Prueba2

OBJETIVO:

- Consolidar los conocimientos adquiridos en clase sobre Java.
- Desarrollo de una aplicación utilizando el paradigma de programación orientado a objetos.

OBJETIVO ALCANZADO:

- Abstraer objetos del mundo real y los modelarlos mediante diagramas de clase.
- Construir clases utilizando el paradigma de programación orientada a objetos estudiados en la unidad.

ACTIVIDADES A DESARROLLAR

1) Desarrollar un diagrama de clases que permita gestionar PROYECTOS.

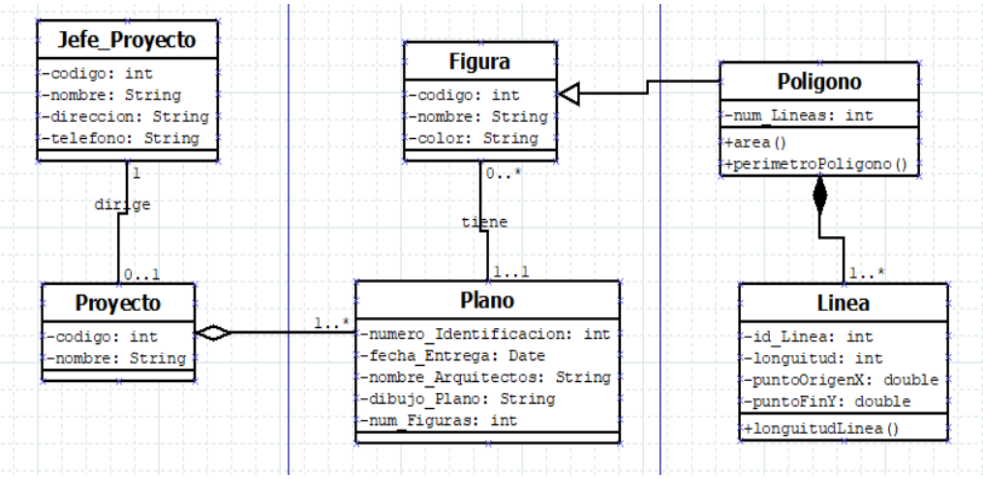
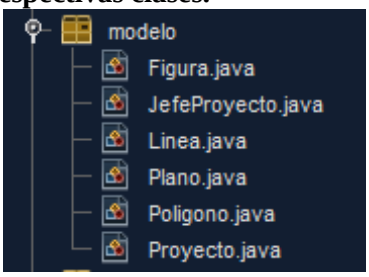


Figura 1: Diagrama de clases para el sistema de Proyectos

2) **Creación de un proyecto de Java en Netbeans. Creación del paquete modelo con sus respectivas clases.**



3) Creación de las relaciones dentro del paquete modelo y creación de sus clases con sus respectivos atributos, getters y setter:

- **3.1**-Se procede a realizar la relación de asociación de la clase **Proyecto** a **JefeProyecto**.

Prueba2

- **3.2-**Un **proyecto** se compone de una serie de **planos**, pero éstos se quieren guardar de modo independiente al proyecto.

```java
public class Proyecto {

    private int codigo;
    private String nombre;
    private JefeProyecto jefeProyecto;

    //Atributo de composicion
    private List<Plano> plano;

    //Constructores
    public Proyecto(int codigo, String nombre, JefeProyecto jefeProyecto, List<Plano>
        this.codigo = codigo;
        this.nombre = nombre;
        this.jefeProyecto = jefeProyecto;
        this.plano = plano;
    }

    public Proyecto(int codigo, String nombre) {
        this.codigo = codigo;
        this.nombre = nombre;
    }
```

```java
*/
public class JefeProyecto  {
    private int id;
    private String nombre;
    private String direccion;
    private String telefono;
    private Proyecto proyecto;

    //Constructores
    public JefeProyecto(int id, String nombre, String direccion, String telefono, Proyecto proyecto) {
        this.id = id;
        this.nombre = nombre;
        this.direccion = direccion;
        this.telefono = telefono;
        this.proyecto = proyecto;
    }

    public JefeProyecto(int id, String nombre, String direccion, String telefono){
        this.id = id;
        this.nombre = nombre;
        this.direccion = direccion;
        this.telefono = telefono;
    }
```

```java
public class Plano {

    private int num_Identificacion;
    private Date fecha_Entrega;
    private String nombre_Arquitectos;
    private  String dibujo_Plano;
    private int num_Figuras;

    //Constructor
    public Plano(int num_Identificacion, Date fecha_Entrega, String nombre_Arquitectos, St
        this.num_Identificacion = num_Identificacion;
        this.fecha_Entrega = fecha_Entrega;
        this.nombre_Arquitectos = nombre_Arquitectos;
        this.dibujo_Plano = dibujo_Plano;
        this.num_Figuras = num_Figuras;
    }
```

- **3.3-** Los planos tienen figuras.

```java
*/
public class Plano {

    private int num_Identificacion;
    private Date fecha_Entrega;
    private String nombre_Arquitectos;
    private  String dibujo_Plano;
    private int num_Figuras;

    //Constructor
    public Plano(int num_Identificacion, Date fecha_Entrega, St
        this.num_Identificacion = num_Identificacion;
        this.fecha_Entrega = fecha_Entrega;
        this.nombre_Arquitectos = nombre_Arquitectos;
        this.dibujo_Plano = dibujo_Plano;
        this.num_Figuras = num_Figuras;
    }

   //Metodos Get y Set
    public int getNum_Identificacion() {
        return num_Identificacion;
    }

    public void setNum_Identificacion(int num_Identificacion) {
        this.num_Identificacion = num_Identificacion;
    }
```

Prueba2

- 3.4- Creación de la clase polígono y su respectiva relación con la clase Figura.

```java
public class Figura{

    private int codigo;
    private String nombre;
    private String color;

    //Constructores
    public Figura(int codigo, String nombre, String color) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.color = color;
    }

     public Figura( String nombre, String color) {
        this.nombre = nombre;
        this.color = color;
    }

    //Metodos Get y Set
    public int getCodigo() {
        return codigo;
    }
```

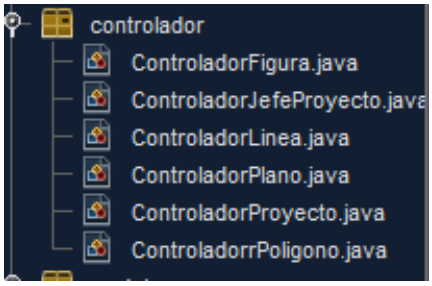
```java
 */
public class Poligono extends Figura{
    private int num_Lineas;

    //Constructor

    public Poligono(int codigo, String nombre, String color) {
        super(codigo, nombre, color);
    }

    public Poligono(int codigo,int num_Lineas, String nombre, String col
       super(codigo, nombre, color);
    }

    //Metodos Get y Set
    public int getNum_Lineas() {
        return num_Lineas;
    }

    public void setNum_Lineas(int num_Lineas) {
        this.num_Lineas = num_Lineas;
    }

    @Override
    public String toString() {
        return "Poligono{" + "Num_Lineas=" + num_Lineas + '}';
    }

}
```

- **3.5-** Cada línea que forma parte de un polígono, creación de la clase línea con su relación de composición hacia la clase polígono.

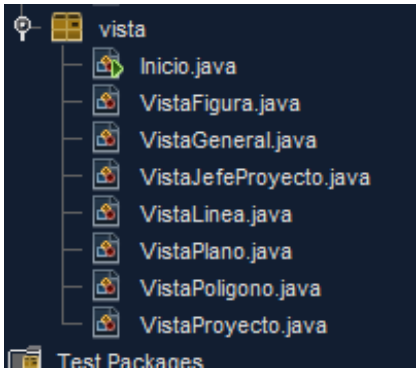
```java
public class Linea {
    private int id_Linea;
    private int longuitud;
    private Double puntoOrigen;
    private Double puntoFin;

    //Atributo de composicion
    private List<Poligono> poligono;

    //Constructores

    public Linea(int id_Linea, int longuitud, Double puntoOrigen, Double puntoFin,
        this.id_Linea = id_Linea;
        this.longuitud = longuitud;
        this.puntoOrigen = puntoOrigen;
        this.puntoFin = puntoFin;
        this.poligono = poligono;
    }

     public Linea(int id_Linea, int longuitud, Double puntoOrigen, Double puntoFin)
        this.id_Linea = id_Linea;
        this.longuitud = longuitud;
        this.puntoOrigen = puntoOrigen;
        this.puntoFin = puntoFin;
    }

     //Metodos Get y Set
    public int getId_Linea() {
        return id_Linea;
```

Prueba2

4) **Creación del paquete Controlador y se instancian los métodos CRUD en cada clase del paquete controlador.**



5) **Creación del paquete Vista.**



6) **Creación de las clases Vista de las diferentes clases del paquete modelo.**

VistaProyecto

```
public Proyecto crear(){
    System.out.println("Ingrese: \n Nombre:");
    s.nextLine();
    String nombre = s.nextLine();
    return controladorProyecto.crear(nombre,vistaJefeProyecto.crear(),vistaPlano.crear());
}
public  Proyecto actualizar(){
    System.out.println("Ingrese: \n Nombre");
    s.nextLine();
    String nombre = s.nextLine();

   return controladorProyecto.actualizar(nombre,vistaJefeProyecto.actualizar());
}
public Proyecto buscar(){
    System.out.println("Ingrese: \n Nombre:");
    s.nextLine();
    String nombre = s.nextLine();
    System.out.println(controladorProyecto.buscar(nombre));
    return controladorProyecto.buscar(nombre);
}
```

Prueba2

VistaPlano

```
public Plano crear(){
    try {
    System.out.println("Fecha (dd/mm/yyyy):");
    String fecha = s.next();
    System.out.println("Ingrese: \n Nombre_Arquitectos:");
    String nombre_Arquitectos = s.next();
    System.out.println("Ingrese: \n Dibujo_Plano:");
    String dibujo_Plano = s.next();
    System.out.println("Ingrese: \n Num_Figuras");
    int num_Figuras = s.nextInt();
    return controladorPlano.crear(formatoFecha.parse(fecha),nombre_Arquitectos,dibujo_Plano,num_Figuras);
    } catch (ParseException ex) {
        System.out.println(ex.getMessage());
    }
    return null;

}
public Plano actualizar(){
     try {
    System.out.println("Fecha (dd/mm/yyyy):");
    String fecha = s.next();
    System.out.println("Ingrese: \n Nombre_Arquitectos:");
    String nombre_Arquitectos = s.next();
    System.out.println("Ingrese: \n Dibujo_Plano:");
    String dibujo_Plano = s.next();
    System.out.println("Ingrese: \n Num_Figuras");
    int num_Figuras = s.nextInt();
     return controladorPlano.crear(formatoFecha.parse(fecha),nombre_Arquitectos,dibujo_Plano,num_Figuras);
    } catch (ParseException ex) {
        System.out.println(ex.getMessage());
    }
    return null;
```

VistaPoligono

```
public Poligono crear(){
    System.out.println("Ingrese: \n Num_Lineas");
    s.nextLine();
    int num_Lineas = s.nextInt();
    System.out.println("Ingrese: \n Nombre:");
    String nombre= s.nextLine();
    System.out.println("Ingrese: \n Color: ");
    String color = s.nextLine();
   return controladorrPoligono.crear(num_Lineas,nombre,color);
}
public Poligono actualizar(){
    System.out.println("Ingrese: \n Num_Lineas");
    s.nextLine();
    int num_Lineas = s.nextInt();
    System.out.println("Ingrese: \n Nombre:");
    String nombre= s.nextLine();
    System.out.println("Ingrese: \n Color: ");
    String color = s.nextLine();
    return controladorrPoligono.crear(num_Lineas,nombre,color);
}
public void buscar(){
    System.out.println("Ingrese: \n Nombre:");
    s.nextLine();
    String nombre = s.nextLine();
    System.out.println(controladorrPoligono.buscar(nombre));
     controladorrPoligono.buscar(nombre);
}
public void eliminar(){
    this.buscar();
    System.out.println("Res: "+ controladorrPoligono.eliminar(controlador
}
```

Prueba2

7) **Creación de la clase Vista General, aquí instanciamos todos los controladores del paquete controlador.**

- En las seis vistas que se crearon, presenta los datos de paquete modelo y lo que vería el usuario.
- Vista General: Se procede a imprimir el código como tal mediante la clase Inicio.

```
public class Inicio {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        VistaGeneral vg = new VistaGeneral();
        vg.menu();

    }
}
```

```
controladorLinea = new ControladorLinea();
controladorPlano = new ControladorPlano();
controladorProyecto = new ControladorProyecto();
controladorJefeProyecto = new ControladorJefeProyecto();

}
public void menu(){
    int op=1;
    while(op >0)
    {
        System.out.println(" 1. Menu Jefe_Proyecto \n 2. Menu Proyecto \n 3. Menu Plano \n 4. Menu Figura \n 5. Menu Poligono \n6. Menu
        op = s.nextInt();

        if(op==1)
            vistaJefeProyecto.menu();
        else if(op==2)
            vistaProyecto.menu();
        else if (op==3)
            vistaPlano.menu();
        else if (op==4)
            vistaFigura.menu();
        else if (op==5)
           vistaPoligono.menu();
        else if (op==6)
            vistaLinea.menu();
    }
```

8) **Salida del presente código.**

- **Se muestra le menú principal**

```
run:
 1. Menu Jefe_Proyecto
 2. Menu Proyecto
 3. Menu Plano
 4. Menu Figura
 5. Menu Poligono
6. Menu Linea
 7. Salir
```

- **Se ingresa al menú proyecto en donde creamos y nos muestra el nombre del proyecto con sus respectivo JefeProyecto y su plano al que pertenece, cada uno de estos cuentan con sus respectivos datos.**

Programación Orientada Objetos

Prueba2

```
Ingrese:
 Nombre:
figuras
Ingrese:
 Nombre:
Ana

Ingrese:
 Direccion:
Cuenca
Ingrese:
 Telefono:
2275609
Fecha (dd/mm/yyyy):
15/08/2021
Ingrese:
 Nombre_Arquitectos:
juan, pedro
Ingrese:
 Dibujo_Plano:
Ingrese:
 Num_Figuras
20
1.crear
2.Actualizar
3.Buscar
```

```
5.listar
5
Proyecto{codigo=1, nombre=figuras, jefeProyecto=Jefe_Proyecto{id=1, nombre=, direccion=Cuenca,
```

```
telefono=2275609}, plano=[Plano{num_Identificacion=1, fecha_Entrega=Fri Jan 15 00:08:00 COT 2021,
```

```
fecha_Entrega=Fri Jan 15 00:08:00 COT 2021, nombre_Arquitectos=juan,, dibujo_Plano=pedro, num_Figuras=20}]}
```

Prueba2

- **Se muestra le menú Línea.**

```
1
Ingrese:
 Longuitud:
40
Ingrese:
 PuntoOrigen
8
Ingrese:
 PuntoFin
9
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
5
Linea{id_Linea=1, longuitud=40, puntoOrigen=8.0, puntoFin=9.0}
1.crear
2.Actualizar
3.Buscar
4.Eliminar
5.listar
```