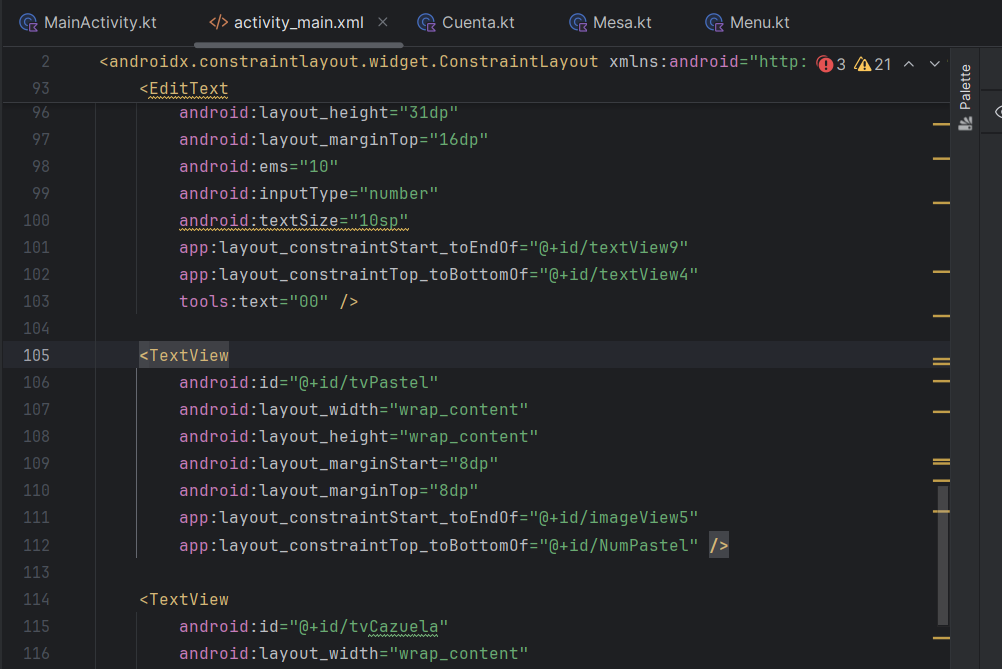
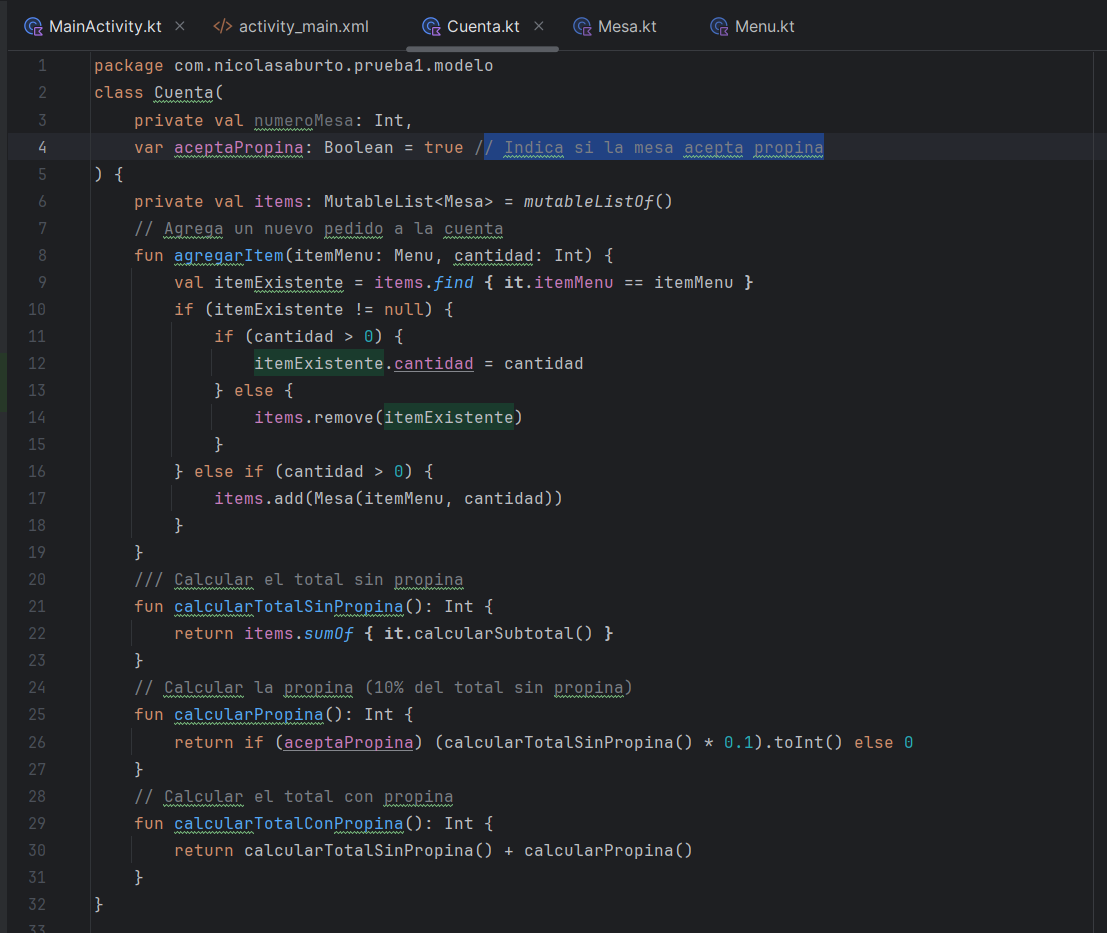
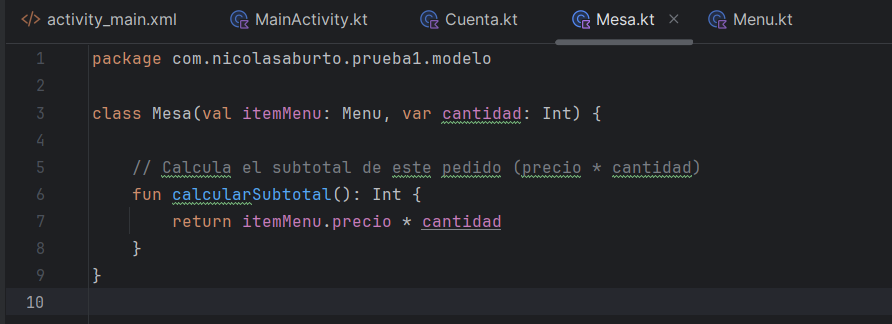
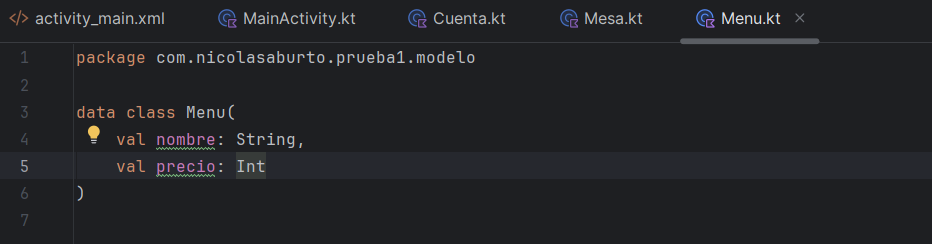
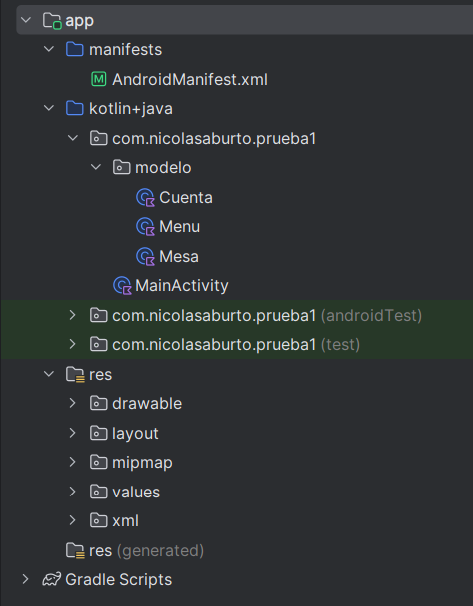
Primero se crea la pantalla principal de la aplicación usando ConstraintLayout, insertando 3 imágenes con imageView, uno para el logo, y 2 para la cazuela y pastel, se insertan 7 textView para los textos del menú y los valores, 2 numbers para ingresar las cantidades, otros 2 textView para mostrar los subtotales de cada plato, y otros 3 textView para mostrar el valor total, la propina y el valor total mas propina:

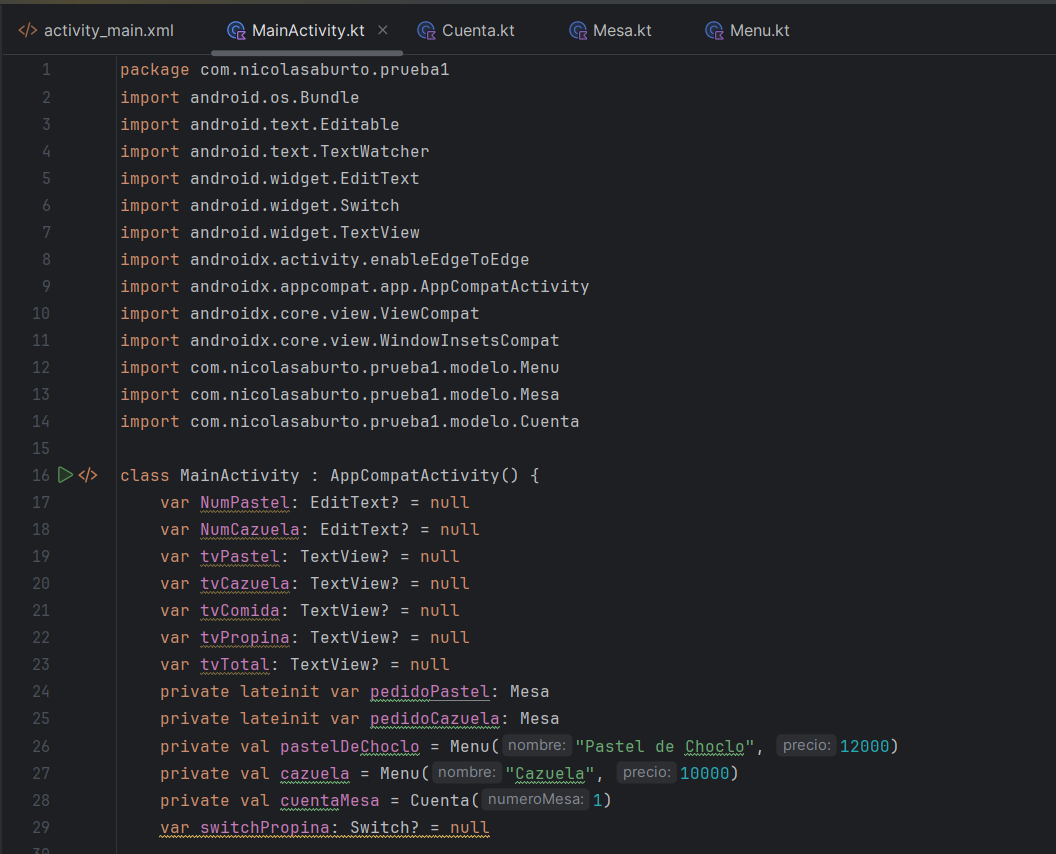




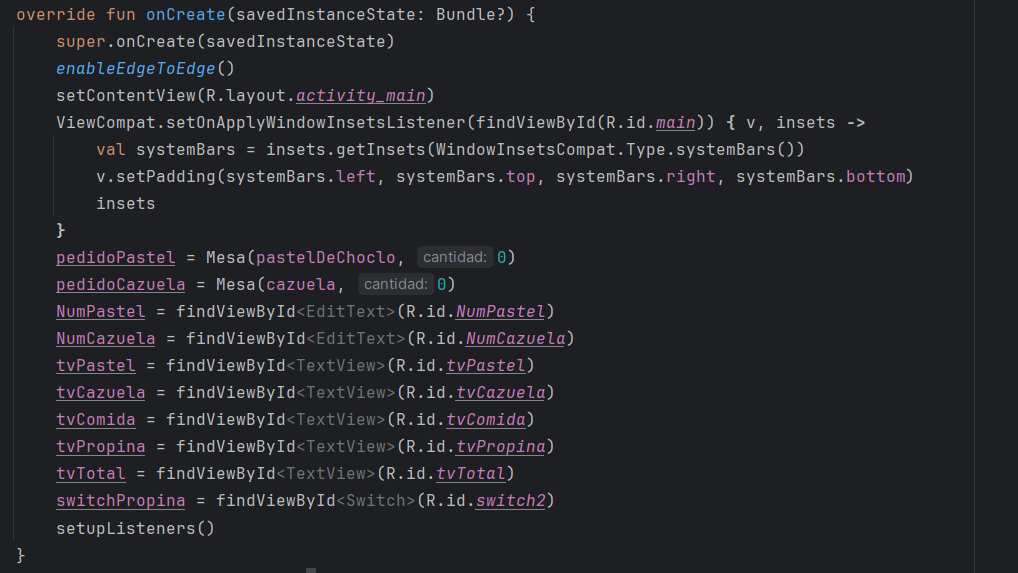
Se construyen las clases modelo para solucionar el problema de Restaurant, se crea un pakage llamado modelo, dentro del cual se crean 3 clases Kotlin, Cuenta, Menu y Mesa, en Menu se crean las variables nombre y precio, en la clase Mesa se crean la variable cantidad, y se llama a la clase Menu como itemMenu, luego se crea el método calcularSubtotal(), con el cual multiplica el precio por la cantidad y se retorna, finalmente la clase Cuenta, donde se crean la variable agregarPropina, y los métodos agregarItem(), calcularTotalSinPropina(), calcularPropina() y calcularTotalConPropina(),



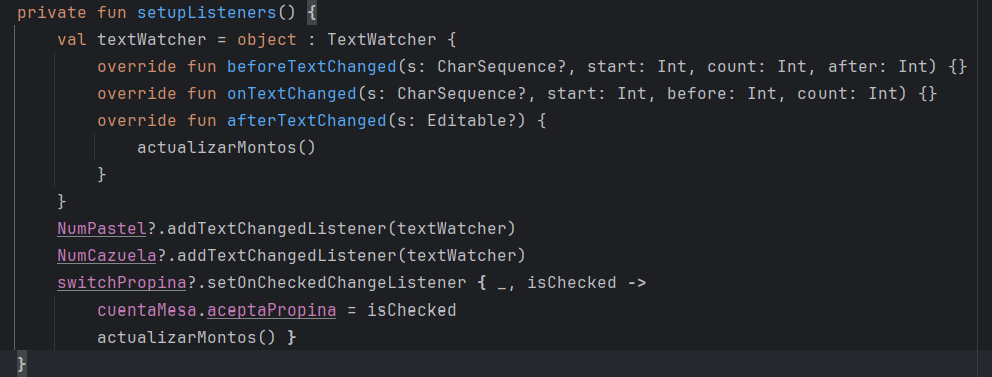
Luego de lo anterior, se modifica el MainActivity, primero las importaciones necesarias, como las clases, luego se crean las variables para referencian los elementos de la interfaz grafica, los Textview, Number y el Switch, se crean las variables pedidoPastel y pedidoCazuela, luego las variables cazuela y pastelDeChoclo, agregando los nombres y valores y enviándolos a la clase Menu, la variable CuentaMesa.



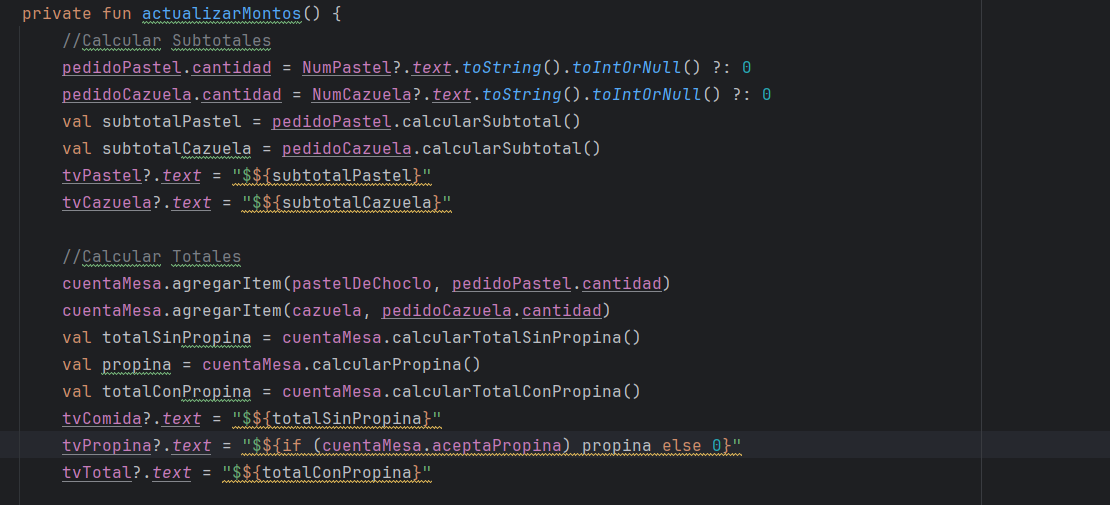
En el método onCreate, con findViewById se busca a los elementos correspondientes del XML por sus IDs.



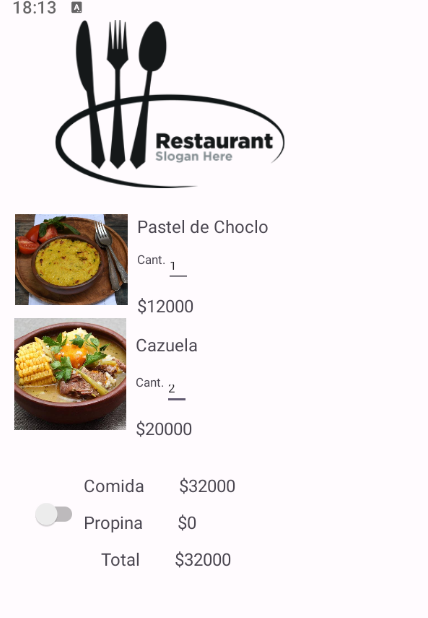
Se crea el método setupListeners para observar los cambios en los campos de texto NumPastel y NumCazuela, esto con addTextChangedListener, y con setOnCheckedChangeListener se observa el cambio en el Switch, y en ambos casos se llama al método actualizarMontos()



Se crea el método actualizarMontos(), primero se calcula los subtotales, envando la cantidad ingresada en NumPastel y NumCazuela, y usando el método calcularSubtotal() de la clase Mesa, luego se muestra los resultados en tvPastel y tvCazuela, la segunda parte se calcula el total, primero se envían los ítems al método agregarItem() de la clase Cuenta, luego se calcula el total sin propina con el método calcularTotalSinPropina(), luego se muestra en tvComida, luego se calcula la propina con el método calcularPropina(), luego se usa la variable aceptaPropina, la cual de tener un valor, se muestra valor en tvPropina, finalmente se calcula el total con propina con el método calcularTotalConPropina(), y mostrándolo en tvTotal



A continuación se muestra la aplicación funcionando, en la primera imagen se muestra los totales con propina, en la segunda sin propina

Problemas Encontrados:

Se encontraron diferentes problemas y dificultades en el desarrollo de la aplicación, como la sintaxis del código, o la importación de diferentes elementos en las clases, pero el problema principal fue el requerimiento del programa Android Studio, el cual llegaba a consumir mas de 2GB de RAM, a eso se del debe sumar la emulación para hacer pruebas, en mi caso baje un emulador alternativo MEMU, ya que el emulador de Android Studio nunca funciono óptimamente, aun así ambos programas abiertos y corriendo, mas la virtualización, superaban los 5GB, en equipos de 8GB de RAM como el mío estos programas no funcionara a su máxima potencia, ya que esos 5GB, mas otros programas abiertos del sistema, como Word y Explorador de Internet supera la capacidad RAM del equipo

