

Android Developer Fundamentals V2

# Actividades e Intentos

Lección 2



# 2.2 Ciclo de vida de la actividad y estado

# Contenidos

- Ciclo de vida de la Actividad
- Callbacks en el ciclo de vida de las Actividades
- Estado de la instancia Actividad
- Guardando y restaurando el estado de la actividad

# Ciclo de vida de la Actividad

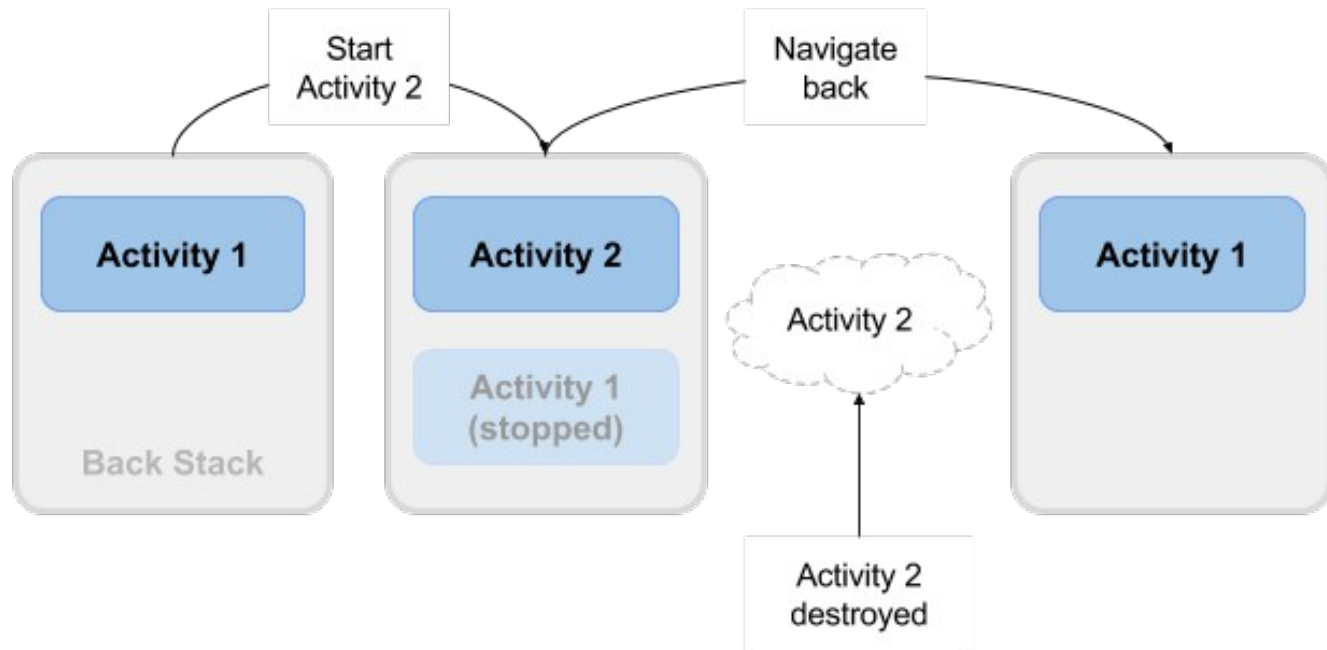
# ¿Qué es el ciclo de vida de la actividad?

- El conjunto de estados en los que puede estar una actividad durante su vida útil, desde que se crea hasta que se destruye

Más formalmente:

- Un diagrama de todos los estados en que puede estar una actividad, y las callbacks asociadas con la transición de cada estado al siguiente

# ¿Qué es el ciclo de vida de la actividad?



# Estados de actividad y visibilidad de la aplicación

- Creada (no visible)
- Iniciada (visible)
- Reanudar (visible)
- Pausada (parcialmente invisible)
- Detenida (oculta)
- Destruída (removida de la memoria)

Los cambios de estado se activan por acción del usuario, cambios de configuración como la rotación del dispositivo o acción del sistema

# Callbacks en el ciclo de vida de la Actividad



# Callbacks y cuando se llaman

**onCreate(Bundle savedInstanceState)**—inicialización estática

**onStart()**—cuando la actividad (pantalla) se hace visible

**onRestart()**—se llama si se detuvo la actividad (se llama onStart())

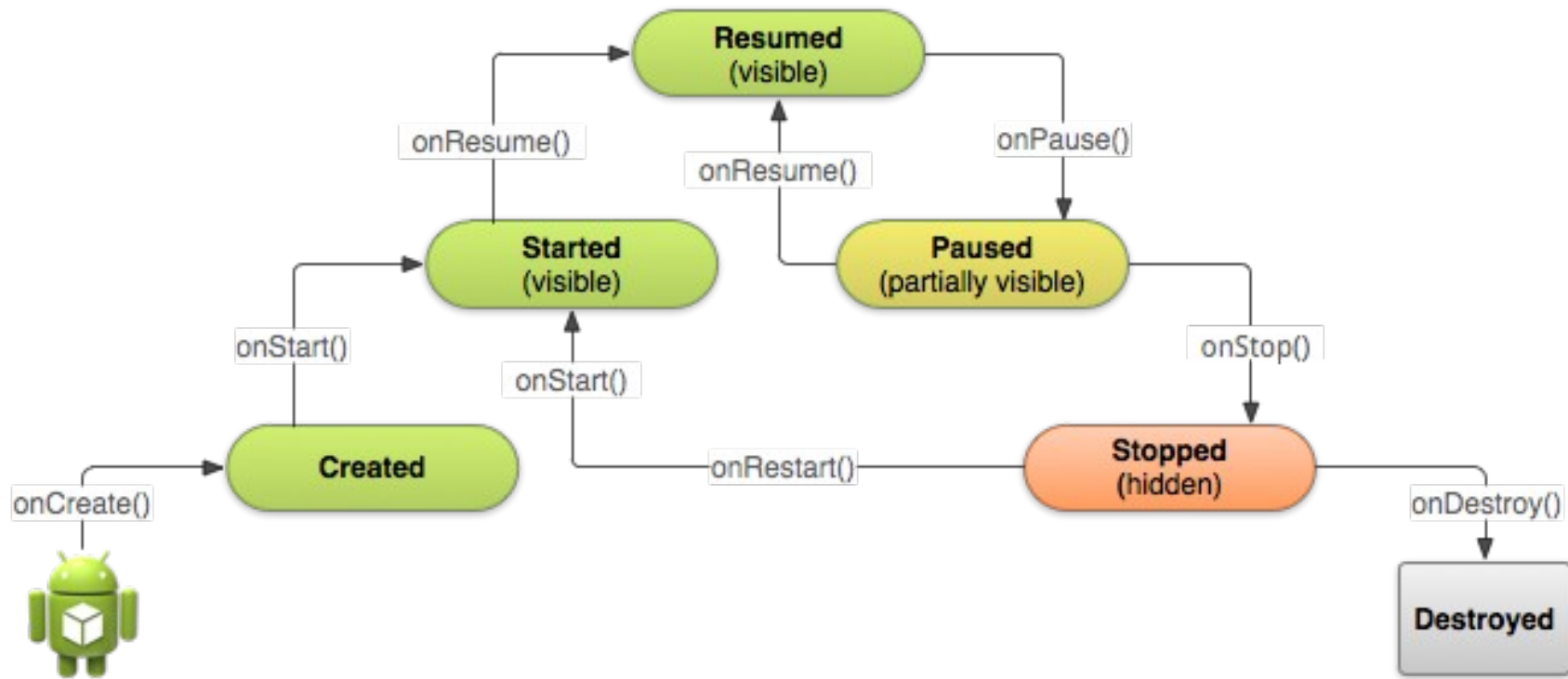
**onResume()**—empieza a interactuar con el usuario

**onPause()**—a punto de reanudar la actividad ANTERIOR

**onStop()**—ya no es visible, pero aún existe y se conserva toda la información del estado

**onDestroy()**—llamada final antes de que el sistema Android destruya la actividad

# Gráfico de estados de la Actividad y callbacks



# Implementar y anular llamadas callback

- Solo se requiere onCreate()
- Reemplace las otras callbacks para cambiar el comportamiento predeterminado

# onCreate() -> Creado

- Se llama cuando se crea la Actividad por primera vez, por ejemplo, cuando el usuario pulsa el icono de lanzador (launcher)
- Realiza la configuración estática: crea vistas, vincula datos a listas, ...
- Se llama una vez durante la vida de una Actividad
- Toma un Bundle con el estado previamente guardado de la Actividad, si hubiera un
- El estado “creado” es siempre seguido por onStart ( )

# onCreate(Bundle savedInstanceState)

```
@Override  
public void onCreate(Bundle  
savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    // The activity is being created.  
}
```

# onStart() -> Iniciado

- Se llama cuando la actividad se hace visible para el usuario
- Se puede llamar más de una vez durante el ciclo de vida
- Seguido por onResume() si la actividad llega al primer plano, o onStop() si se oculta

# onStart()

```
@Override  
protected void onStart() {  
    super.onStart();  
    // The activity is about to become  
    visible.  
}
```

# onRestart() -> Iniciado

- Llamado después de que la Actividad se haya detenido, inmediatamente antes de que se inicie de nuevo
- Estado transitorio
- Always followed by onStart()



# onRestart()

```
@Override
protected void onRestart() {
    super.onRestart();
    // The activity is between stopped and
    // started.
}
```

# onResume() -> Reanudado / En ejecución

- Llamado cuando la Actividad comenzará a interactuar con el usuario
- La Actividad se ha movido a la parte superior de la pila de actividades
- Comienza a aceptar la entrada de usuario
- Estado en ejecución
- Siempre seguido de onPause()

# onResume()

```
@Override  
protected void onResume() {  
    super.onResume();  
    // The activity has become visible  
    // it is now "resumed"  
}
```

# onPause() -> En pausa

- Invocado cuando el sistema está a punto de reanudar una actividad anterior
- La Actividad es parcialmente visible, pero el usuario está abandonando la Actividad
- Normalmente se usa para confirmar cambios no guardados en datos persistentes, detener animaciones y cualquier cosa que consuma recursos
- Las implementaciones debe ser rápidas porque la siguiente actividad no se reanuda hasta que este método regrese
- Seguido por onResume() si la Actividad vuelve al frente, o onStop() si se vuelve invisible para el usuario

# onPause()

```
@Override  
protected void onPause() {  
    super.onPause();  
    // Another activity is taking focus  
    // this activity is about to be "paused"  
}
```

# onStop() -> Detenido

- Se llama cuando la actividad ya no es visible para el usuario
- Se está iniciando una nueva Actividad, una existente se pone delante de esta, o esta está siendo destruida
- Operaciones que eran demasiado pesadas para onPause()
- Seguido por onStart() si la Actividad está regresando para interactuar con el usuario, o onDestroy() si la Actividad está desapareciendo

# onStop()

```
@Override  
protected void onStop() {  
    super.onStop();  
    // The activity is no longer visible  
    // it is now "stopped"  
}
```

# onDestroy() -> Destruida

- Llamada final antes de que la actividad sea destruida
- El usuario navega de nuevo a la Actividad anterior, o cambios de configuración
- La Actividad está terminando o el sistema la está destruyendo para ahorrar espacio
- Llame el método isFinishing() para verificar
- El sistema puede destruir la Actividad sin llamar a esto, así que use onPause() o onStop() para guardar los datos o el estado



# onDestroy()

```
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    // The activity is about to be destroyed.  
}
```

# Instancia de Estado de la Actividad

# ¿Cuándo cambia la configuración?

Los cambios de configuración invalidan el diseño actual u otros recursos en su actividad cuando el usuario:

- Gira el dispositivo
- Elige diferentes idiomas del sistema, por lo cual, cambia la configuración regional
- Entra en modo multi-ventana (desde Android 7)

# ¿Qué sucede en el cambio de configuración?

En cambio de configuración, Android:

**1.** Apaga la Actividad  
llamando a:

- onPause()
- onStop()
- onDestroy()

**2.** Inicia la Actividad  
nuevamente invocando a:

- onCreate()
- onStart()
- onResume()

# Instancia de Estado de la Actividad

- La información de estado se crea mientras la actividad se está ejecutando, como un contador, el texto del usuario y la progresión de la animación
- El estado se pierde cuando se gira el dispositivo, se cambia el idioma, se presiona el botón de retroceso o el sistema borra la memoria

# Guardando y restaurando el estado de la Actividad

# ¿Qué guarda el sistema?

- El sistema guarda solo:
  - Estado de las vistas con ID única (`android:id`) como el texto ingresado en `EditText`
  - El Intento que inició la Actividad y datos en sus extras
- Ud es responsable de guardar otras actividades y datos de progreso de usuario

# Guardando la instancia de Estado

Implemente `onSaveInstanceState()` en su actividad

- Invocado por el motor de ejecución de Android cuando existe la posibilidad de que la Actividad pueda ser destruida
- Guarda datos solo para esta instancia de la actividad durante la sesión actual



# onSaveInstanceState(Bundle outState)

```
@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    // Add information for saving HelloToast counter
    // to the to the outState bundle
    outState.putString("count",
        String.valueOf(mShowCount.getText()));
}
```

# Recuperando la Instancia de Estado

Dos formas de recuperar el paquete guardado

- en `onCreate(Bundle mySavedState)`  
Preferido, para garantizar que su interfaz de usuario, incluido cualquier estado guardado, se vuelva a ejecutar y se ejecute lo más rápido posible
- Implementar el callback (invocado después de `onStart()`)  
[onRestoreInstanceState\(Bundle mySavedState\)](#)

# Restoring in onCreate()

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mShowCount = findViewById(R.id.show_count);

    if (savedInstanceState != null) {
        String count = savedInstanceState.getString("count");
        if (mShowCount != null)
            mShowCount.setText(count);
    }
}
```

# onRestoreInstanceState(Bundle state)

```
@Override
public void onRestoreInstanceState (Bundle mySavedState) {
    super.onRestoreInstanceState(mySavedState);

    if (mySavedState != null) {
        String count = mySavedState.getString("count");
        if (count != null)
            mShowCount.setText(count);
    }
}
```

# Instancia de Estado y reinicio de la aplicación

Cuando detiene y reinicia una nueva sesión de aplicación, los estados de la instancia de Actividad se pierden y sus actividades volverán a su aspecto predeterminado

Si necesita guardar datos de usuario entre sesiones de aplicaciones, use preferencias compartidas o una base de datos.

# FIN