



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor(a): René Adrián Dávila Pérez

Asignatura: POO

Grupo: 01

No de Práctica(s): Practica 4

Integrante(s): 322059179

322125337

322080869

323629814

322113536

*No. de lista o
brigada:* 6

Semestre: 2025-2026

Fecha de entrega: 19/09/25

Observaciones:

CALIFICACIÓN: _____

Índice

1. Introducción	2
2. Marco Teórico	2
3. Desarrollo	3
4. Resultados	4
5. Conclusiones	5
6. Referencias bibliográficas	6

1. Introducción

- **Planteamiento del problema:** Hacer un código que tiene que ser capaz de tomar 2 puntos mediante String []args que en este caso serán pasados a objetos, después calcular la longitud entre esos 2 puntos en un plano con la formula $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, para después imprimir los puntos y resultados en una ventana mediante un botón.
- **Motivación:** Es de gran importancia que se comprendan los conceptos básicos de un lenguaje orientado a objetos para así poder trabajar posteriormente en futuros trabajos teniendo claro que es lo que estamos realizando con cada archivo, clase y objetos que estamos creando.
- **Objetivos:** Principalmente nuestro código tiene que ser capaz de tomar 2 puntos mediante y que seamos capaces de calcular la longitud entre esos 2 puntos teniendo en cuentas que ambos son objetos, para después poder imprimir los puntos y la distancia en una ventana.

2. Marco Teórico

Clase:En Java, una clase es una plantilla o molde que define las características y comportamientos de un tipo de objeto. Contiene atributos (variables) y métodos (funciones o procedimientos) que describen el estado y las acciones que puede realizar un objeto. Las clases son la base de la programación orientada a objetos (POO), permitiendo modelar entidades del mundo real de forma estructurada y reutilizable.[1]

Objeto: Como vimos en clase un objeto es una instancia de una clase. Es una entidad que tiene un estado, definido por sus atributos, y un comportamiento, definido por sus métodos. Cada objeto puede tener valores únicos en sus atributos, aunque se cree a partir de la misma clase.[1]

Atributos: Son las variables que se declaran dentro de una clase y que representan las propiedades o características del objeto. Por ejemplo, en una clase Persona, que tiene atributos como nombre o edad. Los atributos pueden tener distintos niveles de acceso (público, privado, protegido) y se accede a ellos generalmente a través de métodos.

Métodos:Son funciones definidas dentro de una clase que determinan el comportamiento de los objetos creados a partir de ella. Los métodos pueden realizar acciones como modificar atributos, procesar datos o interactuar con otros objetos. Java permite definir métodos con parámetros, valores de retorno y niveles de acceso. Existen también métodos especiales como los constructores, que se usan para inicializar objetos.

Constructores: Un constructor es un tipo especial de método dentro de una clase que se ejecuta automáticamente al crear un objeto. Su propósito principal es inicializar los atributos del objeto con valores predeterminados o específicos.

Interfaz Gráfica de Usuario (GUI): En java nos permite crear aplicaciones interactivas que se comunican con el usuario a través de elementos visuales como botones, ventanas, etiquetas, cuadros de texto, pero en este caso será mediante de ventanas y botones.

3. Desarrollo

Primero comenzamos por documentarnos acerca de las herramientas que se iban a utilizar en la practica, como el uso de ArrayList, el manejo de argumentos en el método main, las clases de apoyo para crear objetos, pero mas que nada nos centramos en la utilización de Swing para generar una interfaz gráfica en Java ya que era algo nuevo para nosotros. Una vez que aprendimos y entendimos como funciona cada herramienta, aplicamos lo aprendido para poder realizar la practica.

Decidimos que las coordenadas de los puntos iban a ser recibidas vía los argumentos del método main (`String[] args`), esto para que cuando se ejecute en la consola se pueden meter los 4 valores para representar 2 puntos. Después almacenamos estos valores en un `ArrayList<Integer>` convirtiendo cada argumento de tipo cadena a entero mediante `Integer.parseInt`, luego hicimos una prueba para asegurarnos que los datos se guardaran correctamente.

Después creamos dos objetos de la clase Punto, uno para el Punto A y otro para el Punto B, los inicializamos con los valores obtenidos en el ArrayList y después calculamos la distancia entre ambos puntos utilizando la fórmula de la distancia todo esto usando `Math.pow` para elevar al cuadrado las diferencias de coordenadas y `Math.sqrt` para obtener la raíz cuadrada del resultado.

Luego definimos la clase Mensajes para centralizar los textos que se mostrarían en la interfaz, de tal forma que cada método devuelve un mensaje específico como PuntoA, PuntoB y Distancia.

En la clase Ventana que es para crear la interfaz gráfica, ahí modificamos el botón visto en clase que, al ser presionado, ejecuta un ActionListener encargado de mostrar un JOptionPane con la información de los puntos y la distancia calculada, también configuramos propiedades de la ventana como el título, tamaño, posición centrada en la pantalla y la acción para que se termine el programa.

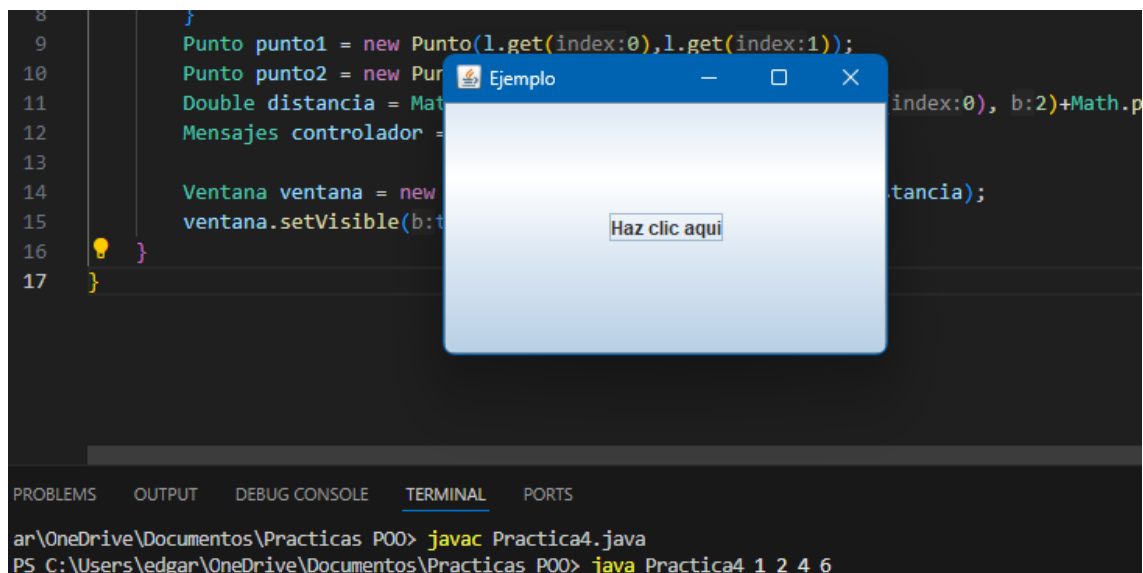
Por ultimo en el main instanciamos un objeto de la clase Ventana pasando como parámetros el controlador de mensajes, los dos puntos y la distancia calculada, y lla-

manos a setVisible(true) para hacerla visible y cuando se ejecuta el programa desde la consola con los argumentos necesarios, se abre una ventana con un botón que al hacer click se muestra un cuadro donde vienen las coordenadas de cada punto y la distancia entre ellos.

4. Resultados

El ejercicio se realizó correctamente ingresando 4 valores desde main y calculando la distancia entre 2 puntos

Ventana



The screenshot shows an IDE with a Java file named Practica4.java. The code defines a Punto class, a Mensajes class, and a main method. The main method takes four command-line arguments, converts them to integers, creates two Punto objects, calculates the distance between them, and displays the result in a window titled 'Ejemplo'. The window contains a button labeled 'Haz clic aqui'. The terminal at the bottom shows the command 'java Practica4 1 2 4 6' being executed.

```
8 }
9 Punto punto1 = new Punto(l.get(index:0),l.get(index:1));
10 Punto punto2 = new Punto(l.get(index:2),l.get(index:3));
11 Double distancia = Math.sqrt(Math.pow(l.get(index:0)-l.get(index:2),2)+Math.pow(l.get(index:1)-l.get(index:3),2));
12 Mensajes controlador = new Mensajes();
13
14 Ventana ventana = new Ventana(distancia);
15 ventana.setVisible(true);
16 }
17 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ar\OneDrive\Documentos\Practicas P00> javac Practica4.java
PS C:\Users\edgar\OneDrive\Documentos\Practicas P00> java Practica4 1 2 4 6

-Recibe argumentos por consola (args), se esperan 4 números enteros (x1, y1, x2, y2).

-Convierte esos valores a enteros y los guarda en una ArrayList.

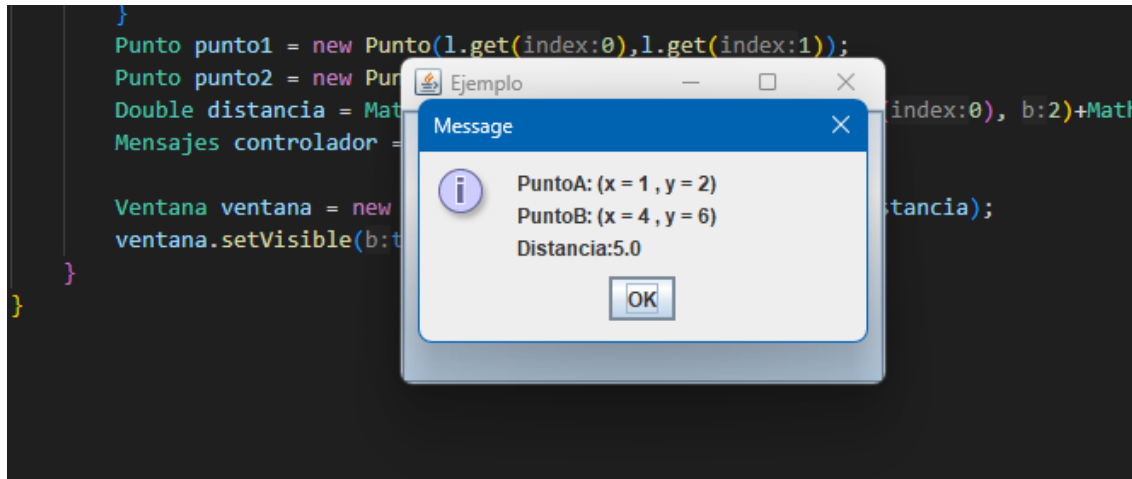
-Crea dos puntos (Punto), uno con (x1, y1) y otro con (x2, y2).

-Calcula la distancia entre los puntos con la fórmula de distancia

-Crea un objeto de la clase Mensajes.

-Abre una ventana (Ventana) que dice "Haz clic aqui"

Resultado



- Por ultimo se abre una ultima ventana donde se muestran los puntos y la distancia entre ellos.
- En este caso los puntos ingresados fueron $x_1=1$, $x_2=4$, $y_1=2$, $y_2=6$
- Se muestra el resultado del calculo, Distancia = 5.0

5. Conclusiones

La práctica nos permitió comprender y poner en practica el uso combinado de argumentos desde la línea de comandos así como herramientas vistas en clase tales como estructuras de datos como ArrayList, creación de objetos (para los puntos) y la construcción de una interfaz gráfica sencilla con Swing; pudimos ver de manera clara cómo los datos ingresados desde consola pueden procesarse para generar objetos que representen entidades reales que en este caso fueron dos puntos en un plano (x,y), y cómo a partir de ellos se pueden realizar cálculos como el de la distancia para después mostrar la misma.

Entender los puntos en un plano como objetos que se pueden relacionar entre sí para generar nueva información (como la distancia) es un excelente acercamiento a la filosofía de la POO. La cual consiste en construir sistemas complejos no a partir de instrucciones, sino a partir de objetos que colaboran entre sí, tal y como ocurre en el mundo real".

Implementando los códigos que se nos proporcionaron como parte de la revisión teórica de la práctica, tan solo fue necesario editar los métodos y los mensajes de las clases que ya estaban definidas, con el objetivo de que al presionar el botón que se generaba al ejecutar el código se imprimiera el resultado de la distancia como una ventana emergente.

6. Referencias bibliográficas

[1] Oracle. (s.f.). Concepts: Classes and Objects.
Recuperado de <https://docs.oracle.com/javase/tutorial/java/concepts/index.html>