

## **Exercício revisão lógica de programação – JavaScript**

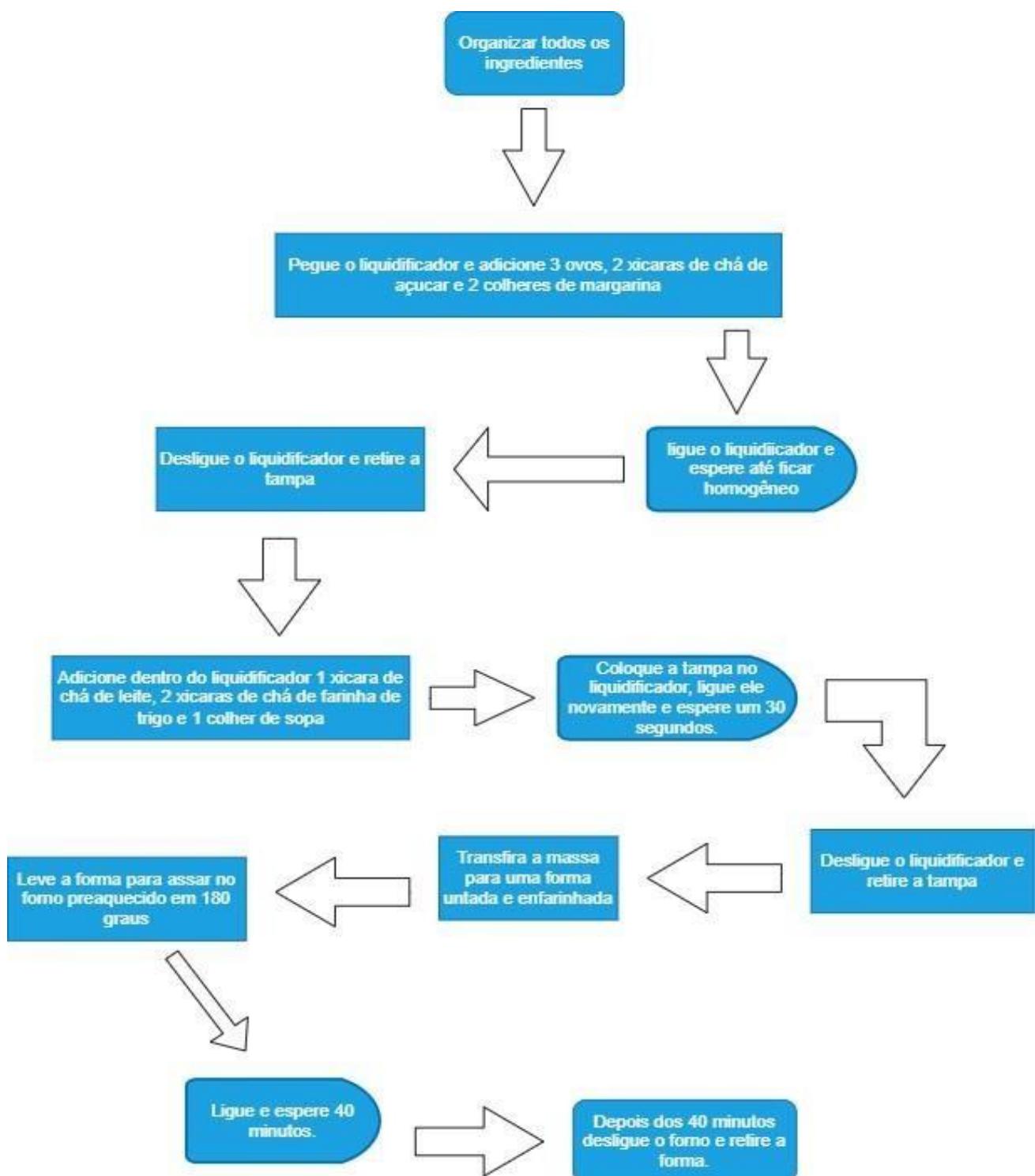
**Valdy Gabriel Cruz Da Silva; MATRICULA : 01667708**

### **Respostas**

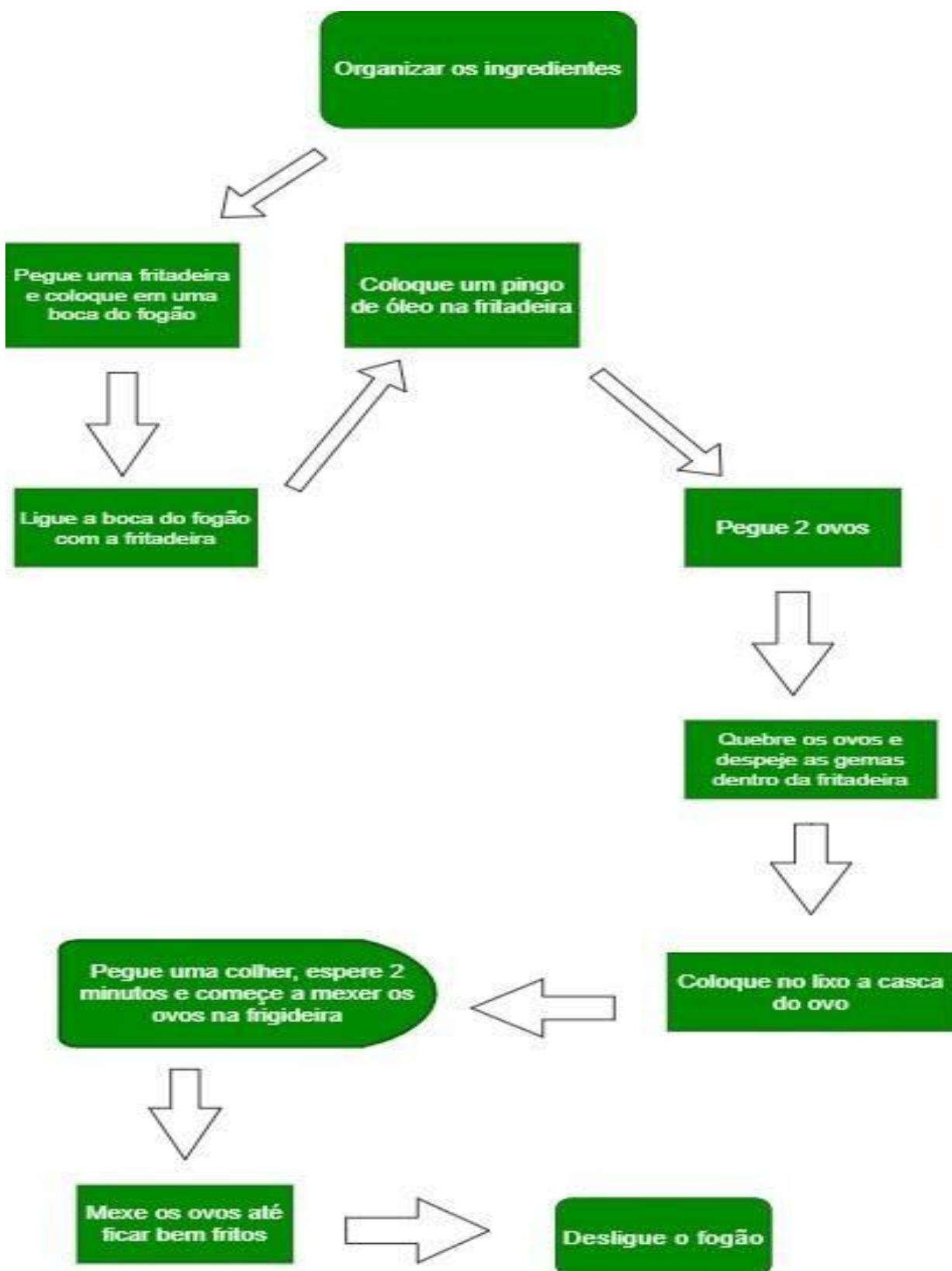
- 1) É basicamente uma instrução em sequencias organizadas para atingir um determinado objetivo.
- 2) São sequencias de instruções para resolver problemas. O pseudocódigo é apenas uma forma de escrever um algoritmo e o fluxograma é o mesmo que um algoritmo apenas diferenciando que o fluxograma é representado por gráficos.
- 3) Receita de bolo:
  - 1- Pegue o liquidificador e adicione 3 ovos, 2 xícaras de chá de açúcar e 2 colheres de margarina
  - 2- Ligue o liquidificador e espere até ficar homogêneo
  - 3- Desligue o liquidificador e retire a tampa
  - 4- Adicione dentro do liquidificador 1 xícara de chá de leite, 2 xícaras de chá de farinha de trigo e 1 colher de sopa
  - 5- Coloque a tampa no liquidificador, ligue ele novamente e espere um 30 segundos.
  - 6- Desligue o liquidificador e retire a tampa
  - 7- Transfira a massa para uma forma untada e enfarinhada
  - 8- Leve a forma para assar no forno preaquecido em 180 graus
  - 9- Ligue e espere 40 minutos.
  - 10- Depois dos 40 minutos desligue o forno e retire a forma.
- 4) Receita para fritar ovo
  - 1- Pegue uma fritadeira e coloque em uma boca do fogão
  - 4- Pegue 2 ovos
  - 5- Quebre os ovos e despeje as gema dentro da fritadeira
  - 6- Coloque no lixo a casca do ovo
  - 7- Pegue uma colher e espere 2 minutos

- 8- Espere 2 minutos e comece a mexer os ovos
- 9- Mexe os ovos até ficar bem fritos
- 10- Desligue o fogão

5) Fluxograma de Receita de bolo:



6) Fluxograma de fritar ovos:



7) Uma constante é um item de dados cujo já tem um valor predefinido, enquanto uma variável é um item de dados que o valor pode ser alterado durante a execução do programa.

8) Podem ter letras do alfabeto(minúsculas ou maiúsculas), dígitos (0 ...) e o caractere underline ( \_ ).

Podem começar com letra maiúscula.

Maiúsculas e minúsculas são diferentes (case-sensitive), ex:  
nomeCliente <> nomecliente .

Não usar hífen para iniciar o nome de variáveis, ex: -nomecliente.

Não pode ser uma palavra reservada, ex: for, if, int;

9) garotaGrande, bola1, carro\_Verde, moto\_Nova\$, bola2.

10. Tipos de dados é uma combinação de valores e de operações que uma variável pode executar.

11. String= 'texto', '001110101', '\$%``\$#%'

Number= 400, -300, 30.5

Boolean= false, true

Null= null

Undefined

12. var Boolean = true;

13. var char = 'Meu nome é Maria';

14. var char = 'carro21!';

15. Sim, porque a variável BOOLEAN só recebe verdadeiro ou falso.
16. Não, porque a variável NUMBER só recebe valores numéricos.
17. Sim, porque a variável NUMBER recebe valores numéricos.

18. Os arrays são estruturas que servem para organizar e guardar dados. Ex:

```
var motos = ['FZ15', 'Factor 150', 'Fazer 150'];  
var marcasDeCarros = ['Fiat', 'Renault', 'Ford'];  
var listaDeFrutas = ['Banana', 'Maçã', 'Abacaxi'];  
var sabores = ['Morango', 'Chocolate', 'Límao'];  
var numeros = ['55', '765', '545'];
```

19. Um objeto é uma entidade independente, com propriedades e tipos. Ex:

```
var minhaMoto = new Object();  
minhaMoto.fabricação = "Yamaha";  
minhaMoto.ano = "2024";  
minhaMoto.valor = "16.500";  
minhaMoto.placa = "ADC1565";  
minhaMoto.cor = "preta";
```

20. Operador é basicamente um elemento de um código que executa uma operação em um ou mais elementos que contêm valores.

Uma expressão são vários elementos de valor combinados com operadores que geram um novo valor.

21. + ( soma ), - ( subtração ), \* ( multiplicação ), / ( divisão ), % ( Modulo ) e ^ ( Exponenciação ). Ex: ( A + B ), ( A - B ), ( A \* B ), ( A / B ), ( A % B ).

22. Atribuição de adição:     $x += y \quad x = x + y$

Atribuição de subtração:     $x -= y \quad x = x - y$

Atribuição de multiplicação:       $x *= y$      $x = x * y$

Atribuição de divisão:       $x /= y$      $x = x / y$

Atribuição de resto:     $x \% = y$        $x = x \% y$

Atribuição exponencial:       $x **= y$        $x = x ** y$

23.    > maior que.    A > B

< menor que    A < B

$\geq$  maior ou igual a    A  $\geq$  B

$\leq$  menor ou igual a    A  $\leq$  B

= igual a    A == B

24.    AND (  $\&\&$  )

OU (  $\|$  )

NOT ( ! )

Ex: if ( x<5  $\&\&$  y>7 );

    if ( idade>18  $\&\&$  salario>1200 );

    if ( x=5  $\|$  y>6 );

    if ( !( x=6 );

    if ( x<3.5  $\&\&$  y>7 );

25. A estrutura de decisão verifica um comando e praticamente executa a decisão dele.

26. If : Permite executar comandos se a condição for verdadeira.

Else: Permite executar comandos se a condição for falsa.

If ( condição1 ) afirmação1

Else ( afirmação2

```
semaforo = "vermelho";

if (semaforo == "verde") {
    mensagem = "Pode passar";
} else {
    mensagem = "Pare";

//////////////////////////////



username = "usuario123";
password = "123456";

if (username == "usuario123" && password == "123456") {
    console.log("Usuário autenticado");
} else {
    console.log("Login e/ou senha incorretos");
//////////////////////////////



cpf = "123.456.678.01";

if (cpf.length == 14) {
    console.log("CPF válido");
} else {
    console.log("CPF inválido");

//////////////////////////////



var nomeUsuario = "Eduardo";
var mensagem = "";

if (nomeUsuario == "Eduardo") {
    mensagem = "Nome igual";
} else {
    mensagem = "Nome diferente";

//////////////////////////////



email = "dunppphys@gmail.com";
password = "123456";

if (email == "usuario123" && password == "123456") {
    console.log("Email autenticado");
} else {
    console.log("Email ou senha incorretos");
```

27. Switch permite executar um bloco de código diferente de acordo com cada opção (cada case) especificada.

```
switch(expressão){  
    case n1:  
        bloco de código 1  
        break;  
  
    case n2:  
        bloco de código 2  
        break;  
    default:  
        bloco de código 3  
}
```

Ex:

```
var alimento = "Gordura";  
  
switch (alimento) {  
    case "Proteína":  
        mensagem = "Carne, leite, aveia, amêndoas";  
        break;  
    case "Carboidrato":  
        mensagem = "Banana, batata doce, feijão, pão";  
        break;  
    default:  
        mensagem = "Cuidado com a alimentação!";  
  
//////////////////////////////////////////////////////////////////  
  
var tipoUsuario = "Gerente";  
  
switch (tipoUsuario) {  
    case "Admin":  
        mensagem = "*|*| Feliz Natal, chefe! |*|*";  
        break;  
    case "Gerente":  
        mensagem = "Boas festas, meu amigo!";  
        break;  
    default:  
        mensagem = "Boas festas!";
```

28. **For** determina que uma ação deve ser executada a partir de uma condição inicial até que seja encontrada outra que interrompa o laço.

```
for (inicialização; condição; incremento) {  
    // código que será repetido  
}
```

Ex:

```
var carros = [  
    { id: 1, modelo: 'Corsa', marca: 'Chevrolet', preco: 45000,  
     createdAt: 1534506085621 },  
    { id: 2, modelo: 'Punto', marca: 'Fiat', preco: 120000,  
     createdAt: 1534506085621 }  
]  
var total = 0  
for(var i = 0; i < carros.length; i++) {  
    total += carros[i].preco  
}
```

29. For in permite iterar sobre as propriedades enumeradas de um objeto de acordo com a ordem em que foram inseridos.

```
for ([índice] in [objeto])  
    declaração
```

ex:

```
var cores = [ 'Verde', 'Amarelo', 'Azul', 'Branco' ]  
  
for (const index in cores) {  
    console.log(index, cores[index])  
}  
//////////  
  
var pessoa = {  
    nome: 'Bruno',  
    idade: 25  
}  
  
for (const index in pessoa) {  
    console.log(`#${index} - ${pessoa[index]}`)  
}
```

30. For of permite percorrer um objetivo iterativo ( Array, Map, Set ) e executar um bloco de código.

```
for ([item] of [iteravel])
    declaracao
```

ex:

```
var numeros = [ 10, 11, 22, 23, 34, 45, 66, 17, 28, 93,
101 ]
```

```
for (const numero of numeros) {
    console.log(numero)
}
```

```
//////////////////////////////
```

```
var artigo = document.querySelectorAll('article > p')

for (const paragrafo of artigo) {
    paragrafo.classList.add("read")
}
```

31. Funções são blocos de construção fundamentais em JavaScript. Uma função é um procedimento de JavaScript - um conjunto de instruções que executa uma tarefa ou calcula um valor. Para usar uma função, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la.

Ex: // 1 função

```
function saudacao(nome) {
    return "Olá, " + nome + "! Bem-vindo ao nosso site.";
}
```

// 2 função

```
function areaTriangulo(b, a) {
    var area = (b * a) / 2
    return area;
}
```

// 3 função

```
function retornoInvestimento(c, i, t) {
    var m = c * (1 + i) ** t;
    return m;
}
```

// 4 função

```

function converteFparaC(F) {
    var celsius = (F - 32) * (5 / 9)
    return celsius;
}

// 5 função
document.write(inverteNum(875));

```

32. São os nomes dados aos atributos que uma função pode receber, além de definir quais argumentos são aceitos por uma função

```

function media(x, y, z)
{
    var media=(x+y+z)/3;
    document.write("A media é: " + media);
}
///////////////////////////////
function myFunction(x, y = 10) {
    return x + y;
}
///////////////////////////////

function somaReturn(320, 568){
    var resultado = valor1 + valor2;
    return resultado;
}

```

33. A estrutura de função apresentada indica que o parâmetro1 e o parametro2 vão ser retornados sendo multiplicados.

34. A sintaxe é basicamente a maneira como as palavras precisam estar sequenciadas corretamente para transmitir o seu significado. Já semântica é basicamente a analise do significado das instruções dadas pelo programa.

35. Devemos evitar código duplicado.

Devemos evitar criar nomes repetidos de variáveis.

Sim, poder criar pode, porem é melhor dar nomes descritivos para as variaveis para tornar o codigo mais legível e comprehensível.

