

Examen 1º Inter ciclo

Programación Orientada a Objetos

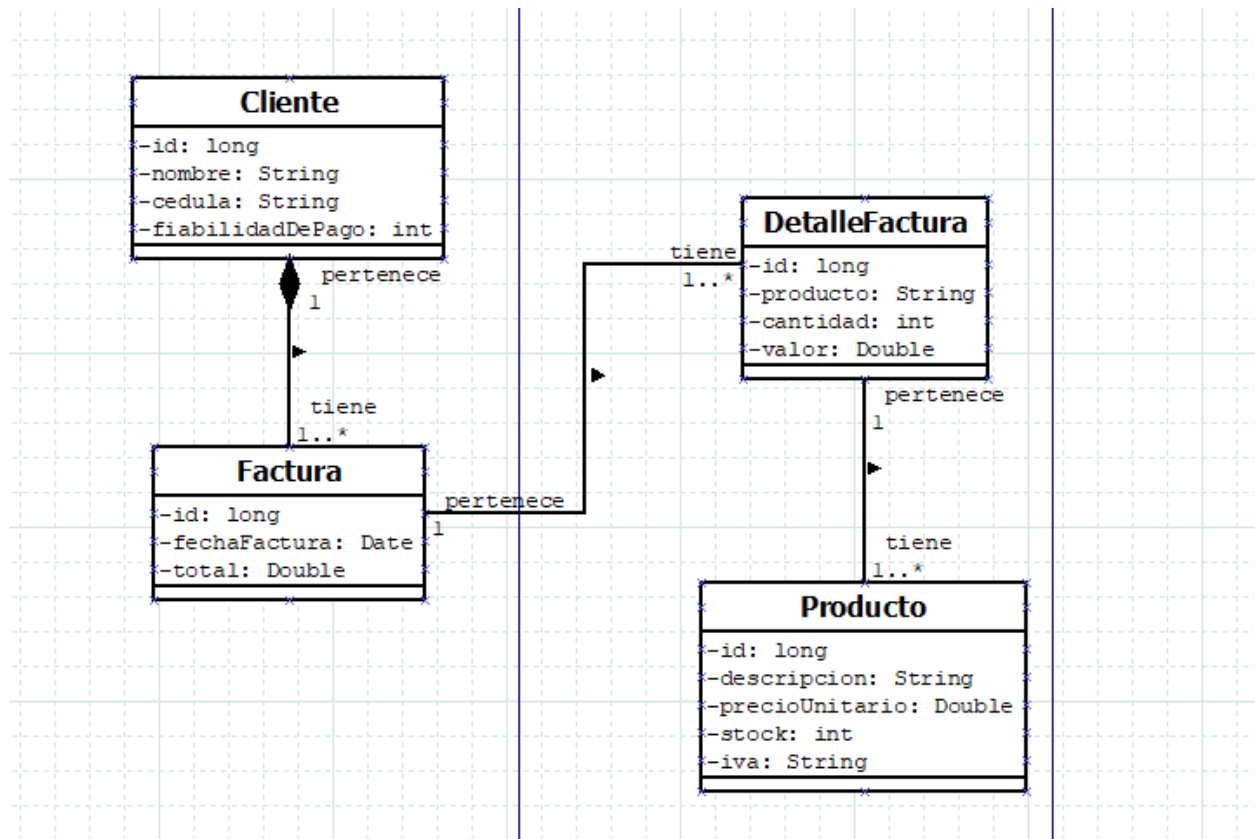
Nombre:

Jaime Viñanzaca

Carrera: Computación

Docente: Lcdo. Vicente Samaniego

DIAGRAMA DE CLASE



CODIGO

```
5
6 package ec.edu.ups.modelo;
7
8 import java.util.ArrayList;
9 import java.util.List;
10
11 /**
12  *
13  * @author jaime andres
14  */
15 public class Cliente
16 {
17     private long id;
18     private String nombre;
19     private String cedula;
20     private String fiabilidadDePago;
21     private Factura factura;
22     private List<Factura> listaFactura;
23
24     public Cliente(long id) {
25         this.id = id;
26     }
27
28     public Cliente(long id, String nombre, String cedula, String fiabilidadDePago) {
29         this.id = id;
30         this.nombre = nombre;
31         this.cedula = cedula;
32         this.fiabilidadDePago = fiabilidadDePago;
33         this.listaFactura = new ArrayList();
34     }
35
36     public Cliente(long id, String nombre, String cedula, String fiabilidadDePago, List<Factura> listaFactura) {
37         this.id = id;
38         this.nombre = nombre;
39         this.cedula = cedula;
40         this.fiabilidadDePago = fiabilidadDePago;
41         this.listaFactura = listaFactura;
42     }
43
44     public long getId() {
45         return id;
46     }
47
48     public String getNombre() {
49         return nombre;
50     }
51
52     public void setNombre(String nombre) {
53         this.nombre = nombre;
54     }
55
56     public String getCedula() {
57         return cedula;
58     }
59
60     public void setCedula(String cedula) {
61         this.cedula = cedula;
62     }
63
64     public String getFiabilidadDePago() {
65         return fiabilidadDePago;
66     }
67
68     public void setFiabilidadDePago(String fiabilidadDePago) {
69         this.fiabilidadDePago = fiabilidadDePago;
70     }
71
72     public List<Factura> getListaFactura() {
73         return listaFactura;
74     }
75
76     public void setListaFactura(List<Factura> listaFactura) {
77         this.listaFactura = listaFactura;
78     }
79
80     @Override
81     public String toString() {
82         return "Cliente(" + "id=" + id + ", nombre=" + nombre + ", cedula=" + cedula + ", fiabilidadDePago=" + fiabilidadDePago + ", listaFactura=" + listaFactura + ')';
83     }
84
85
86
87
88
89 }
```

```

5  */
6  package ec.edu.ups.modelo;
7
8  import java.util.ArrayList;
9  import java.util.Date;
10 import java.util.List;
11
12 /**
13  *
14  * @author jaime andres
15  */
16 public class Factura {
17
18     private long id;
19     private String fechaFactura;
20     private Double total;
21     private Cliente cliente;
22     private List<DetalleFactura> listaDetalleFactura;
23
24     public Factura(long id, String fechaFactura, Double total) {
25         this.id = id;
26         this.fechaFactura = fechaFactura;
27         this.total = total;
28     }
29
30     public Factura(long id, String fechaFactura, Double total, Cliente cliente) {
31         this.id = id;
32         this.fechaFactura = fechaFactura;
33         this.total = total;
34         this.cliente = cliente;
35         this.listaDetalleFactura = new ArrayList();
36     }
37
38     public Factura(long id, String fechaFactura, Double total, Cliente cliente, List<DetalleFactura> listaDetalleFactura) {
39         this.id = id;
40         this.fechaFactura = fechaFactura;
41         this.total = total;
42         this.cliente = cliente;
43         this.listaDetalleFactura = listaDetalleFactura;
44     }
45
46     public long getId() {
47         return id;
48     }
49
50     public void setId(long id) {
51         this.id = id;
52     }
53

```

```

52     }
53
54     public String getFechaFactura() {
55         return fechaFactura;
56     }
57
58     public void setFechaFactura(String fechaFactura) {
59         this.fechaFactura = fechaFactura;
60     }
61
62     public Double getTotal() {
63         return total;
64     }
65
66     public void setTotal(Double total) {
67         this.total = total;
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 List<DetalleFactura> getListaDetalleFactura() {
79         return listaDetalleFactura;
80     }
81
82     public void setListaDetalleFactura(List<DetalleFactura> listaDetalleFactura) {
83         this.listaDetalleFactura = listaDetalleFactura;
84     }
85
86     @Override
87     public String toString() {
88         return "Factura(" + "id=" + id + ", fechaFactura=" + fechaFactura + ", total=" + total + ", cliente=" + cliente + ", listaDetalleFactura=" + listaDetalleFactura + ")";
89     }
90
91
92
93 }
94

```

```

5   */
6  package ec.edu.ups.modelo;
7
8  import java.util.ArrayList;
9  import java.util.List;
10
11  /**
12   *
13   * @author jaime andres
14   */
15 public class DetalleFactura {
16
17     private long id;
18     private String producto;
19     private int cantidad;
20     private Double valor;
21     private Factura factura;
22     private List<Producto> listaProducto;
23
24     public DetalleFactura(long id, String producto, int cantidad, Double valor) {
25         this.id = id;
26         this.producto = producto;
27         this.cantidad = cantidad;
28         this.valor = valor;
29     }
30
31     public DetalleFactura(long id, String producto, int cantidad, Double valor, Factura factura) {
32         this.id = id;
33         this.producto = producto;
34         this.cantidad = cantidad;
35         this.valor = valor;
36         this.factura = factura;
37     }
38
39     public DetalleFactura(long id, String producto, int cantidad, Double valor, List<Producto> listaProducto) {
40         this.id = id;
41         this.producto = producto;
42         this.cantidad = cantidad;
43         this.valor = valor;
44         this.listaProducto = new ArrayList();
45     }

```

```

45     }
46
47     public DetalleFactura(long id, String producto, int cantidad, Double valor, Factura factura, List<Producto> listaProducto) {
48         this.id = id;
49         this.producto = producto;
50         this.cantidad = cantidad;
51         this.valor = valor;
52         this.factura = factura;
53         this.listaProducto = listaProducto;
54     }
55
56     public long getId() {
57         return id;
58     }
59
60     public void setId(long id) {
61         this.id = id;
62     }
63
64     public String getProducto() {
65         return producto;
66     }
67
68     public void setProducto(String producto) {
69         this.producto = producto;
70     }
71
72     public int getCantidad() {
73         return cantidad;
74     }
75
76     public void setCantidad(int cantidad) {
77         this.cantidad = cantidad;
78     }
79
80     public Double getValor() {
81         return valor;
82     }
83
84     public void setValor(Double valor) {
85         this.valor = valor;
86     }
87
88     public Factura getFactura() {
89         return factura;
90     }
91

```

```

91
92     public void setFactura(Factura factura) {
93         this.factura = factura;
94     }
95
96     public List<Producto> getListaProducto() {
97         return listaProducto;
98     }
99
100     public void setListaProducto(List<Producto> listaProducto) {
101         this.listaProducto = listaProducto;
102     }
103
104     @Override
105     public String toString() {
106         return "DetalleFactura(" + "id=" + id + ", producto=" + producto + ", cantidad=" + cantidad + ", valor=" + valor + ", factura=" + factura + ", listaProducto=" + listaProducto + ')';
107     }
108 }
109

```

```

5  //
6  package ec.edu.ups.modelo;
7
8  /**
9   *
10   * @author jaime andres
11   */
12  public class Producto {
13
14      private long id;
15      private String descripcion;
16      private Double precioUnitario;
17      private int stock;
18      private String iva;
19      private DetalleFactura detalleFactura;
20
21      public Producto(long id, String descripcion, Double precioUnitario, int stock, String iva) {
22          this.id = id;
23          this.descripcion = descripcion;
24          this.precioUnitario = precioUnitario;
25          this.stock = stock;
26          this.iva = iva;
27      }
28
29      public Producto(long id, String descripcion, Double precioUnitario, int stock, String iva, DetalleFactura detalleFactura) {
30          this.id = id;
31          this.descripcion = descripcion;
32          this.precioUnitario = precioUnitario;
33          this.stock = stock;
34          this.iva = iva;
35          this.detalleFactura = detalleFactura;
36      }
37
38      public long getId() {
39          return id;
40      }
41
42      public void setId(long id) {
43          this.id = id;
44      }
45
46      public String getDescripcion() {
47          return descripcion;
48      }
49
50      public void setDescripcion(String descripcion) {
51          this.descripcion = descripcion;
52      }

```

```

52  }
53
54      public Double getPrecioUnitario() {
55          return precioUnitario;
56      }
57
58      public void setPrecioUnitario(Double precioUnitario) {
59          this.precioUnitario = precioUnitario;
60      }
61
62      public int getStock() {
63          return stock;
64      }
65
66      public void setStock(int stock) {
67          this.stock = stock;
68      }
69
70      public String getIva() {
71          return iva;
72      }
73
74      public void setIva(String iva) {
75          this.iva = iva;
76      }
77
78      public DetalleFactura getDetalleFactura() {
79          return detalleFactura;
80      }
81
82      public void setDetalleFactura(DetalleFactura detalleFactura) {
83          this.detalleFactura = detalleFactura;
84      }
85
86      @Override
87      public String toString() {
88          return "Producto{" + "id=" + id + ", descripcion=" + descripcion + ", precioUnitario=" + precioUnitario + ", stock=" + stock + ", iva=" + iva + '}';
89      }
90
91  }
92

```

```

5  /*
6  package ec.edu.ups.controlador;
7
8  import ec.edu.ups.modelo.Cliente;
9  import ec.edu.ups.modelo.Factura;
10 import java.util.ArrayList;
11 import java.util.List;
12
13 /**
14  *
15  * @author jaime andres
16  */
17 public class ControladorCliente {
18
19     private List<Cliente> listaCliente;
20     private Cliente seleccionado;
21
22     public ControladorCliente() {
23         listaCliente = new ArrayList<Cliente>();
24         seleccionado = null;
25     }
26
27     public long generarId(){
28         return (listaCliente.size() > 0)? listaCliente.get(listaCliente.size()-1).getId()+1 : 1;
29     }
30
31     public boolean crear(String nombre, String cedula, String fiabilidadDePago)
32     {
33         Cliente cliente = new Cliente(this.generarId(), nombre, cedula, fiabilidadDePago);
34         return listaCliente.add(cliente);
35     }
36
37     public Cliente buscar(String Cedula)
38     {
39         for(Cliente cliente : listaCliente)
40         {
41             if(cliente.getCedula().equals(Cedula))
42             {
43                 return cliente;
44             }
45         }
46         return null;
47     }

```



```

48
49     public boolean actualizar(String nombre, String cedula, String fiabilidadDePago)
50     {
51         Cliente cliente = this.buscar(cedula);
52         if(cliente != null)
53         {
54             int posicion = listaCliente.indexOf(cliente);
55             cliente.setNombre(nombre);
56             cliente.setCedula(cedula);
57             cliente.setFiabilidadDePago(fiabilidadDePago);
58             listaCliente.set(posicion, cliente);
59             return true;
60         }
61         return false;
62     }
63
64     public boolean eliminar(String Cedula)
65     {
66         Cliente cliente = this.buscar(Cedula);
67         if(cliente != null)
68             return listaCliente.remove(cliente);
69         return false;
70     }
71
72     public void imprimir(){
73         for (Cliente client : listaCliente){
74             System.out.println("cliente");
75         }
76     }
77
78     public boolean agregarFactura(Factura factura){
79         return seleccionado.getListFactura().add(factura);
80     }
81
82     public List<Cliente> getListCliente() {
83         return listaCliente;
84     }
85
86     public void setListaCliente(List<Cliente> listaCliente) {
87         this.listaCliente = listaCliente;
88     }
89
90     public Cliente getSeleccionado() {
91         return seleccionado;
92     }
93

```

```

93
94     public void setSeleccionado(Cliente seleccionado) {
95         this.seleccionado = seleccionado;
96     }
97
98
99
100 }
101

```

```

5  /*
6  package ec.edu.ups.controlador;
7
8  import ec.edu.ups.modelo.Cliente;
9  import ec.edu.ups.modelo.DetalleFactura;
10 import ec.edu.ups.modelo.Factura;
11 import java.util.ArrayList;
12 import java.util.Date;
13 import java.util.List;
14
15 /**
16  *
17  * @author jaime andres
18  */
19 public class ControladorFactura {
20
21     private List<Factura> listaFacturas;
22     private List<Cliente> listaClientes;
23     private Factura seleccionada;
24
25     public ControladorFactura(){
26         listaFacturas = new ArrayList();
27         listaClientes = new ArrayList();
28         seleccionada = null;
29     }
30
31     public long generarId(){
32         return (listaFacturas.size() > 0)? listaFacturas.get(listaFacturas.size()-1).getId()+1 : 1;
33     }
34
35     public boolean crear(String fechaFactura, Double total, Cliente cliente){
36         Factura factura = new Factura(this.generarId(), fechaFactura, total, cliente);
37
38         return listaFacturas.add(factura);
39     }
40
41     public Factura buscar(String fechaFactura){
42         for(Factura dato : listaFacturas){
43             if(dato.getFechaFactura().equals(fechaFactura))
44                 return dato;
45         }
46         return null;
47     }
48

```

```

48
49 public boolean actualizar(String fechaFactura, Double total){
50     Factura factura = this.buscar(fechaFactura);
51     if(factura != null){
52         int posicion = listaFactura.indexOf(factura);
53         factura.setFechaFactura(fechaFactura);
54         factura.setTotal(total);
55         listaFactura.set(posicion, factura);
56         return true;
57     }
58     return false;
59 }
60
61 public boolean eliminar(String fechaFactura){
62     Factura factura = this.buscar(fechaFactura);
63     if(factura != null){
64         factura.getCliente().getListaFactura().remove(factura);
65         return listaFactura.remove(factura);
66     }
67     return false;
68 }
69
70
71 public boolean agregarDetalleFactura(DetalleFactura detalleFactura){
72     return seleccionado.getListadetaelleFactura().add(detalleFactura);
73 }
74
75 public List<Factura> getListFactura() {
76     return listaFactura;
77 }
78
79 public void setListaFactura(List<Factura> listaFactura) {
80     this.listaFactura = listaFactura;
81 }
82
83 public Factura getSeleccionado() {
84     return seleccionado;
85 }
86
87 public void setSeleccionado(Factura seleccionado) {
88     this.seleccionado = seleccionado;
89 }
90
91 public List<Cliente> getListCliente() {
92     return listaCliente;
93 }
94

```

```

94
95     public void setListaCliente(List<Cliente> listaCliente) {
96         this.listaCliente = listaCliente;
97     }
98
99
100
101 }
102

```

```

5  /**
6  package ec.edu.ups.controlador;
7
8  import ec.edu.ups.modelo.DetalleFactura;
9  import ec.edu.ups.modelo.Factura;
10 import ec.edu.ups.modelo.Producto;
11 import java.util.ArrayList;
12 import java.util.List;
13
14 /**
15  *
16  * @author jaime andres
17  */
18 public class ControladorDetalleFactura {
19
20     private List<DetalleFactura> listaDetalleFactura;
21     private DetalleFactura seleccionada;
22
23     public ControladorDetalleFactura(){
24         listaDetalleFactura = new ArrayList();
25         seleccionada = null;
26     }
27
28     public long generarId(){
29         return (listaDetalleFactura.size() > 0)? listaDetalleFactura.get(listaDetalleFactura.size()-1).getId()+1 : 1;
30     }
31
32     public boolean crear(String producto, int cantidad, Double valor, Factura factura){
33         DetalleFactura detalleFactura = new DetalleFactura(this.generarId(), producto, cantidad, valor, factura);
34         factura.getListaDetalleFactura().add(detalleFactura);
35         return listaDetalleFactura.add(detalleFactura);
36     }
37
38     public DetalleFactura buscar (String producto){
39         for(DetalleFactura detalleFactura : listaDetalleFactura){
40             if(detalleFactura.getProducto().equals(producto)){
41                 return detalleFactura;
42             }
43         }
44         return null;
45     }
46

```

```

46
47 public boolean actualizar (String producto, int cantidad, Double valor){
48     DetalleFactura detalleFactura = this.buscar(producto);
49     if(detalleFactura != null){
50         int posicion = listaDetalleFactura.indexOf(detalleFactura);
51         detalleFactura.setProducto(producto);
52         detalleFactura.setCantidad(cantidad);
53         detalleFactura.setValor(valor);
54         listaDetalleFactura.set(posicion, detalleFactura);
55         return true;
56     }
57     return false;
58 }
59
60 public boolean eliminar (String producto){
61     DetalleFactura detalleFactura = this.buscar(producto);
62     if(detalleFactura != null)
63         return listaDetalleFactura.remove(detalleFactura);
64     return false;
65 }
66
67 public boolean agregarProducto(Producto producto){
68     return seleccionado.getListProducto().add(producto);
69 }
70
71 public List<DetalleFactura> getListDetalleFactura() {
72     return listaDetalleFactura;
73 }
74
75 public void setListaDetalleFactura(List<DetalleFactura> listaDetalleFactura) {
76     this.listaDetalleFactura = listaDetalleFactura;
77 }
78
79 public DetalleFactura getSeleccionado() {
80     return seleccionado;
81 }
82
83 public void setSeleccionado(DetalleFactura seleccionado) {
84     this.seleccionado = seleccionado;
85 }
86
87
88
89 }
90

```

```

5  /*
6  package ec.edu.ups.controlador;
7
8  import ec.edu.ups.modelo.DetalleFactura;
9  import ec.edu.ups.modelo.Producto;
10 import java.util.ArrayList;
11 import java.util.List;
12
13 /**
14  *
15  * @author jaime andres
16  */
17 public class ControladorProducto {
18
19     private List<Producto> listaProductos;
20     private Producto seleccionada;
21
22     public ControladorProducto(){
23         listaProductos = new ArrayList();
24         seleccionada = null;
25     }
26
27     public long generarId(){
28         return (listaProductos.size() > 0)? listaProductos.get(listaProductos.size()-1).getId()+1 : 1;
29     }
30
31     public boolean crear (String descripcion, Double precioUnitario, int stock, String iva, DetalleFactura detalleFactura){
32         Producto producto = new Producto(this.generarId(), descripcion, precioUnitario, stock, iva, detalleFactura);
33         detalleFactura.getListaProducto().add(producto);
34         return listaProductos.add(producto);
35     }
36
37     public Producto buscar (String descripcion){
38         for(Producto producto : listaProductos){
39             if(producto.getDescripcion().equals(descripcion)){
40                 return producto;
41             }
42         }
43         return null;
44     }
45

```

```

45
46     public boolean actualizar (String descripcionAnterior, String descripcion, Double precioUnitario, int stock, String iva){
47         Producto producto = this.buscar(descripcionAnterior);
48         if(producto != null){
49             int posicion = listaProducto.indexOf(producto);
50             producto.setDescripcion(descripcion);
51             producto.setPrecioUnitario(precioUnitario);
52             producto.setStock(stock);
53             producto.setIva(iva);
54             listaProducto.set(posicion, producto);
55             return true;
56         }
57         return false;
58     }
59
60     public boolean eliminar (String descripcion){
61         Producto producto = this.buscar(descripcion);
62         if(producto != null){
63             producto.getDetalleFactura().getListaProducto().remove(producto);
64             return listaProducto.remove(producto);
65         }
66         return false;
67     }
68
69     public List<Producto> getListaProducto() {
70         return listaProducto;
71     }
72
73     public void setListaProducto(List<Producto> listaProducto) {
74         this.listaProducto = listaProducto;
75     }
76
77     public Producto getSeleccionado() {
78         return seleccionado;
79     }
80
81     public void setSeleccionado(Producto seleccionado) {
82         this.seleccionado = seleccionado;
83     }
84
85
86
87 }
88

```

```

17  /*
18  public class VistaCliente {
19
20      private Scanner teclado;
21      private ControladorFactura controladorFactura;
22      private ControladorCliente controladorCliente;
23
24      public VistaCliente () {
25          this.teclado = new Scanner(System.in);
26          this.controladorCliente = new ControladorCliente();
27          this.controladorFactura = controladorFactura;
28      }
29
30      public void menu(){
31          int opcion = 0;
32          do{
33              System.out.println("-----");
34              System.out.println("Gestion de Clientes");
35              System.out.println("1. Crear");
36              System.out.println("2. Actualizar");
37              System.out.println("3. Buscar");
38              System.out.println("4. Eliminar");
39              System.out.println("5. Listar");
40              System.out.println("6. Salir");
41              System.out.println("-----");
42              opcion = teclado.nextInt();
43              switch(opcion){
44                  case 1:
45                      this.crear();
46                      break;
47                  case 2:
48                      this.actualizar();
49                      break;
50                  case 3:
51                      this.buscar();
52                      break;
53                  case 4:
54                      this.eliminar();
55                      break;
56                  case 5:
57                      System.out.println("listado de Clientes: ");
58                      controladorCliente.imprimir();
59              }
60          }while(opcion < 6);
61      }
62

```



```

62
63 public void crear(){
64     System.out.println("-----");
65     System.out.println("Ingrese los suuiguinetes datos: ");
66     System.out.println("Id: ");
67     long id = teclado.nextLong();
68     System.out.println("Nombre: ");
69     String nombre = teclado.next();
70     System.out.println("Cedula: ");
71     String cedula = teclado.next();
72     System.out.println("Fiabilidad de pago: ");
73     String fiabilidad = teclado.next();
74     boolean resultado = controladorCliente.crear(nombre, cedula, fiabilidad);
75     System.out.println("Cliente creado: "+resultado);
76     System.out.println("-----");
77 }
78
79 public void actualizar(){
80     System.out.println("-----");
81     System.out.println("Nombre: ");
82     String nombre = teclado.next();
83     System.out.println("Cedula: ");
84     String cedula = teclado.next();
85     System.out.println("Fiabilidad: ");
86     String fiabilidad = teclado.next();
87     boolean resultado = controladorCliente.actualizar(nombre, cedula, fiabilidad);
88     System.out.println("Cliente actualizado: "+resultado);
89     System.out.println("-----");
90 }
91
92 public void eliminar(){
93     System.out.println("-----");
94     System.out.println("Eliminar cliente");
95     System.out.println("Cedula: ");
96     String cedula = teclado.next();
97     boolean resultado = controladorCliente.eliminar(cedula);
98     System.out.println("Cliente eliminado: "+resultado);
99     System.out.println("-----");
100 }
101
102 public Cliente buscar(){
103     System.out.println("-----");
104     System.out.println("Cedula: ");
105     String cedula = teclado.next();
106     Cliente cliente = controladorCliente.buscar(cedula);
107     System.out.println(cliente);
108     return cliente;
109 }
110

```

```

110
111     public void asignarSeleccionado(Cliente cliente) {
112         controladorCliente.setSeleccionado(cliente);
113     }
114
115     public ControladorCliente getControladorCliente() {
116         return controladorCliente;
117     }
118
119     public void setControladorCliente(ControladorCliente controladorCliente) {
120         this.controladorCliente = controladorCliente;
121     }
122
123     public ControladorFactura getControladorFactura() {
124         return controladorFactura;
125     }
126
127     public void setControladorFactura(ControladorFactura controladorFactura) {
128         this.controladorFactura = controladorFactura;
129     }
130
131
132 }
133

```

```

5  //
6  package ec.edu.ups.vista;
7
8  import ec.edu.ups.controlador.ControladorCliente;
9  import ec.edu.ups.controlador.ControladorFactura;
10 import ec.edu.ups.modelo.Factura;
11 import java.text.ParseException;
12 import java.text.SimpleDateFormat;
13 import java.util.Date;
14 import java.util.Scanner;
15
16 /**
17  *
18  * @author jaime andres
19  */
20 public class VistaFactura {
21
22     private Scanner scanner;
23     private ControladorCliente controladorCliente;
24     private ControladorFactura controladorFactura;
25     private VistaCliente vistaCliente;
26     private VistaDetalleFactura vistaDetalleFactura;
27     private VistaProducto vistaProducto;
28     private SimpleDateFormat formatoFecha;
29     public static String formato = "dd-MM-yyyy";
30
31     public VistaFactura(VistaCliente vistaCliente, VistaDetalleFactura vistaDetalleFactura, VistaProducto vistaProducto) {
32         controladorFactura = new ControladorFactura();
33         this.vistaCliente = vistaCliente;
34         this.vistaDetalleFactura = vistaDetalleFactura;
35         this.vistaProducto = vistaProducto;
36         scanner = new Scanner(System.in);
37         formatoFecha = new SimpleDateFormat(formato);
38     }
39

```

```

39
40 public void menu() {
41     int opcion = 0;
42     do {
43         System.out.println("1. Crear");
44         System.out.println("2. Actualizar");
45         System.out.println("3. Buscar");
46         System.out.println("4. Eliminar");
47         System.out.println("5. Listar");
48         System.out.println("6. Salir");
49         opcion = teclado.nextInt();
50         switch (opcion) {
51             case 1:
52                 crear();
53                 break;
54             case 2:
55                 actualizar();
56                 break;
57             case 3:
58                 buscar();
59                 break;
60             case 4:
61                 eliminar();
62                 break;
63             case 5:
64                 listar();
65                 break;
66         }
67     } while (opcion < 6);
68 }
69
70
71 public void crear() {
72     System.out.println("-----");
73     System.out.println("Ingrese fecha de emision de la Factura "+formato+" : ");
74     String fechaFactura = teclado.nextLine();
75     System.out.println("Total: ");
76     Double total = teclado.nextDouble();
77     //Date fechaFactura = formatoFecha.parse(teclado.nextLine());
78     System.out.println("Creado: "+ controladorFacturas.crear(fechaFactura, total, controladorCuentas.getSeleccionado()));
79     System.out.println("-----");
80 }

```

```

80     }
81
82     public Factura buscar(){
83         System.out.println("-----");
84         System.out.println("Ingrese fecha "+formato+": ");
85         String fechaFactura = teclado.nextLine();
86         //Date fechaFactura = FormatoFecha.parse(teclado.nextLine());
87         Factura factura = controladorFactura.buscar(fechaFactura);
88         controladorFactura.setSeleccionado(factura);
89         System.out.println(factura);
90         return factura;
91     }
92
93     public void actualizar(){
94         Factura factura = buscar();
95         System.out.println("-----");
96         System.out.println("Ingrese nueva fecha "+formato+": ");
97         String fechaFactura = teclado.nextLine();
98         System.out.println("Ingrese valor nuevo: ");
99         Double valor = teclado.nextDouble();
100        System.out.println("Resultado: "+controladorFactura.actualizar(fechaFactura, valor));
101        System.out.println("-----");
102    }
103
104    public void eliminar(){
105        Factura factura = buscar();
106        System.out.println("-----");
107        System.out.println("Resultado: "+ controladorFactura.eliminar(factura.getFechaFactura()));
108        System.out.println("-----");
109    }
110
111    public void listar(){
112        System.out.println("-----");
113        for(Factura factura : controladorFactura.getListaFactura()){
114            System.out.println("factura");
115        }
116        System.out.println("-----");
117    }
118
119 }
120

```

```

6  package ec.edu.ups.vista;
7
8  import ec.edu.ups.controlador.ControladorDetalleFactura;
9  import ec.edu.ups.controlador.ControladorProducto;
10 import ec.edu.ups.modelo.Producto;
11 import java.text.SimpleDateFormat;
12 import java.util.Scanner;
13
14 /**
15  *
16  * @author jaime andres
17  */
18 public class VistaProducto {
19
20     private ControladorProducto controladorProducto;
21     private ControladorDetalleFactura controladorDetalleFactura;
22     private Scanner teclado;
23
24     public VistaProducto () {
25         this.controladorDetalleFactura = controladorDetalleFactura;
26         controladorProducto = new ControladorProducto();
27         teclado = new Scanner(System.in);
28     }
29

```

```

29
30     public void menu() {
31         int opcion = 0;
32         do {
33             System.out.println("Gestión del Producto");
34             System.out.println("1. Crear");
35             System.out.println("2. Actualizar");
36             System.out.println("3. Buscar");
37             System.out.println("4. Eliminar");
38             System.out.println("5. Listar");
39             System.out.println("6. Salir");
40             opcion = teclado.nextInt();
41             switch(opcion) {
42                 case 1:
43                     this.crear();
44                     break;
45                 case 2:
46                     this.actualizar();
47                     break;
48                 case 3:
49                     this.buscar();
50                     break;
51                 case 4:
52                     this.eliminar();
53                     break;
54                 case 5:
55                     this.imprimir();
56                     break;
57             }
58         } while (opcion < 6);
59     }
60
61     public void crear() {
62         System.out.println("-----");
63         System.out.println("Ingrese descripcion: ");
64         String descripcion = teclado.nextLine();
65         System.out.println("Precio Unitario: ");
66         Double precioUnitario = teclado.nextDouble();
67         System.out.println("Stock: ");
68         int stock = teclado.nextInt();
69         System.out.println("Iva: ");
70         String iva = teclado.next();
71         boolean resultado = controladorProducto.crear(descripcion, precioUnitario, stock, iva, controladorDetalleFactura.getSeleccionado());
72         System.out.println("Creado: "+resultado);
73         System.out.println("-----");
74     }
75

```

```

75
76     public void actualizar(){
77         System.out.println("-----");
78         System.out.println("Ingrese descripcion anterior: ");
79         String descripcionAntigua = scanner.nextLine();
80         System.out.println("Ingrese descripcion nueva: ");
81         String descripcionNueva = scanner.nextLine();
82         System.out.println("Precio Unitario: ");
83         Double precioUnitario = scanner.nextDouble();
84         System.out.println("Stock: ");
85         int stock = scanner.nextInt();
86         System.out.println("Iva: ");
87         String iva = scanner.next();
88         boolean resultado = controladorProducto.actualizar(descripcionAntigua, descripcionNueva, precioUnitario, stock, iva);
89         System.out.println("Actualizado: "+resultado);
90         System.out.println("-----");
91     }
92
93     public Producto buscar(){
94         System.out.println("-----");
95         System.out.println("Ingrese descripcion: ");
96         String descripcion = scanner.nextLine();
97         Producto producto = controladorProducto.buscar(descripcion);
98         System.out.println(producto);
99         return producto;
100     }
101
102     public void eliminar(){
103         System.out.println("-----");
104         System.out.println("Ingrese descripcion: ");
105         String descripcion = scanner.next();
106         boolean resultado = controladorProducto.eliminar(descripcion);
107     }
108
109     public void imprimir(){
110         System.out.println("-----");
111         for(Producto producto : controladorProducto.getListaProducto()){
112             System.out.println(producto);
113         }
114     }
115
116     public void asignarSeleccionado (Producto producto){
117         controladorProducto.setSeleccionado(producto);
118     }
119
120     public ControladorProducto getControladorProducto() {
121         return controladorProducto;
122     }
123

```

```

122     }
123
124     public void setControladorProducto(ControladorProducto controladorProducto) {
125         this.controladorProducto = controladorProducto;
126     }
127
128     public ControladorDetalleFactura getControladorDetalleFactura() {
129         return controladorDetalleFactura;
130     }
131
132     public void setControladorDetalleFactura(ControladorDetalleFactura controladorDetalleFactura) {
133         this.controladorDetalleFactura = controladorDetalleFactura;
134     }
135
136
137
138 }
139

```

```

5  |  */
6  |  package ec.edu.ups.vista;
7  |
8  |  import ec.edu.ups.controlador.ControladorDetalleFactura;
9  |  import ec.edu.ups.controlador.ControladorFactura;
10 |  import ec.edu.ups.modelo.Cliente;
11 |  import ec.edu.ups.modelo.DetalleFactura;
12 |  import java.text.SimpleDateFormat;
13 |  import java.util.Scanner;
14 |
15 |  /**
16 |   *
17 |   * @author jaime andres
18 |   */
19 |  public class VistaDetalleFactura {
20 |
21 |      private ControladorDetalleFactura controladorDetalleFactura;
22 |      private ControladorFactura controladorFactura;
23 |      private Scanner teclado;
24 |      private SimpleDateFormat formatoFecha;
25 |      public static String formato = "dd-MM-yyyy";
26 |
27 |      public VistaDetalleFactura(){
28 |          this.controladorFactura = controladorFactura;
29 |          controladorDetalleFactura = new ControladorDetalleFactura();
30 |          teclado = new Scanner(System.in);
31 |          formatoFecha = new SimpleDateFormat(formato);
32 |      }
33 |

```

```

33
34 public void menu(){
35     int opcion = 0;
36     do {
37         System.out.println("-----");
38         System.out.println("Gestion de Detalle de Facturas");
39         System.out.println("1. Crear");
40         System.out.println("2. Actualizar");
41         System.out.println("3. Buscar");
42         System.out.println("4. Eliminar");
43         System.out.println("5. Listar");
44         System.out.println("6. Salir");
45         System.out.println("-----");
46         opcion = teclado.nextInt();
47         switch(opcion){
48             case 1:
49                 this.crear();
50                 break;
51             case 2:
52                 this.actualizar();
53                 break;
54             case 3:
55                 this.buscar();
56                 break;
57             case 4:
58                 this.eliminar();
59                 break;
60             case 5: this.imprimir();
61                 break;
62         }
63     } while (opcion<6);
64 }
65
66 public void crear(){
67     System.out.println("-----");
68     System.out.println("Ingrese los siguientes datos: ");
69     System.out.println("Producto: ");
70     String producto = teclado.next();
71     System.out.println("Cantidad: ");
72     int cantidad = teclado.nextInt();
73     System.out.println("Valor: ");
74     Double valor = teclado.nextDouble();
75     boolean resultado = controlDetalleFactura.crear(producto, cantidad, valor, controlDetalleFactura.getSeleccionado());
76     System.out.println("Creado: "+resultado);
77     System.out.println("-----");
78 }
79

```



```

79
80     public void actualizar(){
81         System.out.println("-----");
82         System.out.println("Ingrese los datos a actualizar: ");
83         System.out.println("Producto: ");
84         String producto = teclado.next();
85         System.out.println("Cantidad: ");
86         int cantidad = teclado.nextInt();
87         System.out.println("Valor: ");
88         Double valor = teclado.nextDouble();
89         boolean resultado = controladorDetalleFactura.actualizar(producto, cantidad, valor);
90         System.out.println("Actualizado: "+resultado);
91         System.out.println("-----");
92     }
93
94     public DetalleFactura buscar(){
95         System.out.println("-----");
96         System.out.println("Ingrese producto: ");
97         String producto = teclado.next();
98         DetalleFactura detalleFactura = controladorDetalleFactura.buscar(producto);
99         System.out.println(detalleFactura);
100         return detalleFactura;
101     }
102
103     public void eliminar(){
104         System.out.println("-----");
105         System.out.println("Ingrese el producto: ");
106         String producto = teclado.next();
107         boolean resultado = controladorDetalleFactura.eliminar(producto);
108     }
109
110     public void imprimir(){
111         for(DetalleFactura detalleFactura : controladorDetalleFactura.getListaDetalleFactura()){
112             System.out.println(detalleFactura);
113         }
114     }
115
116     public void asignarSeleccionado(DetalleFactura detalleFactura){
117         controladorDetalleFactura.setSeleccionado(detalleFactura);
118     }
119
120     public ControladorDetalleFactura getControladorDetalleFactura() {
121         return controladorDetalleFactura;
122     }
123
124     public void setControladorDetalleFactura(ControladorDetalleFactura controladorDetalleFactura) {
125         this.controladorDetalleFactura = controladorDetalleFactura;
126     }
127

```

```

126     }
127
128     public ControladorFactura getControladorFactura() {
129         return controladorFactura;
130     }
131
132     public void setControladorFactura(ControladorFactura controladorFactura) {
133         this.controladorFactura = controladorFactura;
134     }
135
136
137
138     }
139
140
141

```

```

5  |  /*
6  |  package ec.edu.ups.vista;
7  |
8  |  import java.util.Scanner;
9  |
10 |  /**
11 |   *
12 |   * @author jaime andres
13 |   */
14 |  public class VistaGeneral {
15 |
16 |      private Scanner teclado;
17 |      private VistaFactura vistaFactura;
18 |      private VistaCliente vistaCliente;
19 |      private VistaDetalleFactura vistaDetalleFactura;
20 |      private VistaProducto vistaProducto;
21 |
22 |      public VistaGeneral(){
23 |          vistaCliente = new VistaCliente();
24 |          vistaDetalleFactura = new VistaDetalleFactura();
25 |          vistaProducto = new VistaProducto();
26 |          vistaFactura = new VistaFactura(vistaCliente, vistaDetalleFactura, vistaProducto);
27 |          teclado = new Scanner(System.in);
28 |      }
29 |

```

```

28     }
29
30     public void menu() {
31         int opcion = 0;
32         do {
33             System.out.println("1.- Clientes");
34             System.out.println("2.- Facturas");
35             System.out.println("3.- Detalle Factura");
36             System.out.println("4.- Productos");
37             System.out.println("5.- Salir");
38             opcion = teclado.nextInt();
39             switch (opcion) {
40                 case 1:
41                     vistaCliente.menu();
42                     break;
43                 case 2:
44                     vistaFactura.menu();
45                     break;
46                 case 3:
47                     vistaDetalleFactura.menu();
48                     break;
49                 case 4:
50                     vistaProducto.menu();
51                     break;
52             }
53         } while (opcion < 5);
54     }
55 }
56

```

```
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 examenfactura;
7
8  import ec.edu.ups.vista.VistaGeneral;
9
10 /**
11  *
12  * @author jaime andres
13  */
14 public class ExamenFactura {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21
22         VistaGeneral vistaGeneral = new VistaGeneral();
23         vistaGeneral.menu();
24     }
25
26 }
27
```