**Sesión 3-01 Clase del 4 de diciembre**

En esta sesión vamos a ver cómo se trabaja **intents implícitos** que permiten abrir una **activity** de otra aplicación para procesar un contenido.

Un ***intent***es un objeto a través del cual un componente como una ***activity*** se comunica con otro componente (*activity, service, broadcast receiver, content provider*) para iniciarle y/o pasarle datos.

A través de un ***intent* implícito** el SO llamará a ejecución a una **activity** de otra app que puede realizar la acción requerida (son apps que tienen un filtro que indica que pueden responder a una o varias acciones).

Las partes más importantes que puede utilizar al crear un ***intent* implícito** son:

* La **acción** que se quiere realizar.
* La **URI** del recurso sobre el que se va a hacer la acción. Una URI es la especificación del protocolo de acceso al recurso y la ubicación del recurso.
* El **tipo de datos** relacionado con la acción. A veces no es necesario pues se deduce de la URI. Algunos tipos son **text/plain, text/html, audio/mpeg3.**
* **Categorías opcionales** especifican más información sobre el *intent*. Por ejemplo, permiten agregar un adjunto en un *intent* para enviar un email.

La sintaxis básica para crear y lanzar un **intent implícito** para iniciar una **activity** de otra app es:

val context= *LocalContext*.current

val intent = Intent(*acción*, Uri.parse(*datos*))  
context.startActivity(intent)

1.- Descarga y abre el proyecto Sesion3\_01\_Base. Tiene preparada la pantalla con la que vamos a trabajar está en esta sesión. Hay cuatro botones para abrir una página web con el navegador, abrir maps en una determinada localización, enviar un menaje y compartir un mensaje.

Para todos los **TextFields** se tienen declaradas las variables de estado que almacenan el contenido de esos **TextFields.** Se tiene declarada un variable para almacenar el contexto y que se usará para procesar los **intents implícitos.**

2.- Vamos a programar el **Button VER WEB** para que construya un ***intent*** que permita iniciar una ***activity*** de una **app** navegador web que abra la página web cuya URL se haya escrito en el **TextField** de la url (variable de estado **url**).

En un ***intent* implito** para abrir una página web:

* La acción es **Intent.ACTION\_VIEW.** Esta acción también se usa para ver otros recursos como ver un contacto o reproducir un archivo multimedia.
* La URI se corresponde con la URL de la página web. El tipo de recurso se identifica en la URI por ser un texto que comience con **http:,** la ubicación del recurso es lo que hay después de **http:**.

val intent = Intent(Intent.*ACTION\_VIEW*, Uri.parse("http://campus.educantabria.es"))

Y para que se lance la **app receptora** a través del *intent* creado:

context.startActivity(intent)

Prueba lo anterior. Modifica para que se abra la web correspondiente a la url introducida en el **TextField url.**

3.- Normalmente, todos los procesamientos de intents implícitos se han de realizar con tratamiento de errores y control de excepciones. Para este caso, este sería un código que lo implementa.

if (url.*isBlank*()) {  
 Toast.makeText(context, "Por favor, introduce una URL.",  
 Toast.*LENGTH\_LONG*).show()  
} else {  
 if (!Patterns.*WEB\_URL*.matcher(url).matches()) {  
 Toast.makeText(  
 context, "La URL no tiene un formato válido.",  
 Toast.*LENGTH\_LONG* ).show()  
 } else {  
 try {  
 val intent = Intent(Intent.*ACTION\_VIEW*, Uri.parse(url))  
 context.startActivity(intent)  
 } catch (e: Exception) {  
 Toast.makeText(  
 context, "No se pudo abrir la URL.",  
 Toast.*LENGTH\_LONG* ).show()  
 }  
 }  
}

4.- Antes de realizar la siguiente programación del *Button* **VER MAPS**, inicia la aplicación de mapas para que quedes registrado como usuario. Después puedes cerrar la aplicación de mapas, se trata de iniciarla desde nuestra aplicación.

Vamos a programar el *Button* **VER MAPS** para que construya un ***intent*** que permita iniciar una *activity* de una **app** de mapas mostrando el mapa entorno a la localización indicada por la latitud y la longitud especificadas en los ***TextFields*** de latitud y longitud. La latitud puede tener valores *Double* entre -90 y +90. La longitud puede tener valores entre -180 y +180.

En un *intent* implícito para iniciar una **app** de **maps**:

* La acción es **ACTION\_VIEW.**
* El texto de una URI que especifica una localización en un mapa con un determinado nivel de zoom es:

***geo:latitud,longitud?z=nivel\_zoom***

El nivel de zoom puede tener valores con decimales entre 0 y 21 (máximo zoom).

Por ejemplo, puedes probar esto para abrir el mapa con zoom 17 sobre Torrelavega.

val geoUri = "geo:43.3495,-4.0513?z=17.0"  
val intent = Intent(Intent.*ACTION\_VIEW*, Uri.parse(geoUri))  
context.startActivity(intent)

Sustituye en el código, para que el mapa se abra sobre la localización establecida en los **TextFields** de latitud y longitud. Prueba con diferentes valores de zoom.

5.- Para implementar lo anterior con tratamiento de errores y control de excepciones.

try {

// Covertir String a Double o nulo  
 val lat = latitude.*toDoubleOrNull*()  
 val lon = longitude.*toDoubleOrNull*()  
  
 // Validación de la latitud y longitud  
 if (lat == null || lon == null || lat !in -90.0..90.0 || lon !in -180.0..180.0) {  
 Toast.makeText(context,  
 "Por favor, valores para latitud (-90 a 90) y longitud (-180 a 180).", Toast.*LENGTH\_LONG*).show()  
 } else {  
 // Crear Intent para mostrar la ubicación en una app de mapas  
 val geoUri = "geo:$lat,$lon?z=15.0"  
 val intent = Intent(Intent.*ACTION\_VIEW*, Uri.parse(geoUri))  
 context.startActivity(intent)  
 }  
} catch (e: Exception) {  
 Toast.makeText(context, "Error al intentar mostrar el mapa",  
 Toast.*LENGTH\_*).show()  
}

6.- Vamos a programar el **Button ENVIAR\_A** para que construya un **intent** que permita abrir una app para enviar el mensaje escrito en el **TextField Mensaje** al teléfono escrito en el **TextField Teféfono** indicado.

En este *intent* implícito para iniciar una **app** de mensajería:

* La acción es **ACTION\_SENDTO.**
* El texto de una URI que especifica un **sms** para un teléfono es:

***smsto:numero\_telefono***

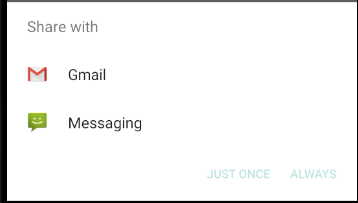
Dado que en una acción **ACTION\_SENDTO** se pueden enviar varios tipos de contenidos:

* Hay que especificar que se enviará el mensaje en la clave **sms\_body**
* Por defecto, un mensaje de texto se envía en formato **text/plain**.

Para construir este *intent* que procese la acción **SENDTO** hay varias alternativas. Una de ellas es la siguiente:

val intent = Intent(Intent.*ACTION\_SENDTO*).*apply* **{** *data* = Uri.parse("smsto:$phoneNumber") // Esquema para SMS  
 putExtra("sms\_body", message) // Mensaje a enviar  
**}**context.startActivity(intent)

En este caso será normal que Android te permita elegir la app que procese el *intent* ya que puede tener muchas disponibles:



Si construimos el *intent* sin pasar el URI, se ejecuta de igual forma, pero pudiendo elegir después en la aplicación de mensajería cual es el destinatario (teléfono u otro tipo de dirección destino).

7.- Por último, vamos a programar el *Button* **COMPARTIR** para compartir el mensaje del **TextField** seleccionando la app destino. En este caso usaremos la acción SEND en lugar de SENDTO. SEND es más genérica y permite enviar otros tipos de contenido, además del tipo texto, seleccionar la aplicación de destino, etc. El siguiente es un ejemplo para compartir, seleccionando la aplicación destino en un **Chooser**.

// Crear el Intent para compartir el mensaje  
val intent = Intent(Intent.*ACTION\_SEND*).*apply* **{** *type* = "text/plain" // Tipo de contenido  
 putExtra(Intent.*EXTRA\_TEXT*, message) // Mensaje a compartir  
**}**// Crear un chooser para seleccionar la aplicación de destino  
val chooser = Intent.createChooser(intent, "Elige una aplicación para compartir")  
context.startActivity(chooser)

Si quisiéramos compartir una imagen, el tipo podría ser **image/\*** y el extra se prepararía con **Intent.EXTRA\_STREAM, UriDeArchivoImagen.**

Si en lugar de lo anterior queremos que el mensaje se comparta sólo a través de la aplicación **Whatsapp**, tendríamos que construir el intent y abrir la aplicación destino de esta forma:

val intent = Intent(Intent.*ACTION\_SEND*).*apply* **{** *type* = "text/plain" // Tipo de contenido  
 putExtra(Intent.*EXTRA\_TEXT*, message) // Mensaje a compartir  
 *`package`* = "com.whatsapp" // Especifica que sea WhatsApp  
 **}**context.startActivity(intent)