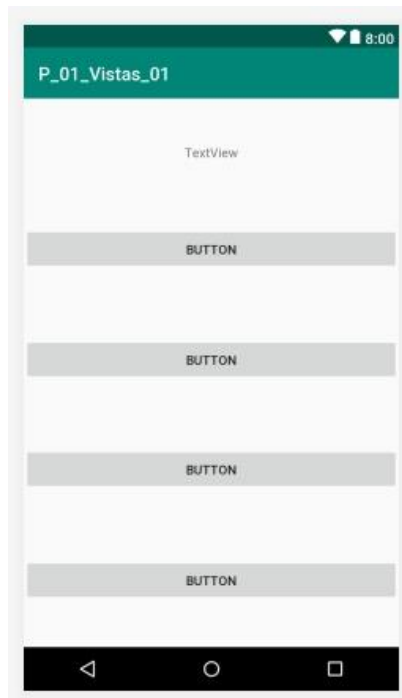
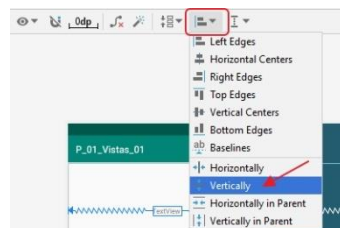


Ejercicio 1: Usando ConstraintLayout

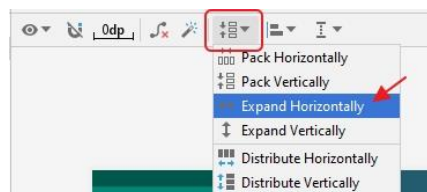
1. Crea un nuevo proyecto llamado P_01_Vistas_01 con plantilla de actividad inicial *Empty Activity*.
2. Diseña una pantalla formada por un `ConstraintLayout` que contiene un `TextView` y cuatro `Button`. Las cinco vistas están distribuidas uniformemente en el layout. Las previsualizaciones son las siguientes:



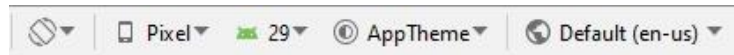
3. Solución guiada:
 - a. Arrastra al layout las cinco vistas desde la paleta. Pon en todas restricciones de `Odp` a izquierda y derecha.
 - b. Para distribuirlas uniformemente, selecciona las 5 a la vez (`May+click`) y distribuye:



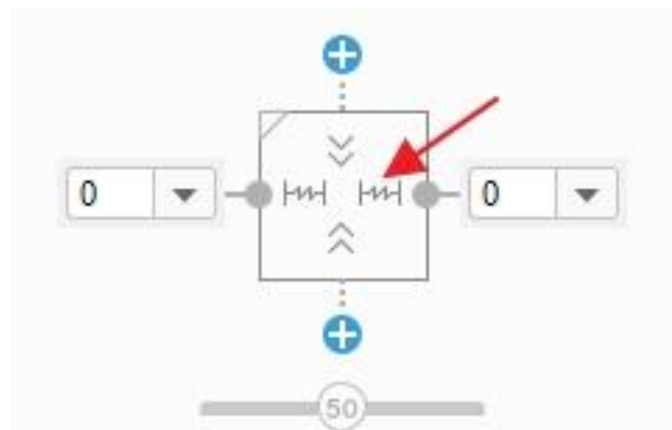
- c. Expande en horizontal los cuatro `Button`:



- d. Previsualiza en horizontal. Utiliza los botones de la barra superior para observar cómo se representará el Layout en diferentes circunstancias:




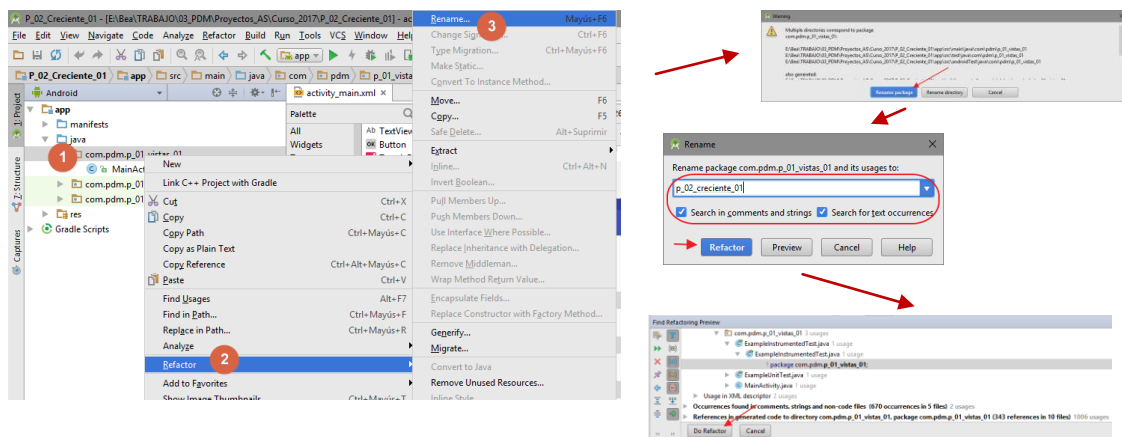
- Las cuatro primeras opciones del primer desplegable permiten escoger **previsualizaciones** (no crean layouts alternativos) entre diferentes modos de pantalla: en orientación vertical (*Portrait*) o horizontal (*Landscape*). También en modos especiales (*Card Dock* significa que está activado el modo automóvil y *Nigth mode* el modo noche,...) pero no siempre!. Las últimas permiten **crear layout alternativos** para diversos soportes (orientación, tamaño, etc.)
 - El segundo desplegable permite tener una previsualización (no crea layouts alternativos) en ejemplos de dispositivos.
 - El siguiente sirve para cambiar la previsualización a las distintas versiones que hayamos instalado desde el SDK (puede tardar un poco en cargarse y fallar por incompatibilidad!).
 - 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.
- e. Para conseguir que los botones ocupen el ancho de la pantalla también en horizontal:



- f. La herramienta Lint nos informa que tenemos errores (son "warnings") que corregiremos en otro ejercicio.
- g. Ejecuta en dispositivo (real o emulado) y cierra el proyecto.

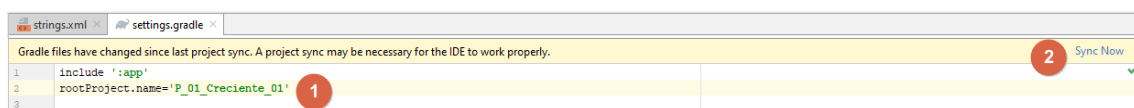
Ejercicio 2: Copiando proyectos

1. Un hecho muy común cuando desarrollamos, es querer trabajar con una copia de un proyecto. Por ejemplo, queremos conservar P_01_Vistas_01 y trabajar sobre una copia llamada P_02_Creciente_01 (hazlo desde el explorador de ficheros!).
2. Abre desde AS () Open an existing Android Studio project) P_02_Creciente_01. Consejo: La manera más rápida de encontrarlo es usando la herramienta "Recent files".
3. Observa que hay muchas referencias a P_01_Vistas_01:
4. La manera más rápida de refactorizar es ejecutar la siguiente secuencia:

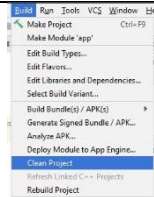
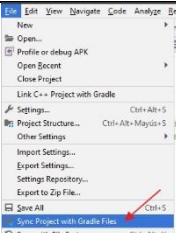
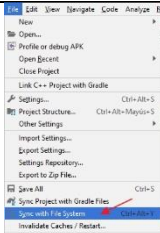


Seguramente tendrás que hacer alguna modificación más (por ejemplo, en strings.xml y sus variantes, querrás cambiar el nombre de la aplicación).

Además:



Ejecuta en dispositivo, seguramente fallará la instalación porque todavía hay referencias al paquete "antiguo". Solución:

1º Limpiar proyecto	2º Sincronizar con ficheros gradle	3º Sincronizar con ficheros sistema
		

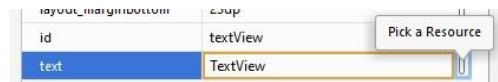
Ejercicio 3: Recursos strings y soporte para idiomas


1. Sigue trabajando en el proyecto P_02_Creciente_01 (va a ser un proyecto que a crecer durante el curso) y utiliza recursos (res/values/strings.xml) para introducir los cinco textos que aparecen:

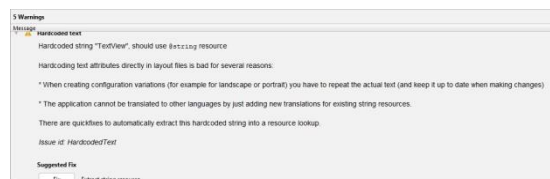



2. Solución rápida: para cada una de las cinco vistas, fija los valores de la propiedad **text**: Hay dos alternativas

- a. haciendo uso del asistente para la creación de recursos:



- b. desde la propia herramienta Lint, seleccionado el "warning" :



3. Observa que han desaparecido los "warning".
4. Ejecuta la aplicación en un dispositivo
5. Pica en  Default (en-us) y da soporte para español. Por ejemplo

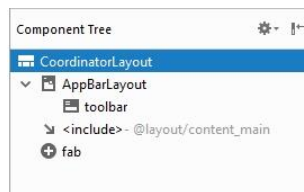
Key	Resource Folder	Untranslatable	Default Value	Spanish (es)
app_name	app/src/main/res	<input checked="" type="checkbox"/>	P_02_Creciente_01	
texto1	app/src/main/res	<input checked="" type="checkbox"/>	Creciente	
texto2	app/src/main/res	<input type="checkbox"/>	Play	Jugar
texto3	app/src/main/res	<input type="checkbox"/>	Settings	Configuración
texto4	app/src/main/res	<input type="checkbox"/>	About	Acerca de
texto5	app/src/main/res	<input type="checkbox"/>	Exit	Finalizar

6. Ejecuta la aplicación en el mismo dispositivo que antes y cambia el idioma de inglés a español y viceversa para comprobar el correcto funcionamiento.

Ejercicio 4: Vistas de entrada de texto

1. Crea un nuevo proyecto con nombre P_03_Vistas_02.
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; lo veremos en una unidad posterior). 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, están en una librería (en concreto las de "design"). Lo estudiaremos en la siguiente unidad.



De momento, trabajaremos con el fichero content_main.xml para conseguir el siguiente diseño de pantalla:



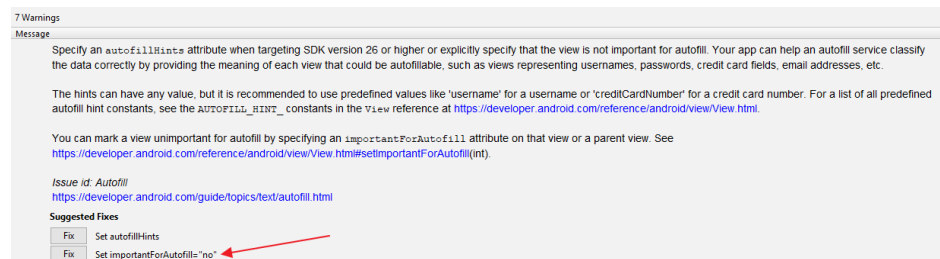
3. Solución guiada:


- c. Borra el TextView de la plantilla suministrada.
- d. Arrastra las vistas de tipo entrada de texto (la primera de tipo normal o Text Plain, debajo una de tipo "Password", otra "Number" debajo y la última "Multiline text". Algunas en el editor casi no se ven!). Puede conocerse su tipo por la propiedad **inputType**



son "engañosas", nos ayudarán con el teclado a mostrar al usuario pero no si los valores son válidos o no!.

- e. 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 (tipo "Por favor, teclee edad"). Recuerda crear los recursos strings como en un ejercicio anterior.
- f. Arregla los warnings, usando el asistente y fija los atributos de autocompletado a No ([Más información](#)):



- g. Permite que AS deduzca las restricciones con 
- h. Ejecuta para ver el resultado y observa como para introducir el texto de una entrada se mostrará un tipo de teclado diferente.
- i. Observa también los tres puntos de la esquina superior derecha, al pulsar se mostrará un menú con la opción Settings. La manera en que se ha definido este menú y otras opciones sobre la barra de acciones se explicarán más adelante.

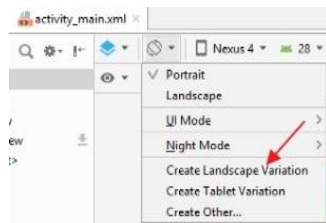
(Consejo: Si no has ido cerrando Proyectos, Android Studio los está inspeccionando continuamente y ralentizando el trabajo. Acostúmbrate a ir cerrando los proyectos cuando los finalices!).

Ejercicio 5: Soporte para orientación

1. Vuelve al proyecto P_02_Creciente_01. Se desea que en orientación horizontal el diseño de pantalla sea:



2. Solución: Crea la variación desde



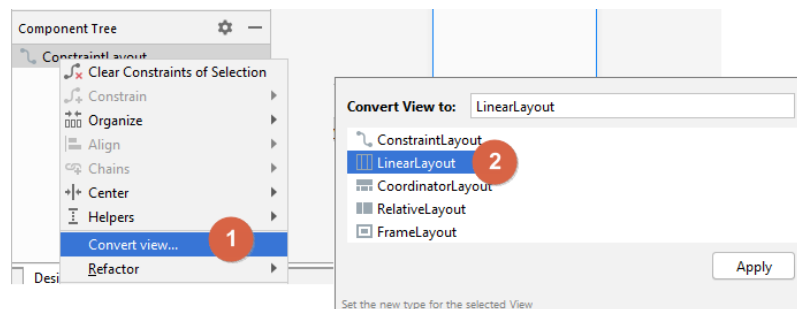
(ya de paso, observa que podríamos crear otro tipo de variaciones!). Se trata de una copia del diseño en vertical. Trabaja sobre ella, borrando las restricciones anteriores y creando las nuevas.








Ejercicios optativos


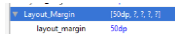
Los siguientes ejercicios son optativos, pero recomendables para conocer otros tipos de Layout que antes de ConstraintLayout eran muy utilizados.

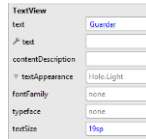
Ejercicio 6: Conociendo LinearLayout

1. Crea un nuevo proyecto con nombre P_04_Vistas_03 para tfnos y tablets con Android como mínimo 4.4 y escoge una plantilla **Empty Activity**. No es necesario que actives "Backwards Compatibility" (que sería necesario para poder trabajar con la "ActionBar")
2. En el fichero *activity_main.xml* selecciona la primera vista que estaba ya creada (*TextView*) y elimínala.
3. Cambia el diseño del *layout* para que sea *LinearLayout*:




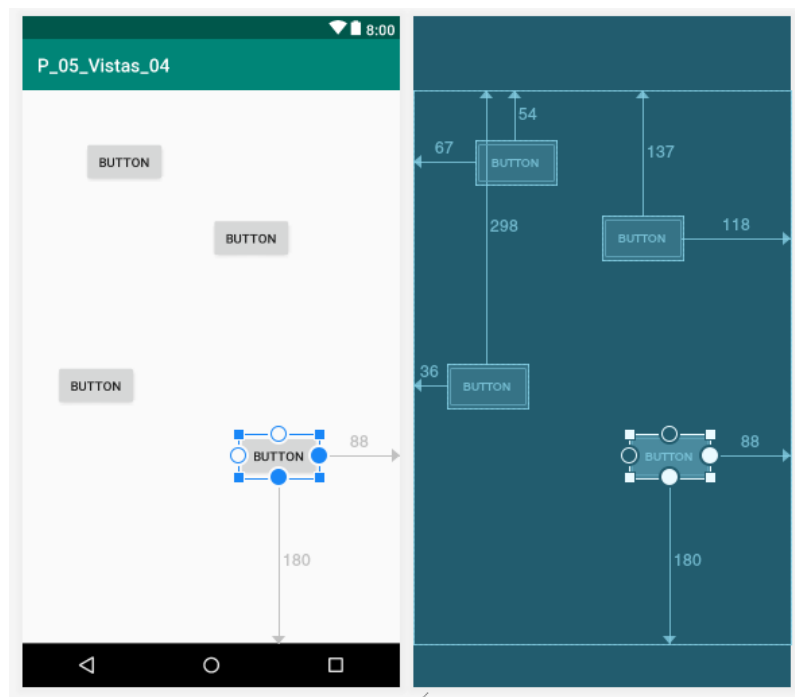
4. Seleccionado desde el árbol de componentes el contenedor (el "LinearLayout") fija en la ventana de propiedades que sea de orientación vertical. 
5. Desde la paleta de izquierda arrastra los siguientes elementos: *ToggleButton*, *CheckBox*, *ProgressBar* y *RatingBar*.
6. Muévete por las diferentes 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.
7. Con el contenedor (layout) seleccionado, pulsa el primer botón  para conseguir que el *LinearLayout* tenga una orientación horizontal. Pulsa otra vez el botón  para volver a una orientación vertical.
8. Selecciona la vista *ToggleButton*. Pulsa el botón . Conseguirás habilitar/deshabilitar que el ancho del botón se ajuste al ancho de su contenedor o a su contenido. Observa los cambios del valor de *layout_width* en el panel de propiedades (*match_parent* o *wrap_content*). Déjalo para que adapte a su contenido. **AS 3.5 no funciona, hay que hacerlo con los atributos.**
9. Pulsa el botón siguiente . Conseguirás habilitar/deshabilitar que el alto del botón se ajuste al alto de su contenedor o de su contenido. 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*). **AS 3.5 no funciona, hay que hacerlo con los atributos.**
10. Observa que hay un espacio sin usar en la parte inferior del *Layout*. Selecciona el layout y sus cuatro vistas. Pulsa el botón . El alto de las vistas se ajustará para que ocupen lo necesario para que no haya espacio sin usar. Es equivalente a poner la propiedad *layout_weight* =1 y la propiedad *layout_height* =0dp para todas las vistas. Con el botón  puede restablecerse el peso de la vista a su condición inicial.

11. Con la vista `CheckBox` seleccionada, introduce el valor 2 en la propiedad `layout_weight`. 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.
12. Selecciona la vista `CheckBox` y fija el margen a "50dp". (para ver la propiedad puedes usar  para buscarla o para ver todas. )
13. En la propiedad `Text` sustituye el valor "CheckBox" por "Guardar" y Text size a "19sp".
14. Hagamos caso a los avisos de Lint y extrae los recursos strings
15. Ejecuta el proyecto para ver el resultado



Ejercicio 7: Conociendo RelativeLayout

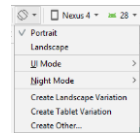
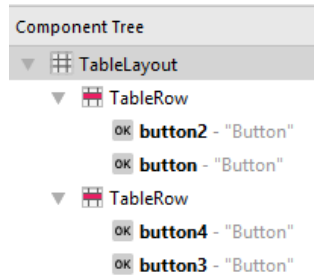
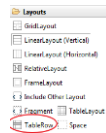
1. Crea un nuevo proyecto con nombre `P_05_Vistas_04` (recuerda que la elección de la plantilla determinará que tengas que trabajar con el fichero `activity_main.xml` en el caso de `Empty` o con el `content_main.xml` en el caso de `Basic`).
2. Borra el `TextView` suministrado por la plantilla y cambia el tipo de layout a `RelativeLayout` (muy utilizado hasta noviembre 2016).
3. Desde AS 3.5, hay que habilitar  para que las vistas se coloquen relativas al contenedor padre. Arrastra al contenedor cuatro botones, de manera similar a como muestra la imagen.



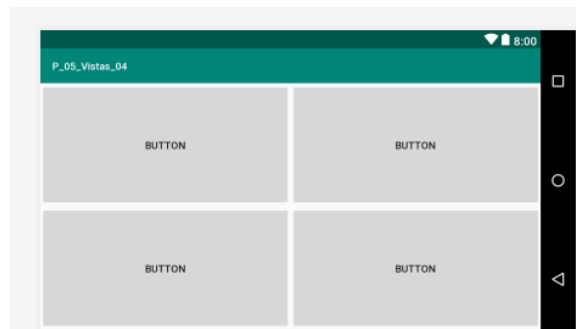
4. Ejecuta el proyecto en un dispositivo.

Ejercicio 8: Conociendo TableLayout

- Trabajando con el mismo proyecto que en el ejercicio anterior, crea la variación para orientación horizontal
- Cambia en la variación el tipo de layout a `TableLayout` para realizar un diseño similar al siguiente. Añade las filas arrastrando desde la paleta 2 `TableRow` y mueve a ellas los botones



- Para cada uno de los botones, borra los posibles márgenes (atributos `layoutMarginStart` y `layoutMarginTop`) y usa botones explicados anteriormente para conseguir un diseño similar al siguiente:



- La herramienta Lint nos advierte que de avisos referidos a estilos o a recursos string no son necesarios que los corrijas, ya trabajaremos sobre dichos conceptos más adelante