

Caso Palíndromos

[1] Contexto del problema

```
function esPalindromo(cadena) {  
    let resultado = "";  
    resultado = cadena.split("").reverse().join("");  
    return cadena === resultado;  
}
```

[1.1]

¿Qué hace el método?

El método verifica si una cadena es un palíndromo.

¿Cómo lo hace?

Toma una cadena de texto como parámetro, y en una variable llamada “resultado” toma esta cadena y la invierte, finalmente verifica si la cadena de texto de resultado es igual a la de la cadena inicial.

¿Cómo lo uso?

Para usar este método, debemos pasarle una cadena de texto como parámetro inicial para que trabaje, luego el método trabajará para mostrar si la cadena es o no es un palíndromo.

Ejemplo

Si le pasamos al método la palabra “oso” este devolverá “true”, ya que esta palabra es un palíndromo, en cambio, si le pasamos la palabra “amor” devolverá “false” ya que esta palabra no es palíndroma.

[2] Entender el método JavaScript

[2.1] Explicación detallada del método

El método se llama “**esPalindromo**”, el cual toma como parámetro una cadena de texto. Dentro del método se crea una variable llamada “**resultado**”, esta variable contendrá la cadena inicial para dividir las en caracteres con “**reverse**”, luego con “**join**” se vuelve a convertir en una cadena de texto, finalmente el método compara la cadena inicial y la cadena que contiene la variable “**resultado**”, y retorna falso o verdadero dependiendo de si es o no es un palíndromo.

[2.2] Versión Java

```
public class esPalindromo {  
    public static boolean esPalindromo(String cadena) {  
  
        // Clase StringBuilder para manipular cadenas de caracteres  
        StringBuilder resultado = new StringBuilder(cadena);  
  
        StringBuilder cadenaInvertida = resultado.reverse();  
  
        String cadenaInvertidaStr = cadenaInvertida.toString();  
  
        boolean esPalindromo = cadena.equals(cadenaInvertidaStr);  
  
        return esPalindromo;  
    }  
}
```

[3] Ok! Si el método funciona ¿Qué puede malir sal? ;-)

[3.1]

Casos de prueba

- Cadenas con mayusculas y minusculas
- Cadenas de números
- Cadenas con espacio
- Cadenas combinadas con las anteriores

[3.2]

Cadenas con mayúsculas y minúsculas:

Input: " Ana"

Output esperado: true

Cadenas de números:

Input: " 54345"

Output esperado: true

Cadenas con espacios:

Input: "buenos días"

Output esperado: false

Cadenas combinadas con las anteriores:

Input: "123 hola"

Output esperado: false

[3. 3]

```
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*;
```

```
public class PalindromoTest {

    @Test
    public void testCadenasConMayusculasYMinusculas() {
        assertTrue(Palindromo.esPalindromo(" Ana"));
    }

    @Test
    public void testCadenasDeNumeros() {
        assertTrue(Palindromo.esPalindromo(" 54345"));
    }

    @Test
    public void testCadenasConEspacios() {
        assertFalse(Palindromo.esPalindromo("buenos días"));
    }

    @Test
    public void testCadenasCombinadas() {
        assertFalse(Palindromo.esPalindromo("123 hola"));
    }
}
```

[3. 4] ¿Qué resultados arrojan sus Test con estas entradas: “aca”, “acas”, “h”?

“aca” arroja verdadero al ser un palíndromo, esta pasa toda las pruebas

“acas” arroja falso al no ser un palíndromo

“h” arroja falso al no ser un palíndromo

[4.1]

Input: 200 (número entero)

Output: ????

Arroja falso, ya que los números no pueden ser palíndromos

Input: "" (cadena vacía)

Output: ????

Arroja verdadero, ya que este caso se encuentra en las pruebas

Input: "aaabccbaaa"

Output: ????

Arroja verdadero, ya que es un palíndromo

Input: "ahabccbaaa"

Output: ????

Arroja falso, ya que NO es un palíndromo

Input: "La tele letal"

Output: ????

Arroja verdadero, ya que es un palíndromo y pasa todas las pruebas del programa

[4.2]

```
public class Palindromo {
    public static boolean esPalindromo(String cadena) {

        // Atributos
        String cadenaEspacios = cadena.replaceAll("\\s+", "").toLowerCase();

        int longitud = cadenaEspacios.length();

        for (int i = 0; i < longitud / 2; i++) {
            if (cadenaEspacios.charAt(i) != cadenaEspacios.charAt(longitud - i -
1)) {
                return false;
            }
        }
        return true;
    }
}
```

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;
```

```
public class PalindromoTest {

    @Test
    public void testCadenaVacía() {
        assertTrue(Palindromo.esPalindromo(""));
    }

    @Test
    public void testPalindromoSimple() {
```

```
        assertTrue(Palindromo.esPalindromo("oso"));
    }

    @Test
    public void testNoPalindromo() {
        assertFalse(Palindromo.esPalindromo("hola"));
    }

    @Test
    public void testPalindromoNumerico() {
        assertFalse(Palindromo.esPalindromo("12321"));
    }

    @Test
    public void testNumeroNoEsPalindromo() {
        assertFalse(Palindromo.esPalindromo("200"));
    }

    @Test
    public void testCadenaAlfanumerica() {
        assertTrue(Palindromo.esPalindromo("aaabccbaaa"));
        assertFalse(Palindromo.esPalindromo("ahabccbaaa"));
    }

    @Test
    public void testCadenaConEspacios() {
        assertTrue(Palindromo.esPalindromo("La tele letal"));
    }
}
```

[4.3]

¿Qué consideraciones tomaron en cuenta?

Prevenir errores como los espacio, mayusculas y minusculas, y como funciona todo con los números

¿Qué mejoró en su método?

El método al prevenir errores, hace que sea más eficiente y confiable, es decir, mejora el criterio del método para determinar si una cadena de texto es o no es Palíndromo

¿Qué rol jugaron las pruebas en mejorar su código?

Mostraron todos los fallos que podía presentar el programa, y que casos nuevos introducir

[5] Conclusión

Obtuvimos un programa con el suficiente criterio para poder determinar si una cadena de texto es o no es Palíndroma, este proyecto incluyó el usar de Java como lenguaje de programación, y JUnit 5 para las pruebas unitarias, estas últimas ayudaron a determinar aquellos casos en los cuales el programa podría verse sobrepasado, permitiéndonos crear excepciones y adaptar el código a la solución de estas