

Roteiro Aula Prática



**ALGORÍTMOS E LÓGICA
DE PROGRAMAÇÃO**

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

Unidade: 2 – ELEMENTOS DE ALGORITMOS

Aula: 3 – Estruturas de repetição

OBJETIVOS

- ✓ Interpretação dos requisitos e desenvolvimento correto do algoritmo abrangendo todas as possibilidades distintas na execução do programa;
- ✓ Aplicar na prática a estrutura de repetição: (“ENQUANTO FAÇA”) no desenvolvimento de algoritmos computacionais com a pseudolinguagem Portugol.

SOLUÇÃO DIGITAL

- Portugol WebStudio

LINK SOLUÇÃO DIGITAL: <https://portugol.dev/>

O Portugol WebStudio é uma ferramenta online de ensino e de aprendizagem de algoritmos baseada no Portugol (Português Estruturado), que é uma pseudolinguagem de programação para fins acadêmicos.

A ferramenta Portugol Web Studio não necessita de instalação, basta somente acessar por meio de uma conexão web o endereço (URL).

PROCEDIMENTO/ATIVIDADE

ATIVIDADE PROPOSTA:

Suponha que você está em uma missão para encontrar um tesouro escondido no meio de um labirinto muito perigoso. O labirinto é composto por uma série de corredores estreitos, bifurcações e salas ocultas. Sua missão é encontrar o caminho para o tesouro, evitando armadilhas e obstáculos ao longo do caminho.

Descrição do Labirinto:

- Existem caminhos que não têm saída, fazendo com que você precise voltar atrás e tentar outro caminho.
- Algumas portas podem levar de volta ao início do labirinto, obrigando-o a recomeçar sua busca.
- O tesouro está escondido em um local diferente a cada vez que o programa iniciar, tornando a exploração do labirinto desafiadora e imprevisível.

Boa sorte em sua busca pelo tesouro!

Com essa descrição, têm-se um cenário envolvente e desafiador para criação de um algoritmo usando o comando "Enquanto-Faça" em Portugol para guiar o explorador pelo labirinto até encontrar o tesouro.

PROCEDIMENTOS PARA A REALIZAÇÃO DA ATIVIDADE:

Aplicação dos fundamentos da lógica de programação e algoritmos, criação de variáveis e constantes, além de comandos da estrutura de repetição "Enquanto-Faça" em Portugol para resolução da atividade.

1. Configuração da Ferramenta:

- Utilizar a linguagem Portugol no Portugol Web Studio.

O estudante deverá utilizar a linguagem de programação "Portugol" por meio da ferramenta Portugol WebStudio acessando-o na url: <https://portugol.dev/>. Na ferramenta o estudante chegará nesta tela e deverá clicar no botão "Novo Arquivo":



- Descrever o cenário do labirinto e a missão do "explorador" em encontrar o tesouro.
Explicar o objetivo da atividade: utilizar o comando "Enquanto-Faça" para guiar o explorador pelo labirinto até encontrar o tesouro.
- 2. **Desenvolvimento do Programa:**
 - Utilizar a linguagem Portugol no Portugol Web Studio.
 - Criar variáveis para representar a posição do explorador no labirinto e outras informações relevantes.
 - a. Exemplo: definir o labirinto como uma matriz de 5 x 5 contendo = 25 espaços.
 - b. Definir o local do tesouro em um desses espaços. $\text{Labirinto}[4][4] = \text{tesouro}$.
 - Implementar um loop "Enquanto-Faça" para controlar a movimentação do explorador até encontrar o tesouro.
 - Definir uma posição inicial para o explorador no início do caminho, ou seja, da matriz:
 - a. $\text{posX} = 0$
 - b. $\text{posY} = 0$

```
// Enquanto não encontrar o tesouro
enquanto (labirinto[posX][posY] != tesouro)
{
    escreva("Explorador está na posição: (", posX, ", ", posY, ") ")
}
```

Definir uma posição iniciar

- Criar condições para determinar os movimentos possíveis do explorador (para frente, para trás, para a esquerda, para a direita) e verificar se ele encontrou o tesouro.

a. Baseado na posição da matriz 5 x 5 (labirinto), definir a próxima posição do jogador:

	0	1	2	3	4
0	Explor.				
1					
2					
3					
4					Tesouro

b. Como sugestão você pode definir o tesouro em um local específico da (matriz / labirinto), e fazer o explorador ir caminhando para encontrar o tesouro.

c. Como sugestão, os valores de cada endereço do labirinto podem ser:

- 2 = tesouro
- 1 = parede normal
- 0 = caminho livre
- Outro valor para punir o usuário e fazer ele voltar ao início do labirinto.
- O importante é criar obstáculos ao explorador

3. Geração Aleatória do Labirinto:

- Implementar um algoritmo para gerar aleatoriamente o layout do labirinto ou gerá-lo da mesma forma iniciando o jogador na posição [0][0].
- Garantir que o tesouro esteja escondido em um local diferente a cada vez que o programa iniciar.

4. Testes e Correções:

- Testar o programa com diferentes configurações de labirinto para garantir que o explorador encontre o tesouro corretamente.
- Corrigir quaisquer bugs ou erros de lógica que possam surgir durante os testes.

5. Documentação e Comentários:

- Documentar o código fonte, explicando a lógica por trás do movimento do explorador e as condições para encontrar o tesouro.
- Incluir comentários explicativos para facilitar a compreensão e manutenção do código.

Exemplo de comentários para a função que inicia o labirinto: // significa linha de comentário

```

funcao inicializarLabirinto()
{
    inteiro i, j

    // Inicializa o labirinto com paredes (1) e caminhos livres (0)
    para (i = 0; i < 5; i++)
    {
        para (j = 0; j < 5; j++)
        {
            labirinto[i][j] = parede
        }
    }

    // Define alguns caminhos livres
    labirinto[0][0] = caminhoLivre
    labirinto[0][1] = caminhoLivre
    labirinto[1][1] = caminhoLivre
}

```

6. Execução do Programa:

- Apresentar o programa, demonstrando como funciona a exploração do labirinto.

```

Explorador está na posição: (0, 0) Explorador está na posição: (0, 1) Explorador está na posição: (1, 1) Explorador está na posição: (2, 1) Explorador está na posição: (2, 2)
Explorador está na posição: (3, 2) Explorador está na posição: (4, 2) Explorador está na posição: (4, 3) Tesouro encontrado na posição: (4, 4)
Programa finalizado. Tempo de execução: 51 milissegundos

```

CHECKLIST:

- Analisar o cenário para criação do código-fonte do programa;
- Gerar aleatoriamente o layout do labirinto em cada execução do programa.
- Garantir que o tesouro esteja escondido em um local diferente a cada vez que o programa iniciar.
- Desenvolver o algoritmo em “Portugol” na ferramenta “Portugol WebStudio”;
- Executar testes e possíveis correções;
- Apresentar o programa e verificar seu correto funcionamento.

RESULTADOS

O estudante deve entregar um arquivo em PDF contendo toda a codificação do exercício documentada, ou seja, para cada trecho do código-fonte, o estudante deve anexar um texto explicando o que acontece quando o trecho do código é executado.

O arquivo deverá conter:

- Capa;
- Folha de rosto com os dados da disciplina e do aluno;
- Codificação completa dos exercícios;
- Referências bibliográficas (quando houver).

RESULTADOS DE APRENDIZAGEM:

O objetivo da atividade é a aplicação do comando “Enquanto-Faça” para guiar o explorador pelo labirinto até encontrar o tesouro, evitando armadilhas e obstáculos ao longo do caminho.

A atividade proporciona uma oportunidade valiosa para o desenvolvimento de habilidades de resolução de problemas, pensamento crítico e raciocínio lógico.