Muchas veces se echa en falta cuando trabajamos con Android capacidad de Dependency Inyection en cuanto a la gestión de vistas se refiere. Normalmente cuando en Android disponemos de dos botones que queremos pulsar el código para referenciarlos se apoya en un el método findViewById().

```
public class MainActivity extends AppCompatActivity {

 Button boton1;
 Button boton2;


 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);

 boton1= (Button) findViewById(R.id.boton1);
 boton2=(Button)findViewById(R.id.boton2);

 boton1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {


Toast.makeText(getApplicationContext(),"boton1",Toast.LENGTH_SHORT).sh
ow();


 }
```

```
 });

 boton2.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {


Toast.makeText(getApplicationContext(),"boton2",Toast.LENGTH_SHORT).sh
ow();


 }
 });


 }
```

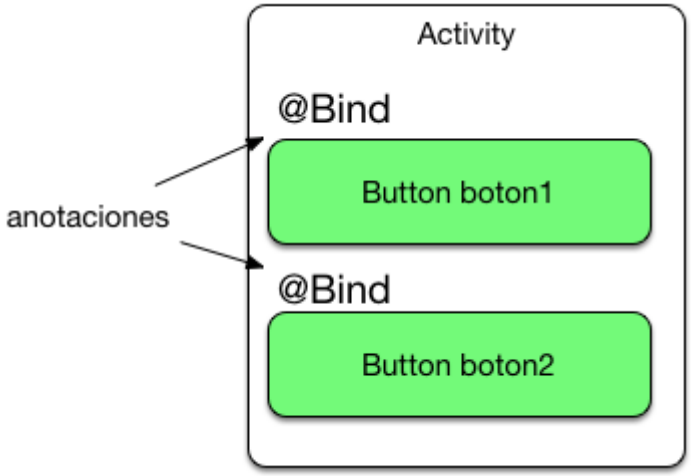
Android Dependency Inyection

Como se puede observar el código a construir es un poco engorroso , sobre todo si estamos acostumbrados a frameworks de inyección de dependencia. Para solventar este problema podemos usar Butter Knife una librería que nos aporta a través del uso de anotaciones una solución rápida al problema. Para ello añadiremos a nuestro proyecto de Android la dependencia de la librería.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.2.1'
    compile 'com.android.support:design:23.2.1'
```

```
    compile 'com.jakewharton:butterknife:7.0.1'
}
```

Una vez esta la librería configurada , hacemos uso de sus anotaciones @Bind para dejar inyectados los campos.



El código es muy similar al anterior pero no hace falta el uso de findViewById.

```
package botones.arquitecturajava.com.miaplicacion2;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
```

```
import android.widget.Button;
import android.widget.Toast;

import butterknife.Bind;

public class MainActivity extends AppCompatActivity {

@Bind(R.id.boton1)
Button boton1;
@Bind(R.id.boton2)
Button boton2;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
ButterKnife.bind(this);

boton1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Toast.makeText(getApplicationContext(), "boton1", Toast.LENGTH_SHORT);

}
});
boton2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Toast.makeText(getApplicationContext(),"boton2",Toast.LENGTH_SHORT);

}
});
```

```
}
```

Acabamos de utilizar las anotaciones de ButterKnife para inyectar las dependencias de las vistas de Android de una forma mucho más sencilla. Recordemos que el sistema que usa ButterKnife para gestionar las Android Dependency Inyection , es en tiempo de compilación. Esto facilita su integración con la mayor parte de las soluciones .

Otros artículos relacionados: Android Intents , Android Eventos