Examen 06/03/2021

EXAMEN DE PROGRAMACION ORIENTADA A OBJETOS

Nombre:
ALEX ZUMBA

Fecha:
06/06/2021

Tema:
PROGRAMA FACTURA

Docente:
ING. DIEGO QUISI

Carrera:
COMPUTACION

Programación Orientada Objetos
Tema: Clases y Sistemas en Java .

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 del detalle de la factura son sólo su producto, cantidad, valor.

Cada clase tendrá los métodos para leer y fijar ("set" y "get") todos sus atributos.

También se debe incluir en la clase Factura un método de "Borrado", que no devuelve ni recibe ningún parámetro.

Los productos que venden la empresa son de carácter de computadores y tiene un id, descripción, precio unitario, stock, iva (si se calcula el iva o no).

Los atributos serán privados y tendrán métodos públicos para acceder a ellos. Se pide dibujar el diagrama UML de las clases Factura, Datos_del_cliente y Datos_del_producto y generar el sistema en java para realizar los procesos de CRUD para generar facturas, productos, clientes.

Se calificará el avance con los siguientes criterios de evaluación:

- Diagrama de Clases: 25%
- MVC: 25%
- Usabilidad – Vista - Menús: 25%
- Ejecución, pruebas y sustentación 25%

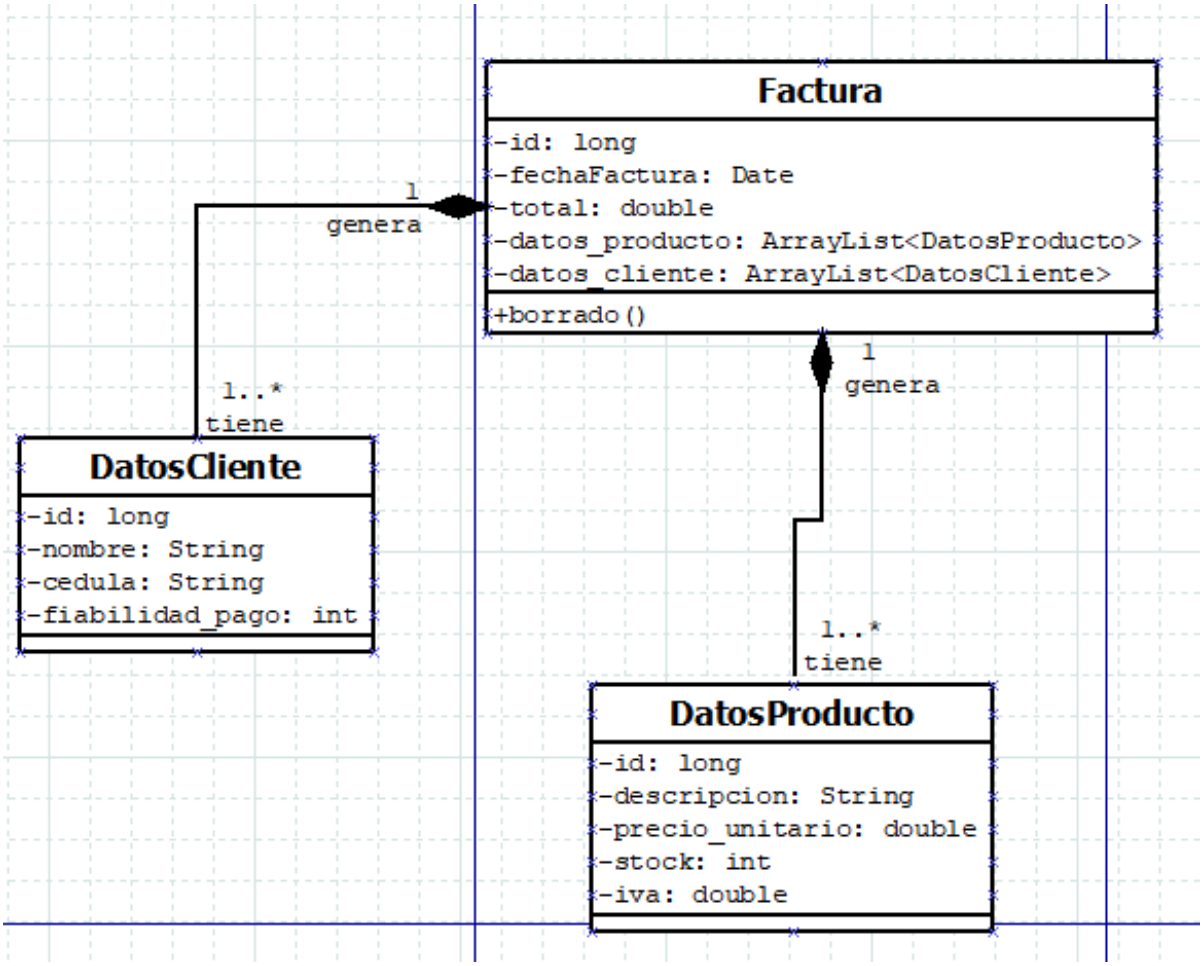
Entrega: Subir al Git el documento en formato PDF de la práctica y código hasta las ***23:55*** del domingo ***06 de junio del 2021***.

Desarrollo del examen:

1. Diagrama de clases

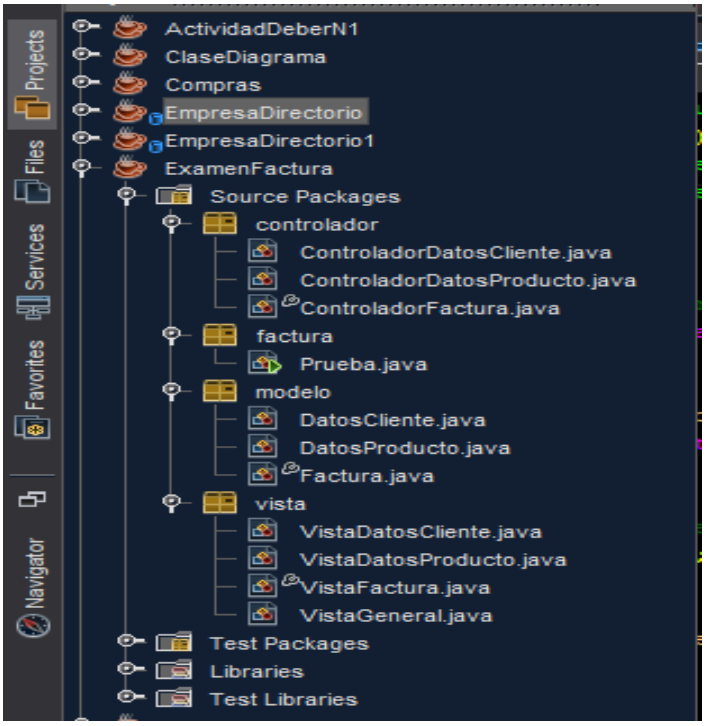
   En este apartado esta la imagen del desarrollo del diagrama de clases en donde se muestra como esta relacionadas las distintas clases.

Examen 06/03/2021



Desarrollo del programa en Java

2. Proyecto
   Implementación del MVC para poder desarrollar correctamente el programa



3. Desarrollo del paquete modelo

   En este paquete esta implementado todas las clases con sus atributos, constructores, getter y setter, que contienen la información que será enviada al controlador.

   - Factura
     La clase Factura, contiene atributos de fecha y total del costo, además de listas de clientes y productos.

Examen 06/03/2021

```java
 * @author Alex Zumba
 */
public class Factura {

    private long id;
    private Date fechaFactura;
    private double total;
    private List<DatosProducto> datosDelProducto;
    private List<DatosCliente> datosDelCliente;

    public Factura() {
    }

    public Factura(long id, Date fechaFactura, double total) {
        this.id = id;
        this.fechaFactura = fechaFactura;
        this.total = total;
    }

    public Factura(long id, Date fechaFactura, double total,
            List<DatosProducto> datosDelProducto, List<DatosCliente> datosDelCliente)
    {
        this.id = id;
        this.fechaFactura = fechaFactura;
        this.total = total;
        this.datosDelProducto = datosDelProducto;
        this.datosDelCliente = datosDelCliente;
    }
```

Examen 06/03/2021

```java
public long getId() {
    return id;
}

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

public Date getFechaFactura() {
    return fechaFactura;
}

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

public double getTotal() {
    return total;
}

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

public List<DatosProducto> getDatosDelProducto() {
    return datosDelProducto;
}

public void setDatosDelProducto(List<DatosProducto> datosDelProducto) {
```

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

public List<DatosProducto> getDatosDelProducto() {
    return datosDelProducto;
}

public void setDatosDelProducto(List<DatosProducto> datosDelProducto) {
    this.datosDelProducto = datosDelProducto;
}

public List<DatosCliente> getDatosDelCliente() {
    return datosDelCliente;
}

public void setDatosDelCliente(List<DatosCliente> datosDelCliente) {
    this.datosDelCliente = datosDelCliente;
}


@Override
public String toString() {
    return "Factura: " + "id: " + id + ", fechaFactura: " + fechaFactura + ", total: " + total;
}
```

- DatosCliente

  La clase DatosCliente contiene atributos, constructores, encapsulamiento y el método toString

Examen 06/03/2021

```java
 * @author Alex Zumba
 */
public class DatosCliente {

    private long id;
    private String nombre;
    private String apellido;
    private String cedula;
    private int fiabilidadPago;

    public DatosCliente() {
    }

    public DatosCliente(long id, String nombre, String apellido,
            String cedula, int fiabilidadPago) {
        this.id = id;
        this.nombre = nombre;
        this.apellido = apellido;
        this.cedula = cedula;
        this.fiabilidadPago = fiabilidadPago;
    }

    public long getId() {
        return id;
    }

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

Examen 06/03/2021

```java
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }

    public String getCedula() {
        return cedula;
    }

    public void setCedula(String cedula) {
        this.cedula = cedula;
    }

    public int getFiabilidadPago() {
        return fiabilidadPago;
    }
```

```java
public String getCedula() {
    return cedula;
}

public void setCedula(String cedula) {
    this.cedula = cedula;
}

public int getFiabilidadPago() {
    return fiabilidadPago;
}

public void setFiabilidadPago(int fiabilidadPago) {
    this.fiabilidadPago = fiabilidadPago;
}

@Override
public String toString() {
    return "DatosCliente: " + "id: " + id + ", nombre: " + nombre + ", apellido: " + apellido
            + ", cedula: " + cedula + ", fiabilidadPago: " + fiabilidadPago;
}
```

- DatosProducto

Examen 06/03/2021

La clase DatosProducto contiene atributos, constructores, encapsulamiento y el método toString

```java
 * @author Alex Zumba
 */
public class DatosProducto {

    private long id;
    private String descripcion;
    private double precioUnitario;
    private int stock;
    private double iva;

    public DatosProducto() {
    }

    public DatosProducto(long id, String descripcion,
            double precioUnitario, int stock, double iva) {
        this.id = id;
        this.descripcion = descripcion;
        this.precioUnitario = precioUnitario;
        this.stock = stock;
        this.iva = iva;
    }

    public long getId() {
        return id;
    }

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

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

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public double getPrecioUnitario() {
        return precioUnitario;
    }

    public void setPrecioUnitario(double precioUnitario) {
        this.precioUnitario = precioUnitario;
    }

    public int getStock() {
        return stock;
    }

    public void setStock(int stock) {
        this.stock = stock;
    }
```

```
    public double getIva() {
        return iva;
    }

    public void setIva(double iva) {
        this.iva = iva;
    }

    @Override
    public String toString() {
        return "DatosProducto: " + "id: " + id + ", descripcion: " + descripcion
                + ", precioUnitario: " + precioUnitario + ", stock: " + stock + ", iva: " + iva;
    }

}
```

4. Desarrollo del paquete Controlador

   En este paquete desarrollamos la información recibida del paquete modelo para procesarla y enviarla a la vista para su ejecución y presentación en consola.

   - ControladorFactura

     En esta clase implementamos los métodos CRUD de la clase Factura

Examen 06/03/2021

```java
 * @author Alex Zumba
 */
public class ControladorFactura {

    private List<Factura> listaFactura;
    private Factura seleccionado;

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

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

    // Creacion de los metodos CRUD

    public boolean crear(Date fechaFactura, double total){
        return listaFactura.add(new Factura(generarId(), fechaFactura, total));
    }

    public Factura buscar(Date fechaFactura){
        for (Factura factura : listaFactura) {
            if(factura.getFechaFactura().equals(fechaFactura)){
                seleccionado = factura;
                return factura;
            } else {
            }
        }
```

```
    public Factura buscar(Date fechaFactura){
        for (Factura factura : listaFactura) {
            if(factura.getFechaFactura().equals(fechaFactura)){
                seleccionado = factura;
                return factura;
            } else {
            }
        }
        return null;
    }

    public boolean actualizar(Date fechaFactura, Date fechaNueva, double total){
        Factura factura = this.buscar(fechaFactura);
        if (factura != null) {
            int posicion = listaFactura.indexOf(factura);
            factura.setFechaFactura(fechaNueva);
            factura.setTotal(total);
            listaFactura.set(posicion, factura);
            return true;
       }
        return false;
    }

    public boolean eliminar(Date fechaFactura){
        Factura factura = buscar(fechaFactura);
        return listaFactura.remove(factura);
    }
```

```
    public boolean agregarCliente(DatosCliente cliente){
        return seleccionado.getDatosDelCliente().add(cliente);
    }

    public boolean agregarProducto(DatosProducto producto){
        return seleccionado.getDatosDelProducto().add(producto);
    }

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

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

    public Factura getSeleccionado() {
        return seleccionado;
    }

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

}
```

- ControladorDatosCliente

Examen 06/03/2021

En esta clase implementamos los métodos CRUD de la clase DatosCliente

```
 * @author Alex Zumba
 */
public class ControladorDatosCliente {

    private List<DatosCliente> listaClientes;
    private DatosCliente seleccionado;

    public ControladorDatosCliente() {
        listaClientes = new ArrayList<>();
    }

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

    // Creacion de los metodos CRUD

    public boolean crear(String nombre, String apellido, String cedula, int fiabilidadPago){
        return listaClientes.add(new DatosCliente(this.generarId(), nombre, apellido, cedula, fiabilidadPago));
    }

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

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

    public boolean eliminar(String cedula){
        DatosCliente cliente = buscar(cedula);
        return listaClientes.remove(cliente);
    }
```

```java
public List<DatosCliente> getListaClientes() {
    return listaClientes;
}

public void setListaClientes(List<DatosCliente> listaClientes) {
    this.listaClientes = listaClientes;
}

public DatosCliente getSeleccionado() {
    return seleccionado;
}

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

- ControladorDatosProdcuto

  En esta clase implementamos los métodos CRUD de la clase DatosProducto

```java
/**
 * @author Alex Zumba
 */
public class ControladorDatosProducto {
    private List<DatosProducto> listaProductos;
    private DatosProducto seleccionado;

    public ControladorDatosProducto() {
        listaProductos = new ArrayList<>();
    }

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

    // Creacion de los metodos CRUD

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

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

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

public boolean eliminar(String descripcion){
    DatosProducto producto = buscar(descripcion);
    return listaProductos.remove(producto);
}
```

```java
public List<DatosProducto> getListaProductos() {
    return listaProductos;
}

public void setListaProductos(List<DatosProducto> listaProductos) {
    this.listaProductos = listaProductos;
}

public DatosProducto getSeleccionado() {
    return seleccionado;
}

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

5. Desarrollo del paquete vista

   En este paquete se desarrolla e implementa la información de los paquetes modelo y controlador, y lo muestra en consola

   - Clase VistaFactura

Examen 06/03/2021

```
 * @author Alex Zumba
 */
public class VistaFactura {

    private ControladorFactura controladorFactura;
    private VistaDatosCliente vistaCliente;
    private VistaDatosProducto vistaProducto;
    private Scanner entrada;
    private DateFormat formatoFecha;

    public VistaFactura(VistaDatosCliente vistaCliente, VistaDatosProducto vistaProducto) {
        controladorFactura = new ControladorFactura();
        entrada = new Scanner(System.in);
        this.vistaCliente = vistaCliente;
        this.vistaProducto = vistaProducto;
        formatoFecha = new SimpleDateFormat("dd/mm/yy");
    }

    public void menu() throws ParseException {
        int opcion = 0;
        do {
            System.out.println("1.- Crear");
            System.out.println("2.- Buscar");
            System.out.println("3.- Actualizar");
            System.out.println("4.- Eliminar");
            System.out.println("5.- Listar");
            System.out.println("6.- Gestionar Cliente");
            System.out.println("7.- Gestionar Producto");
            System.out.println("8.- Salir");
```

```
            switch (opcion) {
                case 1: crear(); break;

                case 2: buscar(); break;
                case 3: actualizar(); break;
                case 4: eliminar(); break;
                case 5: listar(); break;
                case 6: cliente(); break;
                case 7: producto(); break;
                case 8:
                    System.out.println("Finalizado");
                    break;
            }
        } while (opcion < 8);
    }

    public void crear() {
        System.out.print("Ingrese la fecha  (dd/mm/yy): ");
        String fecha = entrada.next();
        System.out.print("Ingrese el total: ");
        double total = entrada.nextDouble();
        try {
            System.out.println("Respuesta: " + controladorFactura.crear(formatoFecha.parse(fecha), total));
        } catch (ParseException ex) {
            System.out.println(ex.getMessage());
        }
    }
```

Examen 06/03/2021

```java
public Factura buscar() throws ParseException {
    System.out.print("Ingrese la fecha  (dd/mm/yy): ");
    String fecha = entrada.next();
    Factura factura = controladorFactura.buscar(formatoFecha.parse(fecha));
    controladorFactura.setSeleccionado(factura);
    System.out.println(factura);
    return factura;
}

public void actualizar() throws ParseException {
    Factura factura = buscar();
    System.out.print("Ingrese la fecha actualizada  (dd/mm/yy): ");
    String fechaNueva = entrada.next();
    System.out.print("Ingrese el total actualizado: ");
    double total = entrada.nextDouble();
    try {
        System.out.println("Resultado: " + controladorFactura.actualizar(factura.getFechaFactura(), formatoFecha.parse(fechaNueva), total));
    } catch (ParseException ex) {
        System.out.println(ex.getMessage());

    }

}
```

```java
public void eliminar() throws ParseException {
    Factura factura = buscar();

    System.out.println("Resultado: " + controladorFactura.eliminar(factura.getFechaFactura()));
}

public void listar() {
    for (Factura factura : controladorFactura.getListaFactura()) {
        System.out.println(factura);
    }
}

public void cliente() throws ParseException {
    Factura factura = buscar();
    if (factura != null) {
        vistaCliente.getControladorCliente().setListaClientes(factura.getDatosDelCliente());
        vistaCliente.menu();
    }
}

public void producto() throws ParseException {
    Factura factura = buscar();
    if (factura != null) {
        vistaProducto.getControladorProducto().setListaProductos(factura.getDatosDelProducto());
        vistaProducto.menu();
    }
}
```

Examen 06/03/2021

```
public ControladorFactura getControladorFactura() {
    return controladorFactura;
}

public void setControladorFactura(ControladorFactura controladorFactura) {
    this.controladorFactura = controladorFactura;
}

public VistaDatosCliente getVistaCliente() {
    return vistaCliente;
}

public void setVistaCliente(VistaDatosCliente vistaCliente) {
    this.vistaCliente = vistaCliente;
}

public VistaDatosProducto getVistaProducto() {
    return vistaProducto;
}

public void setVistaProducto(VistaDatosProducto vistaProducto) {
    this.vistaProducto = vistaProducto;
}
```

- Clase VistaDatosCliente

```
 * @author Alex Zumba
 */
public class VistaDatosCliente {

    private ControladorDatosCliente controladorCliente;
    private Scanner entrada;

    public VistaDatosCliente() {
        controladorCliente = new ControladorDatosCliente();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("1.- Crear");
            System.out.println("2.- Buscar");
            System.out.println("3.- Actualizar");
            System.out.println("4.- Eliminar");
            System.out.println("5.- Listar");
            System.out.println("6.- Salir");

            opcion = entrada.nextInt();

            switch (opcion) {
                case 2: buscar(); break;
                case 3: actualizar(); break;
                case 4: eliminar(); break;
                case 5: listar(); break;
                case 6: System.out.println("Finalizado"); break;
```

Examen 06/03/2021

```
            switch (opcion) {
                case 1: crear(); break;
                case 2: buscar(); break;
                case 3: actualizar(); break;
                case 4: eliminar(); break;
                case 5: listar(); break;
                case 6: System.out.println("Finalizado"); break;
            }
        } while (opcion < 6);
    }

    public void crear() {
        System.out.print("Ingrese el nombre: ");
        String nombre = entrada.next();
        System.out.print("Ingrese el apellido: ");
        String apellido = entrada.next();
        System.out.print("Ingrese la cedula: ");
        String cedula = entrada.next();
        System.out.print("Ingrese la fiabilidad de pago: ");
        int fiabilidad = entrada.nextInt();
        System.out.println("Resultado: " + controladorCliente.crear(nombre, apellido, cedula, fiabilidad));
    }

    public DatosCliente buscar() {
        System.out.print("Ingrese la cedula: ");
        String cedula = entrada.next();
        DatosCliente cliente = controladorCliente.buscar(cedula);
        controladorCliente.setSeleccionado(cliente);
        System.out.println(cliente);
        return cliente;
    }
```

```
    public void actualizar() {
        DatosCliente cliente = buscar();

        System.out.print("Ingrese el nombre: ");
        String nombre = entrada.next();
        System.out.print("Ingrese el apellido: ");
        String apellido = entrada.next();
        System.out.print("Ingrese la cedula: ");
        String cedula = entrada.next();
        System.out.print("Ingrese la fiabilidad de pago: ");
        int fiabilidad = entrada.nextInt();
        System.out.println("Resultado: " + controladorCliente.actualizar(nombre, apellido, cedula, fiabilidad));
    }

    public void eliminar() {
        DatosCliente cliente = buscar();

        if (cliente != null) {
            System.out.println("Resultado: " + controladorCliente.eliminar(cliente.getCedula()));
        }
    }
```

```
public void listar() {
    for (DatosCliente cliente : controladorCliente.getListaClientes()) {
        System.out.println(cliente);
    }
}

public ControladorDatosCliente getControladorCliente() {
    return controladorCliente;
}

public void setControladorCliente(ControladorDatosCliente controladorCliente) {
    this.controladorCliente = controladorCliente;
}
```

- Clase VistaDatosProducto

```
 * @author Alex Zumba
 */
public class VistaDatosProducto {

    private ControladorDatosProducto controladorProducto;
    private Scanner entrada;

    public VistaDatosProducto() {
        controladorProducto = new ControladorDatosProducto();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("1.- Crear");
            System.out.println("2.- Buscar");
            System.out.println("3.- Actualizar");
            System.out.println("4.- Eliminar");
            System.out.println("5.- Listar");
            System.out.println("6.- Salir");

            opcion = entrada.nextInt();
```

Examen 06/03/2021

```
        switch (opcion) {
            case 1: crear(); break;
            case 2: buscar(); break;
            case 3: actualizar(); break;
            case 4: eliminar(); break;
            case 5: listar(); break;
            case 6: System.out.println("Finalizado"); break;
        }
    } while (opcion < 6);
}

public void crear() {
    System.out.print("Ingrese la descripcion: ");
    String descripcion = entrada.next();
    System.out.print("Ingrese el precio unitario: ");
    double precioU = entrada.nextDouble();
    System.out.print("Ingrese el stock: ");
    int stock = entrada.nextInt();
    System.out.print("Ingrese el iva: ");
    double iva = entrada.nextDouble();
    System.out.println("Resultado: " + controladorProducto.crear(descripcion, precioU, stock, iva));
}

public DatosProducto buscar() {
    System.out.print("Ingrese la descripcion: ");
    String descripcion = entrada.next();
    DatosProducto producto = controladorProducto.buscar(descripcion);
    controladorProducto.setSeleccionado(producto);
    System.out.println(producto);
    return producto;
}
```

```
public void actualizar() {
    DatosProducto producto = buscar();

    System.out.print("Ingrese la descripcion: ");
    String descripcion = entrada.next();
    System.out.print("Ingrese el precio unitario: ");
    double precioU = entrada.nextDouble();
    System.out.print("Ingrese el stock: ");
    int stock = entrada.nextInt();
    System.out.print("Ingrese el iva: ");
    double iva = entrada.nextDouble();
    System.out.println("Resultado: " + controladorProducto.actualizar(descripcion, precioU, stock, iva));
}

public void eliminar() {
    DatosProducto producto = buscar();

    if (producto != null) {
        System.out.println("Resultado: " + controladorProducto.eliminar(producto.getDescripcion()));
    }
}

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

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

public ControladorDatosProducto getControladorProducto() {
    return controladorProducto;
}

public void setControladorProducto(ControladorDatosProducto controladorProducto) {
    this.controladorProducto = controladorProducto;
}
}
```

- VistaGeneral

  En esta clase se recibe la información de las de más vistas para que la información sea enviada al main para su impresión.

```
 * @author Alex Zumba
 */
public class VistaGeneral {

    private Scanner entrada;
    private VistaFactura vistaFactura;
    private VistaDatosCliente vistaCliente;
    private VistaDatosProducto vistaProducto;

    public VistaGeneral() {
        vistaCliente = new  VistaDatosCliente();
        vistaProducto = new VistaDatosProducto();
        vistaFactura = new VistaFactura(vistaCliente, vistaProducto);
        entrada = new Scanner(System.in);

    }

    public void menu() throws ParseException {
        int opcion = 0;
        do {
            System.out.println("\n***** MENU PRINCIPAL *****");
            System.out.println("1.- Facturas");
            System.out.println("2.- Clientes");
            System.out.println("3.- Productos");
            System.out.println("4.- Salir");

            opcion = entrada.nextInt();
```

```
            switch (opcion) {
                case 1:
                    vistaFactura.menu();
                    break;
                case 2:
                    vistaCliente.menu();
                            break;
                case 3:
                    vistaProducto.menu();
                    break;
                case 4:
                    System.out.println("PROGRAMA FINALIZADO");
                    break;
            }
        } while (opcion < 4);
    }
```

6. Paquete factura

   En este paquete esta la clase principal, que recibe la información de los demás paquetes y la muestra en pantalla

   - Clase Prueba

```
package factura;

import java.text.ParseException;
import vista.VistaGeneral;

/**
 *
 * @author Alex Zumba
 */
public class Prueba {

    public static void main(String[] args) throws ParseException {

        VistaGeneral vistaGeneral = new VistaGeneral();
        vistaGeneral.menu();
    }
}
```

7. Muestra de la ejecución del programa

   - Gestión de Facturas

     Crear

```
***** MENU PRINCIPAL *****
1.- Facturas
2.- Clientes
3.- Productos
4.- Salir
1

+++++Creacion de Facturas+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Gestionar Cliente
7.- Gestionar Producto
8.- Salir
1
Ingrese la fecha  (dd/mm/yy): 20/11/2020
Ingrese el total: 20
Respuesta: true
```

```
+++++Creacion de Facturas+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Gestionar Cliente
7.- Gestionar Producto
8.- Salir
1
Ingrese la fecha  (dd/mm/yy): 15/07/2020
Ingrese el total: 18
Respuesta: true
```

     Buscar

Examen 06/03/2021

```
+++++Creacion de Facturas+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Gestionar Cliente
7.- Gestionar Producto
8.- Salir
2
Ingrese la fecha  (dd/mm/yy): 17/11/2020
Factura: id: 3, fechaFactura: Fri Jan 17 00:11:00 COT 2020, total: 34.0
```

Actualizar

```
+++++Creacion de Facturas+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Gestionar Cliente
7.- Gestionar Producto
8.- Salir
3
Ingrese la fecha  (dd/mm/yy): 20/11/2020
Factura: id: 1, fechaFactura: Mon Jan 20 00:11:00 COT 2020, total: 20.0
Ingrese la fecha actualizada  (dd/mm/yy): 06/06/2021
Ingrese el total actualizado: 15
Resultado: true
```

Eliminar

```
+++++Creacion de Facturas+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Gestionar Cliente
7.- Gestionar Producto
8.- Salir
4
Ingrese la fecha  (dd/mm/yy): 11/09/2019
Factura: id: 2, fechaFactura: Fri Jan 11 00:09:00 COT 2019, total: 24.0
Resultado: true
```

Listar

```
+++++Creacion de Facturas+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Gestionar Cliente
7.- Gestionar Producto
8.- Salir
5
Factura: id: 1, fechaFactura: Wed Jan 06 00:06:00 COT 2021, total: 15.0
Factura: id: 3, fechaFactura: Fri Jan 17 00:11:00 COT 2020, total: 34.0
Factura: id: 4, fechaFactura: Wed Jan 15 00:07:00 COT 2020, total: 18.0
```

- Gestión Clientes

Crear

```
***** MENU PRINCIPAL *****
1.- Facturas
2.- Clientes
3.- Productos
4.- Salir
2

+++++Creacion de Clientes+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el nombre: Alex
Ingrese el apellido: Zumba
Ingrese la cedula: 0106333651
Ingrese la fiabilidad de pago: 1
Resultado: true
```

Examen 06/03/2021

```
+++++Creacion de Clientes+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el nombre: Piter
Ingrese el apellido: Suarez
Ingrese la cedula: 0106988471
Ingrese la fiabilidad de pago: 2
Resultado: true
```

Buscar

```
+++++Creacion de Clientes+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese la cedula: 0106333651
DatosCliente: id: 1, nombre: Alex, apellido: Zumba, cedula: 0106333651, fiabilidadPago: 1
```

Actualizar

```
+++++Creacion de Clientes+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
3
Ingrese la cedula: 0157
DatosCliente: id: 3, nombre: Arturo, apellido: Perez, cedula: 0157, fiabilidadPago: 2
Ingrese el nombre: Artur
Ingrese el apellido: Gomez
Ingrese la fiabilidad de pago: 2
Resultado: true
```

Eliminar

Examen 06/03/2021

```
+++++Creacion de Clientes+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
4
Ingrese la cedula: 01087461
DatosCliente: id: 5, nombre: Lina, apellido: Gomez, cedula: 01087461, fiabilidadPago: 4
Resultado: true
```

Listar

```
+++++Creacion de Clientes+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
5
DatosCliente: id: 1, nombre: Alex, apellido: Zumba, cedula: 0106333651, fiabilidadPago: 1
DatosCliente: id: 2, nombre: Piter, apellido: Suarez, cedula: 0106988471, fiabilidadPago: 2
DatosCliente: id: 3, nombre: Artur, apellido: Gomez, cedula: 0255, fiabilidadPago: 2
DatosCliente: id: 4, nombre: Teo, apellido: Martinez, cedula: 01098742, fiabilidadPago: 3
```

- Gestión Productos

Crear

```
+++++Creacion de Productos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese la descripcion: chocolate
Ingrese el precio unitario: 0,50
Ingrese el stock: 20
Ingrese el iva: 0,03
Resultado: true
```

Examen 06/03/2021

```
+++++Creacion de Productos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese la descripcion: arroz
Ingrese el precio unitario: 3
Ingrese el stock: 10
Ingrese el iva: 0,10
Resultado: true
```

Buscar

```
+++++Creacion de Productos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese la descripcion: agua
DatosProducto: id: 2, descripcion: agua, precioUnitario: 0.75, stock: 10, iva: 0.008
```

Actualizar

```
+++++Creacion de Productos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
3
Ingrese la descripcion: chocolate
DatosProducto: id: 1, descripcion: chocolate, precioUnitario: 0.5, stock: 20, iva: 2.0
Ingrese la descripcion nueva: manicho
Ingrese el precio unitario: 0,50
Ingrese el stock: 25
Ingrese el iva: 0,03
Resultado: true
```

Mostrar

Examen 06/03/2021

Eliminar

```
+++++Creacion de Productos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
4
Ingrese la descripcion: arroz
DatosProducto: id: 4, descripcion: arroz, precioUnitario: 3.0, stock: 10, iva: 0.1
Resultado: true
```

Listar

```
+++++Creacion de Productos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
5
DatosProducto: id: 1, descripcion: manicho, precioUnitario: 0.5, stock: 25, iva: 0.03

DatosProducto: id: 2, descripcion: agua, precioUnitario: 0.75, stock: 10, iva: 0.008
DatosProducto: id: 3, descripcion: Galletas, precioUnitario: 1.0, stock: 20, iva: 0.04
DatosProducto: id: 5, descripcion: chocolate, precioUnitario: 0.5, stock: 20, iva: 0.03
DatosProducto: id: 6, descripcion: agua, precioUnitario: 0.75, stock: 10, iva: 0.008
DatosProducto: id: 7, descripcion: galleta, precioUnitario: 1.0, stock: 20, iva: 0.04
DatosProducto: id: 8, descripcion: arroz, precioUnitario: 3.0, stock: 14, iva: 0.1

+++++Creacion de Productos+++++
```

Programa finalizado

Examen 06/03/2021

```
+++++Creacion de Productos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
6
Finalizado

***** MENU PRINCIPAL *****
1.- Facturas
2.- Clientes
3.- Productos
4.- Salir
4

******PROGRAMA FINALIZADO*****
```

Nombre: Alex Fabricio Zumba Sangurima