

**Coordinación de Ingeniería en**

**Sistemas Computacionales**



**Materia: Programación Móvil I**

**Nombre Informe:** funciona la distribución de los elementos en un ConstraintLayout. Informe Guiado

**Alumno(s): María Guadalupe Ramírez González**

**Fecha: 09/09/2020**

**Objetivo**

Identificar los aspectos de la creación de nuevos programas para aplicaciones móviles, utilizando todos los elementos de herramientas de diseño gráfico.

# Temas del plan de estudios

TEMA 3: Android Studio

3.5Creación y estructura de un proyecto Android.

3.5.1 Probar el proyecto sobre un dispositivo virtual.

3.5.2 Probar el proyecto sobre un dispositivo móvil.

# Material

1. Un computadora o laptop
2. El Andriod Studio
3. Conocimiento del tema

# Marco Teórico

ConstraintLayout te permite crear diseños grandes y complejos con una jerarquía de vistas plana (sin grupos de vistas anidadas). Es similar a RelativeLayout en cuanto a que se presentan todas las vistas de acuerdo con las relaciones entre las vistas del mismo nivel y el diseño de nivel superior, pero es más flexible que RelativeLayout y más fácil de usar con el editor de diseño de Android Studio.

Toda la potencia de ConstraintLayout está disponible directamente desde las herramientas visuales del editor de diseño, ya que la API de diseño y el editor de diseño se crearon específicamente para funcionar en conjunto. Así, puedes crear todo tu diseño con ConstraintLayout arrastrando y soltando elementos, en lugar de editar el XML.

**Como agregar un ConstraintLayout**

1. Asegúrate de tener el repositorio maven.google.com declarado en el archivo build.gradle de nivel de módulo:

repositories {

google() }

1. Agrega la biblioteca como una dependencia en el mismo archivo build.gradle, como se muestra en el siguiente ejemplo. Ten en cuenta que la última versión puede ser diferente de la que se muestra en el ejemplo:

dependencies {

implementation "androidx.constraintlayout:constraintlayout:2.0.0-rc1"

}

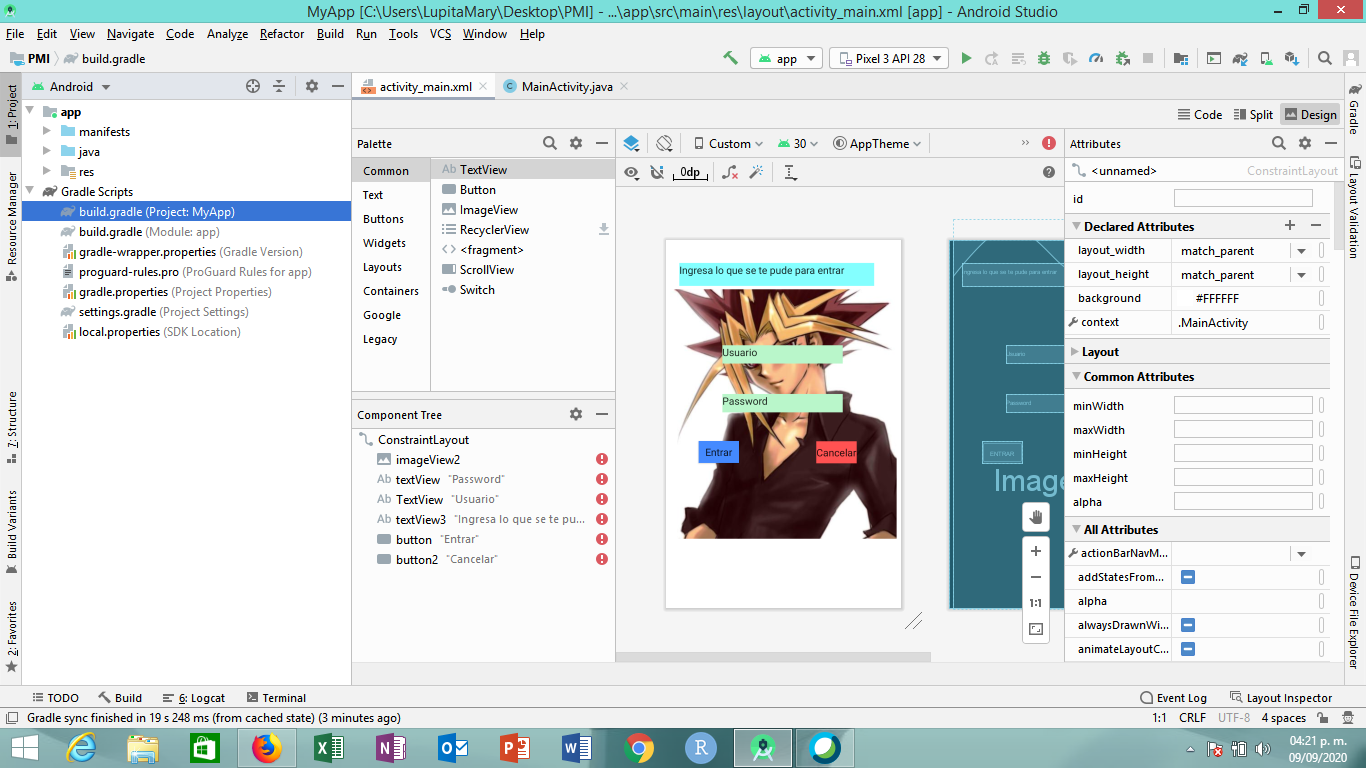
1. En la barra de herramientas o notificación de sincronización, haz clic en Sync Project with Gradle Files.

Ya está todo listo para crear el diseño con ConstraintLayout.

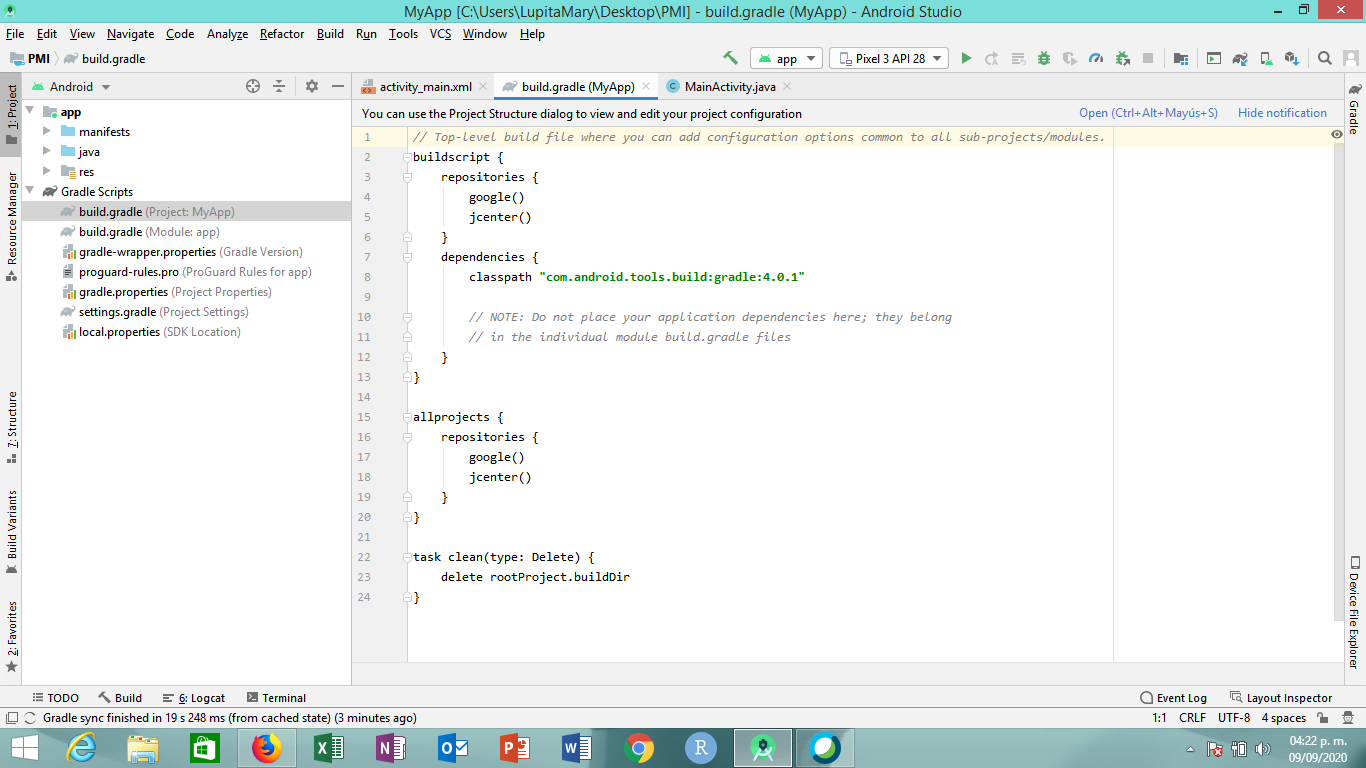
Cómo crear un diseño nuevo

1. Para iniciar un archivo de diseño con restricciones nuevo, sigue estos pasos:
2. En la ventana Project, haz clic en la carpeta del módulo y, luego, selecciona File > New > XML > Layout XML.
3. Ingresa un nombre para el archivo de diseño y "androidx.constraintlayout.widget.ConstraintLayout" en Root Tag.
4. Haz clic en Finish.

# Desarrollo



Para poder agregar o ver si tenemos ConstraintLayou nos tenemos que ir a Gradle Scripts, tenemos que desplegar el menú y seleccionamos la clase buid.gradle.



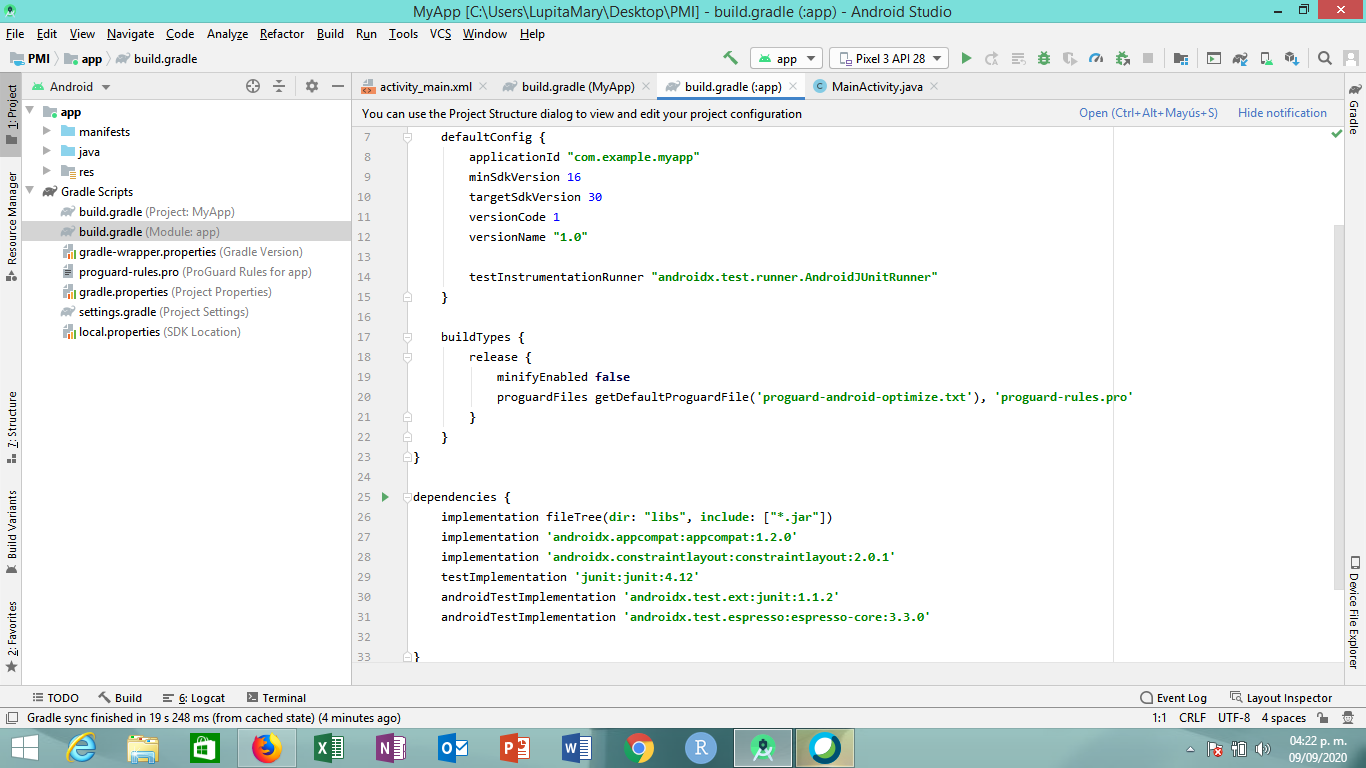
En esta parte nos muestra el código de los repositorios si tiene declarado el método de repositories y dentro del google() está bien podemos empezar a diseñar nuestra aplicación.

Pero si no lo tiene se tiene que poner lo siguiente:

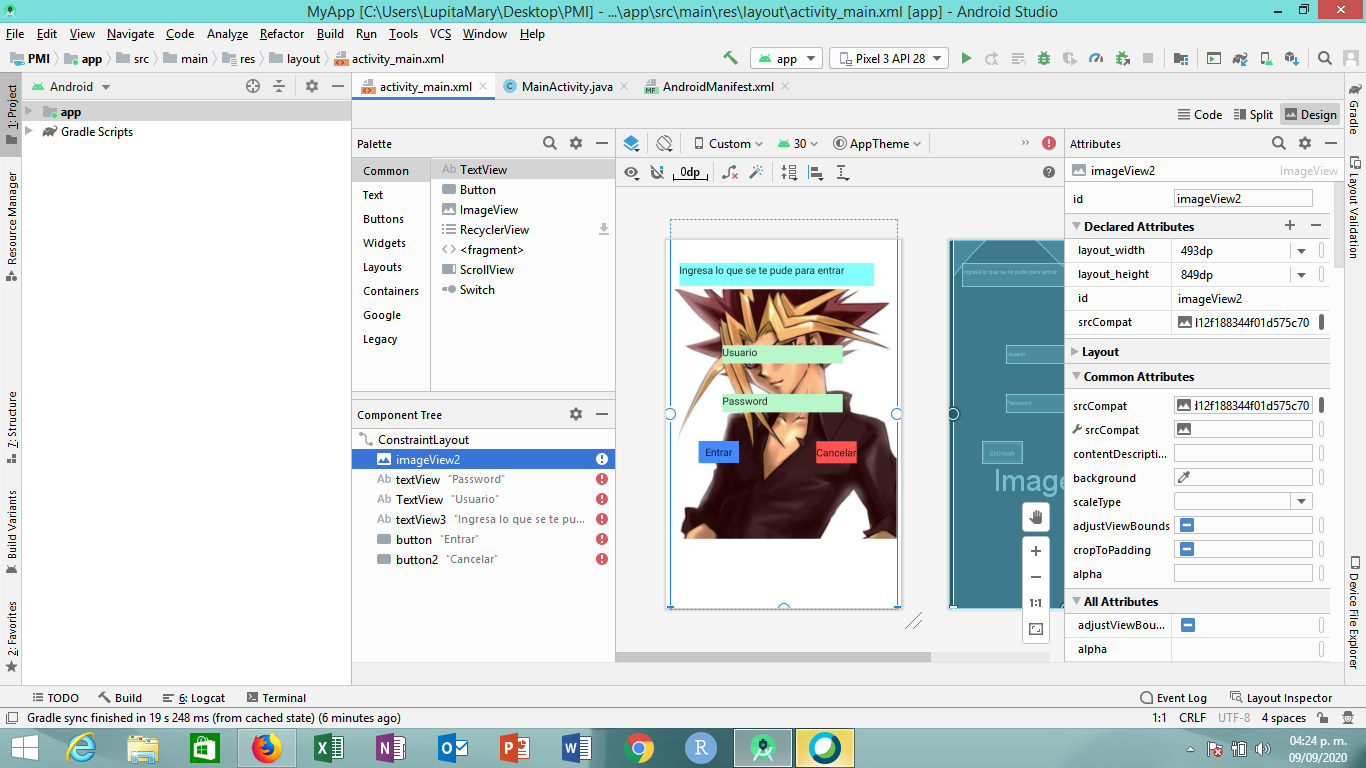
**repositories {**

**google()**

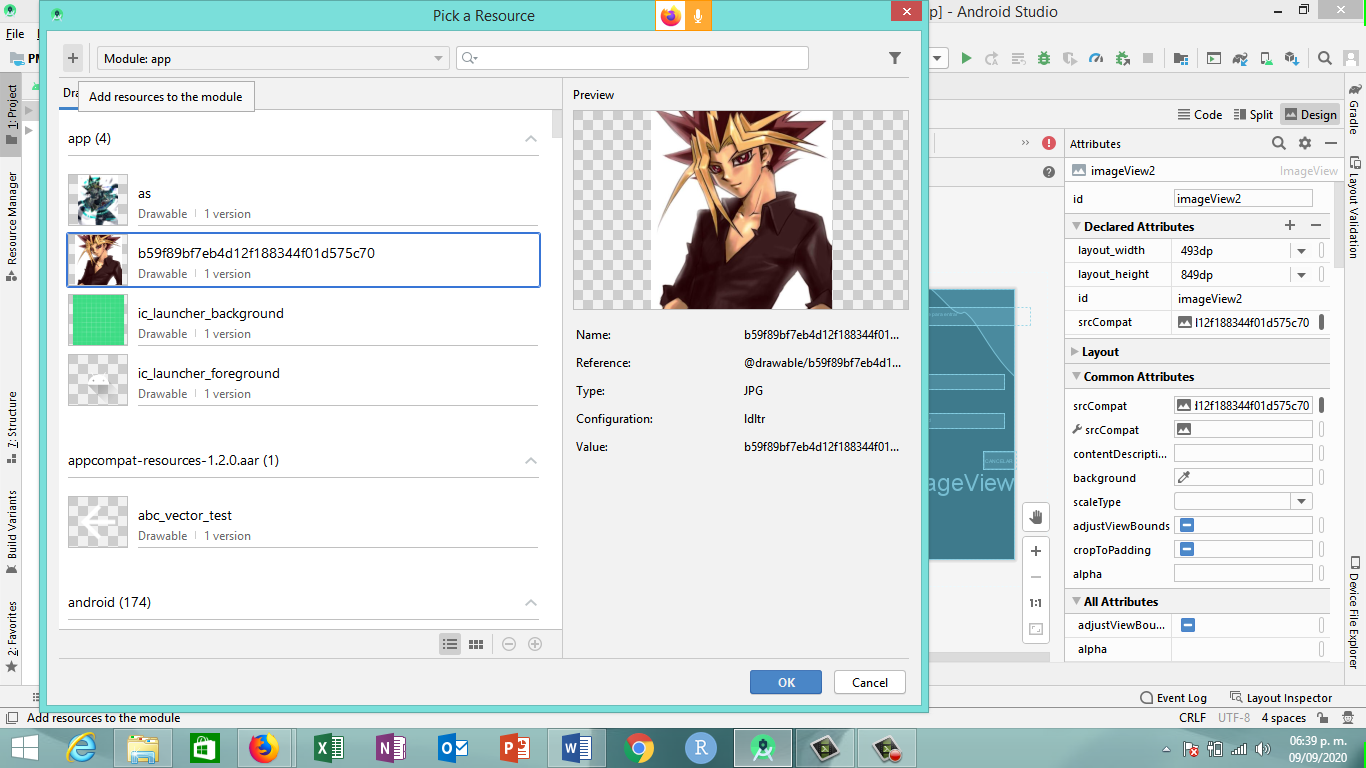
**}**



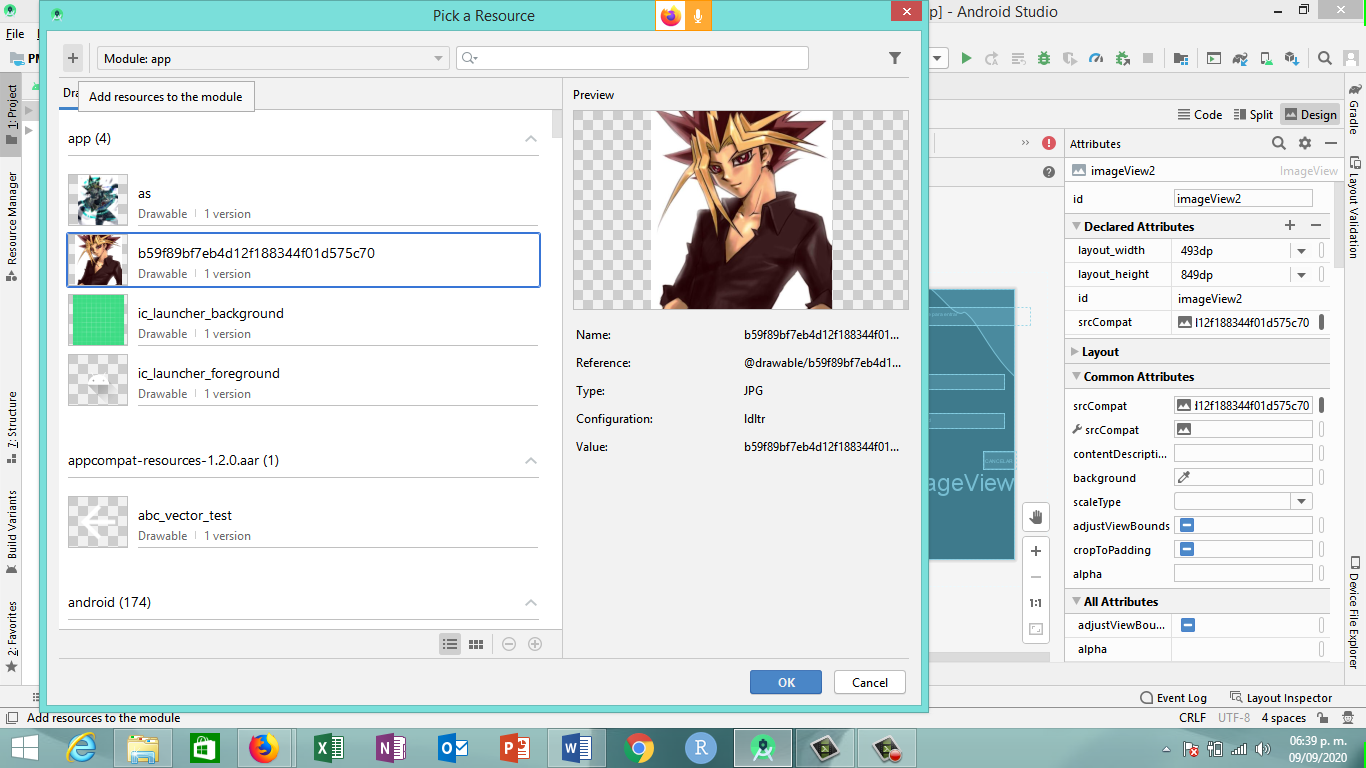
En el segundo buid.gradle tenemos el siguiente código en donde podemos ver que está implementando el androidx.constraintlayout.ConstraintLayout/2.0.1. con esto podemos ver que si está declarado e implementado bien el ConstraintLayout.



En esta parte ya tenemos el proyecto con las etiquetas, es decir en este proyecto de pantalla tenemos los siguientes elementos que son: imageViem, TestViem´s y Button´s.

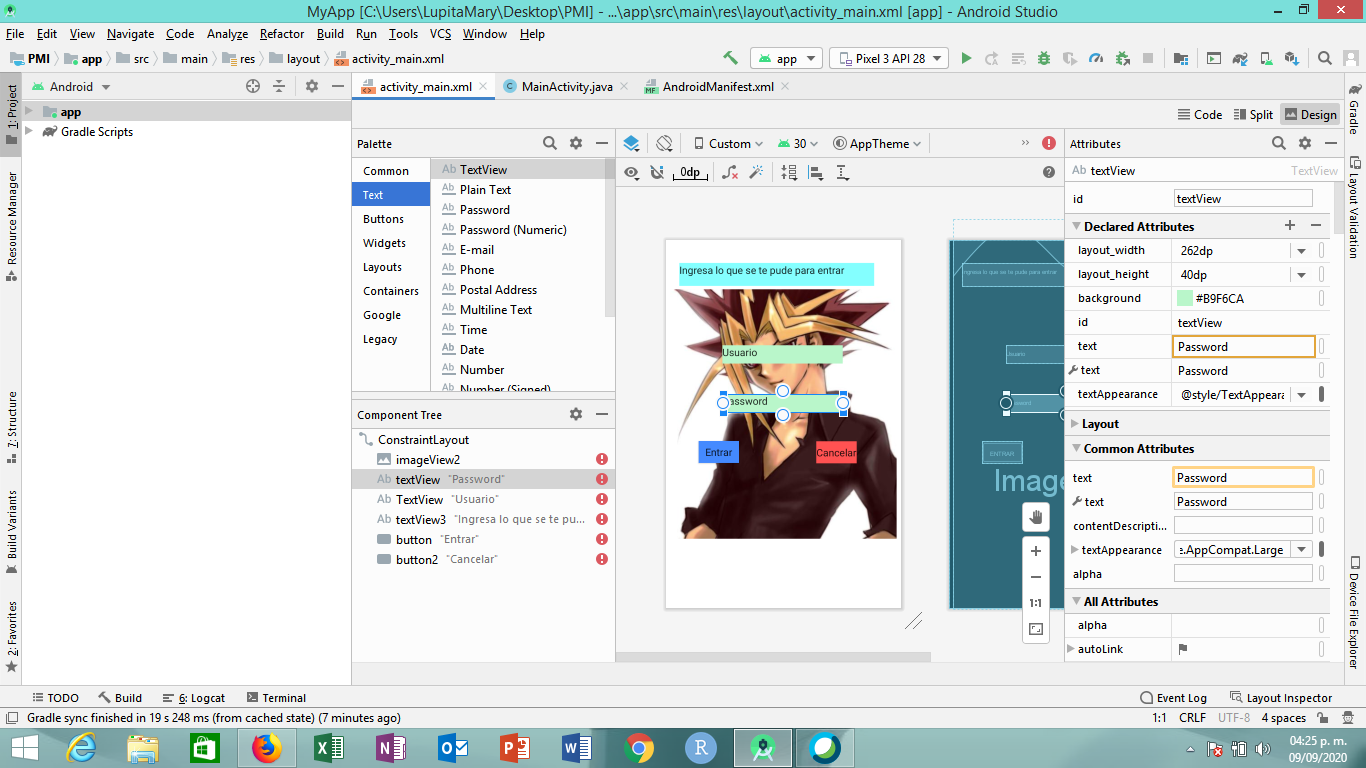


En la imageViem lo que hicimos es importa una imagen cualquiera para eso témenos que irnos a la parte de los atributos srcCompat y después nos aparece esta pestaña en donde la parte superior se encuentra un botón con el símbolo de + y de ahí lo que tiene que hacer es ver dónde está la imagen que queremos importar y seleccionarla para que aparezca como fondo de la aplicación.



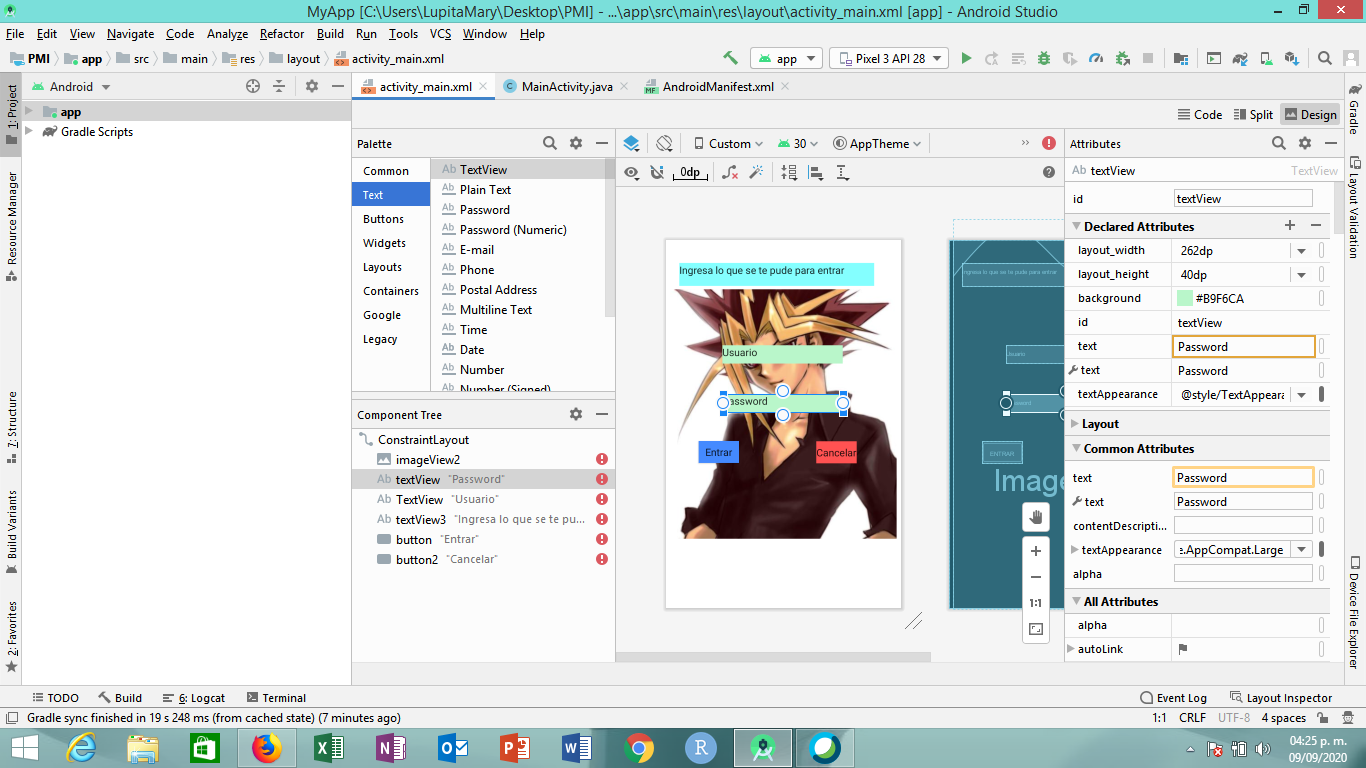
Luego de ahí podemos cambiar su tamaño y posición con los elementos de layout\_widht y height.

Esto también podemos hacerlo con los testViem y los button.



Pero para eso lo que hacemos en los testViem fue cambisr el texto y el color, para esto lo encontramos en backgruop para cambiar el color, text para agregar un texto, textAppearance="@style/TextAppearance.AppCompat.Large" para poder cambiar el tamaño del texto. Esto es lo mismo que hacemos en los botones.

# Resultado



# Evaluación

# Conclusiones

Me pareció un trabajo interesante, por la razón de que la creación o la importación del ConstraintLayout. Y porque tuve que investigar mucho y me pareció interesante lo que cada definición que encontraba y que trabajos podía hacer con esta herramienta.

# Bibliografía

<https://developer.android.com/training/constraint-layout?hl=es-419>