



# Intent Implícito

## Primera práctica

# Intent en Android: CargarTexto

## Código fuente

El código fuente se encuentra en la dirección github del proyecto:

<https://github.com/aaronperez/CargarTextos>

## Funcionamiento de la aplicación

Cargartexto es una aplicación sin lanzador que se ofrece al intentar abrir un archivo de texto plano para mostrarlo. Esta app obtiene el contenido del archivo y lo muestra en un *EditText* dentro de un *scroll* donde podemos modificarlo y posteriormente pulsando un icono salvar el archivo y salir.

Esta es la imagen inicial al cargar la app.:



## Menús

La aplicación dispone de un único proceso, Agregar. Éste estaría en el menú inicial y siempre visible mediante un icono en la barra de navegación.



## Implementación

### Manifiesto

En *AndroidManifest* usamos el permiso de escritura en el almacenamiento externo que es de dónde recibiremos el archivo. Además en *icon* he cambiado el icono por otro creado editado por mí. Además modificamos el intent-filter quitando el launcher añadiendo el action para que al intentar el dispositivo abrir un archivo de texto plano ofrezca nuestra aplicación.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

<application
    .....
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <data android:mimeType="text/plain"/>
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
</application>
```

## Recursos

### Drawable

Aquí hemos añadido 2 imágenes que servirán de iconos a la app y al elemento guardar.

### Layout

El layout de la actividad principal contiene un *EditText* dentro de un *scroll* (*activity\_main.xml*).



## Menú

El menú de la actividad tiene una sola opción para guardar, que tiene la propiedad *showAsAction="ifRoom"* (la muestra si hay espacio disponible) que hace que se mantenga visible en la barra de navegación (*main.xml*).

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context=".Main" >
    <item android:id="@+id/save"
          android:icon="@drawable/save"
          android:title="@string/guardar"
          android:showAsAction="ifRoom" />
</menu>
```

## Código fuente

### Clase Main

El método *onCreate ()* visualiza en el Layout principal e inicia *EditText* y llama al metodo *leer()*.

El método *onOptionsItemSelected ()* infla el menú principal de la actividad (*main.xml*).

El método *onOptionsItemSelected ()* contiene la opción del menú principal de guardar y se encarga de ejecutarla cuando la seleccionamos.

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle presses on the action bar items
    switch (item.getItemId()) {
        case R.id.save:
            guardar();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

El método *tostada ()* lo usamos para mostrar un mensaje emergente desde un *int*.

Los métodos *leer ()* lo llamamos desde el *onCreate* y sirven para recoger mediante un *intent* recoger el archivo que ha llamado a la app y mostrarlo en el *EditText*.

```
public void leer() {
    Intent intent = getIntent();
    Uri data = intent.getData();
    texto = data.getPath();
    f=new File(texto);
    try {
        BufferedReader br = new BufferedReader(new FileReader(f));
        String linea;
        StringBuilder texto = new StringBuilder("");
        while ((linea = br.readLine()) != null) {
            texto.append(linea+"\n");
        }
        br.close();
        etT.setText(texto);
    } catch (IOException e) {}
}
```

El método *guardar ()* se lanza al pulsar su icono y recoge en un *String* el contenido del *EditText* y a través de *OutputStreamWriter* reescribirlo en el archivo del que lo recibimos y cerrando la aplicación.

```
public void guardar(){
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    try {
        texto = etT.getText().toString();
        OutputStreamWriter escribir = new OutputStreamWriter(new FileOutputStream(f));
        escribir.write(texto);
        escribir.flush();
        escribir.close();
        finish();
        tostada(R.string.guardado);
    } catch(IOException e){}
}
```