# Exercício P.3.2. de Estrutura de Dados

Desenvolva um programa que leia os dados de um arquivo de texto, contendo palavras em ordem alfabética, e que armazene cada uma das palavras em um vetor (cada elemento do vetor contém uma palavra).

Depois de carregar os dados no vetor, o programa solicita ao usuário a digitação de uma palavra e então faz a busca binária e informa a posição da palavra no vetor.

O programa também deve fornecer como saída o número de passos de comparação que foram efetuados até que a palavra seja encontrada.

As entradas para o programa devem ser o arquivo de texto contendo as palavras em ordem alfabética e a palavra buscada.

A saída o número de passos e em qual posição a palavra foi encontrada.

## Resolução

1. **Ler o arquivo e armazenar palavras num vetor:** Usando o módulo *fs* para ler o arquivo e separar as palavras em um array;
2. **Solicitar a palavra ao usuário:** Usando o módulo *readline* para ler a palavra digitada;
3. **Implementar a busca binária:** Criar uma função que realize a busca binária e conte o número de comparações;
4. **Resultados:** Exiba a posição da palavra (ou uma mensagem se não for encontrada) e o número de comparações.

## Busca Binária

A busca binária é eficiente porque reduz o espaço de busca pela metade a cada passo, e o número de comparações feitas pode indicar essa eficiência.

Em vez de procurar linha por linha (como na busca linear), a busca binária faz *log₂(n)* comparações no pior caso, onde *n* é o número total de palavras.

* **Passo 1:** Compara a palavra com o elemento do meio do vetor;
* **Passo 2:** Dependendo do resultado, vai para a metade superior ou inferior;
* **Passo 3:** Novamente compara com o meio da nova metade;
* Assim por diante até terminar as comparações.

## Exemplo

**Palavra:** Abacate

### Passo de Comparação 1:

* *meio = Math.floor((0 + 51) / 2) = 25*
* Palavra no índice 25 = "melancia"
* Compara "melancia" com "abacate"
* "melancia" vem depois de "abacate"
* Atualiza: fim = 24

### Passo de Comparação 2:

* *meio = Math.floor((0 + 24) / 2) = 12*
* Palavra no índice 12 = "gato"
* Compara "gato" com "abacate"
* "gato" vem depois de "abacate"
* Atualiza: fim = 11

### Passo de Comparação 3:

* *meio = Math.floor((0 + 11) / 2) = 5*
* Palavra no índice 5 = "cavalo"
* Compara "cavalo" com "abacate"
* "cavalo" vem depois de "abacate"
* Atualiza: fim = 4

### Passo de Comparação 4:

* *meio = Math.floor((0 + 4) / 2) = 2*
* Palavra no índice 2 = "bola"
* Compara "bola" com "abacate"
* "bola" vem depois de "abacate"
* Atualiza: fim = 1

### Passo de Comparação 5:

* *meio = Math.floor((0 + 1) / 2) = 0*
* Palavra no índice 0 = "abacate"
* Compara "abacate" com "abacate"

### Resultado:

* Posição no vetor: 0
* Número de passos de comparação: 5

## Exemplo

**Palavra:** melancia

### Passo de Comparação 1:

* *meio = Math.floor((0 + 51) / 2) = 25*
* Palavra no índice 25 = "melancia"
* Compara "melancia" com "melancia"

### Resultado:

* Posição no vetor: 25
* Número de passos de comparação: 1