Universidad Nacional

Escuela de Informática

EIF-411

 Diseño y programación de plataformas móviles

Reporte de clase

Profesor

Juan de Dios Murillo Morera

Estudiantes:

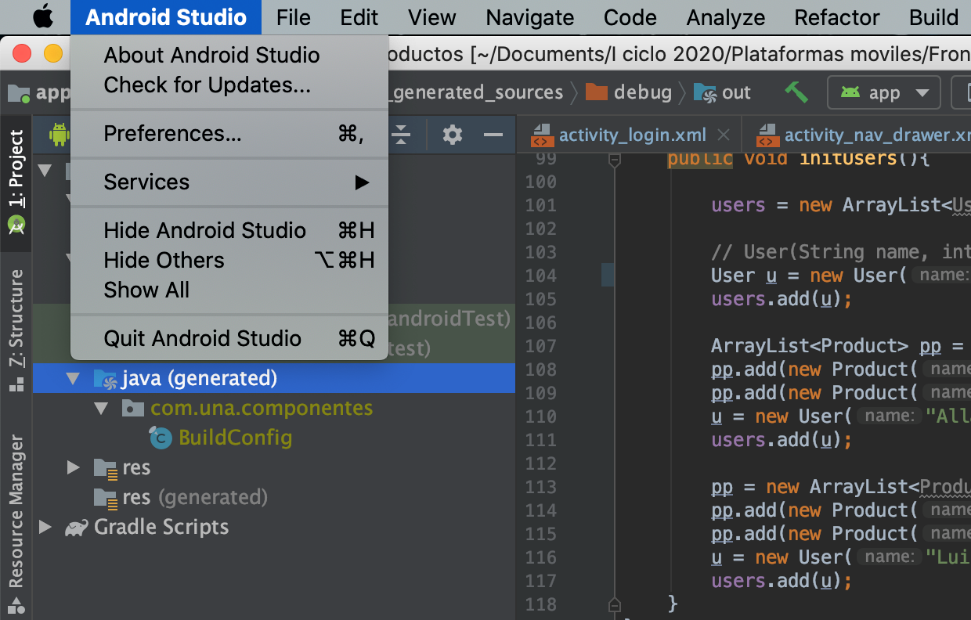
Djenane Hernández Rodríguez - 504210953

Diego Monterrey Benavides - 116810263

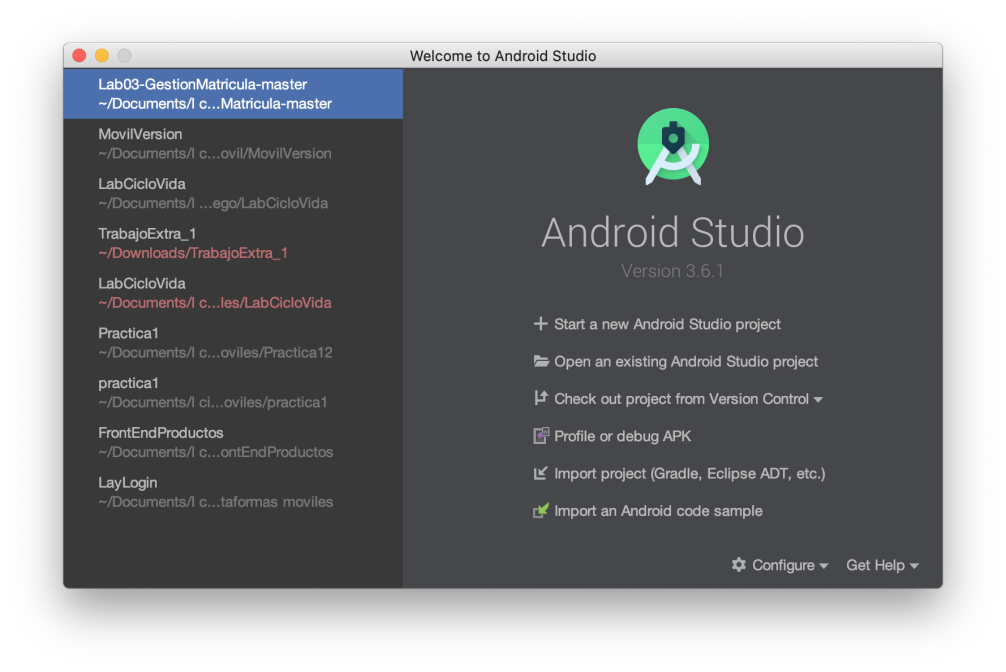
I ciclo 2020

***Reporte Técnico***

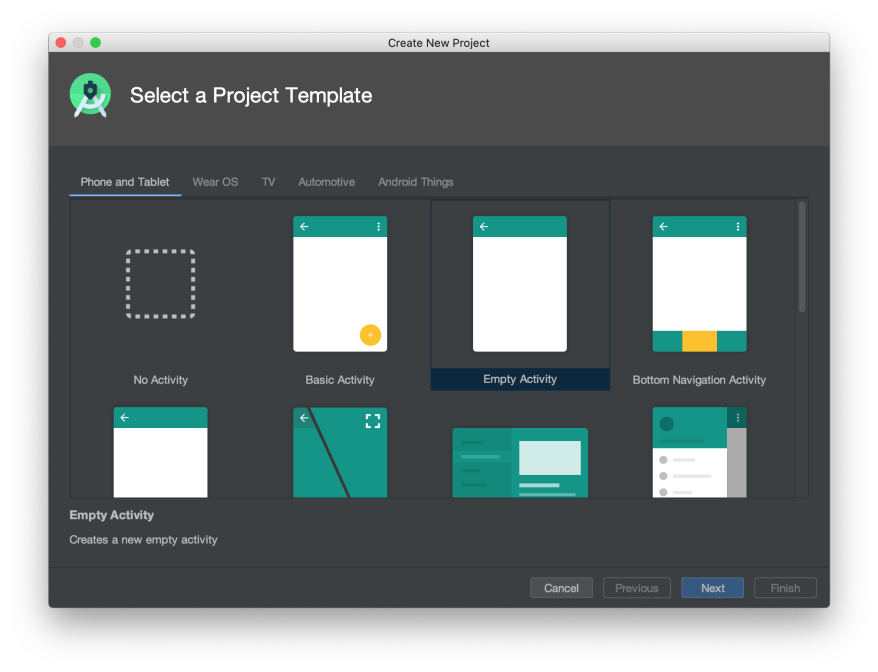
Siempre lo primero que se debe hacer al iniciar Android Studio, es realizar un chequeo de posibles actualizaciones, si este no lo hace automático, nos dirigimos a Android Studio y luego en “Check for updates”, al igual podemos ver la versión en “About Android Studio”. Estar actualizados siempre es mejor, ya que incorpora características nuevas en cada update.



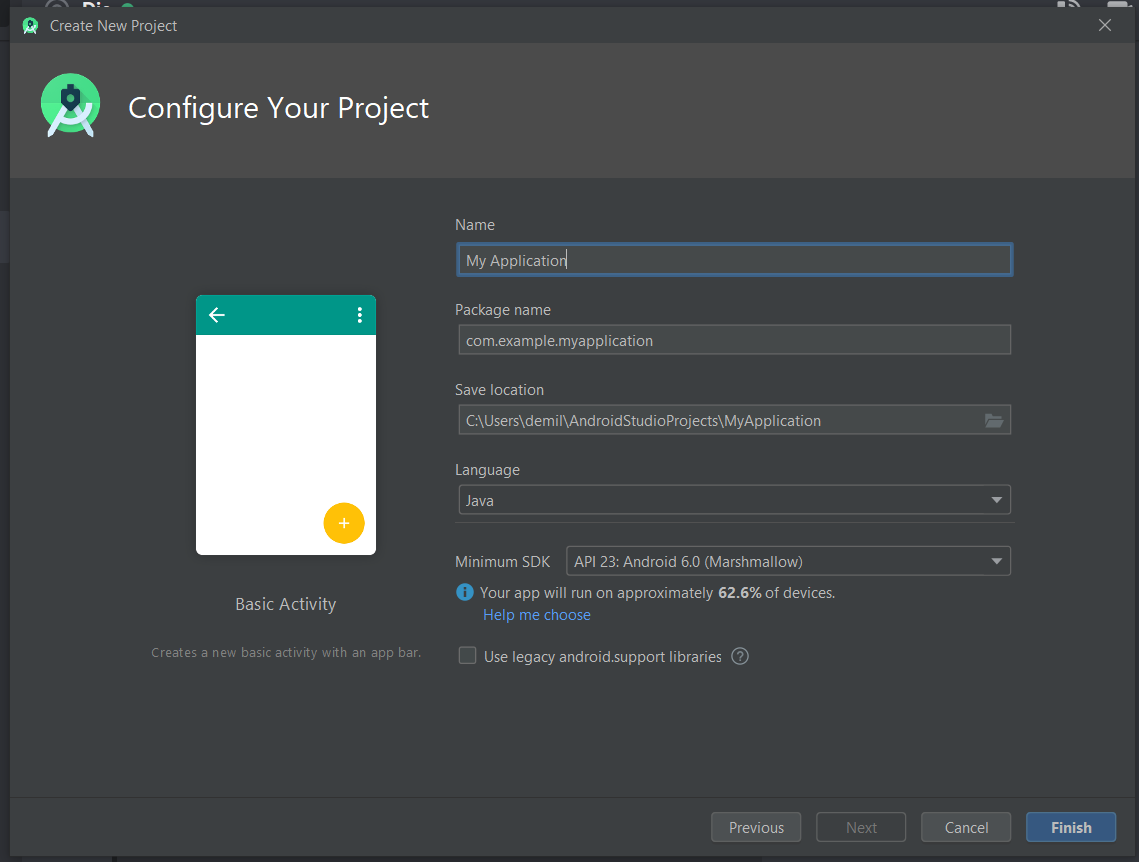
Cuando iniciamos Android Studio, en la pantalla principal, para iniciar un nuevo proyecto, nos vamos en la primera opción, “Start a new Android Studio Project”.



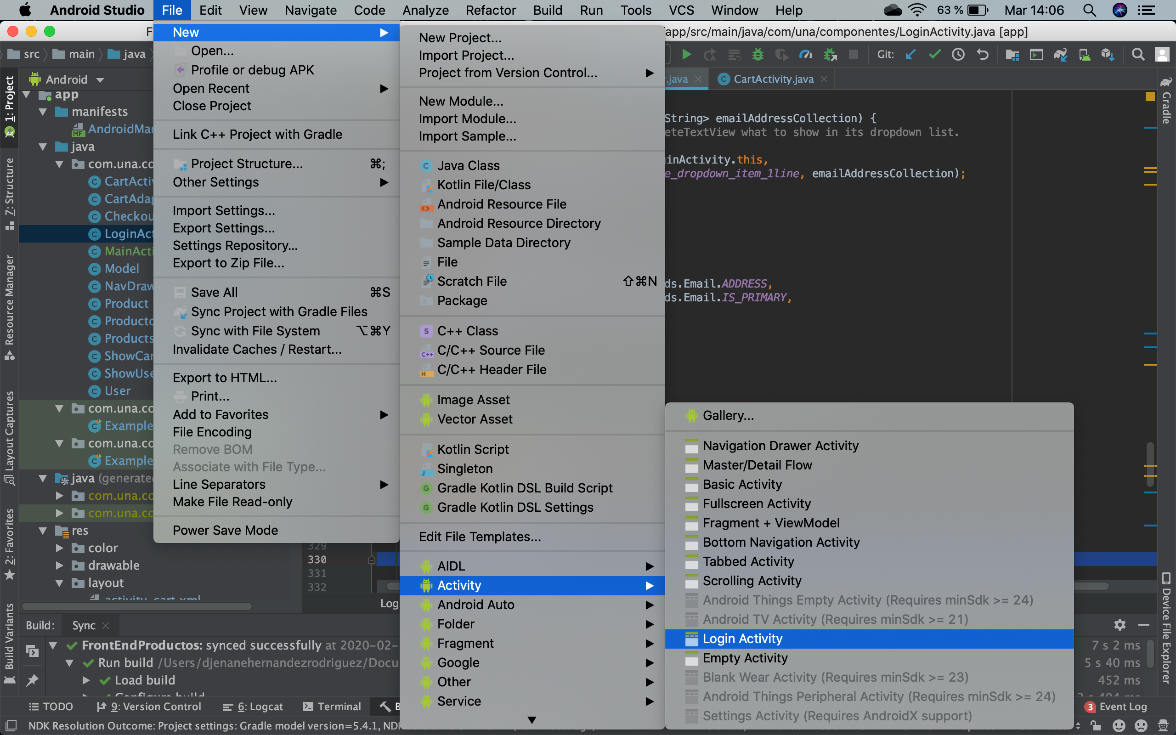
Luego, nos va a mandar a una sección en donde podemos escoger diferentes plantillas, como, por ejemplo, una actividad básica, una actividad de Login, en este caso, abriremos una actividad básica.



Cuando estamos dentro de esas plantillas, nos dará diferentes opciones, tales como el nombre del proyecto, nombre del paquete, la localización en donde podemos guardar el proyecto, incluso el lenguaje en el cual vamos a desarrollar nuestro proyecto, entre ellos: Java y Kotlin. Además de las cosas más importantes, la versión de Android en el cual podemos lanzar el proyecto, esto es bastante importante, ya que tenemos que considerar la cantidad de dispositivos que contienen un sistema operativo más actual, por eso, se debe considerar un sistema operativo mas viejo, para englobar a mas usuarios utilizando Android, no obstante, Android Studio nos indica que tanto porcentaje de usuarios utilizan cada sistema operativo.

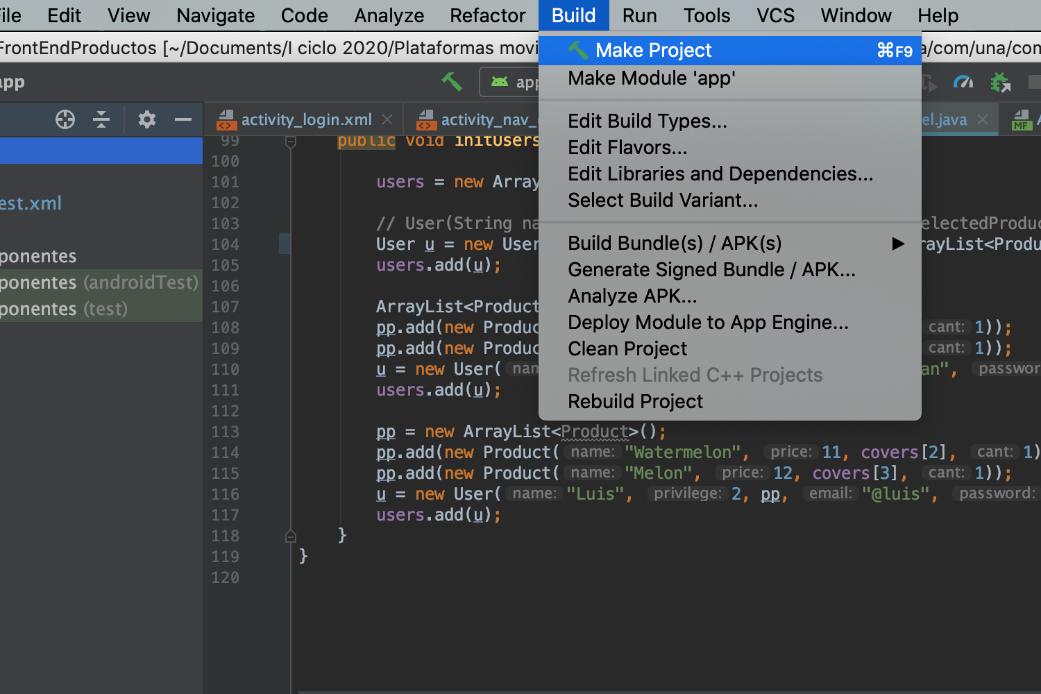


Ya dentro del proyecto, podemos ver que Android Studio tiene muchas opciones o características que podemos utilizar, este reporte se va a enfocar en lo que hemos visto durante el curso de móviles, por ejemplo, cuando vamos a crear una nueva actividad en el proyecto, nos iremos a la opción de file, luego en new, después nos dirigimos en activity y escogemos la actividad la cual vamos a desarrollar.

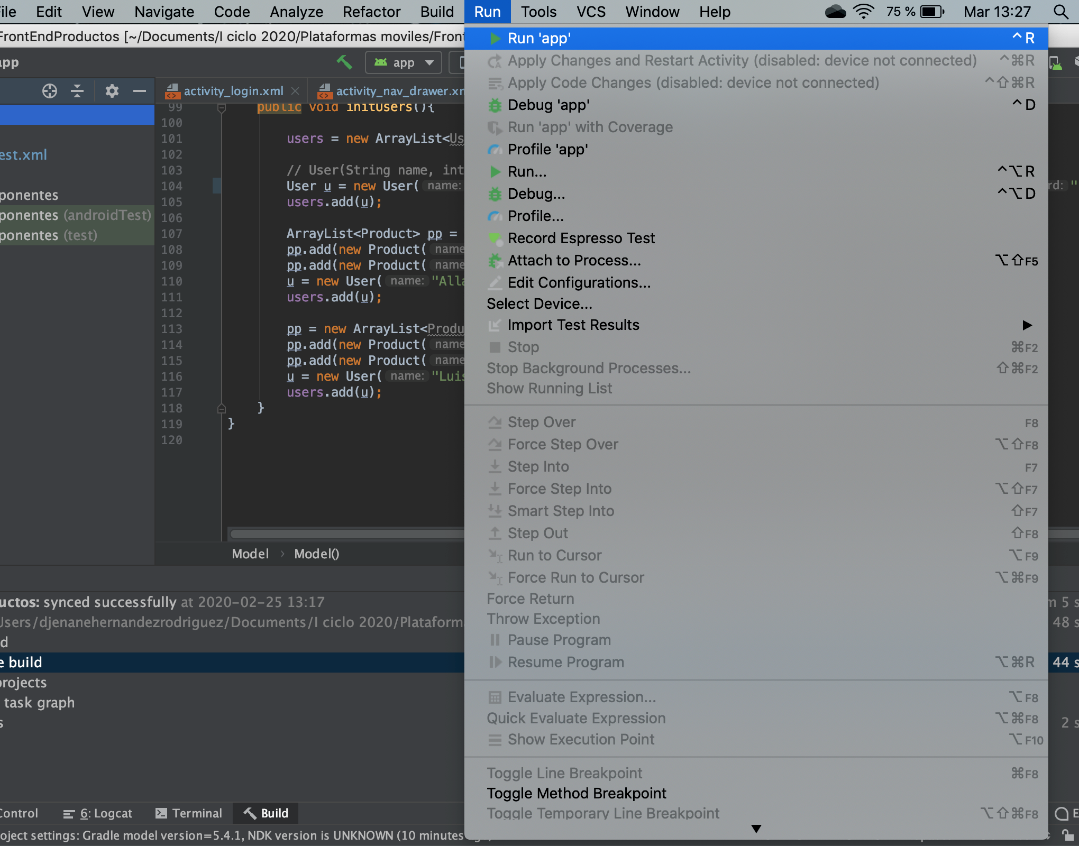


También se puede observar cómo en new, podemos crear un nuevo proyecto, importar, crear módulos, etc. Aquí es donde podemos administrar diferentes componentes del diseño móvil, una de las partes más importantes.

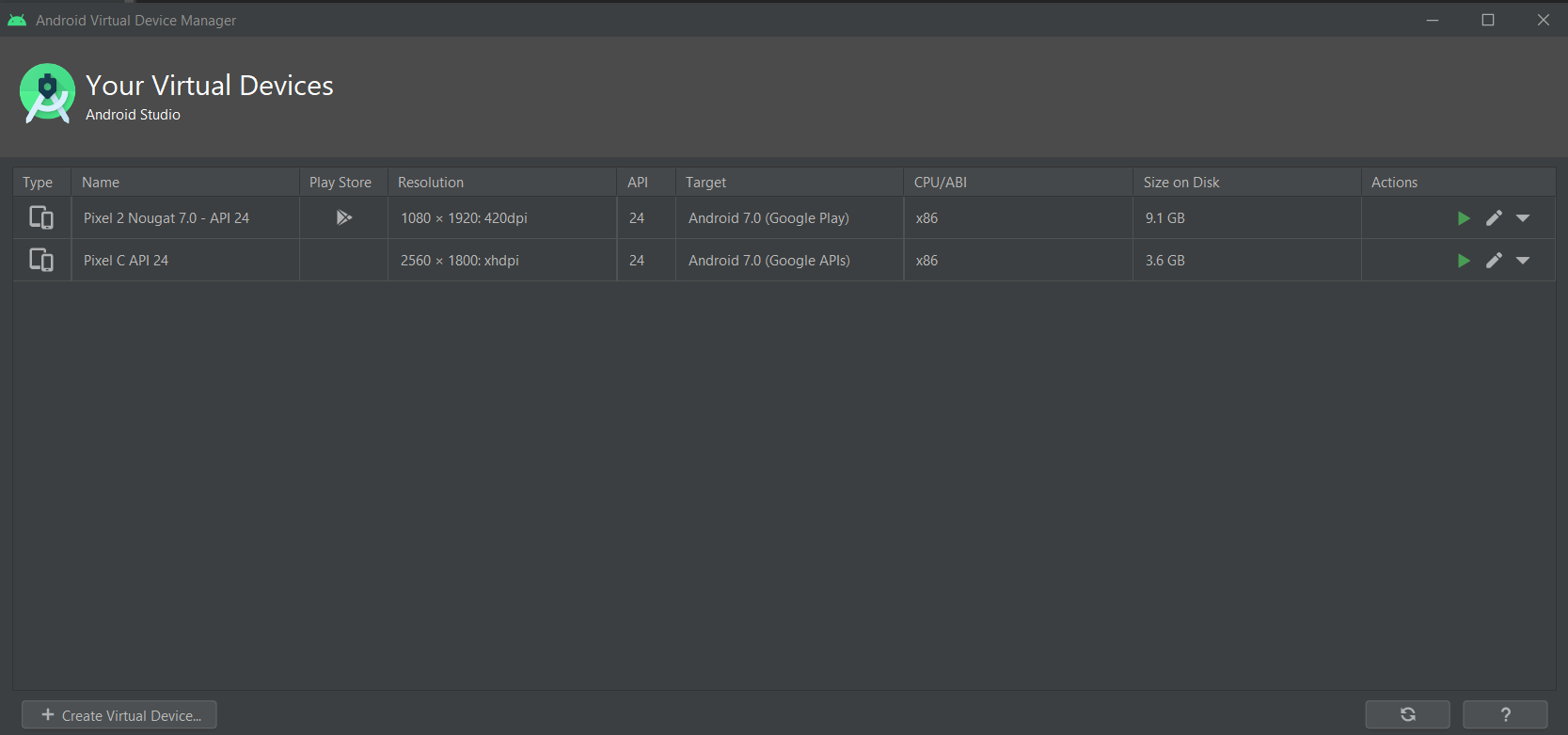
Otra parte importante es el cual nos vamos a la pestaña de built, en esa pestaña vamos a encontrar lo que es make project, en el cual esta función nos va a ayudar a hace una construcción del proyecto o reconstrucción si se selecciona el rebuilt project, para así actualizar el archivo de propiedades, para incorporarlo al proyecto. Es recomendable hacerlo cuando se elimina una actividad o algún layout, para que este se actualice.



Al lado de built, tenemos la pestaña de run, en la cual al darle clic nos va a abrir unas opciones, en las cuales podemos encontrar la de poder correr la aplicación, debuggear la aplicación, entre otras, aquí es cuando se dirige para poder ejecutar la aplicación.



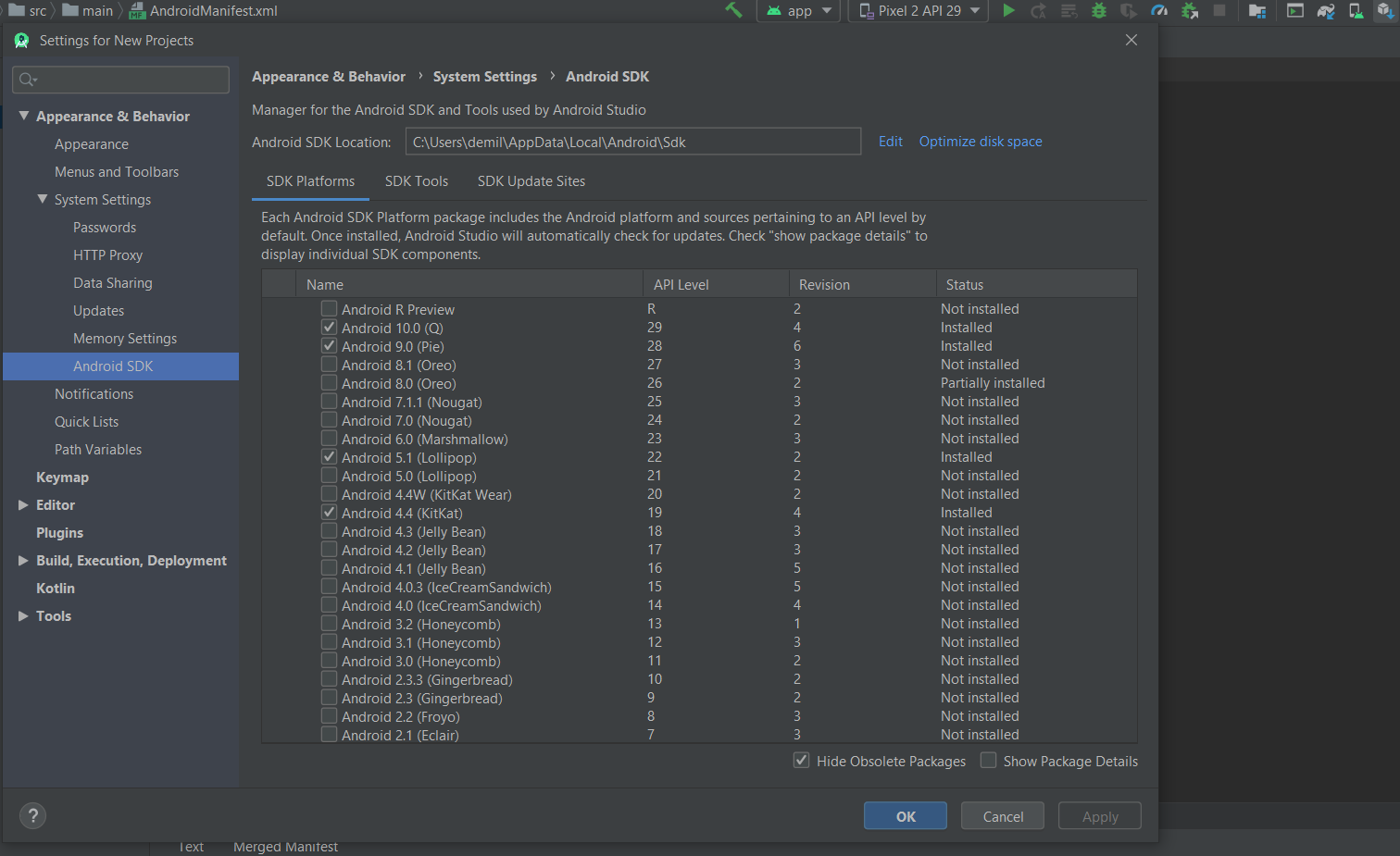
Siguiendo la línea, se puede ver el AVD Manager, en el, podemos ver los dispositivos que tenemos conectados con Android, además de los distintos emuladores de Android que se pueden instalar, por ejemplo, un celular Píxel 2 y una table Píxel C.



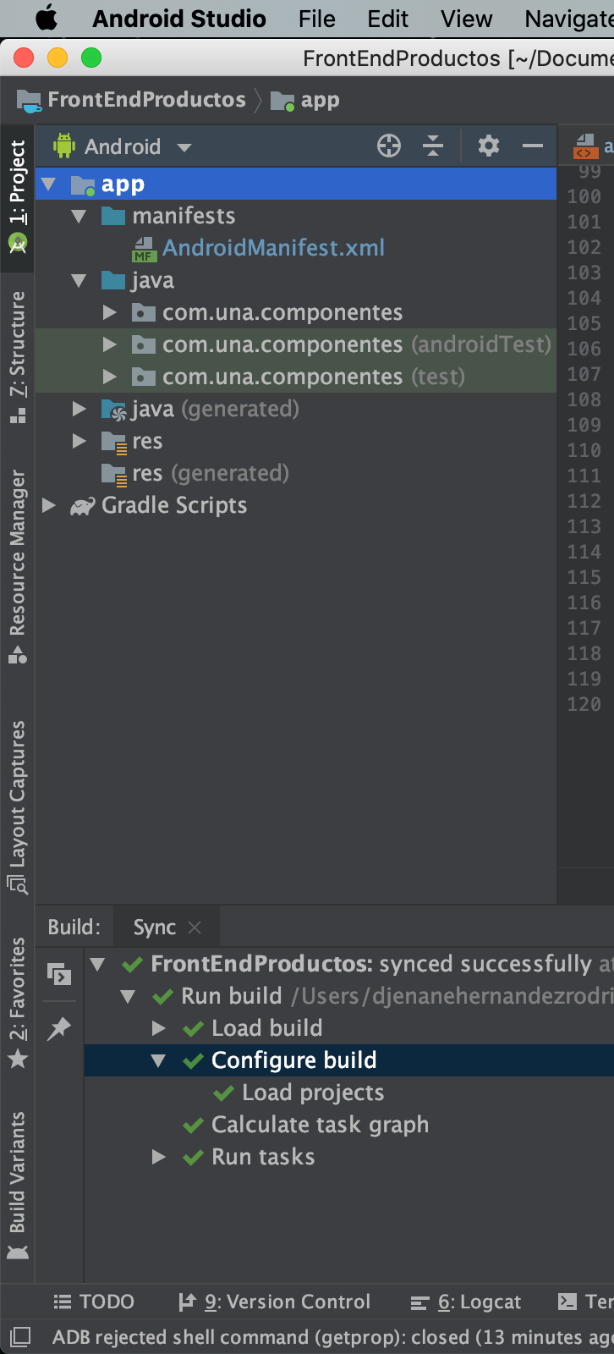
Además, a la par del AVD, podemos ver dos botones importantes, como lo es el poder sincronizar el proyecto con los archivos del Gradle.



El segundo, es el SDK Manager, en el cual dentro, podemos escoger los Sistemas Operativos que queramos para que sea la base del proyecto.

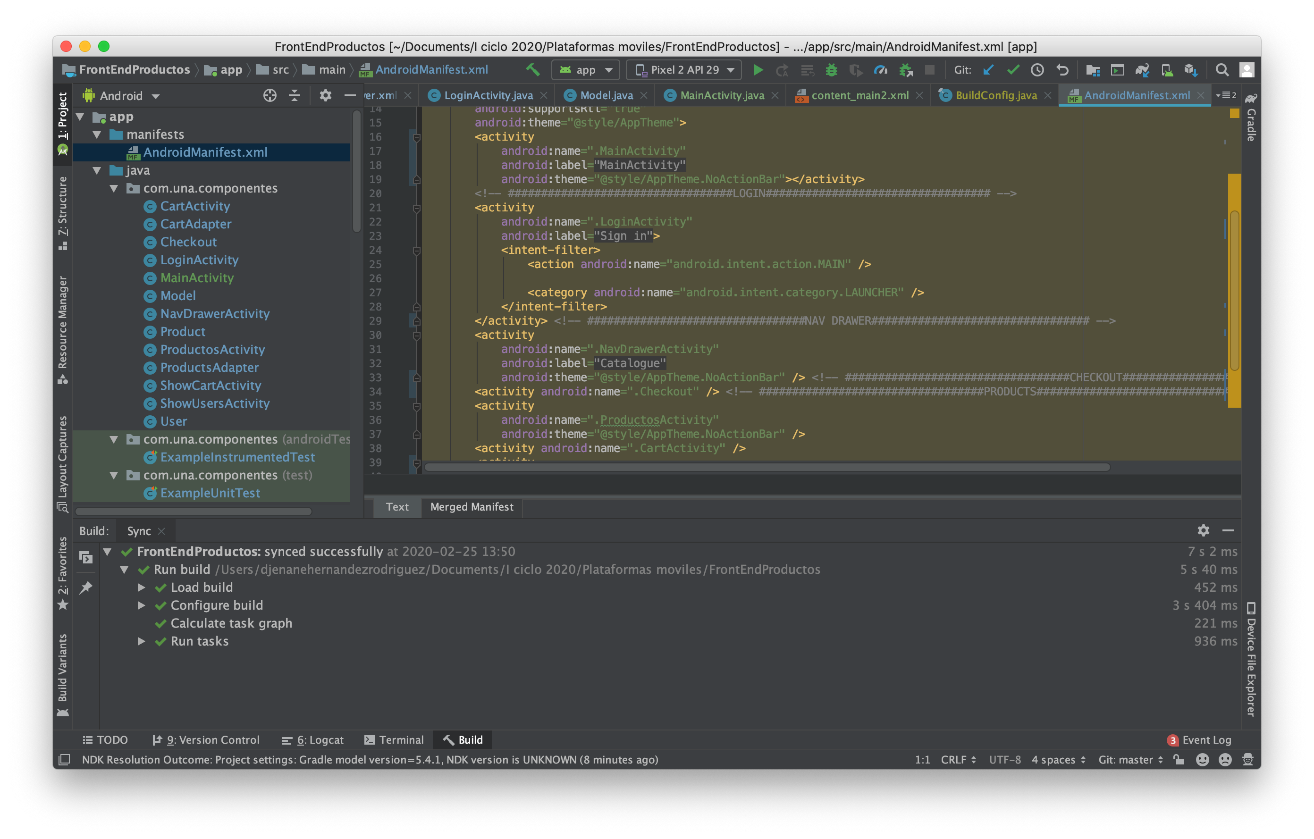


Viendo el lado izquierdo de Android Studio, podemos notar lo que el proyecto genera predeterminadamente, nos genera 3 carpetas importantes.



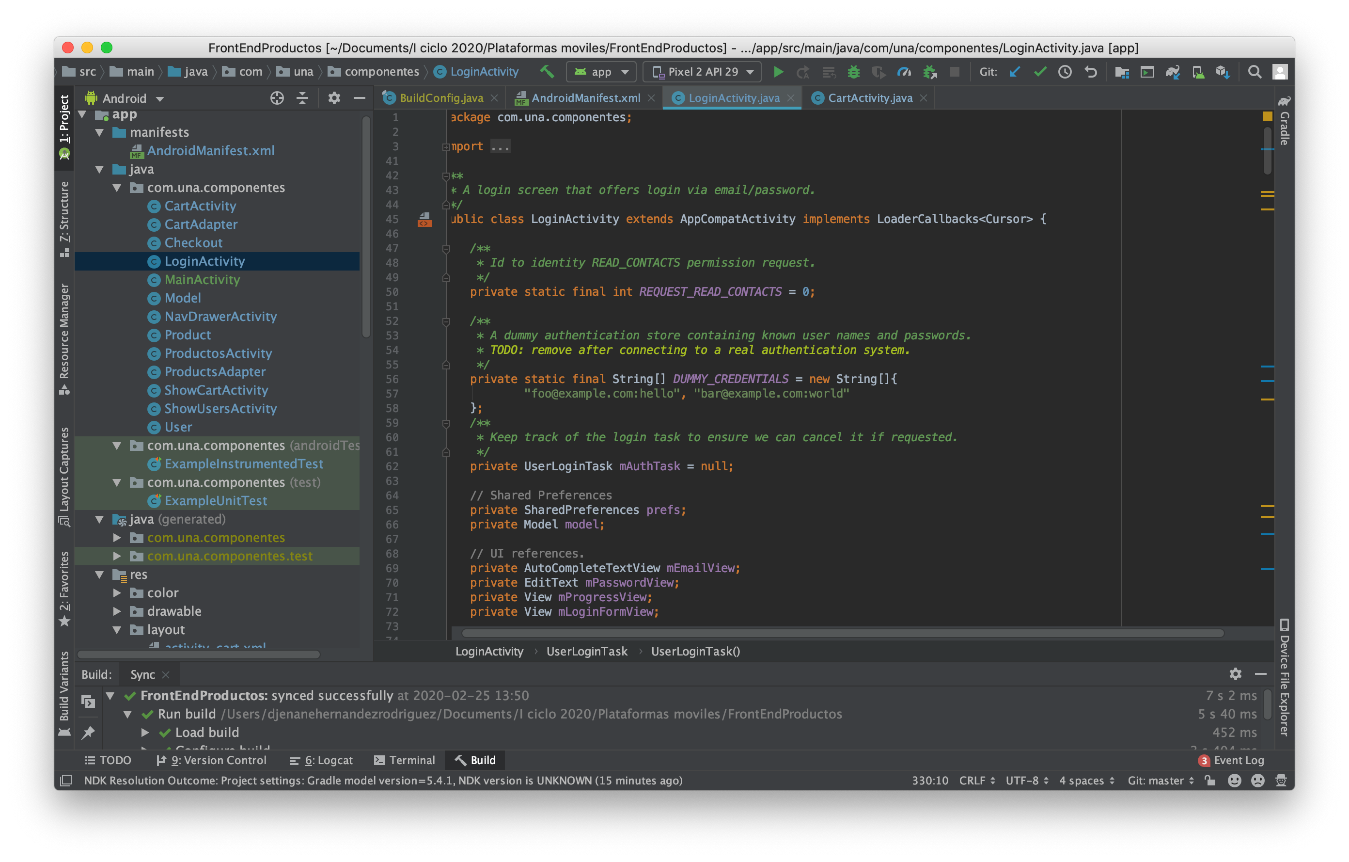
*Manifest*

La primera que encontramos se llama manifest, como su nombre lo indica, es un manifiesto de todas las actividades o fragmentos que la aplicación móvil contiene en todo el programa, es de suma importancia, ya que aquí es cuando se agrega una nueva actividad o se elimina, todo se va a ver reflejado en este XML, es importante saber que no se puede pasar de un layout hacia otro sin pasar por un controlador, además es de suma importancia que el manifest nunca debe de ser manipulado.



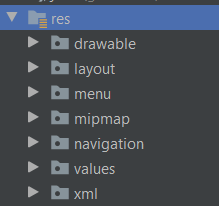
Carpeta Java

La segunda pestaña que se encuentra es la de Java, esta, si se escogió Java como lenguaje de programación, aquí es donde se van a encontrar todas las clases o actividades que se vayan creando a lo largo del proyecto.



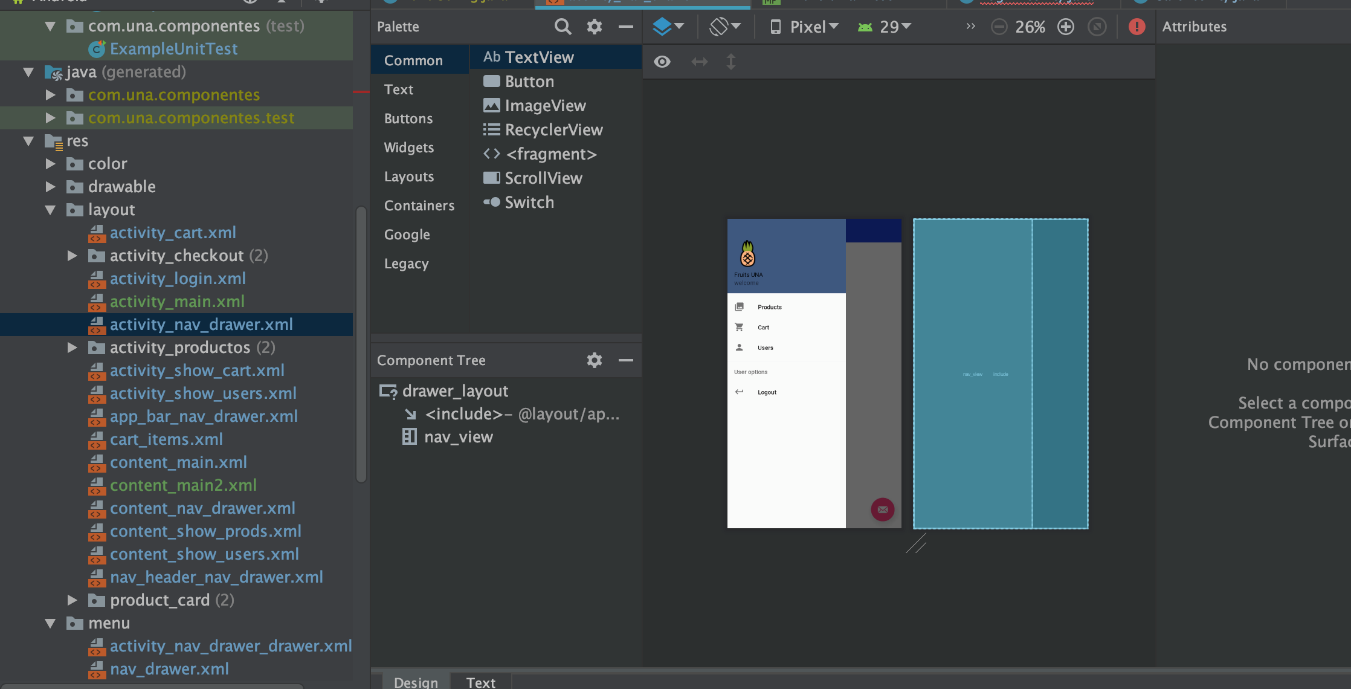
*Res*

La ultima pestaña importante que se puede observar, es la res, aquí se pueden observar varias pestañas más, las cuales son las de color, drawable, menú, values y layout.



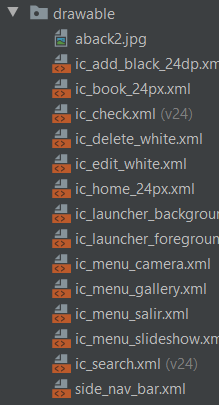
*Layout*

Es uno de los más importantes ya que podemos ver el XML y la parte grafica de la actividad que se agregó al proyecto, aquí podemos agregar botones, sliders, cuadros de texto, etc.



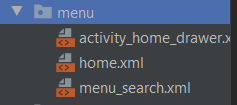
*Drawable*

Describiendo las otras pestañas en res, esta la de drawable, en esta podemos ver las imágenes en formato XML y otros de las que se han agregado a un proyecto y así poder utilizarlas haciendo un llamado en el layout de la actividad “app:srcCompat="@drawable/aback2", este, por ejemplo.



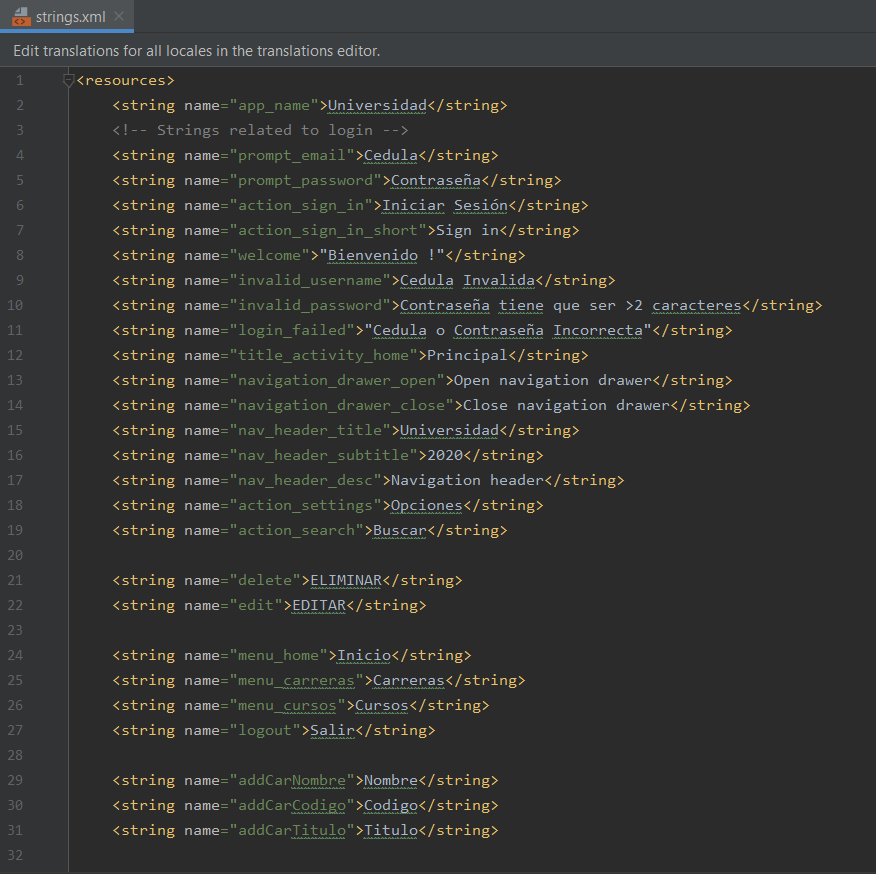
*Menu*

La siguiente es la del menú, donde vemos la barra arriba del Navdrawer o el de una actividad en la cual le podemos poder una barra, por ejemplo, un botón de búsqueda.



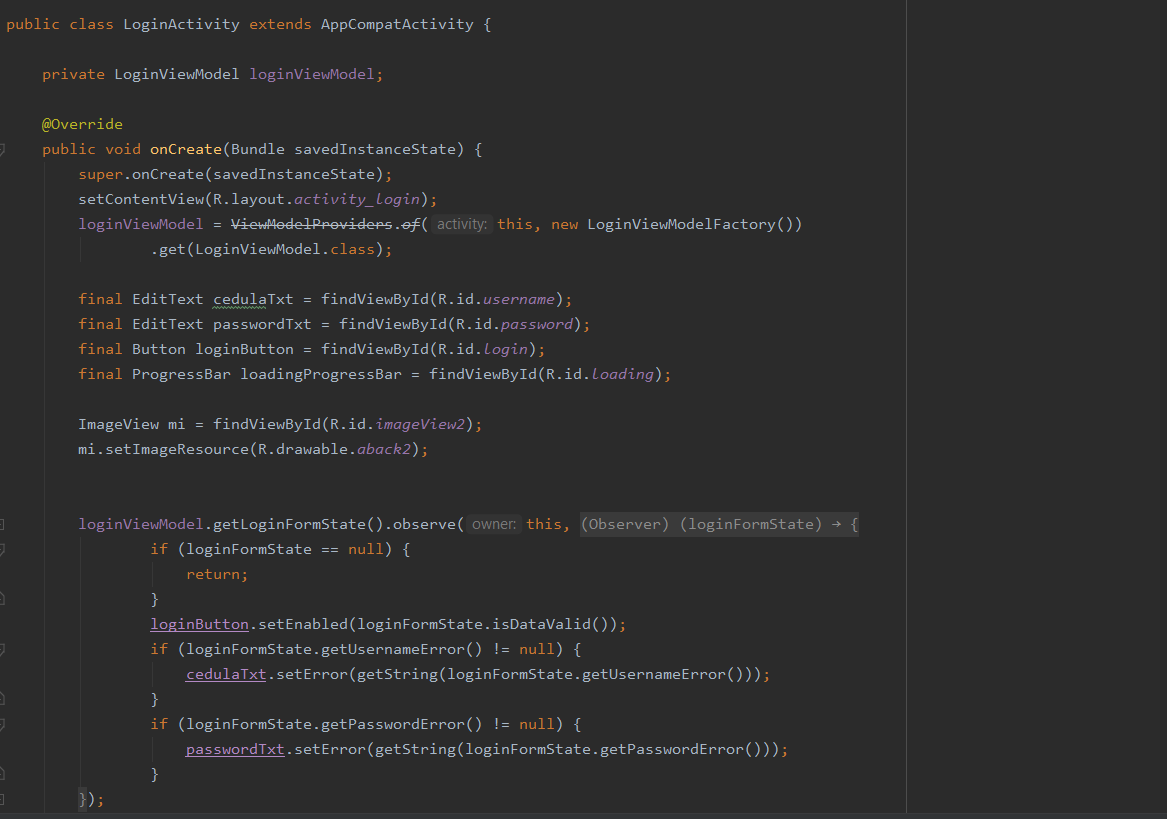
*Strings*

Otra carpeta importante, es la que está en values, llamada “Strings”, aquí es donde podemos ver los nombres de diferentes strings que le podemos asignar digamos a los nombres de los botones del Navdrawer, el nombre de la aplicación, el de las actividades, y más.



*R*

Una variable importante que tenemos que tomar en cuenta, es la R, aquí la R es como una clase de recursos, que esta nos va a permitir llamar a diferentes recursos de la carpeta res, ya sea drawable o un layout, esta es muy importante, ya que sin esta, la conexión entre la parte de programación y sus recursos de “imagen” o “views” no seria accesible.

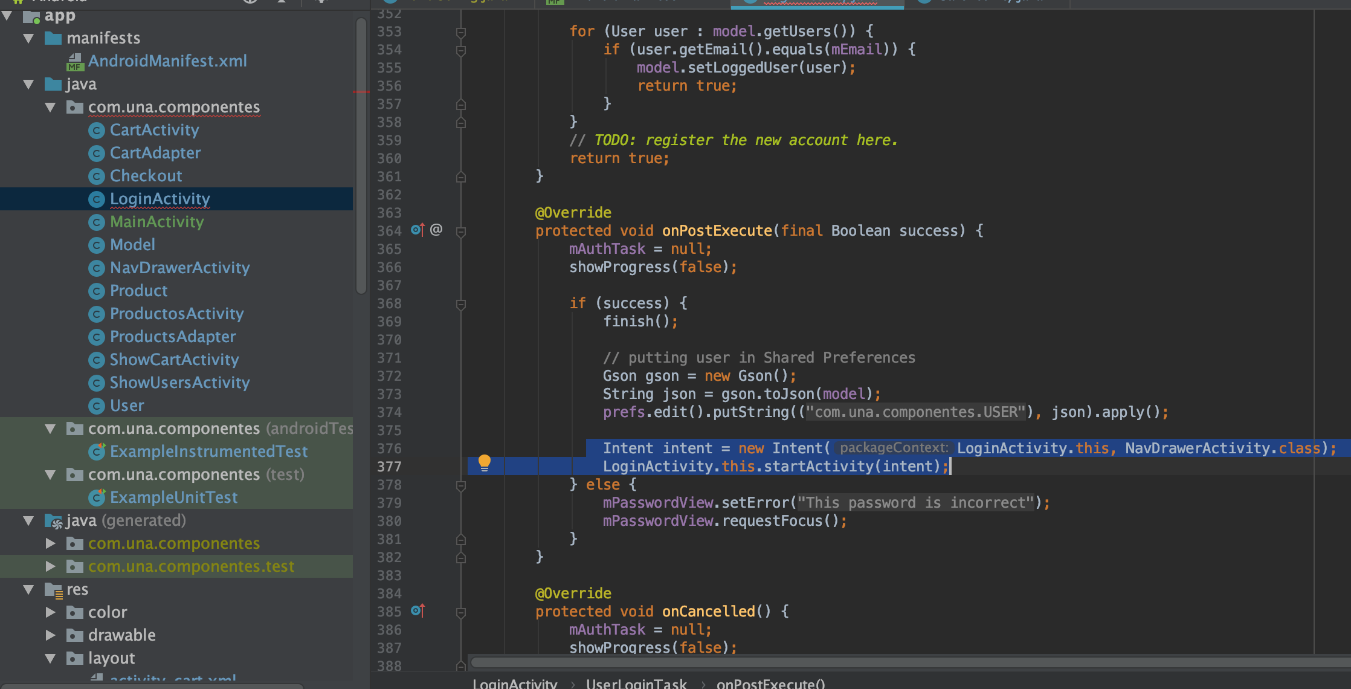


Como podemos ver, en el Login, esta actividad tiene en un layout, que se crea junto a la actividad, para ella, pero, hay que linkearla, y para eso existe R, como se puede observar en la imagen de arriba, cuando seteamos un ContentView, este nos permite setear el layouts que queremos para esta actividad, entonces con R buscamos en los layouts y el nombre del layout que ocupamos, así de sencillo, también con los id´s de los campos, ya sean cedula, contraseña, botones, etc. Con el método de buscar por id, y pasarle el id contenido en el R y en la categoría de id´s, se hace sencillamente, R es parte esencial de ese linkeo entre la actividad y sus diferentes recursos que se van a utilizar en esa actividad.

*Intent*

Como se manifestó en la parte del manifest, no podemos pasar desde un layout hasta otro, esto sin pasar por un medio controlador. Dos formas de enlazar un controlador con un layout, cuando se crea en el evento create y se hace a través del set content view, y la otras con los intents, nos puede ayudar a pasar del controlador A hacia B en cualquier momento.

En el ejemplo podemos ver la función del Intent en la parte del Login activity.



El ejemplo anterior vemos como desde la actividad Login, que es la que esta predeterminada para que se ejecute primero, cuando el usuario es exitosamente loggeado, con el Intent podemos navegar a otra actividad, por ejemplo, al Navdrawer, se crea el Intent y este empieza la actividad de ese Intent.

*Manejo de Activities*

Viendo un panorama variado, el manejo de actividades puede hacerse de diferentes maneras, usando los intents, en el navdrawer, por ejemplo, cuando queramos linkear un botón con una actividad, el navdrawer tiene un método llamado “onNavigationItemSelected(@NonNull MenuItem item)”, este significa que, si el usuario toca un ítem del menú del navdrawer, este ejecuta una acción, y en este caso, es desplazarnos de actividad.



public void cursos(){  
 finish();  
 Intent a = new Intent(this, CursosActivity.class);  
 startActivity(a);  
}

con esto, le damos por finalizado a la actividad donde estamos, después creamos el Intent, que es de esta actividad hacia otra actividad, en este caso Cursos, y luego la iniciamos, al igual que el botón de atrás, hacemos el mismo procedimiento.