SaveBank

SIMULADOR DE CAIXA ELETRÔNICO EM C

UNICEPLAC
CENTRO UNIVERSITÁRIO

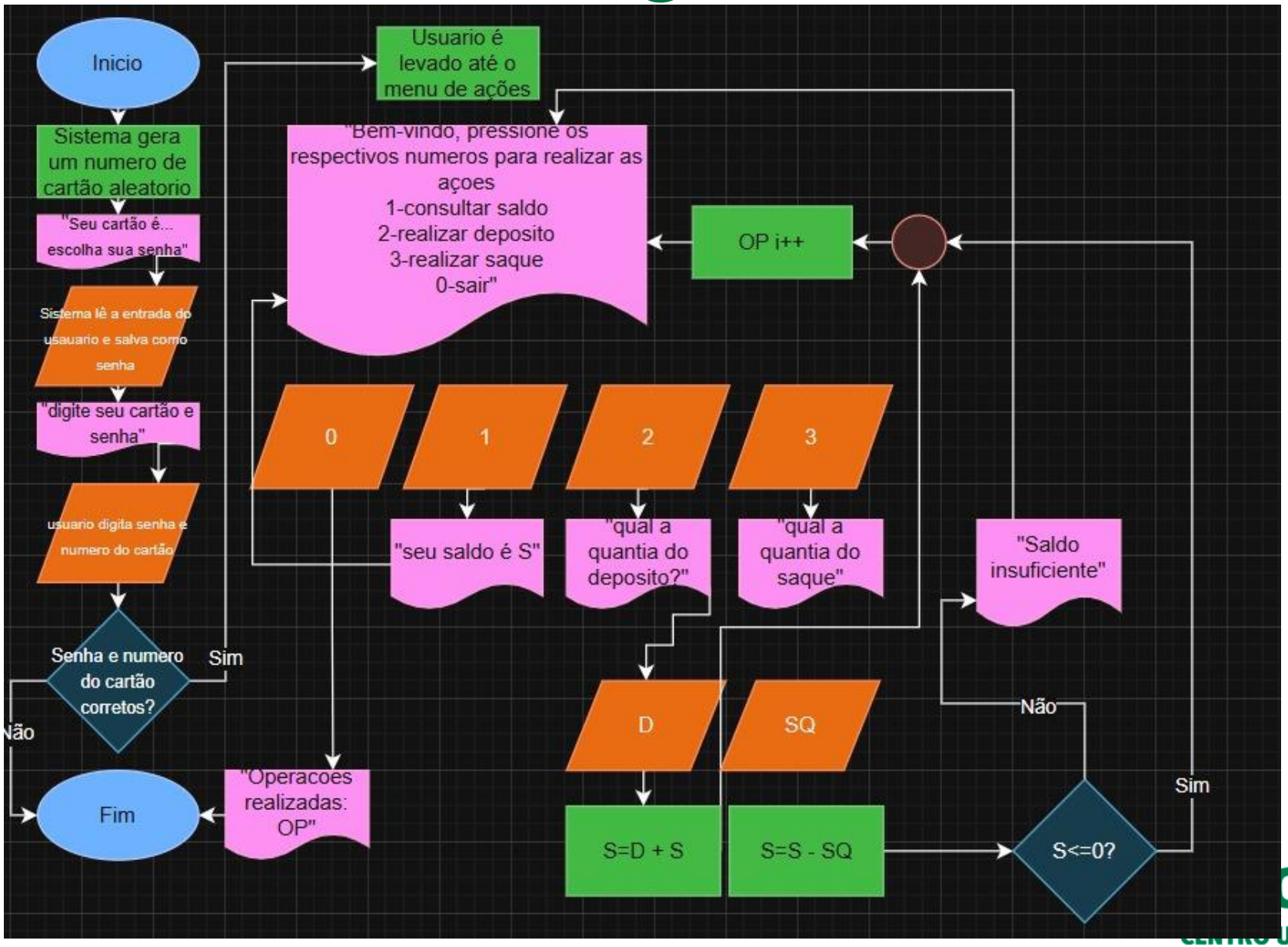
Integrantes e suas funções

Divisão De Módulos:

- Módulo 1-Verificação de login e senha: Italo andrade de sousa
- Módulo 2-Exibe o saldo atual da conta: Guilherme Kalil Pereira Nascimento
- Módulo3-Realiza saques com validação de saldo: Jullie Kessy Pereira de Carvalho
- Módulo 4-Permite adicionar valores à conta : Maria Clara Santos da Costa
- Módulo 5-Armazena e exibe operações feitas: Pedro Yan Barros Magalhães

UNICEPLAC
CENTRO UNIVERSITÁRIO

Fluxograma



CEPLAC
CENTRO UNIVERSITÁRIO

Modulo 1-Verificação de login e senha

```c
#include <stdlib.h>
#include <string.h>
#include <time.h>

// Declarações das funções dos outros arquivos
void visualizarSaldo(float saldo);
void realizarDeposito(float *saldo);
void realizarSaque(float *saldo);
void registrarOperacao(const char *tipo, float valor);
void exibirTotalOperacoes();

// Variáveis globais
int main() {
    int numero_cartao, cartao_digitado;
    char senha[20], senha_digitada[20];
    float saldo = 1000.0; // Saldo inicial para testes
    int opcao;

    // Gerador de número de cartão
    srand(time(NULL));
    numero_cartao = 100000 + rand() % 900000;
// criaçao de conta
    printf("Bem-vindo ao sistema de criacao de conta!\n");
    printf("Seu numero de cartao e: %d\n", numero_cartao);
    printf("Crie uma senha para o seu cartao: ");
    scanf("%19s", senha);

    printf("\nConta criada com sucesso!\n");
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Modulo 1-Verificação de login e senha

```
// verificação de Login
printf("\n=== Login ===\n");
printf("Digite o número do cartao: ");
scanf("%d", &cartao_digitado);
printf("Digite a senha: ");
scanf("%19s", senha_digitada);

if (cartao_digitado == numero_cartao && strcmp(senha, senha_digitada) == 0) {
    printf("Login bem-sucedido!\n");

    do {
        printf("\n== MENU ==\n");
        printf("1 - Consultar saldo\n");
        printf("2 - Realizar deposito\n");
        printf("3 - Realizar saque\n");
        printf("0 - Sair\n");
        printf("Escolha uma opcao: ");
        scanf("%d", &opcao);
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Modulo 1-Verificação de login e senha

```
//encaminha o usuario para as estaçoes de deposito/saque assim como disponibiliza a visualização do saldo e encerramento d sistemaa
            switch (opcao) {
                case 1:
                    visualizarSaldo(saldo);
                    break;
                case 2:
                    realizarDeposito(&saldo);
                    break;
                case 3:
                    realizarSaque(&saldo);
                    break;
                case 0:
                    printf("Saindo...\n");
                    break;
                default:
                    printf("Opção invalida.\n");
            }
//encerra o codigo quando o usario escolhe a opcao de sair
        } while (opcao != 0);

        exibirTotalOperacoes();
//encerra o codigo em caso de senha ou numero de cartao incorretos
    } else {
        printf("Numero do cartao ou senha incorretos.\n");
    }

    return 0;
} // Feito por Italo Andrade De Sousa
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Funcionalidades Implementadas

- **Criação de conta:**
  Geração automática de um número de cartão com rand() e criação de senha digitada pelo usuário via scanf.
- **Login do usuário:**
  Leitura do número do cartão e senha digitados. Validação por meio de comparação direta e com a função strcmp() para a senha.
- **Menu de opções interativo:**
  Exibe opções para consultar saldo, realizar depósito, realizar saque e sair. O menu permanece em execução com do-while até o usuário optar por sair.
- **Controle do fluxo de operações:**
  As opções escolhidas no menu são tratadas com switch-case, chamando diretamente as funções implementadas em outros módulos.
- **Encerramento com resumo:**
  Ao final da execução, exibe a quantidade total de operações realizadas com a função exibirTotalOperacoes().

UNICEPLAC
CENTRO UNIVERSITÁRIO

Principais Decisões Técnicas

- **Uso de switch-case** para garantir legibilidade e modularização nas chamadas de função.
- **Uso de do-while** para manter o menu ativo até o encerramento.
- **Integração com ponteiros** nos parâmetros das funções realizarDeposito() e realizarSaque() para permitir atualização direta do saldo declarado no main.

Demonstração Do Terminal

Geração de numero de cartão

```
Bem-vindo ao sistema de criação de conta!
Seu número de cartão é: 620782
Crie uma senha para o seu cartão:
```

Seleção da senha e criação da conta

```
Bem-vindo ao sistema de criação de conta!
Seu número de cartão é: 620782
Crie uma senha para o seu cartão: senha

Conta criada com sucesso!

=== Login ===
Digite o número do cartão:
```

Digitação da senha e numero do cartão

```
=== Login ===
Digite o número do cartão: 620782
Digite a senha: senha
```

Login realizado com senha e numero certos

```
Login bem-sucedido!

== MENU ==
1 - Consultar saldo
2 - Realizar depósito
3 - Realizar saque
0 - Sair
Escolha uma opção:
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Módulo 2 – Visualização do saldo

```c
#include <stdio.h>

void visualizarSaldo(float saldo) {  // funcao para exibir o saldo
    printf("Usuario: Pedro Kalil Pereira Sousa da Costa\n");
    printf("Seu saldo atual e: R$ %.2f\n", saldo); // Mostra o saldo atual do usuario

    // Estrutura de decisao para informar o usuario sobre seu saldo
    if (saldo >= 100) {
        printf("Voce tem saldo suficiente para realizar uma compra de alto valor.\n");
    } else if (saldo >= 50) {
        printf("Voce pode realizar compras de medio valor.\n");
    } else {
        printf("Seu saldo esta baixo(voce esta liso). Economize mais antes de gastar!.\n");
    }
//Feito por Guilherme Kalil
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Funcionalidades Implementadas

- **Exibição de identificação do usuário**:
- Irá imprimir no console o nome do usuário.

- **Apresentação do saldo atual**:
- Mostrar o valor do saldo utilizando apenas duas casas decimais, determinadas por ‘%.2f’.

- **Estrutura condicional em relação ao saldo**:
- Ao utilizar a estrutura ‘if else’, exibe uma mensagem correspondente ao valor atual do usuário.

UNICEPLAC
CENTRO UNIVERSITÁRIO

Principais Decisões Técnicas

- **Organização por função específica**:

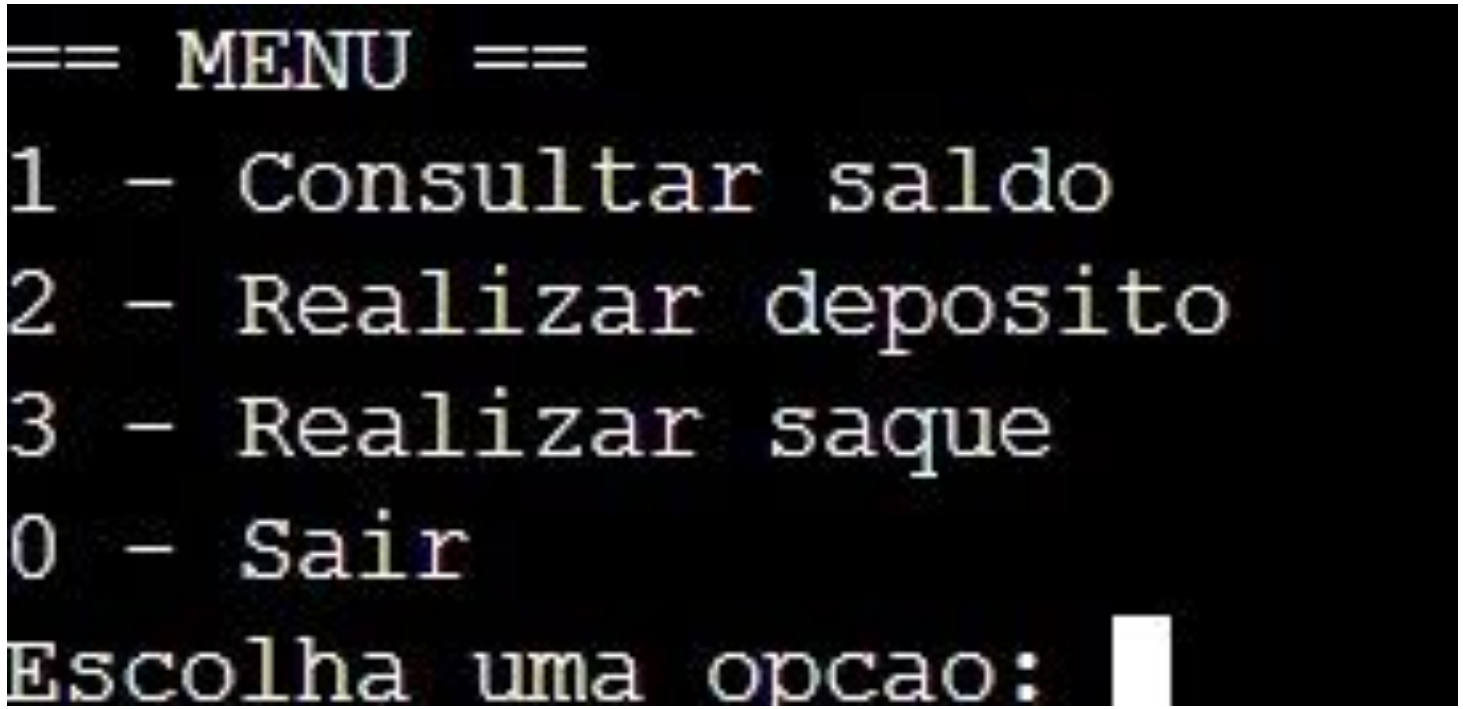
- Para que o saldo seja visualizado dentro de outras partes do programa, foi criada a função ‘visualizarSaldo’ somente para isso.

- **Receber o saldo como entrada**:

- A função recebe o valor do saldo como informação, ou seja, dentro do parâmetro ‘(float saldo)’, podendo ser reutilizada com outros valores.

UNICEPLAC
CENTRO UNIVERSITÁRIO

Demonstração no terminal



UNICEPLAC
CENTRO UNIVERSITÁRIO

Demonstração no terminal

```
Usuario: Pedro Kalil Pereira Sousa da Costa
Seu saldo atual e: R$ 1000.00
Voce tem saldo suficiente para realizar uma compra de alto valor.
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Demonstração no terminal

```
Usuario: Pedro Kalil Pereira Sousa da Costa
Seu saldo atual e: R$ 70.00
Voce pode realizar compras de medio valor.
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Demonstração no terminal

```
Usuario: Pedro Kalil Pereira Sousa da Costa
Seu saldo atual e: R$ 20.00
Seu saldo esta baixo(voce esta liso). Economize mais antes de gastar!.
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Módulo 3- Realiza saques com validação de saldo

```c
#include <stdio.h>

extern void registrarOperacao(const char *tipo, float valor);

void realizarSaque(float *saldo) {
    float valorSaque;
    printf("Digite o valor que deseja sacar: R$ ");
    scanf("%f", &valorSaque);// Lê o valor que o usuário digitou e armazena em valorSaque


    if (valorSaque <= 0) {
        printf("Valor de saque invalido.\n");// Verifica se o valor do saque é menor ou igual a zero
    } else if (valorSaque > *saldo) {// Verifica se o valor do saque é maior que o saldo disponível
        printf("Saldo insuficiente! Voce nao possui R$ %.2f.\n", valorSaque);
    } else {
        *saldo -= valorSaque;// Realiza o saque, subtraindo o valor do saldo
        registrarOperacao("Saque", valorSaque);
        printf("Saque de R$ %.2f realizado com sucesso!\n", valorSaque);
    }
}
// Feito por Jullie Kessy Pereira de Carvalho
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Funcionalidades Implementadas

•**Leitura do valor de saque**:
•O usuário informa quanto deseja sacar via *scanf*.
•**Validação do valor informado**:
•Verifica se o valor é maior que zero.
•Verifica se há saldo suficiente para realizar o saque.
•**Atualização do saldo**:
•Se as validações forem aprovadas, o valor do saque é **subtraído do saldo atual**.
•**Registro da operação**:
•Chama a função *registrarOperacao* para **registrar o saque** (mensagem + contagem).
•**Confirmação ao usuário**:
•Informa se o saque foi realizado ou se houve erro (valor inválido ou saldo insuficiente).

UNICEPLAC
CENTRO UNIVERSITÁRIO

Principais Decisões Técnicas

- **Uso de ponteiro para modificar o saldo**

A função *realizarSaque(float *saldo)* usa um ponteiro para que a modificação no valor do saldo reflita fora da função — ou seja, diretamente no contexto do *main*.

UNICEPLAC
CENTRO UNIVERSITÁRIO

Demonstração Do Terminal

```
Bem-vindo ao sistema de criacao de conta!
Seu numero de cartao e: 122805
Crie uma senha para o seu cartao: 123

Conta criada com sucesso!

=== Login ===
Digite o n·mero do cartao: 122805
Digite a senha: 123
Login bem-sucedido!

== MENU ==
1 - Consultar saldo
2 - Realizar deposito
3 - Realizar saque
0 - Sair
Escolha uma opcao: 3
Digite o valor que deseja sacar: R$
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Demonstração Do Terminal

```
Escolha uma opcao: 3
Digite o valor que deseja sacar: R$ 0
Valor de saque invalido.
```

```
Escolha uma opcao: 3
Digite o valor que deseja sacar: R$ 2000
Saldo insuficiente! Voce nao possui R$ 2000.00.
```

```
Escolha uma opcao: 3
Digite o valor que deseja sacar: R$ 20
Saque de R$ 20.00 realizado com sucesso!
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Modulo 4 –Realiza Depósitos

```c
#include <stdio.h>

extern void registrarOperacao(const char *tipo, float valor);

void realizarDeposito(float *saldo) {
    float deposito;
    printf("Informe o valor a ser depositado: ");
    scanf("%f", &deposito);

    if (deposito > 0) {
        *saldo += deposito;
        registrarOperacao("Deposito", deposito);
        printf("Deposito de R$ %.2f realizado com sucesso\n", deposito);
    } else {
        printf("O valor a ser depositado deve ser positivo!\n");
    }
}  // Ass: Maria Clara
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Funcionalidades Implementadas

- **Leitura do valor do deposito:**

Ele faz a Leitura do valor dado pelo usuário.

- **Validação do valor:**

Verifica se o valor lido e positivo .

- **Atualização do saldo:**

Atualiza o saldo somando o valor novo.

- **Registro da Operação:**

Chama a função registrarOperacao para registrar o deposito.

- **Confirmação do Deposito:**

Exibe pro usuário o valor que foi depositado.

Se o valor lido for menor ou igual a 0 ele exibe uma mensagem falando que o valor do deposito tem que ser positivo.

UNICEPLAC
CENTRO UNIVERSITÁRIO

Principais Decisões Técnicas

- **O uso de ponteiro para Modificar o saldo:**

void realizarDeposito(float *saldo) Permite que qualquer alteração que a função fizer em *saldo irá modificar diretamente o saldo original da conta, que está definido no Modulo 1.

UNICEPLAC
CENTRO UNIVERSITÁRIO

Demonstração do Terminal

```
Bem-vindo ao sistema de criacao de conta!
Seu numero de cartao e: 132039
Crie uma senha para o seu cartao: 1234

Conta criada com sucesso!

=== Login ===
Digite o numero do cartao: 132039
Digite a senha: 1234
Login bem-sucedido!

== MENU ==
1 - Consultar saldo
2 - Realizar deposito
3 - Realizar saque
0 - Sair
Escolha uma opcao: 2
Informe o valor a ser depositado: 200
Deposito de R$ 200.00 realizado com sucesso

== MENU ==
1 - Consultar saldo
2 - Realizar deposito
3 - Realizar saque
0 - Sair
Escolha uma opcao:
```

```
== MENU ==
1 - Consultar saldo
2 - Realizar deposito
3 - Realizar saque
0 - Sair
Escolha uma opcao: 2
Informe o valor a ser depositado: 0
O valor a ser depositado deve ser positivo!

== MENU ==
1 - Consultar saldo
2 - Realizar deposito
3 - Realizar saque
0 - Sair
Escolha uma opcao:
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Modulo 5-Historico de operações

```c
#include <stdio.h>
#include <string.h>

#define MAX_OPERACOES 100

typedef struct {
    char tipo[20];
    float valor;
} Operacao;

Operacao historico[MAX_OPERACOES];
int totalOperacoes = 0;

void registrarOperacao(const char *tipo, float valor) {
    if (totalOperacoes < MAX_OPERACOES) {
        strcpy(historico[totalOperacoes].tipo, tipo);
        historico[totalOperacoes].valor = valor;
        totalOperacoes++;
    }
}

void exibirTotalOperacoes() {
    printf("\n=== Resumo Final ===\n");
    printf("Total de operações realizadas: %d\n", totalOperacoes);
    printf("====================\n");
}//feita por pedro van
```

UNICEPLAC
CENTRO UNIVERSITÁRIO

Funcionalidades Implementadas

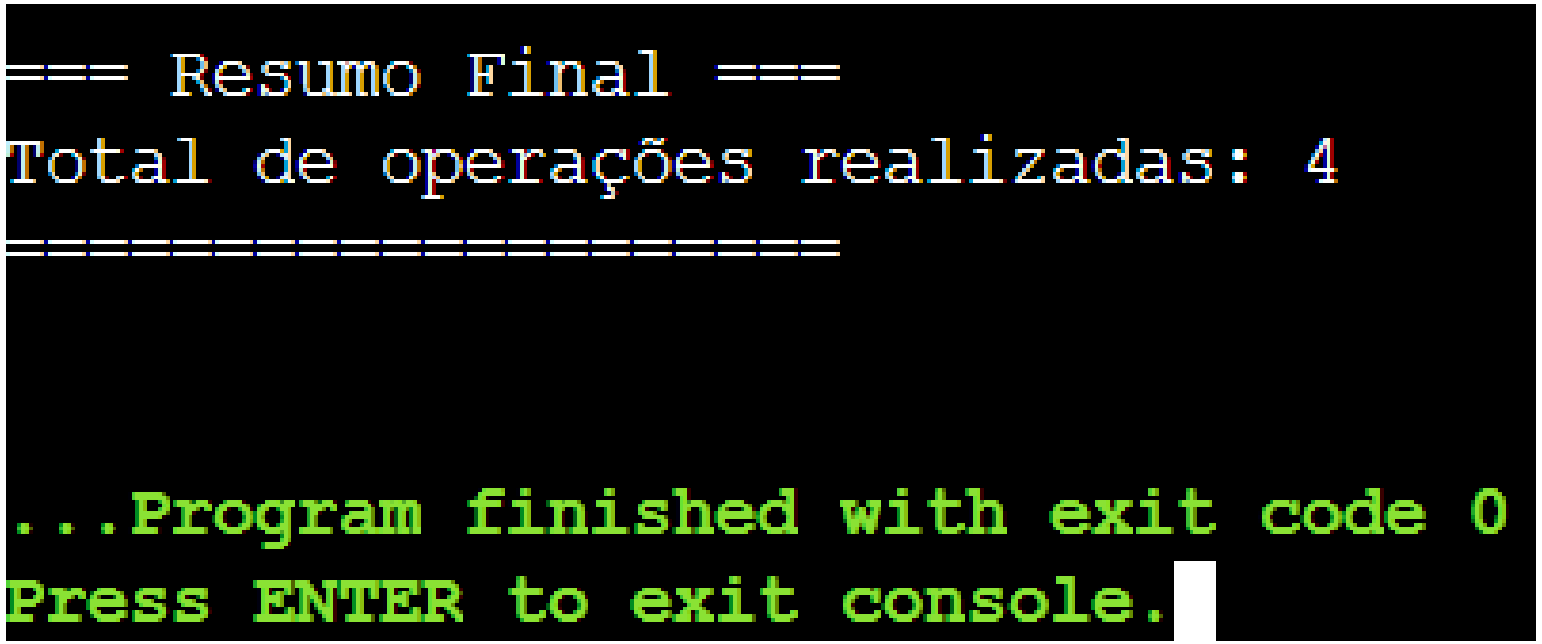
- **Estrutura de armazenamento:**
  Definição de uma struct Operacao contendo os campos tipo (descrição da operação) e valor (valor monetário da operação).
- **Registro das operações:**
  A função registrarOperacao(const char *tipo, float valor) armazena cada transação válida (depósito ou saque) no vetor historico[], incrementando o contador totalOperacoes.
- **Controle de limite de registros:**
  Aceita até 100 registros, definidos pela constante MAX_OPERACOES.
- **Exibição de resumo final:**
  A função exibirTotalOperacoes() imprime ao final do programa o número total de operações feitas pelo usuário.

UNICEPLAC
CENTRO UNIVERSITÁRIO

Principais Decisões Técnicas

- **Uso de vetor de structs (Operacao historico[100])** para organizar o histórico de forma estruturada e escalável.
- **Controle de índice com totalOperacoes** para gerenciar a posição atual de escrita no vetor.
- **Isolamento da lógica de registro em função própria**, o que permite reutilização do código em diferentes módulos (depósito e saque).

Demonstração Do Terminal



UNICEPLAC
CENTRO UNIVERSITÁRIO

Obrigado (a)!

JULLIE KESSY PEREIRA DE CARVALHO
ITALO ANDRADE DE SOUSA
PEDRO YAN BARROS MAGALHÃES
GUILHERME KALIL PEREIRA NASCIMENTO
MARIA CLARA SANTOS DA COSTA

UNICEPLAC
CENTRO UNIVERSITÁRIO