



Instituto Politécnico Nacional

Omar Montoya Romero

7CM1

Desarrollo de Aplicaciones Móviles Nativas

Sistemas Computacionales

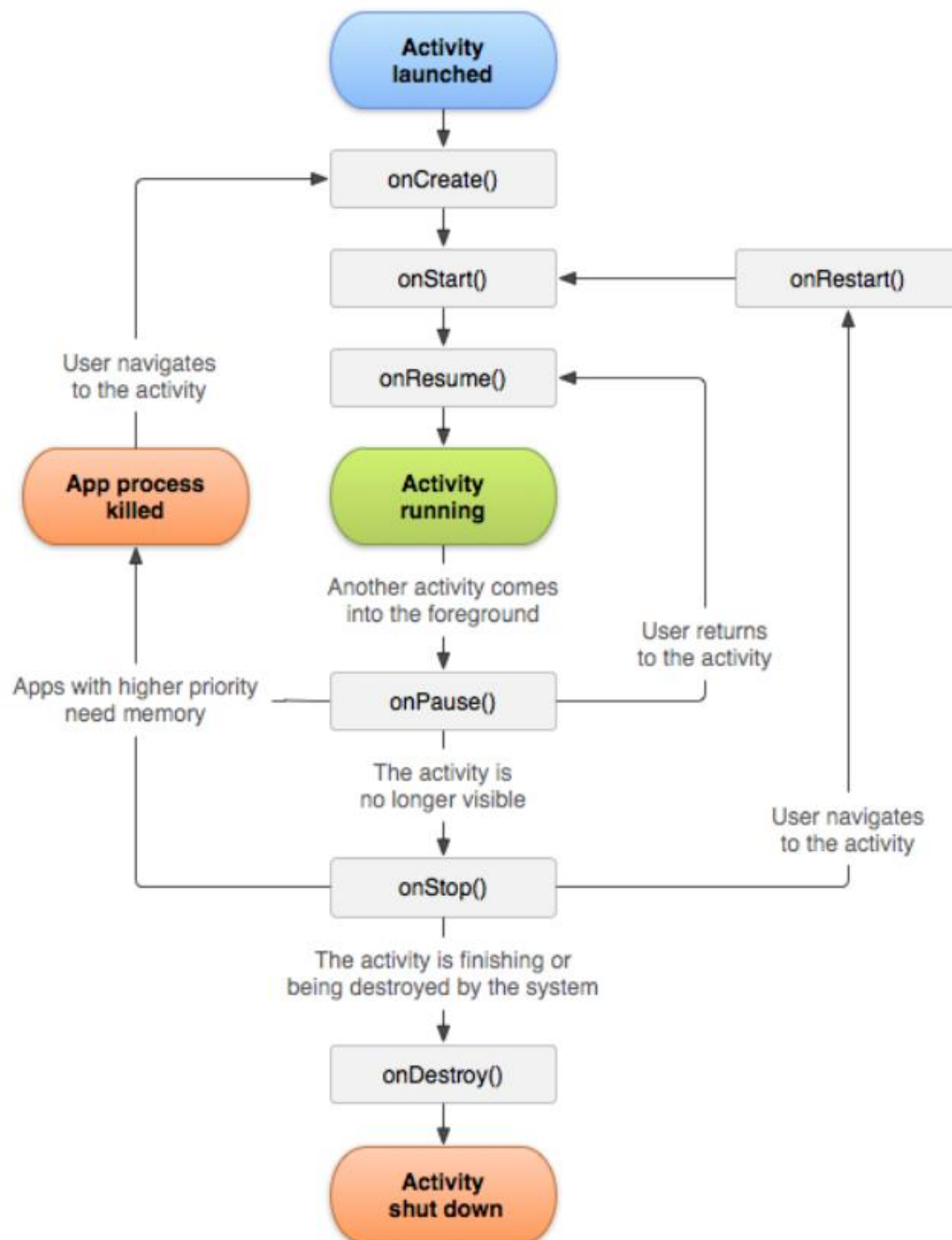
**Actividad II: Investigación sobre el Ciclo de Vida de una
Actividad**

Mtro. Efraín Arredondo Morales

19/09/2023

Conceptos de los ciclos de las actividades

Para navegar por las transiciones entre las etapas del ciclo de vida de una actividad, la clase Activity proporciona un conjunto básico de seis devoluciones de llamadas: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()` y `onDestroy()`. El sistema invoca cada una de estas devoluciones de llamada cuando una operación entra en un nuevo estado.



onCreate()

En el método `onCreate()`, ejecutas la lógica de arranque básica de la aplicación que debe ocurrir una sola vez en toda la vida de la actividad. Por ejemplo, tu implementación de `onCreate()` podría vincular datos a listas, asociar la actividad con un `ViewModel` y crear instancias de algunas variables de alcance de clase. Este método recibe el parámetro `savedInstanceState`, que es un objeto `Bundle` que contiene el estado ya guardado de la actividad. Si la actividad nunca existió, el valor del objeto `Bundle` es nulo.

onStart()

La llamada `onStart()` hace que el usuario pueda ver la actividad mientras la app se prepara para que esta entre en primer plano y se convierta en interactiva. Por ejemplo, este método es donde la app inicializa el código que mantiene la IU.

Cuando la actividad pase al estado `Started`, cualquier componente que priorice el ciclo de vida vinculado al de la actividad recibirá el evento `ON_START`.

El método `onStart()` se completa muy rápido y, al igual que con el estado `Created`, la actividad no permanece en el estado `Started`. Una vez finalizada esta devolución de llamada, la actividad entra en el estado `Resumed`, y el sistema invoca el método `onResume()`.

onResume()

Este es el estado en el que la app interactúa con el usuario. La app permanece en este estado hasta que ocurre algún evento que la quita de foco. Tal evento podría ser, por ejemplo, recibir una llamada telefónica, que el usuario navegue a otra actividad o que se apague la pantalla del dispositivo.

Cuando se reanude la actividad, cualquier componente que priorice el ciclo de vida vinculado al de la actividad recibirá el evento `ON_RESUME`. Aquí es donde los componentes del ciclo de vida pueden habilitar cualquier funcionalidad que necesite ejecutarse mientras el componente esté visible y en primer plano, como, por ejemplo, iniciar una vista previa de la cámara.

Cuando se produce un evento de interrupción, la actividad entra en el estado `Paused` y el sistema invoca la devolución de llamada `onPause()`.

onPause()

El sistema llama a este método a modo de primera indicación de que el usuario está abandonando tu actividad (aunque no siempre significa que está finalizando la actividad); esto indica que la actividad ya no está en primer plano (aunque puede seguir siendo visible si el usuario está en el modo multiventana). Utiliza el método `onPause()` para pausar o ajustar las operaciones que no deben continuar (o que deben continuar con moderación) mientras `Activity` se encuentra en estado `Paused` y que esperas reanudar en breve. Hay varias razones por las que una actividad puede entrar en este estado. Por ejemplo:

- Algunos eventos interrumpen la ejecución de la app, como se describe en la sección `onResume()`. Este es el caso más común.
- En Android 7.0 (API nivel 24) o versiones posteriores, varias apps se ejecutan en el modo multiventana. Debido a que solo una de las apps (ventanas) tiene foco en cualquier momento, el sistema pausa todas las demás.
- Se abre una nueva actividad semitransparente (como un diálogo). Mientras la actividad siga siendo parcialmente visible, pero no esté en foco, se mantendrá pausada.

onStop()

Cuando el usuario ya no puede ver tu actividad, significa que ha entrado en el estado `Stopped`, y el sistema invoca la devolución de llamada `onStop()`. Esto puede ocurrir, por ejemplo, cuando una actividad recién lanzada cubre toda la pantalla. El sistema también puede llamar a `onStop()` cuando haya terminado la actividad y esté a punto de finalizar.

Cuando la actividad pase al estado `Stopped`, cualquier componente que priorice el ciclo de vida vinculado al de la actividad recibirá el evento `ON_STOP`. Aquí es donde los componentes del ciclo de vida pueden detener cualquier funcionalidad que no necesite ejecutarse mientras el componente no sea visible en la pantalla.

En el método `onStop()`, la app debe liberar o ajustar los recursos que no son necesarios mientras no sea visible para el usuario. Por ejemplo, tu app podría pausar animaciones o cambiar de actualizaciones de ubicación detalladas a más generales. Usar `onStop()` en lugar de `onPause()` garantiza que continúe el trabajo relacionado con la IU, incluso cuando el usuario esté viendo tu actividad en el modo multiventana.

onDestroy()

Se llama a `onDestroy()` antes de que finalice la actividad. El sistema invoca esta devolución de llamada por los siguientes motivos:

- La actividad está terminando (debido a que el usuario la descarta por completo o a que se llama a `finish()`).
- El sistema está finalizando temporalmente la actividad debido a un cambio de configuración (como la rotación del dispositivo o el modo multiventana).

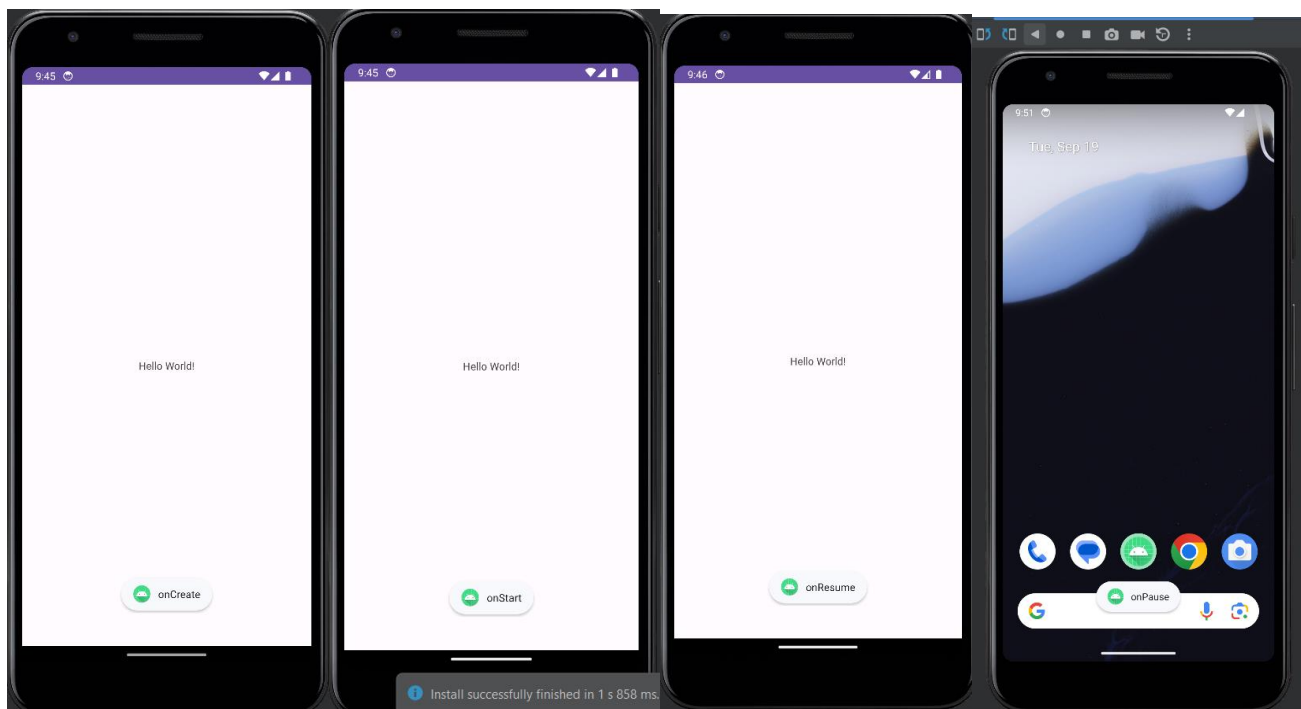
Cuando la actividad pase al estado `Destroyed`, cualquier componente que priorice el ciclo de vida vinculado al de la actividad recibirá el evento `ON_DESTROY`. Aquí es donde los componentes del ciclo de vida pueden recuperar cualquier elemento que se necesite antes de que finalice el objeto `Activity`.

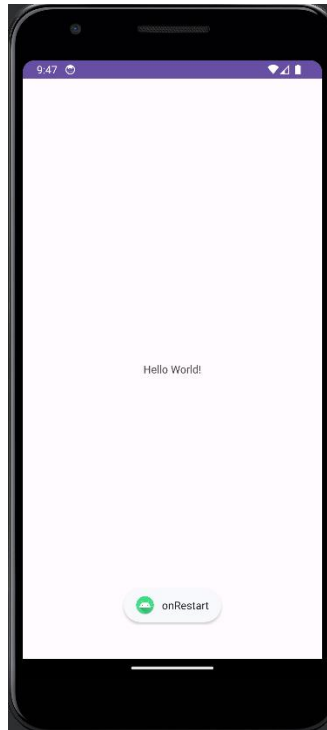
Estado de actividad y expulsión de memoria

Probabilidad de que finalice	Estado del proceso	Estado de la actividad
Menos	Primer plano (en foco o por estar en el)	Created Started Resumed
Mas	Segundo plano (foco perdido)	Pausado
Mayor	Segundo plano (no visible)	Detenido
	Vacío	Finalizado

El sistema nunca finaliza una actividad de forma directa para liberar memoria. En su lugar, finaliza el proceso en el que se ejecuta la actividad para eliminar no solo la actividad, sino también todo lo que se ejecuta en el proceso. [1]

Resultados de la implementación





Referencias

- [1] A. Developers, «Android Developers,» 23 06 2020. [En línea]. Available: <https://developer.android.com/guide/components/activities/activity-lifecycle?hl=es-419#saras>. [Último acceso: 19 09 2019].