

Guía de ejercicios - Elementos de la interfaz, navegación e interacción (II)



¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

La siguiente guía de estudio tiene como objetivo practicar y ejercitar los contenidos que hemos visto en clase.

¡Vamos con todo!



Tabla de contenidos

Actividad guiada: Navegación entre fragmentos con bottom navigation	2
Preguntas de proceso	13
¡Manos a la obra! - Actualización de fragmentos	14
¡Manos a la obra! - Mostrar las Notificaciones desde el Perfil	14
Preguntas de cierre	14
Solución ¡Manos a la obra! - Actualización de fragmentos	15
Solución ¡Manos a la obra! - Mostrar las Notificaciones desde el Perfil	16



¡Comencemos!



Actividad guiada: Navegación entre fragmentos con bottom navigation

Una de las formas de navegación más comunes que se encuentran en las apps en general y Android en particular, es la Bottom Navigation Bar, que nos permite movernos entre distintas pantallas de interés.

En esta actividad guiada construiremos una app con 1 actividad que tiene 1 Bottom Navigation Bar y 1 fragmento para cada pantalla, con un total de 3 fragmentos.

Home

This is home fragment

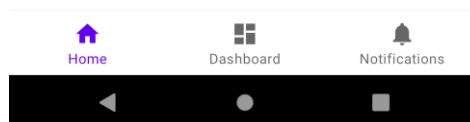


Imagen 01. App a crear.
Fuente: Desafío Latam.

Para lograr este resultado, vamos a seguir los siguientes pasos:

1. Crear un proyecto usando el template de Empty Activity.

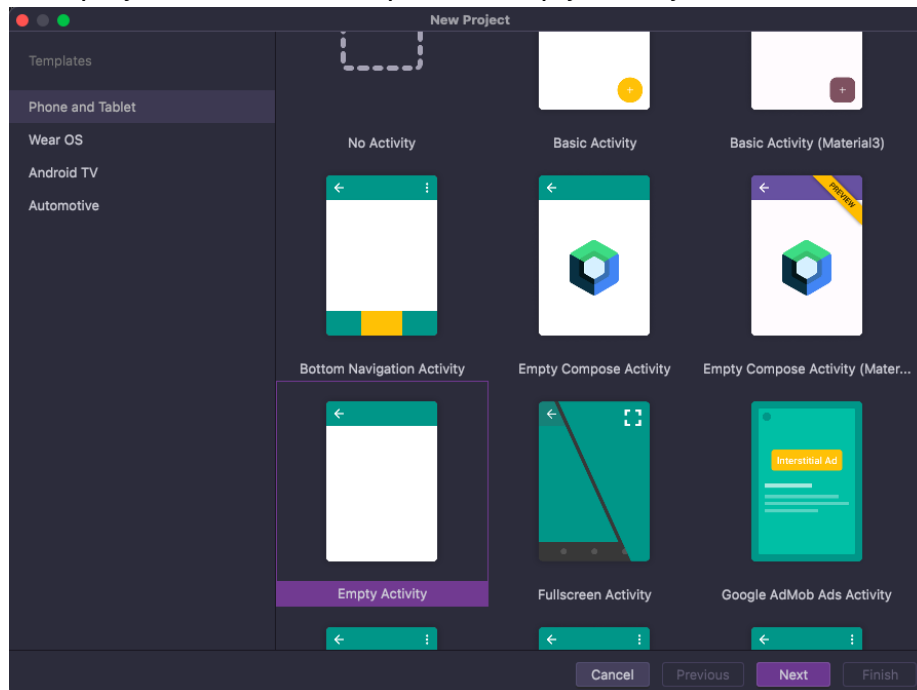


Imagen 02. Empty Activity..

Fuente: Desafío Latam.

2. Agregar los vectores para ocupar en cada elemento de la Bottom navigation bar, usando la biblioteca de material incluida en el SDK. Los tres vectores son de color negro, y automáticamente la Bottom bar lo resalta cuando está seleccionado.

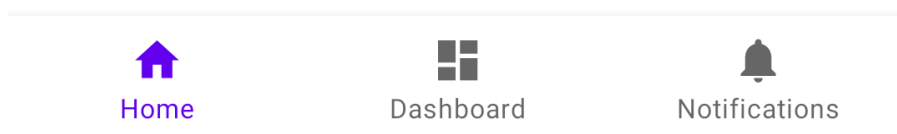


Imagen 03. Agregar los vectores.

Fuente: Desafío Latam.

Los tres vectores serán agregados usando Asset Studio:

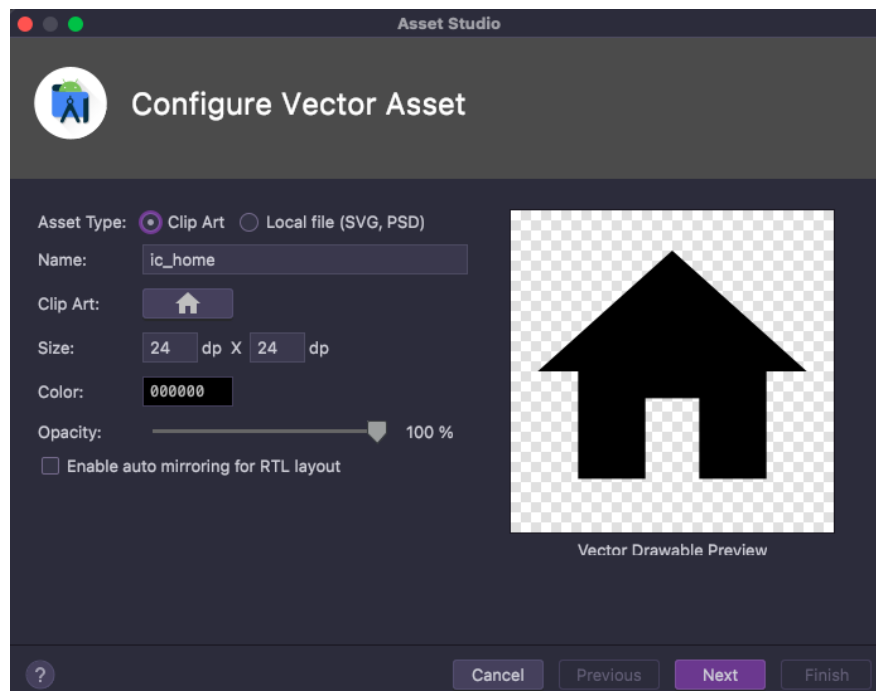


Imagen 04. Agregar los vectores.
Fuente: Desafío Latam.

3. Vamos a crear el archivo de recurso para el menú de la Bottom navigation bar.
 - a. Primero creando un directorio para el menú: con el botón derecho sobre la carpeta res/ -> New -> Android Resource Directory.

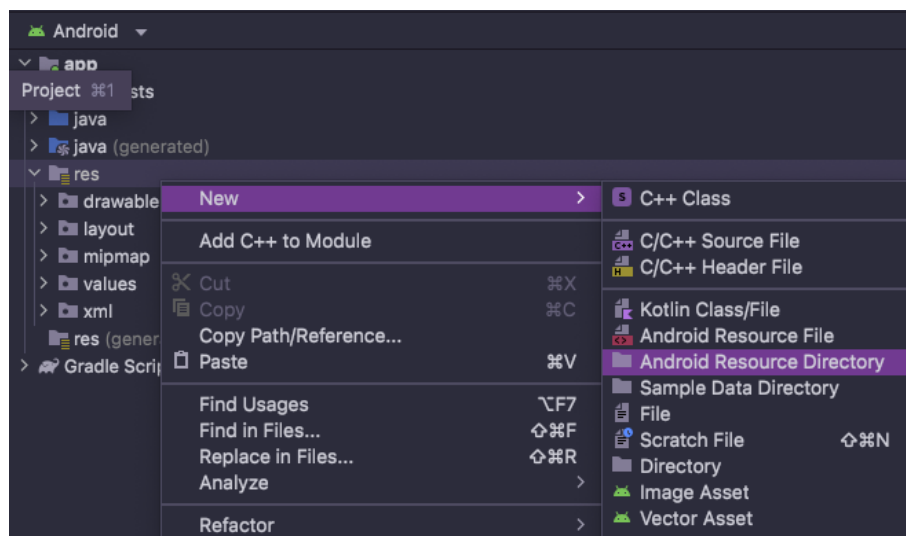


Imagen 05. Crear el archivo de recurso.
Fuente: Desafío Latam.

- b. Seleccionar el tipo de recurso (*Resource type*) que corresponde, en este caso, menú.

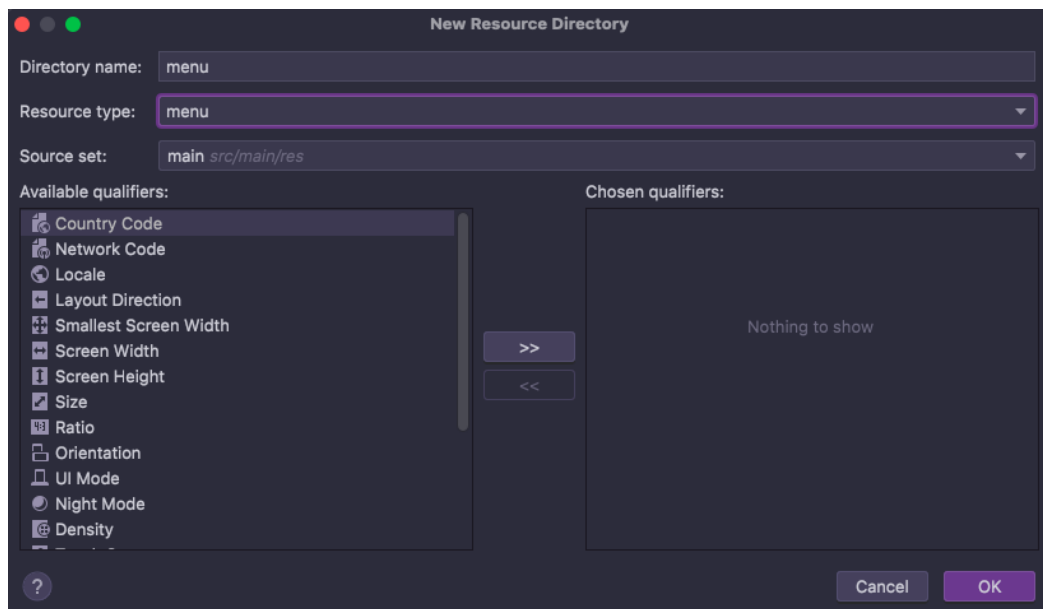


Imagen 06. Selección tipo de recurso.

Fuente: Desafío Latam.

- c. Luego de creada la carpeta de recursos para menú, agregamos un nuevo archivo, seleccionando con el botón derecho sobre el menú.

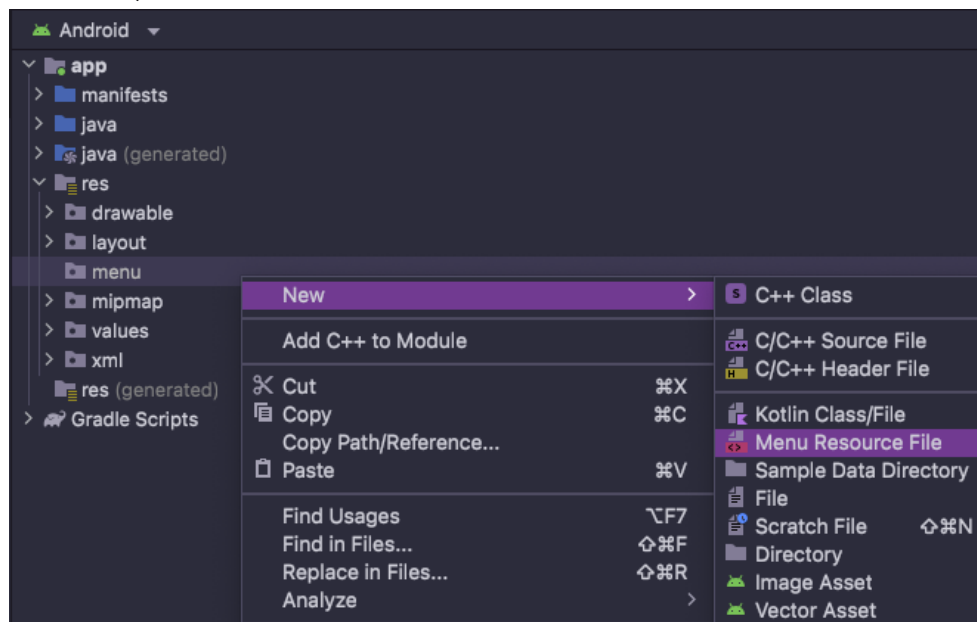


Imagen 07. Agregar nuevo archivo.

Fuente: Desafío Latam.

- d. Este menú va a ser utilizado por la Bottom Nav bar, así que le pondremos un nombre descriptivo para poder tener una referencia intuitiva de su uso. En este caso, la nombramos como bottom_nav_bar.

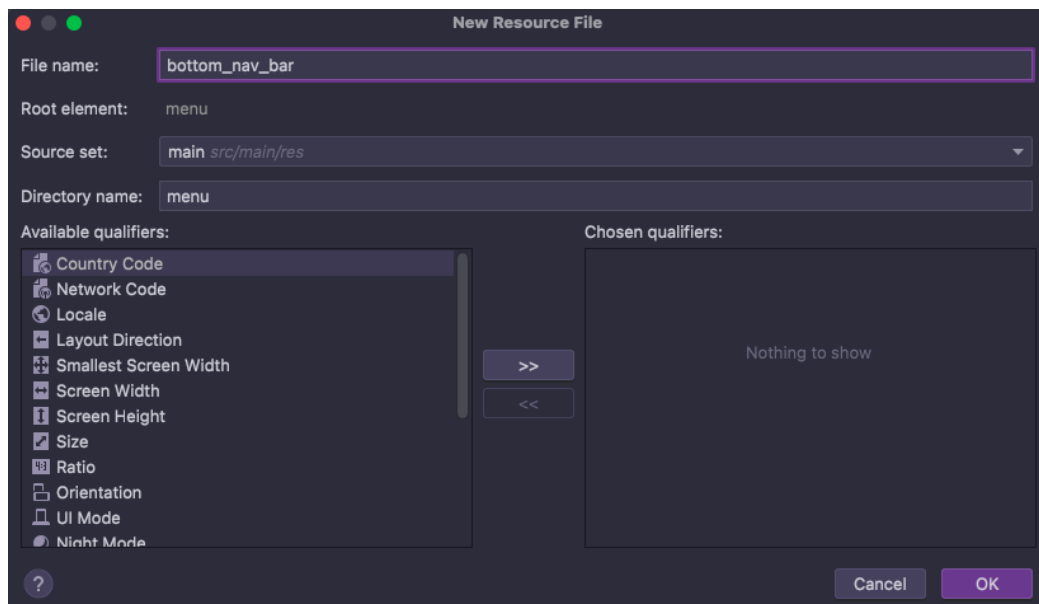


Imagen 08. Nombrar bottom_nav_bar.

Fuente: Desafío Latam.

- e. El archivo de menú contiene la definición de los tres elementos que serán utilizados, indicando identificador, el ícono asociado (creado previamente) y el título que debe ser extraído al archivo strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home"
        android:title="@string/title_home" />

    <item
        android:id="@+id/navigation_dashboard"
        android:icon="@drawable/ic_dashboard"
        android:title="@string/title_dashboard" />

    <item
        android:id="@+id/navigation_notifications"
        android:icon="@drawable/ic_notifications"
        android:title="@string/title_notifications" />
</menu>
```

- 4. Declaramos la Bottom Navigation bar en el layout activity_main.xml, y lo ubicamos en la parte inferior de la pantalla dentro del ConstraintLayout.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_bar" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

5. Vamos a utilizar la biblioteca Navigation de Jetpack para navegar entre los fragmentos de la app.

- a. Para eso, vamos a agregar las dependencias en el archivo build.gradle del módulo.

```
implementation 'androidx.navigation:navigation-fragment:2.5.3'
implementation 'androidx.navigation:navigation-ui:2.5.3'
```

- b. Para luego crear una nueva carpeta de recursos para la navegación, con el botón derecho sobre res/ New ->

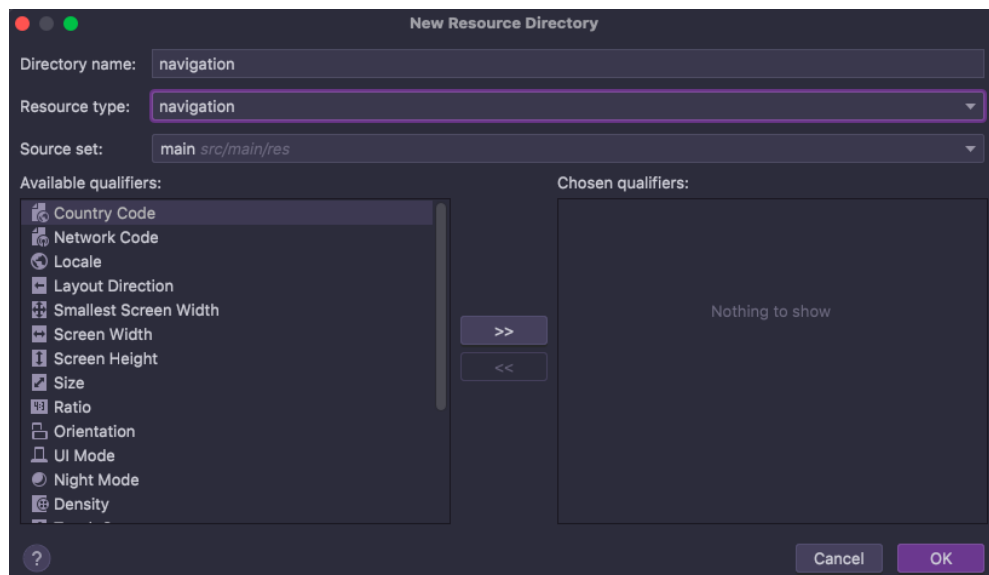


Imagen 09. Crear nueva carpeta de recursos.

Fuente: Desafío Latam.

- c. Después de creada la nueva carpeta de recursos, agregamos el archivo de navegación.

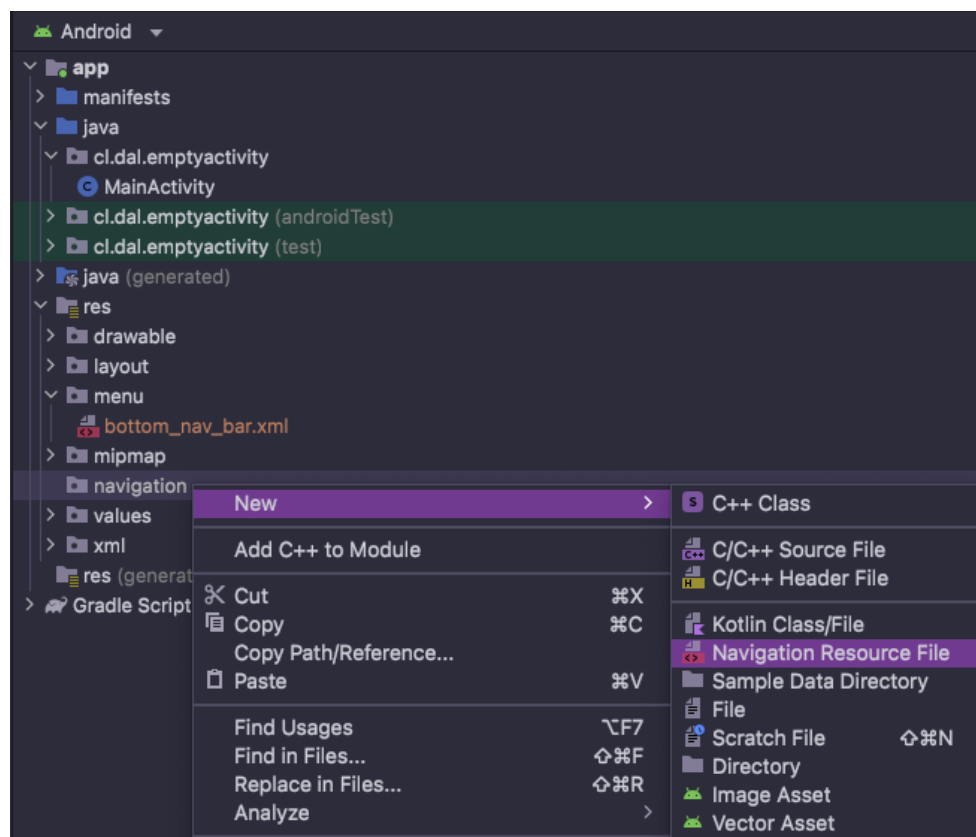


Imagen 10. Agregar el archivo de navegación.

Fuente: Desafío Latam.

- d. Y lo nombramos de forma descriptiva, por ejemplo, nav_graph.

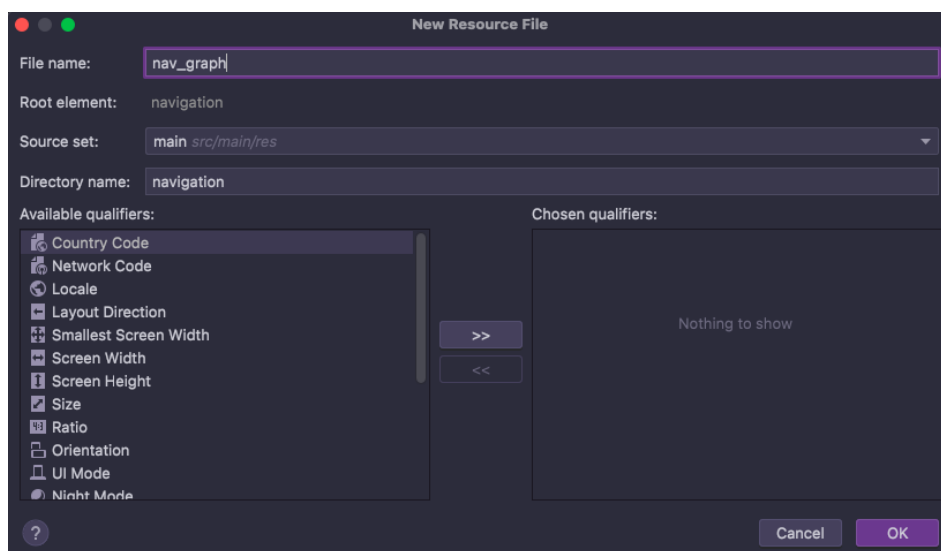


Imagen 11. Nombre descriptivo.

Fuente: Desafío Latam.

6. Agregamos el Navigation host al layout de la actividad. Dentro del Constraint layout agregamos un nuevo fragmento del tipo NavHostFragment para que sea el contenedor donde se muestran los fragmentos.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <fragment
        android:id="@+id/nav_host_fragment_activity_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/nav_graph" />
```

```
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/nav_view"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="0dp"
    android:layout_marginEnd="0dp"
    android:background="?android:attr/windowBackground"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:menu="@menu/bottom_nav_bar" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

En este punto, el gráfico de navegación está vacío, por lo que debemos agregar los fragmentos que corresponden.

7. Agregamos el primer fragmento Home.

Lo podemos hacer desde el menú File -> New -> Fragment o directamente desde el editor del gráfico de navegación seleccionando agregar -> Create new destination, donde crearemos un nuevo fragmento usando el template Fragment (Blank) y lo nombraremos como HomeFragment.

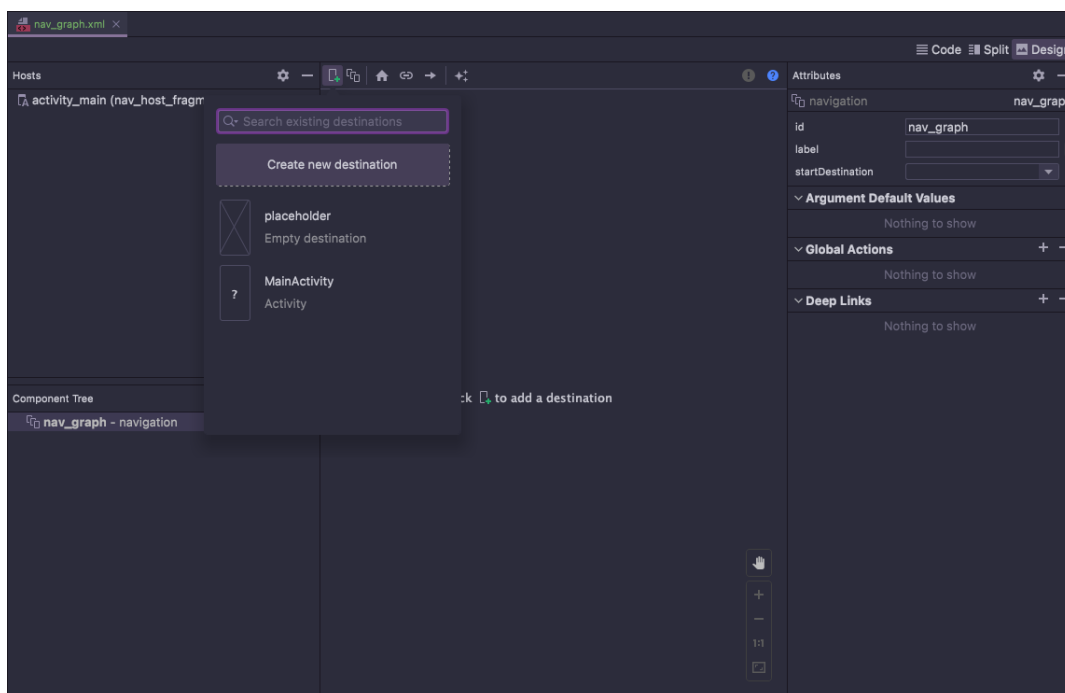


Imagen 12. Agregar el primer fragmento.

Fuente: Desafío Latam.

8. Luego de creado el HomeFragment, editamos el layout fragment_home.xml y modificamos el texto a mostrar.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".HomeFragment">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/home_text" />
</FrameLayout>
```

El string home_text está declarado en el archivo strings.xml

```
<string name="home_text">Home fragment</string>
```

9. Debemos repetir el mismo proceso de creación del fragment y modificación del string para DashboardFragment y NotificationFragment.

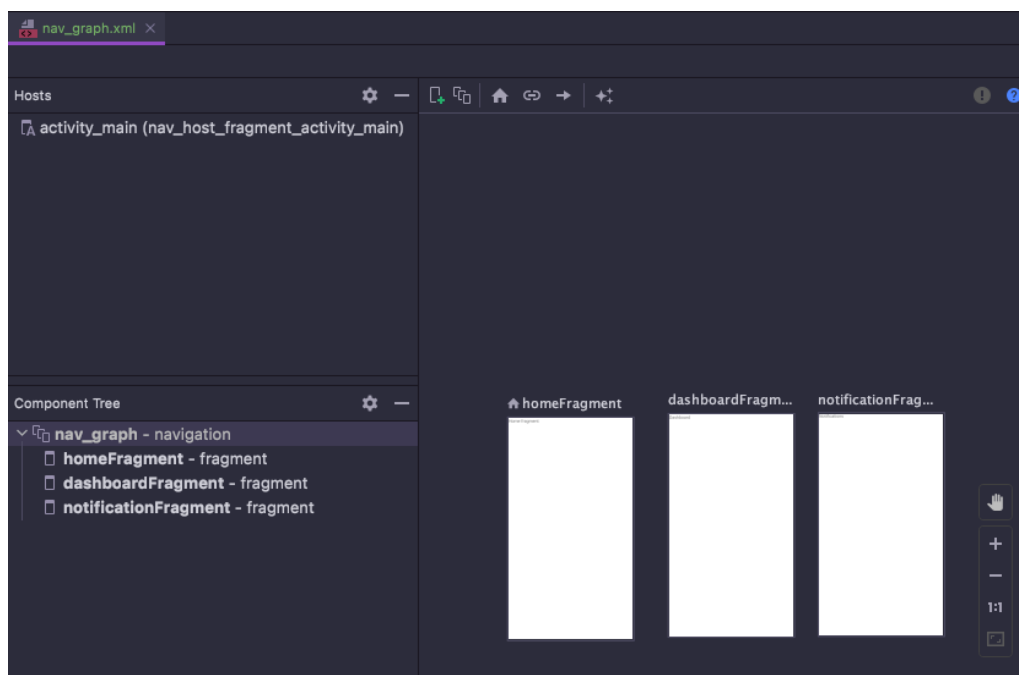


Imagen 13. Repetición de proceso.

Fuente: Desafío Latam.

10. Por último, vamos a enlazar el funcionamiento de la Bottom Navigation bar con Navigation, modificando el método onCreate de MainActivity.java para agregar la llamada al método `initBottomNavBar()`.

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        initBottomNavBar();  
    }  
  
    private void initBottomNavBar() {  
        // Recupera la vista de la barra de navegación declarada en el  
        // archivo activity_main.xml  
        BottomNavigationView navView = findViewById(R.id.nav_view);  
  
        // Recupera el Navigation Controller para la navegación  
        NavController navController =  
        Navigation.findNavController(this,  
        R.id.nav_host_fragment_activity_main);  
  
        // Asocia la Bottom Navigation bar con el NavController  
        NavigationUI.setupWithNavController(navView, navController);  
    }  
}
```

11. Al correr la app en un emulador o dispositivo, podemos ver que el funcionamiento de la Bottom Navigation bar va cambiando al fragmento seleccionado.

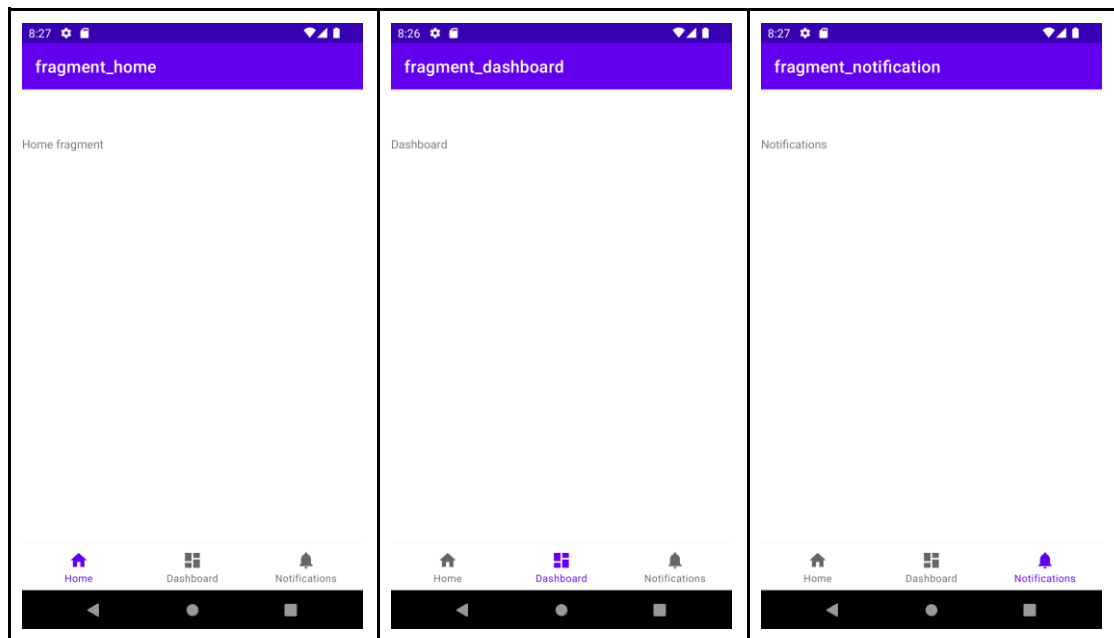


Imagen 14. App en emulador.
Fuente: Desafío Latam.

Preguntas de proceso

Reflexiona:

- ¿Qué representa un fragmento?
- ¿Qué se necesita para navegar desde un fragmento a otro?
- ¿Qué representan los archivos de recursos?





¡Manos a la obra! - Actualización de fragmentos

Imagina que le presentas esta maqueta de app a un posible cliente, quien te indica que le hace más sentido que además del Home y Dashboard, se podría sumar una pantalla de Perfil (Profile) con la información del usuario, dejando la pantalla de notificaciones accesible directamente desde la pantalla de Perfil mediante un botón.

Para lo anterior, deberás agregar un nuevo fragmento Perfil a la navegación y modificar el menú de la Bottom Nav bar para reemplazar el fragmento de notificaciones por el de Perfil.



¡Manos a la obra! - Mostrar las Notificaciones desde el Perfil

Luego de agregar el botón en el layout del fragmento de Perfil, deberás crear una acción de navegación para ser utilizada por el NavController para navegar desde la pantalla de perfil hacia la pantalla de notificaciones.

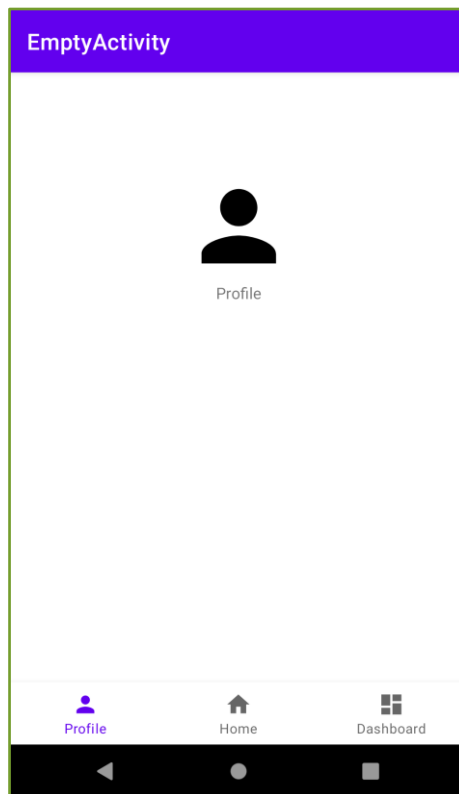


Nota: Recuerda que este contenido fue revisado en la sesión **Elementos de navegación**.

Preguntas de cierre

- ¿De qué forma interactúan los fragmentos al utilizar la biblioteca Navigation?
- ¿Cuántas actividades y fragmentos tenemos en la app?
- ¿Por qué se prefiere tener más fragmentos que actividades en una app?

Solución ¡Manos a la obra! - Actualización de fragmentos



Solución ¡Manos a la obra! - Mostrar las Notificaciones desde el Perfil

