

Lista de Exercícios 9 – INF05008

Siga as instruções sobre elaboração de exercícios de INF05008.

Nesta lista, para todas as **funções recursivas** você deve incluir modelo da solução. Esse modelo, em forma de comentários, deve explicitar como o algoritmo vai funcionar para o(s) caso(s) base e passo(s) da definição da função recursiva. Este modelo deve ter o seguinte formato, e pode ser colocado antes ou permeado ao código:

```
;; SE < ...é o caso base da def. de lista... > ENTÃO < ...resolver o problema ... >
;; SE < ... não é o caso base da def. de lista... > <== pode haver mais de um desses casos, e
                                                pode-se usar SENÃO, se for o último caso

;; ENTÃO < ...combinar soluções... >
;;      < ...fazer algo com... > < o primeiro elemento da lista >
;;      < ...solucionar o problema para... > < o resto da lista >
```

Abra o template da lista 9 e use as definições de dados dos conjuntos **Arquivo**, **Conteúdo** e **Diretorio**, que definem dados sobre arquivos, conteúdos de diretórios e informações sobre diretórios, respectivamente. Note que os tipos **Conteúdo** e **Diretorio** são definições com referência mútua. Além destes tipos, você deve definir todos outros tipos que forem necessários para resolver os exercícios (decidir quais tipos são necessários e defini-los faz parte da solução do problema).

1. Construa a função **arquivo-encontrado-no-nivel?** que, dados o conteúdo de um diretório e um nome de arquivo, verifica se existe um arquivo com este nome neste conteúdo, sem considerar subdiretórios. A função deve devolver o valor booleano correspondente.
2. Construa a função **arquivo-encontrado?** que, dados o conteúdo de um diretório e um nome de arquivo, verifica se existe um arquivo com este nome neste conteúdo, considerando subdiretórios, ou seja, a função retorna verdadeiro se um arquivo com o nome dado existe na lista de conteúdo recebida, ou em qualquer diretório da lista. A função deve devolver o valor booleano correspondente.
3. Desenvolva uma função, **insere-arquivo** que, dado um diretório corrente e um arquivo, insere este arquivo neste diretório, retornando o diretório atualizado. Se já existe um arquivo com este nome no diretório (sem considerar os conteúdos de sub-diretórios), este arquivo será substituído pelo novo arquivo.
4. Defina a função chamada **calcula-tamanho** que, dado um diretório, calcula o tamanho necessário em disco para armazenar este diretório. O tamanho é calculado da seguinte forma: para cada arquivo que faz parte do diretório, soma-se o tamanho do arquivo, e para cada diretório soma-se 10 (para guardar as informações sobre o diretório) ao tamanho de seu conteúdo. O tamanho deve considerar toda a árvore do diretório (ou seja, o diretório e todos seus subdiretórios).
5. Construa a função **mostra-caminho** que, dado um nome de arquivo e um diretório, mostra o caminho para encontrar esse arquivo, ou uma mensagem dizendo que o arquivo não está nesse diretório. O resultado deve ser uma string mostrando um caminho até o arquivo conforme os exemplos abaixo (podem existir mais de um caminho, podem escolher qualquer um para mostrar) ou a mensagem "**Arquivo não encontrado**". Exemplos:

```
(mostra-caminho DIR1 "arq1") = "DIR1 -> arq1"
(mostra-caminho DIR3 "arq1") = "DIR3 -> DIR1 -> arq1"
(mostra-caminho DIR3 "arq6") = "Arquivo não encontrado"
```
6. (Ponto extra) Construa a função **visualiza-diretorio**, que gera uma imagem para visualizar a árvore de um diretório. Não é necessário mostrar o tamanho dos arquivos, somente os nomes (de arquivos e diretórios).