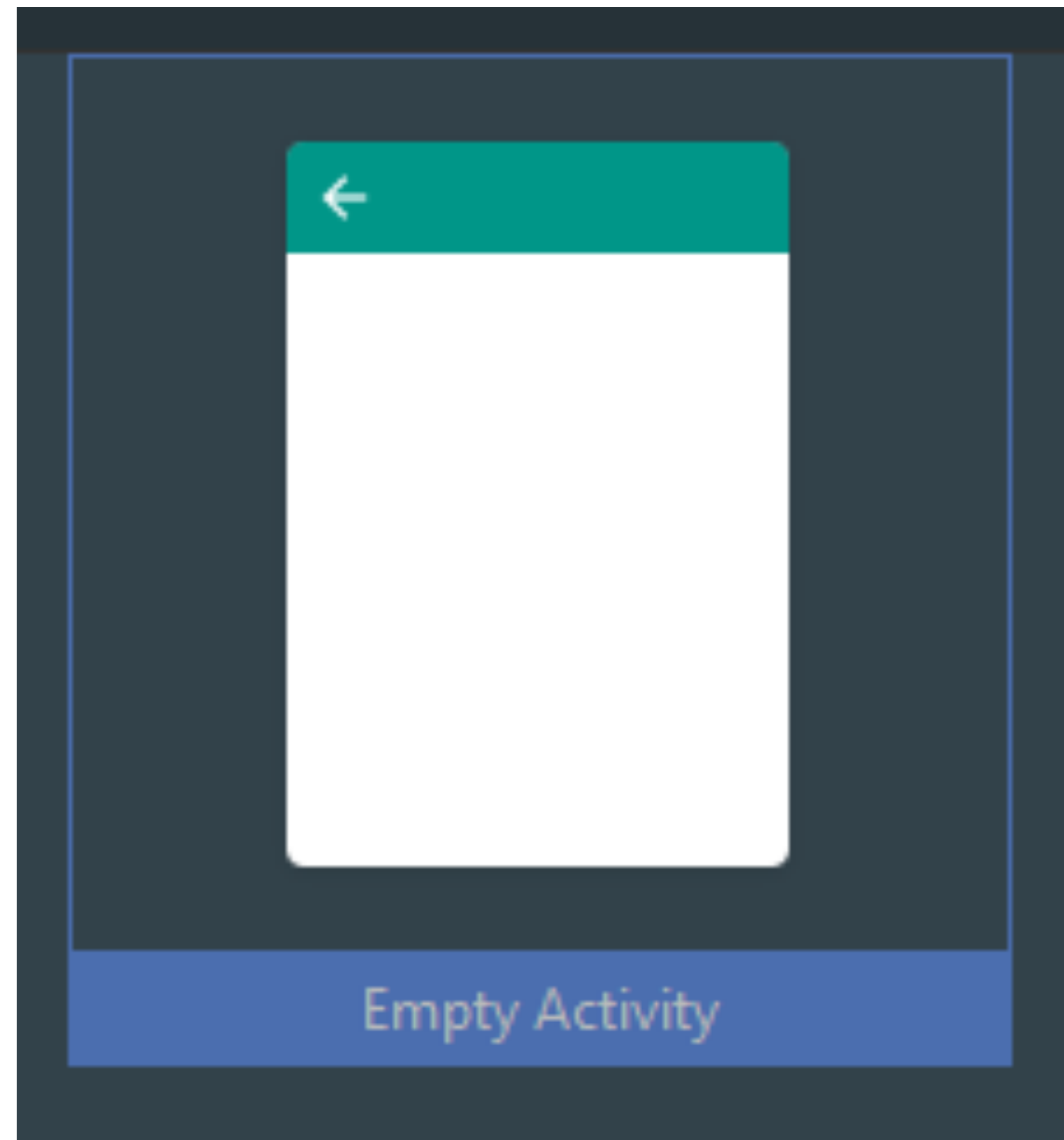




android

Desarrollo de
Aplicaciones Móviles

Actividades

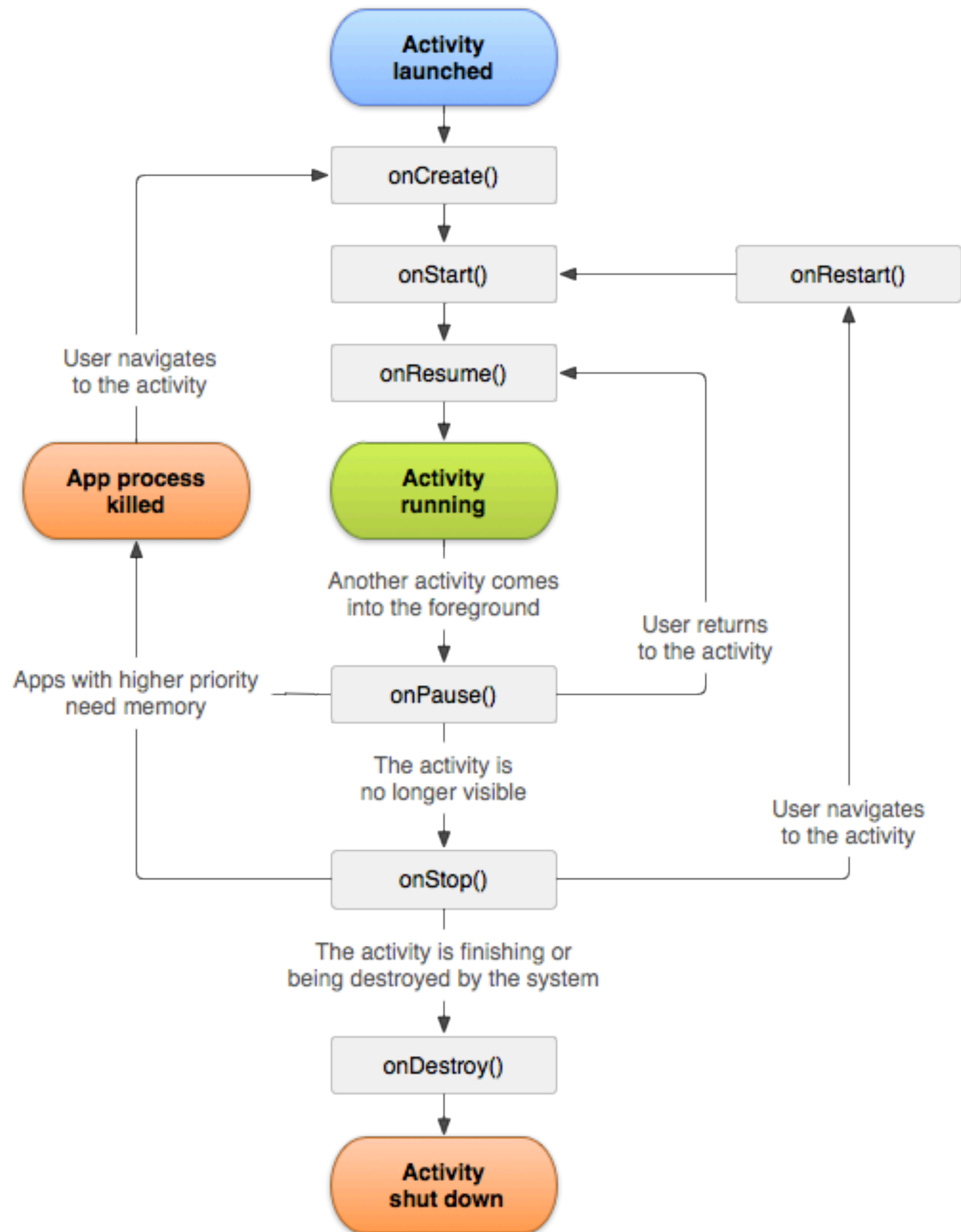


Una activity (actividad en español), representa una pantalla con interfaz de usuario. Por ejemplo, una aplicación de correo electrónico tiene una actividad que muestra una lista de los correos electrónicos nuevos, otra actividad para redactar el correo electrónico.



Ciclo de Vida

Cuando un usuario navega por una aplicación, se realiza diferentes transiciones entre actividades, cada actividad pasa por diferentes estados dentro de su ciclo de vida. La clase Activity provee diferentes métodos que son invocados lo cual le permite saber a la actividad saber que su estado a cambiado.



Ciclos de Vida

`onCreate():`

Este método es invocado cuándo la actividad esta comenzando, aquí realizamos el llamado para inflar la vista de la actividad o inicializar varibales.

`onStart():`

Este método es invocado después de `onCreate()` o después de `onRestart()` cuándo la aplicación se ha detenido, pero esta siendo de nuevo mostrada, es el encargado de hacer la actividad visible al usuario.

`onResume():`

Este método es invocado después de `onStart()` o después de `onPause()`, y es el encargado de la interacción con el usuario, la app se mantiene en este estado hasta que algo le quita el enfoque, ese algo podría ser apagar la pantalla, recibir una llamada o cambiar de actividad.

Ciclos de Vida

`onPause()`:

El sistema llama a este método como primer indicador de que el usuario esta dejando tu actividad, aunque esto no siempre significa que tu actividad está siendo destruida, aquí podemos parar animaciones o cualquier otra cosa que no deba de continuar cuándo la app no está enfocada.

`onStop()`:

Este método es invocado cuándo la actividad ya no es visible al usuario, esto puede suceder por ejemplo cuando una nueva actividad cubre completamente la anterior, así mismo el sistema también puede llamar a este método cuándo tu actividad esta a punto de ser destruida.

Ciclos de Vida

`onRestart()`:

Este método es invocado cuando la actividad está siendo mostrada nuevamente al usuario debido a que el usuario navega de regreso a ella.

`onDestroy()`:

Es la encargada de realizar la limpieza de los recursos usados por la aplicación antes de que esta sea terminada, es invocado por que explícitamente se llama `finish()` o porque el sistema esta temporalmente destruyendo la instancia para salvaguardar recursos. Este método no debe de ser usado para guardar datos, ya que no existe garantía de que sea invocado por el sistema.

Manifiesto

Todas las aplicaciones deben tener un archivo AndroidManifest.xml (con ese nombre exacto) en el directorio raíz. El archivo de manifiesto proporciona información esencial sobre tu aplicación al sistema Android, información que el sistema debe tener para poder ejecutar el código de la app.

Todas las actividades deben estar representadas por elementos <activity> en el archivo de manifiesto. El sistema no verá las que no estén declaradas en el archivo y nunca se ejecutarán.

```
<activity android:name=".MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```


Layout

Un diseño define la estructura visual para una interfaz de usuario, como la IU para una actividad o widget de una app. Puedes declarar un diseño de dos maneras:

Declarar elementos de la IU en XML. Android proporciona un vocabulario XML simple que coincide con las clases y subclases de vistas, como las que se usan para widgets y diseños.

Crear una instancia de elementos del diseño en tiempo de ejecución. Tu aplicación puede crear objetos View y ViewGroup (y manipular sus propiedades) programáticamente.

El framework de Android te ofrece la flexibilidad de usar uno de estos métodos o ambos para declarar y administrar la IU de tu aplicación. Por ejemplo, podrías declarar los diseños predeterminados de la aplicación en XML, incluidos los elementos de pantalla que aparecerán en ellos y sus propiedades. Luego podrías agregar código en tu aplicación para modificar el estado de los objetos de la pantalla, incluidos los declarados en XML, en tiempo de ejecución.

Intent

Un Intent es un objeto de acción que puedes usar para solicitar una acción de otro componente de la aplicación. Aunque las intents facilitan la comunicación entre los componentes de muchas maneras, existen tres casos de uso fundamentales:

Para comenzar una actividad:

Una Activity representa una única pantalla en una aplicación. Puedes iniciar una nueva instancia de una Activity pasando una Intent a `startActivity()`.

Para iniciar un servicio:

Un Service es un componente que realiza operaciones en segundo plano sin una interfaz de usuario.

Para entregar un mensaje:

Un mensaje es un aviso que cualquier aplicación puede recibir. El sistema entrega varios mensajes de eventos del sistema, como cuando el sistema arranca o el dispositivo comienza a cargarse.

Bundle

Un bundle se ocupa para pasar datos a futuro entre actividades en Android, aunque también es utilizado por ejemplo cuando la pantalla del dispositivo gira, si esto pasa el método `onSaveInstanceState(Bundle outstate)` es invocado y la actividad es destruida, posteriormente otra instancia de la actividad es creada y el método `onCreate(Bundle savedInstanceState)` es llamado, cuando la primera instancia de la actividad es creada el bundle es null, si salvamos los estados este bundle ya no es null y podemos restaurar lo que sea que se haya guardado.

Android automáticamente guarda los textos al girar la pantalla, pero no guarda todo, así corresponde a nosotros guardar y restaurar estas variables de manera correcta.

Salvando Datos

`onSaveInstanceState(Bundle outState):`

Cuando tu actividad empieza a pararse, el sistema llama a `onSaveInstanceState()` para que tu actividad pueda guardar la información de estado en una colección de pares clave-valor. La implementación por defecto de este método guarda información acerca del estado de la jerarquía de vistas de la actividad, como el texto en un control `EditText` o la posición de desplazamiento en un `ListView`.

Restaurando Datos

`onCreate(Bundle savedInstanceState):`

Cuando tu actividad se recrea después de haberse destruido previamente, puedes recuperar el estado guardado desde el Bundle que el sistema pasa a tu actividad. Tanto el método de retrollamada `onCreate()` como `onRestoreInstanceState()` reciben el mismo Bundle que contiene la información de estado de la instancia.

Dado que el método `onCreate()` se llama tanto si el sistema está creando una nueva instancia de tu actividad como si está recreando una previa, debes comprobar si el Bundle de estado es nulo antes de intentar leerlo. Si es nulo, entonces el sistema está creando una nueva instancia de la actividad, en lugar de restaurar una anterior que había sido destruida.