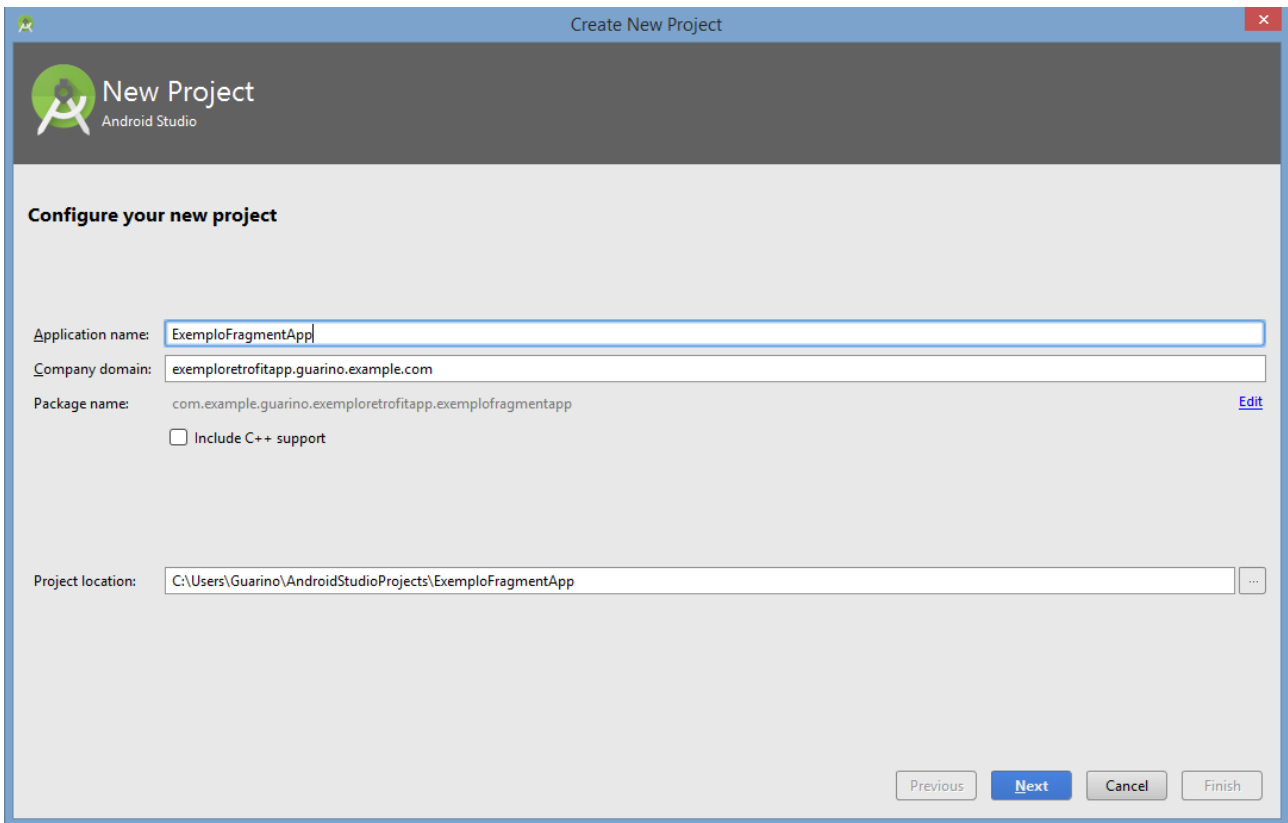


## Fragments

### Criar um novo projeto

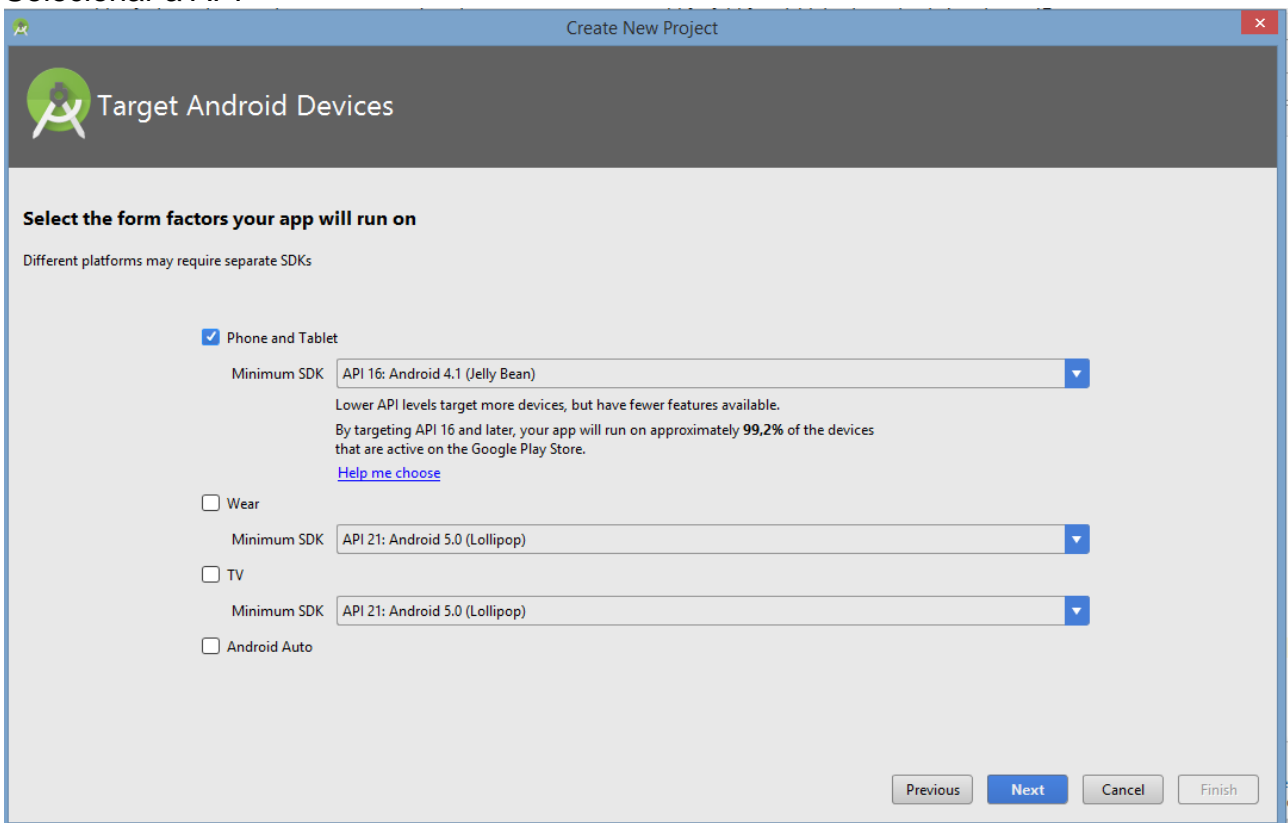


The screenshot shows the 'Create New Project' dialog in Android Studio. The title bar reads 'Create New Project'. The main header area contains the Android Studio logo and the text 'New Project' and 'Android Studio'. Below this, the section 'Configure your new project' is displayed. The form includes the following fields and options:

- Application name:** ExemploFragmentApp
- Company domain:** exmploretrofitapp.guarino.example.com
- Package name:** com.example.guarino.exmploretrofitapp.exemplofragmentapp (with an [Edit](#) link)
- Include C++ support
- Project location:** C:\Users\Guarino\AndroidStudioProjects\ExemploFragmentApp (with a browse button)

At the bottom right, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

### Selecionar a API



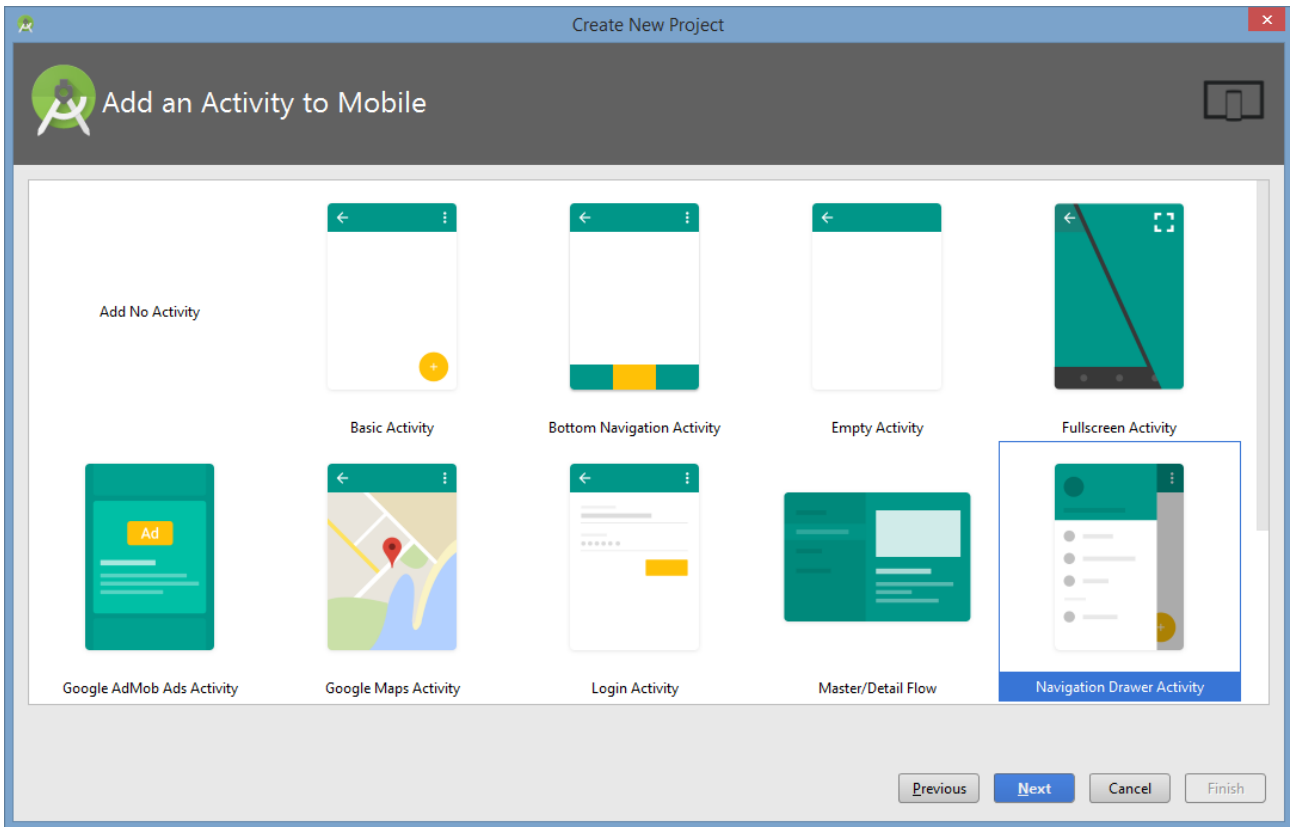
The screenshot shows the 'Create New Project' dialog in Android Studio, specifically the 'Target Android Devices' step. The title bar reads 'Create New Project'. The main header area contains the Android Studio logo and the text 'Target Android Devices'. Below this, the section 'Select the form factors your app will run on' is displayed, with the subtitle 'Different platforms may require separate SDKs'.

The form includes the following options and fields:

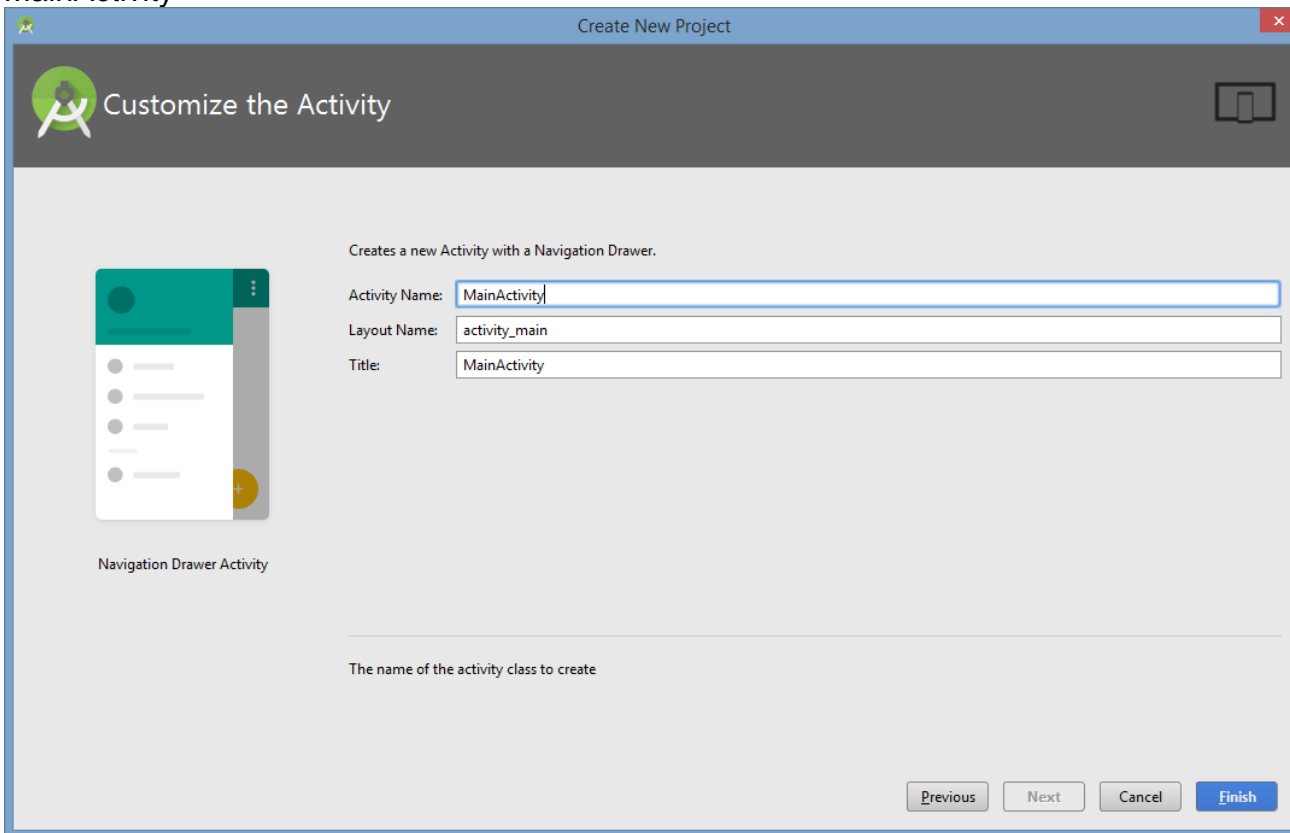
- Phone and Tablet
  - Minimum SDK: API 16: Android 4.1 (Jelly Bean) (with a dropdown arrow)
  - Lower API levels target more devices, but have fewer features available.
  - By targeting API 16 and later, your app will run on approximately **99,2%** of the devices that are active on the Google Play Store.
  - [Help me choose](#)
- Wear
  - Minimum SDK: API 21: Android 5.0 (Lollipop) (with a dropdown arrow)
- TV
  - Minimum SDK: API 21: Android 5.0 (Lollipop) (with a dropdown arrow)
- Android Auto

At the bottom right, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

### Navigation Drawer Activity



## MainActivity



Finish.

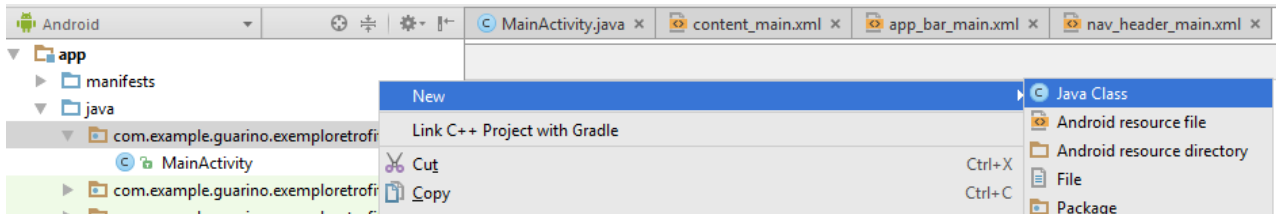
## Adicionar as Dependências do ButterKnife.

- butterknife
- butterknife-compiler.

Veja o tutorial do ButterKnife no site.

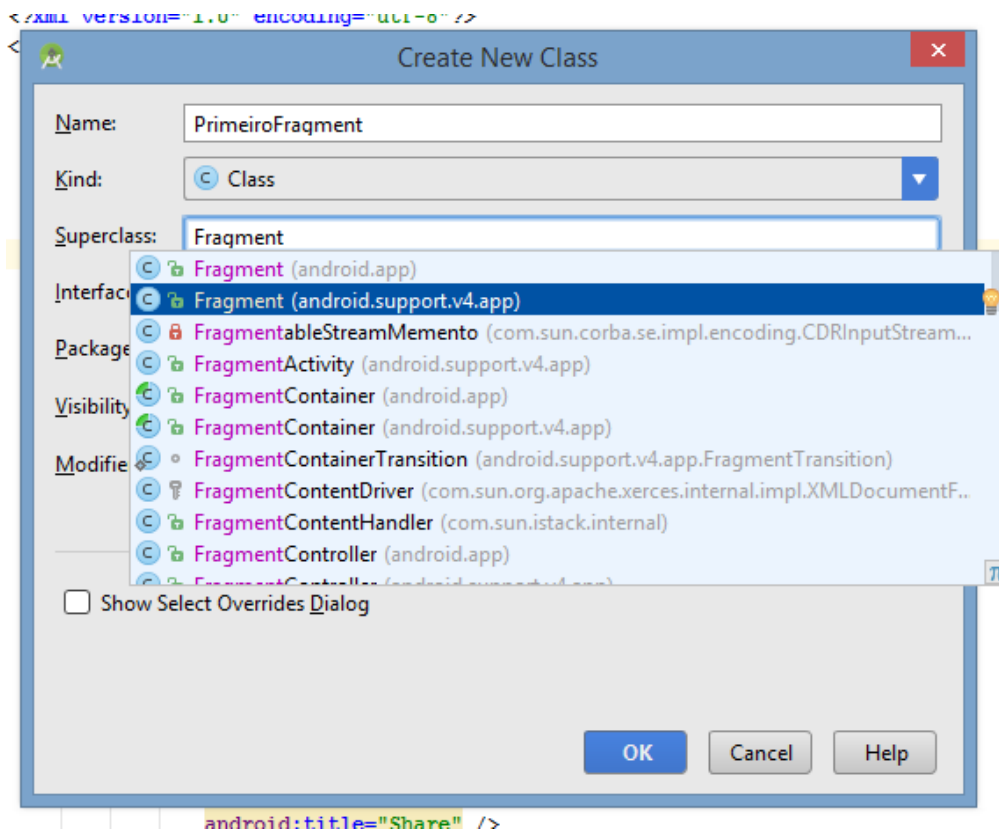
## Adicionar um Fragment.

No pacote principal, clique com o botão direito e adicione uma nova Java Class.

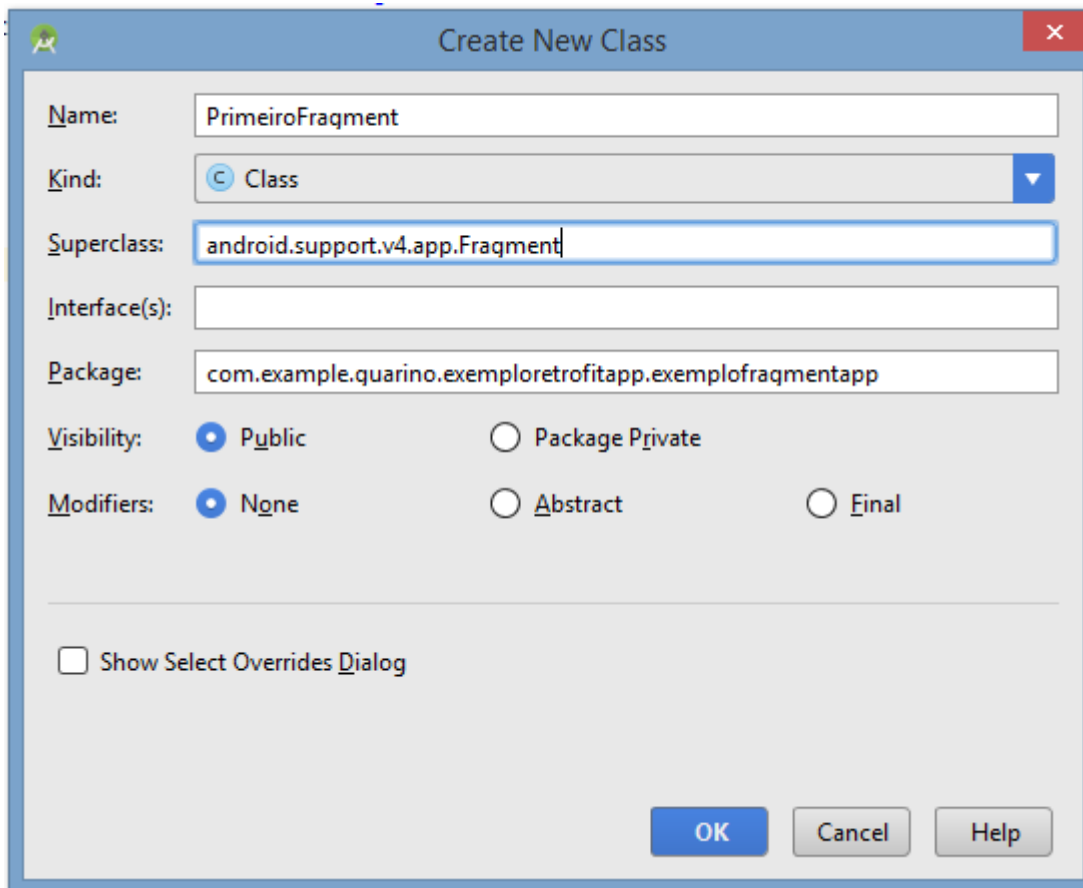


Crie a classe PrimeiroFragment, que deve herdar a classe Fragment.

Porém, selecione o Fragment (android.support.v4.app) para manter a retrocompatibilidade.



A superclass deve ficar como na figura a seguir.



Clique em OK.

Adicione o evento onCreateView

```
/**
 * Created by Guarino on 14/10/2017.
 */

public class PrimeiroFragemt extends Fragment {

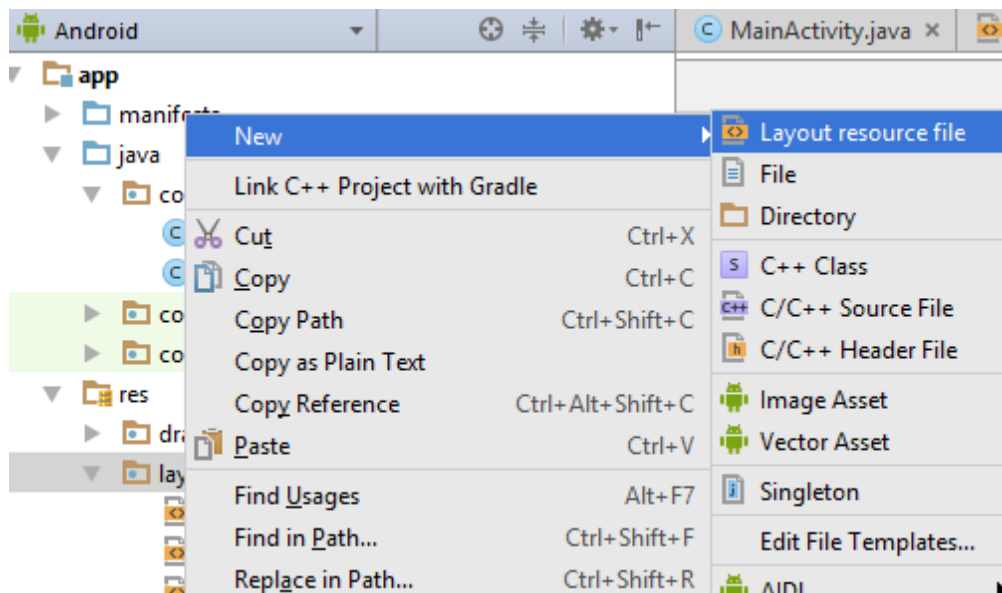
    onCreate

    public Animation onCreateAnimation(transit, enter, ne... Fragment
    public void onCreate(savedInstanceState) {...} Fragment
    public View onCreateView(inflater, container, savedInstanceState) {...} Fragment
    public void onCreateContextMenu(menu, v, menuInfo) {...} Fragment
    public void onCreateOptionsMenu(menu, inflater) {...} Fragment
    public void onActivityResult(savedInstanceState) {...} Fragment
    public void onViewCreated(view, savedInstanceState) {...} Fragment
```

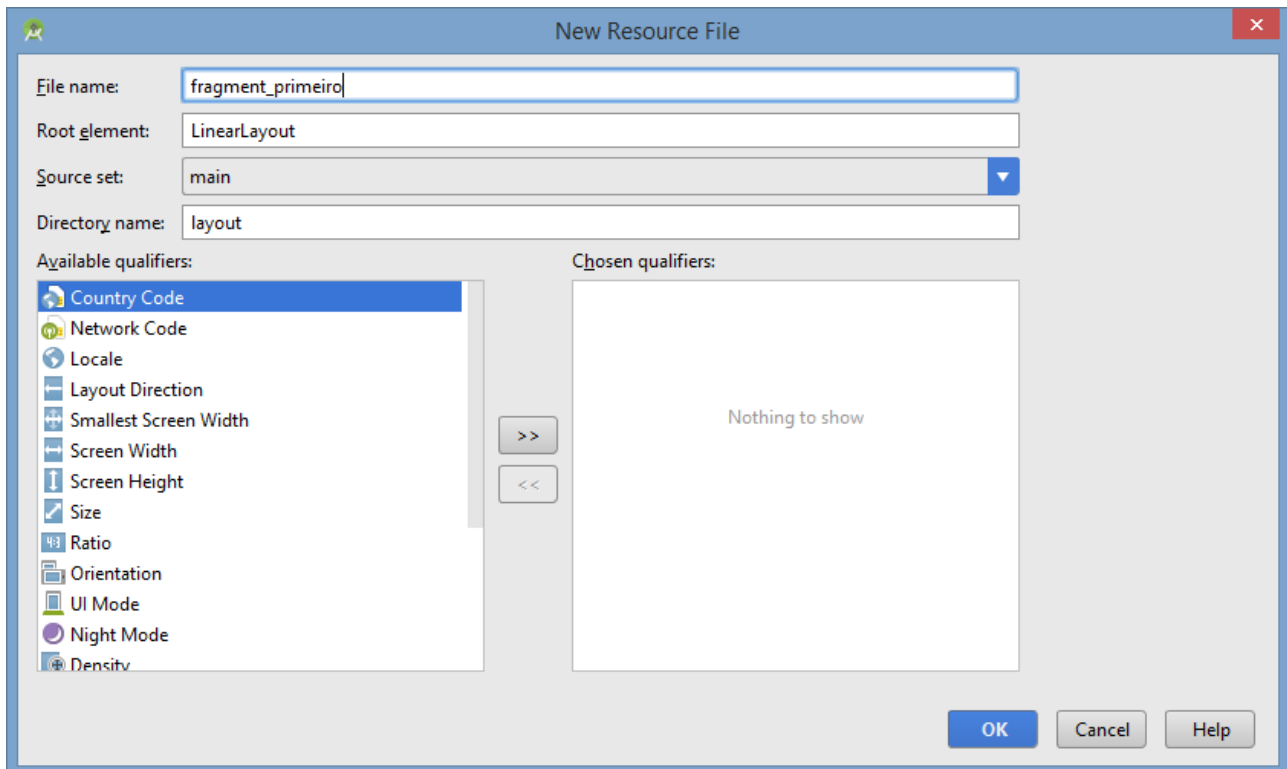
O código inicialmente deve ficar como segue:

```
public class PrimeiroFragment extends Fragment {  
  
    @Nullable  
    @Override  
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup  
container, @Nullable Bundle savedInstanceState) {  
  
        View view = super.onCreateView(inflater, container, savedInstanceState);  
        ButterKnife.bind(this, view);  
  
        return view;  
    }  
}
```

Agora, devemos criar o resource do fragment.  
Na pasta layout, botão direito > Layout resource file.



## Adicione o fragment\_primeiro



Clique em OK.

Remova a linha

```
<?xml version="1.0" encoding="utf-8" ?>
```

do fragment. Deixe o fragment como segue.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"

    tools:context=
"com.example.guarino.exemploretrofitapp.exemplofragmentapp.PrimeiroFragment"
    >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Digite o nome: " />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/edtNome" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Próximo"
        android:id="@+id/btnProximo" />

</LinearLayout>
```

Altere o onCreateView do PrimeiroFragment para

```
public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {

    View view = inflater.inflate(R.layout.fragment_primeiro, container,
false);

    ButterKnife.bind(this, view);

    return view;
}
```

**Alterando a MainActivity para aceitar Fragments.**

Abra o arquivo app\_bar\_main.xml.

Procure pela linha

```
<include layout="@layout/content_main" />
```

Remova esta linha.

Adicione o bloco a seguir do fechamento do AppBarLayout, ou seja, no lugar da linha removida.

```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/fragment_main">
</FrameLayout>
```

Em seguida, coloque o AppBarLayout e o FrameLayout dentro de um LinearLayout.

....

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>
```

```

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/fragment_main">
</FrameLayout>

```

```
</LinearLayout>
```

.....

Na MainActivity, no fechamento do onCreate, adicione a chamada do Fragment.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
        }
    });

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
    drawer.setDrawerListener(toggle);
    toggle.syncState();

    NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);

    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    transaction.add(R.id.fragment_main, new PrimeiroFragment());
    transaction.commit();
}

```

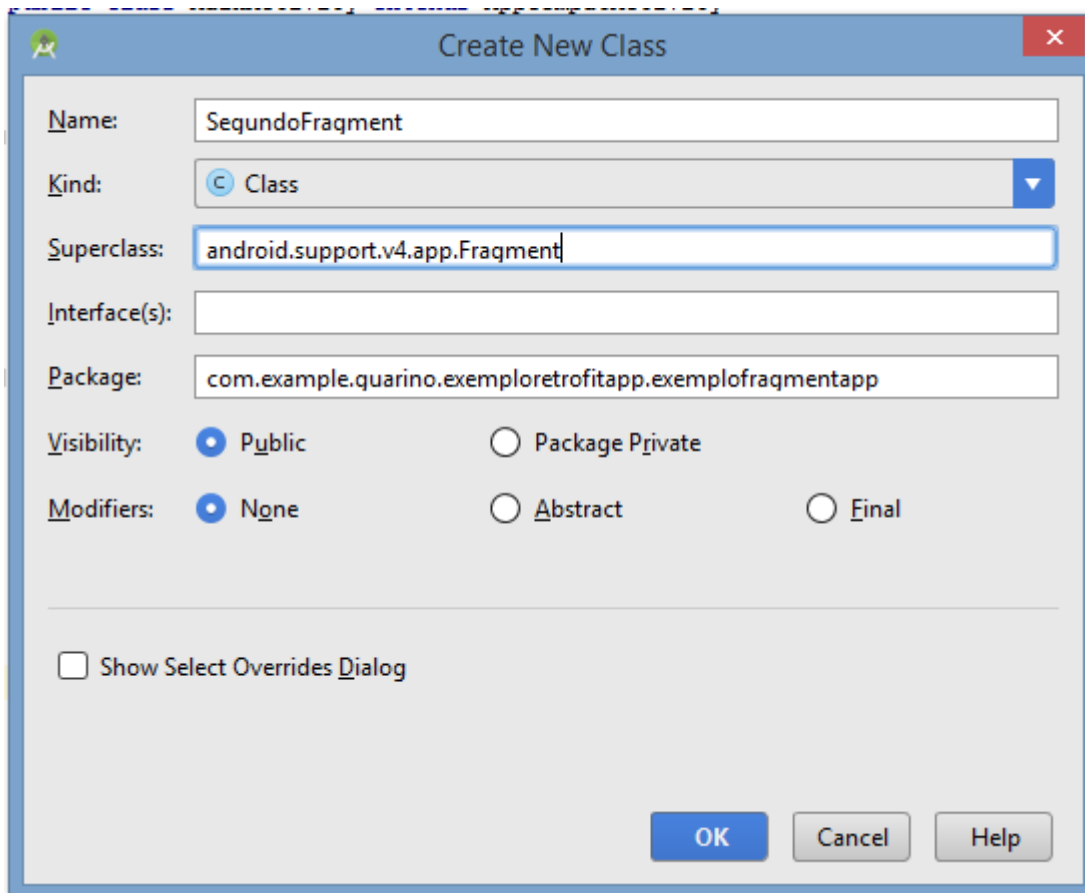
## Execute a aplicação.

Irá aparecer o Fragment dentro da Activity.

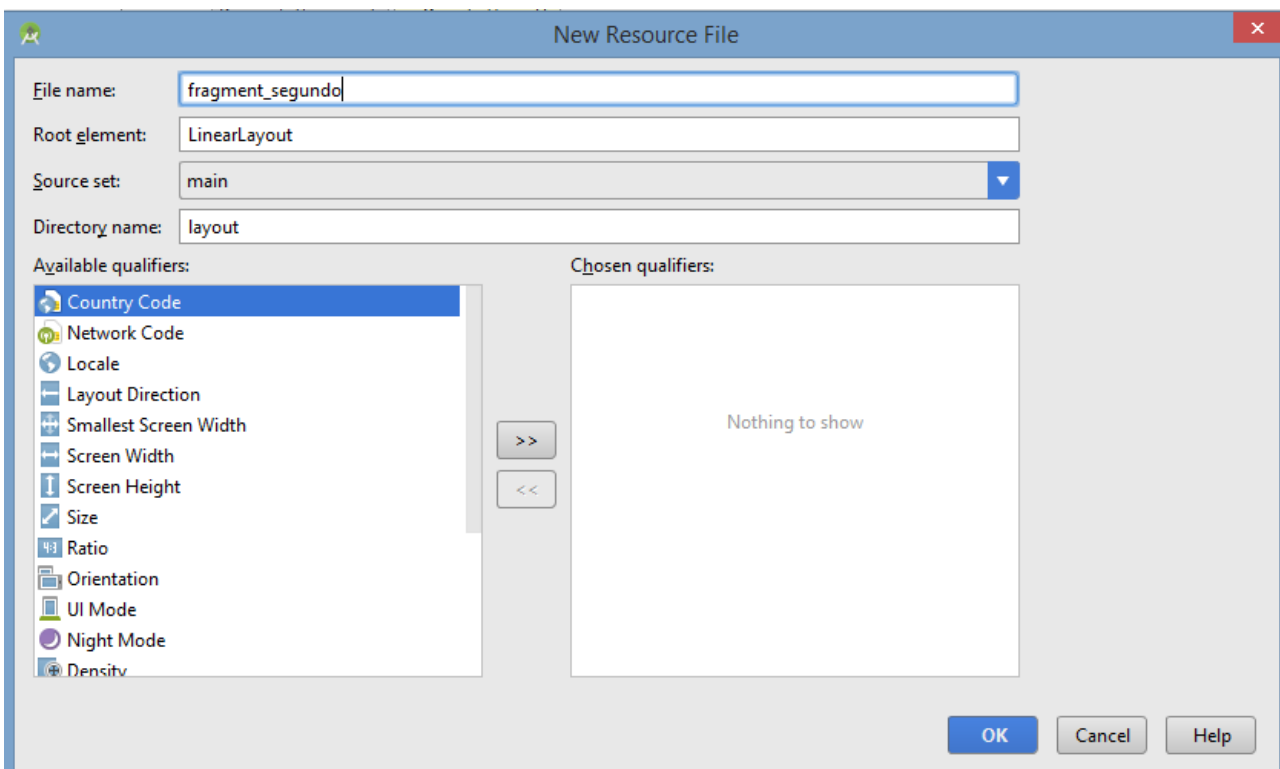


## Adicionando o segundo fragment.

Adicione uma nova Java Classe no pacote principal.



Na pasta layout, crie o resource para o fragment.



Volte no SegundoFragment e adicione o onCreateView.

```
public class SegundoFragment extends Fragment {

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_segundo, container,
false);
        ButterKnife.bind(this, view);

        return view;
    }
}
```

Abra o fragment\_segundo e remova a linha

```
<?xml version="1.0" encoding="utf-8"?>
```

Altere o arquivo para...

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"

    tools:context="com.example.guarino.exemploretrofitapp.exemplofragmentapp.Segundo
Fragment"
    >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Segundo fragment"/>

</LinearLayout>
```

Abra o arquivo activity\_main\_drawer.xml na pasta menu.

Altere os dois primeiros itens para chamar os dois fragments.  
Comente os demais itens.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_primeirofragment"
            android:icon="@drawable/ic_menu_camera"
            android:title="Primeiro Fragment" />
        <item
            android:id="@+id/nav_segundofragment"
            android:icon="@drawable/ic_menu_gallery"
```

```

        android:title="Segundo Fragment" />
</group>

<!--<item android:title="Communicate">-->
    <!--<menu>-->
        <!--<item-->
            <!--android:id="@+id/nav_share"-->
            <!--android:icon="@drawable/ic_menu_share"-->
            <!--android:title="Share" />-->
        <!--<item-->
            <!--android:id="@+id/nav_send"-->
            <!--android:icon="@drawable/ic_menu_send"-->
            <!--android:title="Send" />-->
        <!--</menu>-->
    <!--</item>-->

</menu>

```

Abra a MainActivity e altere a chamada do menu que fica no evento `onNavigationItemSelectedListener`

```

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelectedListener(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    FragmentManager fragmentManager;
    FragmentTransaction transaction;
    if (id == R.id.nav_primeirofragment) {
        fragmentManager = getSupportFragmentManager();
        transaction = fragmentManager.beginTransaction();
        transaction.replace(R.id.fragment_main, new PrimeiroFragment());
        transaction.commit();

    } else if (id == R.id.nav_segundofragment) {

        fragmentManager = getSupportFragmentManager();
        transaction = fragmentManager.beginTransaction();
        transaction.replace(R.id.fragment_main, new SegundoFragment());
        transaction.commit();
    }
    // else if (id == R.id.nav_slideshow) {
    // } else if (id == R.id.nav_manage) {
    // } else if (id == R.id.nav_share) {
    // } else if (id == R.id.nav_send) {
    // }

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}

```

### Execute a aplicação.

Chame o primeiro fragment.

Chame o segundo fragment.