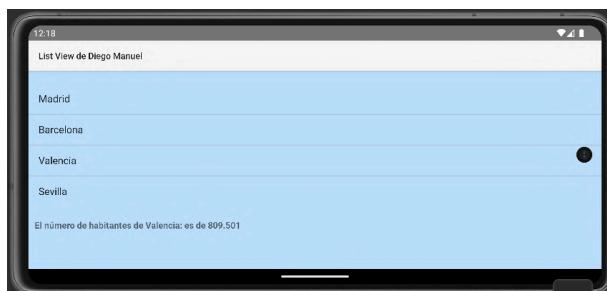
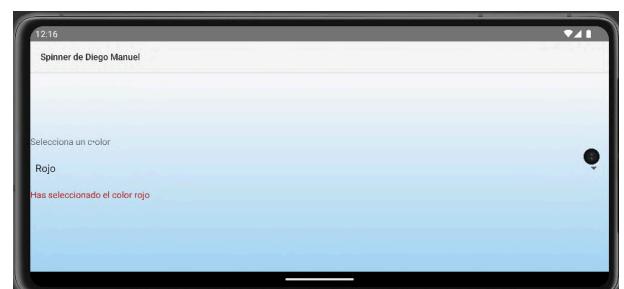
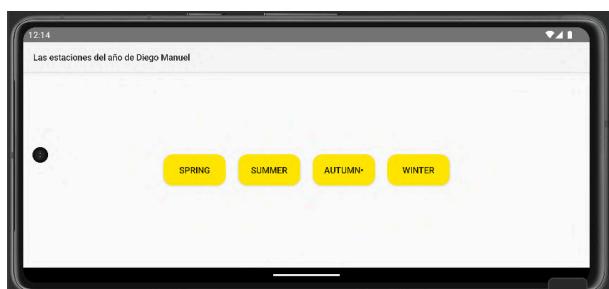
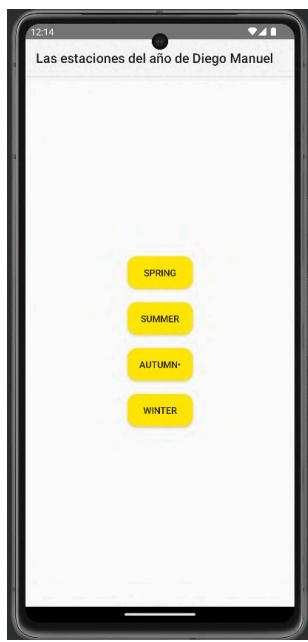


# Tarea 4 para Programación Multimedia y Dispositivos Móviles



Diego Manuel Carrasco Castañares

**PMDM04.- Diseño. Gestores de colocación.**

Internacionalización y localización.

Detalles de la tarea de esta unidad.

Enunciado.

**Tarea UT04.1. Orientación y Localización**

Crea una app que muestre los nombres de las estaciones del año con las siguientes especificaciones:

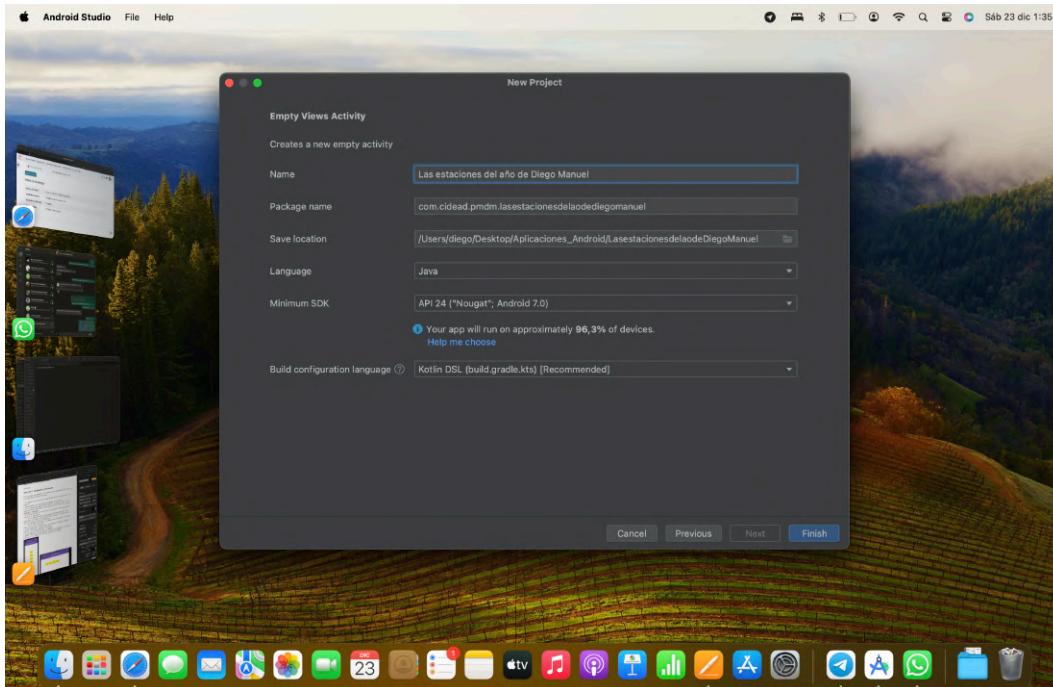
- El nombre de la aplicación será "Las estaciones del año de + Tu nombre"
- Cada estación del año aparecerá dentro de un botón.
- Los botones deben tener fondo amarillo y texto en negro. Sus esquinas serán redondeadas. Usar para ello un fichero de recurso dibujable (drawable).
- El diseño variará en función de la orientación del dispositivo (vertical o apaisada).
  - En el diseño vertical (portrait) las estaciones del año aparecerán centrados horizontal y verticalmente en la pantalla, como una cadena vertical de botones.
  - En el diseño apaisado, la cadena será horizontal (el resto igual que el caso vertical)
- La app debe estar localizada a los idiomas: español, inglés y francés (Todo debe aparecer traducido).
- El idioma por defecto será el español.
- EXTRA: Añade un margen entre los botones de 20dp tal y como se muestra en la imagen. La interfaz tendrá este aspecto



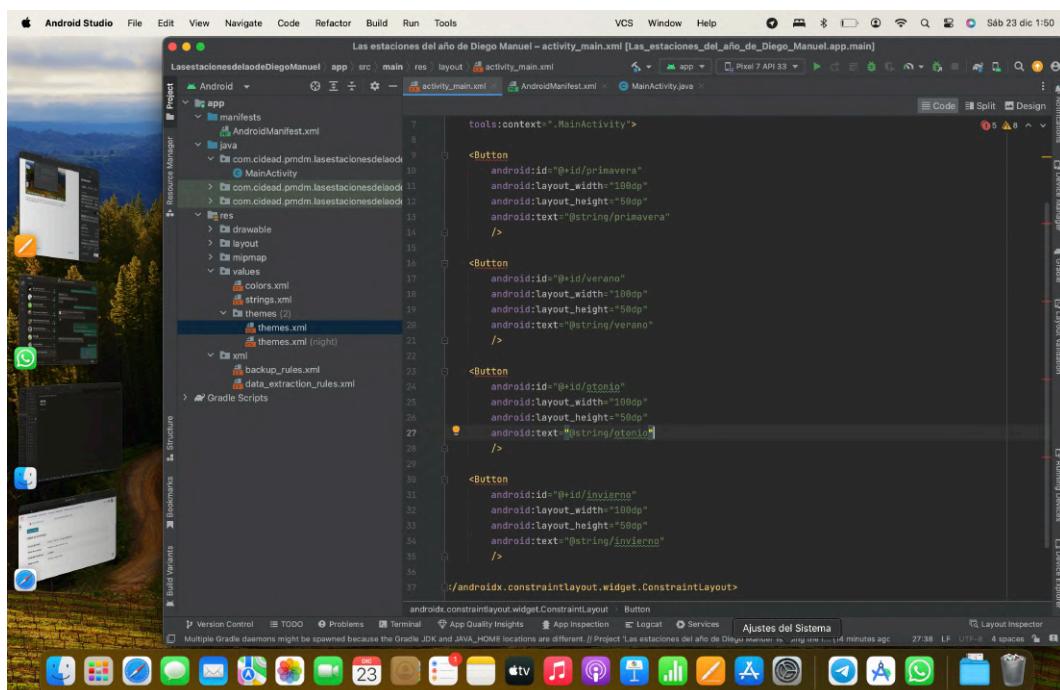
Lo primero que haremos será modificar el tema de la app para que nos muestre el nombre de la misma en la parte superior de la pantalla. Esto lo haremos en manifestos – AndroidManifests.xml.

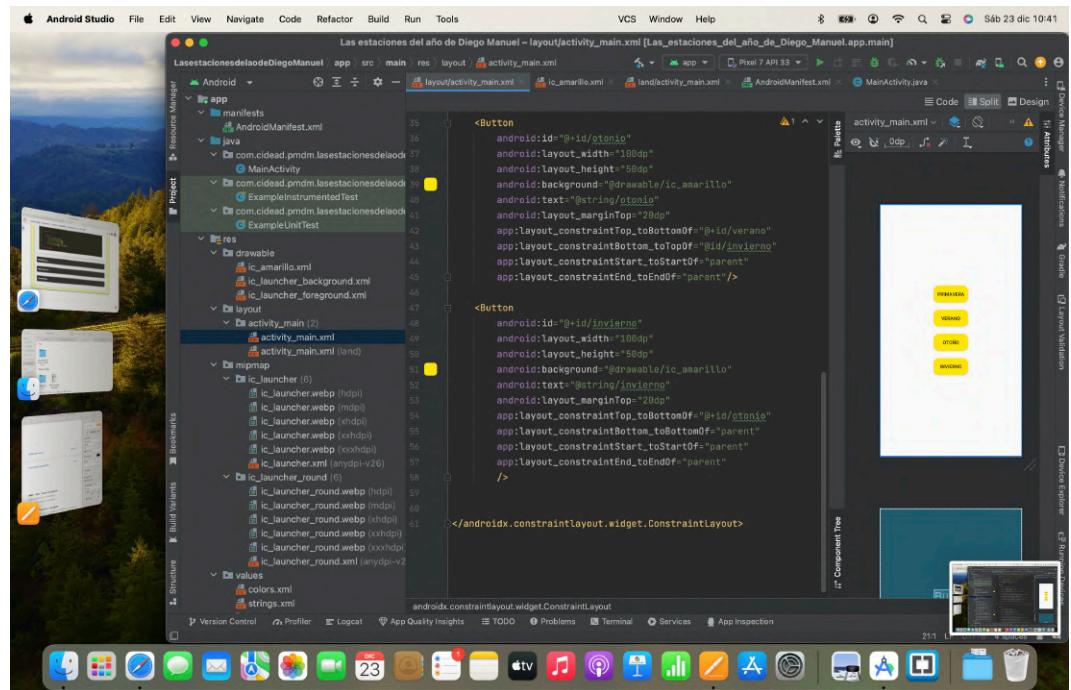
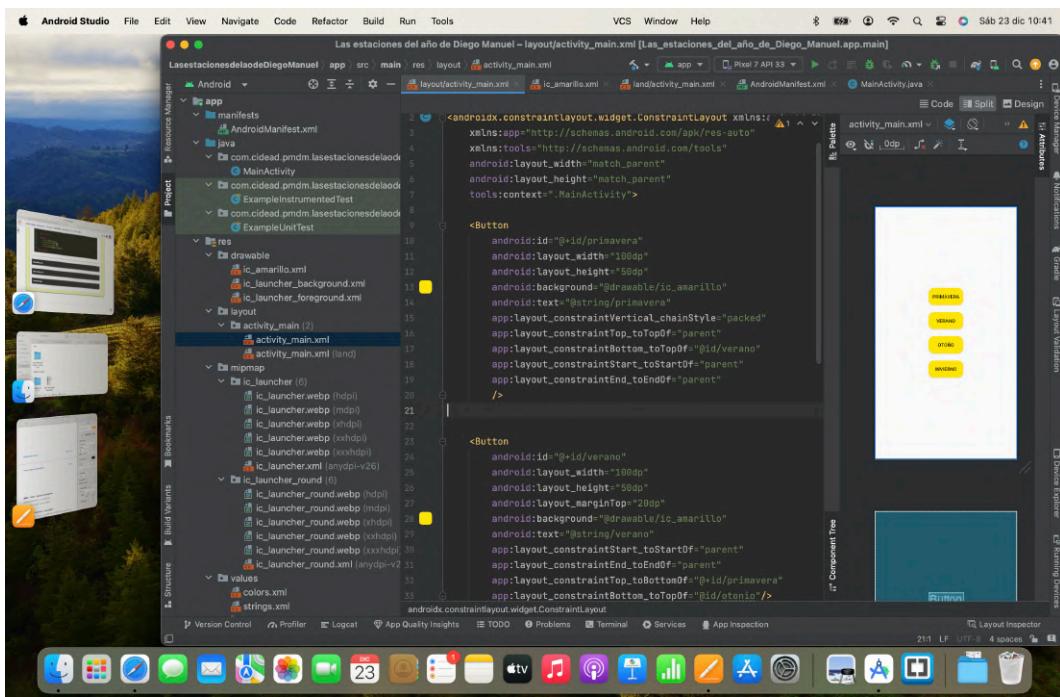
En nuestro caso hemos elegido AppCompact.DayNight.

Posteriormente crearemos el proyecto llamado "Las estaciones del año de Diego Manuel".



A continuación crearemos los botones dentro el archivo activity\_main.xml tal y como se muestra en la imagen.

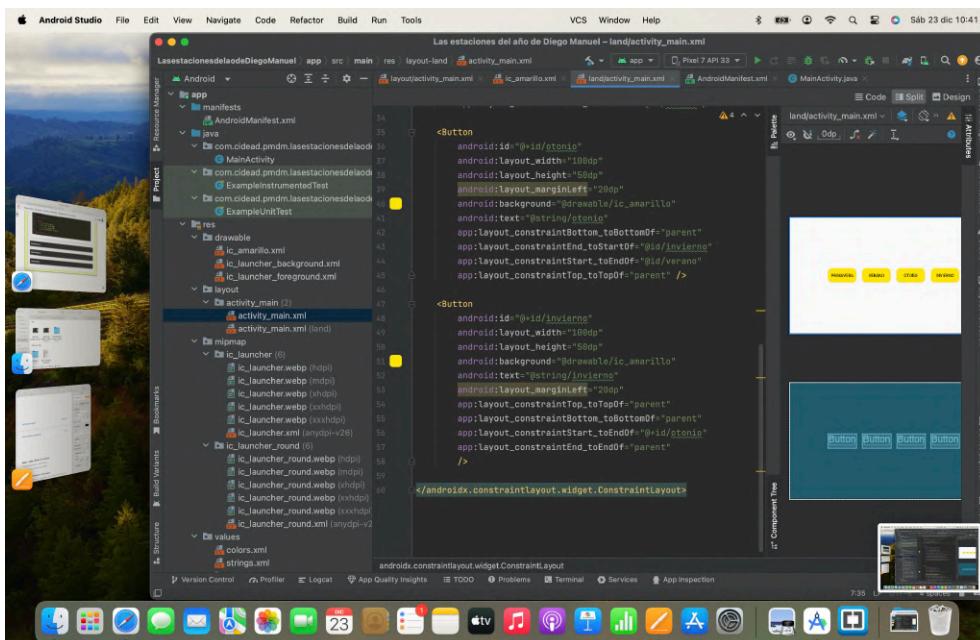
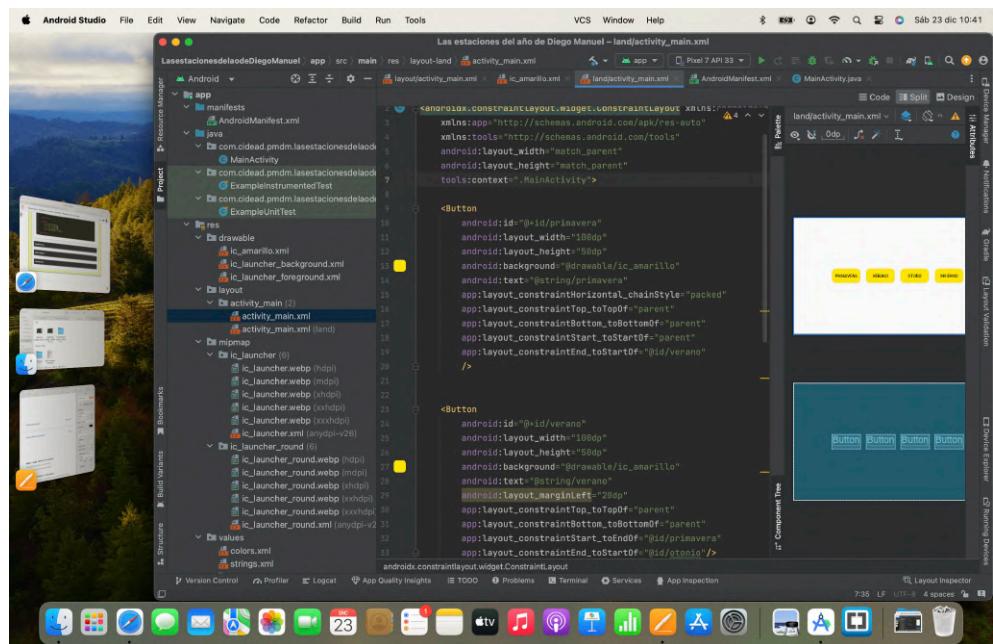
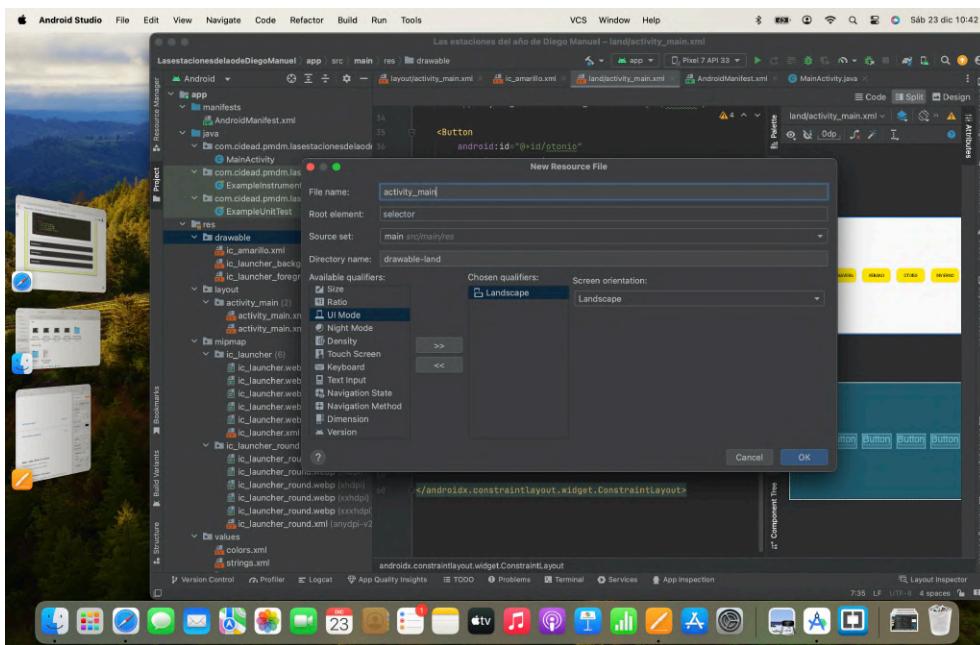




El siguiente paso será crear el archivo xml dentro de lay out para crear la interfaz de usuario para la pantalla apaisada.

Dentro de este xml creamos los botones, elegimos el estilo "packed" y los ajustamos a la pantalla apaisada.

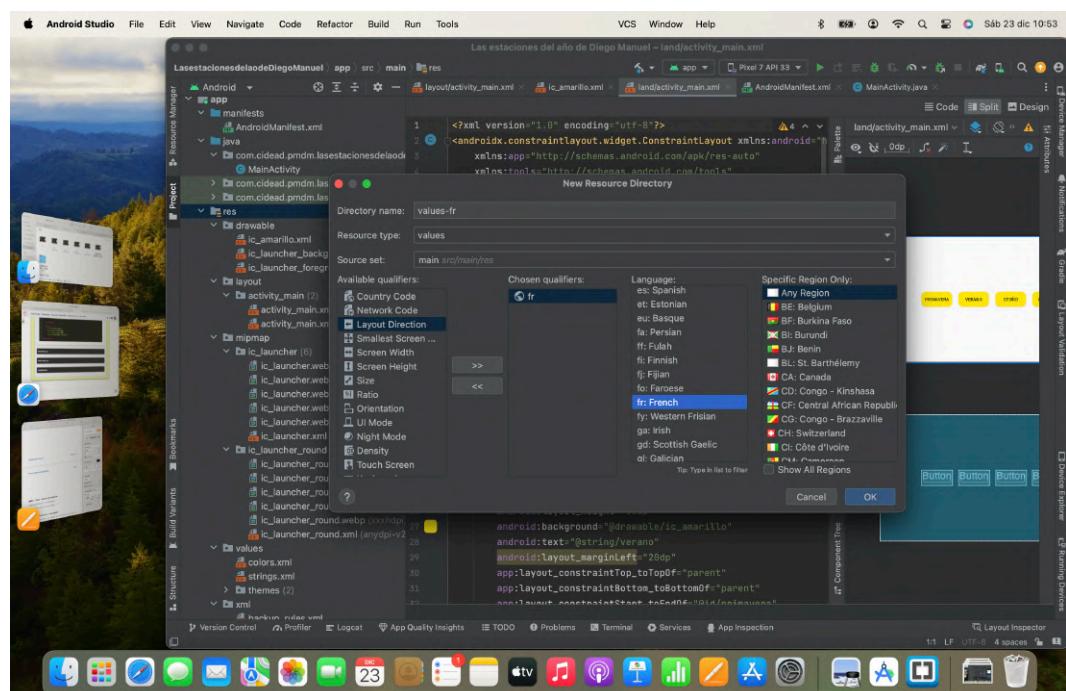
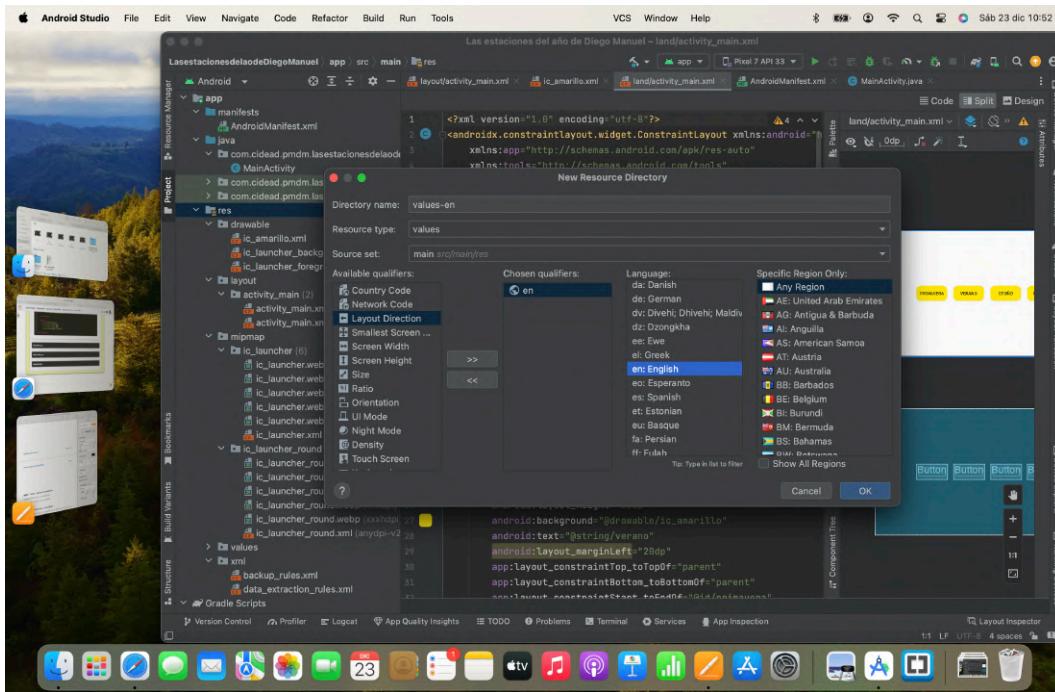
En las siguientes imágenes podemos verlo.

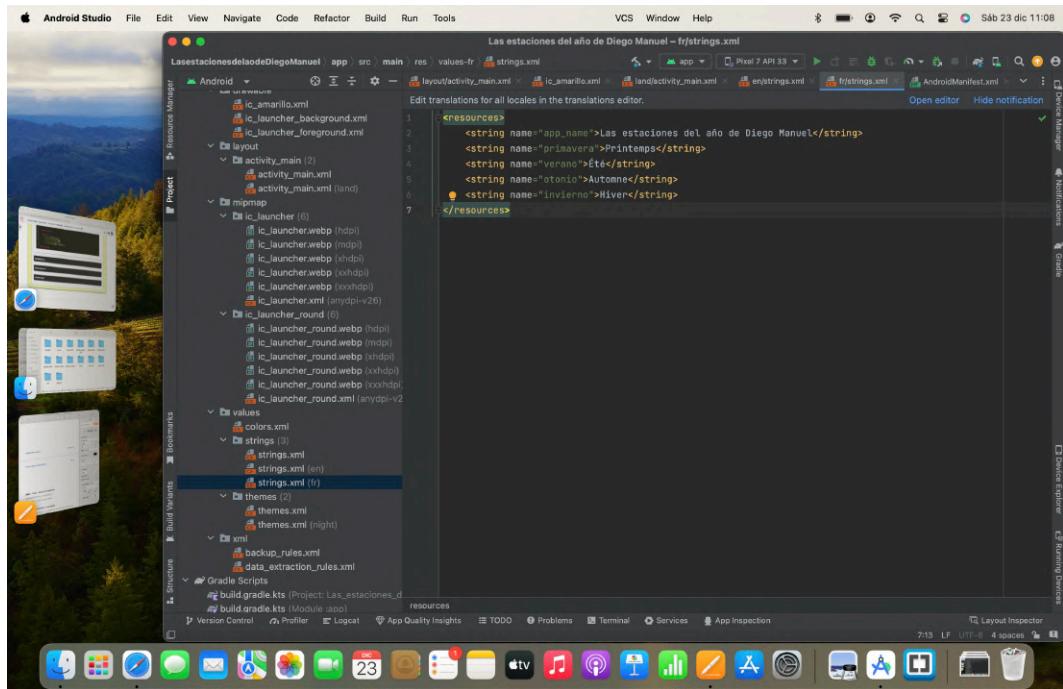
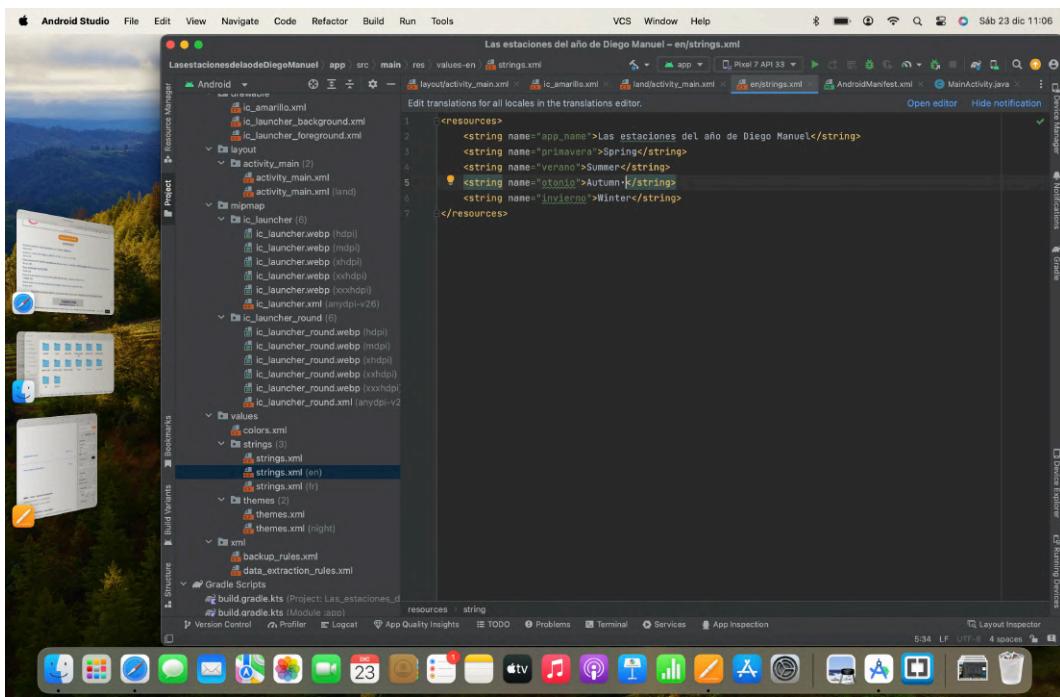


Lo siguiente que haremos será crear los archivos xml para los idiomas. Para ello crearemos dos directorios nuevos, uno llamado values-en y otro llamado values-fr.

Dentro del directorios values-en crearemos el xml para el idioma inglés y dentro del directorio values-fr crearemos el xml para el idioma francés.

En las siguientes imágenes podemos apreciar los pasos y el código.

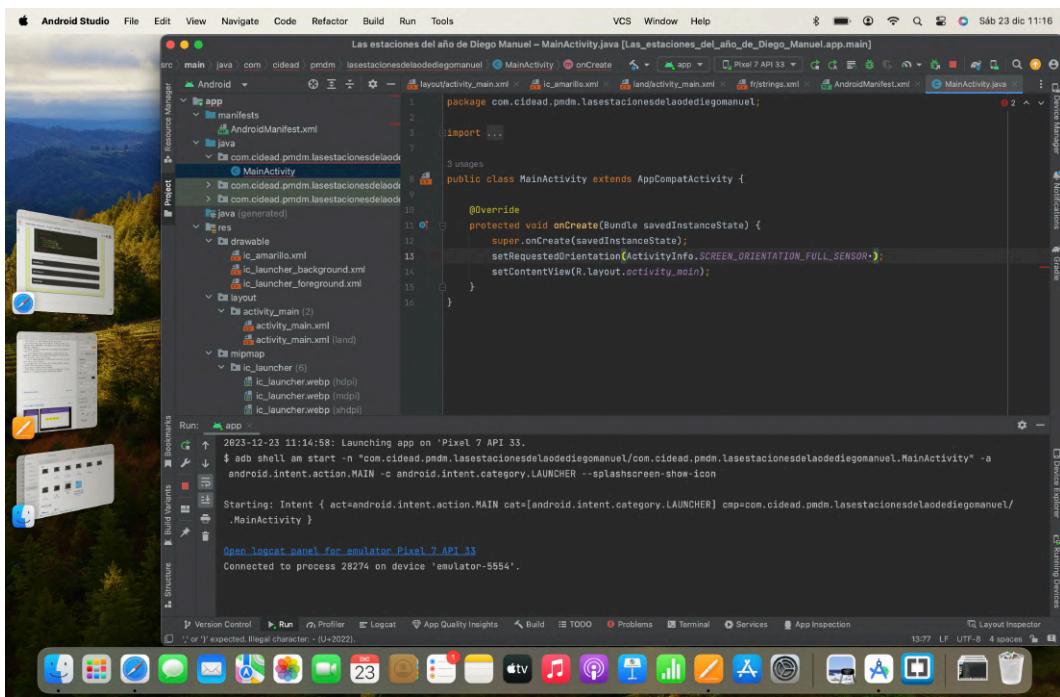




A continuación, dentro de mainactivity.java, mediante el método setRequestOrientation, estableceremos que se cambie el activity usado en función del sensor de orientación estableciéndolo a SCREEN\_ORIENTATION\_FULL\_SENSOR, así cuando el usuario está con el teléfono en vertical se usa la configuración para vertical y al girar a horizontal se usa la configuración para horizontal.

En la siguiente imagen podemos ver el código mencionado.

Y con esto damos por finalizada este apartado de la tarea.



### Tarea UT04.2. Spinner y Orientación

Realiza una app Android que permita seleccionar un color de una lista desplegable (Spinner) y que cuando se cambie de color se muestre el texto que se indica en la imagen adjunta.

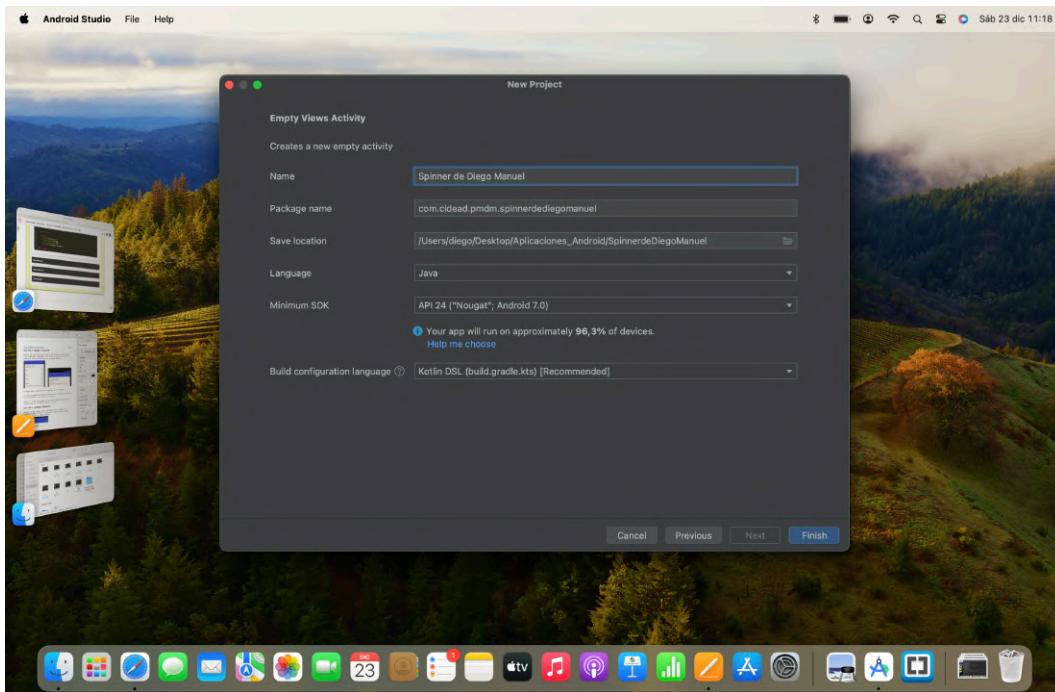


El nombre de la aplicación será "Spinner de + Tu nombre".

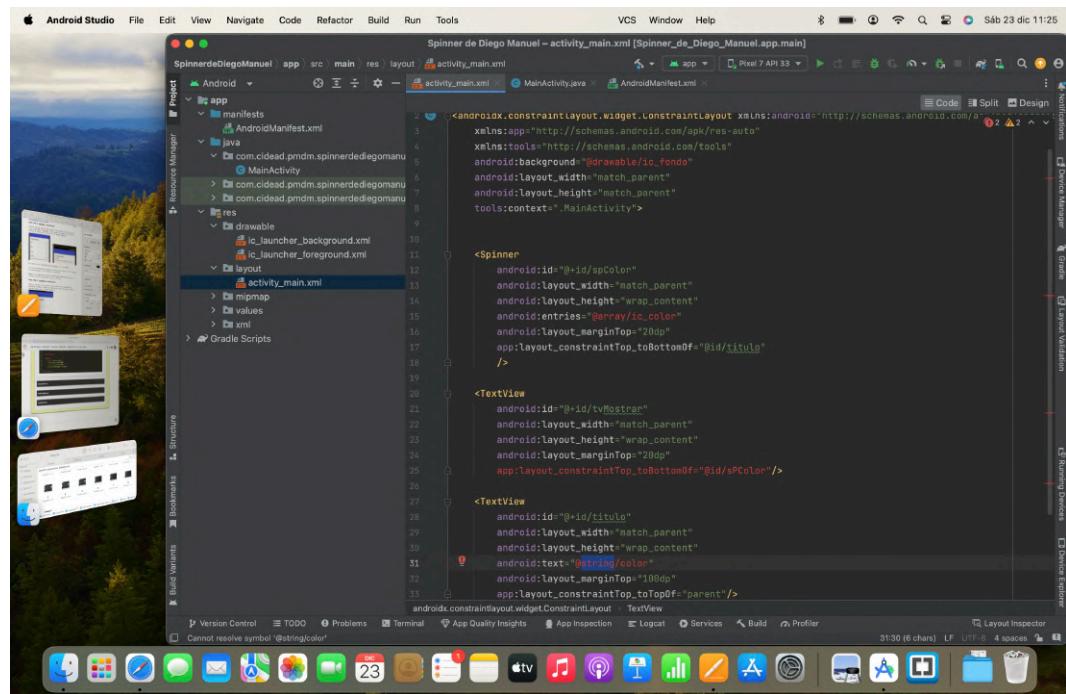
La lista contendrá los cuatro colores que se indican en la imagen y se almacenarán en un fichero de recursos.

Además, el diseño debe quedar perfectamente definido tanto si el teléfono esta en horizontal como en vertical.

Empezaremos creando el proyecto al que llamaremos "Spinner de Diego Manuel".

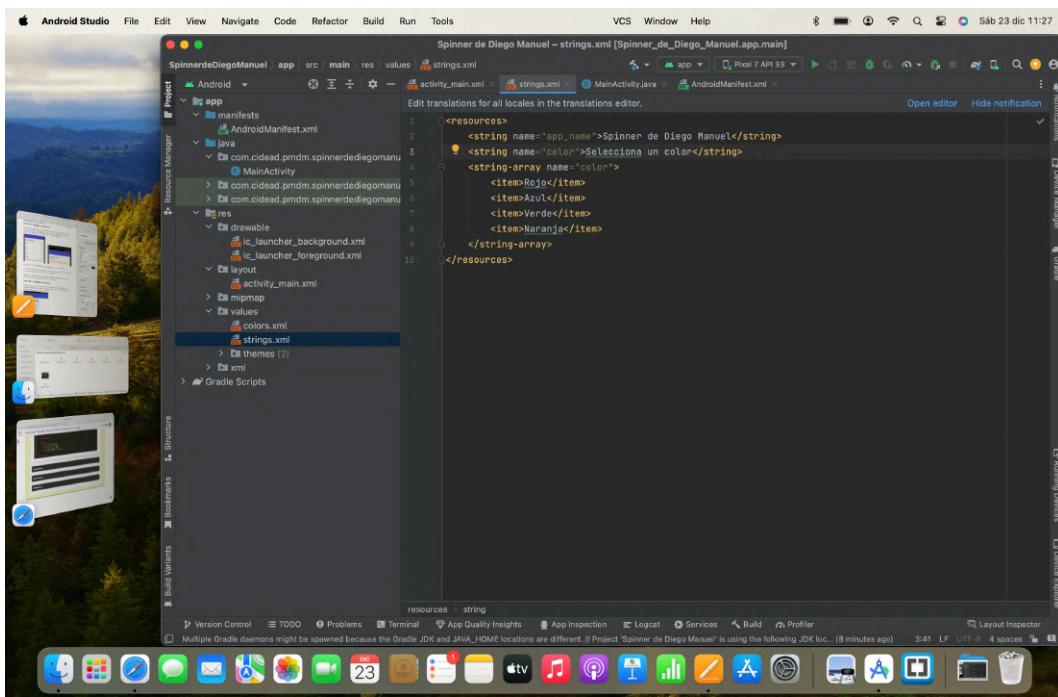


El siguiente paso será crear el spinner y los texview para la configuración vertical.



Posteriormente crearemos, en string, los stirng para contener las cadenas de texto y el array que contendrá la información del spinner.

En la siguiente imagen podemos apreciar el código de dicho archivo.

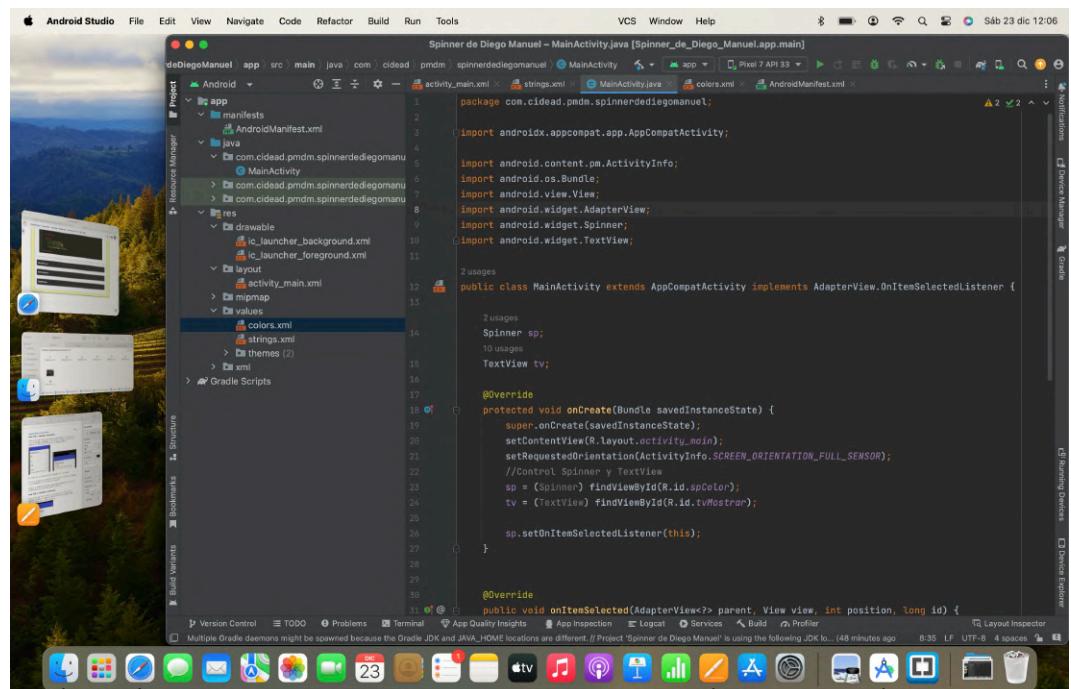


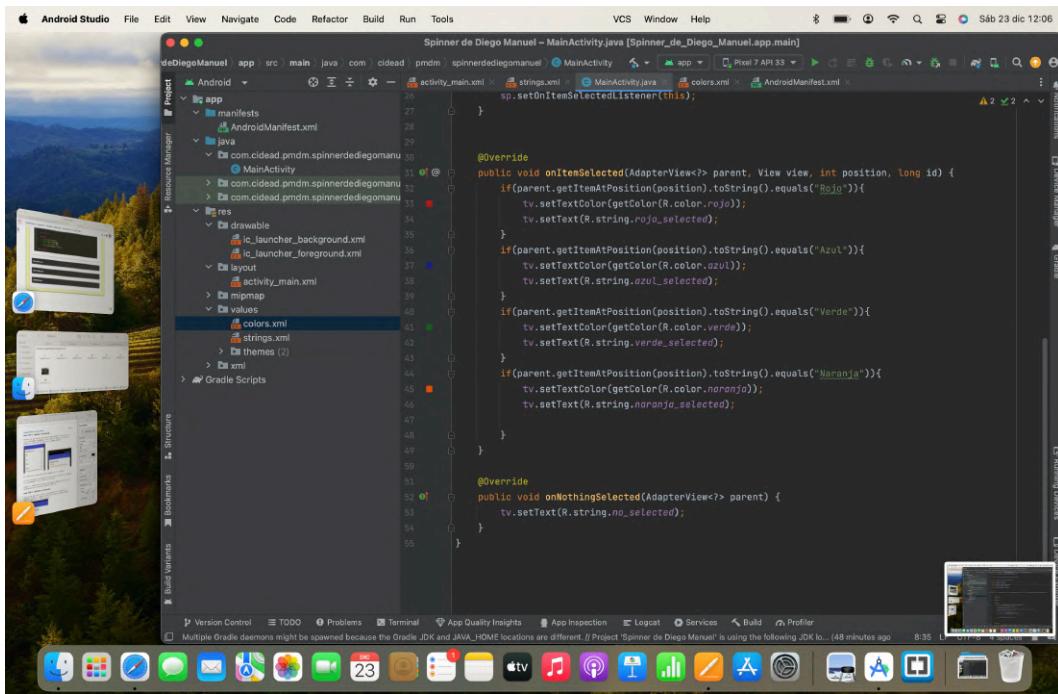
A continuación crearemos el código del mainactivity.

Crearemos dos variables, una de tipo spinner y otra de tipo textView y añadiremos el código para que seleccione el activity en función de la orientación de la pantalla.

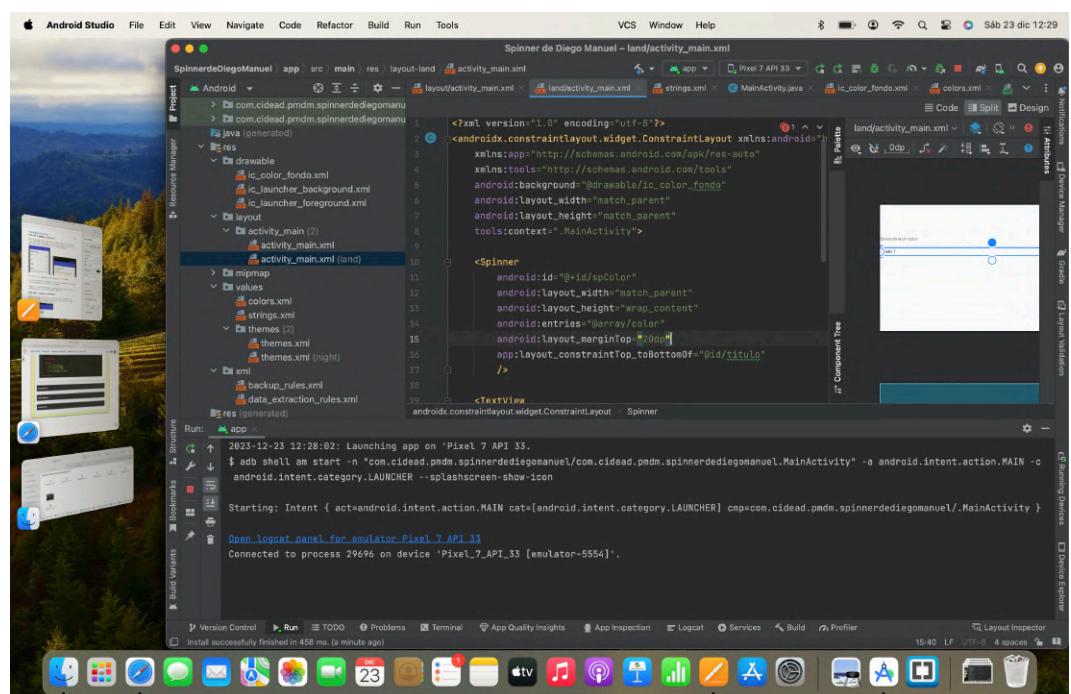
Posteriormente sobreescribiremos el método onItemSelected para que, mediante unos if y en función del ítem seleccionado en el spinner, muestre el texto y el color del texto acorde a ello.

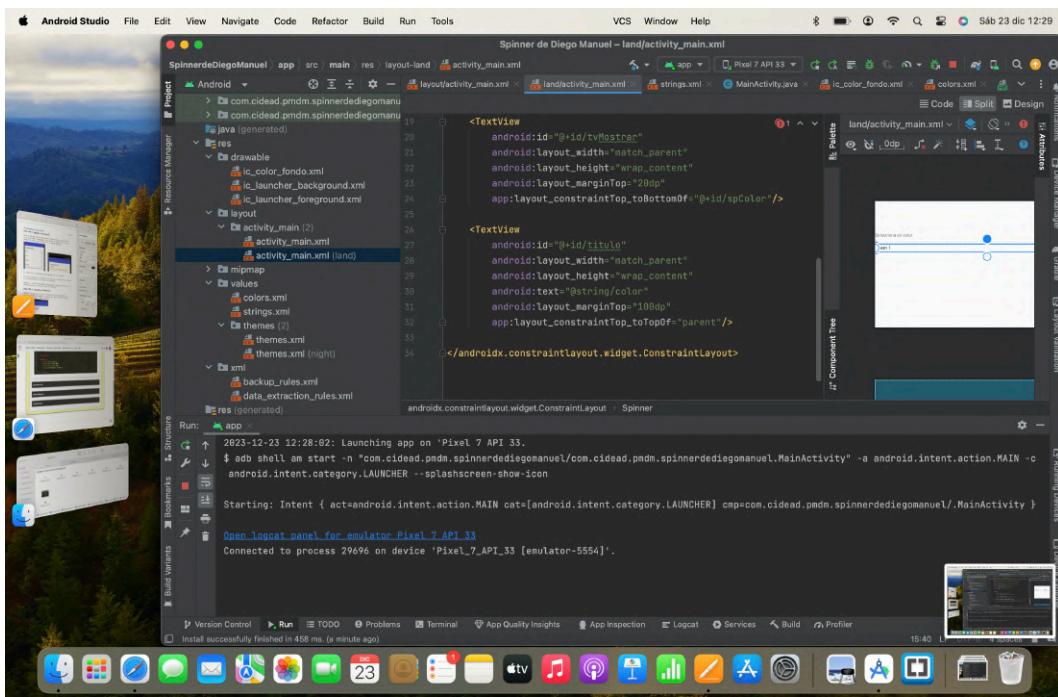
En las siguientes imágenes podemos ver el código.



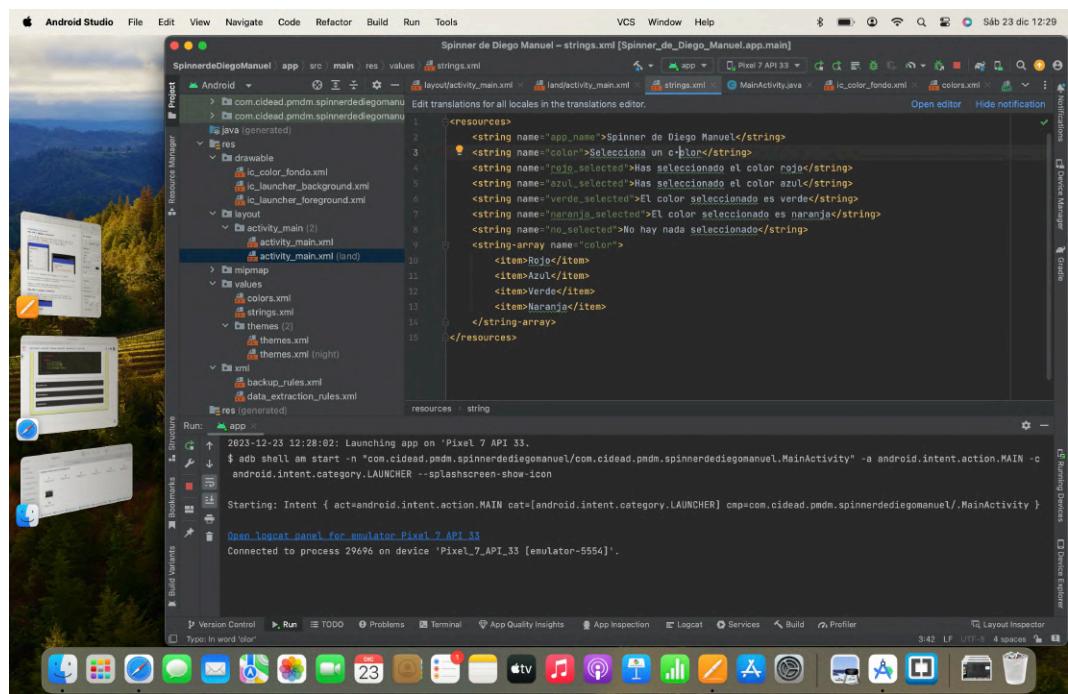


A continuación crearemos y configuraremos el archivo xml para cuando este en uso la pantalla apaisada al igual que hicimos para la pantalla vertical, peor adaptando al tamaño apaisado.





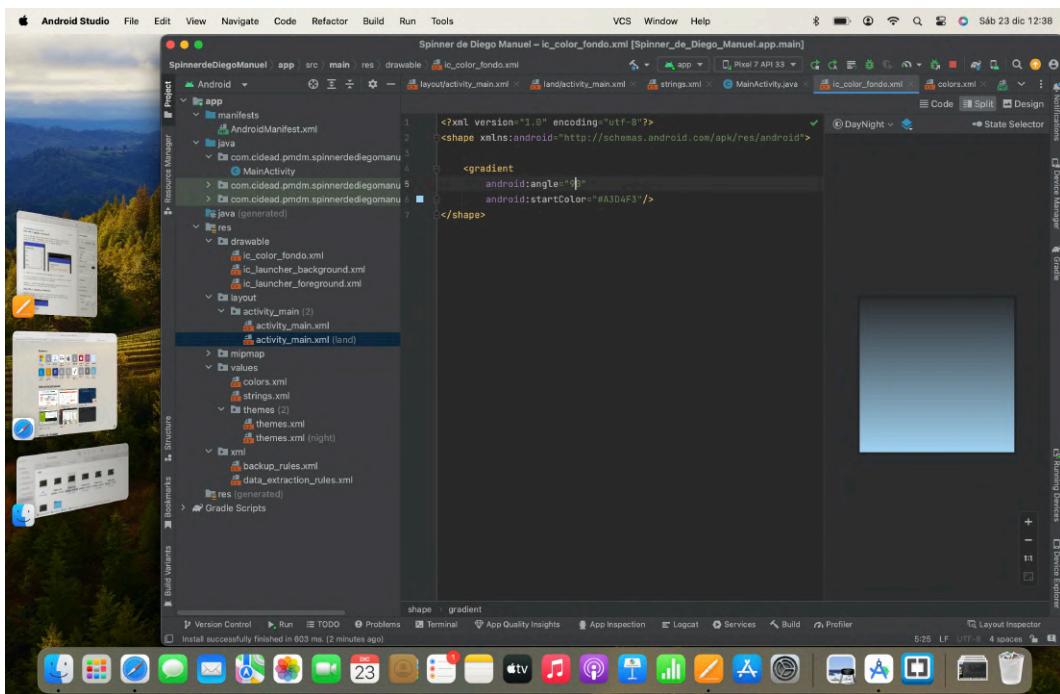
A continuación modificaremos el archivo string.xml para contener los strings de la aplicación y el array donde irán creados los colores.



A continuación crearemos un nuevo archivo xml llamado ic\_color\_fondo que irá ubicado dentro del directorio res - drawable.

Este archivo lo usaremos para establecer el color de fondo de la app.

En la siguiente imagen podemos ver el código del citado archivo.



Y con esto damos por terminado este apartado de la tarea.

#### Tarea UT04.3. ListView y Orientación

Realiza una app Android que al seleccionar una ciudad de un ListView indique el número de habitantes que tiene esa ciudad, como se indica en la imagen adjunta.

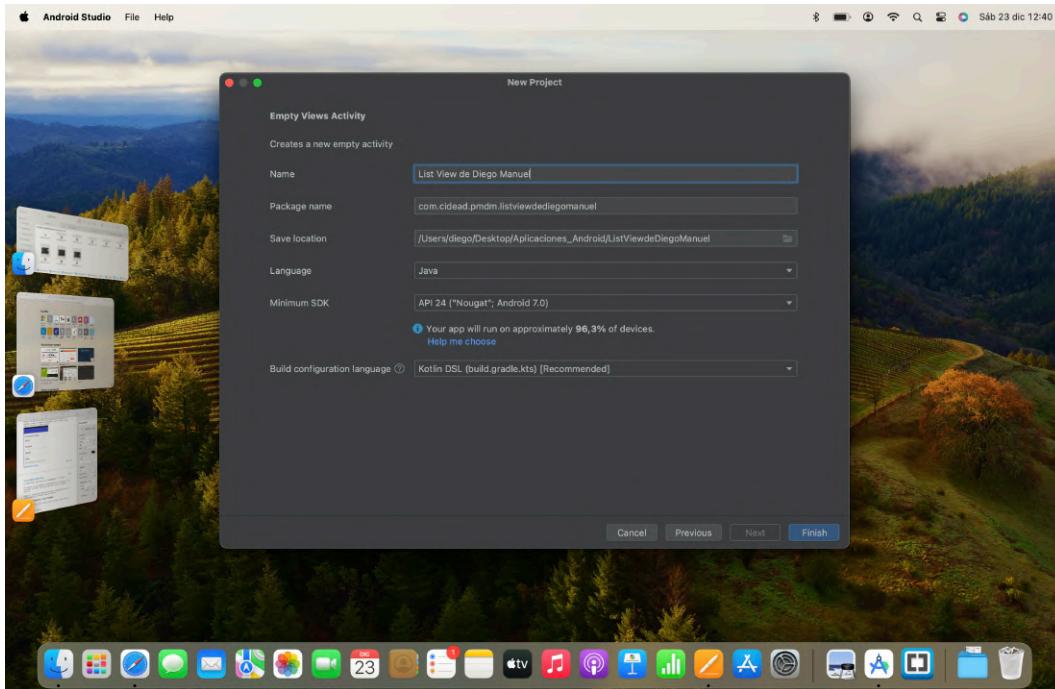


El nombre de la aplicación será "ListView de + Tu nombre".

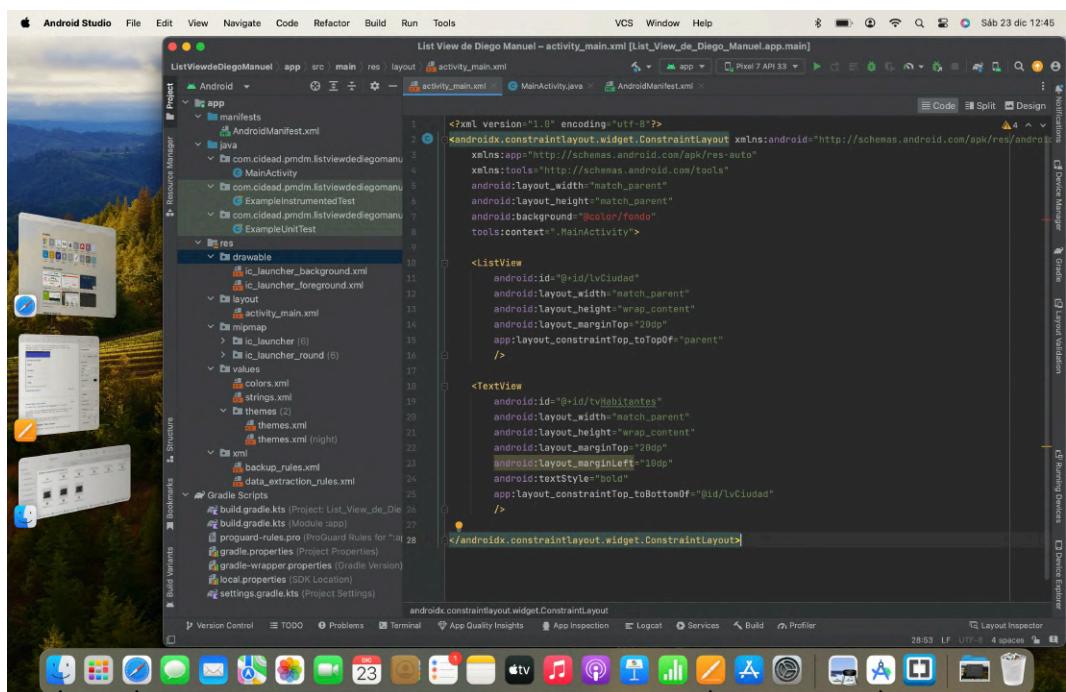
Los nombres de las ciudades se almacenarán en un ArrayList y los habitantes se pueden almacenar como cada alumno/a considere oportuno.

Además, el diseño debe quedar perfectamente definido tanto si el teléfono está en horizontal como en vertical.

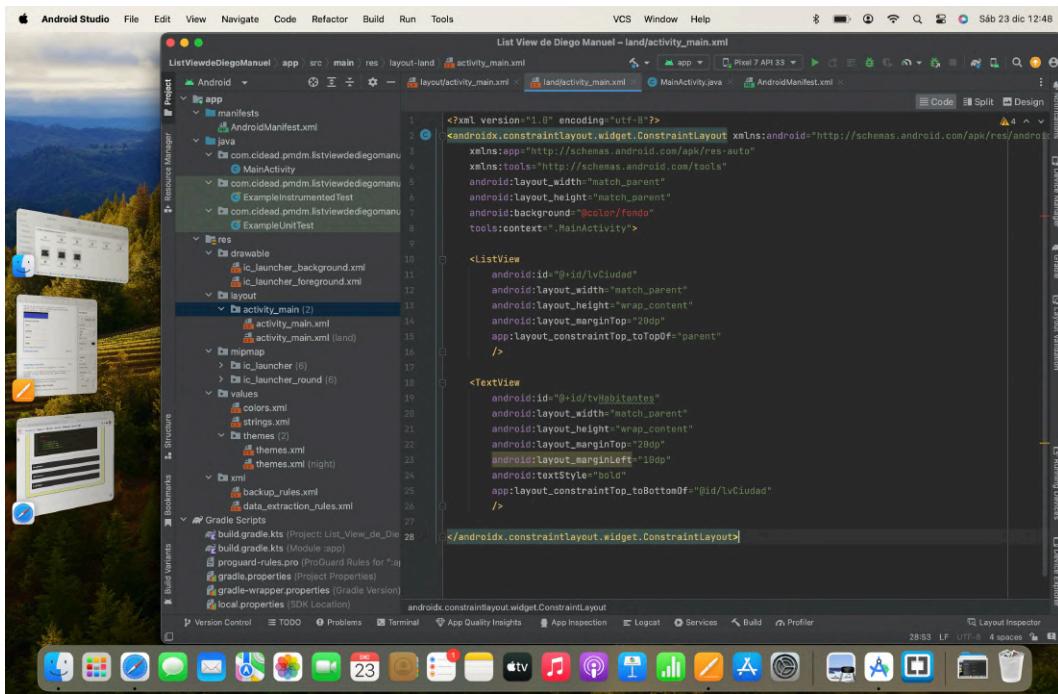
Lo primero que haremos será crear la aplicación llamada "List View de Diego Manuel".



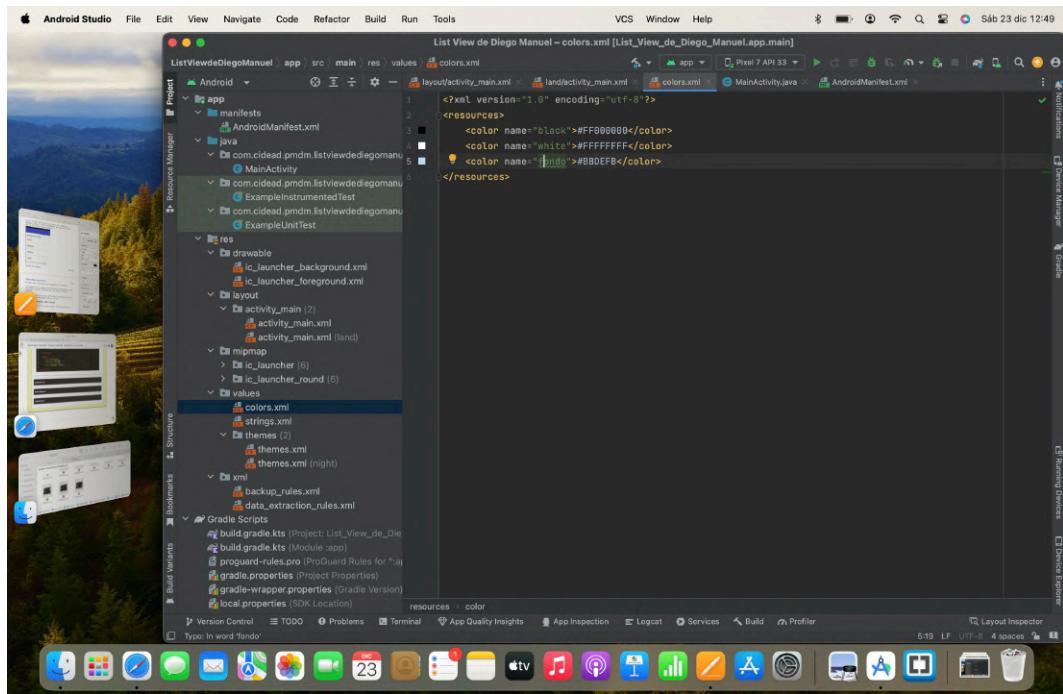
El siguiente paso será crear el contenido del activitymain donde crearemos un listview y un textview.



A continuación crearemos y configuraremos el activity para las pantallas apaisadas.



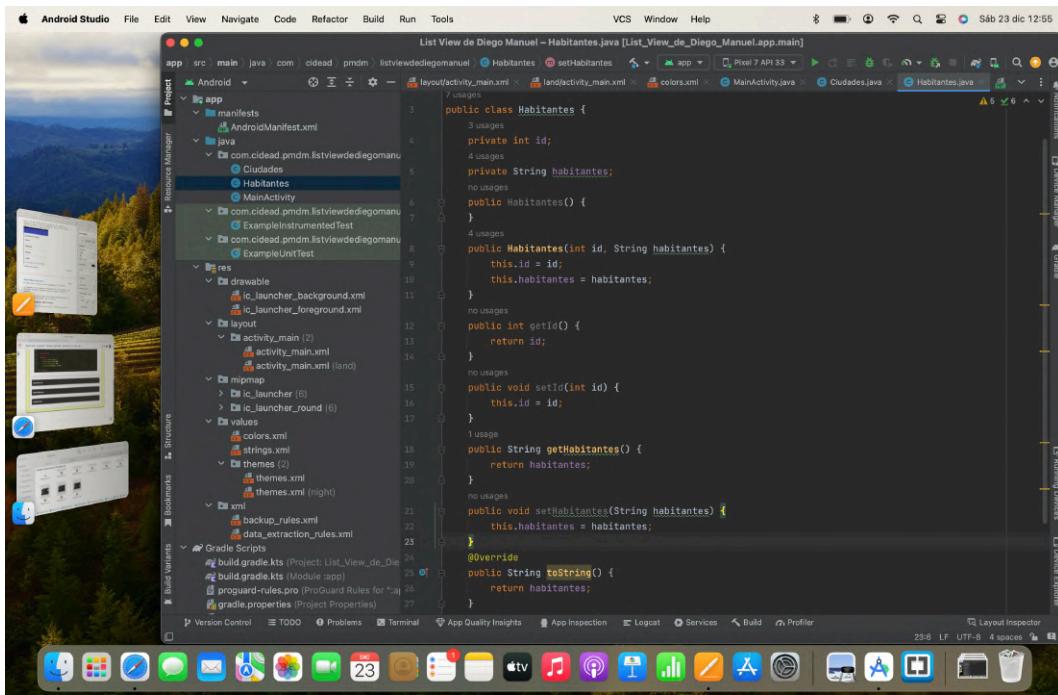
Posteriormente añadiremos al xml de color el color de fondo que vamos a usar en la aplicación.



Lo siguiente sera crear la clase java llamada habitantes.

Dicha clase contendrá dos atributos privados, uno de tipo int llamado id y otro de tipo string llamado habitantes.

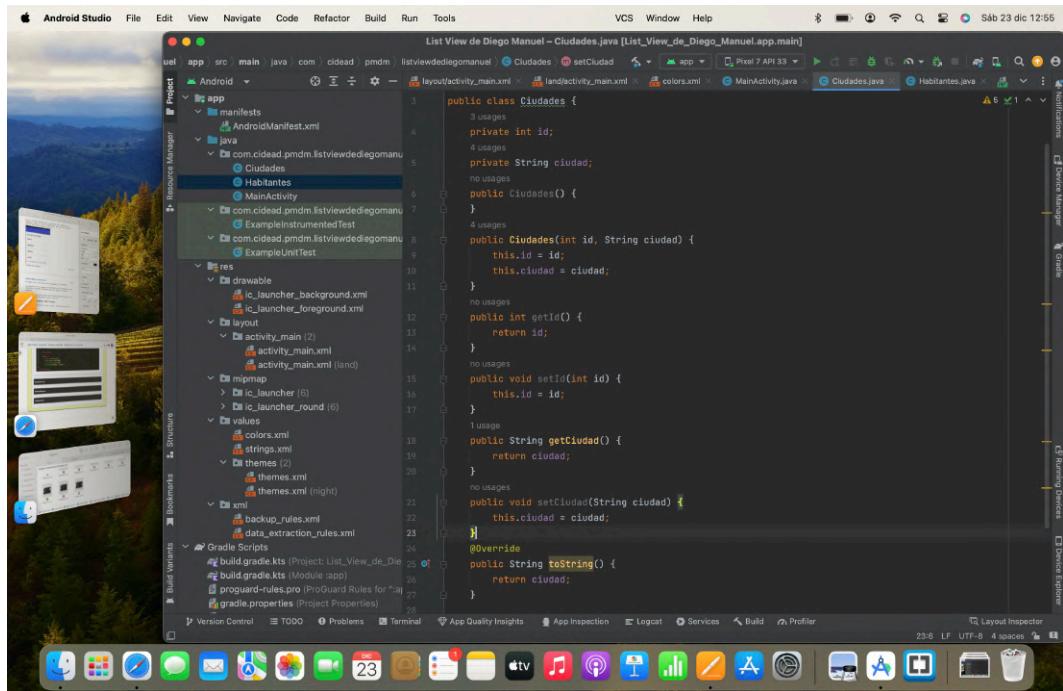
Tambien dispondrá del constructor por defecto, otro constructor con los dos parámetros y los getter and setter.



A continuación crearemos la clase ciudades.

Dicha clase tendrá dos atributos privados, uno de tipo int llamado id y otro de tipo string llamado ciudad.

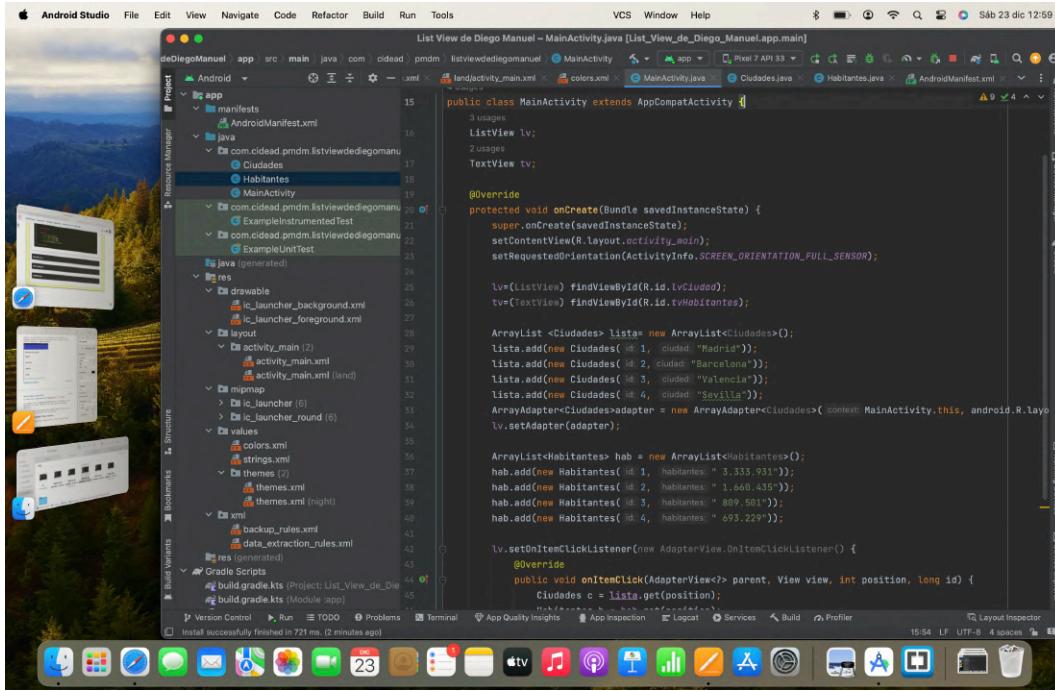
Ademas dispondrá del constructor por defecto, el constructor con los dos parámetros y los guetter and setter.



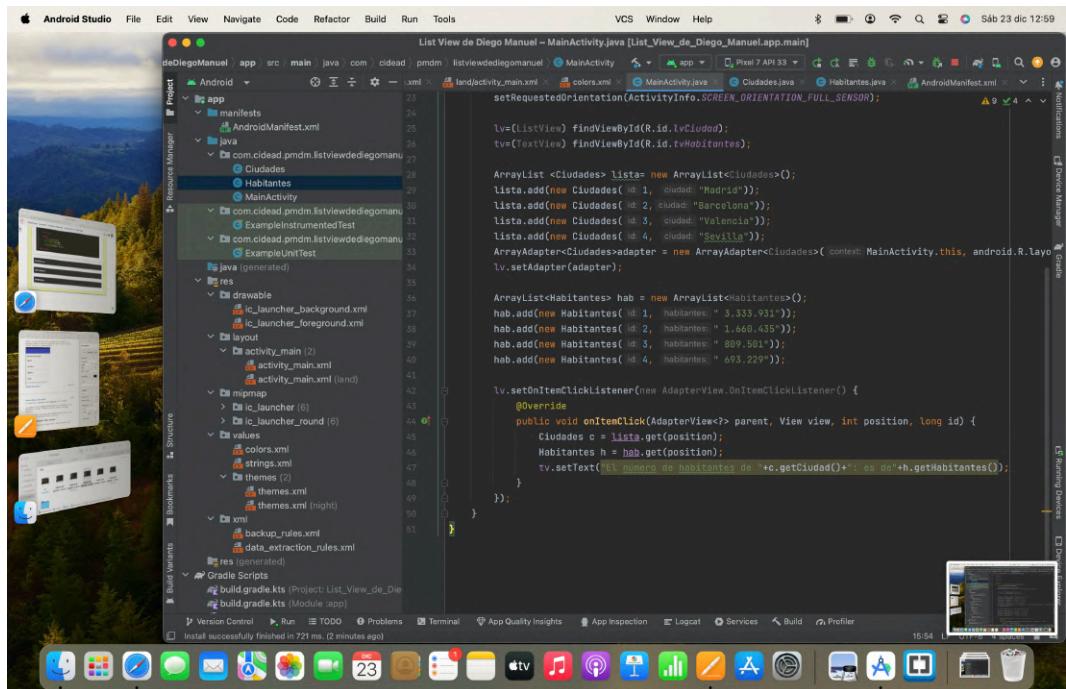
Tras ello crearemos el código de la clase mainactivity.

Dicha clase contendrá dos atributos, uno de tipo listview y otro de tipo textview.

También contendrá dos array, uno de tipo ciudades y otro de tipo habitantes. Dichos array contendrán los nombres de las ciudades y el número de habitantes respectivamente, que añadiera mediante el método add.



Por último sobreescribiremos el método setOnItemClickListener para que escoja de los array los datos en función de lo que seleccionemos en el listview y concatenara los datos extraídos con una frase.



Y con esto damos por finalizada este apartado de la tareas y con ello, la tarea al completo.

**Criterios de puntuación. Total 10 puntos.**

La valoración total de la tarea es de 10 puntos repartidos del siguiente modo:

**Tarea UT04.1 (6 puntos)**

- Localización: la aplicación se muestra correctamente traducida para los 3 idiomas. 1,5 puntos
- Aspecto variable según orientación: la aplicación se muestra tal y como pide el enunciado en ambas orientaciones. 3 puntos.
- Correcta organización de los recursos en ficheros: colores, cadenas de textos, identificadores y aspecto. 1,5 puntos.

**Tarea UT04.2 (2 puntos) y Tarea UT04.3 (2 puntos):**

- Aspecto gráfico: 1 punto
- Programación: 1 punto

**Recursos necesarios para realizar la Tarea.**

- Ordenador con Android Studio instalado y suficientes recursos para ejecutar el emulador.
- Contenidos de la Unidad, muy importantes los ejercicios resueltos de la misma.
- Páginas web de los desarrolladores de los sistemas operativos para móviles.
- Páginas web de estadísticas como gs.statcounter.com.

**Consejos y recomendaciones.**

- Lee atentamente el enunciado y asegúrate de haber entendido lo que has de hacer.
- Intenta reproducir en tu Android Studio los ejercicios resueltos, especialmente los proyectos 2, 3, 6 y 7 de este tema; estos te darán todas las claves para acometer la tarea.
- No dudes en comentarle a tu tutor o tutora cualquier duda que te pueda surgir.
- Envíasela a tu tutor o tutora a través del sistema establecido en la plataforma.
- Las capturas y el contenido de los ficheros deben aparecer en perfecto orden para que esté claro lo que deseas mostrar.
- Los diseños deben mostrarse lo más parecidos posibles a lo que se pide.

Recuerda que es una parte importante de la unidad.

- Debe llevarse a cabo sobre una versión actual de Android.
- Debe mostrarse el fondo de pantalla y la fecha y hora de todas las capturas para garantizar la originalidad del material.

**Indicaciones de entrega.**

Una vez realizada la tarea elaborarás un documento PDF con los pasos realizados para la elaboración de cada una de los ejercicios de la tarea.

Las capturas deben mostrar el fondo de pantalla y la fecha y hora de la barra de estado de tu sistema operativo para garantizar la originalidad del material.

Muy importante que personalices tu nombre en la aplicación.

También se deberá entregar la carpeta comprimida del proyecto con cada uno de los ejercicios en formato .zip. En todos los Activity\_main.java debe aparecer el nombre del autor de la entrega.

Entrega los archivos a través de la plataforma. Se nombrarán siguiendo las siguientes pautas:

apellido1\_apellido2\_nombre\_PMDM04\_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños.

Así por ejemplo la alumna Begoña Sánchez Mañas para la cuarta unidad del MP PMDM, debería nombrar esta tarea como...

sanchez\_manas\_begona\_PMDM04\_Tarea