	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor(a): René Adrián Dávila Pérez

Asignatura: Programación Orientada a Objetos

Grupo: 7

No. de práctica(s): 3

Integrante(s): 320065570
425133840
423020252
322229916
425032224

No. de lista o brigada: 3

Semestre: 2026-1

Fecha de entrega: 27/08/2025

Observaciones:

CALIFICACIÓN: _____

Índice

1. Introducción

Uno de los aspectos esenciales de un programa es la interfaz que éste le proporciona al usuario para navegar y ejecutar la funcionalidad de dicho programa. Por esto, a la hora de construir programas, es necesario diseñar la interfaz con el usuario de tal manera que facilite la navegación y la interacción por la parte del usuario, con el fin de fomentar la inferencia de información acerca del programa. De esta manera, diseñar programas con interfaces gráficas, resulta ser la mejor elección cuando se trata de soluciones donde frecuentemente se requiere interacción con el usuario.

El problema que aborda esta práctica es la construcción de un programa que calcule la distancia entre dos puntos, utilizando una interfaz gráfica para la detonación por el usuario de esta funcionalidad y el despliegue de su salida.

La motivación para elaborar este programa radica en la necesidad de obtener el conocimiento de diseñar interfaces gráficas en el lenguaje de programación Java y así adquirir la capacidad de aprovechar sus beneficios.

Dado lo anterior, el objetivo de esta práctica es familiarizarnos con la biblioteca Swing y con los fundamentos de construcción de interfaces gráficas para desarrollar una base sobre la cual seremos capaces de cimentar futuras habilidades de diseño de interfaces interactivas.

2. Marco Teórico

El paquete Swing es parte de la JFC (Java Foundation Classes) en la plataforma Java, la cual provee recursos para construir GUIs. Al mismo tiempo, Swing es una extensión de AWT (Abstract Window Toolkit); abarca componentes como botones, tablas, marcos, etc..., las cuales pertenecen al paquete javax.swing.[1] Swing ofrece una funcionalidad mejorada respecto a AWT, nuevos componentes, funciones ampliadas para componentes y un excelente manejo de eventos con compatibilidad con la función de arrastrar y soltar.[2]

Desde una perspectiva teórica, las interfaces gráficas se diseñan con base en la

estrategia arquitectónica MVC (Modelo, Vista, Controlador), la cual postula una separación de los aspectos del código que abarcan el modelado de los datos de los aspectos que rigen el flujo de las operaciones a realizar con dichos datos.[3]

3. Desarrollo

Para dar solución al problema planteado, fueron creadas, además de la clase que contiene el método `main`, 3 clases auxiliares: **Mensajes**, **Ventana** y **Punto**.

La clase **Punto** describe las características de un punto en el plano cartesiano – tiene coordenadas horizontal y vertical. Además, describe el comportamiento de este tipo al ser convertido al tipo `String`. Para lograr esto, dentro de la clase en cuestión se definen 2 atributos (variables): `x`, `y` de tipo `float`, con un valor inicial de 0. También se define un constructor con 2 parámetros, de tipo `float`, cuyo valor se le asigna a los atributos dentro de este. Por último, se define un método (función) llamado `toString` que no acepta argumentos y regresa una cadena que muestra el valor de las coordenadas en el formato: (x, y).

La clase **Ventana** describe las características y el comportamiento de la interfaz gráfica mediante la cual el usuario detona la funcionalidad principal del programa. Se declaran 4 atributos: 1 de tipo `JButton`, otro de tipo `Mensajes` y 2 de tipo `Punto`. Se define un constructor de 3 parámetros: 1 de tipo `Mensajes` y 2 de tipo `Punto`, cuyos valores se les asignan a los atributos correspondientes; dentro de este se especifican las siguientes características: título es *Distancia entre 2 puntos*, dimensiones son 300 píxeles por 200 píxeles, la ventana no está ubicada en relación a ningún otro elemento y su operación de cierre, por omisión, es `JFrame.EXIT_ON_CLOSE`. Además, el atributo de tipo `JButton` se inicializa y se etiqueta *Haz clic aqui*. Luego, se plantea que al ser presionado el botón mencionado, este se ha de comunicar con el atributo de tipo `Mensaje`, pasando los atributos de tipo `Punto`, para obtener una salida, la cual se desplegará en una nueva ventana de diálogo.

Por último, la clase **Mensajes** está a cargo de llevar a cabo la funcionalidad principal del programa – calcular la distancia entre 2 puntos dados. Para esto, se

define el método `mensaje`, el cual sirve para realizar los cálculos necesarios y al mismo tiempo para la comunicación con otras clases. Tiene 2 parámetros de tipo `Punto` y regresa una cadena. Primero se comunica con la clase `Math` mediante su método `hypot`, pasando las coordenadas de los puntos recibidos, para calcular la distancia entre ellos. Luego regresa una cadena que muestra el resultado de la operación anterior con un formato específico.

Al inicio del programa, se crea un objeto de tipo `Mensajes`, luego los datos obtenidos como argumentos del programa se convierten al tipo `float` y son almacenados en un arreglo. Consiguientemente, se crean 2 objetos de tipo `Punto` y sus coordenadas se definen a partir de los datos mencionados anteriormente. Finalmente, se crea un objeto de tipo `Ventana` y se habilita su visibilidad. La salida del programa es una ventana titulada *Distancia entre 2 puntos*, con un botón que se extiende por todo el área de la ventana y tiene una etiqueta que dice *Haz clic aqui*. Al oprimir aquel botón, aparece un diálogo titulado *Message* que muestra las coordenadas de los 2 puntos proporcionados al programa, tal como un mensaje que indica la distancia entre aquellos puntos.

4. Resultados

```
[A31ZP@MM1A P4 % javac *.java
[A31ZP@MM1A P4 % java Practica4 4.44301 1.2305 -0.55611 5.2323
```

Figura 1: Ejecución del programa proporcionando las coordenadas de los puntos deseados

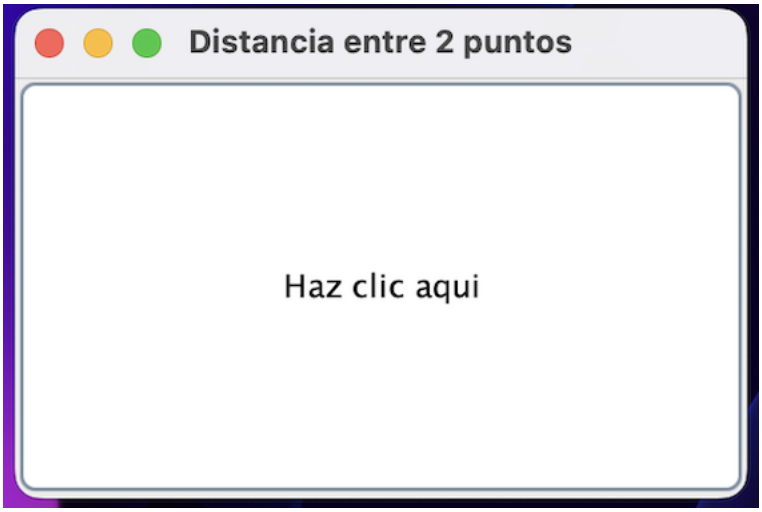

Figura 2: Ventana – elemento de interfaz gráfica correspondiente al objeto de tipo Ventana

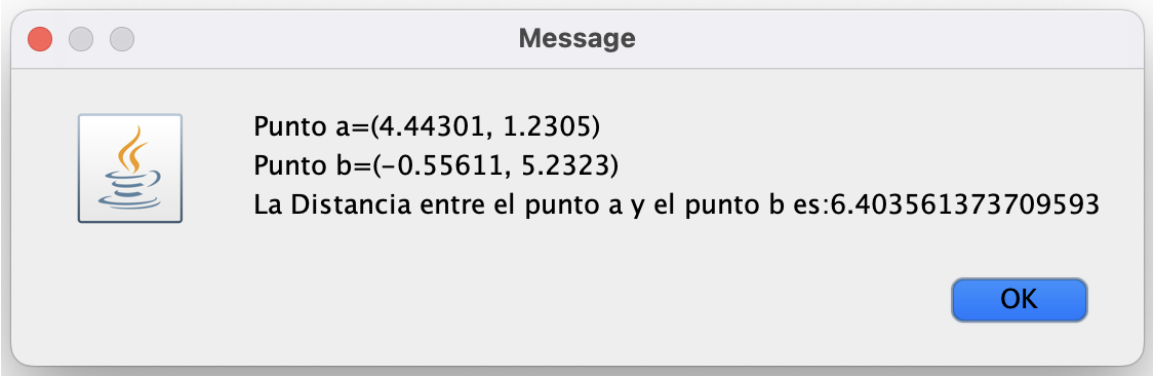

Figura 3: Diálogo – elemento de interfaz gráfica producido a partir de la interacción con el botón

5. Conclusiones

El desarrollo de esta aplicación demostró cómo el uso de Swing permite crear interfaces gráficas que mejoran la interacción entre el usuario y el programa. Al utilizar componentes como ventanas, botones y cuadros de diálogo, se logró que una operación matemática, como calcular la distancia entre dos puntos, se volviera visual y fácil de usar.

La separación del código en partes claras (como la lógica del cálculo y la interfaz) mostró la utilidad de organizar el trabajo para que cada parte cumpla una función específica, haciendo el código más comprensible y fácil de mantener.

El manejo de eventos, como el clic de un botón, permitió que la aplicación respondiera de manera ágil a las acciones del usuario, lo que confirma la eficacia de Swing para crear programas interactivos.

Referencias

[1] SWING. (s. f.). *Introducción a Java Swing*. Recuperado de `https://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/edavis/swing.html`

[2] GeeksforGeeks. (2025, 23 julio). *Introduction to Java Swing*. Recuperado de `https://www.geeksforgeeks.org/java/introduction-to-java-swing/`

[3] MDN. (s. f.). *MVC - Glosario de MDN Web Docs*. Recuperado de `https://developer.mozilla.org/es/docs/Glossary/MVC`