



# Algoritmos y Estructura de Datos

## EJERCICIOS



DOCENTE: MARX DANLY LEÓN TRUJILLO  
FACULTAD DE ING. INDUSTRIAL Y SISTEMAS  
E.P. INGENIERÍA DE SISTEMAS

## EJERCICIOS DE ESTRUCTURAS Y CLASES

---

### **Ejercicio 1: Clase Rectángulo**

Crea una clase llamada Rectángulo que tenga atributos para la longitud y el ancho. Implementa métodos para calcular el área y el perímetro del rectángulo. Crea objetos Rectángulo y muestra sus áreas y perímetros.

### **Ejercicio 2: Clase Estudiante**

Crea una clase Estudiante con atributos como nombre, edad y carné. Crea métodos para obtener y establecer estos atributos. Luego, crea objetos Estudiante y muestra sus detalles.

### **Ejercicio 3: Clase Libro**

Crea una clase Libro con atributos como título, autor y año de publicación. Implementa métodos para mostrar la información del libro. Crea objetos Libro y muestra su información.



## EJERCICIOS DE ESTRUCTURAS Y CLASES

---

### **Ejercicio 4: Clase Calculadora**

Crea una clase Calculadora que tenga métodos para realizar operaciones matemáticas básicas como suma, resta, multiplicación y división. Puedes pasar los números como parámetros a estos métodos y obtener el resultado.

### **Ejercicio 5: Clase Banco**

Crea una clase Banco que tenga una lista de cuentas bancarias. Cada cuenta bancaria debe ser representada como un objeto de una clase CuentaBancaria que tiene atributos como número de cuenta, saldo, etc. La clase Banco debe tener métodos para agregar cuentas, eliminar cuentas y consultar el saldo de una cuenta.



## EJERCICIOS DE ESTRUCTURAS Y CLASES

---

### **Ejercicio 6: Herencia**

Crea una clase base llamada Vehiculo con atributos como marca, modelo y año. Luego, crea clases derivadas como Coche, Camión y Motocicleta que hereden de Vehiculo. Cada una de las clases derivadas debe tener atributos y métodos específicos.

### **Ejercicio 7: Clase Producto**

Crea una clase Producto con atributos como nombre, precio y cantidad en stock. Implementa métodos para actualizar el precio y la cantidad en stock, y para mostrar la información del producto.



# OPERACIONES DE ARCHIVOS

---

## **Lectura de Archivos**

Para leer un archivo en Java, puedes utilizar clases como `FileInputStream` o `BufferedReader`.

## **Escritura de Archivos**

Para escribir en un archivo, puedes usar clases como `FileOutputStream` o `BufferedWriter`.

En la siguiente clase encontramos los métodos para leer y escribir en un archivo.



# OPERACIONES DE ARCHIVOS

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class ManejoDeArchivos {

    // Método para leer un archivo de texto
    public static String leerArchivo(String nombreArchivo) {
        StringBuilder contenido = new StringBuilder();
        try (BufferedReader br = new BufferedReader(new FileReader(nombreArchivo))) {
            String linea;
            while ((linea = br.readLine()) != null) {
                contenido.append(linea).append("\n");
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return contenido.toString();
    }

    // Método para escribir en un archivo de texto
    public static void escribirArchivo(String nombreArchivo, String contenido) {
        try (BufferedWriter bw = new BufferedWriter(new FileWriter(nombreArchivo))) {
            bw.write(contenido);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



# OPERACIONES DE ARCHIVOS

```
// Método para escribir en un archivo de texto
public static void escribirArchivo(String nombreArchivo, String contenido) {
    try (BufferedWriter bw = new BufferedWriter(new FileWriter(nombreArchivo))) {
        bw.write(contenido);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    // Ejemplo de uso
    String nombreArchivo = "mi_archivo.txt";

    // Escribir en el archivo
    String textoParaEscribir = "Hola, este es un ejemplo2 de escritura en archivo.";
    escribirArchivo(nombreArchivo, textoParaEscribir);

    // Leer el archivo y mostrar su contenido
    String contenidoLeido = leerArchivo(nombreArchivo);
    System.out.println("Contenido del archivo:");
    System.out.println(contenidoLeido);
}
```

