



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor(a): M.C. RENE ADRIAN DAVILA PEREZ

Asignatura: PROGRAMACIÓN ORIENTADA A OBJETOS

Grupo: 01

No de Práctica(s): 04

Integrante(s): 322118311

322094028

322092842

322078673

322067738

*No. de lista o
brigada:* 03

Semestre: 2026-1

Fecha de entrega: 19/09/2025

Observaciones:

CALIFICACIÓN: _____

Índice

1. Introducción	2
2. Marco Teórico	2
2.1. Distancia entre dos puntos:	2
2.2. GUI de java:	2
2.3. Biblioteca Swing:	2
2.3.1. JFrame:	2
2.3.2. JButton:	3
2.3.3. JOptionPane:	3
2.4. AWT en java:	3
2.5. Comunicación entre objetos:	3
3. Desarrollo	4
3.1. Archivo Punto.java	4
3.2. Archivo Calculo.java	4
3.3. Archivo Ventana.java	4
3.4. Archivo Practica4.java	4
4. Resultados	5
5. Conclusiones	6
6. Bibliografía	7

1. Introducción

- **Planteamiento del problema.** Buscamos implementar una aplicación en Java que permita calcular la distancia entre dos puntos, tomando estos argumentos desde el main, para posteriormente mostrar los resultados en una ventana emergente al presionar un botón.
- **Motivación.** Esta práctica nos permitiera reforzar los conceptos de Programación Orientada Objetos, así como el uso de interfaz gráfica y el manejo de eventos en Java.
- **Objetivos.** Desarrollar una aplicación que reciba coordenadas desde el main, genere un evento al presionar un botón y muestre en una ventana las coordenadas de los puntos y la distancia calculada entre ellos.

2. Marco Teórico

2.1. Distancia entre dos puntos:

Para calcular la distancia entre dos puntos A y B dentro de un plano cartesiano, trazamos una línea recta que una estos puntos, la cual será la hipotenusa de un triángulo rectángulo formado al trazar líneas paralelas a los ejes del plano (x,y) partiendo desde los puntos A y B, de esta analogía se deduce la siguiente fórmula, basada en el teorema de Pitágoras. [1]

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

2.2. GUI de Java:

La GUI (Graphical User Interface) es un conjunto de componentes gráficos (ventanas, botones, cajas de diálogo, cajas de texto, etc) los cuales permiten una mejor interacción entre el usuario y la aplicación. [2]

2.3. Biblioteca Swing:

Es un conjunto de clases que forman parte de la Java Foundation Classes (JFC) y es utilizada para crear interfaces gráficas de usuario (GUI) que nos ofrece botones, menús, cuadros de texto, tablas y paneles, para un mejor diseño en la aplicación a desarrollar. Entre los componentes que utilizaremos están:

2.3.1. JFrame:

Crea una ventana principal, contiene los siguientes métodos:

setTitle: Para definir el título que tendrá la barra de la ventana.

setSize: Para definir la dimensión de la ventana (ancho X alto).

setDefaultCloseOperation: Para establecer que pasara al cerrar la ventana.

setLocationRelativeTo: Centra la ventana respecto a un componente

add:. Para agregar un componente.

2.3.2. JButton:

Crea un boton que el usuario podra presionar, contiene el siguiente metodo:

addActionListener: Para registrar un “escuchador de eventos”.

2.3.3. JOptionPane:

Crea cuadros de dialogo, contiene el siguiente metodo:

showMessageDialog: Para mostrar un cuadro de dialogo informativo.

[3]

2.4. AWT en java:

La AWT (Abstract Window Toolkit) es un conjunto de clases que permiten que la aplicacion pueda manejar eventos en interfaces gráficas, es decir, acciones que realiza el usuario. Entre los componentes que utilizaremos estan:

ActionListener: Interfaz que permite .^{es}cuchar .^{ev}entos de accion (como presionar un boton.)

ActionEvent: Objeto que representa al evento de accion. [4]

2.5. Comunicación entre objetos:

Los objetos interactuan entre si enviandose mensajes unos a otros, despues de recibir un mensaje el objeto realizara una acción (enviar otro mensaje, realizar una tarea, etc). [5]

3. Desarrollo

3.1. Archivo Punto.java

Dentro de esta archivo estara la clase punto

Clase Punto : Dentro de esta clase empezaremos definiendo un objeto, que representara un punto en el plano cartesiano, que mediante un constructor por defecto inicializara el punto en (1,1). Tambien sobreescribiremos el metodo "toString "que permitira devolver un formato legible del punto.

Metodo punto: En este metodo definiremos un metodo constructor parametrizado que permitira crear un punto para cualquier coordenada

3.2. Archivo Calculo.java

Este archivo contendra la clase Calculo que a su vez tendra el metodo distancia.

Clase Calculo : En esta clase empezaremos definiendo dos valores (a y b) de tipo double, que se utilizaran en el metodo distancia

Metodo distancia: Este metodo recibira dos objetos de tipo punto como parametro, para despues calcular su distancia, haciendo uso de los atributos (x,y) ,aplicando la formula mencionada en el marco teorico, finalmente nos regresara un mensaje con las distancias entre los puntos.

3.3. Archivo Ventana.java

Este archivo contendra la clase ventana, la cual a su vez tendra el metodo ventana.

Clase Ventana : Esta clase, mediante JFrame, permitira crear una ventana grafica la cual tendra un boton y cuadros de dialogo, lo que facilitara la interaccion entre el usuario y el programa.

Metodo ventana: Este metodo recibira como parametros un objeto Calculo y dos objetos punto (los cuales seran usados en el metodo distancia). Posteriormente se inicializara la ventana configurando su dimension, titulo, operacion de cierre y la posición centrada, despues creamos un boton con el mensaje “Haz click aqui” y se le asignara un ActionListener que detectara cuando el usuario lo presione, cuando esto ocurra se llamara al metodo distancia pasando los parametros antes mencionados para finalmente mediante una ventana emergente realizada con JOptionPane se mostrara el resultado.

3.4. Archivo Practica4.java

Finalmente, en este archivo estara la clase Practica4 la cual tendra el metodo main.

Clase Practica4 : La clase Practica4 contendra el metodo main en el cual crearemos dos objetos de tipo punto que seran recibidos como argumentos desde la linea de comandos convirtiendo los de String a Int (mediante Integer.parseInt()), despues se

realizara la instancia de un obeito Calculo el cual se encarga de realizar el calculo de la distancia entre los puntos, posteriormente se creara un objeto ventana utilizando como parametros el controlador y dos puntos, para generar la interfaz grafica, Finalmente mediante setVisible(true) la ventana se hara visible permitiendo al usuario interactuar con ella.

4. Resultados

```
PS C:\Users\marti\distancia> javac *.java  
>  
PS C:\Users\marti\distancia> java Practica4 1 2 4 6
```

Figura 1: Entradas que tomara en el main

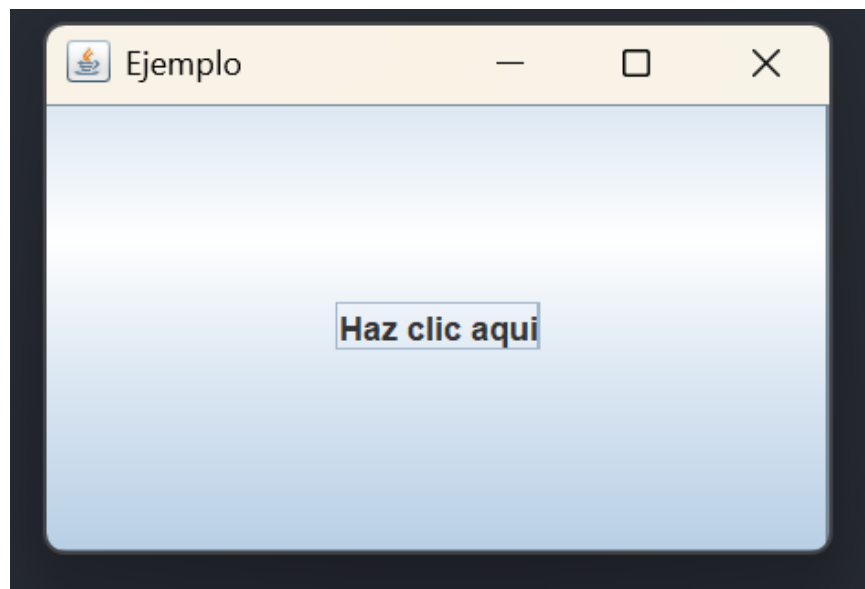


Figura 2: Ventana principal que contiene el boton.

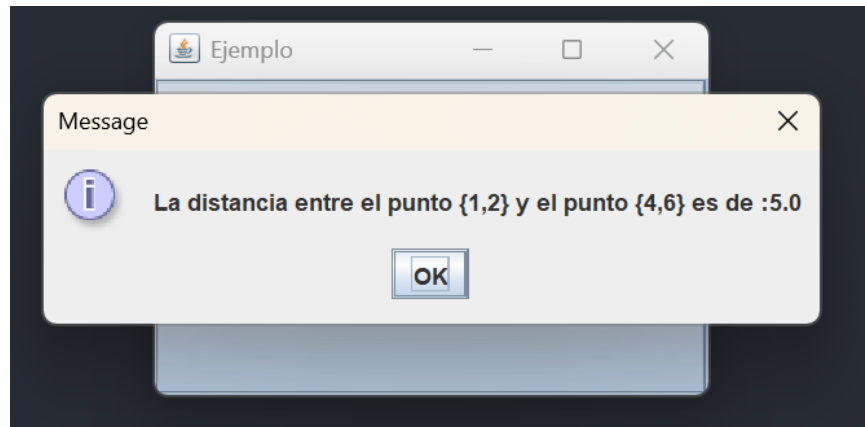


Figura 3: Mensaje que arroja despues de dar click al boton de la ventana principi

5. Conclusiones

En esta practica logramos aplicar conceptos de que es un constructor, clases, objetos y como interactuan entre si. Primero empezamos con las clases que de manera informal son como moldes o plantillas, que nos permiten abstraer objetos de la vida real y plasmarlos a codigo, aunque ya hemos trabajado con estas mismas, debido a que java es un lenguaje de programación orientado a objetos. En este caso creamos 4 clases:

- Calculo.java
- Practica4.java
- Ventana.java
- Punto.java

Los constructores (metodos que tienen el mismo nombre que la clase) cuya funcionalidad es inicializar los atributos de un nuevo objeto, nosotros inicializamos los constructores punto y ventana (en archivos java diferentes que posteriormente interactúan por medio de mensajes) donde ventana nos permite abrir una interfaz y punto pasar los valores a los cuales se les calculara la distancia.

Posteriormente logramos que cada objeto interactúe entre si y formen un solo programa por medio de mensajes que la sintaxis `objeto.metodo()`; esto permitió que los objetos se comuniquen, tal y como lo vemos en ventana: `controlador.distancia(a,b)`; Y como se observa al compilar nos abre una interfaz grafica (con sus tres botones del lado derecho superior) con la cual podemos darle click para que se muestren nuestros calculos realizados, todo esto es por que utilizamos las bibliotecas `import javax.swing.*;` `import java.awt.event.*;` que evidentemente nos permitieron realizar esta interfaz grafica.

6. Bibliografía

- [1] M. Hoyos, “Distancia entre dos puntos,” Curso para la UNAM, 20 de agosto de 2020. [En línea]. Disponible: <https://cursoparalaunam.com/distancia-entre-dos-puntos>. [Accedido: 19-sep-2025].
- [2] “Interfaz Gráfica GUI,” Programación Básica JAVA. [En línea]. Disponible: <https://sites.google.com/site/programacionbasicajava/interfaz-grafica-gui> .[Accedido: 19-sep-2025].
- [3] “Componentes Java Swing,” CodeJavu Blog, sept. 2013. [En línea]. Disponible: <https://codejavu.blogspot.com/2013/09/componentes-java-swing.html>. [Accedido: 19-sep-2025].
- [4] Oracle, “Class Component (Java Platform SE 8),” Oracle, 2015. [En línea]. Disponible: <https://docs.oracle.com/javase/8/docs/api/java/awt/Component.html> . [Accedido: 19-sep-2025].
- [5] C. [Apellido], “Java Básico 3.2,” Profesores FI-B UNAM. [En línea]. Disponible: http://profesores.fi-b.unam.mx/carlos/java/java_basico32.html. [Accedido : 19 – sep – 2025].