|  |
| --- |
| Programação Estruturada em C    **ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**    RELATÓRIO DE AULAS PRÁTICAS    Nome: Igor Pereira da Silva      RA: 2537577      Polo de matrícula: São Paulo - Anchieta      Local da realização da Aula Prática: UNIP Anchieta        Ano da postagem:2025  Professor aula pratica: Fábio Assis |

|  |  |  |
| --- | --- | --- |
| **Instituto de Ciências Exatas e Tecnologia** | **Disciplina:** Programação Estruturada em C **Título da Aula:** Programação Estruturada em C | **Relatório 1** |

**Orientações**: A atividade pode ser realizada individualmente ou em grupo, porém cada aluno deverá elaborar e entregar seu próprio relatório, com 1 a 2 páginas, contendo todas as partes solicitadas:

**Resumo Teórico:** Desenvolva uma explicação teórica, com base no conteúdo estudado, utilizando suas próprias palavras, sobre os seguintes tópicos:

* Estrutura de um programa em C;

Um programa em **C** possui uma estrutura básica composta por **diretivas de pré-processamento,** como #include <stdio.h> para incluir bibliotecas**; declaração de variáveis**, que define os dados que serão usados; e a **função principal** main(), que é o ponto de entrada do programa, onde ficam as instruções a serem executadas. Dentro de main(), são escritos os **comandos e operações**, como cálculos, leitura e exibição de dados. Ao final, normalmente retorna-se um valor inteiro (return 0;) indicando que o programa terminou corretamente.

* Tipos de dados básicos em C;

Em C, os tipos de dados básicos são usados para definir o tipo de informação que uma variável pode armazenar. O tipo **int** guarda números inteiros, o **float** armazena números com casas decimais, enquanto o **double** oferece números reais com maior precisão. O **char** é utilizado para representar caracteres individuais, e o tipo **void** indica ausência de valor, sendo comum em funções que não retornam nada.

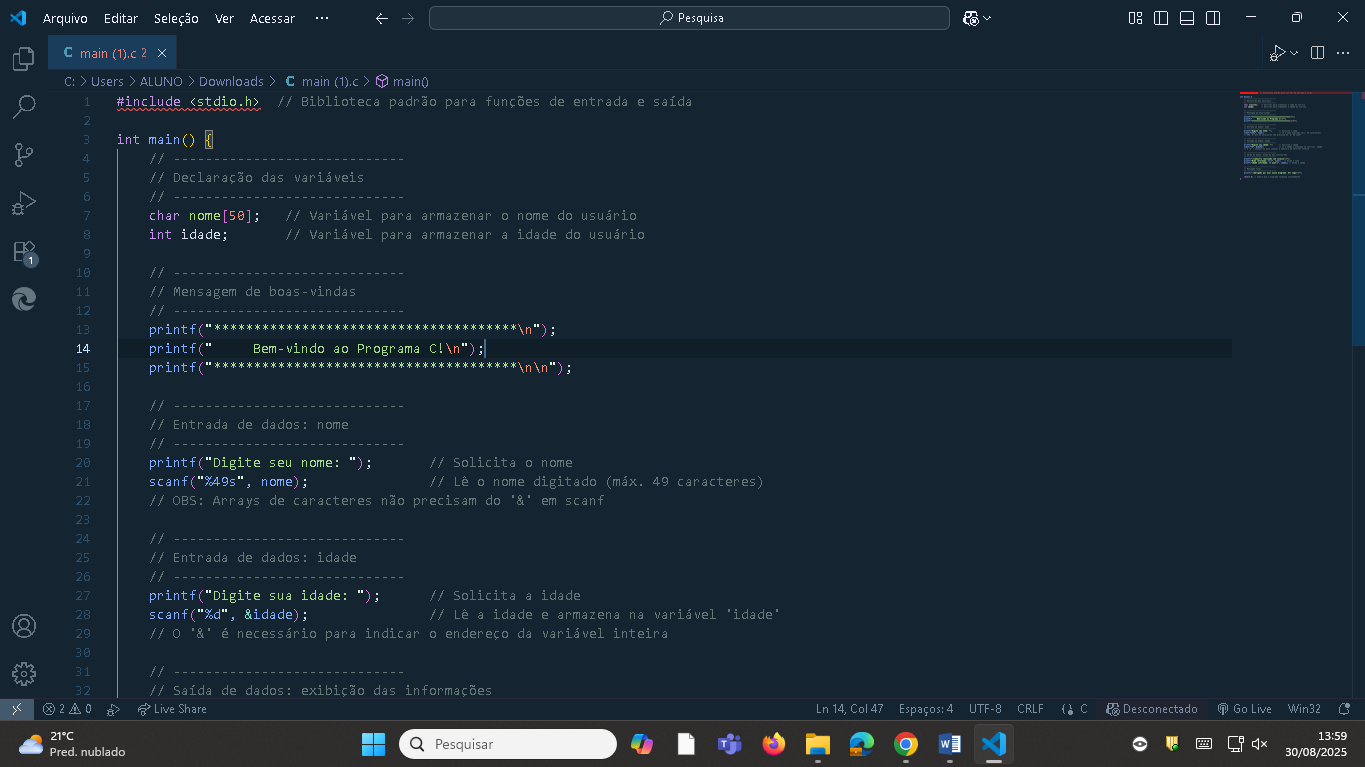
* Comandos básicos em C, como: printf() e scanf();  Identificadores e variáveis.

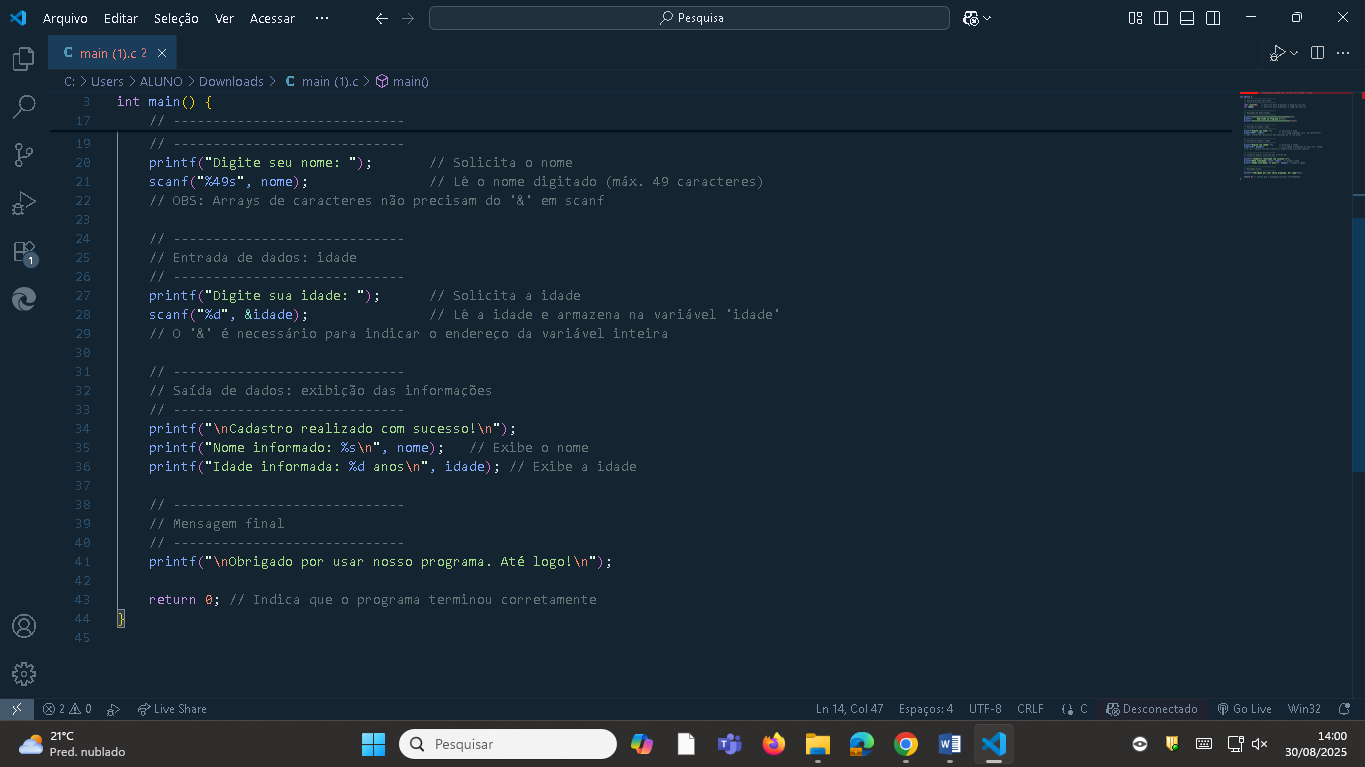
Em C, os **comandos básicos** incluem printf() e scanf(). O printf() é usado para exibir informações na tela, podendo mostrar textos, números ou resultados de operações. Já o scanf() serve para ler dados fornecidos pelo usuário, armazenando-os em variáveis; é necessário usar & antes do nome da variável para indicar seu endereço de memória.

**Identificadores** são nomes dados a variáveis, funções ou constantes, devendo começar com letra ou \_ e não podendo ser palavras reservadas do C. **Variáveis** são espaços de memória que armazenam temporariamente os dados do programa, e cada uma deve ter um tipo definido, como int, float ou char. Por exemplo: int idade; declara uma variável chamada idade do tipo inteiro.

**Código-Fonte Comentado:**

* Inserir o *código*-*fonte* *completo* do roteiro 1, atividade proposta 1.
* Modificar as mensagens do código.
* Comentar as principais linhas, ressaltando o uso de printf(), scanf() e por que variáveis são importantes.





### **Comentários detalhados**

1. **#include <stdio.h>**: permite usar funções de entrada e saída, como printf() e scanf().
2. **Variáveis**: armazenam dados do usuário temporariamente. Sem elas, o programa não poderia manipular ou exibir informações.
3. **printf()**: mostra mensagens no console, ajuda a orientar o usuário e exibe resultados.
4. **scanf()**: captura os dados digitados pelo usuário e os armazena nas variáveis correspondentes.
5. **Comentários (//)**: explicam cada parte do código, facilitando o entendimento e a manutenção futura.

**Referências:**

DEITEL, P. J.; DEITEL, H. M. C: Como Programar. 7. ed. São Paulo: Pearson, 2013.

TANENBAUM, A. S. Programação em C: Guia Prático. Rio de Janeiro: LTC, 2010.

TUTORIALSPOINT. C Programming Language. Disponível em: https://www.tutorialspoint.com/cprogramming/index.htm. Acesso em: 30 ago. 2025.

|  |  |  |
| --- | --- | --- |
| **Instituto de Ciências Exatas e Tecnologia** | **Disciplina:** Programação Estruturada em C  **Título da Aula:** Operadores Básicos em C | **Relatório 2** |

**Orientações**: A atividade pode ser realizada individualmente ou em grupo, porém cada aluno deverá entregar seu próprio relatório, com 2 a 3 páginas, contendo:

**Resumo Teórico:** Desenvolva uma explicação teórica, com base no conteúdo estudado, utilizando suas próprias palavras, sobre os seguintes tópicos:

* Operadores de atribuição;

Os operadores de atribuição são usados para **atribuir valores a variáveis**. O mais comum é o =, que coloca o valor do lado direito na variável do lado esquerdo. Existem também operadores combinados, que realizam uma operação e atribuem o resultado à mesma variável, como += (soma e atribuição), -= (subtração e atribuição), \*= (multiplicação e atribuição) e /= (divisão e atribuição).

* Operadores aritméticos;

Os operadores aritméticos permitem realizar **operações matemáticas básicas** entre valores numéricos. Os principais são: + (adição), - (subtração), \* (multiplicação), / (divisão) e % (módulo, que retorna o resto de uma divisão inteira). Eles são essenciais para cálculos em programas de C.

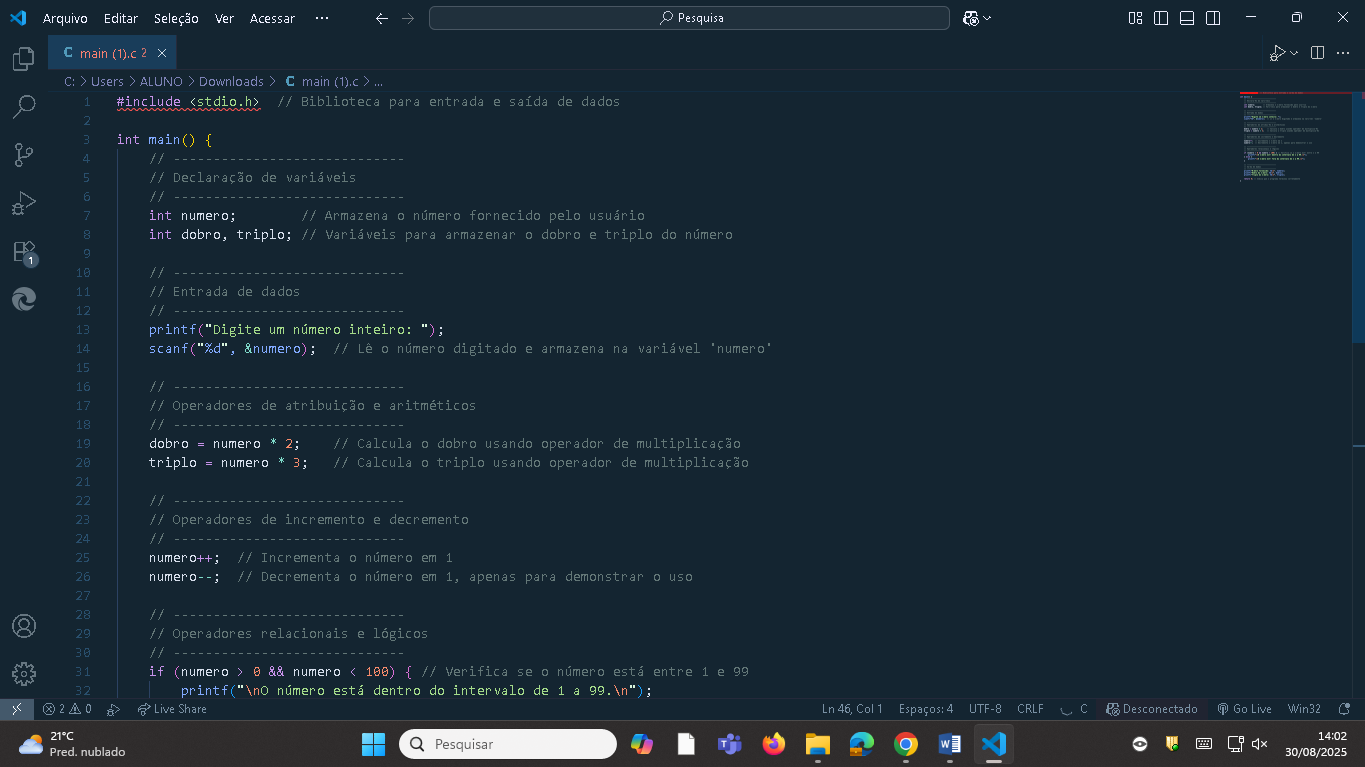
* Operadores relacionais e lógicos;

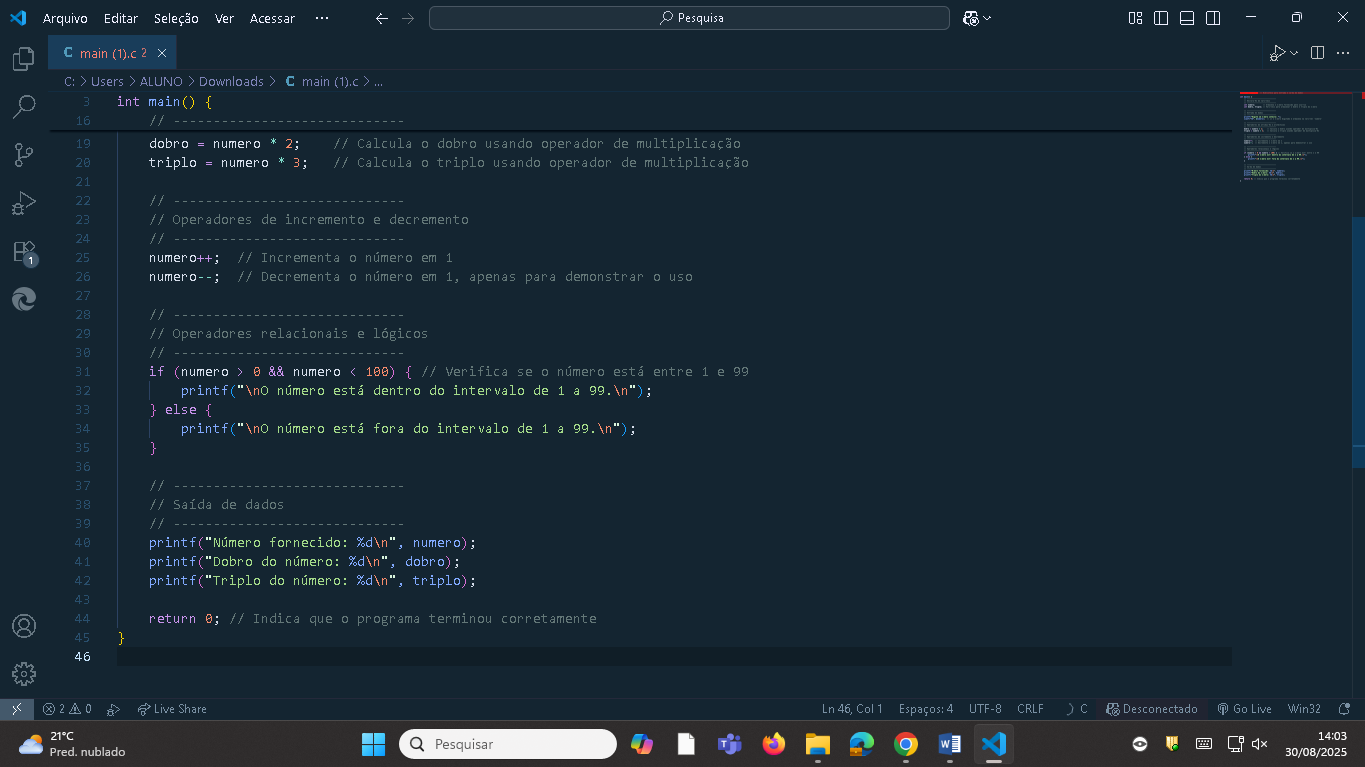
Os operadores **relacionais** comparam valores e retornam verdadeiro (1) ou falso (0). Exemplos: == (igual), != (diferente), > (maior que), < (menor que), >= (maior ou igual), <= (menor ou igual).  
Os operadores **lógicos** são usados para combinar ou inverter condições: && (E lógico), || (OU lógico) e ! (negação). Eles permitem construir decisões mais complexas, como em estruturas condicionais (if) ou laços (while).

* Operadores de incremento e decremento.

Esses operadores **alteram o valor de uma variável em uma unidade**. O ++ incrementa em 1, enquanto o -- decrementa em 1. Eles podem ser usados **antes** (pré-incremento ou pré-decremento) ou **depois** (pós-incremento ou pós-decremento) da variável, influenciando a ordem de avaliação em expressões.

**Códigos-Fontes Comentados:**

* Inserir o *código-fonte* completo do roteiro 2, atividade proposta 1.
* Comentar as principais linhas, ressaltando o uso de operadores.
* O aluno deve implementar pelo menos uma das ampliações sugeridas. **Conclusão:** Descreva como a aula contribuiu para o seu entendimento sobre operadores e explique a importância desses elementos na construção de programas. 



**Referência:**

 DEITEL, P. J.; DEITEL, H. M. C: Como Programar. 7. ed. São Paulo: Pearson, 2013.

 TANENBAUM, A. S. Programação em C: Guia Prático. Rio de Janeiro: LTC, 2010.

 KOCH, S. Programação em C: Fundamentos e Técnicas. 2. ed. Rio de Janeiro: Ciência Moderna, 2015.

 TUTORIALSPOINT. C Programming Language. Disponível em: https://www.tutorialspoint.com/cprogramming/index.htm. Acesso em: 30 ago. 2025.

 PROGRAMIZ. C Operators. Disponível em: https://www.programiz.com/c-programming/c-operators. Acesso em: 30 ago. 2025.

|  |  |  |
| --- | --- | --- |
| **Instituto de Ciências Exatas e Tecnologia** | **Disciplina:** Programação Estruturada em C **Título da Aula:** Operações e Controle de Fluxo | **Relatório 3** |

**Orientações**: A atividade pode ser realizada individualmente ou em grupo, porém cada aluno deverá entregar seu próprio relatório, com 2 a 3 páginas, contendo:

**Resumo Teórico:** Desenvolva uma explicação teórica, com base no conteúdo estudado, utilizando suas próprias palavras, sobre os seguintes tópicos:  Estrutura condicional: if;

* Estruturas condicionais: if-else.

**Códigos-Fontes Comentados:**

* Inserir o *código-fonte* completo do roteiro 3, atividade proposta 1.  Comentar as principais linhas, ressaltando o uso de if-else.

**Conclusão:** Descreva como a aula contribuiu para o seu entendimento sobre estruturas condicionais e explique a importância desses elementos na construção de programas. Cite possíveis aplicações práticas.

**Referências**: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

# CRITÉRIOS PARA AVALIAÇÃO

|  |  |  |
| --- | --- | --- |
| **Critério** | **Peso** | **Descrição** |
| **Clareza do**  **Resumo**  **Teórico** | 3,0 | Clareza e correlação correta entre conceitos. Conclusão bem definida. |
| **Organização e**  **Comentários do**  **Código** | 3,0 | O código deve estar indentado corretamente, usar nomes de variáveis adequados e conter comentários informativos (quando necessários). |
| **Funcionalidade do Código** | 2,0 | Execução sem erros, entrada e saída corretas. |
| **Criatividade e**  **Aprimoramentos** | 2,0 | Funcionalidades extras, simulações prints de tela. |

**Nota** **Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

|  |  |  |
| --- | --- | --- |
| **Instituto de Ciências Exatas e Tecnologia** | **Disciplina:** Programação Estruturada em C  **Título da Aula:** Estruturas de Repetição em C (Laços) | **Relatório 4** |

**Orientações**: A atividade pode ser realizada individualmente ou em grupo, porém cada aluno deverá entregar seu próprio relatório, com 2 a 3 páginas, contendo:

**Resumo Teórico:** Desenvolva uma explicação teórica, com base no conteúdo estudado, utilizando suas próprias palavras, sobre os seguintes tópicos:

* Laços de repetição: while, do-while e for.
* Comando break e comando continue.
* Switch-case.

**Códigos-Fontes Comentados:**

* Inserir o *código-fonte* completo do roteiro 4, atividade proposta 1 e 2.
* Comentar as principais linhas, ressaltando o uso de laços de repetição.

**Conclusão:** Descreva como a aula contribuiu para o seu entendimento dessas estruturas e explique a importância desses elementos na construção de programas.

**Referências**: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

# CRITÉRIOS PARA AVALIAÇÃO

|  |  |  |
| --- | --- | --- |
| **Critério** | **Peso** | **Descrição** |
| **Clareza do**  **Resumo**  **Teórico** | 3,0 | Clareza e correlação correta entre conceitos. Conclusão bem definida. |
| **Organização e**  **Comentários do**  **Código** | 3,0 | O código deve estar indentado corretamente, usar nomes de variáveis adequados e conter comentários informativos (quando necessários). |
| **Funcionalidade do Código** | 2,0 | Execução sem erros, entrada e saída corretas. |
| **Criatividade e**  **Aprimoramentos** | 2,0 | Funcionalidades extras, simulações prints de tela. |

**Nota** **Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

|  |  |  |
| --- | --- | --- |
| **Instituto de Ciências Exatas e Tecnologia** | **Disciplina:** Programação Estruturada em C  **Título da Aula:** Vetores (Arrays Unidimensionais) | **Relatório 5** |

**Orientações**: A atividade pode ser realizada individualmente ou em grupo, porém cada aluno deverá entregar seu próprio relatório, com 2 a 3 páginas, contendo:

**Resumo Teórico:** Desenvolva uma explicação teórica, com base no conteúdo estudado, utilizando suas próprias palavras, sobre os seguintes tópicos:  Vetores (arrays unidimensionais);  Operações com vetores.

**Códigos-Fontes Comentados:**

* Inserir o *código-fonte* completo do roteiro 5, atividade proposta 1.
* Comentar as principais linhas de código e propor mudanças no código.

**Conclusão:** Descreva como a aula contribuiu para o seu entendimento sobre vetores e explique a importância desses elementos na construção de programas. Cite possíveis aplicações práticas.

**Referências**: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

# CRITÉRIOS PARA AVALIAÇÃO

|  |  |  |
| --- | --- | --- |
| **Critério** | **Peso** | **Descrição** |
| **Clareza do**  **Resumo**  **Teórico** | 3,0 | Clareza e correlação correta entre conceitos. Conclusão bem definida. |
| **Organização e**  **Comentários do**  **Código** | 3,0 | O código deve estar indentado corretamente, usar nomes de variáveis adequados e conter comentários informativos (quando necessários). |
| **Funcionalidade do Código** | 2,0 | Execução sem erros, manipulação correta de vetor. |
| **Criatividade e**  **Aprimoramentos** | 2,0 | Funcionalidades extras, simulações prints de tela. |

**Nota** **Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

|  |  |  |
| --- | --- | --- |
| **Instituto de Ciências Exatas e Tecnologia** | **Disciplina:** Programação Estruturada em C **Título da Aula:** Matrizes (Arrays Bidimensionais) | **Relatório 6** |

**Resumo Teórico:** Desenvolva uma explicação teórica, com base no conteúdo estudado, utilizando suas próprias palavras, sobre os seguintes tópicos:

* Matrizes (arrays bidimensionais).
* Manipulação de matrizes: acesso e operações.
* o relatório deve conter prints da execução, caso possível, como forma de comprovação.

**Códigos-Fontes Comentados:**

* Inserir o *código-fonte* completo do roteiro 6, atividade proposta 1.  Comentar as principais linhas do código.

**Conclusão:** Descreva como a aula contribuiu para o seu entendimento sobre matrizes e explique a importância desses elementos na construção de programas. Cite possíveis aplicações práticas.

**Referências**: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

# CRITÉRIOS PARA AVALIAÇÃO

|  |  |  |
| --- | --- | --- |
| **Critério** | **Peso** | **Descrição** |
| **Clareza do**  **Resumo**  **Teórico** | 3,0 | Clareza e correlação correta entre conceitos. Conclusão bem definida. |
| **Organização e**  **Comentários do**  **Código** | 3,0 | O código deve estar indentado corretamente, usar nomes de variáveis adequados e conter comentários informativos (quando necessários). |
| **Funcionalidade do Código** | 2,0 | Execução sem erros, entrada e saída corretas. |
| **Criatividade e**  **Aprimoramentos** | 2,0 | Funcionalidades extras, simulações prints de tela. |

**Nota** **Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

|  |  |  |
| --- | --- | --- |
| **Instituto de Ciências Exatas e Tecnologia** | **Disciplina:** Programação Estruturada em C  **Título da Aula:** Funções em C | **Relatório 7** |

**Orientações**: A atividade pode ser realizada individualmente ou em grupo, porém cada aluno deverá entregar seu próprio relatório, com 2 a 3 páginas, contendo:

**Resumo Teórico:** Desenvolva uma explicação teórica, com base no conteúdo estudado, utilizando suas próprias palavras, sobre os seguintes tópicos:

* Funções em C.
* Chamada de funções e passagem de parâmetros.
* Escopo de variáveis.

**Códigos-Fontes Comentados:**

* Inserir o *código-fonte* completo do roteiro 7, atividade proposta 1.
* Comentar as principais linhas do código.
* Aplicar as melhorias propostas (opcional, mas recomendável).

**Conclusão:** Descreva como a aula contribuiu para o seu entendimento sobre operadores e explique a importância desses elementos na construção de programas.

**Referências**: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

# CRITÉRIOS PARA AVALIAÇÃO

|  |  |  |
| --- | --- | --- |
| **Critério** | **Peso** | **Descrição** |
| **Clareza do**  **Resumo**  **Teórico** | 3,0 | Clareza e correlação correta entre conceitos. Conclusão bem definida. |
| **Organização e**  **Comentários do**  **Código** | 3,0 | O código deve estar indentado corretamente, usar nomes de variáveis adequados e conter comentários informativos (quando necessários). |
| **Funcionalidade do Código** | 2,0 | Execução sem erros, entrada e saída corretas. |
| **Criatividade e**  **Aprimoramentos** | 2,0 | Funcionalidades extras, simulações prints de tela. |

**Nota** **Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

|  |  |  |
| --- | --- | --- |
| **Instituto de Ciências Exatas e Tecnologia** | **Disciplina:** Programação Estruturada em C  **Título da Aula:** Ponteiros e Manipulação de Arquivos | **Relatório 8** |

**Orientações**: A atividade pode ser realizada individualmente ou em grupo, porém cada aluno deverá entregar seu próprio relatório, com 2 a 3 páginas, contendo:

**Resumo Teórico:** Desenvolva uma explicação teórica, com base no conteúdo estudado, utilizando suas próprias palavras, sobre os seguintes tópicos:

* Definição ponteiros em C.
* Manipulação de arquivos (leitura e escrita em arquivos).

**Códigos-Fontes Comentados:**

* Inserir o *código-fonte* completo do roteiro 8, atividade proposta 1.  Comentar as principais linhas, ressaltando o uso de operadores.

**Conclusão:** Descreva como a aula contribuiu para o seu entendimento sobre ponteiros e arquivos e explique a sua importância na construção de programas.

**Referências**: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

# CRITÉRIOS PARA AVALIAÇÃO

|  |  |  |
| --- | --- | --- |
| **Critério** | **Peso** | **Descrição** |
| **Clareza do**  **Resumo**  **Teórico** | 3,0 | Clareza e correlação correta entre conceitos. Conclusão bem definida. |
| **Organização e**  **Comentários do**  **Código** | 3,0 | O código deve estar indentado corretamente, usar nomes de variáveis adequados e conter comentários informativos (quando necessários). |
| **Funcionalidade do Código** | 2,0 | Execução sem erros, entrada e saída corretas. |
| **Criatividade e**  **Aprimoramentos** | 2,0 | Funcionalidades extras, simulações e prints de tela. |

**Nota** **Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.