

# Estruturas de Dados

## Tabela de Dispersão - Exercícios

PONTIFÍCIA UNIVERSIDADE CATÓLICA  
DO RIO DE JANEIRO



# Questão 1

Considere um cadastro de alunos armazenado em uma tabela de dispersão (*hash*) que usa listas encadeadas para o tratamento de colisões, isto é, a tabela é implementada como um vetor de listas encadeadas. Considerando as seguintes declarações relativas à tabela *hash* dos alunos:

```
struct aluno {
    char nome[81];
    int matricula;
    float cr;
    struct aluno* prox;
};
typedef struct aluno Aluno;

#define N 127
typedef Aluno* Hash[N];
```

# Questão 1

Implemente:

- a) Uma função de *hash* que, dado o nome de um aluno, retorne o índice do aluno na tabela, de acordo com protótipo definido a seguir:

```
int hash (char* nome);
```

- b) Uma função que insira um novo aluno na tabela *hash*. Para simplificar, considere que a função estará sempre adicionando um novo aluno ao cadastro, isto é, não existe a hipótese do nome do aluno a ser inserido já constar no cadastro. A função deve receber como parâmetros a tabela *hash*, o nome do aluno, sua matrícula e seu CR, e deve retornar um ponteiro para a estrutura criada para o novo aluno, de acordo com o protótipo definido a seguir:

```
Aluno* insere (Hash tab, char* nome, int mat, float cr);
```

# Questão 1

```
int hash (char* nome)
{
    int i;
    int total = 0;
    for (i=0; nome[i]!='\0'; i++)
        total += nome[i];
    return (total%NTAB);
}
```

# Questão 1

```
Aluno* insere(Hash tab, char* nome, int mat, float cr);
{
    int h = hash(nome);
    Aluno* novo = (Aluno*) malloc(sizeof(Aluno));
    strcpy(novo->nome,nome);
    novo->mat=mat;
    novo->cr=cr;
    novo->prox=tab[h];
    tab[h]=novo;
    return novo
}
```