

# EAT NOW – APP DE PEDIDOS DE COMIDA LOCAL

**Subtítulo: De MVP móvil a Web:  
catálogo, carrito, checkout y  
confirmación.**

Integrantes: Fabián Córdova

Vicente Sánchez

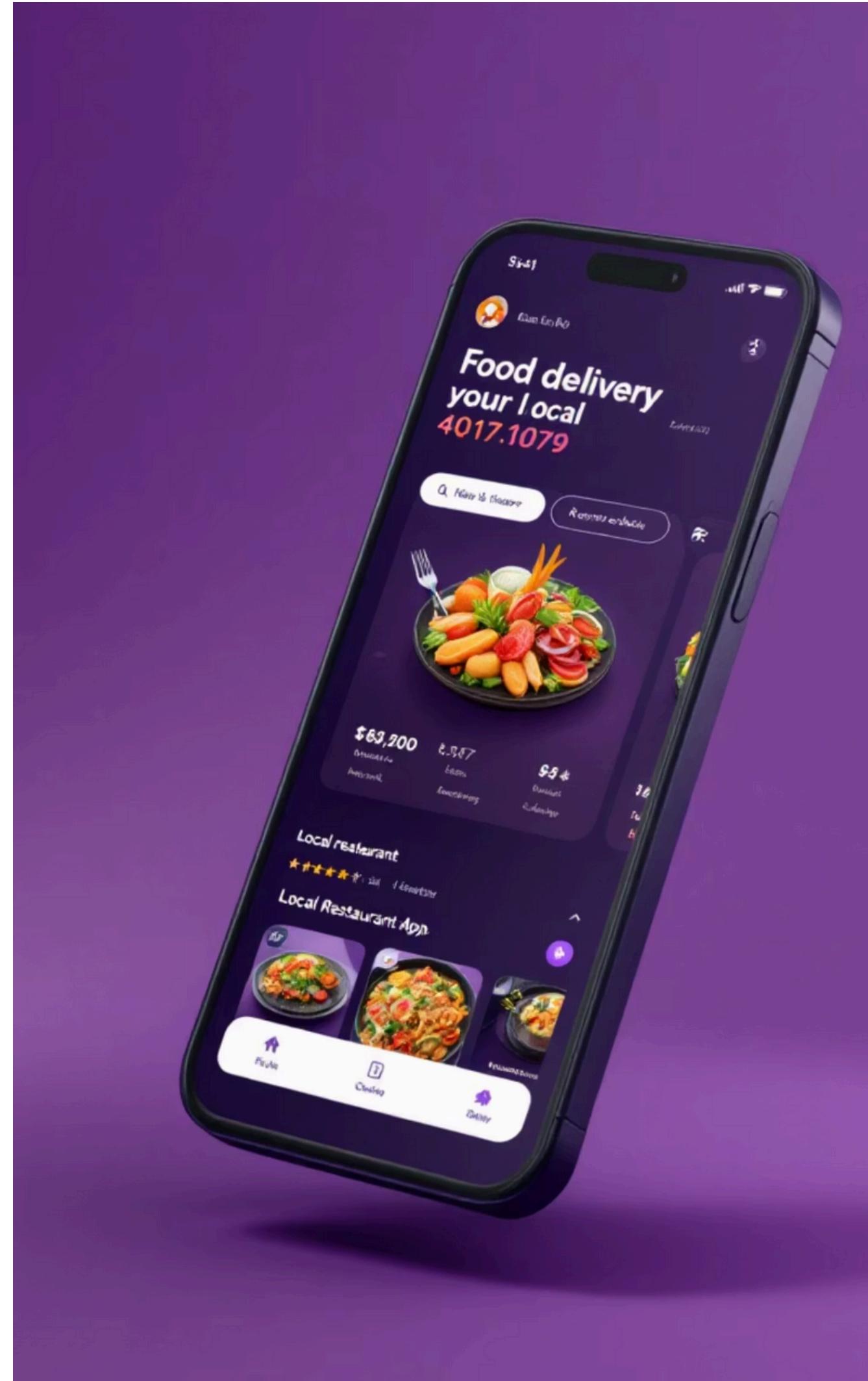
Asignatura: Desarrollo de Aplicaciones Móviles

Sección: 004D

Profesor: Atanacio Montano

Fecha: 24-10-2025

- Fabián Córdova – desarrollo de la aplicación (implementación y demo).
- Vicente Sánchez – arquitectura, diseño UX/UI y validaciones.



# PROBLEMÁTICA IDENTIFICADA

- Emprendimientos sin canal web propio → pedidos por WhatsApp/llamadas.
- Errores de toma de pedido, demoras y poca trazabilidad.
- Falta de visibilidad del menú y precios actualizados.
- Objetivo: habilitar un sitio web responsive para pedidos rápidos, claros y trazables.

Estos seian

Visual sugerido: iconos “ →  → <img alt="Green checkmark icon" data-bbox="468 641 508 678}”.</p>

# PROPUESTA DE SOLUCIÓN: EAT NOW

Convertimos los pedidos locales en una experiencia web rápida, clara y trazable, accesible desde cualquier dispositivo.



## MENÚ DIGITAL COMPLETO

listado de productos con precios y descripciones.



## CARRITO INTELIGENTE

agrega, elimina y confirma pedidos en tiempo real.



## PERFIL PERSONALIZADO

permite visualizar compras anteriores y detalles.



## RECURSOS NATIVOS

Integración con GPS, cámara y sistema de notificaciones para una experiencia completa.

- Diseñada con Material Design 3 para garantizar una experiencia moderna, fluida y accesible en todos los dispositivos Android.

Beneficio: facilita la gestión para el negocio y simplifica la experiencia del cliente.

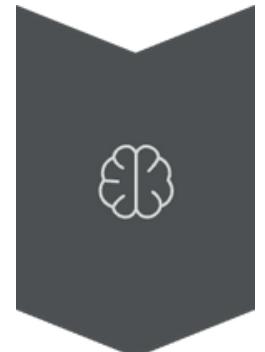
# ARQUITECTURA TÉCNICA DEL PROYECTO

ARQUITECTURA: MVVM + ROOM + JETPACK COMPOSE + MATERIAL 3



## Capa de Presentación (UI)

Diseñada con Jetpack Compose (pantallas de Productos, Carrito, Pedidos, Perfil).



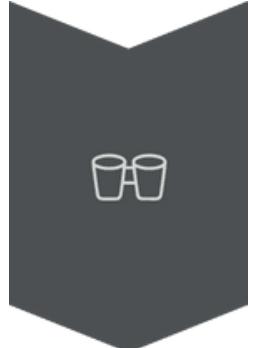
## Capa de Lógica (ViewModel)

Usa StateFlow y LiveData para gestionar el estado y la comunicación con la base de datos.



## Capa de Datos (Room)

Contiene entidades, DAOs y repositorios para guardar productos, pedidos y detalles..



## Repositorio Central

Coordinador el acceso a los datos y mantiene la lógica separada de la UI.



# INTEGRACIÓN DE RECURSOS NATIVOS

PEDIDOSYAPO USA FUNCIONES NATIVAS DEL DISPOSITIVO PARA MEJORAR LA EXPERIENCIA:

- 🔔 **NOTIFICACIONES:** SE GENERA UNA ALERTA VISUAL AL COMPLETAR UN PEDIDO MEDIANTE NOTIFICATIONMANAGERCOMPAT.
- ▢ **COMPATIBILIDAD ANDROID 13+:** SOLICITA PERMISOS DE NOTIFICACIÓN (POST\_NOTIFICATIONS) EN TIEMPO REAL.
- 📡 **INTEGRACIÓN LOCAL:** BASE DE DATOS PERSISTENTE Y MANEJO OFFLINE GRACIAS A ROOM.

**EXPERIENCIA FLUIDA GRACIAS A MATERIAL DESIGN 3 Y COMPONENTES NATIVOS COMPOSE.**

# **SISTEMA DE FORMULARIOS VALIDADOS**

## **FORMULARIOS IMPLEMENTADOS:**

- REGISTRO DE USUARIO (CORREO, CONTRASEÑA).
- FORMULARIO DE PEDIDO (NOMBRE DEL CLIENTE, DIRECCIÓN, COMUNA, MÉTODO DE PAGO).
- CONFIRMACIÓN FINAL ANTES DE REGISTRAR EL PEDIDO.

## **VALIDACIONES:**

- CAMPOS OBLIGATORIOS (NO SE PERMITE PEDIDO VACÍO).
- VALIDACIÓN DE TEXTO MÍNIMO EN CAMPOS COMO "NOMBRE" O "COMUNA".
- MENSAJES VISUALES MEDIANTE TOAST Y SNACKBARS AL AGREGAR PRODUCTOS O CONFIRMAR PEDIDOS.

# DISEÑO Y EXPERIENCIA DE USUARIO



## IDENTIDAD VISUAL:

TEMA MODERNO CON PALETA VERDE OSCURO Y NARANJA BRILLANTE (MATERIAL 3).

## CARACTERÍSTICAS:

⌚ NAVEGACIÓN FLUIDA: ESTRUCTURA CON NAVHOST Y COMPOSABLE DINÁMICOS.

⭐ ANIMACIONES: TRANSICIONES CON ANIMATEDVISIBILITY Y FEEDBACK VISUAL INMEDIATO.

► DISEÑO ADAPTABLE: OPTIMIZADO PARA DISPOSITIVOS MÓVILES Y TABLETS ANDROID.

**RESULTADO:** UNA EXPERIENCIA RÁPIDA, VISUAL Y FUNCIONAL PARA EL USUARIO.



# ALMACENAMIENTO Y GESTIÓN DEL PROYECTO

## Estrategia de almacenamiento:

- 📁 **Room Database:** guarda pedidos, productos y detalles localmente.
- 💾 **Persistencia local:** la información se conserva incluso al cerrar la app.
- 🌐 **Posibilidad de conexión futura con API REST.**

## Metodología de desarrollo:

**GitHub** → control de versiones, ramas de desarrollo y commits del equipo.

**Trello** → planificación de tareas: diseño, integración Room, validaciones, testing.

# Diapositiva 9 - Conclusiones y Proyección Futura

## RESULTADOS OBTENIDOS:

- ✓ SE LOGRÓ UN FLUJO COMPLETO DE PEDIDOS DENTRO DE LA APP.
- ✓ REDUCCIÓN DE ERRORES HUMANOS AL REGISTRAR COMPRAS.
- ✓ VALIDACIONES FUNCIONALES Y PERSISTENCIA DE DATOS CON ROOM.

## PROYECCIÓN FUTURA:

IMPLEMENTAR PASARELA DE PAGO ONLINE.

AÑADIR NOTIFICACIONES PUSH MEDIANTE FIREBASE.

SINCRONIZACIÓN EN LA NUBE (API O BACKEND REST).

## **Demo + Cambio en vivo**

Objetivo: mostrar el flujo completo y agregar una validación nueva sin romper la app.

🎥 Objetivo: mostrar el flujo y una modificación real en el código.

Parte 1 – Fabián (1:30):

Recorrido desde catálogo hasta confirmación de pedido.

Parte 2 – Vicente (1:30):

Agregar campo “Comuna” obligatorio en Checkout, con validación visual y persistencia.

Criterios de éxito:

- Validación bloquea envío hasta corregir.
- Mensaje de confirmación visible (Toast/Snackbar).
- Campo “Comuna” se conserva tras recargar la app.

*The End*

