

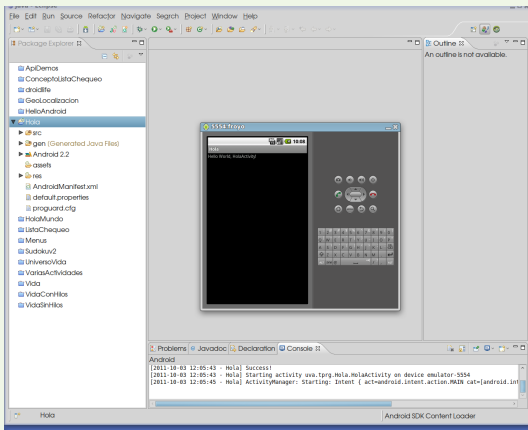
## Tecnología de Programación

## Hola, Mundo

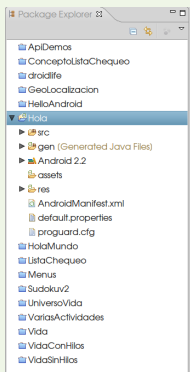
Félix Prieto

Curso 2011/12

## Eclipse y el dispositivo virtual en ejecución



## Elementos de un proyecto Android



- **src:** Código java del proyecto organizado en paquetes
- **gen:** Código java generado por las herramientas de desarrollo
- **res:** Recursos del proyecto almacenados en formato XML
- **AndroidManifest.xml:** Identificación, propiedades de la aplicación, permisos requeridos para su ejecución,...

## La primera Actividad

```

package uva.tprg.Hola;

import android.app.Activity;
import android.os.Bundle;

public class HolaActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}

```

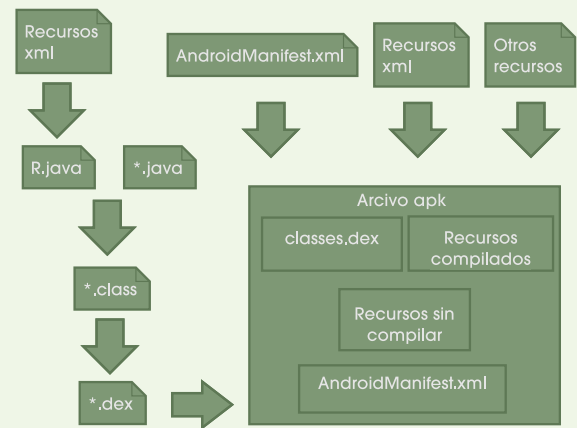
## «Hola, Mundo» en Android

- Elaborar una aplicación «Hola, Mundo» para Android es trivial
  - Abrimos eclipse
  - File -> New -> Android Project
  - Project name: Hola
  - Build target: Android 2.2
  - Package name: uva.tprg.Hola
  - Create Activity: HolaActivity
- Arrancar un dispositivo virtual o conectar un dispositivo físico
- Run -> Run -> Android Application
- ... Y «Hola, Mundo» funciona de modo «mágico»

## Elementos básicos de eclipse

- En eclipse podemos utilizar varias vistas simultáneas
- Un conjunto de vistas constituyen una perspectiva sobre el proyecto
- Disponibles varias perspectivas «predefinidas» (Java, DDMS, Debug, ...)
- Podemos cambiar la configuración por defecto de una Perspectiva
- Podemos recuperar la configuración por defecto de una perspectiva
- Elementos básicos de la perspectiva «Java»
  - Explorador: Selección de Aplicaciones o elementos dentro de ellas
  - Editor
  - JavaDoc: Muestra documentación sobre el código en edición
  - Consola: Mensajes relativos a la compilación y/o ejecución
  - Esquema: Estructura del elemento en edición

## Diagrama general de la construcción



## Clases y ficheros en Java

- Las clases se organizan en «paquetes» que agrupan clases relacionadas
- package se utiliza para declarar el paquete a que pertenecen las clases del fichero
- Para que una clase pueda ser usada fuera de su paquete hay que declararla como pública
- En cada fichero sólo puede existir una clase pública, con el mismo nombre que el fichero
- La importación de otras clases es explícita, mediante la clausula import
- Eclipse inserta los import necesarios (shift-control-o)

## Métodos, atributos y visibilidad

- Todos los métodos son funciones, aunque algunos devuelven `void`
- Se admite el polimorfismo sintáctico
- Existen cuatro niveles de visibilidad:
  - Por defecto, visible para las clases del paquete
  - `public`, visible para todas las clases
  - `private`, visible sólo para instancias de la clase
  - `protected`, visible desde clases de paquete o sus descendientes
- Un atributo público, lo es también para escritura, lo que rompe el principio de encapsulación

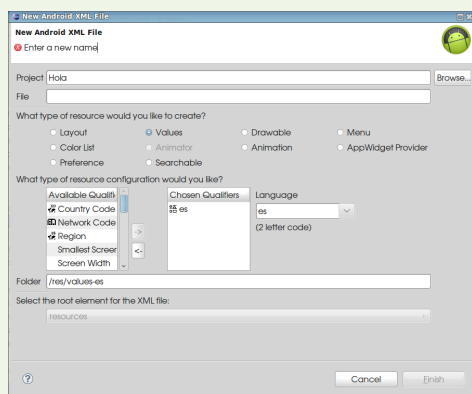
## Definición del primer «layout»

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android=
  "http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <TextView android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="@string/hello" />
</LinearLayout>
```

## Consideraciones sobre el «layout»

- En general es más cómodo utilizar el editor «rico»
- La columna de la izquierda contiene todos los elementos o Vistas que podemos insertar
- La columna de la derecha contiene el árbol de estructura del layout
- Podemos cambiar propiedades de una vista mediante un menú contextual

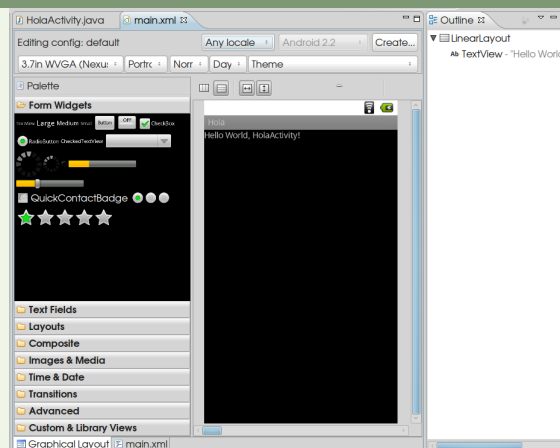
## Creando un nuevo fichero de recursos



## Herencia y redefinición

- Sólo se permite herencia simple entre clases
- `extends` indica la única clase padre de la actual
- La marca `@override` indica que el siguiente método es una redefinición
  - No es obligatorio, pero evita errores relacionados con la redefinición
- `super.onCreate()` hace referencia al precursor del método
- En resumen `HolaActivity` es una clase heredera de `Activity` que redefine el método `onCreate`

## Editando el «layout»



## La estructura del «layout»

- La posición de los elementos siempre es relativa
- En general las vistas se construyen combinando cajas apiladas vertical y horizontalmente
- Las medidas preferidas son `dip` (pixels independientes de la densidad) y `sp` (equivalente para texto)
- No deben aparecer explícitamente:
  - colores
  - tipos de letra
  - cadenas de caracteres
- todos estos elementos aparecen en otros ficheros de recursos

## Ficheros de recursos

- Podemos crear ficheros de recursos de varios tipos
  - Layout, colores, datos, menús,...
- Podemos crear varios ficheros para el mismo recurso
- Se activará uno de ellos en función de un conjunto de propiedades
  - Idioma, tamaño o posición de pantalla,...
- Los ficheros se almacenan en directorios diferentes
- Eclipse es capaz de ocuparse de los detalles

## El siguiente paso

- **Conseguir reacciones a las acciones del usuario**
- **Lanzar otra actividad**
- **Utilizar otros tipos de vista**
- **Ofrecer opciones de configuración de la aplicación**
- **Esto es: Un «Hola, Mundo» con pretensiones**