

Generation

BRASIL

1. Como é feita a **entrada** e **saida** de dados no C#?

- ☐ A leia()
e
escreva()
- ☐ B input()
e
print()
- ☒ C `Console.ReadLine();`
e
`Console.WriteLine();`

2. O que significa uma linguagem ser **fortemente tipada**?

- ☐ A Não possui tipos para definir suas variáveis
- ☐ B `Variáveis e constantes têm um tipo, assim como cada expressão que é avaliada como um valor`
- ☐ C Somente algumas variáveis possuem atributos tipados

3. **C#** possui 4 tipos de variáveis, sendo elas: Tipos internos, tipos personalizados, tipos literais e tipos genéricos.

- ☒ A **Verdadeiro**
- ☐ B Falso

4. A definição: Representam inteiros, valores de ponto flutuante, expressões booleanas, caracteres de texto, valores decimais e outros tipos de dados. É aplicada a qual tipo de variável?

- ☒ A **Tipos Internos**
- ☐ B Tipos Personalizados
- ☐ C Tipos Literais
- ☐ D Tipos genéricos

5. A definição: Utilizado para criar seus próprios tipos personalizados. É aplicada a qual tipo de variável?

- ☐ A Tipos Internos
- ☒ B **Tipos Personalizados**
- ☐ C Tipos Literais
- ☐ D Tipos genéricos

6. A definição: Você pode especificar como um literal numérico deve ser digitado anexando uma letra ao final do número. É aplicada a qual tipo de variável?

- | | | | |
|----------------------------|----------------|----------------------------|----------------------|
| <input type="checkbox"/> A | Tipos Internos | <input type="checkbox"/> B | Tipos Personalizados |
| <input type="checkbox"/> C | Tipos Literais | <input type="checkbox"/> D | Tipos genéricos |

7. A definição: Um tipo pode ser declarado com um ou mais parâmetros de tipo que servem como um espaço reservado para o tipo real (o tipo concreto). É aplicada a qual tipo de variável?

- | | | | |
|----------------------------|----------------|----------------------------|----------------------|
| <input type="checkbox"/> A | Tipos Internos | <input type="checkbox"/> B | Tipos Personalizados |
| <input type="checkbox"/> C | Tipos Literais | <input type="checkbox"/> D | Tipos genéricos |

8. É possível declarar variáveis de diferentes formas no C#, quais das alternativas abaixo está declarando corretamente a variável nome?

- | | | | |
|----------------------------|-----------------------------------|----------------------------|-----------------------------------------|
| <input type="checkbox"/> A | string nome; | <input type="checkbox"/> B | string nome = "Gustavo Boaz"; |
| <input type="checkbox"/> C | string nome = Console.ReadLine(); | <input type="checkbox"/> D | string nome = "Gustavo" + " " + "Boaz"; |

9. Os Operadores (+, -, *, /, %), são operadores:

- | | | | |
|----------------------------|------------|----------------------------|--------------|
| <input type="checkbox"/> A | Lógico | <input type="checkbox"/> B | Relação |
| <input type="checkbox"/> C | Atribuição | <input type="checkbox"/> D | Aritiméticos |

10. Os Operadores (=, +=, -=, *=, /=, %=), são operadores:

- | | | | |
|----------------------------|------------|----------------------------|--------------|
| <input type="checkbox"/> A | Lógico | <input type="checkbox"/> B | Relação |
| <input type="checkbox"/> C | Atribuição | <input type="checkbox"/> D | Aritiméticos |

11. Os Operadores (==, !=, <, <=, >, >=), são operadores:

- | | | | |
|----------------------------|------------|----------------------------|--------------|
| <input type="checkbox"/> A | Lógico | <input type="checkbox"/> B | Relação |
| <input type="checkbox"/> C | Atribuição | <input type="checkbox"/> D | Aritiméticos |

12. Os Operadores (&&, ||), são operadores:

- | | | | |
|----------------------------|------------|----------------------------|--------------|
| <input type="checkbox"/> A | Lógico | <input type="checkbox"/> B | Relação |
| <input type="checkbox"/> C | Atribuição | <input type="checkbox"/> D | Aritiméticos |

13. Quais as finalidades de escrever uma boa **nomenclatura** ao desenvolver um código?

A

Criam uma aparência consistente para o código, para que os leitores possam se concentrar no conteúdo e não no layout.

B

Permitem que os leitores entendam o código com mais rapidez, fazendo suposições com base na experiência anterior.

C

Facilitam a cópia, a alteração e a manutenção do código.

D

Demonstram as práticas recomendadas do C#.

14. O que esta acontecendo no trecho de código:

```
float preco = (float) Convert.ToDouble(Console.ReadLine());
```

A

Esta convertendo de string para double, e depois convertendo para float

B

Esta convertendo de float para double, e depois fazendo um cast para string

C

Esta convertendo de string para double, e depois fazendo um cast para float