



MÓDULO PROYECTO

CFGS Desarrollo de Aplicaciones
Multiplataforma
Informática y Comunicaciones

FocusTrack

"Tu gestor de Cine"

Tutor individual: ROBERTO SANCHEZ DE LA ROSA

Tutor colectivo: CRISTINA SILVAN PARDO

Año: 2025

Fecha de presentación: 13/06/2025

Nombre y Apellidos: Ángel Chicote Veganzones

Email: angel.chv.01@gmail.com





Tabla de contenido

1	Identificación proyecto	4
2	Organización de la memoria	4
3	Descripción general del proyecto	4
3.1	Objetivos	4
3.2	Cuestiones metodológicas	4
3.3	Entorno de trabajo (tecnologías de desarrollo y herramientas).....	5
4	Descripción general del producto.....	6
4.1	Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.....	6
4.2	Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.....	7
4.3	Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha.....	8
5	Planificación y presupuesto	9
5.1	Planificación	9
5.2	Presupuesto.....	9
6	Documentación Técnica: análisis, diseño, implementación y pruebas.....	11
6.1	Especificación de requisitos	11
6.2	Análisis del sistema	11
6.3	Diseño del sistema:	12
6.4	Implementación:	14
6.4.1	Entorno de desarrollo.....	14
6.4.2	Estructura del código.....	14
6.4.3	Cuestiones de diseño e implementación reseñables.....	14
6.5	Pruebas.....	15
7	Manuales de usuario	15



7.1	Manual de usuario.....	15
7.2	Manual de instalación	27
8	Conclusiones y posibles ampliaciones	27
10	Bibliografía	29
11	Anexos	30



1 Identificación proyecto

2 Organización de la memoria

3 Descripción general del proyecto

3.1 Objetivos

Objetivo principal

- Desarrollar la aplicación móvil **FocusTrack**, que permita a los usuarios explorar, descubrir, buscar y organizar películas de manera cómoda e intuitiva a través de listas personalizadas.

Objetivos secundarios

1. Integrar una **API externa de películas (TMDb)** para obtener información detallada y actualizada.
2. Implementar un sistema de **autenticación y gestión de usuarios** mediante **Firebase Authentication**, con opciones de correo/contraseña y Google Sign-In.
3. Implementar funcionalidades de **autocompletado** de credenciales y uso de **Credential Manager** para mejorar la experiencia de inicio de sesión.
4. Diseñar una interfaz **minimalista, accesible e internacionalizada**, siguiendo **Material Design 3** y adaptándose dinámicamente al tema del sistema.
5. Permitir la creación, edición y eliminación de **listas personalizadas**, y la gestión de películas dentro de ellas.
6. Aprender a crear un proyecto con buenas prácticas de diseño y arquitectura, separando la lógica de la aplicación de los servicios utilizados.
7. Desplegar la aplicación en **Google Play Store**, cumpliendo todos los requisitos de publicación (categoría y contenido de la aplicación, ficha de play store con: gráfico de funciones, capturas. Política de privacidad, vinculación con GitHub Pages, Play App Signing, Play Integrity...).
8. Documentar el proceso de desarrollo y elaborar la memoria del proyecto.

3.2 Cuestiones metodológicas

Para la realización de este proyecto se ha seguido un modelo de ciclo de vida en cascada con iteraciones de revisión:



1. Análisis de requisitos: recabar información sobre funcionalidades deseadas (búsqueda, listas, recomendaciones).
2. Diseño: definición de la arquitectura (cliente-servidor), esquemas de navegación (NavHost), estructura de carpetas y diseño de interfaz.
3. Implementación: desarrollo en Kotlin con Jetpack Compose, separación en capas (Core, data, domain, ui), uso de Hilt para inyección de dependencias y Retrofit para las peticiones a la API de TMDb, Firebase para la autenticación, Firebase Firestore para la persistencia de datos, Firebase Crashlytics para las analíticas de fallos en la aplicación una vez publicada.
4. Despliegue: generación de App Bundle firmado, configuración de Play Console (pruebas internas, metadata, integridad).

3.3 Entorno de trabajo (tecnologías de desarrollo y herramientas)

Modelo arquitectónico

- Cliente-servidor: la aplicación consume servicios REST de TMDb y utiliza Firebase (Auth, Firestore) como backend.
- Distribuido: componentes desacoplados en servidor (Firebase) y cliente (Android App).

Tecnologías y lenguajes

- Kotlin (v2.0.21) con Jetpack Compose para UI declarativa.
- Firebase Authentication y Firestore (Firebase BOM v33.14.0).
- TMDb API mediante Retrofit 2 (v2.9.0) y OkHttp (v4.11.0).
- Hilt (v2.51) para inyección de dependencias.
- Coil (v2.4.0) para la obtención y muestra de imágenes.
- Credential Manager (AndroidX Credentials v1.5.0) para la gestión de credenciales.
- Material Design 3 con tema adaptativo (Material3 Compose, Material You).

Herramientas de desarrollo

- Android Studio Arctic Fox o superior.
- Gradle (AGP v8.10.1) con Kotlin DSL + Version Catalog.
- GitHub para control de versiones y GitHub Pages para alojar la política de privacidad y mostrar la aplicación de manera más visual.
- Keytool para generación de claves (upload key) y Play App Signing en Google Play Console.
- Notion para apuntes y seguimiento del avance.



- Microsoft Word para la redacción de la memoria.

4 Descripción general del producto

4.1 *Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.*

FocusTrack está concebido como una aplicación **Android** que ofrece al usuario final un conjunto de funcionalidades enfocadas a la exploración, organización y seguimiento de películas. Se establecen los siguientes límites y elementos clave:

Límites del sistema:

- Se ejecuta exclusivamente en dispositivos con Android API level 24 / Android 7 (nougat) o superior.
- Requiere conexión a Internet para consumir la API de TMDb y para sincronizar datos con Firebase.
- No incluye almacenamiento offline completo; solo se almacenan en caché mínimos datos temporales.

Funcionalidades básicas:

- Autenticación de usuarios: registro, login con correo/contraseña y Google Sign-In, autocompletado y almacenamiento de credenciales.
- Exploración de películas: secciones de tendencias, populares, mejor valoradas, en cines y próximas a estrenarse.
- Búsqueda: campo de búsqueda con sugerencias en tiempo real y resultados listados.
- Gestión de listas personalizadas: creación y eliminación de listas, adición/eliminación de películas.
- Detalle de película: visión completa de metadatos (título, sinopsis, géneros, fecha, puntuación, presupuesto, recaudación, etc.).
- Interfaz adaptativa: uso de Material Design 3 para seguir las guías de accesibilidad y adaptación de tema.
- Navegación avanzada: gestión independiente por cada sección y comportamiento personalizado al navegar a la pantalla anterior.

Usuarios y roles:

Usuario registrado: actualmente solo se dispone de este tipo de cuenta, que accede a todas las funcionalidades.



Sistemas externos:

API TMDb: fuente de datos para películas y metadatos.

Firebase: autenticación, Firestore para sincronización de listas, Crashlytics.

GitHub Pages: alojamiento de la política de privacidad y la web.

4.2 Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.

Arquitectura por capas (Clean Architecture simplificada):

- Capa Core: lógica de negocio y utilidades genéricas (errores, autenticación y navegación).
- Capa Data: repositorios y servicios remotos.
- Capa Domain: modelos de datos y DTOs.
- Capa UI: componentes y pantallas en Compose.

Separación de lógica de negocio con servicios:

- Servicios y repositorios.
- DTO's y clases modelo.
- Excepciones personalizadas.

Inyección de dependencias: Hilt para gestionar instancias de ViewModel, repositorios y servicios.

Comunicación con backend: Retrofit + OkHttp para llamadas HTTP, manejo de respuestas y errores.

Internacionalización: uso de strings.xml para multi-idioma (es-ES, en-US).

Responsive UI:

- Compose adaptativo.
- “WindowInsets” para áreas seguras:
 - Límites físicos: cámara o Notch.
 - límites Android: barra superior con estado o notificaciones, barra inferior para navegación del sistema.

Manejo de estados: arquitectura MVVM con StateFlow y mutableStateOf.



4.3 Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha

Plataforma tecnológica:

- Mínimo: Android 7 / API Level 24.
- Objetivo: Android 11.

Proceso de despliegue:

1. Crear una cuenta de Desarrollador de Google Play.
2. Completar toda la información solicitada por Google.
3. Configurar huellas digitales del certificado SHA de Firebae, que aparece en Google Play Console después de los pasos previos.
4. Agregar Google-services.json proporcionado por Firebase.
5. Compilar (build de reléase) con el comando: `./gradlew buldeRelease`.
6. Firmar con upload Key y Google Play App Signing.
7. Subir App Bundle a Google play Console.
8. Crear prueba cerrada en Google Play Console.
9. Invitación a testers y verificación en dispositivo real.
10. Publicación en producción tras pruebas cerradas.

Proceso de instalación:

- Una vez publicada la versión, se podrá descargar desde Google Play Store.
- Instalar directamente la APK (no recomendado, ya que Google protege que se instale la aplicación desde fuera de Play Store), desde el repositorio en la raíz: [app-release.apk](#).
- Clonar el repositorio de GitHub desde Android Studio.

Puesta en marcha:

- Iniciar sesión o crear una cuenta, mediante correo y contraseña o Google-SignIn (Recomendado).

5 Planificación y presupuesto

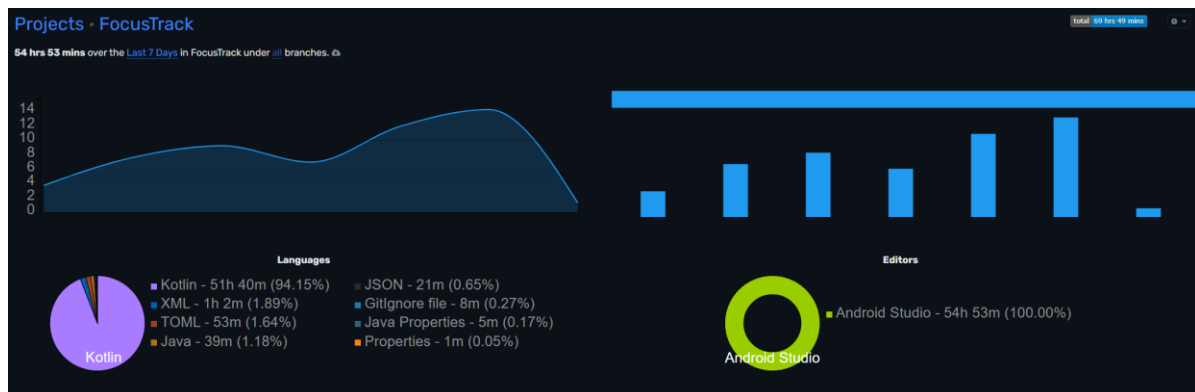
5.1 Planificación

Para organizar el desarrollo de FocusTrack se ha elaborado un **cronograma de actividades** estructurado por semanas y meses, detallado a continuación mediante un diagrama de Gantt simplificado y una tabla de seguimiento.

Tarea	Actividades principales
Investigación	Estudio de aplicaciones similares; revisión de TMDb y Firebase (3 semana)
Diseño arquitectónico	Definición de capas, navegación (1 semanas)
Desarrollo Frontend	Implementación de UI en Compose y componentes (2 semanas)
Desarrollo Backend	Integración con TMDb y Firebase; repositorios y servicios (1 semanas)
Despliegue	Configuración de firma, Play Console, publicación interna (1 semana)
Documentación	Redacción de memoria, manuales y anexos (1 semanas)

5.2 Presupuesto

Tiempo dedicado: 69h 49 min





Licencia Desarrollador GooglePlay	25 €	total	25
Total, estimado primer año			2.067

6 Documentación Técnica: análisis, diseño, implementación y pruebas.

6.1 Especificación de requisitos

6.2 Análisis del sistema

Especificación de requisitos funcionales

A partir de las funcionalidades básicas descritas en el apartado 4, se detallan a continuación los requisitos funcionales de FocusTrack:

1. Autenticación de usuarios

- **Registro:** formulario con campos obligatorios (email, contraseña) o Google Sign-In.
 - Validaciones:
 1. Email con formato válido.
 2. Contraseña con mínimo 6 caracteres.
- **Login:** credenciales correo/contraseña; opción Google Sign-In.
- **Autocompletado:** uso de Credential Manager para sugerir cuentas guardadas.

2. Gestión de películas

- **Listado por categorías:** mostrar tendencias, populares, mejor valoradas, en cines y próximas.
- **Búsqueda:** campo de texto con sugerencias en tiempo real. Visualización de resultados en una lista de previsualizaciones de los detalles de las películas.
- **Detalle de película:** al seleccionar, mostrar metadatos, imagen y enlaces externos.
- **Añadir película a listas:** bottom sheet con checkbox para cada lista del usuario.

3. Listas personalizadas

- **Crear lista:** alert dialog solicita nombre (no vacío).
- **Ver listas:** grid de listas con nombre.



- **Eliminar película:** botón Clear en cada tarjeta; confirma en dialog.
- **Eliminar lista:** opción en detalle de lista; confirma en dialog.

4. Sincronización y cache

- **Firestore:** guardar IDs de películas por usuario y lista.
- **Cache de imágenes:** Coil gestiona almacenamiento local de pósters.

6.3 Diseño del sistema:

6.3.1 Diseño de la Base de Datos

Tipo: NoSQL, orientado a documentos

- **Modelo E/R:** Entidad Usuario —< Lista >— Películas;
- Atributos principales:
 - Usuario (gestionado por Firestore)
 - Lista (id del documento generado por Firestore, nombre)
 - Película (id de tmdb)
- **Esquema en Firestore:** colección users/{userId}/lists/{listId} con campo movieIds (array).
- **Reglas de acceso:**

```
rules_version = '2';
```

```
service cloud.firestore {
```

```
  match /databases/{database}/documents {
```

```
    match /users/{userId}/lists/{listId} {
```

```
      allow read, write: if request.auth != null && request.auth.uid == userId;
```

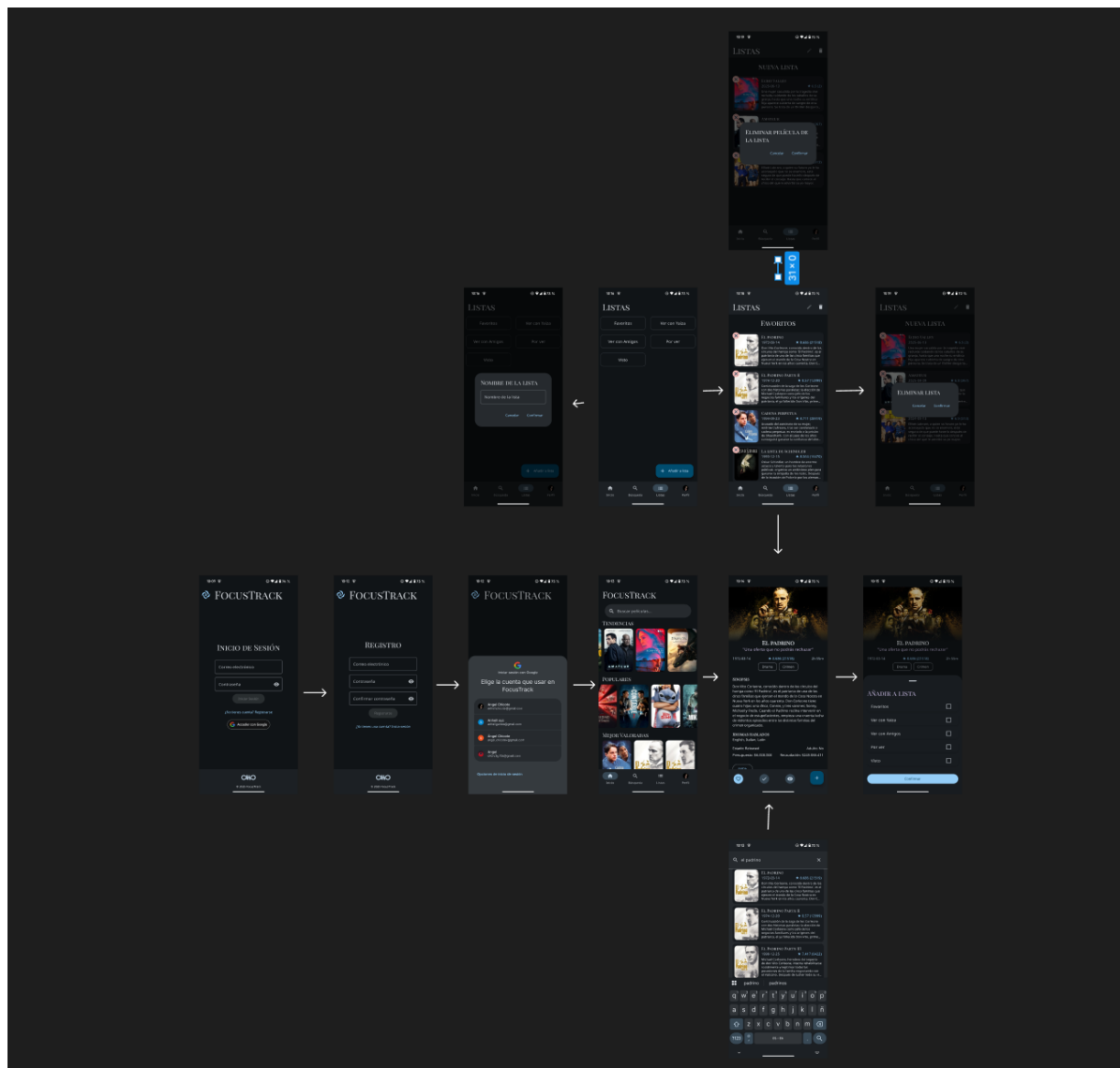
```
    }
```

```
  }
```

```
}
```

6.3.2 Diseño de la Interfaz de usuario

- **Wireframes:**



- **Diagrama de flujo:** uso de NavHost para gestionar rutas por sección; cada flujo independiente.

6.3.3 Diseño de la Aplicación

- **Procesos clave:**
 - Carga inicial: onLaunch en Splash, valida sesión y redirige.
 - Carga de secciones: HomeViewModel consulta TMDb y expone estado UI.
 - Transición a detalle: onClick emite evento navigation.
 - Gestión de listas: ListDetailViewModel sincroniza Firestore y actualiza UIState.



6.4 Implementación:

6.4.1 Entorno de desarrollo.

- **IDE:** Android Studio Electric Eel
- **SDKs:** Android SDK 35, Kotlin 2.0.4 , Compose Compiler 1.4
- **Control de versiones:** Git + GitHub

6.4.2 Estructura del código.

- **Librerías:**
 - Firebase (Auth, Firestore)
 - Retrofit + OkHttp
 - Hilt
 - Coil
- **Patrones:** MVVM, repositorios, inyección de dependencias.
- **Orientación:** mezcla de orientación a objetos (modelos) y declarativa (Compose).

6.4.3 Cuestiones de diseño e implementación reseñables.

- **Material You:** tema dinámico.
- **Navegación avanzada:** con gestión de navegación independiente por sección.
- **Seguridad:** autenticación mediante credential manager.
- **Scaffold dinámico:** único Scaffold adaptativo según ruta/flow, reduciendo duplicación.
- **BottomSheet:** uso de ModalBottomSheetLayout para selección múltiple de listas.
- **SafeArgs:** extensiones NavControllerExtensions para tipo seguro de rutas.
- **Logo app y marca personal:** se han realizado a mano con InkScape.






6.5 Pruebas.

7 Manuales de usuario

7.1 Manual de usuario

Inicio de sesión:



The screenshot shows the FocusTrack login interface on a mobile device. At the top, the status bar displays the time 10:09, signal strength, Wi-Fi, and battery level at 74%. The FocusTrack logo is prominently displayed at the top of the app screen. Below it, the text 'INICIO DE SESIÓN' is centered. There are two input fields: 'Correo electrónico' and 'Contraseña', the latter with a toggle eye icon. A dark 'Iniciar Sesión' button is positioned below the fields. A link for registration is present, followed by a 'G Acceder con Google' button. The bottom of the screen features the CHVO logo and the copyright notice '© 2025 FocusTrack'.



Registro:

10:12 

 FOCUSTRACK

REGISTRO

Correo electrónico

Contraseña 

Confirmar contraseña 

Registrarse

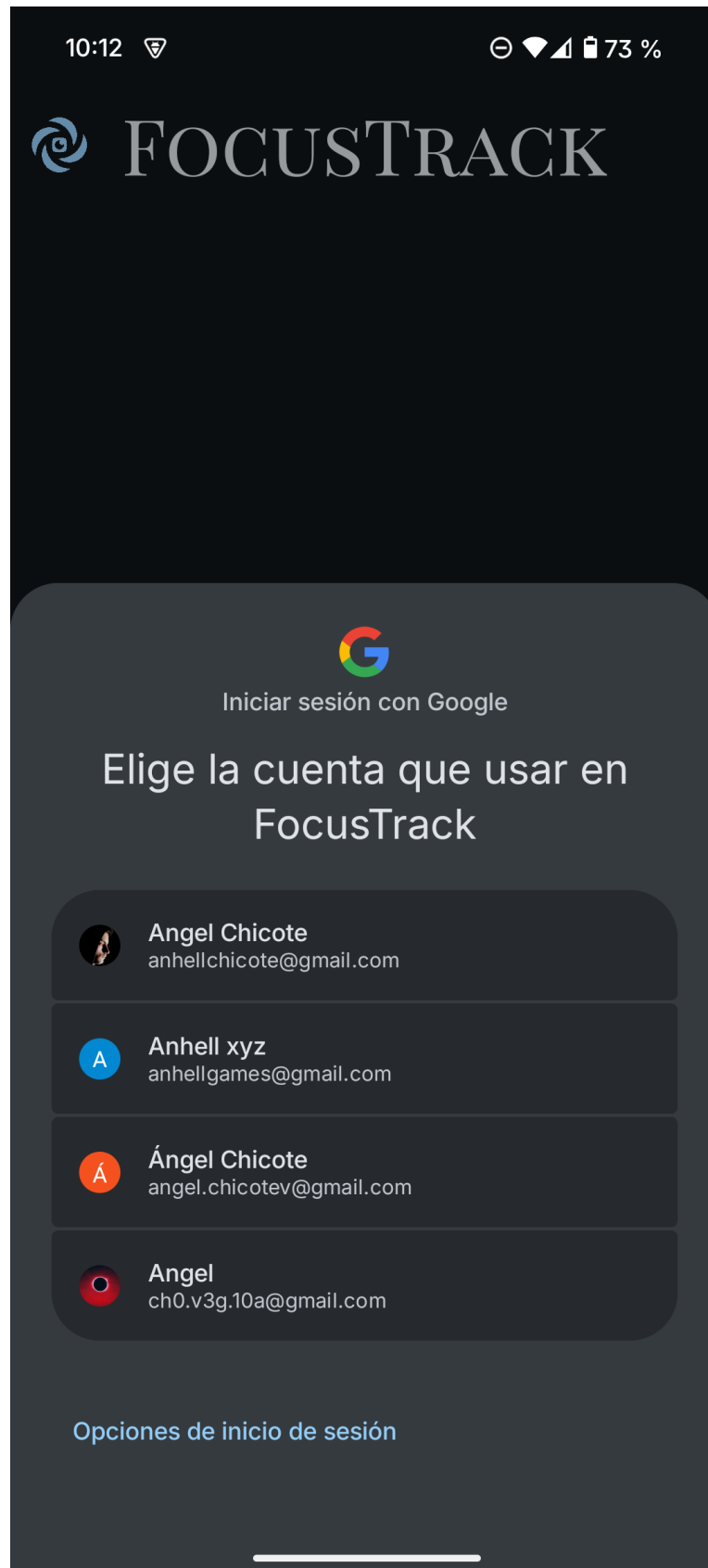
[¿Ya tienes una cuenta? Inicia sesión](#)



© 2025 FocusTrack



Credential Manager:





Home:






Detalle película:

10:14

73 %



EL PADRINO

"Una oferta que no podrás rechazar"

1972-03-14

★ 8.686 (21518)

2h 55m

Drama

Crimen

SINOPSIS

Don Vito Corleone, conocido dentro de los círculos del hampa como 'El Padrino', es el patriarca de una de las cinco familias que ejercen el mando de la Cosa Nostra en Nueva York en los años cuarenta. Don Corleone tiene cuatro hijos: una chica, Connie, y tres varones; Sonny, Michael y Fredo. Cuando el Padrino reclina intervenir en el negocio de estupefacientes, empieza una cruenta lucha de violentos episodios entre las distintas familias del crimen organizado.

IDIOMAS HABLADOS

English, Italian, Latin

Estado: Released

Adulto: No


Presupuesto: \$6.000.000





Recaudación: \$245.066.411


IMDb



Añadir película a las listas:

10:15 

    73 %



EL PADRINO

"Una oferta que no podrás rechazar"

1972-03-14★ 8.686 (21518)2h 55m

Drama

Crimen

AÑADIR A LISTA

Favoritos

☐

Ver con Yaiza

☐

Ver con Amigos

☐

Por ver

☐

Visto

☐

Confirmar



Búsqueda:

10:13
73 %

el padrino

EL PADRINO

1972-03-14 ★ 8.686 (21519)

Don Vito Corleone, conocido dentro de los círculos del hampa como 'El Padrino', es el patriarca de una de las cinco familias que ejercen el mando de la Cosa Nostra en Nueva York en los años cuarenta. Don C...

EL PADRINO PARTE II

1974-12-20 ★ 8.57 (12999)

Continuación de la saga de los Corleone con dos historias paralelas: la elección de Michael Corleone como jefe de los negocios familiares y los orígenes del patriarca, el ya fallecido Don Vito, prime...

EL PADRINO PARTE III

1990-12-25 ★ 7.417 (6422)

Michael Corleone, heredero del imperio de don Vito Corleone, intenta rehabilitarse socialmente y legitimar todas las posesiones de la familia negociando con el Vaticano. Después de luchar toda su vi...

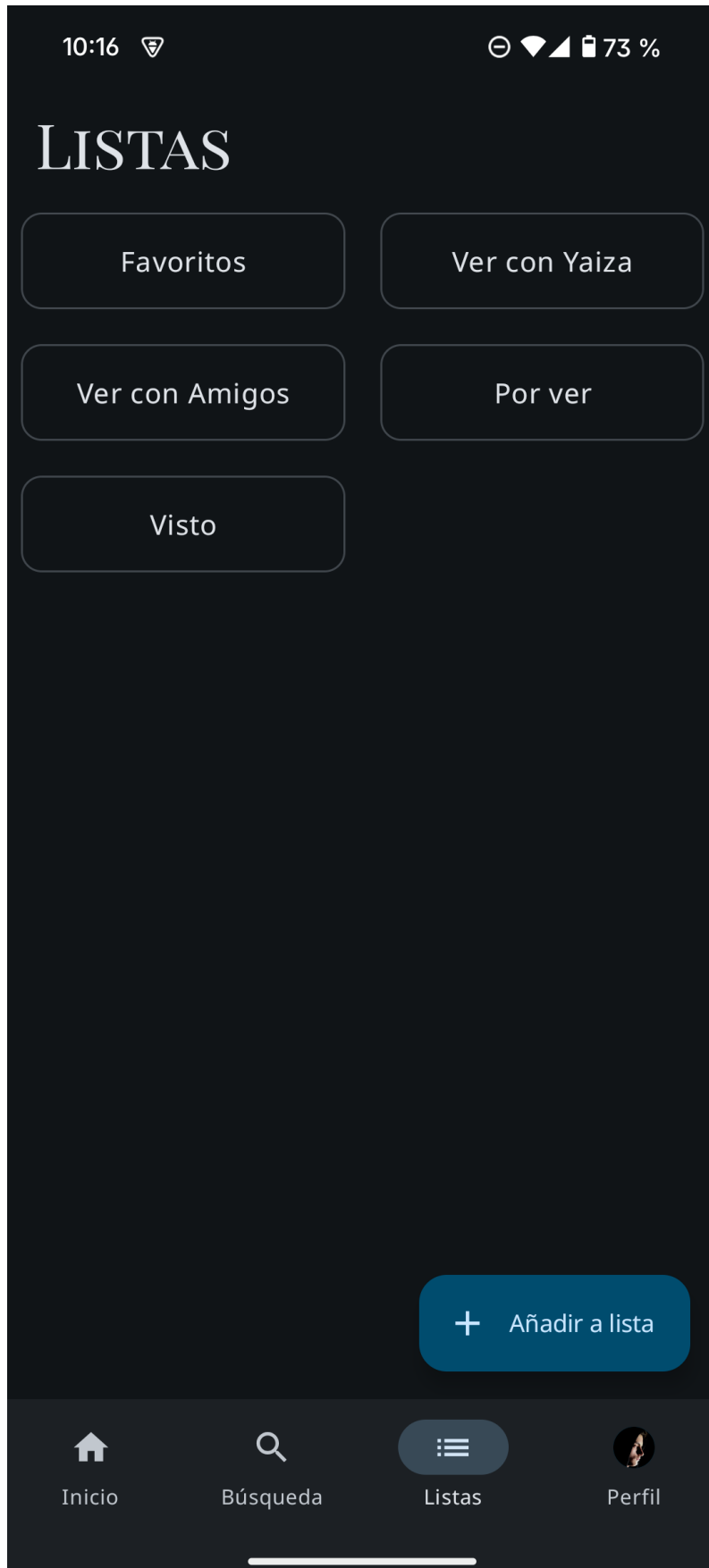
padrino | padrinos

q w e r t y u i o p
a s d f g h j k l ñ
z x c v b n m
?123 , .

ES • EN



Listas:





Detalle Lista:

10:18
72 %

LISTAS

FAVORITOS

EL PADRINO
1972-03-14 ★ 8.686 (21518)

Don Vito Corleone, conocido dentro de los círculos del hampa como 'El Padrino', es el patriarca de una de las cinco familias que ejercen el mando de la Cosa Nostra en Nueva York en los años cuarenta. Don C...

EL PADRINO PARTE II
1974-12-20 ★ 8.57 (12999)

Continuación de la saga de los Corleone con dos historias paralelas: la elección de Michael Corleone como jefe de los negocios familiares y los orígenes del patriarca, el ya fallecido Don Vito, prime...

CADENA PERPETUA
1994-09-23 ★ 8.711 (28419)

Acusado del asesinato de su mujer, Andrew Dufresne, tras ser condenado a cadena perpetua, es enviado a la prisión de Shawshank. Con el paso de los años conseguirá ganarse la confianza del dire...

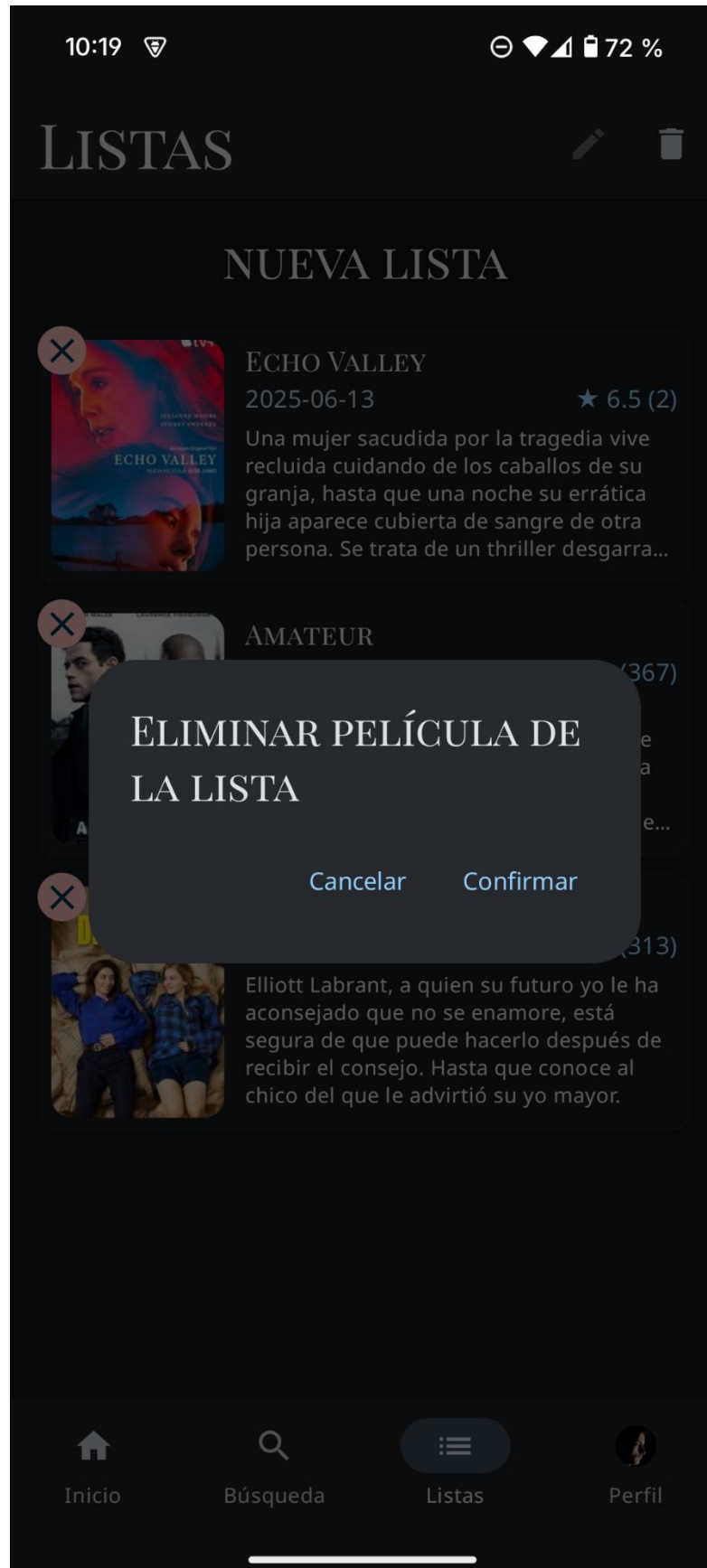
LA LISTA DE SCHINDLER
1993-12-15 ★ 8.564 (16470)

Oskar Schindler, un hombre de enorme astucia y talento para las relaciones públicas, organiza un ambicioso plan para ganarse la simpatía de los nazis. Después de la invasión de Polonia por los aleman...

Inicio
Búsqueda
Listas
Perfil

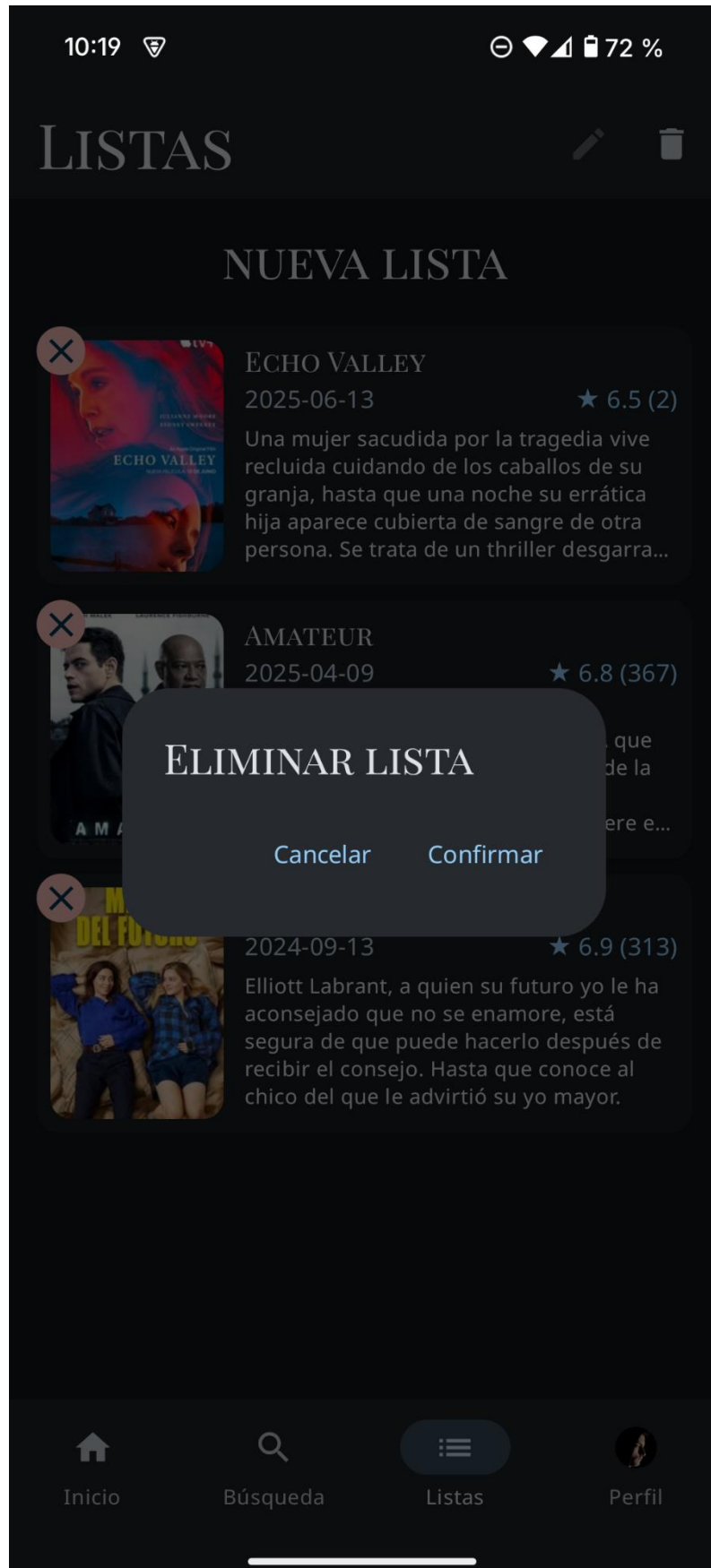


Eliminar película de la lista:



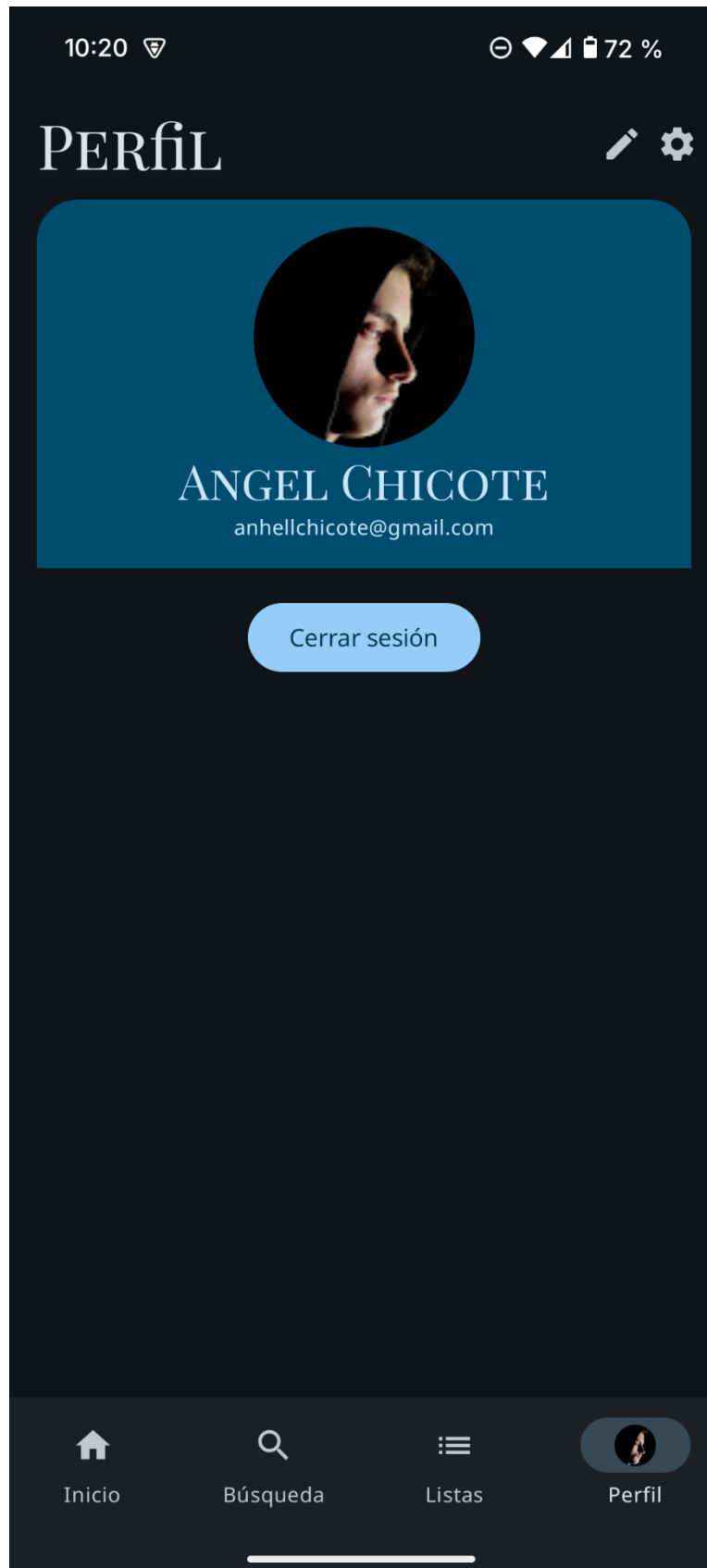


Eliminar lista:





Perfil:





7.2 Manual de instalación

El manual de instalación describe cómo desplegar y arrancar la aplicación en un dispositivo Android:

Requisitos

- Dispositivo con Android 7.0 (API 24) o superior.
- Conexión a Internet activa.

Instalación desde Play Store (prueba interna o producción)

- Recibir enlace de invitación o entrar en Google Play.
- Pulsar Instalar.
- Permitir permisos de Internet cuando el sistema lo solicite.

Instalación manual (APK/AAB local)

- Copiar `app-release.apk` al dispositivo o usar ADB:
- `adb install app-release.apk`
- Hay que asegurar que en ajustes de seguridad está habilitada "Permitir instalación de orígenes desconocidos".

Primer arranque

- Abrir la aplicación desde el cajón de apps.
- Conceder permisos (si se solicitan).
- Seguir manual de usuario para registro e inicio de sesión.

8 Conclusiones y posibles ampliaciones

Al finalizar el proyecto FocusTrack, se pueden extraer diversas conclusiones que reflejan tanto los aciertos como los retos enfrentados durante el desarrollo:

1. Cumplimiento de objetivos:

- **Objetivos principales:** la aplicación permite explorar, buscar y organizar películas mediante listas personalizadas de forma intuitiva, satisfaciendo la necesidad inicial de un gestor de películas minimalista.
- **Objetivos secundarios:** se integró con éxito la API de TMDb, se implementó autenticación segura con Firebase y Google Sign-In, y se creó una UI adaptable a Material Design 3.



- **Evaluación:** todas las funcionalidades definidas en los requisitos iniciales están operativas, excepto poder compartir listas, recomendaciones basadas en los gustos del usuario con una posible implementación de ia y una feed donde se puedan ver trailers.

2. Calidad de la arquitectura y código:

- La adopción de **MVVM** y la inyección de dependencias con **Hilt** facilitaron la modularidad y el testeo.
- La separación en capas (core, data, domain, ui) redujo el acoplamiento y mejoró la mantenibilidad.
- **Lecciones:** fue esencial definir claramente las responsabilidades de cada capa para evitar la proliferación de lógica en la capa UI.

3. Experiencia de usuario (UX/UI):

- **Compose** y Material You proporcionaron una experiencia fluida y coherente.
- El tema dinámico adaptado al fondo de pantalla del usuario mejoró la personalización.
- **Feedback:** mediante Crashlytics, si hay errores en la aplicación, se pueden solucionar rápida y eficazmente.

4. Rendimiento y fiabilidad:

- **Mejora:** prever el uso intensivo de Firestore en listas largas para optimizar consultas con paginación.

5. Desafíos y dificultades:

- **Manejo de estados:** conciliar múltiples flujos de navegación y estados UI en Compose requirió iteraciones y refactorizaciones.
- **Configuración de publicación:** la integración con Play Console (App Signing, Play Integrity, políticas) resultó compleja y consumió un 20% más del tiempo del previsto.
- **Sincronización en tiempo real:** garantizar coherencia entre UIState y Firestore implicó gestionar correctamente los StateFlow y evitar condiciones de carrera.

6. Grado de satisfacción:

- El autor considera que el proyecto ha cumplido con los estándares de calidad esperados: funcionalidad completa, estabilidad y experiencia de usuario cuidada.
- Se alcanzó un alto nivel de dominio de tecnologías avanzadas (Compose, Hilt, Firebase), cumpliendo el objetivo de aprendizaje.



- Diseño totalmente personalizado (Logo aplicación y marca personal de diseño propio)

9 Posibles ampliaciones

- **Funcionalidad social:** añadir un módulo de seguidores, feed de actividad y oportunidades de intercambio de listas.
- **Recomendaciones basadas en IA:** entrenar modelos simples en backend para sugerir películas según historial de visualización.
- **Modo offline:** implementar cache de datos esenciales (listas y últimos resultados de búsqueda) para uso sin conexión.
- **Internacionalización avanzada:** añadir más idiomas (francés, alemán) y gestionarlos vía Remote Config.
- **Integración de streaming:** liga con APIs de servicios de video (YouTube, Vimeo) para reproducir trailers dentro de la app.
- **Diferentes permisos en listas:** que existan diferentes roles para los usuarios que comparten una lista.
- **Recordatorios de estrenos.**
- **Permitir claves de acceso.**

10 Bibliografía

A continuación se recogen las fuentes consultadas durante el desarrollo de FocusTrack, incluyendo documentación oficial, artículos técnicos y tutoriales:

1. **Jetpack Compose Documentation.** Android Developers. Google. 2025.
<https://developer.android.com/jetpack/compose>
<https://developer.android.com/develop?hl=es-419>
<https://developer.android.com/samples>
2. **Google Play console:** <https://play.google.com/console/about/internal-testing/>
https://play.google.com/intl/es_ALL/console/about/production/
3. **Material Design 3 Guidelines.** Material.io. Google. 2024.
<https://m3.material.io>
4. **Firebase Documentation:**



- Authentication: <https://firebase.google.com/docs/auth>
 - Firestore: <https://firebase.google.com/docs/firestore>
 - Crashlytics: <https://firebase.google.com/docs/crashlytics>
5. **The Movie Database (TMDb) API Documentation.** TMDb. 2025.
<https://developers.themoviedb.org/3>
 6. **Retrofit 2 User Guide.** Square. 2023.
<https://square.github.io/retrofit>
 7. **OkHttp Documentation.** Square. 2023.
<https://square.github.io/okhttp>
 8. **Hilt for Android.** Android Developers. Google. 2024.
<https://developer.android.com/training/dependency-injection/hilt-android>
 9. **Android Credential Manager.** AndroidX. 2024.
<https://developer.android.com/guide/credentials>

11 Anexos

<https://github.com/AngelChv/FocusTrack>