

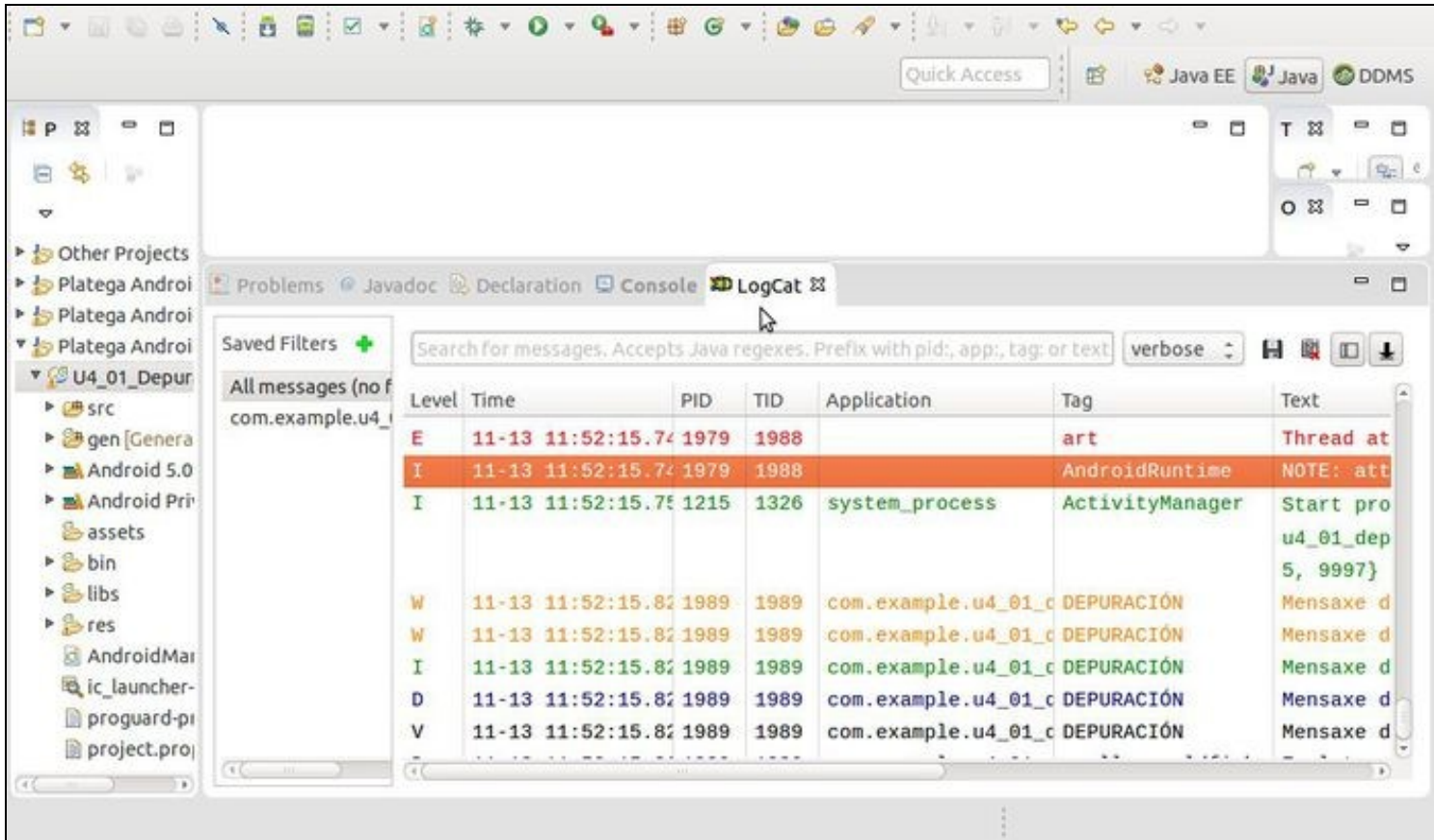
Depuración: LogCat

Sumario

- 1 Introducción
- 2 Caso práctico
 - ♦ 2.1 O XML do Layout
 - ♦ 2.2 O código Java da aplicación

Introdución

- Unha das técnicas para realizar depuración e seguimento das aplicacións é a creación de logs (rexistros) en tempo de execución.
- En Android úsase a clase **Log**, que posúe un conxunto de métodos estáticos que nos van axudar no proceso de depuración/seguimento.



The screenshot shows an IDE interface with the LogCat window open. The LogCat window displays a table of log messages with the following columns: Level, Time, PID, TID, Application, Tag, and Text. The messages are as follows:

Level	Time	PID	TID	Application	Tag	Text
E	11-13 11:52:15.74	1979	1988		art	Thread at
I	11-13 11:52:15.74	1979	1988		AndroidRuntime	NOTE: att
I	11-13 11:52:15.75	1215	1326	system_process	ActivityManager	Start pro u4_01_dep 5, 9997}
W	11-13 11:52:15.82	1989	1989	com.example.u4_01_c	DEPURACIÓN	Mensaxe d
W	11-13 11:52:15.82	1989	1989	com.example.u4_01_c	DEPURACIÓN	Mensaxe d
I	11-13 11:52:15.82	1989	1989	com.example.u4_01_c	DEPURACIÓN	Mensaxe d
D	11-13 11:52:15.82	1989	1989	com.example.u4_01_c	DEPURACIÓN	Mensaxe d
V	11-13 11:52:15.82	1989	1989	com.example.u4_01_c	DEPURACIÓN	Mensaxe d

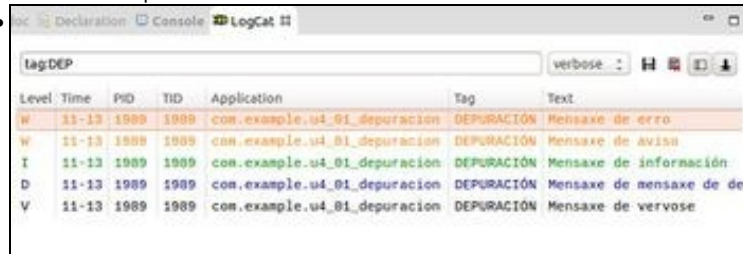
- A imaxe amosa un conxunto de mensaxes que teñen, entre outros, os seguintes campos.
 - ◆ Nivel de criticidade: tipo de mensaxe (Erro, Warning, INfo, Debug, etc)
 - ◆ PID: Código do proceso que produciu a mensaxe.
 - ◆ Tag: Etiqueta que axuda á identificación da mensaxe, moi útil para filtrar resultados.
 - ◆ Mensaxe
- Para cada nivel de criticidade existe un método estático que nós axuda a clasificar as mensaxes que pode producir a aplicación: **e()**, **w()**, **i()**, **d()** e **v()**.
- Cada un dos métodos recibe como parámetro a etiqueta (*tag*) e o texto da mensaxe.
- Referencias:
 - ◆ A Clase Log: <http://developer.android.com/reference/android/util/Log.html>

Caso práctico

- Crear o seguinte proxecto: **U4_01_Depuracion**
- Imos realizar un pequeno proxecto onde amosamos como poder usar o LOG.
- A etiqueta que se está usando no programa é: "DEPURACIÓN"
- E estamos usando un filtro para ver só as etiquetas con ese nome.
- Para ver as mensaxes de log usaremos a ventá **LogCat** que está tanto na perspectiva Java como DDMS.
- Depuración



Lanzamos a aplicación



E no LogCat amosa unha serie de mensaxes con distintos niveis de criticidade.

Estas foron creadas dentro do método onCreate().



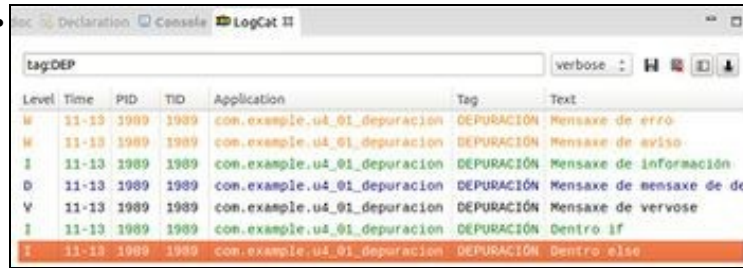
Escribimos un valor correcto



E no LogCat vemos por anda a aplicación ...



Escribimos un valor incorrecto



E volvemos ver por onde anda a aplicación ...

O XML do Layout

- A continuación amósase o Layout da aplicación:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.u4_01_depuracion.U4_01_Depuracion" >

    <EditText
        android:id="@+id/et"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Introduce o século no que vives"
        android:inputType="number" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Preme aquí"
        android:onClick="onButtonClick" />

</LinearLayout>
```

O código Java da aplicación

```
package com.example.u4_01_depuracion;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class U4_01_Depuracion extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u4_01__depuracion);
        Log.e("DEPURACIÓN", "Mensaje de erro");
        Log.w("DEPURACIÓN", "Mensaje de aviso");
        Log.i("DEPURACIÓN", "Mensaje de información");
        Log.d("DEPURACIÓN", "Mensaje de mensaxe de depuración");
        Log.v("DEPURACIÓN", "Mensaje de verbose");
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
```

```

getMenuInflater().inflate(R.menu.u4_01__depuracion, menu);
return true;
}

public void onClick(View v) {
    EditText et = (EditText) findViewById(R.id.et);

    if (et.getText().toString().equals("21")) {
        Toast.makeText(this, "Tes os pes na terra ", Toast.LENGTH_SHORT).show();
        Log.i("DEPURACIÓN BOTÓN", "Dentro if");
    } else if (et.getText().toString().equals("")) {
        Toast.makeText(this, "Non escribiches nada", Toast.LENGTH_SHORT).show();
        Log.i("DEPURACIÓN BOTÓN", "Dentro else if");
    } else
    {
        Toast.makeText(this, "Estás noutro planeta", Toast.LENGTH_SHORT).show();
        Log.i("DEPURACIÓN BOTÓN", "Dentro else");
    }
}
}
}

```

- **Liñas 18-22:** A aplicación emite mensaxes a través dos métodos estáticos con distintos niveis de criticidade.
 - ◆ A etiqueta (tag) é o primeiro campo.
- **Liñas 38,42,48:** emisión de mensaxes que nos axudan a saber porque parte do código se atopa a aplicación.