

Anotacion AfterViews

La documentaci3n de esta anotaci3n la veremos en el siguiente enlace:

<https://github.com/androidannotations/androidannotations/wiki/Injecting-Views#afterviews>

La anotaci3n `@AfterViews` indica que se debe llamar al m3todo que tenga dicha anotaci3n despu3s de que se haya **realizado el binding de vistas**. Puedes anotar m3ltiples m3todos con `@AfterViews`.

El m3todo anotado `AfterInject` se iniciar3 despu3s del m3todo `onCreate`. La sucesi3n de ejecuci3n ser3 la siguiente:

```
\----- m3todo anotado con AfterExtras
  |----- m3todo anotado con AfterInject
    |----- onCreate
      |----- m3todo anotado con AfterView
        |---- onStart
          | - onResume
```

NOTA A TENER EN CUENTA PR3XIMAS CLASES: Cuando se llama a `onCreate()`, los campos `@ViewById` a3n no est3n configurados. Por lo tanto, puedes usar `@AfterViews` para escribir c3digo que dependa de las vistas.

```
@AfterViews
void updateTextWithDate() {
    // Do something
}
```

Don't forget that you should not use any `"@ViewById"` (we will see it in next classes) annotated object in `onCreate()`:

IMPORTANTE: No olvides que no se debe usar ning3n objeto anotado `@ViewById` (lo veremos en las siguientes clases) en `onCreate()` ya que no estar3 inicializado:

```
@EActivity(R.layout.main)
public class MyActivity extends Activity {

    @ViewById
    TextView myTextView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // DON'T DO THIS !! It will throw a NullPointerException, myText
        // myTextView.setText("Date: " + new Date());
    }
```

Puedes asumir con seguridad que los campos `@ViewById`, `@Extra`, etc. ya están inicializados en los métodos anotados con `@AfterViews`.