
Ejercicios 3 Cadenas

- [Ejercicio 1](#)
 - [Ejercicio 2](#)
 - [Ejercicio 3](#)
 - [Ejercicio 4](#)
 - [Ejercicio 5](#)
 - [Ejercicio 6](#)
 - [Ejercicio 7](#)
 - [Ejercicio 8](#)
 - [Ejercicio 9](#)
 - [Ejercicio 10](#)
 - [Ejercicio 11](#)
 - [Ejercicio 12](#)
 - [Ejercicio 13](#)
 - [Ejercicio 14](#)
 - [Ejercicio 15](#)
-

Ejercicio 1

Realiza una función que reciba una cadena y devuelva el número de vocales que tiene.

Ejercicio 2

Realiza una función que reciba una cadena y devuelva otra cadena igual pero sin espacios en blanco. No se pueden utilizar métodos *replace*.

Ejercicio 3

Realiza una función que reciba una cadena y un carácter y devuelva otra cadena igual pero con todas las apariciones del carácter reemplazadas por un asterisco. No se pueden utilizar métodos *replace*.

Ejercicio 4

Realiza una función que reciba una cadena y un carácter y devuelva el número de apariciones del carácter en la cadena. No se pueden utilizar los métodos *indexOf* ni *contains*.

Ejercicio 5

Realiza una función que reciba una cadena y un carácter y devuelva el número de caracteres que hay entre la primera y última aparición del carácter en la cadena. No se pueden utilizar los métodos *indexOf* ni *contains*.

Ejercicio 6

Realiza una función que reciba dos cadenas y devuelva la concatenación de ambas. El método *concat* o el operador `+` solamente se pueden utilizar a nivel de carácter.

Ejercicio 7

Realiza una función que reciba una cadena y un número y devuelva otra cadena con los primeros caracteres de la cadena, tantos como indique el parámetro. No se puede utilizar el método *substring*. Por ejemplo, si recibe "Programación" y 5, devuelve "Progr".

Ejercicio 8

Realiza una función que reciba una cadena y un número y devuelva otra cadena con los últimos caracteres de la cadena, tantos como indique el parámetro. No se puede utilizar el método *substring*. Por ejemplo, si recibe "Programación" y 5, devuelve "ación".

Ejercicio 9

Realiza una función que reciba una cadena y devuelva otra cadena con los caracteres en orden inverso. Por ejemplo, si recibe "hola", devuelve "aloh".

Ejercicio 10

Realiza una función que reciba una cadena y devuelva la suma de todos los dígitos que hay en ella. Por ejemplo, si recibe "abc12de3f4gh", devuelve 10.

Ejercicio 11

Realiza una función que reciba una cadena y devuelva el número de palabras que tiene. Hacer las comparaciones a nivel de carácter. No se pueden utilizar métodos *replace*, *indexOf* ni *trim*. Las palabras dentro de la cadena están separadas por un espacio. Por ejemplo, si recibe "Estoy matriculado en Programación en Entornos y en Marcas", devuelve 9. Si recibe " Estoy matriculado en Programación en Entornos y en Marcas ", es decir, con un espacio al principio y otro al final, devuelve 9.

Ejercicio 12

Realiza una función que reciba una cadena y devuelva el número de palabras que tiene. Hacer las comparaciones a nivel de carácter. No se pueden utilizar métodos *replace* ni *indexOf*. Las palabras dentro de la cadena pueden estar separadas por más de un espacio. No se pueden eliminar los espacios de la cadena. Por ejemplo, si recibe " Estoy matriculado en Programación en Entornos y en Marcas ", devuelve 9.

Ejercicio 13

Realiza una función que reciba dos cadenas y devuelva el número de apariciones de una de ellas en la otra. No se pueden utilizar los métodos *indexOf*, *contains* ni *substring*. Hacer las comparaciones a nivel de carácter. Por ejemplo, si recibe "Estoy matriculado en Programación, en Entornos y en Marcas" y "en", devuelve 3. Si recibe "hola que tal, hola que pasa, hola" y "hola", devuelve 3.

Ejercicio 14

Realiza una función recursiva que reciba una cadena y devuelva otra cadena con los caracteres en orden inverso. Por ejemplo, si recibe "hola", devuelve "aloh".

Ejercicio 15

Realiza una función recursiva que reciba una cadena de caracteres e indique si es o no palíndroma, es decir, si se lee igual de izquierda a derecha que de derecha a izquierda, sin tener en cuenta espacios ni mayúsculas. Ejemplo: "Dabale arroz a la zorra el abad" es palíndroma.