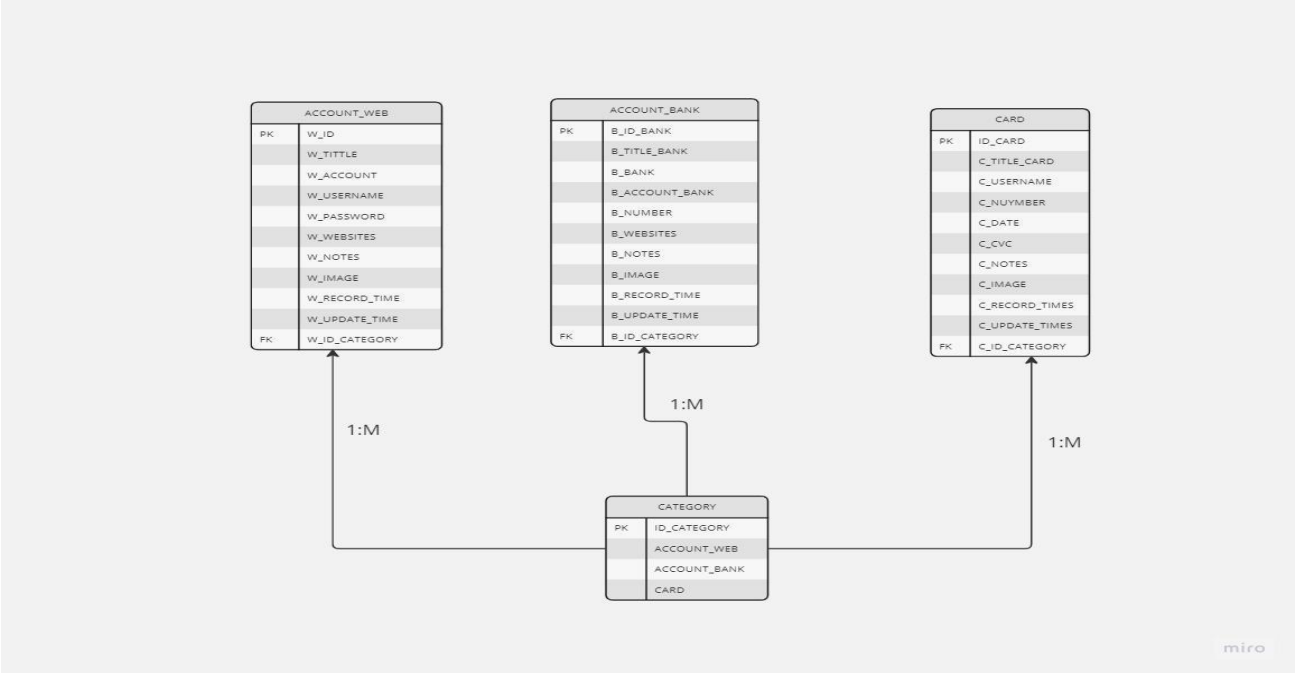
La aplicación la compone una base de datos relacional compuesta por tres tablas. La tabla **"CATEGORY"** es la tabla principal que tiene una relación de uno a muchos con las tablas **"ACCOUNT_WEB", "ACCOUNT_BANK" y "CARD"**. Esta relación indica que una categoría puede tener varios registros en las otras tablas, pero a la vez, cada registro solo puede pertenecer a una categoría única. Esta relación se establece mediante el uso de una clave foránea en las tablas secundarias que hace referencia al ID de la categoría en la tabla principal.

La tabla **"ACCOUNT_WEB"** almacena información relacionada con título, cuentas, nombres de usuario, contraseña, sitio web, notas, imágenes, tiempo en que se realizó el registro y tiempo en que se actualizó el registro.

La tabla **"ACCOUNT_BANK"** almacena información relacionada con cuentas bancarias, como títulos de cuenta, nombres de banco, números de cuenta, enlaces a sitios web de bancos y notas. Además, esta tabla también tiene una relación de muchos a uno con la tabla "CATEGORY".

La tabla **"CARD"** almacena información relacionada con tarjetas de crédito o débito, como el título de la tarjeta, el número

En resumen, la estructura de la base de datos permite al usuario almacenar de manera organizada y segura información relacionada con diferentes tipos de cuentas y tarjetas, y además, permite categorizarlos para un fácil acceso y administración.



Diagramas de Casos de Uso

Registro de usuario:

Descripción: El usuario podrá registrarse en la aplicación.

Actor principal: Usuario.

Flujo básico:

- El usuario ingresa una contraseña y confirma la misma contraseña.
- La aplicación valida los datos ingresados y crea una cuenta de usuario.
- La aplicación muestra un mensaje de confirmación y redirige al usuario a la pantalla de inicio de sesión.

Flujo alternativo: Si los datos ingresados no son válidos, la aplicación muestra un mensaje de error y solicita al usuario que ingrese datos válidos.

Iniciar sesión:

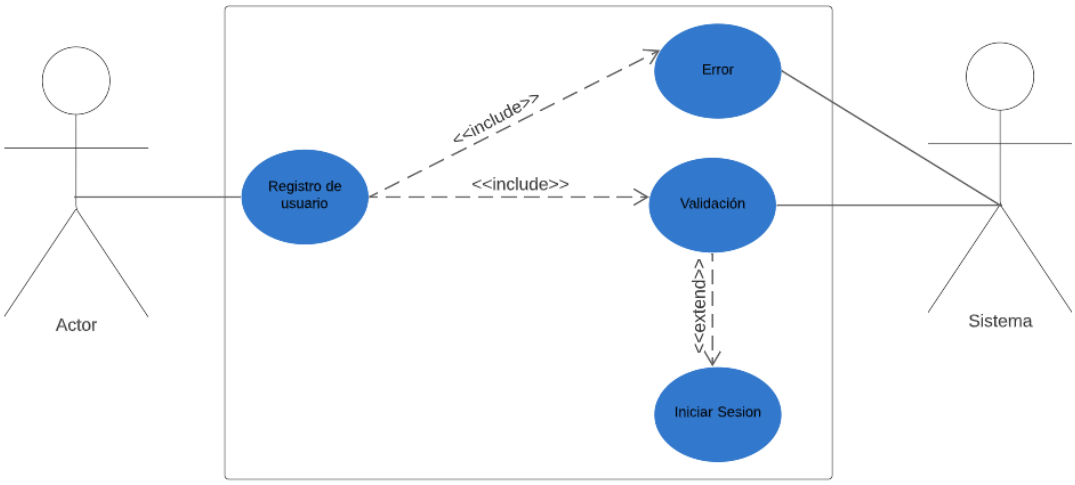
Descripción: El usuario podrá iniciar sesión en la aplicación.

Actor principal: Usuario.

Flujo básico:

- El usuario ingresa su nombre de usuario y contraseña.
- La aplicación valida los datos ingresados y autentica al usuario.
- La aplicación redirige al usuario a la pantalla principal de la aplicación.

Flujo alternativo: Si los datos ingresados no son válidos, la aplicación muestra un mensaje de error y solicita al usuario que ingrese datos válidos.



Añadir cuenta web:

Descripción: El usuario podrá agregar una cuenta web a la aplicación.

Actor principal: Usuario.

Flujo básico:

- El usuario ingresa la información de la cuenta web (título, nombre de usuario, contraseña, URL, notas, imagen, categoría).
- La aplicación valida los datos ingresados y agrega la cuenta web a la base de datos.
- La aplicación muestra un mensaje de confirmación y redirige al usuario a la pantalla principal de la aplicación.

Flujo alternativo: Si los datos ingresados no son válidos, la aplicación muestra un mensaje de error y solicita al usuario que ingrese datos válidos.

Añadir cuenta bancaria:

Descripción: El usuario podrá agregar una cuenta bancaria a la aplicación.

Actor principal: Usuario.

Flujo básico:

- El usuario ingresa la información de la cuenta bancaria (título, banco, número de cuenta, notas, una imagen).
- La aplicación valida los datos ingresados y agrega la cuenta bancaria a la base de datos.
- La aplicación muestra un mensaje de confirmación y redirige al usuario a la pantalla principal de la aplicación.

Flujo alternativo:

- Si los datos ingresados no son válidos, la aplicación muestra un mensaje de error y solicita al usuario que ingrese datos válidos.

Añadir tarjeta de crédito:

Descripción: El usuario podrá agregar una tarjeta de crédito a la aplicación.

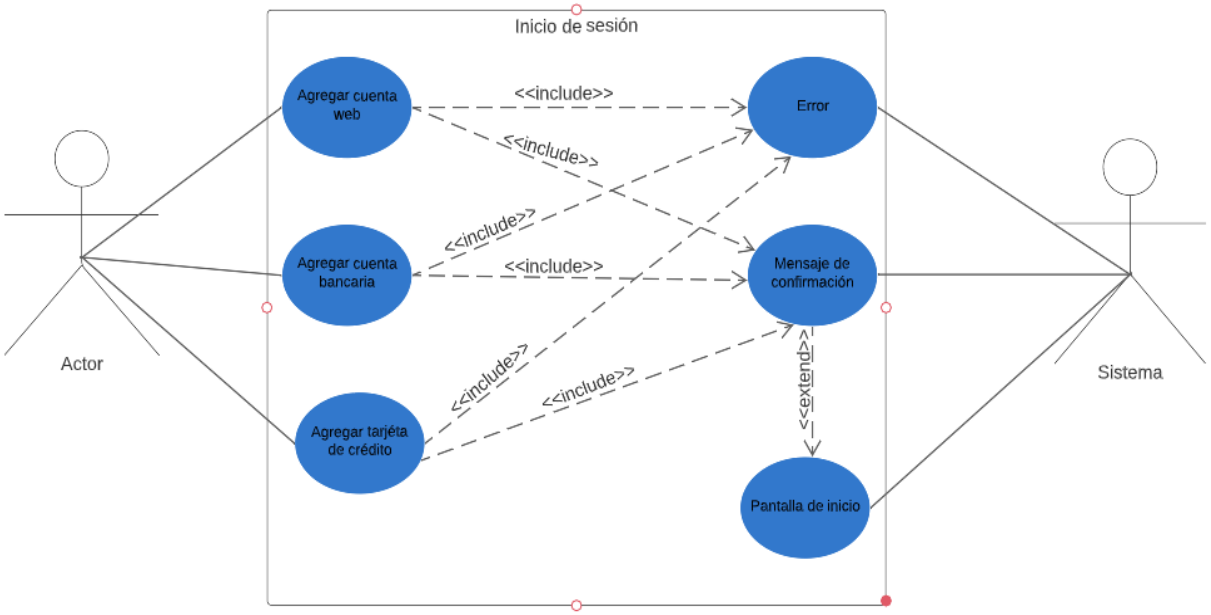
Actor principal: Usuario.

Flujo básico:

- El usuario ingresa la información de la tarjeta de crédito (título, nombre del titular, número de tarjeta, fecha de vencimiento, código CVC, notas, imagen, categoría).
- La aplicación valida los datos ingresados y agrega la tarjeta de crédito a la base de datos.
- La aplicación muestra un mensaje de confirmación y redirige al usuario a la pantalla principal de la aplicación.

Flujo alternativo:

- Si los datos ingresados no son válidos, la aplicación muestra un mensaje de error y solicita al usuario que ingrese datos válidos.



Diagramas de Clases

Para finalizar el proceso de análisis del proyecto, realizamos un diagrama de clases en el que cada una de estas clases se relaciona con la clase Usuario a través de un atributo llamado usuario_id, lo que indica a qué usuario pertenece cada una de las cuentas.

Clase Cuenta Web

Tiene varios atributos importantes, como el título, que es una descripción corta de la cuenta, y el nombre de usuario y la contraseña que se utiliza para iniciar sesión en la cuenta. Además, la clase también tiene un campo para la URL de la cuenta, que es la dirección web donde se encuentra la cuenta, y un campo para notas adicionales que el usuario puede agregar para recordar detalles específicos de la cuenta.

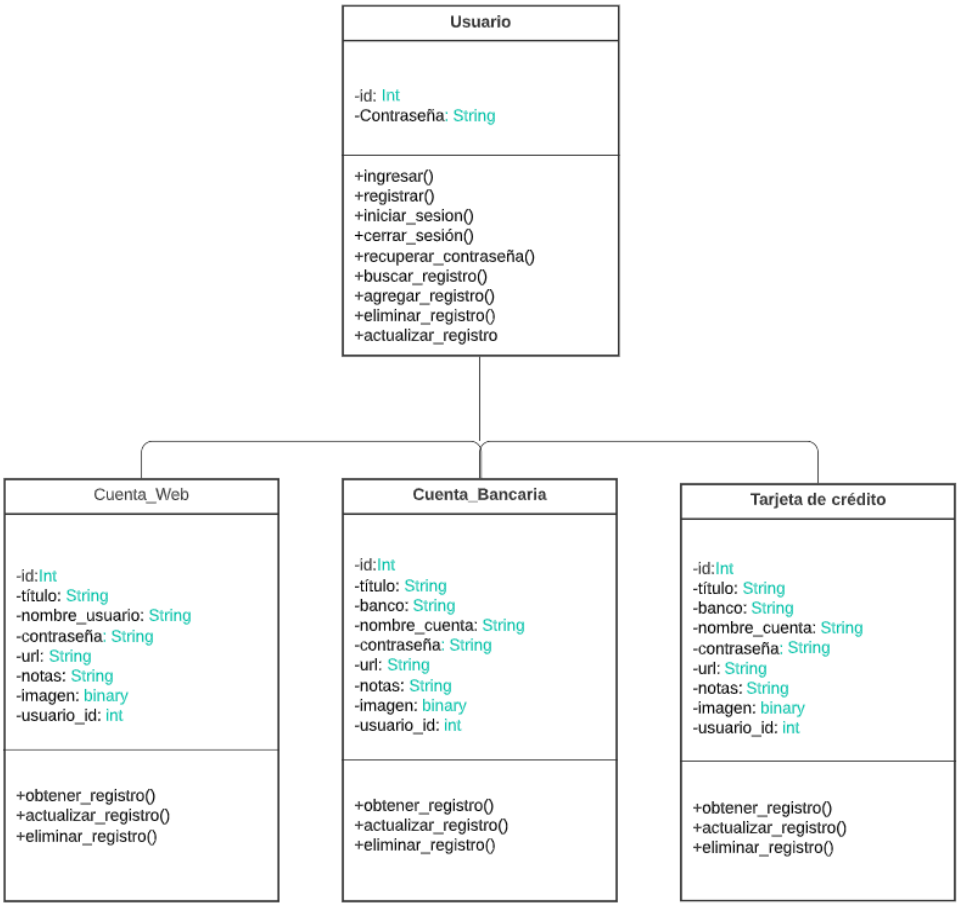
Clase Cuenta Bancaria

También tiene un campo para el banco asociado con la cuenta, el número de cuenta, y un campo para notas adicionales. Estos detalles son importantes para que el usuario pueda mantener un registro de sus cuentas bancarias en un solo lugar.

Clase Tarjeta de Crédito

Tiene atributos adicionales, como el nombre del titular de la tarjeta y la fecha de vencimiento. También tiene un campo para el código CVC, que se utiliza para confirmar la identidad del titular de la tarjeta en ciertas situaciones.

Por lo tanto, este diagrama de clases representa un sistema de administración de cuentas de usuario que permite a los usuarios almacenar y organizar sus cuentas web, bancos y tarjetas de crédito en una sola aplicación. La aplicación está diseñada para ser fácil de usar y proporcionar a los usuarios la seguridad y la comodidad que necesitan para administrar sus cuentas de manera efectiva.



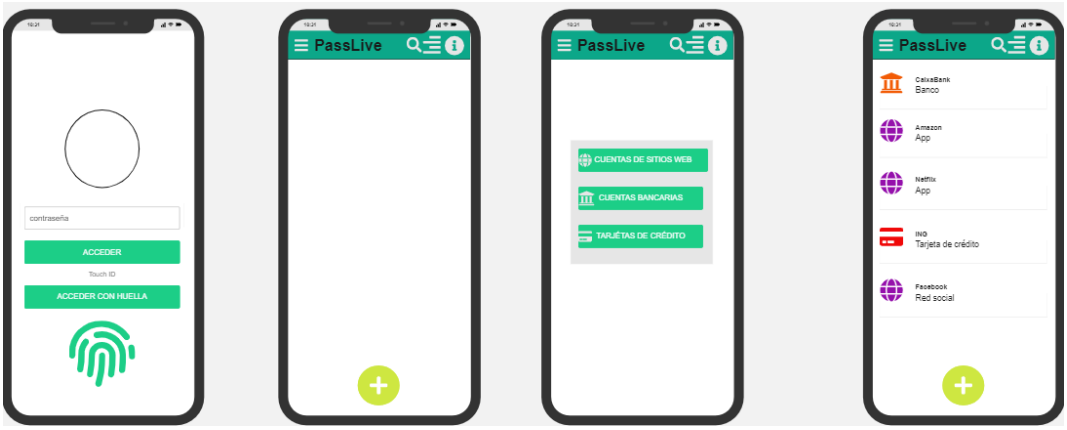
6.Diseño

Para empezar, nos enfocamos en diseñar la interfaz de la aplicación. Utilizamos la herramienta Miro para crear los Wireframes de las diferentes vistas. En primer lugar, diseñamos la pantalla de registro donde se debe ingresar una contraseña y confirmarla. Luego, al iniciar sesión, se accede a la actividad principal y se puede acceder al menú a través de un botón ubicado en la esquina superior izquierda.

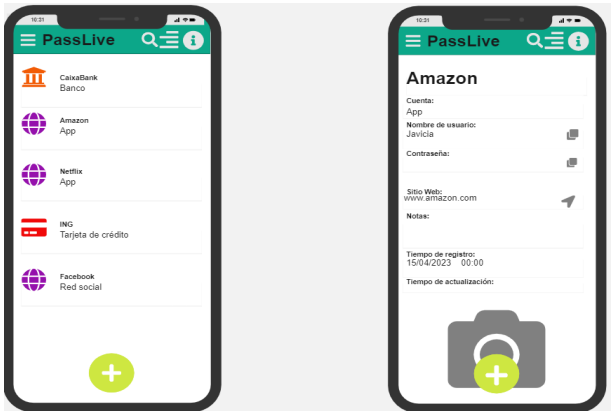
Una vez nos hemos registrado en la aplicación, al acceder a ella se nos muestra la pantalla de inicio de sesión, donde podremos acceder introduciendo nuestra contraseña o mediante el acceso biométrico. Una vez dentro de la actividad principal, podremos acceder a las distintas opciones a través del botón flotante en la parte inferior central, que nos mostrará un cuadro de diálogo con las diferentes categorías disponibles, como cuentas web, cuentas bancarias y tarjetas de crédito.

Al seleccionar una de estas opciones, accederemos a una pantalla de agregar registro correspondiente a dicha categoría. En esta pantalla, podremos introducir los datos necesarios del registro, como el nombre de usuario, la contraseña, etc., y se guardarán en un **RecyclerView** junto con el resto de registros de todas las categorías.

De esta forma, tendremos una lista organizada de todas nuestras cuentas y tarjetas, que podremos consultar, editar u ordenar en cualquier momento.



Después de agregar un registro, tendremos la opción de acceder a él para visualizarlo, copiarlo o editarlo según nuestras necesidades.



Realización del proyecto

El proyecto consiste en una aplicación para gestionar registros de diferentes tipos, como notas, contraseñas y webs. La aplicación cuenta con una base de datos local SQLite para almacenar los registros encriptados, y ofrece diversas funcionalidades como agregar, editar y eliminar registros, así como buscar y filtrar registros.

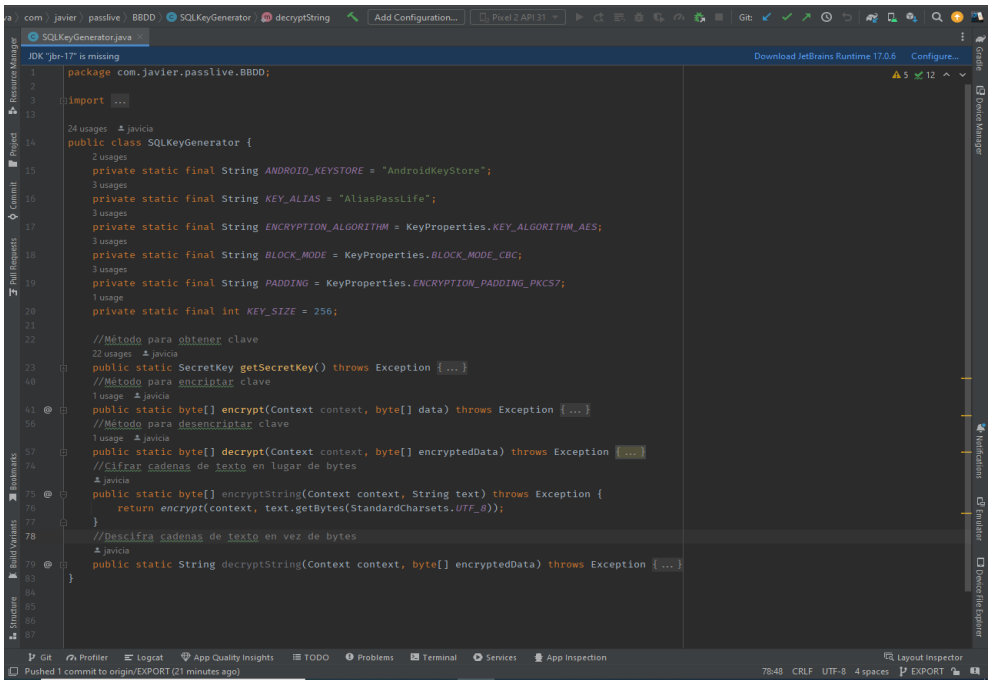
A continuación se detallan las principales clases, funciones y ficheros del proyecto:

Clases:

- **MainActivity:** Es la actividad principal de la aplicación. En ella se muestra una lista de todos los registros almacenados en la base de datos, y se ofrecen diversas opciones para agregar, editar y eliminar registros. También se puede buscar y filtrar registros por diferentes criterios, como tipo, título o fecha de creación.
- **Util -> Util_Bank, Util_Card y Util_Web:** Son actividades que se encargan de agregar o editar registros de cada tipo. Cada actividad cuenta con un formulario donde se pueden ingresar los datos del registro, como título, descripción, fecha de creación, etc.
- **Model -> Record, Bank, Card y Web**: Son clases que representan los diferentes tipos de registros que se pueden almacenar en la base de datos. Cada clase cuenta con atributos que corresponden a los diferentes campos del formulario, como título, descripción, fecha de creación, etc.
- **BBDD ->Helper**: Es una clase que se encarga de crear y actualizar la base de datos local SQLite, y proporciona métodos para obtener una instancia de la base de datos y realizar operaciones en ella.
- **BBDD** -> **SQLkeyGenerator:** Es una clase que proporciona la clave para cifrar la base de datos. Para conseguir una clave segura se utiliza un cifrado AES con un tamaño clave de 256 bits, el modo de bloque CBC y el relleno PKCS7. Los pasos para la primera ejecución son:

- **"getSecretkey()"**.Obtenemos una clave secreta que se utiliza para cifrar y descifrar los datos. Si la clave no existe en el alamacenamiento de claves se creará una nueva clave.
- **"encrypt()".**Este método se utiliza para cifrar los datos utilizando la clave secreta obtenida del método "getSecretKey()"
- **"decrypt()"**. Esta función la utilizamos para descifrar los datos usando la clave secreta obtenida en el método.
- **"encryptString()"** y **"decryptString()".** Estos dos métodos los utilizamos para cifrar cadenas de texto en el lugar de bytes. El método **"encryptString()"**
- toma una de texto como entrada y la cadena convierte en una matriz de bytes antes de descifrarla utilizando el método **"encrypt()".** El método

“decryptString()” toma una matriz de bytes cifrada como entrada y la descifra utilizando el método **“decrypt()”.** Luego convierte los bytes descifrados en una cadena de texto.



- **Adapter-> RecordAdapter, Adapter_bank, Adapter_card y Adapter_web**: Son adaptadores para las listas de registros de cada tipo. Cada adaptador se encarga de mostrar los datos de cada registro en un elemento de la lista, y proporciona opciones para editar o eliminar cada registro.
- **UserRegistration ->Registration:** Esta clase se encarga del registro de usuarios de la aplicación, en primer lugar, se inicializa variables y verifica si la contraseña ya ha sido registrada previamente en la aplicación. Luego, el código establece un listener para el botón de registro y valida si los campos de correo electrónico y contraseña cumplen con los requisitos necesarios antes de guardar los datos en un archivo de preferencias compartidas. Si la política de privacidad no ha sido aceptada, se muestra un diálogo. Si todo está correcto, el código redirige al usuario a la actividad principal de la aplicación.
- **Login ->LoginUser:** Es la clase que se encarga del inicio de sesión cuando el usuario ya esté registrado, en el que incluye un acceso biométrico. La actividad tiene un campo de entrada de texto la contraseña, así como dos botones: uno para iniciar sesión con credenciales y otro para iniciar sesión mediante autenticación biométrica. También utiliza el SharedPreferences para almacenar la contraseña del usuario. Cuando se hace clic en el botón de inicio de sesión con credenciales, el código comprueba si el campo de la contraseña no está

vacío y si coincide con la contraseña almacenada en SharedPreferences. Si las credenciales son válidas, la actividad principal se abre y la actividad actual se cierra. Cuando se hace clic en el botón de inicio de sesión biométrico, aparece un diálogo de autenticación biométrica que solicita al usuario que escanee su huella digital para iniciar sesión. Si la autenticación es exitosa, la actividad principal se abre y la actividad actual se cierra.

Funciones:

onCreate(): Es una función que se ejecuta al crear la actividad principal MainActivity. En ella se inicializa la vista de la lista de registros y se configuran los botones y opciones para agregar, editar y eliminar registros.

updateRecord(): Función que se utiliza para actualizar los registros de contraseñas o notas en la base de datos.

loadRecord(). Es una función que se encarga de cargar los registros correspondientes en la lista. Esta funcion se llaman al abrir MainActivity, al agregar o eliminar un registro, o al aplicar un filtro de búsqueda.

saveRecord(): Función que se utiliza para guardar los registros de contraseñas o notas en la base de datos.

deleteRecord(): Función que se utiliza para eliminar los registros de contraseñas o notas de la base de datos.

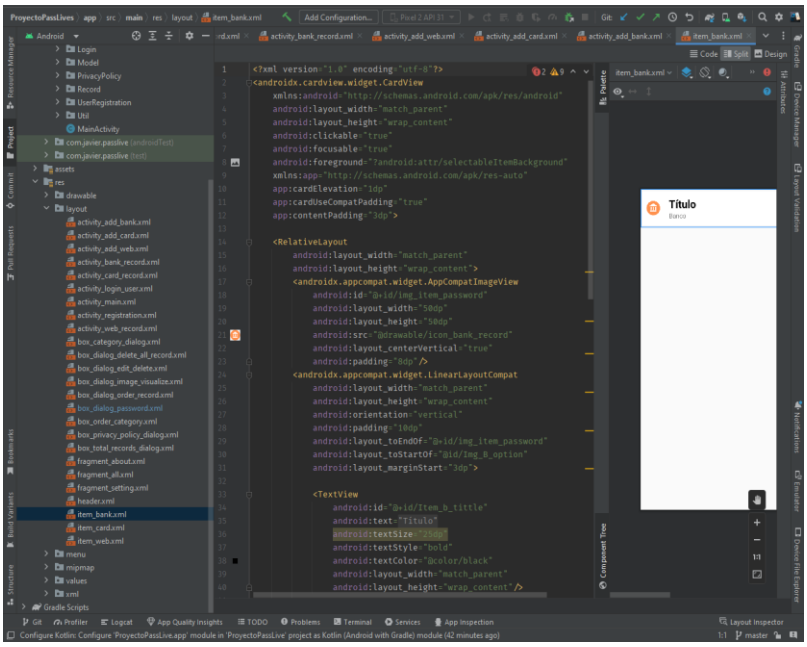
searchRecords(): Es una función que se encarga de buscar registros por un término de búsqueda específico. Esta función se llama al escribir en el campo de búsqueda de MainActivity.

filterRecords(): Es una función que se encarga de filtrar registros por diferentes criterios, como tipo, título o fecha de creación. Esta función se llama al seleccionar una opción de filtro en MainActivity.

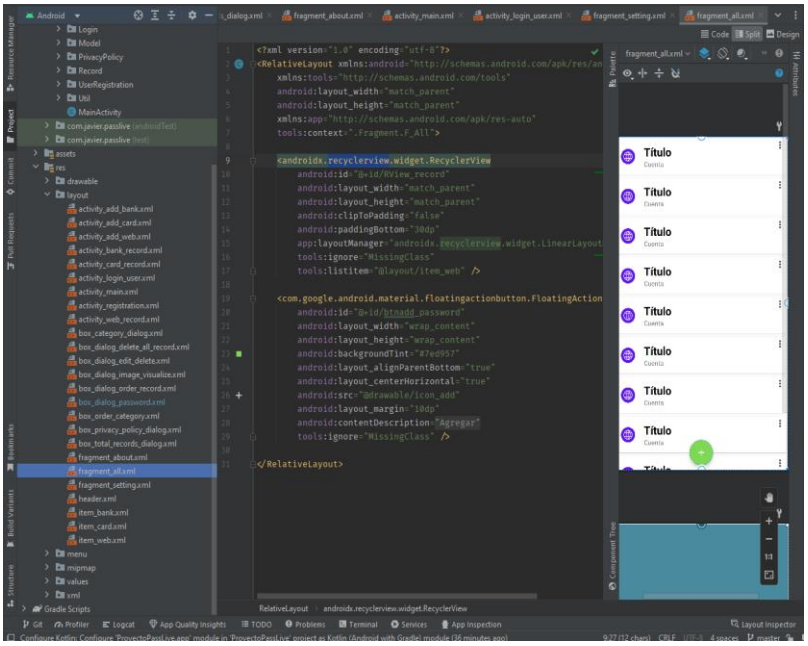
Ficheros:

activity_main.xml: Es el archivo de diseño de la actividad principal MainActivity. En él se define la vista de la lista de registros y se configuran los botones y opciones para agregar, editar y eliminar registros.

item_bank.xml, item_card.xml y ítem_web.xml. Archivo XML que define el diseño de los elementos del **CardView** utilizado para mostrar los registros de contraseñas para sitios web.



fragment_all.xml. Este archivo contiene un RecyclerView en el que se cargará todos los registros de la aplicación.



7.Depliegue y pruebas

Para realizar las pruebas de la aplicación, se han realizado llevado a cabo mediante las pruebas de caja negra o pruebas funcionales para cada una de las funcionalidades que el software deberá cumplir, realizando la siguiente tabla:

Nº	Objetivo probado	Requisitos probados	Pruebas que realizó
1	El usuario debe poder crear una nueva cuenta de usuario.	La aplicación debe permitir el registro de nuevos usuarios. La información del nuevo usuario debe ser guardada correctamente	1. Abrir la aplicación 2. Seleccionar la opción "Registrarse" 3. Introducir los datos necesarios para el registro (email,contraseña y confirmar contraseña) 4. Hacer clic en el botón de registro 5. Compruebe que se muestra un mensaje de éxito y que los datos del nuevo usuario se han almacenado correctamente en la base de datos.
2	El usuario debe poder iniciar sesión en su cuenta.	La aplicación debe permitir el inicio de sesión de los usuarios registrados. La información de inicio de sesión debe ser verificada correctamente	1. Abrir la aplicación. 2. Introducir la contraseña del usuario registrado o acceso biométrico 4. Hacer clic en el botón de inicio de sesión 5 Compruebe que se muestra un mensaje de éxito y que el usuario ha accedido a su cuenta correctamente.
3	El usuario debe poder agregar un nuevo registro.	La aplicación debe permitir la creación de nuevos registros. Los registros deben ser guardados correctamente.	1. Iniciar sesión en la aplicación. 2.Pulsar botón flotante 2. Seleccionar la Categoría. Cuenta Web, cuenta bancaria o tarjeta de crédito. 3. Introducir los datos necesarios del nuevo registro (título, usuario, contraseña) 4. Hacer clic en el botón de guardar 5. Compruebe que se muestra un mensaje de éxito y que la nueva contraseña se ha almacenado correctamente en la base de datos.
4	El usuario debe poder ver todas sus registros guardados.	La aplicación debe permitir la visualización de todos los registros guardadas del usuario. La información de los registros debe ser mostrada correctamente	1. Iniciar sesión en la aplicación 2. Hacer clic en el registro deseado. 3. Comprobar que se muestra una lista con todos las datos del registro guardados por el usuario.

5	El usuario debe poder editar una contraseña existente	La aplicación debe permitir la edición de los diferentes registros existentes. Los registros editadas deben ser almacenados correctamente.	1. Iniciar sesión en la aplicación. 2. Hacer clic en el botón de opciones del registro deseado. 3. Seleccionar el botón “Editar” del cuadro de diálogo. 4. Modificar los datos necesarios del registro (título, usuario, contraseña). 5. Hacer clic en el botón de guardar. 7. Comprobar que se muestra un mensaje de éxito y que los cambios se han guardado correctamente en la bases de datos.
6	El usuario debe poder eliminar una contraseña existente.	La aplicación debe permitir la eliminación de los registros existentes. Las contraseñas eliminadas no deben ser accesibles por el usuario.	1. Iniciar sesión en la aplicación. 2. Hacer clic en el botón superior. derecho del registro deseado. 3. Seleccionar el botón “Eliminar” del cuadro de diálogo. 4. Comprobar que se muestra un mensaje de éxito y que los registros se han eliminado.
7	El usuario debe poder cambiar de contraseña de acceso a la aplicación.	La aplicación debe permitir poder cambiar de contraseña. Las contraseñas deben ser cambiados correctamente.	1. Iniciar sesión en la aplicación. 2. Hacer clic en el botón de menú. 3. Hacer clic en la opción “Ajustes”. 4. Seleccionar la opción “Cambiar password”. 5. Introducir nueva contraseña y la confirmación de contraseña. 6. Hacer clic en el botón cambiar. 7. Comprobar que se muestra un mensaje de éxito y que la contraseña se ha cambiado.
8	El usuario debe poder exportar todos los registros en formato CSV.	La aplicación debe permitir exportar todos sus registros en formato CSV. Los registros deben de ser exportados correctamente.	1. Iniciar sesión en la aplicación. 2. Hacer clic en el botón de menú. 3. Hacer clic en la opción “Ajustes”. 4. Seleccionar la opción “Exportar archivo CSV”. 5. Comprobar que se muestra un mensaje de éxito y que los registros se han exportado correctamente.
9	El usuario debe poder importar todos los registros en formato CSV.	La aplicación debe permitir importar todos sus registros en formato CSV y reflejarse en la pantalla principal de la aplicación. Los registros deben de ser importados correctamente.	1. Iniciar sesión en la aplicación. 2. Hacer clic en el botón de menú. 3. Hacer clic en la opción “Ajustes”. 4. Seleccionar la opción “Importar archivo CSV”. 5. Comprobar que se muestra un mensaje de éxito y que los registros se han exportado correctamente. 6.Examinar que los registros cargan correctamente en pantalla principal.

10	El usuario debe poder eliminar todos los registros a la vez.	La aplicación debe permitir eliminar todos sus a la vez. Los registros deben ser correctamente eliminados.	1. Iniciar sesión en la aplicación. 2. Hacer clic en el botón de menú. 3. Hacer clic en la opción “Ajustes”. 4. Seleccionar la opción “Eliminar todos los registros”. 5. Comprobar que se muestra un mensaje de éxito y que los registros se han eliminado correctamente.
11	El usuario debe poder acceder a las redes sociales de la aplicación.	La aplicación debe permitir el acceso directo a las redes sociales de la aplicación. El acceso a las redes debe ser correctos.	1. Iniciar sesión en la aplicación. 2. Hacer clic en el botón de menú. 3. Hacer clic en la opción “Acerca de”. 4. Seleccionar la red social. 5. Comprobar que nos redirige a la red social pulsada.
12	El usuario debe poder buscar sus registros mediante un buscador.	La aplicación debe permitir buscar registros mediante un buscador. La búsqueda de registros debe de ser correcta.	1.Iniciar sesión en la aplicación. 2. Hacer clic en el botón de “Búsqueda”. 3. Hacer clic en la opción “Acerca de”. 4. Escribir el título del registro 5. Comprobar el registro escrito en el buscador sea el correcto.
13	El usuario debe poder ordenar sus registros.	La aplicación debe permitir ordenar los registros El orden de los registros debe de ser correcto.	1.Iniciar sesión en la aplicación. 2. Hacer clic en el botón de “Ordenar”. 3. Seleccionar la opción deseado (Categorías, recientes, antiguos, A-Z, Z-A). 4. Comprobar que el orden de los registros es el correcto.
14	El usuario debe poder leer la política de privacidad de la aplicación.	La aplicación debe permitir al usuario leer la política de privacidad al registrarse en la aplicación. El dialogo de la política de privacidad debe de ser aceptado por el usuario para su continuidad.	1. Abrir la aplicación. 2. Seleccionar la opción "Registrarse". 3. Introducir los datos necesarios para el registro (contraseña y confirmar contraseña). 4. Aceptar los términos y la política de privacidad de la aplicación. 5. Compruebe que se muestra un mensaje de éxito y que los datos del nuevo usuario se han almacenado correctamente en la base de datos.

15	El usuario debe salir y cerrar la sesión.	La aplicación debe permitir al usuario poder cerrar la sesión de usuario. La aplicación debe realizar el cierre de sesión correcto.	1. Iniciar sesión en la aplicación. 2. Hacer clic en el botón de menú. 3. Hacer clic en la opción “Salir”. 5. Compruebe que se muestra un mensaje de cierre de sesión y que la aplicación realiza el cierre de sesión.

Realizamos pruebas para asegurarnos de que la aplicación se maneja correctamente en situaciones de error, como cuando se ingresan datos inválidos o cuando ocurre algún problema de conexión a la base de datos. También nos aseguramos de que la aplicación funciona correctamente en diferentes dispositivos, para garantizar la compatibilidad con una amplia gama de dispositivos.

Además, las pruebas de caja negra también nos permitieron detectar posibles vulnerabilidades de seguridad y comprobar que los datos sensibles, como las contraseñas, se almacenan de forma segura y se manejan correctamente en la aplicación.

Estas pruebas de caja negra nos permitieron validar que la aplicación cumple con los requisitos funcionales especificados y asegurar que las diferentes funcionalidades son utilizables y cumplen con las expectativas del usuario.

8.Conclusiones

Entre los objetivos que se alcanzaron en el proyecto se encuentran los siguientes:

- **Diseño de una interfaz de usuario intuitiva**: Se desarrolló una interfaz de usuario clara y fácil de usar, lo que facilita la experiencia del usuario y aumenta la usabilidad de la aplicación.
- **Seguridad de la información**: Se implementaron diversas medidas de seguridad para proteger la información almacenada en la aplicación, tales como la encriptación de datos y la autenticación de usuarios mediante contraseñas seguras.
- **Organización de contraseñas**: Se contrató una estructura de datos adecuada para almacenar las contraseñas de manera organizada y fácilmente accesible para el usuario.

Sin embargo, a pesar de haber logrado estos objetivos, también se presentaron ciertas dificultades en el desarrollo del proyecto que no permitieron alcanzar algunos de los objetivos propuestos inicialmente. Una de las principales causas fue la falta de recursos y tiempo limitado para el desarrollo, lo que limitó la capacidad de implementar ciertas funcionalidades y características avanzadas.

En cuanto a la opinión sobre el trabajo realizado, considero que fue un proyecto desafiante pero exitoso en general. Además, se cumplió con los principales objetivos del proyecto y se entregó una aplicación funcional y segura para los usuarios.

En general, se puede decir que el proyecto fue un éxito en términos de los objetivos alcanzados y la calidad de la aplicación desarrollada. Aunque hubo ciertas dificultades y limitaciones, se reparará crear una aplicación útil y efectiva para los usuarios, lo que es lo más importante. En el futuro, se pueden realizar mejoras y agregar nuevas funcionalidades para seguir mejorando la aplicación y satisfacer las necesidades de los usuarios.

9.Vías Futuras

Existen varias vías futuras que podrían explorarse en el proyecto de la aplicación. Algunos de los objetivos planteados inicialmente y que no se pueden incluir en la versión actual podrían ser considerados para una futura actualización.

Una de estas vías podría una posible mejora, sería la integración de un generador de contraseñas seguras dentro de la aplicación. Esto facilitaría a los usuarios la creación de contraseñas fuertes y únicas, lo que aumentaría la seguridad de sus cuentas.

Otra vía futura que podría ser explorada es la integración de la aplicación con navegadores web. Esto permitiría a los usuarios acceder a sus contraseñas ocultas directamente desde el navegador sin tener que abrir la aplicación por separado. Además, podría permitir que la aplicación se integre con la funcionalidad de autocompletado de contraseñas de los navegadores, lo que facilitaría el proceso de inicio de sesión.

La integración con sistemas de gestión de identidad empresarial: para usuarios empresariales, la integración con sistemas de gestión de identidad empresarial, como Active Directory o LDAP, sería una funcionalidad valiosa. Esto permitiría a los empleados acceder a sus contraseñas almacenadas a través de la misma infraestructura de inicio de sesión que utilizan para acceder a otros recursos de la empresa.

También podría ocurrir la integración con servicios de almacenamiento en la nube como Dropbox o Google Drive, lo que permitiría a los usuarios hacer copias de seguridad de sus datos y acceder a ellos desde diferentes dispositivos.

Otra posible vía de mejora sería la implementación de un sistema de alertas que notifique a los usuarios cuando se detecte una brecha de seguridad en una de sus cuentas almacenadas. De esta manera, los usuarios podrían tomar medidas inmediatas para proteger sus cuentas.

En cuanto a la interfaz de usuario, se podría considerar una actualización para hacerla más intuitiva y fácil de usar. También podría incluirse una opción de personalización para permitir a los usuarios ajustar la apariencia de la aplicación a sus preferencias personales.

En conclusión, hay varias vías futuras que podrían explorarse en el proyecto de la aplicación para guardar contraseñas, desde mejoras en la seguridad hasta la integración con servicios en la nube o la actualización de la interfaz de usuario. Todas estas mejoras permitirían que la aplicación evolucione y se mantengan actualizadas a medida que cambien las necesidades de los usuarios.

10.Biografía

Durante la elaboración de la aplicación para guardar contraseñas, se consultaron diversas fuentes para obtener información relevante sobre programación, seguridad de bases de datos, diseño de interfaces de usuario, entre otros temas. Entre las fuentes consultadas se encuentran:

- **Autentia:** Es una consultora de servicios tecnológicos española que entre otras cosas realiza publicaciones de contenidos didácticos. Empleé una recopilación de buenas prácticas y patrones para el desarrollo del software. **https://www.autentia.com/wp-content/uploads/2021/05/Fichas_SoftwareDesign.pdf**

- **Developer.Android.com:** Es la documentación oficial de Android, donde puede encontrar recursos para el desarrollo de la aplicación. **https://developer.android.com/**

- **Develou.com:** Es una plataforma en línea que ofrece tutoriales y artículos sobre programación. **https://develou.com/**

- **Stack Overflow**: Es una plataforma en línea donde los desarrolladores pueden hacer preguntas y obtener respuestas de otros desarrolladores. Es una fuente importante de información que me ayudó a resolver muchos de los problemas de código. **https://stackoverflow.com/**

- **Copyprogramming.com**: Es un portal de tecnología para programadores. En él me inspiré para realizar la estructura de las vistas de XML utilizando Linear Layout Compat. https://copyprogramming.com/howto/what-s-the-difference-between-linearlayout-and-linearlayoutcompat

https://copyprogramming.com/howto/what-s-the-difference-between-linearlayout-and-linearlayoutcompat

- **Desarrollolibre.net:** Es un sitio web donde se pueden encontrar artículos dedicados a la programación. Consultando entre otras cosas la utilización de los CardView.
  https://www.desarrollolibre.net/blog/android/como-crear-cardviews-personalizados-en-android

- **GitHub:** Se trata de una de las principales plataformas para crear proyectos abiertos de herramientas y aplicaciones, y se caracteriza sobre todo por sus funciones colaborativas que ayudan a que todos puedan aportar su granito de arena para mejorar el código. Entre otras cosas, se utilizó para crear los PhotoView que tiene como objetivo ayudar a producir una implementación fácil de usar de un ImageView de Android con zoom.
  https://github.com/Baseflow/PhotoView

- **Devexperto de Antonio Leiva:** Es un sitio web que ofrecen tutoriales, artículos y recursos para desarrolladores de software. En concreto consulté artículos sobre los concatadapter y RecycleWiew.
  https://www.devexperto.com/concatadapter/

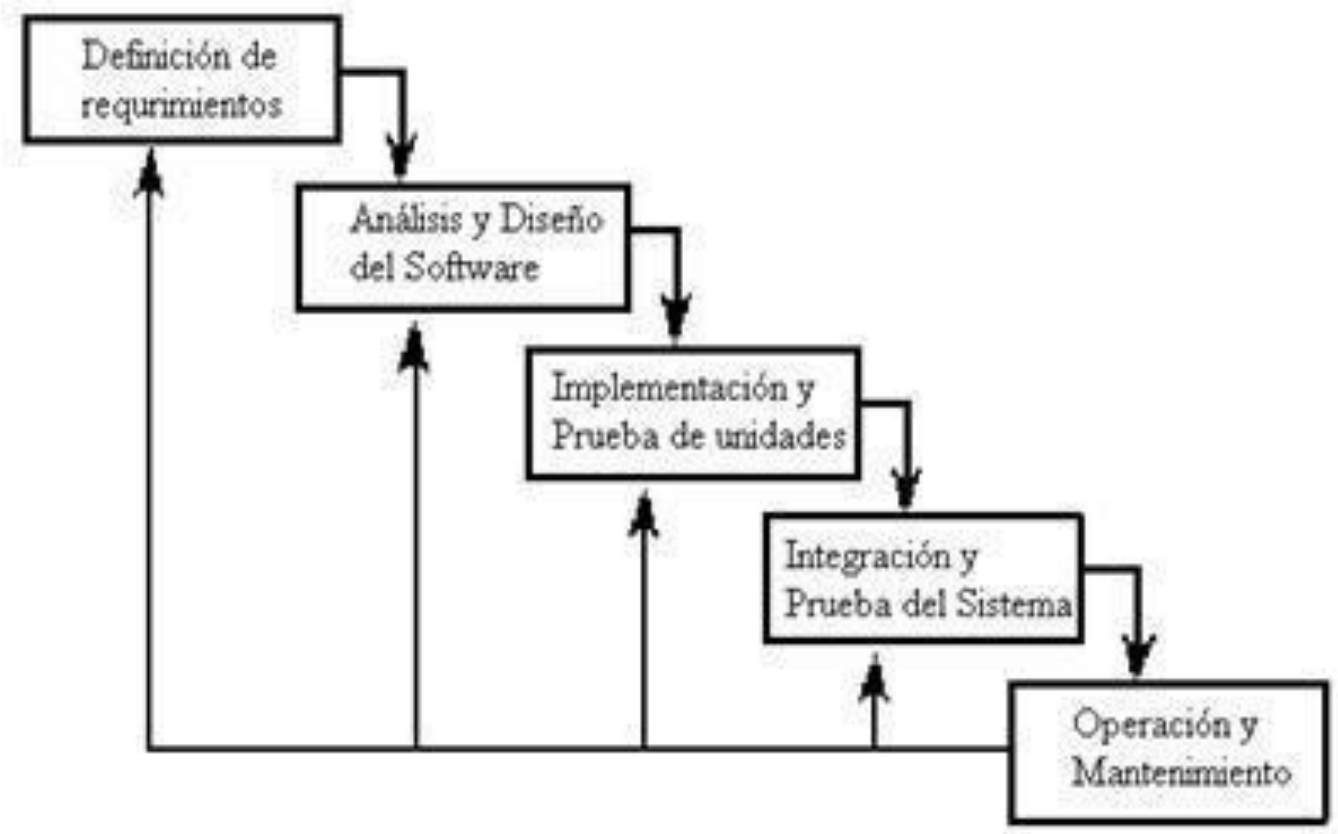
- **El blog de Luis Salvador Roa Rodriguez:** Es una fuente de información sobre la encriptación de bases de datos seguras.
  https://www.enmilocalfunciona.io/base-de-datos-seguras-en-android/

Cada una de estas fuentes ha sido útil para la elaboración de diferentes aspectos de la aplicación, desde el diseño y la implementación hasta la seguridad y la confiabilidad. La información obtenida ha sido crucial para la toma de decisiones informadas en cada etapa del proyecto.

11.Anexos

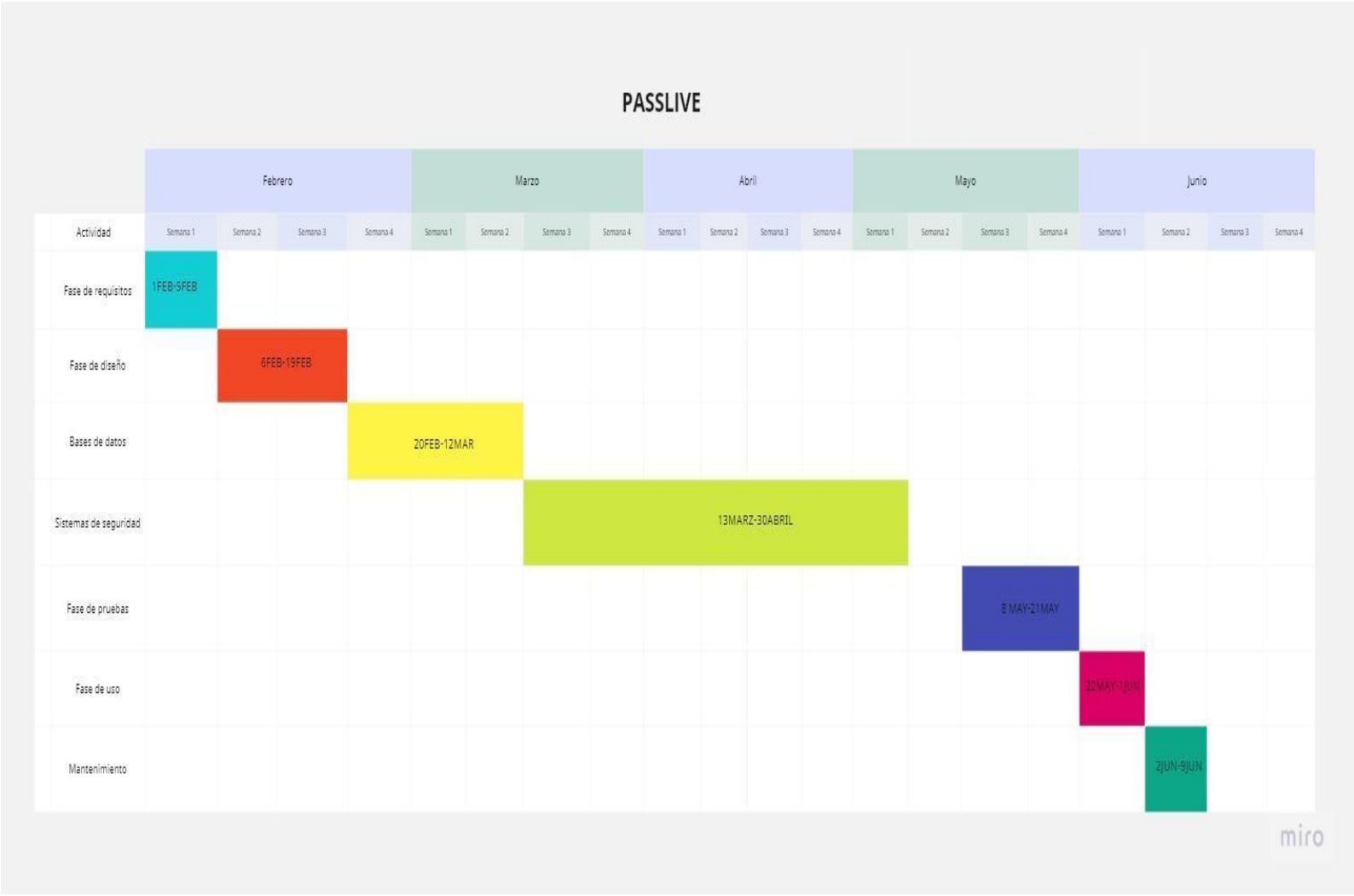
Anexo A- Metodología empleada

A.1.Ciclo de vida del proyecto



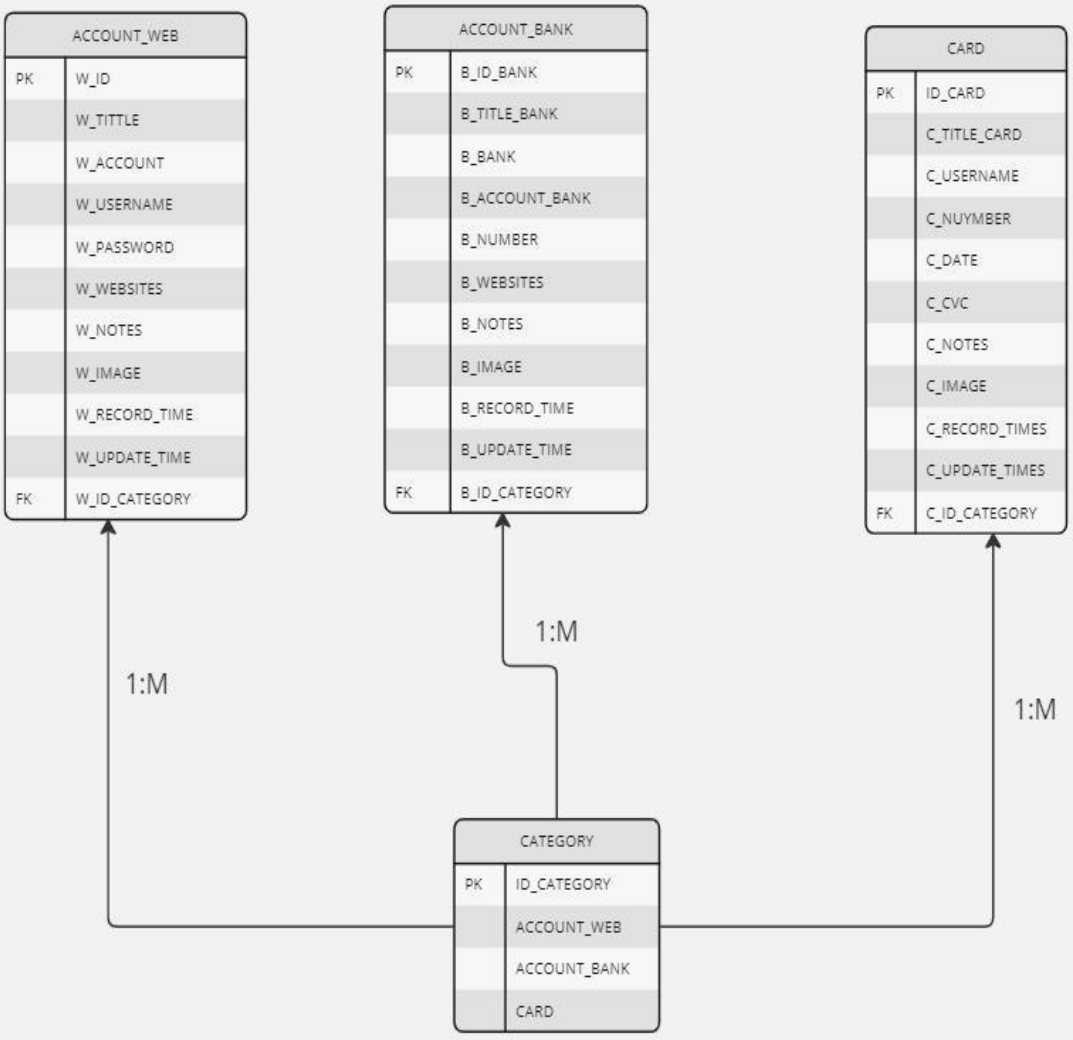
Anexo B. Estimación de recursos y planificación

B.1.Gráfico recursos

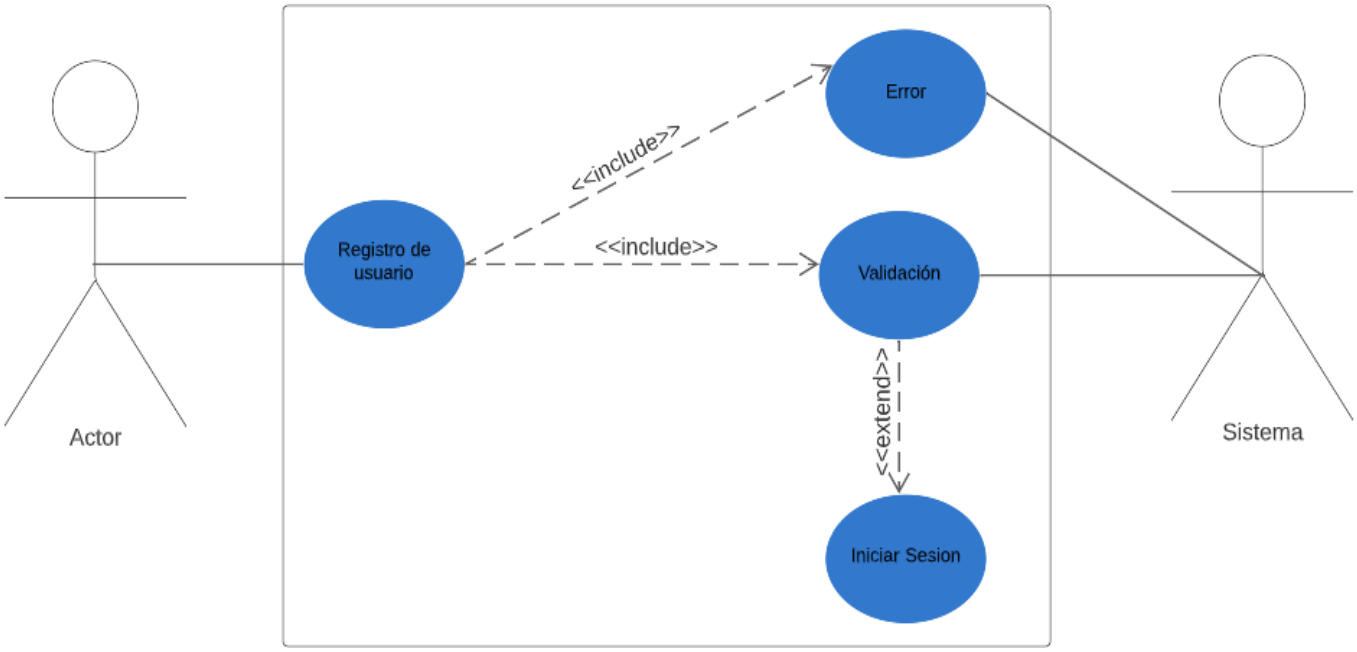


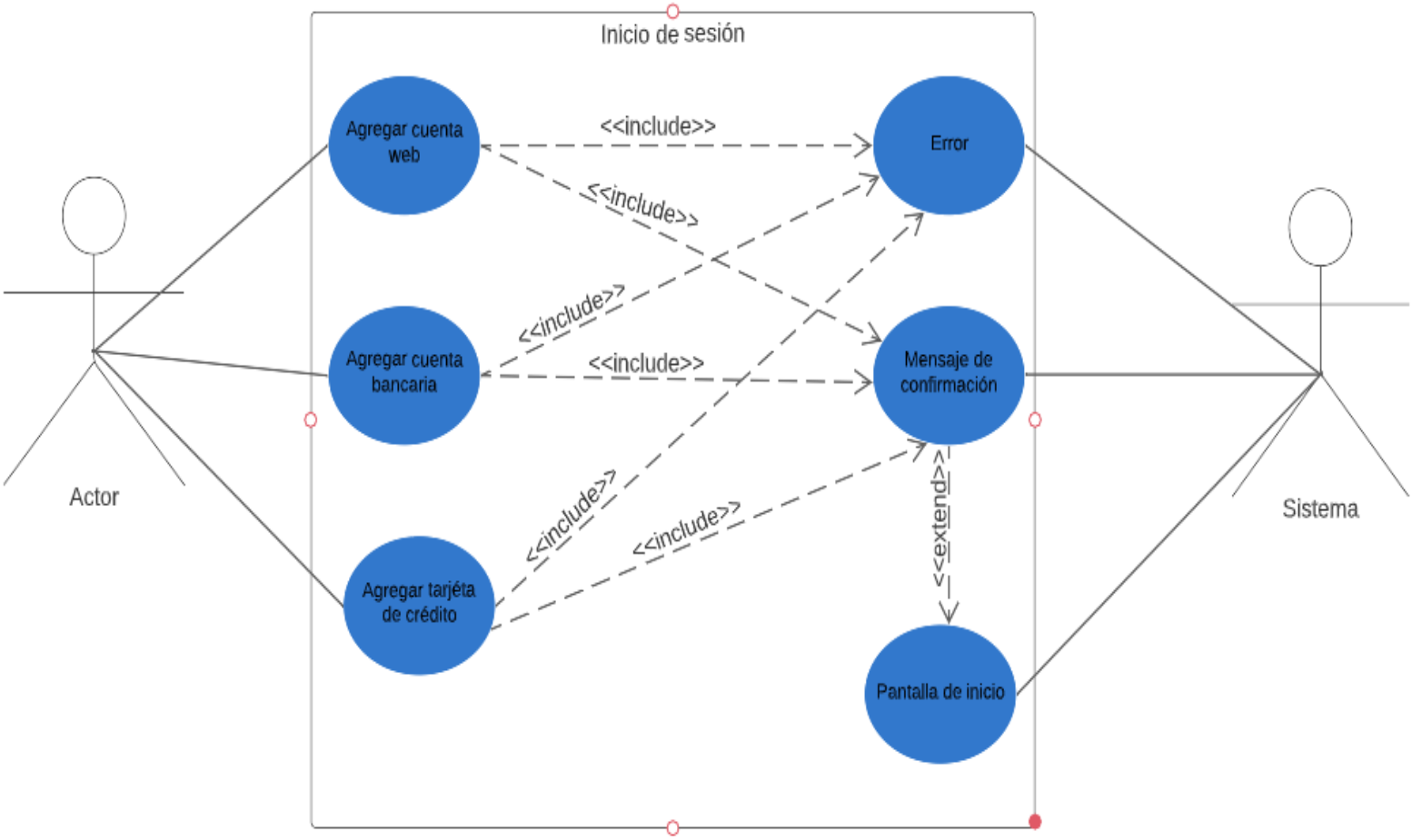
Anexo C. Análisis

C.1.Diagráma Entidad-Relación

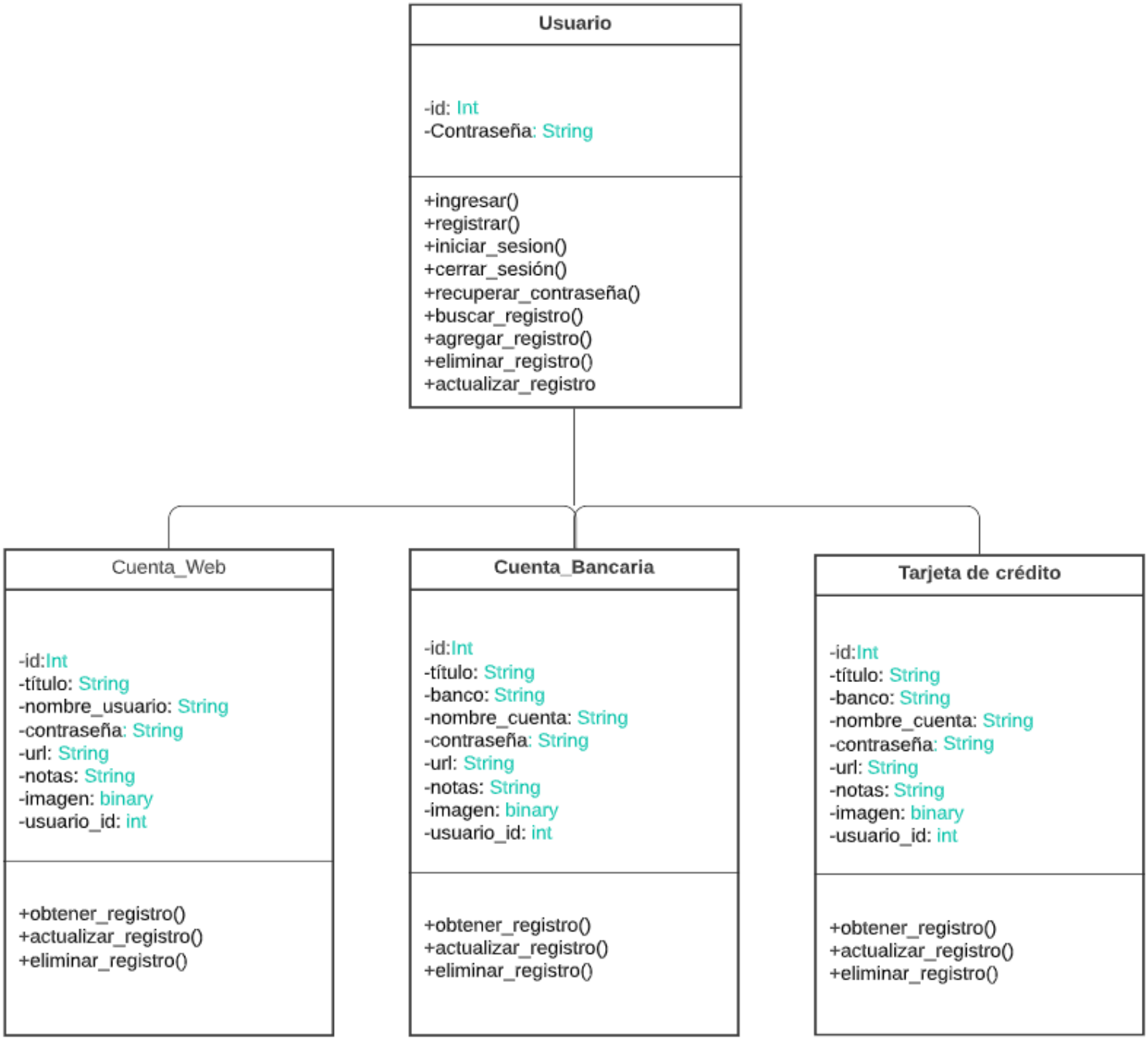


D.2.Diagráma de Casos de Uso



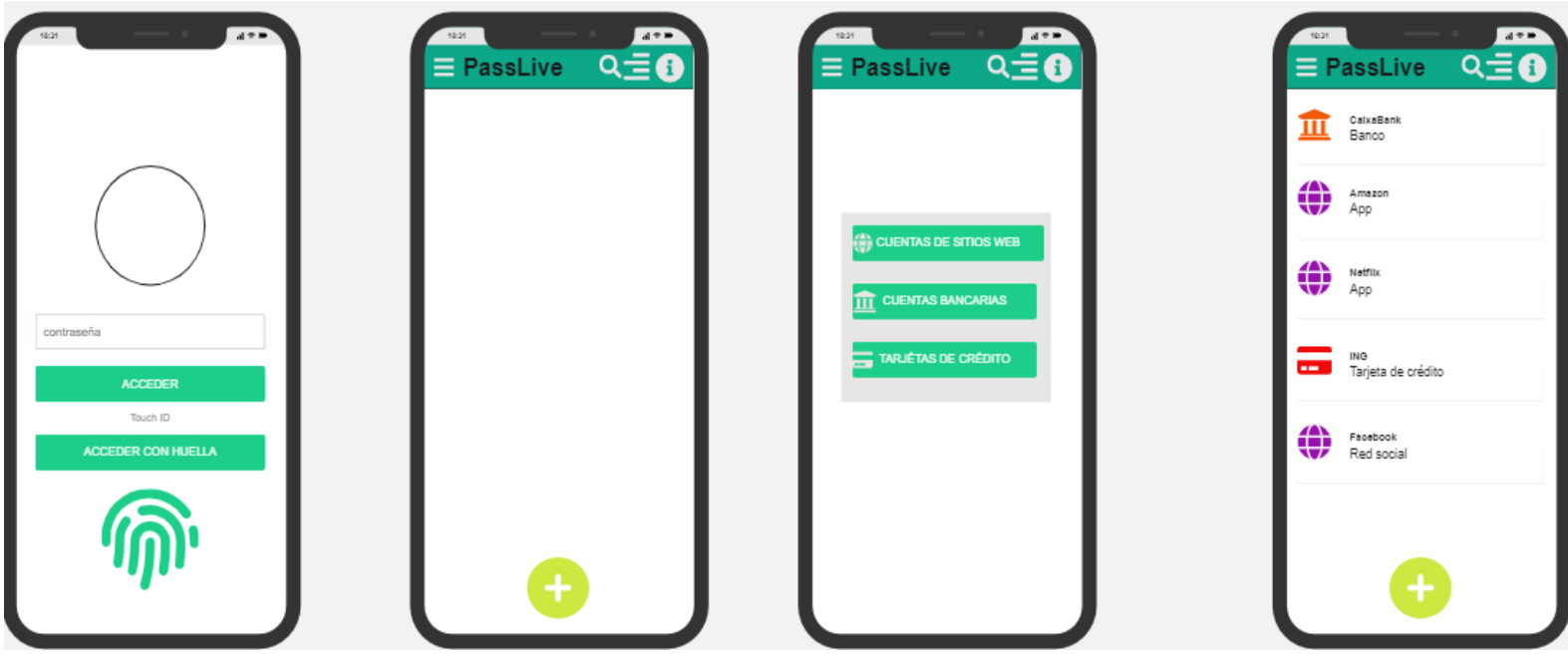


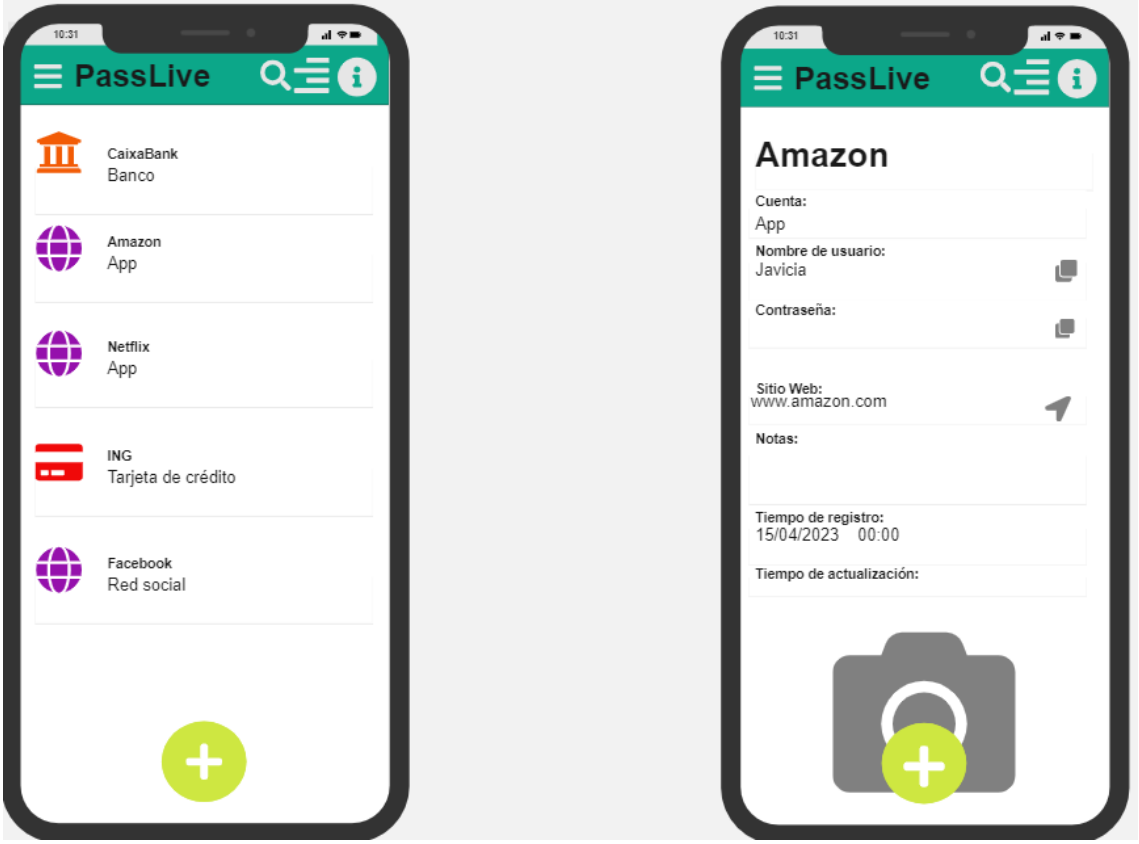
D.3.Diagráma de Clases



Anexo E. Diseño

E.1.Wireframes




10:31
PassLive
CaixaBank
Banco
Amazon
App
Netflix
App
ING
Tarjeta de crédito
Facebook
Red social
10:31
PassLive
Amazon
Cuenta:
App
Nombre de usuario:
Javicia
Contraseña:
Sitio Web:
www.amazon.com
Notas:
Tiempo de registro:
15/04/2023 00:00
Tiempo de actualización: