

Gestión de Información en Dispositivos Móviles



Intents

Javier Abad (abad@decsai.ugr.es)

Dpto. de Ciencias de la Computación e Inteligencia Artificial

<http://decsai.ugr.es>

Recordemos...

- ▶ Una actividad puede iniciar programáticamente otra actividad
 - ▶ Creamos un objeto Intent
 - ▶ Pasamos ese Intent a `startActivity()` o `startActivityForResult()`
- ▶ **Contenidos:**
 - ▶ La clase Intent
 - ▶ Iniciación de actividades
 - ▶ Activación explícita
 - ▶ Activación implícita mediante resolución de intents

La clase Intent

- ▶ Un Intent es una estructura de datos que puede representar:
 - ▶ Una operación a realizar
 - ▶ Un evento que ha ocurrido en el sistema
- ▶ Vamos a centrarnos en el uso de los Intents para especificar operaciones a realizar, no para la notificación de eventos.
- ▶ Veremos el uso de los Intents para la notificación de eventos cuando hablemos de los receptores de anuncios (Broadcast Receivers).

Uso de Intents para especificar operaciones

- ▶ Los Intents proporcionan un lenguaje flexible y cómodo para especificar operaciones que queremos que se realicen.
 - ▶ Seleccionar un contacto
 - ▶ Hacer una foto
 - ▶ Marcar un número de teléfono
 - ▶ Mostrar un mapa
 - ▶ ...
- ▶ El intent lo construye una actividad que quiere que se haga una determinada tarea
- ▶ Android inicia otra actividad para que realice la tarea deseada. Esa actividad recibe el Intent.

Campos del Intent

- ▶ Action
- ▶ Data
- ▶ Category
- ▶ Type
- ▶ Component
- ▶ Extras
- ▶ Flags

Action

- ▶ String que representa o nombra la operación a realizar
- ▶ Ejemplos:
 - ▶ ACTION_DIAL: marcar un número de teléfono
 - ▶ ACTION_EDIT: mostrar datos para edición
 - ▶ ACTION_SYNC: sincronizar datos del dispositivo con un servidor
 - ▶ ACTION_MAIN: iniciar actividad como actividad inicial de una aplicación

Action

- Actions predefinidas para lanzar actividades:

<u>ACTION_MAIN</u>	<u>ACTION_FACTORY_TEST</u>
<u>ACTION_EDIT</u>	<u>ACTION_WEB_SEARCH</u>
<u>ACTION_VIEW</u>	<u>ACTION_SEARCH</u>
<u>ACTION_ATTACH_DATA</u>	<u>ACTION_PICK_ACTIVITY</u>
<u>ACTION_PICK</u>	<u>ACTION_SYNC</u>
<u>ACTION_CHOOSER</u>	<u>ACTION_RUN</u>
<u>ACTION_GET_CONTENT</u>	<u>ACTION_DELETE</u>
<u>ACTION_DIAL</u>	<u>ACTION_INSERT</u>
<u>ACTION_CALL</u>	<u>ACTION_ANSWER</u>
<u>ACTION_SEND</u>	<u>ACTION_SENDTO</u>

Cómo establecer la Action del Intent

- ▶ Pasar la Action al constructor

```
Intent nuevo_intent = new Intent(Intent.ACTION_DIAL);
```

- ▶ Crear el Intent vacío y usar el método setAction

```
Intent nuevo_intent = new Intent();  
nuevo_intent.setAction(Intent.ACTION_DIAL);
```


Data

- ▶ Datos asociados al Intent
 - ▶ Formateados como un URI (*Uniform Resource Identifier*: Identificador de Recursos Uniforme)
- ▶ Ejemplos:
 - ▶ Datos para ver un mapa
`Uri.parse("geo:0,0?q=Recogidas+50+Granada")`
 - ▶ Número para marcar en el teléfono
`Uri.parse("tel:+34900123456")`

`Uri.parse` crea un objeto `Uri` a partir de la cadena que se le pasa

Cómo establecer los Data del Intent

- ▶ Pasar los Data al constructor

```
Intent nuevo_intent = new Intent(Intent.ACTION_DIAL,  
                                Uri.parse("tel:+34900123456"));
```

- ▶ Crear el Intent y usar el método setData

```
Intent nuevo_intent = new Intent(Intent.ACTION_DIAL);  
nuevo_intent.setData(Uri.parse("tel:+34900123456"));
```

Category

- ▶ Información adicional sobre qué componentes pueden manejar el Intent.
- ▶ Ejemplos:
 - ▶ CATEGORY_BROWSABLE: puede ser invocada desde un browser a través de un URI tipo enlace (URL).
 - ▶ CATEGORY_LAUNCHER: puede ser la actividad inicial de una tarea y aparece en el Lanzador de Aplicaciones (Launcher).

Cómo establecer la categoría del Intent

```
Intent.addCategory(String category);
```

Category

► Categorías predefinidas:

<u>CATEGORY_DEFAULT</u>	<u>CATEGORY_PREFERENCE</u>
<u>CATEGORY_BROWSABLE</u>	<u>CATEGORY_TEST</u>
<u>CATEGORY_TAB</u>	<u>CATEGORY_CAR_DOCK</u>
<u>CATEGORY_ALTERNATIVE</u>	<u>CATEGORY_LE_DESK_DOCK</u>
<u>CATEGORY_SELECTED_ALTERNATIVE</u>	<u>CATEGORY_HE_DESK_DOCK</u>
<u>CATEGORY_LAUNCHER</u>	<u>CATEGORY_DESK_DOCK</u>
<u>CATEGORY_INFO</u>	<u>CATEGORY_CAR_MODE</u>
<u>CATEGORY_HOME</u>	<u>CATEGORY_APP_MARKET</u>

Type

- ▶ Especifica el tipo MIME de los datos (Data) del Intent.
- ▶ Ejemplos:
 - ▶ `image/*`, `image/png`, `image/jpg`
 - ▶ `text/html`, `text/plain`
- ▶ Si no especificamos el tipo MIME de los datos, Android intentará inferirlo.

Cómo establecer el tipo del Intent

`Intent.setType(String type)`

`Intent.setDataAndType(Uri data, String type)`

¡Ojo! Si queremos especificar Data (URI) y Type (MIME), no debemos llamar a `setData` y `setType`, porque un método anula al otro. En este caso debemos usar `setDataAndType`.

Component

- ▶ La componente que debe recibir el Intent.
- ▶ Se usa cuando sabemos que hay precisamente una actividad que es la que debe recibir el Intent.

Cómo establecer la componente del Intent

```
Intent nuevo_intent = new Intent(Context packageContext,  
                                Class<?> clase);
```

O bien:

```
Intent nuevo_intent = new Intent ();
```

y usar uno de los métodos:

```
setComponent()
```

```
setClass()
```

```
setClassName()
```

Extras

- ▶ Información adicional asociada al Intent
- ▶ Se tratan como un mapa de parejas clave-valor
- ▶ Ejemplos:
 - ▶ `Intent.EXTRA_EMAIL`: destinatarios de correo

```
Intent nuevo_intent= new Intent(Intent.ACTION_SEND);  
nuevo_intent.putExtra(android.content.Intent.EXTRA_EMAIL,  
    new String[]{  
        "abad@decsai.ugr.es",  
        "tu@tudominio.com",  
        "alcalde@granada.org",  
        "presidente@juntadeandalucia.es"}  
    );
```

Cómo establecer el atributo Extra del Intent

- ▶ El método `putExtra()` está sobrecargado para una gran variedad de tipos.

```
Intent.putExtra(String nombre, String valor);
```

```
Intent.putExtra(String nombre, String[] valor);
```

```
Intent.putExtra(String nombre, float valor);
```

```
...
```


Extras

► Extras predefinidos:

<u>EXTRA_ALARM_COUNT</u>	<u>EXTRA_ORIGINATING_URI</u>
<u>EXTRA_BCC</u>	<u>EXTRA_PHONE_NUMBER</u>
<u>EXTRA_CC</u>	<u>EXTRA_REFERRER</u>
<u>EXTRA_CHANGED_COMPONENT_NAME</u>	<u>EXTRA_REMOTE_INTENT_TOKEN</u>
<u>EXTRA_DATA_REMOVED</u>	<u>EXTRA_REPLACING</u>
<u>EXTRA_DOCK_STATE</u>	<u>EXTRA_SHORTCUT_ICON</u>
<u>EXTRA_DOCK_STATE_HE_DESK</u>	<u>EXTRA_SHORTCUT_ICON_RESOURCE</u>
<u>EXTRA_DOCK_STATE_LE_DESK</u>	<u>EXTRA_SHORTCUT_INTENT</u>
<u>EXTRA_DOCK_STATE_CAR</u>	<u>EXTRA_STREAM</u>
<u>EXTRA_DOCK_STATE_DESK</u>	<u>EXTRA_SHORTCUT_NAME</u>
<u>EXTRA_DOCK_STATE_UNDOCKED</u>	<u>EXTRA_SUBJECT</u>
<u>EXTRA_DONT_KILL_APP</u>	<u>EXTRA_TEMPLATE</u>
<u>EXTRA_EMAIL</u>	<u>EXTRA_TEXT</u>
<u>EXTRA_INITIAL_INTENTS</u>	<u>EXTRA_TITLE</u>
<u>EXTRA_INTENT</u>	<u>EXTRA_UID</u>
<u>EXTRA_KEY_EVENT</u>	

Flags

- ▶ Dan información sobre cómo debería manejarse el Intent.
- ▶ Ejemplos:
 - ▶ FLAG_ACTIVITY_NO_HISTORY: no incluir esta actividad en la pila del historial.
 - ▶ FLAG_DEBUG_LOG_RESOLUTION: mostrar información de logging adicional cuando se procesa el Intent.

Cómo establecer los Flags del Intent

```
Intent nuevo_intent= new Intent(Intent.ACTION_SEND);  
nuevo_intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
```

Category

► Flags predefinidos (para startActivity):

<u>FLAG_ACTIVITY_BROUGHT_TO_FRONT</u>	<u>FLAG_ACTIVITY_NEW_TASK</u>
<u>FLAG_ACTIVITY_CLEAR_TASK</u>	<u>FLAG_ACTIVITY_NO_ANIMATION</u>
<u>FLAG_ACTIVITY_CLEAR_TOP</u>	<u>FLAG_ACTIVITY_NO_HISTORY</u>
<u>FLAG_ACTIVITY_CLEAR_WHEN_TASK_RESET</u>	<u>FLAG_ACTIVITY_NO_USER_ACTION</u>
<u>FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS</u>	<u>FLAG_ACTIVITY_PREVIOUS_IS_TOP</u>
<u>FLAG_ACTIVITY_FORWARD_RESULT</u>	<u>FLAG_ACTIVITY_RESET_TASK_IF_NEEDED</u>
<u>FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY</u>	<u>FLAG_ACTIVITY_REORDER_TO_FRONT</u>
<u>FLAG_ACTIVITY_MULTIPLE_TASK</u>	<u>FLAG_ACTIVITY_SINGLE_TOP</u>
<u>FLAG_ACTIVITY_NEW_DOCUMENT</u>	<u>FLAG_ACTIVITY_TASK_ON_HOME</u>

Iniciación de actividades con Intents

- ▶ Iniciación de actividades programáticamente:
 - ▶ `startActivity(Intent intent,...)`
 - ▶ `startActivityForResult(Intent intent,...)`
- ▶ Resolución de Intents
 - ▶ Explícita, estableciendo la componente del Intent
 - ▶ Implícita, a partir de los datos del Intent y de las características de las actividades instaladas

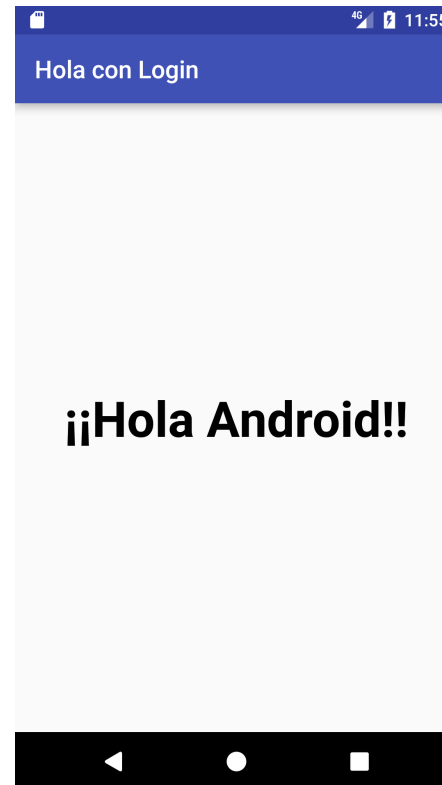
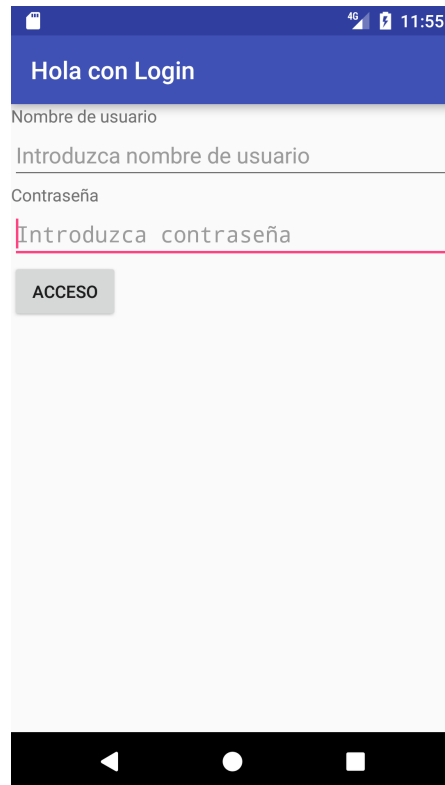
Antes de continuar...

- ▶ Es posible que el dispositivo no disponga de ninguna aplicación para atender el intent que le pasamos a `StartActivity`. En ese caso la llamada fallará y la aplicación abortará su ejecución (crash).
- ▶ Para comprobar que hay una actividad que atenderá el intent, podemos llamar a `resolveActivity()`:

```
if (mi_intent.resolveActivity(getPackageManager())!=null){  
    startActivity(sendIntent);  
}
```

Activación explícita

- ▶ Hola con Login: dos actividades
 - ▶ LoginActivity: comprueba usuario y contraseña e inicia HolaAndroid
 - ▶ HolaAndroid: muestra un mensaje en pantalla



HolaConLogin

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

        final EditText usuario = (EditText) findViewById(R.id.usuario_edittext);
        final EditText password = (EditText) findViewById(R.id.password_edittext);
        final Button boton = (Button) findViewById(R.id.login_button);

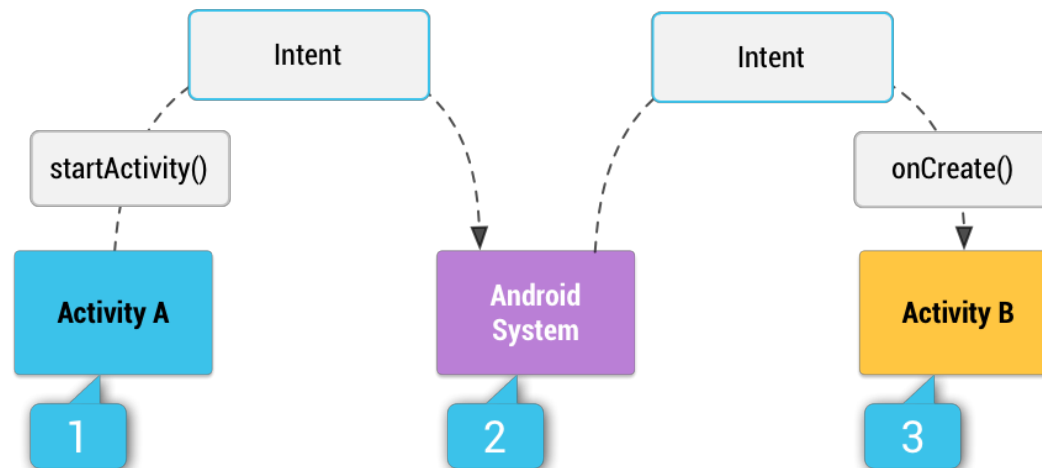
        boton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                if (compruebaPassword(usuario.getText(), password.getText())) {
                    Intent holaAndroidIntent = new Intent(LoginActivity.this, HolaAndroid.class);
                    startActivity(holaAndroidIntent);
                } else {
                    usuario.setText("");
                    password.setText("");
                }
            }
        });
    }
    ...
}
```

Activación implícita

- ▶ Cuando la actividad que queremos activar no se cita de forma explícita, Android tiene que encontrar qué actividades del dispositivo pueden atender el Intent.

```
Intent geoIntent = new Intent(android.content.Intent.ACTION_VIEW,  
    Uri.parse("geo:0,0?q=" + direccion));  
startActivity(geoIntent);
```

Resolución de Intents



Resolución de Intents

- ▶ Usa dos tipos de información:
 - ▶ Intent que describe la operación deseada
 - ▶ Filtros de Intents, que describen qué operaciones puede gestionar una actividad.
 - ▶ Se especifican normalmente en `AndroidManifest.xml` (también se puede hacer programáticamente)
- ▶ Datos empleados en la resolución de Intents:
 - ▶ Action
 - ▶ Data (URI y Type)
 - ▶ Category

Especificación de filtros de Intents

```
<activity ...>
  <intent-filter ...>
    ...
    <action android:name="android.intent.action.DIAL" />
    ...
  </intent-filter>
  ...
</activity>
```

Especificación de filtros de Intents

- ▶ De hecho, todas las aplicaciones tienen un intent-filter:

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

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

para poder ser iniciadas desde el launcher que Android Studio incluye por defecto en AndroidManifest.xml

Más información en el intent-filter

```
<intent-filter ...>
  ...
  <data
    android:mimeType = "string"
    android:scheme = "string"
    android:host = "string"
    android:port = "string"
    android:path = "string"
    android:pathPattern = "string"
    android:pathPrefix = "string"
  />
  ...
</intent-filter>
```

<http://developer.android.com/intl/es/guide/components/intents-filters.html>

Ejemplo

Filtro para una actividad que puede mostrar mapas:

```
<intent-filter ...>  
    ...  
    <data android:scheme="geo"/>  
    ...  
</intent-filter>
```

Especificando una categoría en el filtro

```
<intent-filter ...>  
    ...  
    <category android:name="string" />  
    ...  
</intent-filter>
```

Filtro de Google Maps

```
<intent-filter ...>  
    <action android:name = "android.intent.action.VIEW" />  
    <category android:name = "android.intent.category.DEFAULT"/>  
    <category android:name = "android.intent.category.BROWSABLE"/>  
    <data android:scheme = "geo"/>  
</intent-filter>
```

Nota: Para recibir intents implícitos, la actividad debe incluir un intent-filter con la categoría `android.intent.category.DEFAULT` (para que pueda ser elegida como aplicación por defecto). Todos los intents implícitos se consideran de la categoría `DEFAULT`.

Prioridad

```
<intent-filter android:icon="drawable resource"  
               android:label="string resource"  
               android:priority="integer" >
```

. . .

```
</intent-filter>
```

- ▶ **android:priority** ayuda a Android a elegir entre aplicaciones que pueden gestionar un Intent
- ▶ Valor entre -1000 y 1000
- ▶ Un mayor valor indica mayor prioridad

Mostrar un App Chooser

- ▶ Si queremos dar al usuario la opción de elegir qué aplicación usar para atender un intent, debemos mostrar un Chooser Dialog.

```
Intent sendIntent = new Intent(Intent.ACTION_SEND);  
...  
String title = getResources().getString(R.string.chooser_title);  
// Create intent to show the chooser dialog  
Intent chooser = Intent.createChooser(sendIntent, title);  
  
if (sendIntent.resolveActivity(getPackageManager()) != null) {  
    startActivity(chooser);  
}
```