

Android Developer Fundamentals

Actividades e Intentos

Lección 2



2.1 Actividades

Contenidos

- Actividades
- Definiendo actividades
- Comenzando una nueva actividad con un intento
- Pasando datos entre actividades con extras
- Navegando entre actividades

Actividades (vista de alto nivel)

¿Qué es una actividad?

- Una Activity es un componente de aplicación
- Representa una ventana, una jerarquía de vistas
- Normalmente llena la pantalla, pero puede incrustarse en otra actividad o aparecer como una ventana flotante
- Clase de Java, típicamente una Actividad por archivo

¿Qué hace una actividad?

- Representa una actividad, como ordenar alimentos, enviar correos electrónicos u obtener indicaciones
- Maneja las interacciones del usuario, como los clics en los botones, la entrada de texto o la verificación de inicio de sesión
- Puede iniciar otras actividades en la misma o en otras aplicaciones
- Tiene un ciclo de vida—se crea, se inicia, se ejecuta, se suspende, se reanuda, se detiene, y se destruye

Ejemplos de actividades

Hello Toast

TOAST

4

COUNT

My Food List

Cheese

Pepperoni

Black Olives

Pineapple

Strawberries

Artichokes

Red peppers

Mushrooms

3.1415×2.718

8.538597

7

8

9

DEL

4

5

6

÷

1

2

3

×

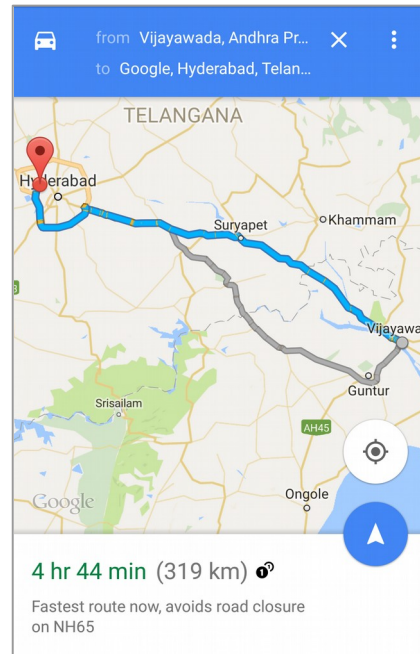
-

.

0

=

+



Aplicaciones y actividades

- Las actividades están ligeramete vinculadas para crear una aplicación
- La primera actividad que ve el usuario se llama típicamente “actividad principal”
- Las actividades se pueden organizar en relaciones entre padres e hijos en el manifiesto de Android para facilitar la navegación

Diseños y Actividades

- Una actividad normalmente tiene un diseño de interfaz de usuario
- El diseño se define generalmente en uno o más archivos XML
- La actividad "infla" el diseño como parte de la creación

Implementando Actividades

Implementar nuevas actividades

1. Definir el diseño en XML
2. Definir la clase Java de la Actividad
 - extiende AppCompatActivity
3. Conectar la Actividad con el diseño
 - Establecer el contentView en onCreate()
4. Declarar la Actividad en el manifiesto de Android

1. Definir el diseño en XML

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent">
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="Let's Shop for Food!" />
```

```
</RelativeLayout>
```

2. Definir la clase Java de la Actividad

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
}
```

3. Conectar la actividad con el diseño

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

recurso es en este archivo
 recurso diseño XML

4. Declarar la actividad en el manifiesto de Android

```
<activity android:name=".MainActivity">
```

4. Declarar actividad principal en manifiesto

La actividad principal debe incluir el intento de comenzar desde el ícono desde el lanzador

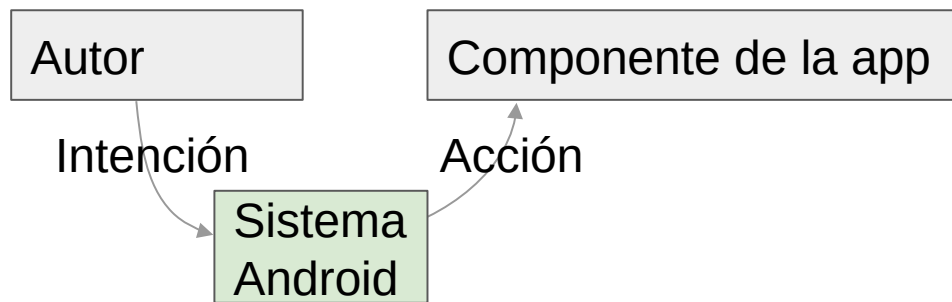
```
<activity android:name=".MainActivity">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>
```


Intentos

¿Qué es un intento?

Un intento es una descripción de una operación a realizar

Un intento es un objeto utilizado para solicitar una acción desde otro componente de la aplicación a través del sistema Android



¿Qué pueden hacer los intentos?

- Iniciar actividades

- Un clic de botón inicia una nueva actividad para la entrada de texto
- Al hacer clic en Compartir se abre una aplicación que te permite publicar una foto

- Iniciar servicios

- Inicia la descarga de un archivo en segundo plano

- Entregar emisiones (broadcasts)

- El sistema informa a todos que el teléfono se está cargando

Intentos explícitos e implícitos

Intentos explícitos

- Inicia una actividad específica
 - Pide té con leche entregado por Nikita
 - La actividad principal inicia la actividad ViewShoppingCart

Intentos implícitos

- Pide al sistema que encuentre una actividad que pueda manejar esta solicitud
 - Encuentra una tienda abierta que venda té verde
 - Al hacer clic en Compartir se abre un selector con una lista de aplicaciones

Iniciando actividades

Iniciar una actividad con un intento explícita

Para comenzar una actividad específica, usa un intento explícito

1. Cree un intento

- `Intent intent = new Intent(this, ActivityName.class);`

2. Use un intento para iniciar la actividad

- `startActivity(intent);`

Iniciar una actividad con un intento implícito

Para pedirle a Android que encuentre una Actividad para manejar su solicitud, use un intento implícito

1. Cree un intento

- `Intent intent = new Intent(action, uri);`

2. Use un intento para iniciar la actividad

- `startActivity(intent);`

Intentos implícitos - Ejemplos

Mostrar una página web

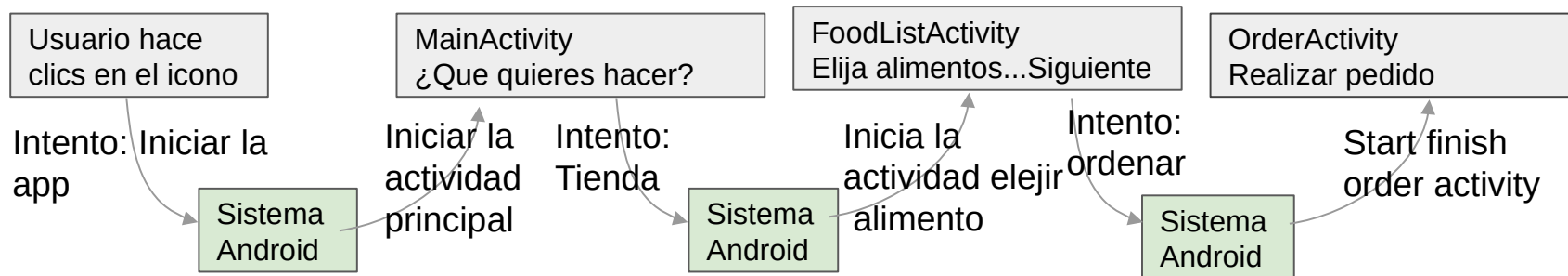
```
Uri uri = Uri.parse("http://www.google.com");  
Intent it = new Intent(Intent.ACTION_VIEW, uri);  
startActivity(it);
```

Marque un número telefónico

```
Uri uri = Uri.parse("tel:8005551234");  
Intent it = new Intent(Intent.ACTION_DIAL, uri);  
startActivity(it);
```


Cómo se ejecutan las actividades

- Todas las actividades son gestionadas por el motor de ejecución de Android
- Comenzando por un "intento", un mensaje al motor de ejecución de Android para ejecutar la actividad



Enviando y recibiendo datos

Dos tipos de envío de datos con intenciones

- Datos—una pieza de información cuya ubicación de datos se puede representar mediante una URI
- Extras—una o más piezas de información como una colección de pares clave-valor en un Bundle

Enviando y recuperando datos

En la primera actividad (enviando):

1. Crea el objeto intento
2. Pone datos o extras en esa intención
3. Inicia la nueva actividad con `startActivity()`

En la segunda actividad (receptora):

4. Obtener el objeto “intento” con el que se inició la actividad
5. Recupera los datos o extras del objeto Intent

Poner una URI como datos de intento

```
// Una URL de una página web
intent.setData(
    Uri.parse("http://www.google.com"));

// una URI de un archivo de ejemplo
intent.setData(Uri.fromFile(
    new File("/sdcard/sample.jpg")));
```

Poner información en intentos extras

- `putExtra(String name, int value)`
⇒ `intent.putExtra("level", 406);`
- `putExtra(String name, String[] value)`
⇒ `String[] foodList = {"Rice", "Beans", "Fruit"};`
`intent.putExtra("food", foodList);`
- `putExtras(bundle);`
⇒ si hay muchos datos, primero cree un paquete y pase el paquete

Envío de datos a una actividad con extras

```
public static final String EXTRA_MESSAGE_KEY =  
  
"com.example.android.twoactivities.extra.MESSAGE";  
  
Intent intent = new Intent(this,  
SecondActivity.class);  
String message = "Hello Activity!";  
intent.putExtra(EXTRA_MESSAGE_KEY, message);  
startActivity(intent);
```

Obtener datos con intentos

- `getData();`
⇒ `Uri locationUri = intent.getData();`
- `getIntExtra (String name, int defaultValue)`
⇒ `int level = intent.getIntExtra("level", 0);`
- `Bundle bundle = intent.getExtras();`
⇒ obtenga todos los datos a la vez como un bundle

Devolviendo datos a la actividad de inicio

1. Use `startActivityForResult()` para iniciar la segunda actividad
2. Para devolver datos de la segunda actividad:
 - Cree un **nuevo** intento
 - Ponga los datos de respuesta en el intento usando `putExtra()`
 - Establece el resultado en `Activity.RESULT_OK` o `RESULT_CANCELED`, si el usuario canceló
 - Llame a `finish()` para cerrar la actividad
3. Implemente `onActivityResult()` en la primer actividad

startActivityResult()

startActivityResult(intent, requestCode);

- Inicia la actividad (intento), le asigna el identificador (requestCode)
- Devuelve datos a través de intentos extras
- Cuando termine, el pop del stack, vuelve a la actividad anterior, y ejecuta el callback onActivityResult() para procesar los datos devueltos
- Utilice requestCode para identificar qué actividad ha “vuelto”

1. Ejemplo de startActivityForResult()

```
public static final int CHOOSE_FOOD_REQUEST = 1;

Intent intent = new Intent(this,
    ChooseFoodItemsActivity.class);
startActivityForResult(intent, CHOOSE_FOOD_REQUEST);
```

Devolver datos y terminar segunda actividad

```
// Crear un intento
Intent replyIntent = new Intent();

// Pon los datos para volver a los extra
replyIntent.putExtra(EXTRA_REPLY, reply);

// Establece el resultado de la actividad a RESULT_OK
setResult(RESULT_OK, replyIntent);

// Termina la actividad actual
finish();
```

3.Implementar onActivityResult()

```
public void onActivityResult(int requestCode,
                             int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == TEXT_REQUEST) { // Identify activity
        if (resultCode == RESULT_OK) { // Activity succeeded
            String reply =
data.getStringExtra(SecondActivity.EXTRA_REPLY);
            // ... do something with the data
        }
    }
}
```

Navegación

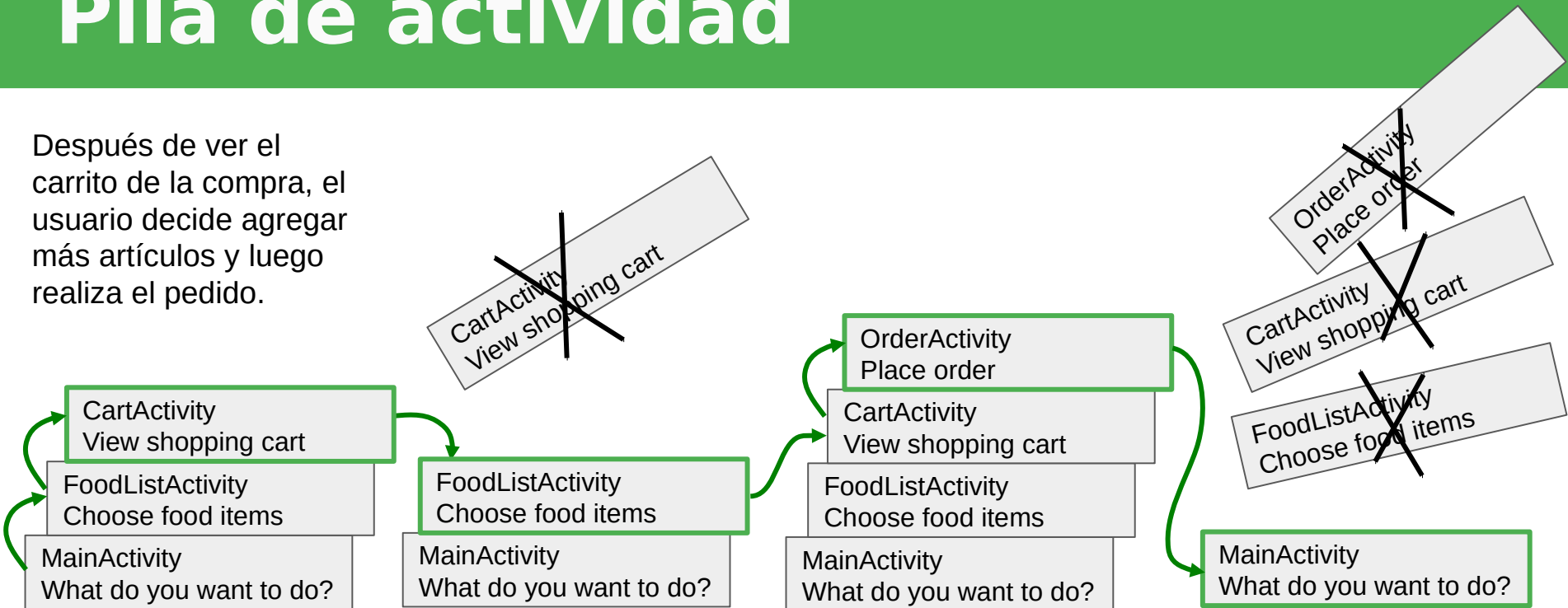
Pila de actividades

- Cuando se inicia una nueva actividad, la actividad anterior se detiene y se empuja en la pila de actividad posterior
- La pila es Last-in-first-out—cuando la actividad actual finaliza o el usuario presiona el botón Atrás, se extrae de la pila y se reanuda la actividad anterior



Pila de actividad

Después de ver el carrito de la compra, el usuario decide agregar más artículos y luego realiza el pedido.



Dos formas de navegacion



Navegación temporal o tracera

- Proporcionada por el botón de retroceso del dispositivo
- Controlado por la pila trasera del sistema Android




Navegación ancestral o ascendente

- Proporcionado por la barra de acción de la aplicación
- Controlado mediante la definición de las relaciones padre-hijo entre actividades en el manifiesto de Android



Navegación de fondo

- La pila de fondo conserva el historial de las pantallas vistas recientemente
- La pila de fondo contiene todas las actividades que el usuario ha iniciado en orden inverso para la tarea actual
- Cada tarea tiene su propia pila de fondo
- Cambio entre las tareas de las actividades contenidas en la pila de fondo
- Iniciar una actividad desde la pantalla de inicio inicia una nueva tarea
- Navegue entre las tareas  con la vista general o la pantalla de tareas recientes



Navegación hacia arriba

- Va al padre de la actividad actual
- Definir el padre de una actividad en el manifiesto de Android
- Establecer `parentActivityName`

```
<activity  
    android:name=".ShowDinnerActivity"  
    android:parentActivityName=".MainActivity" >  
</activity>
```

FIN