

INTRODUÇÃO À LINGUAGEM ORIENTADA A OBJETOS

PROGRAMAÇÃO E SISTEMAS DE INFORMAÇÃO – 2º ANO

CURSO PROFISSIONAL TÉCNICO DE GPSI



“A programação é a arte de organizar e dominar a confusão.”

(Steve Jobs)



Pensar como objeto: Windows Forms

Módulo 10 | Linguagem Orientada a Objetos

Projeto: Evolução da Classe “Empregado”

Objetivos:

- Reforçar o encapsulamento dos dados
- Implementar propriedades calculadas
- Documentar o código

Tarefas a realizar:

1. Encapsular Dados

- Transformar as propriedades Nome, EMail e Telefone em **privadas**.
- Manter apenas o acesso interno dentro da classe.
- As propriedades DataNasc e MoradaAtual permanecem públicas.
- O campo ID continua protected, e o contador autoID mantém-se private static.

2. Criar Propriedades Calculadas para a DataGridView

Implementar **quatro propriedades públicas de leitura** (sem set) que devolvam os valores necessários para a visualização dos empregados na interface gráfica:

Propriedade	Tipo	Descrição
ValorID	int	Devolve o identificador único do empregado
NomeAbreviado	string	Devolve o nome abreviado no formato “APELIDO, Nome”
Localidade	string	Obtém a localidade a partir da morada atual
Cargo	string	Indica o cargo do empregado (padrão: “Empregado”)

3. Tarefa 2 — Implementar Método GetNomeAbreviado()

4. Comentar e Documentar em XML

Adicionar **comentários XML (///)** antes de:

- Cada construtor
- Cada método (GetCargo(), GetNomeAbreviado())
- Cada nova propriedade calculada

Os comentários devem incluir:

```
/// <summary>
/// Descrição da função ou propriedade.
/// </summary>
/// <param name="_nome">Explicação do parâmetro</param>
/// <returns>Explicação do valor devolvido</returns>
```

Desafio:

Fazer validações para **Nome**:

- Deve ter entre 7 e 150 caracteres;
- Deve conter pelo menos duas palavras, ou seja, pelo menos um espaço entre as palavras.

Email:

- Tenta criar um objeto MailAddress com o valor recebido;
- Se ocorrer exceção, o e-mail é inválido. Se o e-mail for inválido, é substituído por uma string vazia ("").

Telefone:

- Usar uma expressão regular (regex) para verificar se o formato é válido.
 - Exemplo: Pode validar expressão com o formato:
 - Opcional sinal “+” no início;
 - Possíveis parênteses para código de país;
 - Apenas dígitos, espaços ou traços.

BOM TRABALHO! TU ÉS CAPAZ! CONSTRÓI O TEU CONHECIMENTO...

Os facilitadores da aprendizagem,

Andreia Quintal | Carlos Almeida | Cristina Gonçalves | Pedro Cardoso