Actividad 1 – B

Instrucciones

1. Lea y comprenda cuidadosamente lo que se le solicite.
2. La Actividad puede ser desarrollada en grupos de 2 o 3 humanoides.
3. La Actividad debe ser desarrollada con el lenguaje de C++ y bajo el Paradigma Orientado a Objetos.
4. Aplicar las Buenas Prácticas para resolver el ejercicio.
5. Procure que los métodos desarrollados sean altamente cohesivos y con bajo acoplamiento.
6. Si los métodos desarrollados no están aplicando el Principio de Responsabilidad Única, considerar en refactorizar el código.
7. Crear un repositorio en GIT y programar con su compañero el código solicitado.
8. Al finalizar esta actividad, agregar el modelo UML y el código final al repositorio, hacer un push a la rama main. Proceda a compartirlo con el profesor. Recuerde como buena práctica, realizar un commit por cada funcionalidad completada.
9. Finalmente, se discutirán de forma rápida en una mesa redonda los principales hallazgos y pensamientos finales sobre el trabajo.

Descripción

Partiendo del programa suministrado, disponible en el Aula Virtual y de acuerdo a la actividad anterior referente a la lista de listas para el manejo de palabras (Diccionario de palabras), tal como se indica en la figura 1. En donde se tiene una lista principal de Letras y a partir de cada nodo se accede a una lista de objetos de tipo Palabra que corresponden con la inicial que se indicó en la lista principal.

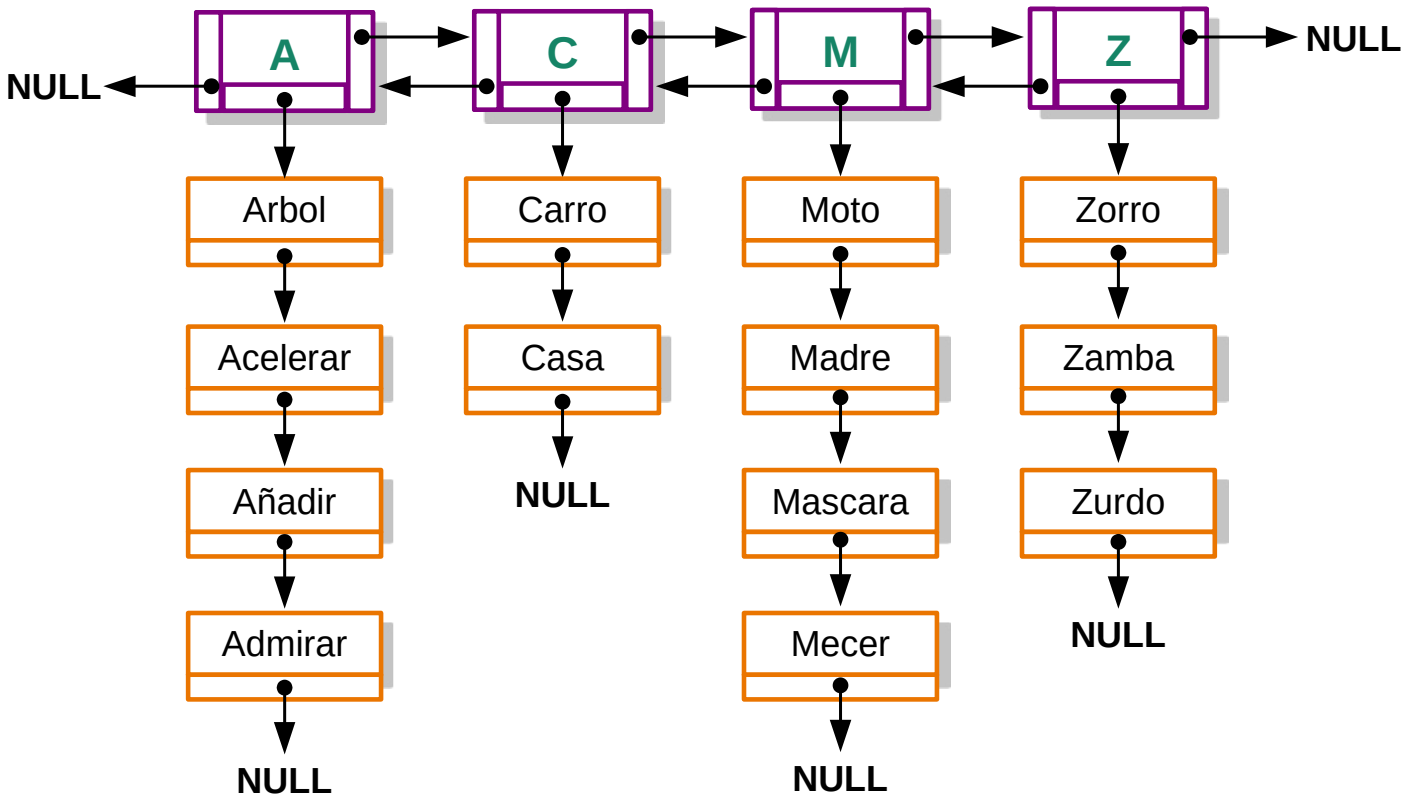

Figura 1. Diccionario de palabras.

Realice lo siguiente:

1. Analizar el código suministrado, comprender la estructura y generar el UML de la solución suministrada.
2. Seguidamente, determinar los elementos (clases y métodos) necesarios que le permitan a un usuario poder realizar las siguientes funcionalidades:
    1. Se deberá contar con una lista doblemente enlazada, en donde el usuario podrá agregar y eliminar Letras (todas las que se requieran). Las letras se agregan en orden alfabético.
    2. Se deberá contar con listas simples, en donde el usuario podrá agregar, modificar y eliminar Palabras (todos los elementos cuantos se quieran). Para agregar una nueva Palabra, se deberá identificar la inicial de la Palabra, recorrer la lista principal para determinar a cual lista incluirla. Cuando la Palabra se agrega a la lista, deberá poseer el comportamiento de una PILA (LIFO).
    3. Se le deberá permitir a un usuario buscar a una Palabra. Una vez encontrada se deberá indicar si se encontró o no.
    4. Partiendo de la funcionalidad anterior, el programa deberá restringir agregar Palabras que ya se encuentren en la lista.
    5. Asimismo, el programa deberá mostrar la cantidad actual de Palabras que hay registradas para cada Letra o en todas (el usuario deberá escoger entre ambas opciones, ambas funcionalidades deberán estar implementadas y disponibles al usuario).
    6. De igual manera, el programa deberá imprimir todas las Palabras de lista de listas o todas las Palabras de una determinada Letra (el usuario deberá escoger entre ambas opciones, ambas funcionalidades deberán estar implementadas y disponibles al usuario).
3. El nuevo código deberá estar refactorizado, convertido a inglés y se deberán con las buenas prácticas.

4. **Pruebas Unitarias:** Realice los métodos necesarios que le permitan verificar el funcionamiento de las funcionalidades, según lo siguiente:
    1. Agregar todas las Palabras que se muestran en la figura 2.
    2. Agregar una Palabra “Mascara” en la lista de listas. Esta acción debería indicar que la palabra ya se encuentra en el Diccionario.
    3. Buscar la Palabra “Zancada” en el Diccionario. Esta acción debería indicar que la palabra no se encuentra en el Diccionario.
    4. Buscar la Palabra “Zamba” en el Diccionario. Esta acción debería indicar que la palabra se encuentra en la letra Z.
    5. Mostrar la cantidad de palabras que se encuentran registradas en el Diccionario. Esta acción deberá mostrar la cantidad de 13 palabras.
    6. Mostrar la cantidad de palabras que se encuentran registradas en el Diccionario con la letra “M”. Esta acción deberá mostrar la cantidad de 4 palabras.
    7. Listar a todas las palabras que se encuentran registradas en el Diccionario con la letra “A”. Esta acción deberá mostrar las palabras [“Arbol”, “Acelerar”, “Añadir”, “Admirar”].

5. **Funcionalidades adicionales:**
    1. Se deberá crear un método que ordene las palabras (ordenadas alfabéticamente), por ejemplo, la palabra “Canario” debe ir antes que “Casa” y antes que “Cazar”.

    2. Se deberán crear los siguientes reportes:
        1. Mostrar el listado de todas las palabras según la inicial, se deberán mostrar en grupos de 5 palabras, tal como se muestra en la figura 2. Se le debe solicitar al usuario indicar a cuál letra desplegar las palabras.

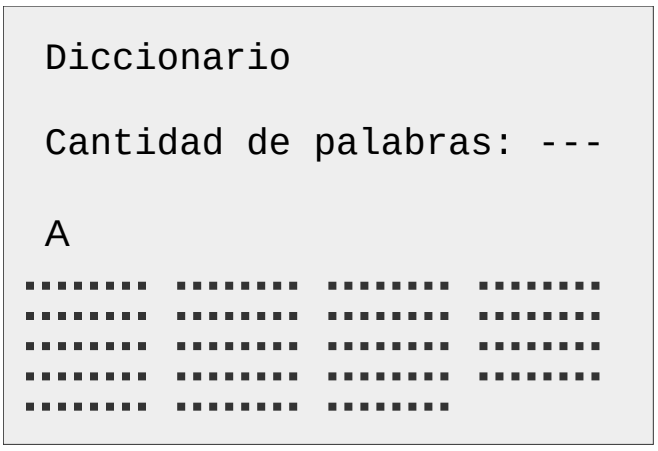

Figura 2. Listado de palabras según la inicial.

2. Mostrar el listado de todas las palabras que se encuentran en el Diccionario según la figura 3. En donde, se deberá mostrar todas las palabras en grupos de 4 letras (paginación), además, se le deberá permitir al usuario desplazarse hacia adelante o atrás según los grupos de palabras.

Figura 3. Listado de todas las palabras.

Nota:
El siguiente método permite posicionar el cursor en cualquier parte de la pantalla.

```
void goToXY(int coordX, int coordY) {
    HANDLE hcon;
    hcon = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD dwPos;
    dwPos.X = coordX;
    dwPos.Y = coordY;
    SetConsoleCursorPosition(hcon, dwPos);
}
```