



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): 7,8

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	2
2. Marco Teórico	3
3. Desarrollo	4
4. Resultados	5
5. Conclusiones	7
6. Referencias	8

1. Introducción

En esta práctica, se trabajó sobre un código proporcionado, al cual se le realizaron las modificaciones adecuadas para la implementación de herencia y polimorfismo; procurando su correcto funcionamiento en todo momento.

Se vuelve necesario la resolución de estos problemas ya que, la herencia y el polimorfismo son la base del diseño de software escalable y sostenible, además, es la única forma de ver lo que se permite y lo que no se a la hora de aplicar estos conceptos. De esta práctica se espera aprender en qué casos nos conviene implementarlos y como promover jerarquías de clases que faciliten el trabajo para la reutilización de código, aumentando así también la eficiencia de este.

2. Marco Teórico

Para la resolución del problema, se utilizó principalmente herencia y polimorfismo. La herencia nos permite extensión de funcionalidades con el objetivo de reutilizar código y usar polimorfismo, se identificó cuál sería la clase padre y las clases derivadas, es decir, quienes heredarían métodos o atributos de la súper clase, en este caso la clase Figura. Otro concepto usado fue una clase abstracta; dentro de ella no se pueden crear objetos, puede tener atributos y métodos, y por lo menos un método abstracto que es el método no definido dentro de una clase para implementarlo en sus clases hijas. Para implementar los métodos abstractos se utilizó la sobre-escritura de métodos, la cual nos permite redefinir el comportamiento de un método. Por último, se usó súper para referirse a la clase padre inmediata de una clase, permitiendo a una subclase llamar a constructores, métodos y variables de su superclase. Su propósito principal es heredar y reutilizar la funcionalidad de la clase padre o evitar la ambigüedad cuando hay métodos o campos con el mismo nombre en ambas clases.

El polimorfismo consiste en conseguir que un objeto de una clase se comporte como un objeto de cualquiera de sus subclases. Se puede aplicar tanto a métodos como a tipos de datos. Los métodos pueden evaluar y ser aplicados a diferentes tipos de datos de manera indistinta. Los tipos polimórficos son tipos de datos que contienen al menos un elemento, cuyo tipo no está especificado. El polimorfismo se puede clasificar en dos grandes grupos:

- Polimorfismo dinámico o paramétrico: es aquel en el que no se especifica el tipo de datos sobre el que se trabaja y, por ende, se puede recibir y utilizar todo tipo de datos compatible. Este tipo de polimorfismo también se conoce como programación genérica.
- Polimorfismo estático o *ad hoc*: es aquel en el que los tipos de datos que se pueden utilizar deben ser especificados de manera explícita antes de ser utilizados. [1]

3. Desarrollo

La clase `Circulo` tiene los atributos privados **radio** y **PI**, la clase `Rectangulo` **alto** y **ancho**, y la clase `TrianguloRectangulo` **altura** y **base**.

Para todas las clases se utilizó la palabra reservada `extends` con el fin de establecer la herencia por parte de estas clases de la clase padre `Figura`, se implementaron métodos de acceso para sus atributos (en la clase `Circulo`, sólo para el atributo **radio**) y se definió un constructor con parámetros de tipo `double` con el fin de habilitar la construcción de objetos de estos tipos a partir de un valor específico para sus atributos. También se implementaron los métodos abstractos heredados de la clase base, definiendo las fórmulas para calcular el área y perímetro de las figuras geométricas representadas por estas clases.

Por último, se empacó el programa con la siguiente estructura de carpetas:

```
mx/
├── unam/
│   ├── fi/
│   │   ├── poo/
│   │   │   └── p78/
│   │   │       ├── Figura.java
│   │   │       ├── Circulo.java
│   │   │       ├── Rectangulo.java
│   │   │       ├── TrianguloRectangulo.java
│   │   │       ├── MainApp.java
│   │   │       ├── PanelDibujo.java
│   │   │       └── NumericTextField.java
```

La clase `MainApp` proporcionada por el profesor, se construye y ejecuta una interfaz gráfica que permite al usuario seleccionar una de las 3 figuras geométricas abarcadas, introducir valores para los atributos definidos en la clase correspondientes a dicha figura y mostrar la figura con las características introducidas, tal como los valores calculados de su área y perímetro.

4. Resultados

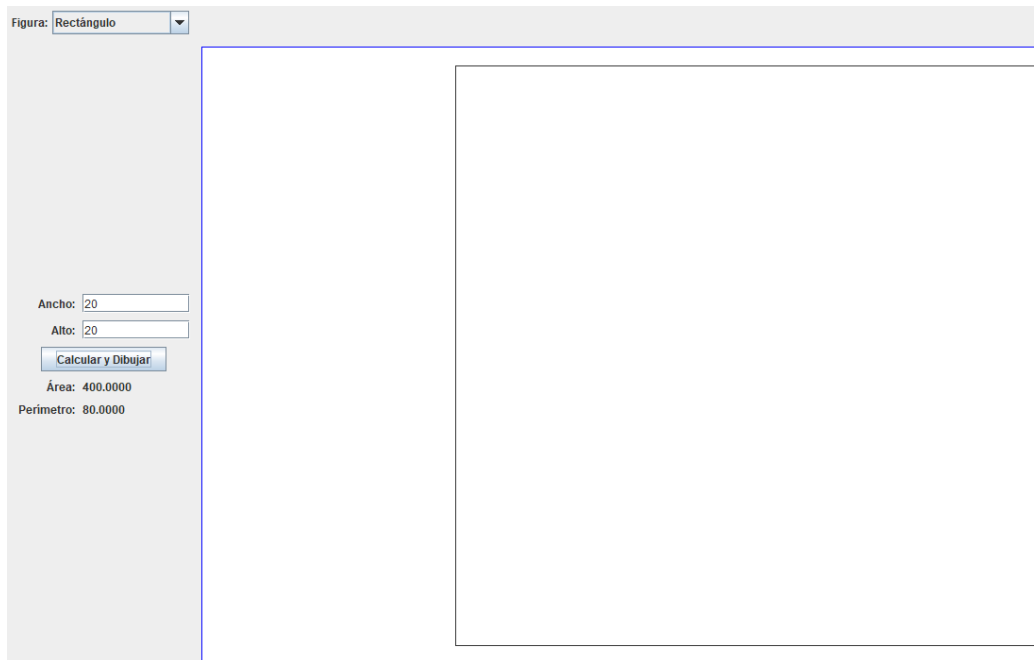


Figura 1: Área y perímetro de rectángulo

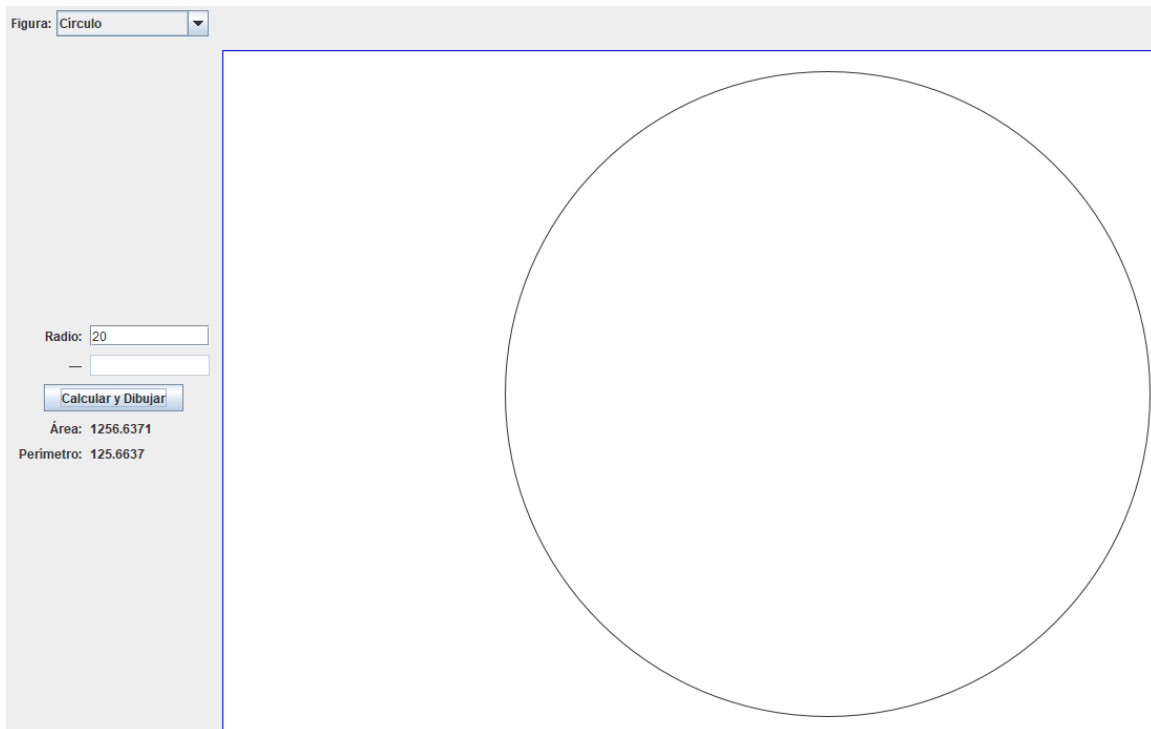


Figura 2: Área y perímetro de círculo

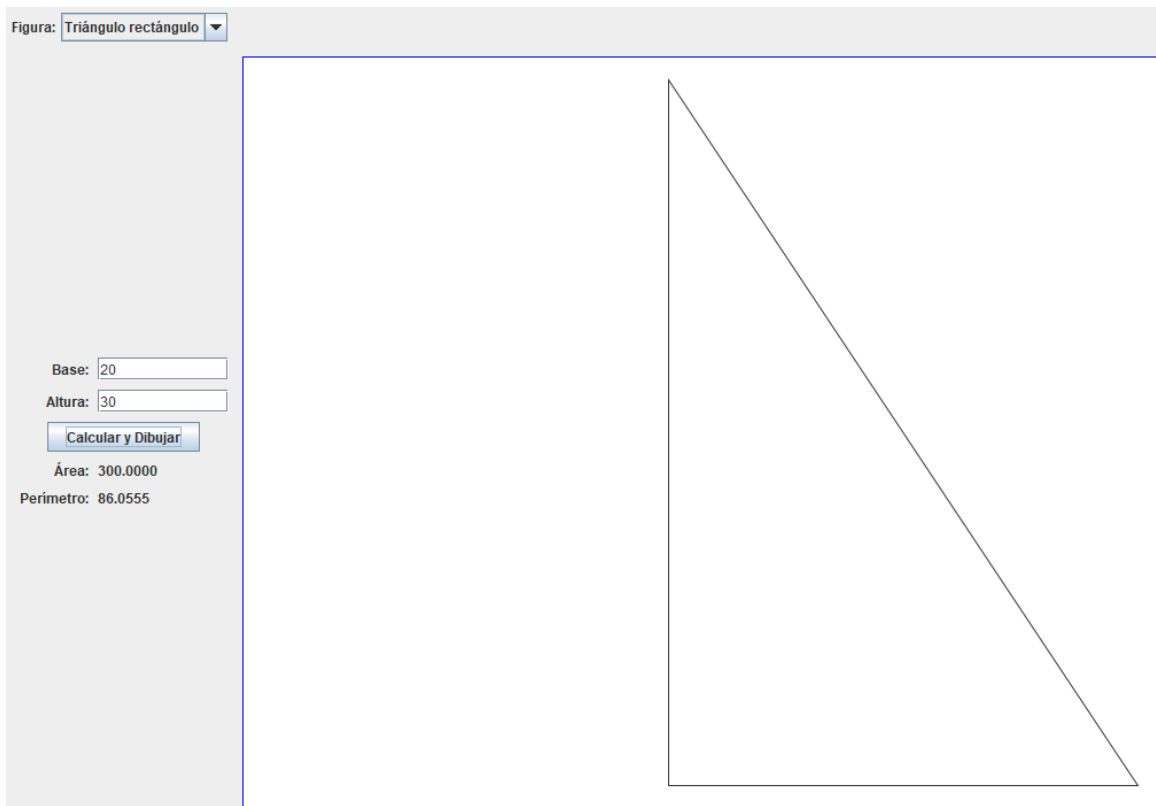


Figura 3: Área y perímetro de triángulo rectángulo

5. Conclusiones

El desarrollo de este trabajo permitió aplicar de manera práctica los principios fundamentales de la Programación Orientada a Objetos, principalmente la herencia, el polimorfismo y el uso de clases abstractas. A través de la clase base **Figura** y sus subclases **Circulo**, **Rectangulo** y **TrianguloRectangulo**, se logró diseñar una estructura jerárquica que favorece la reutilización de código y la organización del proyecto. La declaración de métodos abstractos y su posterior sobrescritura en las clases derivadas permitió establecer comportamientos específicos para cada figura, mientras que el uso de la palabra reservada **super** facilitó la correcta interacción con la clase padre y la reutilización de constructores. En conjunto, este proyecto refleja una comprensión sólida de los conceptos teóricos aplicados y una correcta estructura de empaquetado, evidenciando la importancia de aplicar buenas prácticas de programación orientada a objetos.

6. Referencias

[1] UNADM, "Polimorfismo", Recursos educativos UAPA, 2023. [En línea]. Disponible: https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/3062/mod_resource/content/1/Polimorfismo/index.html