	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: Programación Orientada a Objetos

Grupo: 1

No de Práctica(s): 4

Integrante(s): 322184022

322238682

321570789

322069093

322249723

No. de lista o brigada: 5

Semestre: 2026-1

Fecha de entrega: 19/09/2025

Observaciones:

CALIFICACIÓN: _____

Índice

1. Introducción

En esta práctica, se trabajó con el lenguaje de programación Java de forma que aplicando conceptos como clases, atributos, métodos, constructores y herencia, reforcemos nuestro conocimiento y manejo de este. El proyecto realizado se enfocó en la creación de un programa capaz de calcular la distancia entre dos puntos en un plano mediante el manejo y aplicación de estos conceptos . Para su elaboración, se trabajó con las librerias javax.swing y java.awt, las cuales sirven para la creación de interfaz gráfica. Dicha interfaz fue aplicada para la presentación del resultado mediante una ventana emergente. Con esto podemos decir que es posible relacionar diversas problemáticas, con clases y métodos concretos.

1.1. Planteamiento del problema

El problema se centro de la creación de un programa capaz de calcular la distancia entre dos puntos en un plano. Para su elaboración, se aplicaron conceptos de programación orientada a objetos, como clases, atributos, métodos y constructores. Además se utilizaron las librerias javax.swing y java,awt para desarrollar una interfaz gráfica que presenta el resultado en una ventana emergente.

1.2. Motivación

El reforzamiento del conocimiento y manejo del lenguaje de programación Java mediante la aplicación de conceptos fundamentales de POO. Ademas de adquirir experiencia en el manejo de las librerias swing y awt de interfaz gráfica, pudiendo relacionar la lógica del programa con la interfaz del usuario. Al relacionar una problematica en concreto, se demuestra la capacidad de POO para el modelo y resolución de problemáticas en la vida real.

1.3. Objetivos

- Comprender el uso de conceptos fundamentales en Programación Orientada a Objetos, tales como clases, atributos, métodos y constructores.
- Aplicar el uso de librerías como javax.swing y java.awt para la creación de interfaces gráficas
- Desarrollar habilidades en el manejo de entrada de datos (argumentos de linea de comandos) y la presentación de resultados.

2. Marco Teórico

- **Clases**: son modelos o abstracciones de la realidad que representan un elemento de un conjunto de *objetos* [3]

- **Objeto:** es la estructura o entidad abstracta que se crea a partir de la definición de una *clase*, de tal manera, que ésta permite organizar y relacionar los *datos* de un programa o problema (*atributos*) con sus respectivos *métodos*. [1]
    - **Atributos**: Conjunto de rasgos o características que posee un objeto [1]
    - **Métodos**: Conjunto de procedimientos o funciones para la manipulación o interacción con los datos de un objeto.[1]
        - **Constructor:** Método que tiene el mismo nombre que la clase y cuyo propósito es inicializar los atributos de un nuevo objeto. Se ejecuta automáticamente cuando se crea un objeto o instancia de la clase.[5]
        - **Método estático:** métodos que no requieren un objeto para ser invocados, se pueden ejecutar a través de la clase. [5]
- **Math en Java** : es una clase que pertenece al paquete java.lang, proporciona *métodos estáticos* para realizar operaciones matemáticas avanzadas, así como constantes matemáticas predefinidas. Su principal propósito es facilitar cálculos complejos sin necesidad de implementar algoritmos manualmente, mejorando la precisión y eficiencia del código. [4]
- **GUI**: Interfaces gráficas de usuario (GUI), es la capa visual que permite al usuario comunicarse con la aplicación de manera intuitiva, sin necesidad de usar una línea de comandos.
- **La distancia entre dos puntos:** En un plano cartesiano, la distancia entre dos puntos $A(x_1, y_1)$ y $B(x_2, y_2)$ se calcula utilizando el Teorema de Pitágoras:

$$\text{Distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

[2]

3. Desarrollo

- **Clase Punto**:
  - Atributos:
    - int x=0 : componente en x del punto, la inicializamos en 0
    - int y=0 : componente en y del punto la inicializamos en 0
  - Métodos:
    - Constructor: Recibe como parámetros dos numeros enteros, que los asigna al atributo x y al atributo y del objeto, en ese orden
    - public String toString: Formato de impresión del punto.
    - public double distancia (Punto p1, Punto p2): función que recibe dos objetos de tipo punto, dentro, accede a los atributos x y y de cada objeto, y se llama al *método estático* de Math llamado**hypot()**, que "automáticamente" o "internamente" lo que hace es elevar ambos parámetros al cuadrado, sumarlos y sacar su raíz, es decir, ***aplicar la fórmula de la distancia*** o el teorema de pitágoras. En este caso, los parámetros es la diferencia (resta) de los atributos x y y correspondientes de cada objeto Punto en los parámetros de la función distancia, que finalmente devuelve o retorna un número *double* que es justamente el resultado de la distancia entre ambos puntos.

- **Clase Mensaje**:
  En esta clase lo único que se define es la salida de los datos en la ***Ventana***, recibe como parámetos los dos objetos de tipo ***Punto*** de los que se desea obtener la distancia, ya que dentro de ella, imprimirá cada punto (con el formato que le dimos en la clase punto con toString, simplemente llamándolos a impresión), y llama al método distancia de la clase Punto, y pasa como parámetros los mismos puntos que recibió.

- **Clase Ventana**:
  Se importan las librerías *javax.swing.** y *java.awt.event.** para hacer uso de ellas al hacer la ventana emergente, swing es una biblioteca oficial de Java que contiene un conjunto completo de componentes (como JButton, JTextField, JFrame) para construir estas interfaces.
  *** public class Ventana **extends JFrame** {... : *extends* quiere decir que hereda métodos de *JFrame*, más adelante se llama a *Override*, que de igual forma tiene que ver con este concepto de herencia, el cual no se ahondará en esta práctica, ya que es un tema posterior, solo que en este caso se usa para cumplir con las indicaciones de hacer uso del GUI. [5]
  - Atributos:
    - JButton boton: Nuevo objeto de tipo boton (de la librería java.swing.*)
    - Mensaje msj: Nuevo objeto de tipo Mensaje

- Métodos:
  - Constructor: Recibe como parámetros: **msj** de tipo **Mensaje**, **p1** y **p2**, estos dos últimos ya que dentro de ella los necesitará para los argumentos de msj.
  - Funciones de JFrame para la ventana:
    - SetTitle: establecer el título de la ventana
    - SetSize: qué tan grande será la ventana
    - setLocationRelativeTo(null) : centra en el principal.
  - Para boton: Primero, se declara un nuevo boton de tipo JBotton donde su constructor es un String donde se indica qué debe imprimir.
    Después, con actionPerformed(ActionEvent e) le agrega una acción, que es mostrar el mensaje **msj**, pasándole sus respectivos parámetros (p1 y p2).
    Luego, con un nuevo *String mensaje*, almacena el mensaje de msj.
    Por último, con *JOptionPane.showMessageDialog(null, mensaje)* se muestra el mensaje de la ventana y se toma como argumento el String.

- **Clase Practica4** *(contiene el main)*: Dentro de la función main(String [] args):

1. Dentro de la función main(String [] args), se declaran 4 variables de enteros, los cuales serán los atributos (componentes) de los objetos Punto, y se les asignan los índices 0, 1, 2 y 3 del arreglo args del *main*, esto quiere decir que al momento de ejecutarlo, se debe ingresar en la terminal los valores de x_1, y_1, x_2, y_2, en ese orden, en este caso se debe hacer una función de los Wrappers que es el "desenvolverün dato, es decir, "convertirlo" de un tipo de dato a otro, esto lo hacemos con *Integer.parseInt(args[n])*,ya que quiere decir que el main recibe Strings, por lo que hay que convertirlo a un valor numérico (int).
2. Se declaran dos objetos **p1** y **p2** de tipo **Punto**, donde los parámetros para sus constructores son las variables correspondientes antes mencionadas.
3. Se declara un nuevo objeto **msj** de tipo **Mensaje**
4. Se declara un nuevo objeto de tipo **Ventana**, **vent**, que recibe como parámetros el **Mensaje msj**, los **Puntos p1** y **p2**, para que dentro de ella los comunique con la función de mostrar diálogo.
5. Por último, *vent.setVisible(true)* es un método que hace visible a la ventana.

4. Resultados

```java
public class EjPunto {
    Run | Debug
    public static void main (String [] args){

        //castear los args de string a int
        int x1 = Integer.parseInt(args[0]);
        int y1 = Integer.parseInt(args[1]);
        int x2 = Integer.parseInt(args[2]);
        int y2 = Integer.parseInt(args[3]);

        Punto p1 = new Punto(x1, y1);
        Punto p2 = new Punto(x2, y2);


        Mensaje msj = new Mensaje();


        Ventana vent = new Ventana(msj, p1, p2);
        vent.setVisible(true);


    }
}
```

Figura 1: Main y ejecución del programa

```java
public class Punto {

    // Atributos
    int x =0;
    int y = 0;

    //Métodos
    //Imprimirlo
    public String toString(){
        return "Punto: (x= " + x + ", y= " + y +")";
    }
    //calcular distancia
    public double distancia (Punto p1, Punto p2){
        return Math.hypot((p2.x-p1.x), (p2.y-p1.y));
    }
    //Constructores
    public Punto (){} //Inicializarlo sin argumentos (va a inicializar en los valores anteriores)
    //constructor con argumentos
    public Punto (int x, int y){
        this.x=x;
        this.y=y;
    }
}
```

Figura 2: Puntos y cálculo de la distancia

```java
import javax.swing.*;
import java.awt.event.*;
public class Ventana extends JFrame {
    JButton boton;
    Mensaje msj;

    public Ventana(Mensaje msj, Punto p1, Punto p2){
        this.msj = msj;
        setTitle("Práctica 3");
        setSize(300,200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        boton= new JButton("Calcular la distancia");
        boton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String mensaje = msj.mensaje(p1, p2);
                JOptionPane.showMessageDialog(null, mensaje);
            }
        });

        // Añade el botón a la ventana (JFrame)
        add(boton);

    }

}
```

Figura 3: Ventana de la aplicación

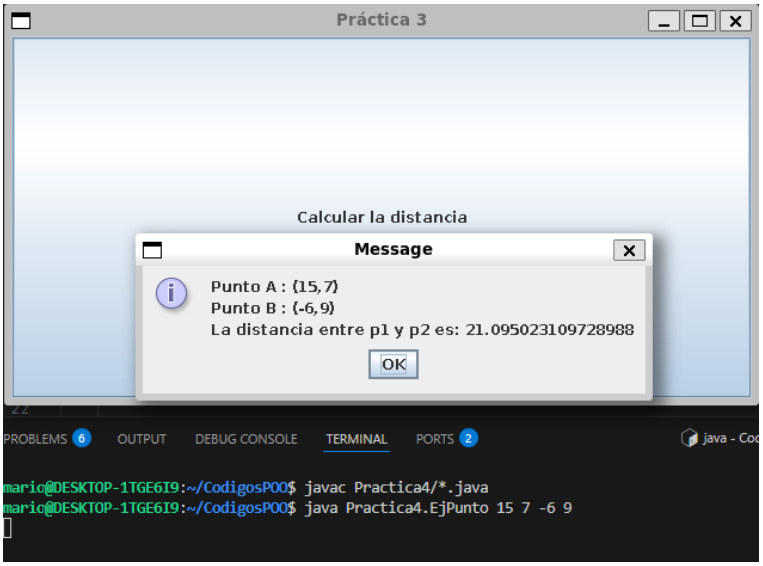

Figura 4: Ejecución del código

5. Conclusiones

En esta práctica se demostro exitosamente como la abstracción y el encapsulamiento son esenciales para el desarrollo de una solución funcional, ya que al definir la

clase *Punto*, encapsulamos lo que serían sus atributos y métodos, logrando trabajar con un par de coordenadas (x,y) como una sola entidad. La creación de dicha clase sirvió como una representación de los datos, la clase *Mensaje* para gestionar la comunicación entre los componentes y la clase *Ventana* para la interfaz gráfica, el poder divir el código en diversas "tareas" nos ayudó a controlar el manejo de datos y poder aplicar la fórmula de la distancia.
Finalmente, al implementar las librerías *javax.swing y java.awt* permite que nuestra solución pase de ser un programa de consola a una herramienta interactiva.

6. Referencias

[1] Alianza BUNAM. Conceptos básicos de programación orientada a objetos. (https://alianza.bunam.unam.mx/cch/conceptos-basicos-de-programacion-orientada-a-objetos/), 2022. Consultado el 18-09-2025.

[2] CUAIEED. Distancia entre dos puntos en el plano. (https://uapas1.bunam.unam.mx/matematicas/distancia$_puntos_en_plano/), 2017. Consultado el 18-09-2025.

[3] Portal Académico del CCH. Clases y objetos. (https://portalacademico.cch.unam.mx/cibernetica1/analisis-y-diseno-en-poo/clases-y-objetos), 2022. Consultado el 18-09-2025.

[4] Oregoom. Clase math en java. (https://oregoom.com/java/math/). Consultado el 18-09-2025.

[5] J. A. Solano. Clases y objetos. (https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/3068/mod$_resource/content/1/UAPA-Clases-Objetos/index.html), 2020. Consultado el 18-09-2025.