



Instituto Tecnológico de Chetumal

Proyecto Final

Manual técnico

Materia: Mobile App

Alumno: Andres Adrián Martin Canto

Profesor: May Canché Isaías

Semestre: 8vo Grupo: U

## Introducción

Este proyecto lo llevamos acabo con varios componentes, algunos de ellos son RecyclerView, textView, LinearLayout, buttonFloat, editText, entre otros, usuamos como base de datos SQLITE, pero con Room, usamos clases de recyclerview asi como el dao para poder saber los query.

## Manual técnico

Como primero lo que importamos es el viewBinding con los siguientes líneas de código todo esto lo ponemos dentro del Gradle Scripts como siguiente buscamos el build.gradle (Module: app) donde pondremos el siguiente comando para que nos reconozca el viewBinding sin embargo dentro de la clase de nuestro fragmente tenemos que poner lo siguiente para ya tener el viewBindin:

```
buildFeatures{  
    viewBinding = true  
}
```

```
private var _binding: FragmentMainBinding? = null  
private val binding get() = _binding!!
```

```
override fun onDestroy() {  
    super.onDestroy()  
    _binding = null  
}
```

Como siguiente pasamos al XML del fragmentMain para poner nuestro formulario y botones que nos ayudara.

Primero ponemos un textView para mostrarle al usuario donde tiene que poner el nombre del producto como siguiente podremos un editText para que pueda ingresar los datos después ponemos dos mas donde ira la cantidad y el precio.

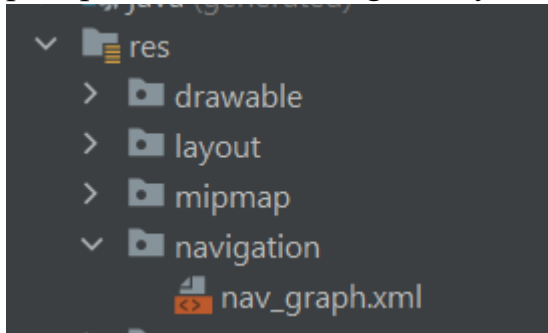
Agregaremos 4 botones para las siguientes cosas:

- Para poder ver la lista completa del usuario y esto lo haremos con la navegación, pero primero tenemos que importar la navegación poniendo lo siguiente en el gradel

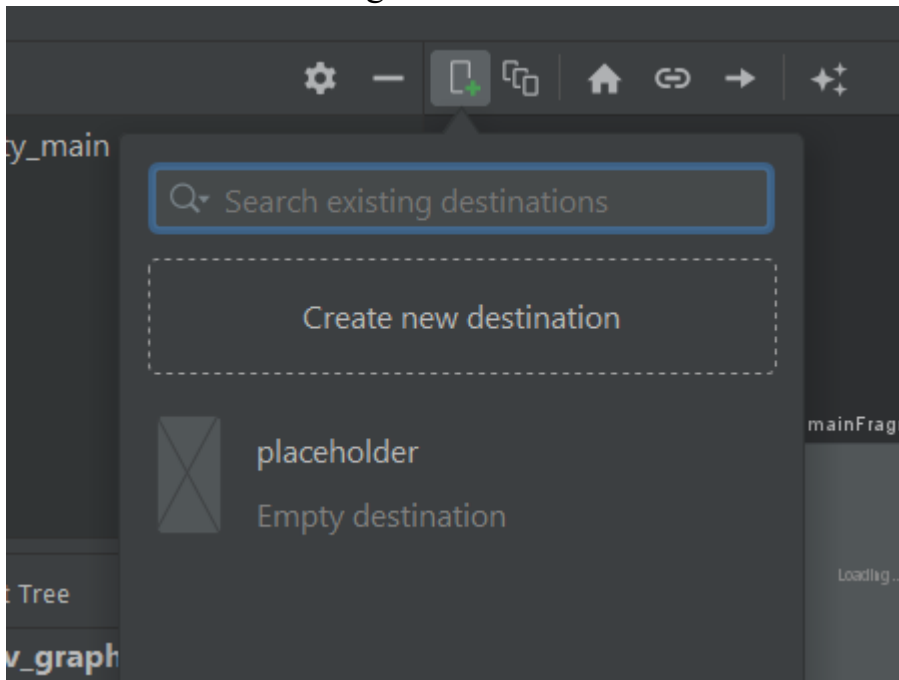
```
implementation 'androidx.navigation:navigation-fragment-ktx:2.5.3'  
implementation 'androidx.navigation:navigation-ui-ktx:2.5.3'  
  
id 'androidx.navigation.safeargs'
```

```
buildscript{  
    dependencies{  
        classpath("androidx.navigation:navigation-safe-args-gradle-plugin:2.5.3")  
    }  
}
```

Una vez ya tenemos esto tendremos que crear el xml para la navegación entre pantallas, una vez creada lo que haremos es traer los fragments para poder crear la navegación y así poner en un botón la navegación.



Esto lo hacemos de la siguiente manera:



Le damos al siguiente icono y de ahí seleccionaremos el fragment y una vez que nos salga dentro uniremos dependiendo para donde vayamos a cada fragment, una vez ya finalizado este proceso, pondremos el evento para ir a la siguiente fragment con el siguiente código:

```
// Evento que crea la accion para navegar a la otra ventana.
binding.btnVerLista.setOnClickListener {
    // Limpiar los paint
    limpiarPaint()
    // Ira a la otra pantalla.
    val action = MainFragmentDirections.navegarALista()
    findNavController().navigate(action)
}
```

Y así podremos ir a la siguiente ventana.

- El botón eliminar me servirá como lo dice eliminar un producto lo único que hago aquí es traer el valor del editText donde se ingresa el nombre del producto para cuando lo ponga ese es el que se va a borrar de igual manera si presiona el botón y no se ingreso nada le notificara un error de que ingrese el nombre del producto para que pueda eliminar y como ultimo borra la casilla de editText producto.

```
// Evento para eliminar un producto
binding.btnEliminar.setOnClickListener { it: View?
    val nombreProducto = binding.editNameProduct.text.toString()
    if (nombreProducto != "") {
        // Le enviamos como parametro el contenido ingresado por el usuario
        viewModel.deleteProduct(binding.editNameProduct.text.toString())
    } else {
        Toast.makeText(context, text: "Ingresa un nombre de producto", Toast.LENGTH_SHORT)
            .show()
    }
    limpiarPaint()
}
```

- El botón buscar hará buscar el elemento que haya ingresado el usuario y lo buscara si no lo encuentra entonces mostrara que no lo encontró así como cuando no ingrese nada le dirá que ponga un dato para buscarlo. Si lo encuentra lo pondrá en el editText correspondiente para que vea el usuario que si existe el dato.
- Boton agregar, agregará siempre y cuando todas las casillas estén llenas y así podrá almacenar el producto.

Como segunda pantalla lo que se muestra es un LinearLayout con dos textView uno para marcarle donde esta el precio total y el otro donde mostrara la cantidad del pago total.

Como segundo dato lo que tendrá es un recyclerview donde saldrán los ítems con cada item y sus datos eso lo hará con un observer que checa que si hay un cambio lo hace.

Y como ultimo hay un botón para regresar al formulario y lo hace de igual manera que el botón para ir a la lista.

En las clases que no se ven son producto que es la clase donde esta la tabla y nos sirve para saber como quedara, la clase DATABASE es la tiene la base de datos, la clase ProductDao tiene las consultas a la base de datos asi como obtener los datos y eliminar.

El productoRepository es una clase que se comunica con viewModel y la base de datos para que pueda pasar toda la información correspondiente.

## Conclusión

Este proyecto nos ayudo a mejorar en Android studio, donde pudimos usar RecyclerView, textView, LinearLayout, buttonFloat, editText, entre otros, usuamos como base de datos SQLITE, pero con Room, entre otras cosas, fue un proyecto maravilloso que nos ayudo a reforzar nuestros conocimientos y poder ayudar a la sociedad con este proyecto.