

EJERCICIOS TDD

Entornos de desarrollo



Metodología de desarrollo donde se escriben las pruebas antes que el código. implica tres pasos:

1. Escribir una prueba que falle.
2. Escribir el código mínimo necesario para que pase la prueba.
3. Refactorizar el código.

Fomenta la creación de código más limpio, confiable y orientado a los requisitos, al tiempo que se acelera la detección temprana de errores.

Antonio Salces Alcaraz (1º DAM)
C.P.I.F.P. Alan Turing
03/12/2024

Índice

1.	<u>STRING INVERTIDO</u>	1
2.	<u>PALABRA PALÍNDROMA</u>	2
3.	<u>NUMERO DE CONSONANTES DE UNA PALABRA</u>	3
4.	<u>PALABRA MAYÚSCULA</u>	4

1. String invertido

Tenemos que crear una función a la que le pasemos un String, y está nos devuelva el contenido invertido. Prueba:

```
@Test
public void reverseWordV1(){
    //Arrange
    String palabra = "amigo";
    String resultadoEsperado = "ogima";

    //Act
    String resultadoReal = StringUtils.reverseWord(palabra);

    //Assert
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}

@Test
public void reverseWordV2(){
    //Arrange
    String palabra = "anticonstitucionalmente";
    String resultadoEsperado = "etnemlanoicutitsnocitna";

    //Act
    String resultadoReal = StringUtils.reverseWord(palabra);

    //Assert
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}
```

Código para superar la prueba:

```
public static String reverseWord(String palabra){
    String palabraInvertida = "";
    for (int i = palabra.length() - 1; i >= 0; i--) {
        palabraInvertida = palabraInvertida + palabra.charAt(i);
    }
    return palabraInvertida;
}
```

2. Palabra palíndroma

Tenemos que crear una función a la que le pasamos un String y nos diga si la palabra es palíndroma o no. Prueba:

```
@Test
public void palabraPalindromaV1(){
    //Arrange
    String palabra = "amigo";
    boolean resultadoEsperado = false;

    //Act
    boolean resultadoReal = StringUtils.palíndromo(palabra);

    //Assert
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}

@Test
public void palabraPalindromaV2(){
    //Arrange
    String palabra = "reconocer";
    boolean resultadoEsperado = true;

    //Act
    boolean resultadoReal = StringUtils.palíndromo(palabra);

    //Assert
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}
```

Código para superar la prueba:

```
public static boolean palíndromo(String palabra){
    boolean esPalindromo = false;
    String palabraInvertida = "";
    for (int i = palabra.length() - 1; i >= 0; i--) {
        palabraInvertida = palabraInvertida + palabra.charAt(i);
    }

    if (palabra.equals(palabraInvertida)){
        esPalindromo = true;
    }
    return esPalindromo;
}
```

3. Numero de consonantes de una palabra

Tenemos que crear una función a la que le pasemos una palabra y nos diga el número de consonantes que tiene. Prueba:

```
@Test
public void numConsonantesV1(){
    //Arrange
    String palabra = "amigo";
    int resultadoEsperado = 2;

    //Act
    int resultadoReal = StringUtils.numConsonantes(palabra);

    //Assert
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}

@Test
public void numConsonantesV2(){
    //Arrange
    String palabra = "transgredir";
    int resultadoEsperado = 8;

    //Act
    int resultadoReal = StringUtils.numConsonantes(palabra);

    //Assert
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}
```

Código para superar la prueba:

```
public static int numConsonantes(String palabra){
    int contConsonantes = 0;
    char letra = ' ';
    for (int i = 0; i <= palabra.length()-1; i++){
        letra = palabra.toLowerCase().charAt(i);
        if (letra != 'a' && letra != 'e' && letra != 'i' && letra !=
'o' && letra != 'u'){
            contConsonantes++;
        }
    }
    return contConsonantes;
}
```

4. Palabra mayúscula

Tenemos que crear una función que le pasamos una palabra y nos diga si está escrita en mayúscula o no. Prueba:

```
@Test
public void palabraMayusculaV1(){
    //Arrange
    String palabra = "HOLA";
    boolean resultadoEsperado = true;

    //Act
    boolean resultadoReal = StringUtils.palabraMayuscula(palabra);

    //Assert
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}

@Test
public void palabraMayusculaV2(){
    //Arrange
    String palabra = "hola";
    boolean resultadoEsperado = false;

    //Act
    boolean resultadoReal = StringUtils.palabraMayuscula(palabra);
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}

@Test
public void palabraMayusculaV3(){
    String palabra = "HoLA";
    boolean resultadoEsperado = false;

    boolean resultadoReal = StringUtils.palabraMayuscula(palabra);

    Assert.assertEquals(resultadoEsperado, resultadoReal);
}
```

Código para superar la prueba:

```
public static boolean palabraMayuscula(String palabra){
    if (palabra.toUpperCase() == palabra) {
        return true;
    }else{
        return false;
    }
}
```