



Objetivo:

- Consolidar los conocimientos adquiridos en clase sobre Java.

Enunciado:

En un programa de ordenador, las facturas tienen necesariamente un conjunto de datos de los productos de los cuales comprar y un conjunto de datos del cliente, un total (valor decimal) y una fecha de la factura.

Los datos del cliente son la cadena de caracteres nombre, cedula, y el entero fiabilidad de pago, mientras que los datos del detalle de la factura son sólo su producto, cantidad, valor.

Cada clase tendrá los métodos para leer y fijar (“set” y “get”) todos sus atributos.

También se debe incluir en la clase Factura un método de “Borrado”, que no devuelve ni recibe ningún parámetro.

Los productos que venden la empresa son de carácter de computadores y tiene un id, descripción, precio unitario, stock, iva (si se calcula el iva o no).

Los atributos serán privados y tendrán métodos públicos para acceder a ellos. Se pide dibujar el diagrama UML de las clases Factura, Datos_del_cliente y Datos_del_producto y generar el sistema en java para realizar los procesos de CRUD para generar facturas, productos, clientes.

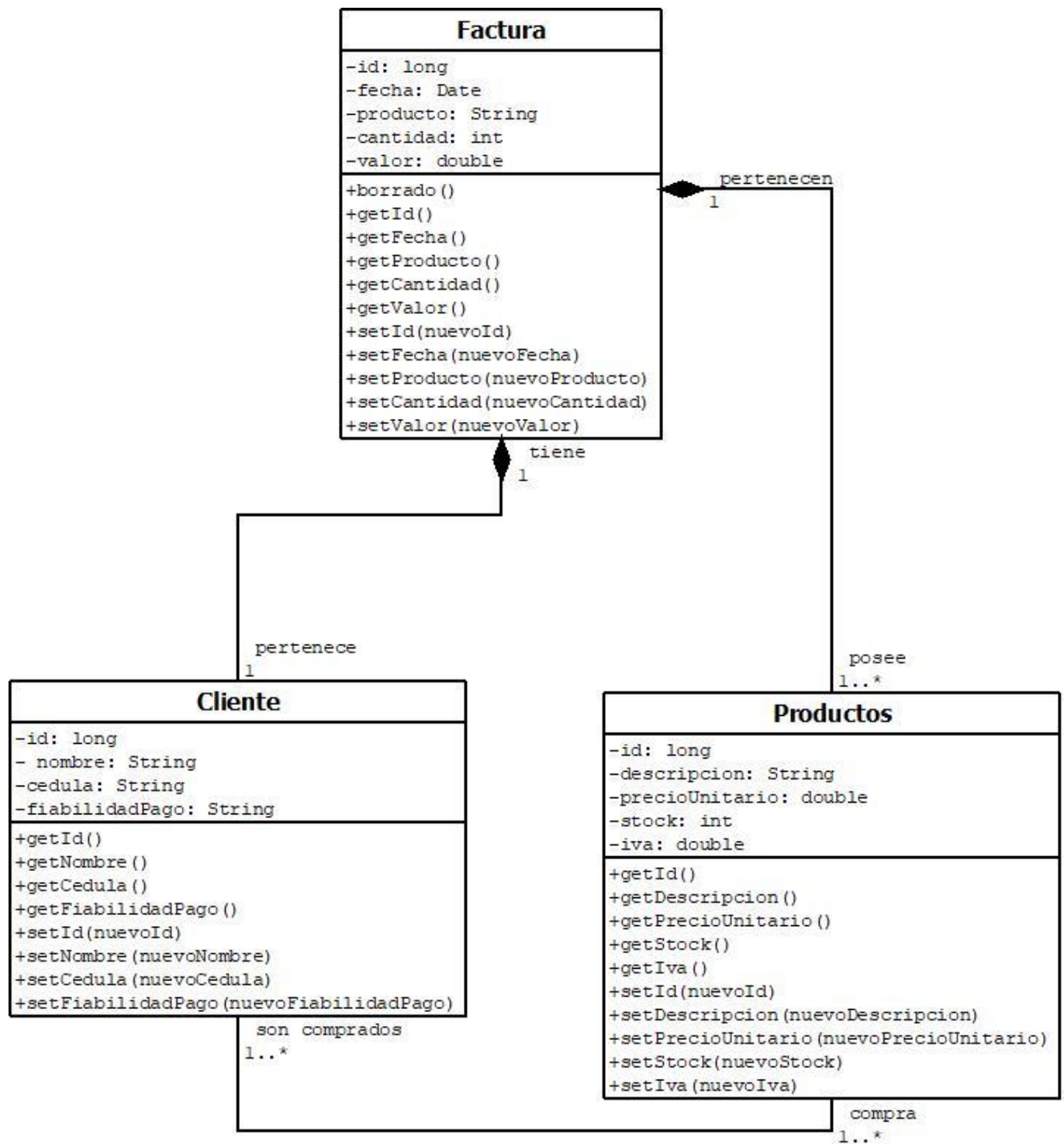
Se calificará el avance con los siguientes criterios de evaluación:

- Diagrama de Clases: 25%
- MVC: 25%
- Usabilidad – Vista - Menús: 25%
- Ejecución, pruebas y sustentación 25%

Entrega: Subir al Git el documento en formato PDF de la practica y código hasta las **23:55** del domingo **06 de Junio del 2021**.



Diagrama UML





Programación Orientada Objetos

Tema: Clases y Sistemas en Java .

Examen

06/03/2021

ExamenFinal - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

357,3/662,0MB

Proje... x Files Services ...or.java

Source History

```
1  /*
2  * To c
3  * To c
4  * and
5  */
6  package
7
8  import
9  import
10 import
11 import
12
13 /**
14 *
15 * @aut
16 */
17 public
18 pri
19 pri
20
21 puk
22 lis
23 sel
24 }
25 // CRUI
26 public
27 Cliente
28 return
29 }
30
31 public
```

Find:

Output - ExamenFinal (run) #13

```
ant -f "C:\Users\GAMER I7 10TH GEN\Documents\NetBeansProjects\Exame
init:
Deleting: C:\Users\GAMER I7 10TH GEN\Documents\NetBeansProjects\ExamenFi
deps-jar:
Updating property file: C:\Users\GAMER I7 10TH GEN\Documents\NetBeansPro
compile:
run:
Seleccione una opción
1. Factura
2. Cliente
3. Producto
4. Salir
1

Administrar la Factura
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
1

Ingrese los siguientes datos:
Id:
1
Fecha: dia/mes/año
6/6/2021
Producto:
Hp
Cantidad:
1
Valor:
1250,75
Creado= true

Administrar la Factura
1. Crear
```

Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

441,0/701,0MB

Proje... x Files Services

Output - ExamenFinal (run) #13

```
5
Factura(id=1, fecha=6/6/2021, producto=Hp, cantidad=1, valor=1250.75)
Factura(id=2, fecha=7/6/2021, producto=Dell, cantidad=2, valor=3100.5)
Factura(id=3, fecha=5/5/2021, producto=Macbook, cantidad=1, valor=2000.0)
Factura(id=4, fecha=8/11/2020, producto=Acer, cantidad=3, valor=4500.79)
Factura(id=5, fecha=14/11/2020, producto=Asus, cantidad=4, valor=6733.54)

Administrar la Factura
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
2
Producto
Acer
Producto Nuevo
Alienware
Actualizado= true

Administrar la Factura
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
5
Factura(id=1, fecha=6/6/2021, producto=Hp, cantidad=1, valor=1250.75)
Factura(id=2, fecha=7/6/2021, producto=Dell, cantidad=2, valor=3100.5)
Factura(id=3, fecha=5/5/2021, producto=Macbook, cantidad=1, valor=2000.0)
Factura(id=4, fecha=8/11/2020, producto=Alienware, cantidad=3, valor=4500.79)
Factura(id=5, fecha=14/11/2020, producto=Asus, cantidad=4, valor=6733.54)

Administrar la Factura
1. Crear
```



Programación Orientada Objetos

Tema: Clases y Sistemas en Java .

Examen

06/03/2021

```
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
1
INGRESE LOS DATOS QUE SE REQUIERAN A CONTINUACIÓN:
Id:
1
Nombre:
Carlos
Cedula:
0302885306
Fiabilidad de Pago:
Tarjeta
Cliente creado: true

Administración de Clientes
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
5
Listado de clientes:
Cliente(id=1, nombre=Carlos, cedula=0302885306, fiabilidadPago=Tarjeta, factura=Factura{id=3, fecha=5/5/2021, producto=Macbook, cantidad=1, valor=2000.

Administración de Clientes
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
```

```
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
1
Crear-Ingresa datos:
Id
1
Descripción
Macbook-i5
Precio Unitario
2000
Stock
4
Iva
0,0
Creado= true

Administar los Productos
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
5
Producto(id=1, descripcion=Macbook-i5, precioUnitario=2000.0, stock=4, iva=0.0, comprador=null)

Administar los Productos
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
```



SUSTENTACIÓN

Este proyecto de Programación Orientada a Objetos fue llevado a cabo a través del ejercicio planteado por el docente; por lo cual en primera instancia se procedió a la elaboración del diagrama UML, en el cual se incluyó la información proporcionada en el ejercicio tal como la creación de 3 clases: Factura, Cliente y Productos. Para ello se extrajo los atributos y métodos que se requerían para su elaboración.

Una vez elaborado el diagrama UML se lo implementó en JAVA mediante MVC, se tomó en cuenta las indicaciones expuestas por el docente, entonces en la parte de modelo se crearon las diferentes clases con sus respectivos atributos y métodos. Luego se elaboraron los controladores teniendo en cuenta el CRUD que se requería usar para que de tal manera reciba, atienda y procese la información que se implementará en el código. Finalmente se creó la interfaz gráfica que se le presentará al usuario, logrando mostrar así un menú en el cual la persona pueda interactuar entre las diferentes opciones, además se le indicará que información deberá introducir el usuario para la creación de su factura.