

Glossário

1. Interpretação

- **Definição:** Processo onde o programa **lê e executa** linha por linha.

Exemplo:

```
programa {  
    funcao inicio() {  
        escreva("Olá, mundo!") // Será interpretado e exibido no terminal  
    }  
}
```

2. Codar

- **Definição:** Escrever instruções em uma linguagem de programação.

Exemplo:

```
programa {  
    funcao inicio() {  
        cadeia nome = "Maria"  
        escreva("Olá, ", nome)  
    }  
}
```

3. IDE (Integrated Development Environment)

- **Definição:** Software que facilita a programação com ferramentas integradas.

Exemplos: VS Code, PyCharm, Thonny.

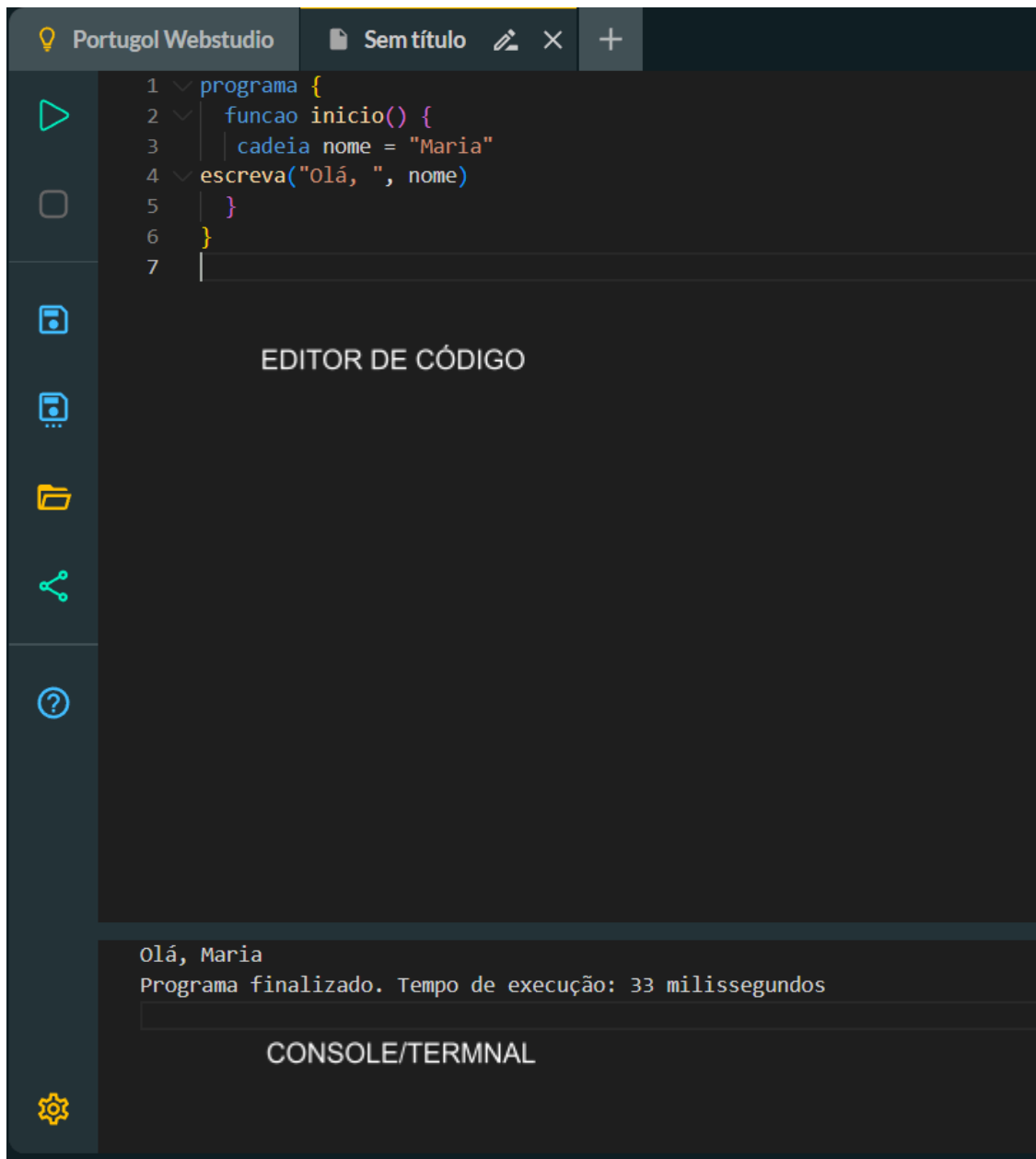


4. Áreas da IDE

Definição: IDE (Ambiente de Desenvolvimento Integrado) possui áreas principais:

- Editor de código
- Console/terminal - SAÍDA
- Área de mensagens
- Gerenciador de arquivos

Exemplo:



5. Identação

- **Definição:** Espaço no início da linha para organizar blocos de código.

Exemplo:

```
1 programa {  
2   funcao inicio() {  
3     cadeia nome = "Maria"  
4     escreva("Olá, ", nome)  
5   }  
6 }
```

6. Entrada/Saída

- **Definição:**
 - **Entrada:** Dados que o usuário fornece.
 - **Saída:** O que o programa exhibe.

Exemplo

```
programa {  
  funcao inicio() {  
    cadeia nome  
    escreva ("Digite seu nome: ", nome) //saída (exibida no terminal)  
    leia(nome) //entrada de dados no sistema (input do usuário)  
    escreva("Olá, ", nome) //saída (exibida no terminal)  
  }  
}
```

7. Comentário

- **Definição:** Texto ignorado pelo interpretador; serve para explicar o código.

Exemplo:

```
escreva("Olá, ", nome) //saída (exibida no terminal) -> ISSO É UM  
COMENTÁRIO
```

8. Biblioteca

- **Definição:** Conjunto de comandos prontos (nem sempre aplicável no Portugol puro).
- **Nota:** Em linguagens reais, como C ou Python, usamos **import** ou **include**.

Exemplo em C (equivalente):

```
#include <math.h>
```

9. Uso das Chaves {}

- **Definição:** Usadas em linguagens como C, Java e JavaScript para definir blocos.

Exemplo em C:

```
if (idade >= 18) {  
  printf("Maior de idade");  
}
```

10. Uso dos Parênteses ()

- **Definição:** Usados para chamar funções ou definir parâmetros.

Exemplo:

```
se (nota >= 7) { //FUNÇÃO SE (definindo parâmetro nota maior ou igual a  
7)  
  escreva("Aprovado") // FUNÇÃO ESCRIVA (definindo o que será exibido  
no terminal de saída)  
} senao {  
  escreva ("Reprovado")
```

```
}
```

11. Uso de Aspas Duplas (")

- Aspas duplas servem para **delimitar uma string** (texto) que será exibida ou usada no código.
- Em muitas linguagens, como Python, JavaScript, C, etc., é possível usar **aspas duplas** ou **simples**, mas as duplas são mais comuns em exibições no terminal.

Exemplo:

```
escreva("Aprovado, parabéns pela sua nota!")
```

12. Funções

- **Definição:** Bloco de código que realiza uma tarefa e pode ser reutilizado.
- Dentro de um código podemos ter várias funções, em Portugal, nós vimos o *leia e escreva*.

13. Variáveis

- **Definição:** Espaço nomeado na memória para armazenar valores.

Exemplo:

```
programa {  
    funcao inicio() {  
        inteiro nota, idade, dataNascimento //aceita números que não possuem  
casas decimais  
        real preco //permite a inclusão de número com casas decimais, como  
R$ 4,90 (quatro reais e noventa centavos)  
        caracter vogal, consoante //aceita apenas 1 caractere, como a ou b  
        cadeia nome //usado quando precisamos armazenar em uma variável, um  
texto ou uma quantidade grande de caracteres  
        logico decisao //determina se uma condição é verdadeira ou falsa.  
    }  
}
```

14. Estrutura do Código

- **Definição:** Organização lógica do programa (início, meio, fim; funções, condições, laços etc).
- Pense no código como peças separadas de um lego:

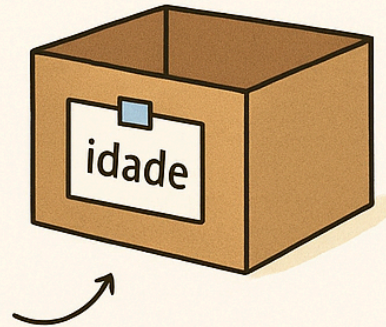


- Cada peça tem sua cor e função na montagem do brinquedo.
 - Declaração de variáveis:
 - **Analogia:** É como criar **caixinhas com etiquetas** para guardar informações.
 - Atribuição de variáveis:
 - **Analogia:** Agora que você tem a caixinha, é hora de **colocar algo dentro dela**.
 - Entrada/Saída:
 - **Analogia:**
 - **Entrada:** Você faz uma **pergunta ao usuário** e espera uma resposta (como um formulário).
 - **Saída:** Você **responde ou mostra algo** no terminal, como um bilhete que aparece na tela.
 - Chamada de funções:
 - **Analogia:** Uma função é como uma **máquina que faz uma tarefa**.
 - Você monta a máquina uma vez (define a função), e depois pode **apertar o botão para usá-la** sempre que quiser.
- No fim, é só juntar todas as peças e montar o brinquedo.

Programar é como organizar a vida com bilhetes e tarefas

1. DECLARAÇÃO DE VARIÁVEIS

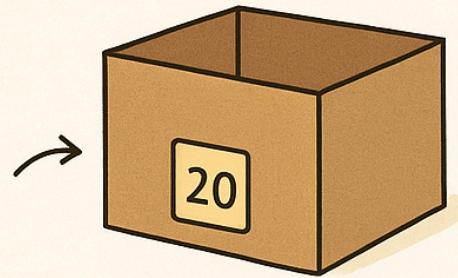
É como criar caixinhas com etiquetas



2. ATRIBUIÇÃO DE VARIÁVEIS

É como colocar algo na caixinha

`idade ← 20`



3. ENTRADA / SAÍDA (LEIA/ESCREVA)

Entrada `leia(nome)`
escrava("Olá, ", nome)
Saída: Mostrar um bilhete na tela



4. CHAMADA DE FUNÇÕES

Máquina `saudacao()`
Aperte o botão

