

Roteiro Aula Prática



**ALGORITMOS E TÉCNICAS
DE PROGRAMAÇÃO**

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: ALGORITMOS E TÉCNICAS DE PROGRAMAÇÃO

Unidade: 2 – ESTRUTURAS DE DECISÃO E REPETIÇÃO

Aula: 3 – ESTRUTURAS DE REPETIÇÃO DETERMINÍSTICAS

OBJETIVOS

- ✓ Compreender a definição e utilização de constantes e variáveis em linguagens de programação.
- ✓ Familiarizar-se com tipos de dados primitivos e compostos.
- ✓ Praticar operações e expressões em C.
- ✓ Explorar o uso de ponteiros e a manipulação de memória.

SOLUÇÃO DIGITAL

- Computador com um compilador C/C++ instalado juntamente com a IDE Code::Blocks.

LINK SOLUÇÃO DIGITAL (EXCETO ALGETEC): <https://www.codeblocks.org/downloads/>

Code::Blocks é um ambiente de desenvolvimento integrado de código aberto e multiplataforma. Ele foi desenvolvido em C++, usando wxWidgets. Sua arquitetura é orientada a plugin, de forma que suas funcionalidades são definidas pelos plugins fornecidos a ele. Code::Blocks é voltado para o desenvolvimento em C/C++.

PROCEDIMENTO/ATIVIDADE

ATIVIDADE PROPOSTA:

Desenvolver um programa em C que utilize variáveis primárias e compostas, ponteiros, constantes e operações matemáticas. Bem como funções de exibição de resultados.

PROCEDIMENTOS PARA A REALIZAÇÃO DA ATIVIDADE:

EXERCÍCIO 1: if, else if, else

Implementar um programa que leia um número inteiro e informe se ele é positivo, negativo ou zero.

Instruções:

Solicitar ao usuário que insira um número.

Usar a estrutura if, else if e else para verificar o valor do número e imprimir a mensagem correspondente.

Código:

```
#include <stdio.h>

int main() {
    int num;

    printf("Digite um número inteiro: ");
    scanf("%d", &num);

    if (num > 0) {
        printf("O número é positivo.\n");
    } else if (num < 0) {
        printf("O número é negativo.\n");
    } else {
        printf("O número é zero.\n");
    }

    return 0;
}
```

Execução e Testes:

Compilar e executar o programa.

Testar com diferentes valores (positivo, negativo e zero).

EXERCÍCIO 2: for

Implementar um programa que imprima os números de 1 a 10.

Instruções:

Usar a estrutura for para iterar de 1 a 10.

Imprimir cada número durante a iteração.

Código:

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

```
int main() {
    for (int i = 1; i <= 10; i++) {
        printf("%d\n", i);
    }

    return 0;
}
```

Execução e Testes:

Compilar e executar o programa.

Verificar a sequência de números impressos.

EXERCÍCIO 3: while

Implementar um programa que leia números inteiros até que o usuário digite zero, e então imprimir a soma desses números.

Instruções:

Usar a estrutura while para ler números até que o usuário digite zero.

Manter um acumulador para somar os números digitados.

Código:

```
#include <stdio.h>

int main() {
    int num, soma = 0;

    printf("Digite um número inteiro (0 para sair): ");
    scanf("%d", &num);

    while (num != 0) {
        soma += num;
        printf("Digite um número inteiro (0 para sair): ");
        scanf("%d", &num);
    }

    printf("A soma dos números é: %d\n", soma);
}
```

```
    return 0;
}
```

Execução e Testes:

Compilar e executar o programa.

Testar com diferentes sequências de números.

EXERCÍCIO 4: do-while

Implementar um programa que leia um número inteiro positivo e calcule seu fatorial.

Instruções:

Solicitar ao usuário que insira um número inteiro positivo.

Usar a estrutura do-while para calcular o fatorial do número.

Código:

```
#include <stdio.h>

int main() {
    int num, fatorial = 1;

    printf("Digite um número inteiro positivo: ");
    scanf("%d", &num);

    if (num < 0) {
        printf("Número inválido.\n");
    } else {
        int i = num;
        do {
            fatorial *= i;
            i--;
        } while (i > 0);

        printf("O fatorial de %d é: %d\n", num, fatorial);
    }

    return 0;
}
```

}

Execução e Testes:

Compilar e executar o programa.

Testar com diferentes valores positivos.

Para verificar o seu entendimento com relação aos conceitos de estruturas condicionais e de repetição, desenvolva um programa em C que leia notas de alunos, calcule a média, identifique a maior e menor nota, conte quantos alunos passaram (nota ≥ 6) e quantos não passaram (nota < 6).

Estrutura do programa:

Estrutura do Programa:

1. Declaração de variáveis.
2. Leitura das notas dos alunos usando um loop (while).
3. Cálculo da média das notas.
4. Identificação da maior e menor nota usando estruturas condicionais (if).
5. Contagem dos alunos que passaram e não passaram.
6. Impressão dos resultados.
7. Testar o Programa: Inserir diferentes notas para verificar se o programa calcula corretamente a média, maior e menor nota, e a contagem de alunos que passaram e não passaram. Testar com notas inválidas para garantir que o programa trate essas entradas corretamente. Testar a finalização do programa ao digitar -1.

CHECKLIST:

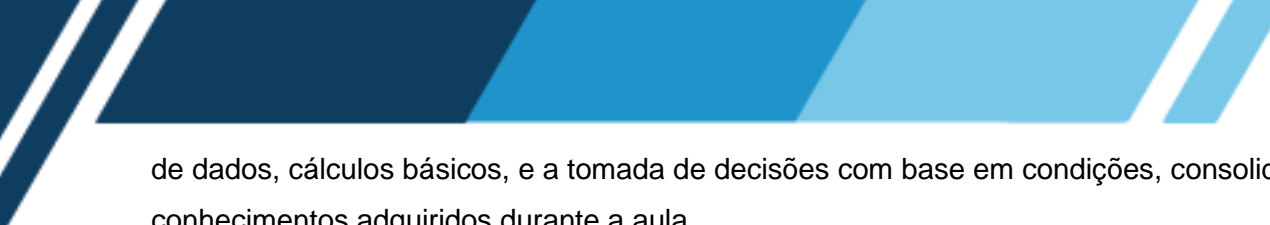
- If-else
- for
- while
- do-while

RESULTADOS

Para comprovar a realização da atividade, é necessário entregar um arquivo .c que contém o seu código da atividade.

RESULTADOS DE APRENDIZAGEM:

Espera-se que o aluno seja capaz de entender os conceitos fundamentais de estruturas condicionais (if, else if, else) e de repetição (while, do-while, for) por meio de práticas com leitura



de dados, cálculos básicos, e a tomada de decisões com base em condições, consolidando os conhecimentos adquiridos durante a aula.