

# Trabalho Final - Sistema de Contas Bancárias em Linguagem C

## Descrição:

Você e sua equipe (de até cinco alunos) foram designados para desenvolver um programa em linguagem C que simule um sistema de contas bancárias. O programa deverá implementar uma série de funções para manipular registros de contas bancárias, armazenando informações como número da conta, nome do cliente, se a conta é especial, e o saldo atual.

Estrutura de Dados: Para representar cada conta bancária, você deve utilizar a seguinte estrutura:

```
typedef struct {  
    int numero;  
    char cliente[51];  
    int especial;  
    float saldo;  
} Conta;
```

A estrutura Conta possui os seguintes campos:

- numero (inteiro): número da conta bancária.
- cliente (string de 51 caracteres): nome do cliente.
- especial (inteiro): indica se a conta é especial (1) ou normal (0).
- saldo (flutuante): saldo atual da conta.

Funções a serem desenvolvidas:

- void inserir(Conta\* contas, int\* totalContas)
  - Descrição: Insere um novo registro de conta em uma posição do vetor de ponteiros.
  - Parâmetros:
    - contas: vetor de ponteiros para as contas bancárias.
    - totalContas: ponteiro para a variável que guarda o total de contas cadastradas.
- void alterar(Conta\* contas, int totalContas)
  - Descrição: Recebe um número de conta digitado pelo usuário e, se encontrada, solicita os novos dados para alteração (permitindo alterar apenas o nome do cliente).
  - Parâmetros:
    - contas: vetor de ponteiros para as contas bancárias.
    - totalContas: total de contas cadastradas.
- void procurar(Conta\* contas, int totalContas)

- Descrição: Recebe um número de conta digitado pelo usuário e localiza a conta no vetor. Se encontrada, exibe as informações da conta no console.
  - Parâmetros:
    - contas: vetor de ponteiros para as contas bancárias.
    - totalContas: total de contas cadastradas.
- void listar(Conta\* contas, int totalContas)
- Descrição: Imprime no console todos os dados de todas as contas bancárias cadastradas no vetor.
  - Parâmetros:
    - contas: vetor de ponteiros para as contas bancárias.
    - totalContas: total de contas cadastradas.
- void depositar(Conta\* contas, int totalContas)
- Descrição: Recebe um valor positivo e efetua o depósito na conta informada, alterando o saldo da conta.
  - Parâmetros:
    - contas: vetor de ponteiros para as contas bancárias.
    - totalContas: total de contas cadastradas.
- void sacar(Conta\* contas, int totalContas)
- Descrição: Recebe um valor positivo e efetua o saque na conta informada, alterando o saldo da conta.
  - Parâmetros:
    - contas: vetor de ponteiros para as contas bancárias.
    - totalContas: total de contas cadastradas.
- void imprimir(Conta\* contas, int totalContas)
- Descrição: Localiza uma conta e imprime os dados desta conta se existir.
  - Parâmetros:
    - contas: vetor de ponteiros para as contas bancárias.
    - totalContas: total de contas cadastradas.
- void saldoGeral(Conta\* contas, int totalContas)
- Descrição: Mostra o saldo total de todas as contas bancárias cadastradas.
  - Parâmetros:
    - contas: vetor de ponteiros para as contas bancárias.
    - totalContas: total de contas cadastradas.

Função main() e Menu: A função main() deve implementar um menu com as opções a seguir:

```
int main() {
    Conta contas[MAX_CONTAS];
    int totalContas = 0;
```

```

int opcao;

do {
    printf("\n----- Menu ----- \n");
    printf("1. Inserir nova conta\n");
    printf("2. Alterar dados de uma conta\n");
    printf("3. Procurar uma conta\n");
    printf("4. Listar todas as contas\n");
    printf("5. Realizar depósito\n");
    printf("6. Realizar saque\n");
    printf("7. Imprimir dados de uma conta\n");
    printf("8. Mostrar saldo geral\n");
    printf("0. Sair\n");
    printf("Opção: ");
    scanf("%d", &opcao);

    switch (opcao) {
        case 1:
            inserir(contas, &totalContas);
            break;
        case 2:
            alterar(contas, totalContas);
            break;
        case 3:
            procurar(contas, totalContas);
            break;
        case 4:
            listar(contas, totalContas);
            break;
        case 5:
            depositar(contas, totalContas);
            break;
        case 6:
            sacar(contas, totalContas);
            break;
        case 7:
            imprimir(contas, totalContas);
            break;
        case 8:
            saldoGeral(contas, totalContas);
            break;
        case 0:
            printf("Encerrando o programa...\n");
            break;
        default:
            printf("Opção inválida!\n");
    }
} while (opcao != 0);

return 0;
}

```

Observações importantes:

- Utilize constantes e macros adequadas, como MAX\_CONTAS, para limitar o tamanho do vetor de contas.
- Os parâmetros das funções podem ser ajustados conforme a necessidade.

- Certifique-se de realizar as validações necessárias (ex: número de conta existente, valor positivo para depósito/saque, etc.).
- Documente seu código com comentários adequados para facilitar a compreensão.

Entrega: O trabalho deverá ser entregue até a data limite especificada no Teams. Certifique-se de identificar cada membro da equipe no envio. Boa sorte no desenvolvimento do programa!