

PROGRAMAÇÃO WEB

Domingos Henriques





Apresentação

Neste módulo, em um período de 5 dias, vamos explorar o desenvolvimento integral e a hospedagem de um website de página única (one page) utilizando o framework Bootstrap para CSS e JavaScript.





Apresentação

Ao longo deste curso, você terá a oportunidade de aprender, na prática, como construir um website funcional e moderno a partir do zero.





Apresentação




O programa será estruturado para proporcionar uma experiência prática e abrangente, capacitando você a criar, aprimorar e publicar seu próprio projeto web.





Apresentação

A abordagem envolverá conceitos práticos, exercícios de codificação, e discussões sobre as melhores práticas, preparando você para enfrentar desafios do mundo real no desenvolvimento web.





Programação Diária

01

Introdução e
Conceitos Básicos

02

Fundamentos da Web e
Tecnologias

03

Construção de Websites
com Frameworks

04

Continuação da
Construção do Website

05

Publicação e Hospedagem








Introdução

- Boas-vindas e Apresentação do Programa
- O que é Programação e por que é importante?
- Iniciação à Lógica de Programação
- Overview do Desenvolvimento Web
- Panorama Atual do Mercado de TI
- Saídas Profissionais e Áreas de Atuação






Objetivos

- Proporcionar uma introdução prática e abrangente ao mundo da programação e desenvolvimento web.
 - Desenvolver as habilidades essenciais necessárias para construir websites funcionais.
 - Familiarizar você com ferramentas, tecnologias e frameworks relevantes.
 - Capacitar você a criar e publicar um website real.
- 
- 
- 





Expectativas

- Participação ativa e envolvimento nas atividades práticas.
 - Compreensão aprofundada dos conceitos apresentados.
 - Aplicação prática dos conhecimentos adquiridos na construção de um website.
 - Interesse em aprender e explorar oportunidades de crescimento profissional.
- 



Ao final do estágio, você sairá com uma base sólida em programação web, projetos práticos concluídos e um entendimento claro das possíveis direções em suas carreiras profissionais.



Programação

Programação é o processo de criação de um conjunto de instruções que informam a um computador como executar uma tarefa específica.



Programação

Essas instruções, conhecidas como código, são escritas em linguagens de programação e são essenciais para o funcionamento de softwares, aplicativos, websites e sistemas em geral.



Linguagens, tecnologias de Programação e de gestão de dados



Programação

A programação está presente em praticamente todos os aspectos da tecnologia moderna.



Indústria automotiva

Em diferentes setores, a programação desempenha papéis cruciais. **Na indústria automotiva, por exemplo, ela é usada para controlar sistemas de produção automatizados.**



Medicina

Na medicina, a programação é utilizada para **modelar processos biológicos, analisar dados de pacientes** e até mesmo em cirurgias assistidas por computador.



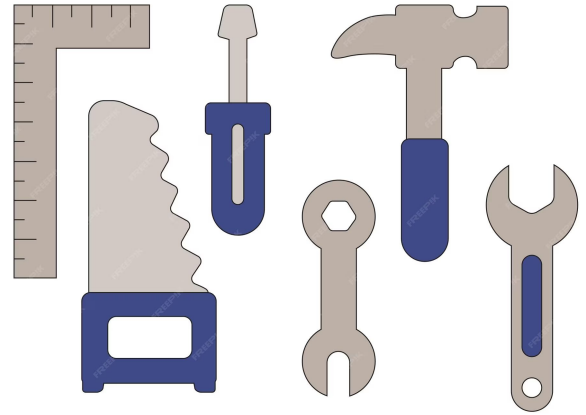
Finanças

No setor financeiro, é fundamental para análise de dados, tomada de decisões e automação de processos bancários.



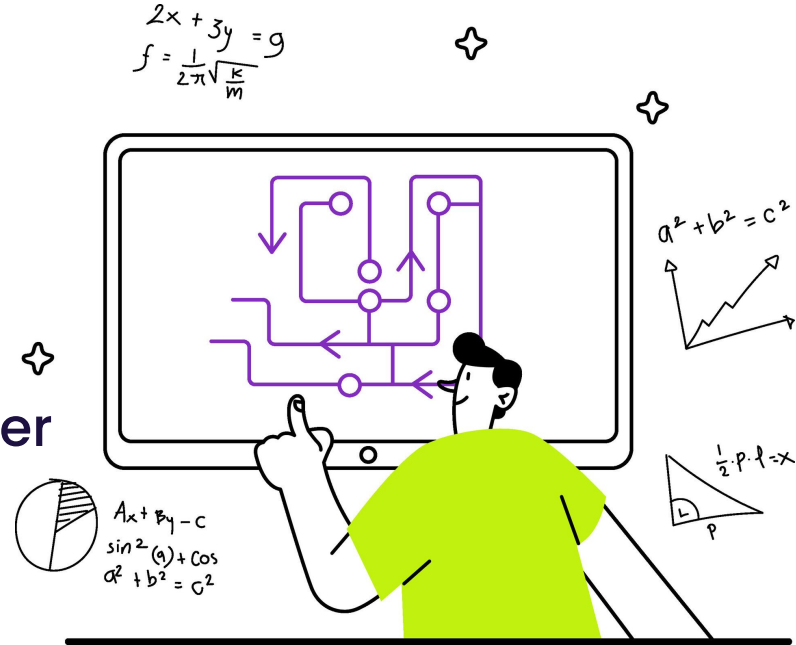
Importância

A programação é uma ferramenta poderosa para a resolução de problemas complexos.



Importância na Resolução de Problemas

Ao decompor um problema em etapas menores e expressá-lo em termos de algoritmos, os programadores podem desenvolver soluções eficientes.



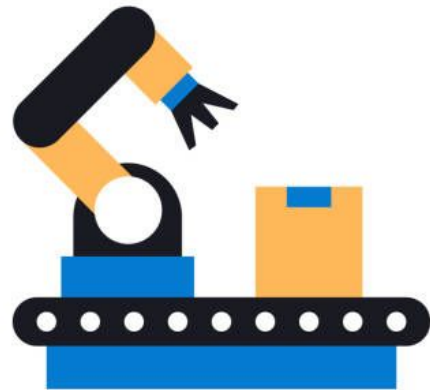
Importância na Resolução de Problemas

A habilidade de resolver problemas é uma das competências mais valiosas no campo da programação.



Automação de Tarefas

A automação de tarefas é outra área crítica. Através da programação, é possível criar scripts e softwares que executam tarefas rotineiras de forma automatizada, economizando tempo e reduzindo erros.



Importância na Automação de Tarefas

Por exemplo, em ambientes de negócios, a automação pode ser aplicada para processar grandes volumes de dados, gerar relatórios automaticamente e realizar tarefas repetitivas.






Exemplos Práticos de Aplicação de Programação em Diversas Áreas

1. Saúde:

Programação é usada para modelagem de dados genéticos, simulação de processos biológicos e desenvolvimento de softwares para monitoramento de pacientes.






Exemplos Práticos de Aplicação de Programação em Diversas Áreas

2. Educação:

Ferramentas de aprendizado online, sistemas de gestão escolar e plataformas educacionais são desenvolvidos com programação.






Exemplos Práticos de Aplicação de Programação em Diversas Áreas

3. Agricultura:

Sistemas de automação para irrigação, monitoramento de safras e otimização de logística agrícola são implementados com programação.






Exemplos Práticos de Aplicação de Programação em Diversas Áreas

4. Finanças:

Algoritmos financeiros, softwares de análise de risco e sistemas de processamento de transações são exemplos de aplicações na área financeira.






Exemplos Práticos de Aplicação de Programação em Diversas Áreas

5. Transporte:

Controle de tráfego, otimização de rotas e desenvolvimento de sistemas de transporte público são áreas onde a programação é essencial.








Exemplos Práticos de Aplicação de Programação em Diversas Áreas



6. Entretenimento:

Desenvolvimento de jogos, aplicativos de streaming, efeitos especiais em filmes e animações são impulsionados pela programação.





Estes exemplos ilustram como a programação está em todos os aspectos da sociedade moderna, proporcionando soluções inovadoras e eficientes em uma variedade de sectores.



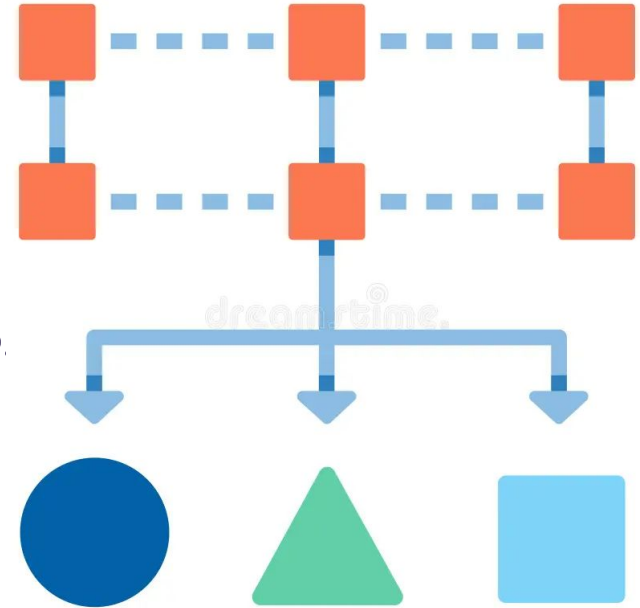
Lógica de programação

Lógica de programação é o modo como se escreve um programa de computador, um **algoritmo**.



Algoritmo

Um algoritmo é uma **sequência** de **passos** para se executar uma função.



Algoritmo – Exemplo

Um exemplo de algoritmo, fora da computação, é uma receita de bolo. Na receita, devem-se seguir os passos para o bolo ficar pronto e sem nenhum problema.



Lógica de programação

A lógica de programação é a base essencial para o desenvolvimento de algoritmos e, por conseguinte, para a programação de software.



Lógica de programação

Esses conceitos fundamentais **são as** **diretrizes** que ajudam os programadores a **criar soluções** lógicas e **eficientes para os** **problemas.**





Processo da lógica

1. Sequência:

A ordem em que as instruções são executadas. As ações ocorrem em sequência, uma após a outra.





Processo da lógica

2. Seleção (ou Tomada de Decisão):

Utilizada para escolher entre diferentes caminhos de execução com base em condições.

Por exemplo, "Se uma condição for verdadeira, faça X; caso contrário, faça Y."





Processo da lógica

3. Repetição (ou Iteração):

Permite que um bloco de código seja repetido várias vezes, com base em uma condição.

Por exemplo, "Enquanto uma condição for verdadeira, repita este bloco de código."






Processo da lógica



4. Modularização:

Divide o código em partes menores e mais gerenciáveis (funções ou procedimentos), facilitando a compreensão e manutenção do código.





Exercícios 1

Abaixo



A prática é crucial para o desenvolvimento da lógica de programação. Aqui estão alguns exercícios práticos que ajudarão a aprimorar o raciocínio lógico








1. Sequência:

- Escreva um algoritmo para calcular a média de três números.
- Desenvolva um código para exibir a soma dos números de 1 a 10.

2. Seleção (Tomada de Decisão):




- Crie um programa que determine se um número é par ou ímpar.
 - Escreva um código que classifique um aluno como "Aprovado" ou "Reprovado" com base em sua nota.
- 
- 
- 



3. Sequência:

- Implemente um algoritmo para calcular o fatorial de um número.
- Desenvolva um programa que imprima os números primos até 50.

4. Modularização:

- Crie uma função que receba dois números e retorne a soma.
 - Desenvolva um programa principal que utilize a função para calcular a média de três números.
- 
- 
- 

Exercícios 2

Abaixo



Algoritmos Simples para Compreensão Prática





1. Algoritmo de Troca:

- Descreva um algoritmo para trocar o valor de duas variáveis sem o uso de uma variável temporária.

2. Algoritmo de Busca Binária:

- Entenda como funciona um algoritmo de busca binária para encontrar um elemento em uma lista ordenada.
- 
- 
- 





3. Algoritmo de Ordenação (ex: Bubble Sort):



- Compreenda o passo a passo de um algoritmo de ordenação simples, como o Bubble Sort.

4. Algoritmo de Fatorial:

- Discuta um algoritmo eficiente para calcular o fatorial de um número usando iteração ou recursão.
- 
- 
- 



Esses exercícios e algoritmos proporcionarão uma base sólida para você, preparando-o para os desafios mais complexos que encontrará ao programar soluções reais.

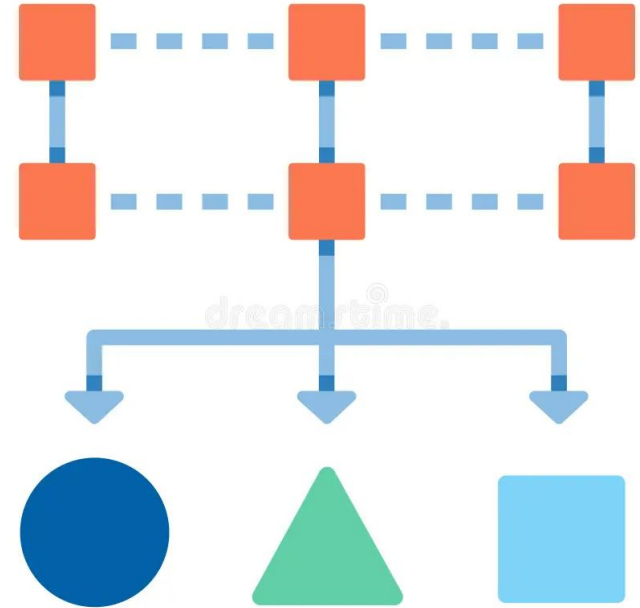


Anexos

Abaixo

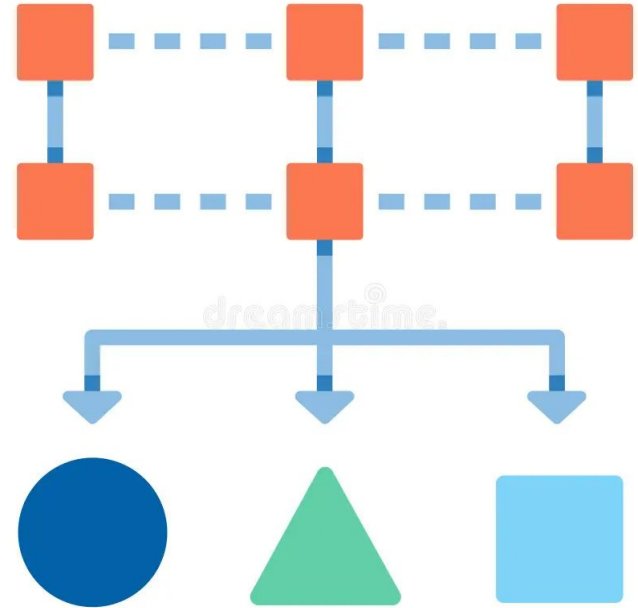
Algoritmo

Algoritmos são sequências estruturadas de instruções que realizam uma tarefa específica.



Algoritmo

A compreensão de algoritmos simples é essencial para a lógica de programação.



Fatorial

O fatorial de um número é o produto de todos os números inteiros positivos menores ou iguais a esse número. É representado pelo símbolo "!".



Fatorial

Por exemplo, o fatorial de 5
(denotado como 5!) é calculado da
seguinte forma:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$



Fatorial

Geralmente, o fatorial é utilizado em problemas combinatórios e probabilísticos, especialmente quando se trata de calcular o número de maneiras diferentes que um conjunto de elementos pode ser organizado.

Fatorial – Fórmula

A fórmula matemática para o fatorial de um número n é:

$$n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$$



Alguns valores conhecidos de fatoriais incluem

$$0! = 1$$

$$0! = 1$$

$$1! = 1$$

$$1! = 1$$

$$2! = 2 \times 1 = 2$$

$$2! = 2 \times 1 = 2$$

$$3! = 3 \times 2 \times 1 = 6$$

$$3! = 3 \times 2 \times 1 = 6$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

E assim por diante. O fatorial cresce rapidamente à medida que o número n aumenta.



JS exe01.js X

JS exe01.js > ...

```
1
2 // Exercício de calculo de média de 3 números
3 function calcularMedia(n1,n2,n3){
4
5     //calculo de média
6     let media = (n1 + n2 + n3)/3
7
8     // retorno do resultado
9     return media
10 }
11
12 // como usar a função acima
13 let resultado = calcularMedia(5, 5, 5)
14 console.log("A média é: ", resultado);
```




```
1
2 // Exercício soma dos números de 1 a 10
3 function soma1a10(){
4
5     // Inicialização da variavel de soma
6     let soma = 0
7
8     // loop para somar os números de 1 a 10
9     for (let i = 1; i <=10; i++) {
10         soma +=1
11     }
12
13     // saída do resultado
14     console.log("A soma dos números de 1 a 10 é: ", soma)
15 }
16
17
18 // chamada da função
19 soma1a10()
```

JS exe03.js X

JS exe03.js > ...

```
1
2 // Exercício determinar se um número é par ou ímpar
3 function parOuImpar(numero){
4
5     // verificar se e par ou ímpar
6     if(numero % 2 === 0){
7         console.log("O número ", numero, " é par...")
8     }else{
9         console.log("O número ", numero, " é ímpar...")
10    }
11 }
12
13 // chamada da função
14 parOuImpar(7)
```

JS exe04.js >  classificarAluno

```
1
2 // Aprovado ou reprovado
3 function classificarAluno(nota){
4
5     // verificar se o aluno aprovou ou reprovou
6     if(nota >= 5){
7         console.log("Aprovado")
8     }else{
9         console.log("Reprovado")
10    }
11 }
12
13 // chamada da função
14 classificarAluno(10)
```

JS exe05.js X

JS exe05.js > calcularFactorial

```
1
2 // Aprovado ou reprovado
3 function calcularFactorial(numero){
4
5     // inicialização do factorial
6     let factorial = 1
7
8     // calcular
9     for (let i = 1; i <= numero; i++) {
10         factorial *=i
11     }
12
13     // retornar resultado
14     return factorial
15 }
16
17 // mostrar resultado
18 let resultadoFactorial = calcularFactorial(5)
19 console.log("O factorial é: ", resultadoFactorial);
```

```
1
2 // Números Primos
3 function numeroPrimoAte50(){
4
5     // loop para verificar os números primos até 50
6     for (let numero = 2; numero <=50; numero++) {
7         let primo = true
8
9         // verificar se o numero é primo
10        for (let i = 2; i < numero; i++) {
11            if(numero % i === 0){
12                primo = false
13                break //não é primo não precisa continuar
14            }
15        }
16
17        // se é primo, mostra
18        if (primo) {
19            console.log(numero);
20        }
21    }
22 }
23
24 // chamada da função
25 numeroPrimoAte50()
```

JS exe07.js X

JS exe07.js > resultado

```
1
2 // Exercício de calculo de soma de 2 números
3 function calcularSoma(n1,n2){
4
5     //calculo da soma
6     let soma = (n1 + n2)/3
7
8     // retorno do resultado
9     return soma
10 }
11
12 // como usar a função acima
13 let resultado = calcularSoma(3, 7)
14 console.log("A média é: ", resultado);
```

Até a próxima aula