


Open · Opened 3 weeks ago by  **Rubén Montero**

Ciclo de vida: onCreate

Resumen

- Introduciremos el concepto de actividad y distinguiremos XML y Java
- Hablaremos brevemente del método `onCreate`
- Mostraremos un *Toast* cuando se crea nuestra actividad

Descripción

La experiencia de usuario con aplicaciones móviles es distinta a las aplicaciones de escritorio. En una *app* móvil no se puede esperar que haya un único punto de entrada. Por ejemplo, si abres la aplicación de Gmail verás tu bandeja de entrada. Pero si seleccionas un PDF desde el explorador de archivos y usas *Compartir con* > *Gmail*, aparecerá directamente la ventana para redactar un nuevo correo con el archivo adjunto.

Las [actividades](#) son el elemento central de la programación Android y facilitan este paradigma.

Debemos entender una *actividad* como una *pantalla*, digamos, un *lienzo* donde nuestra aplicación puede pintar la interfaz de usuario. La mayoría de aplicaciones tienen varias pantallas, por lo que tienen varias actividades.

Hay *tres* piezas que construyen una actividad:

1) Archivo de interfaz XML

Como `activity_main.xml` con el que hemos estado trabajando. Suelen comenzar con `activity_` y luego llevar en `snake_case` el nombre de la actividad.

2) Declaración en el manifiesto XML

El [archivo de manifiesto](#) Android es un único archivo XML que tiene cada aplicación. En él se declaran cosas como el *nombre* de la aplicación, el *icono*, o las *actividades*.

Para declarar una actividad, basta con que se incluya en el sitio adecuado (dentro de `<application>`) una etiqueta como:

```
<activity android:name=".ExampleActivity" />
```

Si echas un vistazo al archivo `app > manifests > AndroidManifest.xml` de nuestra aplicación, verás [cómo está declarada la \(única\) actividad de la aplicación](#).

Observa que tiene un par de cosas adicionales. Entre ellas, `android.intent.category.LAUNCHER` sirve para indicar que es la actividad *inicial*.

3) Clase Java

Asumiendo cierto grado de conocimiento de [programación orientada a objetos \(POO\)](#) diremos que:

- Cada actividad tiene una clase asociada. Esta clase debe *heredar* (ser una subclase) de `Activity` (o `AppCompatActivity`).
- Por ello, el nombre de la clase suele llevar el sufijo `Activity`, e.g.: `MainActivity`.
- Nosotros **no** instanciamos nuestra clase. El sistema Android se encarga de ello.
- Nuestra responsabilidad es *sobreescribir* (`@Override`) los **métodos** que responden a eventos.

¿Métodos que responden a eventos?

Sí.

El método principal es `onCreate`. Se trata de un método que **se va a ejecutar una (1) sólo vez** cuando nuestra actividad sea creada.

La tarea

Abre `app > java > com.afundacion.fp.library > MainActivity.java`. Puedes ver la declaración de una clase, como habíamos prometido, y, además, hay un método llamado `onCreate`:

```
public class MainActivity extends AppCompatActivity {  
  
    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
}
```

Dentro de este método hay un par de líneas típicas que sirven para inicializar la actividad.

A continuación de esas dos líneas, **añade** una línea que sirva para lanzar un `Toast` ¹:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Vamos a lanzar un Toast
    Toast.makeText() ??
}
```

El método estático `makeText` recibe tres parámetros:

- Un `Context`. Esta clase es esencial en muchos métodos Android. Basta saber que una `Activity` *también* es un `Context`, así que podemos enviar `this`.
- Un texto en `String`. Nosotros usaremos `"¡Primera tostada del día!"`.
- Una constante que indique si se mostrará un tiempo corto o largo. Sólo vale `Toast.LENGTH_SHORT` y `Toast.LENGTH_LONG`.

Esto devuelve un objeto tipo `Toast` sobre el que podemos invocar `.show()` para que se muestre:

```
Toast oneToast = Toast.makeText(this, "¡Primera tostada del día!", Toast.LENGTH_SHORT);
oneToast.show();
```

Por brevedad suele escribirse todo en una línea:

```
Toast.makeText(this, "¡Primera tostada del día!", Toast.LENGTH_SHORT).show();
```

¡Adelante! **Añade** esta tostada en ~~nuestro desayuno~~ nuestra aplicación.


```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Vamos a lanzar un Toast
    Toast.makeText(this, "¡Primera tostada del día!", Toast.LENGTH_SHORT).show();
}
```

¡Enhorabuena! Has programado tu primera *tostada*.

Prueba a ejecutar tu *app* en el emulador. ¿Qué ves nada más arranca?

Por último

Sube tus cambios al repositorio en un nuevo *commit*.

1. Un `Toast` es un pequeño elemento visual típico de Android que salta momentáneamente en la pantalla, como una tostada. Es útil para verificar que el código se está ejecutando. 



Rubén Montero @ruben.montero changed milestone to [%Sprint 1](#) 3 weeks ago



Ania Blanco @ania.blanco mentioned in commit [5581a82a](#) 1 week ago