

CARRERA: Computación

ASIGNATURA: Programación Orientada a Objetos

NRO. PRÁCTICA: 1 **TÍTULO PRÁCTICA:** MVC.

OBJETIVO (Colocar el o los objetivos que se alcanzarán al desarrollar la práctica):

- Generar un sistema basado en MVC del siguiente enunciado.
-

INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):

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

2. A continuación, las especificaciones del **Diagrama de clases**:

- Cada proyecto tiene un código y un nombre. Un proyecto tiene uno y solo un jefe de proyecto y un jefe de proyecto sólo puede estar involucrado en un proyecto o en ninguno.
- De cada jefe de proyecto se desean recoger sus datos personales (código, nombre, dirección y teléfono). Un jefe de proyecto se identifica por un código. No hay dos nombres de jefe de proyecto con el mismo nombre.
- Un proyecto se compone de una serie de planos, pero éstos se quieren guardar de modo independiente al proyecto. Es decir, si en un momento dado se dejara de trabajar en un proyecto, se desea mantener la información de los planos asociados.
- De los planos se desea guardar su número de identificación, la fecha de entrega, los arquitectos que trabajan en él y un dibujo del plano general con información acerca del número de figuras que contiene.
- Los planos tienen figuras. De cada figura se desea conocer, el identificador, el nombre, el color, el área y el perímetro. Además, de los polígonos se desea conocer el número de líneas que tienen, además de las líneas que lo forman. El perímetro se desea que sea un método diferido; el área se desea implementarlo como genérico para cualquier tipo de figura, pero además se desea un método específico para el cálculo del perímetro de los polígonos.
- De cada línea que forma parte de un polígono se desea conocer el punto de origen y el de fin (según sus coordenadas, X e Y), así como la longitud. Cada línea tiene un identificador que permite diferenciarlo del resto. La longitud de la línea se puede calcular a partir de sus puntos origen y final.

5. Criterios de valoración:

Criterio	Puntaje
MVC	
• MVC	3
• Usabilidad	1
• Buenas prácticas en el nombre de clases y atributos	1
• Total	5

6. Consideraciones:

- La prueba se realizara individual.
- Tienen hasta 20/06/2021 23:55 para terminar a partir de la explicación.
- Generar un archivo **.pdf** el nombre debe tener el siguiente formato ApellidoNombre con las capturas del funcionamiento del sistema.
- Subir el código fuente y el archivo pdf al Git Personal con el nombre de la caperta PruebaPractica2

ACTIVIDADES POR DESARROLLAR

(Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)

1. Realizar el sistema empleando MVC para la gestión de datos y las buenas practicas de programación.

El diagrama de clases debe cumplir con las buenas prácticas indicadas en clase.

RESULTADO(S) OBTENIDO(S):

Escribir los resultados obtenidos con la realización de la práctica.

- Construye sistemas transaccionales utilizando MVC.

CONCLUSIONES:

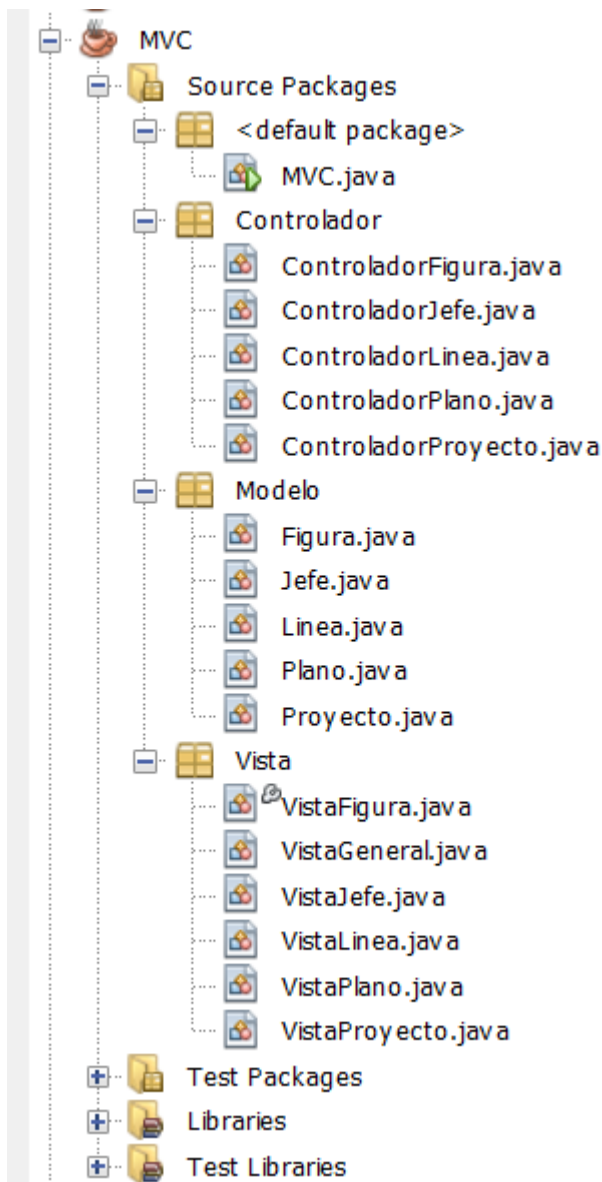
- Cada estudiante podrá desarrollar un diagrama de clases identificando la información primordial en base a un problema.

RECOMENDACIONES:

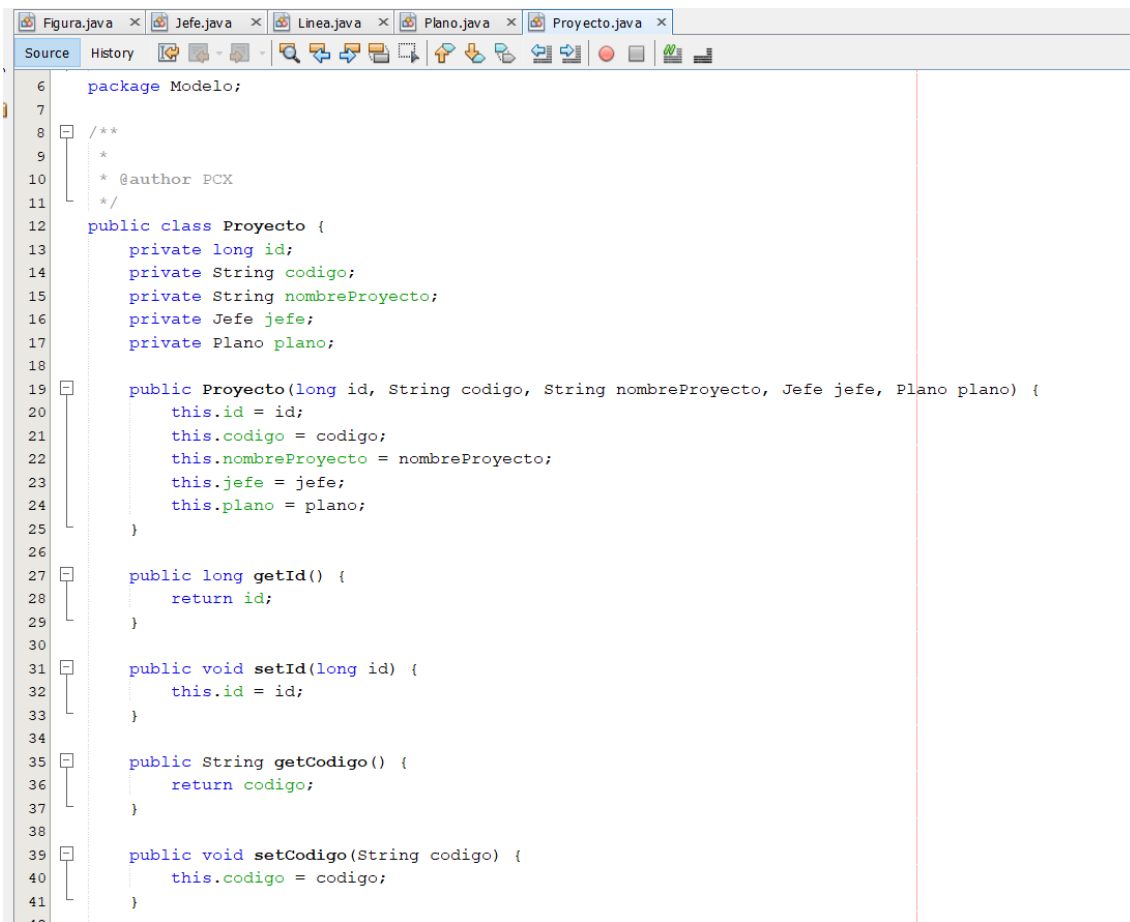
- Revisar el material de los ejercicios.
- Usar Dia.

Docente / Técnico Docente: Inq. Diego Quisi

Firma: _____

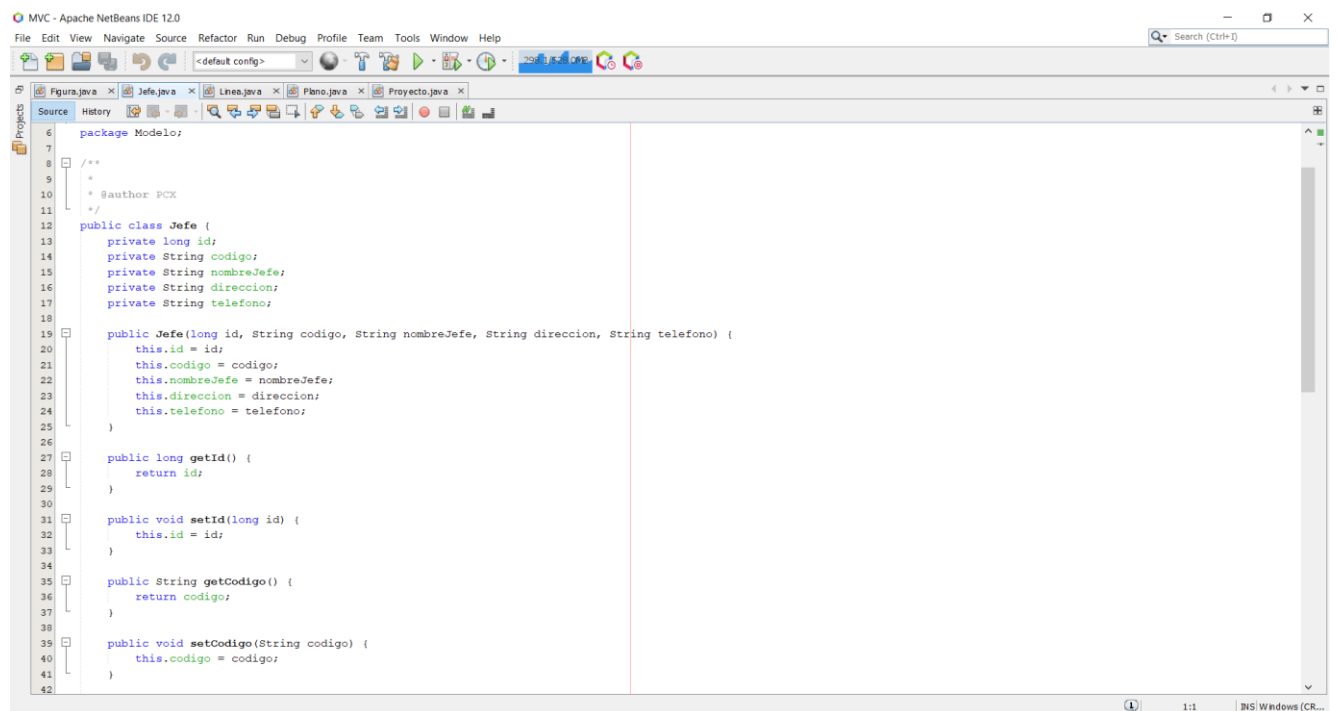


MODELO



The screenshot shows the NetBeans IDE with the 'Proyecto.java' file open. The code defines a package 'Modelo' and a class 'Proyecto'. The class has private attributes for 'id', 'codigo', 'nombreProyecto', 'jefe', and 'plano'. It includes a constructor and four methods: 'getId()', 'setId()', 'getCodigo()', and 'setCodigo()'. The IDE interface includes a toolbar with various icons and a 'Source' tab selected.

```
6 package Modelo;
7
8 /**
9  *
10  * @author PCX
11  */
12 public class Proyecto {
13     private long id;
14     private String codigo;
15     private String nombreProyecto;
16     private Jefe jefe;
17     private Plano plano;
18
19     public Proyecto(long id, String codigo, String nombreProyecto, Jefe jefe, Plano plano) {
20         this.id = id;
21         this.codigo = codigo;
22         this.nombreProyecto = nombreProyecto;
23         this.jefe = jefe;
24         this.plano = plano;
25     }
26
27     public long getId() {
28         return id;
29     }
30
31     public void setId(long id) {
32         this.id = id;
33     }
34
35     public String getCodigo() {
36         return codigo;
37     }
38
39     public void setCodigo(String codigo) {
40         this.codigo = codigo;
41     }
42 }
```



The screenshot shows the NetBeans IDE with the 'Jefe.java' file open. The code defines a package 'Modelo' and a class 'Jefe'. The class has private attributes for 'id', 'codigo', 'nombreJefe', 'direccion', and 'telefono'. It includes a constructor and four methods: 'getId()', 'setId()', 'getCodigo()', and 'setCodigo()'. The IDE interface includes a toolbar with various icons and a 'Source' tab selected.

```
6 package Modelo;
7
8 /**
9  *
10  * @author PCX
11  */
12 public class Jefe {
13     private long id;
14     private String codigo;
15     private String nombreJefe;
16     private String direccion;
17     private String telefono;
18
19     public Jefe(long id, String codigo, String nombreJefe, String direccion, String telefono) {
20         this.id = id;
21         this.codigo = codigo;
22         this.nombreJefe = nombreJefe;
23         this.direccion = direccion;
24         this.telefono = telefono;
25     }
26
27     public long getId() {
28         return id;
29     }
30
31     public void setId(long id) {
32         this.id = id;
33     }
34
35     public String getCodigo() {
36         return codigo;
37     }
38
39     public void setCodigo(String codigo) {
40         this.codigo = codigo;
41     }
42 }
```

MVC - Apache NetBeans IDE 12.0

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Figura.java Jefe.java Linea.java Plano.java Proyecto.java
Source History
package Modelo;
/**
 *
 * @author PCK
 */
public class Plano {
    private long id;
    private String fecha_entrega;
    private String arquitectos;
    private Figura figura;

    public Plano(long id, String fecha_entrega, String arquitectos, Figura figura) {
        this.id = id;
        this.fecha_entrega = fecha_entrega;
        this.arquitectos = arquitectos;
        this.figura = figura;
    }

    public long getId() {
        return id;
    }

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

    public String getFecha_entrega() {
        return fecha_entrega;
    }

    public void setFecha_entrega(String fecha_entrega) {
        this.fecha_entrega = fecha_entrega;
    }

    public String getArquitectos() {
```

MVC - Apache NetBeans IDE 12.0

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Figura.java Jefe.java Linea.java Plano.java Proyecto.java
Source History
package Modelo;
/**
 *
 * @author PCK
 */
public class Figura {
    private long id;
    private int cantidad;
    private String color;
    private double area;
    private double perimetro;
    private Linea linea;

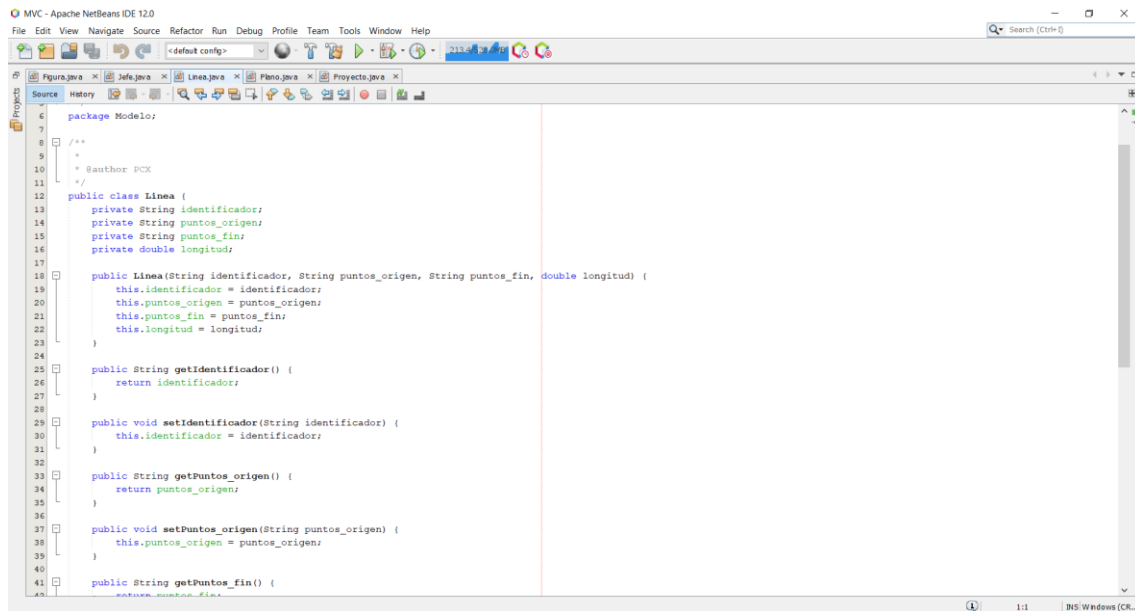
    public Figura(long id, int cantidad, String color, double area, double perimetro, Linea linea) {
        this.id = id;
        this.cantidad = cantidad;
        this.color = color;
        this.area = area;
        this.perimetro = perimetro;
        this.linea = linea;
    }

    public long getId() {
        return id;
    }

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

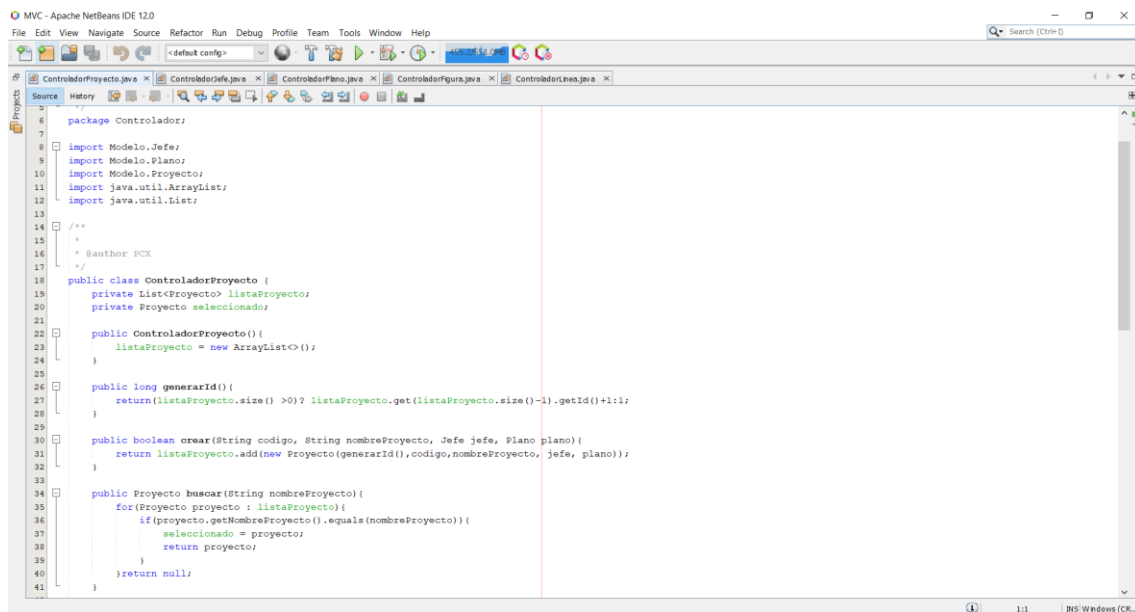
    public int getCantidad() {
        return cantidad;
    }

    public void setCantidad(int cantidad) {
```



```
1 package Modelo;
2
3 /**
4  *
5  * @author PCK
6  */
7 public class Linea {
8     private String identificador;
9     private String puntos_origen;
10    private String puntos_fin;
11    private double longitud;
12
13    public Linea(String identificador, String puntos_origen, String puntos_fin, double longitud) {
14        this.identificador = identificador;
15        this.puntos_origen = puntos_origen;
16        this.puntos_fin = puntos_fin;
17        this.longitud = longitud;
18    }
19
20    public String getIdentificador() {
21        return identificador;
22    }
23
24    public void setIdentificador(String identificador) {
25        this.identificador = identificador;
26    }
27
28    public String getPuntos_origen() {
29        return puntos_origen;
30    }
31
32    public void setPuntos_origen(String puntos_origen) {
33        this.puntos_origen = puntos_origen;
34    }
35
36    public String getPuntos_fin() {
37        return puntos_fin;
38    }
39
40    public void setPuntos_fin(String puntos_fin) {
41        this.puntos_fin = puntos_fin;
42    }
43
44    public double getLongitud() {
45        return longitud;
46    }
47
48    public void setLongitud(double longitud) {
49        this.longitud = longitud;
50    }
51
52    public boolean equals(Object o) {
53        if (o == null) return false;
54        if (o instanceof Linea) {
55            Linea l = (Linea) o;
56            return this.identificador.equals(l.identificador) &&
57                this.puntos_origen.equals(l.puntos_origen) &&
58                this.puntos_fin.equals(l.puntos_fin) &&
59                this.longitud.equals(l.longitud);
60        }
61        return false;
62    }
63
64    public int hashCode() {
65        return this.identificador.hashCode() +
66            this.puntos_origen.hashCode() +
67            this.puntos_fin.hashCode() +
68            this.longitud.hashCode();
69    }
70
71    public String toString() {
72        return "Linea[" + identificador + ", " + puntos_origen + ", " + puntos_fin + ", " + longitud + "]";
73    }
74
75    public static void main(String[] args) {
76        Linea l = new Linea("1", "A", "B", 100);
77        System.out.println(l);
78        l.setIdentificador("2");
79        System.out.println(l);
80        l.setPuntos_origen("C");
81        System.out.println(l);
82        l.setPuntos_fin("D");
83        System.out.println(l);
84        l.setLongitud(200);
85        System.out.println(l);
86    }
87
88 }
```

CONTROLADORES



```
1 package Controlador;
2
3 import Modelo.Jefe;
4 import Modelo.Plano;
5 import Modelo.Proyecto;
6 import java.util.ArrayList;
7 import java.util.List;
8
9 /**
10  *
11  * @author PCK
12  */
13 public class ControladorProyecto {
14     private List<Proyecto> listaProyecto;
15     private Proyecto seleccionado;
16
17     public ControladorProyecto() {
18         listaProyecto = new ArrayList<>();
19     }
20
21     public long generarId() {
22         return (listaProyecto.size() > 0) ? listaProyecto.get(listaProyecto.size() - 1).getId() + 1 : 1;
23     }
24
25     public boolean crear(String codigo, String nombreProyecto, Jefe jefe, Plano plano) {
26         return listaProyecto.add(new Proyecto(generarId(), codigo, nombreProyecto, jefe, plano));
27     }
28
29     public Proyecto buscar(String nombreProyecto) {
30         for (Proyecto proyecto : listaProyecto) {
31             if (proyecto.getNombreProyecto().equals(nombreProyecto)) {
32                 seleccionado = proyecto;
33                 return proyecto;
34             }
35         }
36         return null;
37     }
38
39     public Proyecto getSeleccionado() {
40         return seleccionado;
41     }
42
43     public void setSeleccionado(Proyecto proyecto) {
44         seleccionado = proyecto;
45     }
46
47     public List<Proyecto> getListaProyecto() {
48         return listaProyecto;
49     }
50
51     public void setListaProyecto(List<Proyecto> listaProyecto) {
52         this.listaProyecto = listaProyecto;
53     }
54
55 }
```

```

MVC - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
ControladorProyecto.java x ControladorJefe.java x ControladorPlano.java x ControladorFigura.java x ControladorLinea.java x
Source History
41:
42:
43: public boolean actualizar(String codigo, String nombreProyecto, Jefe jefe, Plano plano) {
44:     Proyecto proyecto = buscar(nombreProyecto);
45:     if (proyecto != null) {
46:         int posicion = listaProyecto.indexOf(proyecto);
47:         proyecto.setCodigo(codigo);
48:         proyecto.setJefe(jefe);
49:         proyecto.getPlano();
50:         listaProyecto.set(posicion, proyecto);
51:         return true;
52:     }
53:     return false;
54: }
55:
56: public boolean eliminar(String nombreProyecto) {
57:     Proyecto proyecto = buscar(nombreProyecto);
58:     return listaProyecto.remove(proyecto);
59: }
60:
61: public List<Proyecto> getListProyecto() {
62:     return listaProyecto;
63: }
64:
65: public void setListaProyecto(List<Proyecto> listaProyecto) {
66:     this.listaProyecto = listaProyecto;
67: }
68:
69: public Proyecto getSeleccionado() {
70:     return seleccionado;
71: }
72:
73: public void setSeleccionado(Proyecto seleccionado) {
74:     this.seleccionado = seleccionado;
75: }
76:
77:

```

```

MVC - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
ControladorProyecto.java x ControladorJefe.java x ControladorPlano.java x ControladorFigura.java x ControladorLinea.java x
Source History
6: package Controlador;
7:
8: import Modelo.Jefe;
9: import java.util.ArrayList;
10: import java.util.List;
11:
12: /**
13:  *
14:  * @author PCK
15:  */
16: public class ControladorJefe {
17:     private List<Jefe> listaJefe;
18:     private Jefe seleccionado;
19:
20:     public ControladorJefe() {
21:         listaJefe = new ArrayList<>();
22:     }
23:
24:     public long generarId() {
25:         return (listaJefe.size() > 0) ? listaJefe.get(listaJefe.size() - 1).getId() + 1 : 1;
26:     }
27:
28:     public boolean crear(String codigo, String nombreJefe, String direccion, String telefono) {
29:         return listaJefe.add(new Jefe(generarId(), codigo, nombreJefe, direccion, telefono));
30:     }
31:
32:     public Jefe buscar(String codigo) {
33:         for (Jefe jefe : listaJefe) {
34:             if (jefe.getCodigo().equals(codigo)) {
35:                 seleccionado = jefe;
36:                 return jefe;
37:             }
38:         }
39:         return null;
40:     }
41:
42:     public boolean actualizar(String codigo, String nombreJefe, String direccion, String telefono) {
43:         Jefe jefe = buscar(codigo);

```

```
MVC - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search [Ctrl-F]

ControladorProyecto.java x ControladorIfe.java x ControladorPlano.java x ControladorFigura.java x ControladorLinea.java x
Source History
package Controlador;
import Modelo.Figura;
import Modelo.Plano;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author PCK
 */
public class ControladorPlano {
    private List<Plano> listaPlano;
    private Plano seleccionado;

    public ControladorPlano() {
        listaPlano = new ArrayList();
    }

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

    public boolean crear(String fecha_entrega, String arquitectos, Figura figura) {
        return listaPlano.add(new Plano(generarId(), fecha_entrega, arquitectos, figura));
    }

    public Plano buscar(String fecha_entrega) {
        for (Plano plano : listaPlano) {
            if (plano.getFecha_entrega().equals(fecha_entrega)) {
                seleccionado = plano;
                return plano;
            }
        }
        return null;
    }

    public boolean actualizar(String fecha_entrega, String arquitectos, Plano plano) {

```

```
MVC - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search [Ctrl-F]

ControladorProyecto.java x ControladorIfe.java x ControladorPlano.java x ControladorFigura.java x ControladorLinea.java x
Source History
package Controlador;
import Modelo.Figura;
import Modelo.Linea;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author PCK
 */
public class ControladorFigura {
    private List<Figura> listaFigura;
    private Figura seleccionado;

    public ControladorFigura() {
        listaFigura = new ArrayList();
    }

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

    public boolean crear(int cantidad, String color, double area, double perimetro, Linea linea) {
        return listaFigura.add(new Figura(generarId(), cantidad, color, area, perimetro, linea));
    }

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

    public boolean actualizar(int cantidad, String color, double area, double perimetro, Linea linea) {
        Figura figura = buscar(color);

```

```

14  *
15  * Saurthor PCK
16  */
17  public class ControladorLinea {
18      private List<Linea> listaLinea;
19      private Linea seleccionado;
20
21      public ControladorLinea() {
22          listaLinea = new ArrayList();
23      }
24
25      public boolean crear(String identificador, String puntos_origen, String puntos_fin, double longitud) {
26          return listaLinea.add(new Linea(identificador, puntos_origen, puntos_fin, longitud));
27      }
28
29      public Linea buscar(String identificador) {
30          for (Linea linea : listaLinea) {
31              if (linea.getIdentificador().equals(identificador)) {
32                  seleccionado = linea;
33                  return linea;
34              }
35          }
36          return null;
37      }
38
39      public boolean actualizar(String identificador, String puntos_origen, String puntos_fin, double longitud) {
40          Linea linea = buscar(identificador);
41          if (linea != null) {
42              int posicion = listaLinea.indexOf(linea);
43              linea.setPuntos_origen(puntos_origen);
44              linea.setPuntos_fin(puntos_fin);
45              linea.setLongitud(longitud);
46              listaLinea.set(posicion, linea);
47              return true;
48          }
49          return false;
50      }
51
52      public boolean eliminar(String identificador) {

```

VISTAS

```

6  package Vista;
7
8  import Controlador.ControladorJefe;
9  import Controlador.ControladorPlano;
10 import Controlador.ControladorProyecto;
11 import Modelo.Proyecto;
12 import java.util.Scanner;
13
14 /**
15  *
16  * Saurthor PCK
17  */
18 public class VistaProyecto {
19     private Scanner teclado;
20     private ControladorProyecto controladorProyecto;
21     private ControladorJefe controladorJefe;
22     private ControladorPlano controladorPlano;
23
24     public VistaProyecto(ControladorProyecto controladorProyecto, ControladorJefe controladorJefe, ControladorPlano controladorPlano) {
25         teclado = new Scanner(System.in);
26         this.controladorProyecto = controladorProyecto;
27         this.controladorJefe = controladorJefe;
28         this.controladorPlano = controladorPlano;
29     }
30
31     public void menu() {
32         int opcion = 0;
33         do {
34             System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Eliminar \n 5. Listar \n 6. Salir");
35             opcion = teclado.nextInt();
36             switch (opcion) {
37                 case 1: this.crear(); break;
38                 case 2: this.actualizar(); break;
39                 case 3: this.buscar(); break;
40                 case 4: this.eliminar(); break;
41                 case 5: this.listar(); break;
42             }

```



```

MVC - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl-F)

VistaProyecto.java VistaJefe.java VistaPaso.java VistaFigura.java VistaPaso.java VistaGeneral.java
Source History
48 teclado.nextLine();
49 String nombreProyecto = teclado.nextLine();
50 System.out.println("Ingresar el código");
51 String codigo = teclado.nextLine();
52 System.out.println("Resultado: " + controladorProyecto.crear(codigo, nombreProyecto, controladorJefe.getSeleccionado(), controladorPlano.getSeleccionado()));
53 }
54
55 public void actualizar() {
56     System.out.println("Ingresar el nombre del proyecto");
57     teclado.nextLine();
58     String nombreProyecto = teclado.nextLine();
59     System.out.println("Ingresar el código");
60     String codigo = teclado.nextLine();
61     System.out.println("Resultado: " + controladorProyecto.actualizar(codigo, nombreProyecto, controladorJefe.getSeleccionado(), controladorPlano.getSeleccionado()));
62 }
63
64
65 public Proyecto buscar() {
66     System.out.println("Ingresar: \n nombre del proyecto");
67     teclado.nextLine();
68     String nombreProyecto = teclado.nextLine();
69     System.out.println("Ingresar el código");
70     String codigo = teclado.nextLine();
71     System.out.println(controladorProyecto.buscar(nombreProyecto));
72     return controladorProyecto.buscar(nombreProyecto);
73 }
74
75 public void eliminar() {
76     this.buscar();
77     System.out.println("Resultado: " + controladorProyecto.eliminar(controladorProyecto.getSeleccionado().getNombreProyecto()));
78 }
79
80 public void listar() {
81     for (Proyecto proyecto : controladorProyecto.getListaProyecto()) {
82         System.out.println(proyecto);
83     }
84 }

```

```

MVC - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl-F)

VistaProyecto.java VistaJefe.java VistaPaso.java VistaFigura.java VistaPaso.java VistaGeneral.java
Source History
8 import Controlador.ControladorJefe;
9 import Modelo.Jefe;
10 import java.util.Scanner;
11
12 /**
13  *
14  * @author PCK
15  */
16 public class VistaJefe {
17     public Scanner teclado;
18     private ControladorJefe controladorJefe;
19
20
21     public VistaJefe(ControladorJefe controladorJefe) {
22         teclado = new Scanner(System.in);
23         this.controladorJefe = controladorJefe;
24     }
25
26     public void menu() {
27         int opcion = 0;
28         do {
29             System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Eliminar \n 5. Listar \n 6. Salir");
30             opcion = teclado.nextInt();
31             switch (opcion) {
32                 case 1: this.crear(); break;
33                 case 2: this.actualizar(); break;
34                 case 3: this.buscar(); break;
35                 case 4: this.eliminar(); break;
36                 case 5: this.listar(); break;
37             }
38         } while (opcion < 6);
39     }
40
41     public void crear() {
42         System.out.println("Ingresar el código");
43         teclado.nextLine();
44         String codigo = teclado.nextLine();

```

```
MVC - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
VistaProyecto.java VistaLife.java VistaPano.java VistaFigura.java VistaLinea.java VistaGeneral.java
Source History
package Vista;
import Controlador.ControladorFigura;
import Controlador.ControladorPlano;
import Modelo.Plano;
import java.util.Scanner;

/**
 *
 * @author PCK
 */
public class VistaPlano {
    public Scanner teclado;
    private ControladorPlano controladorPlano;
    private ControladorFigura controladorFigura;

    public VistaPlano(ControladorPlano controladorPlano, ControladorFigura controladorFigura) {
        teclado = new Scanner(System.in);
        this.controladorPlano = controladorPlano;
    }

    public void menu(){
        int opcion = 0;
        do {
            System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Eliminar \n 5. Listar \n 6. Salir");
            opcion = teclado.nextInt();
            switch(opcion){
                case 1: this.crear(); break;
                case 2: this.actualizar(); break;
                case 3: this.buscar(); break;
                case 4: this.eliminar(); break;
                case 5: this.listar(); break;
            }
        } while (opcion < 6);
    }

    public void crear() {
```

```
MVC - Apache NetBeans IDE 12.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
VistaProyecto.java VistaLife.java VistaPano.java VistaFigura.java VistaLinea.java VistaGeneral.java
Source History
public class VistaFigura {
    public Scanner teclado;
    private ControladorFigura controladorFigura;
    private ControladorLinea controladorLinea;

    public VistaFigura(ControladorFigura controladorFigura, ControladorLinea controladorLinea) {
        teclado = new Scanner(System.in);
        this.controladorFigura = controladorFigura;
        this.controladorLinea = controladorLinea;
    }

    public void menu(){
        int opcion = 0;
        do {
            System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Eliminar \n 5. Listar \n 6. Salir");
            opcion = teclado.nextInt();
            switch(opcion){
                case 1: this.crear(); break;
                case 2: this.actualizar(); break;
                case 3: this.buscar(); break;
                case 4: this.eliminar(); break;
                case 5: this.listar(); break;
            }
        } while (opcion < 6);
    }

    public void crear() {
        System.out.println("Ingresar la cantidad ");
        teclado.nextLine();
        int cantidad = teclado.nextInt();
        System.out.println("Ingresar el color ");
        String color = teclado.next();
        System.out.println("Ingresar el area");
        double area = teclado.nextDouble();
        System.out.println("Ingresar el perimetro");
        double perimetro = teclado.nextDouble();
    }
}
```

```

14  * @author PCX
15  */
16  public class VistaLinea {
17      public Scanner teclado;
18      private ControladorLinea controladorLinea;
19
20
21
22      public VistaLinea(ControladorLinea controladorLinea) {
23          teclado = new Scanner(System.in);
24          this.controladorLinea = controladorLinea;
25      }
26
27
28      public void menu() {
29          int opcion = 0;
30          do {
31              System.out.println(" 1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Eliminar \n 5. Listar \n 6. Salir");
32              opcion = teclado.nextInt();
33              switch(opcion) {
34                  case 1: this.crear(); break;
35                  case 2: this.actualizar(); break;
36                  case 3: this.buscar(); break;
37                  case 4: this.eliminar(); break;
38                  case 5: this.listar(); break;
39              }
40          } while (opcion < 6);
41      }
42
43      public void crear() {
44          System.out.println("Ingrese el identificador ");
45          teclado.nextLine();
46          String identificador = teclado.nextLine();
47          System.out.println("Ingrese los puntos de origen ");
48          String puntos_origen = teclado.nextLine();
49          System.out.println("Ingrese los puntos del fin");
50          String puntos_fin = teclado.nextLine();
51      }
52  }

```

```

5  package Vista;
6
7  import Controlador.ControladorFigura;
8  import Controlador.ControladorJefe;
9  import Controlador.ControladorLinea;
10 import Controlador.ControladorPlano;
11 import Controlador.ControladorProyecto;
12 import Modelo.Figura;
13 import Modelo.Jefe;
14 import Modelo.Linea;
15 import Modelo.Plano;
16 import java.util.Scanner;
17
18
19 /**
20  *
21  * @author PCX
22  */
23 public class VistaGeneral {
24     public Scanner teclado;
25     public VistaProyecto vistaProyecto;
26     public ControladorProyecto controladorProyecto;
27     public VistaJefe vistaJefe;
28     public ControladorJefe controladorJefe;
29     public VistaPlano vistaPlano;
30     public ControladorPlano controladorPlano;
31     public VistaFigura vistaFigura;
32     public ControladorFigura controladorFigura;
33     public VistaLinea vistaLinea;
34     public ControladorLinea controladorLinea;
35
36     public VistaGeneral() {
37         teclado = new Scanner(System.in);
38         controladorProyecto = new ControladorProyecto();
39         controladorJefe = new ControladorJefe();
40         controladorPlano = new ControladorPlano();
41         controladorFigura = new ControladorFigura();
42     }
43 }

```

```

36 public VistaGeneral() {
37     teclado = new Scanner(System.in);
38     controladorProyecto = new ControladorProyecto();
39     controladorJefe = new ControladorJefe();
40     controladorPlano = new ControladorPlano();
41     controladorFigura = new ControladorFigura();
42     controladorLinea = new ControladorLinea();
43     vistaProyecto = new VistaProyecto(controladorProyecto, controladorJefe, controladorPlano);
44     vistaJefe = new VistaJefe(controladorJefe);
45     vistaPlano = new VistaPlano(controladorPlano, controladorFigura);
46     vistaFigura = new VistaFigura(controladorFigura, controladorLinea);
47     vistaLinea = new VistaLinea(controladorLinea);
48 }
49
50
51 public void menu() {
52     int opcion = 0;
53     do {
54         System.out.println("Gestión de Registros");
55         System.out.println(" 1. Proyecto \n 2. Jefe \n 3. Plano \n 4. Figura \n 5 . Linea \n 6. Salir");
56         opcion = teclado.nextInt();
57         switch(opcion){
58             case 1: vistaProyecto.menu();break;
59             case 2: vistaJefe.menu(); break;
60             case 3: vistaPlano.menu();break;
61             case 4: vistaFigura.menu();break;
62             case 5: vistaLinea.menu();break;
63         }
64     } while (opcion < 6);
65 }
66
67
68
69 }
70

```

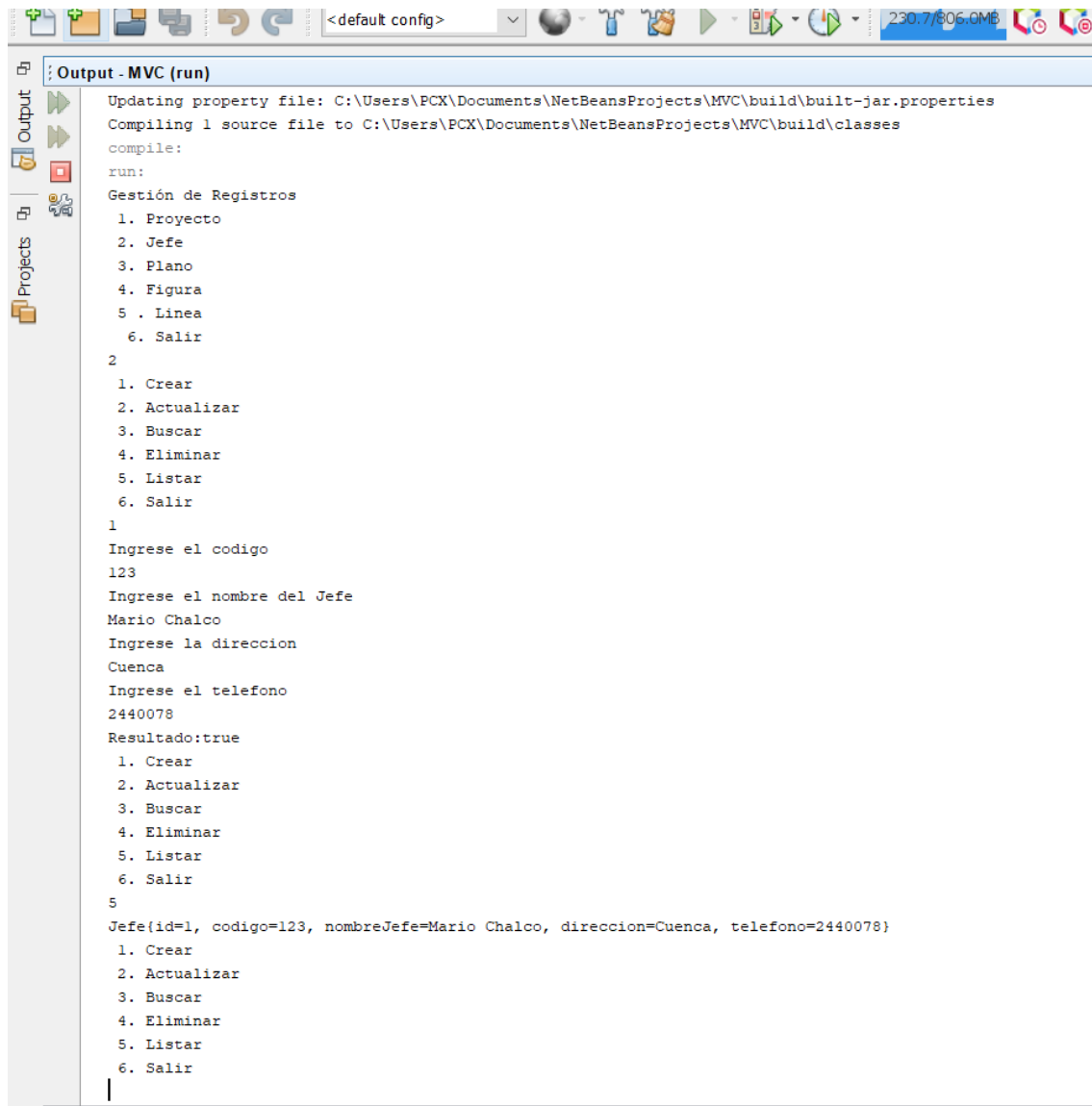
MAIN

```

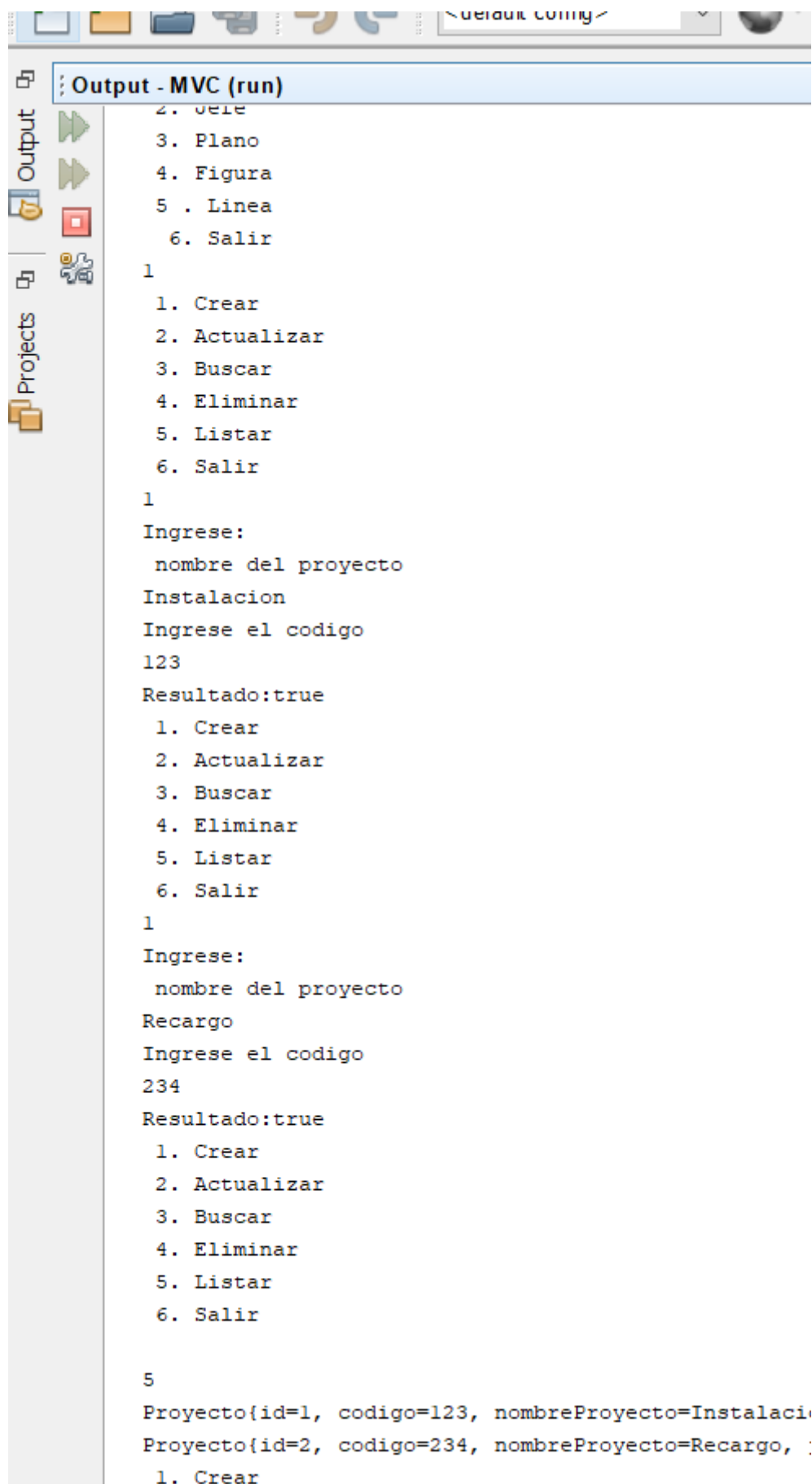
1
2 import Vista.VistaGeneral;
3
4
5 /*
6  * To change this license header, choose License Headers in Project Properties.
7  * To change this template file, choose Tools | Templates
8  * and open the template in the editor.
9  */
10
11 /**
12  *
13  * @author PCK
14  */
15 public class MVC {
16
17     /**
18      * @param args the command line arguments
19      */
20     public static void main(String[] args) {
21         // TODO code application logic here
22         VistaGeneral m=new VistaGeneral();
23         m.menu();
24     }
25 }
26

```

FUNCIONAMIENTO



```
Output - MVC (run)
Ingrese el telefono
2440078
Resultado:true
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
5
Jefe{id=1, codigo=123, nombreJefe=Mario Chalco, direccion=Cuenca, telefono=2440078}
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese el codigo
234
Ingrese el nombre del Jefe
Jorge Barros
Ingrese la direccion
Azogues
Ingrese el telefono
2440098
Resultado:true
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
5
Jefe{id=1, codigo=123, nombreJefe=Mario Chalco, direccion=Cuenca, telefono=2440078}
Jefe{id=2, codigo=234, nombreJefe=Jorge Barros, direccion=Azogues, telefono=2440098}
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
|
```



The screenshot shows an IDE's Output window titled "Output - MVC (run)". The window displays the output of an MVC application. On the left side, there is a vertical toolbar with icons for "Output" (a green arrow), "Projects" (a folder icon), and a "Run" button (a red square with a white play icon). The main area of the window contains the following text:

```
2. Sele
3. Plano
4. Figura
5 . Linea
6. Salir
1
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese:
nombre del proyecto
Instalacion
Ingrese el codigo
123
Resultado:true
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese:
nombre del proyecto
Recargo
Ingrese el codigo
234
Resultado:true
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir

5
Proyecto{id=1, codigo=123, nombreProyecto=Instalaci
Proyecto{id=2, codigo=234, nombreProyecto=Recargo, :
1. Crear
```

