

Noviembre 2025

PEDIDOS YAPO

Asignatura: DSY1105 – Desarrollo de Aplicaciones Móviles

Estudiante: Vicente Sanchez

Profesor: Atanacio Montano

Fecha: 24-11-2025

Vicente Sanchez

ÍNDICE DE CONTENIDOS

01. Inicio del Proyecto

02. Arquitectura

General

03. Microservicios

04. API Externa

05. Persistencia Remota

06. Pruebas Unitarias

07. APK Firmado

08. Colaboración y
Modificación

CONTEXTO DEL PROYECTO

- Aplicación móvil funcional desarrollada en Android Studio (Kotlin + Compose).
- Caso: Sistema de pedidos de comida con carrito, restaurantes, productos, pago y persistencia.
- Integración con:
 - Microservicios Spring Boot
 - APIs externas
 - Room Database
 - Pruebas unitarias
 - APK firmado
- Proyecto usado como base para la Evaluación Final Transversal.

Sobre el Proyecto

INTRODUCCIÓN

Pedidos Yapo es una aplicación móvil desarrollada en Kotlin + Jetpack Compose, diseñada para gestionar pedidos de comida de forma rápida y sencilla. La app integra microservicios en Spring Boot, consumo de API externa, persistencia en MySQL, y arquitectura MVVM, garantizando un funcionamiento fluido entre frontend y backend.

Este proyecto demuestra una solución real y completamente funcional, cumpliendo con los requisitos de la Evaluación Parcial 5 y preparando el camino para el Examen Final Transversal.

O B J E T I V O S

01

Desarrollar una aplicación móvil funcional que permita gestionar pedidos de comida mediante una interfaz clara y fácil de usar.

02

Integrar microservicios en Spring Boot para garantizar un flujo de datos estable entre la app, el backend y la base de datos MySQL.

03

Implementar consumo de API externa, pruebas unitarias y generación del APK firmado para cumplir con los requisitos técnicos de la evaluación.

I N I C I O D E L P R O Y E C T O Y E N T O R N O (1)

Elementos para Android Studio :

- “Verifico la ejecución en emulador/dispositivo físico.”
- “Uso Android Studio Flamingo/Narwhal, SDK 34 y Kotlin.”
- “Mi app está estructurada en MVVM y utiliza Retrofit, Room y Coroutines.”

Mostrar en pantalla:

- ✓ App corriendo
- ✓ Árbol del proyecto
- ✓ build.gradle principal

MICROSERVICIOS Y BACKEND (1)

Microservicios construidos en Spring Boot:

- Productos
- Restaurantes
- Pedidos
- PedidoItem
- PedidoDetalle

Tecnologías:

- Spring Boot / Java
- JPA + MySQL (Railway)
- Swagger
- Pruebas JUnit
- Control de versiones GitHub

ARQUITECTURA GENERAL (2)

Explicar:

- Arquitectura MVVM de la app:
- Model – ViewModel – UI (Compose)
- Arquitectura REST + microservicios

Flujo completo:

1. App → Retrofit → Microservicio
2. Microservicio → MySQL
3. Respuesta JSON → App
4. La UI reacciona con StateFlow

ESTRUCTURA DE PAQUETES (2B)

APP (Android):

- entity/
- dao/
- repository/
- viewmodel/
- ui/screens/
- retrofit/

MICROSERVICIOS:

- controller/
- service/
- repository/
- model/
- config/

CONSUMO DE API EXTERNA (4)

RetrofitService implementado:

- Endpoints
- DTOs
- Coroutines + suspend functions

Explicar diferencia requerida por la pauta:

- API Externa: datos que no controle yo.
- Microservicio propio: datos almacenados en mi app (MySQL).

PERSISTENCIA REMOTA (5)

Flujo de datos:

- App → Retrofit → Microservicio → MySQL → respuesta → ViewModel
- Insert, Update y Delete funcionando
- Acciones realizadas en vivo según el profesor lo pida

Requisito explícito del PDF:

Debe insertar, actualizar o eliminar datos en tiempo real.

PRUEBAS UNITARIAS (6)

Mostrar ejecución REAL de pruebas

(Esp. pruebas con Room, Repository o ViewModel)

Debes decir:

- Qué se prueba
- Qué se mockea
- Por qué
- Qué significa que la prueba esté OK

FIRMA Y GENERACIÓN DEL APK (7)

Explicar paso a paso:

AgreBuild → Generate Signed APK

- Creación del keystore
- Configuración del alias
- Selección de release
- Ubicación del archivo final .apk
- Mostrar tu APK firmado en una carpeta

ar algo de texto

EVIDENCIA DE COLABORACIÓN (8)

GitHub:

- Commits
- Branches
- Pull requests
- Código aportado por ti

Trello:

- Tareas realizadas
- Responsable
- Estado de avance

MODIFICACIÓN EN TIEMPO REAL (9)

Ejemplos que puedes ofrecer:

- Agregar un botón
- Cambiar un texto
- Crear un endpoint
- Modificar un ViewModel
- Ajustar un servicio
- Corregir un bug en vivo

CONCLUSIONES

- Demostraste la app corriendo
- Integración completa con microservicios
- Persistencia implementada
- Pruebas unitarias funcionando
- APK firmado generado
- Trabajo colaborativo evidenciado

P R E G U N T A S

(El profesor debe hacerte preguntas técnicas. No te preocupes, tengo respuestas para todas si quieres practicar.)

Octubre 2030

MUCHAS
GRACIAS

Bruno Lago