

Objetivo o descripción del problema y Solución/Implementación:

El objetivo de esta práctica es la implementación de un programa para verificar la ortografía de una palabra comparándola con un diccionario de palabras correctas.

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.

Se buscaba crear un programa de corrección ortográfica que detectase si una palabra es correcta o incorrecta, y en caso de ser incorrecta nos sugiriera una lista de posibles reemplazos. Un corrector ortográfico puede ser implementado usando un diccionario de palabras válidas para verificar si la palabra es correcta. Para obtener reemplazos en caso de que sea incorrecto, hemos usado una serie de estrategias para generar candidatos. El diccionario de palabras válidas lo implementamos como una tabla de hash con buckets, usando listas encadenadas. Siendo la información que se guarda únicamente las palabras válidas. Para construir el algoritmo recurrente y generar los posibles reemplazos de una palabra incorrecta, tuvimos que implementar varias funciones que generan las palabras candidatas y retener sólo las que se consideren válidas.

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.

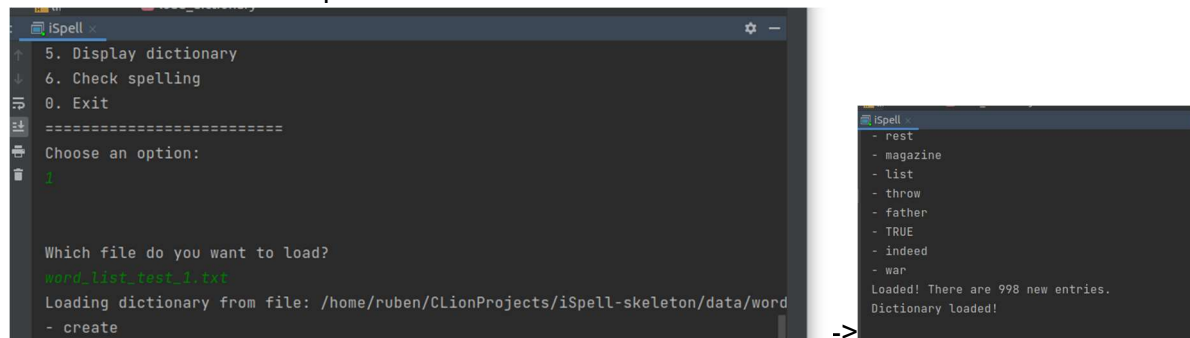
Cómo trabajamos cada miembro del equipo:

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. Función de cargar un nuevo diccionario, se puede ver como el terminal nos devuelve que el diccionario ha sido implementado correctamente.

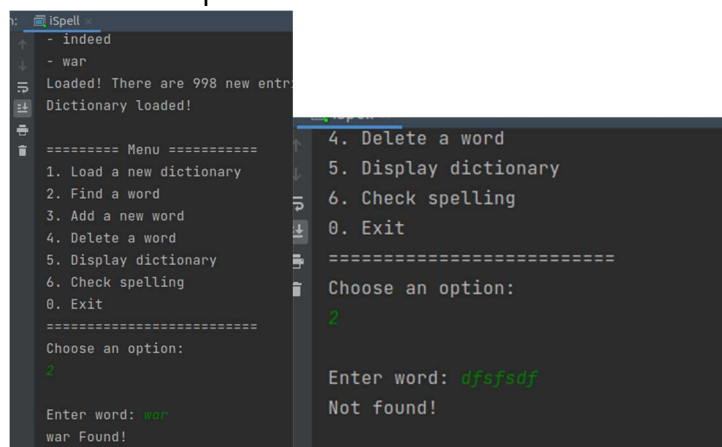


```
iSpell
5. Display dictionary
6. Check spelling
0. Exit
=====
Choose an option:
1

Which file do you want to load?
word_list_test_1.txt
Loading dictionary from file: /home/ruben/CLionProjects/iSpell-skeleton/data/word
- create

iSpell
- rest
- magazine
- list
- throw
- father
- TRUE
- indeed
- war
Loaded! There are 998 new entries.
Dictionary loaded!
```

2. Funcion de encontrar si una palabra esta en el diccionario, en la primera captura se ve un trozo del diccionario donde se ve la palabra war y efectivamente abajo cuando es buscada sale que se ha encontrado. En la segunda captura se ve como si pongo una palabra aleatoria dice que no se ha encontrado.



```
iSpell
- indeed
- war
Loaded! There are 998 new entries.
Dictionary loaded!

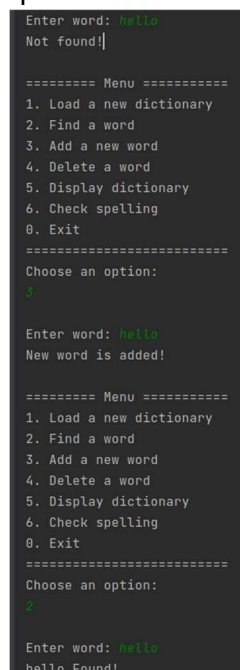
===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary
6. Check spelling
0. Exit
=====
Choose an option:
2

Enter word: war
war Found!

iSpell
4. Delete a word
5. Display dictionary
6. Check spelling
0. Exit
=====
Choose an option:
2

Enter word: dfsfsdf
Not found!
```

3. Al principio el terminal nos dice que la palabra Hello no existe, luego la añadimos, dice que se ha añadido correctamente y cuando la volvemos a buscar sale que la ha encontrado.



```
Enter word: hello
Not found!

===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary
6. Check spelling
0. Exit
=====
Choose an option:
3

Enter word: hello
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
6. Check spelling
0. Exit
=====
Choose an option:
2

Enter word: hello
hello Found!
```

4. Al principio el terminal nos dice que la palabra well se encuentra en el diccionario, luego la eliminamos y al volver a buscarla sale que ya no está en el diccionario.

```
Enter word: well
well Found!

===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary
6. Check spelling
0. Exit
=====
Choose an option:
4

Enter word: well
Deleted!

===== Menu =====
1. Load a new dictionary
2. Find a word
3. Add a new word
4. Delete a word
5. Display dictionary
6. Check spelling
0. Exit
=====
Choose an option:
2

Enter word: well
Not found!
```

5. Se puede apreciar como decimos que vuelva a escribir el diccionario y lo hace.

```
Choose an option:
5

Dictionary has 999 elements:
Bucket #0
yourself
word
way
watch
wall
wait
vote
```

6. En la primera captura se puede ver como el algoritmo donde dada una palabra te devuelve la posible borrando una de las letras de la palabra. En la segunda captura se aprecia como funciona el algoritmo donde devuelve la sugerencia añadiendo una letra en

cualquiera de las posiciones de la palabra. En la tercera se usa el algoritmo que swapea las letras para buscar una palabra que esté en el diccionario. En la ultima se ve que el algoritmo usado es el que reemplaza cada letra por una de las letras de la 'a' a la 'z' para ver si alguna de las palabras formadas coincide con alguna del diccionario.

```
5. Display dictionary
6. Check spelling
0. Exit
=====
Choose an option:
6

Enter word: hooouse
house
```

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

Enter word: huse
huge
use
house
```

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

Enter word: huouse
whose
house
those
```

```
5. Display dictionary
6. Check spelling
0. Exit
=====
Choose an option:
6

Enter word: haouse
cause
house
```

IMPRESIONES Y OPINIÓN PERSONAL

Esta práctica, sin ninguna duda, nos ha ayudado a entender mejor el programa de la corrección ortográfica. 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. 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 mucho más complicado. 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.

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.