Andrea Galindo Cera - 202122477

Taller #2

Objetos y herencia

1. Consola:

**Aplicación**: La clase Aplicación es la única clase del diagrama que es parte del paquete de la consola.

* Ejecutar aplicación: se ejecutan los métodos según la opción que ingrese el usuario.
* Mostrar Menu
* EjecutarMostrarMenuRest: Carga los archivos txt con la información del restaurante e imprime el menú completo del restaurante.
* IniciarPedido: inicializa un pedido con el nombre y dirección del usuario.
* AgregarElemento: Permite al usuario agregar un producto, combo o bebida. Inicialmente imprime los elementos del menú para que el usuario elija cual adicionar. Según la opción seleccionada, se agrega el elemento al pedido actual.
* AgregarIngrediente: Permite al usuario agregar o eliminar un ingrediente de un producto específico. Inicialmente imprime los ingredientes para que el usuario elija si quiere adicionar/eliminar y cual adicionar/eliminar. Según la opción seleccionada se crea un nuevo ProductoAjustado con los ajustes y este se agrega al pedido. Además se elimina el productobase del pedido para solo mantener el ProductoAjustado.
* CerrarPedido: Cierra y guarda el pedido. Devuelve al usuario su factura con el id.
* Conseguir por Id: Permite al usuario consultar una factura según su id.

1. Lógica:

**Restaurante** es el corazón de la lógica de la aplicación y sirve principalmente para agrupar el resto de las cosas que maneja la aplicación.

* IniciarPedido = crea un nuevo pedido con el nombre y la dirección del cliente.
* CerraryGuardarPedido: Guarda la factura del pedido actual. Verifica si ya hubo un pedido exactamente igual al actual. Resetea el pedido actual para que se pueda iniciar uno nuevo.
* ConsultarInfoPedidoPorId: Existe una carpeta “factura” con archivos .txt llamados como el numero de id. Se busca el archivo con nombre igual al id pasado por parámetro y se imprime su factura.
* getPedidoEnCurso
* getMenuBase
* getIngredientes
* getCombo
* getBebidas
* cargarInformacionRestaurante: llama a cada función de carga de datos.
* imprimirInformacionRestaurante: imprime el menú
* cargarIngredientes: lee el archivo con la información de los ingredientes. Separa cada línea por ; donde el primer item es el nombre, el segundo es el precio y el tercero son las calorías. Guarda esa información creando un nuevo ingrediente.
* cargarBebidas: misma implementación que en cargarIngredientes.
* cargarMenu: misma implementación que en cargarIngredientes.
* cargarCombos: misma implementación que en cargarIngredientes.

**Producto** (interface)

**Combo** (implementa producto): guarda el nombre del combo, el descuento y los ítems del combo.

* getCalorias
* getDescuento
* agregarItemACombo
* getItemsCombo
* getPrecio: suma los ítems dentro del combo y les saca el descuento a cada uno.
* generarTextoFactura: con el nombre del combo y su descuento.
* getNombre

**ProductoMenu** (implementa producto): guarda el nombre del producto, su precio base y las calorías.

* getNombre
* getPrecio
* getCalorias
* generarTextoFactura: con el nombre, precio base y calorías del producto.

**ProductoAjustado** (implementa producto): ajusta un producto base agregándole o eliminando ingredientes.

* agregarIngrediente
* eliminarIngrediente
* getNombre
* getCalorias
* getPrecio: si se adiciona un ingrediente, se le suma su precio. Si se elimina un ingrediente no se le resta precio.
* generarTextoFactura: guarda en la factura los elementos adicionados y eliminados.

**Ingrediente**: guarda la información del ingrediente (nombre, costo adicional y calorías)

* getNombre
* getCostoAdicional
* getCalorias

**Pedido**: guarda el nombre del cliente, la dirección del pedido, el id del pedido (que se toma del numero de pedidos) y los productos que hacen parte del pedido.

* getIdPedido
* getNombreCliente
* getDireccionCliente
* agregarProducto: agrega un item de tipo producto al pedido.
* eliminarProducto: elimina un item de tipo producto al pedido.
* getProducto: devuelve los productos del pedido guardados en un arraylist de tipo Producto.
* getProductosStr: devuelve los productos del pedido guardados en un arraylist de tipo string.
* getPrecioNetoPedido: devuelve el precio total del pedido + el valor del iva.
* getPrecioTotalPedido: recorre los productos que hacen parte del pedido y adiciona su costo.
* getPrecioIVApedido
* generarTextoFactura: genera la factura con el id, nombre del cliente, dirección y detalles del pedido.
* GuardarFactura: guarda las facturas en un archivo e imprime la factura al usuario.