Ciclo de vida de componentes

Intent y bundle

{desafío} latam_

Utilizar elementos del ciclo de vida para la implementación de un aplicativo Android que resuelve un problema.

- Unidad 1: Kotlin para el desarrollo de aplicaciones.
- Unidad 2: Ciclo de vida de componentes.
- Unidad 3: Arquitectura en Android.
- Unidad 4: Programación asíncrona en Android.

Te encuentras aquí

{desafío} latam_

¿Qué aprenderás en esta sesión?

- *Enviar datos entre diferentes tipos de vistas y aplicaciones haciendo uso de las herramientas que Android provee como sistema operativo*

Ahora, escribe en el chat:

¿Cómo te imaginas que funciona el envío de datos de una vista a otra?

/* Intents */

Intents

Una intent es un objeto de mensajería que puedes usar para solicitar una acción de otro componente de una app. Si bien las intents facilitan la comunicación entre componentes de varias formas, existen tres casos de uso principales:

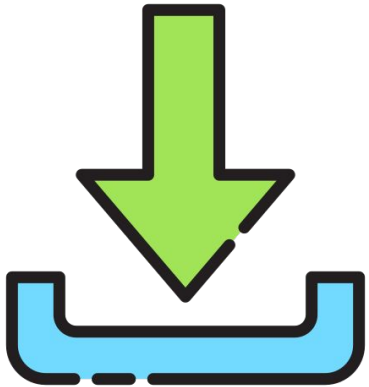
- Iniciar un Activity
- Iniciar un Servicio
- Transmitir una emisión

/* Intent explícitos */

Tipos de Intents explícitos

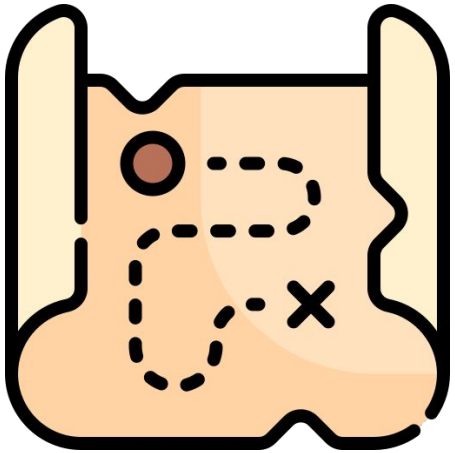
Para hablar de los Intents explícitos, primero debe aclarar que existen dos tipos de intents:

A) **Las intents explícitos** especifican qué aplicación las administrará, ya sea incluyendo el nombre del paquete de la app de destino o el nombre de clase del componente completamente calificado. Normalmente, el usuario usa una intent explícita para iniciar un componente en su propia aplicación porque conoce el nombre de clase de la actividad o el servicio que desea iniciar.



Por ejemplo, puedes utilizarla para iniciar una actividad nueva en respuesta a una acción del usuario o iniciar un servicio para descargar un archivo en segundo plano.

Tipos de Intents explícitos



B) **Las intents implícitos** no nombran el componente específico, pero, en cambio, declaran una acción general para realizar, lo cual permite que un componente de otra aplicación la maneje.

Por ejemplo, si deseas mostrar al usuario una ubicación en un mapa, puedes usar una intent implícita para solicitar que otra aplicación apta muestre una ubicación específica en un mapa.

Ejemplo de Intent explícito

Una intent explícita es una intent que se usa para iniciar un componente específico de la aplicación, como una actividad o un servicio particular en la aplicación.
Para crear una intent explícita, define el nombre de componente del objeto Intent (todas las otras propiedades de la intent son opcionales).

- Por ejemplo, si creaste un servicio en tu app denominado DownloadService, diseñado para descargar un archivo de la Web, puedes iniciarlo con el siguiente código:

```
val downloadIntent = Intent(this,
DownloadService::class.java).apply {
          data = Uri.parse(fileUrl)
}
startService(downloadIntent)
```

- O por ejemplo, un intent muy conocido, iniciar un Activity:

```
startActivity(Intent(context, MainActivity::class.java))
```

/* Intents con resultados */

registerForActivityResult()

Si bien **registerForActivityResult()** registra su devolución de llamada, no inicia la otra actividad ni inicia la solicitud de un resultado. En cambio, esto es responsabilidad de la instancia de ActivityResultLauncher devuelta.

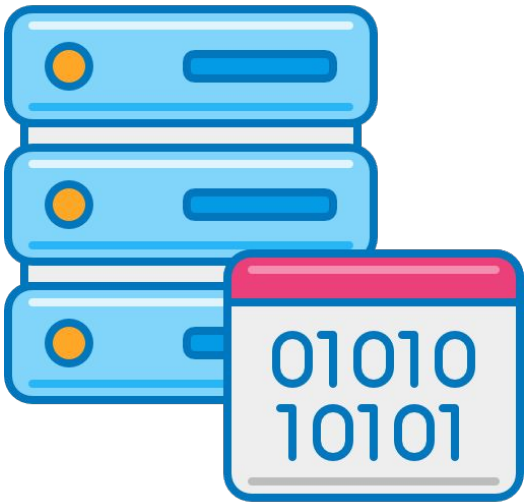
Si existe una entrada, el iniciador toma la entrada que coincide con el tipo de ActivityResultContract. Llamar a launch() inicia el proceso de producir el resultado. Cuando el usuario termina con la actividad subsiguiente y regresa, se ejecuta onActivityResult() de ActivityResultCallback, como se muestra en el siguiente ejemplo:

registerForActivityResult()

```
val getContent = registerForActivityResult(GetContent()) { uri: Uri? ->
    // Handle the returned Uri
}

override fun onCreate(savedInstanceState: Bundle?) {
    // ...

    val selectButton = findViewById<Button>(R.id.select_button)

    selectButton.setOnClickListener {
        // Pass in the mime type you'd like to allow the user to select
        // as the input
        getContent.launch("image/*")
    }
}
```

/* Usar Intents para pasar datos */

Usar Intents para pasar datos



Ya vimos que se puede usar un intent para iniciar un activity, pero **¿Qué pasa si necesitamos enviar datos a ese Activity?** Supongamos que necesitamos abrir un activity el cual recibe un id tipo Int, y hace un llamado a la base de datos local para mostrar los detalles de un usuario.

Lo primero que necesitamos hacer es pasar ese ID, tomamos el ejemplo anterior cambiamos el activity:

```
startActivity(Intent(context,
DetailContactActivity::class.java))
```

Usar Intents para pasar datos

Ahora necesitamos pasar el ID en el Intent:

```
val intent = Intent(context,
DetailContactActivity::class.java)
intent.putExtra("USER_ID", 1234)
startActivity(intent)
```

O en un estilo más Kotlin:

```
startActivity(Intent(context,
DetailContactActivity::class.java).apply {
    putExtra("USER_ID", 1234)
})
```

/* Pasar datos como resultados */

Pasar datos como resultados

Sabemos que podemos utilizar intent para iniciar una Activity y que además podemos pasar datos. ¿Pero cómo hacemos para leer esos datos en otro Acitivity?

Supongamos que estamos enviando datos desde el **MainActivity** a un activity al que llamaremos **DetailContactActivity**, tomando el ejemplo anterior tenemos:

```
MainActivity : AppCompatActivity() {
...
    startActivity(Intent(context,
DetailContactActivity::class.java).apply {
      putExtra("USER_ID", 1234)
  })
```

Pasar datos como resultados

Luego, para leer estos datos en `DetailContactActivity`, lo hacemos leyendo el objeto intent que existe en todo activity y dependiendo del tipo de dato que enviamos, en este caso Int, leemos el valor haciendo uso de la llave

```
val userId = intent.getIntExtra("USER_ID", 0)
```

Existen algunos tipos de datos que requieren valores por defecto, en el caso de Int, requiere que se defina un valor en caso de que la llave que se especifica en el intent no tenga valor.

/* Objeto Bundle */

¿Qué es Bundle?

Los Bundle pueden ser vistos como una forma de empaquetar datos que se quieren enviar entre vistas; es muy común utilizarlos para enviar datos entre activities. Por ejemplo, si queremos enviar un dato a una Activity podemos usar lo siguiente:

```
val intent = Intent(context, MainActivity::class.java)
startActivity(intent.apply {
putExtra("SOME_KEY", "Some value here")
})
```

En caso de necesitar enviar más datos, se pueden agregar más "`putExtra("SOME_KEY", "Some value here")`"

No confundir con Android App Bundle ***Un Android App Bundle es un formato de publicación que incluye todos los recursos y el código compilado de tu app, pero delega la generación del APK y la firma a Google Play.***

Ejemplo de uso de Bundle

Pero qué pasa si lo que necesitamos enviar no es un String o un Int y se trata de un objeto más complejo. Supongamos que queremos para a MainActivity un objeto “Person” que se define de la siguiente forma:

```
@Parcelize
data class Person(
    val name:String,
    val age: Int,
): Parcelable

val person = Person("Samuel", 42)

val intent = Intent(context, MainActivity::class.java)
val bundle = Bundle()
bundle.putParcelable("OBJ_PERSON",person)

intent.putExtra("BUNDLE",bundle)
startActivity(intent)
```

Ya que intent no nos permite enviar objetos como Person, pero sí permite enviar bundle, entonces empaquetamos Person en un bundle

¡Recuerda!

- Los Intent son una de las herramientas más utilizadas en Android.
- Sirven para iniciar activities y enviar datos dentro de tu app y también para interactuar con otras aplicaciones

Próxima sesión...

- *Revisar el material de estudio sincrónico que consiste en una guía de estudio para practicar los conceptos aprendidos en esta sesión.*

{desafío} latam_

{desafío}
latam_

Academia de
talentos digitales