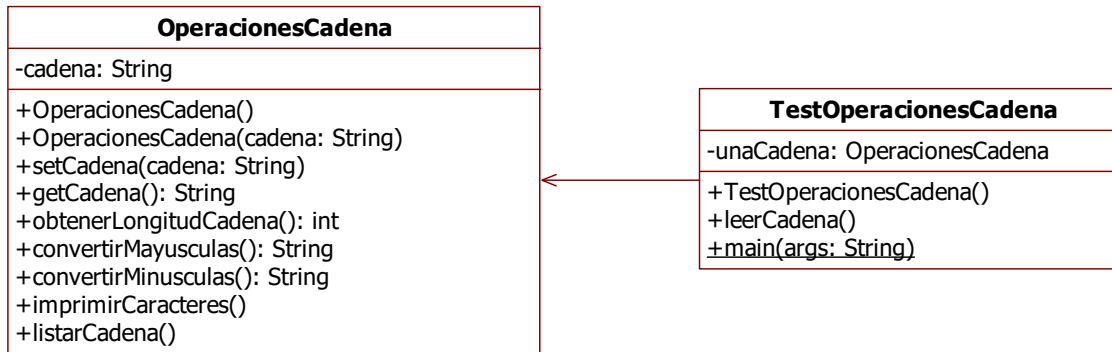


# CADENAS

### Ejercicios.

**Actividad 1.** Digitar y compilar el siguiente código en Java, que permite realizar operaciones de una cadena como medir el tamaño de la cadena, convertir a mayúsculas y minúsculas.

El Siguiente diagrama de clases define unas operaciones básicas de *Strings*



```
public class OperacionesCadena {
    /**
     * atributo de la clases
     */
    private String cadena;

    /**
     * constructor que inicializa la cadena con un espacio en blanco
     */
    public OperacionesCadena( ) {
        this.cadena = "";
    }

    /**
     * Constructor que permite crear cadenas
     * @param cadena
     */
    public OperacionesCadena(String cadena) {
        this.cadena = cadena;
    }

    /**
     * Método que actualiza el contenido de la cadena
     * @param cadena
     */
    public void setCadena(String cadena) {
        this.cadena = cadena;
    }

    /**
     * Método que retorna el valor de la cadena
     * @return cadena
     */
}
```

## Taller 02: Teoría de la computación

---

```
public String getCadena( ) {
    return cadena;
}
/**
 * Método que retorna el tamaño o longitud de la cadena
 * @return tamaño cadena
 */
public int obtenerLongitudCadena( ){
    return cadena.length();
}
/**
 * Método que convierte y retorna la cadena en mayúsculas
 * @return cadena en mayúsculas
 */
public String convertirMayusculas( ) {
    return cadena.toUpperCase();
}

/**
 * Método que convierte y retorna la cadena en minúsculas
 * @return cadena en minúsculas
 */

public String convertirMinusculas( ) {
    return cadena.toLowerCase();
}
/**
 * Método que imprime los caracteres que conforma la cadena
 */

public void imprimirCaracteresCadena( ){
    char c=' ';
    System.out.println("\nImpresion de la Cadena por caracteres");
    for(int i=0;i<cadena.length();i++){
        c=cadena.charAt(i);
        System.out.println(i+" "+c);
    }
    System.out.println(" ");
}
}
```

## Taller 02: Teoría de la computación

---

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
import java.util.Scanner;

public class TestOperacionesCadena {
    private final OperacionesCadena unaCadena;

    public TestOperacionesCadena() {
        this.unaCadena= new OperacionesCadena();
    }

    public void leerCadena(String cadena){
        unaCadena.setCadena(cadena);
        System.out.println( "\nLa Longitud de la cadena: " +
                            unaCadena.obtenerLongitudCadena()+" caracteres ");
        System.out.println( "\nLa cadena texto en mayusculas: " +
                            unaCadena.convertirMayusculas());
        System.out.println( "La cadena texto en minusculas: " +
                            unaCadena.convertirMinusculas());
        unaCadena.imprimirCaracteresCadena();
    }

    public static void main(String[] args) {
        TestOperacionesCadena prueba = new TestOperacionesCadena();
        String cadena="";
        //se lee la cadeena por teclado
        Scanner sc = new Scanner(System.in);
        System.out.print("Digite Cadena ");
        cadena= sc.nextLine();
        prueba.leerCadena(cadena);
    }
}
```

## Taller 02: Teoría de la computación

---

**Actividad 2.** Digitar y compilar el siguiente código en Java, que permite capturar 2 cadenas y define cual es la mayor, y cuando son iguales o desiguales.

```
public class ProcesoCadena {

    public char obtenerCaracter(String cad1, int posicion){
        char caracter=cad1.charAt(posicion);
        return caracter;
    }
    public String obtenerCompareTo(String cadena1, String cadena2){
        int res=cadena1.compareTo(cadena2);
        String respuesta="";
        if (res==0){
            respuesta="Las cadenas son iguales";
        }else
            if (res>0){
                respuesta="La cadena mayor es "+cadena1;
            }else
                respuesta="La cadena mayor es "+ cadena2;
        return respuesta;
    }

    public boolean obtenerComparacion(String cadena1, String cadena2){
        if (cadena1.equals(cadena2)){
            return true;
        }
        else{
            return false;
        }
    }
}
```

## Taller 02: Teoría de la computación

---

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
import java.util.Scanner;

public class TestProcesoCadena {
    public static void main(String[] args) {
        ProcesoCadena app = new ProcesoCadena();
        String cad1 = "";
        String cad2 = "";
        Scanner sc = new Scanner(System.in);
        System.out.print("Digite la Cadena1: ");
        cad1= sc.nextLine();

        System.out.print("Digite la Cadena2: ");
        cad2= sc.nextLine();
        System.out.println(app.obtenerCompareTo(cad1,cad2));

        if(!app.obtenerComparacion(cad1,cad2)) {
            System.out.println("Las cadenas son desiguales. ");
        }
    }
}
```

## Taller 02: Teoría de la computación

---

### Ejercicios propuestos

1. Construir una aplicación que imprima la cadena invertida  
Entrada        salida  
Computador    rodaturpmoC  
Datos         sotaD
2. Implementar una aplicación que permita contar el número de vocales de una cadena.
3. Construir el método que, a partir de una cadena, determine si al leerla de izquierda a derecha o derecha a izquierda significa lo mismo (oro, oso, 1111).