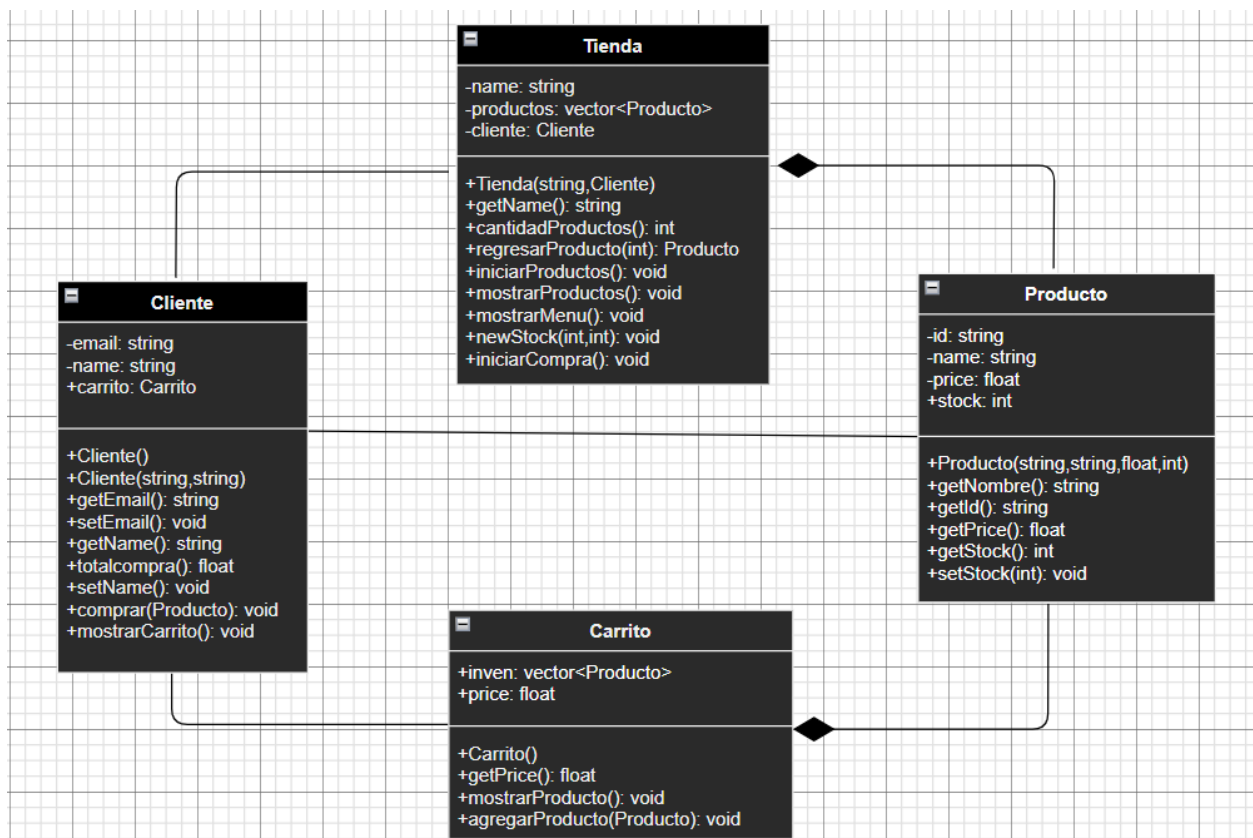


Andres Felipe Garcia Viña

A01800027

Evidencia 2. Solución Situación Problema

Para darle solución a la problemática presentada, se hizo uso de la POO (programación orientada a objetos) el análisis y el prototipo de programación seleccionado fue crear 4 clases una clase llamada Tienda la cual es la encargada de sintetizar toda la información de las demás clase, la clase Producto, la cual hace referencia a los productos que se venden en la tienda, Cliente el cual es el sujeto implicado en la compra de productos en el comercio electrónico que se diseñó, y por ultimo la clase carrito, la cual es la encargada de almacenar los productos que el cliente desea comprar.



Explicación de cada Atributo y cada método de las respectivas clases:

Producto:

Atributo	Tipo dato	explicación
id	String (privado)	Representa el id del producto
name	String (privado)	Representa el nombre del producto
price	Float (privado)	Representa el precio del Producto
stock	Int (publico)	Representa la cantidad disponible de un Producto

Metodo	Tipo retorno	Explicacion
Producto(string,string,float,int)		Constructor de la clase
getNombre()	string	Metodo que retorna el nombre del producto
getId()	String	Metodo que retorna el ID del producto
getPrice()	Float	Metodo que retorna el precio del Producto
getStock()	Int	Metodo que retorna el Stock del Producto
setStock(int)	Void	Metodo que nos permite modificar el Stock del producto

Carrito:

Atributo	Tipo dato	Explicacion
inven	vector<Producto>	Inven es un vector de instancias de la clase producto simula el almacenamiento de productos en un carrito
price	Float	El precio total de los producto ingresados en el carrito

Metodo	Tipo retorno	Explicacion
Carrito()		Constructor
getPrice()	Float	Retorna el precio total del carrito
mostrarProducto()	Void	Muestro todos los producto que hay en el carrito con su descripción de atributos
agregarProducto(Producto)	void	Agrega un producto a inven

Cliente:

Atributo	Tipo dato	Explicacion
Email	string	Email del cliente
name	String	Nombre del cliente
carrito	Carrito	Una instancia de la clase carrito asociada a cliente

Metodo	Tipo retorno	Explicacion
Cliente()		Constructor
Cliente(string,string)		Constructor
getEmail()	String	Retorna el email del cliente
setEmail(string)	Void	establece el email del cliente
getName()	String	Retorna el nombre del cliente
setName(string)	Void	establece el nombre del cliente
totalCompra()	float	Retorna el precio total de los productos ingresados al carrito
Comprar(Producto)	Void	Ingresa un producto al carrito
mostrarCarrito()	Void	Muestra todos los productos que tiene el carrito

Tienda:

Atributo	Tipo dato	Explicación
name	String	El nombre de la tienda
Productos	vector<Producto>	Simula el inventario de una tienda, es un vector que almacena instancias de la clase producto
cliente	Cliente	Le asocia una instancia de la clase Cliente

Metodo	Tipo retorno	Explicacion
Tienda(string,Cliente)		Constructor
getName()	String	Regresa el nombre de la tienda
cantidadProductos()	Int	Regresa la cantidad de productos que tiene la tienda
regresarProducto(int)	Producto	Regresa un producto en la posición seleccionada de el vector de productos
iniciarProducto()	Void	Agrega los productos de un archivo.csv a el vector de productos

mostrarProductos()	Void	Muestra todos los productos disponibles en la tienda
newStock(int,int)	Void	Disminuye el stock de un objeto dependiendo de la compra realizada
iniciarCompra()	Void	Inicia toda la interacción de la tienda

Cuando en el menú principal seleccionamos la opción 1. La tienda desplegará todos los Productos que esta contiene, con su respectivo precio y la cantidad de este producto que esta disponible.

```
antes de iniciar su compra ingrese su nombre: Felipe
```

```
ingrese su correo: demo1@demo.com
```

```
***** MENU PRINCIPAL *****
```

```
1) Seleccionar producto
```

```
2) Ver carrito de compras
```

```
3) Terminar compra
```

```
4) Salir
```

```
Selecciona una opcion: 1
```

```
Lista de productos:
```

```
-----
0: Agua Embotellada Precio: 15.54 cantidad disponible: 25
```

```
-----
1: Bote de Palomitas Precio: 25.65 cantidad disponible: 125
```

```
-----
2: ICE de cereza Precio: 40.23 cantidad disponible: 50
```

```
-----
Que producto deseas comprar (0-2) ? (-1 para cancelar): █
```

Si el producto seleccionado no se encuentra en el rango establecido. Este desplegará un mensaje indicando que por favor seleccione una opción correcta y volverá a preguntar que objeto desea comprar.

```
Que producto deseas comprar (0-2) ? (-1 para cancelar): 3
```

```
Selecciona una opcion correcta.
```

```
Lista de productos:
```

```
-----
0: Agua Embotellada Precio: 15.54 cantidad disponible: 25
```

```
-----
1: Bote de Palomitas Precio: 25.65 cantidad disponible: 125
```

```
-----
2: ICE de cereza Precio: 40.23 cantidad disponible: 50
```

```
-----
Que producto deseas comprar (0-2) ? (-1 para cancelar): █
```

Ya después de haber seleccionado una opción correcta, el programa solicitara la cantidad de este producto que el usuario desea comprar. Si no hay stock suficiente para realizar la compra desplegara el mensaje “No hay Stock disponible para la venta”.

```
Que producto deseas comprar (0-2) ? (-1 para cancelar): 1
Cuantas unidades deseas adquirir?: 150

No hay Stock disponible para la venta.
Cuantas unidades deseas adquirir?:
```

Después de realizada la compra de este producto, este aparecerá en el carrito:

```
Cuantas unidades deseas adquirir?: 120

***** MENU PRINCIPAL *****
1) Seleccionar producto
2) Ver carrito de compras
3) Terminar compra
4) Salir
Selecciona una opcion: 2
-----Producto: 1
id: ID0002
name: Bote de Palomitas
precio Unitario: 25.65
Cantidad a llevar: 120
***** MENU PRINCIPAL *****
1) Seleccionar producto
2) Ver carrito de compras
3) Terminar compra
4) Salir
Selecciona una opcion:
```

Después de agregar varios productos el carrito se vera así:

```
-----Producto: 1
id: ID0002
name: Bote de Palomitas
precio Unitario: 25.65
Cantidad a llevar: 120
-----Producto: 2
id: ID0001
name: Agua Embotellada
precio Unitario: 15.54
Cantidad a llevar: 17
-----Producto: 3
id: ID0003
name: ICE de cereza
precio Unitario: 40.23
Cantidad a llevar: 14
```

En caso de no haber agregado ningún producto al carrito, y seleccionar la opción 2 (ver carrito) se desplegara el siguiente mensaje:

```
***** MENU PRINCIPAL *****
1) Seleccionar producto
2) Ver carrito de compras
3) Terminar compra
4) Salir
Selecciona una opcion: 2

No se han agregado productos.

***** MENU PRINCIPAL *****
1) Seleccionar producto
2) Ver carrito de compras
3) Terminar compra
4) Salir
Selecciona una opcion: █
```

La 3 opcion (Terminar compra), mostrara el total a pagar y un pequeño mensaje de agradecimiento, para la siguiente compra:

```
***** MENU PRINCIPAL *****
1) Seleccionar producto
2) Ver carrito de compras
3) Terminar compra
4) Salir
Selecciona una opcion: 2

-----Producto: 1
id: ID0002
name: Bote de Palomitas
precio Unitario: 25.65
Cantidad a llevar: 20
-----Producto: 2
id: ID0001
name: Agua Embotellada
precio Unitario: 15.54
Cantidad a llevar: 15
-----Producto: 3
id: ID0003
name: ICE de cereza
precio Unitario: 40.23
Cantidad a llevar: 4
```

El valor a pagar fue el siguiente $((25.65*20)+(15.54*15)+(40.23*4)) = 907.02$

Comprobamos con el programa:

```
***** MENU PRINCIPAL *****
1) Seleccionar producto
2) Ver carrito de compras
3) Terminar compra
4) Salir
Selecciona una opcion: 3
Felipe El total de tu compra es: 907.02
Gracias por comprar en: Mitiendita.com!
su ticket sera enviado a: user1@demo.com
```

La opción 4 del menú principal termina el programa.

```
***** MENU PRINCIPAL *****
1) Seleccionar producto
2) Ver carrito de compras
3) Terminar compra
4) Salir
Selecciona una opcion: 4
@A01800027 →/workspaces/intregrador-A01800027 (main) $
```