

1.-

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

- a) El metodo “esPalindromo(cadena)” verifica si una cadena ingresada es palindromo o no.
b) Entrada: recibe una cadena de texto como parámetro.

Proceso:

- Utiliza el metodo split(“”)para dividir la cadena en un array de caracteres únicos.
- Utiliza reverse() para invertir el array
- Utiliza join para unir el array invertido en una nueva cadena.
- Por último compara la cadena inicial con la invertida.

Salida: devuelve True si es igual a la invertida o false en caso contrario.

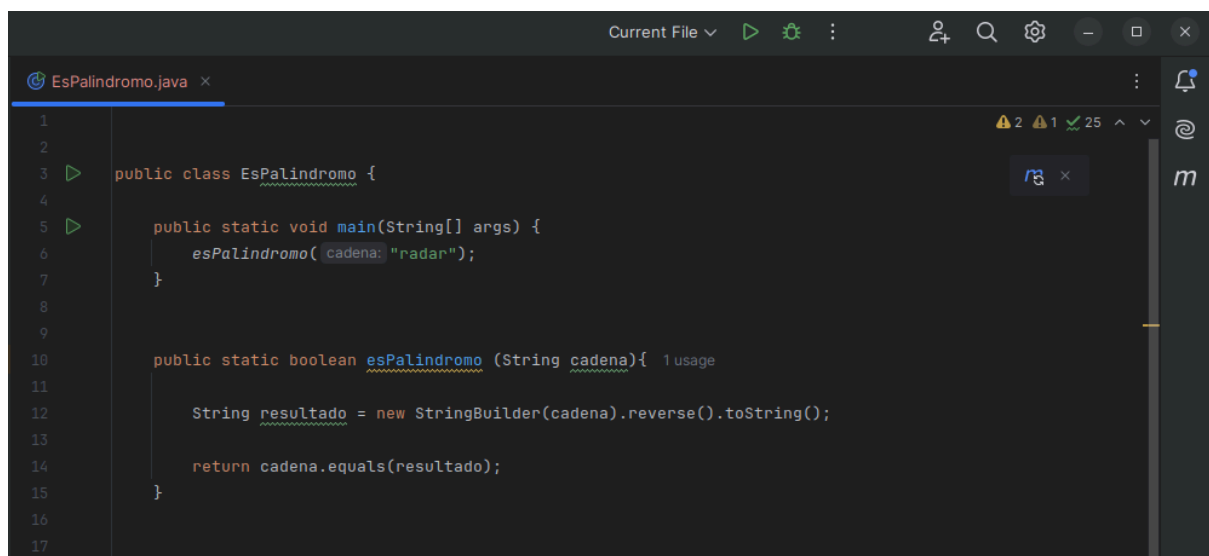
- c) El método se usa llamando a la función y escribiendo cultura cadena de texto como parámetro de esta.

Ej:

```
esPalindromo(“radar”);
```

Esta cadena debería devolver True ya que “radar” es un palíndromo.

2.-



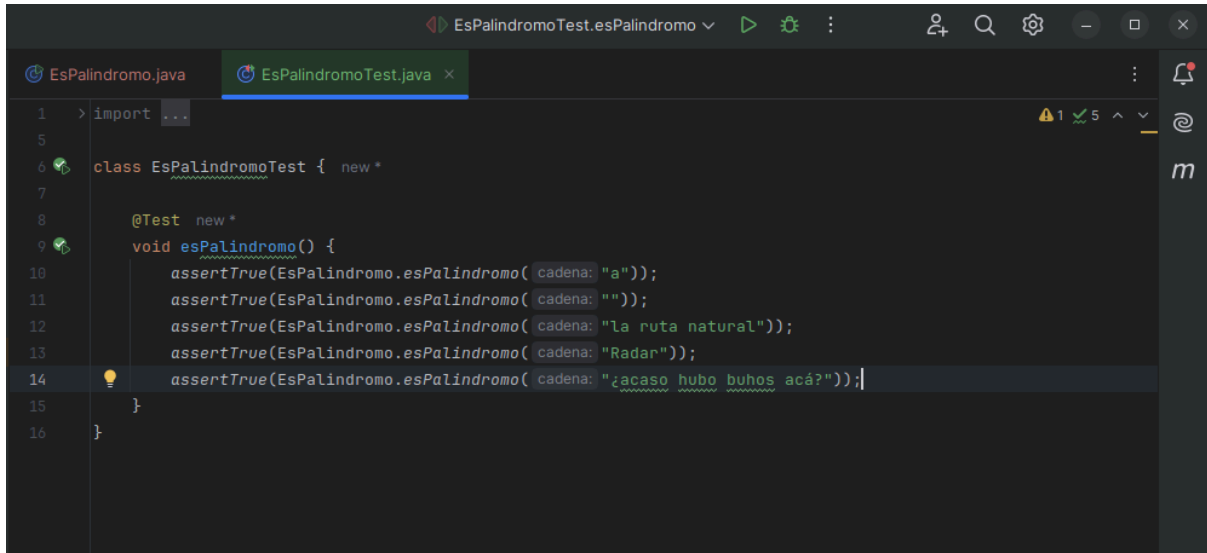
The screenshot shows an IDE window titled 'EsPalindromo.java'. The code is as follows:

```
1  
2  
3 public class EsPalindromo {  
4  
5     public static void main(String[] args) {  
6         esPalindromo( cadena: "radar");  
7     }  
8  
9  
10    public static boolean esPalindromo (String cadena){ 1 usage  
11  
12        String resultado = new StringBuilder(cadena).reverse().toString();  
13  
14        return cadena.equals(resultado);  
15    }  
16  
17
```

3.-

Casos de pruebas unitarias:

- casos para caracteres de 1 dígito ("a", "b", "c", etc)
- casos para cadenas vacías
- casos para cadenas con espacios
- casos para cadenas con mayúsculas y minúsculas.
- casos para cadenas con caracteres especiales ("é", "á", etc..)

A screenshot of an IDE window showing a Java file named 'EsPalindromoTest.java'. The code defines a class 'EsPalindromoTest' with a single test method 'esPalindromo()'. Inside the method, there are five assertions using 'assertTrue()' to test the 'esPalindromo()' method of the 'EsPalindromo' class with various inputs: 'a', an empty string, 'la ruta natural', 'Radar', and '¿acaso hubo buhos acá?'. The IDE interface includes a top toolbar with icons for running, debugging, and searching, and a sidebar on the right with a search icon and a file explorer icon.

```
1 >import ...
5
6 class EsPalindromoTest { new *
7
8     @Test new *
9     void esPalindromo() {
10         assertTrue(EsPalindromo.esPalindromo( cadena: "a"));
11         assertTrue(EsPalindromo.esPalindromo( cadena: ""));
12         assertTrue(EsPalindromo.esPalindromo( cadena: "la ruta natural"));
13         assertTrue(EsPalindromo.esPalindromo( cadena: "Radar"));
14         assertTrue(EsPalindromo.esPalindromo( cadena: "¿acaso hubo buhos acá?"));
15     }
16 }
```

Con las entradas "aca" y "h" el código arroja True, con "acas" arroja False.

4.-

- Caso de prueba 1:

Input: 200

Output: error ya que el código acepta solo Strings

- Caso de prueba 2:

Input: "" (cadena vacía)

Output: probablemente true si lee una cadena vacía como un dato

- Caso de prueba 3:

Input: "aaabccbaaa"

Output: True ya que esa cadena se lee igual al derecho que al revés.

- Caso de prueba 4:

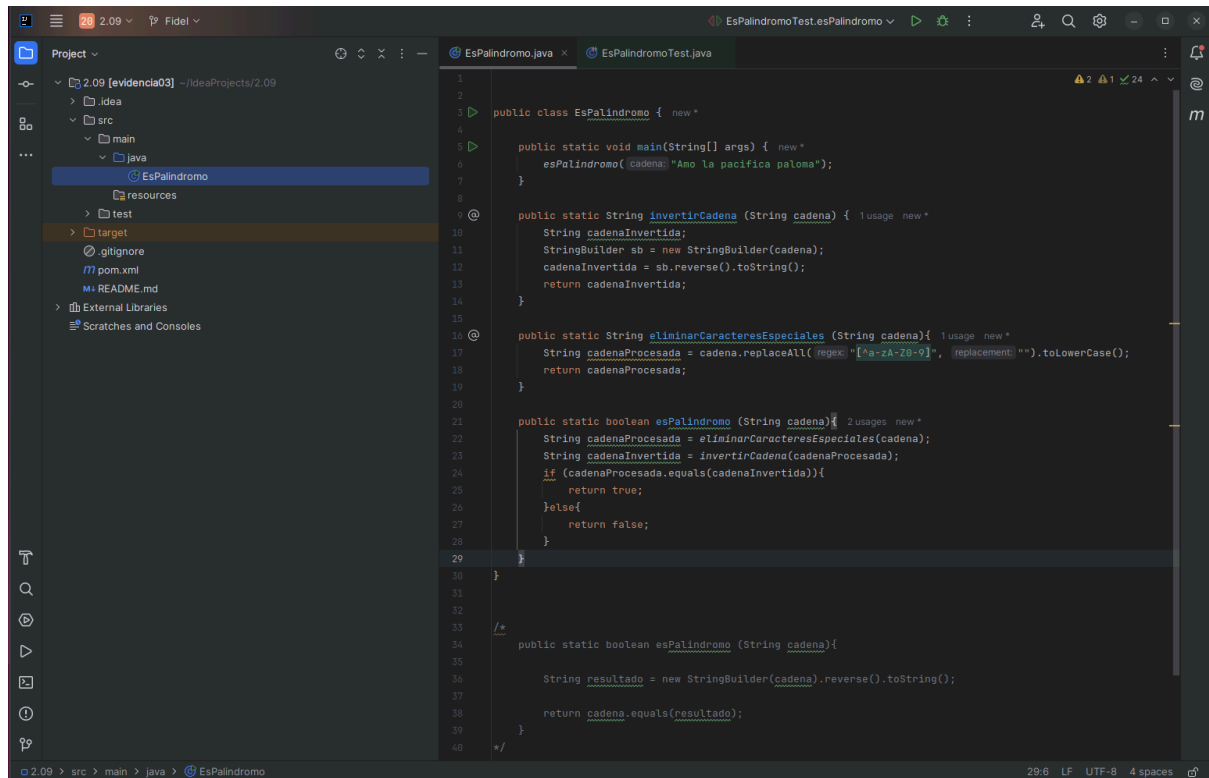
Input: "ahabccbaaa"

Output: Probablemente falso ya que no se lee igual al derecho que al revés.

- Caso de prueba 5:

Input: "La tele letal"

Output: Probablemente falso si los espacios los cuenta como un dato del array.



```
1
2
3 public class EsPalindromo { new *
4
5     public static void main(String[] args) { new *
6         esPalindromo( "Amo la pacifica paloma");
7     }
8
9     public static String invertirCadena (String cadena) { 1 usage new *
10         String cadenaInvertida;
11         StringBuilder sb = new StringBuilder(cadena);
12         cadenaInvertida = sb.reverse().toString();
13         return cadenaInvertida;
14     }
15
16     public static String eliminarCaracteresEspeciales (String cadena){ 1 usage new *
17         String cadenaProcesada = cadena.replaceAll( regex: "[*a-zA-Z0-9]", replacement: "" ).toLowerCase();
18         return cadenaProcesada;
19     }
20
21     public static boolean esPalindromo (String cadena){ 2 usages new *
22         String cadenaProcesada = eliminarCaracteresEspeciales(cadena);
23         String cadenaInvertida = invertirCadena(cadenaProcesada);
24         if (cadenaProcesada.equals(cadenaInvertida)){
25             return true;
26         }else{
27             return false;
28         }
29     }
30 }
31
32
33 /*
34     public static boolean esPalindromo (String cadena){
35
36         String resultado = new StringBuilder(cadena).reverse().toString();
37
38         return cadena.equals(resultado);
39     }
40 */
```

En el código consideramos algunas de los posibles problemas que pueden ocurrir a la hora de verificar la igualdad de las cadenas, en este caso, gracias a la los casos de pruebas anteriores, y a nuestras conclusiones obtenidas de las pruebas del ítem 3, pudimos obtener una idea de lo que se debía de considerar para arreglar y verificar el correcto funcionamiento de el código, así agregamos algunas funcionalidades para evitar problemáticas con espacios, Mayúsculas, y caracteres especiales, de esta forma el código cubre gran parte de las problemáticas que podrían ocurrir a la hora de verificar si una cadena es Palindromo o no.