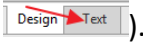

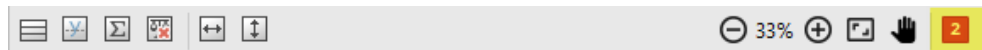




Ejercicio 1: Conociendo el editor de diseño UI




1. Crea un nuevo proyecto con nombre P_01_Vistas_1. Recuerda cambiar el nombre del paquete.
2. Selecciona aplicación solo para tfnos y tablets con Android como mínimo 4.0.3
3. Escoge una plantilla de **Empty Activity**
4. No cambies los nombres que te sugieren. No es necesario que actives "Backwards Compatibility" (que sería necesario para poder trabajar con la "ActionBar")
5. Cambia al fichero *activity_main.xml* (si quieres puedes cerrar el fichero MainActivity.java, de momento no vamos a trabajar con el código java).
6. Selecciona la primera vista que estaba ya creada (*TextView*) y elimínala. Puedes hacerlo desde el árbol de componentes **Component Tree** o desde el editor de diseño (ya sea en su pestaña Diseño o en su pestaña Text ).
7. Cambia el diseño del *layout* para que sea *LinearLayout* desde la pestaña Text.
8. De vuelta a la pestaña Diseño y seleccionado desde el árbol de componentes el contenedor (el "LinearLayout") fija en la ventana de propiedades que sea de orientación vertical.
9. Desde la paleta de izquierda arrastra los siguientes elementos: *ToggleButton*, *CheckBox*, *ProgressBar* y *RatingBar*.
10. Observa que la herramienta Lint nos avisa de errores/avisos , clics sobre el aviso, observa a que se refieren y las posibles correcciones sugeridas, cierra el aviso con Esc, ya los arreglaremos más adelante!



11. Muévete por las diferentes vistas de nuestro layout y observa que la hilera de botones varía dependiendo del tipo de vista seleccionada. Dichos botones son con los que vamos a practicar y sirven para ajustar propiedades rápidamente:

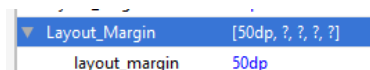


12. Con el contenedor seleccionado, pulsa el primer botón para conseguir que el *LinearLayout* donde están las diferentes vistas tenga una orientación horizontal. Pulsa otra vez el botón para volver a una orientación vertical.
13. Selecciona la vista *ToggleButton*. Pulsa el botón  (*Toggle Fill Width*). Conseguirás que el ancho del botón se ajuste al ancho de su contenedor o de su contenido. Observa los cambios de su valor en el panel de propiedades. Déjalo para que adapte a su contenido.
14. Pulsa el botón siguiente  (*Toggle Fill Height*). Conseguirás que el alto del botón se ajuste al alto de su contenedor. El problema es que el resto de elementos dejan de verse. Vuelve a pulsar este botón para regresar a la configuración anterior (También puedes pulsar *Ctrl-Z*).

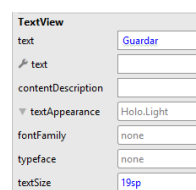
15. Observa que hay un espacio sin usar en la parte inferior del *Layout*. Pulsa el botón  (*Assign All Weights*). El alto de la vista se ajustará para que ocupe lo necesario para que no haya espacio sin usar. Repite asignar la propiedad al resto de las vistas, lo que hace es dividir de forma proporcional el espacio sin usar entre todas las vistas. Es equivalente a poner la propiedad *Layout Weights* =1 y la propiedad *Layout Height* =0dp para todas las vistas de este *Layout*. Con el botón  puede restablecerse el peso de la vista a su condición inicial.
16. Con la vista *CheckBox* seleccionada, visualiza todas sus propiedades , e introduce el valor 2 en la propiedad *Layout Weights*. Selecciona la vista *ToggleButton* y cambia el valor a 0.5. Selecciona la vista *ProgressBar* y cambia el valor 4. Como puedes observar estos pesos permiten ajustar la altura de las diferentes vistas.
17. Utiliza los botones de la barra superior para observar cómo se representará el *Layout* en diferentes circunstancias:



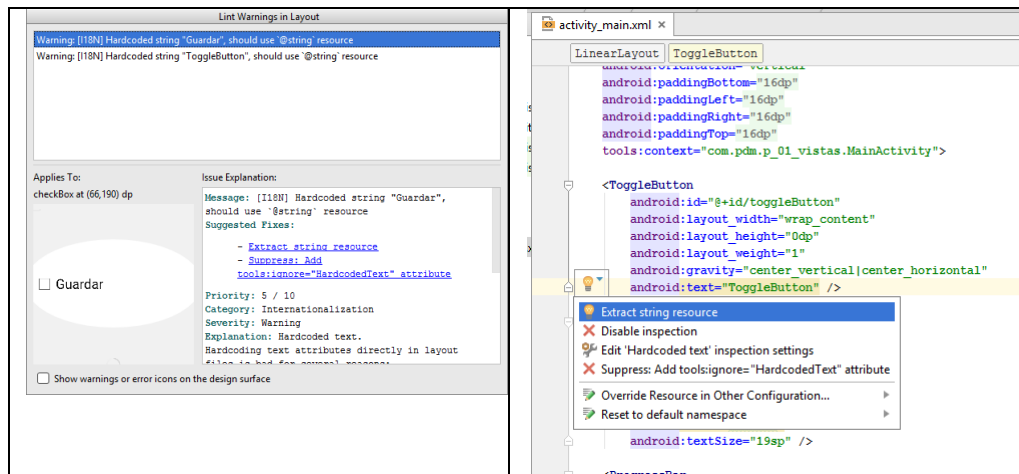
- El primer desplegable permite escoger previsualizaciones entre diferentes modos de pantalla: en orientación vertical (*Portrait*) o horizontal (*Landscape*). También en modos especiales (*Card Dock* significa que está activado el modo automóvil y *Night time* el modo noche,...) pero no siempre!.
 - El segundo desplegable permite tener una previsualización (no crea layout alternativos) en ejemplos de dispositivos.
 - El siguiente desplegable permite ver cómo quedaría el *Layout* tras aplicar un tema (lo veremos en la siguiente unidad).
 - El siguiente permite dotar de recursos alternativos para soporte de idiomas a la aplicación.
 - El siguiente sirve para cambiar la previsualización a las distintas versiones que hayamos instalado desde el SDK (puede tardar un poco en cargarse y en algunas fallar por incompatibilidad!).
 - El último desplegable permite crear layout alternativos para diversos soportes (orientación, tamaño, etc.)
18. Selecciona la vista *CheckBox* y fija el margen a “50dp”.



19. Busca la propiedad *Text* y sustituye el valor “*CheckBox*” por “*Guardar*” y *Text size* a “19sp”.

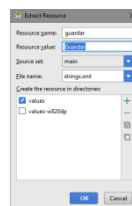


20. Pulsa sobre la pestaña Text y pulsa las teclas *Ctrl-Alt-L* para que formatee adecuadamente el código XML.
21. Hagamos caso a los avisos de Lint: Puede hacerse desde la pestaña Diseño o desde Texto.




En el ejemplo, advierte que te aconseja que es aconsejable extraer los textos "Guardar" y "ToggleButton" al fichero string.xml para que dar soporte para idiomas sea posible. Realmente solo es necesario Guardar, porque el texto del ToggleButton que se visualizará es "ON/OFF".

Ejecuta la solución sugerida (picando en "Extract string resource"):



22. Pica en  y da soporte para español. Por ejemplo

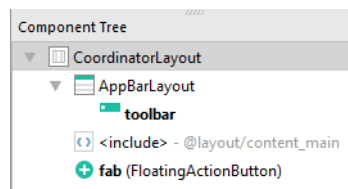
+  Show only keys needing translations			
Key	Default Value	Untra...	
app_name	P_01_Vistas	<input type="checkbox"/>	P_01_Vistas
guardar	Save	<input type="checkbox"/>	Guardar
togglebutton	ToggleButton	<input type="checkbox"/>	

23. Ejecuta el proyecto para ver el resultado (recuerda que la primera vez que lances un tipo de emulador deberás tener paciencia!). Además a veces la pantalla del emulador aparece blanca! Para solucionarlo, minimiza y maximiza el emulador y "voila" (???)

Ejercicio 2: Vistas de entrada de texto

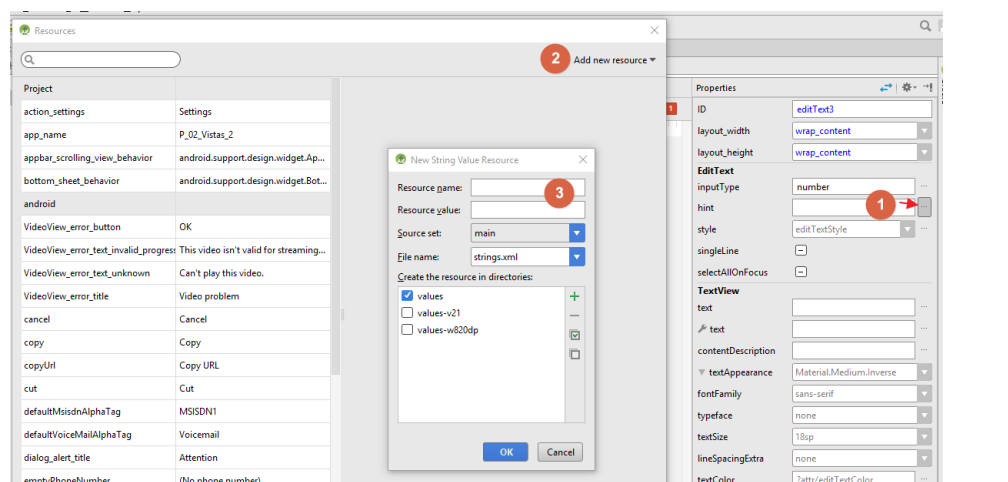
1. Crea un nuevo proyecto con nombre `P_02_Vistas_2`.
2. Escoge una plantilla básica (**Basic Activity**) para que la aplicación tenga un diseño inicial basado en Material Design (una guía para el diseño visual de las aplicaciones, que Google no quiere aplicar exclusivamente a dispositivos móviles, sino que pretende utilizar en todo tipo de contenidos digitales). Observa que ahora tenemos dos ficheros en `res/layout`: `activity_main.xml` y `content_main.xml`.

El fichero `activity_main.xml` contiene una “pantalla customizada” (`CoordinatorLayout`) con tres elementos (la barra de herramientas, un botón de acción y su vista principal que es `content_main.xml`). Desde el modo texto, observa como las etiquetas de las vistas son muy largas. Esto es debido a que no están definidas en el API de Android si no en una librería (en concreto a las `design`). Lo estudiaremos en la siguiente unidad.

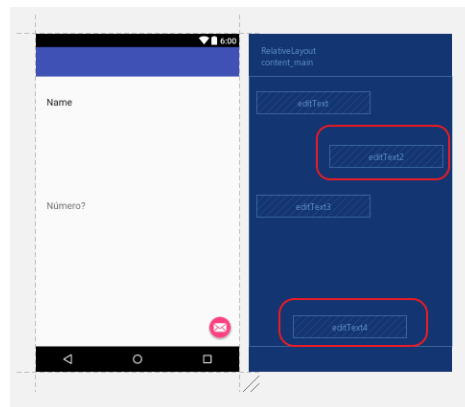


De momento, trabajaremos con el fichero `content_main.xml`

3. Edita en modo Design el layout creado para la actividad y borra el `TextView` de ejemplo.
4. Añade una vista de tipo entrada de texto de tipo normal (Text Plain). Debajo una de tipo palabra secreta (Password). Continua así con otros tipos de entradas de texto (sin perder demasiado tiempo, que nos conocemos!). Algunas en el editor casi no se ven!.
5. Siempre es conveniente dar la mayor ayuda al usuario, acostúmbrate a utilizar la propiedad "hint" en los `EditText`. **NO ES** el valor por defecto, es el valor que se visualizará cuando está vacío.



6. Como el contenedor es de tipo `RelativeLayout` podremos desplazar las vistas que hay y posicionarlas relativamente a otras.

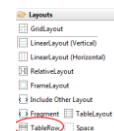




7. Ejecuta el proyecto para ver el resultado y observa como para introducir el texto de una entrada se mostrará un tipo de teclado diferente. Observa también los tres puntos de la esquina superior derecha, al pulsar se mostrará un menú con la opción `Settings`. Como se ha definido este menú y otras opciones sobre la barra de acciones se explicarán más adelante.

(Consejo: Si no has ido cerrando Proyectos, Android Studio los está inspeccionando continuamente y ralentizando el trabajo. Acostúmbrate a ir cerrando los proyectos cuando los finalices!).

Ejercicio 3: Uso de `TableLayout`

1. Crea un nuevo proyecto con nombre `P_03_Vistas_3` (recuerda que la elección de la plantilla determinará que tengas que trabajar con el fichero `activity_main.xml` en el caso de *Empty* o con el `content_main.xml` en el caso de *Basic*).
2. Utiliza un `TableLayout` para realizar un diseño similar al siguiente.

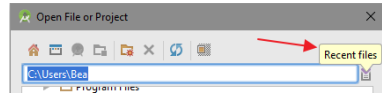


Ayuda: una vez definido el tipo de layout, para añadir las filas utiliza `TableRow` y usa los botones  y  para trabajar más rápidamente.

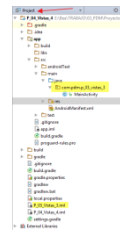
Observa que la herramienta Lint nos da 2 avisos para cada botón, el de los textos guardarlos como recursos "string" (los ignoramos porque son números y no hace falta soporte para idiomas) y el de estilos (lo desarrollaremos en la siguiente unidad).

Ejercicio 4: Copiando proyectos

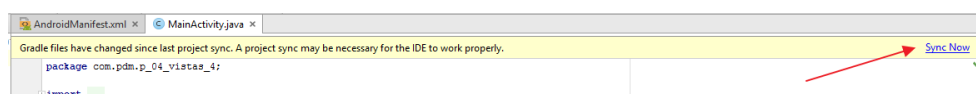
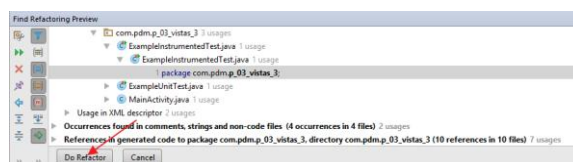
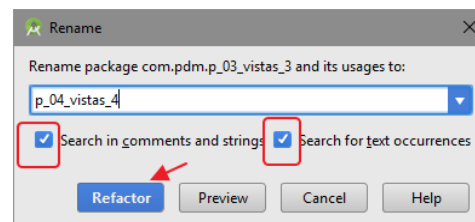
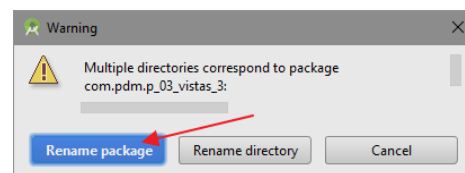
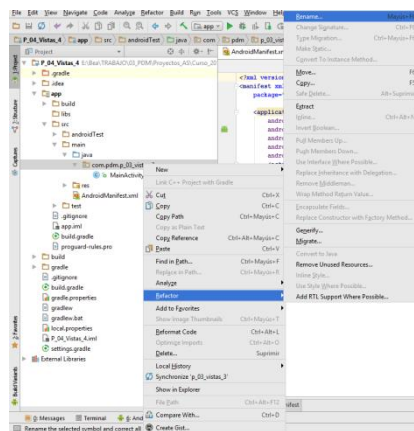
1. Un hecho muy común cuando desarrollamos, es querer trabajar con una copia de un proyecto. Por ejemplo, queremos conservar P_03_Vistas_3 y trabajar sobre una copia llamada P_04_Vistas_4 (hazlo!).
2. Abre desde AS P_04_Vistas_4. Consejo: La manera más rápida de encontrarlo es usando la herramienta "Recent files":



3. Observa que hay muchas referencias a P_03_Vistas_3:



4. Puedes borrar P_03_Vistas_3.iml (es un archivo de módulo creado por AS que guarda los caminos del módulo, dependencias y otros ajustes) ya tenemos P_04_Vistas_4.iml
5. La manera más rápida de refactorizar es ejecutar la siguiente secuencia:





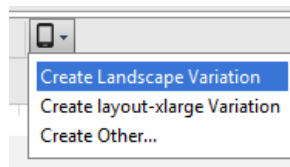
6. Seguramente tendrás que hacer alguna modificación más (por ejemplo, en strings.xml y sus variantes)

Ejercicio 5: Recursos alternativos (orientación)

1. Crea un nuevo proyecto llamado P_05_Creciente_01 (va a crecer a lo largo del curso!) con plantilla *Basic Activity*.
2. Trata de crear una vista similar a la que ves a continuación. Tiene que estar formada por un *LinearLayout* que contiene un *TextView* y cuatro *Button*.



3. Trata de utilizar recursos (*res/values/strings.xml*) para introducir los cinco textos que aparecen y cambiar el título que aparece como nombre de la app (en el fichero *AndroidManifest.xml* es el atributo `android:label="@string/app_name"`):
4. Ejecuta la aplicación en el emulador.
5. Gira la orientación del emulador . Si observas el resultado de la vista que acabas de diseñar en vertical quizás no queda todo lo bien que deseáramos en apaisado. (También lo podríamos haber comprobado pulsando el icono  en AS o previsualizando en los ejemplos representativos)
 - a. Para resolver este problema utiliza el asistente para crear la variante de orientación horizontal




- b. Para dicha configuración horizontal deseamos un contenedor de tipo un *LinearLayout* que contiene un *TextView* y un *TableLayout* con dos *Button* por columna. Desde/hacia el árbol de componentes puedes mover/arrastrar vistas.

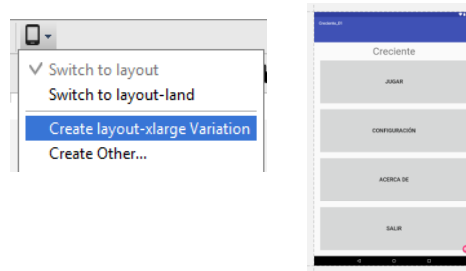


(Para conseguir en un *TableLayout*, que las columnas se ajusten a todo el ancho de la tabla poner `stretchColumns="*"`. `stretchColumns="0"` significa que asigne el ancho sobrante a la primera columna. `stretchColumns="1"` significa que asigne el ancho sobrante a la segunda columna. `stretchColumns="*"` significa que se asigne el ancho sobrante entre todas las columnas).

- c. Vuelve a probar en el emulador.

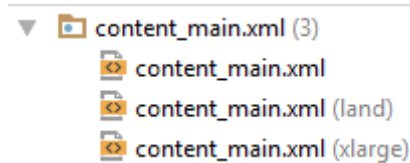
Ejercicio 6: Recursos alternativos (Pantallas grandes)

1. Abre un emulador tipo tablet (10 pulgadas). Si intentas volver a ejecutar la aplicación anterior, se abre por defecto en el dispositivo (real o emulado) en el que se probó por última vez. Para cambiar ese comportamiento "Run Instant", debes parar con  y así en la siguiente ejecución nos solicita escoger entre todos los dispositivos conectados.
2. Si ejecutas la aplicación en una *tablet* observarás que el tamaño de la fuente pequeño, queda mucho espacio vacío, etc.
3. Trata de hacer un diseño alternativo utilizando el asistente.



4. Verifica que la aplicación se visualiza correctamente en todos los tipos de pantalla, tanto en horizontal como en vertical.

Resumiendo, para un buen diseño deberíamos haber configurado los ficheros `content_main.xml` (SIEMPRE EL MISMO NOMBRE!) en carpetas `res/layout` (vista por defecto), `res/layout-land` (vista apaisada), `res/layout-xlarge` (vista en pantallas grandes) y todas las posibles configuraciones. Por otro lado recuerda que siempre es más fácil trabajar con `RelativeLayout`.



Ejercicio 7: Recursos alternativos (Internacionalización)

5. Queremos traducir nuestra aplicación al español, inglés y francés. Utiliza para ello el Editor de traducciones al que se puede acceder desde el fichero `strings.xml`.
6. Ejecuta la aplicación y verifica que la aplicación se visualiza en el idioma en que está configurado el emulador. Cambia el idioma del emulador y comprueba que también cambia en la aplicación.