Inventario de Videojuegos - Documentación del Proyecto

Autor: Roger López  
Fecha: 30/09/2025  
  
Carrera/Asignatura: Programación II  
Primera Parcial

# Índice

1. Objetivo y alcance

2. Requisitos del sistema

3. Arquitectura y tecnologías

4. Modelo de datos (Room)

5. Estructura de paquetes

6. Funcionalidades implementadas (CRUD)

7. Flujo de navegación

8. Pruebas y evidencias

9. Instrucciones de ejecución / instalación

10. Conclusiones y trabajo futuro

11. Referencias

# 1. Objetivo y alcance

Desarrollar una aplicación Android que gestione el inventario de una tienda de videojuegos.

El sistema permite administrar categorías, videojuegos, clientes y ventas, manteniendo la consistencia del stock.

Alcance de esta entrega: CRUD completo para Videojuegos, Clientes y Categorías; registro de Ventas con al menos un detalle y ajuste automático de stock.

# 2. Requisitos del sistema

- Android Studio Giraffe/Koala o superior.

- SDK mínimo 24 (Android 7.0), target 34.

- Gradle con Kotlin DSL (build.gradle.kts) y ViewBinding activado.

- Dispositivo/emulador Android.

# 3. Arquitectura y tecnologías

Patrón por capas simple (data → ui) con navegación por fragments.

Tecnologías clave:

- Room (SQLite): @Entity, @Dao, @Database, @Relation.

- RecyclerView + Adapter para listados.

- ViewBinding para acceso seguro a vistas.

- Navigation Component + BottomNavigation + Toolbar.

- Material Components para UI (FAB, TextInputLayout, CardView, etc.).

# 4. Modelo de datos (Room)

Entidades:

• Categoria(categoriaId, nombre)

• Videojuego(videojuegoId, titulo, precio, stock, categoriaId)

• Cliente(clienteId, nombre, email)

• Venta(ventaId, clienteId, fechaMillis)

• DetalleVenta(detalleId, ventaId, videojuegoId, cantidad)

Relaciones:

- Categoria 1..N Videojuego (FK: categoriaId)

- Cliente 1..N Venta (FK: clienteId)

- Venta 1..N DetalleVenta (FK: ventaId)

- Videojuego 1..N DetalleVenta (FK: videojuegoId)

Reglas de integridad:

- Borrado en cascada de detalles al eliminar una Venta.

- Restricción para no borrar categoría/videojuego referenciados (RESTRICT).

# 5. Estructura de paquetes

com.uninorte.inventario

└─ data

├─ entity (Categoria, Videojuego, Cliente, Venta, DetalleVenta, POJOs @Relation)

├─ dao (CategoriaDao, VideojuegoDao, ClienteDao, VentaDao, DetalleVentaDao)

└─ db (AppDatabase)

└─ ui

├─ videojuegos (InventarioFragment, VideojuegoAdapter)

├─ clientes (ClientesFragment, ClienteAdapter)

├─ ventas (VentasFragment, VentasAdapter)

└─ settings (SettingsFragment)

# 6. Funcionalidades implementadas (CRUD)

- Categorías: crear, editar, eliminar.

- Videojuegos: crear, listar, editar, eliminar, actualización de stock.

- Clientes: crear, listar, editar, eliminar.

- Ventas: crear venta con un detalle mínimo (cliente, videojuego, cantidad). Descuenta stock.

- Eliminar venta: devuelve el stock de cada detalle antes de borrar.

# 7. Flujo de navegación

Activity principal con NavHostFragment, Toolbar y BottomNavigation.

Tabs: Inventario (inicio), Clientes, Ventas, Ajustes.

En Inventario, FAB abre diálogo para alta/edición de videojuegos. Similar para Clientes y Ventas.

# 9. Instrucciones de ejecución / instalación

- Clonar o abrir el proyecto en Android Studio.

- Sincronizar Gradle.

- Ejecutar en dispositivo físico con depuración USB o emulador.

- Si no aparecen categorías en el spinner, desinstalar e instalar de nuevo para disparar el prepopulate().

# 10. Conclusiones y trabajo futuro

La app cumple los requisitos principales del examen: 4+ tablas con relaciones, 2 CRUD completos, navegación y uso de componentes modernos.

Mejoras posibles: búsqueda/filtrado, múltiples detalles por venta en un carrito, validaciones avanzadas, informes y exportación, tema oscuro y pruebas instrumentadas.

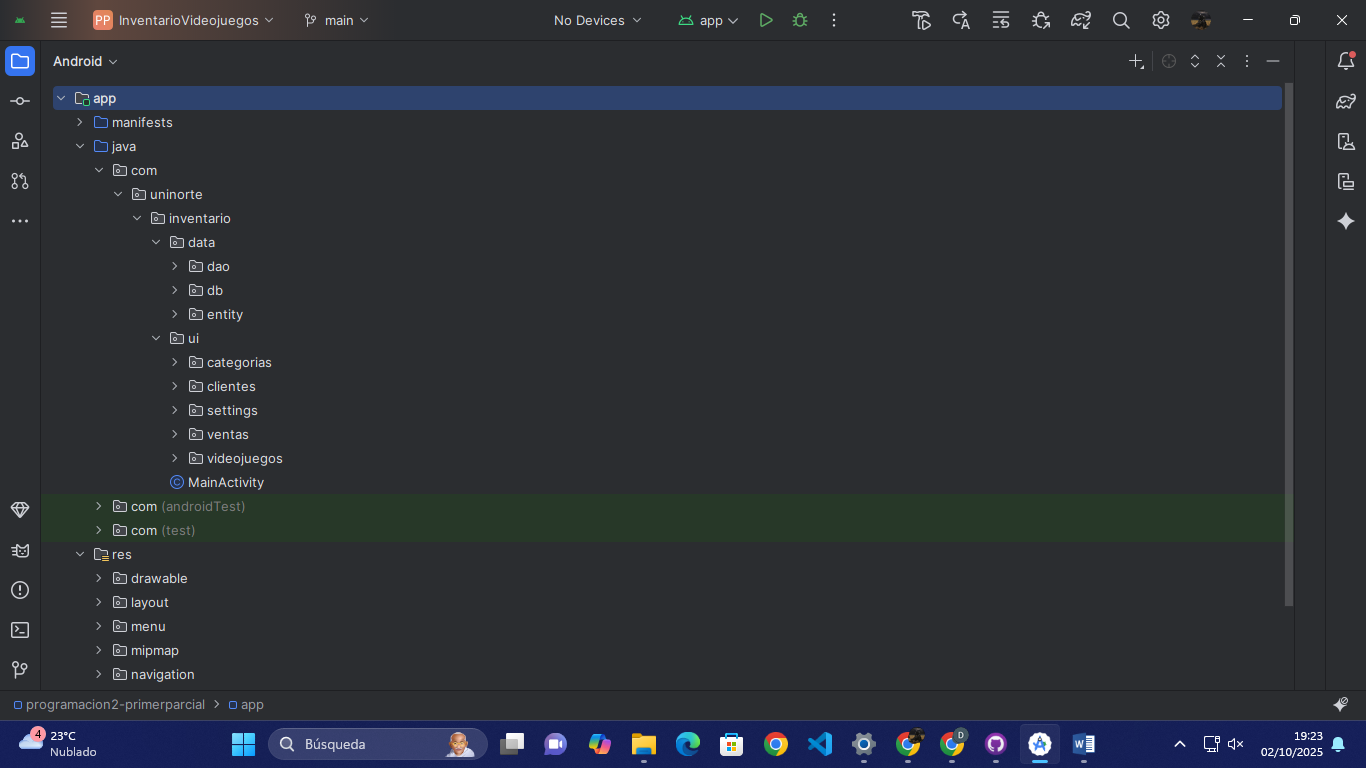
# 11. Referencias

- Android Developers: Room, Navigation, RecyclerView, Material Components.

- Guía de estilos y buenas prácticas de arquitectura en Android.

# 8. Pruebas y evidencias (con capturas)

Fecha de inserción: 30/09/2025 02:15



Captura de la estructura del proyecto.

Figura 1. Captura de pantalla 2025-09-29 225305.png — Evidencia

Caso de prueba específico según la imagen; se valida el comportamiento esperado.

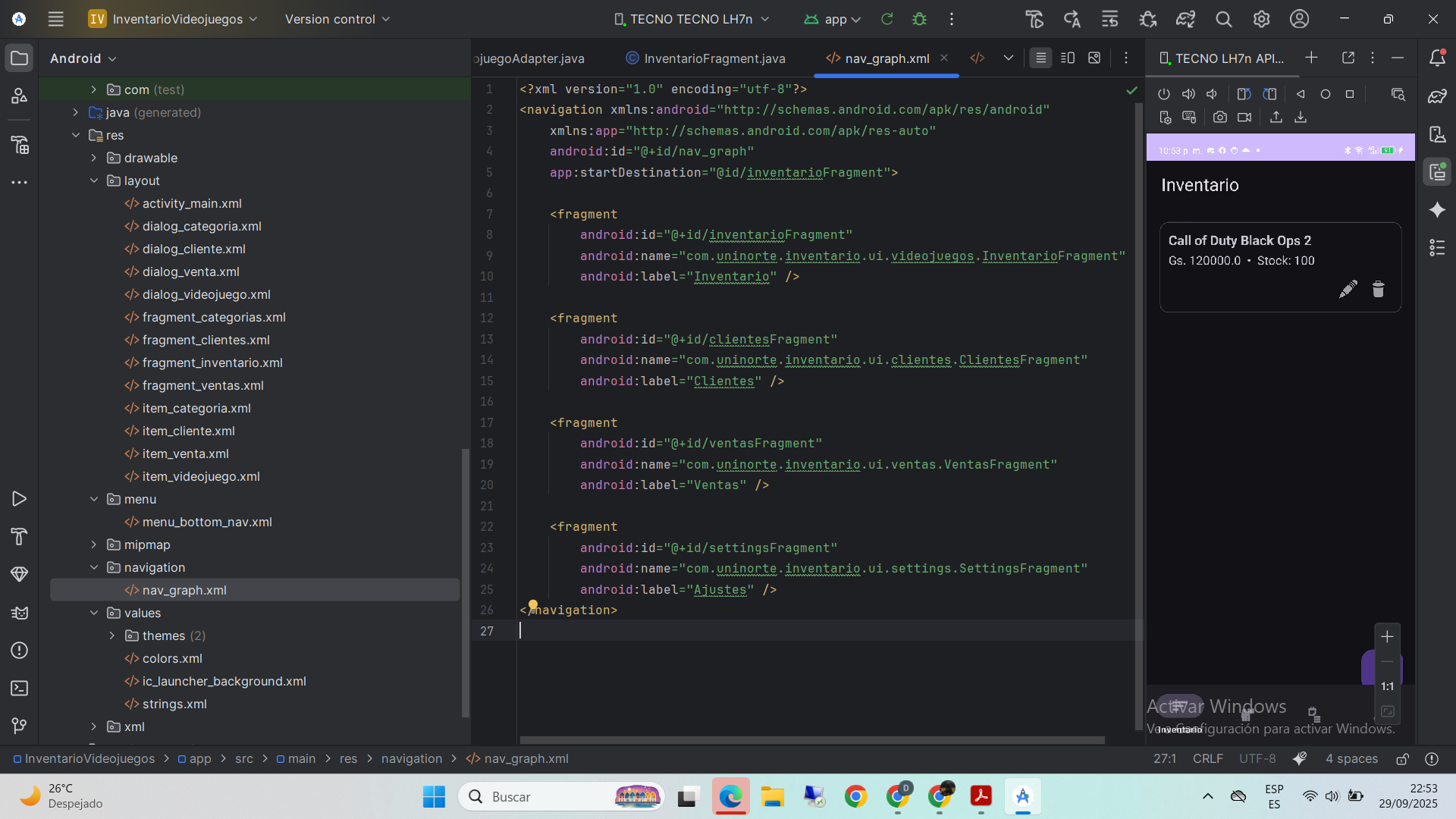


Figura 2. Captura de pantalla 2025-09-29 225343.png — Evidencia

**Navigation Graph** con los fragments.



Figura 3. Screenshot\_20250929\_224615.png — Evidencia

Captura del icono de la app.



Figura 4. Screenshot\_20250929\_224655.png — Evidencia

Captura dentro de la app.

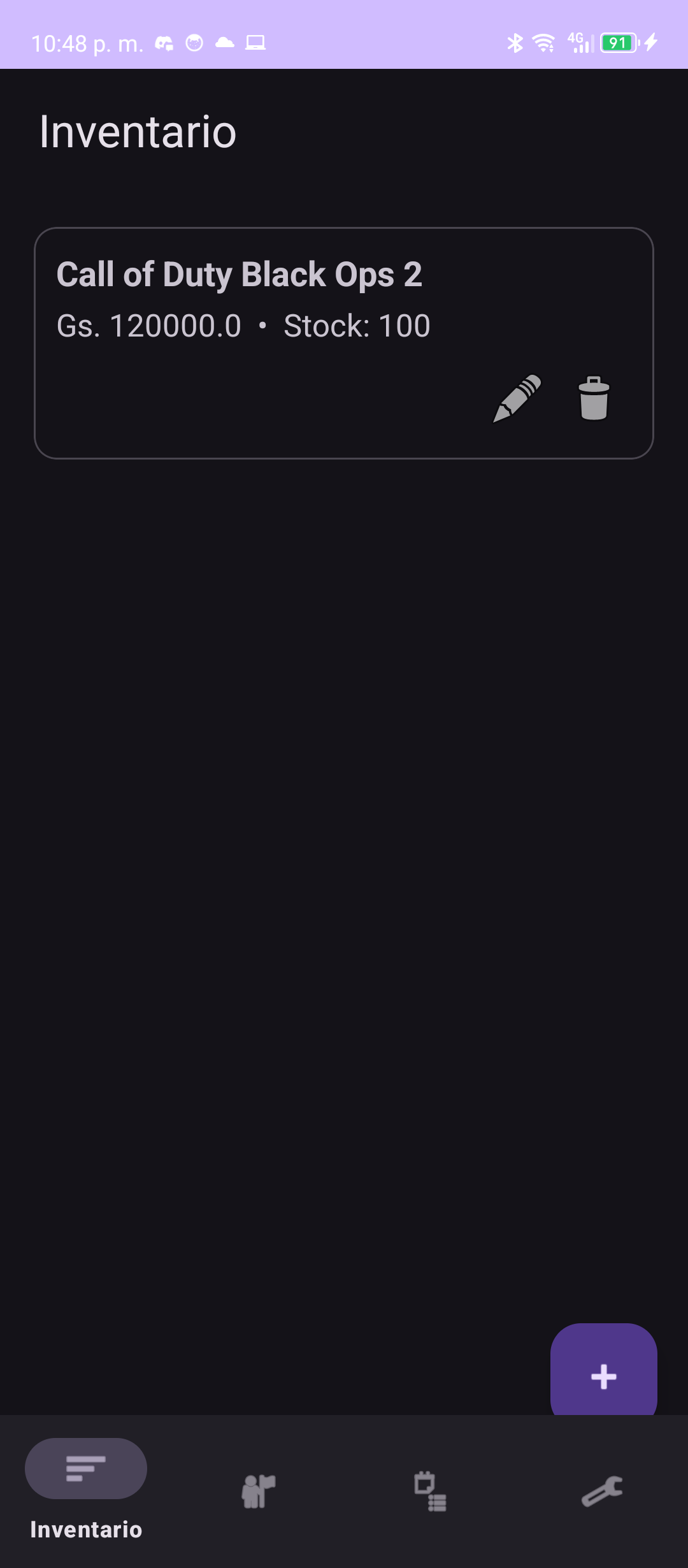


Figura 5. Screenshot\_20250929\_224801.png — Evidencia

Captura de en la app con un articulo añadido.

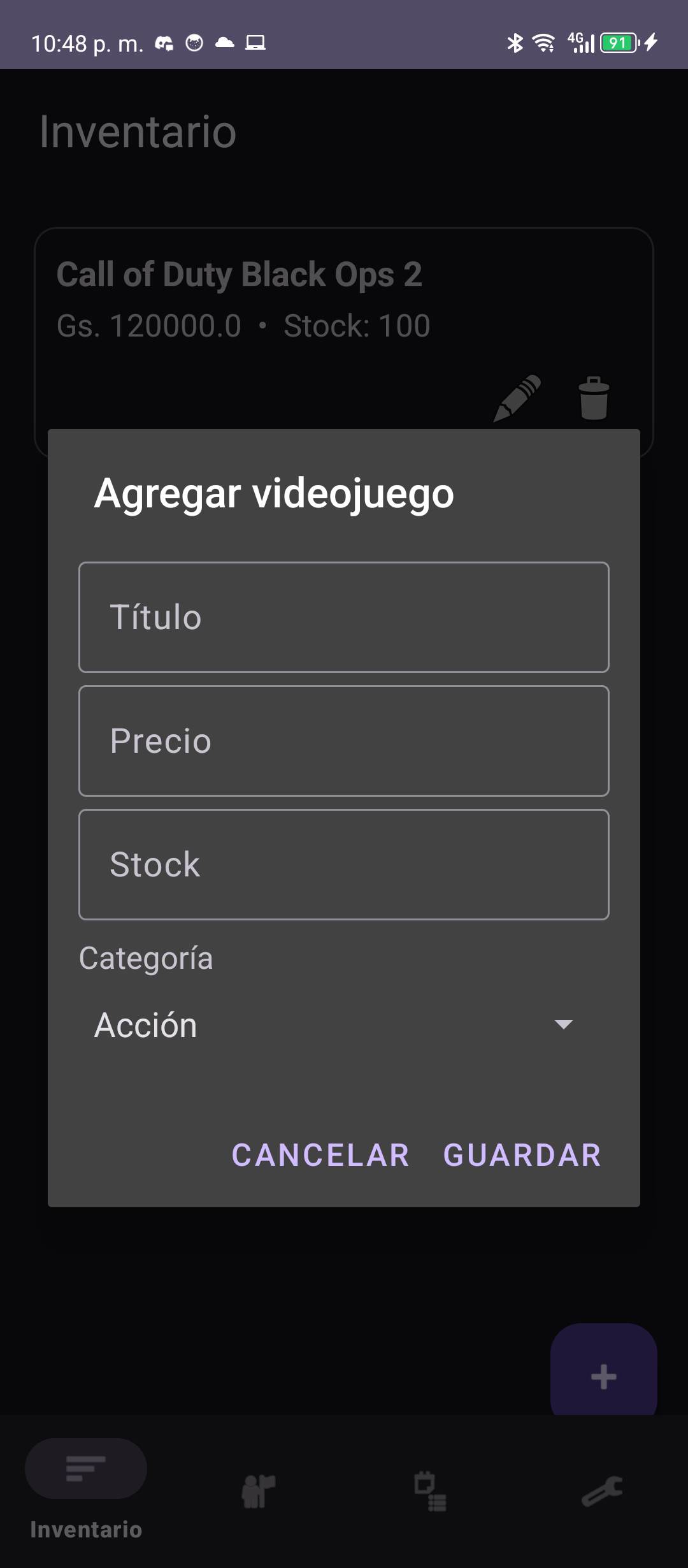


Figura 6. Screenshot\_20250929\_224825.png — Evidencia

Captura de la pantalla para añadir un articulo.

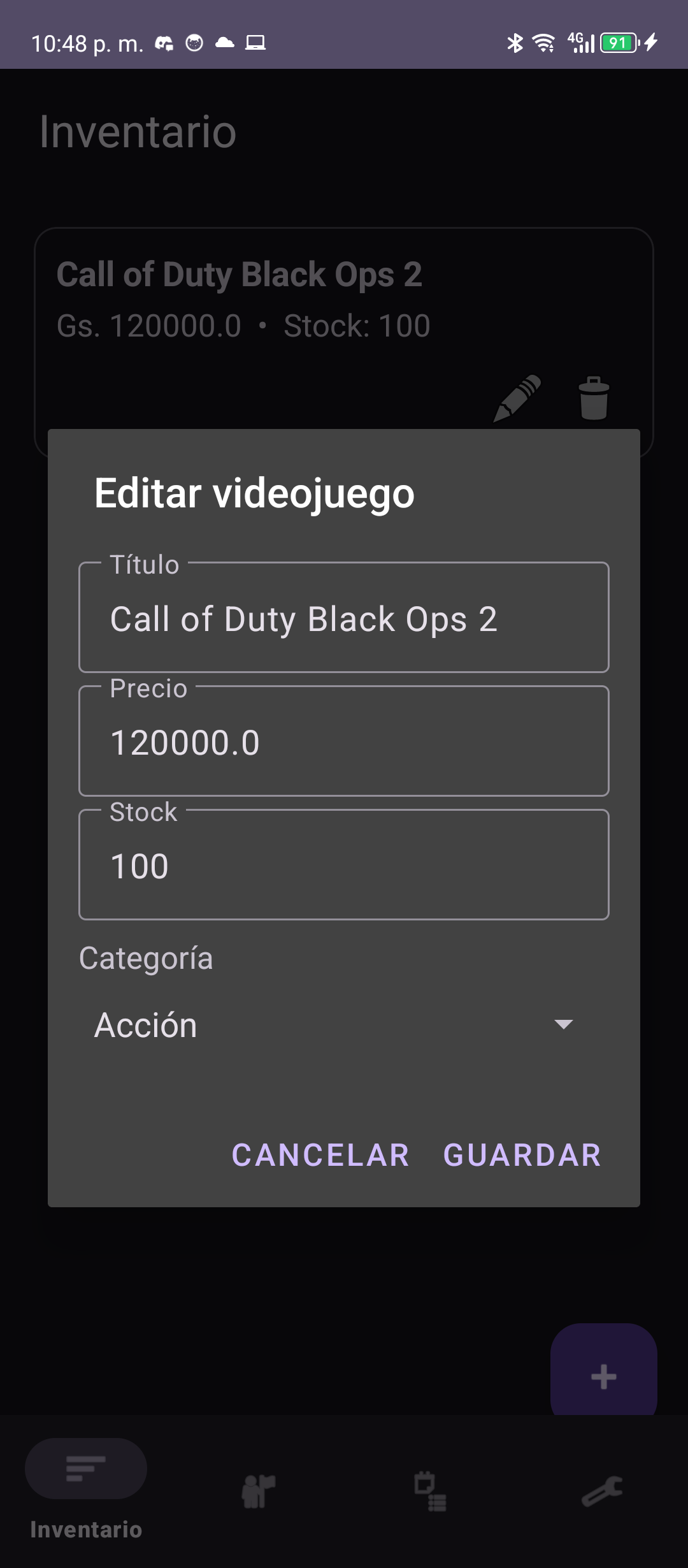


Figura 7. Screenshot\_20250929\_224837.png — Evidencia

Captura con los campos completados.

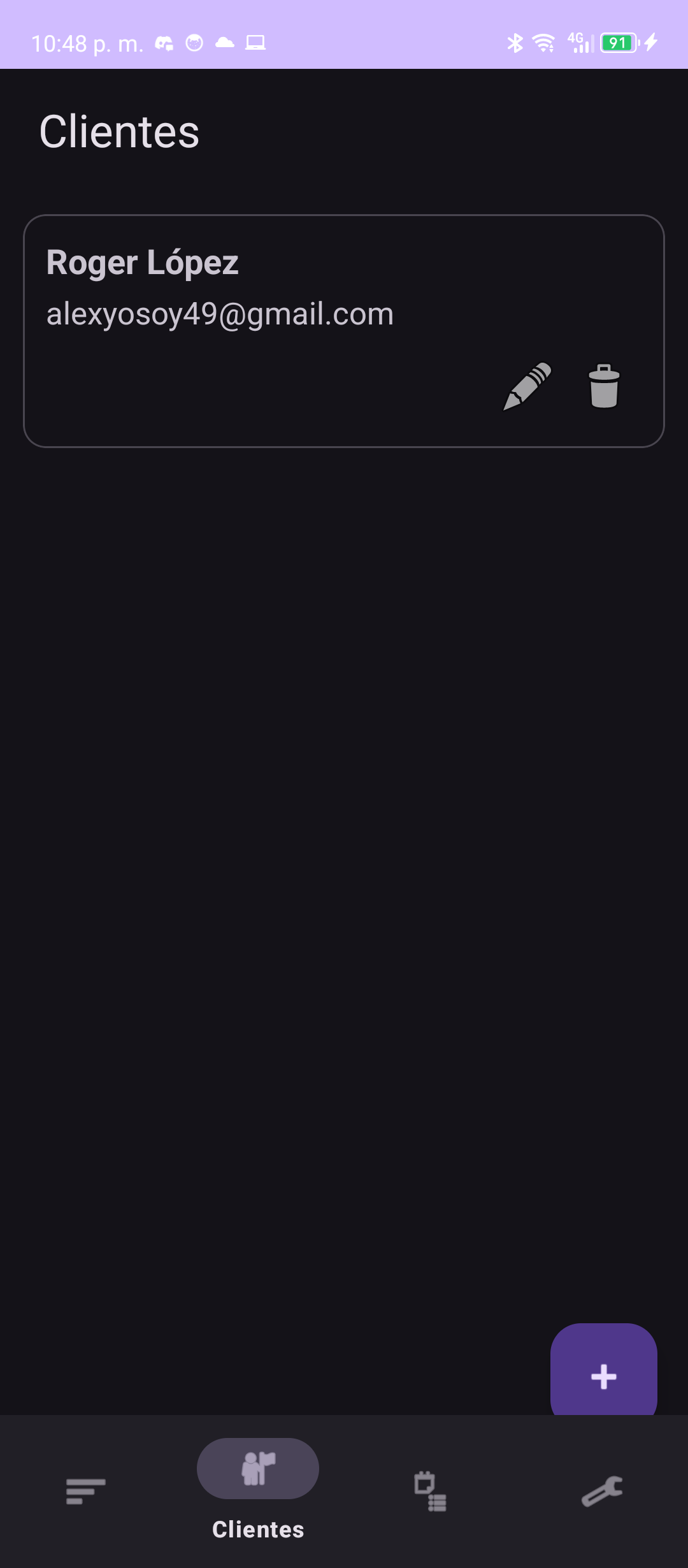


Figura 8. Screenshot\_20250929\_224858.png — Evidencia

Captura de la pantalla de clientes.

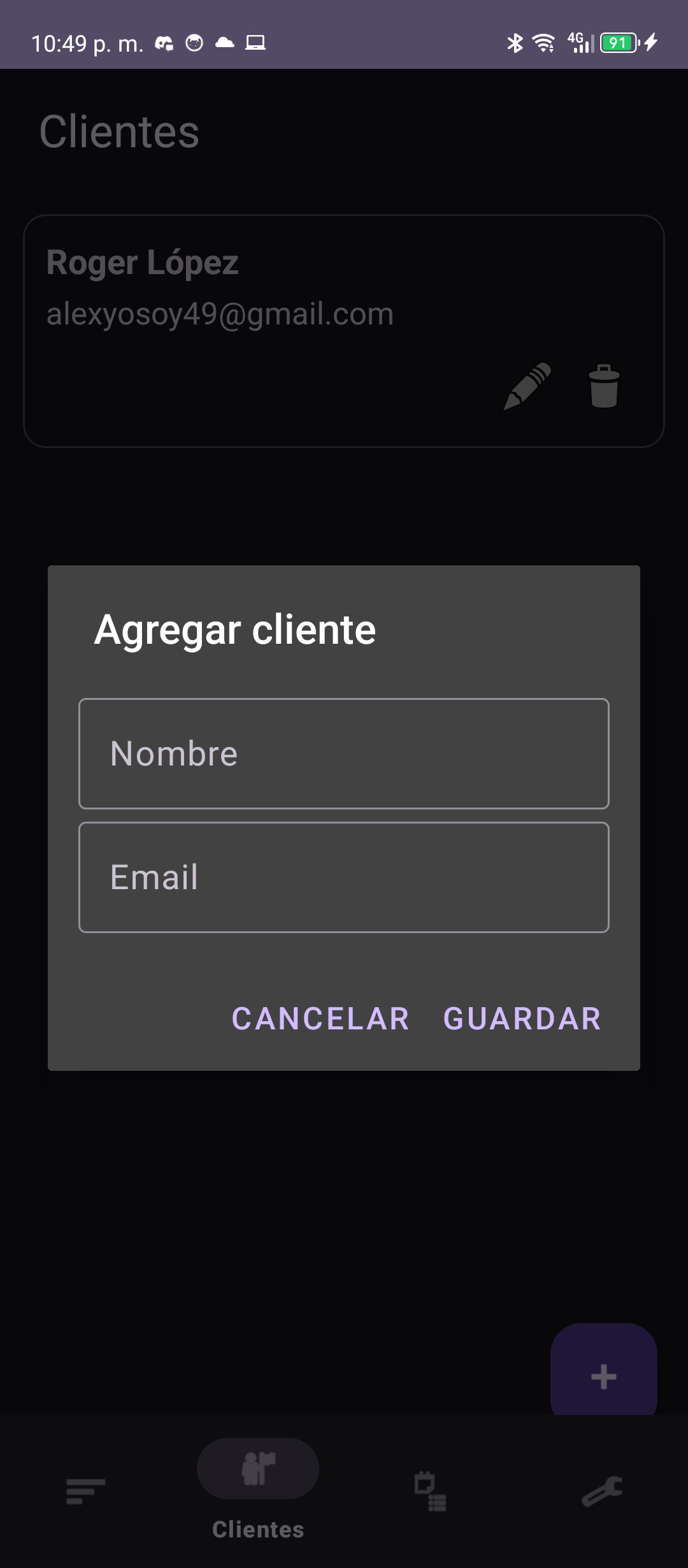


Figura 9. Screenshot\_20250929\_224915.png — Evidencia

Captura de los campos para crear un cliente.

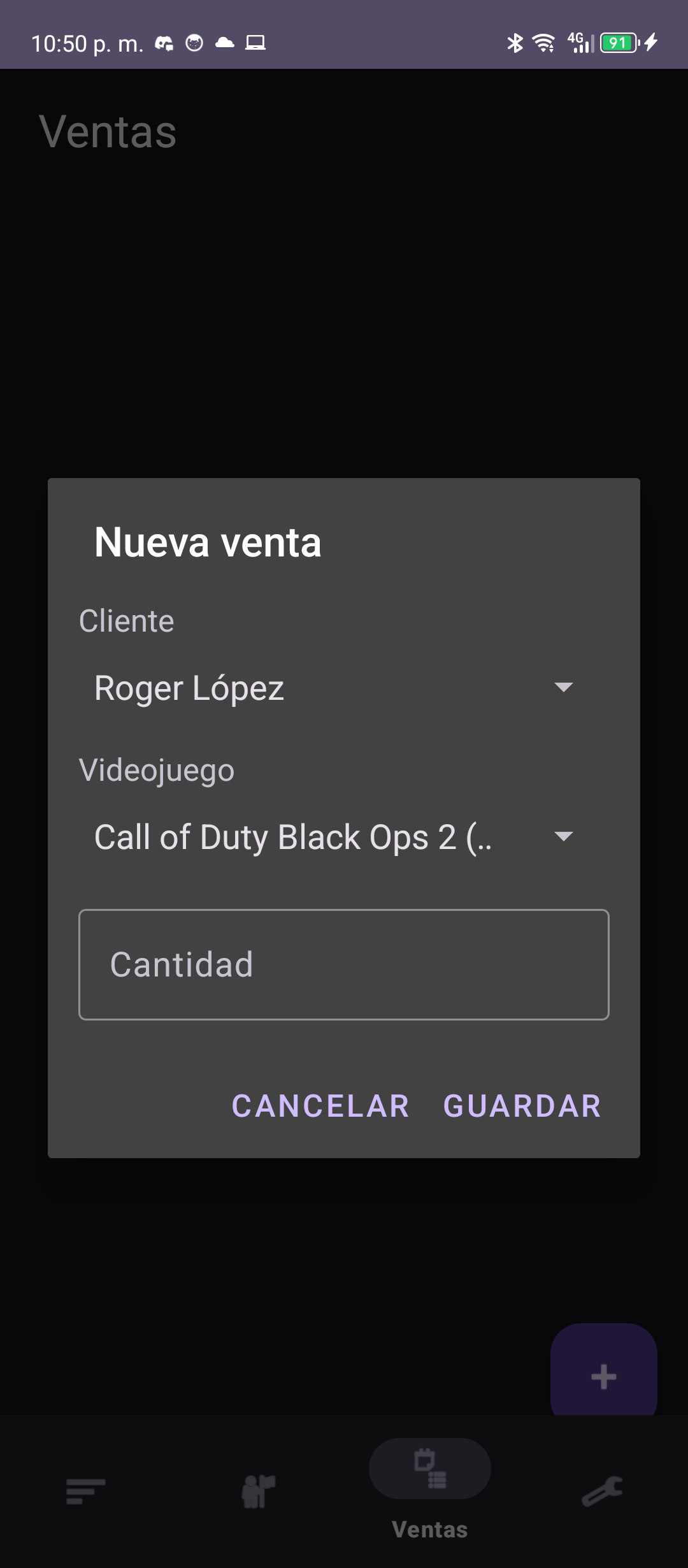


Figura 10. Screenshot\_20250929\_225001.png — Evidencia

Captura donde se añade una nueva venta.

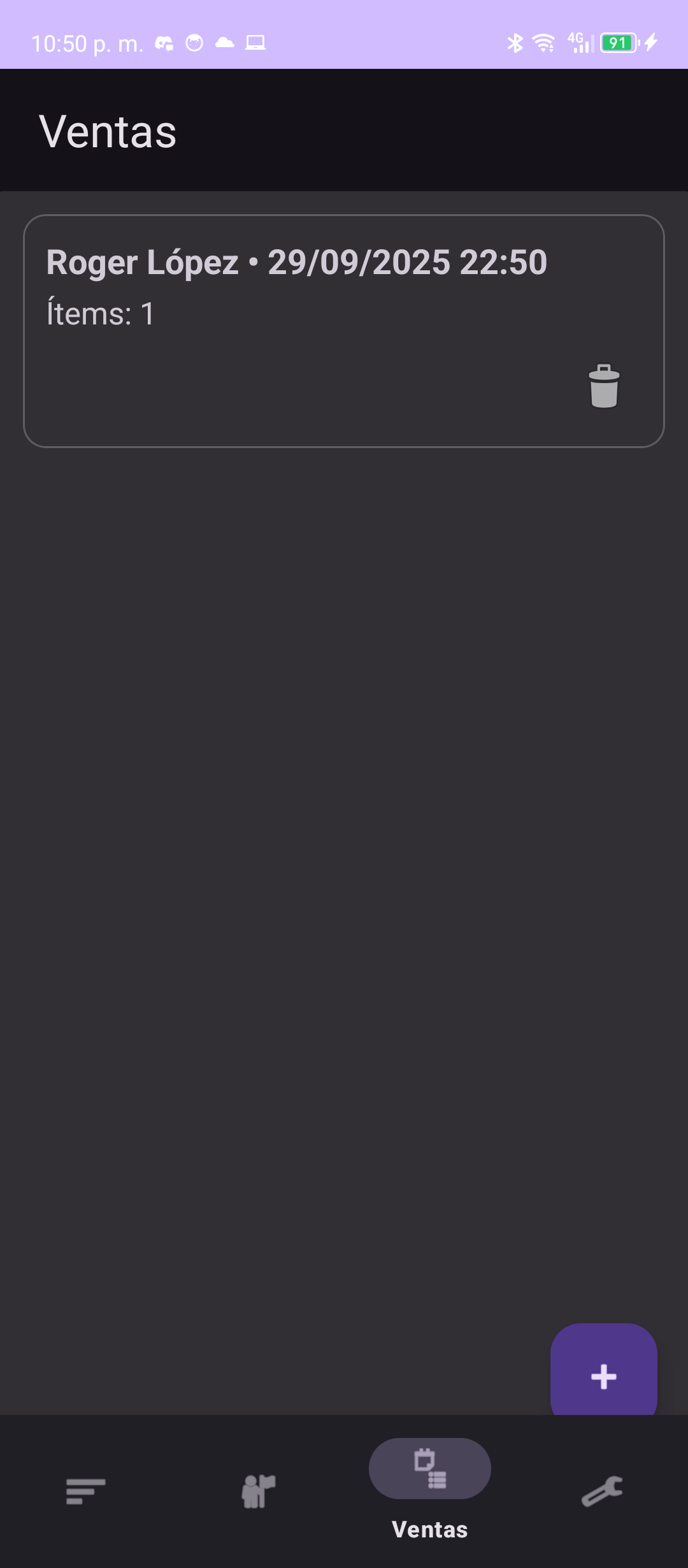


Figura 11. Screenshot\_20250929\_225021.png — Evidencia

Captura de la venta añadida.

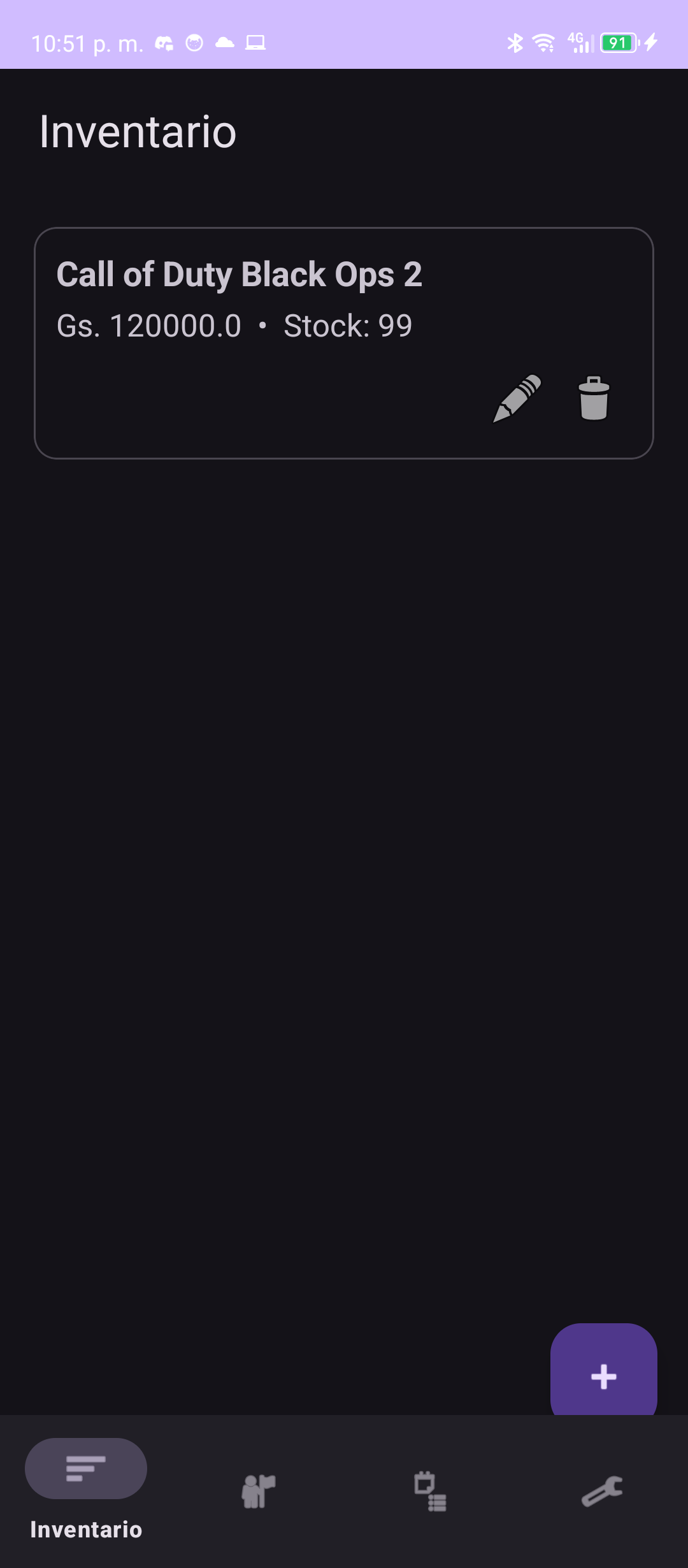


Figura 12. Screenshot\_20250929\_225138.png — Evidencia

Captura del stock luego de añadir una venta.

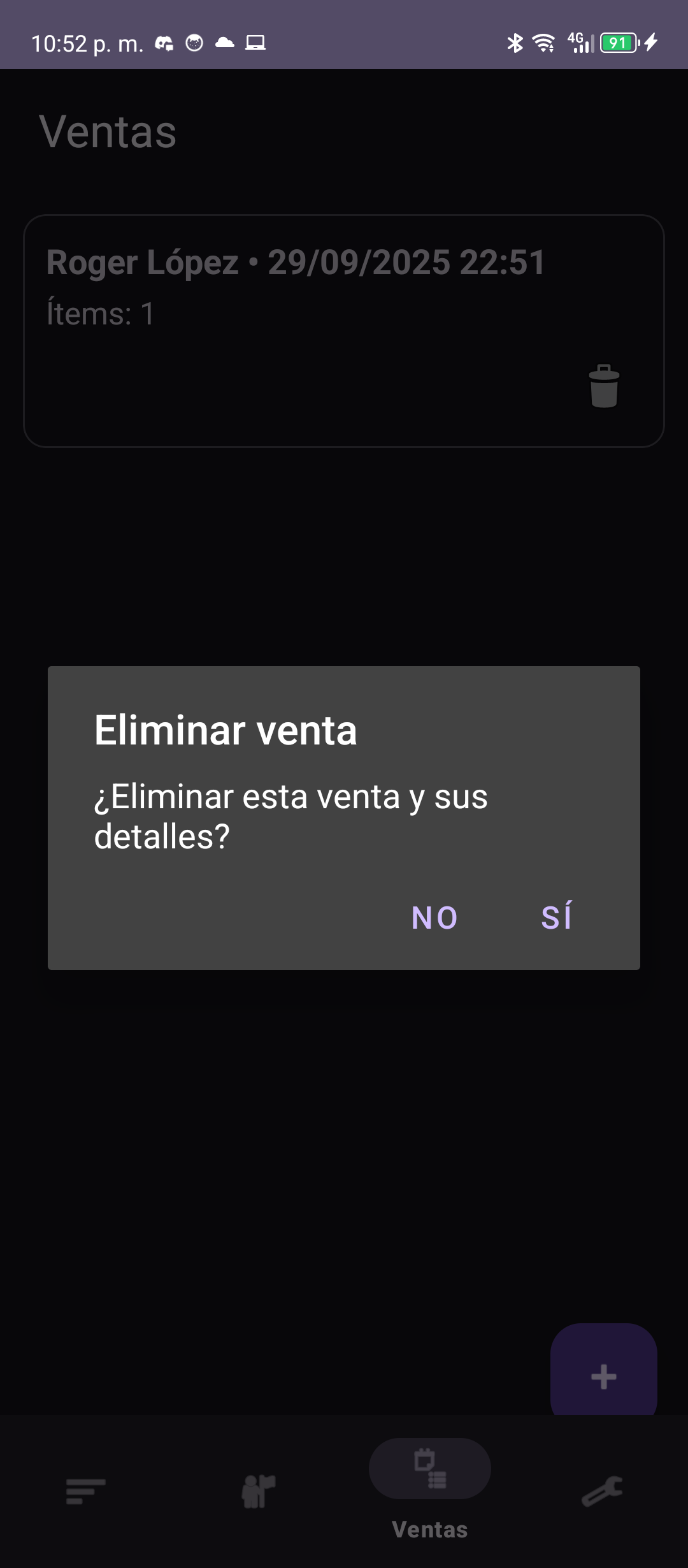


Figura 13. Screenshot\_20250929\_225206.png — Evidencia

Captura donde se elimina una venta.

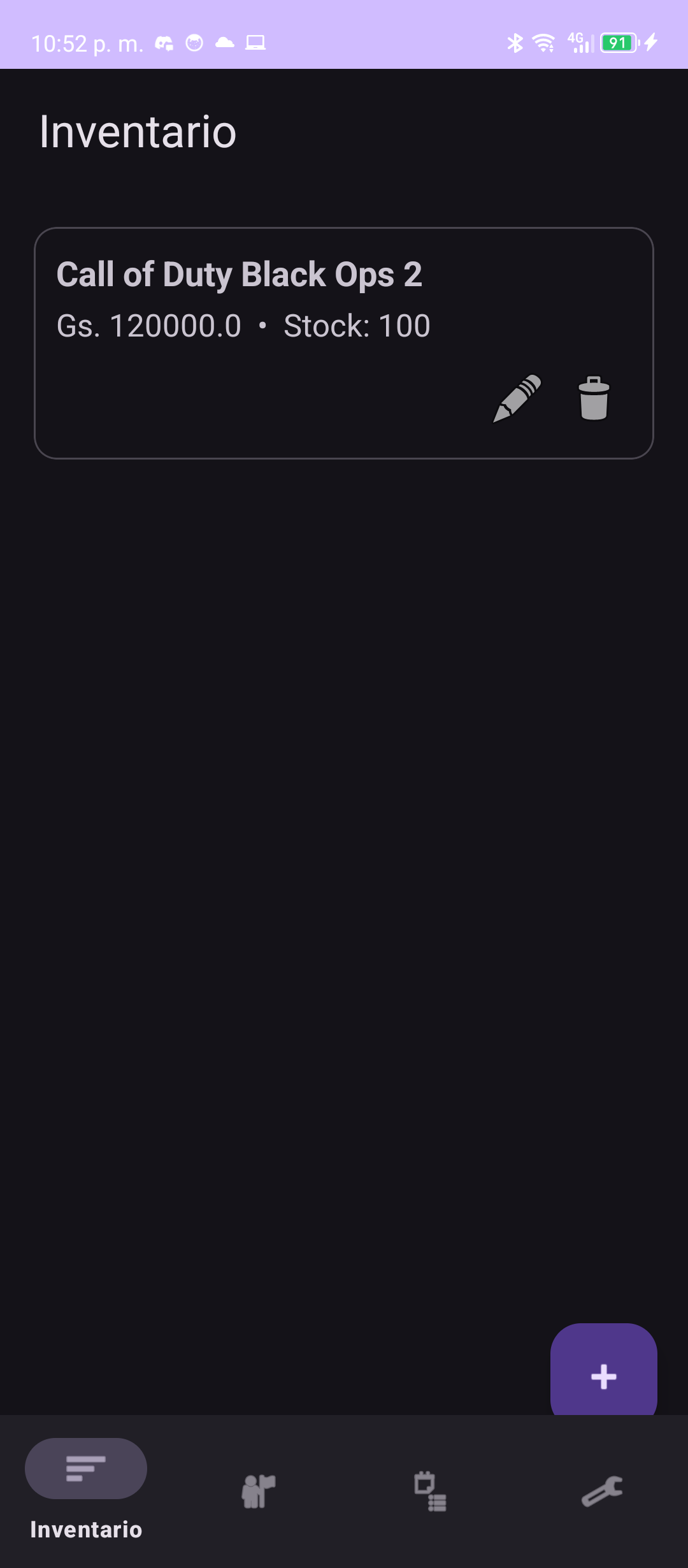


Figura 14. Screenshot\_20250929\_225230.png — Evidencia

Captura del stock luego de eliminar la venta.