



**UNIVERSIDAD CENTRAL DEL ECUADOR
FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICA
INGENIERÍA INFORMÁTICA**

DISPOSITIVOS MÓVILES



CONSULTA # 3

INTEGRANTES: Campaña Ximena

FECHA: 14/05/2019

DESCRIPCIÓN DE LA TAREA

GUARDANDO EL ESTADO DE UNA ACTIVIDAD

Cuando el usuario ha estado utilizando una actividad, y tras cambiar a otras, regresa a la primera, lo habitual es que esta permanezca en memoria y continúe su ejecución sin alteraciones. Como se ha explicado, en situaciones de escasez de memoria, es posible que el sistema haya eliminado el proceso que ejecutaba la actividad. En este caso, el proceso será creado de nuevo, pero se habrá perdido su estado, es decir, se habrá perdido el valor de sus variables y el puntero de programa. Como consecuencia, si el usuario estaba a mitad de un proceso de edición o estaba reproduciendo un audio en un punto determinado perderá esta información.

NOTA: Cuando se ejecuta una actividad sensible a la inclinación del teléfono, es decir puede verse en horizontal o en vertical, se presenta un problema similar al anterior. La actividad es destruida y vuelta a construir con las nuevas dimensiones de pantalla y por lo tanto se llama de nuevo al método `onCreate`. Antes de que la actividad sea destruida también resulta fundamental guardar su estado.

Android provee de dos métodos que se deben sobrescribir en un Activity `onSaveInstanceState` y `onRestoreInstanceState`

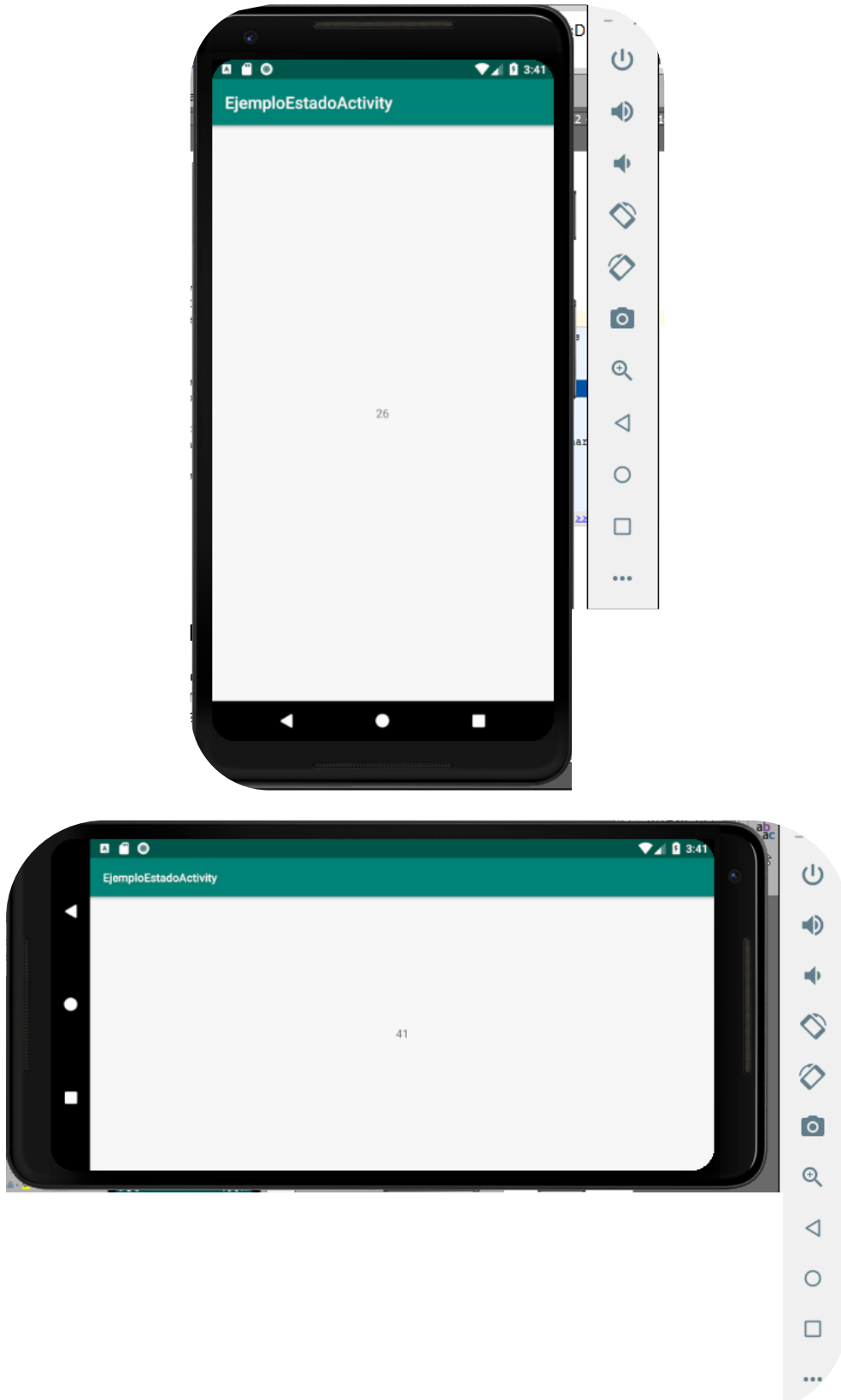
```
public void onSaveInstanceState(Bundle savedInstanceState) {}  
public void onRestoreInstanceState(Bundle savedInstanceState) {}
```

Se debe utilizar el Bundle que recibe como parámetro en ambos métodos.

```
@Override  
public void onSaveInstanceState(Bundle savedInstanceState) {  
    super.onSaveInstanceState(savedInstanceState);  
  
    // Save in savedInstanceState.  
    savedInstanceState.putStringArray(SPINNER_DATA, colors);  
}  
  
@Override  
public void onRestoreInstanceState(Bundle savedInstanceState) {  
    super.onRestoreInstanceState(savedInstanceState);  
  
    // Restore state from the savedInstanceState.  
    if(savedInstanceState != null) {  
        colors = savedInstanceState.getStringArray(SPINNER_DATA);  
        addColors(colors);  
    }  
}
```

Ejemplo

En las siguientes dos capturas se muestra la generación de números aleatorios. Se visualiza que al momento de girar la pantalla el número aleatorio cambia, mostrando así que no se está guardando el estado del activity.



Al agregar el método **onSaveInstanceState** en el archivo Main se guarda el estado del activity.

```
package com.example.ejemploestadoactivity;

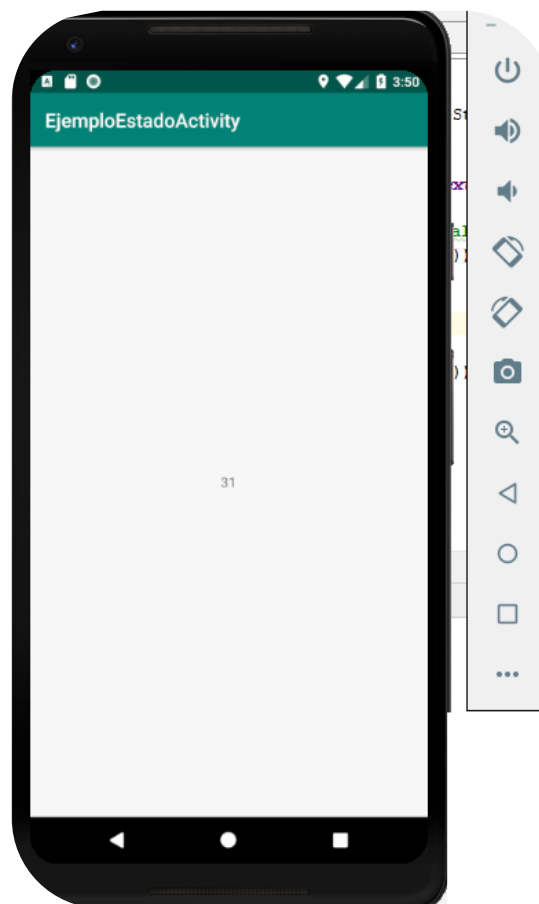
import ...

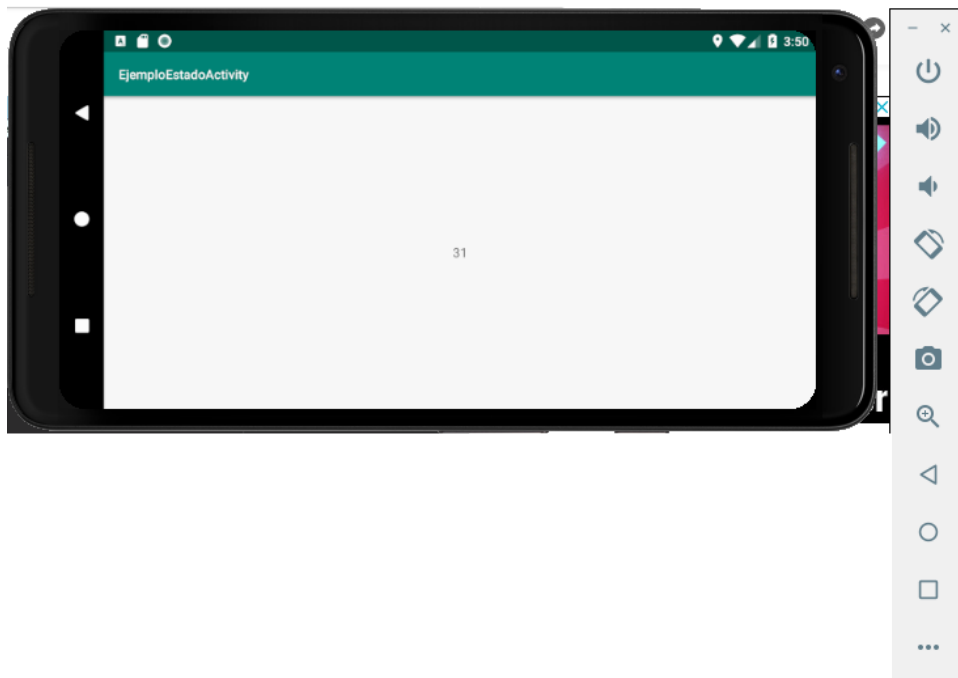
public class MainActivity extends AppCompatActivity {
    TextView aleatorio;
    private int num;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        aleatorio=(TextView)findViewById(R.id.textAleatorio);
        if(savedInstanceState != null){
            num=savedInstanceState.getInt( key: "aleatorio");
            aleatorio.setText(String.valueOf(num));
        }
        else
        {
            num=generarAleatorio( numero: 50);
            aleatorio.setText(String.valueOf(num));
        }
    }

    private int generarAleatorio(int numero){
        return (int) (Math.random()*numero);
    }

    @Override
    public void onSaveInstanceState(Bundle outState) {
        outState.putInt("aleatorio",num);
        super.onSaveInstanceState(outState);
    }
}
```





CONCLUSIONES

Mantener el estado de un activity puede ser de mucha utilidad en varios casos como cuando el sistema cuenta con poca memoria y debe eliminar procesos. Si un proceso es creado nuevamente, este mantendrá el estado, conservará el valor de sus variables.

Cuando el usuario ha estado utilizando una actividad, y tras cambiar a otras, regresa a la primera, lo habitual es que esta permanezca en memoria y continúe su ejecución sin alteraciones. Como se ha explicado, en situaciones de escasez de memoria, es posible que el sistema haya eliminado el proceso que ejecutaba la actividad. En este caso, el proceso será creado de nuevo, pero se habrá perdido su estado, es decir, se habrá perdido el valor de sus variables y el puntero de programa. Como consecuencia, si el usuario estaba a mitad de un proceso de edición o estaba reproduciendo un audio en un punto determinado perderá esta información.

RECOMENDACIONES

En varias de las aplicaciones móviles es importante guardar el estado de una actividad, como es el caso de mantener el usuario de una aplicación registrada en un próximo ingreso a la misma. La realización de aplicaciones usando los métodos `onSaveInstanceState` y `onRestoreInstanceState` ayudarán a mantener el valor de variables y de esta forma guardar del activity.

BIBLIOGRAFÍA

Experto. (22 de marzo de 2017). Recuperado el 11 de mayo de 2019, de <https://experto.dev/android-guardar-estado-activity/>

Universidad Politécnica de Valencia. (2017). Recuperado el 11 de mayo de 2019, de <http://www.androidcurso.com/index.php/recursos/37-unidad-6-multimedia-y-ciclo-de-vida/163-guardando-el-estado-de-una-actividad>