

TRABALHO N2 – PAC I

Criação de Site sobre Lógica e Programação Básica em C

Em grupos, criar um site educativo completo contendo conteúdos sobre lógica e programação básica em C. O objetivo é desenvolver conhecimentos técnicos, capacidade de pesquisa, habilidades de organização da informação e trabalho em equipe.

Conteúdos Obrigatórios:

1. Lógica de Programação

- Definição e importância.
- Como a lógica é usada para resolver problemas.
- Exemplos simples (como fazer um bolo, trocar uma lâmpada, etc).
- Fluxogramas (<http://www.flowgorithm.org>): símbolos básicos e exemplo prático (pode ser um fluxograma de um algoritmo simples).
- Exercícios (utilizando fluxogramas e rachacuca.com.br).

2. Introdução à Programação em C

- História e importância da linguagem C.
- Estrutura básica de um programa em C (inclusão de bibliotecas, função main, uso do return 0).
- Como compilar e executar um programa em C (podem citar compiladores como GCC, Dev C++, CodeBlocks).
- Sintaxe básica (comentários, indentação, boas práticas).

3. Tipos de Dados e Variáveis

- Tipos primitivos: int, float, char, double.
- Declaração e inicialização de variáveis.
- Constantes (`#define` e `const`).
- Regras de nomenclatura para variáveis.
- Exemplos de códigos (atividades resolvidas).

- Lista de atividades (atividades para a comunidade resolver).

4. Operadores e Expressões

- Operadores aritméticos: +, -, *, /, %.
- Operadores relacionais: ==, !=, <, >, <=, >=.
- Operadores lógicos: &&, ||, !.
- Ordem de precedência dos operadores.
- Exemplos de códigos (atividades resolvidas).
- Lista de atividades (atividades para a comunidade resolver).

5. Estruturas Condicionais

- if, else if, else.
- switch e case.
- Uso do break e default.
- Exemplos de códigos (atividades resolvidas).
- Lista de atividades (atividades para a comunidade resolver).

6. Estruturas de Repetição

- while.
- do while.
- for.
- Exemplos de códigos (atividades resolvidas).
- Lista de atividades (atividades para a comunidade resolver).

7. Entrada e Saída de Dados

- Uso da função printf para saída de dados.
 - Uso da função scanf para entrada de dados.
 - Formatação de saída com especificadores (%d, %f, %c, etc).
 - Exemplos de códigos (atividades resolvidas).
 - Lista de atividades (atividades para a comunidade resolver).
-

Requisitos Técnicos do Site:

- O site deve ser feito utilizando **HTML e CSS (é permitido utilizar outras tecnologias em conjunto)**.
- Cada grupo poderá escolher sua própria identidade visual (cores, fontes, layout), mas os conteúdos obrigatórios precisam estar presentes.
- Deve conter um menu com links para cada um dos tópicos listados.
- Os conteúdos devem ser organizados de forma que haja uma ordem lógica (ex: antes dos alunos estudarem tipos de dados, precisam primeiro ver introdução à C).
- **Exemplos de código C** devem estar destacados utilizando as tags `<pre>` e `<code>`.
- Deve conter uma página "Sobre", onde os grupos irão colocar o nome completo de todos os membros e qual foi a função de cada um no desenvolvimento.
- Cada seção deve conter:
 - Título do tópico.
 - Explicação teórica clara e objetiva.
 - Exemplos de código funcionais.
 - Lista de exercícios e desafios.

Cronograma:

- **Entrega e apresentação dos sites: 19/05.**
 - **Entregar a pasta compactada do projeto na atividade do Teams que será criada.**
 - **Em caso de utilização de outras tecnologias (como node e afins), enviar também o link do site (podem utilizar vercel, netlify, ou outros serviços gratuitos).**
- **Será escolhido um site para que seja utilizado nas aulas com a comunidade.**

Critérios de Avaliação:

- **Qualidade e clareza** do conteúdo.
- Correção e funcionamento dos **exemplos de código C**.
- Organização e facilidade de navegação do site.
- Estética e legibilidade (atenção à usabilidade e visual agradável).
- **Participação equilibrada** dos membros do grupo, cada membro deve ter um papel no desenvolvimento e pesquisa.
- Criatividade (exercícios, desafios ou curiosidades extras sobre C são bem-vindos).