

# Unidade 2 - Práctica A

## Sumario

- 1 Introducción
- 2 Aspectos xerais
- 3 Deseño
  - ◆ 3.1 Deseño vertical
  - ◆ 3.2 Deseño en apaisado
  - ◆ 3.3 Internacionalización
- 4 Funcionamento
  - ◆ 4.1 Bloque 1: TextEdit, TextView, Botóns, CheckBox, RadioButtons
  - ◆ 4.2 Bloque 2: Spinner
  - ◆ 4.3 Bloque 3: ImageView
  - ◆ 4.4 Bloque 4: Chronómetro e Botón Switch

## Introdución

- A seguinte aplicación é unha única Activity que resume parte do visto na Unidade 2.
- Hai deseño de distintas vistas, Internacionalización e distintos deseños en apaisado e vertical.

## Aspectos xerais

- Todo:
  - ◆ texto,
  - ◆ cor,
  - ◆ tamaño, dimensión, etc.
- debe estar **definido nun recurso XML**, non a *lume* directamente no Layout.
  
- Os **idiomas** usados para a aplicación son:
  - ◆ Por defecto: Inglés
  - ◆ Personalizados:
    - ◇ Español
    - ◇ Galego (Non funciona nun AVD, pero si nun dispositivo real).
  - ◆ Para facilitar a construción dos recursos tipo Strings facilítase o seguinte ficheiro XML en inglés: [Media:strings.xml](#)
  
- Usar o seguinte nome, **I2\_<username>**, para nome do:
  - ◆ Proxecto
  - ◆ Activity
  
- Para facilitar o seguimento da aplicación axúntase o apk para que se se poida instalar e probar.
  - ◆ Observar o nome: U2\_A\_Teacher
  - ◆ Aplicación: [Media:U2\\_A\\_Teacher.apk](#) ?

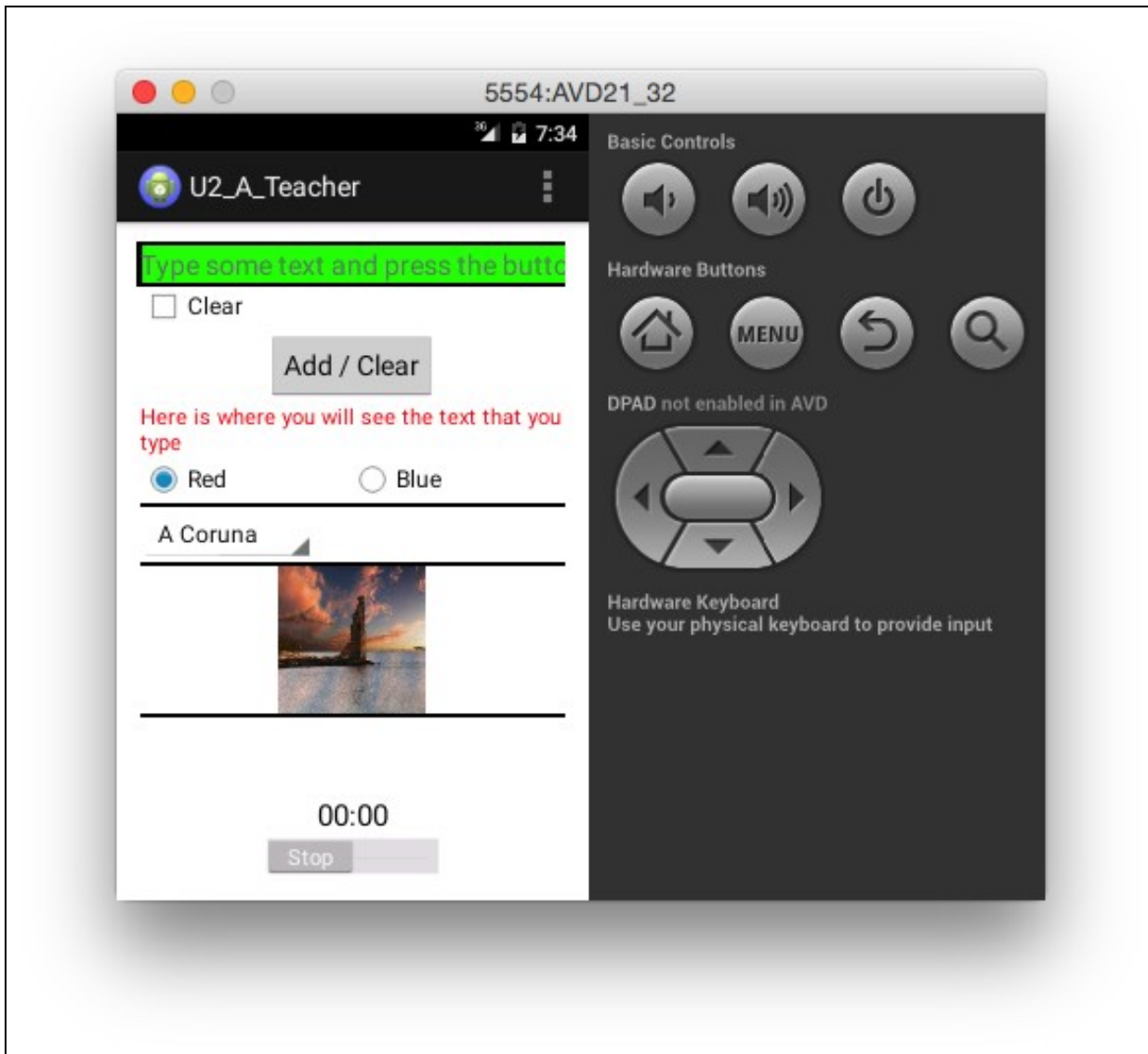
## Deseño

- Comezar antes de nada polo deseño.
- Tratar de aproximarse ao deseño das pantallas que se amosan a continuación.
  - ◆ Posicións dos elementos:
    - ◇ Non se usou ningunha marxe, posición a lume, etc.
    - ◇ Só se usou un recheo no layout principal de 16dp.
- Cores:

♦ Usáronse as cores básicas

• A aplicación está dividida en 4 bloques separados por liñas.

## Deseño vertical



- Observar, botándolle un ollo tamén ao deseño en apaisado:

- ◆ A disposición dos elementos
- ◆ Cales están centrados
- ◆ Cales están aliñados á esquerda
- ◆ Cales están repartidos no peso
- ◆ Cales ocupan todo o ancho
- ◆ Cales están no fondo da pantalla

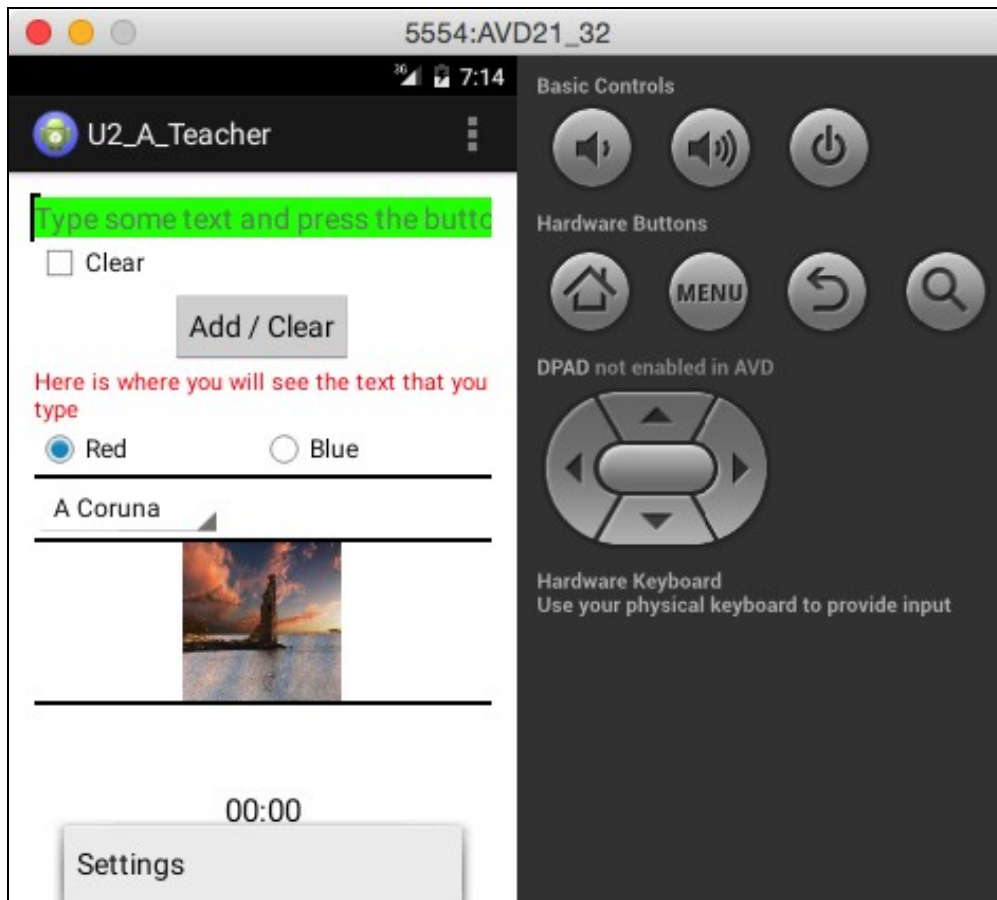
## Deseño en apaisado



- Observar
  - ◆ Non hai imaxe
  - ◆ Só queda unha liña de división con respecto ao deseño en vertical
  - ◆ Os elementos que contén realizan a mesma función que realizan cando o dispositivo en vertical.

## Internacionalización

- Como xa se dixo a aplicación está en 3 idiomas.
- Están traducidos incluso os menús:

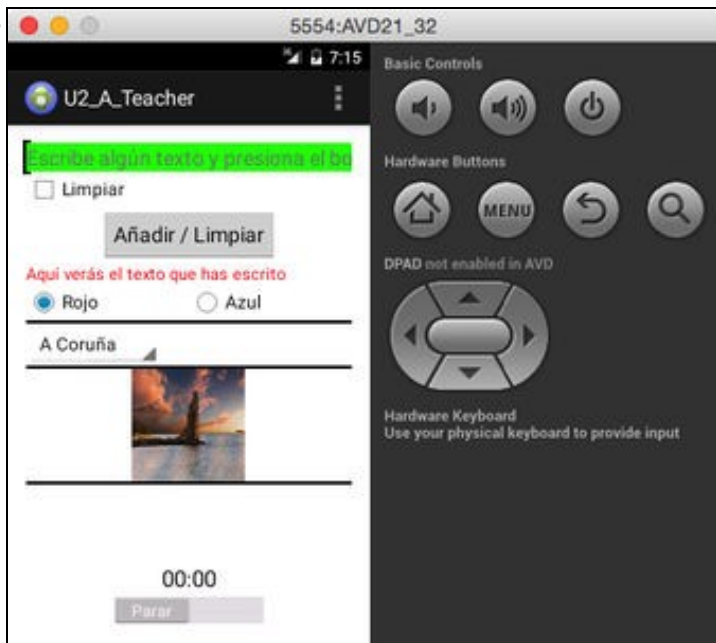


- Cando se traduzan os elementos de Inglés a Español e Galego, observar que nestes 2 idiomas hai "ñ" e tiles.
- As traducións deben estar en recursos XML non en código Java.

## Funcionamento

- A aplicación está dividida en 4 bloques.
- Aconséllase acometer un a un.
- En todo momento pódese probar a aplicación que se facilita para comprobar o funcionamento de cada bloque.

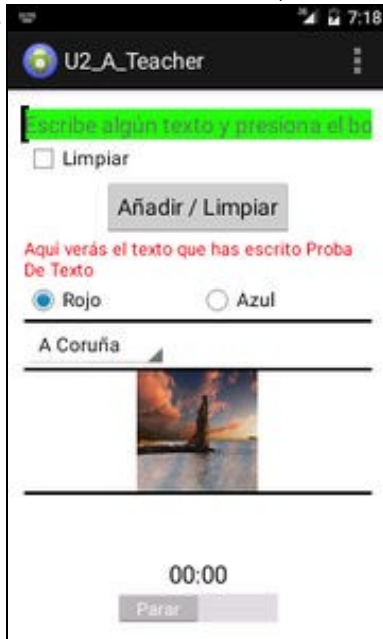
### Bloque 1: TextEdit, TextView, Botóns, CheckBox, RadioButtons



No **TextEdit** vaise introducir texto que se vai engadir á etiqueta de cor vermello (mirar que método permite engadir).  
 - A etiqueta de cor vermello ao lanzar a aplicación ten texto, pero unha vez que se limpa, xa non ten texto.



Cando Se Introduce O Texto, Este Terá A Primeira Letra De Cada Palabra En Maiúsculas De Xeito Automático.

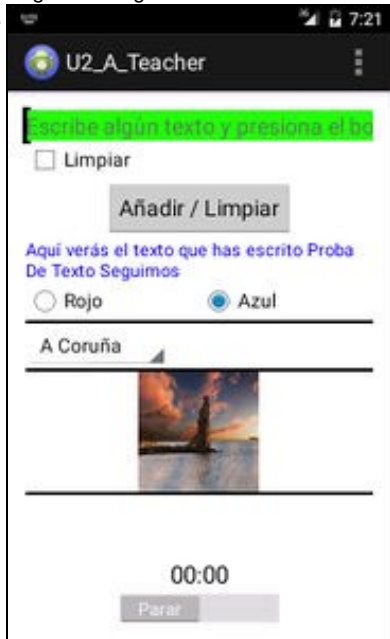


Se prememos no botón **Engadir/Limpiar**:

- Engádesse o texto do TextEdit ao que haxa na etiqueta Vermella (haxa texto ou nada, para comezar de novo).
- Tamén se baleira o TextEdit.



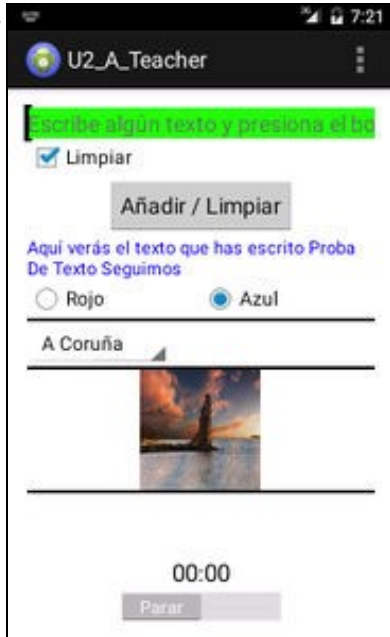
## Seguimos Engadindo Texto



O novo texto engadido á etiqueta, pero ademais:

Os **RadioButtons** permítenos cambiar a cor da etiqueta.

Para facer uso en Java do recurso XML que define a cor, podemos usar: `getResources().getColor(R.color.Nome_da_Cor)`



Se marcamos o **Checkbox** superior **Limpiar** e prememos o Botón **Engadir/Limpiar ...**



... este botón agora cambia o seu comportamento e limpa a etiqueta vermella.

Mentres non se desmarque o **Checkbox**, por moito texto que se introduza no **TextEdit** non se engade nada á etiqueta.

Se desmarca o **Checkbox** e se engade texto, ese texto será o texto polo que comece a etiqueta.

Observar que ao lanzar a aplicación a etiqueta xa tiña un texto que non se limpou até que se marcou o **Checkbox**.

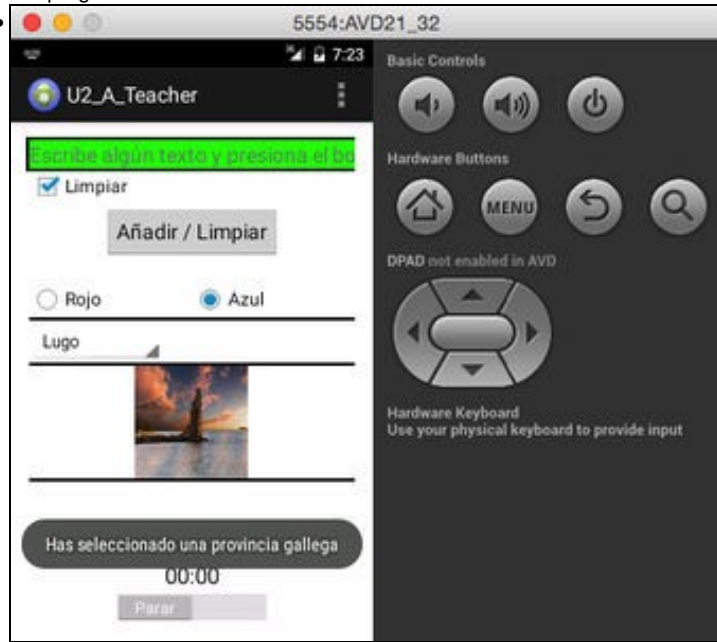
## Bloque 2: Spinner

- Este **Spinner** fai algo moi sinxelo, indica se a provincia seleccionada é galega ou non:
- Usarase un **Toast** para indicar a pertenza da provincia seleccionada
- Como vamos ter que usar un **Listener** e o seu contexto vai ser distinto ao da aplicación que lanza, temos que usar o método `getBaseContext()` para indicarlle ao **Toast** o seu contexto.
- Olo que as mensaxes o **Toast** debe adaptarse ao idioma escollido polo usuario.
- As provincias en Inglés non deben ter tiles.

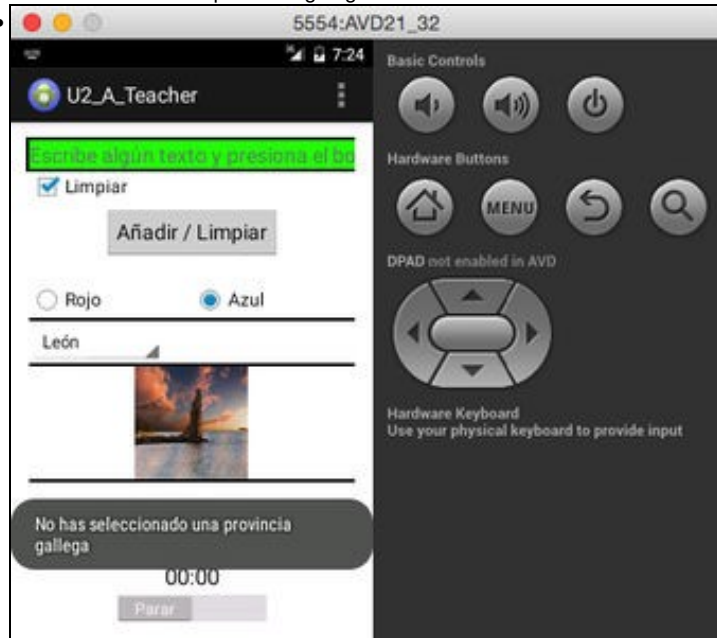




Desprezamos o Toast



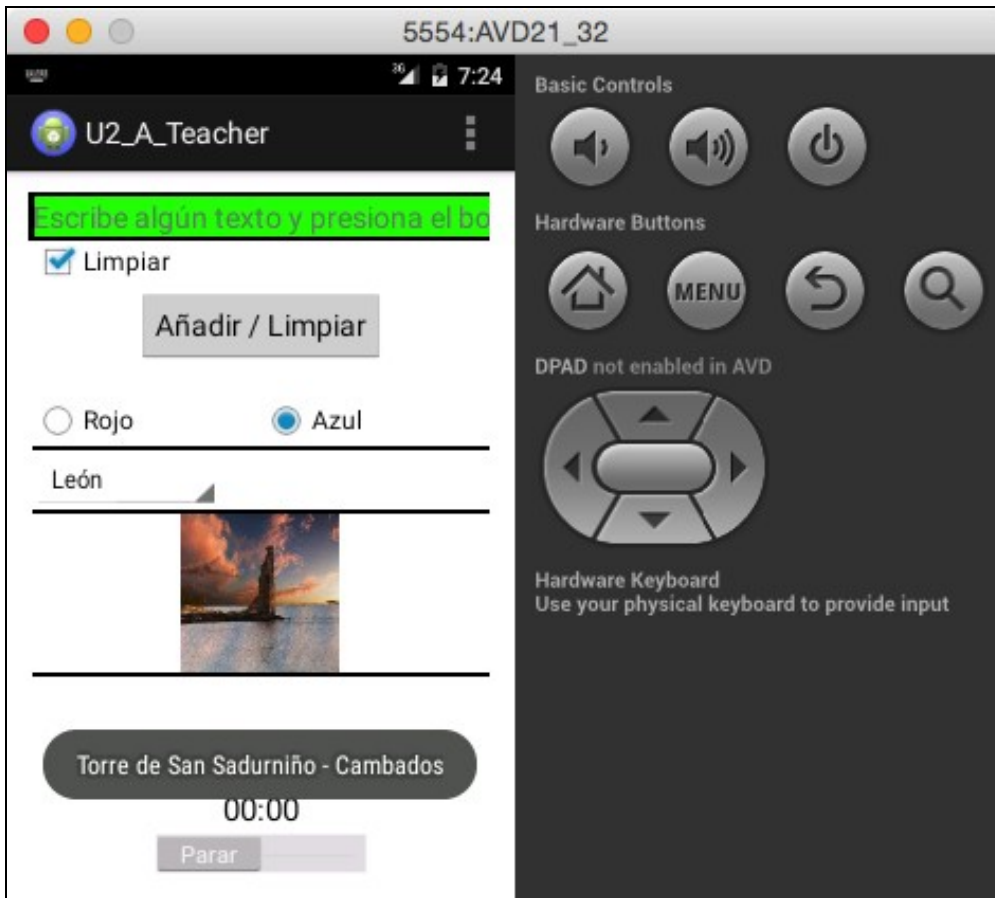
Seleccionamos unha provincia galega e vemos a mensaxe do Toast



Seleccionamos unha provincia non galega e vemos a mensaxe do Toast

### Bloque 3: ImageView

- Para definir a View no layout:
  - ◆ limitouse o alto e ancho da imaxe a 100 dp
  - ◆ e ademais usouse o atributo **scaleType**.
- Escolle unha imaxe característica da tua zoa de nacemento ou de residencia e úsaa para a aplicación.



- Ao premer sobre a imaxe aparece un Toast describindo a imaxe.
- O Toast neste caso cóllese o Tag da imaxe convértese a string.

#### Bloque 4: Chronómetro e Botón Switch



- Cando se activa o cronómetro (co botón Switch) este comeza a contar e cando chega a 15 segundos destrúe a Activity.
- Se se para o cronómetro este ponse a cero.
  - ♦ Olló!! que para iso antes de paralo hai que volver a establecer o tempo base para o cronómetro.
- non chega con usar o atributo onClick da View no Layout, pois este control tamén se desliza que é distinto a a facer click.
- Para poder capturar o "deslizamento", ó único que podemos controlar e o cambio de estado e para iso precisamos usar un Listener: **setOnCheckedChangeListener**.