🧠 Exercício de Fixação 6 — Módulo 1 (POO com Classes Simples)

🎯 Objetivos

- Consolidar fundamentos de Scrum e metodologias ágeis
- Explorar estrutura e uso de classes em C# (POO básica)
- Resolver problemas práticos com uso de propriedades, listas e métodos estáticos
- Desenvolver habilidades de pensamento computacional e organização de lógica

> 💡 **Nota mínima para aprovação: 8,5 pontos (85%)**

📘 PARTE 1 — Exercícios Teóricos

1.1 — Questões Objetivas: Scrum (0,25 ponto cada)

1. Qual prática ajuda o Product Owner a manter o backlog sempre alinhado com os objetivos do negócio?

- [ ] A. Refinamento contínuo com o time e stakeholders
- [ ] B. Congelar o backlog após o primeiro Sprint
- [ ] C. Deixar o Scrum Master revisar o backlog
- [ ] D. Esconder mudanças até o final do projeto

2. Qual o papel do Scrum Master quando o time não está entregando valor nas sprints?

- [ ] A. Reescrever os requisitos sozinho
- [ ] B. Ignorar a situação para evitar atrito
- [ ] C. Facilitar reflexões e remoção de impedimentos
- [ ] D. Cancelar a sprint e recomeçar do zero

3. No Scrum, qual ferramenta permite uma visão visual do progresso diário do time?

- [ ] A. Roadmap
- [ ] B. Quadro Kanban
- [ ] C. Backlog Refinado
- [ ] D. Product Vision Board

4. O que deve ocorrer ao final de cada Sprint?

- [ ] A. A equipe apresenta um relatório financeiro
- [ ] B. A equipe é reorganizada
- [ ] C. Uma entrega potencialmente utilizável é apresentada
- [ ] D. A Sprint Planning da próxima sprint é realizada

5. Qual a melhor forma de lidar com mudanças de escopo frequentes em Scrum?

- [ ] A. Rejeitar todas as mudanças
- [ ] B. Ignorar o backlog
- [ ] C. Atualizar o backlog e reavaliar prioridades com o PO
- [ ] D. Trocar o time de desenvolvimento

6. Quem é o responsável por garantir que o Scrum esteja sendo compreendido e seguido?

- [ ] A. Gerente de Projeto
- [ ] B. Product Owner
- [ ] C. Scrum Master
- [ ] D. Analista de Qualidade

1.2 — Questões Objetivas: C# e Classes (0,25 ponto cada)

7. Qual comando permite criar uma nova instância de uma classe?

- [ ] A. `new`
- [ ] B. `create`
- [ ] C. `instanceof`
- [ ] D. `typeof`

8. Para que serve o método `ToString()` em uma classe C#?

- [ ] A. Converte um objeto para um número
- [ ] B. Cria uma nova instância da classe
- [ ] C. Retorna uma representação textual do objeto
- [ ] D. Envia o objeto para o console

9. O que significa encapsulamento em POO?

- [ ] A. Tornar métodos públicos sempre
- [ ] B. Separar a lógica do código em arquivos

- [ ] C. Ocultar detalhes internos de uma classe
- [ ] D. Dividir variáveis em escopos diferentes

10. Qual dos seguintes trechos representa uma declaração correta de construtor?

- [ ] A. `public void Cliente() { }`
- [ ] B. `Cliente cliente() { }`
- [ ] C. `public Cliente() { }`
- [ ] D. `void Cliente() { return; }`

11. O que acontece se uma propriedade não for inicializada em C#?

- [ ] A. A compilação falha
- [ ] B. Um valor padrão será atribuído automaticamente
- [ ] C. O programa fecha imediatamente
- [ ] D. O objeto se torna `null`

12. O que representa um método `static` dentro de uma classe?

- [ ] A. Um método que só pode ser chamado por objetos
- [ ] B. Um método que pertence à instância
- [ ] C. Um método que pertence à classe, e não ao objeto
- [ ] D. Um método que nunca pode ser alterado

1.3 — Questões Dissertativas (1 ponto cada)

13. Por que a reutilização de código é considerada uma boa prática na programação orientada a objetos? Dê um exemplo onde isso seria útil.

14. Em um time ágil, qual o papel da retrospectiva ao final de cada sprint e como ela pode contribuir para a melhoria contínua?

🛠 PARTE 2 — Desafios Práticos em C# (POO com Classes)

> Os desafios abaixo utilizam apenas **propriedades**, **listas** e **métodos simples**.

🔹 Exercício 2 — Classe `Funcionario` com Cálculo de Bônus (1,0 ponto)

Crie uma classe `Funcionario` com:

- `Nome` (string)
- `SalarioBase` (double)
- `Cargo` (string)

Implemente:

- Método `CalcularBonus()` que retorna:
  - 20% do salário se o cargo for `"Gerente"`
  - 10% se for `"Analista"`
  - 5% para qualquer outro cargo
- Método `SalarioTotal()` que retorna o salário base somado ao bônus

No `Main`:

- Solicite o usuário cadastrar o funcionário
- Exiba o nome, cargo, bônus e salário total de cada um

🔹 Exercício 2 — Classe `Livro` com Validação de Estoque (1,0 ponto)

Crie uma classe `Livro` com:

- `Titulo` (string), `Autor` (string), `QuantidadeEstoque` (int)

Implemente os métodos:

- `AdicionarEstoque(int quantidade)`
- `Vender(int quantidade)` que só permite vender se houver estoque suficiente

No `Main`:

- Solicite ao usuário cadastrar livros, vender caso tenha no estoque e visualizar o estoque dos livros.

🔹 Exercício 3 — Classe `AgendaContatos` com Pesquisa Simples (1,0 ponto)

Crie uma classe `Contato` com:

- `Nome` (string)
- `Telefone` (string)

Na classe `AgendaContatos`, implemente:

- Lista de contatos
- Método `Adicionar(Contato c)`
- Método `BuscarPorNome(string nome)`
- Método `ListarTodos()`

No `Main`:

- Solicite o usuário cadastrar contatos
- Busque um contato pelo nome
- Liste todos os contatos

🔹 Exercício 4 — Classe `PassagemViagem` com Taxas e Validação (1,0 ponto)

Crie a classe `PassagemViagem` com os atributos:

- `NomePassageiro` (string)
- `Destino` (string)
- `ValorBase` (double)
- `TaxaEmbarque` (double)

Implemente os métodos:

- `ValorTotal()` que retorna a soma de `ValorBase + TaxaEmbarque`
- `ValidarValores()` que verifica se `ValorBase` e `TaxaEmbarque` são maiores que zero
- `ResumoPassagem()` que retorna uma string formatada com os dados da passagem (nome, destino, total)

No `Main`:

- Solicite o usuário cadastrar passageiros
- Valide os dados antes de calcular
- Exiba o resumo de cada passagem
- Exiba o valor total arrecadado com todas as passagens

🔹 Exercício 5 — Classe `CartaoPrePago` com Limite de Crédito (1,0 ponto)

Crie a classe `CartaoPrePago` com:

- `NomeUsuario` (string)

- `LimiteDisponivel` (double)

Métodos:

- `AdicionarCredito(double valor)` → adiciona ao limite disponível (valor deve ser positivo)
- `RealizarCompra(double valor)` → só permite se `valor <= LimiteDisponivel`, senão exibe erro
- `ExibirResumo()` → mostra nome do usuário e limite restante formatado

No `Main`:

- Crie um cartão com limite inicial
- Adicione crédito
- Realize 2 compras (uma válida e outra inválida)
- Mostre o resumo após cada operação

✅ Avaliação

Item	Pontuação
Teoria Scrum (6x0,25)	1,5 pts
Teoria C#/Classes (6x0,25)	1,5 pts
Dissertativas (2x1)	2,0 pts
Prática (5x1)	5,0 pts
Nota Máxima	**10,0 pts**
Nota Mínima p/ Aprovação (85%)	**8,5 pts**
