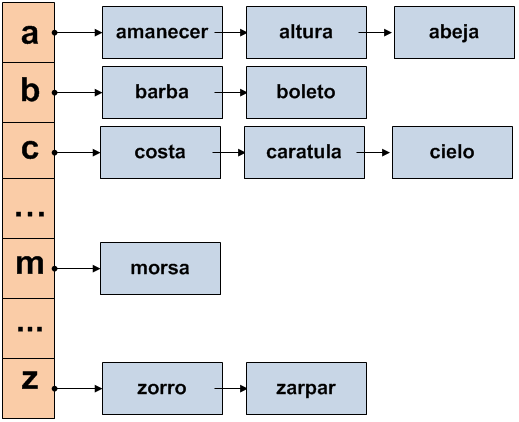
TDA DICCIONARIO

Un diccionario es una recopilación de términos o palabras, y sus respectivas descripciones, que se organizan de acuerdo a un orden alfabético de modo que la búsqueda de un término se realice rápidamente.

En informática, un diccionario puede definirse como una colección de elementos ***(clave, valor)*** que se organizan según un criterio de orden preestablecido. En un diccionario de palabras, la ***clave*** hace referencia a la primera letra del término o palabra, mientras que ***valor*** representa a la palabra propiamente dicha. A fin de agilizar el proceso de búsqueda, las estructuras de datos que implementan diccionarios agrupan los *valores* por *clave*. Por ejemplo, las palabras “amanecer”, “altura”, “arquero” se agrupan bajo la clave “a”, mientras que las “mano”, “moto”, “misterio” se agrupan bajo la clave “m”. La siguiente figura muestra una posible organización de los elementos de un diccionario.



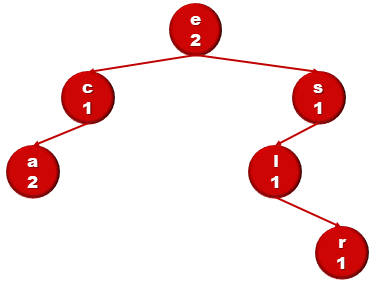
**Figura: Organización de un diccionario.**

Considerando la definición del TDA diccionario se especifican las siguientes operaciones:

* ***Tamaño*:** Retorna el número de palabras del diccionario.
* ***Consulta palabra*:** Determina si una palabra pertenece o no al diccionario.
* ***Consultar* *por clave*:** Muestra las palabras que corresponden a una clave determinada.
* ***Insertar palabra*:** Agrega una palabra en el diccionario de acuerdo a la clave que corresponda.
* ***Borrar* *palabra*:** Remueve una palabra, retornando la palabra extraída.
* ***Listar*:** Muestra todas las palabras del diccionario.

El TDA diccionario puede utilizarse para implementar una agenda electrónica, un diccionario de sinónimos, una base de datos de empleados, las notas de alumnos, etc.

A su vez el tratamiento de elementos de un diccionario puede realizarse utilizando estructuras como árboles binarios. Por ejemplo, al almacenar la palabra “escalera” en un árbol binario



Será sencillo determinar la cantidad de caracteres (longitud de la palabra), cantidad de vocales, cantidad de consonantes, el carácter de mayor frecuencia (el que más se repite), etc. Por ejemplo, las letras “a” y “e” son las que tienen mayor frecuencia (aparecen 2 veces cada una), la palabra tiene longitud 8, la palabra contiene 4 vocales, etc.

**Enunciado**

1. Considerando los conceptos de TDA Diccionario y TDA árbol binario, y sus operaciones asociadas, codifique un programa que implemente un diccionario de palabras con las siguientes operaciones:

**\*\*\*\*\*\*\* MENÚ PRINCIPAL \*\*\*\*\*\*\***

1. **Iniciar diccionario**
2. **Agregar palabra**
3. **Consultar tamaño**
4. **Consultar palabra (despliega sub-menú)**
5. **Borrar palabra**
6. **Listar palabras**
7. **Salir**

**\*\*\*\*\*\*\* MENÚ PRINCIPAL \*\*\*\*\*\*\***

**ELIJA UNA OPCIÓN:**

1. Las operaciones indicadas en el menú deben implementarse utilizando el **listas enlazadas y sus operaciones básicas**. Para ello, tenga en cuenta lo siguiente:

* La opción 1 permite inicializar el diccionario
* La opción 2 permite agregar una palabra, **no se permiten palabras repetidas**. Considere que las palabras deben almacenarse siempre en minúsculas (consulte funciones de conversión de C++).
* La opción 3 permite calcular la cantidad de palabras de la agenda.
* La opción 4 permite elegir una palabra del diccionario y acceder al sub-menú tratamiento de palabras (descripto más adelante).

**\*\*\*\*\*\*\* SUB-MENÚ PALABRAS \*\*\*\*\*\*\***

1. **Generar árbol de letras**
2. **Determinar longitud de la palabra**
3. **Mostrar letra/s con mayor frecuencia**
4. **Contar vocales y contar consonantes**
5. **Mostrar el máximo carácter de la palabra**
6. **Salir**

**\*\*\*\*\*\*\* MENÚ PRINCIPAL \*\*\*\*\*\*\***

**ELIJA UNA OPCIÓN:**

* La opción 5 permite eliminar una palabra del diccionario.
* La opción 6 permite lista todo el contenido del diccionario.
* La opción 7 presenta el mensaje “Fin del Programa” y muestra los datos del autor.

1. Las operaciones indicadas en el sub-menú “Palabras” deben implementarse utilizando **árboles binarios y sus operaciones básicas**. Para ello, tenga en cuenta lo siguiente:

* La opción 1 permite crear un árbol binario con las letras de la palabra elegida (tómese como referencia el árbol presentado en la introducción de este trabajo)
* La opción 2 permite determinar la cantidad de caracteres que componen la palabra. Debe utilizarse el árbol generado (no se permite el uso de la función *strlen* u otras similares).
* La opción 3 permite mostrar la/s letra/s con mayor frecuencia de aparición en la palabra.
* La opción 4 permite determinar la cantidad de constantes y la cantidad de consonantes de la palabra.
* La opción 5 permite mostrar el máximo carácter dela palabra.
* La opción 6 permite salir del sub-menú y regresar al menú principal del programa.

Condiciones de presentación del trabajo

* **Programa fuente** (archivo cpp): El programa presentado debe **COMPILAR** y **EJECUTAR** correctamente, de lo contrario NO será evaluado. Además, se recomienda leer atentamente el enunciado ya **que los trabajos que no se ajusten a lo solicitado serán desestimados. El código fuente debe enviarse al docente al menos 3 días antes del examen.**
* **Informe** (archivo pdf): El informe debe contener descripciones, comentarios, observaciones y/o consideraciones realizadas por el alumno para la implementación del trabajo. Deben explicarse las estructuras de datos elegidas y las operaciones básicas modificadas para el trabajo.
  + - Hoja: A4
    - Carátula: escuela, orientación, asignatura, año lectivo, datos del Alumno.
    - Márgenes: 2 cm superior, 2 cm inferior, 2,5 cm izquierda y 2,5 cm derecha.
    - Encabezado: debe incluir nombre de la asignatura y del alumno.
    - Pie de página: debe incluir número de página.
* **Presentación** (archivo ppt/pptx): Se debe elaborar una presentación (PowerPoint o Software libre alternativo) del trabajo realizado para realizar la defensa oral del trabajo.

🖫🖫🖫