

Jonathan Chavez

Cristobal Ramos

Joaquin Arriagada

Esteban Aguilera

Paso 1: Leer y entender el contexto problema. (15 mins.)

Claro que tenemos un pequeño problema, pues el programador a cargo olvidó dejar la versión Java de la solución y sólo tenemos el siguiente código Javascript:

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

1.1 Discutir y concluir:

¿Qué hace el método?

R. Lo que hace el método es verificar si cierta cadena de texto es palindromo o no, al dividir la cadena, luego invierte el orden de esta y lo junta todo otra vez para guardarla en la variable resultado. Luego verifica si la cadena original es la misma a la invertida

¿Cómo lo hace?

R. El método toma el parámetro recibido "cadena" el cual es un string, a este lo separa en caracteres, los invierte y luego los junta de vuelta, a este string invertido lo guarda en la variable resultado, finalmente con return se devuelve el valor de la comparación de igualdad entre ambas cadenas (original e invertida)

¿Cómo lo uso?

R. El método se utiliza llamando y añadiendo como parámetro una cadena de texto.

Dé al menos un ejemplo de su uso (sin codificar nada).

R. esPalindromo("anilina"), donde separara esta palabra (a,n,i,l,i,n,a), luego la invierte (a,n,i,l,i,n,a) y comprueba si es igual a la original, donde se ve que efectivamente es un palíndromo.

1.2 Discutir resultados entre el grupo y con el profesor.

Paso 2: Entender el método JavaScript (15 mins.)

2.1 Tras una discusión individual, cada grupo deberá explicar que hace el método detalladamente.

2.2 Construya en grupo ahora una versión Java que sea 100% equivalente en funcionalidad (lo bueno y lo malo) al anterior método.

2.3 Discutir resultados entre el grupo y con el profesor.

Paso 3: Ok! Si el método funciona ¿Qué puede malir sal? ;-) (15 mins.)

R. Lo que podría salir mal en un método funciona seria, si se ingresa mayúscula y minúsculas, si se ingresan caracteres no alfanuméricos o si no se ingresa nada.

3.1 Discutir en grupo el diseño de un plan de pruebas para este caso.

3.2 A partir de su plan de pruebas, diseñe los casos de pruebas unitarias a implementar (aún no codifique nada!!!), considere al menos 5.

testClassicSentenceWithPunctuationAndSpaces\_returnsTrue()

stringOneCharacterRemoved()

mixedCaseAndPunctuation\_returnsTrue()

emptyString\_returnsTrue()

singleCharacter\_returnsTrue()

3.3 Estando seguros que sus casos de pruebas unitarias son amplios y relevantes, ahora impleméntelos en Java usando JUnit.

3.4 ¿Qué resultados arrojan sus Test con estas entradas: "aca", "acas", "h"?

R. Los resultados son los siguientes: True,False,True.

Paso 4: **Mejorando el método**, probando más. (15 mins.)

4.1 Considere los siguientes casos de prueba, no codifique NADA, discuta en su grupo.

- Caso de prueba 1:

Input: 200 (número entero)

Output: ERROR int cannot be converted to String

- Caso de prueba 2:

Input: "" (cadena vacía)

Output: true

- Caso de prueba 3:

Input: "aaabccbaaa"

Output: true

- Caso de prueba 4:

Input: "ahabccbaaa"

Output: false

- Caso de prueba 5:

Input: "La tele letal"

Output: true

4.2 De las pruebas analizadas, concluya y construya una versión mejorada de su método. Construya además nuevas pruebas unitarias considerando los casos anteriores y verifique sus resultados teóricos con los empíricos.

¿Qué consideraciones tomaron en cuenta?

R. Tomamos consideraciones previas a verificar el string, como que no esté vacío, luego que se cambie el string a un mejor método para verificar, eliminando mayúsculas, espacio y caracteres no alfanuméricos.

¿Qué mejoró en su método?

Nuestro método mejoró la validación al realizar tratamientos extra al string permitiendo que devuelva correctamente si es palindromo a distintos formatos que antes no se validaban correctamente.

¿Qué rol jugaron las pruebas en mejorar su código?

Las pruebas nos dieron a conocer las falencias que había en nuestro código y donde éste podía fallar causando la detención de su funcionamiento, con los mensajes que nos devuelve la consola del test podemos identificar dónde se encuentran los problemas para así solucionarlos.

Paso 5: Discutir experiencia y resultados con TODO el curso y Concluir!!! (15 mins.)

Link Repositorio Compartido GitHub

[https://github.com/JoacoWn/palindromos\\_2.git](https://github.com/JoacoWn/palindromos_2.git)