UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

LICENCIATURA EN CIBERSEGURIDAD

PROGRAMACIÓN I

INVESTIGACIÓN I

ASESOR ACADÉMICO

NAPOLEON IBARRA

PRESENTADO POR

FÉLIX CABALLERO

MIÉRCOLES 20 DE AGOSTO DE 2025

TABLA DE CONTENIDO

1.3 Construcción de clase (JAVA)

1.3.1 Miembros de una clase

1.3.2 Modificadores de acceso

1.3.3 Otras opciones

¿QUÉ ES LA CONSTRUCCIÓN DE UNA CLASE?

Una clase en Java es una plantilla que permite crear objetos. Define atributos y métodos

Permite organizar el código de forma modular y reutilizable
Base de la Programación Orientada a Objetos
Facilita el mantenimiento del software

DE UNA CLASE

VENTAJAS

- Permiten ordenar datos y comportamientos
- Facilitan la reutilización de código
- Mejoran la comprensión del sistema

DESVENTAJAS

- Si hay muchos miembros, la clase se vuelve difícil de mantener
- Riesgo de exponer información si no se usa encapsulamiento
- Errores si se abusa de métodos o atributos públicos

EJEMPLO

```java
public class Animal {
    String especie;
    int edad;

    void hacerSonido() {
        System.out.println("El animal hace un sonido.");
    }

}
```

¿QUÉ SON LOS MIEMBROS DE UNA CLASE?

Son los componentes internos que forman una clase:

- Atributos
- Métodos
- Constructores

IMPORTANCIA DE LOS MIEMBROS

Permiten definir el comportamiento del objeto

Organizan los datos internos

Dan estructura clara a la clase

MIEMBROS DE UNA CLASE

VENTAJAS

- Permiten ordenar datos y comportamientos
- Facilitan la reutilización de código
- Mejoran la comprensión del sistema

DESVENTAJAS

- Si hay muchos miembros, la clase se vuelve difícil de mantener
- Riesgo de exponer información si no se usa encapsulamiento
- Errores si se abusa de métodos o atributos públicos

EJEMPLO

```java
public class Libro {
    String titulo;
    int paginas;

    Libro(String t, int p) {
        titulo = t;
        paginas = p;
    }

    void leer() {
        System.out.println("Leyendo el libro " + titulo);
    }
}
```

MODIFICADORES DE ACCESO

Los modificadores de acceso controlan la visibilidad de los miembros de una clase:

- public
- private
- protected
- default

IMPORTANCIA DE MODIFICADORES

Protegen datos sensibles

Permiten aplicar encapsulamiento

Evitan accesos no deseados desde otras clases

MODIFICADORES

VENTAJAS

- Otorgan seguridad y control
- Permiten ocultar detalles internos
- Facilitan mantenimiento de datos

DESVENTAJAS

- Complican pruebas rápidas o debugging
- Un mal uso puede limitar acceso útil
- Añade más reglas al diseño

EJEMPLO

```java
public class Empleado {
    private double salario;

    public void setSalario(double s) {
        salario = s;
    }

    public double getSalario() {
        return salario;
    }
}
```

OTRAS OPCIONES

Otras opciones en una clase son las palabras clave como *static final*

Mejoran flexibilidad y rendimiento
Permiten compartir métodos comunes
Refuerzan buenas prácticas en POO

CLASES STATIC FINAL

VENTAJAS

- Optimiza recursos
- Mejora legibilidad del código
- Promueve buenas prácticas y organización

DESVENTAJAS

- Mal uso provoca errores difíciles de detectar
- Puede hacer el código más complejo
- Sobrecarga excesiva confunde al leer el código

EJEMPLO

```java
public class Matematicas {
    static final double PI = 3.1416;

    static int multiplicar(int a, int b) {
        return a * b;
    }
}
```

CONCLUSIÓN

La construcción de clases es la base estructural de la programación en Java y permite modelar objetos reales dentro del código. Dominar sus componentes, como los miembros y los modificadores de acceso, posibilita un desarrollo más seguro, organizado y escalable. Comprender estas herramientas es fundamental para aplicar correctamente los principios de la programación orientada a objetos y garantizar la calidad del software.

BIBLIOGRAFÍA