

# **Prototipo de la aplicación**

# Manual de Usuario de la aplicación FitLife

## 0. INTRODUCCIÓN GENERAL DEL PROYECTO

FitLife es una aplicación Android desarrollada como proyecto de Formación en Centros de Trabajo (FCT) del ciclo de Desarrollo de Aplicaciones Multiplataforma (DAM). El objetivo del proyecto es aplicar de forma práctica los conocimientos adquiridos durante el ciclo, desarrollando una aplicación funcional y completa.

Este documento constituye la **documentación final del proyecto**, integrando en un único archivo PDF la descripción del sistema, el diseño de la base de datos, el manual de usuario y la documentación técnica necesaria para su evaluación por parte del equipo educativo.

### 0.1 Objetivos del proyecto

#### Objetivo general

Desarrollar una aplicación Android que permita gestionar hábitos y rutinas personales.

#### Objetivos específicos

- Gestionar usuarios mediante autenticación.
- Crear y administrar rutinas.
- Implementar recordatorios con notificaciones.
- Mostrar el progreso del usuario.
- Gestionar roles de usuario y administrador.

---

### 0.2 Stack tecnológico utilizado

- Android nativo
- Lenguaje Java
- XML + Material Design

- Firebase Authentication
- Firebase Firestore
- Fragments y Navigation Component

Estas tecnologías se han seleccionado por su adecuación al entorno Android y al alcance del proyecto FCT.

---

## 0.3 Arquitectura general de la aplicación

La aplicación se basa en una arquitectura con **una actividad principal** que contiene múltiples **fragments**, gestionados mediante Navigation Component.

Se distinguen las siguientes capas:

- Presentación: fragments y layouts XML.
- Lógica: controladores y validaciones.
- Datos: Firebase Authentication y Firestore.

Esta estructura favorece el mantenimiento y la escalabilidad del proyecto.

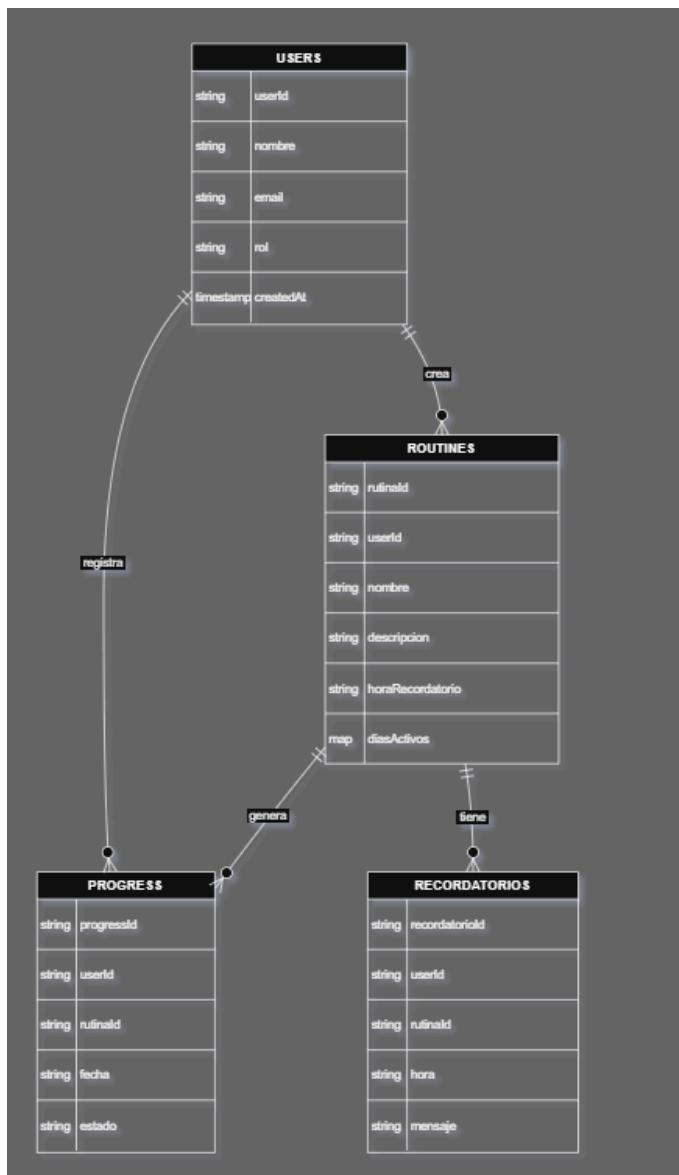
---

## 0.4 Diseño de la base de datos (Firestore)

FitLife utiliza Firebase Firestore, una base de datos NoSQL orientada a documentos.

### Colecciones principales

- users
- routines
- recordatorios
- progress



## 0.5 Prototipo de la aplicación

Antes de la implementación final de la aplicación, se realizó un **prototipo visual** con el objetivo de definir la estructura de las pantallas, la navegación y el diseño general de la interfaz de usuario.

Este prototipo permitió:

- Visualizar la distribución de los elementos en cada pantalla.
- Definir la experiencia de usuario (UX).
- Establecer una línea gráfica coherente basada en Material Design.
- Facilitar la posterior implementación de la aplicación Android.

El diseño del prototipo sigue los principios de **Material Design**, utilizando una interfaz limpia, moderna y orientada a la usabilidad, con navegación inferior y componentes visuales claros.

## Pantallas incluidas en el prototipo

El prototipo de FitLife incluye las siguientes pantallas principales:

- Pantalla de inicio de sesión.
- Pantalla de registro de usuario.
- Panel principal de usuario.
- Gestión de rutinas.
- Gestión de recordatorios.
- Pantalla de progreso y estadísticas.
- Perfil y configuración del usuario.
- Panel de administrador.

---

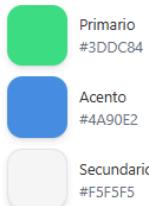
## Diseño visual y sistema de diseño

El sistema de diseño del prototipo se basa en una paleta de colores sencilla y coherente, junto con una tipografía clara y legible.

### Características principales del diseño:

- Uso de colores primarios y secundarios para diferenciar secciones.
- Tipografía legible para facilitar la lectura.
- Esquinas redondeadas y sombras suaves.
- Uso de iconografía clara.
- Navegación inferior para el acceso rápido a las funciones principales.

## Paleta de Colores



Primario  
#3DDC84

Acento  
#4A90E2

Secundario  
#F5F5F5

## Tipografía

Poppins Bold  
Encabezados y Títulos

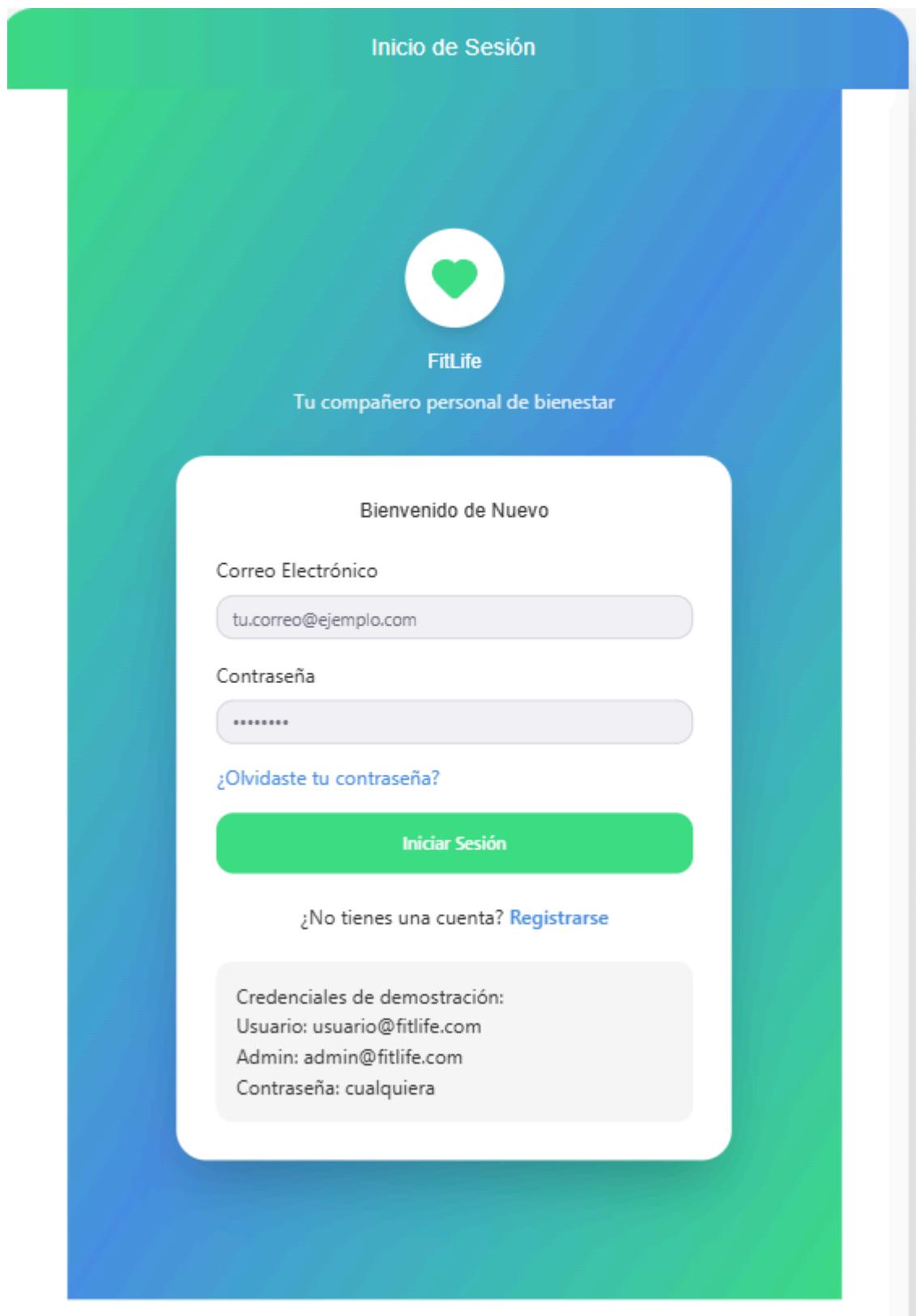
Roboto Regular  
Texto del Cuerpo

## Características Clave

- ✓ Principios Material Design
- ✓ Esquinas Redondeadas y Sombras
- ✓ Encabezados con Gradiente
- ✓ Navegación Inferior (Usuario)
- ✓ Botones de Acción Flotante
- ✓ Gráficas Interactivas

## Prototipo de pantallas principales

A continuación, se muestran algunas de las pantallas definidas en el prototipo de la aplicación.



Pantalla inicial que permite al usuario autenticarse en la aplicación.

## Registro



FitLife

Comienza tu viaje de bienestar hoy

### Crear Cuenta

Nombre Completo

Juan Pérez

Correo Electrónico

tu.correo@ejemplo.com

Contraseña

\*\*\*\*\*

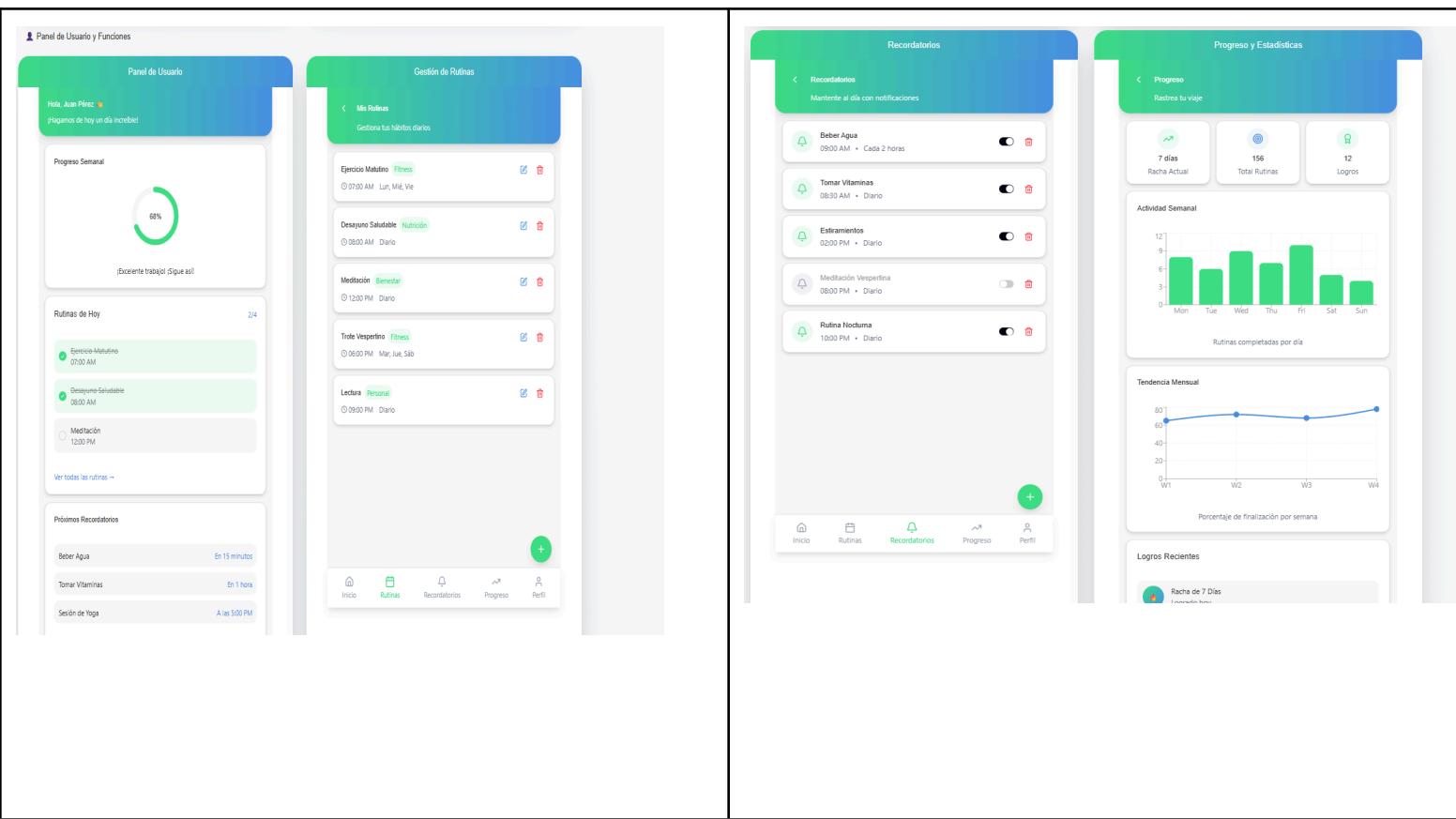
Confirmar Contraseña

\*\*\*\*\*

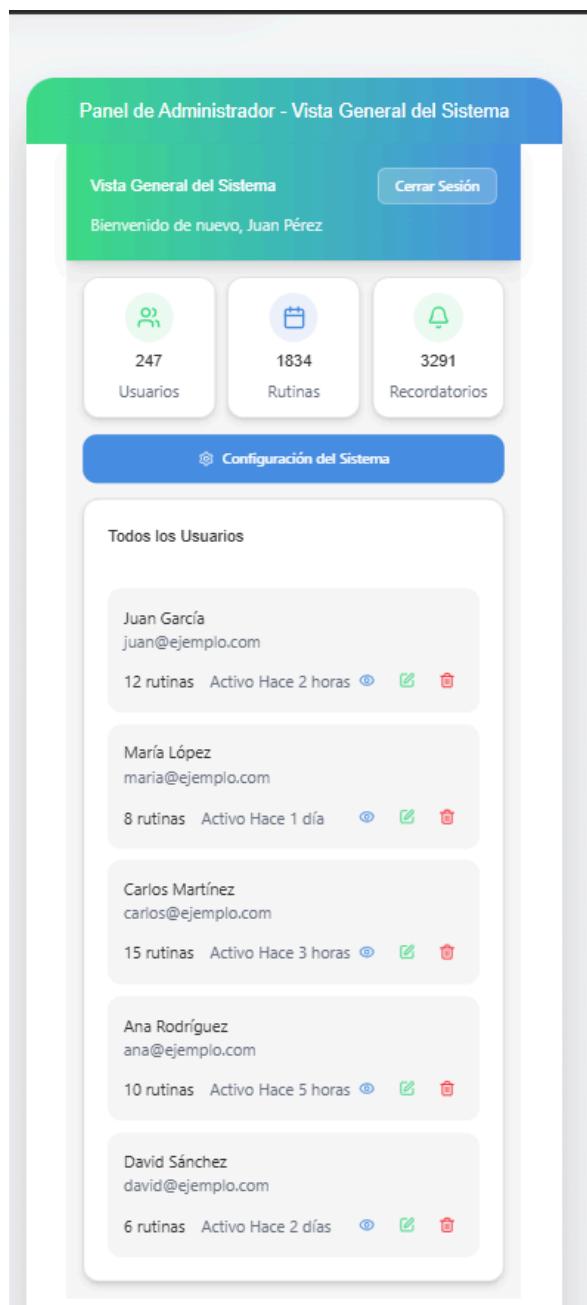
**Crear Cuenta**

¿Ya tienes una cuenta? [Iniciar Sesión](#)

*Formulario de creación de cuenta con los campos necesarios para el registro.*



Pantalla principal del usuario con acceso a rutinas, recordatorios y progreso.



*Vista general del sistema para usuarios con rol administrador.*

## Relación entre prototipo y aplicación final

El prototipo ha servido como referencia directa para el desarrollo de la aplicación final, manteniendo la estructura visual y funcional definida inicialmente.

La mayoría de las pantallas implementadas en la aplicación coinciden con el diseño del prototipo, lo que ha permitido un desarrollo más organizado y coherente.

---

## 2. Requisitos del sistema

- Dispositivo móvil con sistema operativo Android.
  - Versión mínima recomendada: Android 8.0 o superior.
  - Conexión a Internet.
  - Cuenta de usuario registrada en la aplicación.
- 

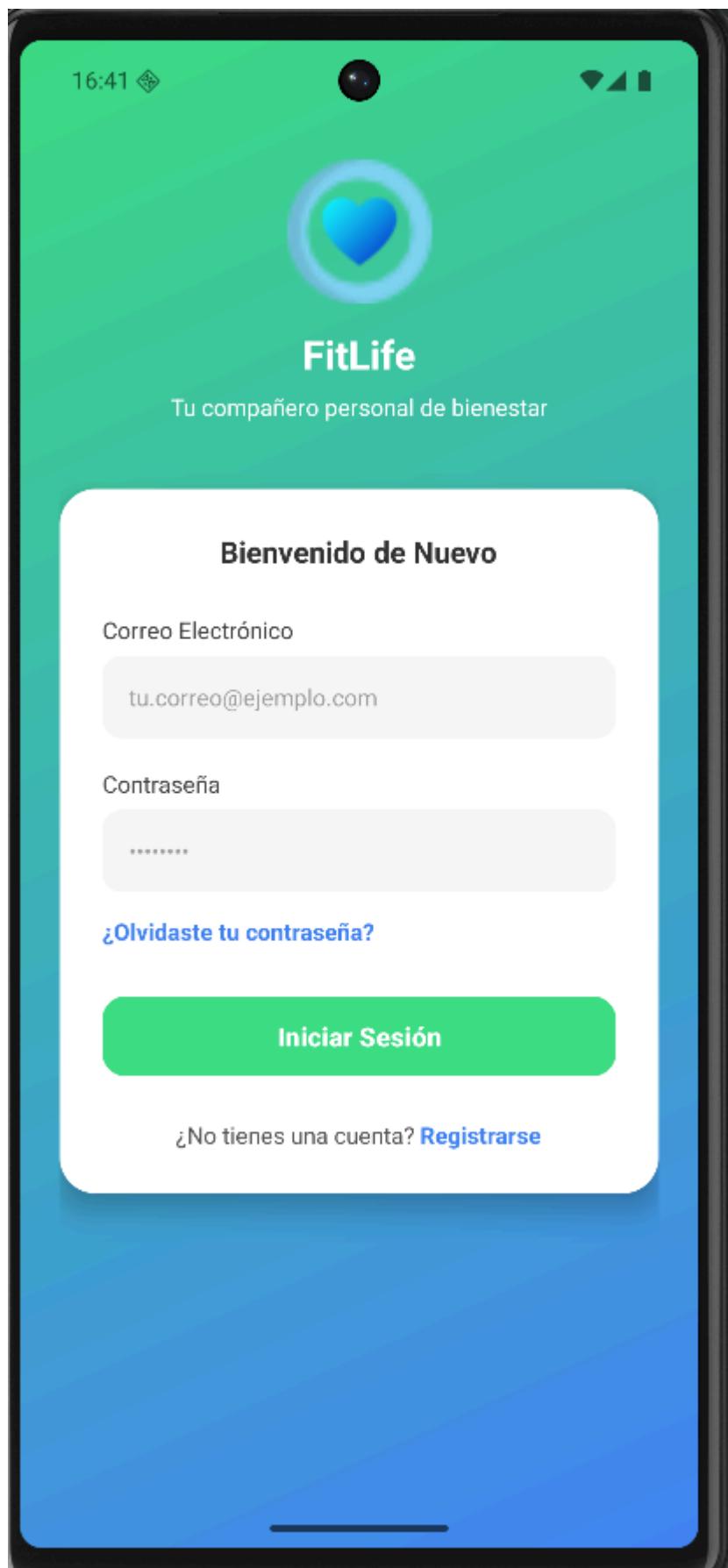
### **3. Pantalla de Inicio de Sesión**

**Descripción:**

**Pantalla inicial de la aplicación que permite al usuario autenticarse mediante correo electrónico y contraseña.**

**Acciones disponibles:**

- Introducir correo electrónico.
- Introducir contraseña.
- Acceder a la aplicación.



*Imagen mostrando el formulario de login con los campos de email y contraseña y el botón de acceso.*

---

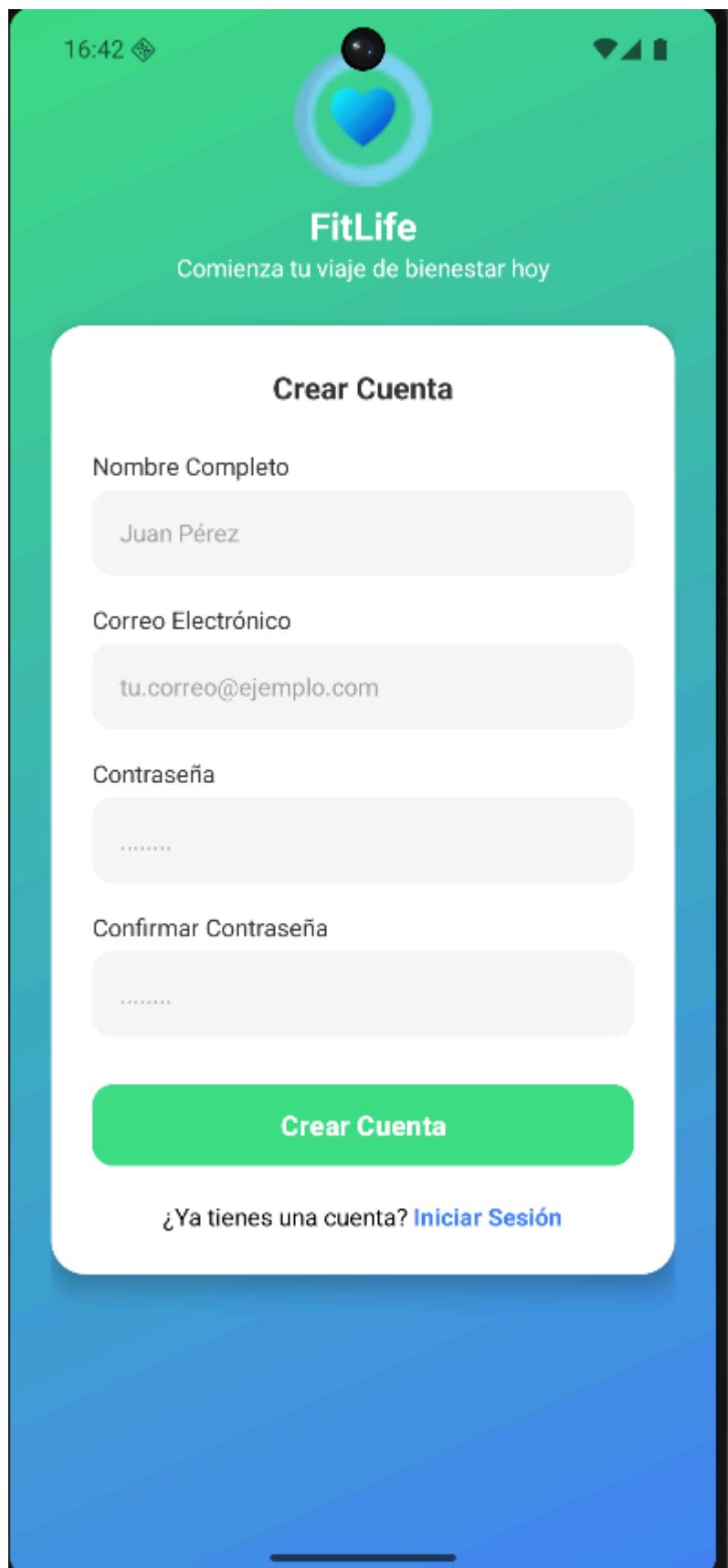
## 4. Pantalla de Registro

### **Descripción:**

Permite al usuario crear una nueva cuenta en la aplicación.

### **Acciones disponibles:**

- Introducir nombre.
- Introducir correo electrónico.
- Introducir contraseña.
- Confirmar registro.



**Imagen mostrando el formulario de registro completo.**

## 5. Dashboard (Panel Principal de Usuario)

### 5.1 Progreso semanal

El progreso semanal se muestra en un **fragment independiente**, accesible desde el menú de navegación inferior en la opción **Progreso**.

Esta pantalla está dedicada exclusivamente al seguimiento de la actividad del usuario a lo largo de la semana.

#### Elementos mostrados:

- Título identificativo del apartado de progreso.
- Gráfico de barras con la **actividad semanal**.
- Eje horizontal con los días de la semana.
- Eje vertical con el número de rutinas completadas por día.

El gráfico se actualiza automáticamente en función de las rutinas que el usuario va marcando como completadas.

#### Objetivo del fragment:

- Permitir al usuario analizar su constancia semanal.
- Ofrecer una visión clara y visual de su rendimiento.
- Complementar la información mostrada en el Dashboard principal.

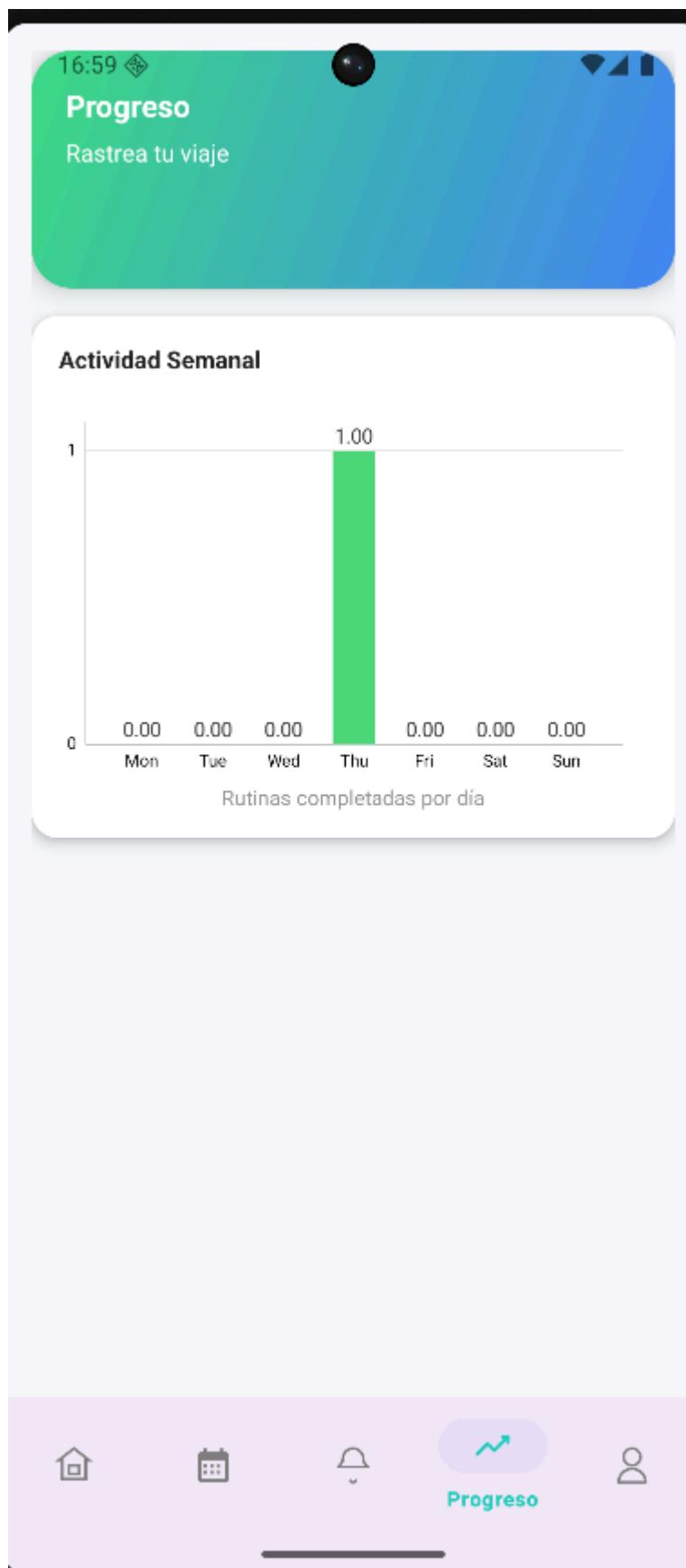


Imagen del fragmento de progreso mostrando el gráfico de barras con las rutinas completadas por día dentro de una semana.

## 5.2 Rutinas del día

Debajo del progreso semanal se muestra la lista de rutinas programadas para el día actual.

**Acciones disponibles:**

- Marcar una rutina como completada pulsando sobre ella.

The figure consists of two side-by-side screenshots of a mobile application interface. Both screenshots show a 'Panel de Usuario' (User Panel) at the top with a welcome message and a progress summary. Below this is a 'Progreso Diario' (Daily Progress) section with a circular progress bar. The left screenshot shows a grey progress bar with the text '¡Has completado 0/1 rutinas de hoy! (0%)'. The right screenshot shows a green progress bar with the text '¡Has completado 1/1 rutinas de hoy! (100%)'. At the bottom of each screenshot is a 'Rutinas de Hoy' (Daily Routines) section listing a single routine named 'Rutina 1' with details: '18:00' and 'defensa dia 12'. The left screenshot shows a grey circle next to 'Rutina 1', while the right screenshot shows a green circle with a checkmark next to it, indicating it has been checked off. The bottom navigation bar includes icons for 'Inicio' (Home), 'Calendario', 'Notificación', 'Más', and 'Perfil'.

**Sin hacer check de rutina**

**Check de rutina**

*Imagen mostrando el panel principal con progreso y lista de rutinas del día.*

---

## 6. Listado de Rutinas

### **Descripción:**

Pantalla que muestra todas las rutinas creadas por el usuario.

### **Elementos mostrados:**

- Nombre de la rutina.
- Hora asociada.
- Acciones de edición y eliminación.

### **Acciones disponibles:**

- Pulsa la rutina o lápiz para poder expandir la rutina y editar o cancelar

16:53



## Mis Rutinas

¡Hagamos de hoy un día increíble!

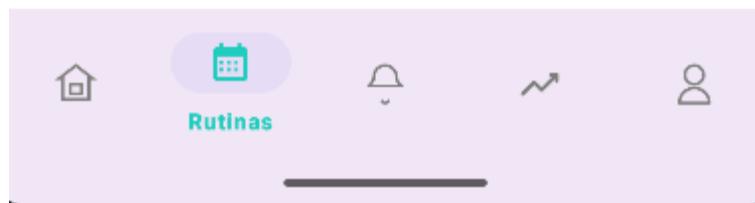
Gestiona tus hábitos diarios y programas

### Rutina 1

⌚ 18:00



+



**Imagen mostrando la lista completa de rutinas del usuario.**

## **7. Crear Rutina**

**Descripción:**

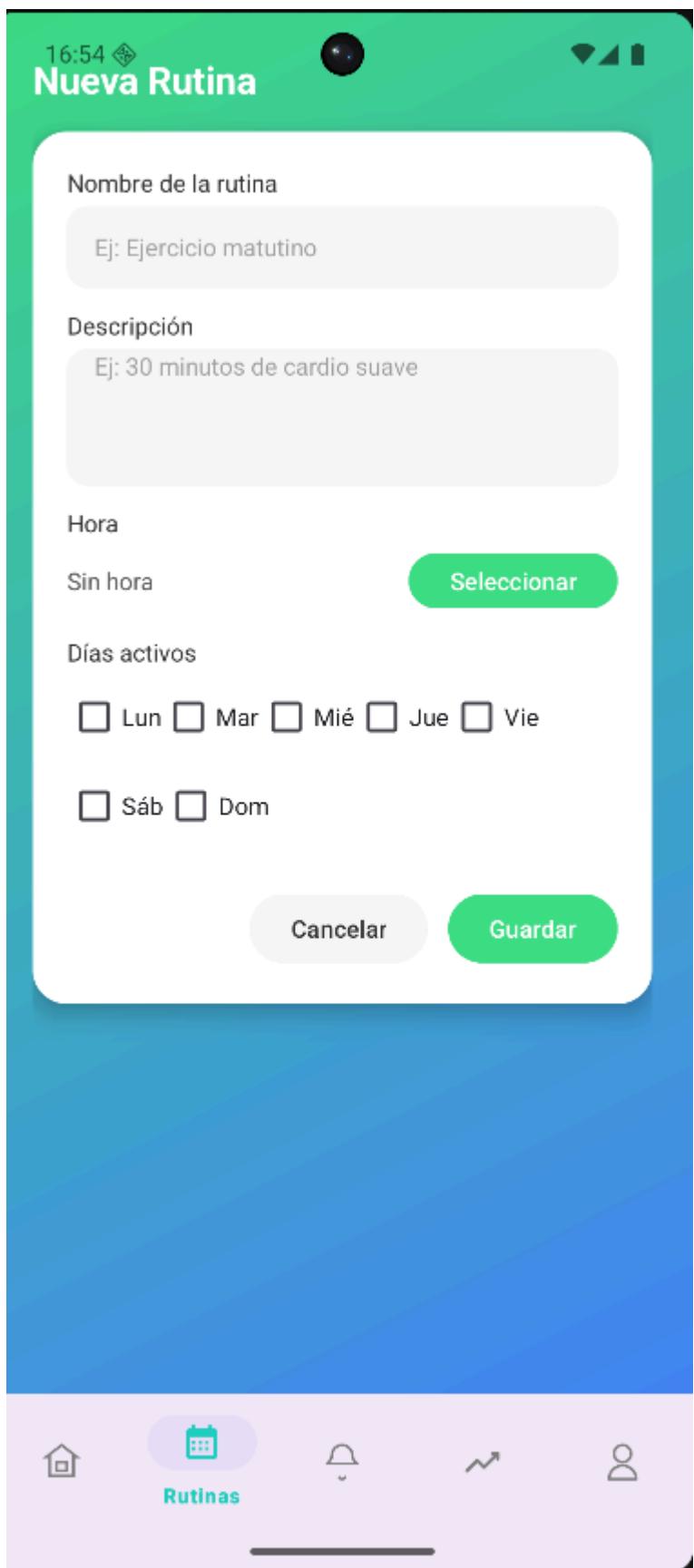
**Formulario para crear una nueva rutina.**

**Campos disponibles:**

- Nombre de la rutina.
- Descripción.
- Hora del recordatorio.
- Selección de días activos.

**Acciones disponibles:**

- Guardar rutina.
- Cancelar creación.



**Imagen del formulario de creación de rutina con todos los campos visibles.**

---

## **8. Editar Rutina**

**Descripción:**

Permite modificar una rutina existente.

**Acciones disponibles:**

- Modificar nombre y descripción.
- Cambiar hora.
- Cambiar días activos.
- Guardar cambios.
- Eliminar rutina.

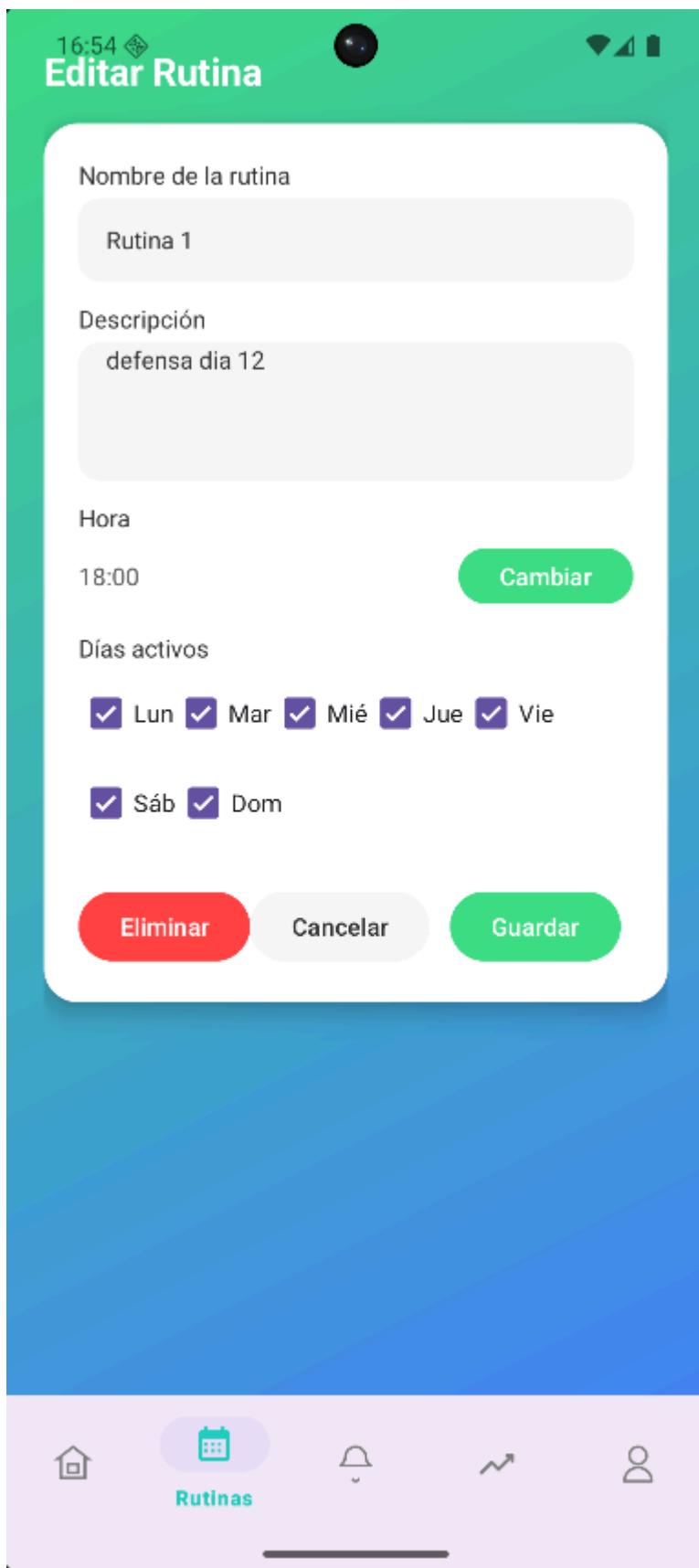


Imagen mostrando el formulario de edición de una rutina ya creada.

---

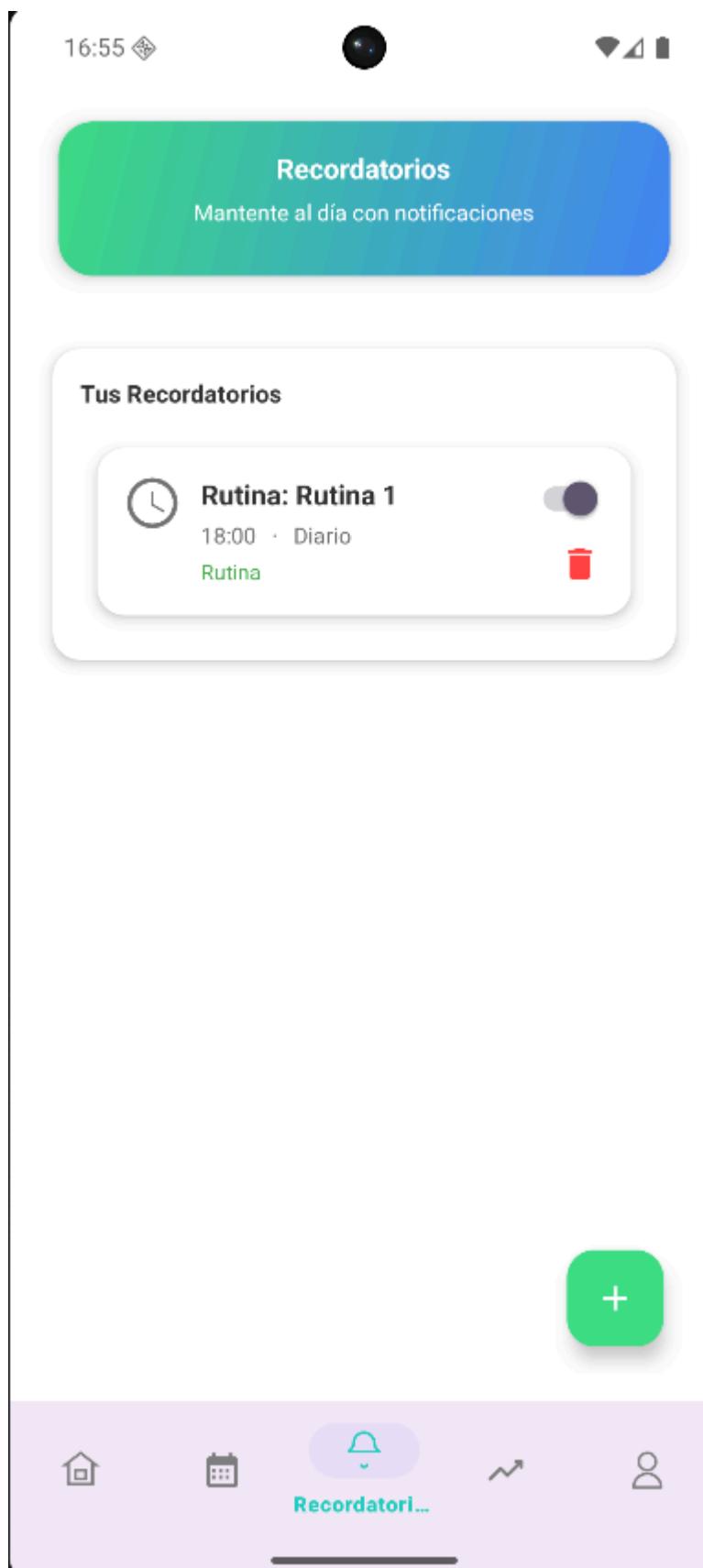
## **9. Listado de Recordatorios**

**Descripción:**

Pantalla que muestra los recordatorios creados por el usuario.

**Elementos mostrados:**

- Título del recordatorio.
- Hora y frecuencia.
- Categoría.
- Interruptor de activación.
- Botón de eliminación.



*Imagen mostrando la lista de recordatorios con interruptores activos/inactivos.*

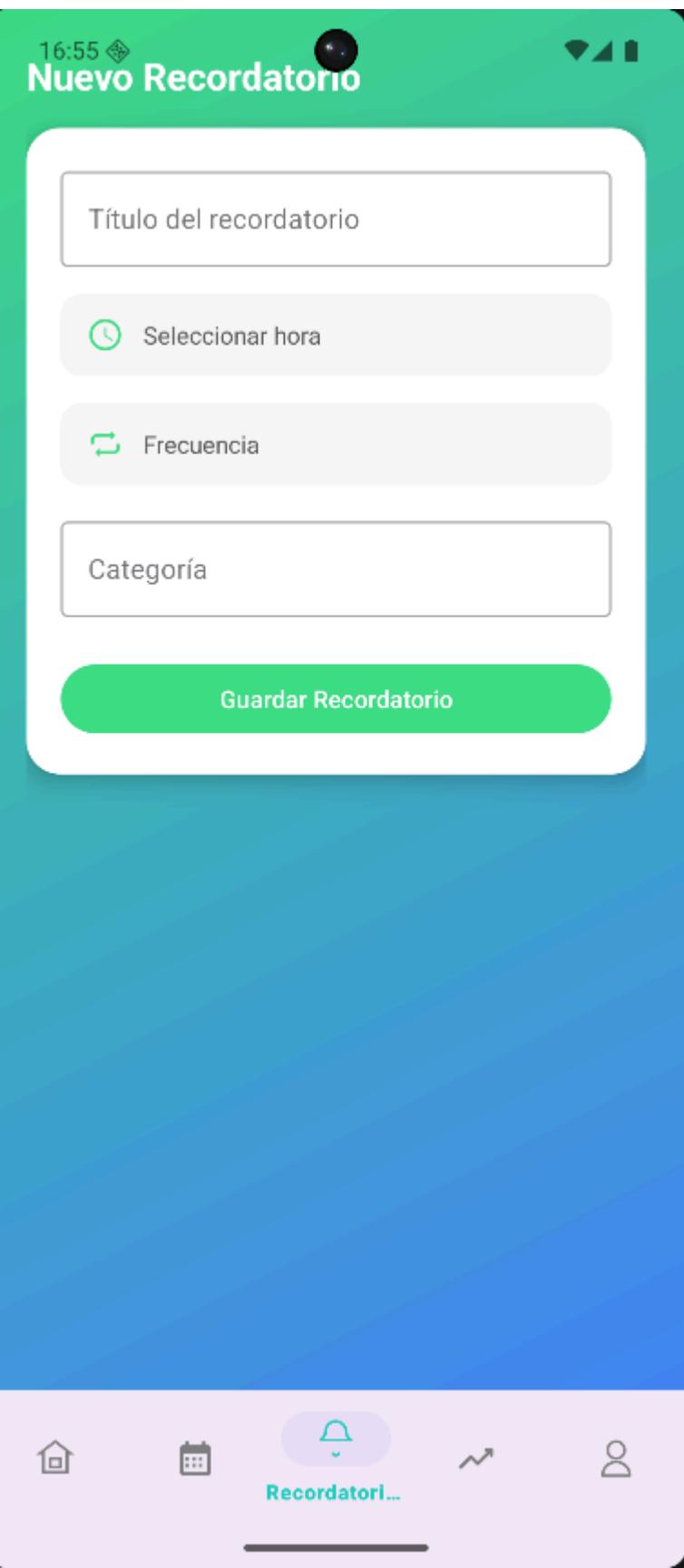
## **10. Crear Recordatorio**

**Descripción:**

**Formulario para crear un nuevo recordatorio con notificación.**

**Campos disponibles:**

- Título.
- Hora.
- Frecuencia.
- Categoría.



**Imagen del formulario de creación de recordatorio.**

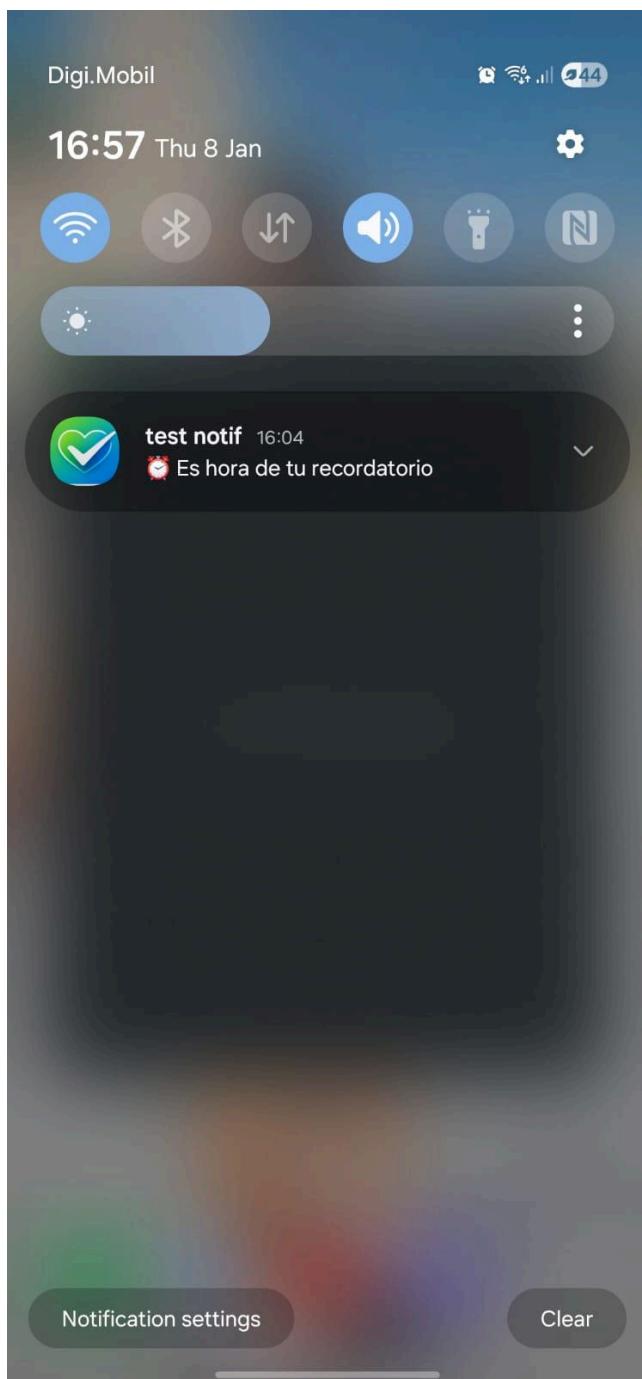
## 11. Notificación del Sistema

### Descripción:

Notificación que aparece cuando se cumple la hora de un recordatorio.

### Contenido:

- Título del recordatorio.
- Mensaje informativo.



*Imagen de la notificación emergente en el dispositivo Android.*

---

## **12. Perfil de Usuario**

**Descripción:**

Pantalla donde el usuario puede consultar su información básica.

**Acciones disponibles:**

- Visualizar datos personales.
- Cerrar sesión.

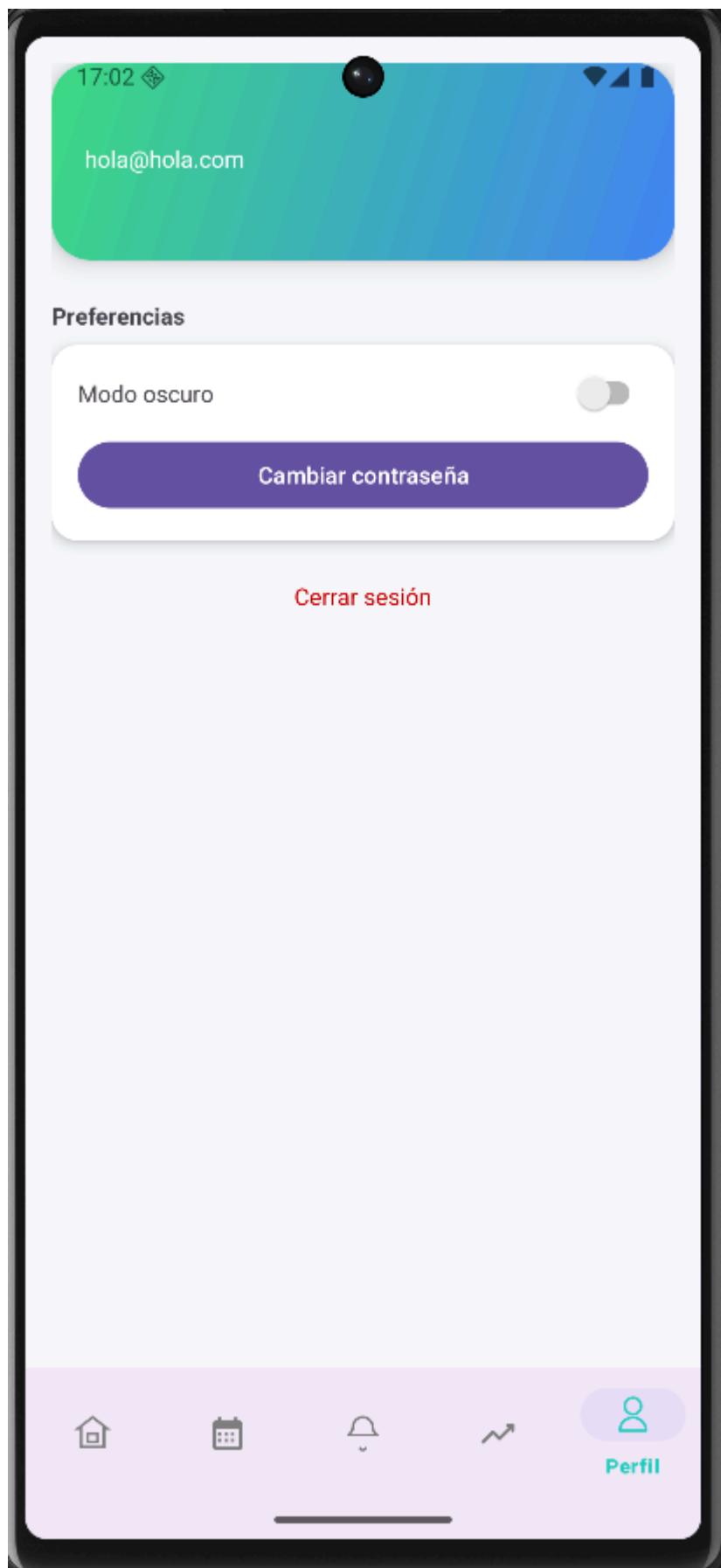


Imagen de la pantalla de perfil con opción de cerrar sesión.

---

## 13. Dashboard de Administrador

### **Descripción:**

**Pantalla exclusiva para usuarios con rol administrador, destinada a la supervisión y gestión general de la aplicación.**

### **Elementos mostrados:**

- Mensaje de bienvenida al administrador.
- Estadísticas generales del sistema:
  - Número total de usuarios.
  - Número total de rutinas.
  - Número total de recordatorios.
- Listado de usuarios registrados en la aplicación.

### 13.1 Gestión de usuarios y roles

Desde el panel de administrador, el administrador puede gestionar los usuarios registrados mediante un menú contextual asociado a cada usuario.

**Al mantener pulsado sobre un usuario del listado, se muestra un menú con las siguientes opciones:**

- Cambiar a usuario: asigna el rol de usuario estándar.
- Cambiar a administrador: asigna el rol de administrador.
- Eliminar usuario: elimina el usuario del sistema.

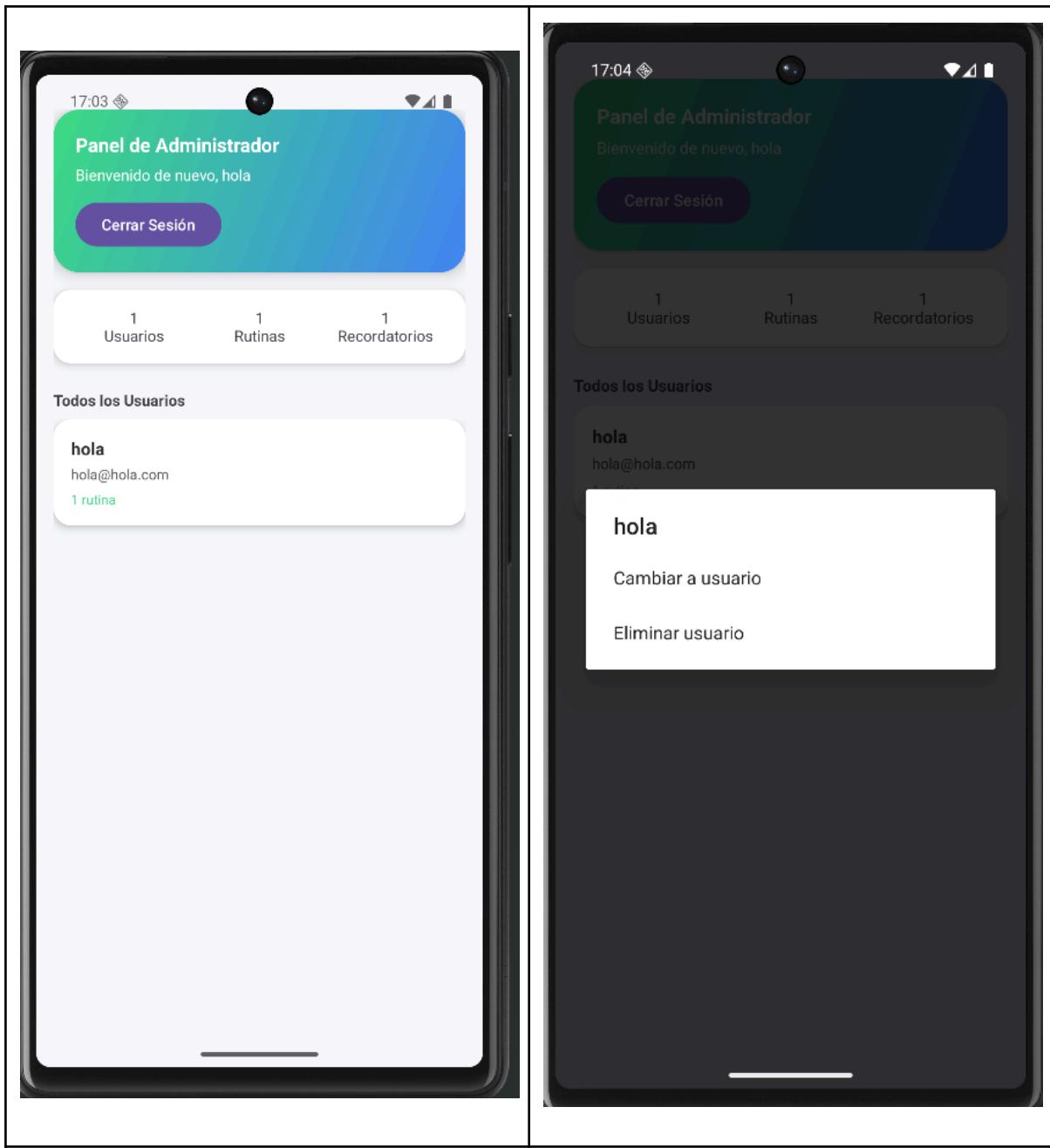
### **Roles disponibles:**

- **Usuario estándar.**
- **Administrador.**

### **Funcionamiento:**

- El administrador puede alternar el rol de un usuario según sea necesario.
- Los cambios de rol se guardan en la base de datos Firestore.
- El rol asignado determina el acceso al panel de administrador.

**Esta funcionalidad permite un control básico de permisos y la administración de usuarios desde la propia aplicación.**



**Imagen del dashboard de administrador con estadísticas y lista de usuarios.**

## 14. Cierre de Sesión

**El usuario puede cerrar sesión desde su perfil o desde el panel de administrador, regresando a la pantalla de inicio de sesión.**

## 15. Consideraciones sobre la Base de Datos (Firestore)

En este proyecto no es necesario aportar scripts de creación de base de datos, ya que FitLife utiliza Firebase Firestore, una base de datos NoSQL orientada a documentos, y no un sistema relacional como MySQL.

A diferencia de las bases de datos relacionales, en Firestore no existen sentencias SQL como:

- **CREATE DATABASE**
- **CREATE TABLE**

### Funcionamiento de Firestore

En Firestore, la estructura de la base de datos se genera de forma automática según el uso de la aplicación:

- Al escribir un documento, se crea automáticamente la colección.
- Al escribir un campo, se crea automáticamente dicho campo.
- No es necesario definir previamente esquemas, tablas o relaciones.

### Ejemplo real aplicado a FitLife

Cuando la aplicación ejecuta el siguiente código:

```
db.collection("users")
  .document(userId)
  .set(usuario);
```

Si la colección users no existe:

- Firestore crea automáticamente la colección users.
- Se crea el documento identificado por userId.
- Se crean todos los campos definidos en el objeto Usuario.

Este comportamiento justifica que no se incluya ningún script SQL en la documentación del proyecto, ya que la estructura de datos se define directamente desde el código Java de la aplicación.

---

## 16. Posibles mejoras y ampliaciones futuras

**Aunque FitLife cumple con los objetivos definidos para el proyecto FCT, la aplicación está diseñada de forma modular, lo que permite su ampliación y mejora en versiones futuras.**

A continuación se describen algunas mejoras que podrían implementarse:

## **16.1 Estadísticas avanzadas de progreso**

- Incorporar gráficos mensuales y anuales que permitan al usuario analizar su evolución a largo plazo.
- Comparativas entre semanas o meses para detectar tendencias de constancia.
- Resúmenes automáticos de progreso.

## **16.2 Integración con APIs externas**

- Integración con APIs de salud o actividad física para importar datos automáticamente.
- Uso de APIs de calendario para sincronizar rutinas y recordatorios.
- Integración con servicios externos que faciliten la personalización de hábitos.

## **16.3 Personalización de la experiencia de usuario**

- Configuración avanzada de notificaciones.
- Recomendaciones automáticas de rutinas en función del historial del usuario.
- Temas visuales personalizables.

## **16.4 Mejoras en el panel de administrador**

- Filtros avanzados en la gestión de usuarios.
- Visualización de estadísticas globales más detalladas.
- Exportación de datos para análisis externo.

**Estas mejoras no se han implementado en la versión actual por motivos de alcance del proyecto FCT, pero demuestran el potencial de crecimiento de la aplicación.**

---

## **17. Consideraciones finales**

FitLife es una aplicación funcional, clara y orientada a la gestión diaria de hábitos. Todas las acciones del usuario se almacenan de forma segura mediante Firebase Authentication y Firestore.

**Este manual refleja fielmente el estado actual de la aplicación en el momento**

# **18. DOCUMENTACIÓN TÉCNICA**

## **18. Documentación técnica**

### **18.1 Autenticación**

Se utiliza Firebase Authentication mediante correo electrónico y contraseña.

### **18.2 Roles de usuario**

El rol se almacena en Firestore y controla el acceso al panel de administrador.

### **18.3 Gestión de rutinas y progreso**

Las rutinas y el progreso se almacenan en Firestore y se sincronizan en tiempo real.

### **18.4 Notificaciones**

Se implementan mediante AlarmManager y BroadcastReceiver.

# **19. RETOS ENCONTRADOS**

Durante el desarrollo se afrontaron los siguientes retos:

- Uso de Firestore sin esquema fijo.
- Gestión de roles y permisos.
- Coordinación de fragmentos.
- Implementación de notificaciones persistentes.

## **19.1 Configuración de reglas de seguridad en Firebase Firestore**

Uno de los principales retos técnicos del proyecto ha sido la **configuración de las reglas de seguridad de Firebase Firestore**, necesarias para controlar correctamente los permisos de creación, lectura, modificación y eliminación de datos.

Firestore no utiliza un sistema de permisos tradicional, sino un sistema de **reglas de seguridad declarativas**, lo que obligó a realizar una fase previa de **investigación y pruebas** para comprender su funcionamiento y adaptarlo a los requisitos del proyecto.

Fue necesario definir reglas que permitiesen:

- Que los usuarios autenticados pudieran **crear y gestionar únicamente sus propios datos**.
- Que un usuario no pudiera acceder ni modificar información perteneciente a otros usuarios.
- Que los usuarios con rol **administrador** dispusieran de permisos adicionales para:
  - Leer todos los usuarios.
  - Modificar roles.
  - Eliminar usuarios.
  - Gestionar rutinas, progreso y recordatorios de cualquier usuario.

Para ello, se implementaron **funciones auxiliares** dentro de las reglas de Firestore que permiten comprobar si un usuario está autenticado y si dispone del rol de administrador, consultando su información almacenada en la colección **users**.

La correcta configuración de estas reglas fue esencial para garantizar la **seguridad de la aplicación**, evitando accesos no autorizados y asegurando que cada usuario solo pudiera operar sobre la información que le corresponde.

Este reto permitió profundizar en el uso de Firebase Firestore más allá del almacenamiento de datos, comprendiendo la importancia de la **seguridad y el control de accesos** en aplicaciones reales.

Reglas de seguridad de FitLifeApp:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    /* =====
       FUNCIONES AUXILIARES
       ===== */
    function isAuthenticated() {
      return request.auth != null;
    }
  }
}
```

```

function isAdmin() {
    return isAuthenticated()
        && get(
            /databases/${database}/documents/users/${request.auth.uid}
        ).data.rol == "admin";
}

/* =====
   USUARIOS
   ===== */
match /users/{userId} {

    // Crear usuario (registro)
    allow create: if isAuthenticated();

    // Leer su propio perfil
    allow read: if isAuthenticated()
        && request.auth.uid == userId;

    // Actualizar su propio perfil
    allow update: if isAuthenticated()
        && request.auth.uid == userId;

    // EL ADMIN PUEDE LEER TODOS
    allow read: if isAdmin();

    // EL ADMIN PUEDE BORRAR USUARIOS
    allow delete: if isAdmin()
        && request.auth.uid != userId; // evita auto-borrado
}

/* =====
   RUTINAS
   ===== */
match /routines/{rutinald} {

    // Crear rutina (solo dueño)
    allow create: if isAuthenticated()
        && request.auth.uid == request.resource.data.userId;

    // Leer / actualizar / borrar
    // - dueño
    // - admin
    allow read, update, delete: if isAuthenticated()
        && (
            request.auth.uid == resource.data.userId
            || isAdmin()
        );
}

/* =====
   PROGRESO
   ===== */
match /progress/{progressId} {

```

```
// Crear progreso (solo dueño)
allow create: if isAuthenticated()
  && request.auth.uid == request.resource.data.userId;

// Leer / actualizar / borrar
// - dueño
// - admin
allow read, update, delete: if isAuthenticated()
  && (
    request.auth.uid == resource.data.userId
    || isAdmin()
  );
}

/* =====
RECORDATORIOS
===== */
match /recordatorios/{recordatorioId} {

  // Crear recordatorio (solo dueño)
  allow create: if isAuthenticated()
    && request.auth.uid == request.resource.data.userId;

  // Leer / actualizar / borrar
  // - dueño
  // - admin
  allow read, update, delete: if isAuthenticated()
    && (
      request.auth.uid == resource.data.userId
      || isAdmin()
    );
}
}
```

