**SESC**

**SENAC**

FELIPE ALVES DE FREITAS

**ALGORITMOS E LÓGICA DE PROGRAMAÇÃO**

**CUIABÁ**

**2024**

**SESC**

**SENAC**

FELIPE ALVES DE FREITAS

**ALGORITMOS E LÓGICA DE PROGRAMAÇÃO**

Trabalho sobre algoritmo e lógica da programação apresentado ao SENAC como requisito para a obtenção de NOTA

Orientador(a): Wanderson Timoteo

**CUIABÁ**

**2024**

**SUMÁRIO**

[ALGORITMO 4](#_Toc160613782)

[**Definição de Algoritmos e sua Importância na Programação:** 4](#_Toc160613783)

[**Lógica de Programação e sua Relação com Algoritmos:** 4](#_Toc160613784)

[ESTRUTURAS DE CONTROLE 4](#_Toc160613785)

[**Estruturas de Controle (Sequência, Seleção e Repetição):** 4](#_Toc160613786)

[**Exemplos de Algoritmos que Utilizam Estruturas de Controle:** 4](#_Toc160613787)

[TIPOS DE DADOS E VARIÁVEIS 5](#_Toc160613788)

[**Tipos de dados básicos:** 5](#_Toc160613789)

[**Exemplos de Declaração e Utilização de Variáveis:** 5](#_Toc160613790)

[FUNÇÕES E MODULARIZAÇÃO 6](#_Toc160613791)

[**Conceito de Função na Programação:** 6](#_Toc160613792)

[**Importância da Modularização na Escrita de Algoritmos:** 6](#_Toc160613793)

[**Exemplos de Funções Simples e sua Utilização em Algoritmos Mais Complexos:** 6](#_Toc160613794)

**ALGORITMO**

**Definição de Algoritmos e sua Importância na Programação:**

**Um algoritmo é um conjunto ordenado e finito de operações que, quando executadas, atingem um objetivo ou resolvem um problema específico. Na programação, os algoritmos são essenciais, pois representam a sequência lógica de passos que um computador deve seguir para realizar uma tarefa.**

**Lógica de Programação e sua Relação com Algoritmos:**

**A lógica de programação refere-se à maneira como os passos de um algoritmo são estruturados para alcançar uma solução. Ela está intimamente ligada aos algoritmos, pois envolve a ordenação e a organização dos comandos de maneira lógica e eficiente.**

****ESTRUTURAS DE CONTROLE****

**Estruturas de Controle (Sequência, Seleção e Repetição):**

**Sequência: Execução linear de comandos, um após o outro.**

**Seleção: Tomada de decisões, onde o programa escolhe entre diferentes caminhos com base em condições.**

**Repetição: Execução repetida de um conjunto de comandos enquanto uma condição é verdadeira.**

**Exemplos de Algoritmos que Utilizam Estruturas de Controle:**

**Sequência: Calcular a média de três números.**

**Seleção: Verificar se um número é positivo, negativo ou zero.**

**Repetição: Calcular a soma dos números de 1 a 100.**

****TIPOS DE DADOS E VARIÁVEIS****

**Tipos de dados básicos:**

**Inteiro: Números inteiros (ex: 5, -2).**

**Ponto Flutuante: Números com casas decimais (ex: 3.14, -0.5).**

**Booleano: Valores verdadeiro ou falso.**

**Exemplos de Declaração e Utilização de Variáveis:**

**# Declaração de variáveis em Python**

**idade = 25**

**altura = 1.75**

**temperatura = 36.5**

**is\_estudante = True**

**# Utilização em um algoritmo simples**

**if idade > 18 and is\_estudante:**

**print("Você é um estudante maior de idade.")**

**else:**

**print("Você não é um estudante maior de idade.")**

****FUNÇÕES E MODULARIZAÇÃO****

**Conceito de Função na Programação:**

**Uma função é um bloco de código que executa uma tarefa específica e pode ser chamado várias vezes ao longo do programa. Ela promove reutilização de código e modularização.**

**Importância da Modularização na Escrita de Algoritmos:**

**A modularização envolve dividir um programa em módulos independentes para facilitar a manutenção e compreensão. Funções são uma forma de modularização, permitindo a criação de blocos reutilizáveis de código.**

**Exemplos de Funções Simples e sua Utilização em Algoritmos Mais Complexos:**

**# Função simples para calcular o quadrado de um número**

**def calcular\_quadrado(numero):**

**return numero \*\* 2**

**# Utilização em um algoritmo mais complexo**

**def calcular\_soma\_quadrados(a, b):**

**soma = calcular\_quadrado(a) + calcular\_quadrado(b)**

**return soma**

**# Exemplo de chamada da função**

**resultado = calcular\_soma\_quadrados(3, 4)**

**print(resultado) # Saída: 25**