

### ANALISIS TALLER 3

#### **Aplicación:**

Es la consola del proyecto, donde se interactúa con el usuario.

- Main: es donde se ubica el método main del proyecto
- mostrarMenu: muestra el menú de opciones al usuario
- ejecutarOpcion: ejecuta la opción que el usuario haya seleccionado

#### **Ferretería:**

Es el modelo que realiza las operaciones requeridas por el usuario.

- Ferretería: es el método que inicia las operaciones de la clase ferretería
- inventarioBase: es el inventario inicial que se carga con el método cargarInventario
- iniciarCompra: es el método que crea una nueva compra a partir de la cedula del cliente
- cerrarYGuardarCompra: cierra la compra que se esta realizando y guarda la información de la misma
- cargarInformacionFerreteria: carga los archivos que serán utilizados en la operación de la aplicación
- cargarInventario: carga el inventario con el que se quiere operar la aplicación
- cargarPaquetes: carga los paquetes dispuestos por la ferretería para incluirlos en los productos

#### **Inventario:**

Es quien tiene la información de cada producto existente en la ferretería, este agotado o no.

- nuevoProducto: crea un nuevo producto en el inventario con su respectivo nombre, precio, descripción y SKU
- modificarProducto: busca un producto ya existente en el inventario con el SKU y modifica su información
- eliminarInfoProducto: busca un producto ya existente en el inventario con el SKU y elimina información de este

#### **Producto:**

Interface que crea los productos con una información básica, como el nombre del producto, el precio, la descripción de su tamaño y el SKU identificador

- getNombre: consulta el nombre del producto
- getPrecio: consulta el precio del producto
- getDescripcion: consulta la descripción del producto
- getSKU: consulta el numero identificador del producto

**ProductoPeligroso:**

Corresponde a los productos que requieren restricción de edad e incluir precauciones para su uso, manipulación y transporte, además requieren verificación de si se han adquirido al menos una vez en el mismo mes

- Precauciones: corresponde a las precauciones que se deben incluir en la información del producto
- Restricciones: corresponde a la restricción de que debe ser manipulado por personas adultas
- cantidadAdquirida: Corresponde a la cantidad de productos adquiridos en ese mes por el cliente
- peligroso: añade los campos necesarios a la información del producto, para que al final cuente con nombre, precio, descripción, SKU, precauciones y restricción de edad, mientras que el atributo cantidad adquirida existe para verificar que no se compre mas de un mismo producto peligroso en el mes

**ProductoPerecedero:**

Corresponde a los productos que requieren tener en cuenta la fecha de caducidad, esta no puede ser menor de 60 días.

- Duración: corresponde al tiempo que tarda el producto en cumplir su fecha de caducidad
- fechaAdquisicion: corresponde a la fecha en que la ferretería adquirió el producto
- Perecedero: añade el campo necesario a la información del producto para que al final cuente con nombre, precio, descripción, SKU y su tiempo de caducidad

**Paquete:**

Corresponde a los paquetes creados por la ferretería para comprar varios productos en uno.

- Paquete: corresponde al paquete con su respectivo SKU y precio
- agregarItemPaquete: agrega productos para crear los paquetes dispuestos por la ferretería
- getSKU: consulta el SKU correspondiente al paquete
- getPrecio: consulta el precio correspondiente al paquete

**Compra:**

Es quien registra la información de cada compra realizada a nombre de la cedula del cliente, adjuntando valores de fecha de la compra, el medio de pago, el valor total de la compra y los productos que adquirió el cliente

- Compra: es el método que crea objetos de la clase compra, usando los parámetros antes mencionados
- getFecha: consulta la fecha del día en que se realiza la compra
- getMedioPago: consulta el medio de pago con el que se cancelan los productos
- getValorTotal: consulta la suma de los precios de todos los productos de la compra

- getProductos: consulta la lista de productos que el cliente adquirió

#### **Cliente:**

Busca si la cedula del cliente está registrada para asignar la nueva compra.

- cedulaCliente: corresponde al numero de cedula del cliente
- clienteExistente: verifica si existe el cliente ya está registrado
- historialCompras: de existir el cliente, consulta el historial de compras de este

#### **nuevoCliente:**

De no estar registrado el cliente se lo registra como nuevo cliente de la ferretería.

- NombreCliente: corresponde al nombre del cliente
- fechaNacimientoCliente: corresponde a la fecha de nacimiento del cliente
- emailCliente: corresponde al email del cliente
- numeroCliente: corresponde al número celular del cliente
- nuevoCliente: a partir de todos los parámetros anteriores se registra el nuevo cliente
- getNombre: consulta el nombre del nuevo cliente
- getFechaNacimiento: consulta la fecha de nacimiento del nuevo cliente
- getEmail: consulta el email del nuevo cliente
- getNumero: consulta el numero celular del nuevo cliente

### **Relaciones entre clases**

#### **Aplicación->ferretería:**

Conecta la consola que interactúa con el usuario con el modelo donde se realizan las operaciones

#### **Ferretería->inventario:**

Conecta la ferretería con el inventario de productos

#### **Inventario->producto:**

El inventario implementa la interfaz producto para manipular la información de cada producto

#### **Paquete, productoPerecedero, productoPeligroso -> producto:**

Son categorías de productos que se diferencian de los productos normales por alguna característica, heredan los métodos de producto puesto que también los utilizan (de 1 a \* productos dependiendo la información del inventario)

#### **Ferretería->paquete:**

La ferreteria proporciona los paquetes que serán creados en la clase Paquete (de 0 a \* paquetes dependiendo la cantidad de paquetes que cree la ferreteria)

#### **Ferreteria->compra:**

La ferreteria registra la nueva compra, 1 a la vez

#### **Compra->cliente:**

En el proceso de compra se consulta si el cliente ya existe y de ser así su historial de compras

#### **nuevoCliente->Cliente:**

De no existir el cliente, lo crea y hereda los atributos y métodos de cliente

**productos Peligrosos->Cliente:**

Es una relación dependiente, dado que dependiendo del número de productos peligrosos adquiridos por el cliente en el mismo mes, se puede o no tener la posibilidad de adquirir el producto