

Aplicaciones Móviles

DOMICIANO RINCÓN

INGENIERÍA TELEMÁTICA
INGENIERÍA DE SISTEMAS
DISEÑO DE MEDIOS INTERACTIVOS

Leer y escribir datos de Views

ANDROID



Views de uso común

Button

1. Leer el texto de un botón

```
String texto = b.getText().toString();
```

En la variable texto queda lo que está escrito en el botón

2. Escribir texto de un botón

```
b.setText("Hola");
```

El botón pasa a tener el texto “Hola”

Views de uso común

EditText

1. Leer el texto de un botón

```
String texto = et.getText().toString();
```

En la variable texto queda lo que está escrito en el EditText

2. Escribir texto de un botón

```
et.setText("Hola");
```

El EditText pasa a tener el texto "Hola"

3. Escribir el hint

```
et.setHint("Escriba su numero");
```

El consejo del editText cambia a "Escriba su numero"

Views de uso común

ImageView

1. Cargar una imagen a partir de un recurso

```
iv.setImageResource(R.drawable.imagen);
```

2. Cargar una imagen a partir de un archivo (Requiere permisos)

```
Bitmap m = BitmapFactory.decodeFile(Environment.getExternalStorageDirectory()+"/foto.jpg");  
iv.setImageBitmap(m);
```

Views de uso común

Checkbox

1. Leer estado del checkbox

```
cb.isChecked()
```

2. Escribir estado del checkbox

```
cb.setChecked(true);
```

Esto hace que el Checkbox quede seleccionado

Views de uso común

RadioButton/RadioGroup

1. Leer estado del radioButton

```
int id = rg.getCheckedRadioButtonId();
```

```
    switch (id){  
        case R.id.mas:  
            //Hacer algo  
            break;  
        case R.id.fem:  
            //Hacer algo  
            break;  
    }
```

Views de uso común

DatePicker

1. Leer la fecha

```
int anho = dp.getYear();
```

```
int mes = dp.getMonth();
```

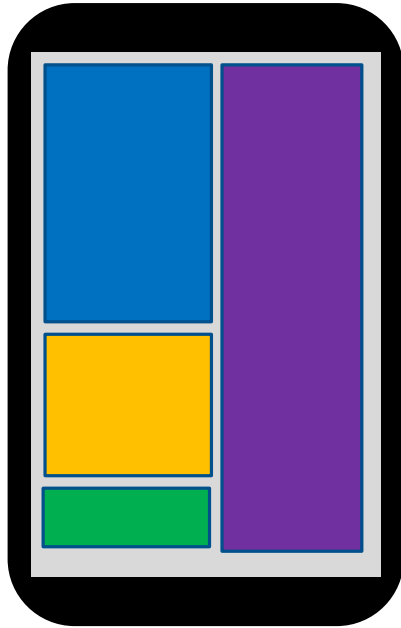
```
int dia = dp.getDayOfMonth();
```

Tenga en cuenta que el método `getMonth()` devuelve número entre 0 y 11.

Fragments

ANDROID

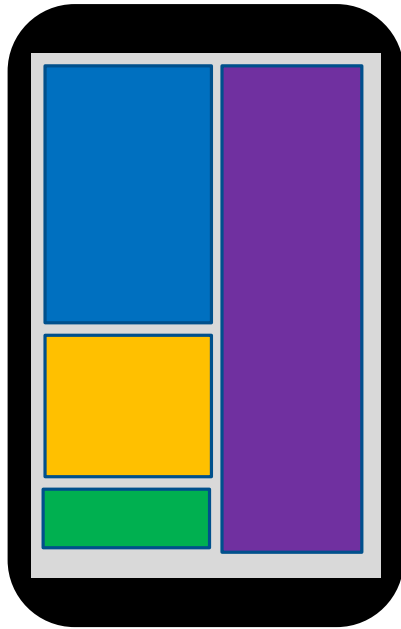
Fragments



Representa una parte o comportamiento de la interfaz de una activity. De esta forma una Activity se puede componer de Fragments.

Tiene también dos partes: JAVA y XML.

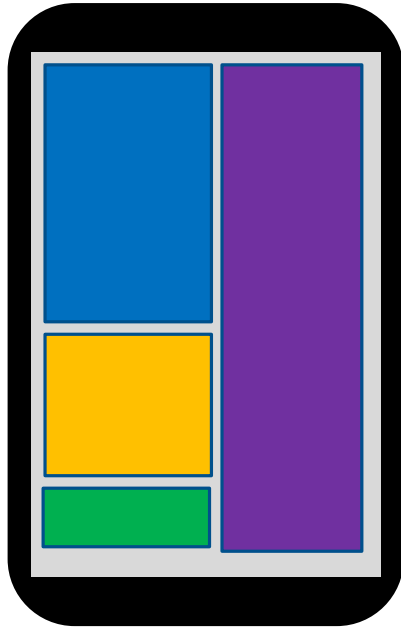
Fragments



En XML se debe cargar un contenedor de fragments llamado `FrameLayout`

```
<FrameLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/contenido"
/>
```

Fragments



Luego, puede cargar cualquier fragment en el contenedor de la siguiente forma

```
FragmentA fragmentA = new FragmentA();
```

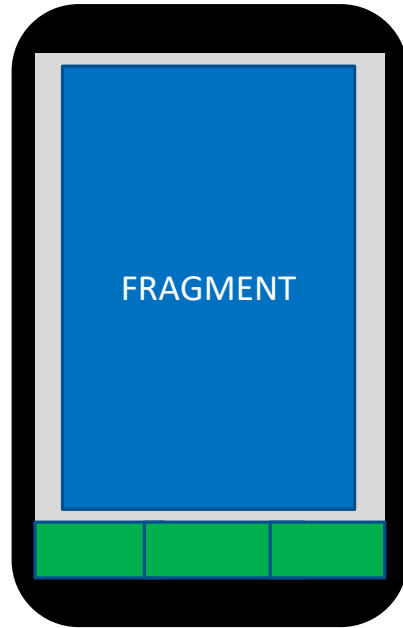
```
FragmentManager fragmentManager = getSupportFragmentManager();
```

```
FragmentTransaction transaction = fragmentManager.beginTransaction();
```

```
transaction.replace(R.id.contenido, fragmentA);
```

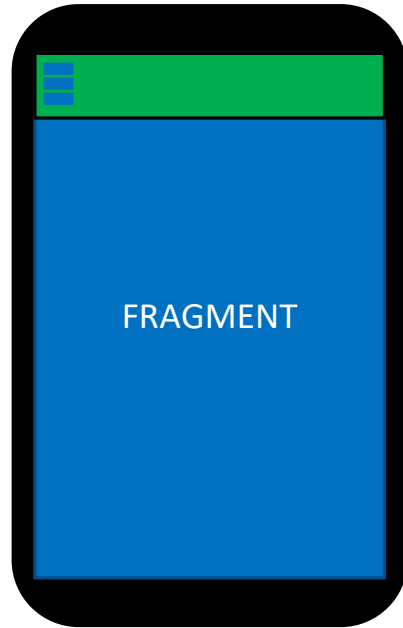
```
transaction.commit();
```

Fragments



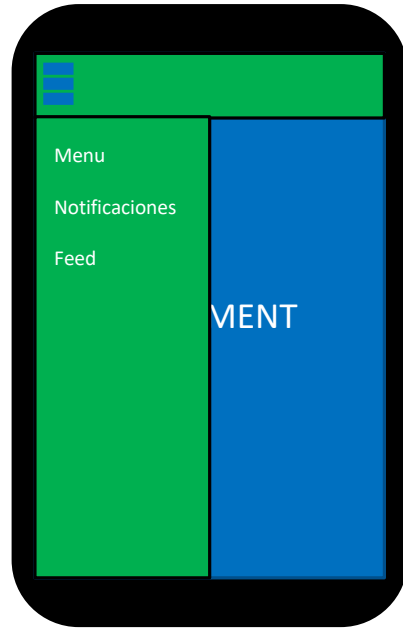
La forma los fragments más usuales
es con barra inferior de navegación

Fragments



0 con Menú estilo hamburguesa

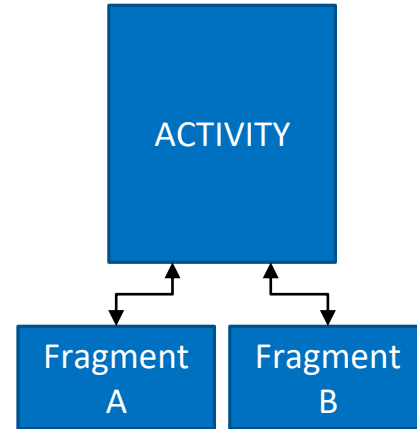
Fragments



0 con Menú estilo hamburguesa

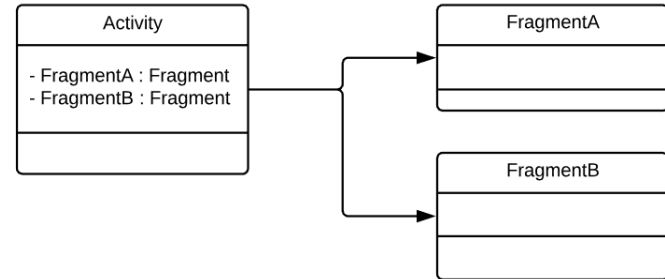
Fragments

Los datos generados por un fragment pueden ser enviados a la actividad. O pueden ser enviados a otro fragment, pero siempre el intermediario es la actividad.



Fragments

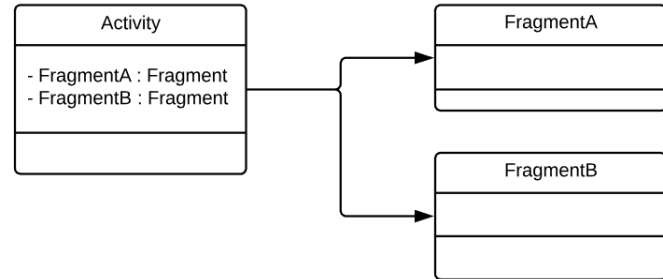
De este modo se deben aplicar técnicas de inversión de control para enviar información del fragment A a la actividad.



Fragments

Para enviar la información de la actividad al fragment se usa un bundle.

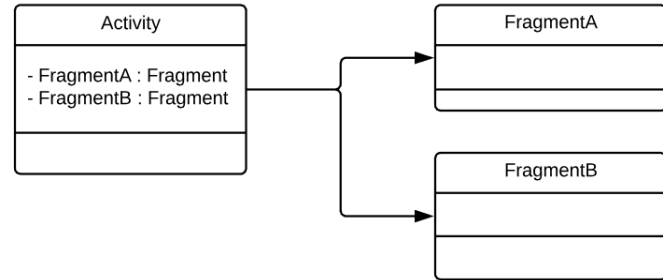
```
Bundle bundle = new Bundle();  
bundle.putString("data", "Prueba");  
fragmentB = new FragmentB();  
fragmentB.setArguments(bundle);  
FragmentManager fragmentManager = getSupportFragmentManager();  
FragmentTransaction transaction = fragmentManager.beginTransaction();  
transaction.replace(R.id.contenido, fragmentB);  
transaction.commit();
```



Fragments

Para enviar la información de la actividad al fragment se usa un bundle.

```
@Override
public View onCreateView(...) {
    ...
    Bundle bundle = this.getArguments();
    ...
}
```



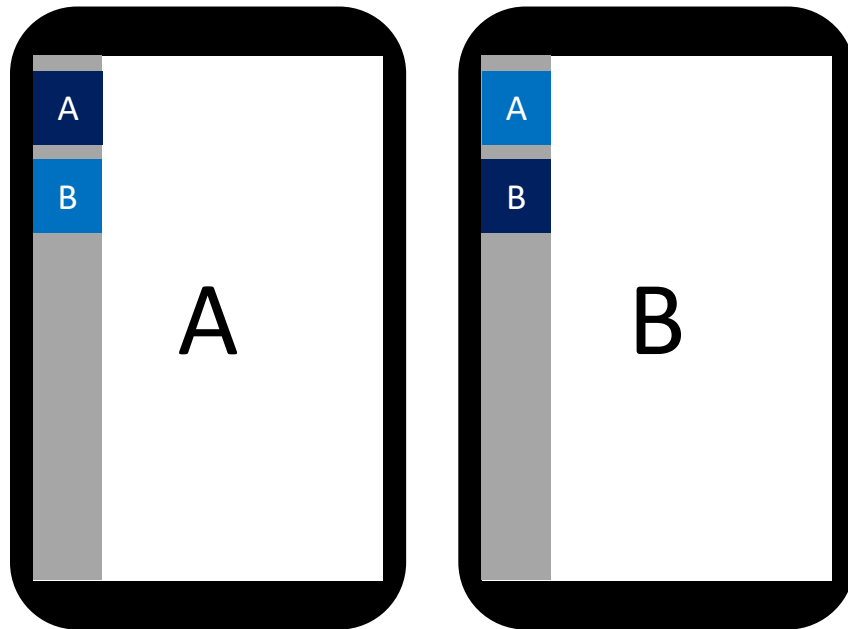
Ejercicio

ANDROID



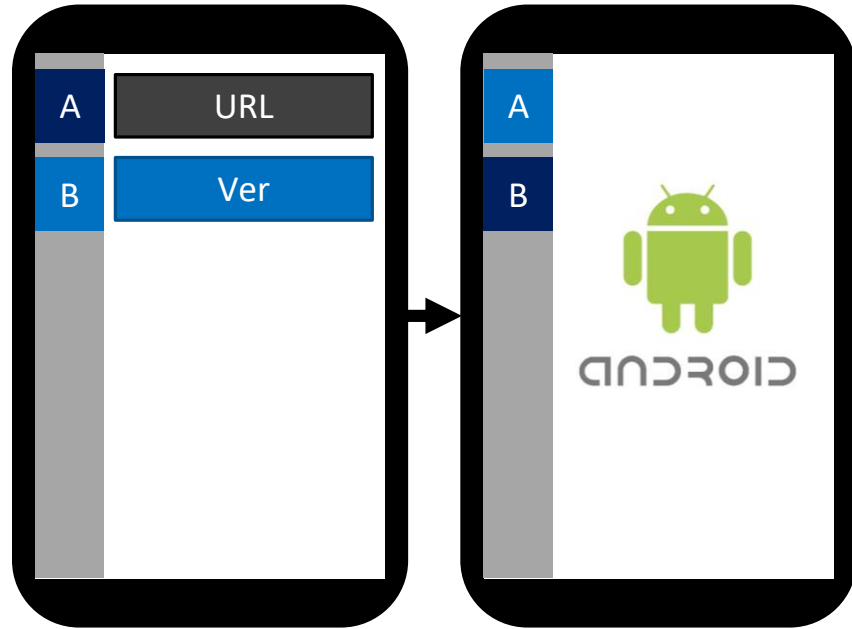
Ejercicio aplicaciones

Cree un menú custom en el que tenga dos botones que al pulsarlos, pueda ver dos fragmentos, el A y el B.



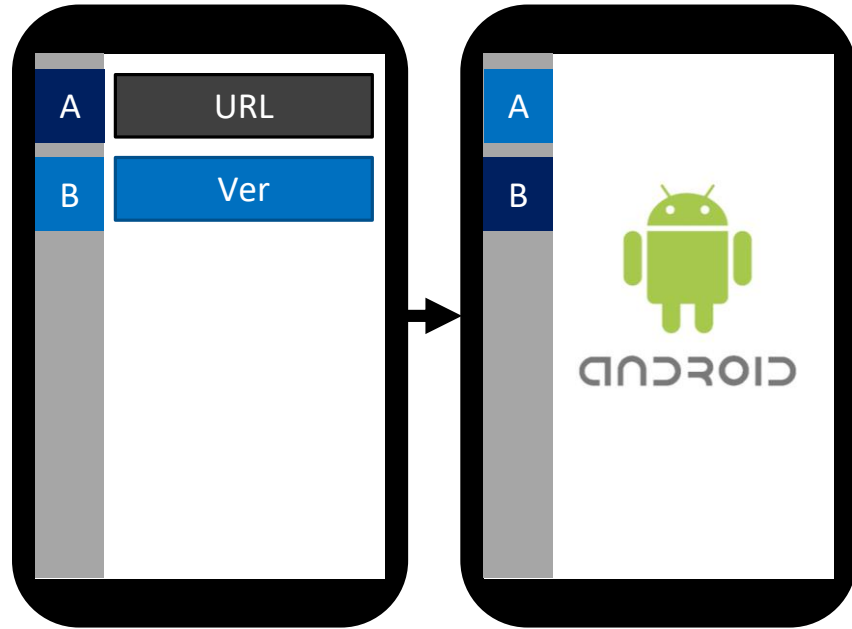
Ejercicio aplicaciones

El fragmento A recibe una URL de una imagen. Al pulsar el botón “ver”, inicia la descarga de una foto y cuando termine, la foto es mostrada en el fragment B.



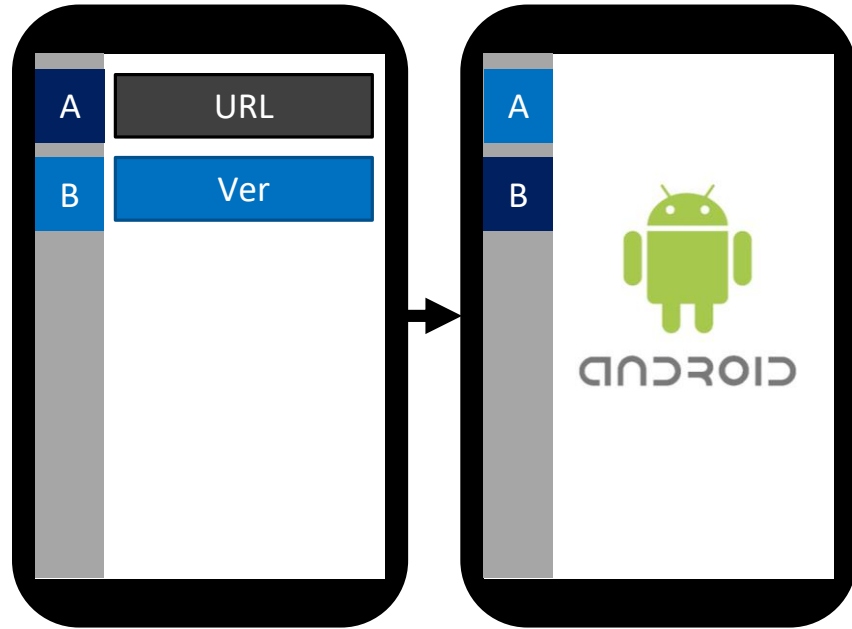
Ejercicio aplicaciones

En el fragment A, encárgese de descargar la foto y asignar una ruta en el teléfono para guardarla.



Ejercicio aplicaciones

Este path envíelo del fragment A al fragment B, para que este finalmente lo cargue



Ejercicio aplicaciones

Pista!

No olvide declarar los permisos en el manifest para internet y lectura-escritura del almacenamiento

Tenga en cuenta que debe pedir los permisos de lectura-escritura en tiempo de ejecución.

```
ActivityCompat.requestPermissions(this, new String[]{  
    Manifest.permission.WRITE_EXTERNAL_STORAGE,  
    Manifest.permission.READ_EXTERNAL_STORAGE  
}, 11);
```

