

CodeLab: Persistencia de datos

40983 – Programación de Aplicaciones Móviles Nativas

Pablo Santana Susilla

Pr. Laboratorio 01.01

Profesor adjunto: Antonio Iván Hernández Fragiel

Curso: 2023/2024

Contenido

Introducción.....	3
Prueba de finalización del CodeLab	3
RUTA DE APRENDIZAJE 1: Introducción a SQL	4
RUTA DE APRENDIZAJE 2: Cómo usar Room para lograr la persistencia de datos	5
1. App de Inventory	5
Se añaden elementos a la base de datos.....	6
Se actualizan elementos de la base de datos	6
Se eliminan elementos de la base de datos.....	7
2. App de Bus Schedule.....	8
RUTA DE APRENDIZAJE 3: Cómo almacenar datos y acceder a ellos mediante claves con DataStore	10
Conclusión.....	11
Enlace a GitHub.....	11
Bibliografía	12

Introducción

Este informe proporciona una visión general de las aplicaciones desarrolladas durante el Codelab de la Unidad 6: "Persistencia de Datos", llevado a cabo en el entorno de desarrollo Android Studio utilizando el lenguaje Kotlin. A lo largo de este documento, se examinará las aplicaciones desarrolladas como parte de este codelab, el cual se enfoca en la persistencia de datos en aplicaciones móviles.

En este informe se presentarán ejemplos concretos de cómo aplicar los conocimientos adquiridos durante el codelab para crear y gestionar bases de datos, almacenar datos localmente en dispositivos móviles y mantener la funcionalidad de las aplicaciones en diversas condiciones de conectividad.

Prueba de finalización del CodeLab

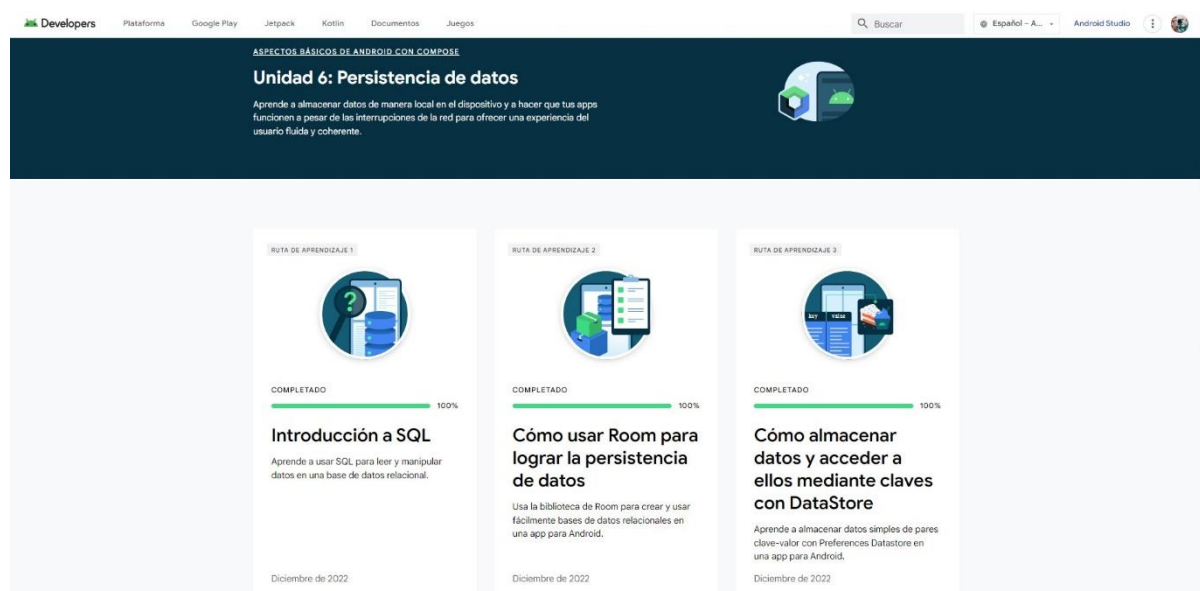


Figura 1. Demostración de la finalización del codelab

RUTA DE APRENDIZAJE 1: Introducción a SQL

En este apartado, se proporcionó una explicación clara y efectiva del lenguaje SQL para la lectura y manipulación de datos en la base de datos. Consideré que esta sección fue muy apropiada como introducción, ya que es útil tanto para repasar conceptos como para aprenderlos en caso de no tener experiencia previa con SQL.

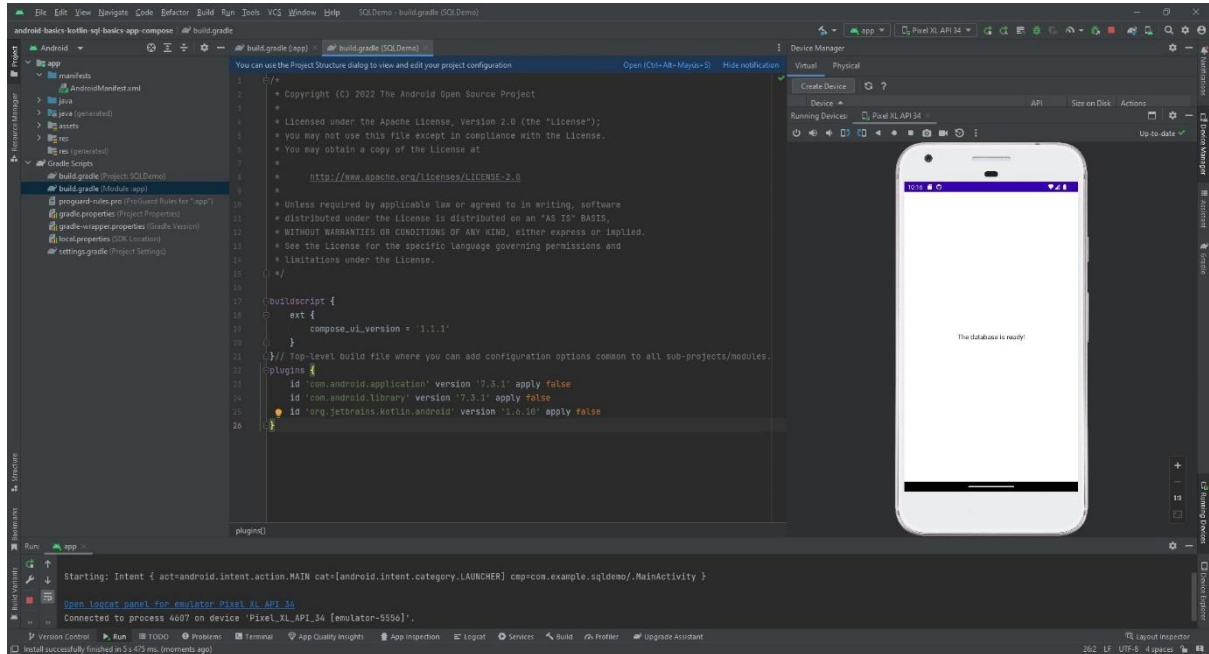


Figura 2. Emulación del proyecto inicial

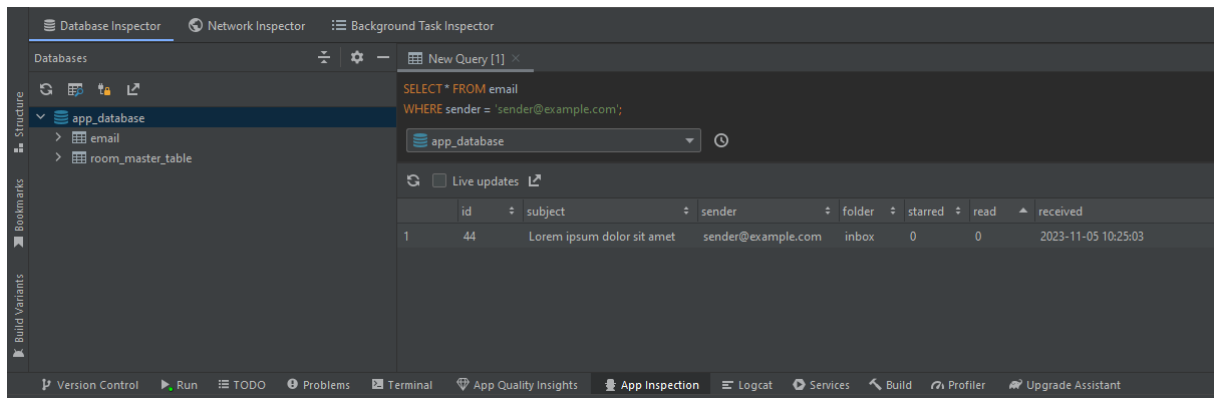


Figura 3. Inspector de base de datos del proyecto

RUTA DE APRENDIZAJE 2: Cómo usar Room para lograr la persistencia de datos

1. App de Inventory

Este apartado se centra en la utilización de la biblioteca Room con el objetivo de lograr la persistencia de datos en la App de Inventory. Iniciamos con el código inicial de la aplicación y luego se procedió a incorporar la capa de la base de datos usando Room.

El propósito final, es permitir a los usuarios acceder a una lista de elementos almacenados en la base de datos. Para lograrlo, se han implementado funcionalidades como añadir nuevos elementos, actualizar elementos y eliminar elementos de la base de datos.

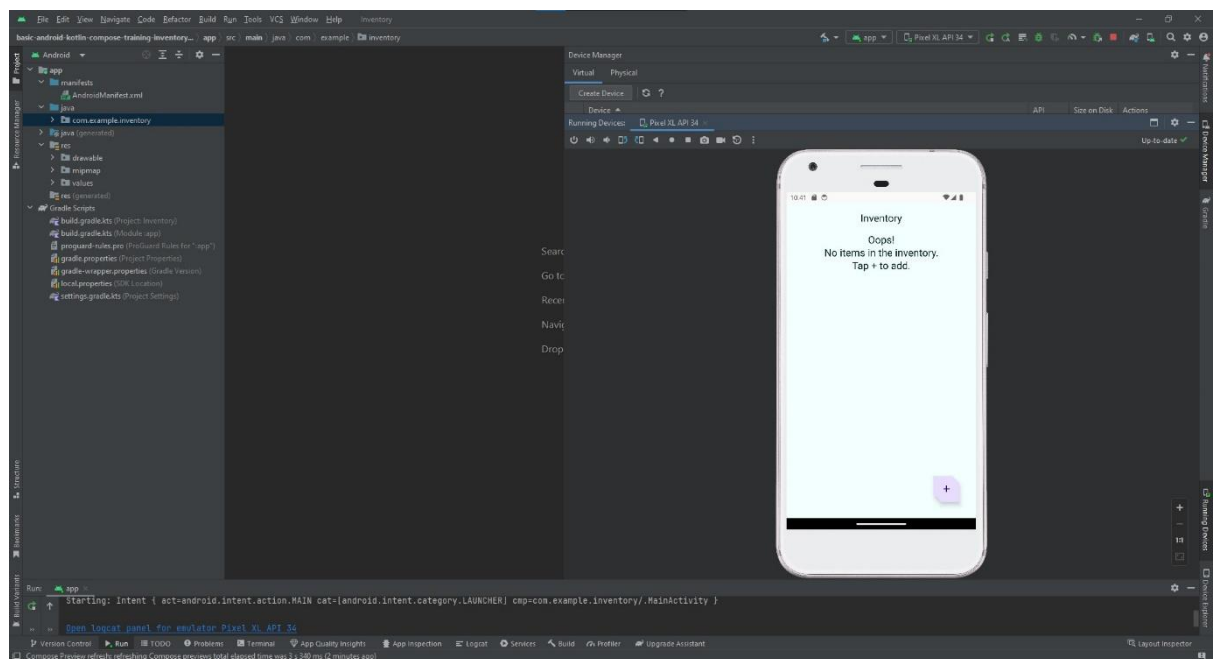


Figura 4: Pantalla inicial de App Inventory

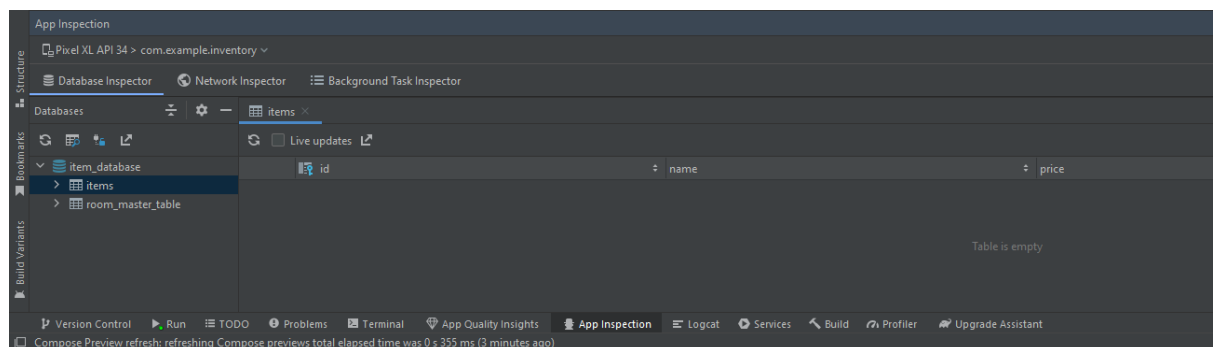


Figura 5: Base de datos

Se añaden elementos a la base de datos

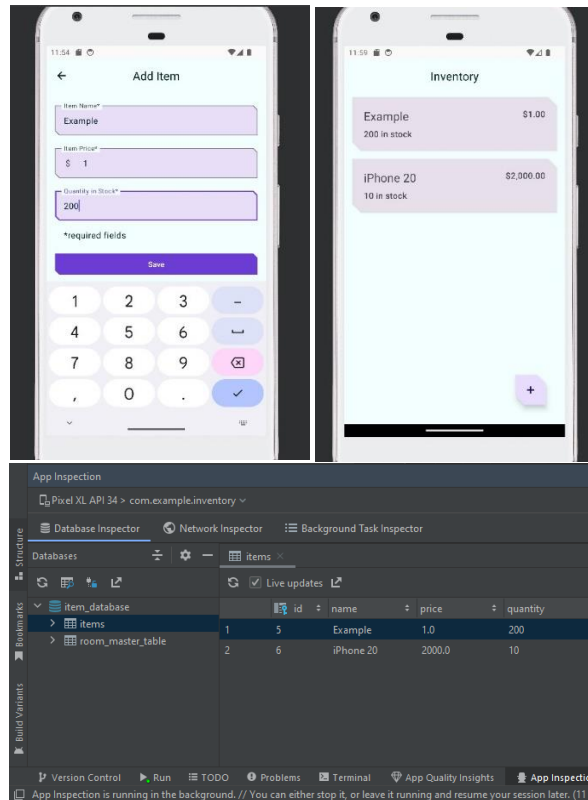


Figura 6: Se añaden elementos

Se actualizan elementos de la base de datos

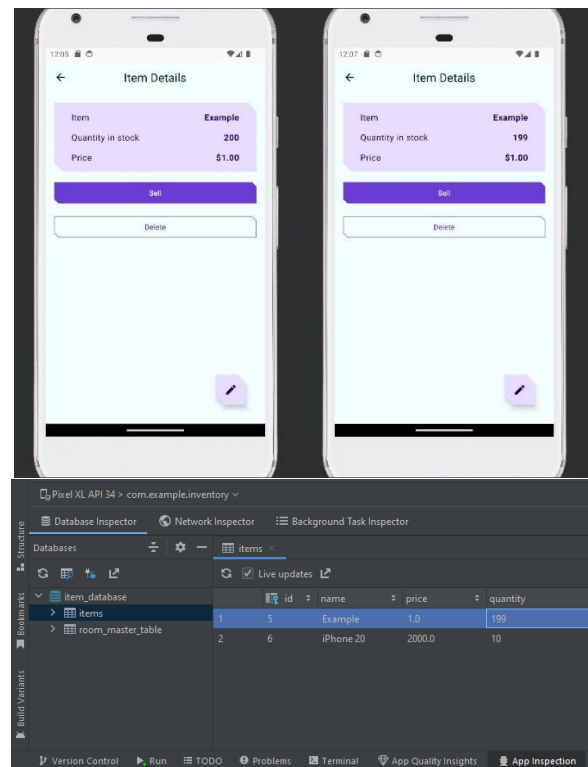


Figura 7: Se 'venden' elementos

Se eliminan elementos de la base de datos

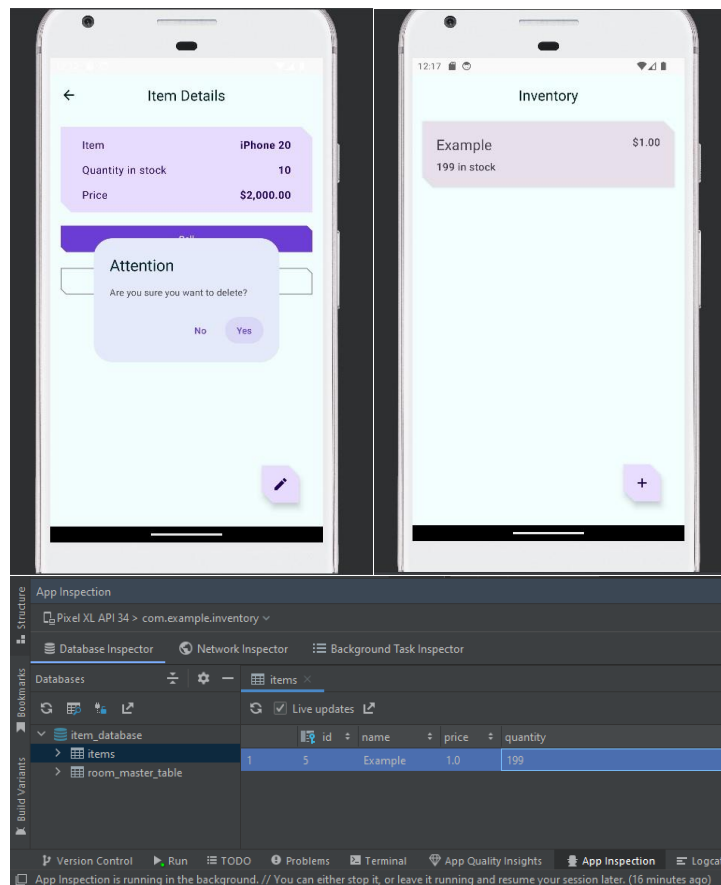


Figura 8: Se eliminan elementos

2. App de Bus Schedule

En este segundo apartado, se ha completado la aplicación de Bus Schedule. El proceso de desarrollo implicó crear una base de datos y la posterior integración de datos en la interfaz de usuario utilizando dicha base de datos. Para ello, se hizo uso de un archivo de base de datos proporcionado como material del codelab. Este suministró los datos necesarios para la aplicación tal y como se puede ver en las figuras XX y XX.

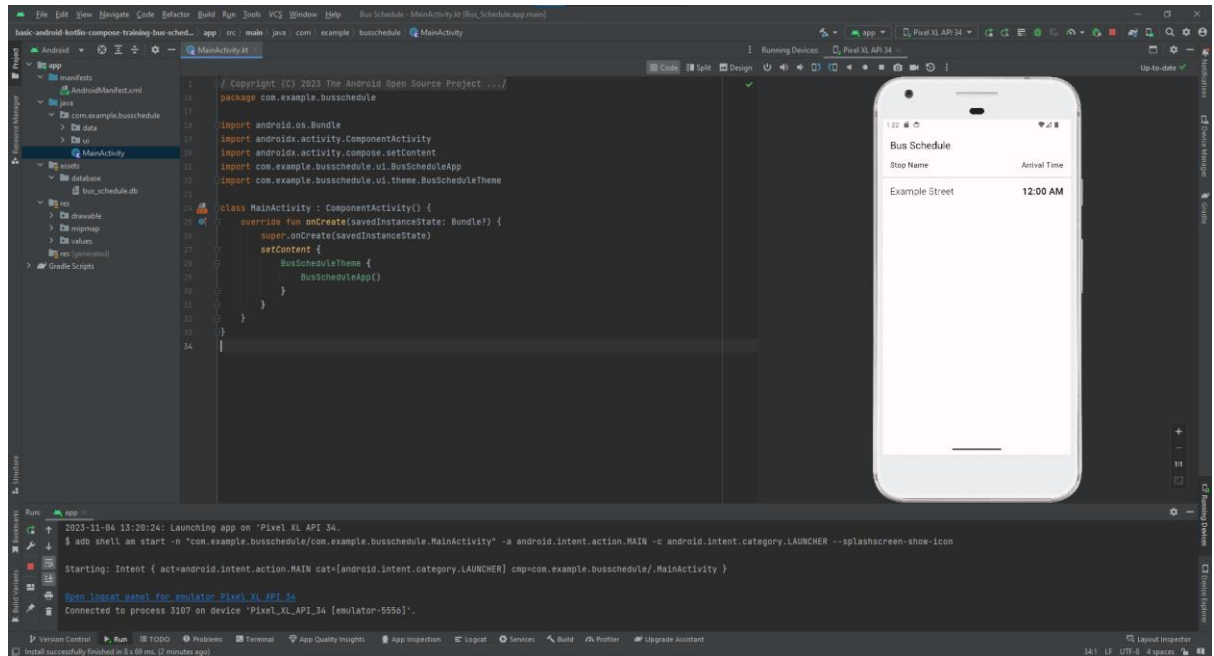


Figura 9: Punto de partida de la App de Bus Schedule

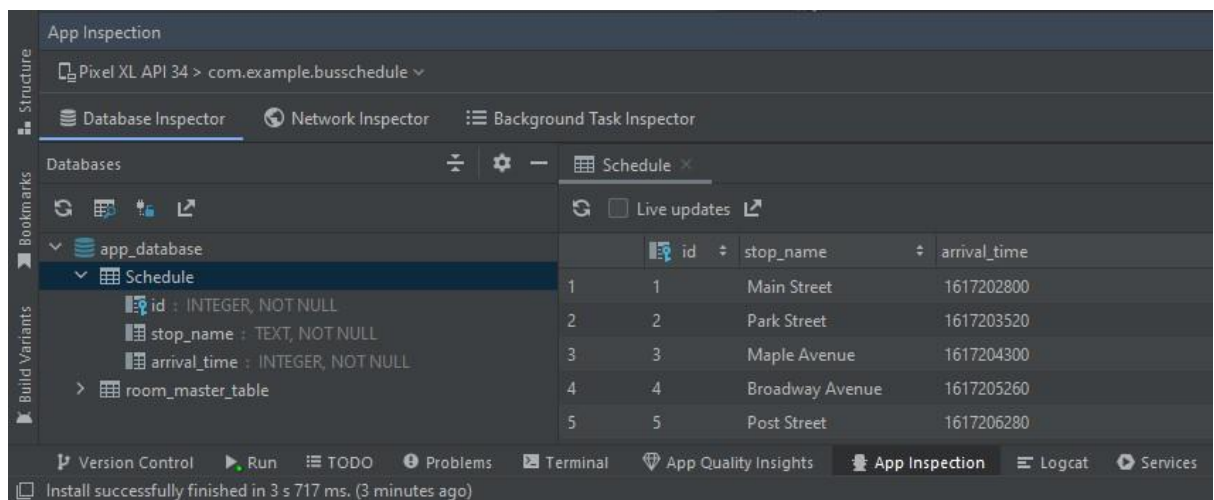


Figura 10: Base de datos de la App de Bus Schedule

Tras seguir las indicaciones del codelab, la aplicación ahora presenta una lista de paradas de guagua con sus respectivos horarios (figura XX). Al hacer 'clic' en una parada, es posible ver información específica sobre dicha parada.

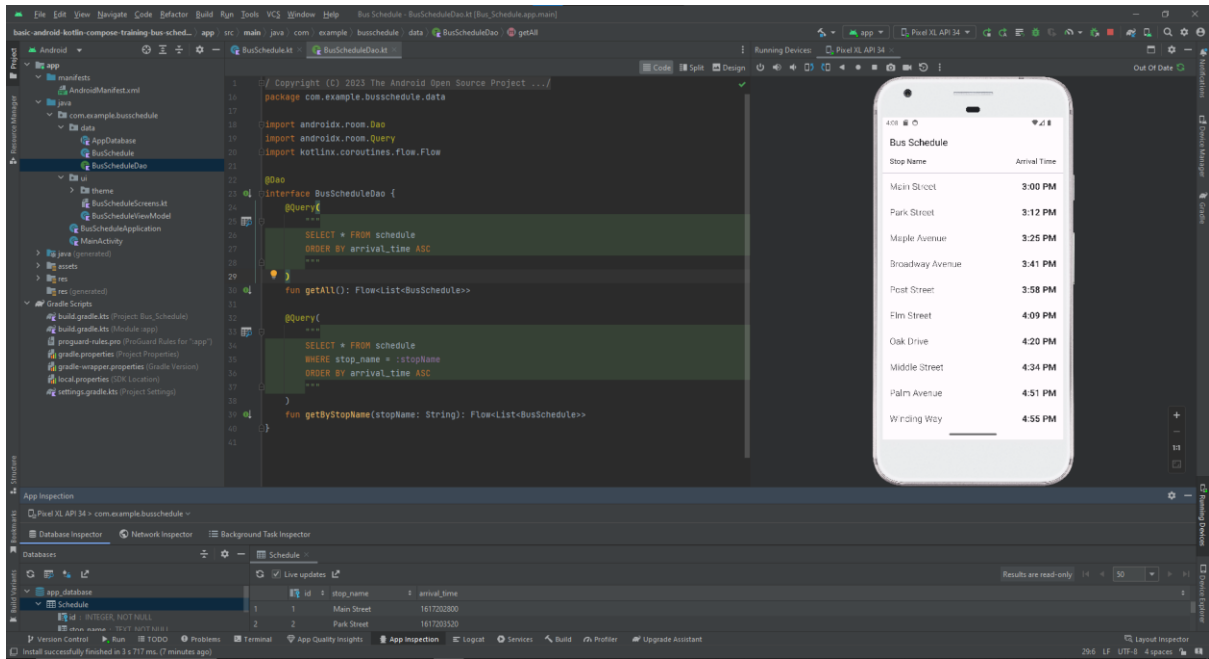


Figura 11. Resultado final de la app de Bus Schedule

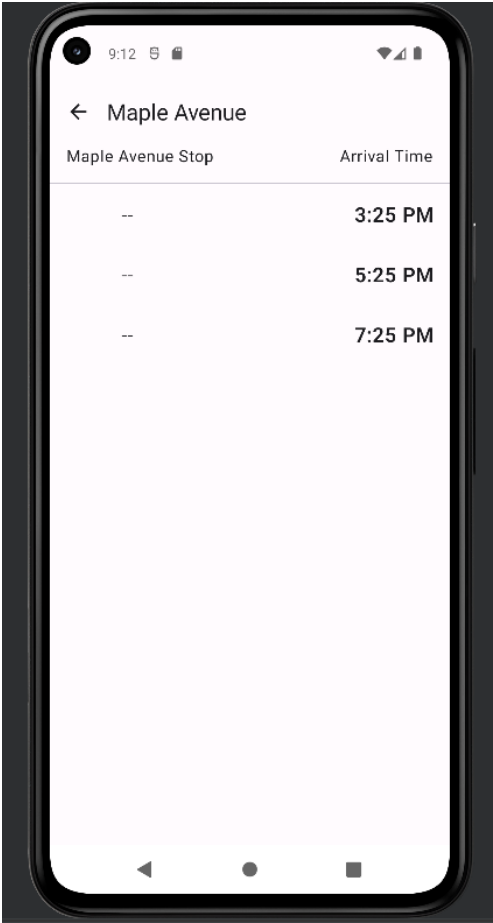


Figura 12. Pantalla al elegir una ruta

RUTA DE APRENDIZAJE 3: Cómo almacenar datos y acceder a ellos mediante claves con DataStore

En este apartado, se ha mejorado la funcionalidad de la aplicación Dessert Release. Inicialmente, la aplicación muestra una lista con las versiones de Android y permitía al usuario cambiar la vista de esta. Sin embargo, no podía conservar la selección de diseño realizada por el usuario.

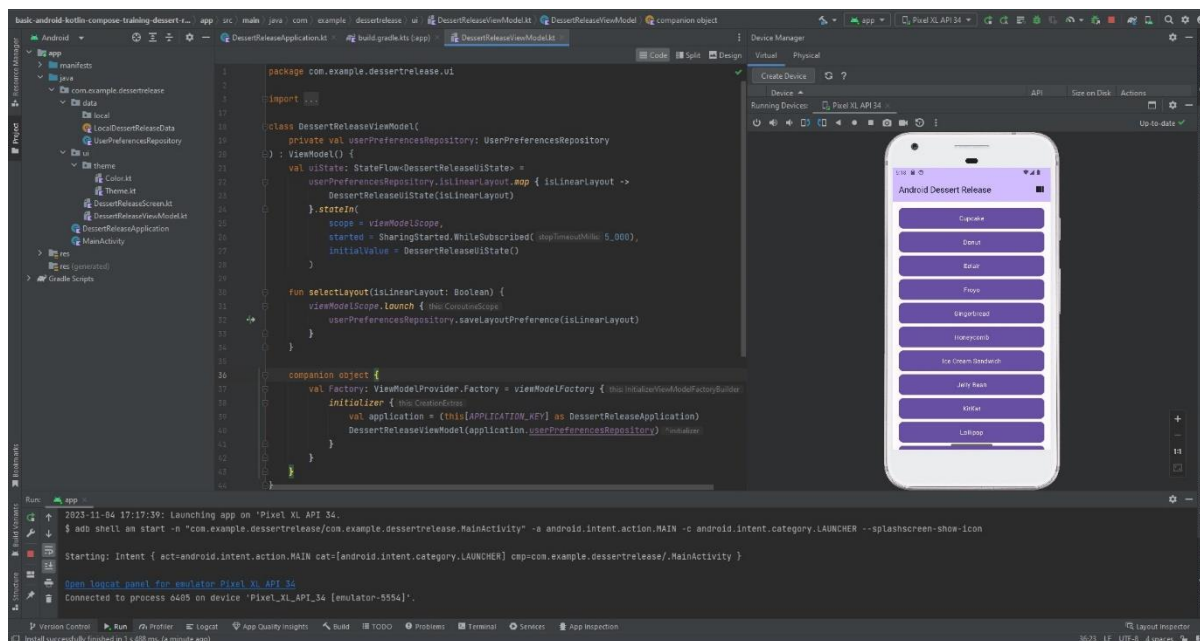


Figura 13: Vista de la lista de versiones Android

En este apartado, se implementó DataStore para solucionar este problema. DataStore se ha utilizado para almacenar la preferencia de selección de diseño, lo que significa que ahora, cuando los usuarios elijan una vista específica, esta preferencia se guarda y persiste incluso después de cerrar la aplicación.

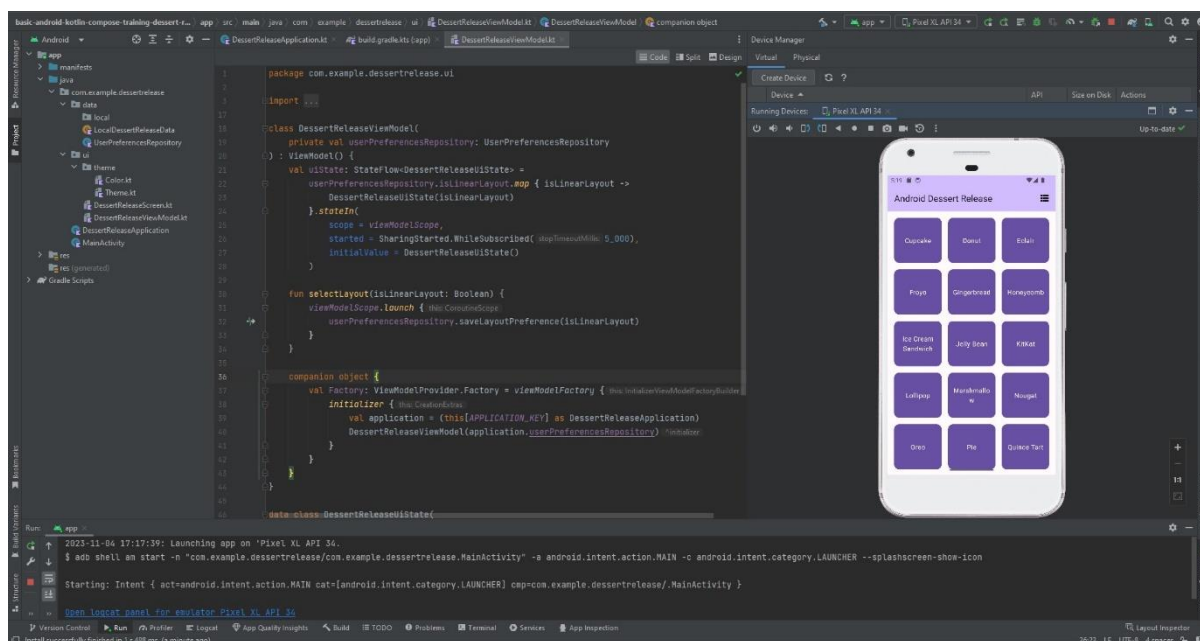


Figura 14: Vista de la cuadrícula de las versiones Android

Conclusión

La realización de este codelab de persistencia de datos en Kotlin ha sido una experiencia educativa positiva. Uno de los puntos que quisiera destacar es el repaso al lenguaje SQL, ya que, esta parte me ayudó a refrescar conceptos y la sintaxis utilizada para manejarlo. La explicación detallada y los ejercicios prácticos relacionados con SQL fueron claros y facilitaron en gran medida la comprensión de los conceptos abordados.

Como punto negativo, me gustaría recalcar el esfuerzo que ha tenido que realizar mi ordenador portátil para ejecutar y poder leer las bases de datos. En muchas ocasiones, tardaba bastante tiempo en cargarlas y tuve algunas dificultades para poder sacar las figuras adjuntas en este informe.

Enlace a GitHub

<https://github.com/pablosanttanaa/PAMN.git>

Bibliografía

[1] Google for Developers, "Unidad 6: Persistencia de datos," [en línea]. Disponible en: <https://developer.android.com/courses/android-basics-compose/unit-6?hl=es-419>. Accedido en: 26/10/23.