

# TP Final Programación 2

## Solución propuesta

### El problema:

Como fue dado en la consigna, el problema inicial a dar solución fue la predicción de texto a la hora de escribir. La solución propuesta es la descrita en cada parte de abajo.

### El programa en C:

Para el programa en C, cuyo objetivo era pasar "En Limpio" los textos dentro de una carpeta de un nombre dado, se creó un archivo de texto donde se guardo cada nombre dentro de la carpeta dada, para luego ir abriendolas en forma de *reading*.

A traves de una funcion declarada se fue iterando de carácter en carácter y, según haya sido tal carácter, pasó a través de un "filtro" que mediante condiciones dadas (ej: **"Si es un salto de linea y el carácter anterior no es un punto, entonces el carácter en limpio es un espacio"**) fue posible escribir el carácter correspondiente al archivo que se crea en Entrada abierto en *writing*.

Una vez se haya pasado todo el texto, los archivos se van cerrando y se guardan los cambios. Seguidamente, se cierran los archivos usados para escribir y para leer los nombres de los textos y se termina ejecutando en system un comando utilizado para ejecutar el programa en python con el argumento inicial pasado.

### El programa en Python:

En el programa escrito en Python, en el cual el objetivo era que dado un nombre y un archivo de frases incompleto se reemplazara la palabra faltante por una adecuada, lo

primero que ocurre es la apertura de los archivos principales a utilizar (**El archivo de Entrada creado en el programa de C, el archivo de Frases incompleto y el archivo de Salida a escribir las frases completas**).

Se guardan todas las palabras de Entrada en una lista, separadas individualmente. Cada frase completa (**y en forma de cadena**) tambien se guarda en una lista.

Se itera sobre cada frase de la lista y se le elimina el salto de linea al final de cada una para asi poder separar palabra por palabra en otra lista.

En la sublista de palabras de la frase creada, se encuentra la posicion de la palabra faltante (**representada por “\_”**) y se ejecuta una funcion declarada en la que se buscan las palabras alrededor de la palabra a reemplazar en cierto rango, y se comparan con cada palabra en el texto de Entrada. Dependiendo de la posicion de la palabra faltante, se guarda la palabra anterior o siguiente correspondiente en el texto de entrada junto con un valor numerico que representa una probabilidad de que sea la palabra faltante dentro de un diccionario (**de la forma {“palabra”:valor}**). Si el diccionario resulta vacio porque las palabras alrededor no son encontradas, se vuelven a buscar las palabras alrededor de “\_” pero con un rango mayor y una probabilidad menor. Una vez encontradas las palabras candidatas a ser la adecuada, se comparan los valores (**probabilidades**) de cada una y se agregan a una lista final de donde se escojeran la de mayor valor o una aleatoria si tienen la misma probabilidad. Finalmente retorna la palabra encontrada.

Una vez dada la palabra encontrada, la palabra faltante es reemplazada por esta en la sublista de palabras de la frase y mediante otra funcion se crea la frase completa con un salto de linea al final.

Se escribe en el archivo de Salida la frase formada y se prosigue con la siguiente frase hasta que no hayan más.

Una vez no hayan mas frases, se cierran los tres archivos abiertos al principio y se guardan los cambios, terminando así el programa y dejando el archivo objetivo

de frases completas en el directorio de Salida, con el nombre pasado como argumento.

Es así como fue hecha la proposición de solución para un programa de predicción de texto al escribir, basado en las probabilidades de palabras precedentes y siguientes en un texto dado, para luego usar esa información para auto-completar palabras en frases.