

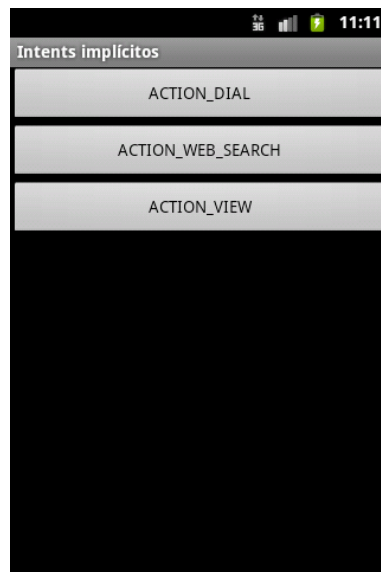
# Intents y navegación entre actividades - Ejercicios

## Índice

1 Intents implícitos.....	2
2 Intents explícitos.....	3
3 Navegación .....	4
4 Actividades en diferentes tareas (*).....	4

## 1. Intents implícitos

En este primer ejercicio vamos a modificar una aplicación que se proporciona en las plantillas de la sesión y que se llama *Implicitos*. La interfaz de la actividad principal de esta aplicación está compuesta por tres botones. El objetivo es que al pulsar cada uno de ellos se lance un `Intent` implícito que sea atendido por alguna de las actividades nativas del sistema.



Interfaz de la aplicación Implicitos

Crea un manejador para todos los botones, de tal forma que cada uno de ellos se encargue de solicitar la realización de la acción indicada. No olvides inicializar los `Intents` con los datos requeridos para que se pueda llevar a cabo la tarea solicitada. Los datos que necesitarás para cada tipo de acción son los siguientes:

- `ACTION_DIAL`: una URI correspondiente a un teléfono. Se puede construir a partir de una cadena con la sintaxis `"tel:[TELÉFONO]"` (donde `[TELÉFONO]` es un número de teléfono) mediante el método `Uri.parse(String)`.
- `ACTION_WEB_SEARCH`: un parámetro extra que será una cadena, cuyo identificador será `SearchManager.QUERY` y cuyo contenido será el texto a buscar en Google.
- `ACTION_VIEW`: en este caso utilizaremos la acción para visualizar nuestra lista de contactos. Para ello debemos añadir al `Intent` la URI `content://contacts/people/`.

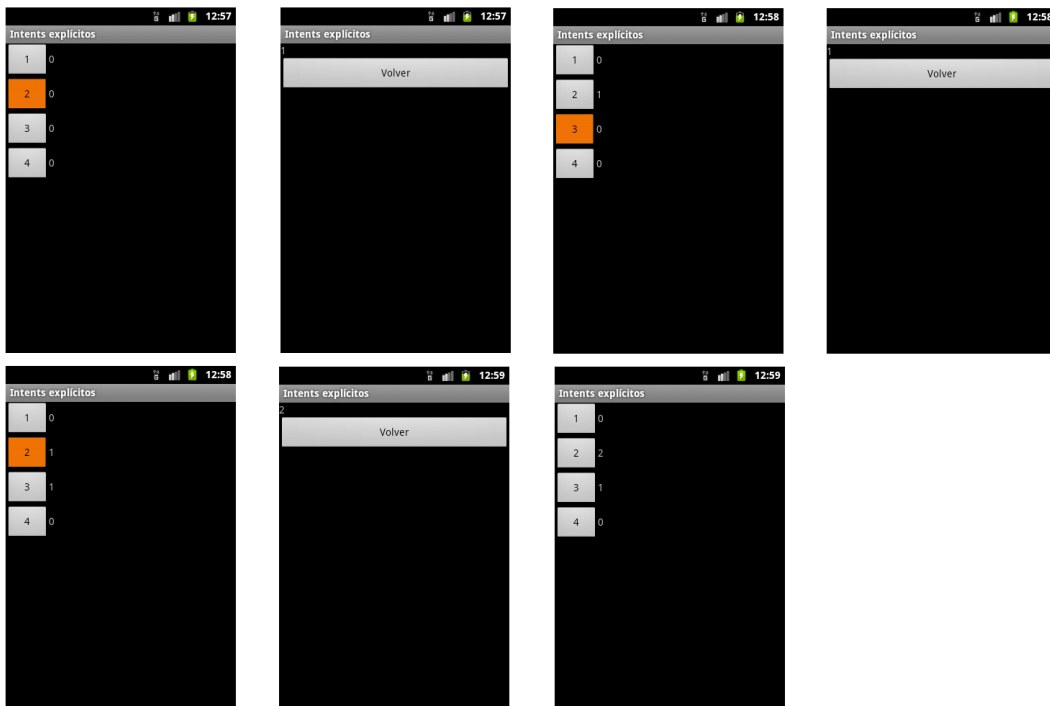
### Nota:

Cuando estés ejecutando el programa, abre el historial de tareas con una pulsación larga de la tecla *HOME*. ¿Puedes ver en este listado alguna de las actividades lanzadas para resolver tus peticiones?

## 2. Intents explícitos

Se te proporciona un proyecto en las plantillas de la sesión de nombre *Explicitos*. La aplicación consta de dos actividades: *Principal* y *Secundaria*. La actividad *Principal* contiene en su interfaz cuatro botones. Al lado de cada uno de estos botones hay un elemento de tipo `TextView`, cuyo valor inicial es cero. Sea cual sea el botón que se pulse se mostrará siempre la actividad *Secundaria*. Esta actividad leerá del `Intent` recibido el valor numérico situado al lado del botón a partir del cual se lanzó desde *Principal* y le sumará uno, mostrándose en la pantalla del dispositivo. Al pulsar el botón de *Secundaria* se deberá volver a *Principal*. En *Principal* se actualizará el valor a la derecha del botón que se pulsó para llamar a *Secundaria*, tomando el valor que se estaba mostrando en dicha actividad antes de pulsar su botón *Volver*.

La siguiente figura muestra un ejemplo de uso, de izquierda a derecha y de arriba a abajo. En primer lugar se pulsa el segundo botón, con lo que se muestra la actividad *Secundaria* mostrando como valor 1 (ya que el valor a la derecha del segundo botón era 0). Cuando pulsamos en volver, el `Intent` de respuesta de *Secundaria* devuelve ese valor 1 que *Principal* utiliza para actualizar el valor a la derecha del segundo botón. Luego se repite el proceso pulsando el tercer botón, y por último se muestra el resultado al volver a utilizar el segundo botón.



Ejemplo de uso de la aplicación de Intents explícitos

**Nota:**

Comprueba que ocurre al volver a lanzar la actividad tanto si vuelves a la pantalla de inicio de Android por medio de la tecla *BACK* como si lo haces por medio de la tecla *HOME* de tu dispositivo.

**Nota:**

Para cambiar el texto mostrado por un elemento de tipo `TextView` debes hacer uso de su método `setText`, que recibe como parámetro un dato de tipo `CharSequence`.

### 3. Navegación

Otro proyecto que te proporcionamos con las plantillas de la sesión es *Navegacion*. Este proyecto consta de cuatro actividades: *A*, *B*, *C* y *D*. Cada actividad contiene un botón etiquetado con el nombre del resto de actividades de la aplicación. Vamos a modificar la manera en la que se navega entre estas actividades, de tal forma que vamos a suponer que existe una actividad principal (actividad *A*) y una serie de actividades secundarias (actividades *B*, *C* y *D*). Cuando se vuelva a la actividad *A* desde cualquiera de las demás, se deberá conseguir que dicha actividad sea la única que quede en la pila. Sigue los siguientes pasos:

- Añade manejadores de evento para los botones de la actividad *A*, de tal forma que cada uno de ellos lance la actividad con la que está etiquetado dicho botón.
- Repite estos mismos pasos con las actividades *B*, *C* y *D*.
- Haz los cambios necesarios para que nunca haya más de una instancia de la actividad *A* en la pila de actividades. Investiga cual de los flags debes añadir al `Intent` a la hora de lanzar la actividad.
- Haz los cambios necesarios para que al pulsar el botón *BACK* estando en la actividad *B*, *C* o *D* se vuelva siempre a la actividad *A*. Al pulsar el botón *BACK* desde la actividad *A* se saldrá de la aplicación. Esto quiere decir que durante la ejecución de la aplicación su tarea contendrá o tan sólo una actividad (la actividad *A*) o como mucho dos (la actividad *A* y cualquiera de entre las otras tres).

**Nota:**

Para poner a uno algún flag de un objeto de la clase `Intent` hacemos uso del método `setFlags`, que recibe como parámetro uno o más identificadores de flag separados por el operador `|` (*OR* binario)

### 4. Actividades en diferentes tareas (\*)

Se te proporciona el proyecto *Tareas* con dos actividades: *Principal* y *Web*. La actividad

*Principal* contiene un botón que servirá para lanzar la actividad *Web*. La actividad *Web* contiene una vista de tipo *WebView*, que trataremos en sesiones posteriores, y que se encarga de mostrar contenido web.

Sigue los siguientes pasos:

- Haz los cambios necesarios para que se pueda lanzar la actividad *Web* desde la actividad *Principal*.
- Lanza la aplicación y pulsa el botón para mostrar la actividad *Web*. Pulsa ahora el botón *HOME* del dispositivo, con lo que volverás a la pantalla principal de Android. A continuación mantén pulsado el botón *HOME* hasta que aparezca el listado de tareas recientes. Selecciona *Tareas*; al hacerlo se deberá mostrar de nuevo la actividad *Web*. Para salir de la aplicación deberás pulsar dos veces el botón *BACK* del dispositivo. Ese es el comportamiento normal de una aplicación Android.
- Haz los cambios necesarios para que cuando se lance la actividad *Web* lo haga en una tarea nueva. Además, debes conseguir que sea la única actividad de esta nueva tarea. Escoge el valor adecuado para el atributo *launchMode*.
- Vamos a repetir el experimento anterior. Ejecuta la aplicación, y pulsa el botón para lanzar la actividad *Web*. Ahora pulsa *HOME*, y vuelve a enviar *Tareas* a primer plano seleccionando la aplicación en el listado que aparece al mantener durante un tiempo pulsada la tecla *HOME*. Verás que en esta ocasión también se mostrará la última actividad activa, la actividad *Web*. La diferencia es que ahora sólo hace falta pulsar *BACK* una vez para abandonar la aplicación.

¿Qué ha sucedido? Las actividades *Principal* y *Web* se encontraban en tareas distintas. Al volver a la actividad *Web* y darle a *BACK* hemos destruido su tarea asociada, pero no hemos vuelto a la tarea que contiene la actividad *Principal*.

**Nota:**

Los valores *standard* y *singleTop* del atributo *launchMode* son los más habituales en aplicaciones Android. Por otra parte, *singleTask* y *singleInstance* simplemente no son adecuados para la inmensa mayoría de aplicaciones. Sea cual sea el valor de *launchMode* que se utilice, es conveniente hacer pruebas de navegación en nuestra aplicación, observando cómo se comporta conforme vamos pulsando el botón *BACK*.

- Veamos como lanzar correctamente la actividad *Web* en una tarea independiente sin confundir al usuario. En el *Manifest* de la aplicación dale como valor al atributo *taskAffinity* de la actividad *Web* la cadena "es.ua.jtech.android.tareas2".
- Ejecuta la aplicación y pulsa el botón para mostrar la actividad *Web*. Ahora pulsa el botón *HOME*. Si ahora pulsas el botón *HOME* durante un rato, aparecerán dos tareas con el nombre *Tareas* en el listado. Una es la tarea que contiene la actividad *Principal* y la otra es la tarea que contiene la actividad *Web*.
- Aun así todo sigue siendo un poco confuso, porque las dos tareas tienen el mismo nombre de aplicación. Cambia en el *Manifest* el valor del atributo *android:label* de la tarea *Web* a *Mi Navegador*.
- Ahora, cuando repitas el experimento anterior y muestres el listado de aplicaciones

recientes verás las aplicaciones correspondientes a las dos tareas mostrándose con nombres diferentes.

- Activemos ahora el flag `FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS` en el `Intent` que utilizamos para lanzar *Web* desde principal. Ejecuta la aplicación, pulsa el botón para lanzar la actividad *Web*, y pulsa el botón *HOME*. Haz ahora una pulsación prolongada del botón *Home*. ¿Aparece la tarea *Mi Navegador* en el listado de tareas recientes?

