



Universidad Don Bosco

Proyecto de catedra

Fase 1

DSM – G01T

- Alfredo Steven Aguilera Salazar. AS240144
- William Adonay Bonilla Perez. BP220765
- Christian Javier Rosales Palacios RP241102
- Tec. Julio Armando García Sánchez
- Técnico en Ingeniería en Computación
- 25 de agosto de 2025

Índice

Introducción

1. Perfil del proyecto

- 1.1 Nombre del proyecto**
- 1.2 Problema a resolver**
- 1.3 Objetivo general**
- 1.4 Metas del proyecto**
- 1.5 Resultados esperados**
- 1.6 Metodología de trabajo**
- 1.7 Área geográfica y beneficiarios**
- 1.8 Factores de éxito**
- 1.9 Carta de presentación del proyecto**

2. Diseño UX/UI (lineamientos de estilo)

3. Lógica de la solución y diagramas UML

4. Arquitectura de software

5. Herramientas a utilizar

6. Presupuesto estimado

7. Cronograma de trabajo

8. Fuentes de consulta (APA)

3. Introducción

En la actualidad, los sistemas de reserva de boletos en cines presentan limitaciones relacionadas con la experiencia del usuario y la gestión interna de las funciones. El presente proyecto propone el desarrollo de CinePlus, una aplicación móvil nativa en Kotlin para Android, orientada a optimizar la compra de boletos y productos de confitería mediante una experiencia digital segura, ágil y sin filas.

Este documento describe la propuesta de solución, incluyendo la lógica del sistema, la arquitectura de software, el diseño UX/UI, herramientas empleadas, cronograma de ejecución y presupuesto inicial.

4. Perfil del proyecto

4.1 Nombre del proyecto

CinePlus – Sistema de gestión digital para cines.

4.2 Problema a resolver

Los clientes enfrentan problemas como:

- Filas largas en taquilla.
- Falta de claridad sobre cartelera y horarios en tiempo real.
- Riesgo de duplicación en la asignación de asientos.
- Pérdida de tiempo al llegar y encontrar funciones agotadas.

4.3 Objetivo general

Desarrollar una aplicación móvil en **Kotlin con Android** que permita la consulta de cartelera, reserva y compra de boletos y combos, selección de asientos en tiempo real, y emisión de tickets digitales mediante **código QR**.

4.4 Metas del proyecto

- Diseñar e implementar una interfaz amigable y optimizada para Android.
- Desarrollar la lógica de negocio con un backend escalable.
- Asegurar la consistencia de asignación de asientos en tiempo real.
- Implementar un sistema de autenticación seguro para usuarios.
- Ofrecer notificaciones push con recordatorios y promociones.

4.5 Resultados esperados

- Aplicación móvil funcional en Android.
 - Reducción de filas en taquilla en al menos un 60%.
 - Incremento en la venta digital de boletos y combos.
 - Mejora en la experiencia del usuario.
-

4.6 Metodología de trabajo

Se utilizará **Scrum** como metodología ágil.

- **Sprint** de 2 semanas.
 - Roles: Product Owner (profesor), Scrum Master (coordinador), Developers (equipo).
 - Herramienta de gestión: Trello/Notion.
-

4.7 Área geográfica y beneficiarios

- **Área geográfica:** Cines en El Salvador (fase inicial).
 - **Beneficiarios:** Clientes de cines y empresas de entretenimiento.
-

4.8 Factores de éxito

- Confiabilidad en la asignación de asientos.

- Interfaz intuitiva y fluida.
- Seguridad en transacciones digitales.
- Capacidad de escalabilidad a otros cines del país.

4.9 Carta de presentación del proyecto

CinePlus busca transformar la experiencia del cine en El Salvador a través de un sistema digital eficiente que optimice el tiempo del cliente y mejore la gestión de los cines, integrando tecnología móvil y buenas prácticas de UX/UI.

5. Diseño UX/UI

(solo lineamientos gráficos según lo solicitado)





Sistema de Gestión de Entradas y Reservas de
Cine

Iniciar Sesión

Correo electrónico

Contraseña

Iniciar Sesión

¿Olvidaste tu contraseña?

¿No tienes una cuenta?

Registrarse



© 2023 CineGest - Todos los derechos reservados

Películas en Cartelera

Descubre las mejores películas disponibles hoy

Todos

Acción

Comedia

Drama



El Último Horizonte

★ 8.7

Ciencia Ficción • 142 min

Detalles

Reservar

Amor en París

★ 7.9

Romance • 118 min

Detalles

Reservar

El Último Suspiro

ESTRENO ★ 8.7/10

Drama Suspense 142 min +12

Sinopsis

Un detective atormentado por su pasado debe enfrentarse a sus propios demonios mientras investiga una serie de misteriosas desapariciones en una pequeña ciudad costera. A medida que profundiza en el caso, descubre una conspiración que podría cambiar su vida para siempre.

Reparto



Carlos
Mendoza



Lucía Vega



Martín
Rojas



Elena
Durán

Horarios disponibles

CineGest Plaza Central

2.3 km

14:30

17:15

19:45

22:00

CineGest Norte

4.8 km

15:00

18:30

21:15

Seleccionar asientos

Volver a cartelera



Inicio



Entradas



Reservas




Perfil

Selección de Asientos

Película: Avatar: El Camino del Agua
Sala: Sala 3
Hora: 19:30 - 22:15

PANTALLA

A	1	2	3	4	5	6	A
B	1	2	3	4	5	6	B
C	1	2	3	4	5	6	C
D	1	2	3	4	5	6	D
E	1	2	3	4	5	6	E
F	1	2	3	4	5	6	F
G	1	2	3	4	5	6	G

 Disponible  Ocupado  Seleccionado

Asientos seleccionados: F2,F3, F4
Total: \$24.00

Confirmar selección

Cambiar película



Inicio



Entradas



Reservas



Perfil

Resumen de Reserva

Dune: Parte 2

Ciencia ficción | 166 min

★ 8.5/10

Director: Denis Villeneuve

Fecha: Viernes, 12 de Julio 2024
Hora: 19:30
Sala: Sala 3 (IMAX)

Asientos seleccionados

F5 F6 F7

PANTALLA

F2 F3 F4

 Cambiar asientos

Resumen de pago

3 Entradas (Adulto)	€27.00
Suplemento IMAX	€6.00
Cargo por servicio	€1.50
Total	€34.50

Confirmar y pagar

Volver a cartelera



Inicio



Entradas



Reservas



Perfil

- **Paleta de colores:**
 - Primario: #56413E (marrón oscuro)
 - Secundario: #FFA590 (naranja claro)
 - Neutros: #E4E2AF, #FFF1D7, #E5CBB4
 - **Tipografía:**
 - Encabezados: Montserrat Bold, 20–24 pt
 - Subtítulos: Roboto Medium, 16–18 pt
 - Texto principal: Roboto Regular, 14–16 pt
 - **Iconografía:**
 - Íconos consistentes de Material Design (boletos, combos, asientos, notificaciones).
 - **Imágenes:**
 - Portadas de películas en formato JPG/PNG, resolución mínima 1080x1920 px.
 - **Navegación móvil:**
 - **Bottom Navigation Bar** con secciones: Inicio, Cartelera, Mis boletos, Perfil.
-

6. Lógica de la solución y diagramas UML

Flujo principal:

1. Usuario inicia sesión.
2. Consulta cartelera en tiempo real.
3. Selecciona película, sala, horario y asientos.
4. Sistema valida disponibilidad de asientos.
5. Realiza pago digital (tarjeta o billetera electrónica).

6. Se genera ticket QR único.
7. El sistema registra compra en historial del usuario.

Diagramas UML a incluir:

- **Diagrama de casos de uso** (usuarios, administrador, sistema).
 - **Diagrama de clases** (Usuario, Película, Función, Asiento, Ticket, Compra).
 - **Diagrama de secuencia** (reserva de asiento y emisión de ticket).
-

7. Arquitectura de software

Modelo Cliente-Servidor

- **Frontend (móvil):** Kotlin – Android Studio.
 - **Backend (API REST):** Node.js con Express o Laravel (según recursos del equipo).
 - **Base de datos:** PostgreSQL o MySQL.
 - **Comunicación:** API REST con JSON sobre HTTPS.
 - **Seguridad:**
 - Autenticación con JWT.
 - Hash de contraseñas con bcrypt.
 - Validación contra inyección SQL, XSS y CSRF.
-

8. Herramientas a utilizar

- **Lenguaje móvil:** Kotlin.
- **IDE:** Android Studio.
- **Backend:** Node.js + Express / Laravel.

- **DBMS:** PostgreSQL o MySQL.
- **Versionamiento:** Git + GitHub.
- **Gestión:** Trello/Notion.
- **UX/UI:** Android.

9. Presupuesto estimado

Concepto	Costo aproximado (USD)
Desarrollo (3 integrantes x 3 meses, \$200 c/u)	\$1,800
Servidor backend (VPS básico 3 meses)	\$90
Dominio + SSL	\$30
Licencias y APIs externas (ej. pasarela de pago)	\$50
Total estimado	\$1,970

10. Cronograma de trabajo

0. Cronograma de Trabajo (12 semanas - Sprints de 2 semanas)

Sprint 0 (Semana 1): Definición final del proyecto, setup de entornos de desarrollo, creación de repositorios y tablero Trello.

Sprint 1 (Semanas 2-3): Diseño completo de la DB y la API. Mockups finales de todas las pantallas. Desarrollo del Login/Registro en la app y endpoint de autenticación en el backend.

Sprint 2 (Semanas 4-5): Implementación del catálogo de películas (API y consumo en app). Pantalla de Detalle de Película.

Sprint 3 (Semanas 6-7): Núcleo del proyecto. Implementación del sistema de selección de horarios y el sistema de bloqueo de asientos en tiempo real (Backend + Frontend).

Sprint 4 (Semanas 8-9): Flujo de checkout y pasarela de pago simulada. Generación del ticket QR.

Sprint 5 (Semanas 10-11): Desarrollo del Historial de Compras. Integración con FCM para notificaciones. Testing intensivo y depuración.

Sprint 6 (Semana 12): Preparación para entrega final. Documentación (README, manual de usuario), video de demostración, y publicación de APK.

12. Qué es una API?

Una API REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedia distribuidos, como la World Wide Web.

Según la documentación de Android Developers (Google) y el estándar arquitectónico:

- **Principio Base:** Una API RESTful se basa en el protocolo HTTP y utiliza sus métodos estándar para interactuar con recursos (datos u objetos) identificados por una URI (Uniform Resource Identifier).
- **Métodos HTTP y Operaciones CRUD:** Los métodos de solicitud HTTP se mapean a las operaciones fundamentales para la gestión de recursos:
 - **GET:** Se utiliza para leer o recuperar datos del servidor.
 - **POST:** Se utiliza para crear nuevos datos en el servidor.
 - **PUT/PATCH:** Se utiliza para actualizar datos existentes.
 - **DELETE:** Se utiliza para eliminar datos del servidor.
- **Formato de Datos:** La respuesta del servidor es típicamente formateada en uno de los formatos comunes de datos, siendo JSON (JavaScript Object Notation) el más utilizado por su ligereza y facilidad de análisis.
- **Stateless (Sin Estado):** El concepto central de REST es que cada solicitud del cliente al servidor debe contener toda la información necesaria para que el servidor la entienda, sin que el servidor dependa de la información de sesiones anteriores.

13. Consumo de API REST en Android

El consumo de APIs REST es esencial en el desarrollo moderno de Android para obtener datos de un servicio backend. La documentación oficial de Google recomienda el uso de librerías externas para simplificar el proceso, asegurar que las operaciones de red se realicen en un hilo de fondo y seguir las mejores prácticas de la plataforma.

1. Librerías de Red (Recomendación Oficial)

La forma más recomendada para conectar una aplicación Android a un servicio web RESTful y obtener una respuesta es mediante la librería Retrofit de Square, Inc.

- Retrofit: Es una librería *type-safe* de HTTP para Java y Kotlin. Permite definir la estructura de la API como una interfaz, lo que simplifica enormemente las llamadas a la API.

Uso: Se combina con un *convertidor*, como Gson o `kotlinx.serialization`, para analizar automáticamente la respuesta JSON del servidor y convertirla en objetos de datos de Kotlin/Java.

- Documentación de Google: Los *code labs* oficiales de Android para "Obtener datos de internet" enseñan a implementar la capa de red con Retrofit, destacando su facilidad para "simplificar en gran medida la obtención de datos" y "ayudar a la aplicación a seguir las mejores prácticas de Android, como realizar operaciones en un hilo de fondo."

2. Concurrencia e Hilos

Las operaciones de red deben realizarse siempre fuera del hilo principal (UI thread) para evitar bloqueos y el error *Application Not Responding* (ANR).

- Corrutinas de Kotlin: La práctica moderna en Android, fuertemente promocionada en la documentación oficial, es usar Corrutinas de Kotlin (Kotlin Coroutines) en conjunto con Retrofit.
 - Ventaja: Permite escribir código asíncrono y concurrente de manera secuencial y legible (suspend functions), lo que simplifica la gestión de las peticiones de red y sus respuestas.

14. Conclusión

El concepto fundamental de una API REST se establece como un estilo de arquitectura de *software* para sistemas distribuidos, como la World Wide Web, que opera sobre el protocolo HTTP. Estas APIs utilizan los métodos HTTP estándar (como GET para recuperar datos o POST para crear nuevos) para interactuar con recursos identificados por una URI. Un principio clave de esta arquitectura es que es Sin Estado, exigiendo que cada solicitud del cliente contenga toda la información necesaria para ser procesada, sin dependencia de sesiones previas. La comunicación suele darse en formato JSON debido a su ligereza y facilidad de análisis.

Para el desarrollo moderno en Android, la documentación oficial de Google recomienda encarecidamente consumir estas APIs utilizando librerías externas como

Retrofit. Retrofit simplifica enormemente las llamadas HTTP, permitiendo definir la estructura de la API como una interfaz y facilitando la conversión automática del JSON de la respuesta a objetos de datos de Kotlin/Java mediante un convertidor como Gson. Es indispensable que estas operaciones de red se realicen fuera del hilo principal de la interfaz de usuario para prevenir errores como el ANR. Por ello, la práctica actual, fuertemente promovida por Google, integra Corrutinas de Kotlin con Retrofit, lo que permite gestionar el código asíncrono de manera concisa y legible a través de funciones suspend. Este método es esencial para obtener datos de un servicio backend y seguir las mejores prácticas en la plataforma.

15. Fuentes de consulta (APA)

- Sommerville, I. (2016). *Ingeniería del Software* (9ª ed.). Pearson.
- Pressman, R. S. (2014). *Ingeniería del Software: Un enfoque práctico*. McGraw-Hill.
- Google. (2024). *Android Developers Documentation*.
<https://developer.android.com>
- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice*. Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Android Developers. (2023)
- PostgreSQL Global Development Group. (2023)