Examen TA_2 – Aplicación "Comidita"

Jorge Vargas

2025-08-18

Portada

Este informe corresponde al examen TA_2 de la asignatura de Programación I. Se presenta la aplicación **Comidita**, un sistema básico de restaurante desarrollado en Android con Kotlin y ConstraintLayout.

Objetivos

- Desarrollar una aplicación Android que permita simular el flujo de un pedido en un restaurante.
- Aplicar conceptos de programación orientada a objetos mediante las clases `ItemMenu`, `ItemMesa` y `CuentaMesa`.
- Usar **ConstraintLayout** para diseñar la interfaz de usuario.
- Implementar la lógica de pedidos: agregar, quitar, calcular totales y propina.
- Simular el envío de un pedido y reinicio de la aplicación.

Desarrollo

Repositorio github:

https://github.com/afromankenobi/comidita

Clases principales

1. `ItemMenu`: representa un plato disponible (nombre y precio).
2. `ItemMesa`: representa un ítem pedido por la mesa (plato + cantidad).
3. `CuentaMesa`: gestiona el pedido total, sumando ítems, calculando subtotales, totales y propinas.

Actividad principal

- Se creó la actividad `MenuActivity`, que conecta los elementos de UI con la lógica de las clases.
- Se usó **ConstraintLayout** para organizar botones, textos y controles.
- Se implementaron botones para agregar y quitar ítems, un switch para la propina, y botones de control: **Nuevo pedido** y **Aceptar**.

Lógica de flujo

- El usuario selecciona platos y cantidades.
- La aplicación calcula subtotales, total y propina (10%).
- Al presionar **Aceptar**, se simula el envío del pedido (**ProgressBar**), y luego se confirma con un mensaje **Toast**.
- Finalmente, la app reinicia la cuenta para tomar un nuevo pedido.

Validación de negativos

Un detalle importante de la implementación es la validación para evitar cantidades negativas:

- Cuando la cantidad de un plato llega a 0, el botón de **restar** se deshabilita automáticamente.
- Esto asegura que no se puedan ingresar valores inválidos y que la lógica de los cálculos (subtotal, total y propina) siempre sea consistente.
- La validación se realiza tanto en el método `refrescarUI()` como en `setControlesEnabled()`, comprobando que la cantidad sea mayor a 0 antes de habilitar el botón de restar.

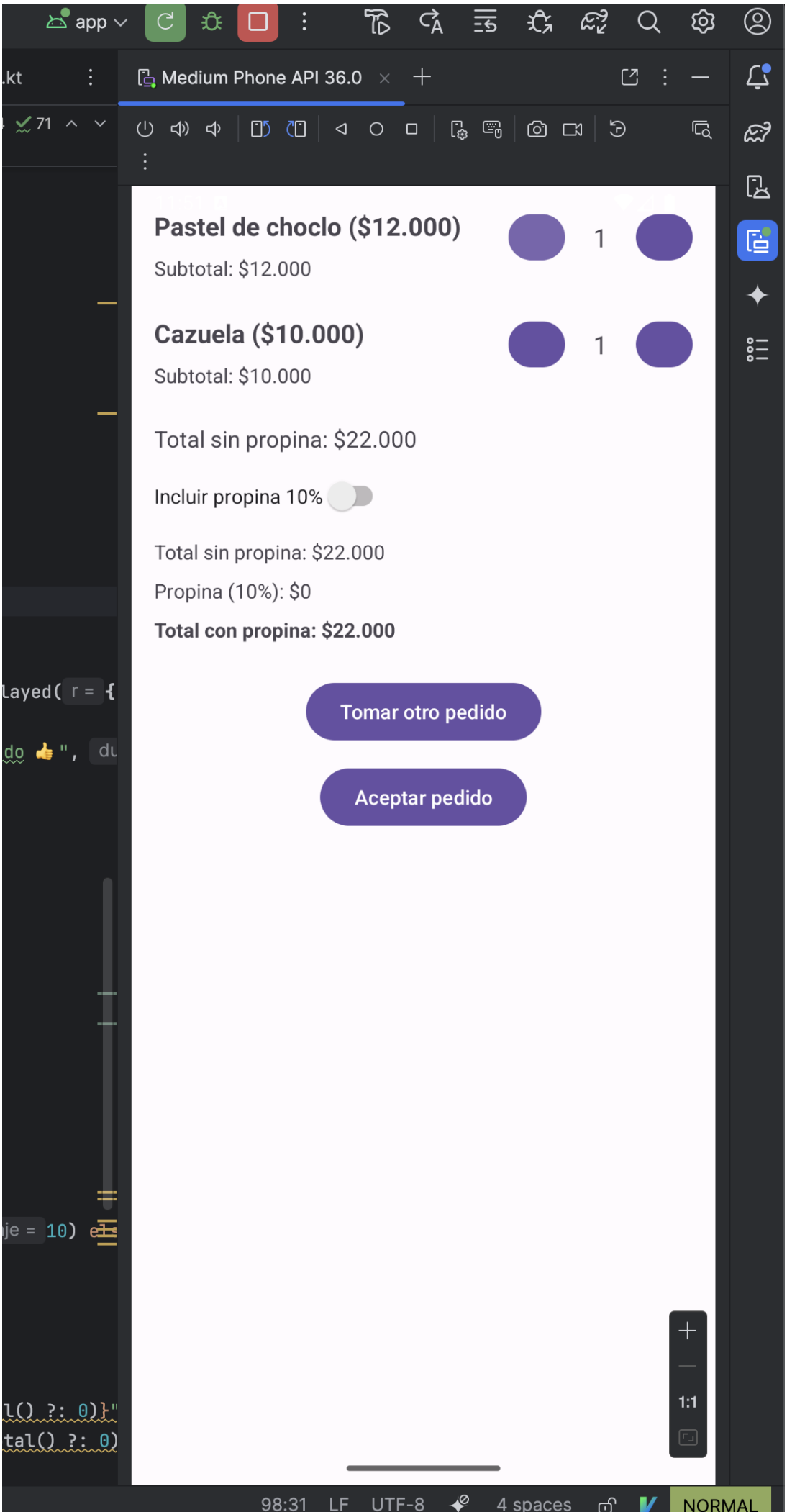
Falta de imagenes

Me faltaron las imagenes, lo se. Perdón :(

Capturas de Pantalla ATTACH

Aquí se insertarán las capturas de la aplicación en funcionamiento:

- Interfaz principal con los platos.


app
Medium Phone API 36.0
Pastel de choclo ($12.000)
1
Subtotal: $12.000
Cazuela ($10.000)
1
Subtotal: $10.000
Total sin propina: $22.000
Incluir propina 10%
Total sin propina: $22.000
Propina (10%): $0
Total con propina: $22.000
Tomar otro pedido
Aceptar pedido
1:1
98:31
LF
UTF-8
4 spaces
NORMAL

• Simulación de envío del pedido (**ProgressBar**).

Pastel de choclo ($12.000)

1

Subtotal: $12.000

Cazuela ($10.000)

2

Subtotal: $20.000

Total sin propina: $32.000

Incluir propina 10%

Total sin propina: $32.000

Propina (10%): $3.200

Total con propina: $35.200

Tomar otro pedido

Aceptar pedido

Conclusión

La aplicación cumple con los requerimientos de la prueba

- Implementa las clases necesarias
- Integra la lógica con la interfaz en Android usando Kotlin y ConstraintLayout.
- Valida adecuadamente que las cantidades de ítems no sean negativas.
- Ofrece una experiencia básica pero completa de simulación de un pedido en un restaurante.
- Permite reiniciar el proceso para múltiples pedidos.

> Se logró aplicar los conceptos de POO y Android Studio, entregando un producto funcional, claro y extensible.