

Entorno de Desarrollo



Ejercicio 1

Utilizando el enfoque del TDD, desarrolla una función que devuelva un String en el orden invertido de letras.

Ejemplo:

Con la entrada: **amigo**

Debería devolver: **ogima**

```
import org.junit.Assert;
import org.junit.Test;

public class StringUtilsTest {
    @Test
    public void reverse_word_amigo() {
        String palabra = "amigo";
        String resultadoEsperado = "ogima";

        String resultadoReal = StringUtils.reverseWord(palabra);
        Assert.assertEquals(resultadoEsperado, resultadoReal);
    }
}
```

Al principio hacemos el Test. Y ya después hacemos el código para que funcione:

```
public static String reverseWord(String palabra){
    String reverseString = "";
    // crear la firma. La firma es todo menos lo que hay entre las llaves.
    for(int i = palabra.length() - 1; i >= 0; i--){
        reverseString = reverseString + palabra.charAt(i);
    }

    return reverseString;
}
```

Ejercicio 2

Utilizando el enfoque del TDD, desarrolla una función que te diga de una palabra es un palíndromo o no. Ejemplos:

Con la entrada: amigo -> Debería devolver: falso

Con la entrada: reconocer -> Debería devolver: true

Aquí hacemos los test mínimos para demostrar que funcione el encontrar si es palindromo o no:

```
@Test
public void espalindromocaso1() {
    String palabra = "amigo";
    Boolean resultadoEsperado=false;
    Boolean resultadoReal = StringUtils.palindromo(palabra);
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}

@Test
public void espalindromocaso2() {
    String palabra = "reconocer";
    Boolean resultadoEsperado=true;
    Boolean resultadoReal = StringUtils.palindromo(palabra);
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}
```

ahora hacemos el código:

```
public static Boolean palindromo(String palabra){
    Boolean escierto= false;
    String reverseString = "";
    // crear la firma. La firma es todo menos lo que hay entre las llaves.
    for(int i = palabra.length() -1; i>=0; i--){
        reverseString = reverseString + palabra.charAt(i);
    }
    if(reverseString.equals(palabra)){
        escierto=true;
    }else{
        escierto=false;
    }
    return escierto;
}
```

Ejercicio 3

Desarrollé una función que cuente el número de consonantes que hay en una palabra.
Con la entrada: amigo -> Debería devolver: 2

```
@Test
public void constantetest() {
    String palabra = "amigo";
    int resultadoEsperado=2;
    int resultadoReal = StringUtils.consonanteWord(palabra);
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}
```

Al principio hacemos el Test. Y ya después hacemos el código para que funcione:

```
public static int consonanteWord(String palabra){
    int contador = 0;
    // crear la firma. La firma es todo menos lo que hay entre las llaves.
    for(int i = palabra.length() -1; i>=0; i--){
        switch (palabra.charAt(i)) {
            case 'a','e','i','o','u':
                break;
            default:
                contador=contador+1;
                break;
        }
    }
    return contador;
}
```

Ejercicio 4

Desarrolla una función que te diga si una palabra tiene solamente mayúsculas.

con la entrada: HOLA -> debería devolver: true

con la entrada: Hola -> debería devolver: false

```
@Test
public void estodomayusculacaso1() {
    String palabra = "HOLA";
    boolean resultadoEsperado=true;
    boolean resultadoReal = StringUtils.solomayusculas(palabra);
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}
public void estodomayusculacaso2() {
    String palabra = "Hpla";
    boolean resultadoEsperado=false;
    boolean resultadoReal = StringUtils.solomayusculas(palabra);
    Assert.assertEquals(resultadoEsperado, resultadoReal);
}
```

Al principio hacemos el Test. Y ya después hacemos el código para que funcione:

```
public static boolean solomayusculas(String palabra){
    Boolean respuesta =false;
    if(palabra.equals(palabra.toUpperCase())){
        respuesta=true;
    }
    // crear la firma. La firma es todo menos lo que hay entre las llaves.
    return respuesta;
}
```

La pruebas de los test que salen correctos:

