

Objetivo o descripción del problema y Solución/Implementación:

El objetivo de esta práctica es la implementación de los BST . Para ello hemos tenido que implementar distintas funciones, entre otras algunas de estas son InOrder, PostOrder, generate_candidates,... Funciona de tal modo que programa lee palabras desde terminal y nos informa si la palabra ingresada es válida. En caso que la palabra no sea válida, el programa sugerirá una lista de posibles palabras correctas. Además tenemos tres opciones donde el diccionario se imprime ordenado el árbol binario, en este caso puede ordenarse de tres maneras diferentes. InOrder, PostOrder y PreOrder.

Se buscaba crear un diccionario de palabras utilizando un Binary Search Tree (BST) que pueda implementar diversas funcionalidades. Los programas que tenemos que implementar variarán respecto a la anterior práctica debido a que en esta ya no usaremos un hash, sino que debemos implementar un BST para almacenar palabras, en el que cada nodo esté compuesto por un ítem de información (la palabra) y dos punteros (izquierdo y derecho) a nodos del mismo tipo.

Debemos implementar todas las funciones propias de cualquier diccionario online empleado hoy en día en cualquier ordenador, como la inserción, búsqueda y borrado de palabras o funciones para listar el contenido del BST. Asimismo también debemos implementar varios programas que cumplan funciones más complejas, como tal que dado un fichero txt que contenga una lista de palabras, las almacene en la memoria y controle que no contenga repeticiones e informar en caso de que sí existan. Otro programa que debemos incorporar sería que, al igual que el anterior, dado un fichero txt que contenga una lista de palabras, sea capaz de verificar si cada palabra es correcta, y en caso contrario, que nos pueda sugerir posibles alternativas o correcciones ordenadas en orden alfabético.

Solución/Implementación

Respecto a la implementación del proyecto procedimos a organizarnos previamente las reuniones para llevar a cabo el trabajo al día y de este modo poder trabajar mejor en grupo. La solución a esta práctica ha sido la compenetración que hemos tenido los miembros del grupo y la facilidad de ponernos de acuerdo. En un principio nos llevó algo más de tiempo llevar adelante el proyecto pero finalmente una vez que entendimos bien cual era el principal objetivo de la práctica avanzamos mucho en cada reunión.

Cómo trabajamos cada miembro del equipo:

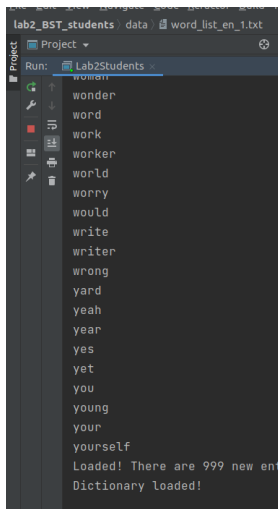
Al ser nosotros un grupo de trabajo muy polivalente y con experiencia trabajando juntos, desde el primer momento supimos cómo organizarnos para repartirnos las diferentes partes del trabajo y para saber cómo realizarlas. Pese a ello, la mayor parte del trabajo la hicimos

juntos, aunque en diferente medida, los tres intentamos aportar todo lo que pudimos para sacar el trabajo adelante.

La metodología seguida para hacer el trabajo fue clara, desde la propuesta de elaboración de esta práctica concordamos varias horas semanales para llevarla a cabo, ya sea viéndonos físicamente o de manera online y así llevarla lo mejor posible en cada momento.

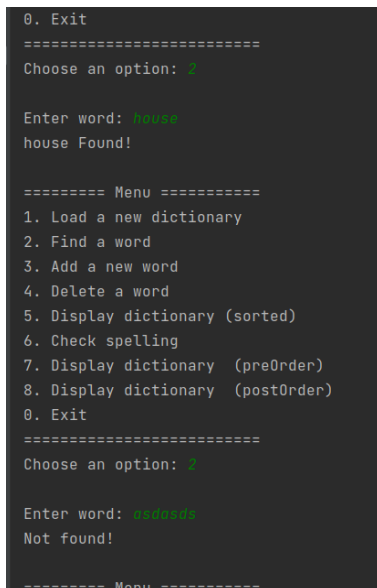
Ejemplos de ejecución:

1.- CARGAR DICCIONARIO:



```
lab2_BST_students / data / word_list_en_1.txt
Project
Run: Lab2Students
women
wonder
word
work
worker
world
worry
would
write
writer
wrong
yard
yeah
year
yes
yet
you
young
your
yourself
Loaded! There are 999 new ent
Dictionary loaded!
```

2.- ENCONTRAR PALABRA EN ARBOL:



```
0. Exit
=====
Choose an option: 2

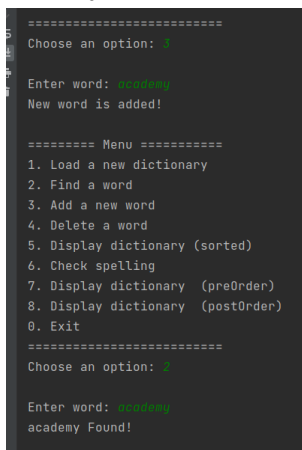
Enter word: house
house Found!

===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary (sorted)
6. Check spelling
7. Display dictionary (preOrder)
8. Display dictionary (postOrder)
0. Exit
=====
Choose an option: 2

Enter word: academy
Not found!

===== Menu =====
```

3.- AÑADIR PALABRA EN EL ARBOL(Comprobamos tambien que se haya añadido bien y que si ya esta en el arbol no se añade):

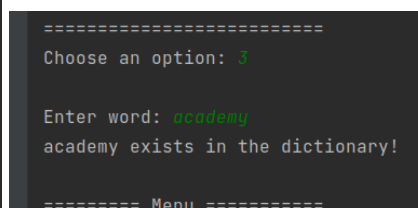


```
=====
Choose an option: 3

Enter word: academy
New word is added!

===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary (sorted)
6. Check spelling
7. Display dictionary (preOrder)
8. Display dictionary (postOrder)
0. Exit
=====
Choose an option: 3

Enter word: academy
academy Found!
```



```
=====
Choose an option: 3

Enter word: academy
academy exists in the dictionary!

===== Menu =====
```

4.- ELIMINAMOS UNA PALABRA(comprobamos que si intentamos eliminar una que no estaba, lo dice):

```
=====
Choose an option: 4

Enter word: academy
Deleted!

===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary (sorted)
6. Check spelling
7. Display dictionary (preOrder)
8. Display dictionary (postOrder)
0. Exit
=====
Choose an option: 2

Enter word: academy
Not found!

===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary (sorted)
6. Check spelling
7. Display dictionary (preOrder)
8. Display dictionary (postOrder)
0. Exit
=====
Choose an option: 4

Enter word: academy
Not found!
```

5.-Imprimimos INORDER(Cargamos primero el arbol de word_list_en_3.txt):

```
word_list_en_3.txt
Loading dictionary from file: ../data/word_list_3.txt
xxx
aaa
ccc
bbb
zzz
Loaded! There are 5 new entries.
Dictionary loaded!

===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary (sorted)
6. Check spelling
7. Display dictionary (preOrder)
8. Display dictionary (postOrder)
0. Exit
=====
Choose an option: 5

El arbol tiene 5 elementos
aaa
bbb
ccc
xxx
zzz
```

7.-Imprimimos PREORDER:

```
=====
Choose an option: 7

xxx
aaa
ccc
bbb
zzz
```

8.-Imprimimos POSTORDER:

```
=====
Choose an option: 8

bbb
ccc
aaa
zzz
xxx
```

6.-Generate Candidates:

By substitution: Si reemplazamos la 'a' por una 'o' obtenemos house.

```
=====
Choose an option: 6

Enter word: house
Substitutes:
cause
house
```

By Swapping: Si cambias 'uo' por 'ou' obtenemos house.

```
=====
Choose an option: 6

Enter word: house
Substitutes:
house
those
whose
```

By insertion: Si insertas 'o', obtenemos house.

```
0. Exit
=====
Choose an option: 6

Enter word: huse
Substitutes:
house
huge
use
```

By deletion: Si eliminas una 'o' obtenemos house.

```
=====
Choose an option: 6

Enter word: hoouse
Substitutes:
house
```

Task 3:

```
Which file do you want to load to check the spell of the words?  
spell_this_1.txt  
Loading words to check spell from file: ../data/spell_this_1.txt  
Substitutes for agrement:  
agreement  
Substitutes for althogh:  
although  
Substitutes for althugh:  
although  
Substitutes for busines:  
business  
Substitutes for business:
```

En caso que no se encuentre sustitución a la palabra:

```
Which file do you want to load to check the spell of the words?  
spell_this_1.txt  
Loading words to check spell from file: ../data/spell_this_1.txt  
Substitutes for agrement:  
Esta palabra no está en el diccionario, prueba a añadir otro diccionario con la opción 1 o quizá la palabra tenga demasiados errores para adivinar que querías decir.  
Substitutes for althogh:  
Esta palabra no está en el diccionario, prueba a añadir otro diccionario con la opción 1 o quizá la palabra tenga demasiados errores para adivinar que querías decir.  
Substitutes for althugh:  
Esta palabra no está en el diccionario, prueba a añadir otro diccionario con la opción 1 o quizá la palabra tenga demasiados errores para adivinar que querías decir.
```

Finished in 781 ms (2 minutes ago)

1181

IMPRESIONES Y OPINIÓN PERSONAL

Esta práctica, sin ninguna duda, nos ha servido para entender mejor el programa de los árboles binarios (BST). Mientras las sesiones de teoría nos han servido mucho para entender el concepto general, aportando ideas al código actual y a la forma de afrontar los problemas propuestos.

En general, como es evidente algunas funciones han sido más complicadas de implementar que otras, pero sin embargo, la práctica tampoco nos ha supuesto demasiada dificultad ya que el grupo ha podido interiorizar perfectamente los conceptos propuestos. Igualmente, la última clase práctica que tuvimos, nos sirvió mucho para aclarar alguna duda que teníamos sobre algunas funciones.

Por otra parte, el hecho de que las prácticas sean en grupos de tres personas nos ha permitido escribir código más rápido y comparar diferentes ideas cuando estábamos pensando la implementación de la función. Por consiguiente, si hubiese sido un trabajo individual, hubiera sido algo más complicado y nos hubiese llevado más tiempo. Además, entendemos que este método de trabajo en equipo es muy útil especialmente para programar, ya que las diferentes ideas que proponen los integrantes del grupo siempre son bien recibidas. Sin embargo, también creemos que el hecho de hacer esta práctica con más personas hubiera supuesto más problemas y nos hubiera dificultado más el proyecto, debido a que tendríamos que escuchar muchas opiniones diferentes y eso nos llevaría a ir más lento.

A todo esto, la función más difícil ha sido `delete_node` por el hecho de que una vez eliminado el nodo, el hecho de restablecer el árbol en función de los hijos que tenga el nodo a eliminar nos ha parecido algo complejo y complicado de entender rápido.

Finalmente, pensamos que esta práctica ha sido una herramienta para implementar nuestro conocimiento adquirido tanto en las clases teóricas como en los seminarios. También esperamos que nuestro interés con la asignatura y las horas dedicadas en este proyecto sean transparentes en nuestra entrega.