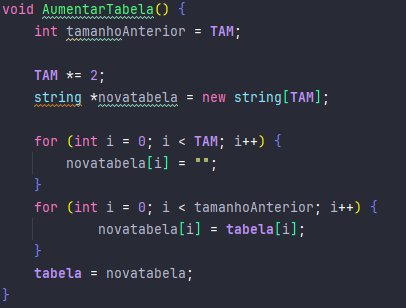
**Tabela Hash com Tratamento de Colisão via três métodos de Endereçamento Aberto**

### Introdução

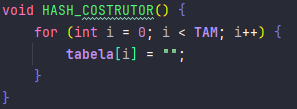
Este código implementa uma tabela hash dinâmica utilizando três tipos de tratamento de colisões por endereçamento aberto: **sondagem linear**, **sondagem quadrática** e **sondagem dupla**. O principal objetivo dessa estrutura de dados é armazenar e recuperar elementos de forma eficiente, mesmo em casos de colisões, que ocorrem quando diferentes chaves geram o mesmo índice. A função hash é responsável por determinar a posição inicial na tabela para cada chave inserida. Quando uma colisão é detectada, os métodos de sondagem buscam uma nova posição disponível na tabela. Além disso, o código implementa a expansão dinâmica da tabela, duplicando seu tamanho sempre que a ocupação atinge 75% da capacidade, garantindo a eficiência da operação de inserção.

### Funções do Código

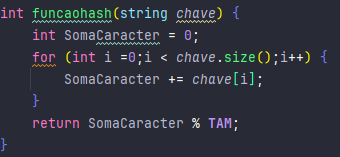
**Função AumentarTabela:** essa função é responsável por expandir a tabela hash quando a ocupação atinge 75% da capacidade. Primeiramente, ela armazena o tamanho atual da tabela em uma variável tamanhoAnterior. Depois, ela dobra o valor de TAM (tamanho da tabela) e cria uma nova tabela com o novo tamanho. A nova tabela é preenchida com strings vazias, e os elementos da tabela antiga são copiados para a nova. Esse processo é necessário para garantir que a tabela tenha espaço suficiente para novos elementos sem comprometer a performance das operações.



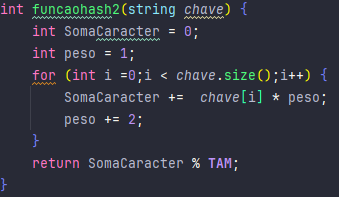
**Função HASH\_COSTRUTOR:** inicializa a tabela hash, atribuindo um valor vazio ("") a cada elemento da tabela. Essa função é chamada quando o programa é iniciado ou quando uma nova tabela é criada após o aumento do tamanho. Ela assegura que todos os espaços da tabela comecem vazios, evitando que dados de inserções anteriores interfiram nas novas operações.



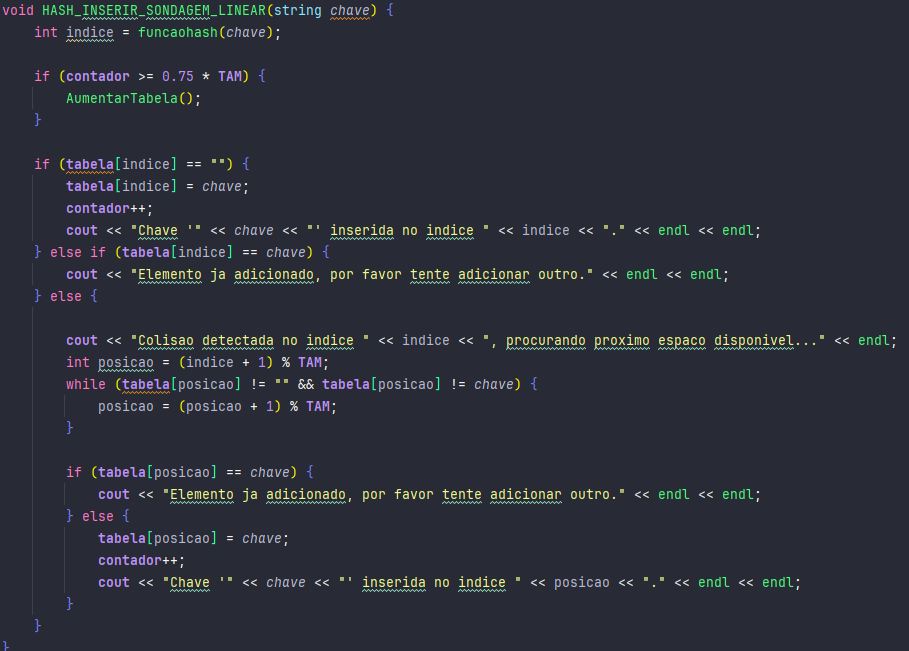
**Função funcaohash:** é responsável por calcular o índice da tabela para uma chave fornecida. O cálculo é feito somando os valores ASCII de cada caractere da chave e aplicando o operador módulo com o tamanho da tabela (TAM). O valor resultante define a posição inicial onde o elemento será armazenado. Esse método é simples e eficaz, mas pode gerar colisões, o que justifica o uso dos métodos de sondagem para tratar essas colisões.



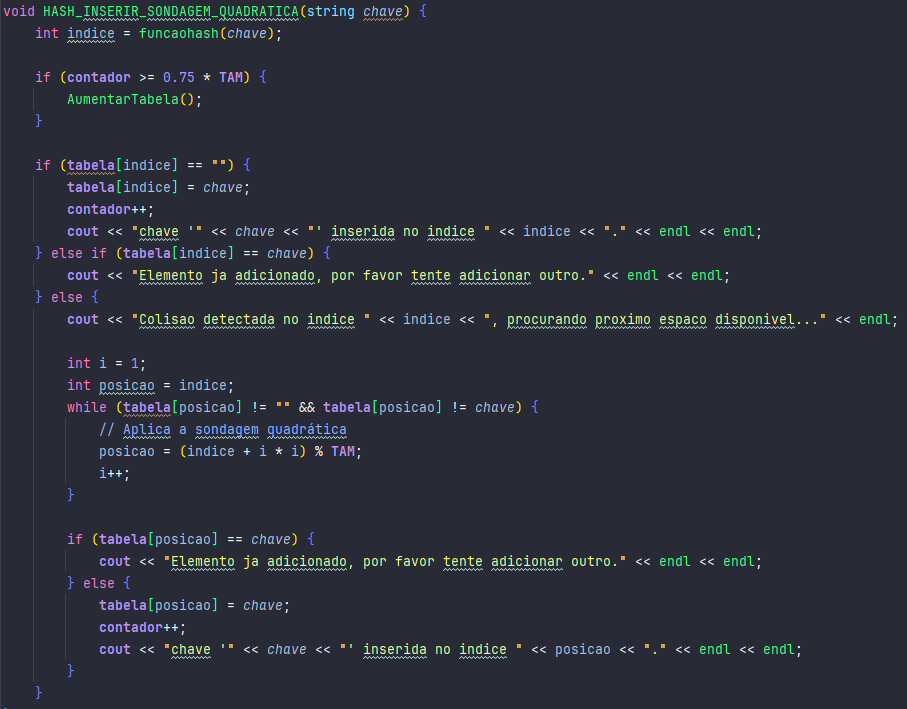
**Função funcaohash2:** é uma variação da função funcaohash, sendo utilizada especificamente na sondagem dupla. Ela aplica uma fórmula diferente para calcular o índice, utilizando um fator constante (31) para modificar o hash a cada caractere da chave. Esse hash resultante é usado para calcular o deslocamento em caso de colisões, ajudando a evitar padrões repetitivos de colisões e proporcionando uma distribuição mais uniforme das chaves na tabela.



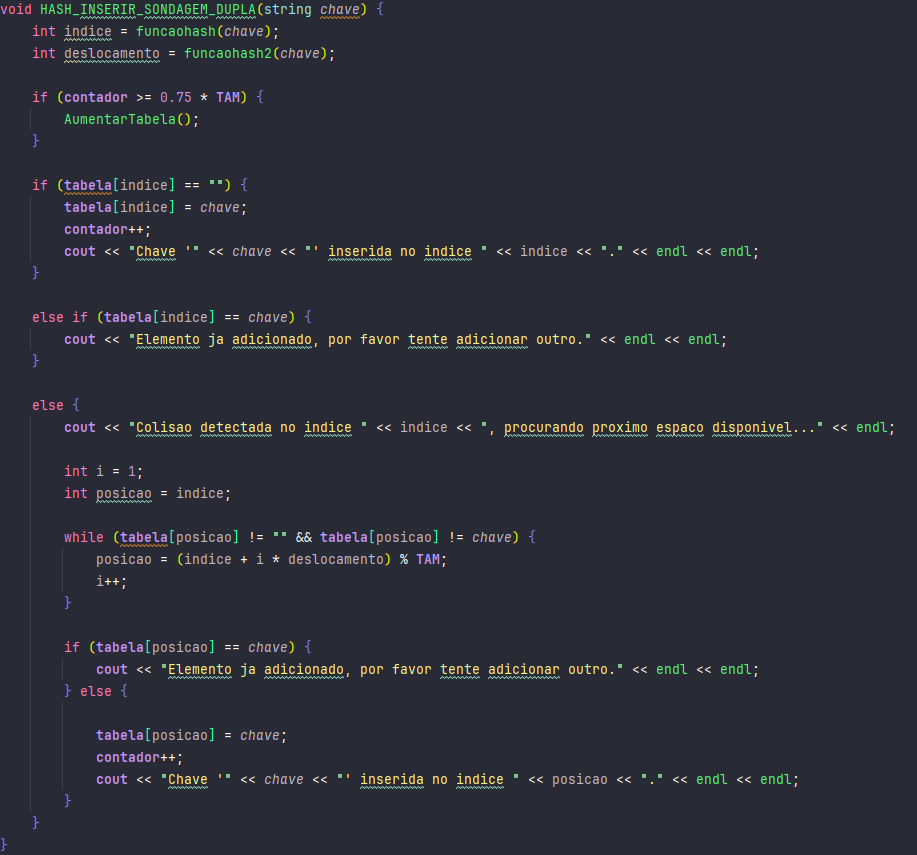
**Função HASH\_INSERIR\_SONDAGEM\_LINEAR:** esta função implementa o tratamento de colisões por **sondagem linear**. Quando uma colisão ocorre no índice calculado pela função hash, a função tenta inserir o elemento na próxima posição da tabela (índice + 1) até encontrar uma posição vazia ou até a tabela ser cheia. O processo de sondagem linear é simples e percorre a tabela sequencialmente. Caso a tabela atinja sua capacidade máxima, o código emite uma mensagem informando que a chave não pode ser inserida. A sondagem linear, apesar de ser eficiente para tabelas pequenas, pode levar a problemas de agrupamento de colisões (clusters) em tabelas maiores.



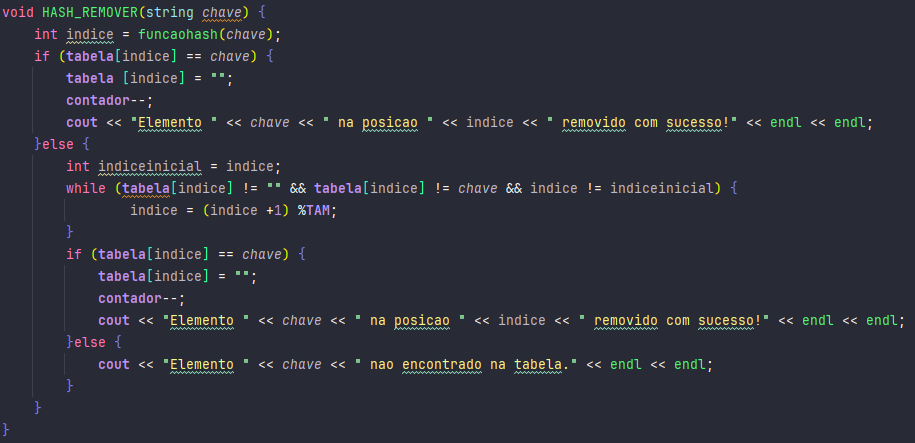
**Função HASH\_INSERIR\_SONDAGEM\_QUADRATICA:** resolve colisões utilizando **sondagem quadrática**. Quando ocorre uma colisão, em vez de simplesmente mover para a próxima posição (como na sondagem linear), o índice é recalculado de acordo com uma fórmula quadrática. A cada tentativa, o valor de deslocamento aumenta (1, 4, 9, 16, etc.), o que ajuda a espalhar as colisões de forma mais dispersa e reduz o agrupamento de colisões. Esse método geralmente proporciona uma distribuição mais uniforme e melhora a eficiência da tabela em comparação com a sondagem linear.



**Função HASH\_INSERIR\_SONDAGEM\_DUPLA:** utiliza **sondagem dupla** para resolver colisões. Ao contrário da sondagem linear e quadrática, a sondagem dupla usa duas funções de hash: uma para calcular o índice inicial e outra para calcular o deslocamento (caso ocorra uma colisão). O deslocamento é calculado com a segunda função hash (funcaohash2). Isso permite que a sondagem explore outras posições mais aleatórias na tabela, o que pode diminuir a probabilidade de colisões subsequentes e melhorar a eficiência de inserção.



**Função HASH\_REMOVER:** permite remover uma chave da tabela hash. Ela primeiro verifica o índice calculado pela função hash para encontrar o elemento. Caso a chave não esteja nesse índice, a função percorre a tabela utilizando a sondagem linear até encontrar a chave ou até concluir que o elemento não está presente. Uma vez que a chave é encontrada, ela é removida e o contador de elementos é decrementado. Caso o elemento não seja encontrado, o código informa que a chave não está presente na tabela.



**Função HASH\_BUSCAR:** busca uma chave na tabela hash. Assim como na função de remoção, ela começa verificando o índice calculado pela função hash. Caso a chave não esteja nesse índice, a função percorre a tabela utilizando o método de sondagem até encontrar o elemento ou concluir que ele não está presente. A função retorna uma mensagem indicando a posição da chave ou informando que a chave não foi encontrada na tabela.

