

A continuación, se describen las modificaciones de la aplicación, con relación al modelo UML planteado en el enunciado del proyecto.

## Modificaciones a la clase aplicación

Se añadieron varios métodos a la clase aplicación. Las adiciones se hicieron con el fin de hacer el código más modular, y más legible. Sin embargo, el funcionamiento de la clase es el mismo.

## Modificaciones a la clase restaurante

Se reemplaza el método cargarMenu() por dos métodos cargarBebidas() y cargarNoBebidas(). Esto con el fin de manejar las bebidas por separado de los otros elementos del menú. Así, la clase restaurante tiene muchos combos, muchas bebidas y muchas no bebidas. Adicionalmente, se añaden los métodos getBebidas(), getNoBebidas() y getCombos, y se elimina el método getMenuBase(). Finalmente se añaden los métodos getProductosMenu() y getProductosMenuHT(), que devuelven una ArrayList o una Hashtable, con todos los productos base del menú (sin incluir combos).

## Modificaciones a la clase Pedido

La clase pedido ya lo almacena todos los Productos bajo una misma relación con la interfaz producto. Por el contrario, almacena combos, bebidas, no bebidas y productos ajustados por separado. Lo anterior se hace para facilitar la comparación de 2 pedidos (revisar si son idénticos).

Por lo anterior el método agregarProducto(), se convierte en 4 métodos diferentes, cada uno para las 4 clases de productos. Adicionalmente, se añade el método getProds(), que retorna un ArrayList(), con todos los productos (incluyendo combos) que hay en un pedido. Esto, con el fin de facilitar la obtención de precios, y calorías del pedido.

## Modificaciones a interfaz Producto

Se añade el método getCalorias() a la interfaz producto (y a todas las clases que la implementan, por ende), para poder obtener las calorías de los productos, e incluir ese dato en la factura.

## Modificaciones a clase Ingrediente

Se añade el atributo calorías (int) y el método getCalorias(), con el fin de calcular las calorías de los productos modificados.

## Modificaciones a la clase ProductoAjustado

Se añaden los métodos agregarIngrediente() y eliminarIngrediente(), con el fin de facilitar la modificación de productos ajustados. Por otro lado, se añaden los métodos getAdedIngredients() y getDelIngredients, para facilitar la comparación de dos productos ajustados (revisar si son idénticos).

## Cambios en clase ProductoMenu

Para facilitar la separación entre bebidas y no bebidas, la clase ProductoMenu se convierte en una clase abstracta, de la cual heredan las clases bebida y noBebida.

## Cambios en la clase combo

El método agregarItemACombo(), ya no recibe por parámetro un Producto en general, si no específicamente un ProductoMenu. Lo anterior, con el fin de evitar que se añadan combos a los combos.

## Creación de la clase format

La clase format, contiene dos métodos estáticos que facilitan la generación de los textos para la factura.

## Adición del método equals

Se agregó el método equals() a las clases Pedido, ProductoMenu, Combo, Ingrediente y ProductoAjustado, con el fin de comparar si dos pedidos son idénticos.