



## Examen

06/03/2021

### 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**.



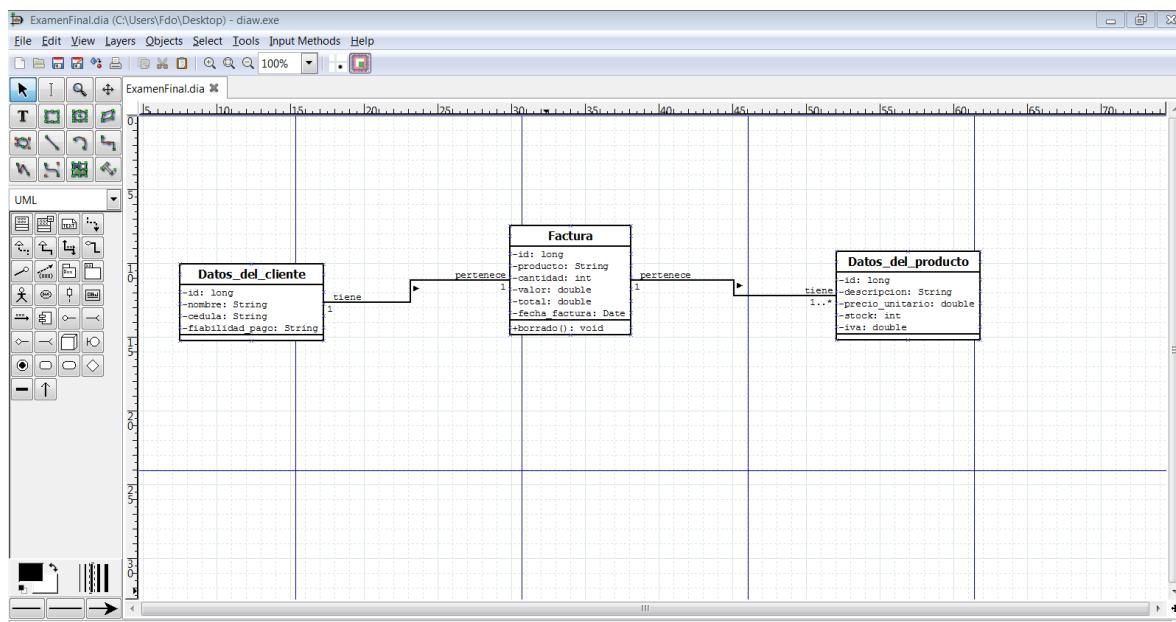
Examen

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:





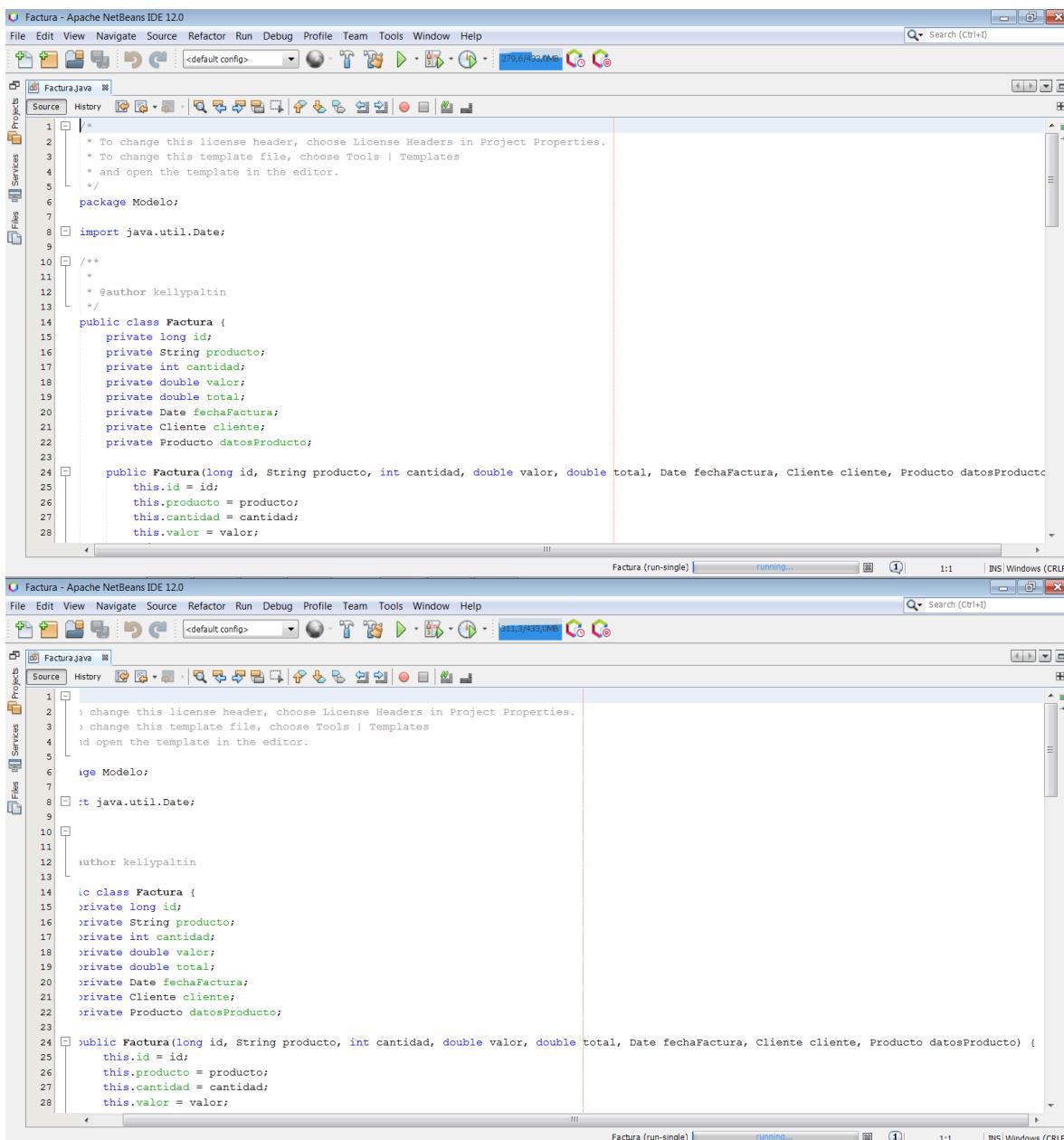
## Examen

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:



```
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
<default config> 275.6/321.0MB 311.9/433.0MB
Projects Services Files
Factura.java [Source]
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
10 /**
11  *
12  * @author kellypaltin
13  */
14 public class Factura {
15     private long id;
16     private String producto;
17     private int cantidad;
18     private double valor;
19     private double total;
20     private Date fechaFactura;
21     private Cliente cliente;
22     private Producto datosProducto;
23
24     public Factura(long id, String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto) {
25         this.id = id;
26         this.producto = producto;
27         this.cantidad = cantidad;
28         this.valor = valor;
}
Factura (run-single) running... 1:1 INS Windows (CRLF)
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
<default config> 311.9/433.0MB
Projects Services Files
Factura.java [Source]
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
10 /**
11  *
12  * @author kellypaltin
13  */
14 public class Factura {
15     private long id;
16     private String producto;
17     private int cantidad;
18     private double valor;
19     private double total;
20     private Date fechaFactura;
21     private Cliente cliente;
22     private Producto datosProducto;
23
24     public Factura(long id, String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto) {
25         this.id = id;
26         this.producto = producto;
27         this.cantidad = cantidad;
28         this.valor = valor;
}
Factura (run-single) running... 1:1 INS Windows (CRLF)
```



## Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
23     public Factura(long id, String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto) {
24         this.id = id;
25         this.producto = producto;
26         this.cantidad = cantidad;
27         this.valor = valor;
28         this.total = total;
29         this.fechaFactura = fechaFactura;
30         this.cliente = cliente;
31         this.datosProducto = datosProducto;
32     }
33
34     public long getId() {
35         return id;
36     }
37
38     public void setId(long id) {
39         this.id = id;
40     }
41
42     public String getProducto() {
43         return producto;
44     }
45
46     public void setProducto(String producto) {
47         this.producto = producto;
48     }
49
50     public int getCantidad() {
51         return cantidad;
52     }
53
54     public double getValor() {
55         return valor;
56     }
57
58     public double getTotal() {
59         return total;
60     }
61
62     public Date getFechaFactura() {
63         return fechaFactura;
64     }
65
66     public void setFechaFactura(Date fechaFactura) {
67         this.fechaFactura = fechaFactura;
68     }
69
70     public Cliente getCliente() {
71         return cliente;
72     }
73
74     public void setCliente(Cliente cliente) {
75         this.cliente = cliente;
76     }
77
78     public Producto getDatosProducto() {
79         return datosProducto;
80     }
81
82     public void setDatosProducto(Producto datosProducto) {
83         this.datosProducto = datosProducto;
84     }
85
86     public void borrado() {
87
88     }
89
90 }
```

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

Factura - Apache NetBeans IDE 12.0

```
74     public Date getFechaFactura() {
75         return fechaFactura;
76     }
77
78     public void setFechaFactura(Date fechaFactura) {
79         this.fechaFactura = fechaFactura;
80     }
81
82     public Cliente getCliente() {
83         return cliente;
84     }
85
86     public void setCliente(Cliente cliente) {
87         this.cliente = cliente;
88     }
89
90     public Producto getDatosProducto() {
91         return datosProducto;
92     }
93
94     public void setDatosProducto(Producto datosProducto) {
95         this.datosProducto = datosProducto;
96     }
97
98     public void borrado() {
99
100    }
101 }
```

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



## Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
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;
}

public void borrado(){
}

@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 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;
}

public void borrado(){
}

@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:**

```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
Source History <default config> 343.1/438.4MB
Factura.java Producto.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 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 }
```

```
Factura - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
Source History <default config> 288.5/438.4MB
Factura.java Producto.java
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

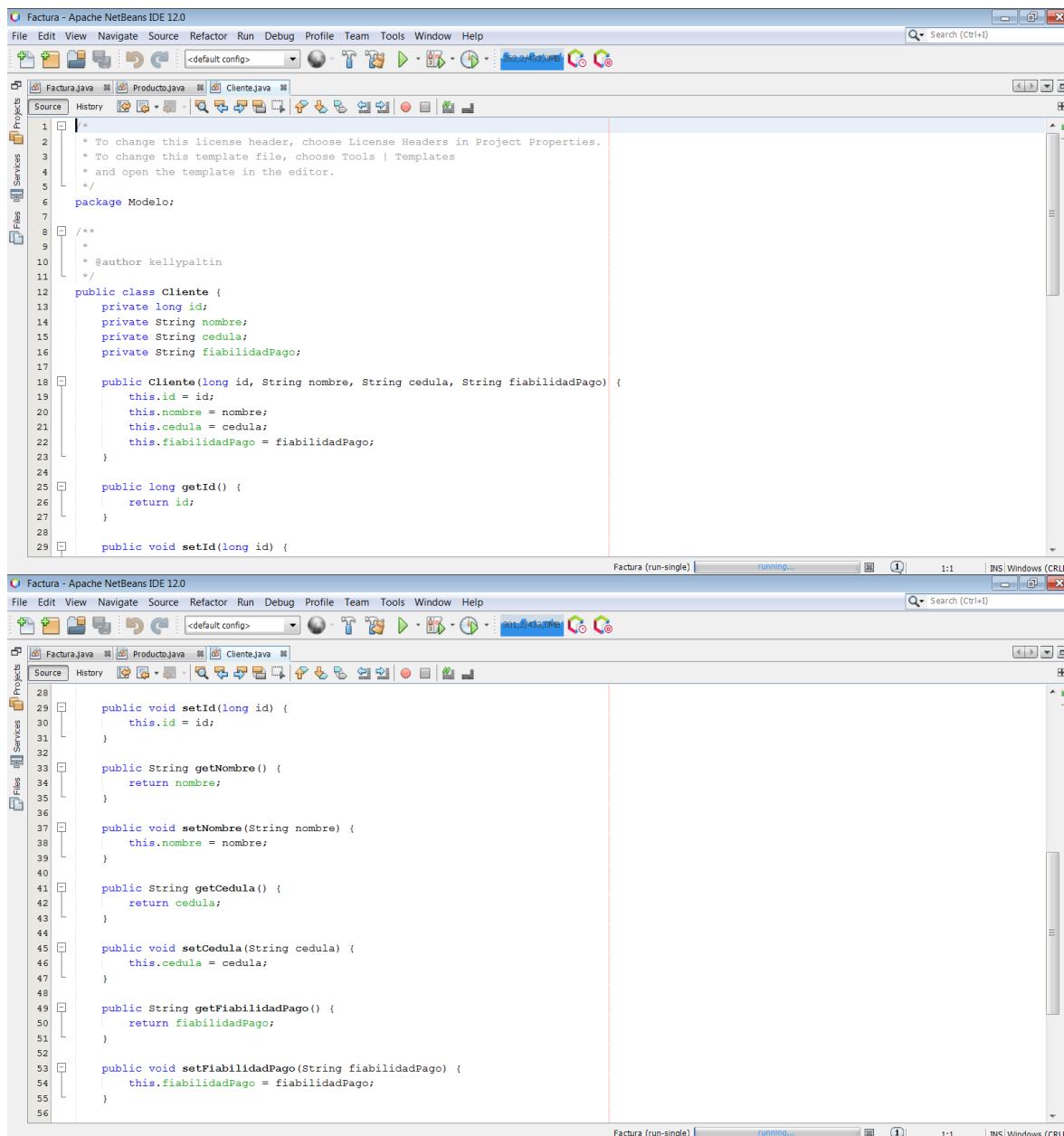
```
44     irn precioUnitario;
45
46
47     void setPrecioUnitario(double precioUnitario) {
48         .precioUnitario = precioUnitario;
49     }
50
51     int getStock() {
52         irn stock;
53     }
54
55     void setStock(int stock) {
56         .stock = stock;
57     }
58
59     double getIva() {
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  /*
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 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
Factura.java Producto.java Cliente.java
Source History <default config> Search (Ctrl+F)
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
57     @Override
58     public String toString() {
59         return "Cliente{" + "id=" + id + ", nombre=" + nombre + ", cedula=" + cedula + ", fiabilidadPago=" + fiabilidadPago + '}';
60     }
61 }
62 }
```

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

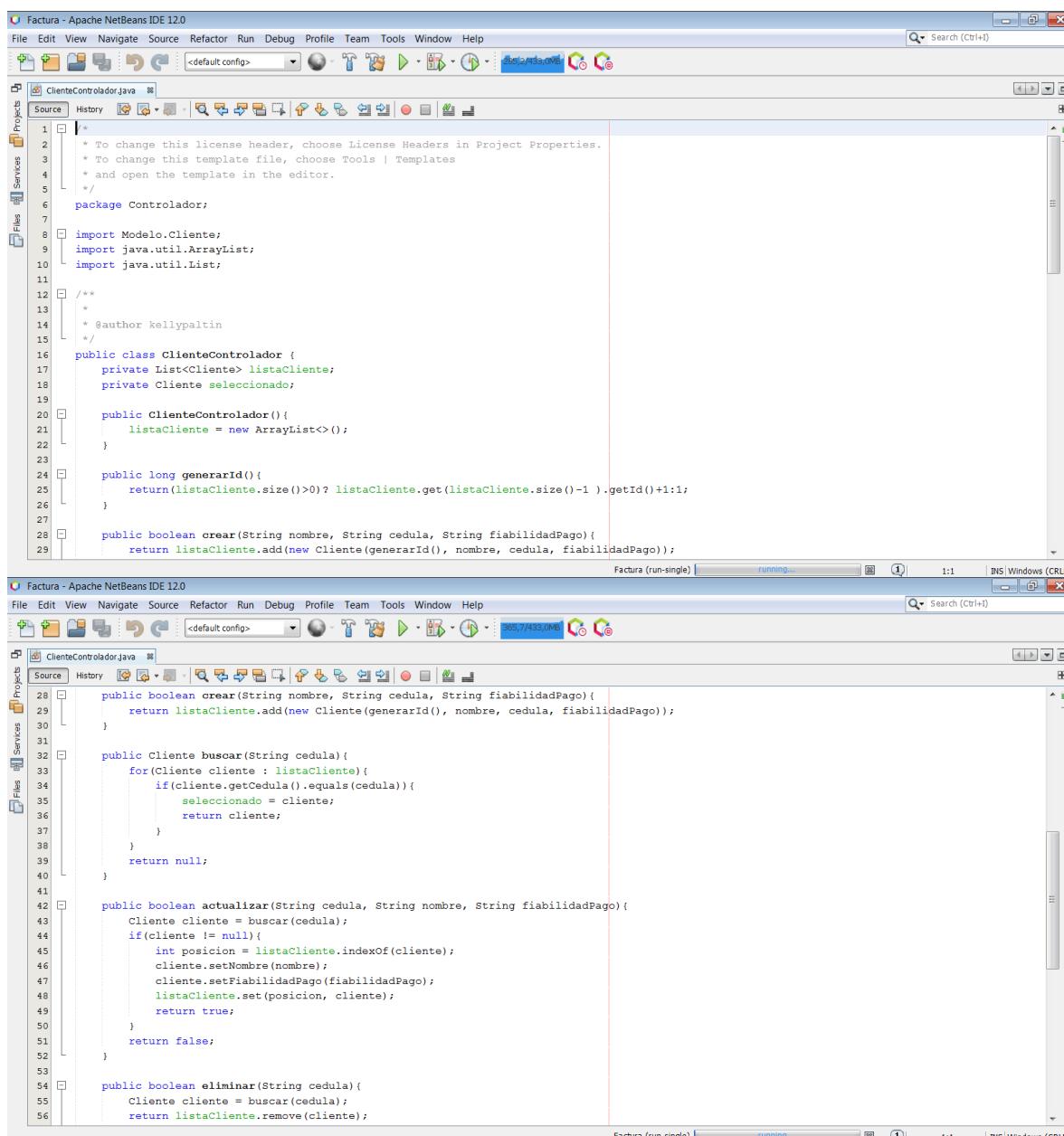


## Examen

06/03/2021

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
Projects Services Files
ClientControlador.java
Source History <default config> Search (Ctrl+I)
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 kellypaltin
15 */
16 public class ClientControlador {
17     private List<Cliente> listaCliente;
18     private Cliente seleccionado;
19
20     public ClientControlador(){
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
32     public Cliente buscar(String cedula){
33         for(Cliente cliente : listaCliente){
34             if(cliente.getCedula().equals(cedula)){
35                 seleccionado = cliente;
36                 return cliente;
37             }
38         }
39         return null;
40     }
41
42     public boolean actualizar(String cedula, String nombre, String fiabilidadPago){
43         Cliente cliente = buscar(cedula);
44         if(cliente != null){
45             int posicion = listaCliente.indexOf(cliente);
46             cliente.setNombre(nombre);
47             cliente.setFiabilidadPago(fiabilidadPago);
48             listaCliente.set(posicion, cliente);
49             return true;
50         }
51         return false;
52     }
53
54     public boolean eliminar(String cedula){
55         Cliente cliente = buscar(cedula);
56         return listaCliente.remove(cliente);
57     }
58 }
```

The screenshot shows two identical windows of the Apache NetBeans IDE displaying the same Java code for a controller class named ClientControlador. The code implements a CRUD interface for managing Cliente objects. It includes methods for generating IDs, creating new clients, searching by ID, updating client information, and deleting clients. The code uses Java 8 features like the var keyword and the Stream API.



## Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
49         return true;
50     }
51     return false;
52 }
53
54 public boolean eliminar(String cedula){
55     Cliente cliente = buscar(cedula);
56     return listaCliente.remove(cliente);
57 }
58
59 public List<Cliente> getListaCliente() {
60     return listaCliente;
61 }
62
63 public void setListaCliente(List<Cliente> listaCliente) {
64     this.listaCliente = listaCliente;
65 }
66
67 public Cliente getSeleccionado() {
68     return seleccionado;
69 }
70
71 public void setSeleccionado(Cliente seleccionado) {
72     this.seleccionado = seleccionado;
73 }
74
75 }
76 }
```

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

### -Controlador factura:

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 Controlador;
7
8 import Modelo.Cliente;
9 import Modelo.Factura;
10 import Modelo.Producto;
11 import java.util.ArrayList;
12 import java.util.Date;
13 import java.util.List;
14
15 /**
16  * 
17  * @author kellypaltin
18 */
19 public class FacturaControlador {
20     private List<Factura> listaFactura;
21     private Factura seleccionado;
22     public double valorIva = 1.12;
23
24     public FacturaControlador(){
25         listaFactura = new ArrayList<>();
26     }
27
28     public long generarId(){
29         return (listaFactura.size()>0)? listaFactura.get(listaFactura.size()-1).getId():1;
```

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



## 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 datosProduct
Factura factura = buscar(producto);
if(factura != null){
    int posicion = listaFactura.indexOf(factura);
    factura.setProducto(producto);
    factura.setCantidad(cantidad);
    factura.setValor(valor);
    factura.setTotal(total);
    factura.setFechaFactura(fechaFactura);
    factura.setDatosProducto(datosProducto);
```

Factura (run-single) | running... | 1:1 | INS

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;
}

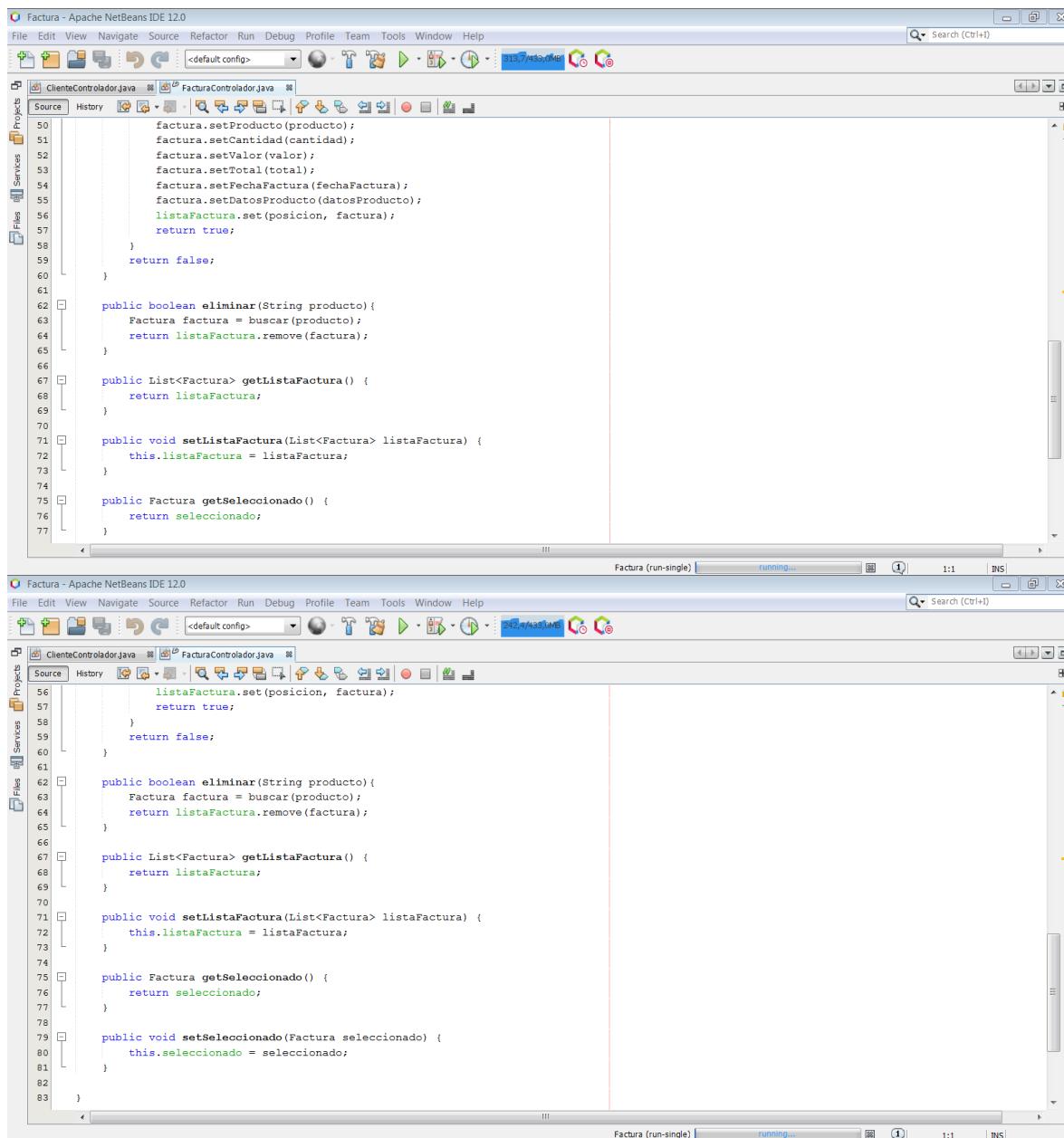
public boolean actualizar(String producto, int cantidad, double valor, double total, Date fechaFactura, Cliente cliente, Producto datosProducto){
Factura factura = buscar(producto);
if(factura != null){
    int posicion = listaFactura.indexOf(factura);
    factura.setProducto(producto);
    factura.setCantidad(cantidad);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

Factura (run-single) | running... | 1:1 | INS | Windows (CR/LF)



## Examen

06/03/2021



The image shows two side-by-side windows of the Apache NetBeans IDE 12.0 interface, both displaying the same Java code for the `FacturaController.java` class.

```
50     factura.setProducto(producto);
51     factura.setCantidad(cantidad);
52     factura.setValor(valor);
53     factura.setTotal(total);
54     factura.setFechaFactura(fechaFactura);
55     factura.setDatosProducto(datosProducto);
56     listaFactura.set(posicion, factura);
57     return true;
58   }
59   return false;
60 }
61 public boolean eliminar(String producto){
62   Factura factura = buscar(producto);
63   return listaFactura.remove(factura);
64 }
65
66 public List<Factura> getListaFactura() {
67   return listaFactura;
68 }
69
70 public void setListaFactura(List<Factura> listaFactura) {
71   this.listaFactura = listaFactura;
72 }
73
74 public Factura getSeleccionado() {
75   return seleccionado;
76 }
77
78 public void setSeleccionado(Factura seleccionado) {
79   this.seleccionado = seleccionado;
80 }
81
82 }
```

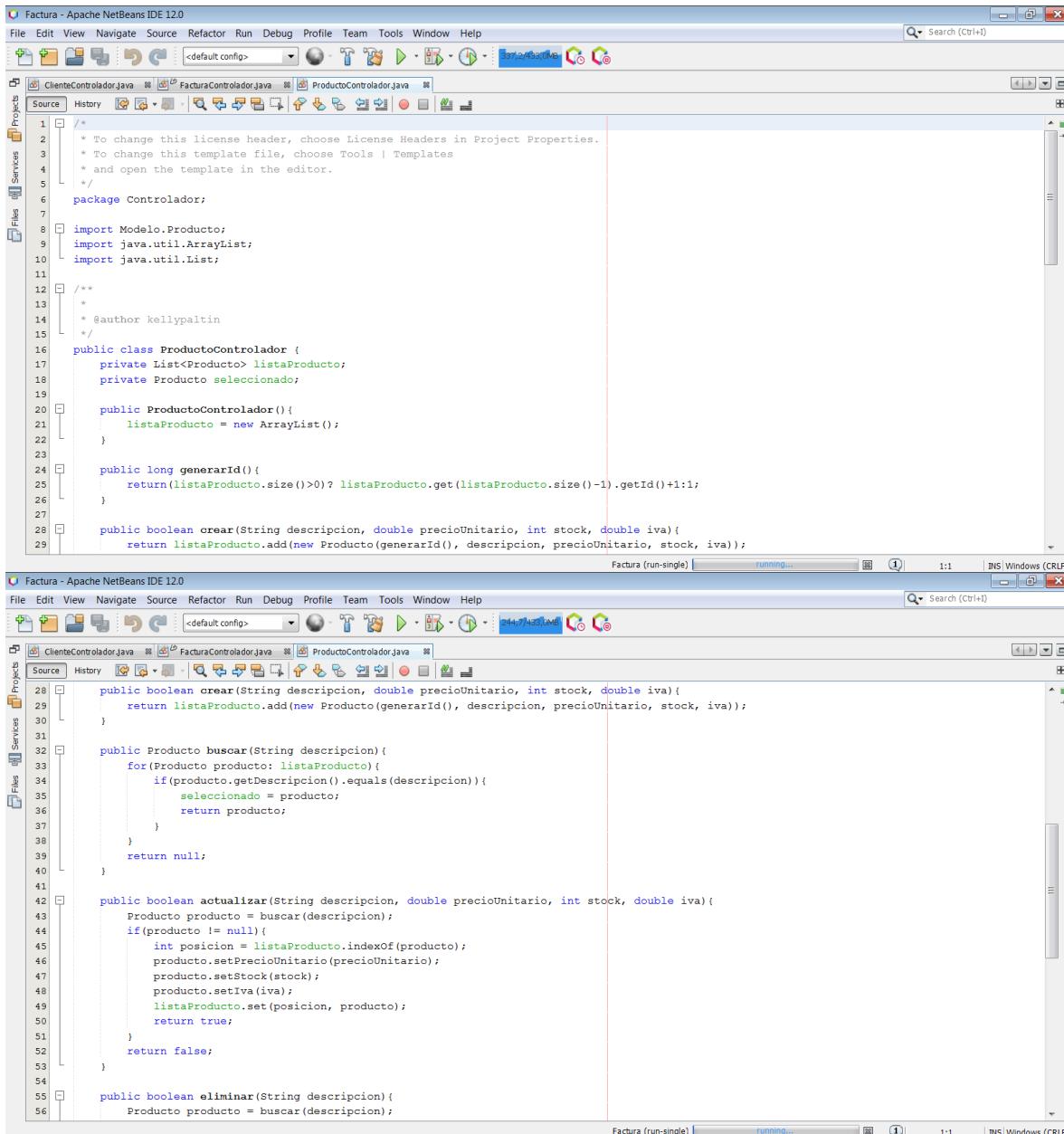
The code implements a controller for managing `Factura` objects. It includes methods for updating a factura's properties, adding it to a list, removing it from the list, getting the list of facturas, setting the list of facturas, and getting the selected factura. The `seleccionado` field is used to store the currently selected factura.



## Examen

06/03/2021

### -Controlador producto:



```
/* 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.Producto;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author kellypaltin
 */
public class ProductoControlador {
    private List<Producto> listaProducto;
    private Producto seleccionado;

    public ProductoControlador() {
        listaProducto = new ArrayList();
    }

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

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

    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);
        if(producto != null){
            listaProducto.remove(producto);
            return true;
        }
        return false;
    }

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

```
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;
}

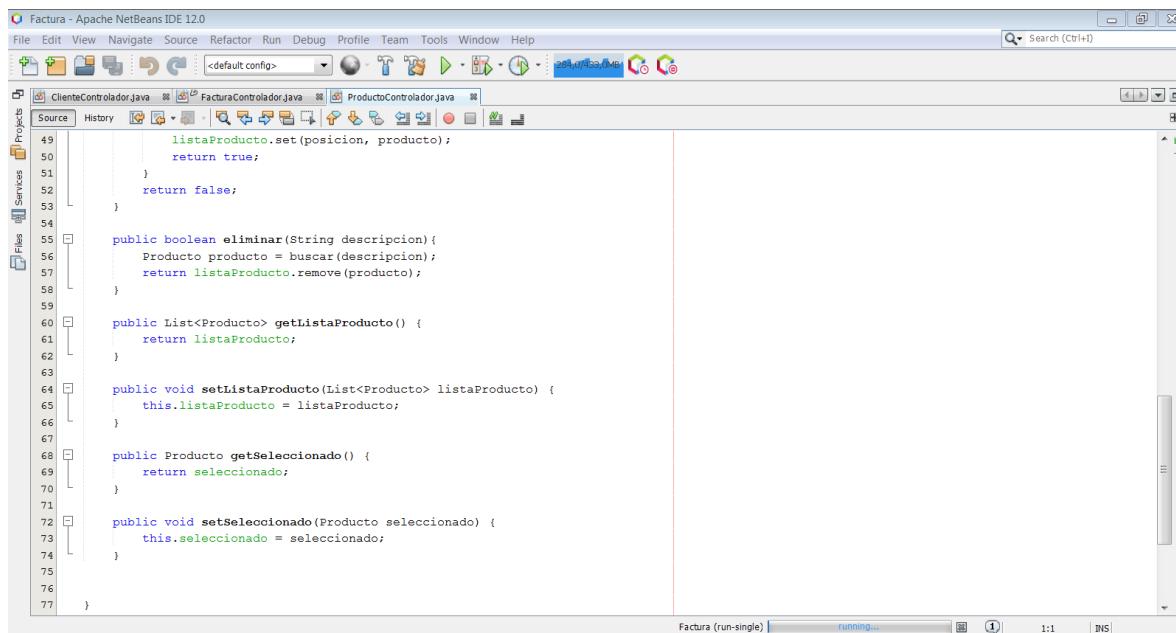
public boolean actualizar(String descripcion, double precioUnitario, int stock, double iva){
    Producto producto = buscar(descripcion);
    if(producto != null){
        int posicion = listaProducto.indexOf(producto);
        producto.setPrecioUnitario(precioUnitario);
        producto.setStock(stock);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

public boolean eliminar(String descripcion){
    Producto producto = buscar(descripcion);
    if(producto != null){
        listaProducto.remove(producto);
        return true;
    }
    return false;
}
```



## Examen

06/03/2021



```
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   public List<Producto> getListadoProducto() {
59     return listaProducto;
60   }

61   public void setListadoProducto(List<Producto> listaProducto) {
62     this.listaProducto = listaProducto;
63   }

64   public Producto getSeleccionado() {
65     return seleccionado;
66   }

67   public void setSeleccionado(Producto seleccionado) {
68     this.seleccionado = seleccionado;
69   }

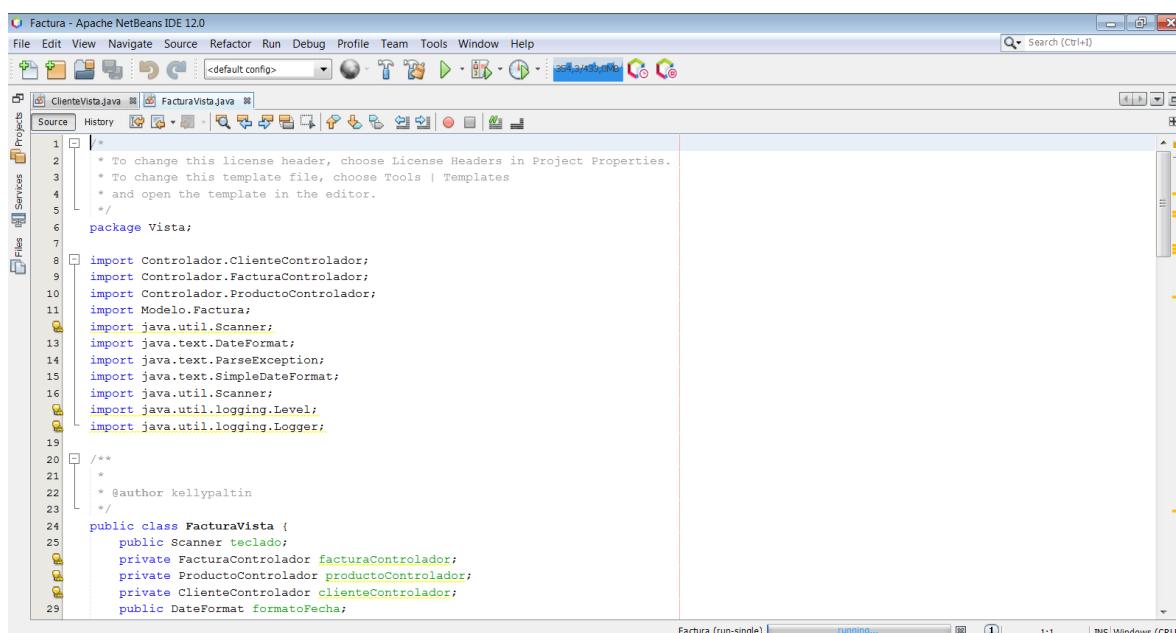
70 }

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 apunte 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.

### -Vista factura:



```
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.ClienteControlador;
9 import Controlador.FacturaControlador;
10 import Controlador.ProductoControlador;
11 import Modelo.Factura;
12 import java.util.Scanner;
13 import java.text.DateFormat;
14 import java.text.ParseException;
15 import java.text.SimpleDateFormat;
16 import java.util.Scanner;
17 import java.util.logging.Level;
18 import java.util.logging.Logger;
19
20 /**
21  *
22  * @author kellypaltin
23  */
24 public class FacturaVista {
25   public Scanner teclado;
26   private FacturaControlador facturaControlador;
27   private ProductoControlador productoControlador;
28   private ClienteControlador clienteControlador;
29   public DateFormat formatoFecha;
```



## 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> 7205485.java 2021-03-06 10:45:00
Projects Services Files
ClientVistaJava FacturaVistaJava
Source
22  * @author kellypaltin
23  */
24  public class FacturaVista {
25      private Scanner teclado;
26      private FacturaControlador facturaControlador;
27      private ProductoControlador productoControlador;
28      private ClienteControlador clienteControlador;
29      public DateFormat formatoFecha;
30
31      public FacturaVista(FacturaControlador facturaControlador, ProductoControlador productoControlador, ClienteControlador clienteControlador) {
32          teclado = new Scanner(System.in);
33          this.facturaControlador = facturaControlador;
34          this.productoControlador = productoControlador;
35          this.clienteControlador = clienteControlador;
36          formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
37      }
38
39      public void menu() {
40          int opcion = 0;
41          do {
42              System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Listar \n 5. Eliminar \n 6. Borrado \n 7. Salir");
43              opcion = teclado.nextInt();
44              switch(opcion){
45                  case 1: this.crear(); break;
46                  case 2: this.actualizar(); break;
47                  case 3: this.buscar(); break;
48                  case 4: this.listar(); break;
49                  case 5: this.eliminar(); break;
50                  case 6: this.borrado(); break;
51              }
52          } while (opcion < 7);
53      }
54
55      public void crear() {
56          System.out.println("Ingrese: \nproducto");
57          String producto = teclado.next();
58          System.out.println("cantidad");
59          int cantidad = teclado.nextInt();
60          System.out.println("valor");
61          double valor = teclado.nextDouble();
62          System.out.println("total");
63          double total = teclado.nextDouble();
64          System.out.println("Fecha (dd/MM/yyyy):");
65          String fechaFactura = teclado.next();
66          try {
67              System.out.println("Res:" + facturaControlador.crear(producto, cantidad, valor, total, formatoFecha.parse(fechaFactura), clienteControlador));
68          }catch (ParseException ex) {
69              System.out.println(ex.getMessage());
70          }
71      }
72  }
```

Factura (run-single) 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
Source History <default config> 7205485.java 2021-03-06 10:45:00
Projects Services Files
ClientVistaJava FacturaVistaJava
Source
43  opcion = teclado.nextInt();
44  switch(opcion){
45      case 1: this.crear(); break;
46      case 2: this.actualizar(); break;
47      case 3: this.buscar(); break;
48      case 4: this.listar(); break;
49      case 5: this.eliminar(); break;
50      case 6: this.borrado(); break;
51  }
52  } while (opcion < 7);
53
54
55  public void crear() {
56      System.out.println("Ingrese: \nproducto");
57      String producto = teclado.next();
58      System.out.println("cantidad");
59      int cantidad = teclado.nextInt();
60      System.out.println("valor");
61      double valor = teclado.nextDouble();
62      System.out.println("total");
63      double total = teclado.nextDouble();
64      System.out.println("Fecha (dd/MM/yyyy):");
65      String fechaFactura = teclado.next();
66      try {
67          System.out.println("Res:" + facturaControlador.crear(producto, cantidad, valor, total, formatoFecha.parse(fechaFactura), clienteControlador));
68      }catch (ParseException ex) {
69          System.out.println(ex.getMessage());
70      }
71  }
72  }
```

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



## Examen

06/03/2021

Factura - Apache NetBeans IDE 12.0

```
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67     >ducto, cantidad, valor, total, formatoFecha.parse(fechaFactura), clienteControlador.getSeleccionado(), productoControlador.getSeleccionado());
68
69
70
```

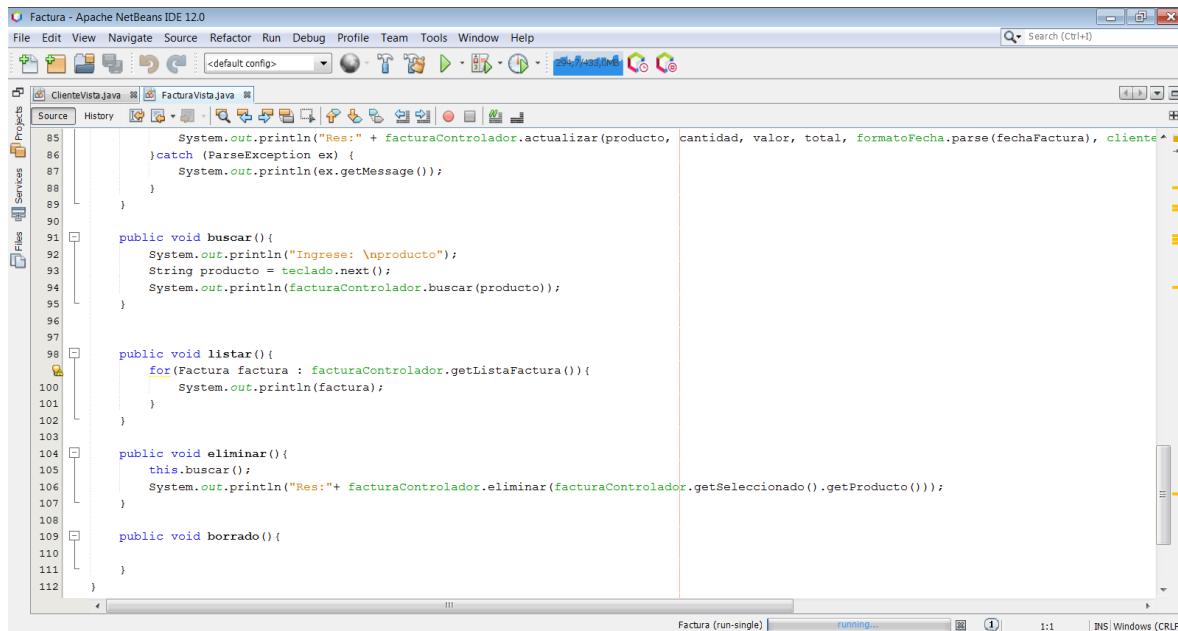
Factura - Apache NetBeans IDE 12.0

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



## Examen

06/03/2021

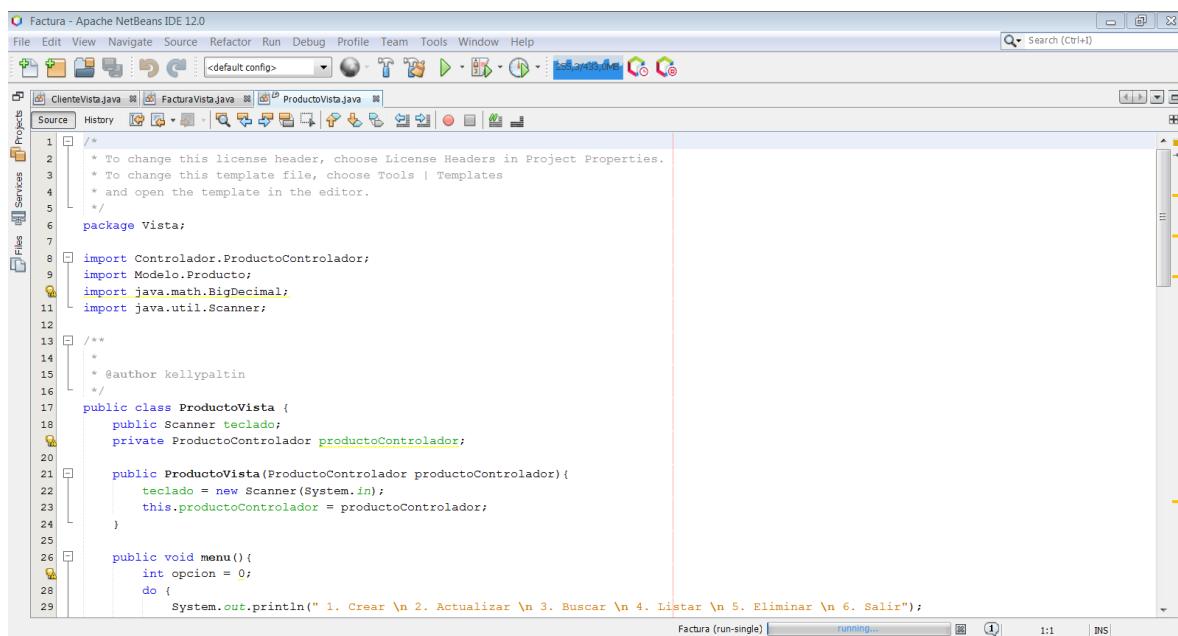


The screenshot shows the Apache NetBeans IDE 12.0 interface with the title bar "Factura - Apache NetBeans IDE 12.0". The main window displays the code for "Factura.java". The code implements a command-line interface for managing facturas. It includes methods for updating, searching, listing, and deleting facturas. The code uses System.out.println statements to output messages and results.

```
85     System.out.println("Res:" + facturaControlador.actualizar(producto, cantidad, valor, total, formatoFecha.parse(fechaFactura), cliente);
86   }catch (ParseException ex) {
87     System.out.println(ex.getMessage());
88   }
89 }
90
91 public void buscar(){
92   System.out.println("Ingrese: \nproducto");
93   String producto = teclado.next();
94   System.out.println(facturaControlador.buscar(producto));
95 }
96
97
98 public void listar(){
99   for(Factura factura : facturaControlador.getListaFactura()){
100     System.out.println(factura);
101   }
102 }
103
104 public void eliminar(){
105   this.buscar();
106   System.out.println("Res:" + facturaControlador.eliminar(facturaControlador.getSeleccionado().getProducto()));
107 }
108
109 public void borrado(){
110 }
111 }
112 }
```

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 main window displays the code for "ProductoVista.java". This class represents a product view, interacting with a "ProductoControlador" object to handle user input via a Scanner and display menu options.

```
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.ProductoControlador;
9 import Modelo.Producto;
10 import java.math.BigDecimal;
11 import java.util.Scanner;
12
13 /**
14  *
15  * @author kellypaltin
16  */
17 public class ProductoVista {
18   public Scanner teclado;
19   private ProductoControlador productoControlador;
20
21   public ProductoVista(ProductoControlador productoControlador){
22     teclado = new Scanner(System.in);
23     this.productoControlador = productoControlador;
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: \n 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.print("Res:" + productoControlador.crear(descripcion, precioUnitario, stock, iva));
}

public void actualizar(){
    System.out.print("Ingrese: \n 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.print("Res:" + productoControlador.actualizar(descripcion, precioUnitario, stock, iva));
}

public Producto buscar(){
    System.out.print("Ingrese: \n descripcion");
    String descripcion = teclado.next();
    System.out.print(productoControlador.buscar(descripcion));
    return productoControlador.buscar(descripcion);
}

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

public void eliminar(){}
```

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

Factura - Apache NetBeans IDE 12.0

```
public void actualizar(){
    System.out.print("Ingrese: \n 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.print("Res:" + productoControlador.actualizar(descripcion, precioUnitario, stock, iva));
}

public Producto buscar(){
    System.out.print("Ingrese: \n descripcion");
    String descripcion = teclado.next();
    System.out.print(productoControlador.buscar(descripcion));
    return productoControlador.buscar(descripcion);
}

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

public void eliminar(){}
```

Factura (run-single) | 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
Projects Services Files
Source History <default config> 2021-03-06 10:43 AM
ClienteVista.java FacturaVistaJava.java ProductoVistaJava.java
58     System.out.println("Stock");
59     int stock = teclado.nextInt();
60     System.out.println("IVA");
61     double iva = teclado.nextDouble();
62     System.out.println("Res:"+ productoControlador.actualizar(descripcion, precioUnitario, stock, iva));
63 }
64
65 public Producto buscar(){
66     System.out.println("Ingrese: \n descripcion");
67     String descripcion = teclado.next();
68     System.out.println(productoControlador.buscar(descripcion));
69     return productoControlador.buscar(descripcion);
70 }
71
72
73 public void listar(){
74     for(Producto producto : productoControlador.getListaProducto()){
75         System.out.println(producto);
76     }
77 }
78
79 public void eliminar(){
80     this.buscar();
81     System.out.println("Res:"+ productoControlador.eliminar(productoControlador.getSeleccionado().getDescripcion()));
82 }
83
84 }
85 
```

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

### -Vista cliente:

Factura - Apache NetBeans IDE 12.0

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Services Files
Source History <default config> 2021-03-06 10:43 AM
ClienteVista.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.ClienteControlador;
9 import Modelo.Cliente;
10 import java.util.Scanner;
11
12 /**
13  * @author kellypaltin
14 */
15 public class ClienteVista {
16     public Scanner teclado;
17     private ClienteControlador clienteControlador;
18
19     public ClienteVista(ClienteControlador clienteControlador){
20         teclado = new Scanner(System.in);
21         this.clienteControlador = clienteControlador;
22     }
23
24     public void menu(){
25         int opcion = 0;
26         do {
27             System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Listar \n 5. Eliminar \n 6. Salir");
28             opcion = teclado.nextInt();
29         } 
```

Factura (run-single) running... 1:1 INS



## 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.println("Ingrese: \n nombre");
    String nombre = teclado.next();
    System.out.println("Cedula");
    String cedula = teclado.next();
    System.out.println("Fiabilidad de Pago");
    String fiabilidadPago = teclado.next();
    System.out.println("Res:"+ clienteControlador.crear(nombre, cedula, fiabilidadPago));
}

public void actualizar(){
    System.out.println("Ingrese: \n nombre");
    String nombre = teclado.next();
    System.out.println("Cedula");
}
```

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

Factura - Apache NetBeans IDE 12.0

```
public void actualizar(){
    System.out.println("Ingrese: \n nombre");
    String nombre = teclado.next();
    System.out.println("Cedula");
    String cedula = teclado.next();
    System.out.println("Fiabilidad de Pago");
    String fiabilidadPago = teclado.next();
    System.out.println("Res:"+ clienteControlador.actualizar(nombre, cedula, fiabilidadPago));
}

public Cliente buscar(){
    System.out.println("Ingrese: \n cedula");
    String cedula = teclado.next();
    System.out.println(clienteControlador.buscar(cedula));
    return clienteControlador.buscar(cedula);
}

public void listar(){
    for(Cliente cliente : clienteControlador.getListaCliente()){
        System.out.println(cliente);
    }
}

public void eliminar(){
    this.buscar();
    System.out.println("Res:"+ clienteControlador.eliminar(clienteControlador.getSeleccionado().getCedula()));
}
```

Factura (run-single) | running... | 1:1 | INS

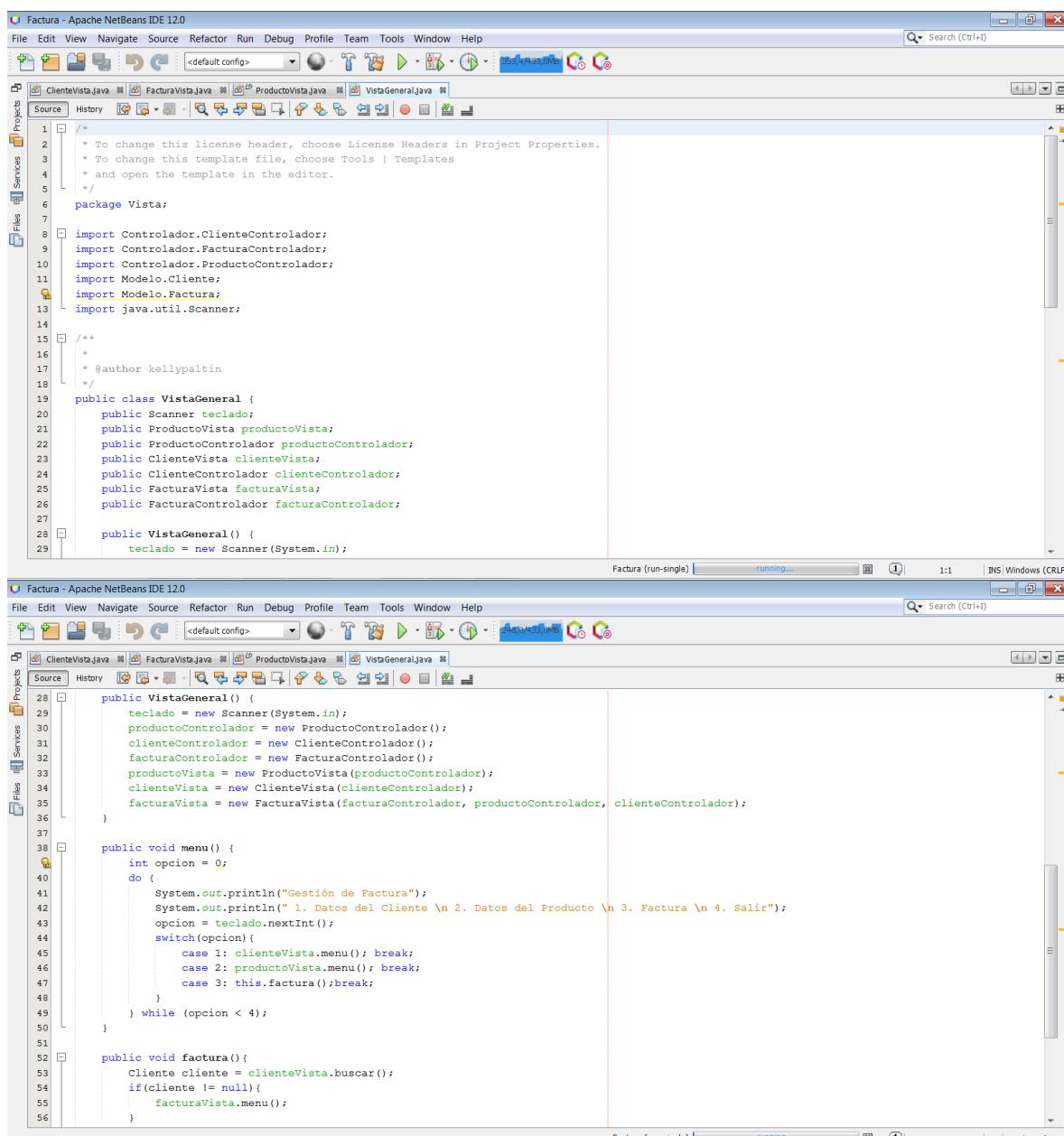


## Examen

06/03/2021

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.

### -Vista general:



The screenshot shows two windows of the Apache NetBeans IDE displaying the same Java code for the `VistaGeneral` class. The code is as follows:

```
/*
 * 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.ClienteControlador;
import Controlador.FacturaControlador;
import Controlador.ProductoControlador;
import Modelo.Cliente;
import Modelo.Factura;
import java.util.Scanner;

/**
 *
 * @author kellypaltin
 */
public class VistaGeneral {
    public Scanner teclado;
    public ProductoVista productoVista;
    public ProductoControlador productoControlador;
    public ClienteVista clienteVista;
    public ClienteControlador clienteControlador;
    public FacturaVista facturaVista;
    public FacturaControlador facturaControlador;

    public VistaGeneral() {
        teclado = new Scanner(System.in);
    }
}

public VistaGeneral() {
    teclado = new Scanner(System.in);
    productoControlador = new ProductoControlador();
    clienteControlador = new ClienteControlador();
    facturaControlador = new FacturaControlador();
    productoVista = new ProductoVista(productoControlador);
    clienteVista = new ClienteVista(clienteControlador);
    facturaVista = new FacturaVista(facturaControlador, productoControlador, clienteControlador);
}

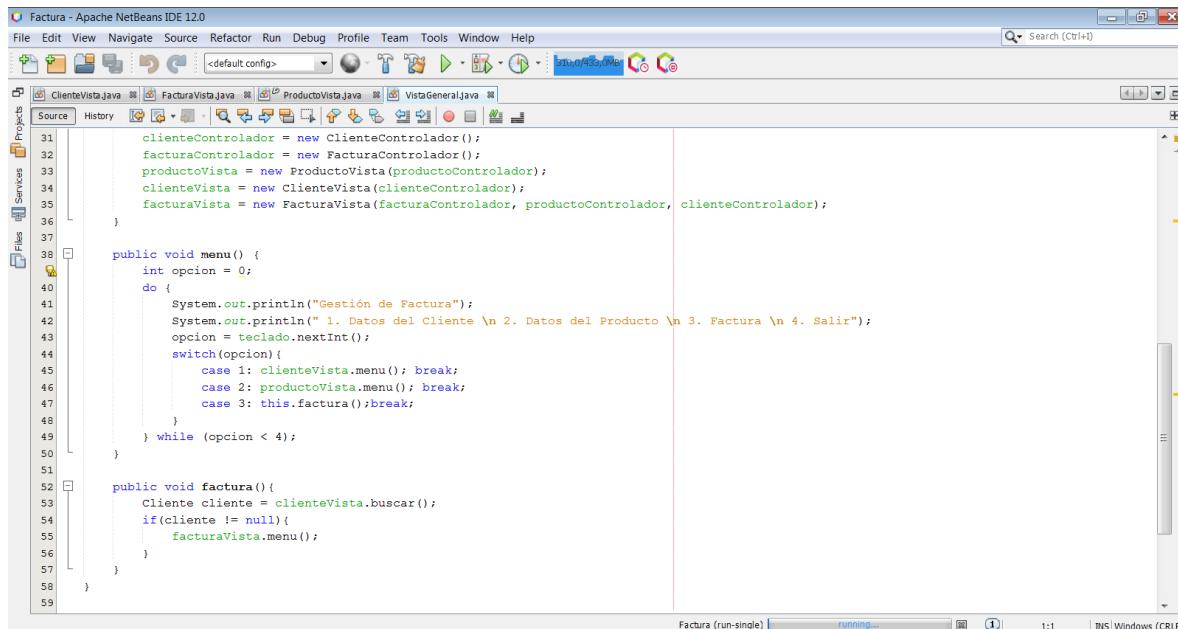
public void menu() {
    int opcion = 0;
    do {
        System.out.println("Gestión de Factura");
        System.out.println(" 1. Datos del Cliente \n 2. Datos del Producto \n 3. Factura \n 4. Salir");
        opcion = teclado.nextInt();
        switch(opcion) {
            case 1: clienteVista.menu();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

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



## Examen

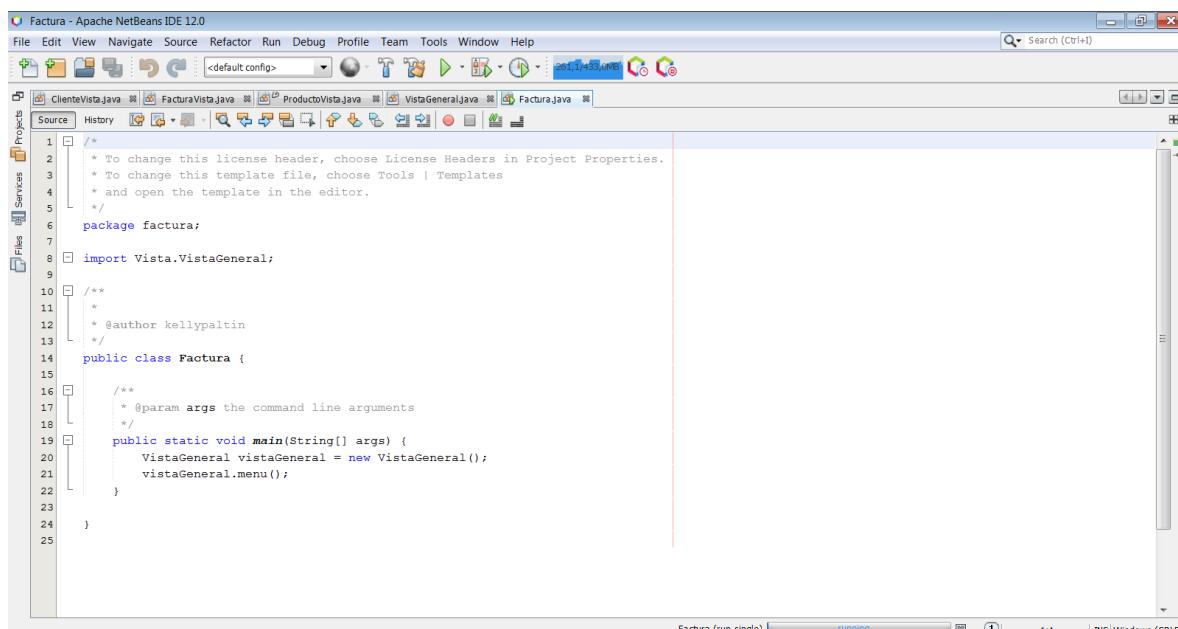
06/03/2021



```
31     clienteControlador = new ClienteControlador();
32     facturaControlador = new FacturaControlador();
33     productoVista = new ProductoVista(productoControlador);
34     clienteVista = new ClienteVista(clienteControlador);
35     facturaVista = new FacturaVista(facturaControlador, productoControlador, clienteControlador);
36 }
37
38 public void menu() {
39     int opcion = 0;
40     do {
41         System.out.println("Gestión de Factura");
42         System.out.println(" 1. Datos del Cliente \n 2. Datos del Producto \n 3. Factura \n 4. Salir");
43         opcion = teclado.nextInt();
44         switch(opcion) {
45             case 1: clienteVista.menu(); break;
46             case 2: productoVista.menu(); break;
47             case 3: this.factura();break;
48         }
49     } while (opcion < 4);
50 }
51
52 public void factura(){
53     Cliente cliente = clienteVista.buscar();
54     if(cliente != null){
55         facturaVista.menu();
56     }
57 }
```

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:



```
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  *
12  * @author kellypaltin
13  */
14 public class Factura {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         VistaGeneral vistaGeneral = new VistaGeneral();
21         vistaGeneral.menu();
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:**

Factura - Apache NetBeans IDE 12.0

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

Output - Factura (run)

```
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
030
Fiabilidad de Pago
debito
Res:true
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
5. Eliminar
<!-->
!!!
```

Factura - Apache NetBeans IDE 12.0

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

Output - Factura (run)

```
Factura (run-single) running... 1:1 INS
```

```
Output - Factura (run)
Res:true
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Salir
4
Cliente(id=1, nombre=kelly, cedula=030, fiabilidadPago=debito)
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
helados
Precio Unitario
<!-->
!!!
```



# **Programación Orientada Objetos**

## **Tema: Clases y Sistemas en Java .**



## 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)

```
descripción
helados
Precio Unitario
1,5
Stock
10
IVA
0,12
Res: true
1. Crear
2. Actualizar
3. Buscar
4. Listar
5. Eliminar
6. Salir
6
Producto(id=1, descripción=helados, 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
030
```

Factura (run-single) running... 1:1 INS

Factura - Apache NetBeans IDE 12.0

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

Output - Factura (run)

```
Ingresé:
cedula
030
Cliente(id=1, nombre=kelly, cedula=030, fiabilidadPago=debito)
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):
06/06/2021
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=Wed Jan 06 00:06:00 COT 2021, cliente=Cliente(id=1, nombre=kelly, cedula=030, fiabilidadPago=debito), datos
```

Output (run-single) running... 1:1 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)

```
3
valor
4.5
total
5.04
Fecha (dd/mm/yyyy) :
06/06/2021
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=Wed Jan 06 00:06:00 COT 2021, cliente=Cliente(id=1, nombre=kelly, cedula=030, fiabilidadPago=debito), datos
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: 1 minute 51 seconds)
```

Output

Factura (run-single) running... 1:1 INS