

---

# EDA I

---

## Práctica 02 Ejercicio 3

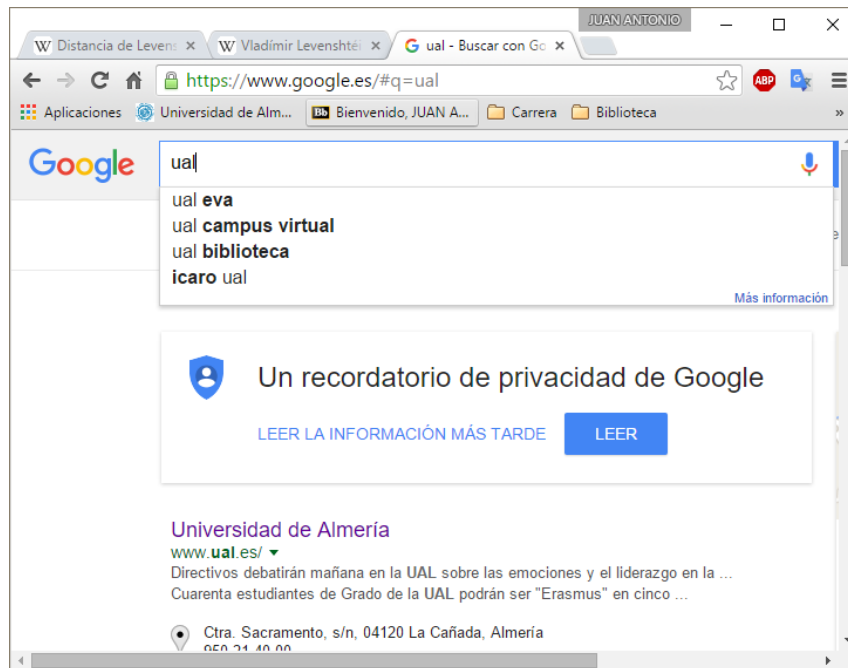
---

Juan Antonio Rodríguez Baeza

---

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

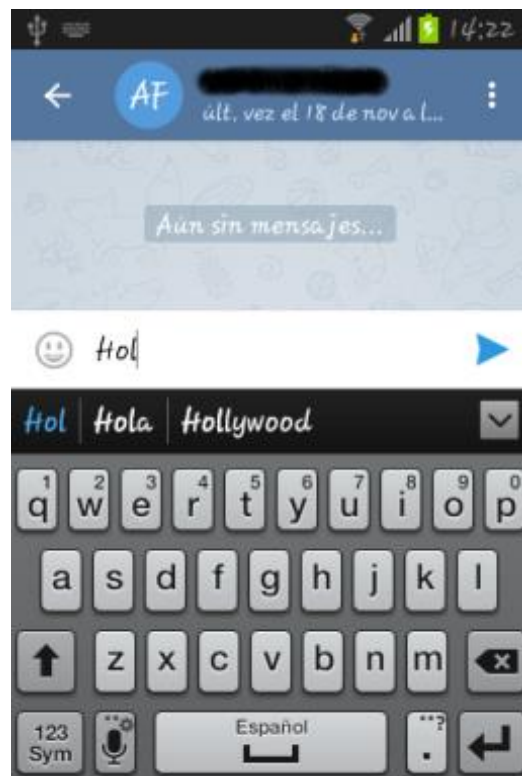
El algoritmo de la Distancia de Levenshtein ([Vladimir Levenshtein](#)), trata de ayudarnos al escribir, ofreciéndonos posibilidades que se acercan a la palabra que estemos escribiendo en ese momento, hoy en día es uno de los algoritmos mas utilizados a nivel mundial, en múltiples variantes. Google es mantiene una patente sobre su propia modificación del algoritmo que podemos consultar en el siguiente enlace <http://www.google.com/patents/US7664343>, cuando hacemos usamos el motor de búsqueda:



El sistema nos ofrece una serie de posibles búsquedas relacionadas con la palabra que hemos introducido.

Otro ejemplo es la aplicación de mensajería móvil Telegram.

En la captura podemos ver como al escribir una serie de caracteres la aplicación nos ofrece las opciones que se acercan.



- ¿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?

Para adaptar el algoritmo a nuestras necesidades lo hemos implementado de forma que cada palabra que le llega la analizara comparándola sobre un diccionario de palabras extrayendo de este las n palabras que más se le acercan al aplicarle el algoritmo a sendos Strings (palabra analizada y palabra comparada).

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

Si implementásemos un BSTree al cargar el archivo para reellenar la estructura “fallaría” completamente dicha implementación, es decir, como estamos cargando un fichero de texto de palabras ordenadas alfabéticamente, cada palabra será la “hija izquierda” de la anterior, degenerando una lista enlazada o en un ArrayList.