Especificação do Tipo Abstrato de Dados (TAD) - Lista de Bebidas

**Especificação do TAD Lista de Bebidas**

Este documento apresenta a especificação do Tipo Abstrato de Dados (TAD) para uma lista ordenada de bebidas, contendo as operações solicitadas. Cada operação é descrita em termos de entrada, pré-condição, processo, saída e pós-condição, sem foco em detalhes de implementação.

# Operação: insere\_ord

## Entrada:

* + Uma lista l de bebidas (ordenada pelo nome).
  + Uma estrutura nova do tipo Bebida, contendo os campos nome, volume e

preco.

## Pré-condição:

* + A lista l deve ser válida (não nula).
  + A lista l não deve estar cheia.
  + A estrutura nova deve conter dados válidos (nome não nulo, volume e preço não negativos).

## Processo:

* + Insere a bebida nova na lista l de forma ordenada, com base no campo nome

(ordem alfabética).

* + Garante que a ordenação da lista seja mantida após a inserção.

## Saída:

* + Um inteiro: 1 se a inserção for bem-sucedida, 0 caso contrário (lista nula ou cheia).

## Pós-condição:

* + A bebida nova está inserida na posição correta na lista l, mantendo a orde- nação alfabética.
  + O tamanho da lista é incrementado em 1, se a inserção for bem-sucedida.
  + Se a inserção falhar, a lista permanece inalterada.

# Operação: remove\_ord

## Entrada:

* + Uma lista l de bebidas (ordenada pelo nome).
  + Uma string nome representando o nome da bebida a ser removida.

## Pré-condição:

* + A lista l deve ser válida (não nula).
  + A lista l não deve estar vazia.
  + A string nome deve ser válida (não nula).

## Processo:

* + Busca a bebida com o nome nome na lista l.
  + Se encontrada, remove a bebida e reorganiza a lista para manter a ordenação.

## Saída:

* + Um inteiro: 1 se a remoção for bem-sucedida, 0 caso contrário (lista nula, vazia ou bebida não encontrada).

## Pós-condição:

* + Se a remoção for bem-sucedida, a bebida com o nome nome não estará mais na lista, e a lista permanecerá ordenada.
  + O tamanho da lista é decrementado em 1, se a remoção for bem-sucedida.
  + Se a remoção falhar, a lista permanece inalterada.

# Operação: get\_elem\_pos

## Entrada:

* + Uma lista l de bebidas.
  + Um inteiro pos representando a posição do elemento desejado (base 0).
  + Um ponteiro b para uma estrutura do tipo Bebida onde o elemento será ar- mazenado.

## Pré-condição:

* + A lista l deve ser válida (não nula).
  + A posição pos deve ser válida (0 *≤* pos *<* tamanho da lista).
  + O ponteiro b deve ser válido (não nulo).

## Processo:

* + Recupera o elemento da lista l na posição pos e o armazena em b.

## Saída:

* + Um inteiro: 1 se a recuperação for bem-sucedida, 0 caso contrário (lista nula ou posição inválida).
  + O ponteiro b contém os dados da bebida na posição pos, se bem-sucedido.

## Pós-condição:

* + Se bem-sucedida, a estrutura apontada por b contém os dados da bebida na posição pos.
  + A lista l permanece inalterada.
  + Se a operação falhar, b permanece inalterado.

# Operação: esvazia\_lista

## Entrada:

* + Uma lista l de bebidas.

## Pré-condição:

* + A lista l deve ser válida (não nula).

## Processo:

* + Remove todos os elementos da lista l, deixando-a vazia.

## Saída:

* + Um inteiro: 1 se a operação for bem-sucedida, 0 caso contrário (lista nula).

## Pós-condição:

* + A lista l está vazia (tamanho igual a 0).
  + Se a operação falhar, a lista permanece inalterada.

# Operação: apaga\_lista

## Entrada:

* + Um ponteiro l para uma lista de bebidas.

## Pré-condição:

* + O ponteiro l deve ser válido (não nulo).
  + A lista apontada por l deve ser válida (não nula).

## Processo:

* + Libera a memória alocada para a lista e define o ponteiro l como nulo.

## Saída:

* + Um inteiro: 1 se a operação for bem-sucedida, 0 caso contrário (ponteiro ou lista nulos).

## Pós-condição:

* + A lista apontada por l é liberada, e l aponta para NULL.
  + Se a operação falhar, o ponteiro l permanece inalterado.

# Operação: imprime\_lista

## Entrada:

* + Uma lista l de bebidas.

## Pré-condição:

* + Nenhuma, já que a operação lida com casos de lista nula ou vazia.

## Processo:

* + Exibe no console as informações de todas as bebidas na lista l, incluindo posição, nome, volume e preço.
  + Se a lista for nula ou vazia, exibe uma mensagem indicando isso.

## Saída:

* + Nenhuma (saída é exibida diretamente no console).

## Pós-condição:

* + A lista l permanece inalterada.
  + As informações das bebidas são exibidas no console, ou uma mensagem de erro, se aplicável.