

TFG

TRABAJO DE FINAL DE GRADO



I.E.S. BARAJAS

Desarrollo de Aplicaciones Multiplataforma

Aplicación Android para seguimiento y consulta de información sobre criptomonedas

Android application for tracking and consulting information about cryptocurrencies

Damián Peña-Marín Samaniego
18-11-2024

Copyright © 2024 Damián Peña-Marín Samaniego
Este trabajo está licenciado bajo CC BY-NC-SA 4.0



“Dedicado a mis padres y mi familia, gracias a ellos hoy soy mejor y es posible que este trabajo se haya realizado. Por su confianza, su amor y su ayuda. También a mis amigos, en especial a los que me acompañaron en clase, por los buenos momentos y el apoyo. Gracias a mi profesora Macarena y el resto de profesores. A todos por esta etapa académica que culmina”

“Y por último, agradecimientos a mí mismo por seguir adelante”

Índice

I.	Introducción	2
II.	Abstract	2
III.	Objetivos del Proyecto	3
IV.	Temporalización del proyecto y diagrama de Gantt	4
V.	Metodología de Desarrollo.....	5
VI.	Tecnologías utilizadas	7
	1. Lenguajes de programación.....	7
	2. Softwares y sistemas	8
VII.	Desarrollo.....	11
	3. Requisitos	11
VIII.	Leyes y normativas aplicables.....	13
IX.	Versiones de la aplicación.....	13
X.	Requisitos funcionales (Casos de uso)	17
XI.	Diseño de pantallas y explicación básica de funcionalidades	20
	1. Diagrama de flujo de pantallas	20
	2. Pantalla de Inicio	21
	3. Pantalla de mercado y detalle de criptomonedas.....	22
	4. Pantalla cartera de criptomonedas.....	25
	5. Pantalla de cuenta de usuario (inicio de sesión, registro y detalles de la cuenta)....	29
XII.	Estructura de datos de la base de datos	34
XIII.	Pruebas y resultados	36
	1. Prueba de uso correcto de las funcionalidades de la aplicación	36
	2. Prueba de uso incorrecto y manejo de errores	42
XIV.	Conclusiones	47
XV.	Proyección a futuro	48
XVI.	Bibliografía	49
XVII.	Autoría y derechos de la obra.....	50

I. Introducción

En el dinámico y volátil mundo de las criptomonedas, la gestión eficiente de las inversiones es crucial para el éxito. Este Trabajo de Fin de Grado presenta una aplicación móvil diseñada para abordar esta necesidad, proporcionando a los usuarios una herramienta integral para el seguimiento y análisis de sus portafolios de criptomonedas.

La aplicación permite a los usuarios registrar sus activos de criptomonedas. A través de la integración con APIs de datos de mercado en tiempo real, la aplicación ofrece una visión actualizada del valor del portafolio, el rendimiento individual de cada criptomoneda y las fluctuaciones del mercado.

Con funcionalidades como gráficos interactivos y noticias del mercado, la aplicación empodera a los usuarios con la información necesaria para tomar decisiones de inversión informadas. Este proyecto no solo aborda un problema real en el creciente mercado de las criptomonedas, sino que también demuestra la aplicación práctica de los principios de desarrollo de software móvil aprendidos durante el grado.

II. Abstract

In the dynamic and volatile world of cryptocurrencies, efficient investment management is crucial to success. This Bachelor's Degree Final Project presents a mobile application designed to address this need, providing users with a comprehensive tool for tracking and analyzing their cryptocurrency portfolios.

The application allows users to record their cryptocurrency assets. Through integration with real-time market data APIs, the application offers an up-to-date view of the portfolio's value, the individual performance of each cryptocurrency, and market fluctuations.

With functionalities such as interactive charts and market news, the application provides users with the information needed to make informed investment decisions. This project not only addresses a real-world problem in the growing cryptocurrency market, but also



demonstrates the practical application of the mobile software development principles learned during the degree.

III. Objetivos del Proyecto

El objetivo principal de este Trabajo de Fin de Grado es desarrollar una aplicación móvil para Android que facilite la gestión y seguimiento de portafolios de criptomonedas, proporcionando a los usuarios herramientas útiles para analizar sus inversiones y tomar decisiones informadas. Para alcanzar este propósito, se establecen los siguientes objetivos específicos:

1. **Registrar y gestionar activos de criptomonedas:** Permitir a los usuarios añadir y gestionar sus inversiones en diversas criptomonedas, a partir de una lista actualizada obtenida mediante la API de CoinGecko [6].
2. **Integrar datos de mercado en tiempo real:** Implementar la integración con APIs de datos de criptomonedas, como CoinGecko y CoinMarketCap [7], para ofrecer información actualizada sobre el valor de mercado y las fluctuaciones en tiempo real.
3. **Proporcionar visualización clara de datos:** Ofrecer herramientas de visualización como gráficos interactivos que permitan a los usuarios analizar el rendimiento de sus activos y detectar tendencias en el mercado de criptomonedas.
4. **Facilitar el seguimiento personalizado del portafolio:** Permitir a los usuarios hacer seguimiento del valor de su portafolio en euros, actualizando la información de sus activos de forma fácil y dinámica.
5. **Implementar funcionalidades de autenticación y gestión de cuentas:** Incluir un sistema de inicio de sesión que permita a los usuarios crear y actualizar sus datos de cuenta de manera segura.
6. **Implementar funcionalidades guardado “en la nube”:** Disponer de un almacenamiento en la nube asociado a la cuenta de usuario almacenar y recuperar sus datos en cualquier momento y desde cualquier lugar.

7. **Desarrollar una aplicación intuitiva y accesible:** Diseñar una interfaz de usuario fácil de usar, que permita una navegación fluida y una experiencia positiva para usuarios con distintos niveles de conocimiento sobre criptomonedas.

IV. Temporalización del proyecto y diagrama de Gantt

Fecha de Inicio: Septiembre 2024

Fecha de Finalización: Del 9 al 12 de diciembre 2024

Tareas	Septiembre				Octubre				Noviembre				Diciembre	
	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Investigación preliminar y definición														
Diseño de base de datos y selección de tecnologías														
Creación de Actividades, fragmentos y diseño de UI														
Desarrollo de la pantalla de inicio														
Desarrollo de la pantalla de mercado														
Implementación de detalle de mercado														
Desarrollo de pantalla de cartera														
Implementación de inicio de sesión y registro														
Desarrollo de la pantalla de cuenta de usuario														
Funcionalidad de copia de seguridad														
Pruebas de usabilidad y corrección de errores														
Pruebas finales y retoques finales														
Memoria del TFG y preparación de presentación														
Revisión final														

Figura 1. Diagrama de Gantt en formato tabla

Diagrama de Gantt resumido por meses:

1. **Septiembre: Investigación, planificación, diseño y desarrollo inicial**
 - Investigación de las APIs y tecnologías, y desarrollo de la pantalla de inicio.
 - Definición de base de datos, tecnologías y desarrollo de la pantalla de mercado.
2. **Octubre: Desarrollo de activities y funcionalidades**
 - Conexiones a base de datos, desarrollo del login, implementación de distintas pantallas.



3. **Noviembre: Comienzo de la memoria del TFG y más desarrollo**
 - Desarrollo de más funcionalidades.
 - Comienzo de la memoria del TFG.
 - Revisión de la app y arreglos antes de la entrega.
4. **Diciembre: Finalización de la memoria del TFG y preparación de la presentación**
 - Preparación final de la presentación y preparación para la presentación tras el puente de diciembre.

V. Metodología de Desarrollo

El desarrollo de la aplicación **Cripto Utilidades Damian** se llevó a cabo siguiendo una metodología iterativa e incremental, centrada en la creación de funcionalidades clave de forma progresiva y asegurando la integración continua de componentes. Se empleó un enfoque modular para asegurar que cada parte de la aplicación fuese desarrollada, probada e integrada de manera eficiente, lo que permitió ajustes rápidos en cada iteración. Los aspectos clave de esta metodología incluyen:

1. **Análisis de Requisitos y Definición de Historias de Usuario:** El desarrollo comenzó con la identificación clara de los requisitos, a partir de los cuales se definieron las historias de usuario que guiarían la implementación de las funcionalidades. Cada historia de usuario cubría una funcionalidad específica de la aplicación, como el registro de usuarios, la visualización de la lista de criptomonedas o el almacenamiento de criptomonedas en el monedero. Esto permitió descomponer el proyecto en pequeños componentes manejables.
2. **Diseño e Implementación Modular:** Cada funcionalidad de la aplicación fue desarrollada de manera independiente, siguiendo un enfoque modular. Esto permitió trabajar en diferentes aspectos de la aplicación, como la conexión con APIs externas, la gestión de bases de datos locales y en la nube, y la creación de la interfaz de usuario. Para la implementación, se emplearon patrones de diseño

como el Modelo-Vista-ViewModel (MVVM) [8], que separa la lógica de negocio de la interfaz gráfica, facilitando el mantenimiento del código.

3. **Integración de APIs y Servicios Externos:** La aplicación se conecta a APIs externas como CoinGecko para obtener información en tiempo real sobre criptomonedas, y utiliza CoinMarketCap para proporcionar datos detallados a través de una WebView. Se empleó la librería **Retrofit** [9] para realizar las peticiones dinámicas a la API, lo que permitió cargar y visualizar listas de criptomonedas de forma eficiente en un RecyclerView [10], combinando con **Glide** para la carga de imágenes.
4. **Gestión de Datos Locales y Remotos:** Para el almacenamiento de datos, se utilizaron dos sistemas complementarios:
 - **SQLite** [11] para gestionar el almacenamiento local de las criptomonedas que los usuarios agregan a su monedero, permitiendo un seguimiento offline.
 - **MySQL** para la gestión de cuentas de usuarios, con una base de datos en la nube que permite la creación, edición y autenticación de usuarios, así como el respaldo de la información de las criptomonedas del monedero.
5. **Diseño y Usabilidad de la Interfaz de Usuario:** Se implementó un diseño de interfaz sencillo e intuitivo, asegurando una experiencia fluida para los usuarios. Se empleó **Data Binding** para vincular los datos observables con los elementos de la interfaz de usuario de forma declarativa. Además, se utilizó un RecyclerView con carga dinámica de elementos para permitir la navegación por grandes volúmenes de datos.
6. **Autenticación y Funcionalidades de Cuenta:** Se desarrolló un sistema de autenticación que permite a los usuarios registrarse, iniciar sesión y editar su información básica. Además, se incorporó la opción de recordar las credenciales (email y contraseña) mediante el almacenamiento seguro de la información utilizando SQLite. Los datos de las criptomonedas almacenadas en el monedero también pueden respaldarse en la nube, lo que permite a los usuarios restaurar su información en diferentes dispositivos.
7. **Pruebas y Validación:** Durante el desarrollo, se realizaron pruebas continuas en dispositivos Android para validar el funcionamiento de cada funcionalidad. Las

pruebas incluyeron tanto la verificación de las historias de usuario como la validación del rendimiento y la correcta integración de los datos obtenidos de las APIs. Las pruebas unitarias y de integración se realizaron especialmente en áreas críticas como la autenticación y la gestión de datos.

8. **Uso de Librerías Externas:** El proyecto utilizó diversas librerías de terceros para optimizar el desarrollo, tales como:
- **Volley** para realizar peticiones HTTP.
 - **Glide** para la carga eficiente de imágenes.
 - **Retrofit** para la gestión de peticiones a la API de CoinGecko. Estas herramientas facilitaron la implementación de funcionalidades clave de la aplicación.

VI. Tecnologías utilizadas

1. Lenguajes de programación

1.1. Java

Para la implementación de la aplicación se ha utilizado **Java** [1].

Java es un lenguaje de programación orientado a objetos, ampliamente utilizado y de código abierto. Comenzó a comercializarse en 1995 por Sun Microsystems, y desde entonces ha sido adoptado para el desarrollo de diversas aplicaciones, especialmente por su robustez, seguridad y capacidad multiplataforma. Java es fiable y rápido, lo que lo convierte en una opción ideal para el desarrollo de aplicaciones Android, donde se requiere rendimiento y escalabilidad.



Figura 2. Logo de Java

2. Softwares y sistemas

1.2. Android

Android [2] es un sistema operativo creado por Google y presentado en 2008. Está basado en el núcleo de Linux y en diversos softwares de código abierto. Fue concebido para usarse en dispositivos móviles con pantalla táctil, como smartphones, tablets, relojes inteligentes, vehículos y televisores.



Figura 3. Logo de Android

1.3. Android Studio

Para el desarrollo del proyecto se ha utilizado Android Studio [3].

Android Studio es el entorno de desarrollo integrado (IDE) oficial para Android, basado en IntelliJ IDEA. Se seleccionó debido a sus numerosas ventajas, tales como:

- Un editor de código potente y fácil de usar.
- Herramientas de desarrollo de IntelliJ integradas.
- Compilación flexible basada en Gradle, que permite configurar y personalizar las builds.
- Plantillas y herramientas que simplifican la creación de aplicaciones Android.
- Soporte para pruebas y debugging en una variedad de marcos de trabajo.



Figura 4. Logo de Android Studio

1.4. GitHub

GitHub [4] es una plataforma de desarrollo colaborativo que permite alojar proyectos y gestionar su evolución mediante el sistema de control de versiones Git. Es ampliamente utilizada para desarrollar y compartir código fuente de programas informáticos. Las principales ventajas y características de GitHub son:

- **Colaboración eficiente:** Facilita el trabajo en equipo al permitir que varios desarrolladores trabajen en el mismo proyecto simultáneamente.
- **Control de versiones:** Git registra todos los cambios realizados en un proyecto, facilitando la revisión, el deshacer errores y el seguimiento del historial.
- **Almacenamiento en la nube:** GitHub ofrece un espacio centralizado para alojar proyectos, accesible desde cualquier lugar.

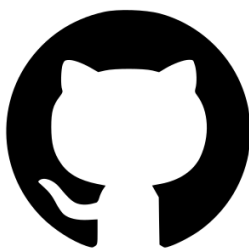


Figura 5. Logo de GitHub

1.5. MySQL

Para la persistencia de datos y la autenticación de usuarios se ha utilizado una **base de datos MySQL [5] alojada en la nube**. MySQL es un sistema de gestión de bases de datos relacional basado en SQL, ampliamente utilizado en el desarrollo de aplicaciones web y móviles debido a su eficiencia, fiabilidad y capacidad para manejar grandes volúmenes de datos. Esta solución se ha implementado en el

proyecto para gestionar el almacenamiento remoto de la información de los usuarios y sus portafolios de criptomonedas.

Las principales ventajas de utilizar **MySQL en la nube** son:

- **Escalabilidad:** MySQL es capaz de manejar grandes cantidades de datos y escalar según la demanda.
- **Seguridad:** Proporciona múltiples capas de seguridad para la gestión de usuarios y el acceso a la base de datos.
- **Interoperabilidad:** Se puede integrar fácilmente con diferentes tecnologías y plataformas de desarrollo.
- **Disponibilidad:** Al estar alojado en la nube, garantiza que los datos estén disponibles en todo momento y desde cualquier lugar.



Figura 6. Logo de MySQL

Para la autenticación y la gestión de usuarios, se ha implementado un sistema basado en **MySQL**, lo que permite a los usuarios registrarse, iniciar sesión y editar su información. Los datos se almacenan en una tabla específica de usuarios, permitiendo la recuperación y actualización de los mismos. A diferencia de soluciones NoSQL, MySQL permite realizar consultas complejas que optimizan la gestión de grandes volúmenes de información.



VII. Desarrollo

3. Requisitos

A continuación, se define una lista de los requisitos funcionales que debe cubrir la aplicación:

1. Usuario

1.1. Creación de un usuario

La aplicación permite a los usuarios crear una cuenta nueva ingresando su correo electrónico y una contraseña segura.

1.2. Inicio de sesión

La aplicación permite a los usuarios iniciar sesión utilizando las credenciales creadas previamente (correo electrónico y contraseña).

1.3. Cierre de sesión

La aplicación permite cerrar la sesión del usuario en cualquier momento.

1.4. Edición de datos del usuario

El usuario puede modificar ciertos datos personales, como el correo electrónico o contraseña, desde su perfil, así como poder mostrar la contraseña en texto plano con un “switch”.

1.5. Recordar credenciales

La aplicación permite recordar las credenciales del usuario si selecciona la opción "recordar contraseña" en la pantalla de inicio de sesión.

2. Criptomonedas

2.1. Listar criptomonedas

La aplicación permite al usuario visualizar una lista actualizada de criptomonedas obtenida a través de la API de CoinGecko.

2.2. Visualización de información detallada

La aplicación permite acceder a la información detallada de cada criptomoneda.

2.3. Añadir criptomonedas al monedero

El usuario puede agregar criptomonedas a su monedero personal para hacer un seguimiento de su valor en tiempo real.



2.4. Eliminar criptomonedas del monedero

La aplicación permite eliminar criptomonedas del monedero del usuario.

2.5. Seguimiento del valor de las criptomonedas

El usuario puede consultar el valor actual de las criptomonedas añadidas a su monedero, actualizado en tiempo real.

3. Sincronización y almacenamiento de datos

3.1. Almacenamiento local de criptomonedas

La aplicación almacena los datos del monedero del usuario localmente utilizando SQLite, permitiendo el acceso a la información sin conexión.

3.2. Respaldo en la nube.

El usuario puede respaldar su monedero en la nube, utilizando una base de datos MySQL, para garantizar que los datos no se pierdan.

3.3. Restauración desde la nube

La aplicación permite restaurar los datos del monedero desde la nube en caso de cambio de dispositivo o reinstalación de la aplicación.

4. Funcionalidades adicionales

4.1. Contador de días hasta la entrega del TFG

La aplicación muestra en la pantalla principal un contador con el número de días restantes hasta el 1 de septiembre (fecha de entrega del TFG) (esta característica fue mandada hacer obligatoriamente por la anterior profesora).

4.2. Visualización de criptomonedas en la pantalla principal

La aplicación muestra una vista resumida de criptomonedas seleccionadas en la pantalla principal, facilitando su seguimiento.

5. Otros requisitos

5.1. Las acciones del usuario (como carga de datos, imágenes, actualización de datos, etc.) deben completarse dentro de tiempos razonables.

5.2. La aplicación debe contener manejo de excepciones de manera que la aplicación no se cierre de forma inesperada o presente comportamientos anormales ante uso impreciso del usuario.

VIII. Leyes y normativas aplicables

Ley de protección de Datos

La aplicación está diseñada para que los usuarios puedan crear cuentas y guardar sus portafolios de criptomonedas. Los datos del usuario, incluido el email, se consideran datos personales por la normativa vigente. Mi aplicación cumple con el Reglamento General de Protección de Datos (GDPR):

El **Reglamento General de Protección de Datos (GDPR)** es la normativa de la Unión Europea que regula el tratamiento de datos personales de las personas físicas. En vigor desde mayo de 2018, establece estándares para garantizar la privacidad y seguridad de los datos:

- **Consentimiento:** Los datos personales solo pueden ser tratados si el usuario da su consentimiento claro y explícito.
- **Derechos de los usuarios:** Protege derechos como el acceso, rectificación, eliminación (derecho al olvido), portabilidad y oposición al tratamiento de los datos.
- **Transparencia:** Exige a las empresas informar claramente a los usuarios sobre cómo se recopilan, usan y almacenan sus datos.
- **Seguridad:** Obliga a implementar medidas para proteger los datos personales y notificar violaciones de seguridad en un plazo máximo de 72 horas.
- **Alcance global:** Se aplica no solo a empresas de la UE, sino también a organizaciones de fuera que manejan datos de ciudadanos europeos.

La aplicación cumple con la normativa española, que complementa al GDPR, para garantizar un manejo adecuado de los datos personales proporcionados.

Se incluye un aviso de privacidad visible dentro de la app, en la pantalla de usuarios, donde se informa que se recogerán los datos y que el usuario acepta otorgarlos si se registra.

También se mencionan los derechos que pueden ejercerse respecto a la información personal.

Estas acciones aseguran el cumplimiento de las leyes de privacidad aplicables.

IX. Versiones de la aplicación

En el proceso de creación de mi aplicación y con el uso de Git en GitHub cree un repositorio de código y varias versiones de la aplicación así como muchos commits.

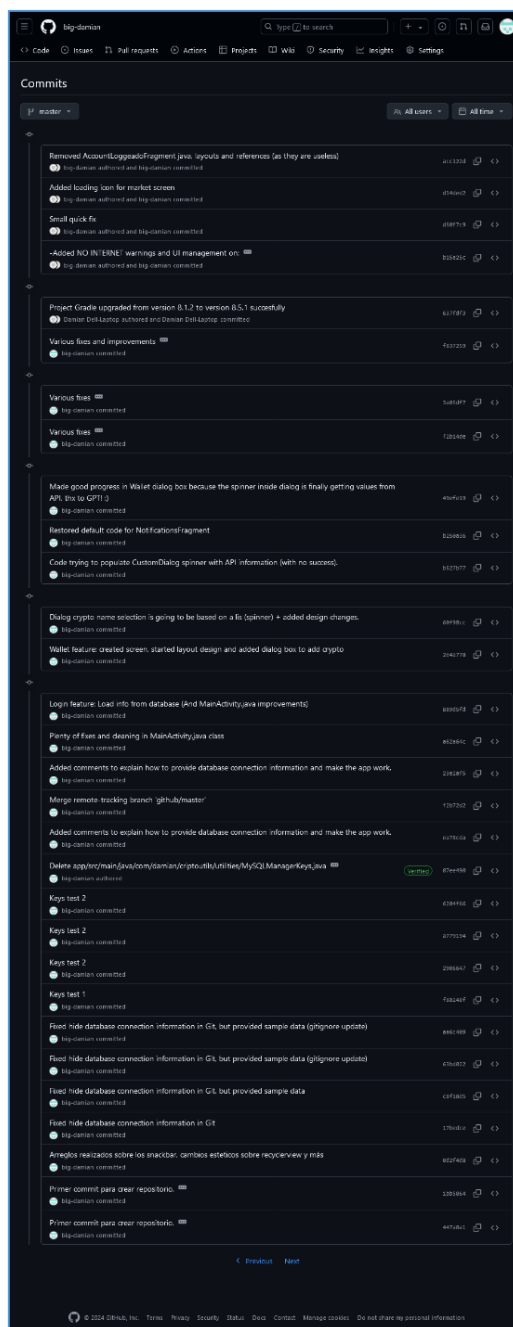


Figura 7. Lista de primeros commits en GitHub

Con motivo de que este proyecto pueda quedar en mi portfolio profesional en un futuro en GitHub está en su mayoría escrito en inglés, aunque aquí lo pondré en español.

He seguido una estructura de versiones mayores (las que suponen actualizaciones muy importantes) y versiones menores (que incluyen cualquier otra actualización), siendo las versiones mayores el primer dígito empezando en cero y las versiones menores el segundo dígito.

Mi primera versión fue la 0.50, siendo esta y las siguientes versiones preliminares hasta llegar a la versión 1.0 que consideraría la versión final. A partir de esta serían todas las actualizaciones menores sobre la versión final para corregir errores, mejorar la aplicación o añadir más funcionalidades.

- **Versión 0.50 – “Primera versión, funcionalidad básica”**

Descripción: Aplicación aún en desarrollo, funcionalidad básica lograda. 3 de 4 pantallas ya se ven como si se les deba, pero todavía carece de características (login, por ejemplo). Comienzo con Git.

- **Versión 0.51 – “Función de inicio de sesión: Perfil de las cargas de información de la base de datos al iniciar sesión”**

Descripción: La información del perfil ahora se carga desde la base de datos al iniciar la creación de código (también de limpieza y optimizaciones de código).

- **Versión 0.52 – “Empezado con la pantalla cartera”**

Descripción: La selección de nombres de criptos en el cuadro de dialogo se va a basar en una lista (spinner)...

- **Versión 0.53 – “Buenos progresos en la pantalla de cartera”**

Descripción: La pantalla de cartera tiene un cuadro de diálogo con un Spinner que ahora carga datos de la API.

- **Versión 0.54 – “Lanzamiento de versión de sólo código (Arreglos y mejoras)”**

Descripción: Añadidas múltiples correcciones y mejoras.

- **Versión 0.55 – “Comportamiento para cuando no hay internet y muchas mejoras visuales”**

Descripción: Manejadas situaciones en las que no hay internet en cada pantalla y muchas mejoras visuales (Y añadir número de versión en cada nombre de Release de GitHub).

- **Versión 0.56 – “CASI TERMINADO, la pantalla mercado muestra los detalles de cada moneda, la pantalla de cartera guarda los datos de forma persistente”**

Descripción: Arreglado la función de registro en la pantalla de login (Cargando datos desde la base de datos).



- **Versión 0.57 – “Empezada implementación de característica ‘actualizar datos del usuario’, grandes avances visuales y varios arreglos”**

Descripción: Actualizados los TODO's para mejor auto-organización.

- **Versión 0.58 – “Probablemente la última versión preliminar! Todas las pantallas funcionan como deberían”**

Descripción: Funcionalidad de la pantalla Mis Criptos (cartera) terminada (el botón de eliminar funciona!).

- **Versión 1.0 – “TFG terminado! Todo funcionando y todas las características previstas inicialmente añadidas”**

Descripción: TFG terminado! Incluso aunque esta es la versión final 1.0, todavía pretendo hacer unos cuantos cambios que tienen sentido como hacer que "Mis Criptomonedas" o pantalla cartera se guarde en MySQL asociado a tu cuenta de la aplicación, así como hacer cambios en la pantalla inicial para hacerla más bonita.

- **Versión 1.1 – “Backup en la nube y restauración implementado para Mis Criptos”**

Descripción: Ahora puedes guardar en la nube y restaurar tus criptos asociadas a tu cuenta.

- **Versión 1.2 – “Profesionalizados los iconos y añadidas correcciones para la entrega del TFG”**

Descripción: Basado en “Nuevo commit para la entrega del TFG”

- **Versión 1.3 – “Agregada política de privacidad y restaurados iconos de login para la entrega del TFG”**

Descripción: Última versión con los retoques finales. Basada en el último commit para la entrega del TFG

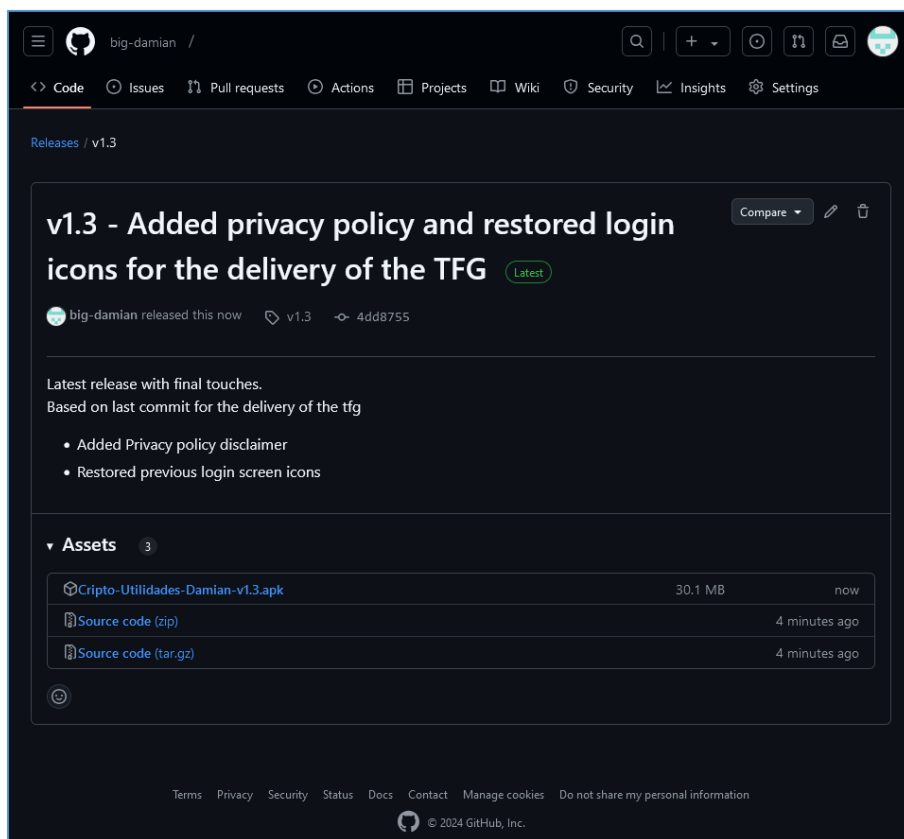


Figura 8. Última versión lanzada en GitHub

X. Requisitos funcionales (Casos de uso)

En esta sección se definen los casos de uso de la aplicación.

1. Un usuario se registra en el sistema

El usuario accede a la aplicación y selecciona la opción de "registrarse".

Deberá ingresar su correo electrónico y crear una contraseña segura.

La aplicación valida los datos y registra la cuenta del usuario en la base de datos MySQL.

Si el proceso es exitoso, se muestra un mensaje de confirmación y el usuario es redirigido a la pantalla de inicio de sesión.

2. Un usuario inicia sesión

El usuario abre la aplicación e ingresa su correo electrónico y contraseña en el panel



de inicio de sesión.

La aplicación verifica las credenciales con los datos almacenados en la base de datos MySQL.

Si la autenticación es correcta, el usuario accede a la pantalla principal de la aplicación, donde puede visualizar su monedero y los datos del mercado.

3. Un usuario edita los datos de su perfil

El usuario accede al apartado de "Cuenta" en el menú de la aplicación.

En esta sección, se despliega un formulario con los datos actuales del usuario, como el correo electrónico y la contraseña.

Presionará el botón "Actualizar datos" para empezar la edición.

El usuario puede modificar uno o más campos, como cambiar su dirección de correo o actualizar su contraseña.

Al finalizar, selecciona la opción "Actualizar datos".

La aplicación valida los datos y actualiza la información del usuario en la base de datos MySQL en la nube.

Una vez completada la actualización, se muestra un mensaje de confirmación indicando que los cambios han sido guardados correctamente.

4. Un usuario cierra sesión

El usuario accede al menú principal de la aplicación.

Selecciona la opción de "cerrar sesión".

La aplicación elimina las credenciales almacenadas en la sesión activa y redirige al usuario a la pantalla de inicio de sesión.

5. Un usuario añade una criptomoneda al monedero

El usuario accede al apartado de "Lista de Criptomonedas".

Selecciona una criptomoneda de la lista obtenida a través de la API de CoinGecko.

La aplicación muestra un formulario donde el usuario puede ingresar la cantidad de criptomonedas que posee.

Al confirmar, la criptomoneda se agrega al monedero del usuario y los datos se guardan en la base de datos SQLite local.

6. Un usuario edita una criptomoneda en su monedero

El usuario accede al apartado "Monedero" donde puede ver las criptomonedas que ha añadido.



Selecciona la criptomoneda que desea modificar.

La aplicación despliega un formulario donde el usuario puede cambiar la cantidad de la criptomoneda en su monedero.

Al guardar los cambios, la aplicación actualiza los datos tanto en la base de datos SQLite local como en la base de datos MySQL en la nube si el usuario está sincronizado.

7. Un usuario elimina una criptomoneda del monedero

El usuario accede al apartado "Monedero" y selecciona una criptomoneda que desea eliminar.

La aplicación solicita confirmación antes de proceder.

Una vez confirmada, la criptomoneda se elimina del monedero del usuario, y los datos se actualizan en la base de datos local y en la nube si la sincronización está habilitada.

8. Un usuario respalda su monedero en la nube

El usuario accede a la opción de sincronización en el menú de la aplicación.

Selecciona la opción "Respalda en la nube".

La aplicación sube los datos del monedero a la base de datos MySQL en la nube.

Una vez completado, se muestra un mensaje de éxito indicando que los datos han sido respaldados correctamente.

9. Un usuario restaura su monedero desde la nube

El usuario accede al menú de la aplicación y selecciona "Restaurar desde la nube".

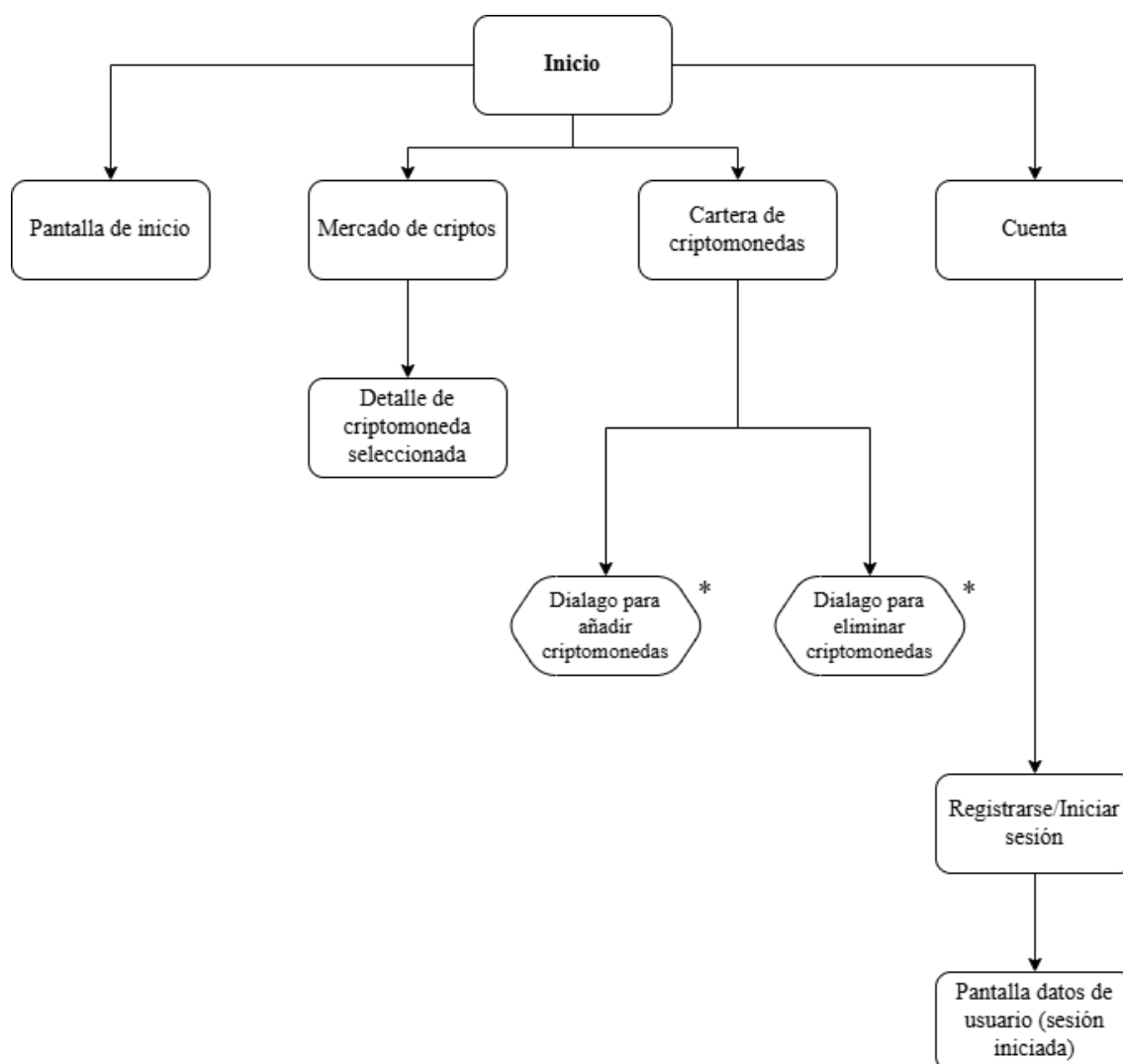
La aplicación verifica la existencia de un respaldo previo y descarga los datos almacenados en la base de datos MySQL.

Los datos del monedero se actualizan en la base de datos SQLite local y se muestra un mensaje confirmando la restauración exitosa.

XI. Diseño de pantallas y explicación básica de funcionalidades

1. Diagrama de flujo de pantallas

A continuación, se muestran las diferentes pantallas de la aplicación y una breve explicación de su funcionamiento. En primer lugar, se muestra un diagrama en el que se define el flujo de las pantallas.



* Los bloques hexagonales indican que no se trata de pantallas completas sino cuadros de dialogo

Figura 9. Diagrama de flujo de las pantallas de la aplicación

2. Pantalla de Inicio

La pantalla de inicio de la aplicación muestra un diseño centrado en la criptomoneda Bitcoin y da la bienvenida a la aplicación.



Figura 10. Pantalla de inicio de la aplicación

En la parte superior, se encuentra el título “Inicio”, indicando que esta es la página principal de la aplicación.

En el centro de la pantalla, hay una representación gráfica animada para hacer la pantalla de inicio más agradable a la vista.

Debajo de los elementos gráficos, hay un texto que da la bienvenida al proyecto de fin de curso.

Más abajo se muestra el precio actual de Bitcoin en euros, actualizado a través de la API o el último valor conocido en caso de ejecutar la aplicación sin conexión a internet.

Debajo hay un botón dedicado a actualizar el precio de Bitcoin manualmente, el cual mostrará mensajes tipo *Snackbar* cuando se presione para indicar el estado de la actualización

En la parte inferior de la pantalla, hay iconos de navegación típicos de una interfaz de aplicación. Cada de estos iconos estará resaltado, indicando cual es la vista actual. Este flujo de pantallas está diseñado para ofrecer una experiencia sencilla y rápida para el usuario, permitiéndole acceder fácilmente a todas las funcionalidades de la aplicación criptomonedas.

3. Pantalla de mercado y detalle de criptomonedas

3.1. Pantalla de mercado crypto

La función principal de esta pantalla es ver la lista completa de criptomonedas disponibles y su información más importante.

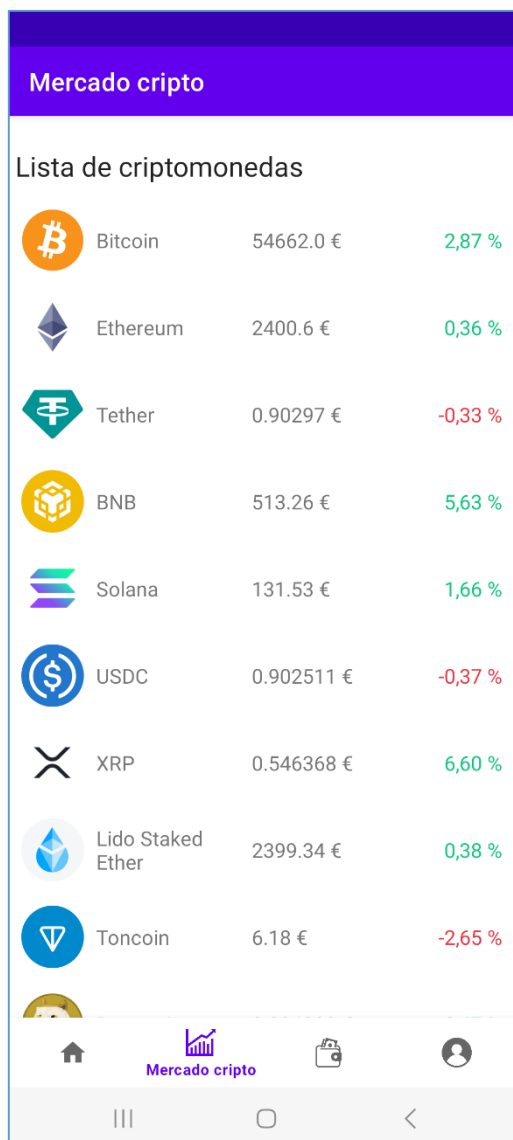


Figura 11. Captura de pantalla de mercado de criptomonedas

En el centro de la pantalla, hay una lista de criptomonedas creada con un RecyclerView, con sus respectivos nombres, iconos, precios en euros y el porcentaje de cambio en verde o rojo para indicar si ha subido o bajado.

Los datos de las monedas se obtienen mediante una API desde la que recibo la información, que luego cargo en un RecyclerView. Para los íconos, recibo las URLs de las imágenes que se almacenan en un servidor público y las cargo con una librería llamada Glide. Esta librería carga dinámicamente los iconos

en el RecyclerView a medida que se cargan, lo que ayuda a ahorrar recursos del dispositivo.

La barra de menú de la aplicación, como siempre, aparece en la parte inferior.

En esta pantalla podemos hacer clic en cualquier elemento de la lista para acceder a la siguiente pantalla, que es la sección de detalles de criptomonedas, donde veremos la información completa de la criptomoneda. Detallaré dicha pantalla a continuación.

3.2. Pantalla detalle de criptomonedas

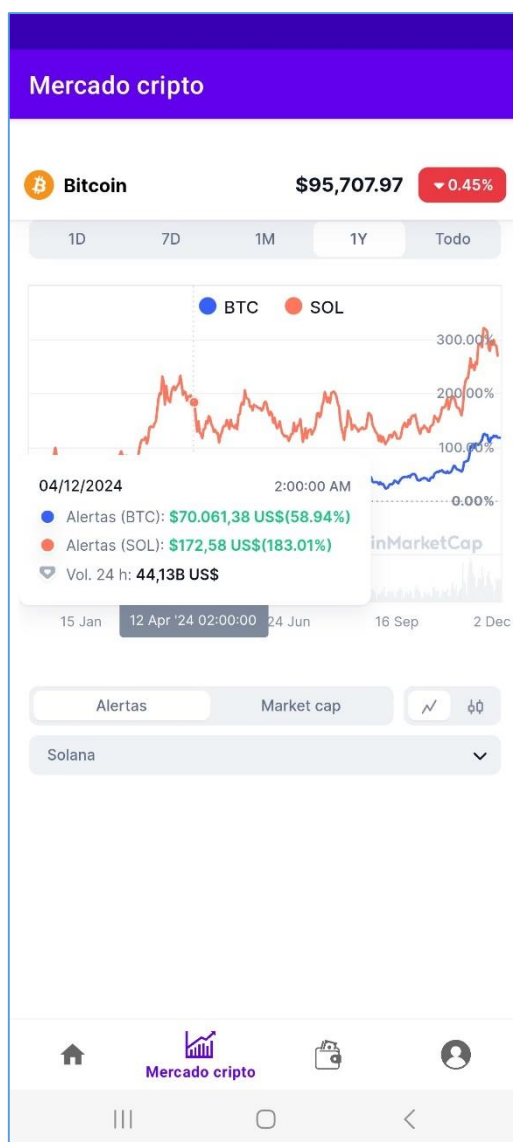


Figura 12. Captura de pantalla de detalle de una criptomoneda del mercado de criptomonedas

En esta pantalla encontraremos la información principal de la moneda seleccionada en la pantalla anterior. Veremos un gráfico interactivo con el precio de la moneda, la opción para ver la capitalización de mercado de la misma, y múltiples opciones de tiempo para ajustar el gráfico, tanto en línea como en velas japonesas.

También se ofrece la opción de comparar con otras criptomonedas a través de una lista.

Toda esta información se obtiene gracias a una conexión a ‘CoinMarketCap’.

4. Pantalla cartera de criptomonedas

4.1. Vista principal y lista de criptomonedas

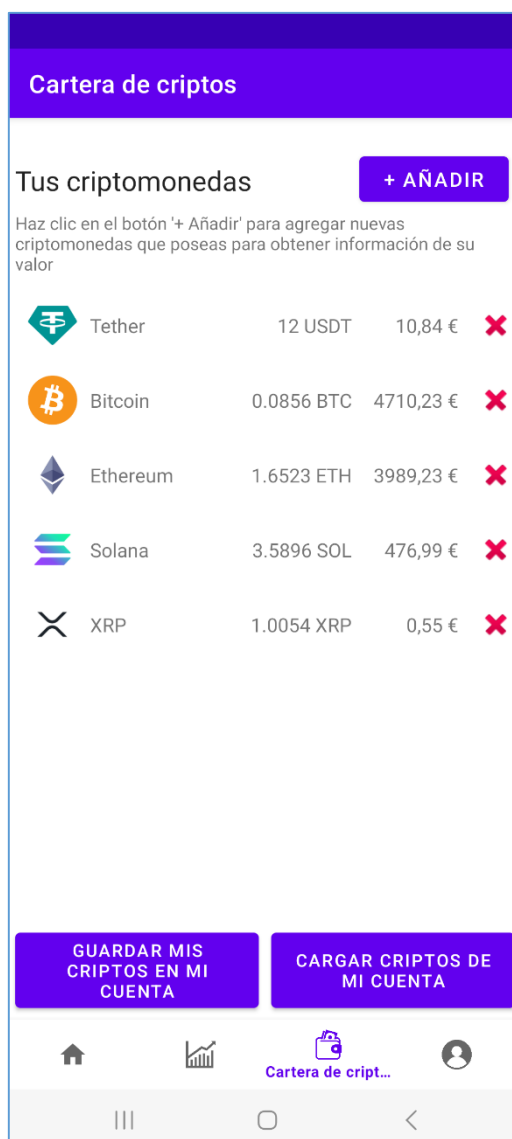


Figura 13. Captura de pantalla de detalle de una criptomoneda del mercado de criptomonedas

En esta pantalla, el usuario puede añadir las criptomonedas que posee en la plataforma que sea y hacer un seguimiento de su valor.

En la parte superior de la pantalla, tenemos el texto introductorio y el botón de "Añadir", el cual abrirá el diálogo para la adición de criptomonedas, que detallaremos en el siguiente punto. En el centro de la pantalla veremos un RecyclerView con las monedas añadidas por el usuario.



Cada fila representa un registro y consta del ícono, cargado de la misma forma que en la pantalla de mercado de criptomonedas, así como el nombre, el número de unidades de moneda y el precio.

Al final de cada fila hay una "X" roja que se puede presionar para eliminar la criptomoneda de la lista.

En la parte inferior de la pantalla, tenemos dos botones: uno para guardar las monedas en la cuenta del usuario (hacer un backup en la nube) y otro para cargar las monedas desde la cuenta del usuario (recuperar el backup desde la nube).

Las monedas agregadas a la lista se almacenarán en la memoria de la aplicación mediante SQLite, por lo que surge la necesidad de hacer un backup de las mismas. Al pulsar el botón de "Guardar en la nube", se recorren todas las monedas de la tabla de criptomonedas del usuario en la base de datos interna de la app y se envían a una tabla de la base de datos en la nube asociada al usuario. Si ya existe un registro en la base de datos para una criptomoneda, este se elimina.

Al cargar las monedas desde la base de datos, se realiza un proceso casi idéntico pero inverso. Es decir, se leen las monedas existentes para el usuario en la base de datos y se cargan en la aplicación; si existe un registro para una moneda en particular, se sobrescribe.

4.2. Dialogo de adición de criptomonedas de usuario

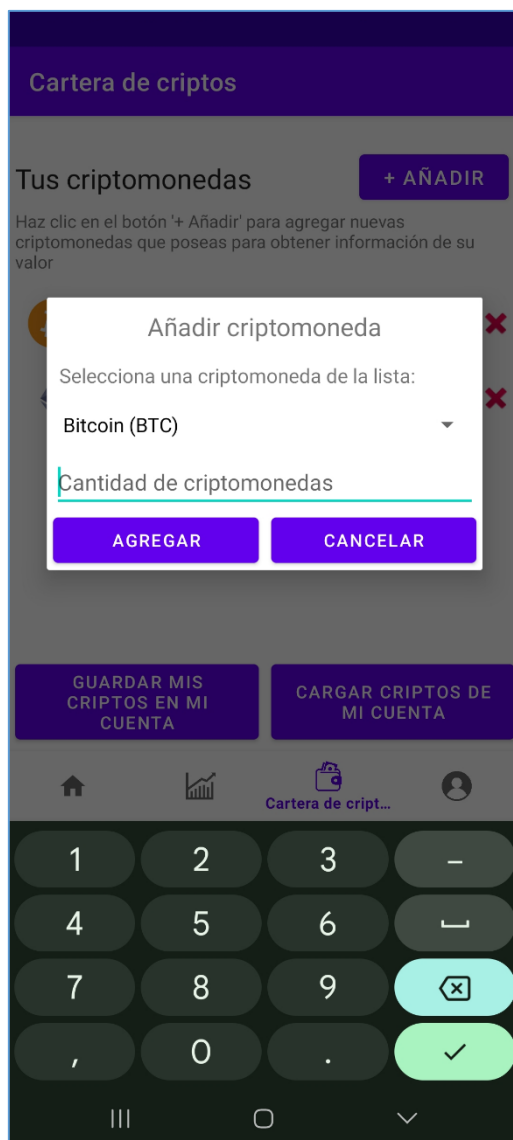


Figura 14. Captura de pantalla de la ventana de dialogo para añadir criptomonedas a la cartera de criptomonedas con el teclado numérico abierto

Este cuadro de diálogo se abre cuando en la pantalla de cartera se presiona el botón "Añadir". En el cuadro de diálogo veremos varios elementos:

Primero, el título y un breve texto explicativo. Luego, encontraremos un 'Spinner', también conocido como 'ComboBox'.

Este selector desplegable se puebla con los nombres y símbolos de las monedas actualizados, ya que se obtienen también de la API, y se ordenan por importancia/popularidad para que se puedan encontrar con facilidad.



Debajo, tenemos un campo de texto simple en el cual el usuario introducirá la cantidad de monedas que posee.

Este campo está programado para aceptar solo números y un punto; solo se podrá introducir un número entero o un valor decimal o 'float', además hay una validación extra: el valor debe ser mayor a cero y no puede estar vacío para que se guarde. Si la condición no se cumple y se presiona el botón de 'Agregar', se mostrará un mensaje tipo Toast mencionando la condición que falta por cumplir, para que el usuario pueda corregir el valor introducido.

5. Pantalla de cuenta de usuario (inicio de sesión, registro y detalles de la cuenta)

5.1. Pantalla de inicio de sesión y registro

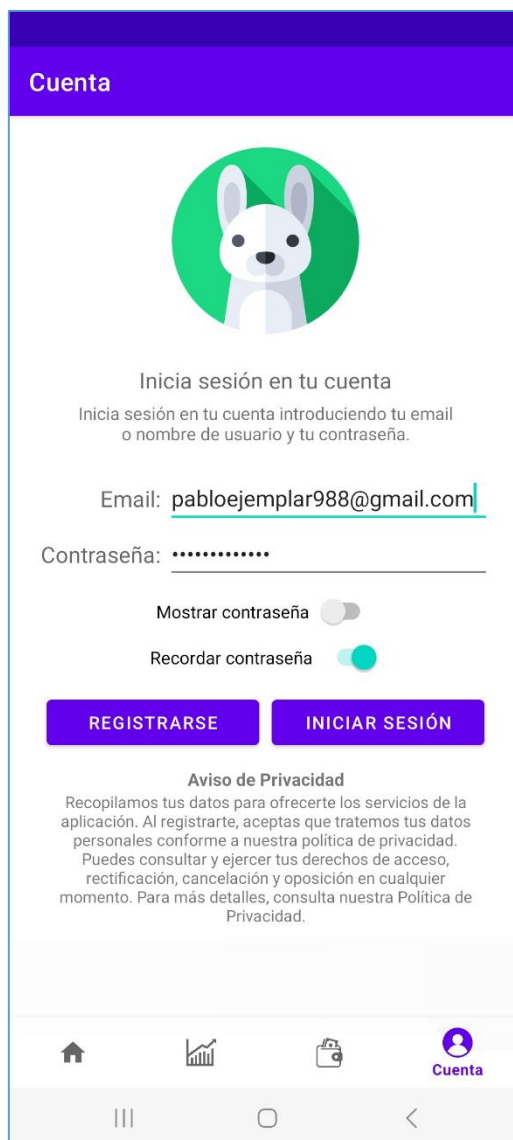


Figura 15. Captura de pantalla de la primera parte de la última pantalla, el inicio de sesión/registro

En esta pantalla, el usuario puede registrar una nueva cuenta en la aplicación o iniciar sesión en una cuenta existente, lo que le dará acceso a la información de su cuenta y a sus criptomonedas guardadas.

En esta pantalla se presentan varios elementos:

Primero, se muestra una imagen que representa un usuario genérico, seguida de textos que introducen y explican brevemente al usuario cómo usar el sistema de inicio de sesión.



Debajo, se encuentra el campo para ingresar el email o nombre de usuario, seguido por el campo de contraseña.

Bajo los campos de texto, hay dos interruptores (switches) con funcionalidades específicas:

Mostrar contraseña: Permite, como su nombre lo indica, visualizar el texto que se está escribiendo en el campo de contraseña en lugar de los puntos.

Mantener sesión iniciada: Este interruptor asegura que, si el usuario inicia sesión o se registra con éxito, su sesión permanecerá activa incluso si cierra y vuelve a abrir la aplicación.

El mecanismo de inicio de sesión funciona enviando a la base de datos el nombre de usuario o correo electrónico introducido junto con la contraseña, y verificando que exista en la tabla de usuarios.

Para el proceso de registro, se comprueba que no exista previamente ningún usuario en la tabla de la base de datos (BBDD) con el mismo correo electrónico o nombre de usuario. Si no se encuentra, se crea un nuevo usuario y se aprueba el inicio de sesión.

Si existe un registro para el usuario en la base de datos, el intento de inicio de sesión se da por válido. A su vez, si el inicio de sesión se da por válido y la función "Recordar contraseña" está activada, se guardan en la base de datos interna de la aplicación (SQLite) los datos y una 'flag' de que el usuario ha iniciado sesión correctamente y con esto, si el usuario sale de la aplicación y vuelve a entrar, el proceso de login se realiza nuevamente de forma automática.

5.2. Pantalla detalle de datos de la cuenta

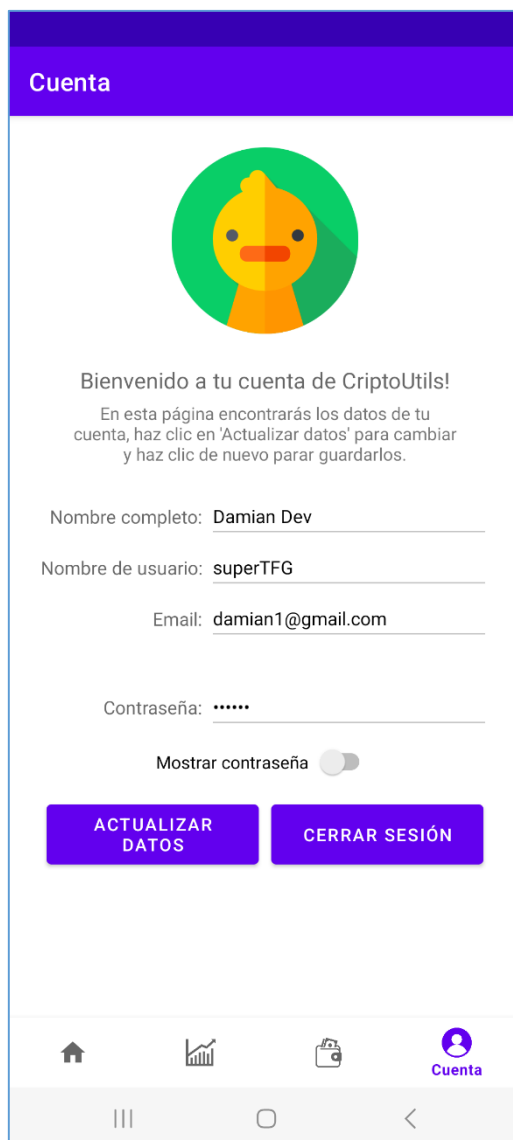


Figura 16. Captura de pantalla de la última pantalla, la pantalla de datos del usuario

En esta pantalla encontramos varios elementos similares a la pantalla anterior, junto con otros datos del usuario.

Primero, se muestra un ícono de usuario. Luego, hay un texto introductorio que explica la funcionalidad de esta pantalla y el proceso de actualización de los datos del usuario, que también detallaremos aquí.

Debajo de este texto, se encuentran los siguientes campos:

- **Nombre completo del usuario.**



- **Nombre de usuario.**
- **Correo electrónico del usuario**, que será el mismo utilizado para iniciar sesión.
- **Contraseña.**

Debajo del último campo de texto, se incluye un interruptor (switch) para mostrar u ocultar la contraseña.

Finalmente, hay dos botones:

- Un botón para actualizar o modificar los datos del usuario.
- Un botón para cerrar la sesión del usuario.

Proceso de actualización de datos:

Al pulsar el botón "Actualizar datos", este comenzará a parpadear, indicando que se ha activado el modo de edición. Los campos de texto que contenían los datos del usuario, anteriormente de solo lectura, estarán habilitados para su modificación (excepto el campo de correo electrónico, ya que, al ser la clave primaria de identificación, no puede modificarse).

Después de realizar los cambios deseados, se debe pulsar nuevamente el botón "Actualizar datos". Este se desactivará brevemente mientras los datos ingresados se envían y se guardan en la base de datos.

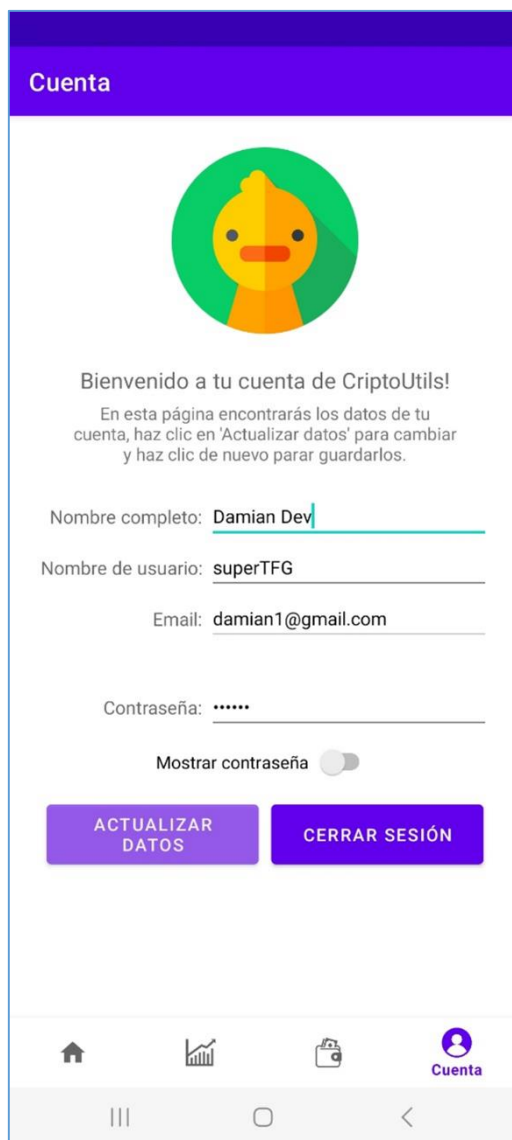


Figura 17. Captura de pantalla de la pantalla de datos del usuario en el modo actualización de datos, con los campos de texto desbloqueados y el botón actualizar datos parpadeando

XII. Estructura de datos de la base de datos

Para la implementación de la base de datos de la aplicación, se ha utilizado MySQL, una conocida base de datos relacional SQL. Esta contiene principalmente dos tablas, una tabla llamada **usuarios**, que almacena la información necesaria para gestionar las cuentas de usuario y permitir el inicio de sesión y una tabla de criptomonedas de

usuarios para guardar las criptomonedas que cada usuario tiene guardadas. A continuación, se describe su estructura y los campos que las componen:

1. Tabla de usuarios

Esta tabla (“*usuarios*”) está diseñada para almacenar los datos básicos de los usuarios de la aplicación y permitir el registro e inicio de sesión.

- **id_usuarios (int):** Un identificador único para cada usuario. Este campo es de tipo entero, se genera automáticamente con la opción **AUTO_INCREMENT**, y está definido como clave única (**UNIQUE**).
- **username (varchar(50)):**
Un nombre de usuario personalizado, opcional.
- **email (varchar(100)):**
Dirección de correo electrónico del usuario. Este campo es obligatorio (**NOT NULL**) y se utiliza como clave principal de la tabla (**PRIMARY KEY**) para garantizar que cada email sea único, se utiliza para localizar a cada usuario de forma inequívoca.
- **contraseña (varchar(50)):**
Almacena la contraseña del usuario. Este campo es obligatorio (**NOT NULL**).
- **nombre (varchar(100)):**
Nombre completo del usuario, opcional.
- **fecha_creacion (datetime):**
Fecha y hora en que se creó la cuenta del usuario.

	id_usuarios	username	email	contraseña	nombre	fecha_creacion
1		big_damian	damian@gmail.com	damian	Damián Peña	1000-01-01 00:00:00
3		Ganamoss	borguitaxd@gmail.com	Colegones	Borja Merchan McDonalds	1000-01-01 00:00:00
4		MacarenaProfee	macarena@gmail.com	macarena	Macarena <3	1000-01-01 00:00:00
5			gerry@gmail.com	gerry		1000-01-01 00:00:00
7			daniJones@gmail.com	dani		1000-01-01 00:00:00
8		superTFG	damian1@gmail.com	damian	Damian Dev	1000-01-01 00:00:00
19		yail	yail@gmail.com	kira01	yaiza montero <3	1000-01-01 00:00:00
21		polla	ayllon@gmail.com	ayllon	gthff	1000-01-01 00:00:00
25		criptopablo98	pabloejemplar98@gmail.com	pabloejemplar98	Pablo Romero García	1000-01-01 00:00:00

Figura 17. Representación visual de la tabla de usuarios

2. Tabla de criptomonedas de usuarios

Esta tabla (“*criptomonedas_usuarios*”) está diseñada para almacenar las criptomonedas que los usuarios agregan su cartera y se relaciona con los datos los usuarios, permite el guardado y restauración de los datos desde la nube.

- **id_criptomoneda_usuario (int):**
Un identificador único para cada registro en esta tabla. Es de tipo entero, se genera automáticamente con la opción **AUTO_INCREMENT** y se define como la clave primaria (**PRIMARY KEY**) para garantizar que cada entrada sea única.

- **email_usuario (varchar(100)):**
Correo electrónico del usuario propietario de las criptomonedas. Este campo es obligatorio (**NOT NULL**) y actúa como referencia para relacionar la información de esta tabla con los usuarios de la tabla **usuarios**.
- **nombre_cripto (varchar(45)):**
Nombre de la criptomoneda que el usuario ha registrado, como "Bitcoin" o "Ethereum".
- **simbolo_cripto (varchar(45)):**
Símbolo de la criptomoneda, como "BTC" o "ETH".
- **cantidad_cripto (varchar(45)):**
Cantidad de la criptomoneda registrada por el usuario.

	id_criptomoneda_usuario	email_usuario	nombre_cripto	simbolo_cripto	cantidad_cripto
	73	damian@gmail.com	Bitcoin	BTC	0.00977044
	74	damian@gmail.com	Ethereum	ETH	16
	75	damian1@gmail.com	Tether	USDT	12
	76	damian1@gmail.com	Bitcoin	BTC	0.0856
	77	damian1@gmail.com	Ethereum	ETH	1.6523
	78	damian1@gmail.com	Solana	SOL	3.5896
	79	damian1@gmail.com	Polkadot	DOT	1
	80	pabloejemplar98@gmail.com	Bitcoin	BTC	0.025
▶	81	pabloejemplar98@gmail.com	Solana	SOL	10
*	NULL	NULL	NULL	NULL	NULL

Figura 18. Representación visual de la tabla de criptomonedas de usuarios

XIII. Pruebas y resultados

A continuación, se mostrarán unos ejemplos del uso de la aplicación con los resultados obtenidos.

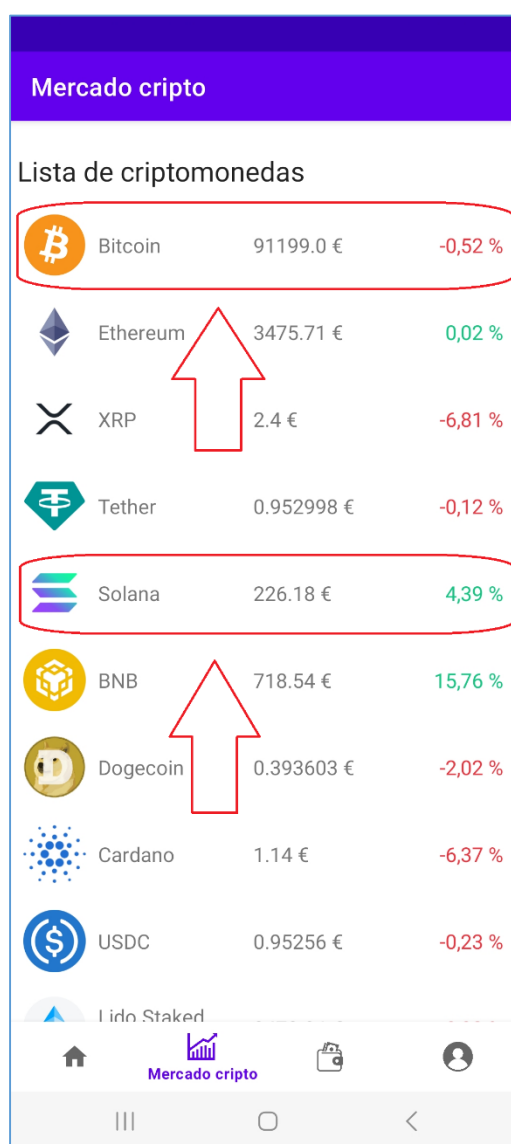
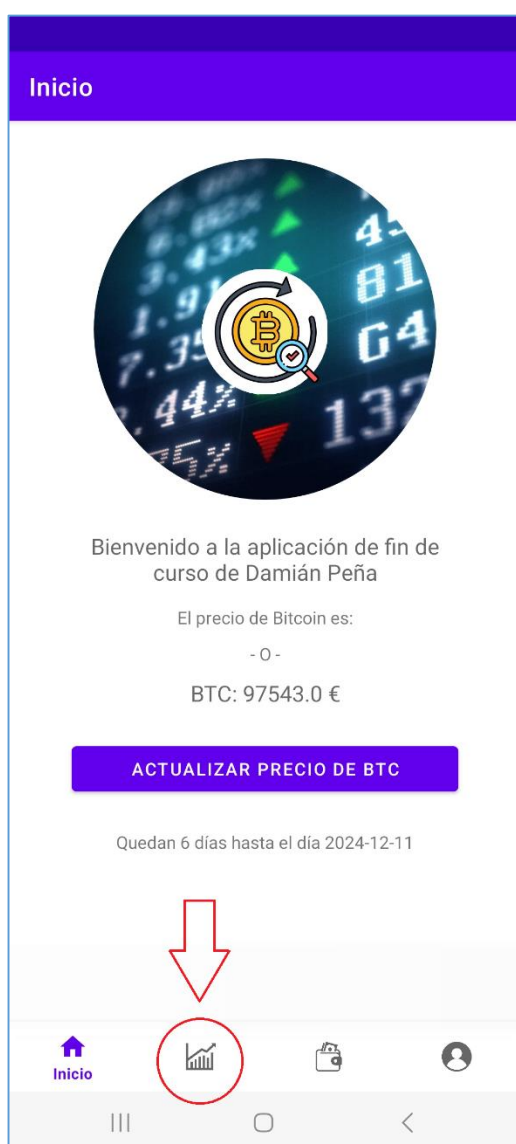
1. Prueba de uso correcto de las funcionalidades de la aplicación

Para el primer ejemplo mostraremos un uso correcto y sin errores de la aplicación (ya que daremos por hecho que disponemos de conexión a internet y el usuario ya tiene todos los datos correctos para usar la aplicación), accederemos como un nuevo usuario y consultaremos el rendimiento de un par de monedas para después añadirlas a la cartera y poder hacer un seguimiento de su valor, además el usuario se registrará en la aplicación, completará todos los datos de su perfil y hará una copia de seguridad de sus datos.

Para este ejemplo tomaremos al inversor ‘Pablo’, que compró en diferentes lugares las criptomonedas “Bitcoin” y “Solana”. En cantidades de 0.025 unidades de Bitcoin y 10 unidades de Solana.

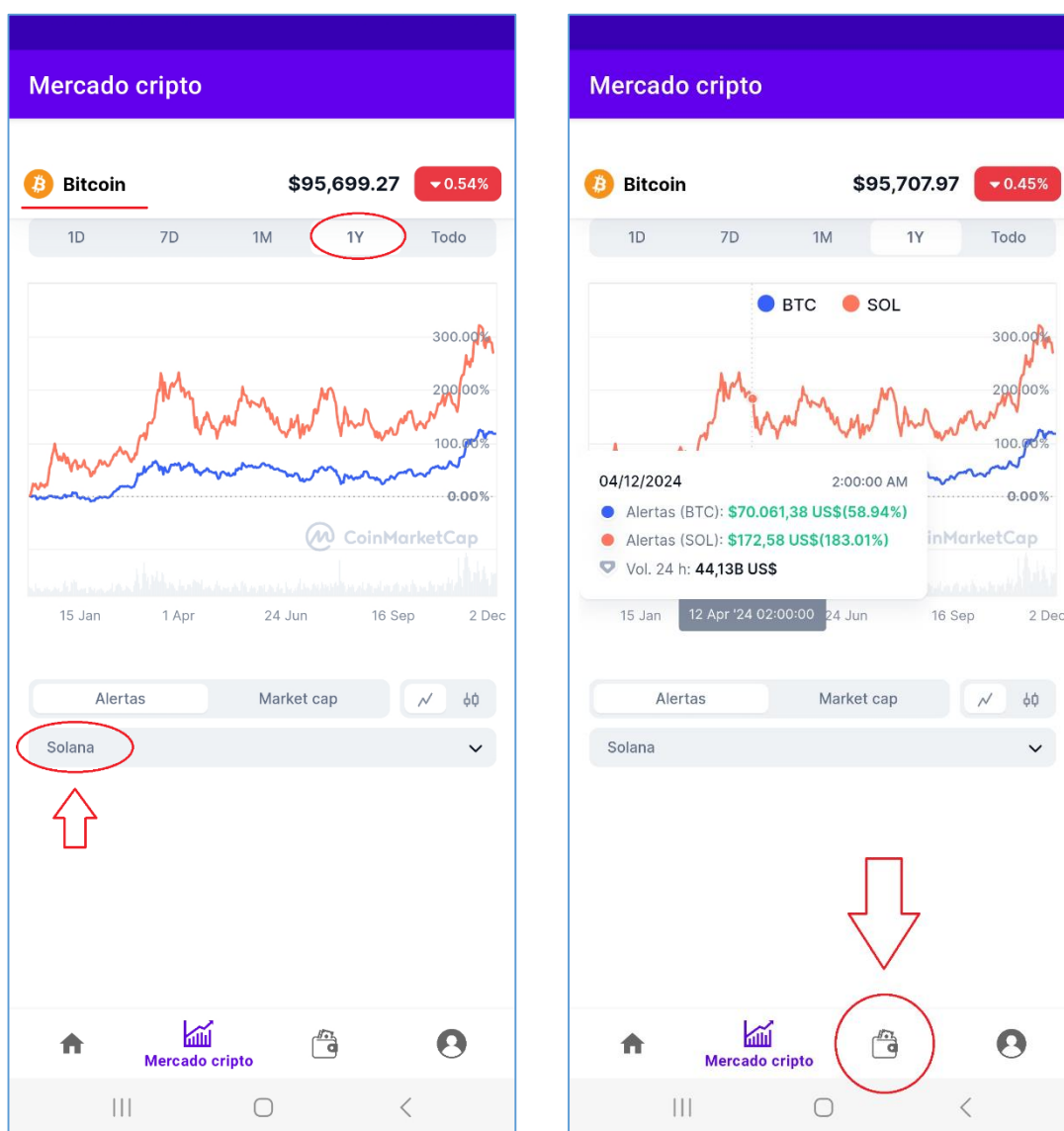
Tras abrir la aplicación, se abrirá la pantalla principal y haremos clic en el icono del menú de navegación inferior ‘Mercado crypto’.

Una vez hecho esto se nos mostrará la lista de todas las criptomonedas actualmente existentes en el mercado.



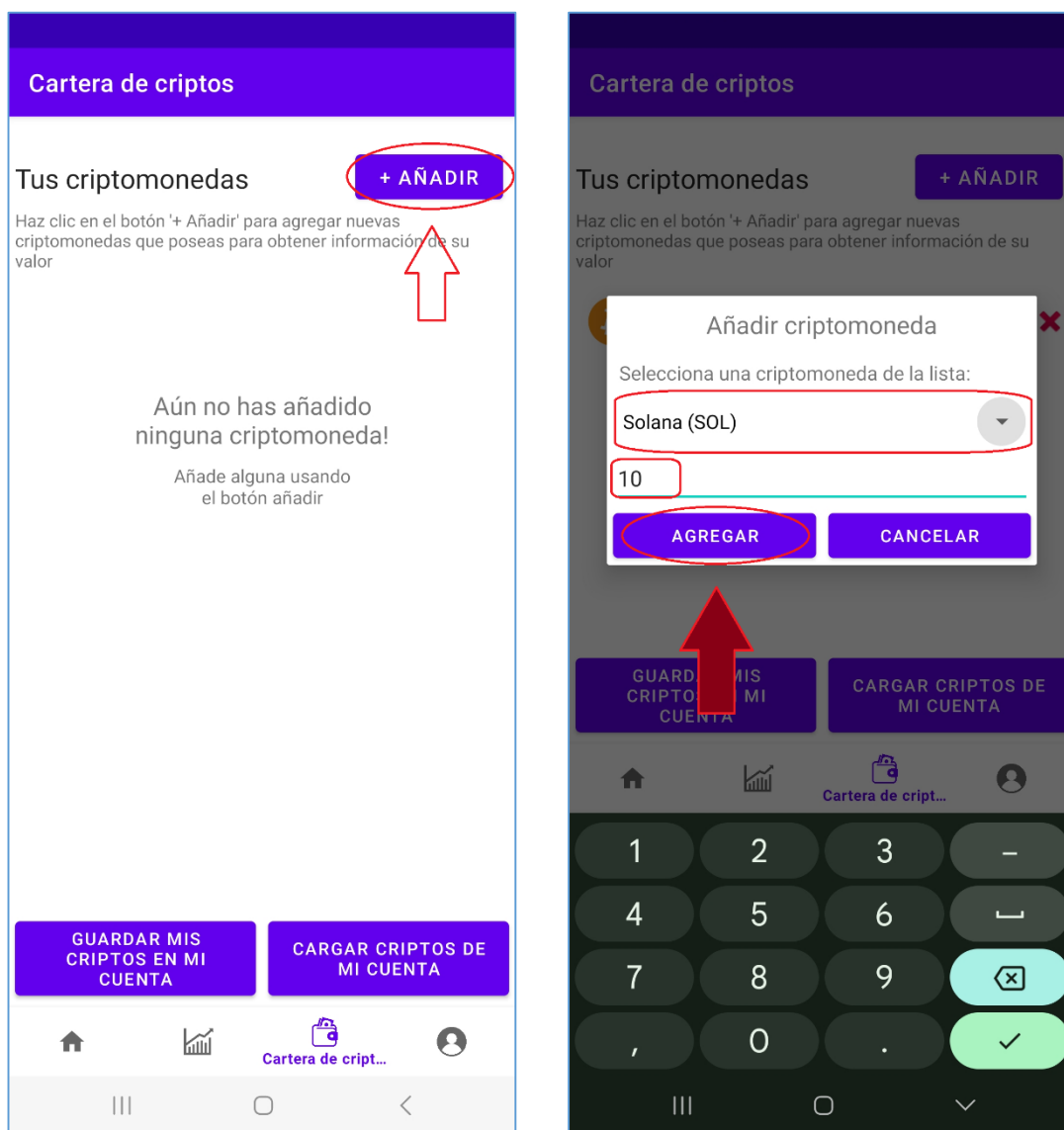
En este caso al ser dos criptomonedas de mucha importancia nos aparecen sin necesidad de bajar en la lista, accederemos primero a los detalles de la moneda Bitcoin haciendo clic sobre el elemento del RecyclerView.

Se nos abrirá la pantalla de detalles de criptomonedas y allí el usuario consultará toda la información de que precise, en este caso compararemos el crecimiento anual de una criptomoneda con la otra con la función comparar.



Hecho esto accederemos a la pantalla de cartera. Pasaremos directamente usando el menú inferior de navegación.

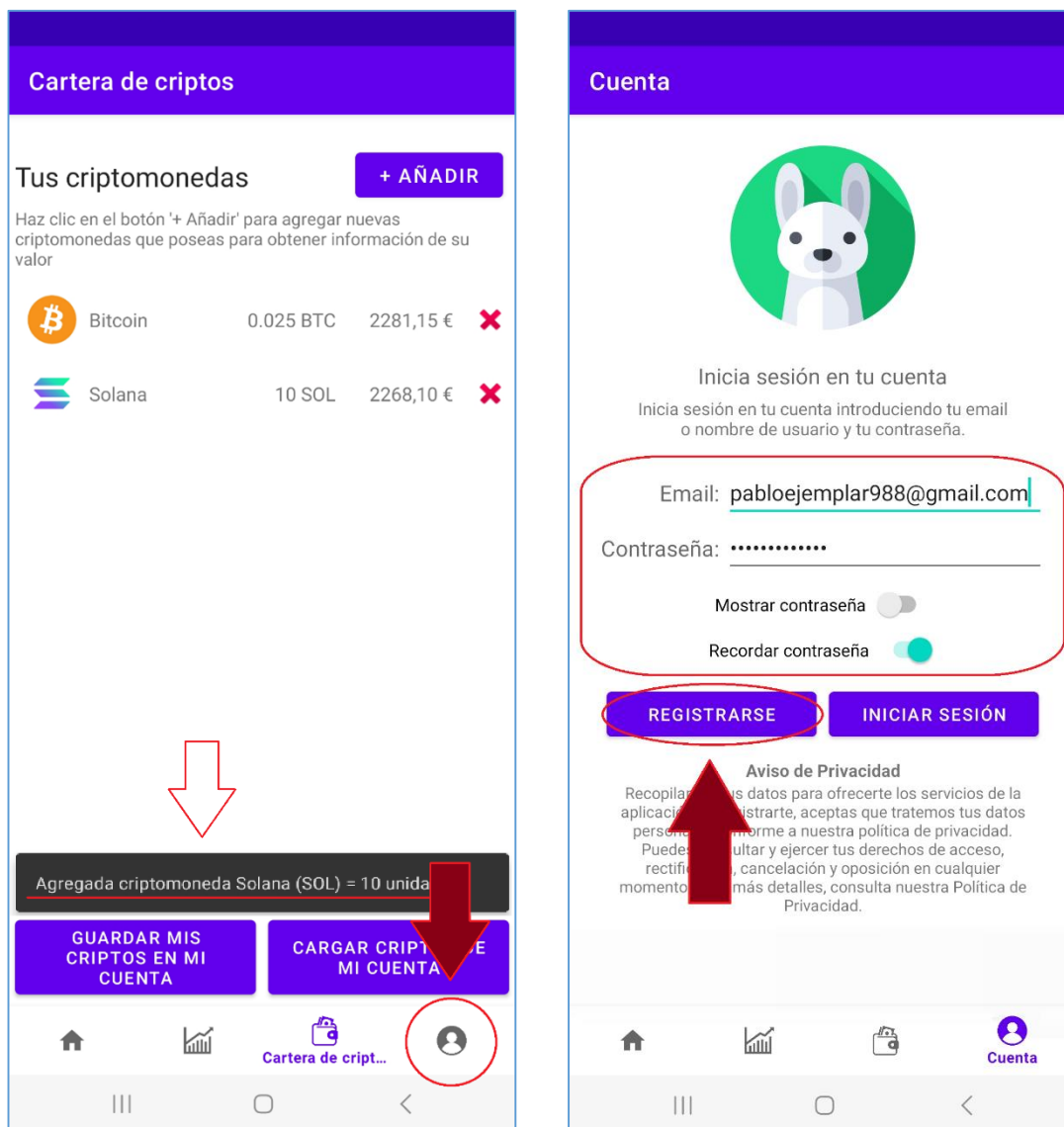
En la pantalla de cartera veremos la lista inicialmente vacía. Añadiremos las criptomonedas que tiene el inversor Pablo pulsando en el botón ‘Añadir’.



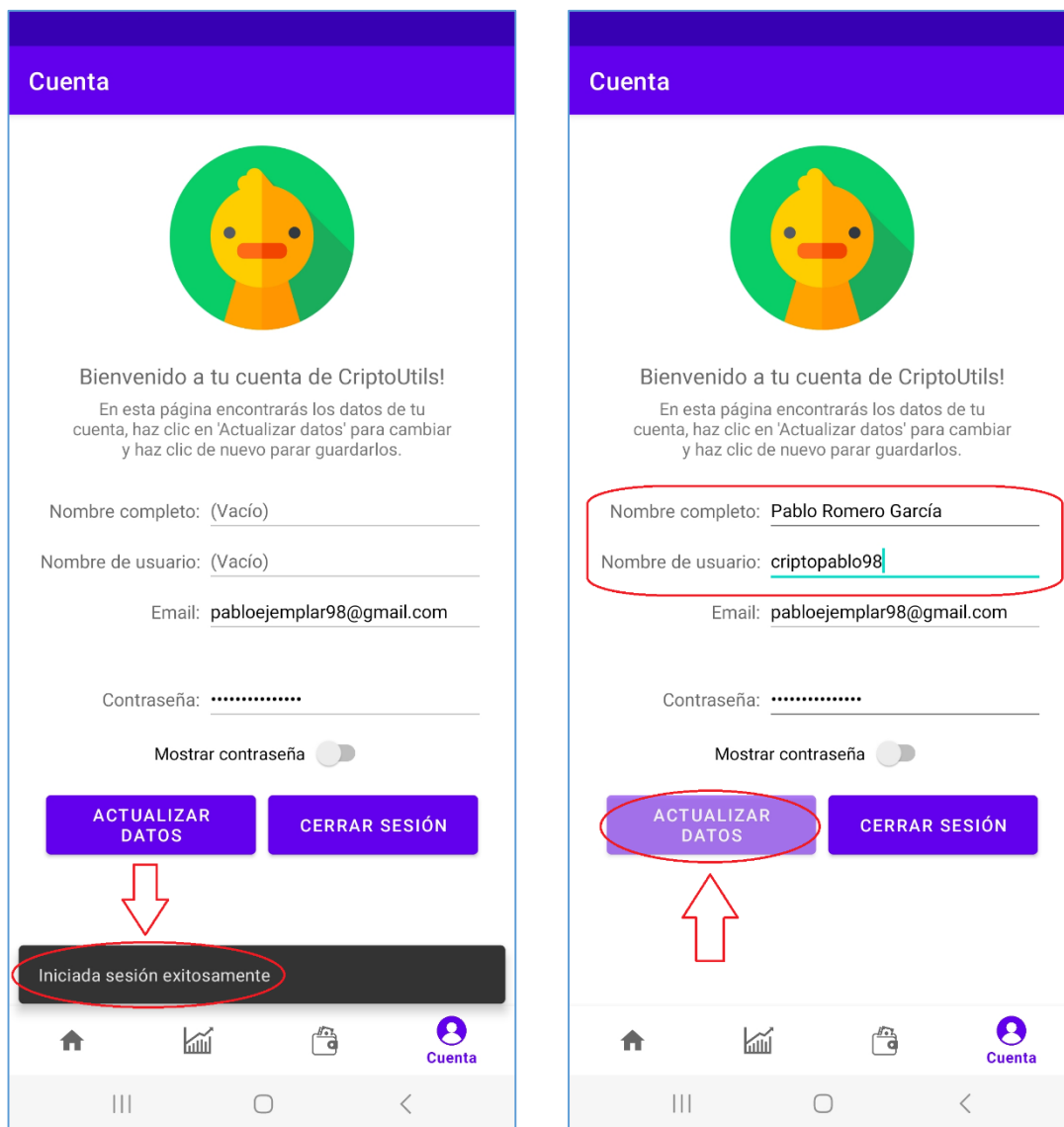
Una vez añadidas las dos monedas veremos la lista completa y el valor de las mismas en euros. Ahora el usuario podrá acceder de forma sencilla a su portfolio unificado en un solo lugar y ver su valor.

Continuaremos pasando a la pantalla ‘Cuenta’. Haremos nuevamente clic en el icono correspondiente del menú de navegación.

En la pantalla de ‘Cuenta’ introduciremos los datos del inversor Pablo y procederemos a registrarlo en la aplicación



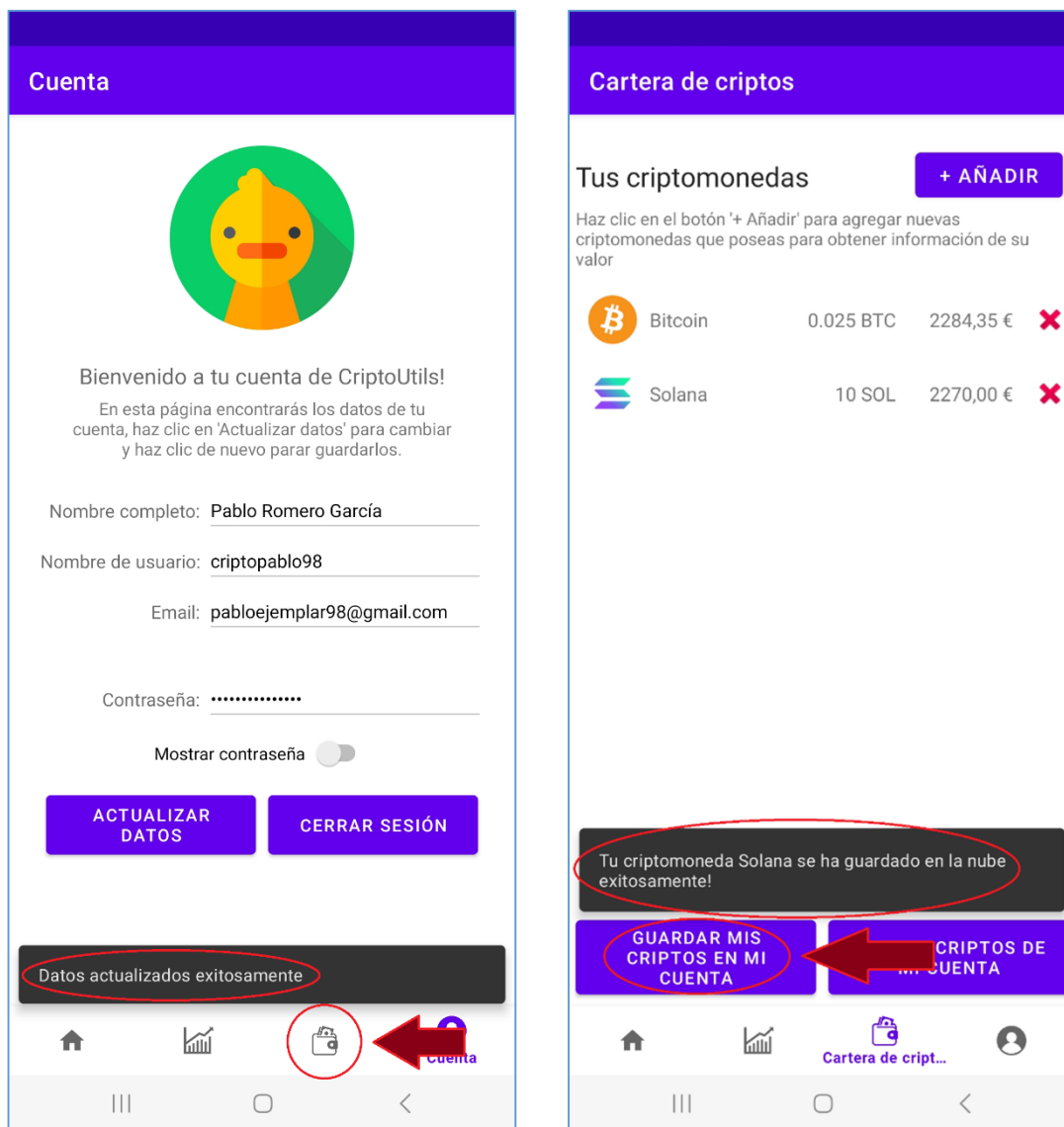
Una vez el usuario ha introducido el correo y la contraseña, que en este caso son “pabloejemplar98@gmail.com” y “pabloejemplar98” respectivamente, procederemos a clicar en el botón ‘Registrar’ para registrar al usuario. El usuario podrá usar el switch de mostrar contraseña y recordar contraseña si así lo desea.



Los datos del nuevo usuario se comprobarán y la aplicación dará acceso al usuario a la pantalla de su nueva cuenta. Al ser un nuevo usuario, los campos ‘Nombre completo’ y ‘Nombre de usuario’ estarán vacíos. El siguiente paso será introducir dichos datos.

Haciendo clic en el botón ‘Actualizar datos’ entraremos en el modo de edición de los datos del usuario e introduciremos el nombre completo de Pablo: “Pablo Romero García” y su nombre de usuario: “criptopablo98”. Una vez introducido todo se hará clic de nuevo en el botón ‘Actualizar datos’ para confirmar los cambios en la base de datos.

Hecho esto volveremos a la pantalla ‘Cartera’ para proceder al último paso, guardar una copia de los datos del usuario.



En la pantalla 'Cartera' pulsaremos el botón 'Guardar mis criptos en mi cuenta' y en cuanto el proceso se termine se mostrará un mensaje tipo *Snackbar* indicando que el proceso se ha completado correctamente.

2. Prueba de uso incorrecto y manejo de errores

En esta segunda prueba tomaremos los mismos datos de ejemplo del caso anterior pero esta vez consideraremos que el usuario puede introducir incorrectamente algunos datos y usar la aplicación sin una conexión a internet.



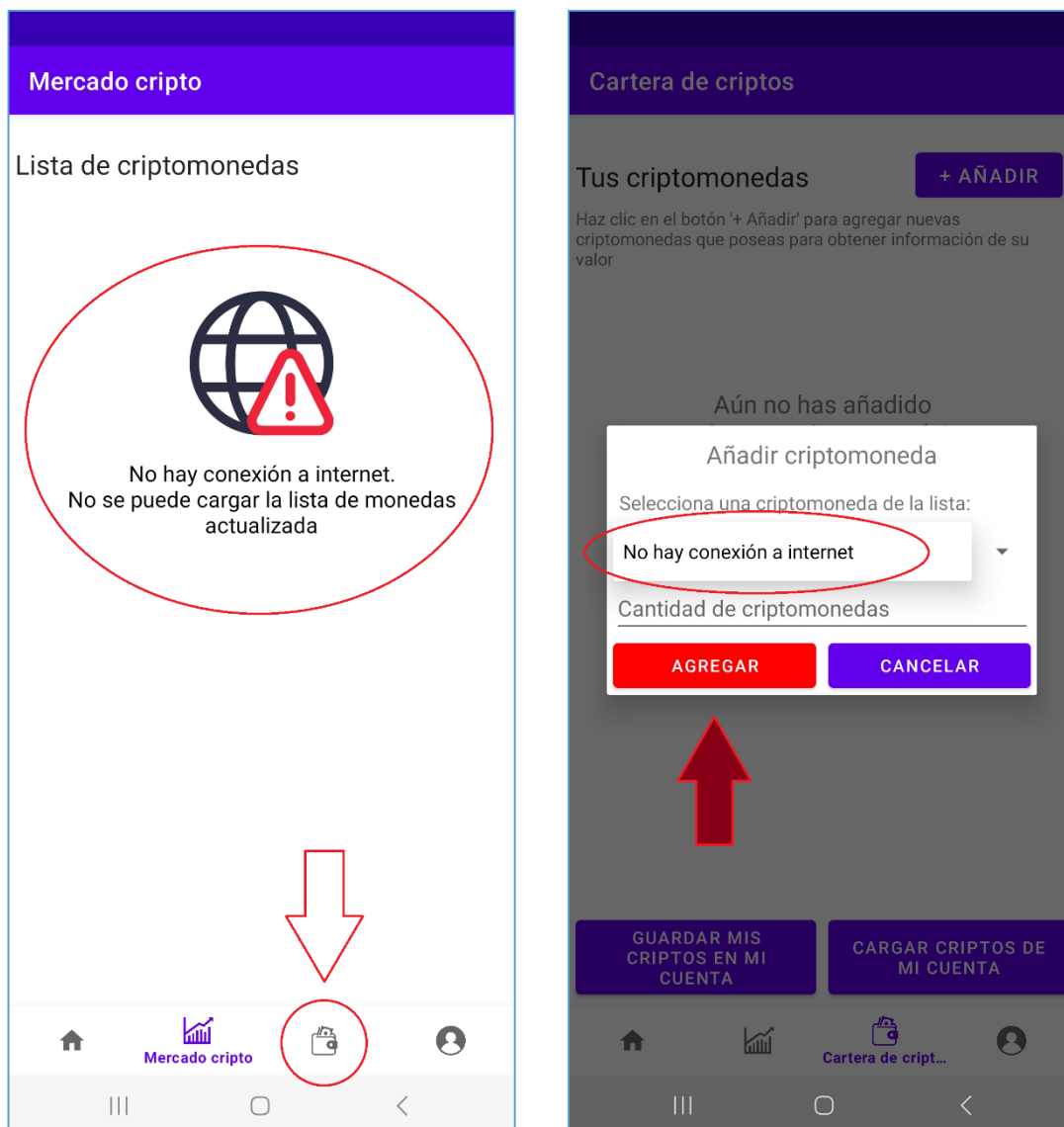
El usuario intentará consultar el mercado de criptomonedas y guardar una nueva criptomoneda en la cartera sin disponer de una conexión a internet, así como intentará hacer login.

Una vez encontrándose el dispositivo desconectado de internet, se abrirá la pantalla principal. Lo primero que podemos notar es un mensaje en la pantalla que nos indica que se está intentando actualizar el precio de las criptomonedas, pero si hacemos clic en el botón de actualizar se nos muestra otro mensaje *SnackBar* de que no se ha podido conectar a la API.

Aun así el precio se sigue mostrando pero este ya no será el precio actualizado sino que será el último precio que se obtuvo cuando la aplicación se conectó a internet por última vez.

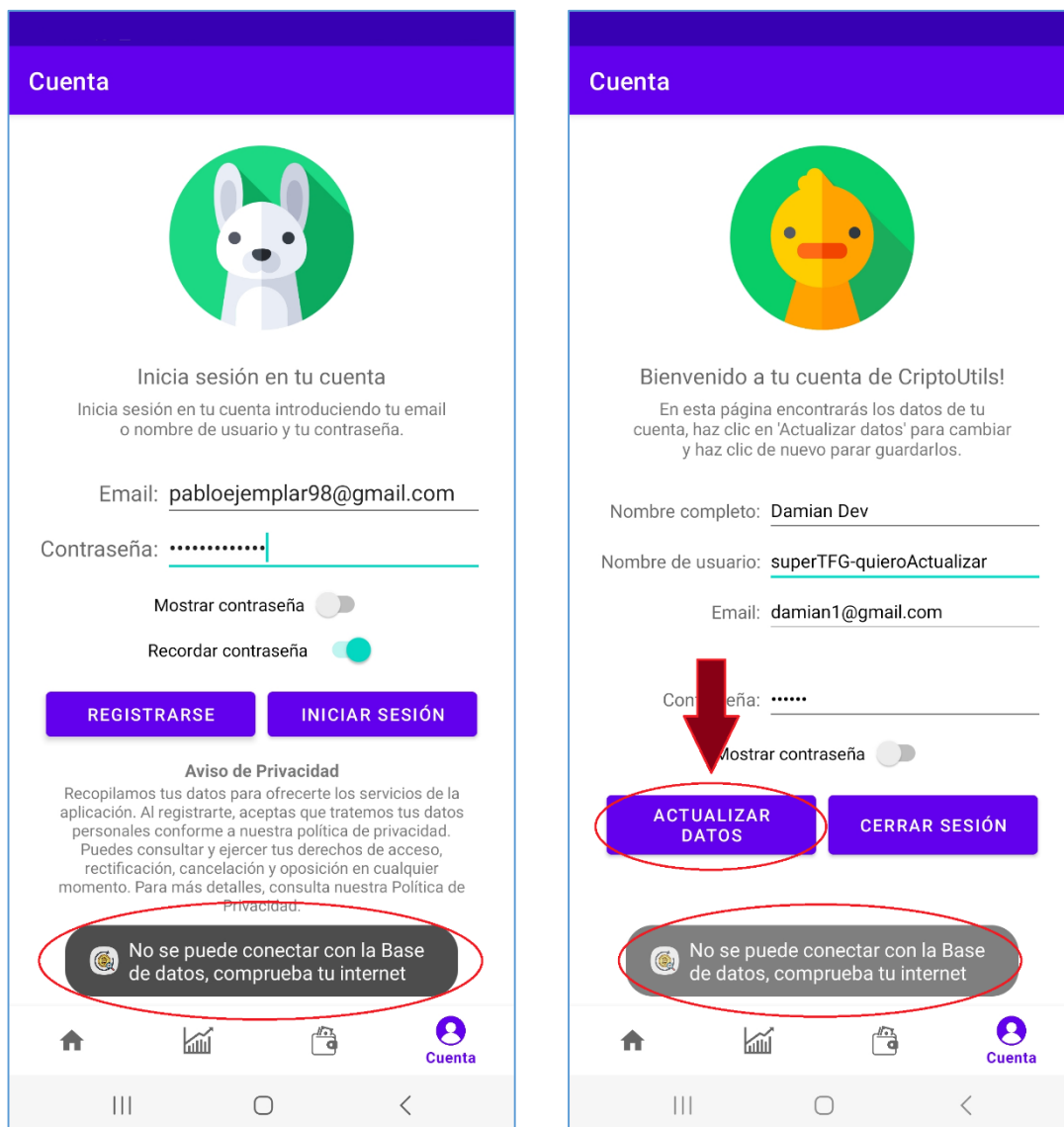


Después haremos clic en el icono del menú de navegación inferior 'Mercado cripto'. Al no tener conexión a internet no veremos la lista de criptomonedas en la pantalla si no un mensaje con un icono que nos indica que no tenemos conexión y que no se pueden obtener los datos para la lista de monedas.



Después accederemos a la pantalla ‘Cartera’ e intentaremos añadir una criptomoneda al seguimiento. Tras hacer clic sobre el botón Añadir, se nos abrirá el cuadro de dialogo pero el usuario verá un mensaje en el selector de monedas que dice “No hay conexión a internet” y el botón ‘Agregar’ se encontrará desactivado, la única opción será pulsar ‘Cancelar’ y revisar la conexión a internet.

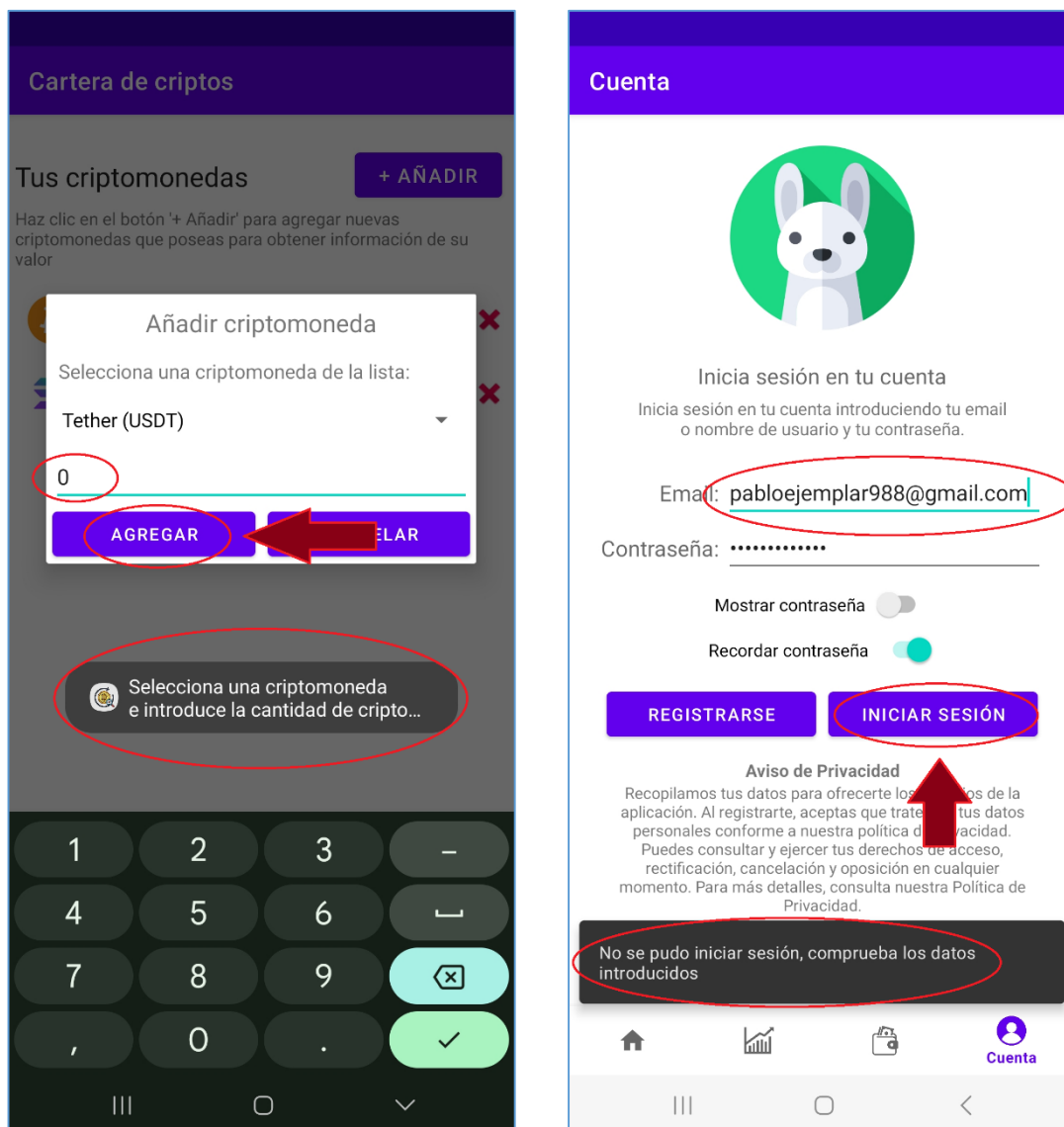
Por último iremos a la pantalla de inicio de sesión y tras rellenar los datos del usuario intentaremos iniciar sesión pulsando el botón correspondiente. Se nos mostrará un mensaje de error tipo *toast* indicando que no se puede conectar con la base de datos. Lo mismo aplica para el botón de registrarse.



Lo mismo ocurrirá si el usuario dispone de conexión al iniciar sesión y a la hora de actualizar pierde la misma, se mostrará un mensaje indicando la imposibilidad de conectar a la base de datos.

Ahora probemos el caso de que el usuario recupera la conexión a internet pero como indicábamos al principio, comete errores al introducir los datos.

En la pantalla de cartera supongamos que el usuario se equivoca al escribir la cifra de criptomonedas y escribe 0 e intenta guardar. El usuario recibirá un mensaje indicando que debe introducir un valor y este debe ser mayor a cero.



Por ultimo supongamos que el usuario se equivoca en el correo o contraseña e intenta iniciar sesión. El usuario recibirá un mensaje que le indica que revise los datos introducidos.

XIV. Conclusiones

El proyecto ha cumplido con los objetivos iniciales de proporcionar una herramienta para la gestión y análisis de portafolios de criptomonedas, integrando datos en tiempo real y ofreciendo funcionalidades como gráficos interactivos y noticias del mercado. En

cuanto al desarrollo, contaba con conocimientos previos en programación con Java y desarrollo de aplicaciones Android, lo que me permitió abordar el proyecto con confianza y centrarme en aprender nuevas tecnologías, como la integración de APIs de datos de mercado.

A nivel personal, esta experiencia ha sido muy enriquecedora, ya que me ha permitido no solo consolidar conocimientos técnicos, sino también gestionar un proyecto completo, desde su concepción hasta su implementación final. Lo que más me ha motivado ha sido trabajar en un problema real (también a nivel personal) y relevante en el creciente mercado de las criptomonedas, descubriendo el potencial de las tecnologías empleadas. Este proyecto me impulsa a seguir explorando el desarrollo móvil y su aplicación en contextos innovadores en el futuro.

XV. Proyección a futuro

Con los objetivos iniciales del proyecto cumplidos, se han identificado varias áreas de mejora para futuras versiones. Una prioridad será implementar un sistema de notificaciones a nivel de dispositivo y un sistema de recuperar la contraseña en caso de que el usuario se olvide. También se plantea ampliar la personalización de la aplicación, incluyendo opciones como la elección entre tema claro y oscuro.

Para incrementar el alcance de usuarios, se podría desarrollar una versión para iOS o una versión web, permitiendo así que la herramienta sea accesible desde más plataformas. Estas mejoras no solo enriquecerán la experiencia del usuario, sino que también posicionarán la aplicación como una solución más competitiva y versátil en el mercado de las criptomonedas.



XVI. Bibliografía

- [1] «Wikipedia,» Java, [En línea]. Available:
[https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- [2] «Wikipedia,» Android, [En línea]. Available:
<https://es.wikipedia.org/wiki/Android>
- [3] «Wikipedia,» Android Studio, [En línea]. Available:
https://es.wikipedia.org/wiki/Android_Studio
- [4] «Wikipedia,» GitHub, [En línea]. Available:
<https://es.wikipedia.org/wiki/GitHub>
- [5] «Wikipedia,» MySQL, [En línea]. Available:
<https://es.wikipedia.org/wiki/MySQL>
- [6] «CoinGecko,» CoinGecko, [En línea]. Available:
<https://docs.coingecko.com/reference/introduction>
- [7] «CoinMarketCap,» CoinMarketCap, [En línea]. Available:
<https://coinmarketcap.com/about/>
- [8] «Wikipedia,» Modelo–vista–modelo de vista, [En línea]. Available:
https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93modelo_de_vista
- [9] « Retrofit,» Retrofit, [En línea]. Available: <https://square.github.io/retrofit/>
- [10] « Android Developers » Views, [En line]. Available:
<https://developer.android.com/develop/ui/views/layout/recyclerview?hl=es-419>
- [11] « SQLite,» SQLite, [En línea]. Available: <https://www.sqlite.org/>



XVII. Autoría y derechos de la obra

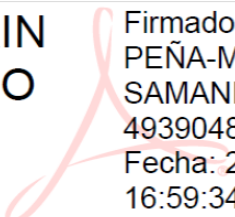
Copyright © 2024 Damián Peña-Marín Samaniego.

Esta obra está licenciada bajo una **Licencia Creative Commons:**

Atribución-No Comercial-Compartir Igual 4.0 Internacional (CC BY-NC-SA 4.0)



Los mismos derechos y la misma licencia aplican al código de esta aplicación disponible en GitHub o junto a este archivo.

PEÑA-MARIN SAMANIEGO DAMIAN - 49390486X	 Firmado digitalmente por PEÑA-MARIN SAMANIEGO DAMIAN - 49390486X Fecha: 2024.12.05 16:59:34 +01'00'
--	---