



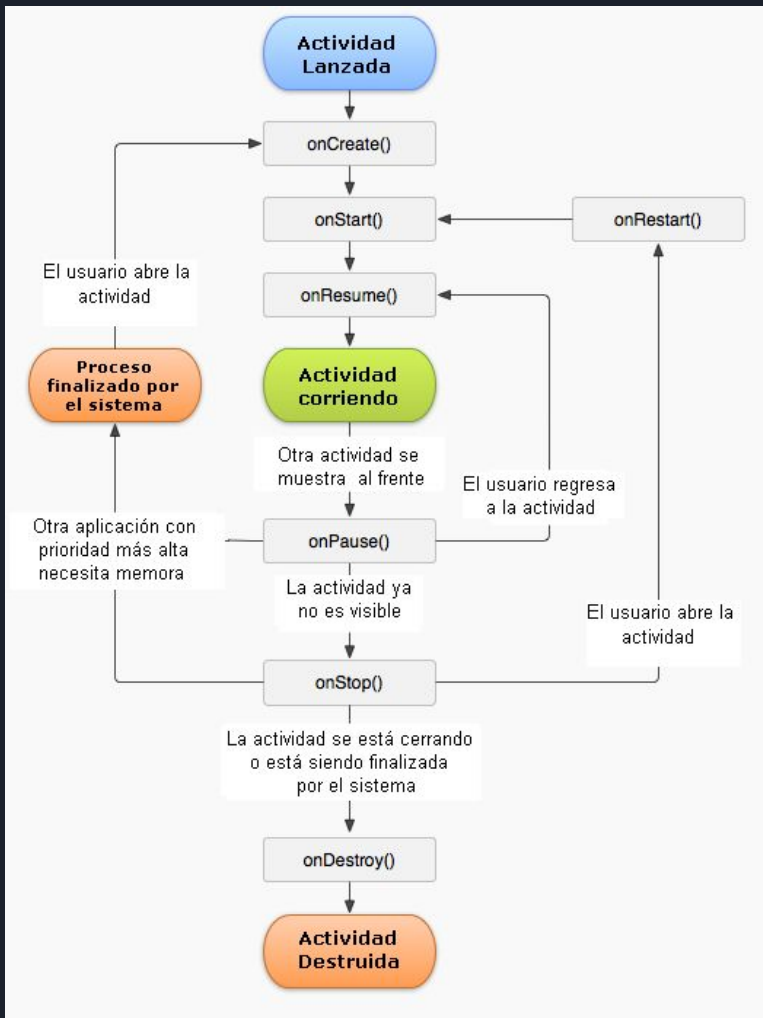
HLC

2° DAW



FAMILIARIZARNOS CON LAS ACTIVITY

Estudiamos el ciclo de vida de una Activity



- **onCreate():** Configura la actividad por primera vez.
- **onStart():** Hace que la actividad sea visible.
- **onResume():** La actividad pasa al primer plano e interactúa con el usuario.
- **onPause():** La actividad pierde el foco.
- **onStop():** La actividad ya no es visible.
- **onDestroy():** Libera todos los recursos antes de que la actividad sea destruida.

Información más ampliada:

<https://developer.android.com/guide/components/activities/activity-lifecycle.html#es-419>

Ciclo de vida de las Activity

Vayamos de nuevo a nuestra vista lógica en Android Studio.

```
MainActivity.java x
1 package com.daw.miprimerapp;
2
3 > import ...
10
11 <> public class MainActivity extends AppCompatActivity {
12
13     @Override
14     @ protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
17         setContentView(R.layout.activity_main);
18         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
19             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
20             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
21             return insets;
22         });
23     }
24 }
25
```

Como podemos comprobar, ya está creado el método `onCreate()`

Ciclo de vida de las Activity

Insertamos los siguientes métodos a continuación del método onCreate() en nuestra clase MainActivity.java

```
@Override
protected void onStart() {
    super.onStart();
    Toast.makeText(this, "OnStart", Toast.LENGTH_SHORT).show();
    // La actividad está a punto de hacerse visible.
}

@Override
protected void onResume() {
    super.onResume();
    Toast.makeText(this, "OnResume", Toast.LENGTH_SHORT).show();
    // La actividad se ha vuelto visible (ahora se "reanuda").
}

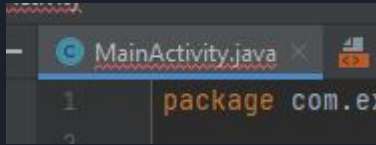
@Override
protected void onPause() {
    super.onPause();
    Toast.makeText(this, "OnPause", Toast.LENGTH_SHORT).show();
    // Enfocarse en otra actividad (esta actividad está a punto de ser "detenida").
}

@Override
protected void onStop() {
    super.onStop();
    Toast.makeText(this, "OnStop", Toast.LENGTH_SHORT).show();
    // La actividad ya no es visible (ahora está "detenida")
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Toast.makeText(this, "OnDestroy", Toast.LENGTH_SHORT).show();
    // La actividad está a punto de ser destruida.
}
```

Ciclo de vida de las Activity

Comprobamos que tras insertar el código, nos salta un error en nuestra clase, eso es porque debemos importar `Toast`

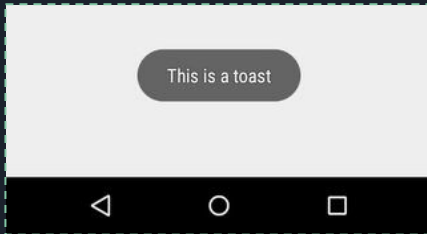


```
import android.widget.Toast;
```

Mensajes emergentes con Toast

Un **toast** es un mensaje que se muestra en pantalla durante unos segundos al usuario para luego volver a desaparecer automáticamente sin requerir ningún tipo de actuación por su parte, y sin recibir el foco en ningún momento (o dicho de otra forma, sin interferir en las acciones que esté realizando el usuario en ese momento).

Su utilización es muy sencilla. Esta clase dispone de un método estático **makeText()** al que deberemos pasar como parámetro el contexto de la actividad, el texto a mostrar, y la duración del mensaje, que puede tomar los valores **LENGTH_LONG** o **LENGTH_SHORT**, dependiendo del tiempo que queramos que la notificación aparezca en pantalla. Tras obtener una referencia al objeto Toast a través de este método, ya sólo nos quedaría mostrarlo en pantalla mediante el método **show()**.



```
Toast.makeText(this, "Suspense", Toast.LENGTH_LONG).show();
```

Ciclo de vida de las Activity

Más adelante veremos la funcionalidad de `Toast`, pero una de las cosas que nos aclara nuestro código, es que nos podrá mostrar el estado actual del ciclo de vida que está ejecutando en cada momento.

```
13      setContentView(R.layout.activity_main);
14  }
15
16  @Override
17  protected void onStart() {
18      super.onStart();
19      Toast.makeText( context: this, text: "OnStart", Toast.LENGTH_SHORT).show();
20      // La actividad está a punto de hacerse visible.
21  }
22  @Override
23  protected void onResume() {
24      super.onResume();
25      Toast.makeText( context: this, text: "OnResume", Toast.LENGTH_SHORT).show();
26      // La actividad se ha vuelto visible (ahora se "reanuda").
27  }
28  @Override
29  protected void onPause() {
30      super.onPause();
31      Toast.makeText( context: this, text: "OnPause", Toast.LENGTH_SHORT).show();
32      // Enfocarse en otra actividad (esta actividad está a punto de ser "detenida").
33  }
```


Ciclo de vida de las Activity

Vamos a añadir este par de líneas al método `onCreate()`, para que también podamos visualizar por pantalla cuando estamos en ese punto de la vida de la activity.

No olvidemos modificar el texto:

```
14      @Override
15      protected void onCreate(Bundle savedInstanceState) {
16          super.onCreate(savedInstanceState);
17          EdgeToEdge.enable( $this$enableEdgeToEdge: this);
18          setContentView(R.layout.activity_main);
19          ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
20              Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
21              v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
22              return insets;
23          });
24          Toast.makeText( context: this, text: "OnCreate", Toast.LENGTH_SHORT).show();
25          // La actividad está creada.
26      }
```

Ciclo de vida de las Activity

- Ejecutamos, o emulamos, ahora nuestra App y comprobamos cómo se muestra por pantalla cada uno de los eventos.

OnCreate

OnStart

OnResume



- Si minimizamos: onPause, onStop. Quiere decir que aún está funcionando
- Abrimos otra App y dejamos la nuestra en segundo plano.
- La ponemos de nuevo en primer plano y observamos los mensajes Toast
- La cerramos

Debuggeo de aplicaciones en Android

Un **BreakPoint** es un punto de corte, que en programación son unas líneas/s específicas/s en la cual queremos que se detenga el flujo habitual de programa.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    [...]
    int matematicas = 5;
    int quimica = 5;
    int fisica = 5;
    int resultado = 0;

    resultado = (matematicas + quimica + fisica) / 3;

    if(resultado >=5){
        Toast.makeText(this, "Aprobado", Toast.LENGTH_LONG).show();
    } else if(resultado <5){
        Toast.makeText(this, "Suspendido", Toast.LENGTH_LONG).show();
    }
}
```

