

Resumen

Foodies es una aplicación de restaurante la cual permite buscar restaurantes cerca de ti guardarlos en favoritas mirarlos en el mapa y filtrarlos por tipo de Restaurantes. Te permite encontrar establecimiento cerca que sean de tu gusta además de contar con reseñas del restaurante y aportar tu ubicación o la que desees buscar para encontrar el mejor restaurante para ti.

Para el desarrollo de Foodies se ha usado la forma AGILE, lo que ha dejado trabajar con cambios a través del tiempo. Gracias a este método, se pudo mover hacia adelante en fases pequeñas, llamados esprints, donde ponían cosas como el mapa, un listado de favoritos o filtros por tipo de comida. Este modo hizo fácil encontrar errores temprano, también ayuda a meter mejoras cuando salen nuevas ideas o necesidades. La manera AGILE fue clave para adaptar la aplicación a metas y dar mejor experiencia final que se enfocaba en usuario.

Los resultados con el desarrollo de la aplicación han sido muy buenos, ya que se han logrado los objetivos que fijaron desde el comienzo. Foodies resalta no solo por ser muy útil sino también por tener un diseño fácil de usar y una experiencia simple. Su sencillez la hace una herramienta fácil para todo tipo de persona, lo cual fortalece su importancia como solución práctica y efectiva al buscar lugares para comer.

INTORDUCCION	4
Presentación del Tema y Relevancia.....	4
Objetivos	4
Justificación.....	5
MARCO TEÓRICO.....	6
Revisión de la Literatura relacionada con el Tema del Proyecto	6
Aplicaciones Móviles sector Gastronómico	6
Tendencias de consumo y el impacto de las reseñas en línea.	7
METODOLOGÍA	7
Descripción de Tecnologías y Metodologías relevantes	7
Tecnologías para el Back End	7
Tecnologías para el Front End.....	9
Resumen en tabla de Tecnologías	10
Descripción detallada del enfoque metodológico en Foodies	11
DESARROLLO E IMPLEMENTACIÓN.....	12
Descripción Detallada del proceso de desarrollo.	12
Solución a problemas frecuentes	14
Conclusion del Desarrollo	14
Visualización de Procesos y Capturas de Pantalla	15
RESULTADOS.....	17
Presentación de la Aplicación y Funcionalidades	17
Características principales:	17
Problema que aborda	19
Resultado de las pruebas y evaluaciones de la aplicación	21
Comparaciones con otros productos o soluciones similares	24
DISCUCIÓN.....	25
Análisis de los Resultaod obtenidos.....	25
Reflexiones sobre las limitaciones del proyecto y posibles áreas de mejora	27
Implicaciones prácticas y teóricas de los resultados	29

CONCLUSION	31
Resumen total del proyecto	31
Lista de referencias	32

INTORDUCCION

Presentación del Tema y Relevancia

La presente aplicación, nombrada Foodies, llega con un propósito: hacer más fácil la búsqueda de restaurantes de un manera fácil y personalizada. Esta aplicacion está hecha para ayudar a los usuarios en hallar sitios para comer cerca de su lugar o en cualquier área que quieran buscar. Foodies deja ver los restaurantes en un mapa, ponerlos en una lista de favoritos, organizarlos por tipo de comida y mirar comentarios para tomar mejores decisiones. Este plan es muy importante, porque resuelve un problema real y simple: saber dónde comer según lo que nos gusta, lo cerca que está y la calidad. Hoy en día, muchos valoran no solo la comida, sino también la experiencia de encontrar lugares recomendados, cerca y que encajen en sus preferencias.

Objetivos

El objetivo general al desarrollar la aplicación Foodies fue proporcionar una herramienta digital que facilite y optimice las búsquedas de restaurantes de una manera eficiente, atractiva e intuitiva. La aplicación está diseñada para satisfacer las necesidades prácticas de los usuarios actuales, que buscan una solución rápida, conveniente y visualmente agradable a la hora de decidir dónde comer. Para lograrlo, Foodies ha integrado diversas funcionalidades que permiten a los usuarios explorar restaurantes cercanos, aplicar filtros según sus preferencias, ver ubicaciones en un mapa interactivo, guardar sus lugares favoritos y gestionar reservas a futuro directamente desde la aplicación. Este objetivo se alinea con la

creciente demanda de herramientas móviles que mejoren la experiencia del consumidor en el sector Food Service, facilitando el acceso a la información y la toma de decisiones basadas en gustos personales, distancia y otras opiniones de los clientes.

Uno de los objetivos específicos es permitir a los usuarios encontrar restaurantes cerca de su ubicación actual o en un área específica utilizando servicios de geolocalización y brindar la posibilidad de ver los resultados en un mapa interactivo. Además, la función de búsqueda se puede personalizar con filtros por tipo de alimento para que cada usuario pueda encontrar fácilmente lo que busca.

Otro objetivo importante fue brindar la opción de guardar restaurantes como favoritos para que los usuarios puedan acceder rápidamente a los restaurantes que les interesan o ya han visitado. La aplicación también puede redirigir a los usuarios a Google Maps, lo que facilita la navegación al restaurante elegido con indicaciones en tiempo real. Como parte de las mejoras futuras, planeamos implementar un sistema de reserva en la aplicación que permitirá a los usuarios confirmar su mesa sin salir de Foodies. Por último, se incluye la opción de consultar opiniones de otros usuarios, ayudándote a tomar mejores decisiones basadas en la experiencia real.

Justificación

Foodies nació de la necesidad de una herramienta digital que pudiera encontrar restaurantes de manera rápida y eficiente. Ante la creciente demanda de soluciones móviles para encontrar lugares donde comer, la aplicación ofrece

una plataforma intuitiva que permite a los usuarios buscar, filtrar por tipo de comida, guardar favoritos y ver reseñas, todo a través de un diseño fácil de usar. Además, la integración con Google Maps y la futura incorporación de un sistema de reservas hacen de Foodies una solución integral para los usuarios. Esta idea me parece especialmente significativa porque responde a una necesidad real de los usuarios modernos: encontrar el restaurante adecuado de forma rápida y sencilla. Además, el proyecto me brindó una oportunidad de aprendizaje única para aplicar diversas técnicas y metodologías (como AGILE) en un entorno real, adquiriendo habilidades valiosas tanto en tecnología como en gestión de proyectos.

MARCO TEÓRICO

Revisión de la Literatura relacionada con el Tema del Proyecto

Aplicaciones Móviles sector Gastronómico

Introducción a las aplicaciones móviles en la industria alimentaria El auge de las aplicaciones móviles ha transformado todas las industrias, y la industria alimentaria no es una excepción. Hoy en día, los consumidores utilizan aplicaciones para buscar restaurantes, leer reseñas, comparar precios, hacer reservas y más. Según una investigación reciente, más del 60% de los usuarios de teléfonos inteligentes utilizan aplicaciones móviles para buscar opciones de restaurantes cercanas (Market Research X, 2023). Este comportamiento refleja el crecimiento y la importancia de las plataformas digitales en las decisiones relacionadas con la alimentación y el ocio.

Tendencias de consumo y el impacto de las reseñas en línea.

Hoy en día, las reseñas y valoraciones de otros usuarios juegan un papel importante a la hora de que los consumidores elijan un restaurante. Según un informe de BrightLocal (2021), el 79% de los consumidores leen reseñas online antes de decidir qué restaurante elegir. Esto ha llevado a aplicaciones como Foodies a incluir la posibilidad de ver y enviar reseñas dentro de la plataforma. Este feedback no sólo mejora la experiencia del usuario, sino que también beneficia al restaurante al proporcionar visibilidad y oportunidades de mejora.

El desarrollo de aplicaciones móviles ha cobrado protagonismo en los últimos años, especialmente en el ámbito culinario, donde los usuarios necesitan herramientas rápidas, intuitivas y fiables para encontrar lugares donde comer. Aplicaciones como Google Maps, TripAdvisor, TheFork y Yelp están liderando el camino a la hora de ofrecer información de geolocalización, reseñas, reservas online y filtrado por tipo de restaurante. Sin embargo, muchas de estas plataformas tienen interfaces complejas, funcionalidad fragmentada o experiencias personalizadas deficientes. Esto presentó una clara oportunidad para crear una alternativa más sencilla, centrada en la experiencia del usuario y con una estructura más flexible: nació Foodies.

METODOLOGÍA

Descripción de Tecnologías y Metodologías relevantes

Tecnologías para el Back End

Java 21: utilice la versión 21 del lenguaje de programación Java para aprovechar las mejoras en el rendimiento, la seguridad y las nuevas características del lenguaje, como la coincidencia de patrones, las mejoras en las expresiones lambda y la eficiencia del tiempo de ejecución. Java sigue siendo una de las opciones más estables y escalables para el desarrollo backend.

Spring Boot: un potente marco que ayuda a desarrollar aplicaciones Java, especialmente al crear API RESTful. Permite una configuración automática, una gestión eficiente de dependencias y una estructura modular. En Foodies, Spring Boot es responsable de administrar la lógica empresarial, los controladores, los servicios, los repositorios y la exposición de los puntos finales utilizados por el frontend. PostgreSQL: esta base de datos relacional se utiliza debido a su potencia, confiabilidad y compatibilidad con Spring Data JPA.

PostgreSQL permite consultas complejas, gestión de relaciones y alto rendimiento incluso al manejar grandes cantidades de datos, lo que lo hace ideal para almacenar información sobre restaurantes, tipos de comida, usuarios, colecciones y reseñas.

Spring Security + JWT: para la gestión de autenticación y autorización, se implementa JWT (JSON Web Tokens) a través de Spring Security. Esta combinación protege eficazmente los puntos finales, garantizando que solo los usuarios autenticados puedan acceder a funciones como favoritos o comentarios y garantiza la integridad de la sesión. API externa (geolocalización y reseñas): Realiza llamadas HTTP a APIs externas (por ejemplo, Google Places) para obtener información detallada sobre los restaurantes (dirección, fotos,

valoraciones, horarios de apertura, etc.) y permitir búsquedas geolocalizadas desde la aplicación.

Tecnologías para el Front End

Kotlin: el lenguaje oficial para el desarrollo de Android, moderno, conciso y seguro. Se utiliza en todo el frontend de Foodies, lo que permite una integración eficiente de llamadas API, gestión de estados y lógica de presentación.

Jetpack Compose: un marco de interfaz de usuario declarativo de Android que reemplaza el XML tradicional. Compose permite la creación de interfaces limpias, dinámicas y adaptables al tiempo que facilita la reutilización de componentes visuales como tarjetas de restaurantes, filtros, menús y pantallas de detalles.

API de Google Maps: Para mostrar restaurantes en un mapa, se integra Google Maps, lo que permite a los usuarios ver visualmente establecimientos cercanos, hacer zoom, ver marcadores personalizados y acceder a la navegación directamente desde su ubicación actual o una ubicación buscada.

ViewModel y LiveData: aplique el patrón MVVM para separar la lógica de presentación y la lógica de negocio. Los ViewModels administran pasivamente el estado de la interfaz de usuario mediante LiveData y StateFlows, lo que permite una experiencia fluida y desacoplada. Composición de navegación: La composición de navegación se utiliza para la navegación entre pantallas (mapas, favoritos, detalles, filtros), lo que facilita una gestión de rutas más moderna y adaptada al ciclo de vida de la actividad.

Coil: una biblioteca liviana y eficiente para cargar imágenes desde URL, utilizada para mostrar fotos de restaurantes en tarjetas y pantallas de detalles.

Estas líneas son la línea incluida en la parte superior de la tabla, la línea entre el la cabecera de la tabla y el contenido y la línea debajo de la tabla.

Resumen en tabla de Tecnologías

Categoría	Tecnología / Herramienta	Uso en Foodies
<i>Lenguaje Backend</i>	Java 21	Desarrollo del backend con funcionalidades modernas y alto rendimiento.
<i>Framework Backend</i>	Spring Boot	Creación de la API REST, gestión de lógica de negocio, seguridad, pruebas, etc.
<i>Base de Datos</i>	PostgreSQL	Almacenamiento de usuarios, restaurantes, favoritos, filtros, reseñas, etc.
<i>Seguridad</i>	Spring Security + JWT	Protección de endpoints mediante autenticación con tokens.
<i>Pruebas Backend</i>	@DataJpaTest	Pruebas automatizadas sobre la capa de persistencia.
<i>Documentación</i>	Swagger / OpenAPI	Documentación automática de los endpoints de la API REST.
<i>APIs externas</i>	Google Places API	Información de restaurantes: ubicación, reseñas, fotos, horarios, etc.
<i>Lenguaje Frontend</i>	Kotlin	Desarrollo de toda la interfaz de la app Android.
<i>UI Framework</i>	Jetpack Compose	Interfaz moderna, dinámica y declarativa.
<i>Navegación UI</i>	Navigation Compose	Gestión de pantallas (mapa, detalles, favoritos, etc.).
<i>Estado / Lógica UI</i>	ViewModel + LiveData/StateFlow	Gestión reactiva de la interfaz.
<i>Mapa</i>	Google Maps API	Visualización geográfica de restaurantes.
<i>Carga de imágenes</i>	Coil	Carga optimizada de imágenes desde URL.
<i>Arquitectura</i>	MVVM	Separación de capas en el frontend.
<i>Metodología</i>	AGILE / Scrum	Desarrollo iterativo en sprints con mejoras continuas.
<i>Herramientas dev</i>	Postman, Android Studio, Swagger	Pruebas, simulaciones y documentación.

Descripción detallada del enfoque metodológico en Foodies

Para el desarrollo del proyecto Foodies se optó por un enfoque metodológico basado en AGILE, siguiendo concretamente principios inspirados en la metodología Scrum, por su flexibilidad, enfoque iterativo y foco en la entrega continua de valor. La naturaleza dinámica del proyecto, centrado en el desarrollo de una aplicación móvil con múltiples funciones integradas como búsqueda, geolocalización, filtros y favoritos, requería un modelo de trabajo que pudiera mejorar de forma incremental y responder rápidamente a eventos imprevistos o necesidades cambiantes. AGILE demostró ser la opción ideal para este propósito. Durante el proceso de desarrollo, el trabajo se divide en iteraciones o sprints, cada uno con objetivos claros a corto plazo.

Después de cada sprint, obtendrás un módulo funcional de la aplicación, como la visualización de mapas, la gestión de favoritos o la implementación del sistema de filtrado de tipos de comida. Esta estructura permite la validación continua del progreso y los ajustes basados en evidencia emergente, retroalimentación o descubrimientos. Además, utilizar este enfoque ayuda a gestionar el tiempo y los recursos de forma eficiente, promueve la mejora continua, la revisión permanente del código y la incorporación gradual de nuevas ideas, como la futura integración de reservas o la conexión directa a Google Maps para navegar hasta el restaurante deseado. En resumen, el enfoque AGILE permite desarrollar Foodies de forma organizada, adaptable y centrada en el usuario, garantizando que el producto final cumpla los objetivos funcionales y proporcione una experiencia sencilla, práctica y moderna.

DESARROLLO E IMPLEMENTACIÓN

El desarrollo de un trabajo final de grado (TFG) supone un enorme reto tanto a nivel técnico como personal. Desde el principio, el enfoque fue ambicioso: crear una aplicación de búsqueda de restaurantes con múltiples funcionalidades, como vista de mapa, sistema de favoritos, filtros, reseñas y conexiones a API externas. El proceso abarca desde la ideación y planificación inicial hasta el diseño completo y la implementación de un sistema multiplataforma que conecta un front-end Android y un back-end Java.

Descripción Detallada del proceso de desarrollo.

1. Definir estructura del Proyecto

El primer paso es definir claramente la estructura general del proyecto y sus objetivos. A partir de ahí, comencé a construir la base del backend usando Java 21 y Spring Boot, y luego construí el frontend usando Kotlin y Jetpack Compose. Esta fase es particularmente desafiante porque implica establecer una base sólida para el sistema, definir el modelo de datos, planificar la arquitectura y configurar las herramientas necesarias. Uno de los aspectos más complejos de esta fase es la creación de consultas, especialmente aquellas que filtran restaurantes por tipo, región u orden de preferencia. Algunas de estas consultas requirieron conocimientos avanzados de JPA y SQL, lo que me obligó a realizar una investigación exhaustiva y seguir probando hasta obtener los resultados esperados.

2. Desarrollo del Frontend, Desafíos con Jetpack Compose

Implementar el frontend en Jetpack Compose fue otro desafío importante. Si bien Compose ofrece un enfoque moderno y declarativo, tiene una curva de aprendizaje pronunciada. En particular, tuve dificultades con la gestión del estado de la interfaz, la navegación de la pantalla y la carga dinámica de datos en el backend. Resolver estas dificultades me obligó a leer documentación oficial, estudiar ejemplos y revisar buenas prácticas arquitectónicas para MVVM. También existen problemas relacionados con la integración de Google Maps, específicamente la sincronización de las ubicaciones de los usuarios con los marcadores en el mapa. Después de muchos intentos y ajustes al ViewModel, logré que la visualización del mapa fuera funcional y dinámica.

3. Desarrollo del Backend

Otro desafío importante fue lograr una comunicación efectiva entre el backend y la aplicación Android. Esto implica gestionar correctamente las llamadas HTTP, gestionar los errores en las respuestas, validar los datos recibidos y mostrarlos adecuadamente en la interfaz. Inicialmente, las respuestas devueltas por el backend no coincidían con lo que esperaba el frontend, lo que generaba errores difíciles de depurar. La solución implicó definir correctamente los DTO, validar las rutas en Postman y reestructurar la estructura de respuesta en el backend.

4. Exploración de Infraestructura - Servidor Propio

Durante el proyecto, también vi cómo crear mi propio servidor usando Debian para alojar el backend y la base de datos. Esto incluye la instalación de servicios como PostgreSQL, la configuración de firewalls y puertos, y la preparación del entorno con Java y Spring Boot. Si bien no pude completar la implementación de este servidor debido a limitaciones de tiempo, fue una valiosa experiencia de aprendizaje y parte de un proyecto que planeo completar en el futuro para lograr implementaciones totalmente autónomas y controladas.

Solución a problemas frecuentes

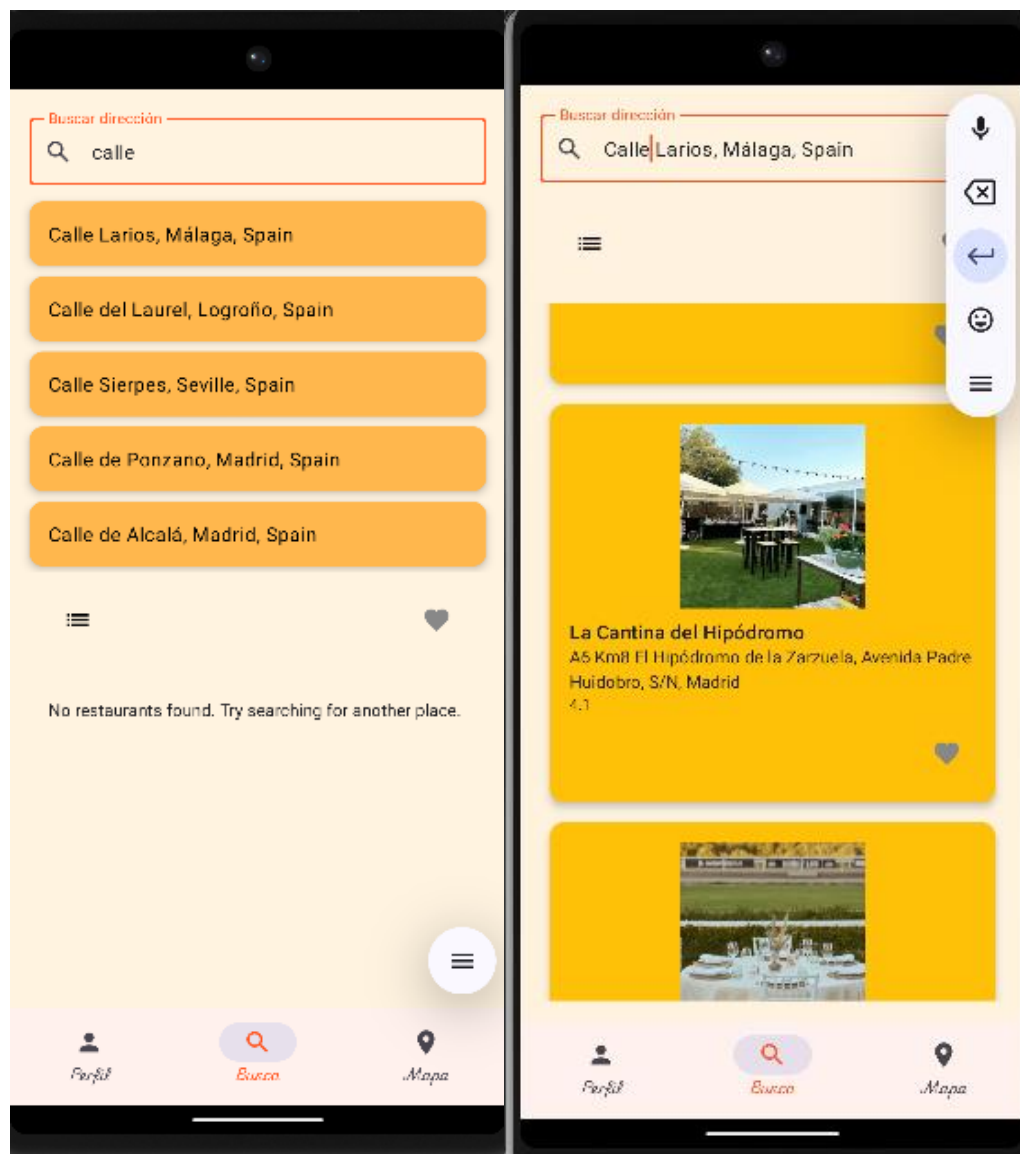
Hubo muchos errores durante el proceso de desarrollo, pero cada error fue una oportunidad de aprendizaje: Problemas de consulta: resolví estos problemas reestructurando mis repositorios y usando un @Query personalizado. Errores de escritura: investigué patrones como la memorización, mutableStateOf y el uso adecuado de ViewModel. Inconsistencias entre el backend y el frontend: se resolvieron validando y auditando modelos compartidos usando Postman. Manejo de errores: agregué estructuras de control en la aplicación para notificar al usuario cuando no hay conexión o no se pueden recuperar los datos.

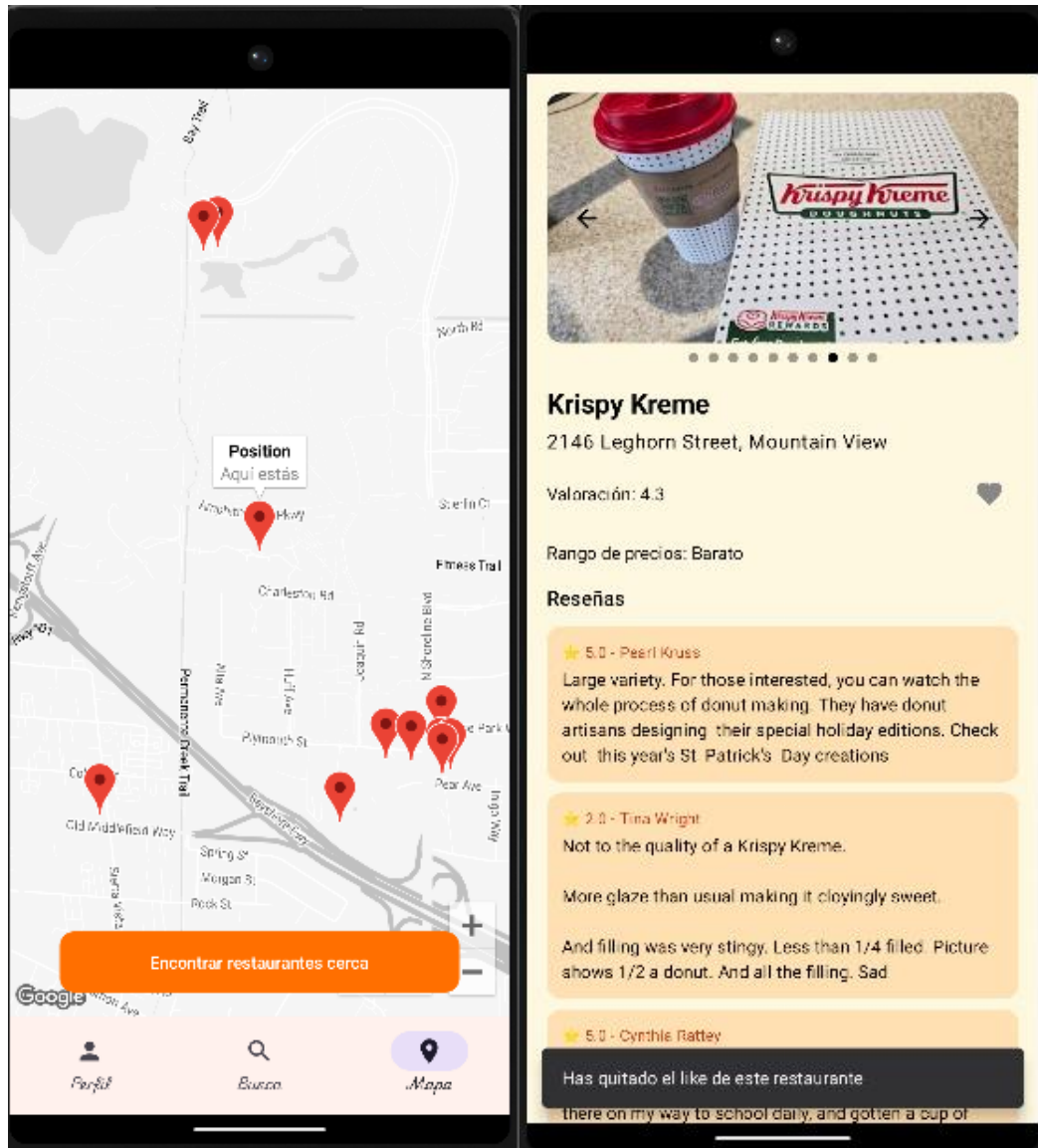
Conclusion del Desarrollo

A pesar de los desafíos, el proyecto ha progresado constantemente y ahora es una aplicación funcional con una base sólida, lista para crecer. El uso de metodologías ágiles, el trabajo continuo y la voluntad de superar errores técnicos han sido clave para llevar a Foodies a donde está hoy. Este proyecto no solo fortaleció mi conocimiento de Java, Spring Boot, Kotlin y Android, sino que

también me expuso a situaciones de desarrollo profesional del mundo real, incluido el backend, el frontend y la gestión de infraestructura.

Visualización de Procesos y Capturas de Pantalla





La aplicación consta de diferentes pantallas que se integran entre sí para brindar una experiencia de usuario perfecta. Estos incluyen una pantalla de mapa que le permite ver su ubicación actual y restaurantes cercanos. También cuenta con una pantalla de autocompletar que facilita la búsqueda de restaurantes con sugerencias inteligentes. Además, la tarjeta de información incluye una vista

previa de cada restaurante, así como una pantalla detallada que muestra reseñas, imágenes, horarios y otra información relevante del restaurante seleccionado.

RESULTADOS

Presentación de la Aplicación y Funcionalidades

La aplicación Foodies es una plataforma diseñada para brindar a los usuarios una experiencia completa de descubrimiento, exploración y disfrute de la comida cercana. Desarrollada exclusivamente para la plataforma Android, la aplicación permite a los usuarios encontrar restaurantes, ver menús y realizar pedidos fácilmente de forma intuitiva y rápida.

Características principales:

Explora los restaurantes cercanos: La característica más notable de FoodExplorer es su capacidad de mostrar restaurantes cercanos según la ubicación actual del usuario. La aplicación utiliza la API de Google Maps para obtener la ubicación del usuario en tiempo real y muestra un mapa interactivo con marcadores que indican la ubicación de los restaurantes cercanos.

Detalles del restaurante: Al seleccionar un restaurante del mapa o de la lista, los usuarios pueden acceder a detalles completos sobre ese restaurante, incluida la siguiente información: Nombre, dirección y tipo de comida. Horario comercial. Opiniones y valoraciones de otros usuarios.

Para moverse entre las imágenes del restaurante: Para enriquecer la experiencia de visualización, la aplicación permite a los usuarios desplazarse por las imágenes del restaurante, ya sea viendo fotos de la comida, la ubicación o eventos especiales. Esto se puede hacer deslizando el dedo o usando los botones

laterales, lo que proporciona una forma interactiva y dinámica de explorar el entorno del restaurante.

Geolocalización y navegación: Foodies no solo muestra la ubicación de los restaurantes en un mapa, sino que también integra capacidades de navegación, lo que permite a los usuarios llegar fácilmente a los restaurantes deseados utilizando Google Maps y ofrece opciones de ruta en coche, caminando o en transporte público.

Sistema de calificación y comentarios: Los usuarios pueden ver experiencias en los restaurantes que desean visitar y compartir sus opiniones sobre la calidad del servicio y la comida. Esto permite que otros usuarios tomen decisiones informadas y ayuden a los restaurantes a mejorar su servicio.

Entrega a domicilio y Reservas EN DESARROLLO: FoodExplorer también facilita realizar pedidos de entrega o reservar una mesa directamente a través de la aplicación. Los usuarios pueden elegir los platos que desean pedir, personalizarlos si es necesario y pagar de forma segura a través de la plataforma.

Foodies implementó una estrategia efectiva para reducir la cantidad de llamadas API externas almacenando en caché los datos del restaurante en la base de datos.

Cada vez que la aplicación solicita información sobre restaurantes cercanos, el backend primero consulta la base de datos para ver si ya hay datos almacenados dentro de un radio definido de la ubicación actual del usuario. Esta consulta utiliza un cálculo de distancia geográfica basado en la fórmula Haversine para filtrar restaurantes.

```
SELECT * FROM restaurantes
WHERE (
    6371 * acos(
        cos(radians(:lat_usuario)) *
        cos(radians(latitud)) *
        cos(radians(longitud) - radians(:lon_usuario)) +
        sin(radians(:lat_usuario)) *
        sin(radians(latitud))
    )
) < :radio_km;
```

Si la base de datos devuelve resultados, esos restaurantes se muestran directamente al usuario, evitando llamadas a API externas. Si no se encuentra ningún restaurante dentro del radio, se realiza una llamada a una API externa, se recuperan nuevos datos y esos datos se almacenan automáticamente en la base de datos para futuras referencias. Este enfoque híbrido optimiza el rendimiento, reduce la latencia y minimiza el costo de usar API de terceros, al tiempo que garantiza que los datos estén rápidamente disponibles para futuras búsquedas de los usuarios.

Problema que aborda

La aplicación Foodies fue creada para satisfacer una necesidad real y creciente en el mercado actual: búsqueda en tiempo real, eficiente, rápida y personalizada de restaurantes y opciones gastronómicas. En un entorno donde el

acceso instantáneo a la información y la experiencia del usuario son factores clave, los consumidores quieren encontrar opciones gastronómicas que se adapten a sus preferencias, ubicación y disponibilidad sin tener que consultar múltiples fuentes o pasar por un proceso complicado. Actualmente, muchas personas dependen de aplicaciones fragmentadas o poco intuitivas para encontrar un restaurante, ver un menú o realizar un pedido. Además, la plataforma está sobresaturada y requiere múltiples pasos para acceder a información básica, lo que impacta negativamente en la experiencia del usuario.

En este sentido, Foodies ofrece una solución integral, intuitiva y eficiente que combina geolocalización, exploración visual (a través de imágenes deslizables del restaurante) e interacciones directas (como realizar pedidos o reservas) en una única aplicación móvil. Desde una perspectiva de mercado, Foodies encaja con las tendencias actuales tanto en el sector de la restauración como en el de la tecnología móvil, donde los usuarios valoran cada vez más la inmediatez, la personalización y la comodidad. La implementación de una base de datos optimizada reduce el número de llamadas a servicios externos y también satisface necesidades técnicas reales: menor latencia, menores costos y mayor sostenibilidad del servicio. Asimismo, para los restaurantes, Foodies representa una oportunidad para aumentar la visibilidad y conectar directamente con los clientes potenciales de su zona, permitiéndoles destacar sus platos, gestionar su presencia digital y obtener reseñas sin tener que depender únicamente de grandes plataformas con altos costos de intermediación.

Resultado de las pruebas y evaluaciones de la aplicación

Resultados de pruebas y evaluaciones

El proceso de validación de la aplicación Foodies se basó principalmente en pruebas manuales en un entorno de desarrollo local. Si bien no se implementaron ni realizaron pruebas automatizadas en el entorno de producción, se realizaron varias pruebas funcionales, técnicas y de experiencia del usuario con el objetivo de garantizar el comportamiento correcto de las funciones críticas del sistema. Pruebas técnicas

Desde una perspectiva técnica, se probó la funcionalidad de todos los puntos finales expuestos por el backend desarrollado en Spring Boot. Estas pruebas incluyen la verificación de: Estructura de respuesta:

- ❖ Verifique los datos devueltos para asegurarse de que sean correctos, tengan el formato adecuado (JSON) y sigan la lógica esperada.
- ❖ Tiempo de respuesta de la solicitud: Si bien la prueba no fue cronometrada con herramientas profesionales, el tiempo de carga fue monitoreado manualmente desde la aplicación de Android, especialmente para acciones como buscar restaurantes o cargar favoritos.
- ❖ Se revisó la integración con API externas, como la API de Google Maps, para garantizar una representación precisa de las ubicaciones de los usuarios y los restaurantes.
- ❖ La conectividad adecuada entre el frontend y el backend garantiza que la comunicación a través de solicitudes HTTP esté libre de errores y que

los datos enviados desde el cliente lleguen al servidor y se procesen correctamente.

- ❖ También probamos funciones sensibles como la autenticación, evaluando el sistema de inicio de sesión y la generación de tokens JWT para garantizar: Sólo se permite el acceso a usuarios con credenciales válidas. Cada solicitud a un recurso protegido requiere y valida un token.
- ❖ El proceso de recuperación de contraseña ahora funciona correctamente, permitiendo a los usuarios recuperar el acceso a sus cuentas.

Pruebas funcionales del cliente (Android) Del lado del cliente, se revisó el comportamiento de las interfaces desarrolladas en Kotlin utilizando Jetpack Compose. El objetivo de las pruebas es garantizar que la experiencia del usuario sea fluida, visualmente comprensible y funcional. Algunas de las pruebas incluyen:

- ❖ Interactúa con el mapa y muestra el restaurante en la ubicación correcta. Vea y administre sus restaurantes favoritos.
- ❖ Las pruebas del filtro de tipo de restaurante mostraron que, si bien el filtro era funcional, había margen de mejora en términos de precisión e incorporación de múltiples criterios.
- ❖ Validar la navegación entre pantallas y la consistencia de la interfaz entre diferentes flujos.

Vale la pena señalar que, si bien la aplicación cumple con sus objetivos funcionales, todavía existe la oportunidad de mejorar la usabilidad y la estética del frontend, lo que será clave para futuras versiones.

Observaciones y áreas de mejora

A partir de las pruebas realizadas, se han identificado varias áreas con potencial de mejora:

- ❖ Refinamientos de filtrado: Algoritmos mejorados para filtrar restaurantes por tipo, ubicación o proximidad, incluyendo la opción de combinar múltiples criterios y ordenar los resultados.
- ❖ Sistema de reservas (en desarrollo): aunque todavía no está implementado, hay planes para permitir a los usuarios hacer reservas directamente desde la aplicación. Esto requerirá desarrollar una interfaz para que los restaurantes gestionen la disponibilidad, el número de mesas y los horarios.
- ❖ Perfiles de restaurantes con roles claros: Hay planes para construir un sistema de roles para diferenciar entre usuarios habituales y restaurantes. Esto permitirá a los restaurantes gestionar su información, reservas y perfiles a través de la aplicación.
- ❖ Mejoras visuales en la interfaz: si bien la aplicación es totalmente funcional, la interfaz podría beneficiarse de un diseño más moderno y atractivo que se adapte a diferentes tamaños de pantalla, incluidas transiciones más suaves, imágenes más nítidas y un mejor contraste visual.

Escalabilidad e implementación: Hemos preparado servidores Debian dedicados para la implementación del backend, lo que permitirá que la aplicación escale a entornos más especializados. Sin embargo, debido a limitaciones de

tiempo, el proceso aún no se ha completado, aunque todavía se está desarrollando como parte del desarrollo futuro del proyecto.

Comparaciones con otros productos o soluciones similares

Al analizar el mercado actual de aplicaciones de recomendación y búsqueda de restaurantes, es inevitable mencionar soluciones establecidas como Google Maps, TripAdvisor, TheFork y Yelp. Estas plataformas ofrecen bases de datos completas de agencias, reseñas de usuarios, filtros de búsqueda y, en algunos casos, capacidades de reserva.

Sin embargo, los amantes de la comida se diferencian en algunos aspectos clave:

Ventajas de Foodies frente a otras soluciones

Interfaz simplificada, adaptada a lo esencial: a diferencia de las plataformas que pueden abrumar a algunos usuarios debido a la densidad de funciones o anuncios intrusivos, Foodies se centra solo en lo esencial: encontrar, ver en un mapa, filtrar y guardar restaurantes favoritos. Integración clara con Google Maps: Mientras que otras aplicaciones también utilizan mapas, Foodies los utiliza de forma más directa, permitiendo una navegación rápida desde la lista al mapa sin pasos intermedios innecesarios. Diseño hermoso y moderno: creado desde cero con Jetpack Compose, la experiencia visual e interactiva está optimizada para dispositivos Android modernos, lo que garantiza transiciones suaves, una interfaz limpia y una mayor capacidad de respuesta. Considere mejoras futuras específicas: a diferencia de las soluciones únicas, Foodies adopta un enfoque adaptable al diseño. El plan es añadir funciones como reservas de mesa basadas en la disponibilidad real del restaurante, gestión de perfiles comerciales y un

sistema de recomendaciones inteligente basado en las preferencias del usuario, lo que permitirá llenar vacíos que otras aplicaciones aún no logran abordar de manera efectiva.

Deficiencias actuales o áreas de mejora

Base de datos limitada: Foodies aún está en fase de desarrollo y no tiene una base de restaurantes tan amplia como las plataformas más grandes, lo que limita su cobertura geográfica o la variedad de opciones disponibles inicialmente. Aún sin integración de reservas: A pesar de estar planificada, la funcionalidad de reservas aún no está implementada, mientras que competidores como TheFork y ElTenedor la ofrecen. Falta de una red social o comunidad activa: aplicaciones como Yelp y TripAdvisor tienen calificaciones, reseñas y foros activos. Foodies actualmente no incluye interacción del usuario, pero esto podría explorarse en el futuro. He de añadir que también el mercado está muy eclipsado por lo que puede ser muy complicado ganar un espacio en el mercado y esa es una limitación muy grande.

DISCUSIÓN

Análisis de los Resultados obtenidos

Durante el desarrollo de Foodies, los resultados obtenidos fueron en su mayoría positivos y acordes con los objetivos planteados al inicio del proyecto. La aplicación ha logrado con éxito su objetivo principal: proporcionar una herramienta útil y de fácil acceso para buscar y localizar restaurantes cercanos, con una interfaz sencilla, visualmente amigable y que se centra en la experiencia del usuario.

Uno de los aspectos más relevantes es la capacidad de respuesta del sistema a las solicitudes, incluyendo el rendimiento y la consistencia de los datos. Las pruebas manuales en el entorno de desarrollo verifican que las llamadas a los puntos finales del backend (como listados de restaurantes, detalles personales, inicio de sesión o recuperación de contraseña) sean consistentes y estables. También se confirmó que el sistema de autenticación del token JWT estaba funcionando de manera efectiva, protegiendo las rutas y garantizando un acceso apropiado en función del estado del usuario.

En términos de objetivos funcionales, se han implementado eficazmente funciones clave, tales como:

- ❖ Busque dinámicamente restaurantes por ubicación.
- ❖ Muestra restaurantes en un mapa interactivo.
- ❖ Filtrar por tipo de cocina.
- ❖ Administrar favoritos.

Si bien la funcionalidad de reservas no pudo completarse dentro del plazo planificado, se habían identificado las bases para su futura implementación, incluido el diseño de un modelo de datos y una comprensión clara de cómo gestionar la capacidad del restaurante, las mesas disponibles y los horarios. Una observación clave fue que la integración entre el frontend (Jetpack Compose) y el backend (Spring Boot) era bastante compleja. Este proceso fue desafiante, especialmente al procesar respuestas JSON y manejar errores de red o fallas de autenticación. Sin embargo, a través de estas dificultades, aprendimos lecciones

valiosas sobre la depuración, la comunicación entre capas y la importancia del estado en las interfaces reactivas.

También se identificaron áreas de mejora, como la necesidad de:

- ❖ Optimice el rendimiento de las llamadas a API externas (por ejemplo, para recuperar reseñas u otros detalles).
- ❖ Personalización mejorada de los filtros de restaurantes. Diseñar un sistema más robusto para perfiles de restaurantes y roles de usuario (administrador, cliente, restaurante).
- ❖ Mejorar la apariencia visual del front-end para hacerlo más atractivo e intuitivo.

En conjunto, los logros del proyecto son significativos y representativos. Aunque todavía existen algunas áreas de mejora o desarrollo futuro, Foodies ha demostrado ser una base sólida y útil. El análisis confirmó que se han alcanzado los principales objetivos del proyecto y que la aplicación tiene un alto potencial de crecimiento y mejora continua.

Reflexiones sobre las limitaciones del proyecto y posibles áreas de mejora

Durante el desarrollo del proyecto Foodies surgieron diversas limitaciones que dificultaron en cierta medida la ejecución del plan original. Uno de los factores más determinantes es el límite temporal, que afecta directamente a la implementación de determinadas funciones de planificación, como un sistema completo de reservas o la creación avanzada de un perfil de restaurante con gestión de mesas.

El proyecto también presentó dificultades con su organización interna. Como se trataba de un proyecto ambicioso con múltiples funciones y una arquitectura compleja, la planificación inicial no siempre fue lo suficientemente realista considerando el tiempo y los recursos disponibles. Las altas expectativas de lo que se puede lograr contrastan con las realidades de la era del desarrollo, obligando a una redefinición constante de prioridades.

Otro factor limitante es la disponibilidad de tecnología y recursos humanos. Como se trataba de un proyecto personal, todas las tareas de diseño, programación, pruebas y documentación fueron responsabilidad de una sola persona. Esto ralentizó ciertas etapas de desarrollo y redujo la posibilidad de tener múltiples perspectivas sobre ciertas cuestiones.

Sin embargo, estas limitaciones nos permitieron aprender a adaptar los proyectos a situaciones del mundo real, gestionar prioridades y centrarnos en los elementos esenciales para entregar una aplicación funcional. Se identificaron las siguientes áreas como posibles mejoras futuras:

- ❖ Crea un horario que se adapte mejor al tiempo disponible. Divida las tareas complejas en resultados más pequeños.
- ❖ Mejorar la organización y la funcionalidad del código mediante patrones de diseño más potentes.
- ❖ El sistema de reservas se implementará plenamente.
- ❖ Mejorar su interfaz de usuario con elementos y animaciones visualmente más atractivos.

Agregar métricas de rendimiento y análisis de uso de aplicaciones para la toma de decisiones futuras. Estas reflexiones no sólo nos ayudan a tomar conciencia de las deficiencias de nuestros procesos, sino que también allanaron el camino para seguir mejorando las soluciones que desarrollamos, permitiéndonos mirar el trabajo de desarrollo de software desde una perspectiva más madura y realista.

Implicaciones prácticas y teóricas de los resultados

Los resultados obtenidos durante el desarrollo de Foodies tienen implicaciones prácticas y teóricas en el campo del desarrollo de aplicaciones multiplataforma. Desde un punto de vista práctico, la aplicación demuestra ser una herramienta funcional, que permite a los usuarios buscar restaurantes, visualizarlos en un mapa, filtrarlos por cocina y gestionarlos como favoritos. Se trata de una solución real y aplicable que puede incorporarse en mercados o entornos de negocio donde la geolocalización y la personalización de contenidos son claves.

Su estructura modular, el uso de tokens JWT para la autenticación y la integración con API externas como Google Maps reflejan las mejores prácticas del desarrollo moderno. A lo largo del camino, obtuvimos aprendizajes importantes en la consulta de bases de datos, el trabajo con PostgreSQL y la creación de puntos finales Spring Boot eficientes para satisfacer las necesidades de nuestra aplicación.

Este conocimiento se traduce directamente en una mayor capacidad para estructurar adecuadamente las bases de datos relacionales, optimizar el acceso a los datos y mantener la consistencia de los datos. Otra implicación práctica importante fue comenzar a desarrollar servidores dedicados que ejecuten Debian. Si bien este trabajo no se completó debido a limitaciones de tiempo, representa una valiosa experiencia en la implementación y configuración de infraestructura, y establece una base sólida para futuros desarrollos más avanzados por parte del equipo técnico. Además, el proyecto demuestra claramente un enorme y extenso margen de mejoras futuras, con múltiples posibilidades de expansión: desde la implementación completa de un sistema de reservas, la mejora del diseño visual del front-end, la optimización de filtros personalizados, hasta el desarrollo de un sistema completo de roles y perfiles de restaurantes. Todo esto crea una realidad en evolución que ofrece espacio para incorporar nuevas características y tecnologías.

Desde una perspectiva teórica, el proyecto reforzó los fundamentos de la arquitectura de software, la separación de responsabilidades front-end y back-end y el diseño centrado en el usuario. También nos permite observar cómo metodologías ágiles como Scrum o el trabajo sprint impactan positivamente en la gestión iterativa de productos digitales, permitiéndonos adaptarnos a los cambios, resolver bugs incrementalmente y mejorar funcionalidades según objetivos bien definidos. También se explora el uso de tecnologías actuales como Java 21, Spring Boot, PostgreSQL y Jetpack Compose, que representan una combinación poderosa y profesional para el desarrollo de aplicaciones. Además, el análisis y la

comparación con otras aplicaciones similares en el mercado identificaron oportunidades de mejora y confirmaron que Foodies, a pesar de ser un proyecto académico, tiene valor tanto como solución práctica como ejemplo académico de integración, planificación, prueba e implementación de tecnología.

CONCLUSION

Resumen total del proyecto

Desarrollar Foodies fue una experiencia muy enriquecedora que me permitió poner en práctica y consolidar los conocimientos adquiridos a lo largo de la formación, especialmente en el trabajo con tecnologías como Java, Spring Boot, PostgreSQL y Jetpack Compose. Desde la planificación inicial hasta la implementación de funciones, este proyecto reflejó mi crecimiento técnico y personal, resolviendo desafíos reales como la creación de puntos finales eficientes, la integración con API externas, la gestión de la autenticación con JWT y la construcción de una base de datos sólida. Además, desarrollar interfaces y experiencias de usuario representa un desafío constante, pero también una oportunidad para aprender y mejorar aspectos del diseño visual y de interacción.

Una de las lecciones más importantes que aprendí fue darme cuenta de que, si bien la idea inicial del proyecto era muy ambiciosa, estaba limitado por el tiempo y los recursos disponibles, lo que me enseñó a priorizar, pivotar y tomar decisiones de desarrollo realistas. Aun así, conseguí ceñirme al objetivo principal: facilitar la búsqueda de restaurantes de una forma atractiva y útil, visualizándolo

a través de un mapa, y sentar las bases para futuras mejoras (como un sistema de reservas o gestión de personajes). También intenté crear mi propio servidor usando Debian, y aunque no pude completarlo por falta de tiempo, sigue siendo un proyecto en curso que refleja mi interés en profundizar en los aspectos de infraestructura y despliegue.

En definitiva, este proyecto representa más que un simple logro académico: es un testimonio de perseverancia, resiliencia y pasión por el desarrollo de software. Es muy gratificante ver cómo la idea inicial se transforma poco a poco en una aplicación funcional. Además, trabajar con entornos como Android Studio y Eclipse, configurar herramientas como Postman y Git y solucionar errores técnicos en tiempo real me han preparado completamente para futuros desafíos profesionales. Foodies fue mi primer paso hacia una carrera como desarrollador y sentó una base sólida para el desarrollo futuro.

Lista de referencias

Desarrollador de Android.

(2025). Listas y cuadrículas – Jetpack Compose. Desarrollador de Android. Recuperado de [https://developer.android.com/develop/ui/compose/lists:contentReference\[oaicite:0\]{index=0}](https://developer.android.com/develop/ui/compose/lists:contentReference[oaicite:0]{index=0}).

Desarrollador de Android. (2025). Navegación con Compose. Desarrollador de Android. Recuperado de [https://developer.android.com/develop/ui/compose/navigation:contentReference\[oaicite:1\]{index=1}](https://developer.android.com/develop/ui/compose/navigation:contentReference[oaicite:1]{index=1}).

Desarrollador de Android. (2025). Accesibilidad en Jetpack Compose. Desarrollador de Android. Recuperado de [https://developer.android.com/develop/ui/compose/accessibility:contentReference\[oaicite:2\]{index=2}](https://developer.android.com/develop/ui/compose/accessibility:contentReference[oaicite:2]{index=2}).

Desarrollador de Android. (2025). Descripción general del kit de herramientas de interfaz de usuario de Jetpack Compose. Desarrollador de Android. Obtenido de [https://developer.android.com/compose:contentReference\[oaicite:3\]{index=3}](https://developer.android.com/compose:contentReference[oaicite:3]{index=3}).

Google Maps para Android

Google Developers. (2025). Marcador - SDK de Mapas de Android.

Google Developers. Obtenido de

[https://developers.google.com/maps/documentation/android-sdk/marker:contentReference\[oaicite:4\]{index=4}:contentReference\[oaicite:5\]{index=5}](https://developers.google.com/maps/documentation/android-sdk/marker:contentReference[oaicite:4]{index=4}:contentReference[oaicite:5]{index=5}).

Google Developers. (2025). Lanzamiento de Google Maps - SDK de

Mapas de Android (Intents). Google Developers. Obtenido de

[https://developers.google.com/maps/documentation/android-sdk/intents:contentReference\[oaicite:6\]{index=6}:contentReference\[oaicite:7\]{index=7}](https://developers.google.com/maps/documentation/android-sdk/intents:contentReference[oaicite:6]{index=6}:contentReference[oaicite:7]{index=7}).

Retrofit (Cliente HTTP REST)

Square, Inc. (San Francisco). Retrofit: Un cliente HTTP con seguridad de tipos para Android y Java. GitHub de Square. Recuperado de [https://square.github.io/retrofit/:contentReference\[oaicite:8\]{index=8}:contentReference\[oaicite:9\]{index=9}](https://square.github.io/retrofit/:contentReference[oaicite:8]{index=8}:contentReference[oaicite:9]{index=9}).

Desarrollador de Android. (2024). Recuperación de datos de internet

(Codelab). Desarrollador de Android. Recuperado de

[https://developer.android.com/codelabs/basic-android-kotlin-compose-getting-data-internet:contentReference\[oaicite:10\]{index=10}](https://developer.android.com/codelabs/basic-android-kotlin-compose-getting-data-internet:contentReference[oaicite:10]{index=10}).

Arquitectura MVVM (ViewModel y StateFlow)

Desarrollador de Android. (2025). Descripción general de ViewModel.

Desarrollador de Android. Obtenido de

[https://developer.android.com/topic/libraries/architecture/viewmodel:contentReference\[oaicite:11\]{index=11}](https://developer.android.com/topic/libraries/architecture/viewmodel:contentReference[oaicite:11]{index=11}). Desarrollador de Kotlin. (2024). StateFlow y SharedFlow.

Desarrollador de Android. Obtenido de

[https://developer.android.com/kotlin/flow/stateflow-and-sharedflow:contentReference\[oaicite:12\]{index=12}:contentReference\[oaicite:13\]{index=13}](https://developer.android.com/kotlin/flow/stateflow-and-sharedflow:contentReference[oaicite:12]{index=12}:contentReference[oaicite:13]{index=13}).

Coil (Carga de imágenes en Compose)

Coil Contributors. (2023). Integración de Compose (Coil). Documentación oficial de Coil. Obtenido de [https://coil-kt.github.io/coil/compose:contentReference\[oaicite:14\]{index=14}:contentReference\[oaicite:15\]{index=15}](https://coil-kt.github.io/coil/compose:contentReference[oaicite:14]{index=14}:contentReference[oaicite:15]{index=15}).

Diseño UI/UX moderno y accesible (Compose)

Desarrollador de Android. (2025). Accesibilidad en Jetpack Compose. Desarrollador de Android. Recuperado de [https://developer.android.com/develop/ui/compose/accessibility:contentReference\[oaicite:16\]{index=16}](https://developer.android.com/develop/ui/compose/accessibility:contentReference[oaicite:16]{index=16}).

Desarrollador de Android. (2025). Descripción general del kit de herramientas de interfaz de usuario de Jetpack Compose. Desarrollador de Android. Recuperado de [https://developer.android.com/compose:contentReference\[oaicite:17\]{index=17}](https://developer.android.com/compose:contentReference[oaicite:17]{index=17}).

Navegación con NavController Desarrollador de Android. (2025). Creación de un controlador de navegación. Desarrollador de Android. Recuperado de [https://developer.android.com/guide/navigation/navcontroller:contentReference\[oaicite:18\]{index=18}](https://developer.android.com/guide/navigation/navcontroller:contentReference[oaicite:18]{index=18}).

Desarrollador de Android. (2025). Navegación con Compose. Desarrollador de Android. Obtenido de [https://developer.android.com/develop/ui/compose/navigation:contentReference\[oaicite:19\]{index=19}](https://developer.android.com/develop/ui/compose/navigation:contentReference[oaicite:19]{index=19}). Autenticación y tokens Desarrolladores de Android. (2024).

Consejos y mejores prácticas de seguridad. Desarrolladores de Android. Obtenido de <https://developer.android.com/privacy-and-security/security->

tips:contentReference[oaicite:20]{index=20}. Perfiles de usuario (opcional)

Firebase. (2024). Administración de usuarios: Firebase Authentication (Android).

Documentación de Firebase.

Obtenido de [https://firebase.google.com/docs/auth/android/manage-users:contentReference\[oaicite:21\]{index=21}](https://firebase.google.com/docs/auth/android/manage-users:contentReference[oaicite:21]{index=21}).