



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**.



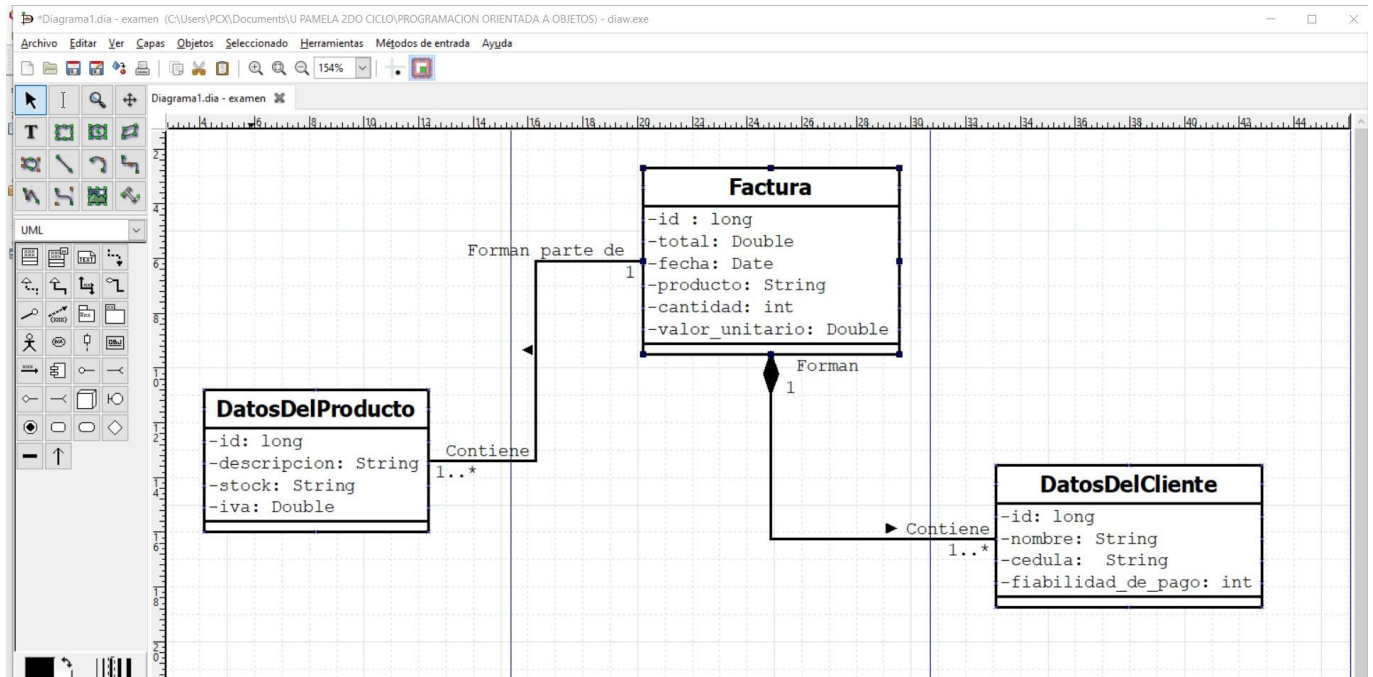
Programación Orientada Objetos

Tema: Clases y Sistemas en Java .

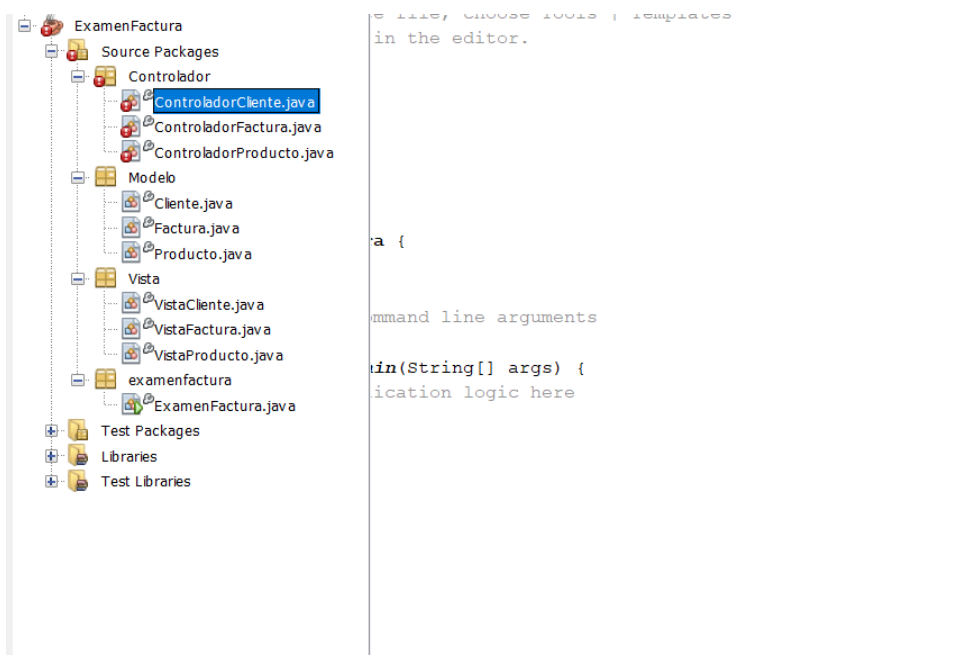
Examen

06/03/2021

DIAGRAMA UML



MVC





Programación Orientada Objetos

Tema: Clases y Sistemas en Java .

Examen

06/03/2021

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 24.7.6/373.0MB

Files Projects Services
Burbuja
BusquedaSecuencial
CobroParquadero
EmpresaDirectorio
ExamenFactura
Source Packages
Controlador
Modelo
  Cliente.java
  Factura.java
  Producto.java
Vista
examenfactura
Test Packages
Libraries
Test Libraries
GUI
SerieFibonacci

Source History
13 private long id;
14 private String descripcion;
15 private String sotck;
16
17 public Producto(long id, String descripcion, String sotck) {
18     this.id = id;
19     this.descripcion = descripcion;
20     this.sotck = sotck;
21 }
22
23 public long getId() {
24     return id;
25 }
26
27 public void setId(long id) {
28     this.id = id;
29 }
30
31 public String getDescripcion() {
32     return descripcion;
33 }
34
35 public void setDescripcion(String descripcion) {
36     this.descripcion = descripcion;
37 }
38
39 public String getSotck() {
40     return sotck;
41 }
42
43 public void setSotck(String sotck) {
44     this.sotck = sotck;
45 }
46
47 @Override
48 public String toString() {
49     return "Producto{" + "id=" + id + ", descripcion=" + descripcion + ", sotck=" + sotck + '}';
```



Examen

06/03/2021

```
VistaCliente.java x VistaCliente.java x VistaFactura.java x Factura.java x ControladorCliente.java x Cliente.java x
Source History
6 package Modelo;
7
8 /**
9  *
10  * @author PCX
11  */
12 public class Cliente {
13     private long id;
14     private String nombre;
15     private String cedula;
16
17     public Cliente(long id, String nombre, String cedula) {
18         this.id = id;
19         this.nombre = nombre;
20         this.cedula = cedula;
21     }
22
23     public long getId() {
24         return id;
25     }
26
27     public void setId(long id) {
28         this.id = id;
29     }
30
31     public String getNombre() {
32         return nombre;
33     }
34
35     public void setNombre(String nombre) {
36         this.nombre = nombre;
37     }
38
39     public String getCedula() {
40         return cedula;
41     }
42 }
```



Examen

06/03/2021

```
VistaCliente.java x VistaCliente.java x VistaFactura.java x Factura.java x ControladorCliente.java x Cliente.java x
Source History
5  */
6  package Modelo;
7
8  import java.util.Date;
9
10 /**
11  *
12  * @author PCX
13  */
14 public class Factura {
15     private long id;
16     private Date fecha;
17     private String producto;
18     private int cantidad;
19     private Double valor_unitario;
20     private Double total;
21
22     public Factura(long id, Date fecha, String producto, int cantidad, Double valor_unitario, Double total) {
23         this.id = id;
24         this.fecha = fecha;
25         this.producto = producto;
26         this.cantidad = cantidad;
27         this.valor_unitario = valor_unitario;
28         this.total = total;
29     }
30
31     public long getId() {
32         return id;
33     }
34
35     public void setId(long id) {
36         this.id = id;
37     }
38
39     public Date getFecha() {
40         return fecha;
41     }
42 }
```



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package Controlador;
7
8  import Modelo.Cliente;
9  import java.util.ArrayList;
10 import java.util.List;
11
12 /**
13  *
14  * @author PCX
15  */
16 public class ControladorCliente {
17     private List<Cliente> datos;
18     private Cliente seleccionado;
19
20     public ControladorCliente() {
21         datos = new ArrayList<Cliente>();
22         seleccionado = null;
23     }
24
25     public boolean crear(long id, String nombre, String cedula) {
26         Cliente cliente = new Cliente( id, nombre, cedula);
27         return datos.add(cliente);
28     }
29
30     public Cliente buscar(String cedula){
31         for (Cliente cliente : datos) {
32             if(cliente.getCedula().equals(cedula) == true){
33                 return cliente;
34             }
35         }
36         return null;
37     }
38 }
```



Programación Orientada Objetos

Tema: Clases y Sistemas en Java .

Examen

06/03/2021

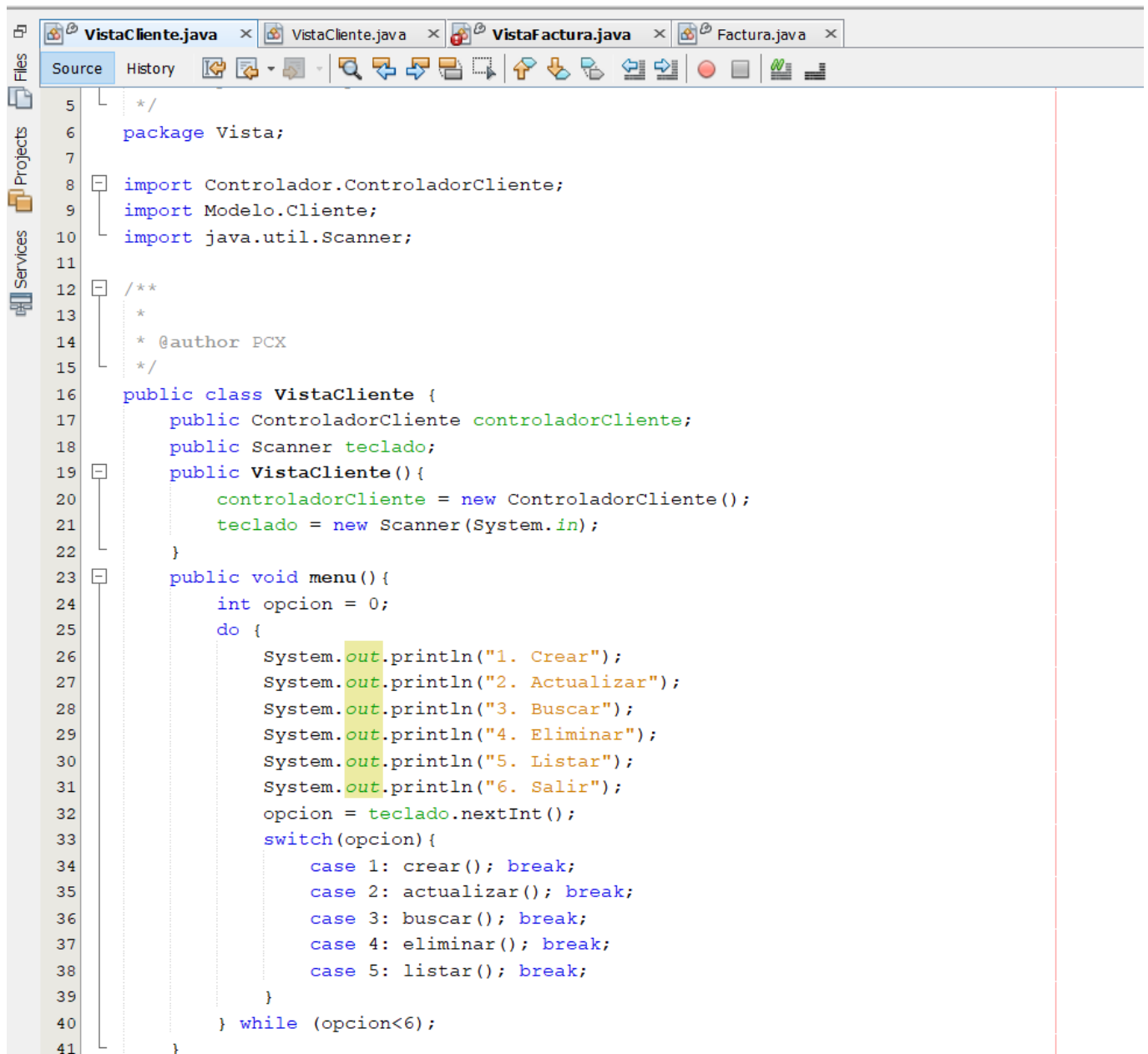
```
Producto.java x ControladorFactura.java x
Source History

5 //
6 package Controlador;
7
8 import Modelo.Cliente;
9 import Modelo.Factura;
10 import java.util.ArrayList;
11 import java.util.Date;
12 import java.util.List;
13
14 /**
15  *
16  * @author PCX
17  */
18 public class ControladorFactura {
19     private List<Cliente> datos;
20     private Cliente seleccionado;
21
22     public ControladorFactura () {
23         datos = new ArrayList<Factura>();
24         seleccionado = null;
25     }
26
27     public boolean crear(long id, Date fecha, String producto, int cantidad, Double valor_unitario, Double total) {
28         Factura factura = new Factura( id, fecha, producto, cantidad, valor_unitario, total);
29         return datos.add(factura);
30     }
31
32     public Factura buscar(String cedula, Factura producto){
33         for ( Cliente factura : datos) {
34             if(factura .getproducto().equals(producto) == true){
35                 return producto;
36             }
37         }
38         return null;
39     }
40     public boolean actualizar(long id, Date fecha, String producto, int cantidad, Double valor_unitario, Double total) {
41         Factura factura;
```

```
Services
sSecuencial
queadero
)rectorio
actura
e Packages
ontrolador
B ControladorCliente.java
B ControladorFactura.java
B ControladorProducto.java
B Factura.java
odelo
B Cliente.java
B Factura.java
B Producto.java
ista
amenfactura
Packages
es
Libraries
nacci

Producto.java x ControladorFactura.java x ControladorProducto.java x
Source History

5 //
6 package Controlador;
7
8 import Modelo.Cliente;
9 import Modelo.Producto;
10 import java.util.ArrayList;
11 import java.util.List;
12
13 /**
14  *
15  * @author PCX
16  */
17 public class ControladorProducto {
18     private List<Cliente> datos;
19     private Cliente seleccionado;
20
21     public ControladorProducto() {
22         datos = new ArrayList<Cliente>();
23         seleccionado = null;
24     }
25
26     public boolean crear(long id, String descripcion, String stock, String stock, Cliente Producto) {
27         Producto producto = new Producto( id, descripcion , stock);
28         return datos.add(Producto);
29     }
30
31     public Producto buscar(String descripcion, Producto Producto){
32         for (Cliente producto : datos) {
33             if(producto.getdescripcion().equals(descripcion) == true){
34                 return Producto;
35             }
36         }
37         return null;
38     }
39     public boolean actualizar(long id, String descripcion, String stock) {
40         Producto producto = this.buscar(descripcion );
41         if(producto != null) {
```



```
5  /*
6  package Vista;
7
8  import Controlador.ControladorCliente;
9  import Modelo.Cliente;
10 import java.util.Scanner;
11
12 /**
13  *
14  * @author PCX
15  */
16 public class VistaCliente {
17     public ControladorCliente controladorCliente;
18     public Scanner teclado;
19     public VistaCliente() {
20         controladorCliente = new ControladorCliente();
21         teclado = new Scanner(System.in);
22     }
23     public void menu() {
24         int opcion = 0;
25         do {
26             System.out.println("1. Crear");
27             System.out.println("2. Actualizar");
28             System.out.println("3. Buscar");
29             System.out.println("4. Eliminar");
30             System.out.println("5. Listar");
31             System.out.println("6. Salir");
32             opcion = teclado.nextInt();
33             switch(opcion) {
34                 case 1: crear(); break;
35                 case 2: actualizar(); break;
36                 case 3: buscar(); break;
37                 case 4: eliminar(); break;
38                 case 5: listar(); break;
39             }
40         } while (opcion<6);
41     }
42 }
```




```
Services
Product.java
ControladorFactura.java
ControladorProducto.java
VistaCliente.java
VistaFactura.java

Source
History
package vista;

import Controlador.ControladorCliente;
import Modelo.Cliente;
import java.util.Date;
import java.util.Scanner;

/**
 * @author PCX
 */
public class VistaFactura {
    public ControladorCliente controladorCliente;
    public Scanner teclado;
    public VistaFactura() {
        controladorCliente = new ControladorCliente();
        teclado = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("1. Crear");
            System.out.println("2. Actualizar");
            System.out.println("3. Buscar");
            System.out.println("4. Eliminar");
            System.out.println("5. Listar");
            System.out.println("6. Salir");
            opcion = teclado.nextInt();
            switch(opcion) {
                case 1: crear(); break;
                case 2: actualizar(); break;
                case 3: buscar(); break;
                case 4: eliminar(); break;
                case 5: listar(); break;
            }
        } while (opcion < 6);
    }
}
```