# CURSO F220042AA IFCM018PO PROGRAMACIÓN DE APLICACIONES ANDROID AF25 GR3

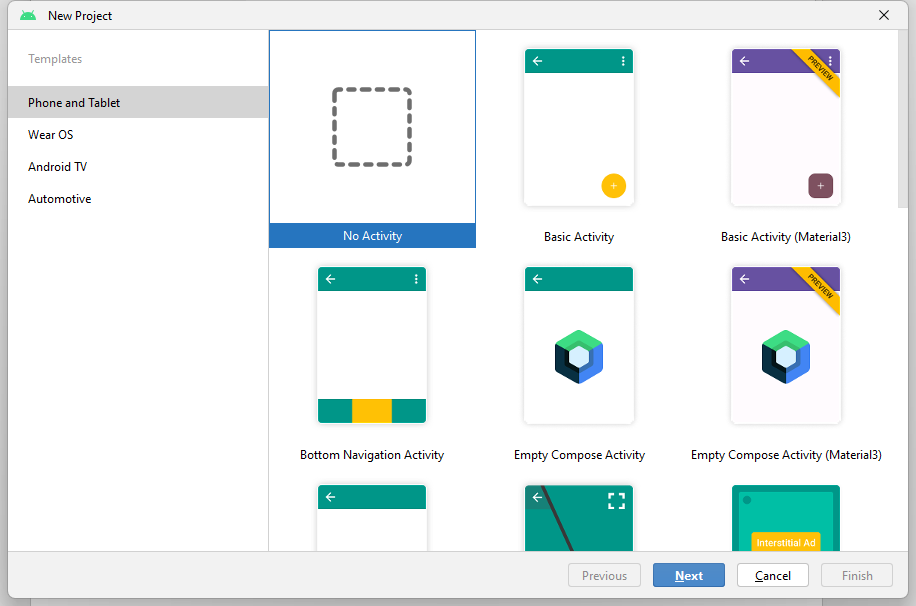
**EVALUACION FINAL**

SERGIO RODRIGUEZ GIL

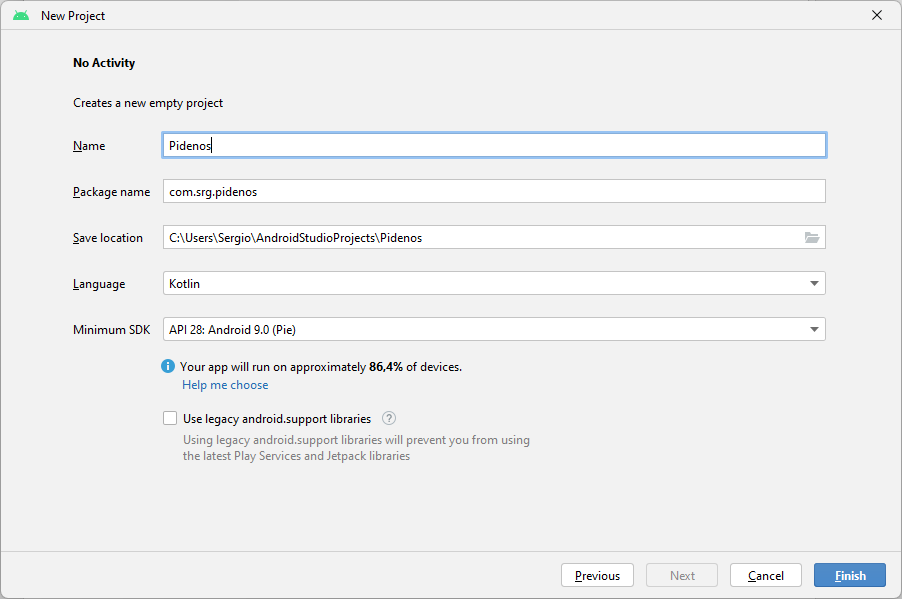
Aplicación PIDENOS

COMIENZO:

Instalamos el Android Studio, lo actualizamos y procedemos a crear un proyecto nuevo. En este caso sin actividad inicial.

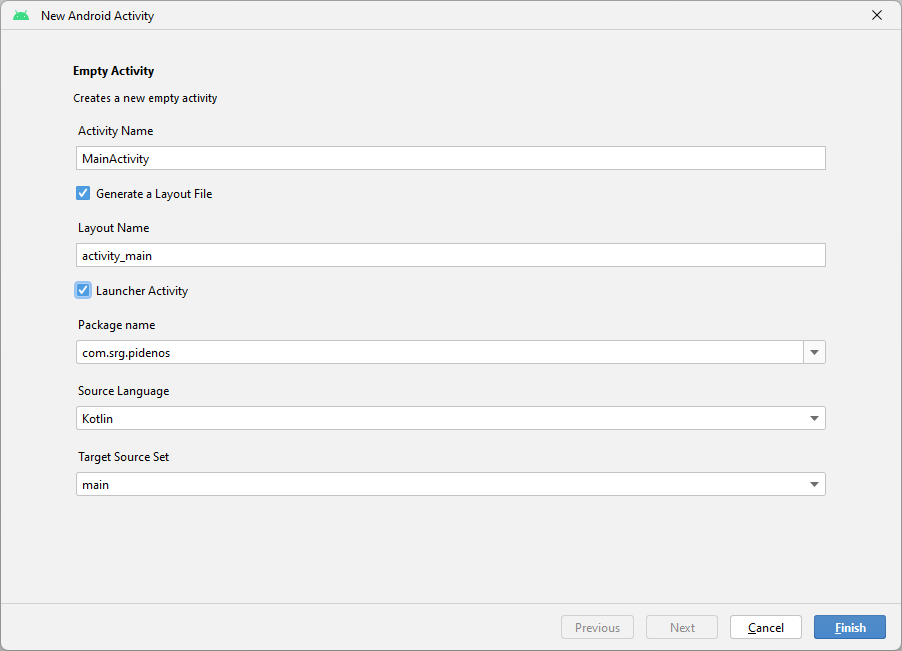


Introducimos las características requeridas para esta aplicación, y pinchamos en Finish. Con esto ya se crea el proyecto.

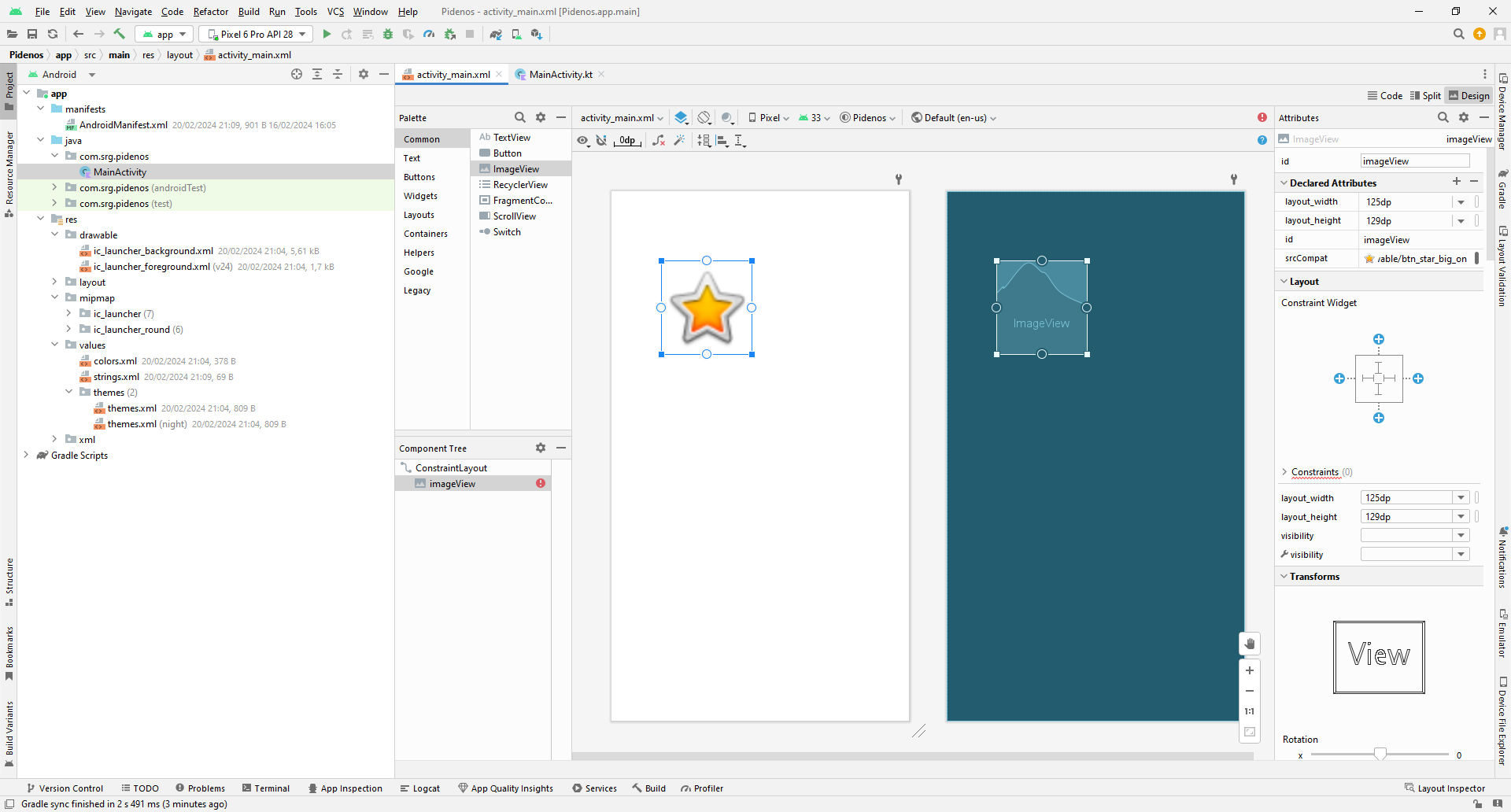


PARTE 1.

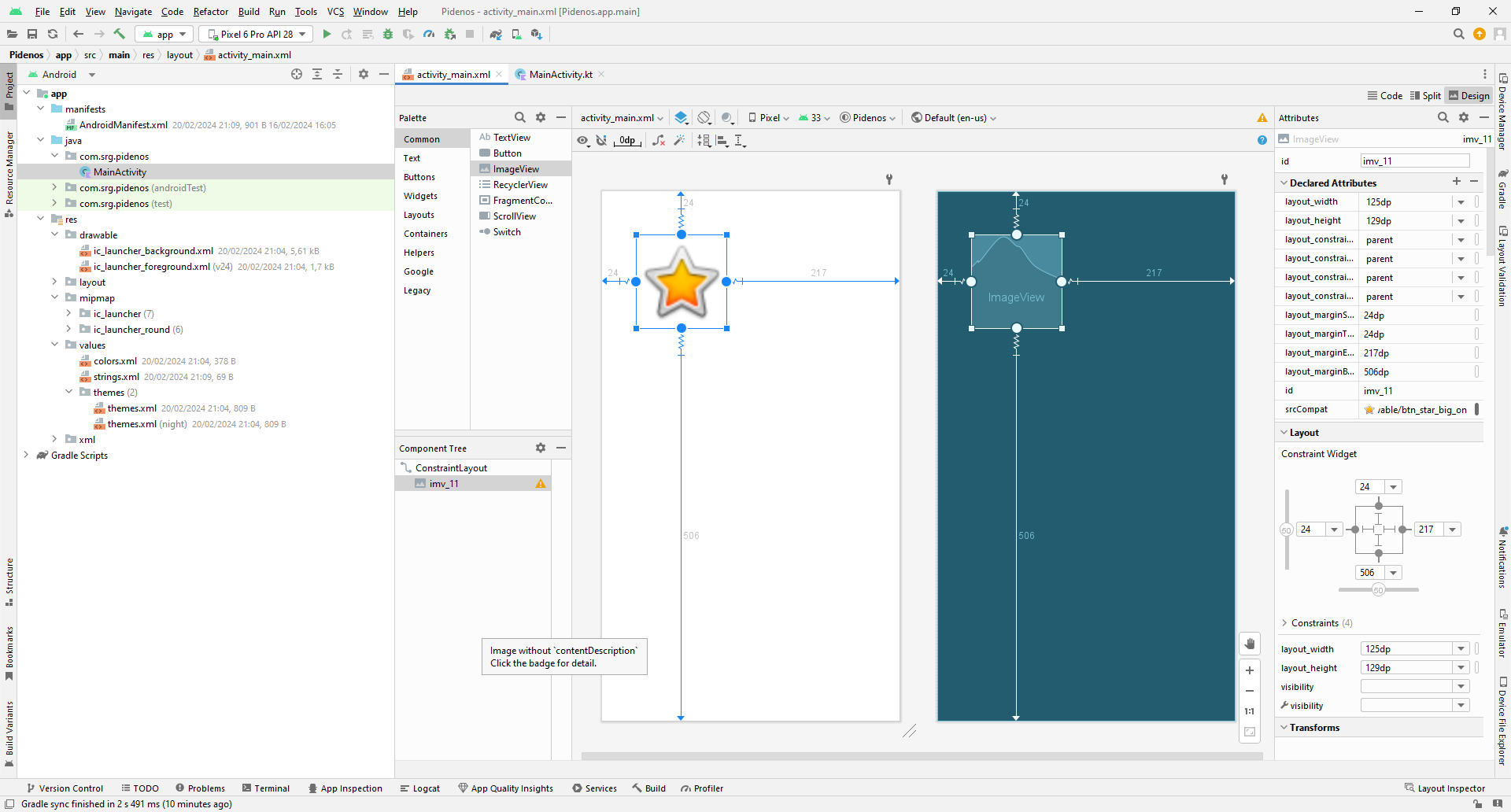
Seleccionamos *File/New/Activity/Empty Activity*. Dejamos todos los valores por defecto menos el de “Launcher activity” que marcaremos.



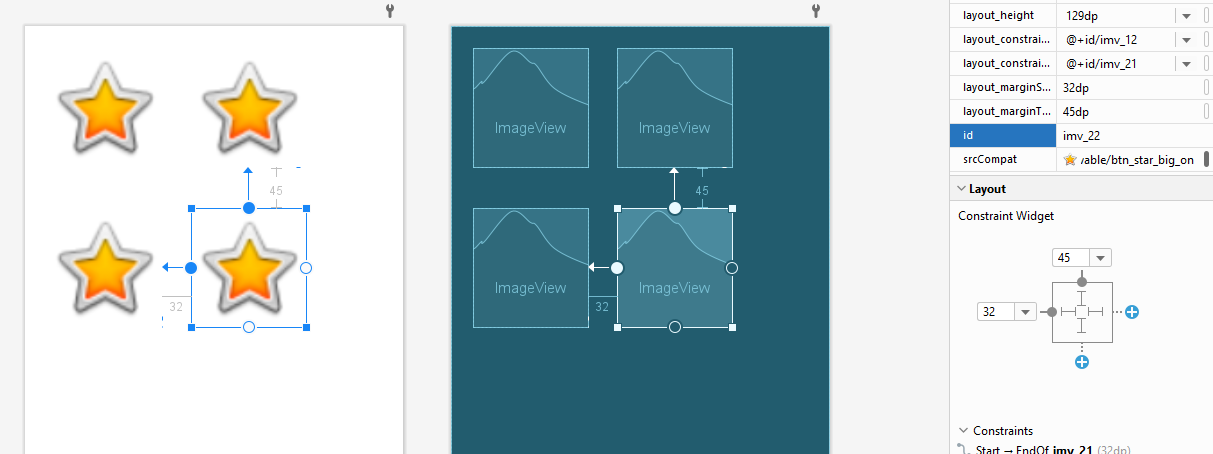
Ahora, desde el archivo *activity\_main.xml* procedemos a insertar un elemento de tipo ImageView, y añadimos una imagen (en este caso una del mismo repositorio del Android Studio).

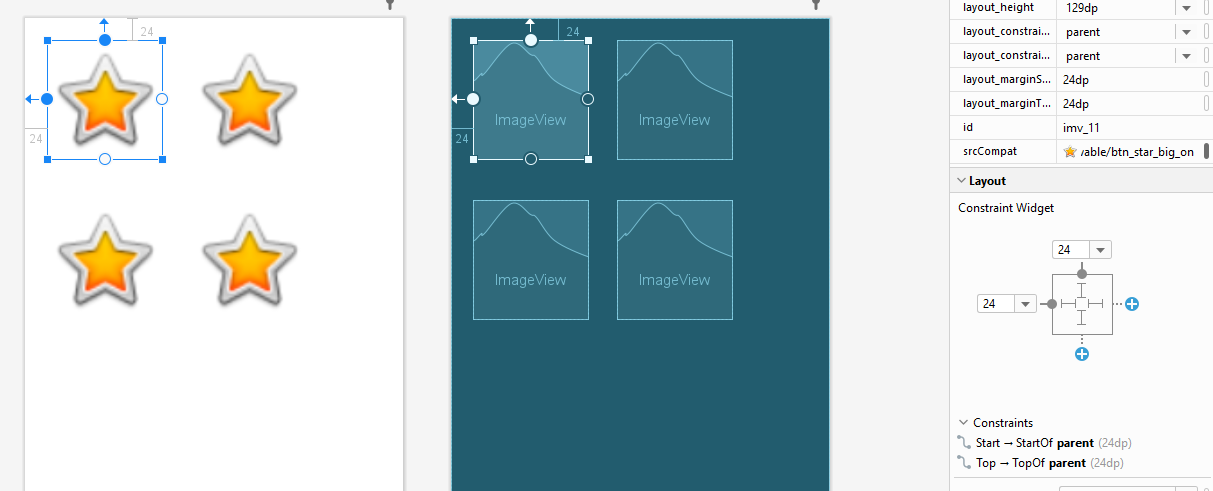


Pinchando en la sección Layout, en los ajustes de Constraints establecemos la distancia superior y la izquierda a 24. Tambien aprovechamos para cambiar el id a imv\_11, para tener claramente identificada la imagen.

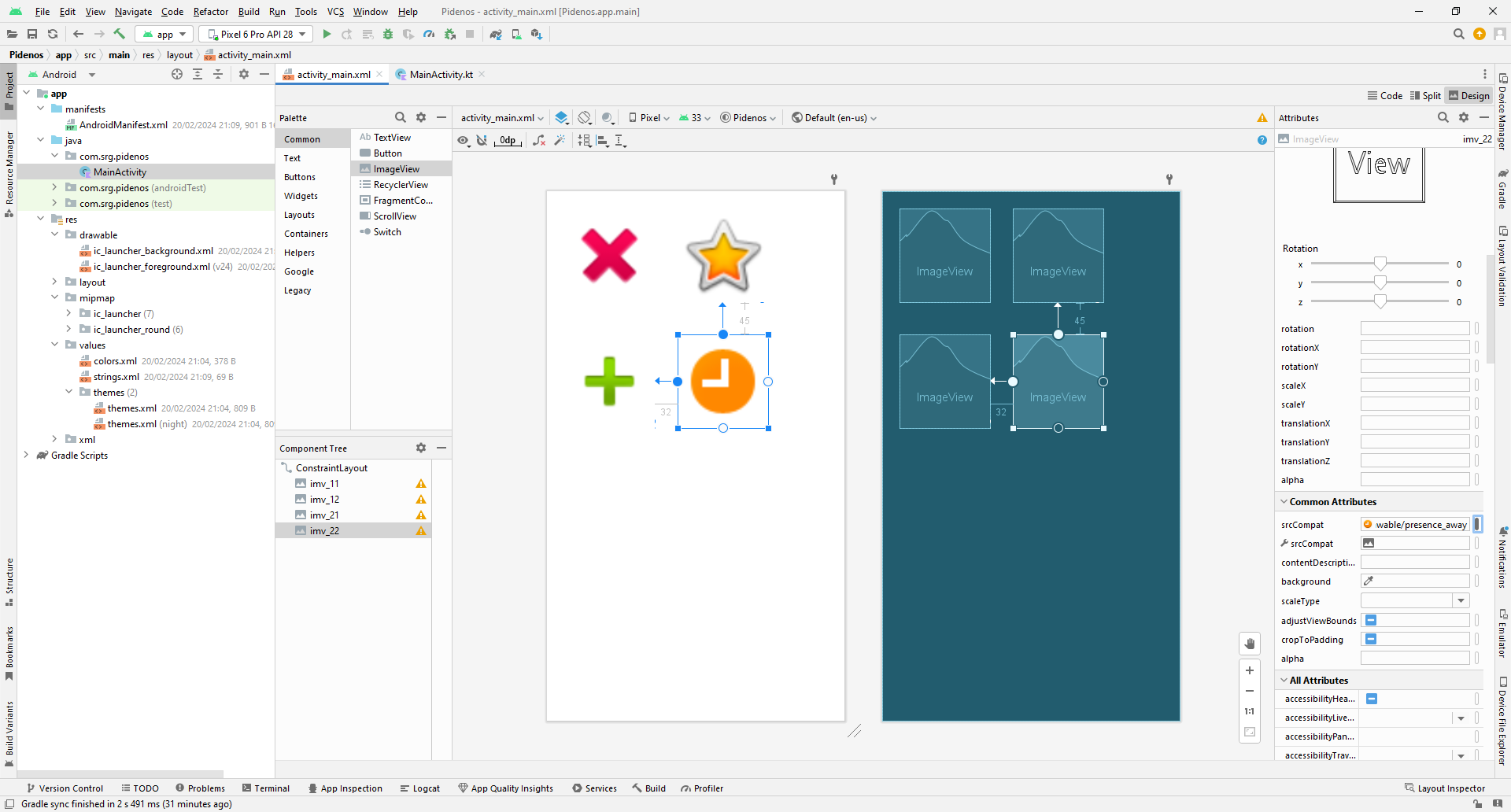


Y después podemos hacer un copia/pega de la imagen estableciendo el resto de constraints, siguiendo las especificaciones.

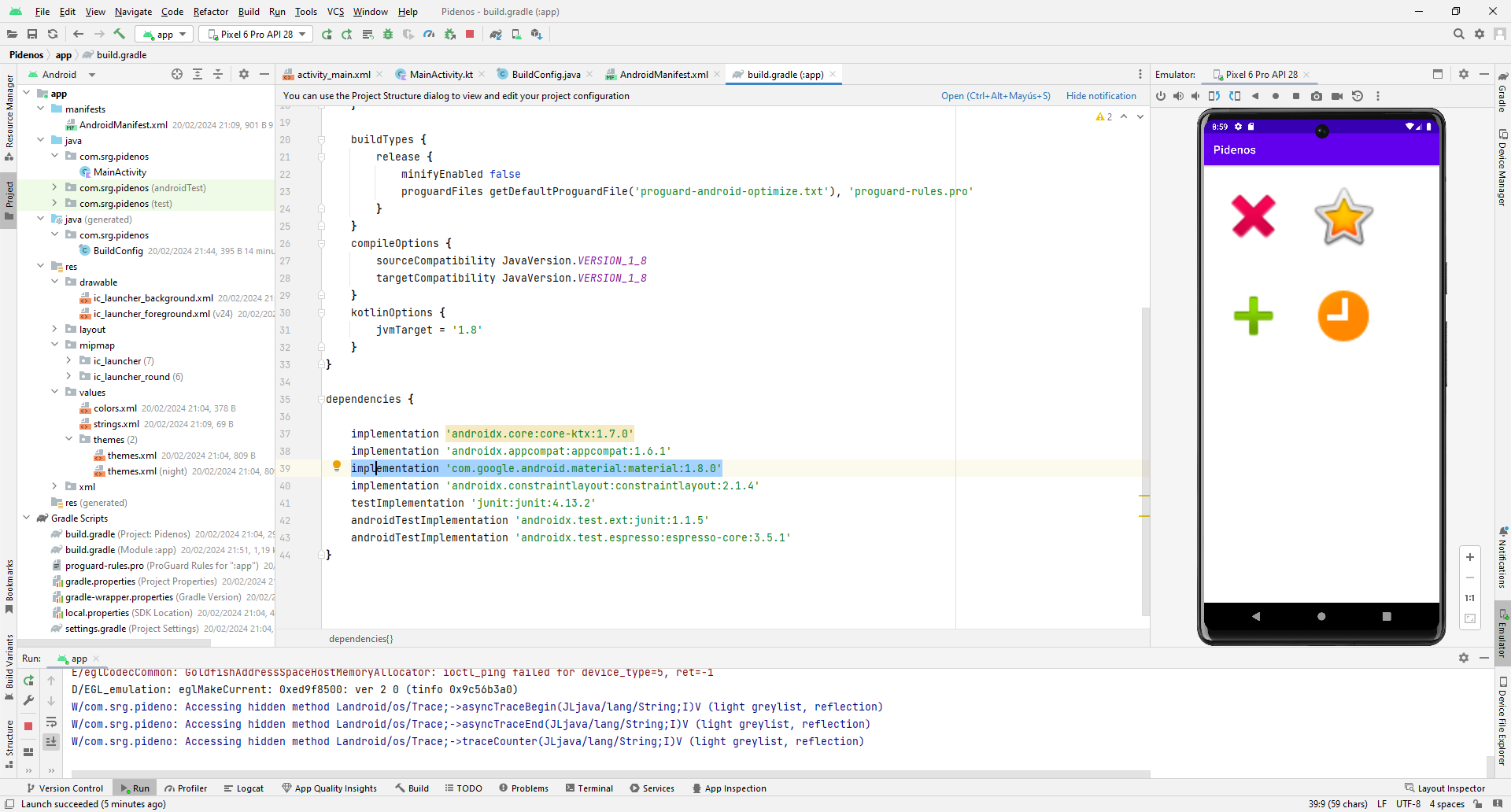




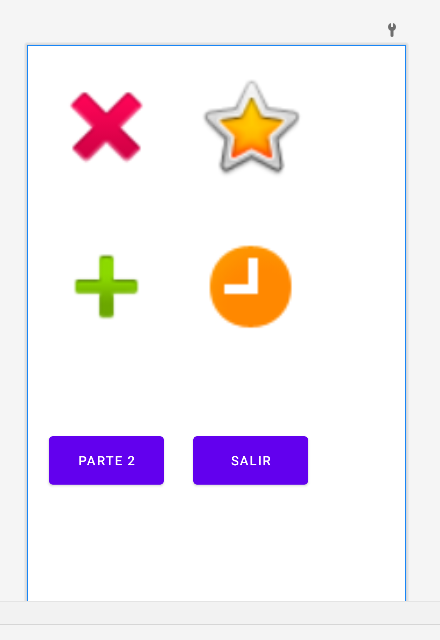
El resto de imágenes las renombramos siguiendo un patron de columna-fila, quedando asi: imv\_11, imv\_12, imv\_21, imv\_22. Por ultimo en la propiedad srcCompat podemos seleccionar otras imágenes para poder distinguirlas.



Antes de pasar a la siguiente tarea comprobamos que funciona (NOTA. Aquí tuve que cambiar el valor *implementation 'com.google.android.material:material:1.8.0'* en el fichero de configuración build.gradle, para que funcionase. También aprovecho para indicar que ya había montado un emulador Pixel 6 Pro en la API 28.



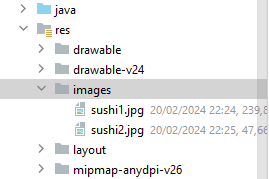
En este punto me he permitido añadir dos botones para realizar todas las partes de la practica en una sola aplicación, de manera que se pueda saltar de una a otra.



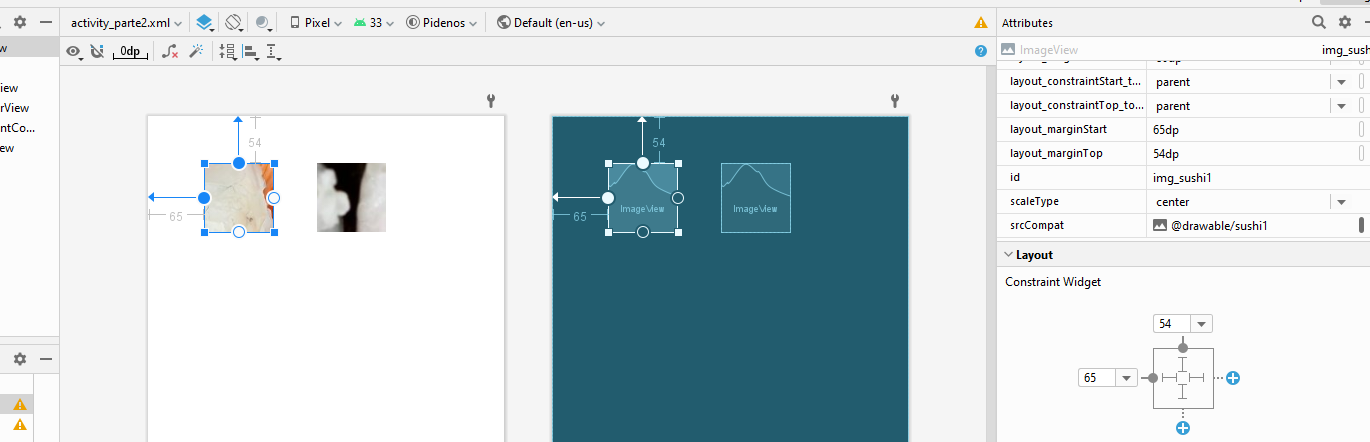
PARTE 2.

Repitiendo los primeros pasos de la Parte 1, creamos un nuevo activity, en este caso, llamado activity\_parte2.

En este caso me descargo dos imágenes de sushi de internet y las copio al proyecto, a la carpeta res/images (images la he creado yo mismo).

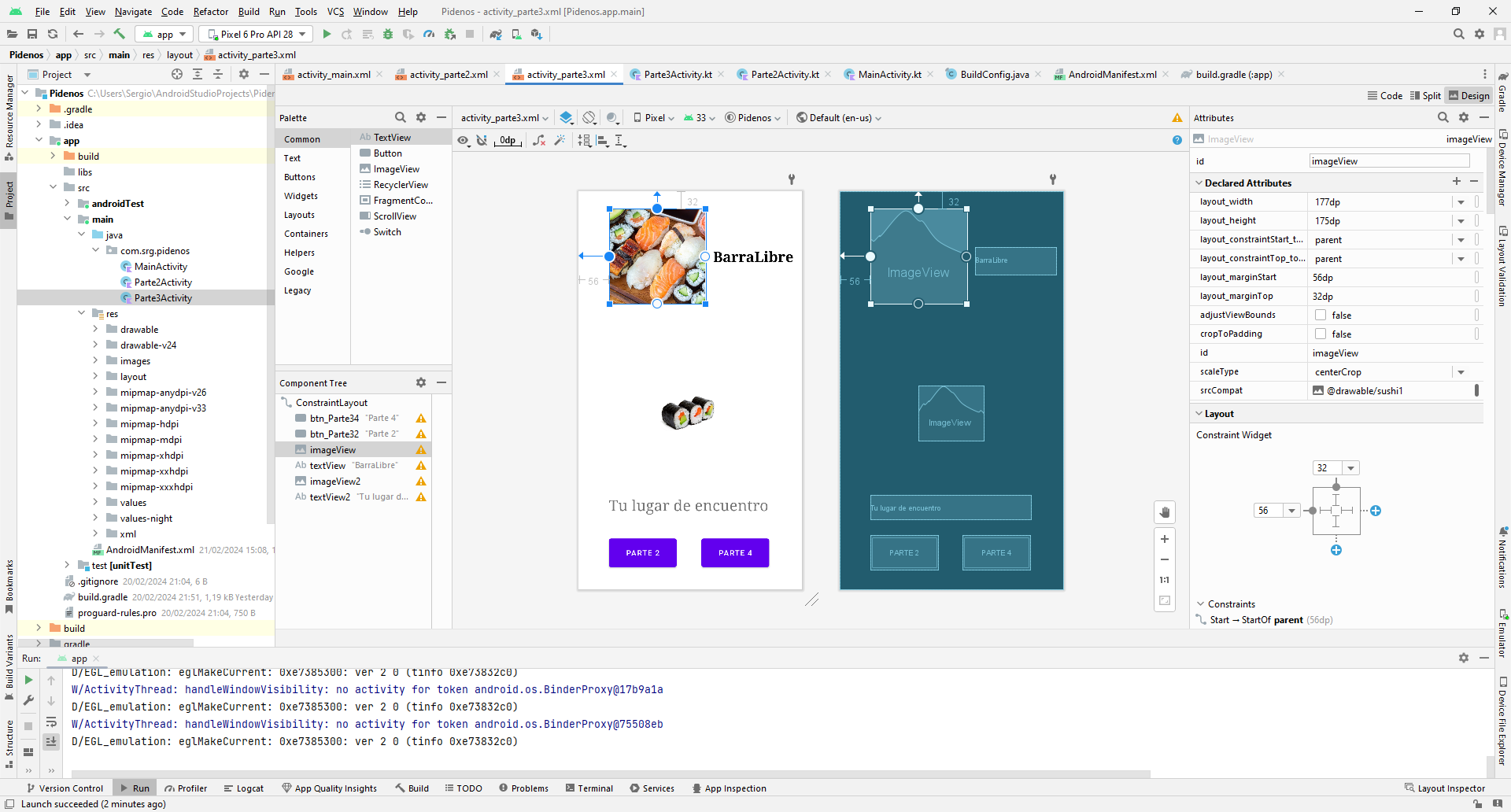


Insertamos un *imageView* y añadimos la imagen *sushi1.jpg*, luego establecemos el tamaño en la parte derecha, en *declared attributes*. Cambiamos el id por *img\_sushi1*. Por ultimo establecemos la opción *center* en el atributo *scaleType*, para que quede recortada y centrada.



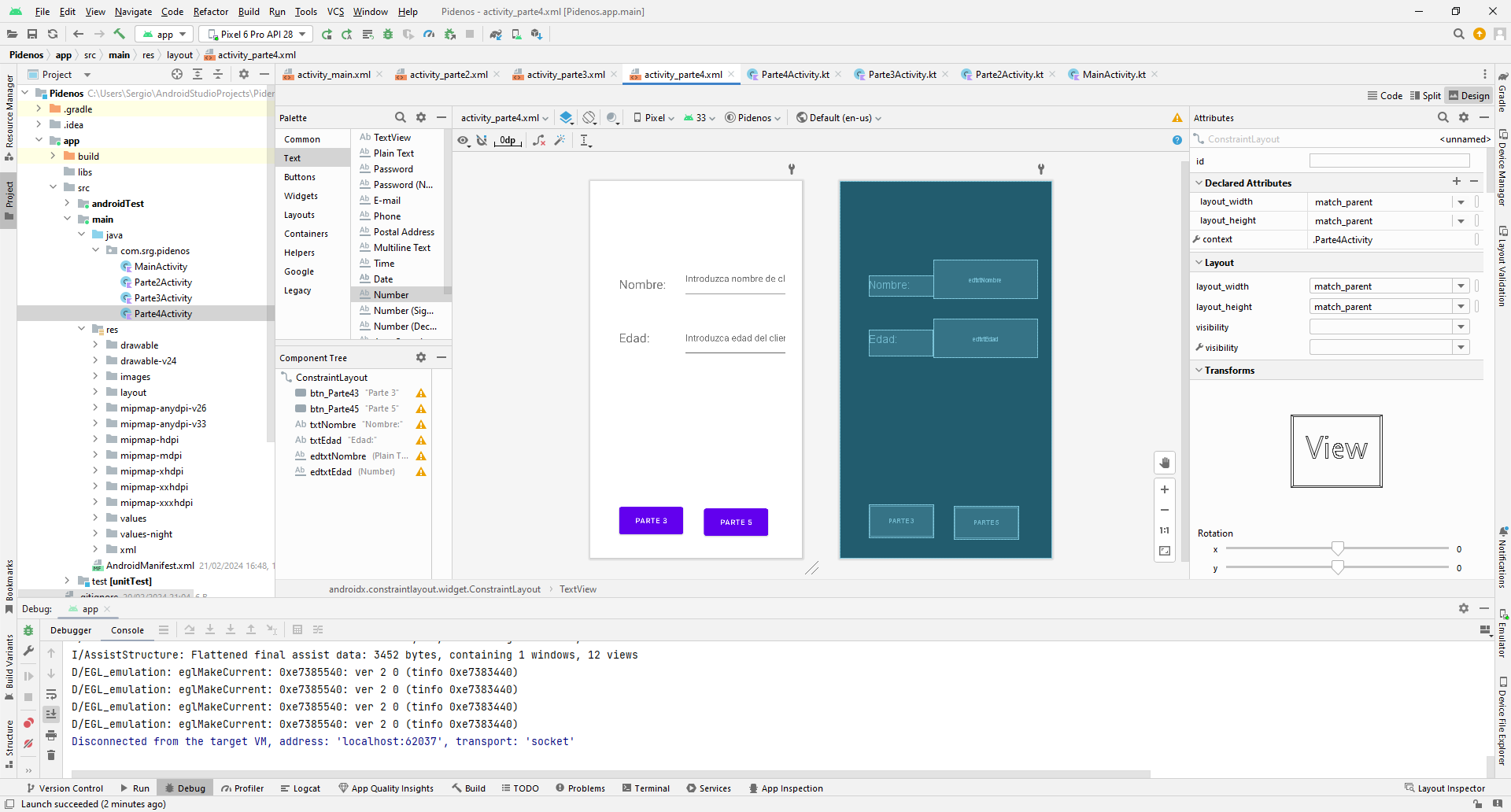
PARTE 3.

Para esta parte repetimos los pasos de la parte 2, salvo que en esta ocasión establecemos el atributo scaleType a centerCrop. También añadimos varios elementos TextView. En estos últimos modificamos los atributos textStyle/bold/true, textColor/black, autoSizeTextType/uniform, text, fontFamily/serif. Por ultimo ajustamos los constraints de todos los elementos para queden debidamente ubicados.

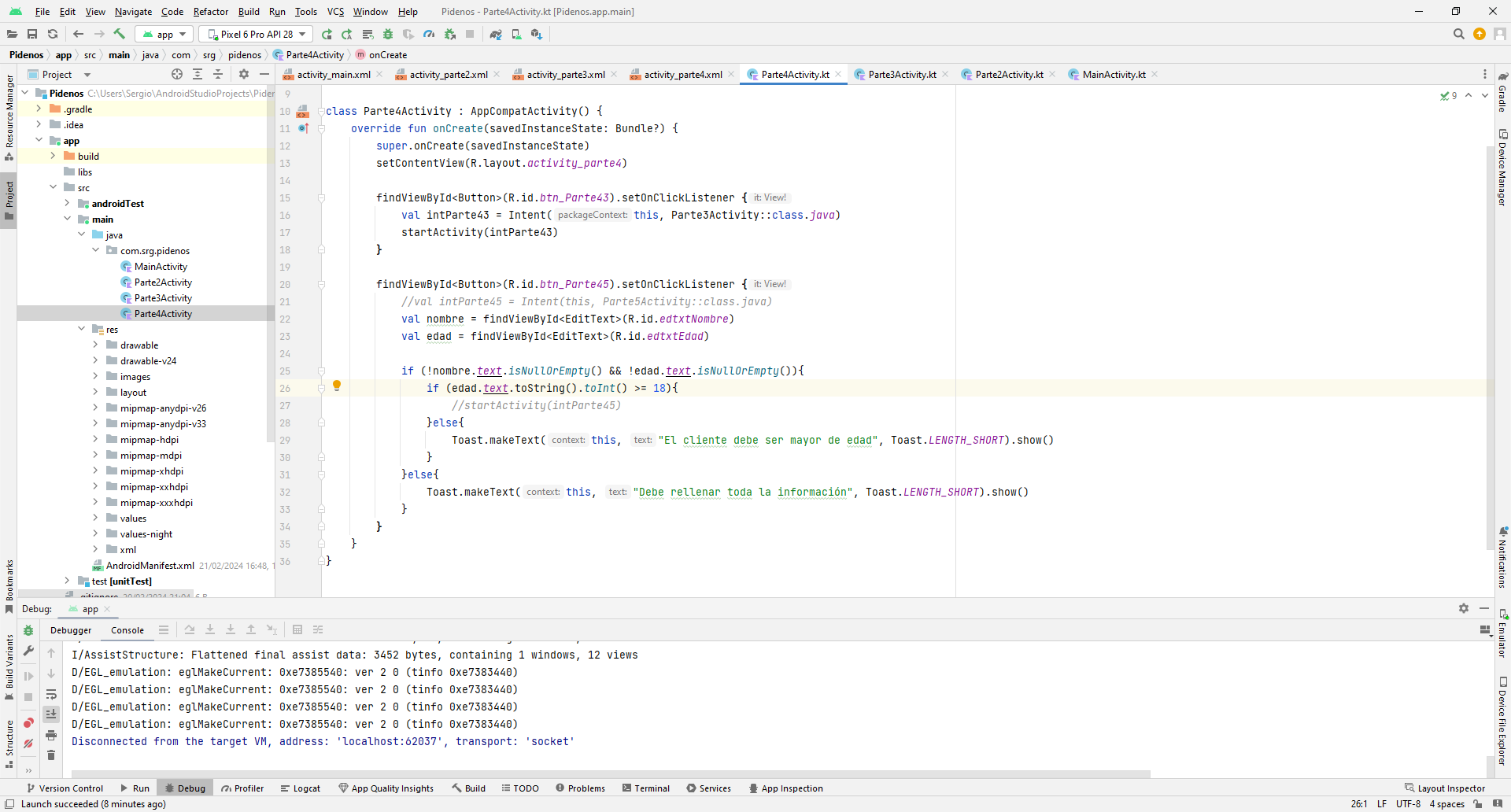


PARTE 4.

Metemos dos viewText igual que en los casos anteriores, y luego introducimos dos elementos de tipo Plain Text y Number, que será donde el usuario introduzca sus datos. Es requisito rellenar el atributo de Hint con la “pista” de la info a rellenar.

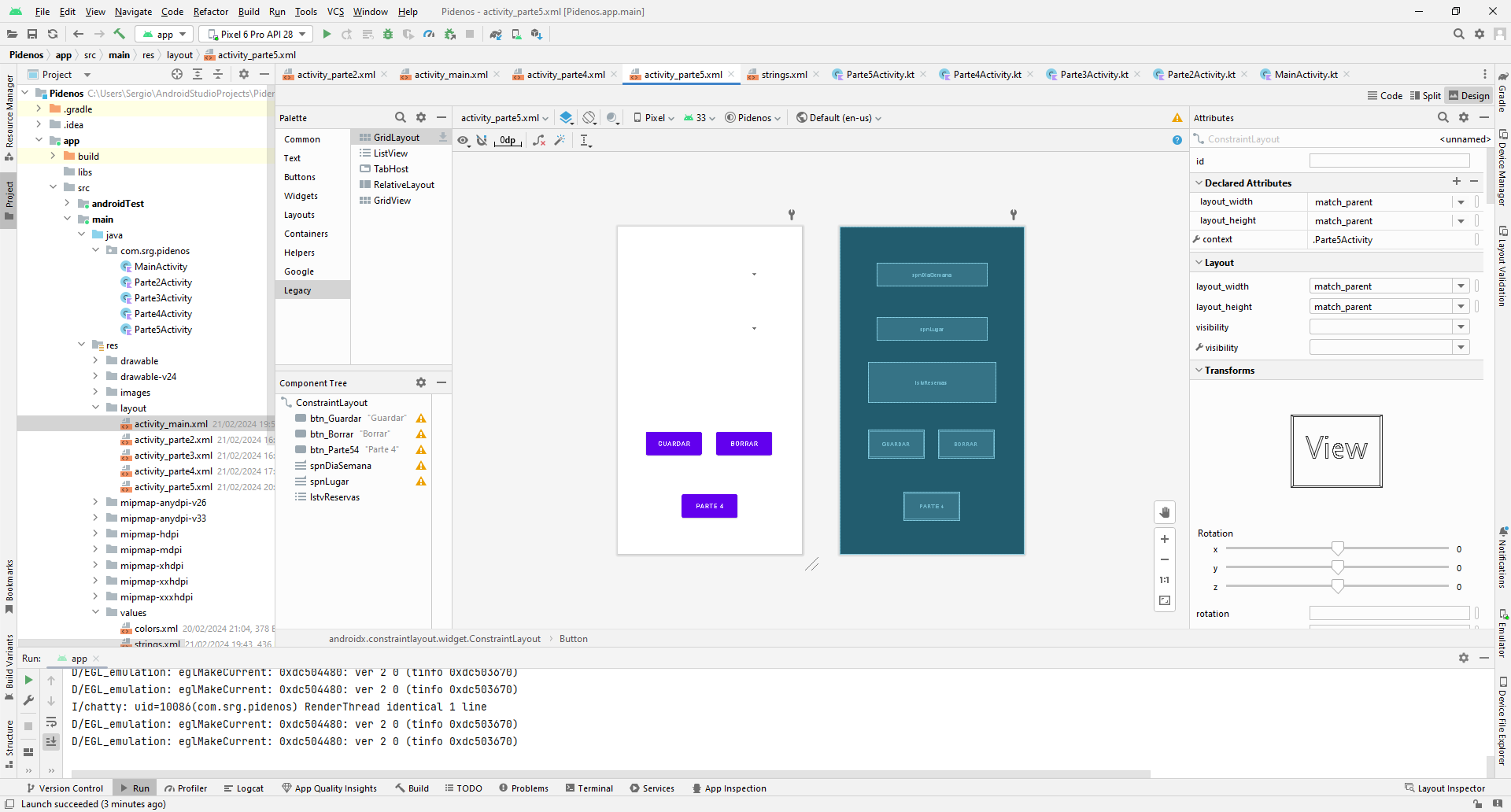


En la parte de código debemos instanciar los campos edit text y los botones. Metemos dos en este caso, uno para regresar a la parte anterior, y el otro para avanzar hasta la parte 5. Justo en este ultimo metemos las validaciones. En la primera validación se pide que se hayan rellenado todos los datos, de no ser asi saltara un mensaje de tipo Toast indicándolo. Si pasa la primera validación salta a la segunda donde metemos un parseo al edit text, primero para extraer la info introducida y luego para pasarla a tipo entero. (NOTA. Como ya pusimos que el campo para introducir la edad es de tipo number no aceptara otros caracteres, por lo que si no esta vacio es que es un numero, es decir, no deberían de colarse otros caracteres, ya que de ser asi podría saltar un error en tiempo de ejecución). Las líneas que validan el salto al siguiente activity están comentadas, ya que todavía no lo hemos creado.

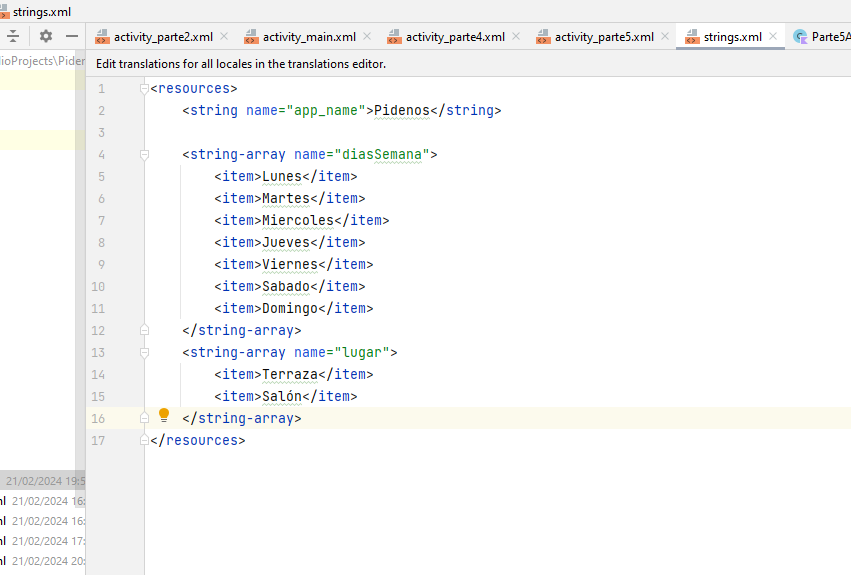


PARTE 5.

Creamos la nueva activity y añadimos dos spinner del bloque de Containers, en la paleta. Aunque no forma parte de la practica me he permitido añadir un ListView para guardar todas las reservas que se realicen, a parte de dos botones, uno de guardar y otro de borrar. Por ultimo, el botón de regresar a la parte 4.



En la parte de código he rellenado en dos resources la lista de días de la semana y los lugares de la reserva.



En la parte de código, cabe destacar que he creado un adaptador para configurar la propiedad adapter de cada spinner y una función común al botón de Guardar y Borrar para refrescar el ListView. Obteniendo el resultado final deseado.

Después de compilar y ejecutar en el simulador queda este aspecto.

