**Modelo Implementado**

**Aplicación:**

* El método ejcutarOpcion(int opcionSeleccionada) es remplazado por ejecutarAplicacion() que directamente llama a una de las funciones mostrarMenu(), recibirPedido(), modificarPedido(), cerrarPedido() y consultarPedido() según la opción seleccionada por el usuario.

**Restaurante:**

* Los métodos getMenuBase() y getIngredientes() que retornaba un ArrayList ahora retornan un Map con llave de tipo Integer y valor un Producto, Ingrediente o Combo. Se realizo este cambio para poder almacenar el id del producto como llave en el Map sin necesidad de afectar el modelo de los Productos. Los clientes pedirán sus productos a través de la consola con el id de cada producto, con el fin de evitar que tengan que ingresar el nombre completo del producto.
* También se creó un nuevo atributo denominado ultimoIdDisponible, ya que si bien combos, productos e ingredientes son diferentes mapas, y no habría problema en repetir el id en cada mapa, cuando se imprima en consola en el menú se busca que cada id sea diferente sin importar que cambien de categoría. Para obtener este id se debe usar el método ultimoId() que retorna el id guardado en el atributo e incrementa en uno su valor.
* El método iniciarPedido() cambio su firma, dado que además del nombre y dirección del cliente, también recibirá listas de números enteros que representaran el id de un producto para dada categoría, es por ello que recibe 4 listas, combos, productos, adiciones y ingredientesRemovidos.

**Pedido:**

* Se creo un nuevo atributo llamado iva que es de tipo Double y es estatico para almacenar el valor del impuesto del iva que en este caso es del 19% pero en decimal seria 0.19.

**Ingrediente:**

* Se creo un nuevo método que es el de generarTextoFactura(), ya que por la forma en que se decidió construir el texto de las facturas, el solo nombre de los ingredientes no era suficiente. También se necesitaba su valor y para evitar llamar ambos métodos get, se consideró crear un método específico para ello.

**ProductoAjustado:**

* A la clase producto ajustado se agregaron los métodos de agregar y eliminar ingredientes que en ambos casos reciben un ingrediente para agregar a la lista de agregados o para agregar a la lista de eliminados.

**Generales:**

* En general también se dejaron de usar tipos primitivos y se usaron solamente Clases, es decir int -> Integer y double -> Double