



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.

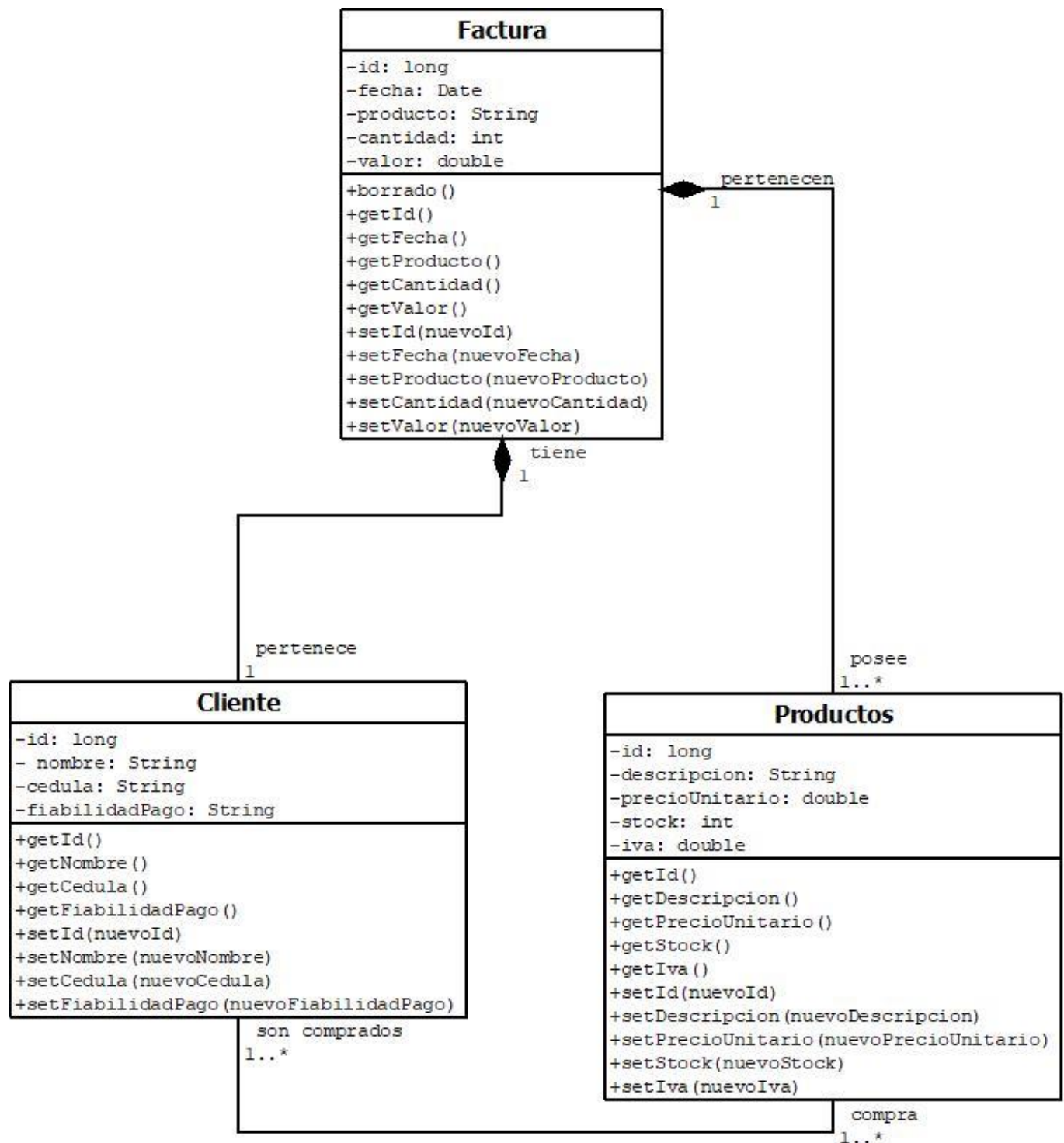
Se calificará el avance con los siguientes criterios de evaluación:

- 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

The screenshot shows the Apache NetBeans IDE 12.0 interface. The left sidebar displays the project structure for 'ExamenFinal', including packages like 'Controlador', 'Modelo', and 'Vista'. The main editor window shows the source code of 'ExamenFinal.java'. The code includes a menu system with options like 'Eliminar', 'Listar/Imprimir', 'Salir', and 'Ingresar los siguientes datos:'. The output window on the right shows the execution results, including the menu options and the data entered for a product (Dell) with a value of 2500.5.

```
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64
```

Output:

```
4. Eliminar  
5. Listar/Imprimir  
6. Salir  
1  
Ingresar los siguientes datos:  
Id:  
1  
Fecha: dia/mes/año  
6/6/2021  
Producto:  
Dell  
Cantidad:  
1  
Valor:  
2500,50  
Creador= 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=Dell, cantidad=1, valor=2500.5)  
Administrar la Factura  
1. Crear  
2. Actualizar  
3. Buscar/Leer  
4. Eliminar  
5. Listar/Imprimir  
6. Salir
```

The screenshot shows the Apache NetBeans IDE 12.0 interface. The left sidebar displays the project structure for 'ExamenFinal', including packages like 'Controlador', 'Modelo', and 'Vista'. The main editor window shows the source code of 'ExamenFinal.java'. The code includes a menu system with options like 'Eliminar', 'Listar/Imprimir', 'Salir', and 'Ingresar los siguientes datos:'. The output window on the right shows the execution results, including the menu options and the data entered for a product (Acer) with a value of 3700.0.

```
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64
```

Output:

```
1  
Ingresar los siguientes datos:  
Id:  
4  
Fecha: dia/mes/año  
18/4/2021  
Producto:  
Acer  
Cantidad:  
2  
Valor:  
3700  
Creador= 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=Dell, cantidad=1, valor=2500.5)  
Factura(id=2, fecha=7/6/2021, producto=Hp, cantidad=2, valor=3450.0)  
Factura(id=3, fecha=7/5/2021, producto=Asus, cantidad=1, valor=1900.0)  
Factura(id=4, fecha=18/4/2021, producto=Acer, cantidad=2, valor=3700.0)  
Administrar la Factura  
1. Crear  
2. Actualizar  
3. Buscar/Leer  
4. Eliminar  
5. Listar/Imprimir  
6. Salir
```



# Programación Orientada Objetos

## Tema: Clases y Sistemas en Java .

### Examen

06/03/2021

The screenshot shows the Apache NetBeans IDE with the project 'ExamFinal'. The source code in 'ExamFinal.java' includes a menu for 'Administrar la Factura' with options: 1. Crear, 2. Actualizar, 3. Buscar/Leer, 4. Eliminar, 5. Listar/Imprimir, 6. Salir. The output window shows the execution results for 'ExamFinal (run) #12' and '#13', displaying facturas with details like id, fecha, producto, cantidad, and valor.

The screenshot shows the Apache NetBeans IDE with the project 'ExamFinal'. The source code in 'ExamFinal.java' includes a menu for 'Administración de Clientes' with options: 1. Crear, 2. Actualizar, 3. Buscar/Leer, 4. Eliminar, 5. Listar/Imprimir, 6. Salir. The output window shows the execution results for 'ExamFinal (run) #13', displaying a list of clients with details like id, nombre, apellido, cedula, and fiabilidadPago.



# 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
<default config> 340.8/615.0MB

Output - ExamenFinal (run) #13
5
Listado de clientes:
Cliente(id=1, nombre=Carlos, apellido=Saico, cedula=0302885307, fiabilidadPago=Tarjeta, factura=Factura(id=1, fecha=6/6/2021, producto=Dell, cantidad=1
Cliente(id=2, nombre=Pepito, apellido=Perez, cedula=0301410216, fiabilidadPago=Efectivo, factura=Factura(id=2, fecha=7/6/2021, producto=Hp, cantidad=2,
Cliente(id=3, nombre=Franklin, apellido=Fernandes, cedula=03092998, fiabilidadPago=Tarjeta, factura=Factura(id=3, fecha=7/5/2021, producto=Asus, cantid

Administración de Clientes
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
4
Eliminar Cliente
Cedula
0301410216
Cliente eliminado: 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, apellido=Saico, cedula=0302885307, fiabilidadPago=Tarjeta, factura=Factura(id=1, fecha=6/6/2021, producto=Dell, cantidad=1
Cliente(id=3, nombre=Franklin, apellido=Fernandez, cedula=03092998, fiabilidadPago=Tarjeta, factura=Factura(id=3, fecha=7/5/2021, producto=Asus, cantid

Administración de Clientes
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
```

```
ExamenFinal - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 588.3/615.0MB

Output - ExamenFinal (run) #13
Ingreso Descripcion
Computador-H-15
null

Administrar los Productos
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
3
Ingreso Descripcion
Computador-Hp-15
Producto(id=2, descripcion=Computador-Hp-15, precioUnitario=1700.0, stock=3, iva=0.0, comprador=null)

Administrar los Productos
1. Crear
2. Actualizar
3. Buscar/Leer
4. Eliminar
5. Listar/Imprimir
6. Salir
5
Producto(id=1, descripcion=Computador-Dell-17, precioUnitario=2500.5, stock=6, iva=75.0, comprador=null)
Producto(id=2, descripcion=Computador-Hp-15, precioUnitario=1700.0, stock=3, iva=0.0, comprador=null)
Producto(id=3, descripcion=Computador-Asus-19, precioUnitario=1900.0, stock=5, iva=250.0, comprador=null)

Administrar 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.