

**Reto 4** – Mi primera aplicación móvil

**Nombre completo: Frausto Becerra Juan de Dios**

**Fecha de elaboración: 03/11/2023**

**Nombre del Asesor: Ignacio de Jesús Segovia Domínguez**

**Introducción**

# El propósito de este Reto es que desarrolles tu primera aplicación móvil en *Android Studio,* a partir de los conocimientos adquiridos en las lecciones correspondientes a este Reto.

# **Instrucciones:**

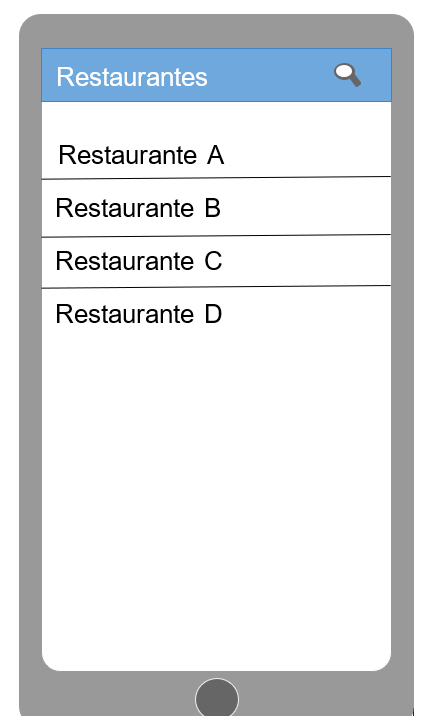
Desarrolla una aplicación que muestre los alimentos que vende cada uno de los restaurantes en la zona centro de la Ciudad de México, de tal forma que los visitantes conozcan las opciones que pueden consumir a su alrededor.

Lee cuidadosamente las dos partes que integran este Reto y realiza lo que se te indica:

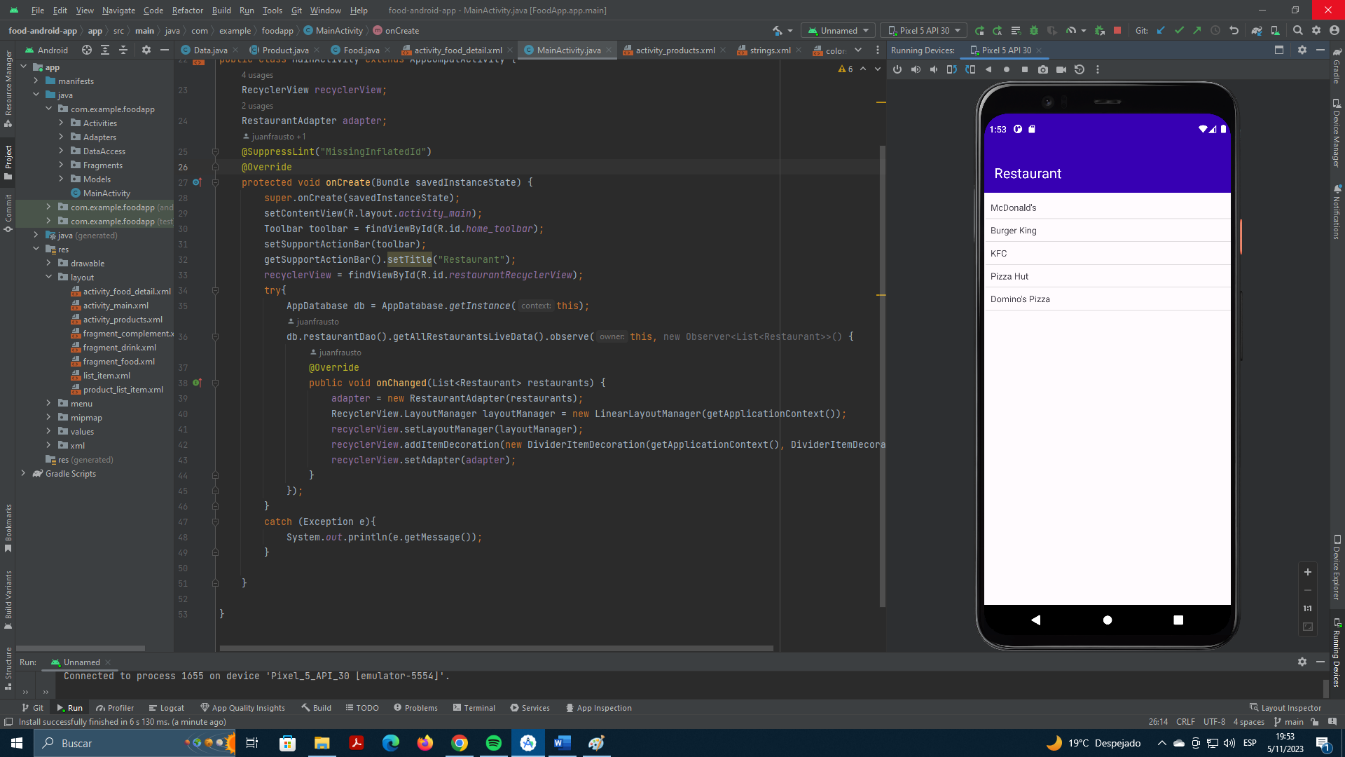
**Parte 1: Estructura y flujo de la aplicación:**

La aplicación que desarrollaremos debe contener los siguientes puntos:

1. Una lista que muestre el listado con los nombres de los restaurantes.



En la imagen se muestra la lista de los Restaurantes, en este caso son Mc Donalds, Burger King, KFC, Pizza Hut y Domino’s Pizza.

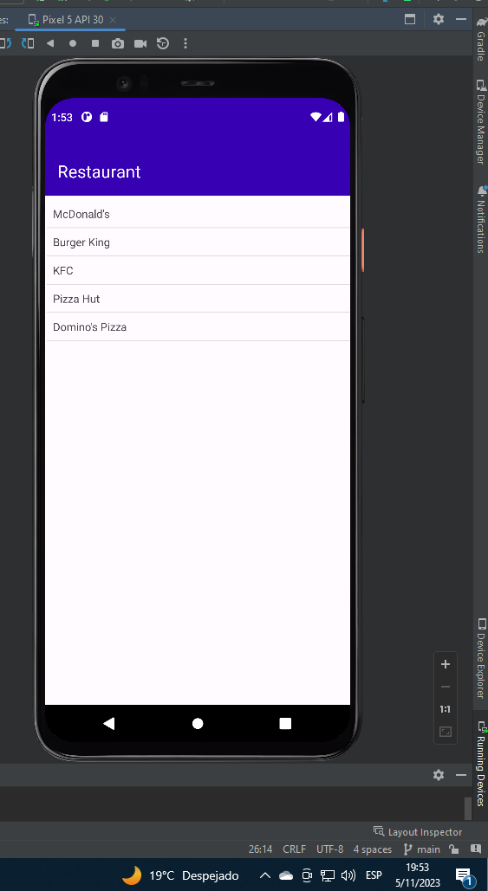


1. Cuando se presione sobre el nombre de alguno de los restaurantes listados deberá ir a un nuevo activity, el cual contendrá tres pestañas y cada una mostrará los siguientes elementos:
   1. **Pestaña comida:** Listado de los nombres y precios de cada alimento que puede consumir el usuario.
   2. **Pestaña bebidas:** Listado con los nombres y precios de las bebidas que puede consumir el usuario.
   3. **Pestaña complementos:** Listado con los nombres y precios de los complementos que puede consumir el usuario.

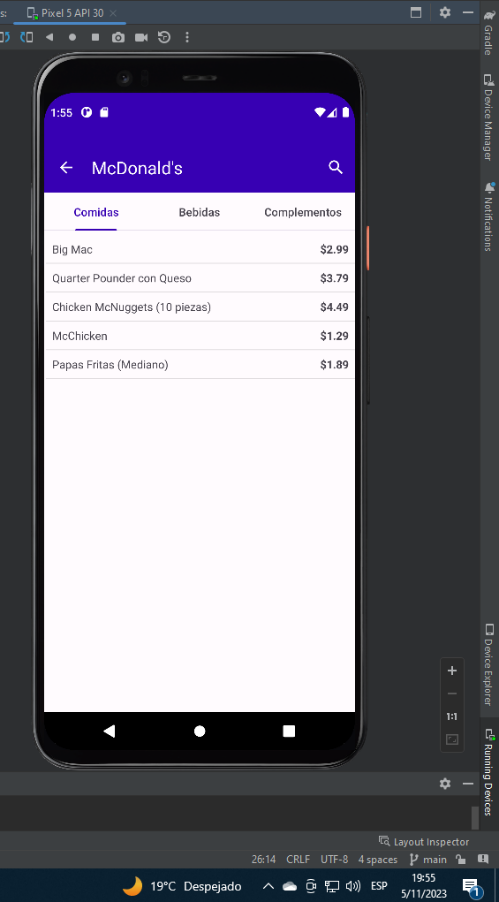
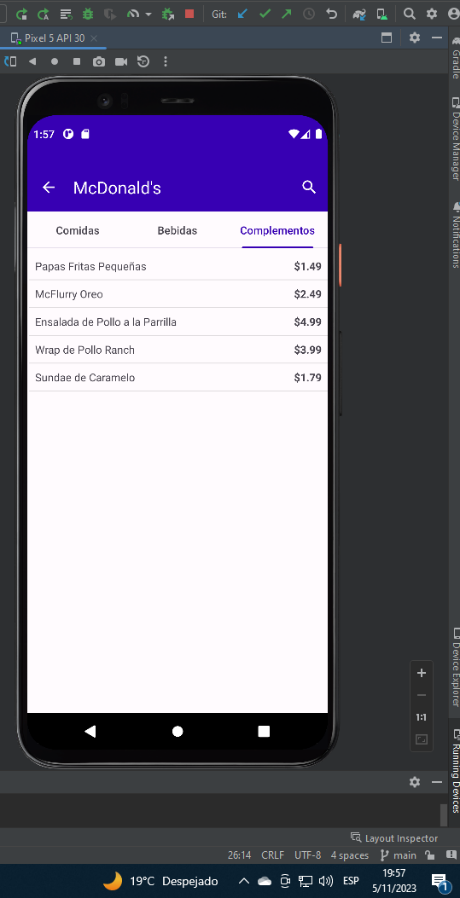


En las imágenes se muestra las pestañas comida, bebidas y complementos

Pestaña Bebidas



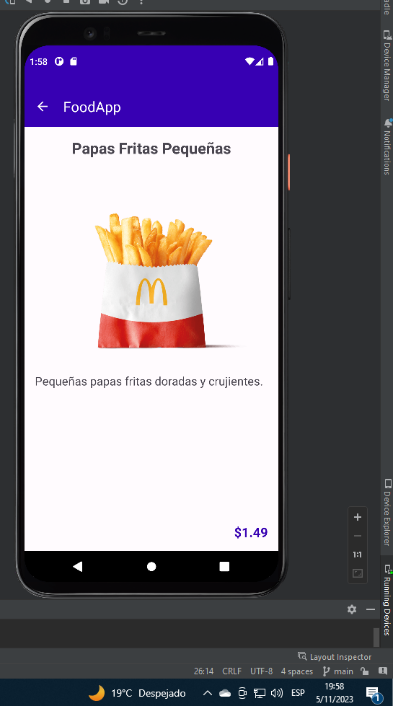
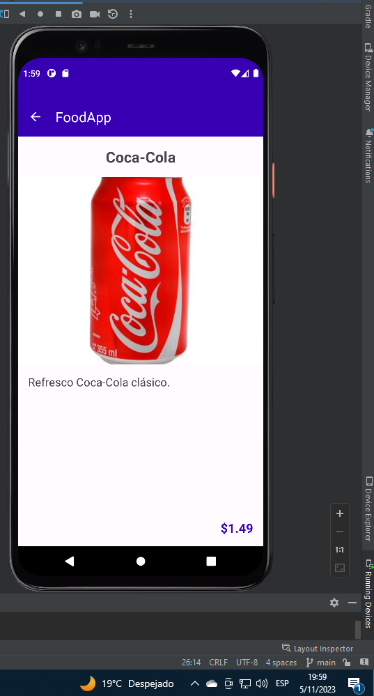
Pestaña comida



Pestaña Complementos

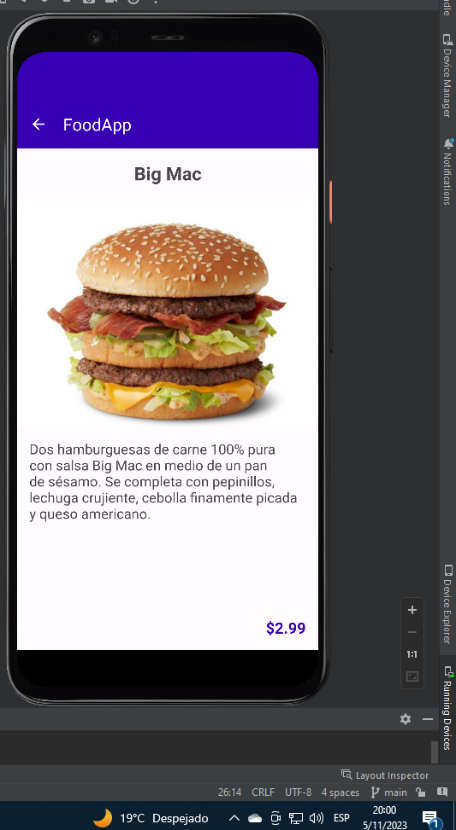
1. Cuando el usuario presione sobre alguno de los listados en las diferentes pestañas realizadas en el punto anterior, la aplicación deberá mostrar un nuevo activity donde se le dará al usuario una descripción más específica del alimento seleccionado.





Descripción del alimento

Descripción del alimento



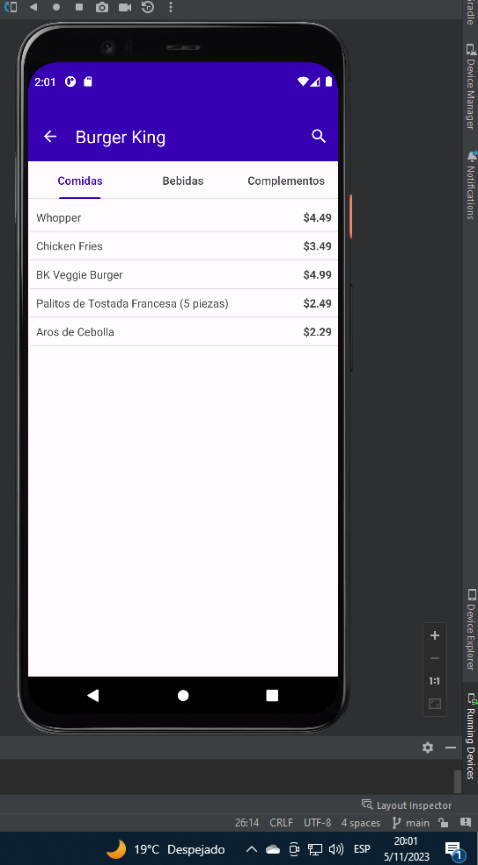
Descripción del alimento

1. La aplicación deberá tener un Toolbar personalizado, con las siguientes características:
   1. Título correspondiente a cada activity.
   2. Botón de búsqueda en caso de que contenga listas el activity.
   3. Botón para regresar al activity anterior.

Botón de busqueda

Botón de regreso

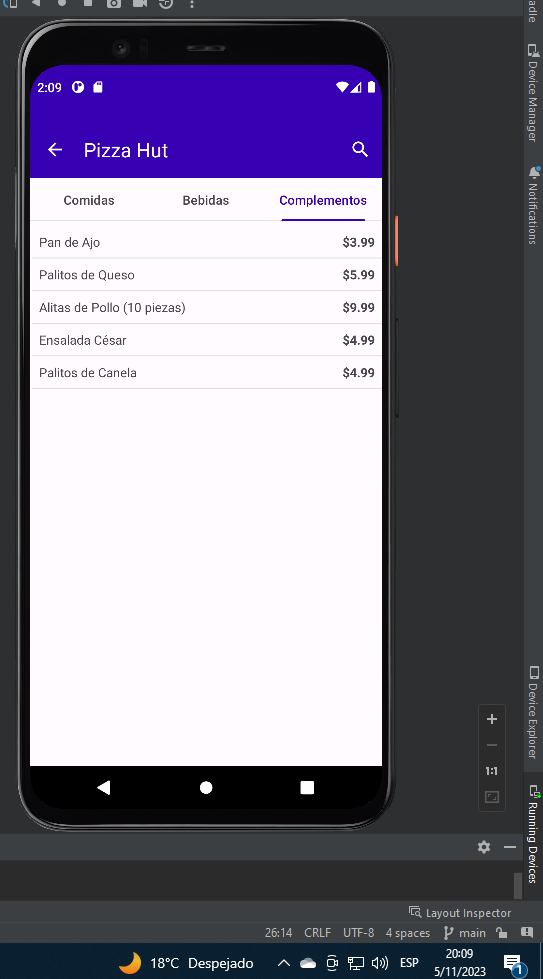
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



1. En el listado de los restaurantes, cada elemento deberá tener un menú contextual que muestre las opciones: comida, bebida y complementos. Cuando se seleccione alguna de estas opciones, debe abrir el activity en la pestaña correspondiente.

Opciones:comida, bebida complementos

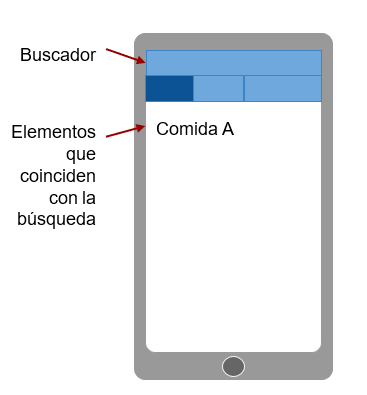




1. La aplicación deberá incluir un buscador sobre los activities que contengan listas.

Buscador

Captura de pantalla de computadora

Descripción generada automáticamente

# 

# **Parte 2: Entrega**

1. Entregar un archivo zip con la carpeta correspondiente al proyecto generado de la Persistencia de datos.
2. Entrega un documento Word, que contenga:
   1. Los datos de identificación.
   2. Una descripción de los pasos realizados, en el que se detalle el proceso que se llevó a cabo para la creación de una aplicación para la persistencia de datos con capturas de pantalla que muestran la ejecución del proceso.

**Descripción de los pasos realizados:**

Food App

- Diagrama de la base de datos

- Crear proyecto

- importación de dependencias

- creación de estructura del proyecto

- Modelos

- Data Access

- Adapters

- Menus

- Activities

- Fragments

Diagrama de la base de datos

Se hicieron 4 tablas.

- Restaurants

- Foods

- Drinks

- Complements

Las tablas de restaurantes tienen una relación uno a muchos con las tablas de foods, drinks y complementos.

Crear proyecto

Se creo proyecto en Android Studio, lenguaje Java con min SDK 24 y con el lenguaje Groovy para la gestión de dependencias en el aradle.

Importación de dependencias

Para este proyecto se utilizó Room como librería para la administración de base de datos y Picasso para la renderización de imágenes.

implementation "androidx.room:room-runtime:$room\_version"

annotationProcessor "androidx.room:room-compiler:$room\_version"

implementation 'com.squareup.picasso:picasso:2.8'

Estructura del proyecto

Se creo las carpetas de Adapters, Activities, Fragments, DataAccess y Models

Modelos.

Se creo una entidad base llamada Product. Es una clase abstracta de la cual van a heredar sus propiedades Food, Drink y Complement. Se utiliza la anotación de Room para demostrar que es una entidad

Se crean los modelos de Food, Drink y Complement

Se crea modelo de Restaurant

Data Access

Se creo un archivo de Java llamada Data con información de prueba, contiene arreglos para cada uno de los modelos.

Se creo un archivo llamado AppDatabase.java en el cual se hace el set de Room. Se cargan los datos de los modelos durante la primera ejecución de la base de datos haciendo uso del Callback de Room en su método OnCreate

Para cada modelo se creó una interfaz para que actúe como DAO. Esto permite que se puedan realizar consultas a la base de datos

Adapters

Se necesitaron 3 adaptadores para el funcionamiento de la aplicación.

ProductAdapter : Este es el adaptador que se encargara de renderizar los productos (independientemente si son food, complementarios o drink) en un recyclerview. Se hereda de la siguiente clase RecyclerView.Adapter<ProductAdapter.ViewHolder> para sobrescribir los métodos onCreateViewHolder, onBindViewHolder y getItemCount, estos serán los encargados de ligar los datos con la vista. Asimismo, se implementa la interfaz Filterable, esto permitirá filtrar los datos de la lista por el nombre.

Para el correcto funcionamiento de este adaptador se creó un layout de xml el cual se llama product\_item\_list.xml el cual es una vista que muestra el nombre del producto y el precio

RestaurantAdapter: Este adaptador se crea para renderizar los restaurantes en un reciclarview, se hereda de la siguiente clase:

RecyclerView.Adapter<RestaurantAdapter.ViewHolder> para sobreescribir los métodos onCreateViewHolder, onBindViewHolder y getItemCount, estos serán los encargados de ligar los datos con la vista.

Para el correcto funcionamiento de este adaptador se creó un layout de xml el cual se llama list\_item.xml el cual es una vista que muestra el nombre del restaurante

ViewPagerAdapter: Para poder crear en una sola activity la funcionalidad de los tabs se necesita un viewpager y este necesita su adaptador el cual le dirá que fragment se debe renderizar

Menus

Para hacer uso de los options menus se agrega un nuevo Android Resource Package llamado menú. El cual en su interior tiene un menu.xml que lleva el comportamiento del option menú. En este caso se utilizó un icono de search y el actionViewClass será un searchview el cual permitirá filtrar los datos

Activities

Se crean 2 activities

FoodDetailActivity: En esta activity se renderiza el detalle del producto. Se hace una llamada a la base de datos para obtener la información del producto, asimismo se utiliza Picasso para la renderización de la imagen, la cual es una url que se encuentra en internet.

ProductsActivity: Esta activity actúa como un Holder para los fragments. Aquí se define el comportamiento del tablayout y del viewpager. El cual van sincronizados.

MainActivity: Esta activity es el punto de entrada de la aplicación, aquí se renderizan los restaurantes. Se utiliza una consulta que trae los restaurantes en una lista de LiveData. Esto debido a que la primera vez que carga la aplicación los datos no están listos en el momento que el MainActivity es ejecutado. Por lo que haciendo uso de LiveData podemos escuchar el momento cuando la información es cargada y renderizarla utilizando un recyclerview.

Fragments:

Los 3 fragments son exactamente lo mismo. Consultan el restaurantId que fue pasado desde el main activity, y consultan los productos disponibles. Una vez que se tiene la lista se sobrescriben los métodos de onCreateOptions menú para poder realizar el proceso de filtrado en el recyclerview dependiendo del nombre que se escriba

Capturas de Pantalla

Captura de pantalla mostrando las Activities, Adaptadores, la DataAcess, los Fragments, Models, DAO, drawable, layout, menú, xml entre otras carpetas.

Captura de pantalla de computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente