

## Ejercicio 2.1

- Importar y exportar aplicaciones
- Para importar proyectos de Android a Android studio
  - Buscar la aplicación en el repositorio: Github, Bitbucket, GoogleCode, etc.
  - Importar en Android Studio
  - Corregir errores de Gradle
  - Importar librerías de Android

LBS 2014-2015

1

## Ejercicio 2.1

<https://bitbucket.org/jmunoz/android-example-el tiempo/>

Click en Downloads

Download repository

Overview

Last updated	2012-04-05	1	0
Language	Java	Branch	Tags
Access level	Read (revoke)	0	2
		Forks	Watchers

Recent activity

- jmunoz/android-example-el tiempo  
Repository watched  
Jorge Gaspar Saez Salinas - 2012-04-17
- velho: Initial commit with code for the ...  
Commit pushed to jmunoz/android-example-el tiempo  
Javier Muñoz - 2012-04-05
- jmunoz/android-example-el tiempo  
Repository created  
Javier Muñoz - 2012-04-05

Downloads

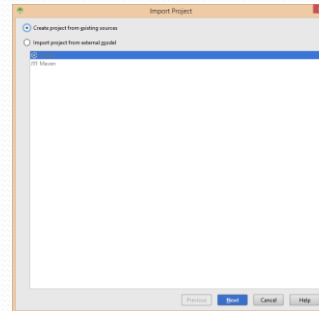
name	Size	Uploaded by	Downl
<a href="#">Download repository</a>	16.9 KB		

LBS 2014-2015

2

## Ejercicio 2.1

- Importar y exportar aplicaciones
- Para importar proyectos de Android a Android estudio

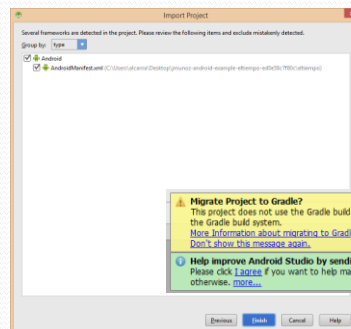
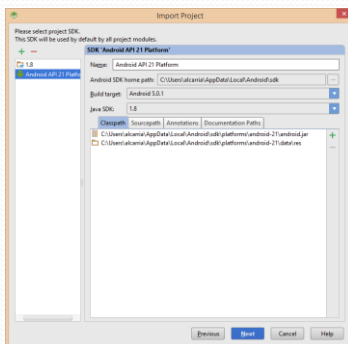


LBS 2014-2015

3

## Ejercicio 2.1

- Importar y exportar aplicaciones
- Para importar proyectos de Android a Android estudio



LBS 2014-2015

4

## Ejercicio 2.1

- Corrección de errores
  - Android Studio: Gradle:
    - *Failed to find target android-7*
  - Ver app/build.gradle
    - Cambiar `compileSdkVersion` y `targetSdkVersion` a nuestra última versión de API de Android.
    - Para Android 5.0 es el API **21**.

LBS 2014-2015

5

## Ejercicio 2.1

- Una vez importado y ejecutado el proyecto vemos que no funciona, por qué?
- El icono del clima se ve muy pequeño (está hecho para pantallas de poca resolución), ¿cómo lo arreglaríamos?

- Resultado esperado:

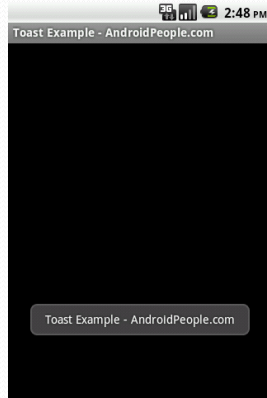


LBS 2014-2015

6

## Ejercicio 2.2

- Creación de actividad con botón y toast



LBS 2014-2015

7

## Ejercicio 2.2

- Crear un botón
  - Utilizamos el asistente gráfico para crear un botón en el layout

```
<Button android:id="@+id/Boton1"
  android:text="Púlsame"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:onClick="myMethod" />
```

- Copiar en la clase principal

```
public void myMethod(View v) {
    //Código a Ejecutar... Ej. Llamar a servidor web, capturar
    datos etc..  button.setEnabled(false);
}
```

LBS 2014-2015

8

## Ejercicio 2.2

- Crear un botón
  - En R.java podremos ver el Nuevo identificador que se le da al botón

```

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class id {
        public static final int BtnBotonI=0x7f050000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
        public static final int main2=0x7f030001;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}

```

LBS 2014-2015

9

## Ejercicio 2.2

- Crear una notificación mediante la clase Toast
  - En nuestra clase, en el método que ejecutamos al pulsar un botón:

```

Context context = getApplicationContext();
CharSequence text = "Hola en notificación";
int duration = Toast.LENGTH_SHORT;

```

```

Toast toast = Toast.makeText(context, text, duration);
toast.show();

```

- O más resumido:

```

Toast.makeText(this, "Nuestro texto", Toast.LENGTH_LONG).show();

```

LBS 2014-2015

10

## Ejercicio 2.3a

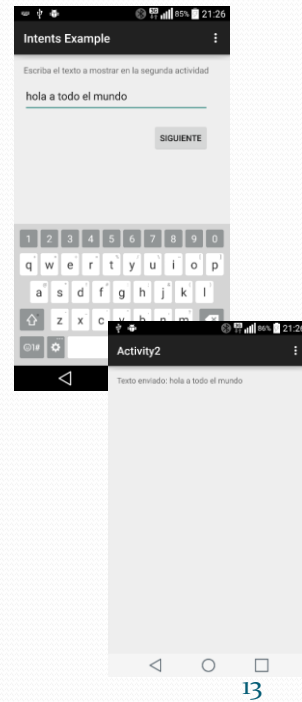
- Paso de información entre actividades mediante intents
- En una actividad creamos un EditText para escribir un nombre
- También creamos un botón
- Mediante un intent se lo pasamos a una segunda actividad, que lo mostrará en un TextView

## Ejercicio 2.3a

- Pistas
  - Crear un nuevo proyecto con los siguientes elementos:
  - Actividad 1: MainActivity.java
  - Actividad 2: Activity2.java
  - Layout 1: activity\_main.xml
  - Layout 2: activity\_activity2.xml

## Ejercicio 2.3a

- Pistas
  - *Agregar a: activity\_main.xml*
    - *Un TextView todo lo ancho de la pantalla*
    - *Un EditText todo lo ancho de la pantalla*
  - *En activity\_activity2.xml*
    - *Crear un TextView*
  - *En strings.xml*
    - *Creamos nuevos textos para mostrar en los TextView de las dos actividades*



LBS 2014-2015

13

## Ejercicio 2.3a

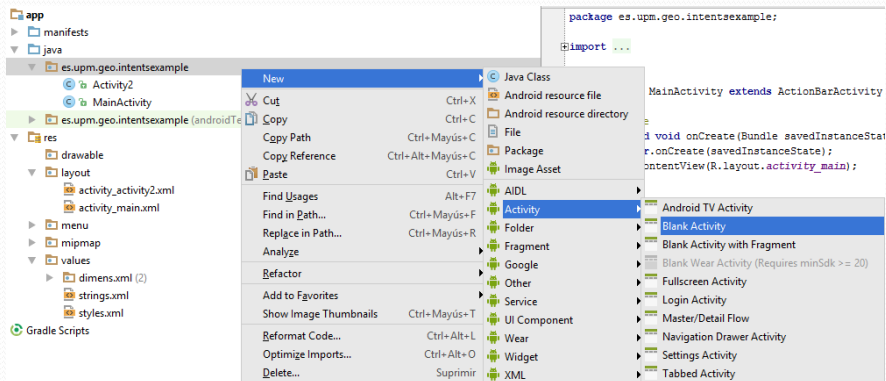
- Pistas
  - *En MainActivity*
    - *Creamos el método a ejecutar cuando se pulsa el botón*
    - *En este método creamos un intent y pasamos el text que el usuario escribe*
    - *Arrancamos la nueva actividad Activity2*
  - *En Activity2*
    - *En onCreate: Recuperamos el TextView*
    - *Recuperamos la información pasada en el Intent*
    - *Construimos el mensaje a mostrar y lo pasamos al TextView.*

LBS 2014-2015

14

## Ejercicio 2.3a

- El asistente de nueva actividad facilita mucho
  - Tiene en cuenta un nuevo layout
  - Registra la actividad en el manifiesto



LBS 2014-2015

15

## Ejercicio 2.3b

- Modificar el ejercicio anterior para añadir algunos intents implícitos
  - Probamos con el de mapas

```
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setData(geoLocation);
if (intent.resolveActivity(getPackageManager() != null) {
    startActivity(intent);
}
```

- Siendo geolocation
  - `geo:latitude,longitude?z=zoom`
  - `geo:o,o?q=UPM+Campus+SUR`
- Ver más ejemplos en:
  - <http://developer.android.com/guide/components/intents-common.html>

LBS 2014-2015

16



## Ejercicio 2.4a

- Uso de servicios:
  - Importar proyecto *NotificationService*
  - Estudiar su funcionamiento
  - Ejecutar aplicación y ver qué hace el servicio cuando está arrancado y cuando se para.

## Ejercicio 2.5

- Creación de un broadcast receiver para que reaccione a cambios en el nivel de batería:
  - Crear un Nuevo Proyecto, y una Actividad con un layout que tenga un textview.
  - Agregar Permiso Battery Statt
  - Al final del onCreate
  - `IntentFilter filter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);`
  - `registerReceiver(battery_receiver, filter);`

## Ejercicio 2.5

- Creamos el broadcast receiver

```
private BroadcastReceiver battery_receiver = new BroadcastReceiver() {

    @Override
    public void onReceive(Context context, Intent intent) {
        int nivel = intent.getIntExtra(BatteryManager.EXTRA_LEVEL, 0);
        String tecnologia =
            intent.getStringExtra(BatteryManager.EXTRA_TECHNOLOGY);
        int estado = intent.getIntExtra(BatteryManager.EXTRA_STATUS, 0);

        String info = "Nivel: " + nivel + "%\n";
        info += ("Tecnologia: " + tecnologia + "\n");
        info += ("Estado: " + obtenerEstado(estado) + "\n");

        TextView tv = (TextView) findViewById(R.id.textview);
        tv.setText(info);
    }
};
```

LBS 2014-2015

22

## Ejercicio 2.5

- Creamos el broadcast receiver

En el `onDestroy` no olvidar el  
`unregisterReceiver(battery_receiver);`

**Método obtenerEstado**

```
@Override
protected void onDestroy() {
    super.onDestroy();
    unregisterReceiver(battery_receiver);
}

private String obtenerEstado(int estado){
    String plugType = "Desconocido";

    switch(estado)
    {
        case BatteryManager.BATTERY_STATUS_DISCHARGING: plugType = "Descargando";break;
        case BatteryManager.BATTERY_STATUS_CHARGING: plugType = "Cargando";break;
        case BatteryManager.BATTERY_STATUS_FULL: plugType = "Batería llena";break;
        case BatteryManager.BATTERY_STATUS_NOT_CHARGING: plugType = "No cargando";break;
    }
    return plugType;
}
```

LBS 2014-2015

23