

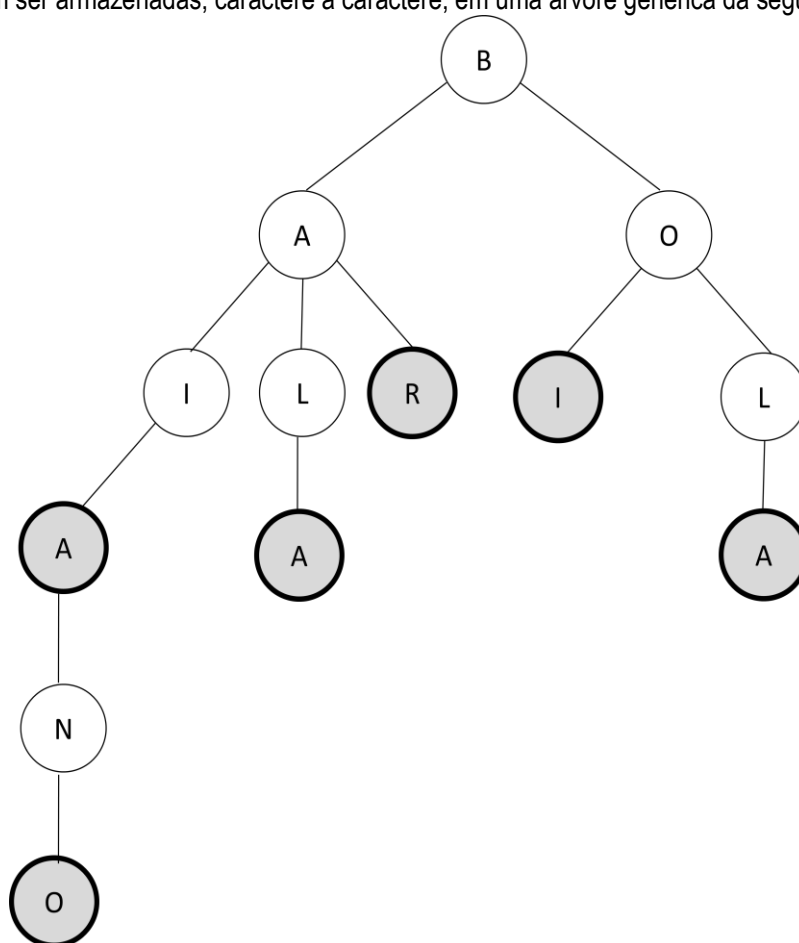
## Trabalhando com Árvores

### Instruções

O objetivo deste trabalho, que pode ser **feito em dupla ou individualmente**, é desenvolver uma solução para gerenciar uma árvore de palavras. Para isto, deverá ser lido um arquivo de palavras que deverá ser armazenado em uma árvore. A partir de dois ou três caracteres digitados, deverá ser apresentada a lista de palavras que começam com estes caracteres.

### Detalhamento

As palavras devem ser armazenadas, caractere a caractere, em uma árvore genérica da seguinte maneira:



Deve haver uma maneira de indicar que um nó é o último caractere que forma uma palavra. Assim, as palavras são compostas pelos caracteres armazenados nos nós visitados da raiz até o último caractere que forma a palavra.

A entrada deve ser um conjunto de caracteres, e a saída deve ser as possíveis palavras que podem ser formadas a partir deste conjunto de caracteres. Considerando a árvore exemplificada acima, se fosse digitado “ba”, a lista de palavras de saída seria: baia, baiano, bala e bar.

### Estrutura de Dados:

A estrutura de dados a ser desenvolvida **deve ser baseada em uma árvore genérica**. Além disso, é necessário fazer uma **análise da complexidade de cada algoritmo** implementado para esta estrutura de dados.

## Aplicação:

Para usar e testar a estrutura de dados desenvolvida deverá ser feita uma aplicação para um dicionário. O funcionamento deve ser o seguinte:

- É fornecido um conjunto de caracteres;
- Após a pesquisa na árvore é retornada uma lista de palavras que iniciam com os caracteres fornecidos;
- É escolhida uma palavra desta lista;
- É apresentado o significado desta palavra.

Neste caso, a string com o significado da palavra poderá indicar que um nodo é o último caractere que forma uma palavra.

## Formato do Arquivo de Entrada:

O arquivo do dicionário deverá ter o seguinte formato:

Palavra 1; significado da palavra 1 Palavra 2; significado da palavra 2 Palavra 3; significado da palavra 3 Palavra 4; significado da palavra 4
--

Importante: este arquivo não pode conter caracteres com acento!

Como exemplo, e para testar os algoritmos desenvolvidos, está sendo passado junto à especificação do trabalho, um arquivo que contém um conjunto de nomes com os seus significados.

## Relatório:

No final deve ser feito um relatório descrevendo a solução proposta, contendo, obrigatoriamente:

- Um parágrafo introdutório contendo uma descrição do trabalho realizado, e incluindo seus objetivos;
- Uma descrição da estrutura de dados desenvolvida, incluindo a sua representação gráfica e uma análise de complexidade dos algoritmos;
- Uma descrição dos **principais algoritmos** apresentados em **português estruturado**;
- Comentários sobre as dificuldades encontradas para elaboração do trabalho;
- Exemplos dos resultados obtidos com a execução do trabalho (podem ser *prints* da execução da aplicação).

## Tarefas:

- Implementar a árvore e os algoritmos que permitam armazenar e consultar as informações de um dicionário;
- Implementar um programa que permita testar a estrutura de dados e os algoritmos implementados através das operações da aplicação com o arquivo de exemplo;
- Escrever o relatório (lembre de incluir no início o nome do aluno, ou os nomes dos alunos no caso de trabalhos feitos em dupla);
- **Entregar no Moodle um arquivo zip contendo apenas: os arquivos do código fonte (somente os .java) e o arquivo .pdf com o relatório.**

## Observações:

- Os trabalhos que NÃO FOREM ENTREGUES através do Moodle seguindo as regras estabelecidas, até o dia e horário especificado, não serão considerados!
- Trabalhos que apresentarem erro de compilação não serão considerados.
- Trabalhos que apresentarem cópias das soluções de outros colegas ou de outras fontes resultarão em nota zero para todos os alunos envolvidos.