



Programación Orientada Objetos

Tema: Clases y Sistemas en Java.



Examen

CARRERA: Computación

NOMBRE: Erika Contreras

OBJETIVO:

- Consolidar los conocimientos adquiridos en clase sobre Java.
- Desarrollo de una aplicación utilizando el paradigma de programación orientado a objetos.

OBJETIVO ALCANZADO:

- Abstractar objetos del mundo real y los modelarlos mediante diagramas de clase.
- Construir clases utilizando el paradigma de programación orientada a objetos estudiados en la unidad.

ACTIVIDADES A DESARROLLAR

- 1) Se pide dibujar el diagrama UML de las clases Factura, Datos_del_cliente y Datos_del_producto.

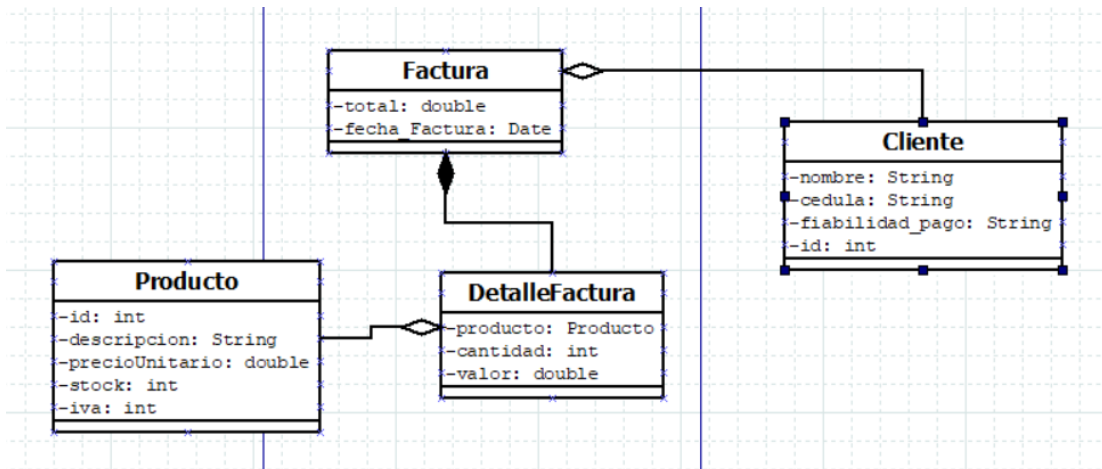


Figura 1: Diagrama de clases para el sistema de Facturas

- 2) Generar en el sistema de java los procesos de CRUD para generar facturas, productos, clientes. Siguiendo las siguientes instrucciones.

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,



Programación Orientada Objetos

Tema: Clases y Sistemas en Java.

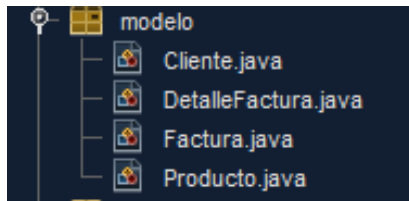


Examen

precio unitario, stock, iva (si se calcula el iva o no).

Los atributos serán privados y tendrán métodos públicos para acceder a ellos.

- 3) Creación de un proyecto de Java en Netbeans. Creación del paquete modelo con sus respectivas clases.



- 4) Creación del paquete modelo y de clase persona con sus respectivos atributos, getters y setter:

- Se procede a realizar la relación de la clase Cliente a la clase factura.

```
public class Cliente {  
    private String nombre;  
    private String apellido;  
    private String cedula;  
    private String fiabilidad_Pago;  
    private int id;  
  
    //Constructor con todos los atributos  
    public Cliente(String nombre, String apellido, String cedula,  
        this.nombre = nombre;  
        this.apellido = apellido;  
        this.cedula = cedula;  
        this.fiabilidad_Pago = fiabilidad_Pago;  
        this.id = id;  
    }  
  
    //Métodos Getters y Setters  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getApellido() {  
        return apellido;  
    }  
  
    public void setApellido(String apellido) {  
        this.apellido = apellido;  
    }  
}
```

```
public class Factura {  
    private double total;  
    private Date fecha_Factura;  
    private Cliente cliente;  
    private int id;  
  
    //Atributo de agregacion  
    private List<DetalleFactura> facturas;  
  
    //Constructor  
    public Factura() {  
        facturas= new ArrayList<>();  
    }  
  
    // Constructor  
    public Factura(double total, Date fecha_Factura,int id,i  
        this.total = total;  
        this.fecha_Factura = fecha_Factura;  
        this.id=id;  
    }  
  
    // Metodo Borrado  
    public static void borrado(){  
    }  
}
```



Programación Orientada Objetos

Tema: Clases y Sistemas en Java.



Examen

- Se procede a realizar la relación de AGREGACION de la clase Producto a la clase DetalleFactura.

```
*/
public class Producto {
    private int id;
    private String descripcion;
    private double precioUnitario;
    private int stock;
    private double iva;

    // Creación de un constructor con todos los atributos
    public Producto(int id, String descripcion, double precioUnitario) {
        this.id = id;
        this.descripcion = descripcion;
        this.precioUnitario = precioUnitario;
        this.stock = stock;
        this.iva = iva;
    }

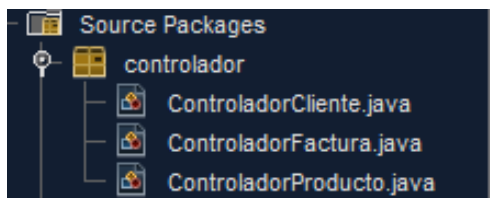
    // Metodos Getters y Setters
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDescripcion() {
        return descripcion;
    }
}
```

```
4  */
5  public class DetalleFactura {
6      private int cantidad;
7      private double valor;
8      private Producto producto;
9
10     //Metodos Getters y Setters
11     public int getCantidad() {
12         return cantidad;
13     }
14
15     public void setCantidad(int cantidad) {
16         this.cantidad = cantidad;
17     }
18
19     public double getValor() {
20         return valor;
21     }
22
23     public void setValor(double valor) {
24         this.valor = valor;
25     }
26
27     public Producto getProducto() {
28         return producto;
29     }
30
31     public void setProducto(Producto producto) {
32         this.producto = producto;
33     }
}
```

5) Creación del paquete Controlador.



6) Se instancian los métodos CRUD en cada clase del paquete controlador.



Programación Orientada Objetos

Tema: Clases y Sistemas en Java.



Examen

```
//Metodos CRUD
public boolean crear( String descripcion, double precioUnitario,
    Producto datosproductos = new Producto(this.generarId(),desc
    return listaDatoProducto.add(datosproductos);
}

public Producto buscar( String descripcion){
    for (Producto datosproductos : listaDatoProducto) {
        if(datosproductos.getDescripcion().equals(descripcion))
            return datosproductos;
    }
    return null;
}

public boolean actualizar( String descripcion, double precioUnit
    Producto datosproductos = this.buscar(descripcion);
    if(datosproductos!= null){
        int posicion = listaDatoProducto.indexOf(datosproductos)
        datosproductos.setDescripcion(descripcion);
        datosproductos.setPrecioUnitario(precioUnitario);
        datosproductos.setStock(stock);
        datosproductos.setIva(iva);
        listaDatoProducto.set(posicion, datosproductos);
        return true;
    }
    return false;
}

public boolean eliminar(String id){
    Producto datosproductos = this.buscar(id);
    if(datosproductos != null)
        return listaDatoProducto.remove(datosproductos);
    return false;
}

public List<Producto> getListDatosProductos() {
    return listaDatoProducto;
}
```

- En el ControladorFactura se crean los metodos CRUD Tanto de La clase Factura como de DetalleFactura, debido a que se realizo una relacion de composición.

```
public class ControladorFactura {

    private List<Factura> listaFactura;
    private Factura seleccionado;

    public ControladorFactura(){
        listaFactura=new ArrayList();
    }

    public int generarId(){
        return(listaFactura.size()>0)? listaFactura.get(listaFactura.size()-1).getI
    }

    //Metodos CRUD
    public boolean crear( double total, Date fecha_Factura,int cantidad, double
        Factura facturas = new Factura(total,fecha_Factura,this.generarId(),ca
        return listaFactura.add(facturas);
    }

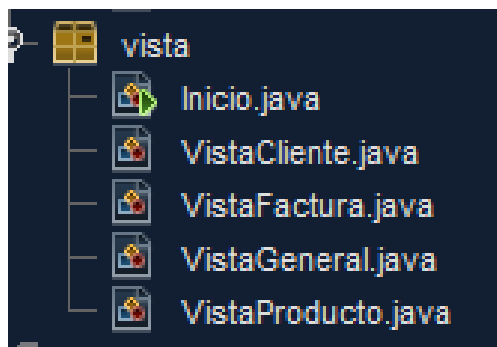
    public Factura buscar( double total ){
        for (Factura facturas : listaFactura) {
            if(facturas.getTotal()==(total)){
                seleccionado=facturas;
            }
            return facturas;
        }
        return null;
    }

    public boolean actualizar( double total,Date fecha_Factura){
        Factura facturas = this.buscar(total);
        if(facturas!= null){
            int posicion = listaFactura.indexOf(total);
            facturas.setFecha_Factura(fecha_Factura);
            listaFactura.set(posicion, facturas);

            return true;
        }
        return false;
    }

    public boolean eliminar(double total){
        Factura facturas = this.buscar(total);
        if(facturas != null)
            return listaFactura.remove(facturas);
        return false;
    }
}
```

7) Creación del paquete Vista.





Programación Orientada Objetos

Tema: Clases y Sistemas en Java.



Examen

- En las tres vistas tanto de Cliente, Factura y Producto, presenta los datos de paquete modelo y lo que verá el usuario.
- Vista General: Se procede a imprimir el código como tal mediante la clase Inicio.

```
/**
 *
 * @author Erika
 */
public class VistaGeneral {

    VistaCliente vistaCliente = new VistaCliente();
    VistaProducto vistaProducto = new VistaProducto();
    VistaFactura vistaFactura = new VistaFactura();

    private Scanner s;
    private ControladorCliente controladorCliente;
    private ControladorProducto controladorProducto;
    private ControladorFactura controladorFactura;

    public VistaGeneral() {
        s = new Scanner(System.in);
        controladorCliente = new ControladorCliente();
        controladorProducto = new ControladorProducto();
        controladorFactura = new ControladorFactura();
    }

    public void menu()
    {
        int op=1;
        while(op>0)
        {
            System.out.println("***** Sistema de Facturación ***** ");
            System.out.println("\n1.Ingrese el CLIENTE \n2.Ingrese el PRODUCTO");
            op=s.nextInt();
            if(op==1)
            {
                vistaCliente.menu();
            }
            else if(op==2)
            {
                vistaProducto.menu();
            }
            else if (op==3)
            {
                vistaFactura.menu();
            }
        }
    }
}
```

```
/**
 *
 * @author Erika
 */
package vista;

public class Inicio {

    public static void main(String[] args) {
        /* VistaFactura factura=new VistaFactura();
        factura.menu() */
        VistaGeneral vg= new VistaGeneral();
        vg.menu();
    }
}
```

8) Salida del presente código.



Programación Orientada Objetos

Tema: Clases y Sistemas en Java.



Examen

Cliente

```
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
1
Ingrese el nombre
Ana
Ingrese el apellido
Lopez
Ingrese la cedula
0106333966
Ingrese la fiabilidad de pago
excelente
Resultado: true
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
5
Factura0Cliente{nombre=Ana, apellido=Lopez, cedula=0106333966, fiabilidad_Pago=excelente, id=1}
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
```

Producto

```
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
1
Ingrese la descripcion
pollo
Ingrese el precio unitario
5
Ingrese el stock
10
Ingrese el IVA
12
Resultado: true
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
5
Factura0Producto{id=1, descripcion=pollo, precioUnitario=5.0, stock=10, cantidad=, iva=12.0}
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
```



Programación Orientada Objetos

Tema: Clases y Sistemas en Java.



Examen

Facturación.

```
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
1
Ingrese el total
15
Ingrese la fecha ("dd/mm/yyyy"):
20/05/2021
Ingrese la cantidad
5
Ingrese el valor
3
Resultado: true
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
0
***** Sistema de Facturación *****

1.Ingrese el CLIENTE
2.ingrese el PRODUCTO
3.ingrese la FACTURA
0 Salir
|
```