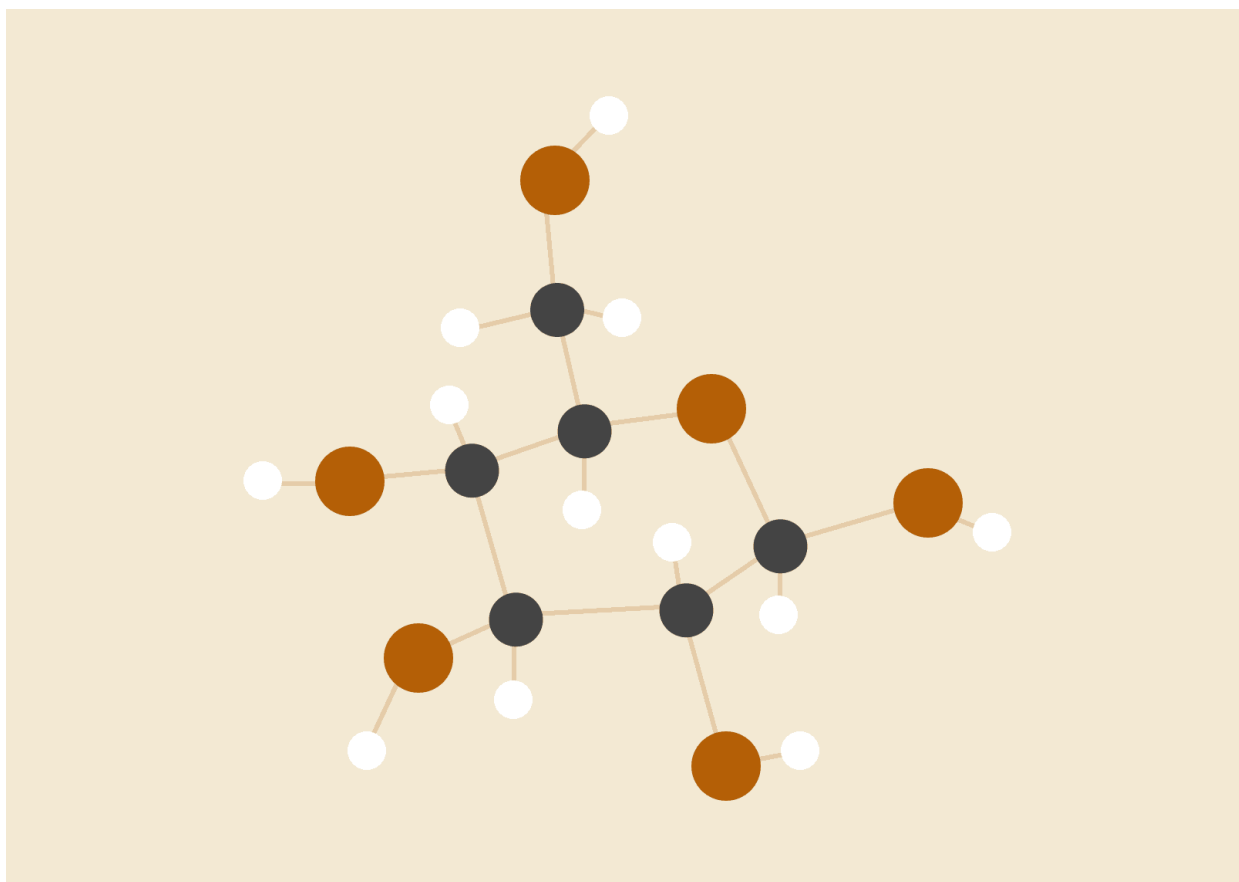


TP FINAL - LABORATORIO II

Universidad tecnológica nacional - Sede Mar del Plata



Equipo de trabajo:

- Devincenzi, Anabella Carla
- Gaitan, Lucas Agustin
- D'Aleo, Braian Ariel

Viernes 26 de Noviembre de 2021



DICCIONARIO

> **Terminos.c / Terminos.h**

guardarNomTextArchivo ()

Genera un archivo de nombres de tipo InfoTexto a partir de un nombre ingresado por consola, generando un ID autoincremental cada vez que esto sucede. Esto se usa para luego generar el diccionario.

generarStringTxt ()

Abre el archivo .BIN de texto y retorna un string de la dimensión exacta del archivo de texto.

validarChar ()

Recibe un carácter y retorna 1 si el carácter forma parte del vocabulario. Retorna 0 si el carácter es un símbolo especial, número, etc.

contarCantidadPalabras ()

Recibe un string y cuenta las palabras delimitado por caracteres especiales. Esto nos permite obtener la dimensión del arreglo de términos.

textoToArregloTermino ()

Divide un string en palabras, cargando un arreglo de tipo “Término”. Además si encuentra una minúscula la convierte a minúscula. Esto se hace mediante una validación, al igual que el contador, evaluando char por char mediante bloques de caracteres válidos (palabras). Cuando llega al final de un segmento válido le agrega el fin de cadena “/0”.

guardarTextoEnArchivoDic ()

Recibe un arreglo de Términos, y los validos, y los guarda en un archivo de tipo Diccionario con un arreglo de Términos.

MOTOR DE BÚSQUEDA

> NodoA & NodoT

crearArbolDiccionario ()

Carga el árbol en orden alfabético, recorriendo el archivo diccionario.

existeNodo ()

Verifica si una palabra se encuentra en el árbol. Retorna el nodo donde se encuentra la palabra.

insertarNodoT ()

NodoT es la lista de ocurrencias que tiene cada palabra, se ordena primero por ID y luego por posición.

FUNCIONES DE USUARIO

1. Buscar todas las apariciones de un término en algún documento.

Utiliza existeNodo, y en caso de existir se muestra por pantalla.

2. Buscar todas las apariciones de un término en un documento y otro.

El usuario ingresa la palabra a buscar, y los ids de los dos textos. Verifica que la palabra se encuentre en ambos textos al mismo tiempo. Lo muestra por pantalla.

buscarIdOcurrencias ()

Recorre la lista de ocurrencias hasta encontrar el ID correspondiente. En caso de éxito retorna 1, sino 0.

buscarPalabraporAnd ()

Por lo retornado mediante buscarIdOcurrencias compara si los dos valores devolvieron 1, y lo muestra.

3. Buscar la aparición de más de un término en un mismo documento.

buscaPalabraMismoDoc ()

Busca en una lista de palabras ingresadas por el usuario, que todas se encuentren en un mismo documento.

4. Buscar una frase completa

leerPalabra ()

El usuario ingresa una frase, y se lee caracter a caracter, delimitando mediante espacios generando palabras hasta que se presione la tecla enter. La frase puede tener cualquier tamaño, y no se limita con anterioridad. Las palabras se insertan en una lista.

La totalidad de términos ingresados deben coincidir contiguamente por posición mediante el mismo ID.

5. Busca la palabra de mayor frecuencia en algún documento

Recorre el árbol filtrando por ID y buscando la palabra con mayor frecuencia.

Usa una función auxiliar que retorna la cantidad de frecuencias de una palabra en un ID especificado, y realizar la comparacion en cada nodo del arbol

6. Distancia de Levenshtein

Recorre el árbol buscando palabras con una diferencia menor a 3 según la función de Levenshtein, y las muestra por pantalla.

