

PROGRAMACION ORIENTADA A OBJETOS

Nombre:
ALEX ZUMBA

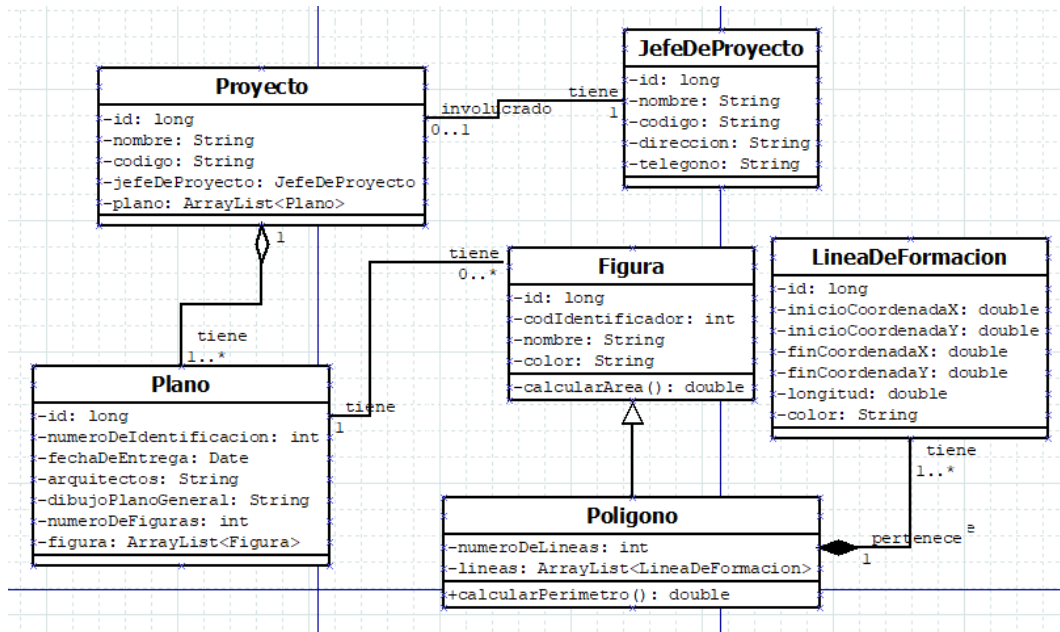
Fecha:
23/06/2021

Tema:
Programa MVC

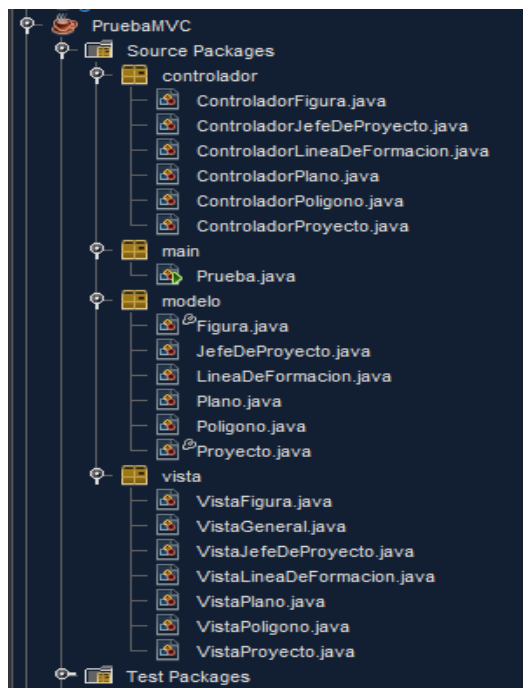
Docente:
ING. DIEGO QUISI

Carrera:
COMPUTACION

1. Diagrama de clases



2. Desarrollo del programa con los datos del diagrama de clases



a) Paquete modelo

Este paquete contiene todas las clases que se utilizarán en el programa, con sus respectivos atributos, constructores, getter y setter, y el método toString

- Clase Proyecto

```
/*
 * 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 modelo;

import java.util.List;

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

    private long id;
    private String nombre;
    private String codigo;
    private JefeDeProyecto jefeDeProyecto;
    private List<Plano> planos;

    public Proyecto() {
    }

    public Proyecto(long id, String nombre, String codigo) {
        this.id = id;
        this.nombre = nombre;
        this.codigo = codigo;
    }

    public Proyecto(long id, String nombre, String codigo,
        JefeDeProyecto jefeDeProyecto) {
        this.id = id;
        this.nombre = nombre;
        this.codigo = codigo;
        this.jefeDeProyecto = jefeDeProyecto;
    }

    public Proyecto(long id, String nombre, String codigo,
        JefeDeProyecto jefeDeProyecto, List<Plano> planos) {
        this.id = id;
        this.nombre = nombre;
        this.codigo = codigo;
        this.jefeDeProyecto = jefeDeProyecto;
        this.planos = planos;
    }

    public long getId() {
        return id;
    }

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

```

    public String getNombre() {
        return nombre;
    }

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

    public String getCodigo() {
        return codigo;
    }

    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }

    public JefeDeProyecto getJefeDeProyecto() {
        return jefeDeProyecto;
    }

    public void setJefeDeProyecto(JefeDeProyecto jefeDeProyecto) {
        this.jefeDeProyecto = jefeDeProyecto;
    }

    public List<Plano> getPlanos() {
        return planos;
    }

    public void setPlanos(List<Plano> planos) {
        this.planos = planos;
    }

    @Override
    public String toString() {
        return "\nProyecto: " + "id: " + id + ", nombre: " + nombre +
", codigo: " + codigo;
    }
}

```

```

*****
*****

```

- Clase Plano

```

/*
 * 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 modelo;

import java.util.Date;
import java.util.List;

/**

```

```

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

    private long id;
    private int numeroDeIdentificacion;
    private Date fechadeEntrega;
    private String arquitectos;
    private String dibujoPlanoGeneral;
    private int numeroDeFiguras;
    private List<Figura> figuras;

    public Plano() {
    }

    public Plano(long id, int numeroDeIdentificacion, Date
fechadeEntrega,
        String arquitectos, String dibujoPlanoGeneral, int
numeroDeFiguras) {
        this.id = id;
        this.numeroDeIdentificacion = numeroDeIdentificacion;
        this.fechadeEntrega = fechadeEntrega;
        this.arquitectos = arquitectos;
        this.dibujoPlanoGeneral = dibujoPlanoGeneral;
        this.numeroDeFiguras = numeroDeFiguras;
    }

    public Plano(long id, int numeroDeIdentificacion, Date
fechadeEntrega,
        String arquitectos, String dibujoPlanoGeneral, int
numeroDeFiguras,
        List<Figura> figuras) {
        this.id = id;
        this.numeroDeIdentificacion = numeroDeIdentificacion;
        this.fechadeEntrega = fechadeEntrega;
        this.arquitectos = arquitectos;
        this.dibujoPlanoGeneral = dibujoPlanoGeneral;
        this.numeroDeFiguras = numeroDeFiguras;
        this.figuras = figuras;
    }

    public long getId() {
        return id;
    }

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

    public int getNumeroDeIdentificacion() {
        return numeroDeIdentificacion;
    }

    public void setNumeroDeIdentificacion(int numeroDeIdentificacion)
{
        this.numeroDeIdentificacion = numeroDeIdentificacion;
    }

    public Date getFechadeEntrega() {
        return fechadeEntrega;
    }

```

```

    }

    public void setFechaDeEntrega(Date fechadeEntrega) {
        this.fechadeEntrega = fechadeEntrega;
    }

    public String getArquitectos() {
        return arquitectos;
    }

    public void setArquitectos(String arquitectos) {
        this.arquitectos = arquitectos;
    }

    public String getDibujoPlanoGeneral() {
        return dibujoPlanoGeneral;
    }

    public void setDibujoPlanoGenera(String dibujoPlanoGenera) {
        this.dibujoPlanoGeneral = dibujoPlanoGenera;
    }

    public int getNumeroDeFiguras() {
        return numeroDeFiguras;
    }

    public void setNumeroDeFiguras(int numeroDeFiguras) {
        this.numeroDeFiguras = numeroDeFiguras;
    }

    public List<Figura> getFiguras() {
        return figuras;
    }

    public void setFiguras(List<Figura> figuras) {
        this.figuras = figuras;
    }

    @Override
    public String toString() {
        return "\nPlano: " + "id: " + id + ",
numeroDeIdentificacion: " + numeroDeIdentificacion
        + ", fechadeEntrga: " + fechadeEntrega + ",
arquitectos: " + arquitectos
        + ", dibujoPlanoGenera: " + dibujoPlanoGeneral + ",
numeroDeFiguras: " + numeroDeFiguras;
    }
}

```

```

*****
*****

```

- Clase JefeDeProyecto

```
/*
 * 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 modelo;
```

```
/**
```

```
 *
```

```
 * @author Alex Zumba
```

```
 */
```

```
public class JefeDeProyecto {
```

```
    private long id;
```

```
    private String nombre;
```

```
    private String codigo;
```

```
    private String direccion;
```

```
    private String telefono;
```

```
    public JefeDeProyecto() {
```

```
    }
```

```
    public JefeDeProyecto(long id, String nombre, String codigo,
        String direccion, String telefono) {
```

```
        this.id = id;
```

```
        this.nombre = nombre;
```

```
        this.codigo = codigo;
```

```
        this.direccion = direccion;
```

```
        this.telefono = telefono;
```

```
    }
```

```
    public long getId() {
```

```
        return id;
```

```
    }
```

```
    public void setId(long id) {
```

```
        this.id = id;
```

```
    }
```

```
    public String getNombre() {
```

```
        return nombre;
```

```
    }
```

```
    public void setNombre(String nombre) {
```

```
        this.nombre = nombre;
```

```
    }
```

```
    public String getCodigo() {
```

```
        return codigo;
```

```
    }
```

```
    public void setCodigo(String codigo) {
```

```
        this.codigo = codigo;
```

```
    }
```

```
    public String getDireccion() {
```

```
        return direccion;
```

```
    }
```

```
    public void setDireccion(String direccion) {
```

```

        this.direccion = direccion;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }

    @Override
    public String toString() {
        return "\nJefeDeProyecto:  " + "id: " + id + ", nombre: " +
nombre + ", codigo: " + codigo
        + ", direccion: " + direccion + ", telefono: " +
telefono;
    }
}

```


- Clase Figura

```

/*
 * 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 modelo;

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

    private long id;
    private int codIdentificador;
    private String nombre;
    private String color;

    public Figura() {
    }

    public Figura(long id, int codIdentificador, String nombre, String
color) {
        this.id = id;
        this.codIdentificador = codIdentificador;
        this.nombre = nombre;
        this.color = color;
    }
}

```



```

    }

    public long getId() {
        return id;
    }

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

    public int getCodIdentificacion() {
        return codIdentificador;
    }

    public void setCodIdentificacion(int codIdentificador) {
        this.codIdentificador = codIdentificador;
    }

    public String getNombre() {
        return nombre;
    }

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

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    @Override
    public String toString() {
        return "\nFigura:  " + "id: " + id + ", codIdentificador: " +
codIdentificador
            + ", nombre: " + nombre + ", color: " + color;
    }
}

```

```

*****
*****

```

- Clase Poligono

Esta clase hereda los atributos de la clase Figura

```

/*
 * 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 modelo;

import java.util.List;

/**
 *
 * @author Alex Zumba
 */
public class Poligono extends Figura {

    private int numeroDeLineas;
    private List<LineaDeFormacion> figuras;

    public Poligono() {
    }

    public Poligono(int numeroDeLineas) {
        this.numeroDeLineas = numeroDeLineas;
    }

    public Poligono(long id, int codIdentificador, String nombre,
String color) {
        super(id, codIdentificador, nombre, color);
    }

    public Poligono(long id, int codIdentificador, String nombre,
String color,
        int numeroDeLineas) {
        super(id, codIdentificador, nombre, color);
        this.numeroDeLineas = numeroDeLineas;
    }

    public Poligono(List<LineaDeFormacion> figuras, long id, int
codIdentificador,
        String nombre, String color, int numeroDeLineas) {
        super(id, codIdentificador, nombre, color);
        this.numeroDeLineas = numeroDeLineas;
        this.figuras = figuras;
    }

    public int getNumeroDeLineas() {
        return numeroDeLineas;
    }

    public void setNumeroDeLineas(int numeroDeLineas) {
        this.numeroDeLineas = numeroDeLineas;
    }

    public List<LineaDeFormacion> getFiguras() {
        return figuras;
    }

    public void setFiguras(List<LineaDeFormacion> figuras) {
        this.figuras = figuras;
    }

    @Override
    public String toString() {
        return "\nPoligono: " + "numeroDeLineas: " + numeroDeLineas
```

```
        + super.toString();  
    }  
}
```

```
*****  
*****
```

- Clase LineaDeFormacion

```
/*  
 * 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 modelo;  
  
/**  
 *  
 * @author Alex Zumba  
 */  
public class LineaDeFormacion {  
  
    private long id;  
    private double iniciocoordenadaX;  
    private double iniciocoordenadaY;  
    private double finCoordenadaX;  
    private double finCoordenadaY;  
    private double longitud;  
    private String color;  
  
    public LineaDeFormacion() {  
    }  
  
    public LineaDeFormacion(long id, double iniciocoordenadaX, double  
iniciocoordenadaY,  
        double finCoordenadaX, double finCoordenadaY, double  
longitud, String color) {  
        this.id = id;  
        this.iniciocoordenadaX = iniciocoordenadaX;  
        this.iniciocoordenadaY = iniciocoordenadaY;  
        this.finCoordenadaX = finCoordenadaX;  
        this.finCoordenadaY = finCoordenadaY;  
        this.longitud = longitud;  
        this.color = color;  
    }  
  
    public long getId() {  
        return id;  
    }  
  
    public void setId(long id) {  
        this.id = id;  
    }  
}
```

```

    }

    public double getIniciocoordenadaX() {
        return iniciocoordenadaX;
    }

    public void setIniciocoordenadaX(double iniciocoordenadaX) {
        this.iniciocoordenadaX = iniciocoordenadaX;
    }

    public double getIniciocoordenadaY() {
        return iniciocoordenadaY;
    }

    public void setIniciocoordenadaY(double iniciocoordenadaY) {
        this.iniciocoordenadaY = iniciocoordenadaY;
    }

    public double getFinCoordenadaX() {
        return finCoordenadaX;
    }

    public void setFinCoordenadaX(double finCoordenadaX) {
        this.finCoordenadaX = finCoordenadaX;
    }

    public double getFinCoordenadaY() {
        return finCoordenadaY;
    }

    public void setFinCoordenadaY(double finCoordenadaY) {
        this.finCoordenadaY = finCoordenadaY;
    }

    public double getLongitud() {
        return longitud;
    }

    public void setLongitud(double longitud) {
        this.longitud = longitud;
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    @Override
    public String toString() {
        return "\nLineaDeFormacion:  " + "id: " + id + ",
iniciocoordenadaX: " + iniciocoordenadaX
        + ", iniciocoordenadaY: " + iniciocoordenadaY + ",
finCoordenadaX: " + finCoordenadaX
        + ", finCoordenadaY: " + finCoordenadaY + ", longitud:
" + longitud + ", color: " + color;
    }

```

```
}
```

```
*****  
*****
```

b) Paquete controlador

En este paquete creamos los métodos CRUD, que procesan la información obtenida del modelo y lo envía a la vista para ser mostrado en pantalla

- Clase ControladorProyecto

```
/*  
 * 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 java.util.ArrayList;  
import java.util.List;  
import modelo.Proyecto;  
  
/**  
 *  
 * @author Alex Zumba  
 */  
public class ControladorProyecto {  
  
    private List<Proyecto> listaProyecto;  
    private Proyecto seleccionado;  
  
    public ControladorProyecto() {  
        listaProyecto = new ArrayList<>();  
        seleccionado = null;  
    }  
  
    public long generarId() {  
        return (listaProyecto.size() > 0) ?  
listaProyecto.get(listaProyecto.size() - 1).getId() + 1 : 1;  
    }  
  
    public boolean crear(String nombre, String codigo) {  
        Proyecto proyecto = new Proyecto(this.generarId(), nombre,  
codigo);  
        return listaProyecto.add(proyecto);  
    }  
}
```

```
    }

    public Proyecto buscar(String codigo) {
        for (Proyecto proyecto : listaProyecto) {
            if (proyecto.getCodigo().equals(codigo)) {
                return proyecto;
            }
        }
        return null;
    }

    public boolean actualizar(String nombre, String codigo, String
codigoNuevo) {
        Proyecto proyecto = this.buscar(codigo);
        if (proyecto != null) {
            int posicion = listaProyecto.indexOf(proyecto);
            proyecto.setNombre(nombre);
            proyecto.setCodigo(codigoNuevo);
            listaProyecto.set(posicion, proyecto);
            return true;
        }
        return false;
    }

    public boolean eliminar(String codigo) {
        Proyecto proyecto = this.buscar(codigo);
        if (proyecto != null) {
            return listaProyecto.remove(proyecto);
        }
        return false;
    }

    public List<Proyecto> getListaProyecto() {
        return listaProyecto;
    }

    public void setListaProyecto(List<Proyecto> listaProyecto) {
        this.listaProyecto = listaProyecto;
    }

    public Proyecto getSeleccionado() {
        return seleccionado;
    }

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

```
*****
*****
```

- Clase ControladorPlano

```

/*
 * 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 java.util.ArrayList;
import java.util.Date;
import java.util.List;
import modelo.Plano;

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

    private List<Plano> listaPlanos;
    private Plano seleccionado;

    public ControladorPlano() {
        listaPlanos = new ArrayList<>();
        seleccionado = null;
    }

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

    public boolean crear(int numeroDeIdentificacion, Date
fechadeEntrega, String arquitectos,
        String dibujoPlanoGeneral, int numeroDeFiguras) {
        Plano plano = new Plano(this.generarId(),
numeroDeIdentificacion, fechadeEntrega, arquitectos,
        dibujoPlanoGeneral, numeroDeFiguras);
        return listaPlanos.add(plano);
    }

    public Plano buscar(String dibujoPlanoGenera) {
        for (Plano plano : listaPlanos) {
            if
(plano.getDibujoPlanoGeneral().equals(dibujoPlanoGenera)) {
                return plano;
            }
        }
        return null;
    }

    public boolean actualizar(int numeroDeIdentificacion, Date
fechadeEntrega, String arquitectos,
        String dibujoPlanoGeneral, String NuevoDibujoPlano, int
numeroDeFiguras) {
        Plano plano = this.buscar(dibujoPlanoGeneral);
        if (plano != null) {
            int posicion = listaPlanos.indexOf(plano);

```

```

        plano.setNumeroDeIdentificacion(numeroDeIdentificacion);
        plano.setFechaDeEntrega(fechaDeEntrega);
        plano.setArquitectos(arquitectos);
        plano.setDibujoPlanoGenera(NuevoDibujoPlano);
        plano.setNumeroDeFiguras(numeroDeFiguras);
        listaPlanos.set(posicion, plano);

        return true;
    }
    return false;
}

public boolean eliminar(String dibujoPlanoGeneral) {
    Plano plano = this.buscar(dibujoPlanoGeneral);
    if (plano != null) {
        return listaPlanos.remove(plano);
    }
    return false;
}

public List<Plano> getListaPlanos() {
    return listaPlanos;
}

public void setListaPlanos(List<Plano> listaPlanos) {
    this.listaPlanos = listaPlanos;
}

public Plano getSeleccionado() {
    return seleccionado;
}

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

```

```

*****
*****

```

- Clase ControladorJefeDeProyecto

```

/*
 * 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 java.util.ArrayList;
import java.util.List;
import modelo.JefeDeProyecto;

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

    private List<JefeDeProyecto> listaJefeDeProyecto;
    private JefeDeProyecto seleccionado;

    public ControladorJefeDeProyecto() {
        listaJefeDeProyecto = new ArrayList<>();
        seleccionado = null;
    }

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

    public boolean crear(String nombre, String codigo, String
direccion, String telefono) {
        JefeDeProyecto jefeProyecto = new
JefeDeProyecto(this.generarId(), nombre, codigo, direccion, telefono);
        return listaJefeDeProyecto.add(jefeProyecto);
    }

    public JefeDeProyecto buscar(String codigo) {
        for (JefeDeProyecto jefeProyecto : listaJefeDeProyecto) {
            if (jefeProyecto.getCodigo().equals(codigo)) {
                return jefeProyecto;
            }
        }
        return null;
    }

    public boolean actualizar(String nombre, String codigo, String
codigoNuevo, String direccion, String telefono) {
        JefeDeProyecto jefeProyecto = this.buscar(codigo);

        if (jefeProyecto != null) {
            int posicion = listaJefeDeProyecto.indexOf(jefeProyecto);
            jefeProyecto.setNombre(nombre);
            jefeProyecto.setCodigo(codigoNuevo);
            jefeProyecto.setDireccion(direccion);
            jefeProyecto.setTelefono(telefono);
            listaJefeDeProyecto.set(posicion, jefeProyecto);

            return true;
        }
        return false;
    }

    public boolean eliminar(String codigo) {
        JefeDeProyecto jefeProyecto = this.buscar(codigo);
        if (jefeProyecto != null) {
```

```

        return listaJefeDeProyecto.remove(jefeProyecto);
    }
    return false;
}

public List<JefeDeProyecto> getListaJefeDeProyecto() {
    return listaJefeDeProyecto;
}

public void setListaJefeDeProyecto(List<JefeDeProyecto>
listaJefeDeProyecto) {
    this.listaJefeDeProyecto = listaJefeDeProyecto;
}

public JefeDeProyecto getSeleccionado() {
    return seleccionado;
}

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

```

```

*****
*****

```

- Clase ControladorFigura

```

/*
 * 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 java.util.ArrayList;
import java.util.List;
import modelo.Figura;

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

    private List<Figura> listaFiguras;
    private Figura seleccionado;

    public ControladorFigura() {
        listaFiguras = new ArrayList<>();
        seleccionado = null;
    }
}

```

```
    }

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

    public boolean crear(int codIdentificador, String nombre, String
color) {
        Figura figura = new Figura(this.generarId(), codIdentificador,
nombre, color);
        return listaFiguras.add(figura);
    }

    public Figura buscar(String color) {
        for (Figura figura : listaFiguras) {
            if (figura.getColor().equals(color)) {
                return figura;
            }
        }
        return null;
    }

    public boolean actualizar(int codIdentificador, String nombre,
String color, String colorNuevo) {
        Figura figura = this.buscar(color);

        if (figura != null) {
            int posicion = listaFiguras.indexOf(figura);
            figura.setCodIdentificacion(codIdentificador);
            figura.setNombre(nombre);
            figura.setColor(colorNuevo);

            return true;
        }
        return false;
    }

    public boolean eliminar(String color) {
        Figura figura = this.buscar(color);

        if (figura != null) {
            return listaFiguras.remove(figura);
        }
        return false;
    }

    public List<Figura> getListaFiguras() {
        return listaFiguras;
    }

    public void setListaFiguras(List<Figura> listaFiguras) {
        this.listaFiguras = listaFiguras;
    }

    public Figura getSeleccionado() {
        return seleccionado;
    }

    public void setSeleccionado(Figura seleccionado) {
```

```
        this.seleccionado = seleccionado;
    }

}
```

```
*****
*****
```

- Clase ControladorPoligono

```
/*
 * 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 java.util.ArrayList;
import java.util.List;
import modelo.Poligono;

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

    private List<Poligono> listaPoligonos;
    private Poligono seleccionado;

    public ControladorPoligono() {
        listaPoligonos = new ArrayList<>();
        seleccionado = null;
    }

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

    public boolean crear(int codIdentificador, String nombre, String
color, int numeroDeLineas) {
        Poligono poligono = new Poligono(numeroDeLineas,
codIdentificador, nombre, color, numeroDeLineas);
        return listaPoligonos.add(poligono);
    }

    public Poligono buscar(String color) {
        for (Poligono poligono : listaPoligonos) {
            if (poligono.getColor().equals(color)) {
                return poligono;
            }
        }
    }
}
```

```
        }
    }
    return null;
}

public boolean actualizar(int codIdentificador, String nombre,
String color,
    String colorNuevo, int numeroDeLineas) {
    Poligono poligono = this.buscar(color);

    if (poligono != null) {
        int posicion = listaPoligonos.indexOf(poligono);

        poligono.setCodIdentificacion(codIdentificador);
        poligono.setNombre(nombre);
        poligono.setColor(colorNuevo);
        poligono.setNumeroDeLineas(numeroDeLineas);

        return true;
    }
    return false;
}

public boolean eliminar(String color) {
    Poligono poligono = this.buscar(color);

    if (poligono != null) {
        return listaPoligonos.remove(poligono);
    }
    return false;
}

public List<Poligono> getListaPoligonos() {
    return listaPoligonos;
}

public void setListaPoligonos(List<Poligono> listaPoligonos) {
    this.listaPoligonos = listaPoligonos;
}

public Poligono getSeleccionado() {
    return seleccionado;
}

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

*****
*****
```

- Clase ControladorLineaDeFormacion

```

/*
 * 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 java.util.ArrayList;
import java.util.List;
import modelo.LineaDeFormacion;

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

    private List<LineaDeFormacion> listaLineaDeFormacion;
    private LineaDeFormacion seleccionado;

    public ControladorLineaDeFormacion () {
        listaLineaDeFormacion = new ArrayList<>();
        seleccionado = null;
    }

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

    public boolean crear(double iniciocoordenadaX, double
iniciocoordenadaY,
        double finCoordenadaX, double finCoordenadaY, double
longitud, String color) {
        LineaDeFormacion linea = new
LineaDeFormacion(this.generarId(), iniciocoordenadaX,
iniciocoordenadaY, finCoordenadaX,
        finCoordenadaY, longitud, color);
        return listaLineaDeFormacion.add(linea);
    }

    public LineaDeFormacion buscar(String color) {
        for (LineaDeFormacion linea : listaLineaDeFormacion) {
            if (linea.getColor().equals(color)) {
                return linea;
            }
        }
        return null;
    }

    public boolean actualizar(double iniciocoordenadaX, double
iniciocoordenadaY,
        double finCoordenadaX, double finCoordenadaY, double
longitud,
        String color, String colorNuevo) {
        LineaDeFormacion linea = this.buscar(color);
    }

```

```

        if (linea != null) {
            int posicion = listaLineaDeFormacion.indexOf(linea);
            linea.setIniciocoordenadaX(iniciocoordenadaX);
            linea.setIniciocoordenadaY(iniciocoordenadaY);
            linea.setFinCoordenadaX(finCoordenadaX);
            linea.setFinCoordenadaY(finCoordenadaY);
            linea.setColor(colorNuevo);

            return true;
        }
        return false;
    }

    public boolean eliminar(String color) {
        LineaDeFormacion linea = this.buscar(color);

        if (linea != null) {
            return listaLineaDeFormacion.remove(linea);
        }
        return false;
    }

    public List<LineaDeFormacion> getListaLineaDeFormacion() {
        return listaLineaDeFormacion;
    }

    public void setListaLineaDeFormacion(List<LineaDeFormacion>
listaLineaDeFormacion) {
        this.listaLineaDeFormacion = listaLineaDeFormacion;
    }

    public LineaDeFormacion getSeleccionado() {
        return seleccionado;
    }

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

```

c) Paquete vista

En este paquete se crea la forma en que será mostrada la información recibida por el controlador en la pantalla, para el usuario.

- Clase VistaProyecto

```
/*
 * 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.ControladorProyecto;
import java.util.Scanner;
import modelo.Proyecto;

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

    private ControladorProyecto controladorProyecto;
    private Scanner entrada;

    public VistaProyecto() {
        controladorProyecto = new ControladorProyecto();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("\n+++++Gestion de Proyectos+++++");
            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 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 codigo: ");
    String codigo = entrada.next();
    System.out.println("Resultado: " +
controladorProyecto.crear(nombre, codigo));
}

public Proyecto buscar() {
    System.out.print("Ingrese el codigo: ");
    String codigo = entrada.next();
    Proyecto proyecto = controladorProyecto.buscar(codigo);
    controladorProyecto.setSeleccionado(proyecto);
    System.out.println(proyecto);
    return proyecto;
}

public void actualizar() {
    Proyecto proyecto = buscar();

    System.out.print("Ingrese el nombre nuevo: ");
    String nombre = entrada.next();
    System.out.print("Ingrese el codigo nuevo: ");
    String codigo = entrada.next();
    System.out.println("Resultado: " +
controladorProyecto.actualizar(nombre, proyecto.getCodigo(), codigo));
}

public void eliminar() {
    Proyecto proyecto = buscar();

    if (proyecto != null) {

        System.out.println("Resultado: " +
controladorProyecto.eliminar(proyecto.getCodigo()));
    }

}

public void listar() {
    for (Proyecto proyecto :
controladorProyecto.getListaProyecto()) {
        System.out.println(proyecto);
    }
}

public ControladorProyecto getControladorProyecto() {
    return controladorProyecto;
}

public void setControladorProyecto(ControladorProyecto
controladorProyecto) {
    this.controladorProyecto = controladorProyecto;
}
}
```


- Clase VistaPlano

```
/*
 * 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.ControladorPlano;
import java.text.DateFormat;
import java.text.ParseException;
import java.util.Scanner;
import modelo.Plano;

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

    private ControladorPlano controladorPlano;
    private Scanner entrada;
    private DateFormat formatoFecha;

    public VistaPlano() {
        controladorPlano = new ControladorPlano();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("\n+++++Gestion de Planos+++++");
            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 1: crear();
                    break;
                case 2: buscar();
                    break;
                case 3: actualizar();
                    break;
                case 4: eliminar();
            }
        }
    }

    private void crear() {
        // TODO Auto-generated method stub
    }

    private void buscar() {
        // TODO Auto-generated method stub
    }

    private void actualizar() {
        // TODO Auto-generated method stub
    }

    private void eliminar() {
        // TODO Auto-generated method stub
    }
}
```

```
        break;
    case 5: listar();
        break;
    case 6: System.out.println("-----Finalizado-----");
        break;
    }
} while (opcion < 6);
}

public void crear() {
    System.out.print("Ingrese el numero de identificacion: ");
    int numeroDeIdentificacion = entrada.nextInt();
    System.out.print("Ingrese la fecha de entrega (dd/mm/yy): ");
    String fechadeEntrega = entrada.next();
    System.out.print("Ingrese el arquitecto: ");
    String arquitectos = entrada.next();
    System.out.print("Ingrese el dibujo del plano: ");
    String dibujoPlanoGeneral = entrada.next();
    System.out.print("Ingrese el numero de figuras: ");
    int numeroDeFiguras = entrada.nextInt();

    try {
        System.out.println("Resultado: " +
            controladorPlano.crear(numeroDeIdentificacion,
                formatoFecha.parse(fechadeEntrega), arquitectos,
                dibujoPlanoGeneral, numeroDeFiguras));
    } catch (ParseException ex) {
        System.out.println(ex.getMessage());
    }
}

public Plano buscar() {
    System.out.print("Ingrese el dibujo del plano: ");
    String dibujo = entrada.next();
    Plano plano = controladorPlano.buscar(dibujo);
    controladorPlano.setSeleccionado(plano);
    System.out.println(plano);
    return plano;
}

public void actualizar() {
    Plano plano = buscar();

    System.out.print("Ingrese el nuevo numero de identificacion:
");
    int numero = entrada.nextInt();
    System.out.print("Ingrese la nueva fecha de entrega
(dd/mm/yy): ");
    String fecha = entrada.next();
    System.out.print("Ingrese el nuevo arquitecto: ");
    String arquitecto = entrada.next();
    System.out.print("Ingrese el nuevo dibujo del plano: ");
    String dibujo = entrada.next();
    System.out.print("Ingrese el nuevo numero de figuras: ");
    int figuras = entrada.nextInt();

    try {
        System.out.println("Resultado: " +
            controladorPlano.actualizar(numero, formatoFecha.parse(fecha),
                fecha, arquitecto, plano.getDibujoPlanoGeneral(),
                figuras));
    }
```

```

    } catch (ParseException ex) {
        System.out.println(ex.getMessage());
    }

}

public void eliminar() {
    Plano plano = buscar();

    if (plano != null) {

        System.out.println("Resultado: " +
controladorPlano.eliminar(plano.getDibujoPlanoGeneral()));
    }

}

public void listar() {
    for (Plano plano : controladorPlano.getListaPlanos()) {
        System.out.println(plano);
    }
}

public ControladorPlano getControladorPlano() {
    return controladorPlano;
}

public void setControladorPlano(ControladorPlano controladorPlano)
{
    this.controladorPlano = controladorPlano;
}
}

```

```

*****
*****

```

- Clase VistaJefeDeProyecto

```

/*
 * 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.ControladorJefeDeProyecto;
import java.util.Scanner;
import modelo.JefeDeProyecto;

```

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

    private ControladorJefeDeProyecto controladorJefeDeProyecto;
    private Scanner entrada;

    public VistaJefeDeProyecto() {
        controladorJefeDeProyecto = new ControladorJefeDeProyecto();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("\n+++++Jefes de Proyectos+++++");
            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 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 codigo: ");
        String codigo = entrada.next();
        System.out.print("Ingrese la direccion: ");
        String direccion = entrada.next();
        System.out.print("Ingrese el telefono: ");
        String telefono = entrada.next();
        System.out.println("Resultado: " +
            controladorJefeDeProyecto.crear(nombre, codigo, direccion, telefono));
    }

    public JefeDeProyecto buscar() {
        System.out.print("Ingrese el codigo: ");
        String codigo = entrada.next();
    }
}
```

```

        JefeDeProyecto proyecto =
controladorJefeDeProyecto.buscar(codigo);
        controladorJefeDeProyecto.setSeleccionado(proyecto);
        System.out.println(proyecto);
        return proyecto;
    }

    public void actualizar() {
        JefeDeProyecto proyecto = buscar();

        System.out.print("Ingrese el nombre nuevo: ");
        String nombre = entrada.next();
        System.out.print("Ingrese el codigo nuevo: ");
        String codigo = entrada.next();
        System.out.print("Ingrese la direccion nuevo: ");
        String direccion = entrada.next();
        System.out.print("Ingrese el telefono nuevo: ");
        String telefono = entrada.next();
        System.out.println("Resultado: " +
            controladorJefeDeProyecto.actualizar(nombre,
proyecto.getCodigo(), codigo, direccion, telefono));
    }

    public void eliminar() {
        JefeDeProyecto proyecto = buscar();

        if (proyecto != null) {

            System.out.println("Resultado: " +
controladorJefeDeProyecto.eliminar(proyecto.getCodigo()));
        }

    }

    public void listar() {
        for (JefeDeProyecto proyecto :
controladorJefeDeProyecto.getListajefeDeProyecto()) {
            System.out.println(proyecto);
        }
    }

    public ControladorJefeDeProyecto getControladorJefeDeProyecto() {
        return controladorJefeDeProyecto;
    }

    public void setControladorJefeDeProyecto(ControladorJefeDeProyecto
controladorJefeDeProyecto) {
        this.controladorJefeDeProyecto = controladorJefeDeProyecto;
    }
}

```

```

*****
*****

```

- **Clase VistaFigura**

```

/*
 * 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.ControladorFigura;
import java.util.Scanner;
import modelo.Figura;

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

    private ControladorFigura controladorFigura;
    private Scanner entrada;

    public VistaFigura() {
        controladorFigura = new ControladorFigura();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("\n+++++Gestion de Figuras+++++");
            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 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 codigo: ");
        int codigo = entrada.nextInt();
        System.out.print("Ingrese el nombre: ");
        String nombre = entrada.next();
        System.out.print("Ingrese la color: ");
        String color = entrada.next();

        System.out.println("Resultado: " +
            controladorFigura.crear(codigo, nombre, color));
    }

    public Figura buscar() {
        System.out.print("Ingrese el color: ");
        String color = entrada.next();
        Figura figura = controladorFigura.buscar(color);
        controladorFigura.setSeleccionado(figura);
        System.out.println(figura);
        return figura;
    }

    public void actualizar() {
        Figura figura = buscar();

        System.out.print("Ingrese el codigo nuevo: ");
        int codigo = entrada.nextInt();
        System.out.print("Ingrese el nombre nuevo: ");
        String nombre = entrada.next();
        System.out.print("Ingrese la color nuevo: ");
        String colorNuevo = entrada.next();

        System.out.println("Resultado: " +
            controladorFigura.actualizar(codigo, nombre, figura.getColor(),
            colorNuevo));
    }

    public void eliminar() {
        Figura figura = buscar();

        if (figura != null) {
            System.out.println("Resultado: " +
                controladorFigura.eliminar(figura.getColor()));
        }
    }

    public void listar() {
        for (Figura figura : controladorFigura.getListaFiguras()) {
            System.out.println(figura);
        }
    }

    public ControladorFigura getControladorFigura() {
        return controladorFigura;
    }

```



```
public void setControladorFigura(ControladorFigura
controladorFigura) {
    this.controladorFigura = controladorFigura;
}

}
```

```
*****
*****
```

- Clase VistaPoligono

```
/*
 * 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.ControladorPoligono;
import java.util.Scanner;
import modelo.Poligono;

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

    private ControladorPoligono controladorPoligono;
    private Scanner entrada;

    public VistaPoligono() {
        controladorPoligono = new ControladorPoligono();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("\n+++++Gestion de Poligonos+++++");
            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();
        } while (opcion != 6);
    }
}
```

```

        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 codigo: ");
    int codigo = entrada.nextInt();
    System.out.print("Ingrese el nombre: ");
    String nombre = entrada.next();
    System.out.print("Ingrese la color: ");
    String color = entrada.next();
    System.out.print("Ingrese el numero de lineas: ");
    int numLineas = entrada.nextInt();

    System.out.println("Resultado: " +
        controladorPoligono.crear(codigo, nombre, color, numLineas));
}

public Poligono buscar() {
    System.out.print("Ingrese el color: ");
    String color = entrada.next();
    Poligono poligono = controladorPoligono.buscar(color);
    controladorPoligono.setSeleccionado(poligono);
    System.out.println(poligono);
    return poligono;
}

public void actualizar() {
    Poligono poligono = buscar();

    System.out.print("Ingrese el codigo nuevo: ");
    int codigo = entrada.nextInt();
    System.out.print("Ingrese el nombre nuevo: ");
    String nombre = entrada.next();
    System.out.print("Ingrese la color nuevo: ");
    String colorNuevo = entrada.next();
    System.out.print("Ingrese el nuevo numero de lineas: ");
    int numLineas = entrada.nextInt();

    System.out.println("Resultado: " +
        controladorPoligono.actualizar(codigo, nombre,
        poligono.getColor(), colorNuevo, numLineas));
}

public void eliminar() {
    Poligono poligono = buscar();

    if (poligono != null) {

```

```

        System.out.println("Resultado: " +
        controladorPoligono.eliminar(poligono.getColor()));
    }

    }

    public void listar() {
        for (Poligono poligono :
        controladorPoligono.getListaPoligonos()) {
            System.out.println(poligono);
        }
    }

    public ControladorPoligono getControladorPoligono() {
        return controladorPoligono;
    }

    public void setControladorPoligono(ControladorPoligono
    controladorPoligono) {
        this.controladorPoligono = controladorPoligono;
    }
}

```

```

*****
*****

```

- Clase VistaLineaDeFormacion

```

/*
 * 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.ControladorLineaDeFormacion;
import java.util.Scanner;
import modelo.LineaDeFormacion;

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

    private ControladorLineaDeFormacion controladorLineaDeFormacion;
    private Scanner entrada;

    public VistaLineaDeFormacion() {

```

```

        controladorLineaDeFormacion = new
ControladorLineaDeFormacion();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("\n+++++Lineas de Formacion+++++");
            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 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 inicio de coordenada X: ");
        double inicioX = entrada.nextDouble();
        System.out.print("Ingrese inicio de coordenada Y: ");
        double inicioY = entrada.nextDouble();
        System.out.print("Ingrese fin de coordenada X: ");
        double finX = entrada.nextDouble();
        System.out.print("Ingrese fin de coordenada Y: ");
        double finY = entrada.nextDouble();
        System.out.print("Ingrese la longitud: ");
        double longitud = entrada.nextDouble();
        System.out.print("Ingrese el color: ");
        String color = entrada.next();

        System.out.println("Resultado: " +
            controladorLineaDeFormacion.crear(inicioX, inicioY, finX,
            finY, longitud, color));
    }

    public LineaDeFormacion buscar() {
        System.out.print("Ingrese el color: ");
        String color = entrada.next();
        LineaDeFormacion lineaF =
controladorLineaDeFormacion.buscar(color);
        controladorLineaDeFormacion.setSeleccionado(lineaF);
        System.out.println(lineaF);
    }

```

```

        return lineaF;
    }

    public void actualizar() {
        LineaDeFormacion lineaF = buscar();

        System.out.print("Ingrese el nuevo inicio de coordenada X: ");
        double inicioX = entrada.nextDouble();
        System.out.print("Ingrese el nuevo inicio de coordenada Y: ");
        double inicioY = entrada.nextDouble();
        System.out.print("Ingrese el nuevo fin de coordenada X: ");
        double finX = entrada.nextDouble();
        System.out.print("Ingrese el nuevo fin de coordenada Y: ");
        double finY = entrada.nextDouble();
        System.out.print("Ingrese la longitud nueva: ");
        double longitud = entrada.nextDouble();
        System.out.print("Ingrese el color nuevo: ");
        String color = entrada.next();

        System.out.println("Resultado: " +
            controladorLineaDeFormacion.actualizar(inicioX, inicioY, finX,
            finY, longitud, lineaF.getColor(), color));
    }

    public void eliminar() {
        LineaDeFormacion lineaF = buscar();

        if (lineaF != null) {
            System.out.println("Resultado: " +
                controladorLineaDeFormacion.eliminar(lineaF.getColor()));
        }
    }

    public void listar() {
        for (LineaDeFormacion lineaF :
            controladorLineaDeFormacion.getListLineaDeFormacion()) {
            System.out.println(lineaF);
        }
    }

    public ControladorLineaDeFormacion
    getControladorLineaDeFormacion() {
        return controladorLineaDeFormacion;
    }

    public void
    setControladorLineaDeFormacion(ControladorLineaDeFormacion
    controladorLineaDeFormacion) {
        this.controladorLineaDeFormacion =
        controladorLineaDeFormacion;
    }
}

```


- Clase VistaGeneral

```
/*
 * 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 java.util.Scanner;

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

    private Scanner entrada;
    private VistaProyecto vistaProyecto;
    private VistaPlano vistaPlano;
    private VistaJefeDeProyecto vistaJefeDeProyecto;
    private VistaFigura vistaFigura;
    private VistaPoligono vistaPoligono;
    private VistaLineaDeFormacion vistaLineaDeFormacion;

    public VistaGeneral() {
        vistaProyecto = new VistaProyecto();
        vistaPlano = new VistaPlano();
        vistaJefeDeProyecto = new VistaJefeDeProyecto();
        vistaFigura = new VistaFigura();
        vistaPoligono = new VistaPoligono();
        vistaLineaDeFormacion = new VistaLineaDeFormacion();
        entrada = new Scanner(System.in);
    }

    public void menu() {
        int opcion = 0;
        do {
            System.out.println("\n***** MENU PRINCIPAL *****");
            System.out.println("1.- Proyectos");
            System.out.println("2.- Planos");
            System.out.println("3.- Jefe-Proyectos");
            System.out.println("4.- Figura");
            System.out.println("5.- Poligono");
            System.out.println("6.- linea-Formacion");
            System.out.println("7.- Salir");

            opcion = entrada.nextInt();

            switch (opcion) {
```

```

        case 1: vistaProyecto.menu();
            break;
        case 2: vistaPlano.menu();
            break;
        case 3: vistaJefeDeProyecto.menu();
            break;
        case 4: vistaFigura.menu();
            break;
        case 5: vistaPoligono.menu();
            break;
        case 6: vistaLineaDeFormacion.menu();
        case 7:
            System.out.println("\n*****PROGRAMA
FINALIZADO*****");
            System.out.println("");
            break;
    }
} while (opcion < 7);
}
}

```


d) Paquete main

En este paquete se encuentra la clase principal que contiene el método main, que sirve para imprimir los datos que se crearon.

```

/*
 * 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 main;

import vista.VistaGeneral;

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

```

```
public static void main(String[] args) {

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

}

}
```

```
*****
*****
```

3. Ejecucion del programa

- Proyecto

Crear

```
+++++Gestion de Proyectos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el nombre: Nimbo
Ingrese el codigo: 10N
Resultado: true

+++++Gestion de Proyectos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
```

```
1
Ingrese el nombre: RobbotT
Ingrese el codigo: 12r
Resultado: true

+++++Gestion de Proyectos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el nombre: Meyer
Ingrese el codigo: 10M
Resultado: true
```


Buscar

```
+++++Gestion de Proyectos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el codigo: 10N

Proyecto: id: 1, nombre: Nimbo, codigo: 10N

+++++Gestion de Proyectos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el codigo: 10S

Proyecto: id: 4, nombre: Serpent, codigo: 10S
```

Actualizar

```
+++++Gestion de Proyectos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
3
Ingrese el codigo: 12r

Proyecto: id: 2, nombre: RobbotI, codigo: 12r
Ingrese el nombre nuevo: ROB-BOT
Ingrese el codigo nuevo: 10R
Resultado: true
```

Eliminar

```
+++++Gestion de Proyectos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
4
Ingrese el codigo: 10M

Proyecto: id: 3, nombre: Meyer, codigo: 10M
Resultado: true
```

Listar

```
+++++Gestion de Proyectos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
5

Proyecto: id: 1, nombre: Nimbo, codigo: 10N

Proyecto: id: 2, nombre: ROB-BOT, codigo: 10R

Proyecto: id: 4, nombre: Serpent, codigo: 10S

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

- Plano

Crear

```

+++++Gestion de Planos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el numero de identificacion: 12
Ingrese la fecha de entrega (dd/mm/yy): 10/06/2021
Ingrese el arquitecto: Pablo-Martinez
Ingrese el dibujo del plano: boceto-casa
Ingrese el numero de figuras: 21
Resultado: true

+++++Gestion de Planos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el numero de identificacion: 10
Ingrese la fecha de entrega (dd/mm/yy): 15/06/2021
Ingrese el arquitecto: Lilian-Godoy
Ingrese el dibujo del plano: boceto-edificio
Ingrese el numero de figuras: 36
Resultado: true
    
```

Buscar

```

1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el dibujo del plano: boceto-edificio

Plano: id: 2, numeroDeIdentificacion: 10, fechadeEntrega: Fri Jan 15 00:06:00 COT 2021, arquitectos: Lilian-Godoy, dibujoPlanoGenera: boceto-edificio, numeroDeFiguras: 36

+++++Gestion de Planos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
    
```

Actualizar

```

+++++Gestion de Planos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
3
Ingrese el dibujo del plano: boceto-restaurant

Plano: id: 3, numeroDeIdentificacion: 165555, fechadeEntrga: Thu Jan 02 00:02:00 COT 2020, arquitectos: Hector-MuDos, dibujoPlanoGenera: boceto-restaurant, numeroDeFiguras: 14
Ingrese el nuevo numero de identificacion: 16
Ingrese la nueva fecha de entrega (dd/mm/yy): 24/06/2021
Ingrese el nuevo arquitecto: Dorian-Baculima
Ingrese el nuevo dibujo del plano: boceto-puente
Ingrese el nuevo numero de figuras: 25
Resultado: true
    
```

Eliminar

```

+++++Gestion de Planos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
4
Ingrese el dibujo del plano: boceto-puente

Plano: id: 3, numeroDeIdentificacion: 16, fechadeEntrga: Sun Jan 24 00:06:00 COT 2021, arquitectos: Dorian-Baculima, dibujoPlanoGenera: boceto-puente, numeroDeFiguras: 25
Resultado: true

+++++Gestion de Planos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
    
```

Listar

```

+++++Gestion de Planos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
5

Plano: id: 1, numeroDeIdentificacion: 12, fechadeEntrga: Sun Jan 10 00:06:00 COT 2021, arquitectos: Pablo-Martinez, dibujoPlanoGenera: boceto-casa, numeroDeFiguras: 21
Plano: id: 2, numeroDeIdentificacion: 10, fechadeEntrga: Fri Jan 15 00:06:00 COT 2021, arquitectos: Lilian-Godoy, dibujoPlanoGenera: boceto-edificio, numeroDeFiguras: 36
Plano: id: 4, numeroDeIdentificacion: 15, fechadeEntrga: Sun Jan 10 00:07:00 COT 2021, arquitectos: Jon-Perez, dibujoPlanoGenera: boceto-puente, numeroDeFiguras: 20

+++++Gestion de Planos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
    
```

- JefeDeProyecto

Crear

```
++++Jefes de Proyectos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el nombre: Pablo-Martinez
Ingrese el codigo: 010
Ingrese la direccion: Cuenca
Ingrese el telefono: 096354
Resultado: true

++++Jefes de Proyectos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el nombre: Lilian-Godoy
Ingrese el codigo: 020
Ingrese la direccion: Cuenca
Ingrese el telefono: 068745
Resultado: true
```

Buscar

```
++++Jefes de Proyectos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el codigo: 010

JefeDeProyecto:  id: 1, nombre: Pablo-Martinez
codigo: 010, direccion: Cuenca
telefono: 096354

++++Jefes de Proyectos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el codigo: 020

JefeDeProyecto:  id: 2, nombre: Lilian-Godoy
codigo: 020, direccion: Cuenca
telefono: 068745
```

Actualizar

```
++++Jefes de Proyectos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
3
Ingrese el codigo: 011

JefeDeProyecto:  id: 4, nombre: Jon-Herrera
codigo: 011, direccion: Giron
telefono: 06845
Ingrese el nombre nuevo: Axel
Ingrese el codigo nuevo: 030
Ingrese la direccion nuevo: Giron
Ingrese el telefono nuevo: 08746
Resultado: true
```

Eliminar

```
++++Jefes de Proyectos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
4
Ingrese el codigo: 123

JefeDeProyecto:  id: 3, nombre: Martin
codigo: 123, direccion: Cuenca
telefono: 05478
Resultado: true
```

Listar

```
++++Jefes de Proyectos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
5

JefeDeProyecto: id: 1, nombre: Pablo-Martinez
codigo: 010, direccion: Cuenca
telefono: 096354

JefeDeProyecto: id: 2, nombre: Lilian-Godoy
codigo: 020, direccion: Cuenca
telefono: 068745

JefeDeProyecto: id: 4, nombre: Axel
codigo: 030, direccion: Giron
telefono: 08746

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

- Figura

Crear

```
++++Gestion de Figuras++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el codigo: 1
Ingrese el nombre: Cuadrado
Ingrese la color: azul
Resultado: true

++++Gestion de Figuras++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el codigo: 55
Ingrese el nombre: triangulo
Ingrese la color: amarillo
Resultado: true
```

Buscar

```
+++++Gestion de Figuras+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el color: rojo

Figura:   id: 5, codIdentificador: 77
         nombre: Rombo, color: rojo
```

Actualizar

```
+++++Gestion de Figuras+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
3
Ingrese el color: amarillo

Figura:   id: 2, codIdentificador: 55
         nombre: triangulo, color: amarillo
Ingrese el codigo nuevo: 88
Ingrese el nombre nuevo: circulo
Ingrese la color nuevo: verde-claro
Resultado: true
```

Eliminar

```
+++++Gestion de Figuras+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
4
Ingrese el color: verde

Figura:   id: 4, codIdentificador: 10
         nombre: circulo, color: verde
Resultado: true
```

Listar


```
++++Gestion de Figuras++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
5

Figura: id: 1, codIdentificador: 1
nombre: Cuadrado, color: azul

Figura: id: 2, codIdentificador: 88
nombre: circulo, color: verde-claro

Figura: id: 3, codIdentificador: 67
nombre: rectangulo-cuadrado, color: amarillo

Figura: id: 5, codIdentificador: 77
nombre: Rombo, color: rojo

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

- Poligono

Crear

```
++++Gestion de Poligonos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el codigo: 111
Ingrese el nombre: triangulo
Ingrese la color: rosa
Ingrese el numero de lineas: 3
Resultado: true

++++Gestion de Poligonos++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el codigo: 222
Ingrese el nombre: circulo
Ingrese la color: marron
Ingrese el numero de lineas: 1
Resultado: true
```

Buscar

```
+++++Gestion de Poligonos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el color: verde

Poligono:  numeroDeLineas: 7

Figura:  id: 7, codIdentificador: 777
        nombre: Heptagono, color: verde

+++++Gestion de Poligonos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el color: fuccia

Poligono:  numeroDeLineas: 4

Figura:  id: 4, codIdentificador: 123
        nombre: cuadrado, color: fuccia
```

Actualizar

```
+++++Gestion de Poligonos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
3
Ingrese el color: fuccia

Poligono:  numeroDeLineas: 4

Figura:  id: 4, codIdentificador: 123
        nombre: cuadrado, color: fuccia
Ingrese el codigo nuevo: 444
Ingrese el nombre nuevo: Cuadrilatero
Ingrese la color nuevo: negro
Ingrese el nuevo numero de lineas: 4
Resultado: true
```

Eliminar

```
+++++Gestion de Poligonos+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
4
Ingrese el color: rosa

Poligono:  numeroDeLineas: 3

Figura:  id: 3, codIdentificador: 111
nombre: triangulo, color: rosa
Resultado: true
```

Listar

```
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
5

Poligono:  numeroDeLineas: 1

Figura:  id: 1, codIdentificador: 222
nombre: circulo, color: marron

Poligono:  numeroDeLineas: 7

Figura:  id: 7, codIdentificador: 777
nombre: Heptagono, color: verde

Poligono:  numeroDeLineas: 4

Figura:  id: 4, codIdentificador: 444
nombre: Cuadrilatero, color: negro

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

- LineaDeFormacion

Crear

```
+++++Lineas de Formacion+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el inicio de coordenada X: 0
Ingrese inicio de coordenada Y: 2
Ingrese fin de coordenada X: 3
Ingrese fin de coordenada X: 3
Ingrese la longitud: 3
Ingrese el color: negro
Resultado: true

+++++Lineas de Formacion+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
1
Ingrese el inicio de coordenada X: 1,5
Ingrese inicio de coordenada Y: 2,5
Ingrese fin de coordenada X: 7
Ingrese fin de coordenada X: 8
Ingrese la longitud: 6,5
Ingrese el color: marron
Resultado: true
```

Buscar

```
+++++Lineas de Formacion+++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
2
Ingrese el color: negro

LineaDeFormacion:  id: 1, iniciocoordenadaX: 0.0
iniciocoordenadaY: 2.0, finCoordenadaX: 3.0
finCoordenadaY: 3.0, longitud: 3.0, color: negro
```

Actualizar

```
++++Lineas de Formacion++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
3
Ingrese el color: amarillo

LineaDeFormacion: id: 4, iniciocoordenadaX: 2.0
  iniciocoordenadaY: 7.0, finCoordenadaX: 10.5
  finCoordenadaY: 11.0, longitud: 9.0, color: amarillo
Ingrese el nuevo inicio de coordenada X: 1
Ingrese el nuevo inicio de coordenada Y: 3
Ingrese el nuevo fin de coordenada X: 12
Ingrese el nuevo fin de coordenada X: 12
Ingrese la longitud nueva: 11
Ingrese el color nuevo: rojo
Resultado: true
```

Eliminar

```
++++Lineas de Formacion++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
4
Ingrese el color: morado

LineaDeFormacion: id: 3, iniciocoordenadaX: 0.0
  iniciocoordenadaY: 1.0, finCoordenadaX: 3.0
  finCoordenadaY: 3.0, longitud: 2.0, color: morado
Resultado: true
```

Listar

```
++++Lineas de Formacion++++
1.- Crear
2.- Buscar
3.- Actualizar
4.- Eliminar
5.- Listar
6.- Salir
5

LineaDeFormacion: id: 1, iniciocoordenadaX: 0.0
iniciocoordenadaY: 2.0, finCoordenadaX: 3.0
finCoordenadaY: 3.0, longitud: 3.0, color: negro

LineaDeFormacion: id: 2, iniciocoordenadaX: 1.5
iniciocoordenadaY: 2.5, finCoordenadaX: 7.0
finCoordenadaY: 8.0, longitud: 6.5, color: marron

LineaDeFormacion: id: 4, iniciocoordenadaX: 1.0
iniciocoordenadaY: 3.0, finCoordenadaX: 12.0
finCoordenadaY: 12.0, longitud: 9.0, color: rojo

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