



**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.

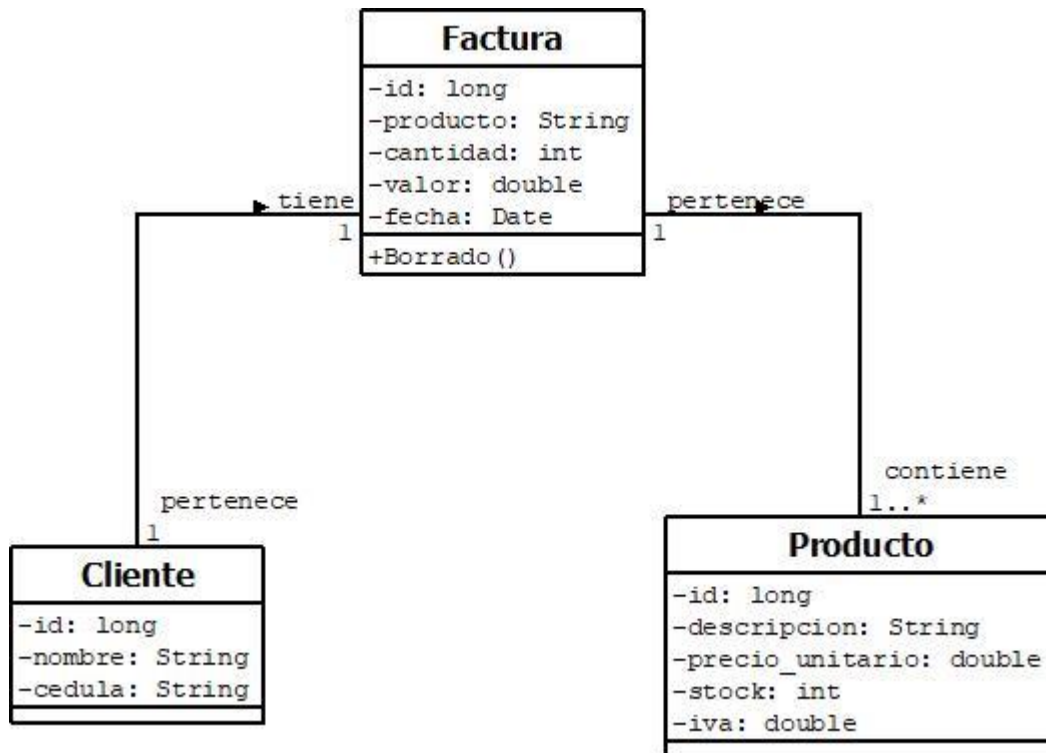
Se calificará el avance con los siguientes criterios de evaluación:

- 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 práctica y código hasta las **23:55** del domingo **06 de Junio del 2021**.



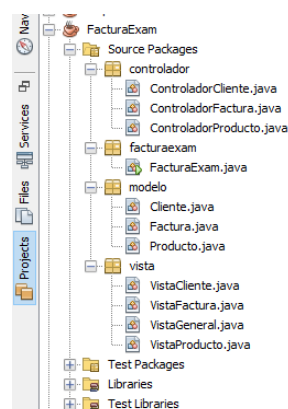
- DIAGRAMA DE CLASES (con el software dia)



El cliente tiene una factura, la factura pertenece a un solo cliente. Los productos perteneces a una factura y la factura contiene uno o más productos.

- PROGRAMA EN JAVA (con NetBeans)

Se divide el programa en paquetes, primero el paquete modelo que contiene las tres clases, seguido del paquete controlador y por último el paquete vista.





El programa se encuentra en el git, a continuación, se mostrará capturas del funcionamiento. Primero el menú principal "Gestión de Tienda", si elegimos Factura y no existe bases de datos de clientes o productos nos envía directamente a los menús respectivos para crear ya sea productos o clientes. Se creo un cliente y se grabo en la base de datos.

```
Output - FacturaExam (run)
Gestion de la Tienda:
1. Productos
2. Clientes
3. Facturas
4. Salir
3
No existe Clientes en la base de Datos
Menu de Clientes:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese nombre del nuevo cliente:
Santiago Berrezueta
Ingrese cedula del nuevo cliente:
0106276256
Res:true
Menu de Clientes:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
|
```

A continuación, grabamos dos productos para la tienda manzanas y peras. Se listan al final para que se confirme.

```
Output - FacturaExam (run)
Gestion de la Tienda:
1. Productos
2. Clientes
3. Facturas
4. Salir
1
Menu de Productos:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese descripcion del nuevo producto:
Manzana
Ingrese precio unitario del nuevo producto:
0,2
Ingrese stock del nuevo producto:
100
Res:true
Menu de Productos:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
```



```
Output - FacturaExam (run)
Menu de Productos:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese descripcion del nuevo producto:
Pera
Ingrese precio unitario del nuevo producto:
0,5
Ingrese stock del nuevo producto:
150
Res:true
Menu de Productos:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
5
Producto(id=1, descripcion=Manzana, precio_uni=0.2, stock=100)
Producto(id=2, descripcion=Pera, precio_uni=0.5, stock=150)
Menu de Productos:
1. Crear
2. Actualizar
```

Con los clientes y productos en las bases de datos, ya se puede crear una factura. Al final listamos la factura para revisar que los datos se guardaran correctamente. Al inicio nos pide el numero de productos diferentes que se compraran en este caso dos (manzana, pera), nos piden ingresar la cedula de un cliente en la base de datos, seguido por el producto con los detalles del IVA, si tiene o no y la cantidad que se comprara del producto para mermar del stock y para hacer los cálculos. Al final nos pide poner fecha a la factura en base al formato, para terminar con listar para comprobar (los caracteres son largos por lo que se dividirá en varias imágenes).

```
Output - FacturaExam (run)
Menu de Facturas:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese cuantos productos diferentes compra:
2
Ingrese la cedula del cliente nuevo:
0106276256
Ingrese la descripcion del producto a comprar:
Manzana
El producto a comprar tiene iva (si/no):
si
Ingrese la cantidad del producto a comprar:
20
```

```
Output - FacturaExam (run)
Ingrese la descripcion del producto a comprar:
Pera
El producto a comprar tiene iva (si/no):
no
Ingrese la cantidad del producto a comprar:
15
Ingresar la Fecha de la nueva Factura con formato (dd/mm/yyyy):
22/07/2020
Res:true
```



```
Menu de Facturas:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
5
Factura{id=1, fecha=Wed Jan 22 00:07:00 COT 2020,
```

```
, listaProductosClie=[Producto{id=1, descripcion=Manzana, precio_uni=0.2, stock=80}, Producto{id=2, descripcion=Pera, precio_uni=0.5, stock=135}],
```

```
, valor=7.5, cliente=Cliente{id=1, nombre=Santiago Berrezueta, cedula=0106276256}]}
```

En los menús nos permiten eliminar actualizar, listar y eliminar, cualquiera de las clases ya sea los clientes, los productos o las facturas.