Documentação da Classe `Tabela_Hash`

Descrição
A classe `Tabela_Hash` implementa uma tabela hash que permite armazenar e gerenciar pares chave-valor. Ela utiliza uma técnica de resolução de colisões por meio de listas encadeadas. A classe fornece métodos para inserir, buscar, remover e listar elementos na tabela, bem como para avaliar seu desempenho.

Métodos Públicos

`Tabela_Hash(int tamanho)`
- **Descrição**: Construtor da classe `Tabela_Hash`.
- **Parâmetros**:
  - `tamanho` (int): O tamanho desejado da tabela hash.
- **Comportamento**: Inicializa a tabela hash com o tamanho especificado e configura os atributos internos.

`void inserir_tabela(double chave, const string &dado)`
- **Descrição**: Insere um elemento na tabela com a chave e o dado especificados.
- **Parâmetros**:
  - `chave` (double): A chave do elemento a ser inserido.
  - `dado` (const string&): O dado a ser armazenado.
- **Comportamento**: Calcula o índice da tabela com base na chave, verifica se a chave já existe na lista correspondente e, se não, insere o novo elemento. Lida com colisões e fornece feedback visual.

`void busca_chave(double chave)`
- **Descrição**: Busca um elemento na tabela pelo valor da chave e imprime o resultado.
- **Parâmetros**:
  - `chave` (double): A chave do elemento a ser buscado.
- **Comportamento**: Calcula o índice da tabela com base na chave e busca o elemento correspondente. Exibe os resultados na saída padrão.

`void busca_dado(const string &dado)`
- **Descrição**: Busca elementos na tabela pelo valor dos dados e imprime os resultados.
- **Parâmetros**:
  - `dado` (const string&): O dado a ser buscado.
- **Comportamento**: Percorre toda a tabela e suas listas para encontrar elementos com o dado especificado e exibi-los na saída padrão.

`void remove_chave(double chave)`
- **Descrição**: Remove um elemento da tabela pelo valor da chave.
- **Parâmetros**:
  - `chave` (double): A chave do elemento a ser removido.
- **Comportamento**: Calcula o índice da tabela com base na chave e remove o elemento correspondente, liberando a memória alocada. Lida com situações em que a chave não é encontrada.

`void remove_dado(const string &dado)`
- **Descrição**: Remove elementos da tabela pelo valor dos dados.
- **Parâmetros**:
  - `dado` (const string&): O dado a ser removido.
- **Comportamento**: Percorre toda a tabela e suas listas para encontrar elementos com o dado especificado e remove-os, liberando a memória alocada.

`void print_tabela()`
- **Descrição**: Exibe todos os elementos da tabela na saída padrão.
- **Comportamento**: Percorre toda a tabela e suas listas, exibindo as chaves e dados dos elementos.

`void mostrar_desempenho()`
- **Descrição**: Exibe informações sobre o desempenho da tabela hash, incluindo o número total de elementos, o número de listas não vazias, o fator de carga e o número de colisões.
- **Comportamento**: Calcula as métricas de desempenho com base no estado atual da tabela e exibe os resultados na saída padrão.

`~Tabela_Hash()`
- **Descrição**: Destrutor para liberar a memória alocada.
- **Comportamento**: Libera a memória alocada para os elementos da tabela hash ao destruir a instância da classe.

Atributos Privados

- `vector<vector<Item *>> tabela`: Vetor de listas de ponteiros para objetos do tipo `Item`, que armazenam os elementos da tabela hash.
- `int tamanho_tabela`: O tamanho da tabela hash.
- `int colisoes`: O contador de colisões.

Função Privada

- `int hash(int chave)`: Função de hash simples que calcula o índice da tabela com base na chave fornecida.