

Guía de ejercicios - Ambiente de desarrollo y sus elementos de configuración



¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

La siguiente guía de estudio tiene como objetivo practicar y ejercitar los contenidos que hemos visto en clase.

Tabla de contenidos

Actividad guiada: Pantalla con texto e imagen	2
Preguntas de proceso	9
¡Manos a la obra! - Personaliza tu icono	9
¡Manos a la obra! - Personaliza tu app	10
Preguntas de cierre	10
Referencias bibliográficas	10
Solución	11
¡Manos a la obra! - Personaliza tu icono	11
¡Manos a la obra! - Personaliza tu app	12



¡Comencemos!



Actividad guiada: Pantalla con texto e imagen

Vamos a crear una pantalla que tenga como encabezado un texto y una imagen centrada en la pantalla.

Para lograrlo, comenzaremos desde un proyecto creado usando la plantilla de Empty Activity, y seguiremos los siguientes pasos:

1. Abrir el archivo `activity_main.xml`.
 - a. Por defecto, el layout tiene un `ConstraintLayout` para ordenar nuestras vistas.

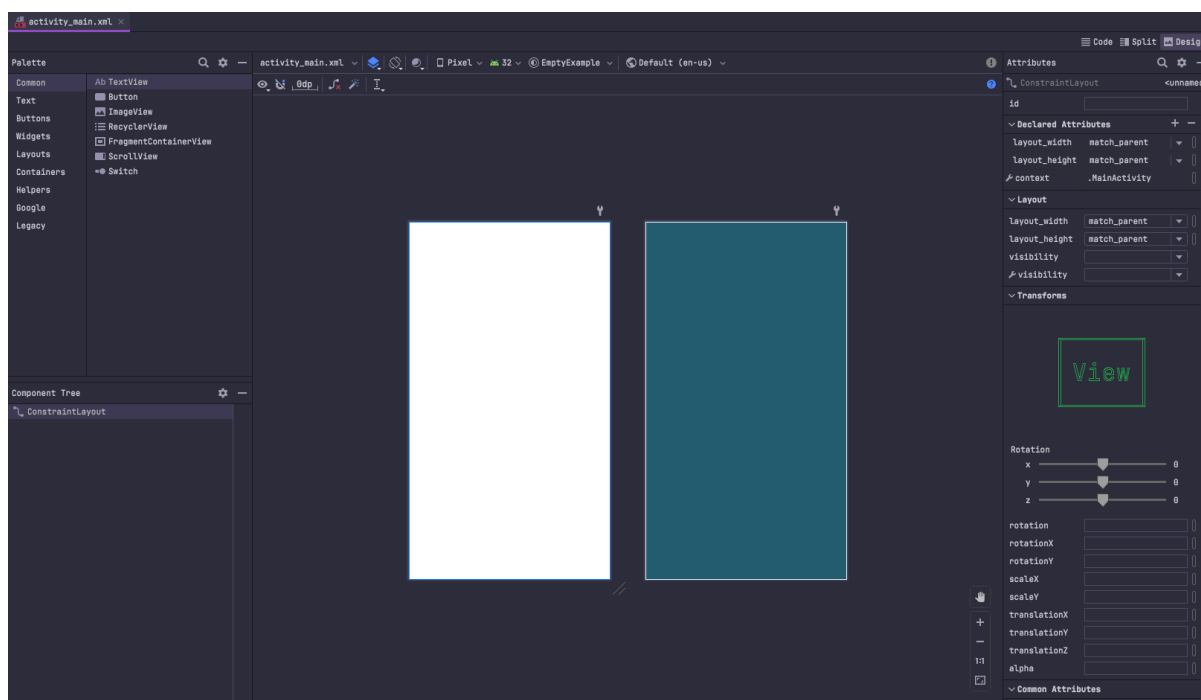


Imagen 01. Paso 1.
Fuente: ADL.

2. En la paleta de vistas (Palette) seleccionar desde Common -> TextView y arrastrar la vista hacia la pantalla de diseño.

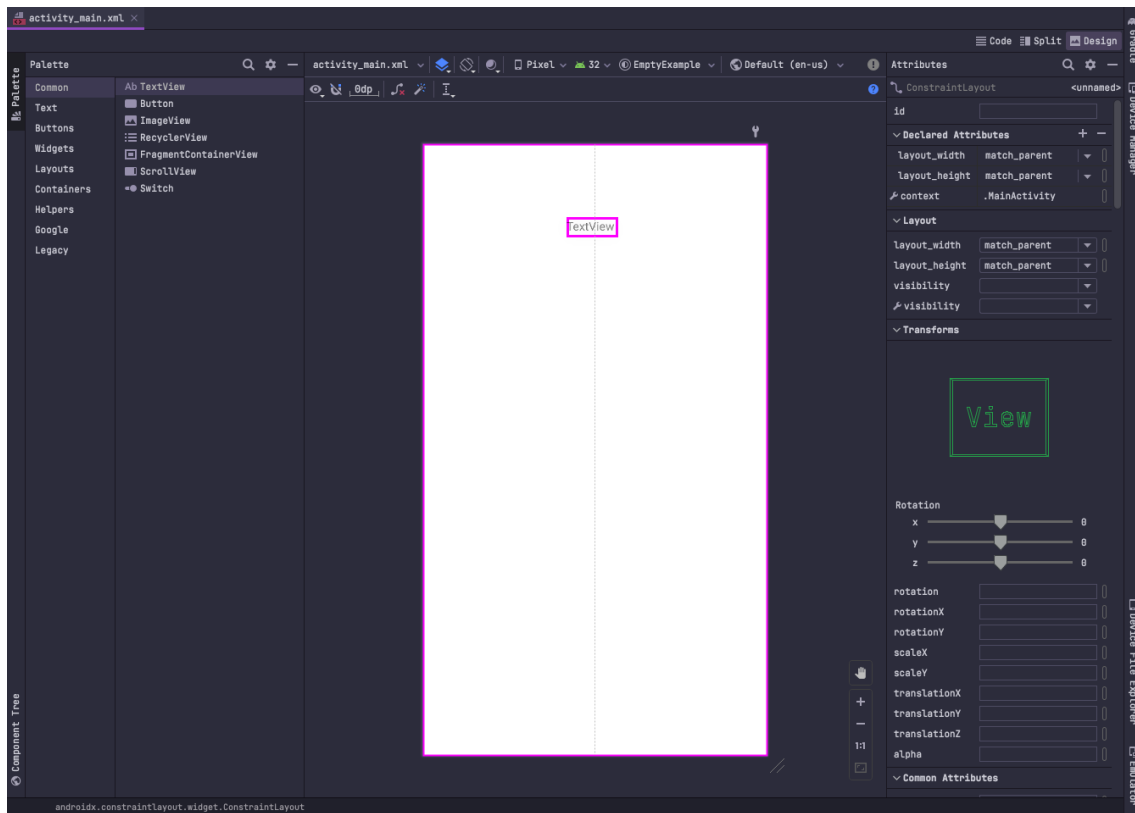
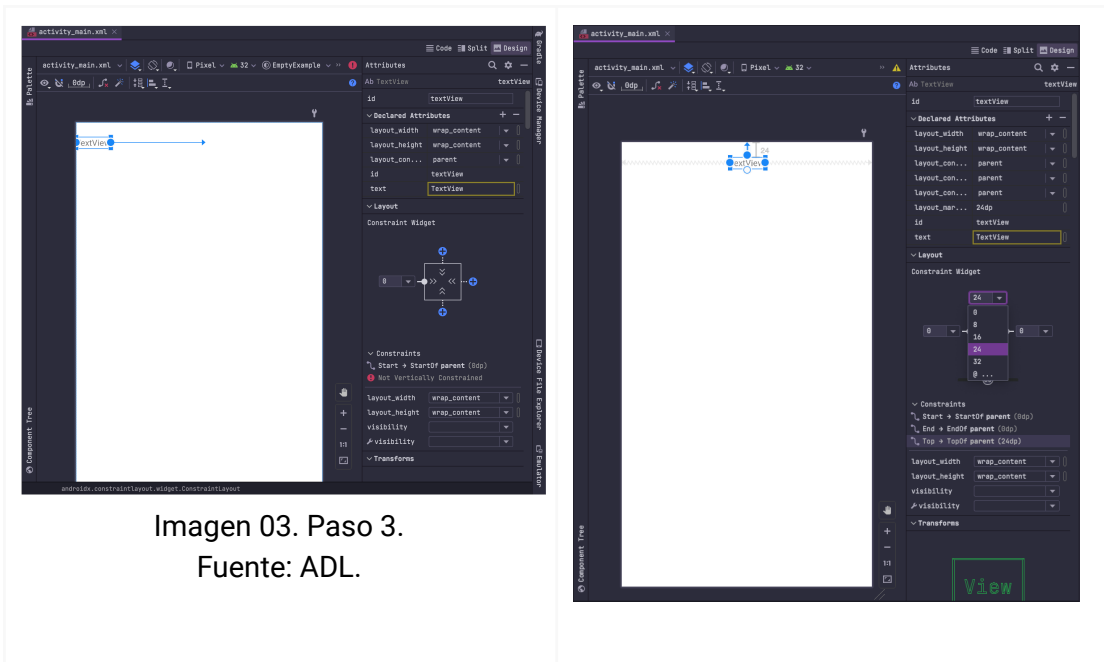


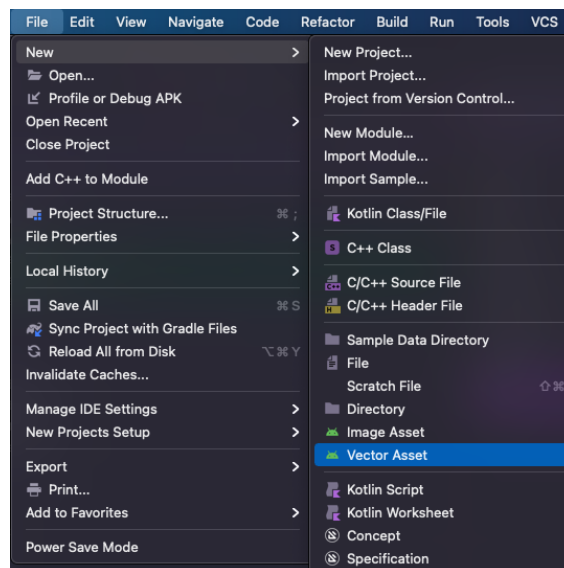
Imagen 02. Paso 2.

Fuente: ADL.

3. Vamos a ubicar el TextView centrado horizontalmente y a 32dp del borde superior.
 - a. Para esto debemos agregar las constraints al TextView.
 - b. Recuerda tomar desde uno de los 4 círculos según corresponda y crear una flecha hacia el extremo que desees.



4. Vamos a agregar un nuevo Vector Asset, seleccionando el menú File -> New -> Vector Asset.



5. Seleccionar desde los iconos de material design el icono castle, y luego finalizar la creación del asset.

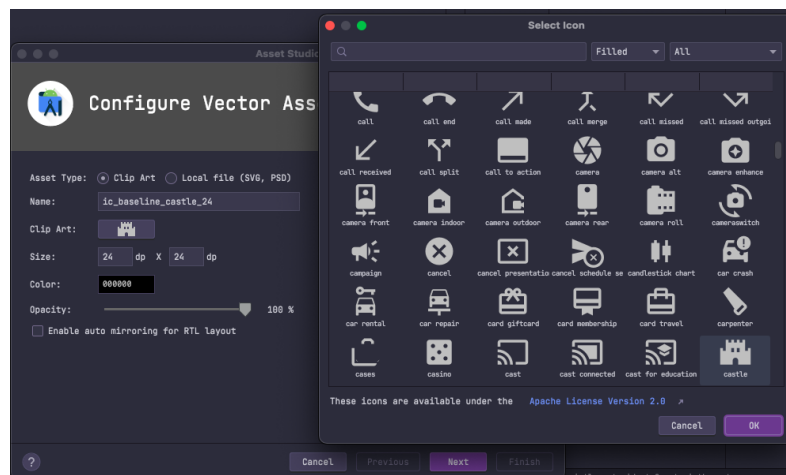


Imagen 05. Paso 5.
Fuente: ADL.

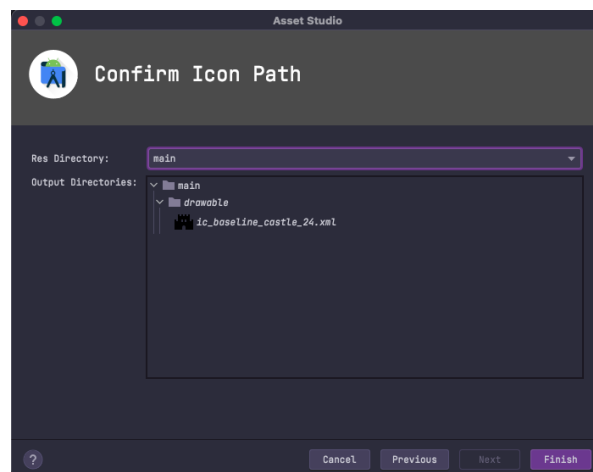


Imagen 06. Paso 5.
Fuente: ADL.

6. Agregar un ImageView al layout activity_main arrastrando desde la paleta, siguiendo la misma dinámica del paso 2.

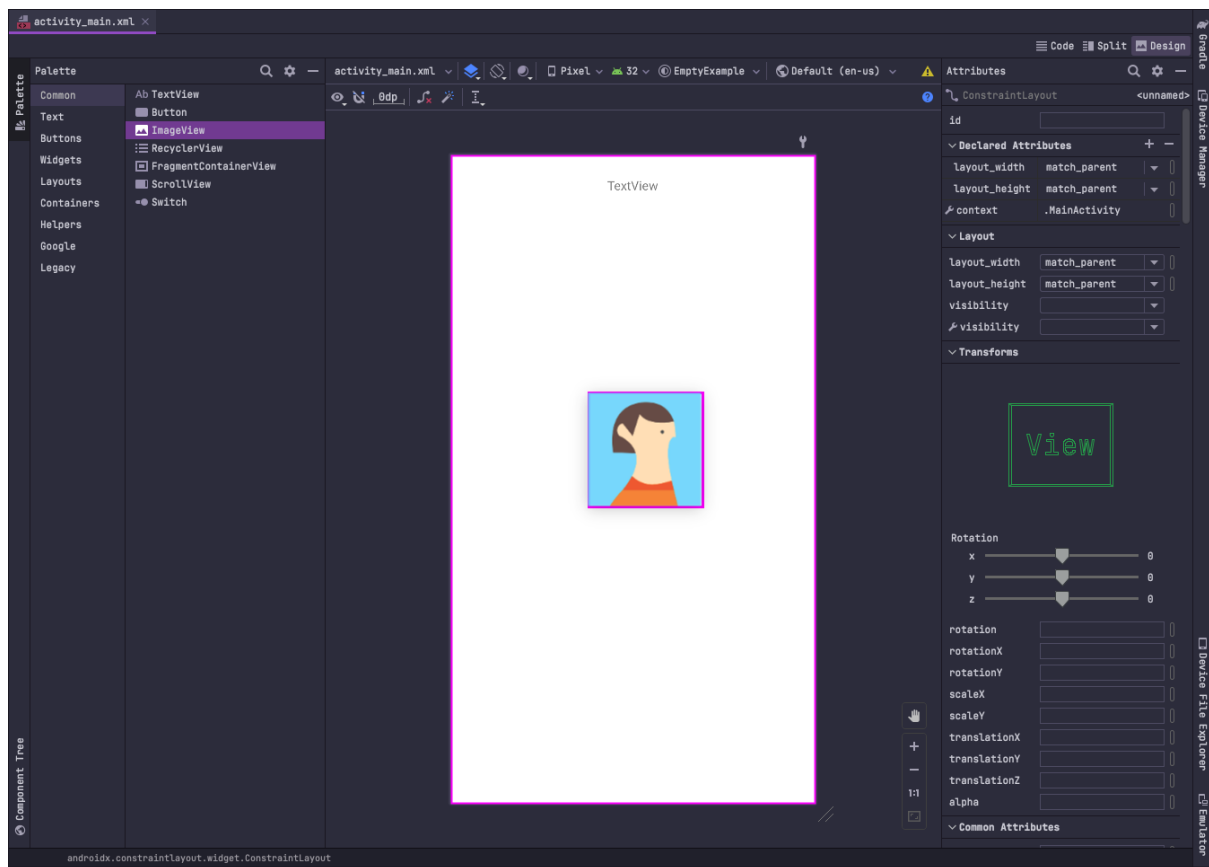


Imagen 07. Paso 6.
Fuente: ADL.

7. Seleccionar el recurso que vamos a utilizar, en nuestro caso, el castillo.

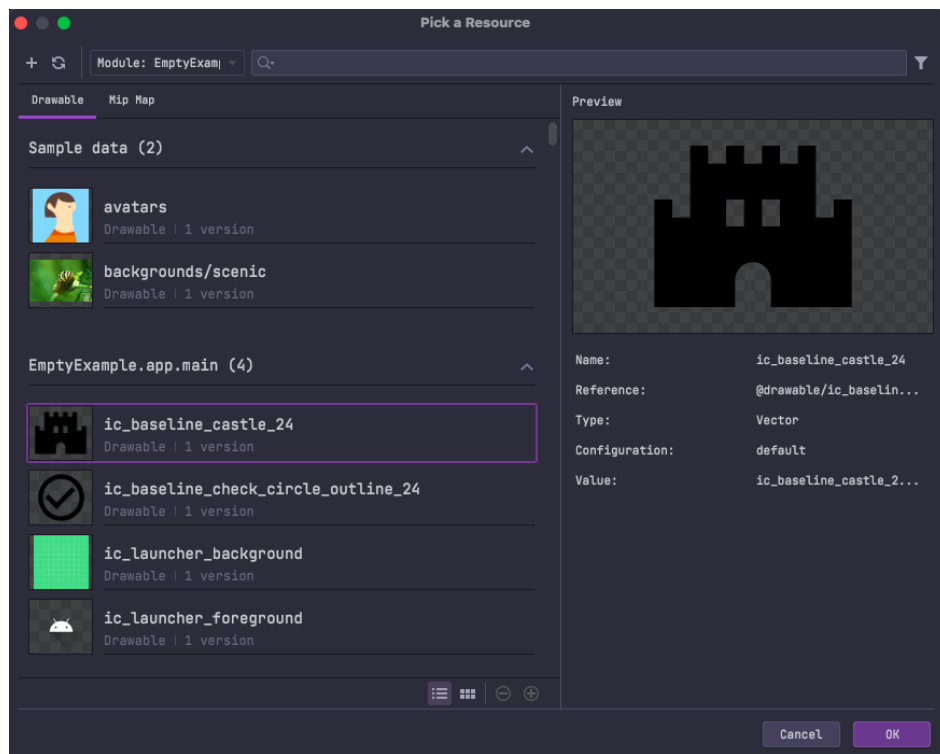


Imagen 08. Paso 7.

Fuente: ADL.

8. Agregar las constraints para que la imagen se ubique al centro de la pantalla, tanto horizontal como vertical.

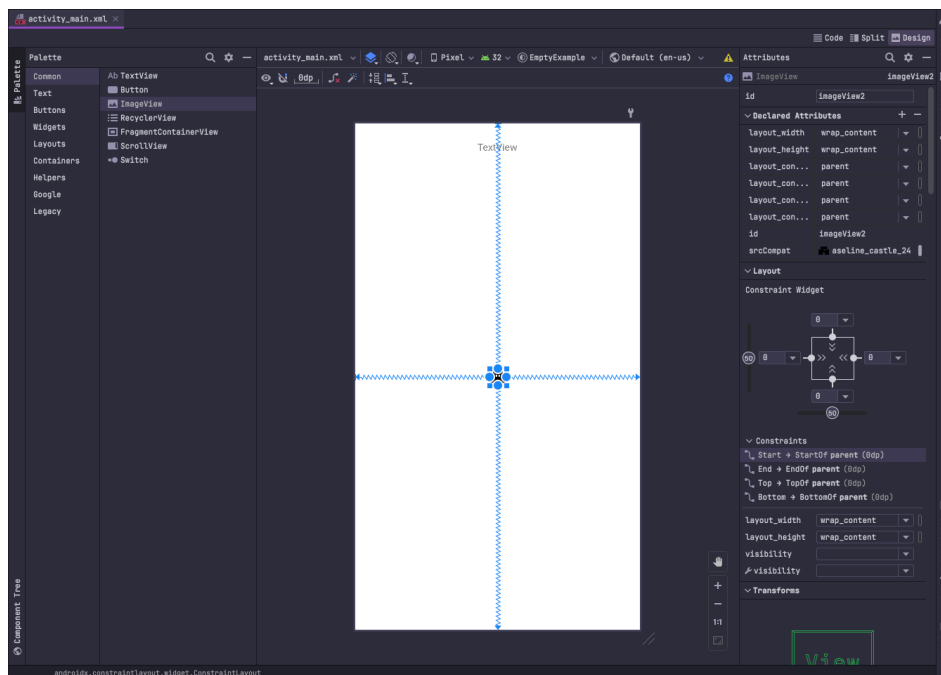


Imagen 09. Paso 8.

Fuente: ADL.

9. Cambiar el tamaño de la imagen a 300dp x 300dp.

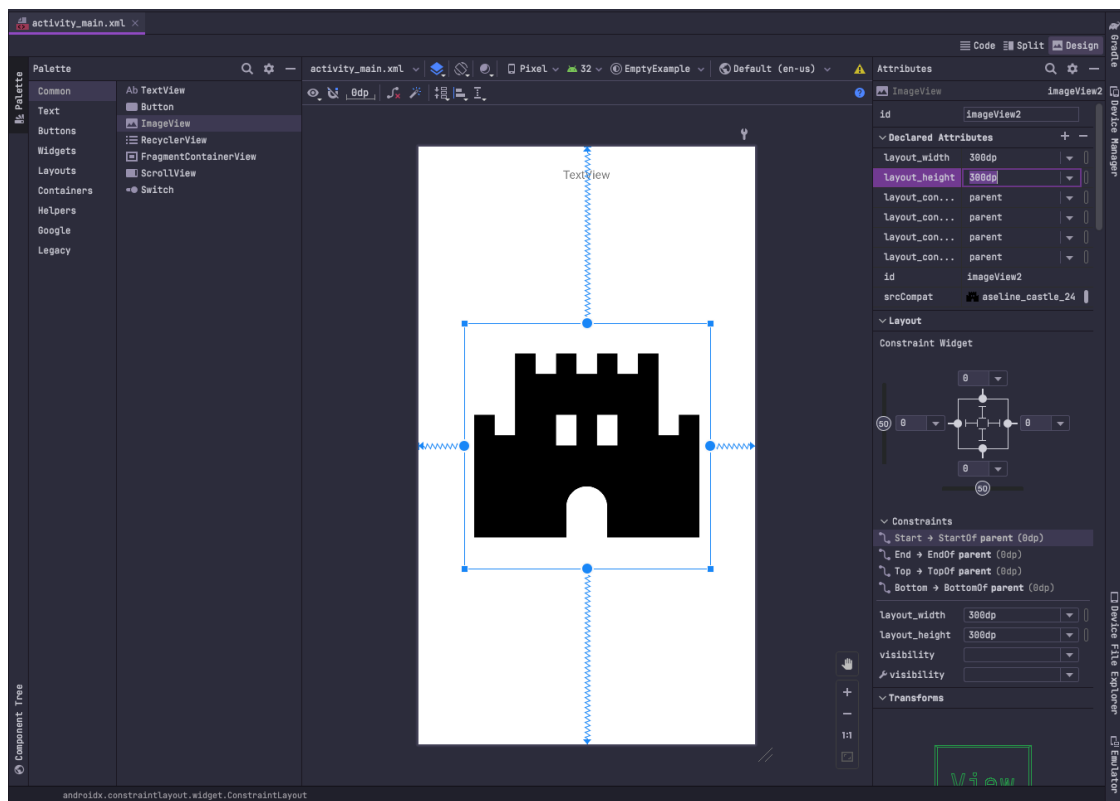


Imagen 10. Paso 9.
Fuente: ADL.

10. Probar en el dispositivo.

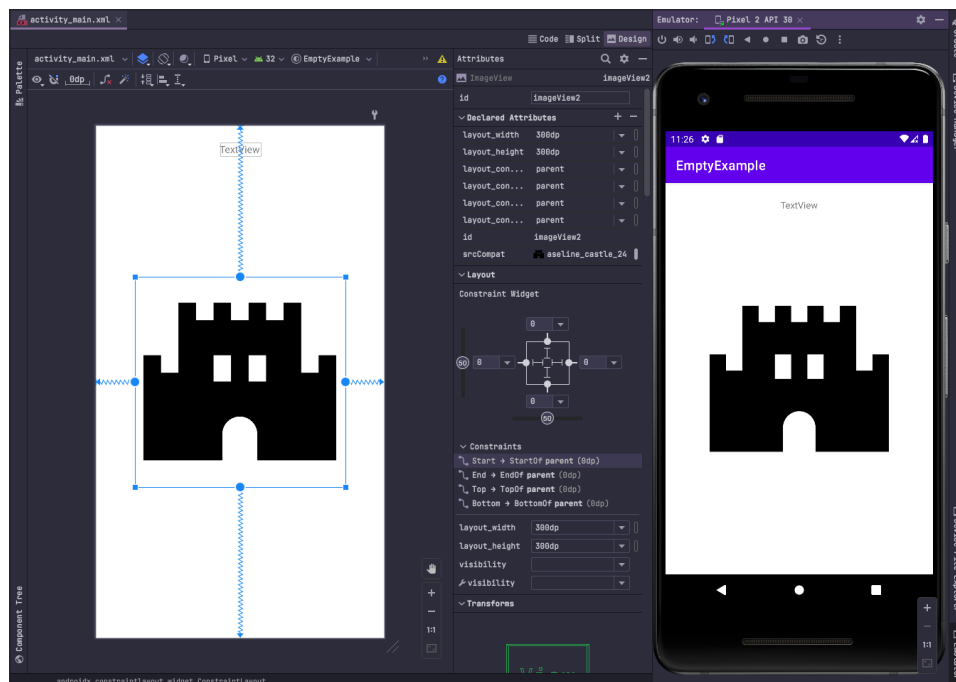


Imagen 11. Paso 10.
Fuente: ADL.

Preguntas de proceso

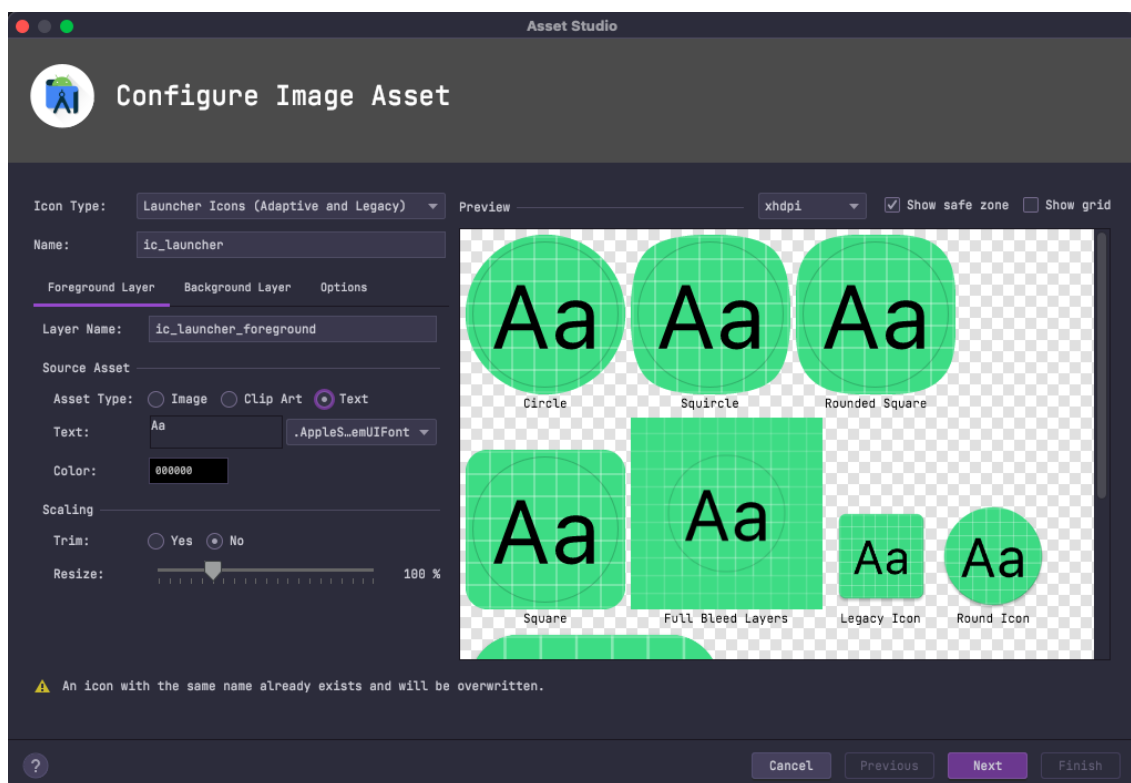
Reflexiona:

- ¿Son todos los tipos de assets iguales?
- ¿Cada elemento que veo en una pantalla es una “vista” (los botones, imágenes, textos, la barra superior)?
- En una aplicación de reseñas de películas (como imdb), donde se puede ver mucha información, ¿usarías un layout especial para dispositivos de pantalla grande, como una tablet?



¡Manos a la obra! - Personaliza tu icono

En la siguiente actividad, deberás personalizar el icono de la app, eligiendo desde los iconos de material design disponibles o escribiendo un texto. ¡Tú eliges!





¡Manos a la obra! - Personaliza tu app

A continuación deberás cambiar colores y textos de la app para darle el aspecto que más te guste. Recuerda que puedes modificar los archivos strings.xml, colors.xml, styles.xml, themes.xml para personalizar tus apps.

Preguntas de cierre



- ¿Cuál es la diferencia entre Image Asset y Vector Asset?
- ¿Qué unidad de medida utilizarías para agrandar la fuente dentro de un TextView?
- ¿Qué unidad de medida utilizarías para agrandar la imagen dentro de un ImageView?

Referencias bibliográficas

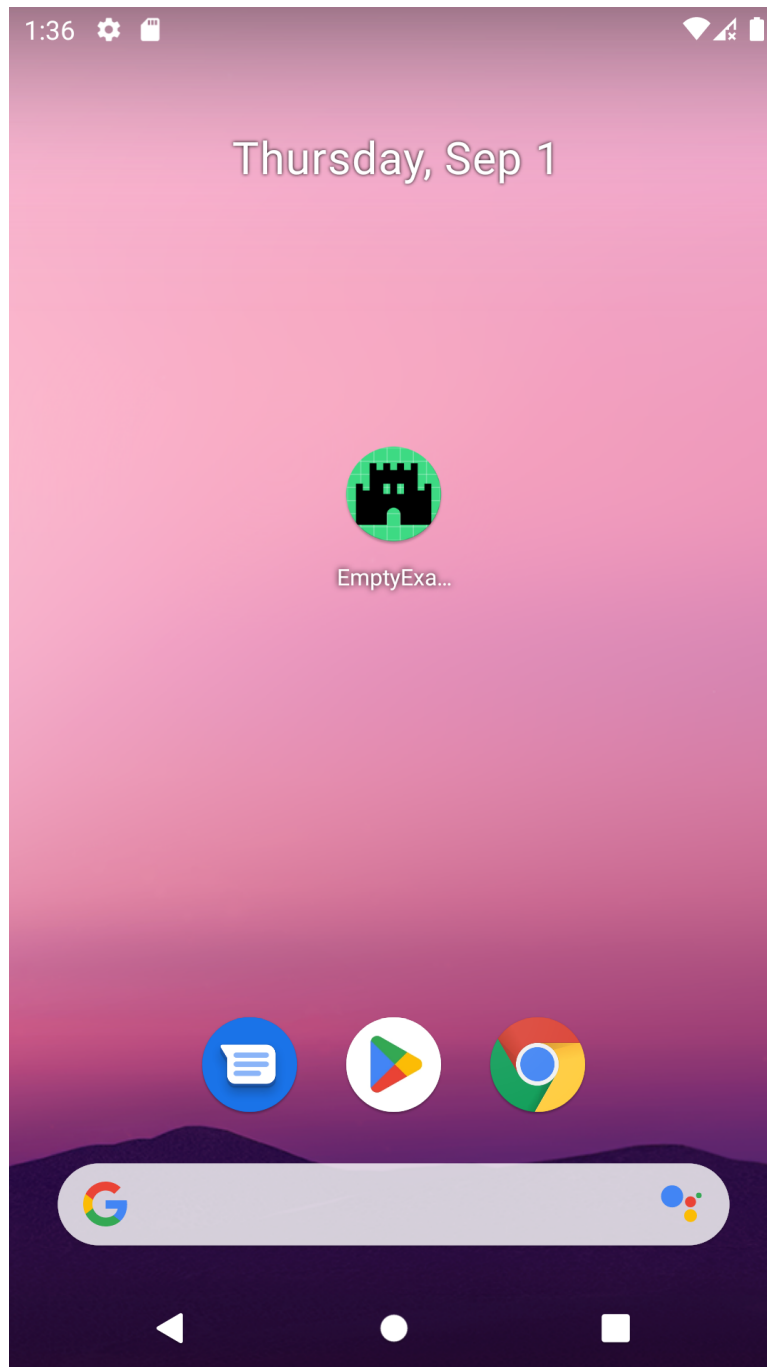


- *Cómo brindar compatibilidad con diferentes densidades de píxeles.* Visitado el 01 de septiembre de 2022. <https://developer.android.com/training/multiscreen/screensizes>
- *Material Design icons Icons.* Visitado el 03 de enero de 2024. material.io/styles/icons

Solución

¡Manos a la obra! - Personaliza tu icono

Al crear un nuevo icono desde Image Asset, por defecto sobrescribe el icono de la app llamado ic_launcher. Una forma de ver que efectivamente se ha cambiado el icono de la app es instalarla y ver el icono utilizado. Por ejemplo:



¡Manos a la obra! - Personaliza tu app

Por ejemplo, luego de modificar los atributos del TextView:

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/margin_top"
    android:text="@string/welcome"
    android:textColor="@color/teal_700"
    android:textSize="@dimen/header_text_size"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Se agregan a dimens.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="header_text_size">24sp</dimen>
    <dimen name="margin_top">24dp</dimen>
</resources>
```

El resultado sería:

