

CodeLab: Persistencia de datos



Introducción

Este informe ofrece un vistazo a las aplicaciones resultantes del Codelab de la Unidad 6: "Persistencia de Datos" realizado en el entorno de desarrollo Android Studio con el lenguaje Kotlin. A lo largo de este documento, exploraremos las aplicaciones desarrolladas como parte de este codelab, que se centra en la persistencia de datos en aplicaciones móviles. La persistencia de datos es un componente esencial para lograr una experiencia de usuario fluida y coherente, incluso en situaciones de interrupción de la red.

Este informe presentará ejemplos concretos de cómo hemos aplicado 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.

Ruta 1: Introducción a SQL

Durante este apartado se explicó de forma sencilla y eficaz el lenguaje SQL para leer y manipular datos en la base de datos. Me pareció muy adecuada esta sección como primera toma de contacto, debido a que sirve tanto para refrescar conceptos como para aprenderlos en caso de no haber trabajado antes con SQL.

En concreto, se ha descargado un proyecto con un "conjunto de datos inicial" en Android Studio con el objetivo de obtener experiencia con sentencias SQL y el inspector de bases de datos.



Ilustración 1: Emulación proyecto "Conjunto de datos inicial"

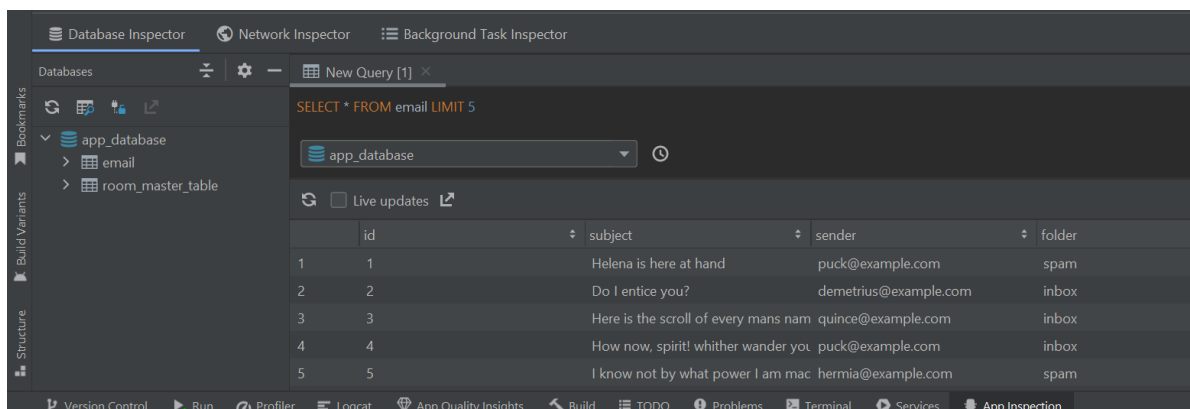


Ilustración 2: Inspector de bases de datos en proyecto "Conjunto de datos inicial"

Ruta 2: Cómo usar Room para lograr la persistencia de datos

1. App Inventory

Este apartado se enfoca en el uso de la biblioteca Room para lograr la persistencia de datos en la aplicación Inventory. Comenzamos con el código base de la aplicación y procedimos a agregar la capa de la base de datos utilizando Room.

El objetivo final es permitir a los usuarios acceder a una lista de elementos almacenados en la base de datos. Para lograr esto, hemos trabajado en funcionalidades como agregar nuevos elementos, actualizar elementos existentes y eliminar elementos de la base de datos.

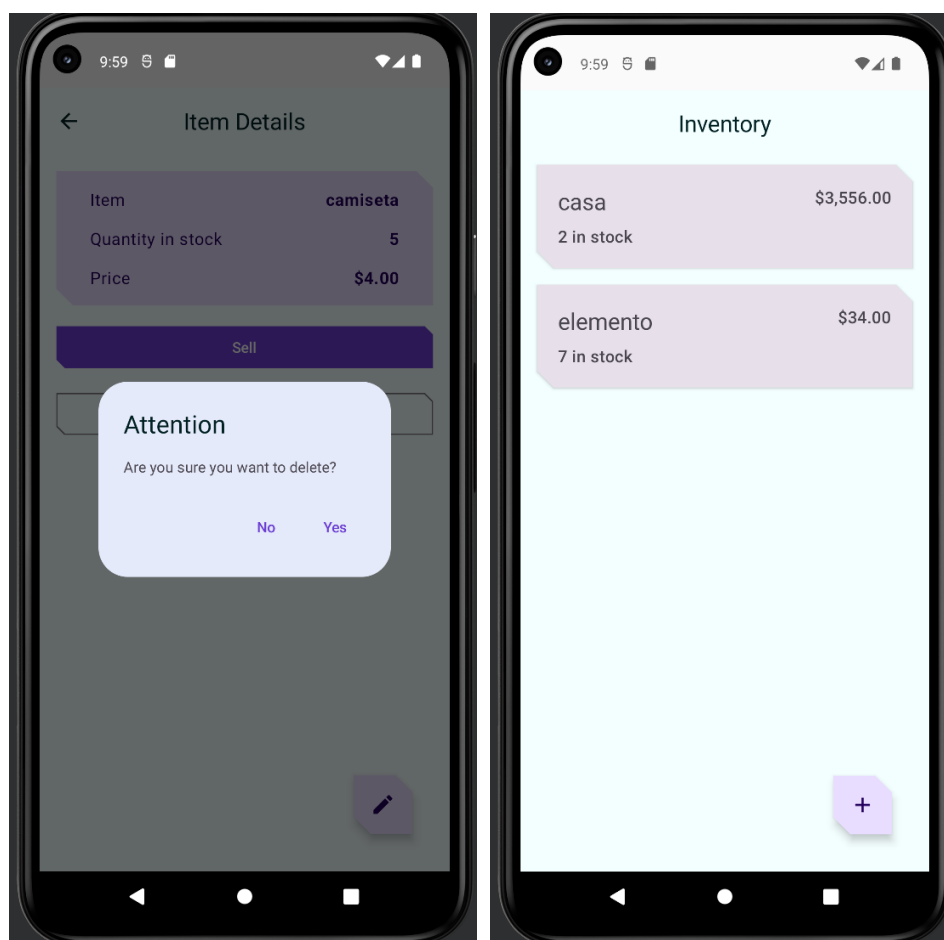


Ilustración 6 y 7: app Inventory (Pantalla Item Details eliminando Item y Pantalla principal)

2. App Bus Schedule

A continuación, se ha llevado a cabo la finalización de la aplicación Bus Schedule. El proceso de desarrollo incluyó la implementación de una base de datos y la posterior integración de datos en la interfaz de usuario utilizando esta base de datos. Para este propósito, se utilizó un archivo de base de datos ubicado en el directorio de recursos, el cual proporcionaba los datos necesarios para la aplicación. Estos datos se cargaron en una base de datos, permitiendo así su accesibilidad desde la aplicación.

Como resultado, la aplicación ahora muestra una lista de paradas de autobús junto con las horas de llegada correspondientes. Los usuarios tienen la capacidad de seleccionar un elemento de la lista, lo que los dirige a una pantalla de detalles donde pueden acceder a información específica sobre esa parada.

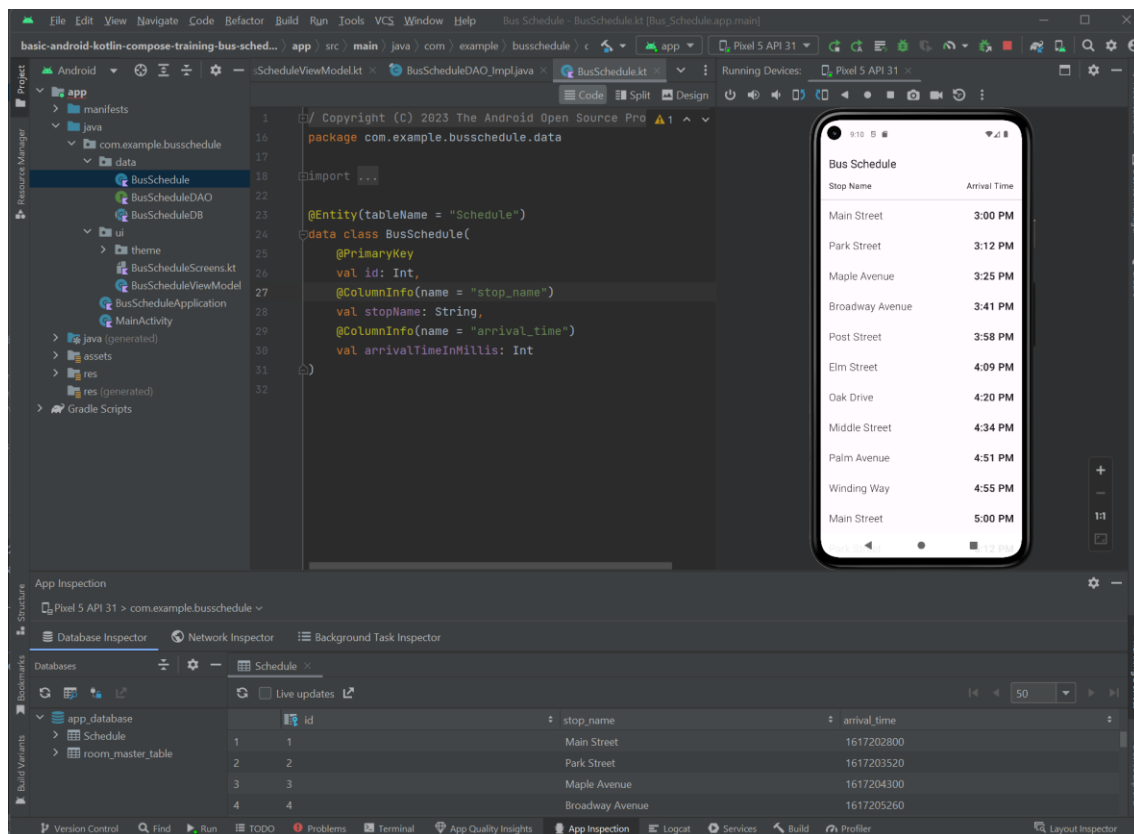


Ilustración 8: app Bus Schedule en Android Studio (Pantalla principal)

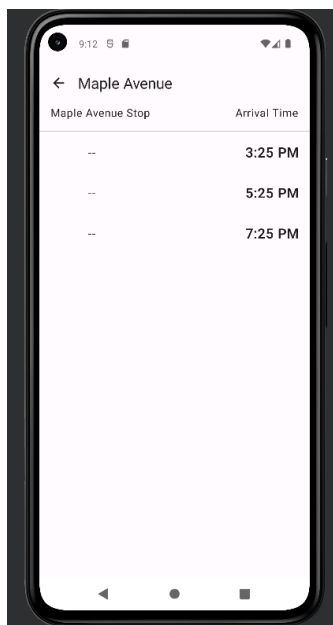


Ilustración 9: app Bus Schedule (Pantalla ruta)

Ruta 3: Cómo almacenar datos y acceder a ellos mediante claves con DataStore

En este apartado, se ha trabajado en la mejora de la funcionalidad existente de la aplicación Dessert Release. La aplicación inicialmente muestra una lista de versiones de Android y permite a los usuarios alternar entre una vista de cuadrícula y una vista de lista mediante un ícono en la barra de la aplicación.

Sin embargo, en su estado anterior, la aplicación carecía de la capacidad de conservar la selección de diseño realizada por el usuario. Al cerrar la aplicación, la configuración se reiniciaba automáticamente a la selección predeterminada.

En este proyecto, se ha implementado DataStore para abordar esta limitación. 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.

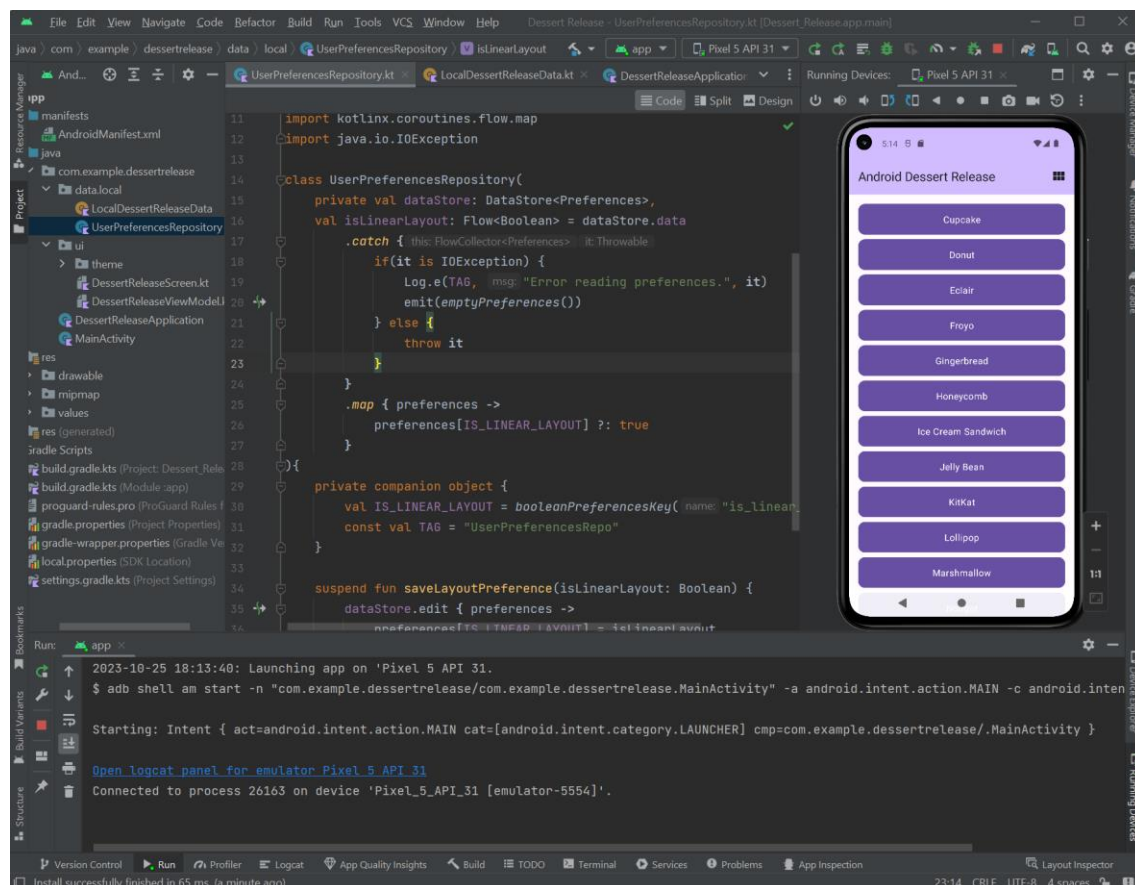


Ilustración 9: app Dessert Release en AndroidStudio (Vista de Lista)

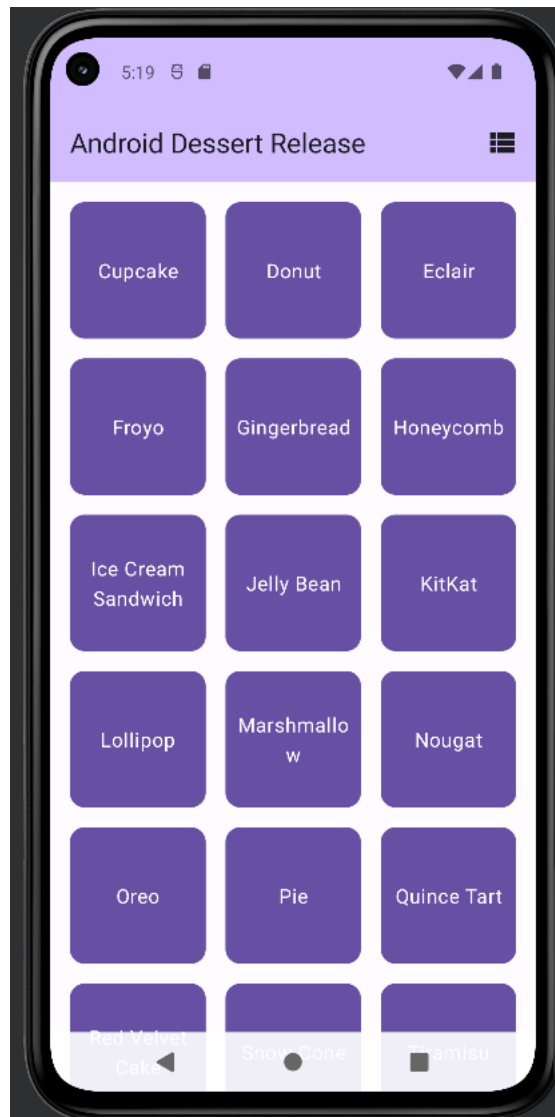


Ilustración 10: app Dessert Release (Vista de Cuadrícula)

Opinión personal sobre el CodeLab

El codelab de persistencia de datos en Kotlin ha sido una experiencia educativa positiva, aunque considero que presenta algunos aspectos que podrían mejorarse, especialmente para personas que están dando sus primeros pasos en Kotlin.

Uno de los aspectos más destacados del codelab es la revisión del lenguaje SQL. Esta parte fue especialmente beneficiosa, ya que ayudó mucho a refrescar conceptos y sintaxis. La explicación detallada y los ejercicios prácticos relacionados con SQL fueron muy claros, y ayudaron a comprender conceptos clave.

Sin embargo, mi principal crítica radica en el nivel de complejidad de Kotlin que se abordó en este codelab. Desde mi perspectiva, el codelab parecía estar orientado a un nivel más avanzado de conocimientos en el lenguaje. A pesar de que se proporcionaron explicaciones y ejemplos, encontré que algunos de los conceptos y la implementación de código eran un desafío para alguien que está en las etapas iniciales de aprender el lenguaje. Hubiera sido beneficioso tener una introducción más gradual a Kotlin antes de abordar este proyecto.

Bibliografía

Unidad 6: Persistencia de datos. Google for Developers.

<https://developer.android.com/courses/android-basics-compose/unit-6?hl=es-419> [en línea]
26/10/23