

**PROGRAMACIÓN DE APLICACIONES MÓVILES NATIVAS**

**2022/2023**

**NETBOND - RED SOCIAL**



Jiten Rajesh Parwani  
Aashish Sanjay Hathiramani

# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>3</b>
<b>DESARROLLO</b>	<b>4</b>
FASES DEL DESARROLLO	4
TEMPORIZACIÓN	18
HERRAMIENTAS	19
ESTRUCTURA Y EJECUCIÓN DE LA APLICACIÓN	20
ESTRUCTURA DE DESARROLLO	20
EJECUCIÓN	22
<b>CONCLUSIÓN</b>	<b>39</b>

# INTRODUCCIÓN

Para el trabajo de curso de esta asignatura, se ha optado por desarrollar una red social llamada **Netbond**. El nombre de la red social es una palabra compuesta en inglés, formada por *Net* (red) y *bond* (vínculo). El nombre viene a representar una **serie de vínculos en la red**.

El propósito de la red social es **demostrar quién conoce mejor a sus amigos** mediante puntuaciones. Existe lo que se denominan los **bonds** (publicaciones), a través de los cuales se pueden conseguir los puntos. Una publicación consiste en una pregunta y varias respuestas, aunque, de todas las respuestas sólo una será la correcta. La pregunta formulada, en principio, tendría que relacionarse con algún gusto personal del usuario.

De este modo, se puede **interactuar con otros usuarios**, enviando una solicitud de seguimiento para empezar a interactuar con sus *bonds* acumulando una serie de puntos.

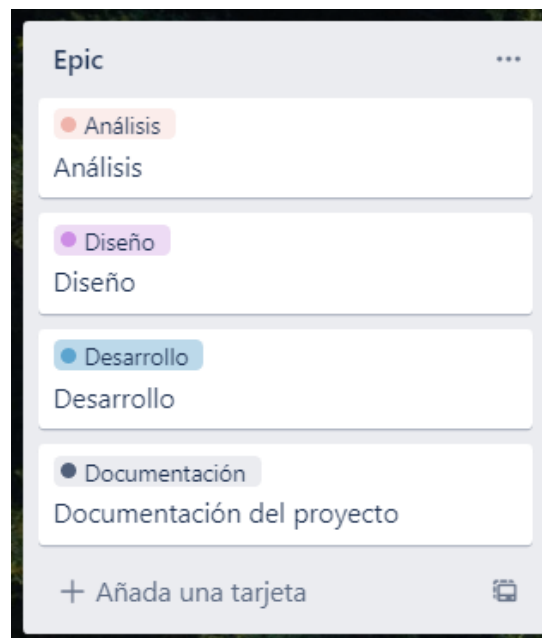
# DESARROLLO

Para el desarrollo de este trabajo, se ha seguido una **metodología de desarrollo ágil**, concretamente llamada **Scrum**. Siguiendo el marco de trabajo de Scrum, se ha dividido el trabajo a realizar en dos *sprints* (iteraciones), aunque se hablará con más detalle sobre esta planificación a lo largo de la memoria.

El desarrollo de la memoria se dividirá en cuatro partes: fases del desarrollo, temporización, herramientas utilizadas y ejecución de la aplicación.

## FASES DEL DESARROLLO

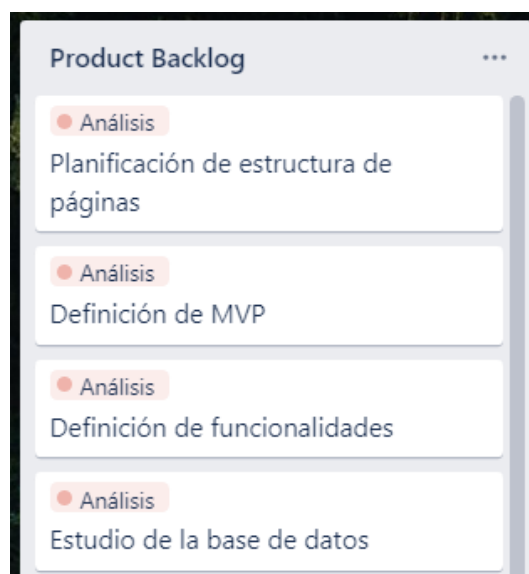
La planificación para el trabajo de curso se ha dividido en cuatro fases: análisis de los requisitos, diseño, desarrollo y documentación del trabajo. Cada una de estas fases, se ha relacionado con una **historia épica**, como se observa en la Figura 1.



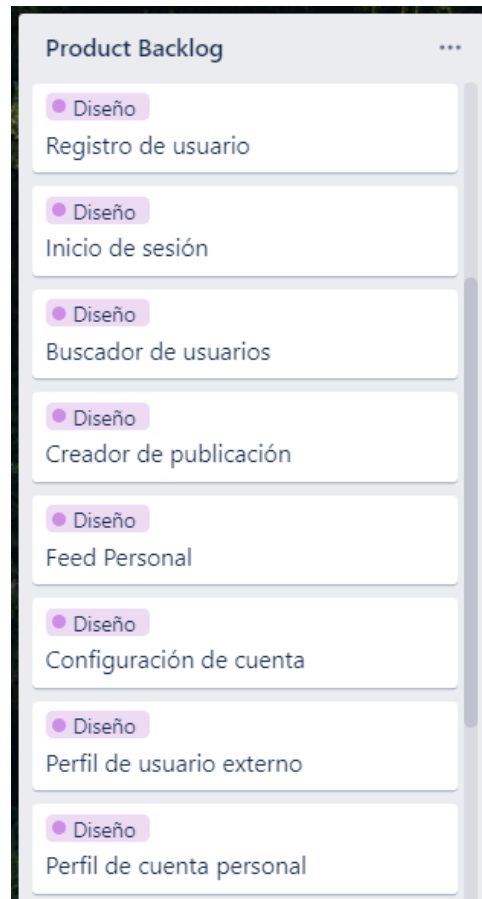
**Figura 1:** Fases del desarrollo.

Como resultado, se obtienen cuatro historias épicas, las cuales, cada una de ellas, han sido desglosadas en **historias de usuario**. La suma de todas las historias de usuario resultantes conforman el **product backlog** (pila de producto), el cual, representaría el **MVP** (*Minimum Viable Product*) que se ha planteado para este trabajo.

A continuación, en las siguientes figuras se muestra la pila de producto resultante, con las historias de usuario correspondientes por cada historia épica (cada fase de desarrollo). Cabe destacar que, se ha evitado la utilización de prioridades en las historias de usuario y solamente se ha utilizado el título, principalmente para acoger la simplicidad.



**Figura 2:** Fase de análisis.



**Figura 3:** Fase de diseño.



**Figura 4:** Fase de desarrollo.


Una vez mostrado el *product backlog* con todas las tareas que conforman el MVP, se procede a explicar las distintas páginas que contiene la aplicación (Figura 3) y posteriormente su correspondiente diseño (fase de diseño). En cuanto a la fase de análisis, la mayor parte de su desarrollo era básicamente un proceso de **brainstorming**. La fase de desarrollo se explicará más adelante en su correspondiente apartado.

A continuación, se muestra una pequeña descripción por cada página de la aplicación.

- **Registro de usuario:** página en la cual se registra un usuario.
- **Inicio de sesión:** página que aparece al acceder a la aplicación sin haber iniciado sesión previamente con una cuenta de usuario.
- **Buscador de usuarios:** página con un buscador de usuarios, en el que se muestran las coincidencias encontradas.
- **Feed personal:** página deslizable verticalmente (*scroll*), con *bonds* (publicaciones) de otros usuarios a los que “sigue” el usuario actual.
- **Perfil de usuario externo:** página que se accede tras seleccionar un usuario (desde el buscador, por ejemplo), en el que muestra información pública de ese usuario. Por defecto, todas las cuentas de usuarios son privadas, por lo que se debe “seguir” enviando una solicitud y ser aceptado para poder ver e interactuar con sus publicaciones en el correspondiente *feed*.
- **Perfil de cuenta personal:** página que muestra todas las publicaciones del usuario actual, así como su información personal.
- **Configuración de cuenta:** página en la que se pueden realizar cambios de la información personal de la cuenta.
- **Creador de publicación:** página en la que se crea un *bond* (publicación).

Una vez introducidas las páginas de la aplicación, en las siguientes figuras se observa el diseño de cada una de ellas.





netbond

START CREATING BONDS

Name

Email


Username

Password

SIGN UP

Already using Netbond? [Log In.](#)

**Figura 5:** Registro de usuario.



netbond

WELCOME TO YOUR BONDS

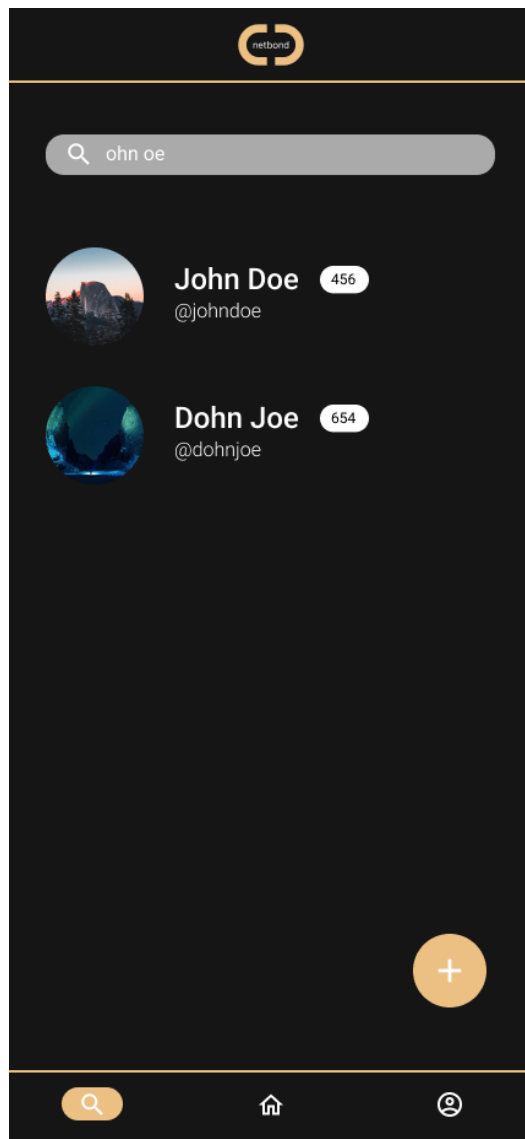
Email

Password

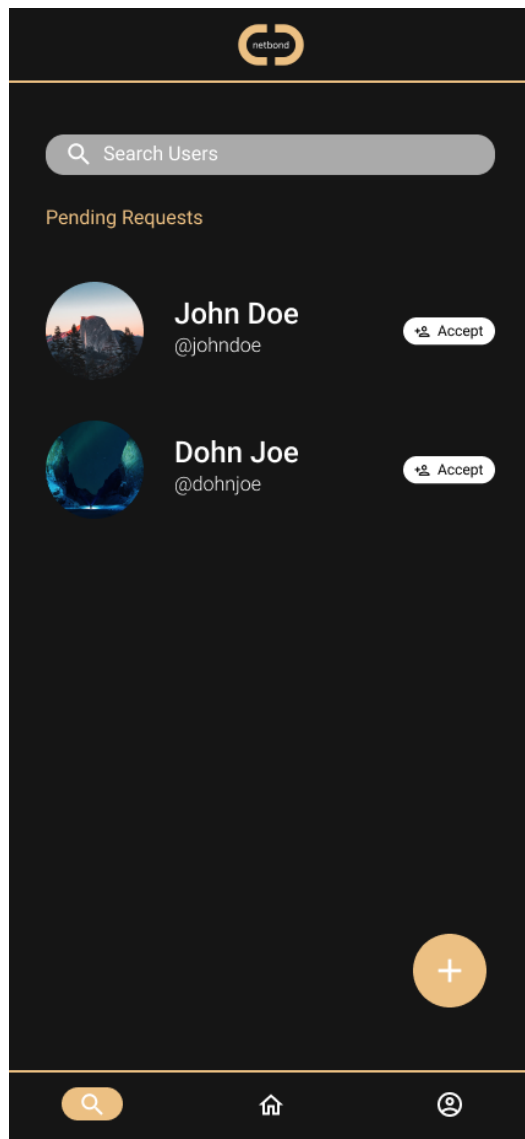
LOG IN

New to Netbond? [Sign Up.](#)

**Figura 6:** Inicio de sesión.

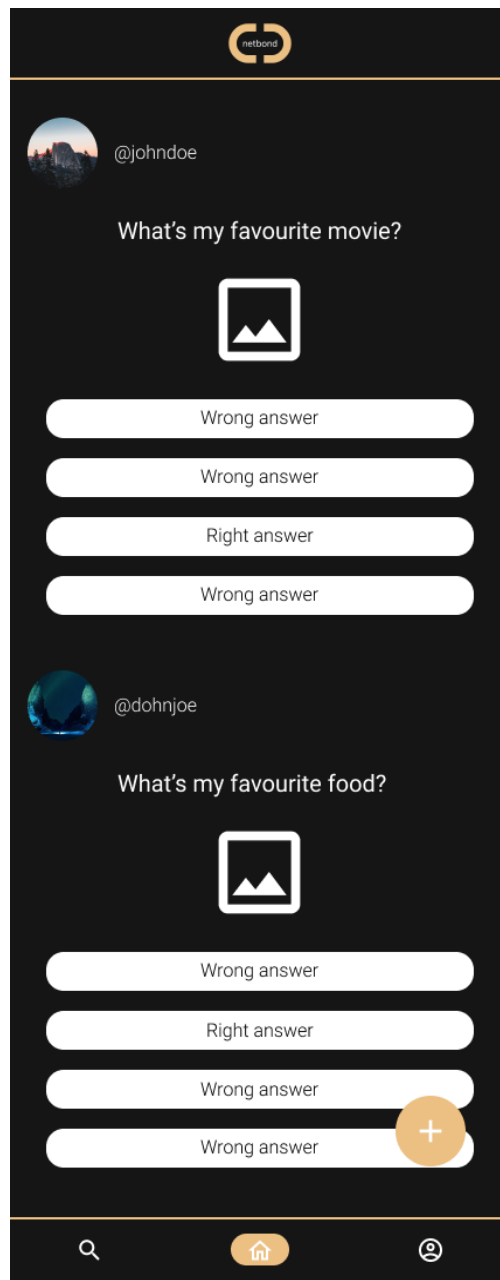


**Figura 7:** Buscador de usuarios.

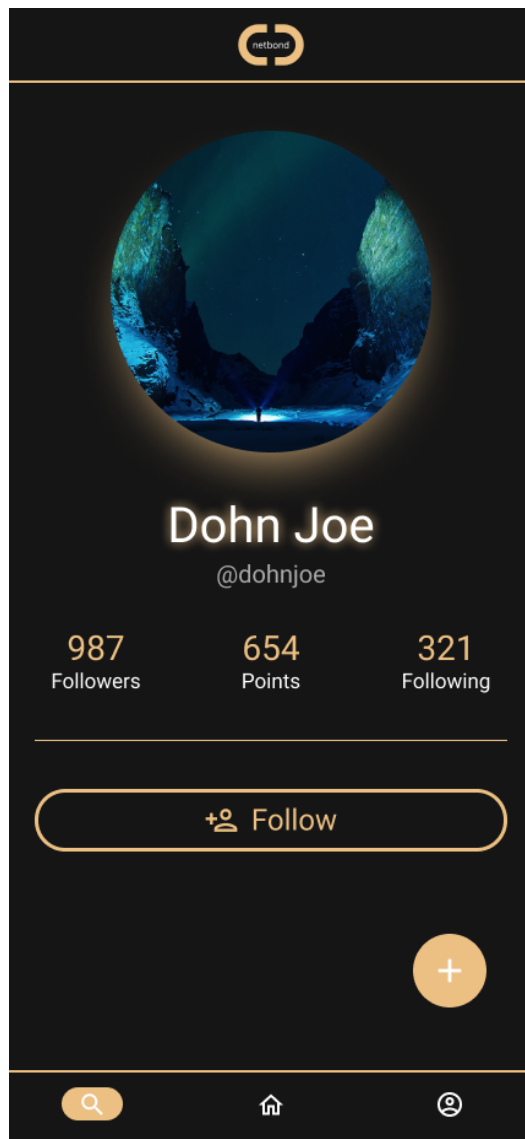


**Figura 8:** Buscador de usuarios (solicitudes pendientes).

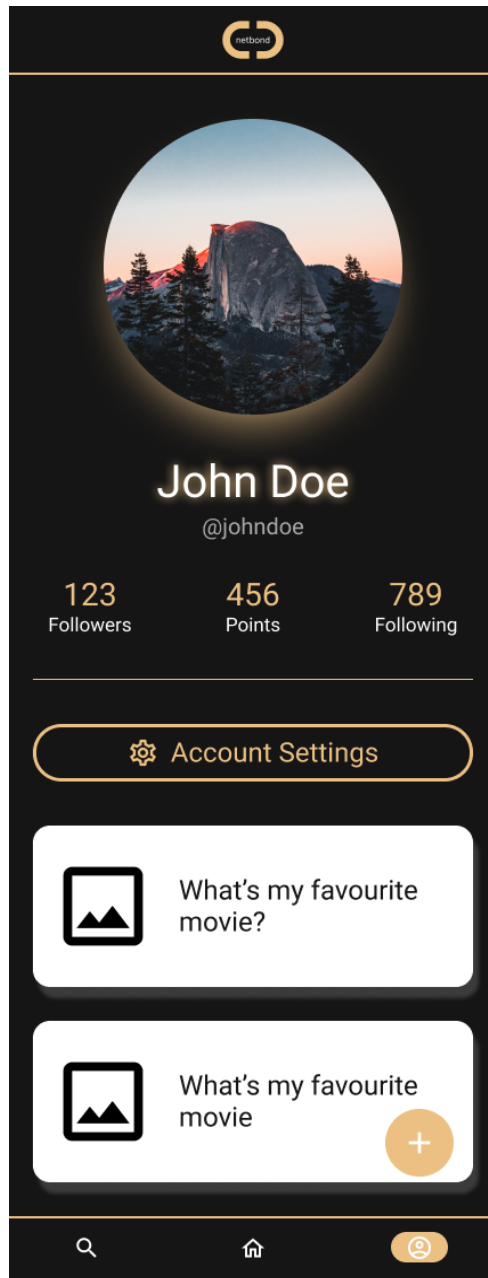
A diferencia de la Figura 7, en la que se muestran las coincidencias encontradas, como se observa en la Figura 8, si no se escribe nada en el buscador, aparecen las **solicitudes pendientes** (solicitudes recibidas). Es decir, si el buscador está vacío aparecen las solicitudes pendientes, y al contrario, aparecen las coincidencias encontradas.



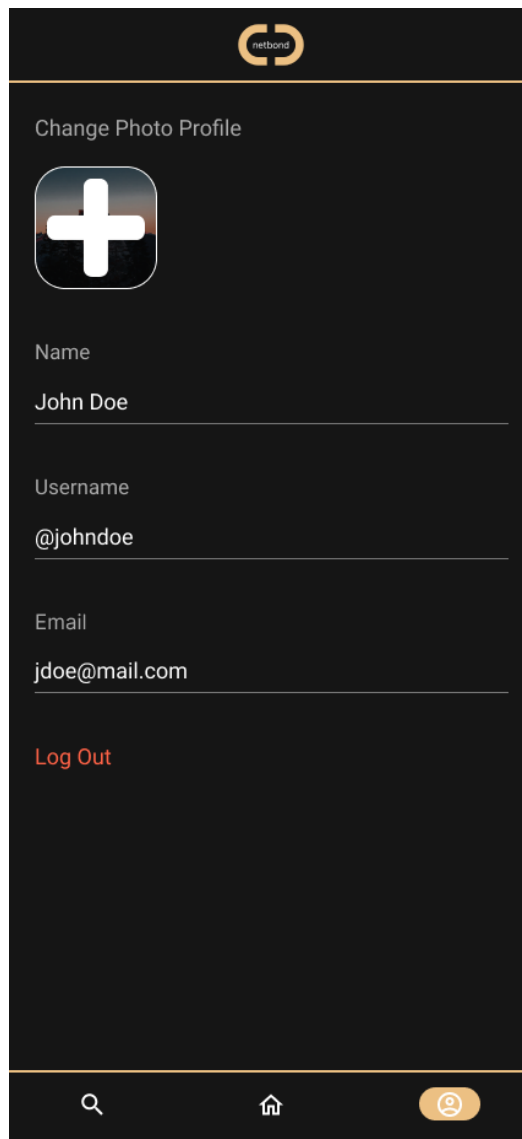
**Figura 9:** *Feed* personal.



**Figura 10:** Perfil de usuario externo.

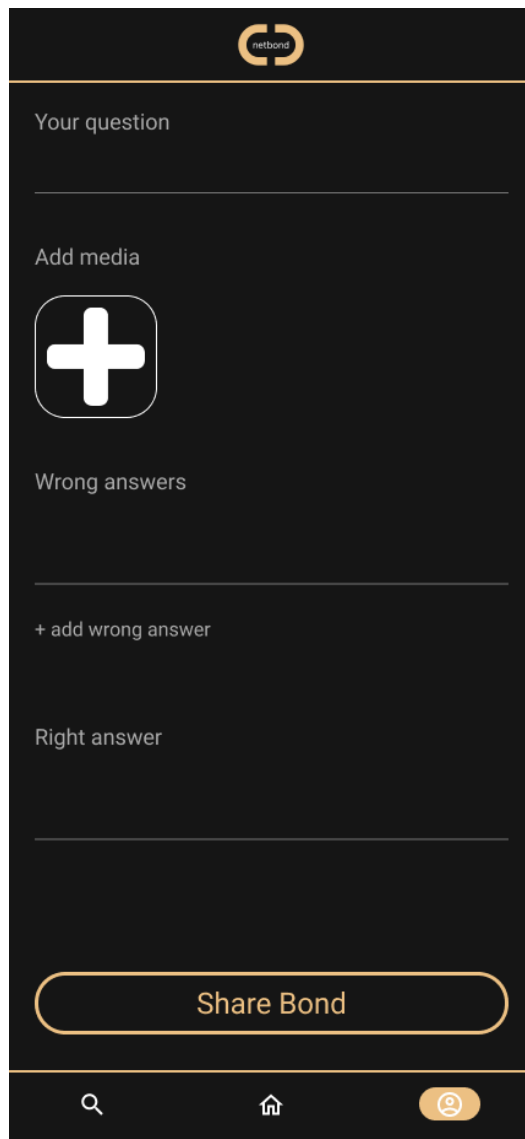


**Figura 11:** Perfil de cuenta personal.



**Figura 12:** Configuración de cuenta.

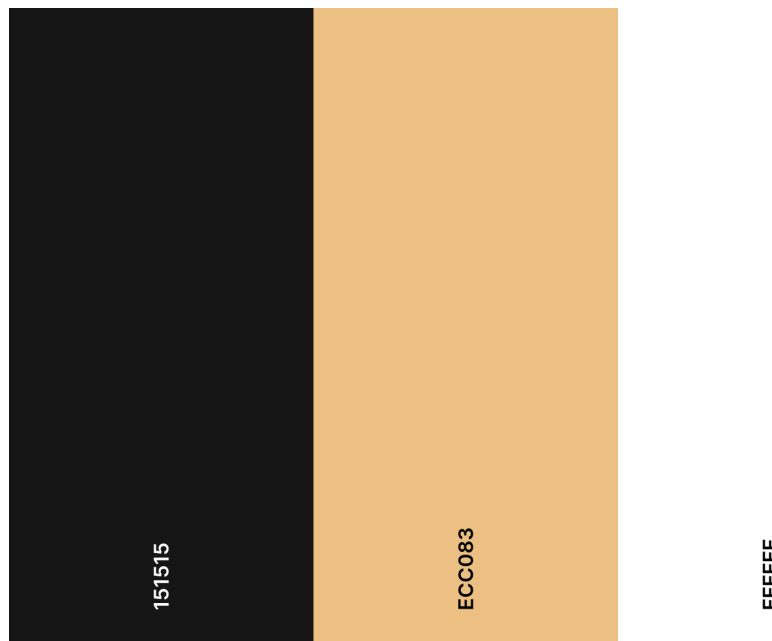




**Figura 13:** Creador de publicación.

Finalmente, destacar que en la mayoría de las páginas existe un **botón flotante** con un icono en forma de “+”, el cual, al interactuar con él, navega a la página de creador de publicación, como se observa en la Figura 13.

A continuación, en la Figura 14, se muestra la paleta de colores que se ha seguido para diseñar la aplicación.



**Figura 14:** Paleta de colores.

## TEMPORIZACIÓN

En cuanto a la temporización del trabajo, como se comentó anteriormente, se ha dividido el trabajo a realizar en dos *sprints* (iteraciones). Concretamente, cada uno de estos *sprints* se ha designado como una **duración de dos semanas**. Para separar las tareas a realizar en cada sprint, se decidió crear un **sprint backlog**, en el cual, se almacenaban las tareas a realizar en la correspondiente iteración.

Durante el **primer sprint**, se planificó para trabajar con las fases de análisis y diseño, con las correspondientes historias de usuario para cada fase en concreto.

Durante el **segundo sprint**, se planificó para trabajar con las dos últimas fases; desarrollo de la aplicación y documentación del trabajo (esta misma memoria).

A continuación, se muestra la planificación en horas por persona de cada fase:

- **Análisis:** 3 horas por persona, lo que suma un total de 6 horas totales.
- **Diseño:** 4 horas por persona, lo que suma un total de 8 horas totales.

- **Desarrollo:** 7 horas por persona, lo que suma un total de 14 horas totales.
- **Documentación:** 2 horas y media por persona, lo que suma un total de 5 horas totales.

## HERRAMIENTAS

Para las fases de diseño, desarrollo y documentación se han utilizado diversas herramientas que se van a mencionar a continuación. Para la fase de análisis, como se ha comentado anteriormente, simplemente era una tormenta de ideas.

A continuación, se muestran las herramientas utilizadas para cada fase.

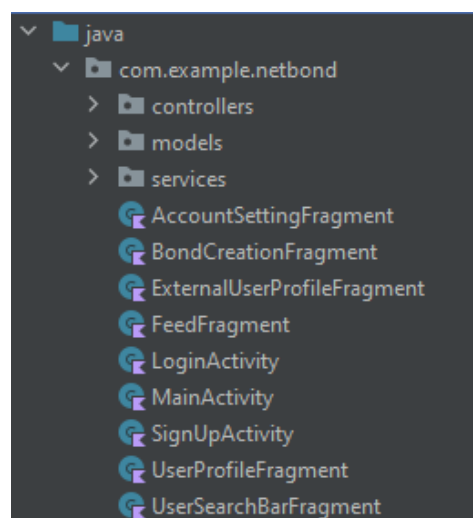
- **Diseño:** se ha utilizado la aplicación web de **Figma**. Se puede acceder a los diseños de las páginas a través del siguiente enlace: <https://www.figma.com/file/Ilf643P1voUgcJgKQclnWk/netbond?node-id=0%3A1&t=99wKQXgvQ8TbvOj-1>.
- **Desarrollo:** se ha utilizado el IDE (*Integrated Development Environment*) de **Android Studio** para la programación en **Kotlin**. Se puede acceder al repositorio GitHub de la aplicación a través del siguiente enlace: <https://github.com/G10on/netbond>.
- **Documentación:** se ha utilizado **Google Docs** para confeccionar la memoria del trabajo.

Finalmente, para la organización de las tareas a realizar, se ha utilizado la aplicación web de **Trello**. En ella, se ha creado el correspondiente *product backlog*, *sprint backlog*, etc., como se ha ido mostrando durante la memoria. Se puede acceder al tablero de Trello a través del siguiente enlace: <https://trello.com/invite/b/BoYWIJAD/ATTI0f256985690fb3ff99cf66c4ba4047be3D6E50BC/pa mn>.

# ESTRUCTURA Y EJECUCIÓN DE LA APLICACIÓN

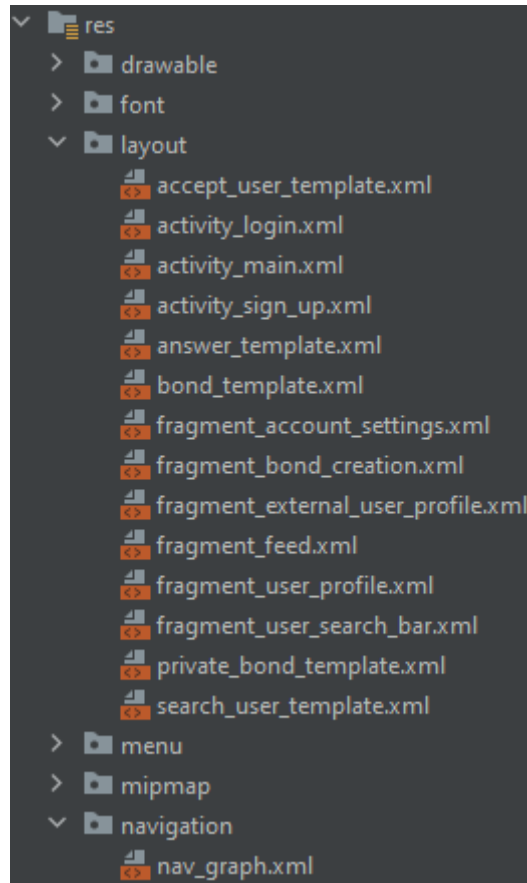
## ESTRUCTURA DE DESARROLLO

Ahora, se procederá a mostrar los aspectos más relevantes de la aplicación, tanto desde la implementación, como de su demostración. Desde el punto de vista de la implementación, se optó por una arquitectura compuesta por los XML, actividades y fragmentos, así como paquetes diferenciados para los modelos, controladores y servicios. Para las vistas, se emplearon tres actividades. Las dos primeras actividades se asignaron al inicio de sesión y al registro de usuario, con sus correspondientes plantillas de XML. La tercera actividad sería el contenedor que contendría el resto de los fragmentos, al que se accedería tras el inicio de sesión.



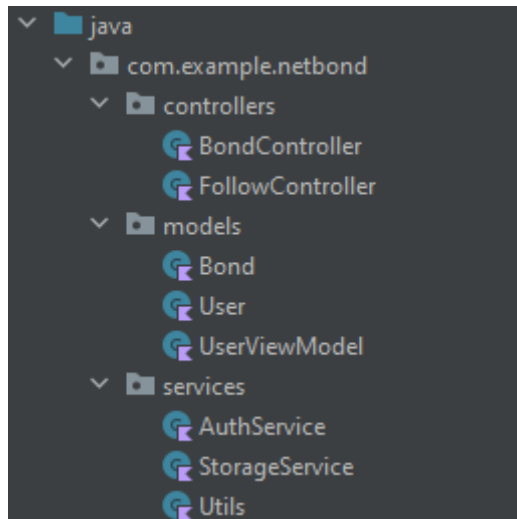
**Figura 15:** Directorio con actividades.

Además, en esta tercera actividad se encontraría la barra de navegación, mapeado mediante la herramienta de grafos.



**Figura 16:** Directorios de layouts y grafos de navegación.

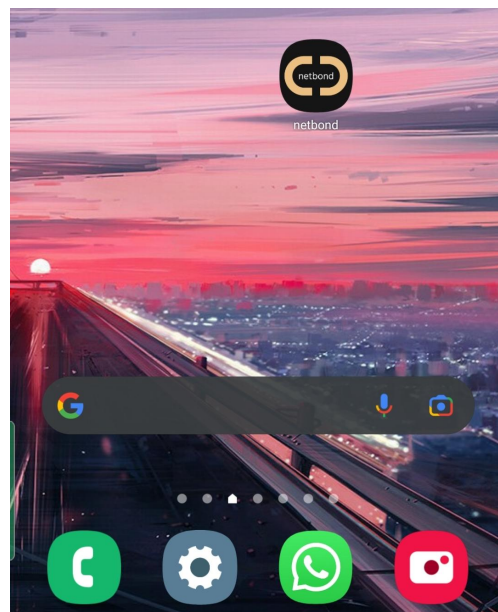
Se emplearon modelos como 'User' y 'Bond', que se encontraban en el paquete 'models' para facilitar los accesos a los datos en Firebase. Estos accesos a Firebase se realizaban mayoritariamente mediante las clases en el paquete de 'services', como las autenticaciones, subidas de imágenes, etc. Para abstraer algunas de las funcionalidades y organizar algunas de las operaciones más empleadas, se hizo uso de las clases del paquete de 'controllers', como operaciones para seguir usuarios, solicitar seguimientos, borrar bonds, etc. En la siguiente figura se muestran estas estructuras.



**Figura 17:** Directorios de paquetes de ayuda.

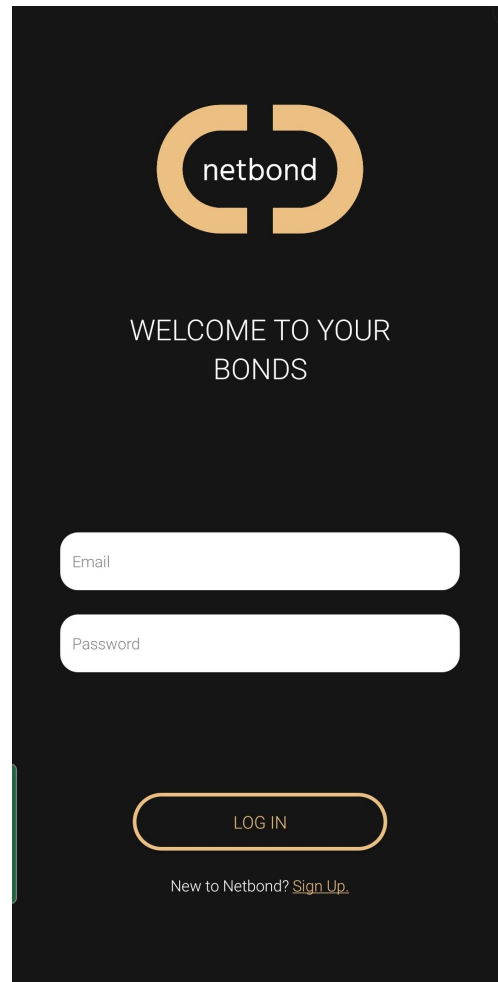
## EJECUCIÓN

En cuanto a la ejecución de la aplicación, en primer lugar, se presenta en el escritorio del usuario con el diseño de su propio logo, como se puede observar en la Figura 5. Esto se logró mediante herramientas online que permitían diseñar logos personales.



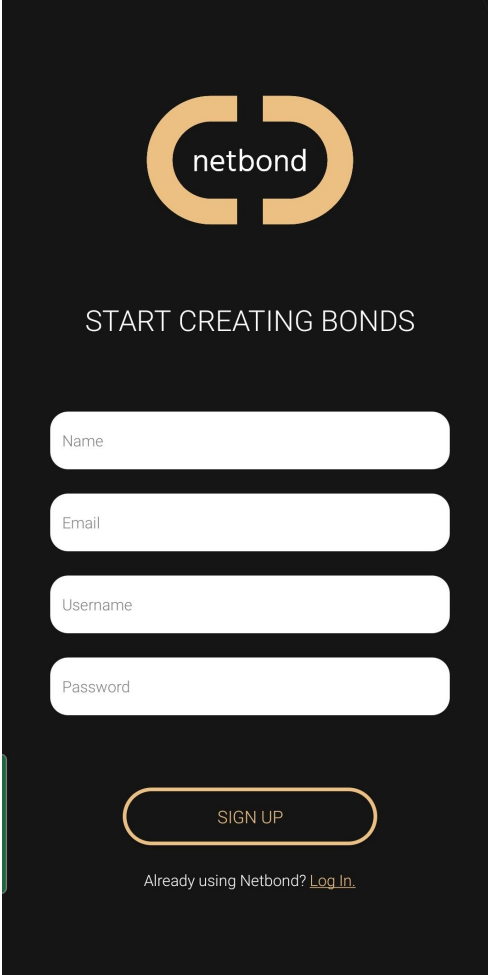
**Figura 18:** Icono de la aplicación.

Una vez se accede a la aplicación, la primera pantalla que aparece es la del inicio de sesión, como se muestra en la siguiente figura. Esto se debe porque es necesario una cuenta de usuario para poder hacer uso de la aplicación. Si el usuario no tuviese previamente una cuenta, puede registrar una nueva pulsando la opción *Sign Up*.



**Figura 19:** Pantalla de inicio de sesión.

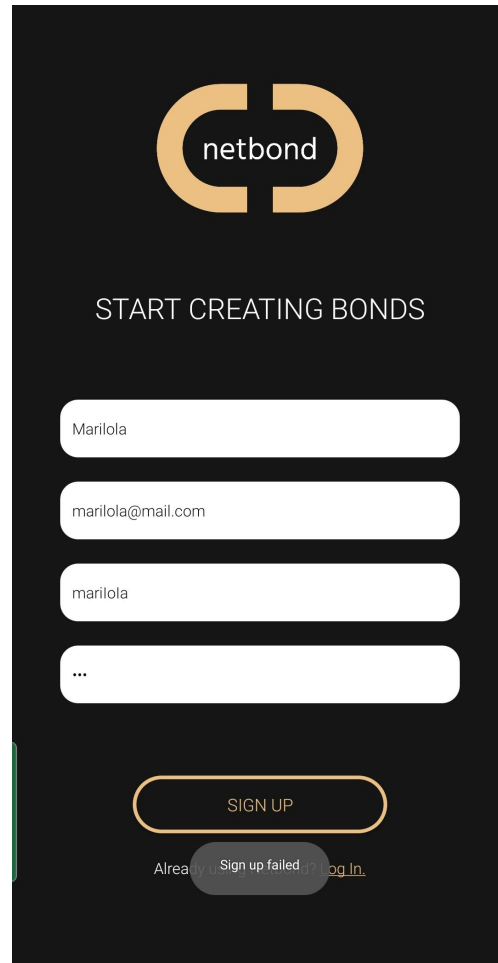
En la siguiente figura se muestra la pantalla para registrar una nueva cuenta de usuario. Desde esta, se le debe indicar un nombre, email, username y una contraseña.

The image shows a mobile application screen for user registration. At the top, there is a logo consisting of two interlocking orange rings with the word "netbond" in white lowercase letters. Below the logo, the text "START CREATING BONDS" is displayed in white uppercase letters. Underneath this text are four white input fields with rounded corners, each containing a placeholder label: "Name", "Email", "Username", and "Password". At the bottom of the form is a large, rounded orange button with the text "SIGN UP" in white uppercase letters. Below the button, there is a line of text that reads "Already using Netbond? [Log In](#)", where "Log In" is a blue hyperlink. The entire screen has a dark background.


**Figura 20:** Pantalla de nuevo registro.



Como se muestra en las siguientes figuras, el registro falla si el email o el username ya existen, o si la contraseña es menor de 6 caracteres (requisito de Firebase).



**Figura 21:** Demostración de un registro fallido, debido a una contraseña corta.



netbond

START CREATING BONDS

Marilola

marilola@mail.com

jiten

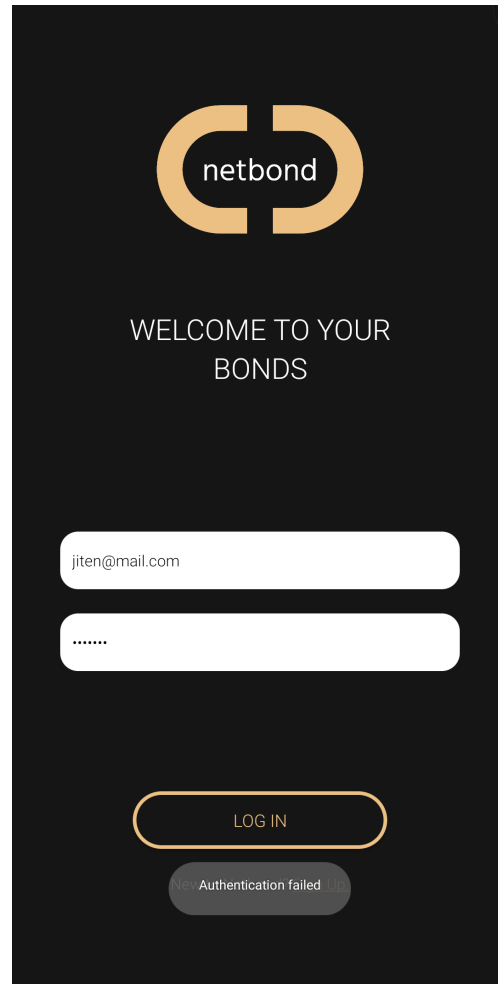
\*\*\*\*\*

SIGN UP

Alt This username already exists

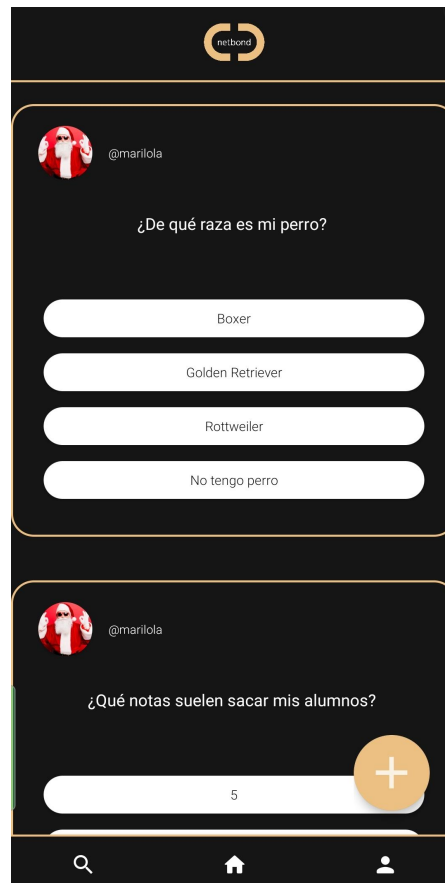
**Figura 22:** Registro de nuevo usuario fallido debido a que *username* ya existe.

Se procederá a realizar el inicio de sesión con una cuenta existente. En la siguiente figura se muestra cómo el inicio de sesión falla si la contraseña es incorrecta.



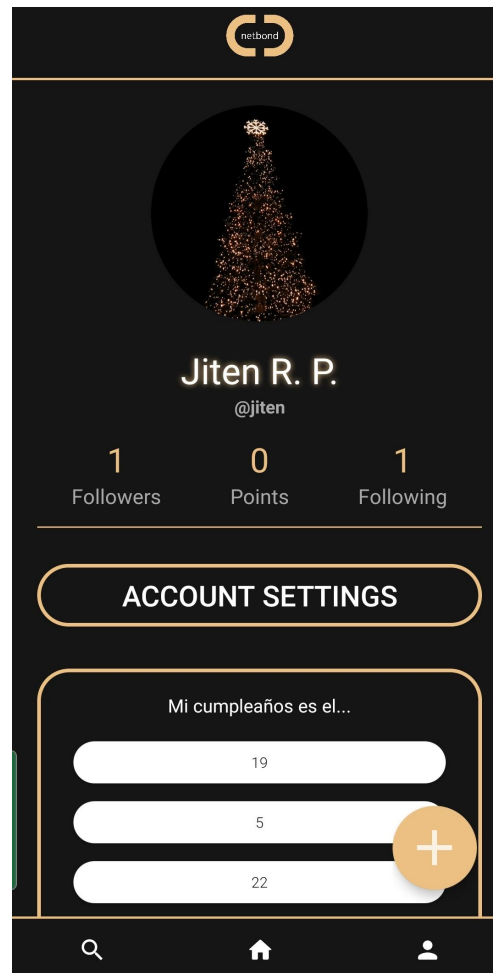
**Figura 23:** Demostración de inicio de sesión fallido debido a contraseña incorrecta.

Una vez se haya registrado o iniciado sesión, el primer fragmento que aparece es el *Feed*. En este, el usuario observa todos los bonds de usuarios a los que sigue y que no ha completado. Asimismo, tiene la posibilidad de responder a estos desde este *Feed*. En la siguiente figura, se muestra un ejemplo de su visualización. Se anota que este fragmento se puede acceder pulsando en el icono central de *Home* en la barra de navegación.



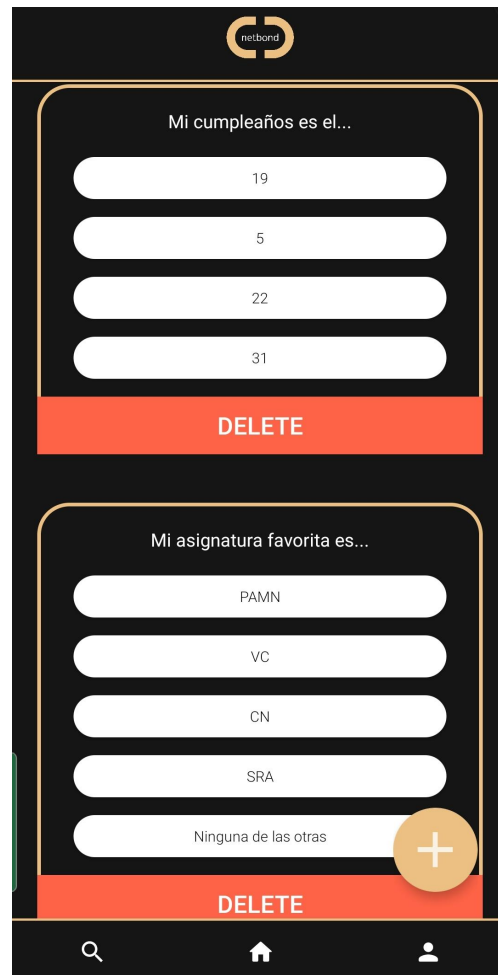
**Figura 24:** Feed del usuario, tras iniciar sesión.

Pulsando en el icono derecho del usuario en la barra de navegación, se accede al fragmento que se muestra en la siguiente figura. En este, el usuario puede visualizar su perfil, así como los contadores de seguimiento, puntos, foto de perfil, y bonds publicados.



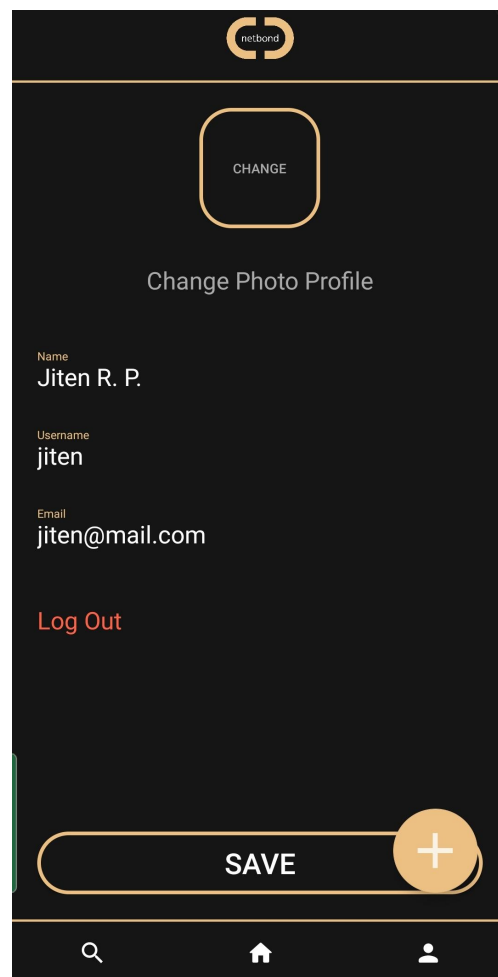
**Figura 25:** Visualización de cuenta personal del usuario.

En la siguiente figura, se muestra cómo se verían los bonds del usuario si se deslizara hacia abajo en el fragmento del perfil de usuario. Aquí, el usuario tiene la posibilidad de eliminar un bond pulsando sobre *DELETE*.



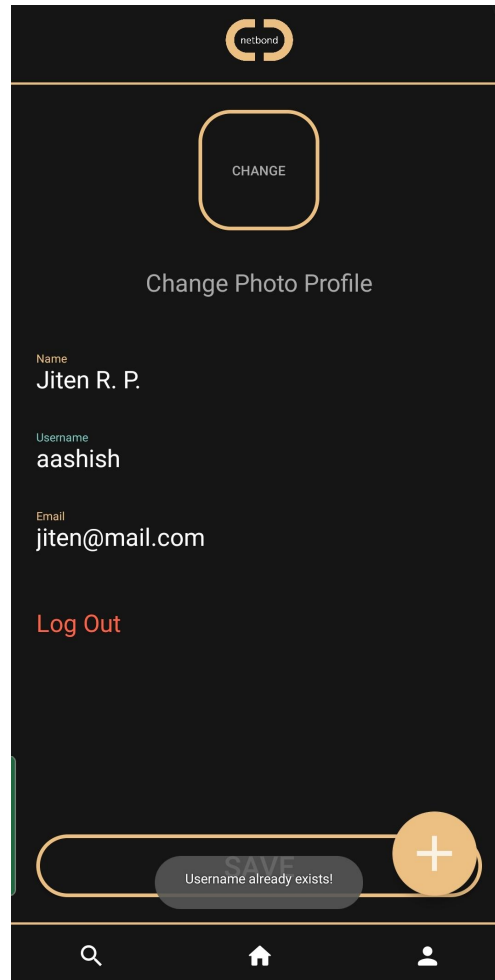
**Figura 26:** Visualización de los bonds personales del usuario.

Como se vio en el fragmento del perfil de usuario, hay un botón llamado *ACCOUNT SETTINGS*, que lleva a un fragmento como el que se muestra en la siguiente figura. En este, el usuario tiene la posibilidad de modificar varios de los datos que introdujo inicialmente, así como cambiar su foto de perfil o incluso cerrar sesión. Para guardar los cambios, el usuario pulsaría en el botón *SAVE*.



**Figura 27:** Ajustes de la cuenta del usuario.

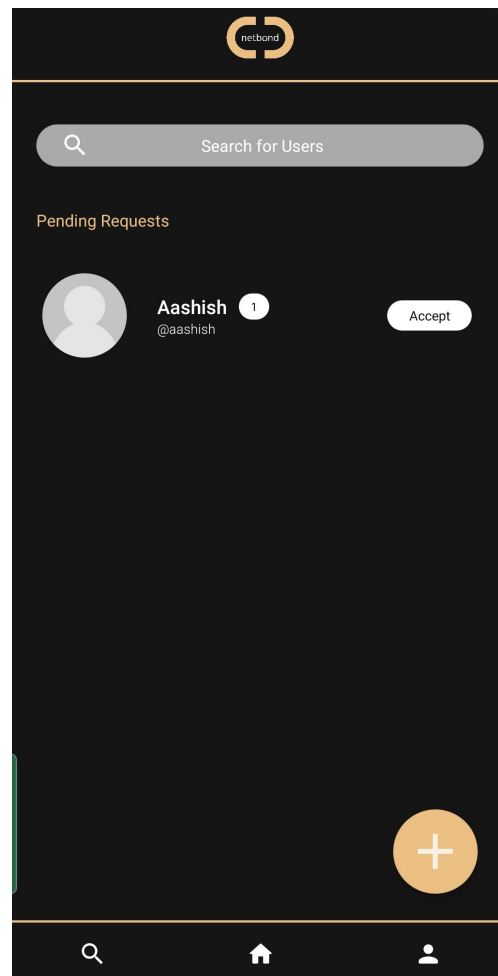
Si durante la modificación de los datos el usuario introduce datos indebidos (email o username existentes), se notifica que se produjo un fallo durante el guardado de los cambios.



**Figura 28:** Demostración de cambio de *username* fallido, debido a que ya existe.

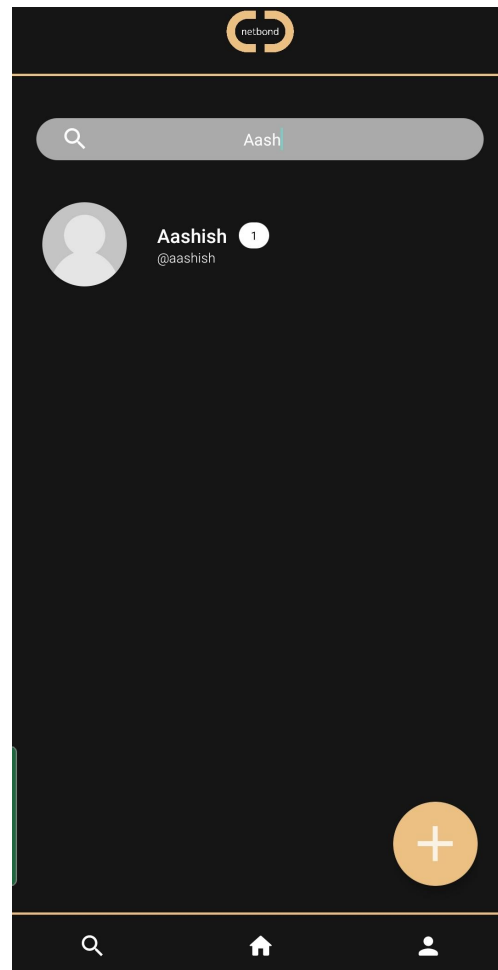


Seguidamente, si se pulsa en el icono izquierdo de la lupa en la barra de navegación, se accede al fragmento que se muestra en la siguiente figura. En este fragmento, el usuario tiene la posibilidad de observar solicitudes entrantes de otros usuarios, así como la opción de aceptar la solicitud. En la figura se muestra un ejemplo de un usuario que ha recibido una solicitud de seguimiento. Se destaca, además, que desde esta solicitud se puede acceder al perfil de ese usuario.



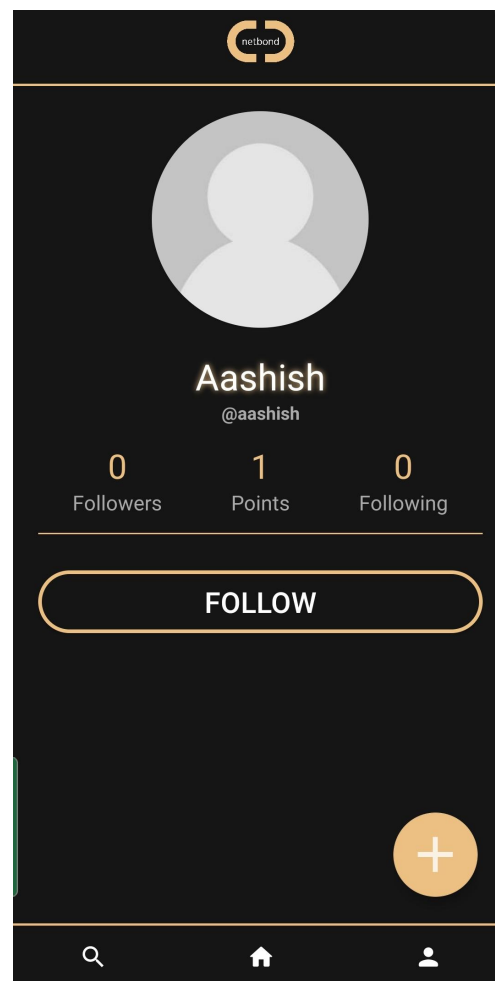
**Figura 29:** Pantalla de solicitudes y búsquedas de usuarios.

Sobre la sección de solicitudes pendientes, se encuentra un buscador de usuarios, desde el que se pueden realizar búsquedas de otros usuarios. En la siguiente figura, se muestra un ejemplo de la búsqueda de un usuario.



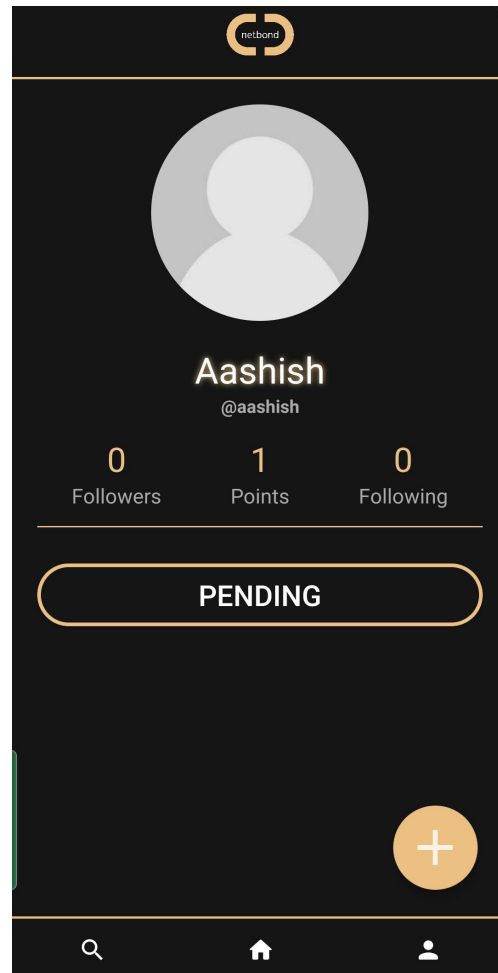
**Figura 30:** Demostración de búsqueda de usuarios.

Pulsando sobre el resultado de la búsqueda, se accede al perfil de este usuario, como se puede observar en la siguiente figura. En este fragmento, el usuario puede observar muchos datos del perfil que visita, como el número de seguimientos, puntos, foto de perfil... Además, tiene la opción de pulsar el botón de *FOLLOW*, lo que le permitiría enviar una solicitud de seguimiento como el que recibió en las figuras anteriores.



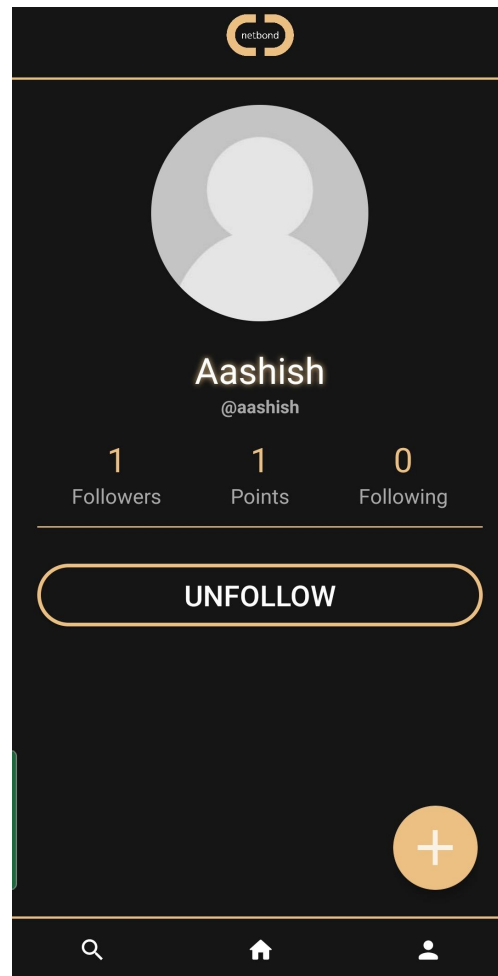
**Figura 31:** Visualización de un usuario externo no seguido.

Una vez que se pulsa en el botón de seguimiento cuando está en estado *FOLLOW*, este cambia al estado de *PENDING*, indicando que se ha enviado la solicitud y aún no ha sido aceptada. Si el usuario vuelve a pulsar este botón, la solicitud se eliminará y volverá al estado de *FOLLOW*. En la siguiente figura se muestra un ejemplo de este estado.



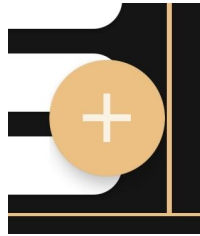
**Figura 32:** Visualización de un usuario externo al que se envió una solicitud.

Una vez que el otro usuario haya aceptado la solicitud de seguimiento, el estado de este botón cambiará a *UNFOLLOW*, como se observa en la siguiente figura. Este estado indicaría que desde ese momento el usuario sigue al usuario del perfil que se encuentra visitando. Además, este estado informaría que, si se desea dejar de seguir a ese usuario, simplemente debe pulsar en ese botón.



**Figura 33:** Visualización de un usuario externo al que se sigue.

Como se indicó en anteriores secciones, hay un botón flotante en la parte derecha inferior que está presente en la mayoría de los fragmentos, como se puede ver en la siguiente figura.



**Figura 34:** Botón flotante para crear un bond.

Pulsando sobre este botón, se accede a un fragmento desde el que se puede crear y compartir un bond. En la siguiente figura, se muestra una demostración para crear un nuevo bond. En este, primero se rellena el campo de la pregunta, y seguidamente se añaden las posibles respuestas. Finalmente, se selecciona la respuesta correcta pulsando sobre ella, y se publica el bond.

Una captura de pantalla de una aplicación móvil con fondo negro. En la parte superior hay un logo 'mitbond' con un icono de flechas. El texto 'Your question' precede a la pregunta '¿Cuál es mi asignatura favorita?'. Debajo, el texto 'Now write all the possible answers and select the right one' introduce la sección 'Answers'. Se muestran cuatro opciones: 'VC', 'PAMN' (destacada con un fondo verde brillante), 'SRA' y 'CN'. Debajo de estas opciones hay un botón naranja que dice '+ Add wrong answer'. En la parte inferior de la pantalla hay un botón redondeado con el texto 'SHARE BOND'. El menú de navegación de la app está visible en la base.

**Figura 35:** Demostración de la publicación de un bond.

## CONCLUSIÓN

Con este proyecto de fin de asignatura, se ha adquirido conocimientos esenciales respecto a la programación en sistemas operativos de Android. Se ha podido experimentar las distintas etapas del ciclo de vida de su desarrollo, desde el lenguaje nativo Kotlin, hasta el uso de Firebase para la base de datos.

Llevar a cabo el desarrollo con las metodologías ágiles permitió aprender y tomar un primer contacto sobre el desarrollo que se suele realizar en el mundo laboral. Haciendo uso de Trello, se adquirieron mejores capacidades para organizar y distribuir los procesos y sus tareas de una forma eficaz, así como obtener resultados de forma progresiva.

Hay varias funcionalidades que se planearon implementar en futuros sprints (hipotéticamente). Una posible característica a añadir en los siguientes sprints en la aplicación, es una guía inicial cuando se realiza un registro. Esto facilitaría al usuario la usabilidad y familiarización con la interfaz de la aplicación.

Otra funcionalidad a implementar en futuros sprints sería que se pueda acceder a la lista de usuarios seguidos o que se siguen, pulsando en los contadores correspondientes en el perfil de un usuario. Con ello, los usuarios dispondrían de una forma para poder comprobar todos aquellos usuarios a los que sigue, así como a sus seguidores.