

## ¿Cuál es la idea básica de este método de similitud?

La idea básica de este método consiste en el número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra. Cuando hablamos de “transformar” hablamos de una operación, bien una inserción, eliminación o la sustitución de un carácter

## ¿Cómo utilizaremos esta distancia de edición (edit distance, como también se le conoce a la distancia de Levenshtein) para recuperar del AVL las n palabras más similares)?

Vamos llamando a nuestra función de LevenshteinDistance que toma dos cadenas, str1 de longitud lenStr1, y str2 de longitud lenStr2, y calcula la distancia Levenshtein entre ellas, si la distancia obtenida es menor que la esperada por nuestra parte para que exista similitud entre las palabras almacenaremos dicha palabra en un AVL que finalmente recorreremos y contendrá las palabras más similares.

## ¿Qué pasaría si el diccionario lo implementásemos mediante un BSTree, en lugar de con un AVLTree?

El BStree es una estructura de árbol que no mantiene ningún mecanismo de equilibrado. Cuando leemos desde un fichero nuestro diccionario si fuese almacenado en un BSTree el árbol resultante quedaría desequilibrado por lo que influiría en las búsquedas de palabras similares.