



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 producto 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**.



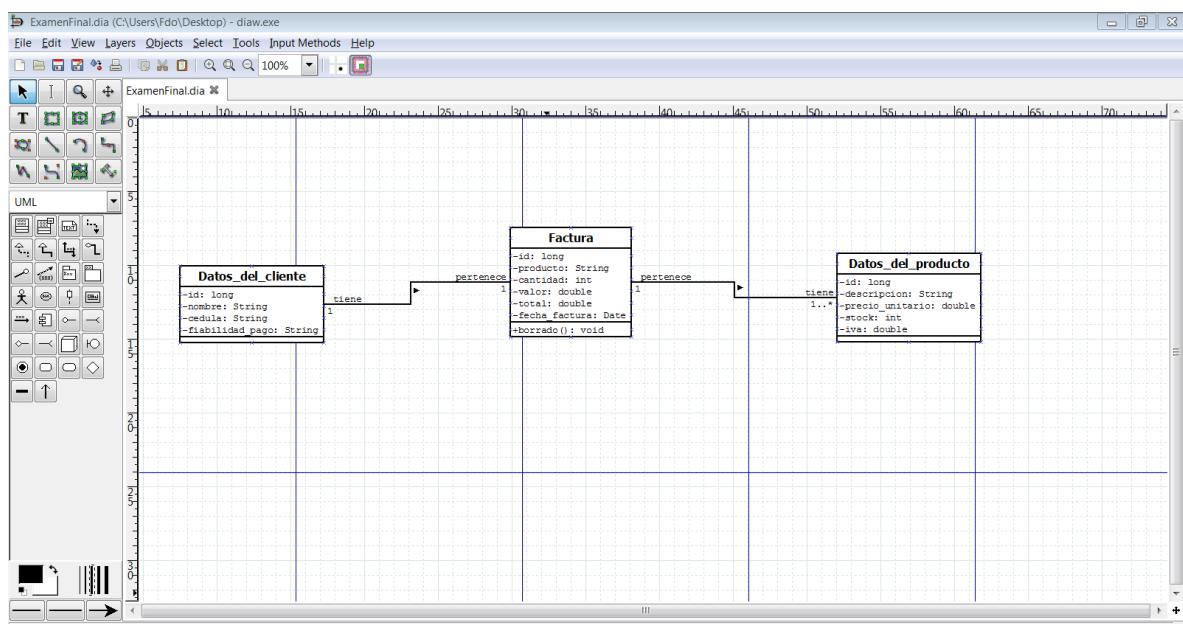
Examen

06/03/2021

Realización del Examen Final

Lo primero que hice para poder realizar todo este proyecto fue analizar el enunciado y junté los atributos y métodos en sus clases adecuados, es decir, fuí organizando la información respectivamente, y al ordenarlos fuí creando el diagrama de día ya que eso ayudaría en visualizar la información del sistema y las relaciones entre las clases.

Diagrama de día:





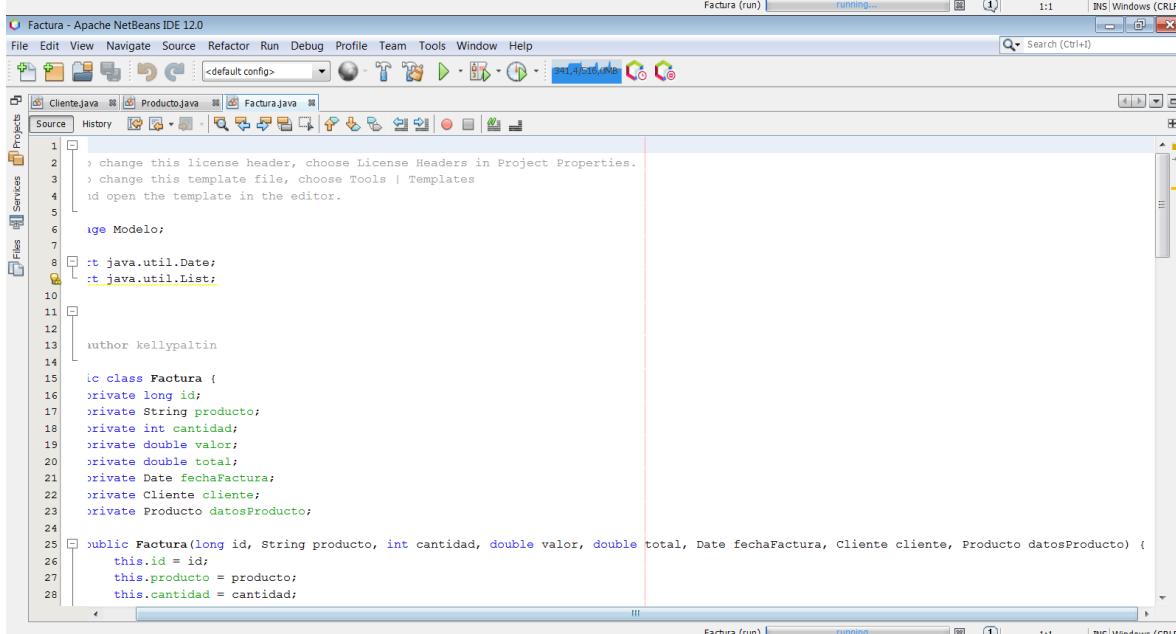
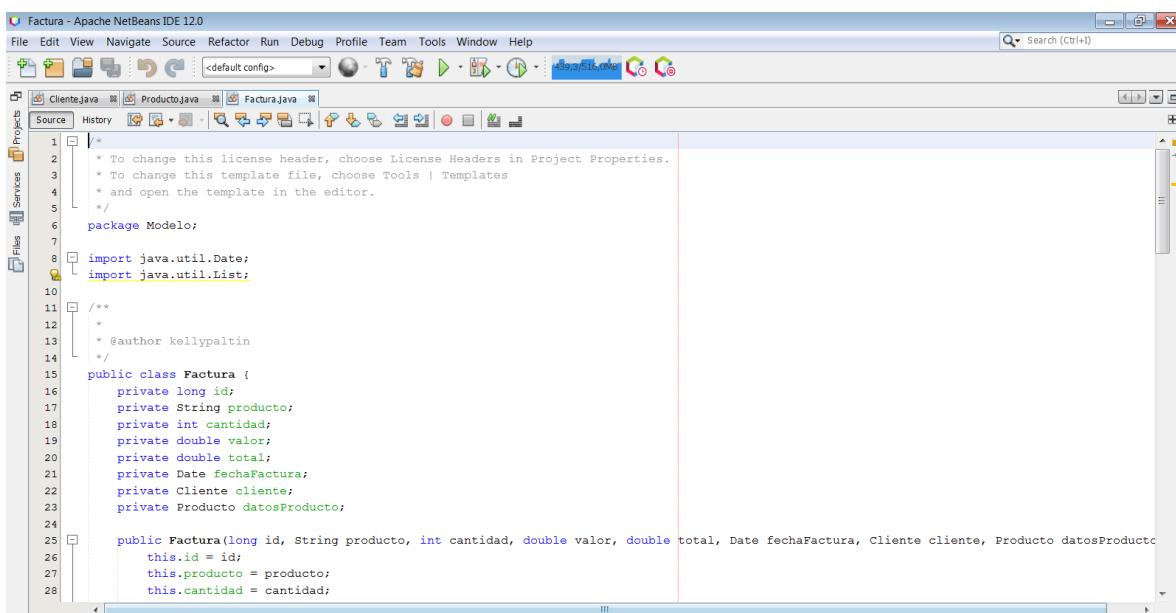
Examen

06/03/2021

Ahora esa información le pasamos a java (NetBeans) comenzando con los modelos, con los atributos y métodos de las clases creamos sus modelos correspondientes, colocamos los atributos de modo privado, ingresamos sus constructores y sus encapsulamientos y hacemos una llamado a los getter y setter, sus métodos debidos, y los terminamos con un override del método `toString` para que nos devuelva una cadena con la descripción de su clase, este proceso le aplicamos a las tres clases (factura, producto, cliente).

MVC:

-Modelo factura:



```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Services Files
Client.java Producto.java Factura.java
Source History <default config> Search (Ctrl+F)
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 Modelo;
7
8 import java.util.Date;
9 import java.util.List;
10
11 /**
12  *
13  * @author kellypaltin
14  */
15 public class Factura {
16     private long id;
17     private String producto;
18     private int cantidad;
19     private double valor;
20     private double total;
21     private Date fechaFactura;
22     private Cliente cliente;
23     private Producto datosProducto;
24
25     public Factura(long id, String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto) {
26         this.id = id;
27         this.producto = producto;
28         this.cantidad = cantidad;
29     }
30
31     @Override
32     public String toString() {
33         return "Factura{" + "id=" + id + ", producto=" + producto + ", cantidad=" + cantidad + ", valor=" + valor + ", total=" + total + ", fechaFactura=" + fechaFactura + ", cliente=" + cliente + ", datosProducto=" + datosProducto + '}';
34     }
35 }
```

The code shows the `Factura` class definition within the `Modelo` package. It includes fields for `id`, `producto`, `cantidad`, `valor`, `total`, `fechaFactura`, `cliente`, and `datosProducto`. A constructor is provided to initialize these fields. An `toString` method is overridden to return a string representation of the object.



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
public Factura(long id, String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto) {
    this.id = id;
    this.producto = producto;
    this.cantidad = cantidad;
    this.valor = valor;
    this.total = total;
    this.fechaFactura = fechaFactura;
    this.cliente = cliente;
    this.datosProducto = datosProducto;
}

public long getId() {
    return id;
}

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

public String getProducto() {
    return producto;
}

public void setProducto(String producto) {
    this.producto = producto;
}

public int getCantidad() {
```

Factura - Apache NetBeans IDE 12.0

```
public int getCantidad() {
    return cantidad;
}

public void setCantidad(int cantidad) {
    this.cantidad = cantidad;
}

public double getValor() {
    return valor;
}

public void setValor(double valor) {
    this.valor = valor;
}

public double getTotal() {
    return total;
}

public void setTotal(double total) {
    this.total = total;
}

public Date getFechaFactura() {
    return fechaFactura;
}
```



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
public void setFechaFactura(Date fechaFactura) {
    this.fechaFactura = fechaFactura;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

public Producto getDatosProducto() {
    return datosProducto;
}

public void setDatosProducto(Producto datosProducto) {
    this.datosProducto = datosProducto;
}

@Override
public String toString() {
    return "Factura(" + "id=" + id + ", producto=" + producto + ", cantidad=" + cantidad + ", valor=" + valor + ", total=" + total + ", fechaFactura=" + fechaFactura + ", cliente=" + cliente + ", datosProducto=" + datosProducto + ')';
}
```

Factura - Apache NetBeans IDE 12.0

```
public void setFechaFactura(Date fechaFactura) {
    this.fechaFactura = fechaFactura;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

public Producto getDatosProducto() {
    return datosProducto;
}

public void setDatosProducto(Producto datosProducto) {
    this.datosProducto = datosProducto;
}

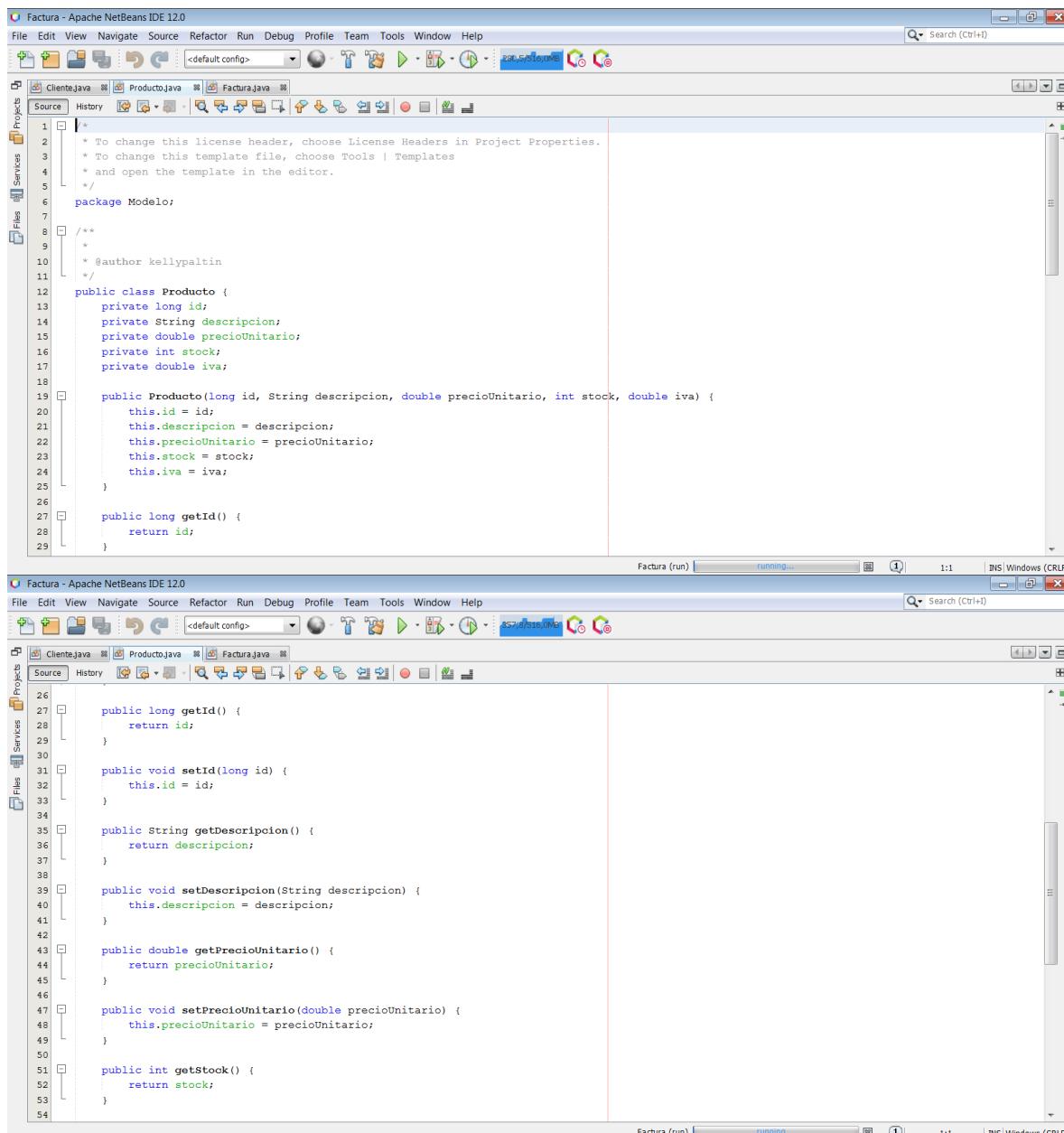
@Override
public String toString() {
    return "Factura(" + "id=" + id + ", producto=" + producto + ", cantidad=" + cantidad + ", valor=" + valor + ", total=" + total + ", fechaFactura=" + fechaFactura + ", cliente=" + cliente + ", datosProducto=" + datosProducto + ')';
}
```



Examen

06/03/2021

-Modelo producto:



```
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 Modelo;
7
8 /**
9  * 
10 * @author kellypaltin
11 */
12 public class Producto {
13     private long id;
14     private String descripcion;
15     private double precioUnitario;
16     private int stock;
17     private double iva;
18
19     public Producto(long id, String descripcion, double precioUnitario, int stock, double iva) {
20         this.id = id;
21         this.descripcion = descripcion;
22         this.precioUnitario = precioUnitario;
23         this.stock = stock;
24         this.iva = iva;
25     }
26
27     public long getId() {
28         return id;
29     }
30
31     public void setId(long id) {
32         this.id = id;
33     }
34
35     public String getDescripcion() {
36         return descripcion;
37     }
38
39     public void setDescripcion(String descripcion) {
40         this.descripcion = descripcion;
41     }
42
43     public double getPrecioUnitario() {
44         return precioUnitario;
45     }
46
47     public void setPrecioUnitario(double precioUnitario) {
48         this.precioUnitario = precioUnitario;
49     }
50
51     public int getStock() {
52         return stock;
53     }
54 }
```



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
44     return precioUnitario;
45 }
46
47     public void setPrecioUnitario(double precioUnitario) {
48         this.precioUnitario = precioUnitario;
49     }
50
51     public int getStock() {
52         return stock;
53     }
54
55     public void setStock(int stock) {
56         this.stock = stock;
57     }
58
59     public double getIva() {
60         return iva;
61     }
62
63     public void setIva(double iva) {
64         this.iva = iva;
65     }
66
67     @Override
68     public String toString() {
69         return "Producto(" + "id=" + id + ", descripcion=" + descripcion + ", precioUnitario=" + precioUnitario + ", stock=" + stock + ", iva=" + iva + ')';
70     }
71 }
```

Factura - Apache NetBeans IDE 12.0

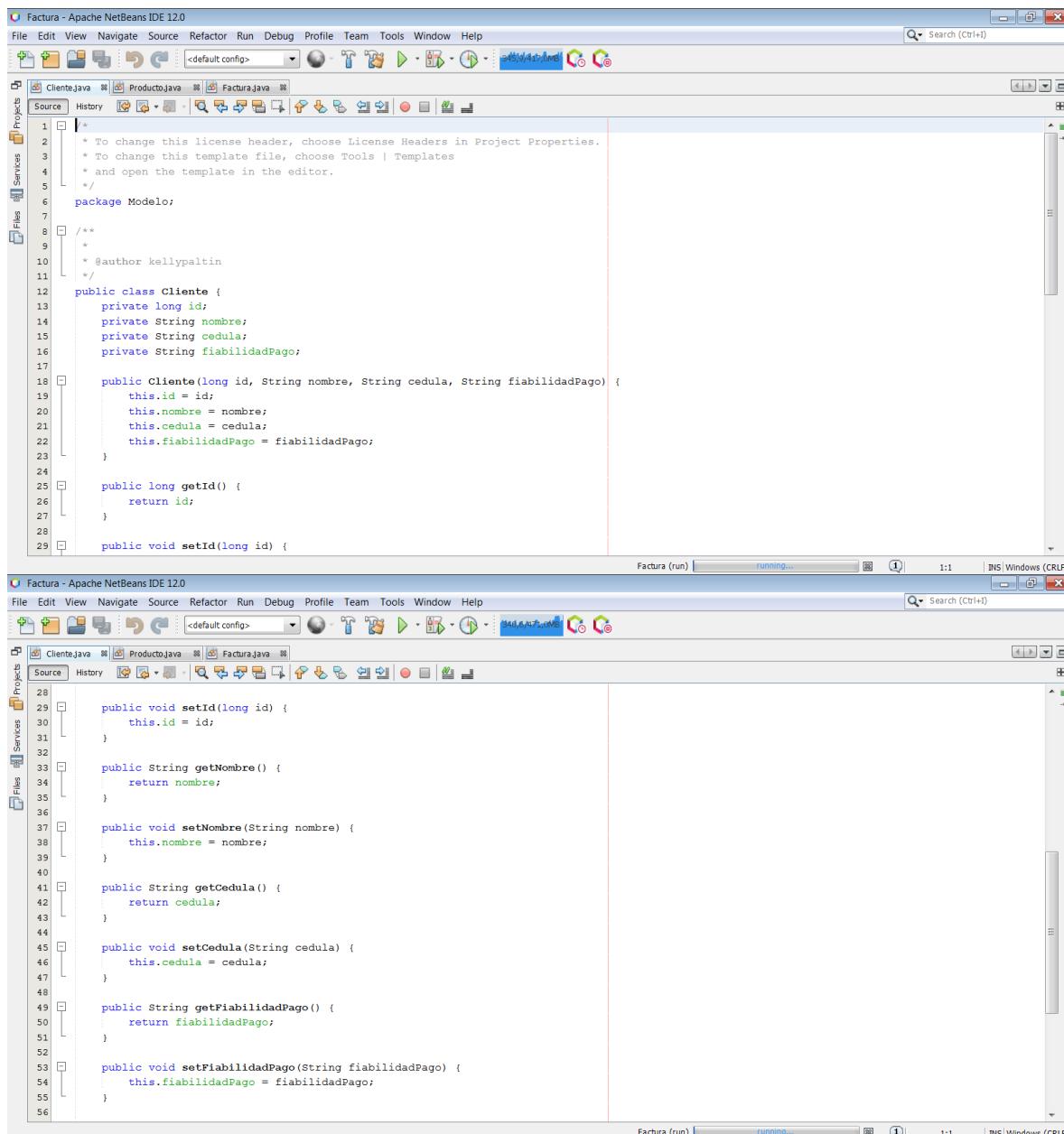
```
44     irn precioUnitario;
45
46
47     void setPrecioUnitario(double precioUnitario) {
48         .precioUnitario = precioUnitario;
49     }
50
51     int getStock() {
52         irn stock;
53     }
54
55     void setStock(int stock) {
56         .stock = stock;
57     }
58
59     double getIva() {
60         irn iva;
61     }
62
63     void setIva(double iva) {
64         .iva = iva;
65     }
66
67     ie
68     string toString() {
69         irn "Producto(" + "id=" + id + ", descripcion=" + descripcion + ", precioUnitario=" + precioUnitario + ", stock=" + stock + ", iva=" + iva + ')';
70     }
71 }
```



Examen

06/03/2021

-Modelo cliente:



```
1 // To change this license header, choose License Headers in Project Properties.
2 // To change this template file, choose Tools | Templates
3 // and open the template in the editor.
4 /*
5  * Author: kellypaltin
6  */
7 package Modelo;
8
9 /**
10  * @author kellypaltin
11  */
12 public class Cliente {
13     private long id;
14     private String nombre;
15     private String cedula;
16     private String fiabilidadPago;
17
18     public Cliente(long id, String nombre, String cedula, String fiabilidadPago) {
19         this.id = id;
20         this.nombre = nombre;
21         this.cedula = cedula;
22         this.fiabilidadPago = fiabilidadPago;
23     }
24
25     public long getId() {
26         return id;
27     }
28
29     public void setId(long id) {
30         this.id = id;
31     }
32
33     public String getNombre() {
34         return nombre;
35     }
36
37     public void setNombre(String nombre) {
38         this.nombre = nombre;
39     }
40
41     public String getCedula() {
42         return cedula;
43     }
44
45     public void setCedula(String cedula) {
46         this.cedula = cedula;
47     }
48
49     public String getFiabilidadPago() {
50         return fiabilidadPago;
51     }
52
53     public void setFiabilidadPago(String fiabilidadPago) {
54         this.fiabilidadPago = fiabilidadPago;
55     }
56 }
```



Examen

06/03/2021

```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Projects Services Files
Source History <default config> Search (Ctrl+I)
Client.java Producto.java Factura.java
37     public void setNombre(String nombre) {
38         this.nombre = nombre;
39     }
40
41     public String getCedula() {
42         return cedula;
43     }
44
45     public void setCedula(String cedula) {
46         this.cedula = cedula;
47     }
48
49     public String getFiabilidadPago() {
50         return fiabilidadPago;
51     }
52
53     public void setFiabilidadPago(String fiabilidadPago) {
54         this.fiabilidadPago = fiabilidadPago;
55     }
56
57     @Override
58     public String toString() {
59         return "Cliente{" + "id=" + id + ", nombre=" + nombre + ", cedula=" + cedula + ", fiabilidadPago=" + fiabilidadPago + '}';
60     }
61
62 }
63 
```

Para los controladores creamos su generarId para crear su id fácilmente, y creamos además su CRUD (create/crear, read/buscar, update/actualizar, delete/eliminar) estos nos ayudarán como funciones que podremos realizar con los atributos de cada clase, así que con esto se organiza nuevamente en los parámetros que se deben ingresar por parte del usuario según los datos que provee cada clase y se crea su lista y seleccionado para su fácil acceso y realizamos su correspondiente get y set. Nuevamente realizamos este proceso a todas las clases.

-Controlador cliente:

```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Projects Services Files
Source History <default config> Search (Ctrl+I)
ControladorCliente.java ControladorProducto.java ControladorFactura.java
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  * @author kellypaltin
14  */
15
16 public class ControladorCliente {
17     private List<Cliente> listaCliente;
18     private Cliente seleccionado;
19
20     public ControladorCliente() {
21         listaCliente = new ArrayList<>();
22     }
23
24     public long generarId(){
25         return(listaCliente.size()>0)? listaCliente.get(listaCliente.size()-1).getId():1;
26     }
27
28     public boolean crear(String nombre, String cedula, String fiabilidadPago){
29         return listaCliente.add(new Cliente(generarId(), nombre, cedula, fiabilidadPago));
30     }
31 } 
```



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

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

public boolean crear(String nombre, String cedula, String fiabilidadPago){
    return listaCliente.add(new Cliente(generarId(), nombre, cedula, fiabilidadPago));
}

public Cliente buscar(String cedula){
    for(Cliente cliente : listaCliente){
        if(cliente.getCedula().equals(cedula)){
            seleccionado = cliente;
            return cliente;
        }
    }
    return null;
}

public boolean actualizar(String cedula, String nombre, String fiabilidadPago){
    Cliente cliente = buscar(cedula);
    if(cliente != null){
        int posicion = listaCliente.indexOf(cliente);
        cliente.setNombre(nombre);
        cliente.setFiabilidadPago(fiabilidadPago);
        listaCliente.set(posicion, cliente);
        return true;
    }
    return false;
}

public boolean eliminar(String cedula){
    Cliente cliente = buscar(cedula);
    return listaCliente.remove(cliente);
}

public List<Cliente> getListaCliente() {
    return listaCliente;
}

public void setListaCliente(List<Cliente> listaCliente) {
    this.listaCliente = listaCliente;
}

public Cliente getSeleccionado() {
    return seleccionado;
}

public void setSeleccionado(Cliente seleccionado) {
    this.seleccionado = seleccionado;
}
```

Factura (run) | running... | 1:1 | INS Windows (CRLF)

Factura - Apache NetBeans IDE 12.0

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

public boolean crear(String nombre, String cedula, String fiabilidadPago){
    return listaCliente.add(new Cliente(generarId(), nombre, cedula, fiabilidadPago));
}

public Cliente buscar(String cedula){
    for(Cliente cliente : listaCliente){
        if(cliente.getCedula().equals(cedula)){
            seleccionado = cliente;
            return cliente;
        }
    }
    return null;
}

public boolean actualizar(String cedula, String nombre, String fiabilidadPago){
    Cliente cliente = buscar(cedula);
    if(cliente != null){
        int posicion = listaCliente.indexOf(cliente);
        cliente.setNombre(nombre);
        cliente.setFiabilidadPago(fiabilidadPago);
        listaCliente.set(posicion, cliente);
        return true;
    }
    return false;
}

public boolean eliminar(String cedula){
    Cliente cliente = buscar(cedula);
    return listaCliente.remove(cliente);
}

public List<Cliente> getListaCliente() {
    return listaCliente;
}

public void setListaCliente(List<Cliente> listaCliente) {
    this.listaCliente = listaCliente;
}

public Cliente getSeleccionado() {
    return seleccionado;
}

public void setSeleccionado(Cliente seleccionado) {
    this.seleccionado = seleccionado;
}
```

Factura (run) | running... | 1:1 | INS Windows (CRLF)



Examen

06/03/2021

-Controlador factura:

```
/* To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Controlador;

import Modelo.Cliente;
import Modelo.Factura;
import Modelo.Producto;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

/**
 *
 * @author kellypaltin
 */
public class ControladorFactura {
    private List<Factura> listaFactura;
    private Factura seleccionado;
    public double valorIva = 0.12;

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

    public long generarId(){
        return (listaFactura.size()>0)? listaFactura.get(listaFactura.size()-1).getId()+1:1;
    }

    public boolean crear(String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto){
        return listaFactura.add(new Factura(generarId(), producto, cantidad, valor, total, fechaFactura, cliente, datosProducto));
    }

    public Factura buscar(String producto){
        for (Factura factura : listaFactura){
            if(factura.getCliente().equals(producto)){
                seleccionado = factura;
                return factura;
            }
        }
        return null;
    }

    public boolean actualizar(String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto){
        Factura factura = buscar(producto);
        if(factura != null){
            int posicion = listaFactura.indexOf(factura);
            factura.setProducto(producto);
            factura.setCantidad(cantidad);
            factura.setValor(valor);
            double sub = valor*valorIva;
            total= valor+sub;
            factura.setTotal(total);
        }
    }
}
```



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
public long generarId() {
    return (listaFactura.size() > 0) ? listaFactura.get(listaFactura.size() - 1).getId() + 1 : 1;
}

public boolean crear(String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto) {
    return listaFactura.add(new Factura(generarId(), producto, cantidad, valor, total, fechaFactura, cliente, datosProducto));
}

public Factura buscar(String producto) {
    for (Factura factura : listaFactura) {
        if (factura.getCliente().equals(producto)) {
            seleccionado = factura;
            return factura;
        }
    }
    return null;
}

public boolean actualizar(String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto) {
    Factura factura = buscar(producto);
    if (factura != null) {
        int posicion = listaFactura.indexOf(factura);
        factura.setProducto(producto);
        factura.setCantidad(cantidad);
        factura.setValor(valor);
        double sub = valor * valorIva;
        total = valor + sub;
        factura.setTotal(total);
    }
    return true;
}

public boolean eliminar(String producto) {
    Factura factura = buscar(producto);
    return listaFactura.remove(factura);
}

public List<Factura> getListaFactura() {
    return listaFactura;
}

public void setListaFactura(List<Factura> listaFactura) {
    this.listaFactura = listaFactura;
}
```

Factura - Apache NetBeans IDE 12.0

```
public boolean actualizar(String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProduct) {
    Factura factura = buscar(producto);
    if (factura != null) {
        int posicion = listaFactura.indexOf(factura);
        factura.setProducto(producto);
        factura.setCantidad(cantidad);
        factura.setValor(valor);
        double sub = valor * valorIva;
        total = valor + sub;
        factura.setTotal(total);
        factura.setFechaFactura(fechaFactura);
        listaFactura.set(posicion, factura);
        return true;
    }
    return false;
}

public boolean eliminar(String producto) {
    Factura factura = buscar(producto);
    return listaFactura.remove(factura);
}

public List<Factura> getListaFactura() {
    return listaFactura;
}

public void setListaFactura(List<Factura> listaFactura) {
    this.listaFactura = listaFactura;
}
```



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Services Projects Files
Source History <default config> ControladorCliente.java ControladorProducto.java ControladorFactura.java
58     return true;
59 }
60 }
61
62 public boolean eliminar(String producto){
63     Factura factura = buscar(producto);
64     return listaFactura.remove(factura);
65 }
66
67 public List<Factura> getListaFactura() {
68     return listaFactura;
69 }
70
71 public void setListaFactura(List<Factura> listaFactura) {
72     this.listaFactura = listaFactura;
73 }
74
75 public Factura getSeleccionado() {
76     return seleccionado;
77 }
78
79 public void setSeleccionado(Factura seleccionado) {
80     this.seleccionado = seleccionado;
81 }
82
83
84 }
85 }
```

Factura (run) running... 1:1 INS Windows (CRLF)

-Controlador producto:

Factura - Apache NetBeans IDE 12.0

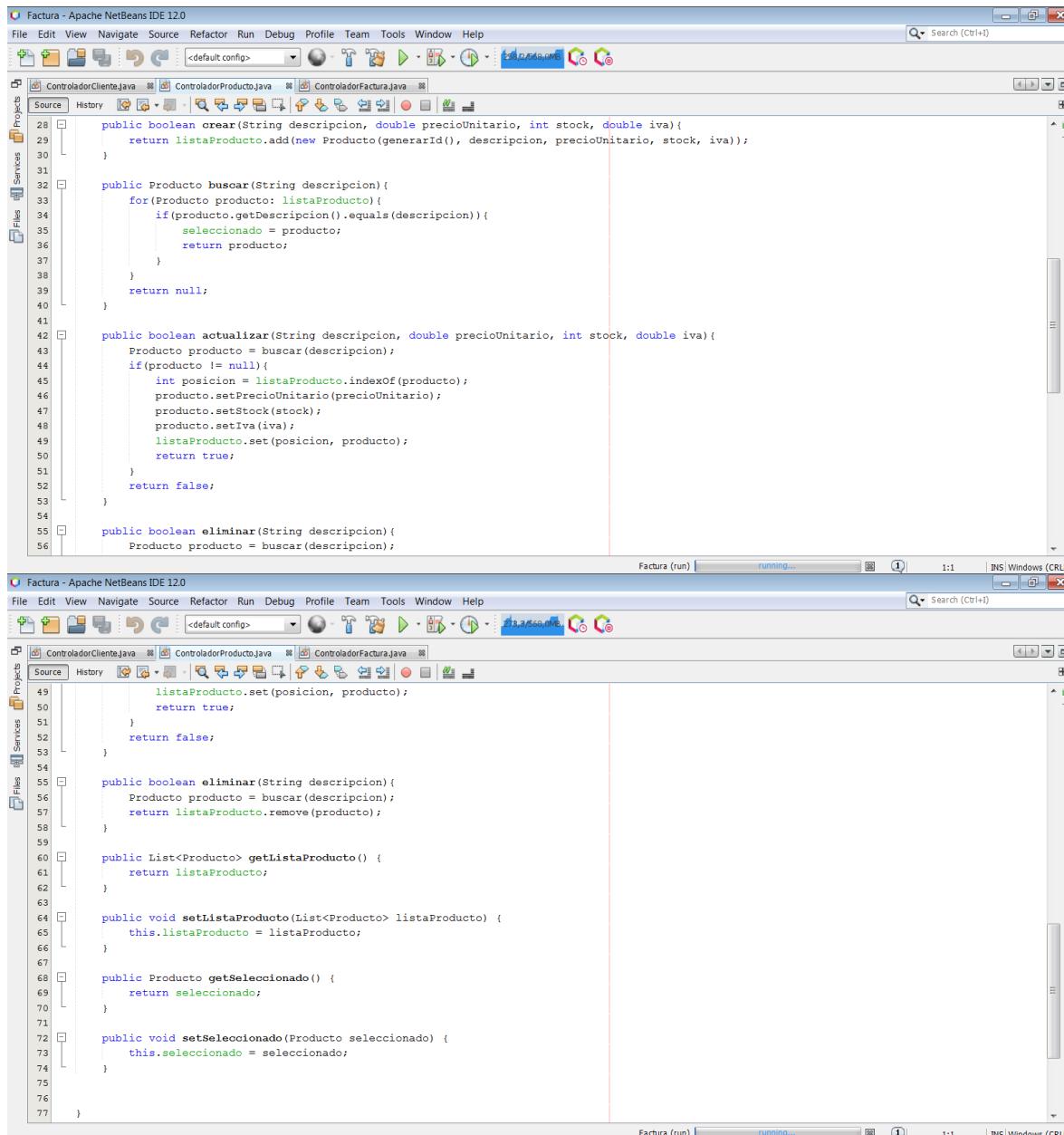
```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Services Projects Files
Source History <default config> ControladorCliente.java ControladorProducto.java ControladorFactura.java
1 /*
2  * To change this license header, choose License Headers in Project Properties.
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.Producto;
9 import java.util.ArrayList;
10 import java.util.List;
11
12 /**
13  *
14  * @author kellypaltin
15  */
16 public class ControladorProducto {
17     private List<Producto> listaProducto;
18     private Producto seleccionado;
19
20     public ControladorProducto() {
21         listaProducto = new ArrayList();
22     }
23
24     public long generarId(){
25         return (listaProducto.size()>0)? listaProducto.get(listaProducto.size()-1).getId():1;
26     }
27
28     public boolean crear(String descripcion, double precioUnitario, int stock, double iva){
29         return listaProducto.add(new Producto(generarId(), descripcion, precioUnitario, stock, iva));
30     }
31 }
```

Factura (run) running... 1:1 INS Windows (CRLF)



Examen

06/03/2021



The screenshot shows two instances of the Apache NetBeans IDE 12.0 interface. Both instances have the title bar "Factura - Apache NetBeans IDE 12.0". The left instance displays the code for the "ControladorCliente.java" class, which contains methods for creating, searching, updating, and deleting clients. The right instance displays the code for the "ControladorProducto.java" class, which contains methods for creating, searching, updating, and deleting products. Both classes interact with a "listaProducto" list and a "seleccionado" variable.

```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

ControladorCliente.java ControladorProducto.java ControladorFactura.java

public boolean crear(String descripcion, double precioUnitario, int stock, double iva){
    return listaProducto.add(new Producto(generarId(), descripcion, precioUnitario, stock, iva));
}

public Producto buscar(String descripcion){
    for(Producto producto: listaProducto){
        if(producto.getDescripcion().equals(descripcion)){
            seleccionado = producto;
            return producto;
        }
    }
    return null;
}

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

public boolean eliminar(String descripcion){
    Producto producto = buscar(descripcion);
    listaProducto.remove(producto);
    return true;
}

public List<Producto> getListado(){
    return listaProducto;
}

public void setListado(List<Producto> listaProducto) {
    this.listaProducto = listaProducto;
}

public Producto getSeleccionado() {
    return seleccionado;
}

public void setSeleccionado(Producto seleccionado) {
    this.seleccionado = seleccionado;
}
```

```
Factura (run) running... 1:1 INS Windows (CRLF)

ControladorCliente.java ControladorProducto.java ControladorFactura.java

49     listaProducto.set(posicion, producto);
50     return true;
51 }
52 return false;
53 }

54 public boolean eliminar(String descripcion){
55     Producto producto = buscar(descripcion);
56     return listaProducto.remove(producto);
57 }
58 }

59 public List<Producto> getListado(){
60     return listaProducto;
61 }

62 }

63 public void setListado(List<Producto> listaProducto) {
64     this.listaProducto = listaProducto;
65 }

66 public Producto getSeleccionado() {
67     return seleccionado;
68 }

69 public void setSeleccionado(Producto seleccionado) {
70     this.seleccionado = seleccionado;
71 }

72 }
```

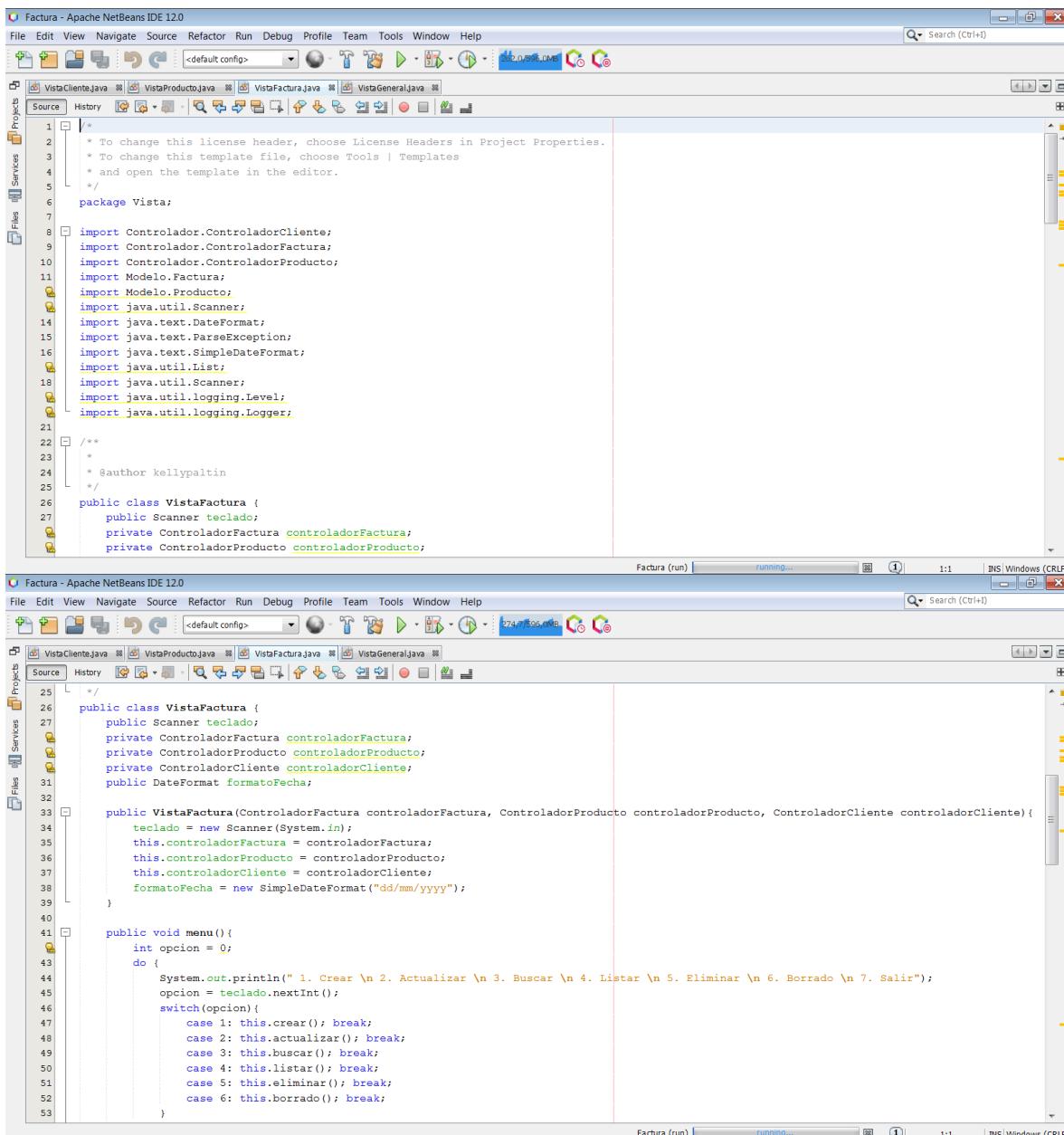
Basado en información aparte de sus propios atributos y métodos, indicamos la relación que tiene con los demás controladores para su ejecución. Ya que al crear su CRUD será necesario el aporte de los controladores de cliente y producto. Se realiza todo organizándolo en un menú para su visualización adecuada. E incluimos el método borrado con void para que en su llamado no devuelva ni reciba ningún parámetro. Además realizamos un try y catch para que en caso de algún error en ingresar la fecha para la factura.



Examen

06/03/2021

-Vista factura:



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Vista;

import Controlador.ControladorCliente;
import Controlador.ControladorFactura;
import Controlador.ControladorProducto;
import Modelo.Factura;
import Modelo.Producto;
import java.util.Scanner;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author kellypaltin
 */
public class VistaFactura {
    public Scanner teclado;
    private ControladorFactura controladorFactura;
    private ControladorProducto controladorProducto;
}

/*
 */
public class VistaFactura {
    public Scanner teclado;
    private ControladorFactura controladorFactura;
    private ControladorProducto controladorProducto;
    private ControladorCliente controladorCliente;
    public DateFormat formatoFecha;

    public VistaFactura(ControladorFactura controladorFactura, ControladorProducto controladorProducto, ControladorCliente controladorCliente) {
        teclado = new Scanner(System.in);
        this.controladorFactura = controladorFactura;
        this.controladorProducto = controladorProducto;
        this.controladorCliente = controladorCliente;
        formatoFecha = new SimpleDateFormat("dd/mm/yyyy");
    }

    public void menu(){
        int opcion = 0;
        do {
            System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Listar \n 5. Eliminar \n 6. Borrado \n 7. Salir");
            opcion = teclado.nextInt();
            switch(opcion){
                case 1: this.crear(); break;
                case 2: this.actualizar(); break;
                case 3: this.buscar(); break;
                case 4: this.listar(); break;
                case 5: this.eliminar(); break;
                case 6: this.borrado(); break;
            }
        } while(opcion != 7);
    }
}
```



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
case 6: this.borrado(); break;
} while (opcion < 7);

public void crear() {
    System.out.println("Ingrese:");
    System.out.println("Producto:");
    String producto = teclado.next();
    System.out.println("Cantidad:");
    int cantidad = teclado.nextInt();
    System.out.println("Valor:");
    double valor = teclado.nextDouble();
    System.out.println("Total:");
    double total = teclado.nextDouble();
    System.out.println("Fecha (dd/mm/yyyy):");
    String fechaFactura = teclado.next();
    try {
        System.out.println("Res:" + controladorFactura.crear(producto, cantidad, valor, total, formatoFecha.parse(fechaFactura), controladorCliente.getSeleccionado(), controladorProducto.getSeleccionado()));
    }catch (ParseException ex) {
        System.out.println(ex.getMessage());
    }
}

public void actualizar() {
    System.out.println("Ingrese:");
    System.out.println("Producto:");
    String producto = teclado.next();
    
```

Factura (run) running... 1:1 INS Windows (CRLF)

Factura - Apache NetBeans IDE 12.0

```

52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79

```

```
, cantidad, valor, total, formatoFecha.parse(fechaFactura), controladorCliente.getSeleccionado(), controladorProducto.getSeleccionado()));


```

Factura (run) running... 1:1 INS Windows (CRLF)



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

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

VistaCliente.java VistaProducto.java VistaFactura.java VistaGeneral.java

Source History

```
67     System.out.println("Fecha (dd/mm/yyyy):");
68     String fechaFactura = teclado.next();
69     try {
70         System.out.println("Res:" + controladorFactura.crear(producto, cantidad, valor, total, formatoFecha.parse(fechaFactura), controladorCliente.getSeleccionado(), controladorProducto.getSeleccionado()));
71     }catch (ParseException ex) {
72         System.out.println(ex.getMessage());
73     }
74 }
75
76 public void actualizar() {
77     System.out.println("Ingrese:");
78     System.out.println("Producto:");
79     String producto = teclado.next();
80     System.out.println("Cantidad:");
81     int cantidad = teclado.nextInt();
82     System.out.println("Valor:");
83     double valor = teclado.nextDouble();
84     System.out.println("Total:");
85     double total = teclado.nextDouble();
86     System.out.println("Fecha (dd/mm/yyyy):");
87     String fechaFactura = teclado.next();
88     try {
89         System.out.println("Res:" + controladorFactura.actualizar(producto, cantidad, valor, total, formatoFecha.parse(fechaFactura), controladorCliente.getSeleccionado(), controladorProducto.getSeleccionado()));
90     }catch (ParseException ex) {
91         System.out.println(ex.getMessage());
92     }
93 }
```

Factura (run) running... 1:1 INS Windows (CRLF)

Factura - Apache NetBeans IDE 12.0

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

VistaCliente.java VistaProducto.java VistaFactura.java VistaGeneral.java

Source History

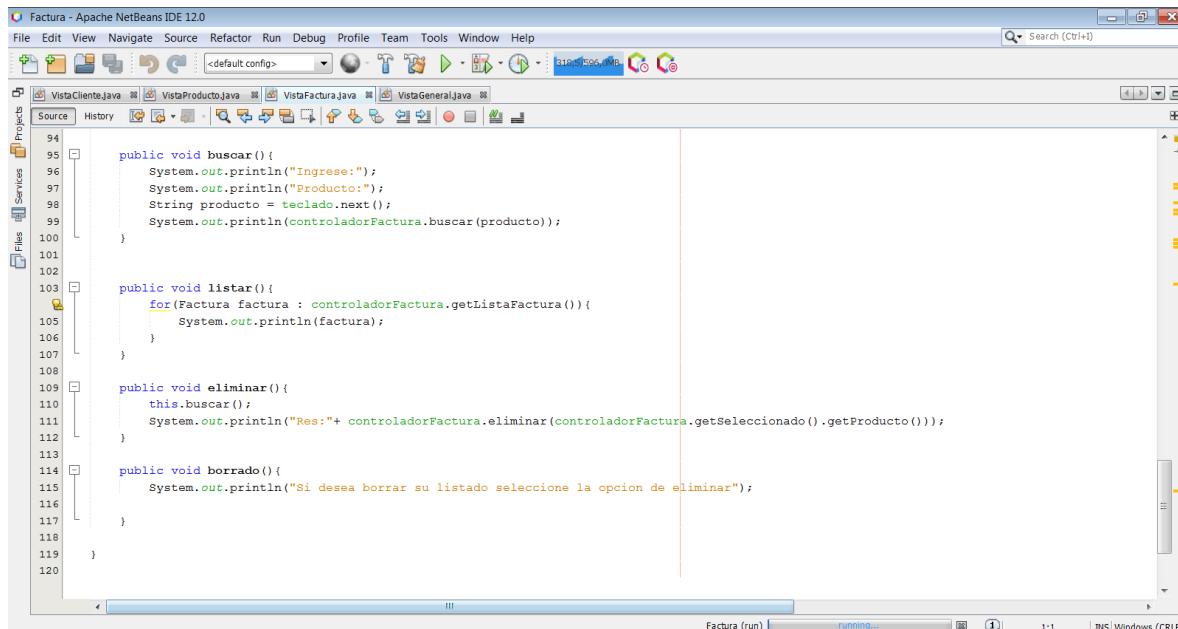
```
67
68
69
70     , cantidad, valor, total, formatoFecha.parse(fechaFactura), controladorCliente.getSeleccionado(), controladorProducto.getSeleccionado());
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89     producto, cantidad, valor, total, formatoFecha.parse(fechaFactura), controladorCliente.getSeleccionado(), controladorProducto.getSeleccionado());
90
91
92
93
```

Factura (run) running... 1:1 INS Windows (CRLF)



Examen

06/03/2021

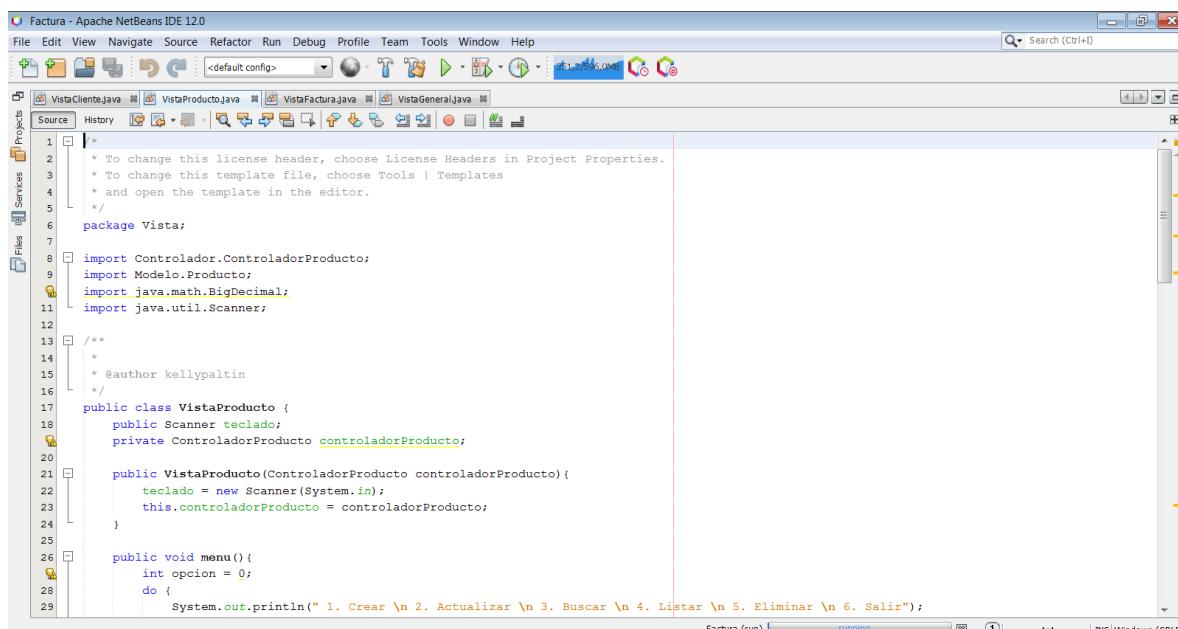


The screenshot shows the Apache NetBeans IDE 12.0 interface with the title bar "Factura - Apache NetBeans IDE 12.0". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, Run, Stop, and Help. The bottom status bar shows "Factura (run) running..." and other settings.

```
94
95     public void buscar() {
96         System.out.println("Ingrese:");
97         System.out.println("Producto:");
98         String producto = teclado.next();
99         System.out.println(controladorFactura.buscar(producto));
100    }
101
102    public void listar() {
103        for(Factura factura : controladorFactura.getListaFactura()){
104            System.out.println(factura);
105        }
106    }
107
108    public void eliminar() {
109        this.buscar();
110        System.out.println("Res:" + controladorFactura.eliminar(controladorFactura.getSeleccionado().getProducto()));
111    }
112
113    public void borrado() {
114        System.out.println("Si desea borrar su listado seleccione la opcion de eliminar");
115    }
116
117}
118
119}
120}
```

Al entrar en la vista se genera un menú que indica su CRUD, las funciones que se puede operar con la información el producto da a indicar que datos son necesarios para su búsqueda, actualización, listado y eliminación. Y esto realizamos igualmente para las dos vistas de producto y cliente.

-Vista producto:



The screenshot shows the Apache NetBeans IDE 12.0 interface with the title bar "Factura - Apache NetBeans IDE 12.0". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, Run, Stop, and Help. The bottom status bar shows "Factura (run) running..." and other settings.

```
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 Vista;
7
8 import Controlador.ControladorProducto;
9 import Modelo.Producto;
10 import java.math.BigDecimal;
11 import java.util.Scanner;
12
13 /**
14  *
15  * @author kellypaltin
16  */
17 public class VistaProducto {
18     public Scanner teclado;
19     private ControladorProducto controladorProducto;
20
21     public VistaProducto(ControladorProducto controladorProducto) {
22         teclado = new Scanner(System.in);
23         this.controladorProducto = controladorProducto;
24     }
25
26     public void menu(){
27         int opcion = 0;
28         do {
29             System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Listar \n 5. Eliminar \n 6. Salir");
30         }
```



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

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

public void crear(){
    System.out.print("Ingrese:");
    System.out.print("Descripcion:");
    String descripcion = teclado.next();
    System.out.print("Precio Unitario:");
    double precioUnitario = teclado.nextDouble();
    System.out.print("Stock:");
    int stock = teclado.nextInt();
    System.out.print("IVA:");
    double iva = teclado.nextDouble();
    System.out.println("Res:"+ controladorProducto.crear(descripcion, precioUnitario, stock, iva));
}
```

Factura (run) running... 1:1 INS Windows (CRLF)

Factura - Apache NetBeans IDE 12.0

```
public void actualizar(){
    System.out.print("Ingrese:");
    System.out.print("Descripcion:");
    String descripcion = teclado.next();
    System.out.print("Precio Unitario:");
    double precioUnitario = teclado.nextDouble();
    System.out.print("Stock:");
    int stock = teclado.nextInt();
    System.out.print("IVA:");
    double iva = teclado.nextDouble();
    System.out.println("Res:"+ controladorProducto.actualizar(descripcion, precioUnitario, stock, iva));
}

public Producto buscar(){
    System.out.print("Ingrese:");
    System.out.print("Descripcion:");
    String descripcion = teclado.next();
    System.out.println(controladorProducto.buscar(descripcion));
    return controladorProducto.buscar(descripcion);
}

public void listar(){
    for(Producto producto : controladorProducto.getListaProducto()){
        System.out.println(producto);
    }
}
```

Factura (run) running... 1:1 INS Windows (CRLF)



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
60     System.out.println("Stock:");
61     int stock = teclado.nextInt();
62     System.out.print("IVA:");
63     double iva = teclado.nextDouble();
64     System.out.println("Res:" + controladorProducto.actualizar(descripcion, precioUnitario, stock, iva));
65 }
66
67 public Producto buscar(){
68     System.out.println("Ingrese:");
69     System.out.println("Descripcion:");
70     String descripcion = teclado.next();
71     System.out.println(controladorProducto.buscar(descripcion));
72     return controladorProducto.buscar(descripcion);
73 }
74
75
76 public void listar(){
77     for(Producto producto : controladorProducto.getListaProducto()){
78         System.out.println(producto);
79     }
80 }
81
82 public void eliminar(){
83     this.buscar();
84     System.out.println("Res:" + controladorProducto.eliminar(controladorProducto.getSeleccionado().getDescripcion()));
85 }
86
87 }
88 }
```

Factura (run) running... 1:1 INS Windows (CRLF)

-Vista cliente:

Factura - Apache NetBeans IDE 12.0

```
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 Vista;
7
8 import Controlador.ControladorProducto;
9 import Modelo.Producto;
10 import java.math.BigDecimal;
11 import java.util.Scanner;
12
13 /**
14  *
15  * @author kellypaltin
16 */
17 public class VistaProducto {
18     public Scanner teclado;
19     private ControladorProducto controladorProducto;
20
21     public VistaProducto(ControladorProducto controladorProducto){
22         teclado = new Scanner(System.in);
23         this.controladorProducto = controladorProducto;
24     }
25
26     public void menu(){
27         int opcion = 0;
28         do {
29             System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Listar \n 5. Eliminar \n 6. Salir");
30         }
31     }
32 }
```

Factura (run) running... 1:1 INS Windows (CRLF)



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
1 * To change this license header, choose License Headers in Project Properties.
2 * To change this template file, choose Tools | Templates
3 * and open the template in the editor.
4 */
5 package Vista;
6
7 import Controlador.ControladorCliente;
8 import Modelo.Cliente;
9 import java.util.Scanner;
10
11 /**
12 *
13 * @author kellypaltin
14 */
15
16 public class VistaCliente {
17     public Scanner teclado;
18     private ControladorCliente controladorCliente;
19
20     public VistaCliente(ControladorCliente controladorCliente){
21         teclado = new Scanner(System.in);
22         this.controladorCliente = controladorCliente;
23     }
24
25     public void menu(){
26         int opcion = 0;
27         do {
28             System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Listar \n 5. Eliminar \n 6. Salir");
29             opcion = teclado.nextInt();
30
31         } while (opcion < 6);
32
33     }
34
35     public void crear(){
36         System.out.println("Ingrese:");
37         System.out.println("Nombre:");
38         String nombre = teclado.next();
39         System.out.println("Cedula:");
40         String cedula = teclado.next();
41         System.out.println("Fiabilidad de Pago:");
42         String fiabilidadPago = teclado.next();
43         System.out.println("Res:"+ controladorCliente.crear(nombre, cedula, fiabilidadPago));
44     }
45
46     public void actualizar(){
47         System.out.println("Ingrese:");
48         System.out.println("Nombre:");
49     }
50
51 }
```

Factura - Apache NetBeans IDE 12.0

```
25 public void menu(){
26     int opcion = 0;
27     do {
28         System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Listar \n 5. Eliminar \n 6. Salir");
29         opcion = teclado.nextInt();
30         switch(opcion){
31             case 1: this.crear(); break;
32             case 2: this.actualizar(); break;
33             case 3: this.buscar(); break;
34             case 4: this.listar(); break;
35             case 5: this.eliminar(); break;
36         }
37     } while (opcion < 6);
38
39
40     public void crear(){
41         System.out.println("Ingrese:");
42         System.out.println("Nombre:");
43         String nombre = teclado.next();
44         System.out.println("Cedula:");
45         String cedula = teclado.next();
46         System.out.println("Fiabilidad de Pago:");
47         String fiabilidadPago = teclado.next();
48         System.out.println("Res:"+ controladorCliente.crear(nombre, cedula, fiabilidadPago));
49     }
50
51     public void actualizar(){
52         System.out.println("Ingrese:");
53         System.out.println("Nombre:");
54     }
55 }
```



Examen

06/03/2021

```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Services Files
VistaCliente.java VistaProducto.java VistaFactura.java VistaGeneral.java
Source History <default config> 398.5/591 MB ① 1:1 INS Windows (CRLF)
51     public void actualizar(){
52         System.out.println("Ingrese:");
53         System.out.println("Nombre:");
54         String nombre = teclado.next();
55         System.out.println("Cedula:");
56         String cedula = teclado.next();
57         System.out.println("Fiabilidad de Pago:");
58         String fiabilidadPago = teclado.next();
59         System.out.println("Res:"+ controladorCliente.actualizar(nombre, cedula, fiabilidadPago));
60     }
61
62     public Cliente buscar(){
63         System.out.println("Ingrese:");
64         System.out.println("Cedula:");
65         String cedula = teclado.next();
66         System.out.println(controladorCliente.buscar(cedula));
67         return controladorCliente.buscar(cedula);
68     }
69
70     public void listar(){
71         for(Cliente cliente : controladorCliente.getListaCliente()){
72             System.out.println(cliente);
73         }
74     }
75
76     public void eliminar(){
77         this.buscar();
78         System.out.println("Res:"+ controladorCliente.eliminar(controladorCliente.getSeleccionado().getCedula()));
79     }
80 }

Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Services Files
VistaCliente.java VistaProducto.java VistaFactura.java VistaGeneral.java
Source History <default config> 398.5/591 MB ① 1:1 INS Windows (CRLF)
53     System.out.println("Nombre:");
54     String nombre = teclado.next();
55     System.out.println("Cedula:");
56     String cedula = teclado.next();
57     System.out.println("Fiabilidad de Pago:");
58     String fiabilidadPago = teclado.next();
59     System.out.println("Res:"+ controladorCliente.actualizar(nombre, cedula, fiabilidadPago));
60
61
62     public Cliente buscar(){
63         System.out.println("Ingrese:");
64         System.out.println("Cedula:");
65         String cedula = teclado.next();
66         System.out.println(controladorCliente.buscar(cedula));
67         return controladorCliente.buscar(cedula);
68     }
69
70     public void listar(){
71         for(Cliente cliente : controladorCliente.getListaCliente()){
72             System.out.println(cliente);
73         }
74     }
75
76     public void eliminar(){
77         this.buscar();
78         System.out.println("Res:"+ controladorCliente.eliminar(controladorCliente.getSeleccionado().getCedula()));
79     }
80 }
```

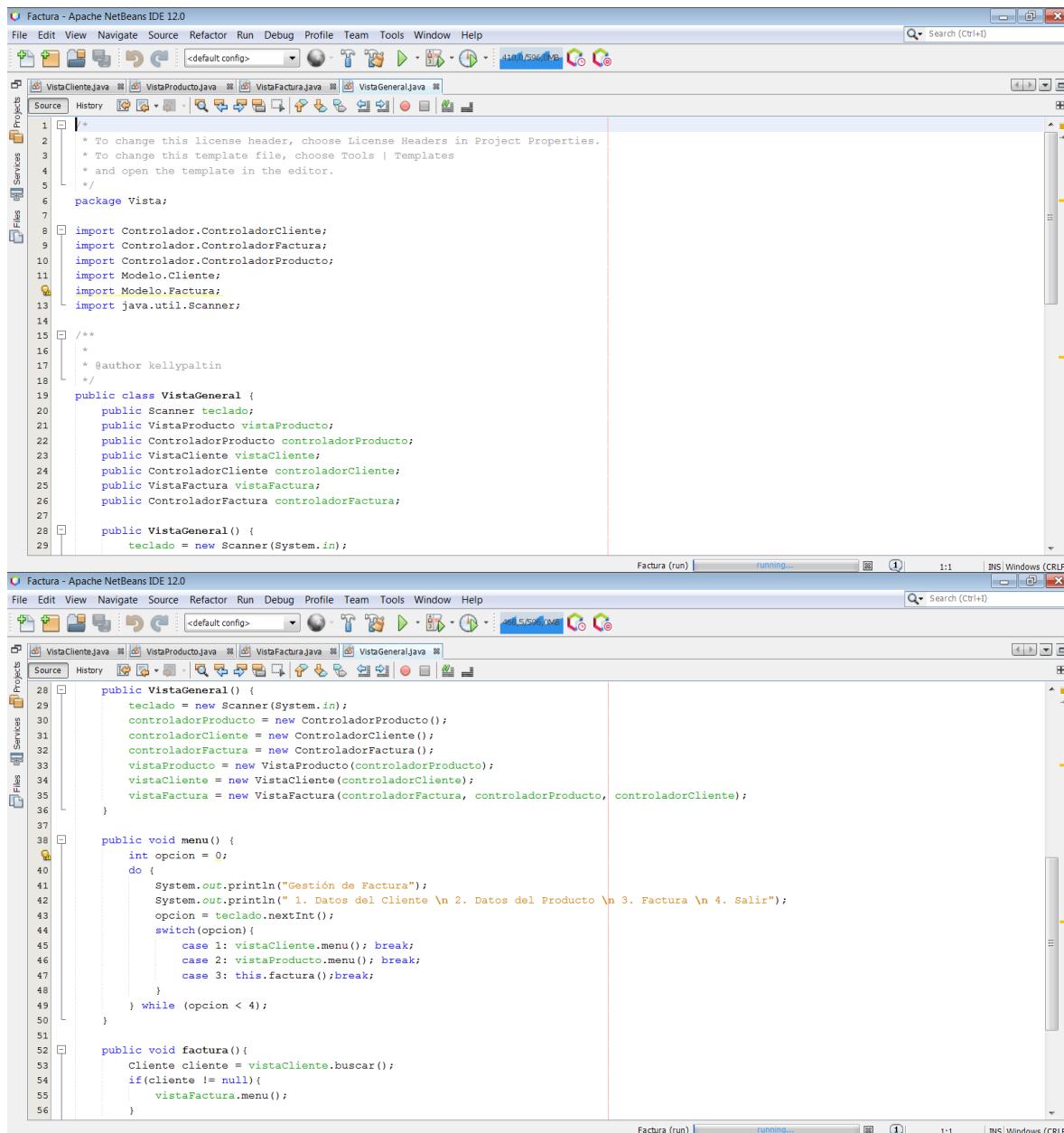
Esta vista generalizado nos ayuda a unir todas las vistas y sus respectivos controladores, y aquí se hace el primer menú que le aparecerá al usuario, es decir, le da la opción de elegir que quiere gestionar primero, da la opción de datos de cliente, datos del producto y la factura, y estos están propuestos en el orden que se mas sugiere ir en, ya que es necesario ingresar los datos del cliente y producto para poder gestionar la factura correspondiente. Y además damos la opción de salir al último cuando el usuario termina de usar el programa o haya terminado de generar su factura.



Examen

06/03/2021

-Vista general:

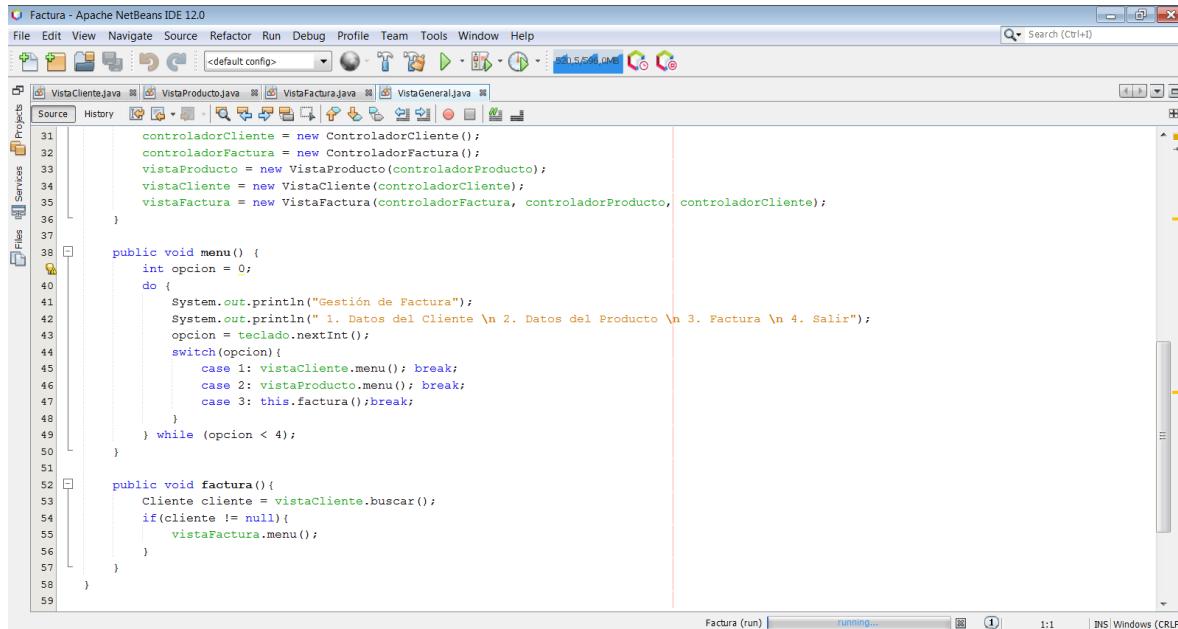


```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 416.596 MB Search (Ctrl+F)
Projects Services Files
Source History < VistaCliente.java VistaProducto.java VistaFactura.java VistaGeneral.java >
1 /*
2  * To change this 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 Vista;
7
8 import Controlador.ControladorCliente;
9 import Controlador.ControladorFactura;
10 import Controlador.ControladorProducto;
11 import Modelo.Cliente;
12 import Modelo.Factura;
13 import java.util.Scanner;
14
15 /**
16  *
17  * @author kellypaltin
18  */
19 public class VistaGeneral {
20     public Scanner teclado;
21     public VistaProducto vistaProducto;
22     public ControladorProducto controladorProducto;
23     public VistaCliente vistaCliente;
24     public ControladorCliente controladorCliente;
25     public VistaFactura vistaFactura;
26     public ControladorFactura controladorFactura;
27
28     public VistaGeneral() {
29         teclado = new Scanner(System.in);
30     }
31
32     public VistaGeneral() {
33         teclado = new Scanner(System.in);
34         controladorProducto = new ControladorProducto();
35         controladorCliente = new ControladorCliente();
36         controladorFactura = new ControladorFactura();
37         vistaProducto = new VistaProducto(controladorProducto);
38         vistaCliente = new VistaCliente(controladorCliente);
39         vistaFactura = new VistaFactura(controladorFactura, controladorProducto, controladorCliente);
40     }
41
42     public void menu() {
43         int opcion = 0;
44         do {
45             System.out.println("Gestión de Factura");
46             System.out.println(" 1. Datos del Cliente \n 2. Datos del Producto \n 3. Factura \n 4. Salir");
47             opcion = teclado.nextInt();
48             switch(opcion){
49                 case 1: vistaCliente.menu(); break;
50                 case 2: vistaProducto.menu(); break;
51                 case 3: this.factura();break;
52             }
53         } while (opcion < 4);
54     }
55
56     public void factura(){
57         Cliente cliente = vistaCliente.buscar();
58         if(cliente != null){
59             vistaFactura.menu();
60         }
61     }
}
```



Examen

06/03/2021



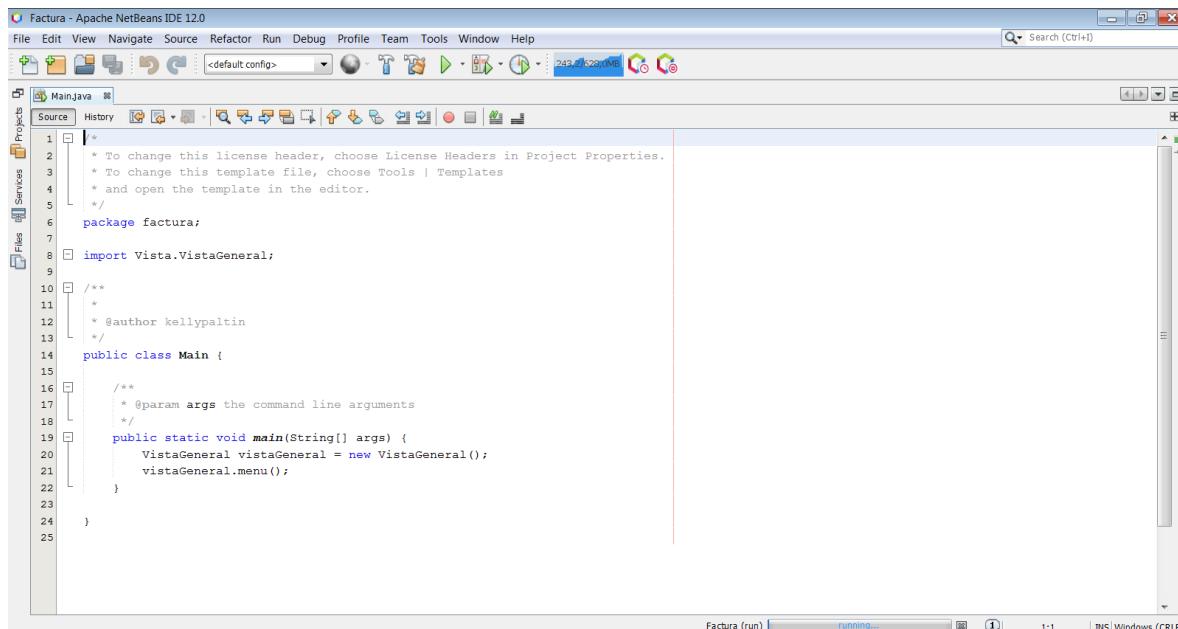
```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Source History <default config> Search (Ctrl+F)
Projects Services Files
VistaCliente.java VistaProducto.java VistaFactura.java VistaGeneral.java
controladorCliente = new ControladorCliente();
controladorFactura = new ControladorFactura();
vistaProductos = new VistaProducto(controladorProducto);
vistaCliente = new VistaCliente(controladorCliente);
vistaFactura = new VistaFactura(controladorFactura, controladorProducto, controladorCliente);

public void menu() {
    int opcion = 0;
    do {
        System.out.println("Gestión de Factura");
        System.out.println(" 1. Datos del Cliente \n 2. Datos del Producto \n 3. Factura \n 4. Salir");
        opcion = teclado.nextInt();
        switch(opcion) {
            case 1: vistaCliente.menu(); break;
            case 2: vistaProducto.menu(); break;
            case 3: this.factura();break;
        }
    } while (opcion < 4);
}

public void factura(){
    Cliente cliente = vistaCliente.buscar();
    if(cliente != null){
        vistaFactura.menu();
    }
}
```

Aquí simplemente se hace una llamado a la vista general y su menú en el main en donde aquí se podrá ejecutar el proyecto.

-Main factura:



```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Source History <default config> Search (Ctrl+F)
Projects Services Files
Main.java
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 factura;
7
8 import Vista.VistaGeneral;
9
10 /**
11  * @author kellypaltin
12  */
13 public class Main {
14
15     /**
16      * @param args the command line arguments
17     */
18     public static void main(String[] args) {
19         VistaGeneral vistaGeneral = new VistaGeneral();
20         vistaGeneral.menu();
21     }
22 }
23
24
25
```



Examen

06/03/2021

Para concluir, ahí esta una pequeña ejecución del programa en donde muestra la creación y listando de un cliente si fuera a comprar un producto como en este caso helados con las características necesarias de aquel producto para la gestión de la factura.

-Ejecución:

```
ant -f C:\Users\Fdo\Documents\NetBeansProjects\Factura -Dnb.internal.action.name=run run
init!
Deleting: C:\Users\Fdo\Documents\NetBeansProjects\Factura\build\built-jar.properties
deps-jar:
Updating property file: C:\Users\Fdo\Documents\NetBeansProjects\Factura\build\built-jar.properties
compile:
run:
Gestión de Factura
 1. Datos del Cliente
 2. Datos del Producto
 3. Factura
 4. Salir
 1
 1. Crear
 2. Actualizar
 3. Buscar
 4. Listar
 5. Eliminar
 6. Salir
 1
Ingresar:
Nombre:
kelly
Cedula:
03
Fiabilidad de Pago:
excelente
Res:true
 1. Crear
 2. Actualizar
 3. Buscar
 4. Listar
 5. Eliminar
 4
Cliente(id=1, nombre=kelly, cedula=03, fiabilidadPago=excelente)
 1. Crear
 2. Actualizar
 3. Buscar
 4. Listar
 5. Eliminar
 6. Salir
 6
Gestión de Factura
 1. Datos del Cliente
 2. Datos del Producto
 3. Factura
 4. Salir
 2
 1. Crear
 2. Actualizar
 3. Buscar
 4. Listar
 5. Eliminar
 6. Salir
 1
Ingresar:
Descripción:
alimento
Precio Unitario:
1,5
Stock:
10
```



Examen

06/03/2021

```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Output - Factura (run)
1,5
Stock:
10
IVA:
0,12
Res:true
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Salir
4
Producto(id=1, descripcion=alimento, precioUnitario=1.5, stock=10, iva=0.12)
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Salir
6
Gestión de Factura
1. Datos del Cliente
2. Datos del Producto
3. Factura
4. Salir
3
Ingresé:
Cedula:
03
Cliente(id=1, nombre=kelly, cedula=03, fiabilidadPago=excelente)
1. Crear
2. Actualizar

Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Output - Factura (run)
Cliente(id=1, nombre=kelly, cedula=03, fiabilidadPago=excelente)
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Borrado
7. Salir
1
Ingresé:
Producto:
helado
Cantidad:
3
Valor:
4,5
Total:
5,04
Fecha (dd/mm/yyyy):
07/06/2021
Res:true
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Borrado
7. Salir
4
Factura(id=1, producto=helado, cantidad=3, valor=4,5, total=5,04, fechaFactura=Thu Jan 07 00:06:00 COT 2021, cliente=Cliente(id=1, nombre=kelly, cedula=03, fiabilidadPago=excelente), dat
1. Crear
2. Actualizar
3. Buscar

Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Output - Factura (run-single) [running...]
(1 more...) 111:114 | INS
```



Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

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

Output - Factura (run)

```
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Borrado
7. Salir

4
Factura(id=1, producto=helado, cantidad=3, valor=4.5, total=5.04, fechaFactura=Thu Jan 07 00:06:00 COT 2021, cliente=Cliente(id=1, nombre=kelly, cedula=03, fiabilidadPago=excelente), dat
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Borrado
7. Salir

6
Si desea borrar su listado seleccione la opcion de eliminar
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Borrado
7. Salir

7
Gestión de Factura
1. Datos del Cliente
2. Datos del Producto
3. Factura
4. Salir
4
BUILD SUCCESSFUL (total time: 2 minutes 30 seconds)
```

Output

Factura (run-single) running... (1 more...) 111:114 INS