



INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo



Actividades, intents y fragmentos en el desarrollo móvil

Grupo: 7CV3

Materia: Desarrollo de Aplicaciones
Móviles Nativas

Profesor: M. en C. Gabriel Hurtado Avilés



Actividades

Las actividades es el componente más importante en el desarrollo de las aplicaciones móviles Android, y lo podemos definir o ver como las interfaces que son la comunicación entre el sistema y los usuarios de estos sistemas móviles.

}

Se le puede decir actividad a cualquier interfaz dentro de la aplicación, y cada actividad puede contener funcionalidades lógicas diferentes[1].

Actividades

Término técnico

Siendo un poco más técnicos la actividad define en Android mediante la clase Activity, y su propósito es manipular y desarrollar la interfaz de usuario y responder a las interacciones del usuario, como toques, desplazamientos y gestos.

Las actividades son importantes para ofrecerle al usuario de la aplicación móvil , una interactividad más fluida y efectiva, ya que permiten la navegación entre diferentes pantallas y manipulación de diferentes botones y características dentro de las mismas.[1][2]

Actividades

Ciclo de vida

El ciclo de vida es importante desarrollarlo ya que habla sobre las etapas o estados que tendrá el desarrollo de las aplicaciones móviles, desde creación, avance o inclusive la destrucción del mismo.

Estas actividades ya tienen métodos dentro de los entornos de desarrollo de Android, para poder tener una manipulación dentro de la plataforma de una forma más fácil[1][2][3].

Actividades

Métodos relevantes en Android

Método	Descripción
onCreate()	Se llama cuando se crea la actividad por primera vez. Aquí se inicializan los componentes de la interfaz de usuario.
onStart()	Se invoca cuando la actividad se vuelve visible para el usuario.
onResume()	Se ejecuta justo antes de que la actividad comience a interactuar con el usuario.
onPause()	Se llama cuando el sistema está a punto de iniciar otra actividad o cuando la actividad pierde el foco.
onStop()	Se invoca cuando la actividad ya no es visible para el usuario.
onRestart()	Se llama cuando una actividad detenida vuelve a ser visible.
onDestroy()	Se ejecuta antes de que la actividad sea destruida, liberando recursos.

Actividades

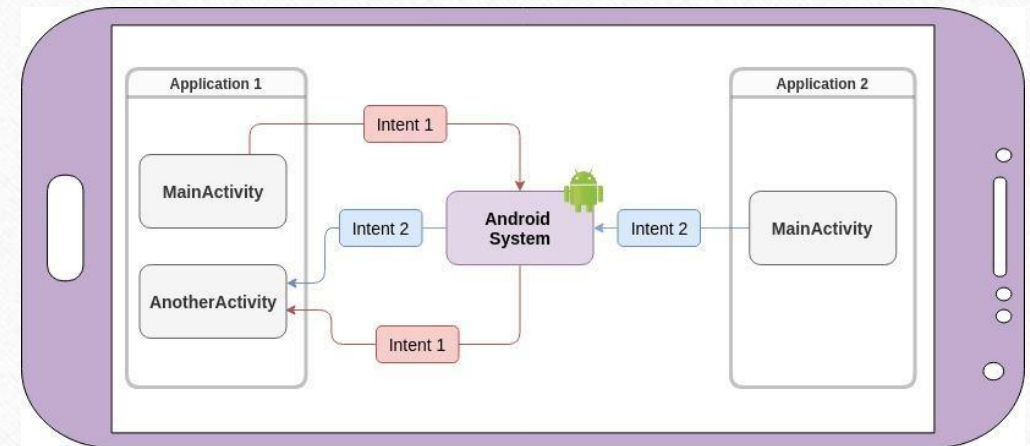
Estados del Ciclo de vida

Estado	¿En memoria?	¿Visible al usuario?	¿En primer plano?
Inexistente	NO	NO	NO
Detenida	SI	NO	NO
Pausada	SI	SI	NO
En marcha	SI	SI	SI

Cada uno de estos estados tiene implicaciones sobre cómo se manejan los recursos y cómo se interactúa con el usuario[3].

Intents

Un Intent es un objeto de mensajería que puedes usar para solicitar una acción desde otro componente de la aplicación. Usando intents podemos facilitar la comunicación entre los componentes de la aplicación.



Tipos de Intents

Explícitos

Los intents explícitos especifican qué componente de qué aplicación satisfará el intent. Se usan los intents explícitos para iniciar un componente en tu propia app.

- `ComponentName`

Implícitos

Los intents implícitos no nombran un componente específico, sino que declaran una acción general, lo que permite que un componente de otra app lo controle.

Como crear un intent

Un objeto Intent contiene información que usa el sistema Android para determinar qué componente debe recibir el intent, además de información que el componente receptor utiliza en para realizar la acción correctamente:

- Nombre del componente (ComponentName)
- Acción (ACTION_)
- Datos (setData(), setType(), setDataAndType())
- Categoría (CATEGORY_BROWSABLE, CATEGORY_LAUNCHER)
- Extra
- Marcas (setFlags())

Ejemplos

Implícitos

```
Intent sendIntent = new Intent();  
sendIntent.setAction(Intent.ACTION_SEND);  
sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);  
sendIntent.setType("text/plain");  
startActivity(sendIntent);
```

Explícitos

```
Intent downloadIntent = new Intent(this,  
DownloadService.class);  
downloadIntent.setData(Uri.parse(fileUrl));  
startService(downloadIntent);
```


Fragmentos

Descripción

Un fragmento representa una parte reutilizable del UI de nuestra aplicación. Los fragmentos no pueden existir por sí solos. Deben estar alojados por una actividad u otro fragmento. Se trata de mini actividades contenidas dentro de una actividad anfitriona, manejando su propio diseño y ciclo de vida.

Fragmentos

Término Técnico

Un fragmento es una parte modular de la interfaz de usuario dentro de una actividad. Un fragmento tiene su propio ciclo de vida, recibe sus propios eventos de entrada, y podemos agregar o quitar fragmentos mientras se ejecuta la actividad que lo contiene.[5]

Fragmentos

Ciclo de vida

El ciclo de vida de un fragmento depende del ciclo de vida de una actividad, cuando una actividad está en estado de reanudación, todos los fragmentos que esta contiene pueden actuar independientemente y pasar a sus otros estados sin ningún problema. Pero si en algún momento la actividad pasa a pausa, entonces todos los fragmentos siguen su comportamiento, al igual que si pasa a detención o se destruye.[4]

Fragmentos

Código básico

Para crear un fragmento, está representado en el API de Android por la clase `Fragment`, por lo que cada vez que vayamos a crear un fragmento personalizado debemos crear una nueva clase que herede sus propiedades y comportamientos. Luego de ello debes sobrescribir el método `onCreateView()` para retornar el `View` del fragmento hacia la jerarquía de la actividad anfitriona. Para eso debemos inflar el código java desde el layout personalizado del fragmento. [5]

Fragmentos

Código básico

```
public static class FragmentoEjemplo extends Fragment {  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.fragmento_ejemplo, container, false);  
    }  
}
```



Fragmentos

Métodos relevantes en Android

Método	Descripción
onAttach()	Es invocado cuando el fragmento ha asociado a la actividad anfitriona
onActivityCreated()	Se ejecuta cuando la actividad anfitriona ya ha terminado la ejecución de su método.
onCreate()	Este método es llamado cuando el fragmento se está creando.
onCreateView()	Se llama cuando el fragmento será dibujado por primera vez en la interfaz de usuario.
onStart()	Se llama cuando el fragmento esta visible ante el usuario.
onResume()	Es ejecutado cuando el fragmento esta activo e interactuando con el usuario.
onStop()	Se llama cuando un fragmento ya no es visible para el usuario.
onPause()	Se ejecuta cuando se detecta que el usuario dirigió el foco por fuera del fragmento.
onDestroyView()	Este método es llamado cuando la jerarquía de views a la cual ha sido asociado el fragmento ha sido destruida.
onDetach()	Se llama cuando el fragmento ya no está asociado a la actividad anfitriona.

Conclusiones

Podemos observar 3 elementos básicos de la programación en android y como se pueden implementar, con esta información se pueden realizar ciertos procesos que no resultan tan simples dentro de un sistema Android, como puede ser la comunicación por lo que aunque pueda parecer información sencilla, puede llegar a salvar nuestros proyectos cuando necesitemos de ciertas “actividades” en específico, aunque también cabe resaltar que esta presentación no está cerca de contener toda la información que podemos saber de cada elementos que hemos repasado y ciertamente debemos mantenernos siempre revisando las documentaciones oficiales para conocer todo lo que conlleva esta información

Bibliografías

- 1 <https://keepcoding.io/blog/ciclo-de-vida-de-un-activity-en-android/>
- 2 <https://learn.microsoft.com/es-es/previous-versions/xamarin/android/app-fundamentals/activity-lifecycle/>
- 3 <https://developer.android.com/codelabs/basic-android-kotlin-compose-activity-lifecycle?hl=es-419#0>
- 4 <https://developer.android.com/guide/components/intents-filters?hl=es-419>
- 5 <https://developer.android.com/guide/fragments?hl=es-419>
- 6 <https://www.develou.com/android-aplicaciones-fragmento/>