

**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
<b>MVC</b>	
• 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:** \_\_\_\_\_

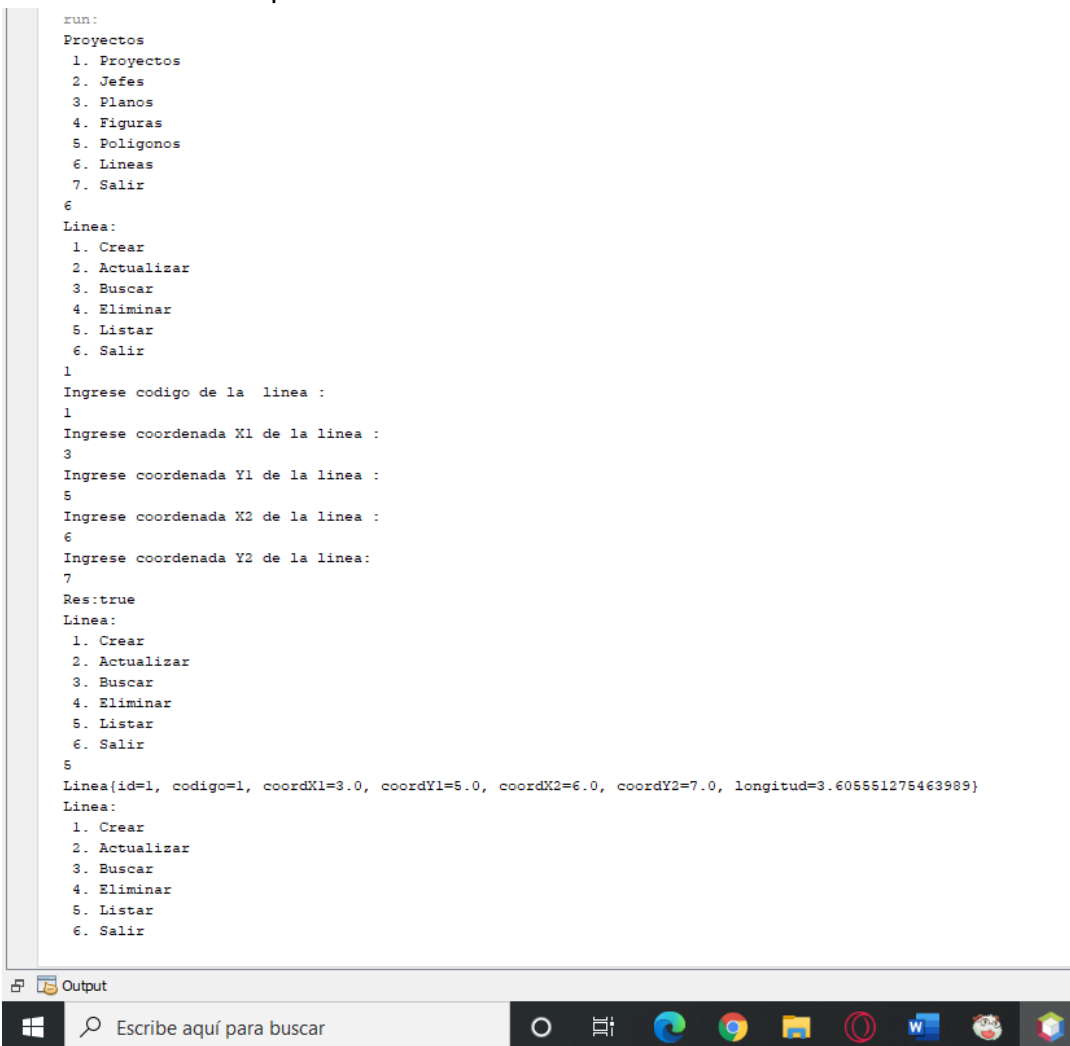
### Prueba MVC

Nombre: Edison Pintado

#### -USABILIDAD

##### 1-Primero tenemos que crear una linea

```
run:
Proyectos
1. Proyectos
2. Jefes
3. Planos
4. Figuras
5. Poligonos
6. Lineas
7. Salir
6
Linea:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese codigo de la linea :
1
Ingrese coordenada X1 de la linea :
3
Ingrese coordenada Y1 de la linea :
5
Ingrese coordenada X2 de la linea :
6
Ingrese coordenada Y2 de la linea:
7
Res:true
Linea:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
5
Linea{id=1, codigo=1, coordX1=3.0, coordY1=5.0, coordX2=6.0, coordY2=7.0, longitud=3.605551275463989}
Linea:
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
```



## 2-tenemos que crear una figura

```
4. figuras
5. Poligonos
6. Lineas
7. Salir
4
Figura
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
1
Ingrese color de la nueva figura a crear:
verde
Elija una Figura:
1. Cuadrado
2. Rectangulo
3. Triangulo
4. Circulo
5. Salir
2
Ingrese la longitud de la base del rectangulo:
5
Ingrese la longitud de la altura del rectangulo:
4
Ingrese el nombre del Rectangulo (R#):
03
Res:true
Figura
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
5
Figura{id=1. nombre=03. color=verde. area=20.0. perimetro=18.0. plano=null}
```

## 3 tenemos que crea un plano

```
-
Ingrese codigo del nuevo plano:
planos
Ingrese cuantos arquitectos diferentes agregar al nuevo plano:
2
Ingrese cuantas figuras diferentes agregar al nuevo plano:
2
Ingrese cuantos poligonos diferentes agregar al nuevo plano:
1
Ingrese el nombre del arquitecto 1 a agregar:
Edisson
Ingrese el nombre del arquitecto 2 a agregar:
Juan


Ingrese el nombre de la figura 1 a agregar:
03
Ingrese el nombre de la figura 2 a agregar:
03
Ingrese el nombre del poligono 1 a agregar:
03
Ingresar la Fecha de entrega del plano con formato (dd/mm/yyyy):
15/04/2021
Res:true
Planos
1. Crear
2. Actualizar
3. Buscar
4. Eliminar
5. Listar
6. Salir
5
Plano{id=1, fechaentrega=Fri Jan 15 00:04:00 COT 2021, listaArquitectos=[Edisson, ], listaFiguras=[Figura{id=1,
Planos
1. Crear
```

#### 4-tenemos que crear un jefe

```
Jefe:
  1. Crear
  2. Actualizar
  3. Buscar
  4. Eliminar
  5. Listar
  6. Salir
1
Ingrese nombre del nuevo jefe a crear:
Edisson
Ingrese direccion del nuevo jefe a crear:
Indurama
Ingrese telefono del nuevo jefe a crear:
4111900
Res:true
Jefe:
  1. Crear
  2. Actualizar
  3. Buscar
  4. Eliminar
  5. Listar
  6. Salir
5
Jefe{id=1, nombre=Edisson, direccion=Indurama, telefono=4111900, proyecto=null}
Jefe:
  1. Crear
  2. Actualizar
  3. Buscar
  4. Eliminar
  5. Listar
  6. Salir
|
```

#### 5 -para finalizar tenemos que crear el plano

```
Proyectos
  1. Proyectos
  2. Jefes
  3. Planos
  4. Figuras
  5. Poligonos
  6. Lineas
  7. Salir
1
Proyectos
  1. Crear
  2. Actualizar
  3. Buscar
  4. Eliminar
  5. Listar
  6. Salir
1
Ingrese nombre del nuevo proyecto:
proyecto01
Ingrese cuantos planos diferentes agregar:
3
Ingrese el nombre del nuevo plano a agregar:
planos
Ingrese nombre del jefe del nuevo proyecto:
Edisson
Res:true
Proyectos
  1. Crear
  2. Actualizar
  3. Buscar
  4. Eliminar
  5. Listar
  6. Salir
5
Proyecto{id=1, nombre=proyecto01, listaPlanos=(Plano{id=1, fechaentrega=Fri Jan 16 00:04:00 COT 2021, listaArquitectos=(Edisson, ), listaFiguras=(Figura{id=1, nombre=03, color=verde, area=20.0, perimetro=18.0, plano=null}), Figura{id=
Proyectos
  1. Crear
  2. Actualizar
  3. Buscar
  4. Eliminar
  5. Listar
  6. Salir
```



-MVC

