# CAPÍTULO 1 – INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO

## Conteúdo

A lógica de programação é a base para o desenvolvimento de qualquer software. Ela consiste na criação de uma sequência organizada de passos para resolver um problema ou executar uma tarefa.

Quando criamos um programa, precisamos informar ao computador exatamente o que deve ser feito. O computador não possui capacidade de interpretação semelhante à dos seres humanos, portanto todas as instruções precisam ser claras e organizadas.

O estudo da lógica de programação ajuda a desenvolver o raciocínio lógico, a análise de problemas e a busca por soluções eficientes.

## Exemplos

Exemplo 1 – Escovar os dentes

1. Pegar a escova.
2. Colocar pasta.
3. Escovar.
4. Enxaguar.

Exemplo 2 – Fazer café

1. Aquecer água.
2. Colocar pó.
3. Passar café.
4. Servir.

Exemplo 3 – Ligar o computador

1. Apertar Power.
2. Fazer login.
3. Abrir navegador.

## Atividade

1. O que é lógica de programação?
2. O que é um algoritmo?

3. Cite uma atividade que segue uma sequência lógica.
4. Qual a importância da lógica?
5. Onde utilizamos algoritmos?
6. Faça um resumo da aula.

# CAPÍTULO 2 – INTRODUÇÃO AO JAVASCRIPT

## Conteúdo

JavaScript é uma linguagem de programação utilizada principalmente para criar interatividade em páginas web. Atualmente é uma das linguagens mais utilizadas do mundo.

Uma grande vantagem do JavaScript é que ele pode ser executado diretamente pelo navegador, sem a necessidade de instalar programas complexos.

Nesta disciplina utilizaremos JavaScript puro para aprender os conceitos fundamentais da programação.

## Exemplos

Exemplo 1

Código:

```
alert("Olá Mundo");
```

Resultado:

Mensagem exibida na tela.

Exemplo 2

Código:

```
console.log("Primeiro programa");
```

Resultado:

Texto exibido no console.

Exemplo 3

Código:

document.write("Bem-vindo");

Resultado:

Texto exibido na página.

## Atividade

1. O que é JavaScript?
2. Onde ele é executado?
3. Qual sua principal vantagem?
4. O que faz o alert()?
5. O que faz o console.log()?
6. Faça um resumo.

# CAPÍTULO 3 – VARIÁVEIS

## Conteúdo

Variáveis são espaços utilizados para armazenar informações na memória do computador.

Podemos guardar nomes, idades, notas, preços e qualquer outro dado necessário para o programa.

Em JavaScript utilizamos principalmente let e const.

## Exemplos

Exemplo 1

let nome = "Maria";

Resultado:

Maria

Exemplo 2

let idade = 16;

Resultado:

16

Exemplo 3

```
const PI = 3.14;
```

Resultado:

```
3.14
```

## Atividade

1. O que é uma variável?
2. Qual a diferença entre let e const?
3. Crie uma variável chamada cidade.
4. Crie uma variável chamada idade.
5. O que acontece ao alterar uma constante?
6. Faça um resumo.

# CAPÍTULO 4 – TIPOS DE DADOS

## Conteúdo

Os tipos de dados indicam quais informações serão armazenadas em uma variável.

Os principais tipos são:

- String
- Number
- Boolean
- Null
- Undefined

Cada tipo possui características específicas e aplicações diferentes.

## Exemplos

```
let nome = "João";

let nota = 85;

let aprovado = true;
```

## Atividade

1. O que é um tipo de dado?

2. O que é String?
3. O que é Number?
4. O que é Boolean?
5. Crie exemplos de cada tipo.
6. Faça um resumo.

# CAPÍTULO 5 – OPERADORES MATEMÁTICOS

## Conteúdo

Os operadores matemáticos permitem realizar cálculos dentro dos programas.

Principais operadores:

- Soma
- Subtração
- Multiplicação

/ Divisão

% Resto da divisão

## Exemplos

10 + 5

Resultado: 15

8 * 7

Resultado: 56

10 % 3

Resultado: 1

## Atividade

1. Qual operador realiza soma?

2. Qual operador realiza multiplicação?
3. O que faz o operador %?
4. Faça uma soma.
5. Faça uma divisão.
6. Faça um resumo.

# CAPÍTULO 6 – ENTRADA E SAÍDA DE DADOS

## Conteúdo

Os programas precisam receber informações dos usuários e apresentar resultados.

Para isso utilizamos:

- prompt()
- alert()
- console.log()

## Exemplos

```
prompt("Digite seu nome");

alert("Bem-vindo");

console.log("Teste");
```

## Atividade

1. O que é entrada de dados?
2. O que é saída de dados?
3. Para que serve prompt()?
4. Para que serve alert()?
5. Para que serve console.log()?
6. Faça um resumo.

# CAPÍTULO 7 – IF E ELSE

## Conteúdo

As estruturas condicionais permitem que o programa tome decisões.

Utilizamos if para verificar condições e else para executar ações alternativas.

## Exemplos

if(idade >= 18)

if(nota >= 60)

if(temperatura > 30)

## Atividade

1. O que é uma condição?
2. Para que serve if?
3. Para que serve else?
4. Crie uma condição de aprovação.
5. Crie uma condição de maioridade.
6. Faça um resumo.

# CAPÍTULO 8 – OPERADORES RELACIONAIS E LÓGICOS

## Conteúdo

Permitem comparar valores e combinar condições.

Operadores:

## ==

<

=

<=

&&

||
!

## Exemplos

idade >= 18

nota >= 60 && frequencia >= 75

idade < 18

## Atividade

1. O que é comparação?
2. O que faz ==?
3. O que faz >?
4. O que faz &&?
5. O que faz ||?
6. Faça um resumo.

# CAPÍTULO 9 – SWITCH CASE

## Conteúdo

O switch é utilizado quando existem várias opções possíveis.

Ele facilita a organização do código.

## Exemplos

switch(dia)

switch(opcao)

switch(mes)

## Atividade

1. O que é switch?
2. Quando utilizá-lo?
3. O que faz case?

4. O que faz break?
5. O que faz default?
6. Faça um resumo.

# CAPÍTULO 10 – LAÇO FOR

## Conteúdo

O for é utilizado quando sabemos quantas vezes uma repetição será executada.

É muito utilizado para tabuadas, contagens e percorrer listas.

## Exemplos

```
for(let i=1;i<=10;i++)

for(let i=0;i<=100;i+=10)

for(let i=10;i>=1;i--)
```

## Atividade

1. O que é repetição?
2. Para que serve for?
3. O que é incremento?
4. Faça uma contagem até 10.
5. Faça uma tabuada.
6. Faça um resumo.

# CAPÍTULO 11 – LAÇO WHILE

## Conteúdo

O while executa repetições enquanto uma condição for verdadeira.

## Exemplos

```
while(i <= 10)

while(senha != "123")
```

while(vidas > 0)

## Atividade

1. O que é while?
2. Qual a diferença para for?
3. Quando utilizá-lo?
4. Crie uma repetição simples.
5. Explique uma aplicação.
6. Faça um resumo.

# CAPÍTULO 12 – FUNÇÕES

## Conteúdo

Funções são blocos de código reutilizáveis.

Elas ajudam a organizar programas.

## Exemplos

function saudacao()

function soma(a,b)

function dobro(numero)

## Atividade

1. O que é uma função?
2. Qual sua vantagem?
3. O que é parâmetro?
4. O que é retorno?
5. Crie uma função simples.
6. Faça um resumo.

# CAPÍTULO 13 – ARRAYS

## Conteúdo

Arrays armazenam vários valores em uma única variável.

## Exemplos

let frutas = ["Maçã","Banana","Uva"];

frutas[0]

frutas.push("Laranja")

## Atividade

1. O que é um array?
2. Qual sua utilidade?
3. Como acessar um elemento?
4. Como adicionar um elemento?
5. Crie um array de nomes.
6. Faça um resumo.

# CAPÍTULO 14 – PROJETOS SIMPLES

## Conteúdo

Aplicação prática dos conhecimentos estudados.

## Exemplos

- Calculadora
- Média Escolar
- Conversor de Temperatura

## Atividade

1. O que é um projeto?
2. Qual projeto você faria?
3. Como calcular média?
4. Como converter temperatura?
5. Qual projeto foi mais interessante?
6. Faça um resumo.

# CAPÍTULO 15 – PROJETO FINAL

## Conteúdo

Desenvolvimento de um sistema simples utilizando variáveis, operadores, condicionais, repetições e funções.

## Exemplos

- Cadastro de alunos
- Sistema de notas
- Jogo de adivinhação

## Atividade

1. Qual projeto você escolheu?
2. Quais recursos utilizou?
3. Qual foi a maior dificuldade?
4. O que aprendeu?
5. O projeto funcionou corretamente?
6. Faça uma conclusão sobre a disciplina.