## Parcial Lenguajes de Programación 2020-II: Punto #5

Manual de Usuario

Jhon Sebastian Rojas Rodriguez

La aplicación interactúa con el usuario mediante la consola. Para hacer uso de la aplicación se debe tener instalado el lenguaje Go en el equipo.

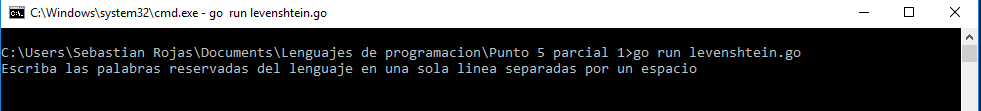
Antes de poder usar el programa, se debe importar la librería **gonum** por lo que se debe ejecutar el siguiente comando en la terminal después de haber instalado el lenguaje:

go get -u gonum.org/v1/gonum/...

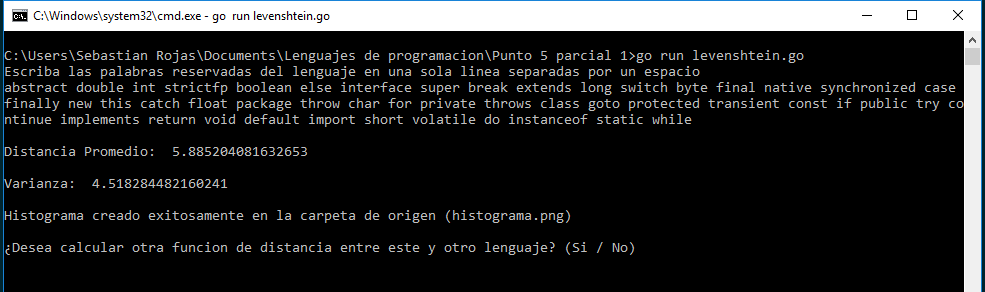
Una vez importada la librería, se puede utilizar el programa desde una línea de comandos ubicada en el archivo fuente levenshtein.go.

go run levenshtein.go

Una vez ejecutado el programa, este le solicitará que ingrese las palabras reservadas del lenguaje que se quiere procesar: Se deben ingresar todas las palabras del lenguajes en una sola linea y separadas por un espacio.

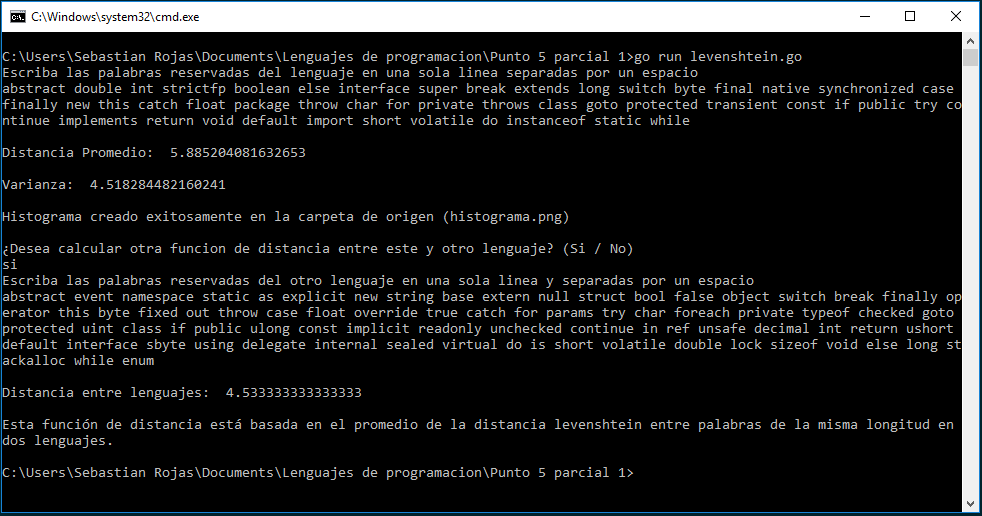


Una vez ingresado el lenguaje, el programa calcula la distancia Damerau-Levenshtein promedio, junto con la varianza. Se genera el histograma del lenguaje en el directorio donde se encuentra el archivo levenshtein.go.



Por último si el usuario desea hacer uso de la función de distancia propuesta, el programa permite ingresar Si (también si) en caso afirmativo, en caso contrario cualquier otra cadena ingresada terminará la ejecución del programa.

En caso de ingresar si, el programa solicita ingresar las palabras reservadas del segundo lenguaje, luego de esto imprime el resultado junto con una breve descripción de la función sugerida y se termina la ejecución.



Parcial Lenguajes de Programación 2020-II: Punto #5

Manual Técnico

Jhon Sebastian Rojas Rodriguez

El programa utiliza una variable global, una matriz **distancias** de tamaño 10000x10000 para realizar el cálculo de la distancia Damerau-Levenshtein. Las funciones que realizan este cálculo guardan los valores calculados para evitar computar varias veces el mismo valor (memoization).

El programa implementa varias funciones auxiliares para su funcionamiento:

***inicializarMatriz()****:* Aparta el espacio de memoria de la matriz distancias.

***levenshtein(s1 string, s2 string) int****:* Función que implementa el algoritmo, asigna los valores base de la recursión en la matriz distancias de acuerdo al número de palabras reservadas del lenguaje. Y realiza el primer llamado a la función recursiva retornando su valor.

***distance(x int, y int) int****:* Función recursiva de programación dinámica.

cost(x int, y int) int: función auxiliar de la función recursiva.

***Min2(x, y int) int y Min3(x, y , z int) int:*** Funciones que calculan el mínimo de dos y tres números enteros, se implementó debido a que no se encontró dentro del estándar del lenguaje funciones para realizar esta tarea.

***calcularPromedio(matriz [][]int, n int) float64:*** función que a partir de la matriz de distancias para el lenguaje, calcula la distancia promedio entre todas sus palabras.

***calcularVarianza(matriz [][]int, n int, promedio float64) float64:*** Función que calcula la varianza de la distancia entre todas las palabras, hace uso de la matriz de distancias entre las palabras del lenguaje y la distancia promedio entre estas.

***imprimirMatriz(matriz [][]int, n int, m int):*** Función implementada para imprimir en la consola una matriz de enteros, se implementó con fines de depuración del programa.

***crear\_histograma(matriz\_distancias [][]int, N int) bool:*** Función que genera la imagen del histograma del lenguaje en el directorio de origen. hace uso del modulo plot de la libreria gonum, regresa true si se pudo generar la imagen de manera satisfactoria y false en otro caso. Transforma la matriz de distancias en un vector de flotantes con las distancias entre palabras.

***custom\_distance(lenguaje1 []string, lenguaje2 []string) float64:*** Función que implementa la función de distancia recomendada, necesita de los dos arreglos de palabras reservadas de los lenguajes y retorna el resultado como un flotante. realiza una sumatoria sobre todos los pares de palabras de igual longitud y se divide entre el número de estos pares para calcular el promedio.

Finalmente la función Main inicializa la matriz de programación dinámica y solicita al usuario que ingrese las palabras del lenguaje a procesar, calcula la matriz de distancias iterando sobre todas las palabras del lenguaje y luego de esto imprime el resultado de las llamadas a las funciones de promedio y varianza.

Pregunta si se quiere hacer uso de la función de distancia sugerida, en cuyo caso solicita las palabras del segundo lenguaje e imprime el resultado del llamado a la respectiva función.