



## **LABORATÓRIO DE DESENVOLVIMENTO DE SOFTWARE | 21179**

### **Data e hora de realização**

17 de junho de 2024, às 10h00 de Portugal Continental

### **Duração da prova**

90 minutos + 60 minutos de tolerância

### **Temática / Tema / Conteúdos**

Matéria teórica do semestre e sua aplicação no projeto

### **Objetivos**

- Aferir a compreensão dos temas teóricos estudados
- Aferir a compreensão da aplicação práticas desses temas no contexto de aplicação durante o projeto prático desenvolvido nos e-fólios.

### **Normas a respeitar**

- Todas as páginas do documento que entregar devem ser numeradas.
- Deve explicitar a que pergunta, alínea e subalínea está a responder.  
No código-fonte, pode dar essas indicações como comentários.
- Utilize a Folha de Resolução disponibilizada e anexe ficheiros de código-fonte, compactando-os juntamente com a folha de resolução num único ficheiro .zip.
- Carregue o ficheiro para a plataforma no dispositivo E-fólio Global até à data-limite e hora-limite de entrega. Evite a entrega próximo da hora-limite para se precaver contra eventuais problemas. O ficheiro não deve exceder 8 MB.

Votos de bom trabalho!

Leonel Morgado

## Trabalho a desenvolver

### 1.ª Parte (4 Valores)

#### Código de referência

```
1 class Program {
2     static void Main() {
3         // Classe Produto definida noutra parte do código, omitida aqui
4         List<Produto> produtos = new List<Produto>();
5         // Início do menu interativo
6         while (true) {
7             // Apresentação do menu ao utilizador
8             Console.WriteLine("1. Adicionar Produto");
9             Console.WriteLine("2. Listar Produtos");
10            Console.WriteLine("3. Sair");
11            // Leitura da opção do utilizador
12            var escolha = Console.ReadLine();
13            switch (escolha) {
14                // Se escolheu adicionar produto à lista
15                case "1":
16                    // Lê nome do produto
17                    Console.Write("Nome: ");
18                    string nome = Console.ReadLine();
19                    float preco;
20                    bool valido;
21                    // Só deixa avançar se for inserido um preço válido
22                    do {
23                        //Lê o preço do produto
24                        Console.Write("Preço: ");
25                        string precoStr = Console.ReadLine();
26                        // Verifica se é um número
27                        valido = float.TryParse(precoStr, out preco);
28                        // Informa que houve um erro
29                        if (!valido) {
30                            Console.WriteLine("Valor inválido. Preço:");
31                        }
32                    } while (!valido);
33                    // Inseere o produto novo na lista
34                    Produto produto
35                        = new Produto { Nome = nome, Preco = preco };
36                    produtos.Add(produto);
37                    break;
38                case "2":
39                    // Listar produtos
40                    foreach (var p in produtos) {
41                        Console.WriteLine($"{p.Nome} - {p.Preco}");
42                    }
43                    break;
44                case "3":
45                    // Sair do programa
46                    return;
47                default:
48                    // Escolha inválida
49                    Console.WriteLine("Escolha uma opção válida.");
50                    break;
51            }
52        }
53    }
54 }
```

As alíneas seguintes são todas de resposta curta.

**1.**

- a)** Indique as linhas do código de referência onde há operações de *input*.
- b)** Indique as linhas do código de referência onde há operações de *output*.

Caso tenha tido dúvidas de interpretação do código que lhe afetaram a resposta, exponha-as por escrito.

- 2.** Para reescrever o código de referência segundo o estilo arquitetónico MVC, indique como distribuir pelos componentes as tarefas descritas nos comentários desse código. Por ex., se considerar que a tarefa “// Apresentação do menu ao utilizador” (linha 7) corresponde ao Controller, marque a célula C7, se acha que corresponde ao Model, marque a célula M7, para indicar que corresponde à View, marque a célula V7.

- a)** Segundo a abordagem de Krasner & Pope (1988).

Linha →	<b>3</b>	<b>5</b>	<b>7</b>	<b>11</b>	<b>14</b>	<b>16</b>	<b>21</b>	<b>23</b>	<b>26</b>	<b>28</b>	<b>33</b>	<b>39</b>	<b>45</b>	<b>48</b>
↓ Componente														
M														
V														
C														

- b)** Segundo a abordagem de Curry & Grace (2008).

Linha →	<b>3</b>	<b>5</b>	<b>7</b>	<b>11</b>	<b>14</b>	<b>16</b>	<b>21</b>	<b>23</b>	<b>26</b>	<b>28</b>	<b>33</b>	<b>39</b>	<b>45</b>	<b>48</b>
↓ Componente														
M														
V														
C														

- c)** Explique as dúvidas ou dilemas com que se debateu para responder às alíneas a) e b) e justifique as principais opções que tomou ao dar as suas respostas.

**3.** Suponha que antes de adicionar um produto à lista (linha 36), queria verificar se o produto já existia na lista e, se assim fosse, notificar o utilizador dessa situação, evitando duplicar produtos.

- a)** Explique como devem os componentes MVC do seu código articular-se para verificar a existência do produto na lista e decidir se deve ser adicionado ou não o produto novo, agindo depois em conformidade (adicionar o produto novo ou apenas informar o utilizador de que o produto já existe).
- b)** Discuta se esta situação da alínea anterior deveria ser ou não implementada com recurso a eventos e delegados, fundamentando a sua resposta com base nos princípios da qualidade do software.

**4.** Neste código de exemplo, ao se obter o preço do produto, é feita diretamente a sua validação; caso esta falhe, é indicado ao utilizador que é preciso inserir novamente o preço. O programa fica neste ciclo até se introduzir um preço válido.

- a)** Como poderia modificar a forma como o código faz a validação, para melhorar a qualidade de software “independência para fomentar a reutilização”?
- b)** Quando o utilizador introduz um preço não numérico, está-se perante uma falha, um erro ou uma falta? Justifique a sua resposta.

## **2.ª Parte (8 Valores)**

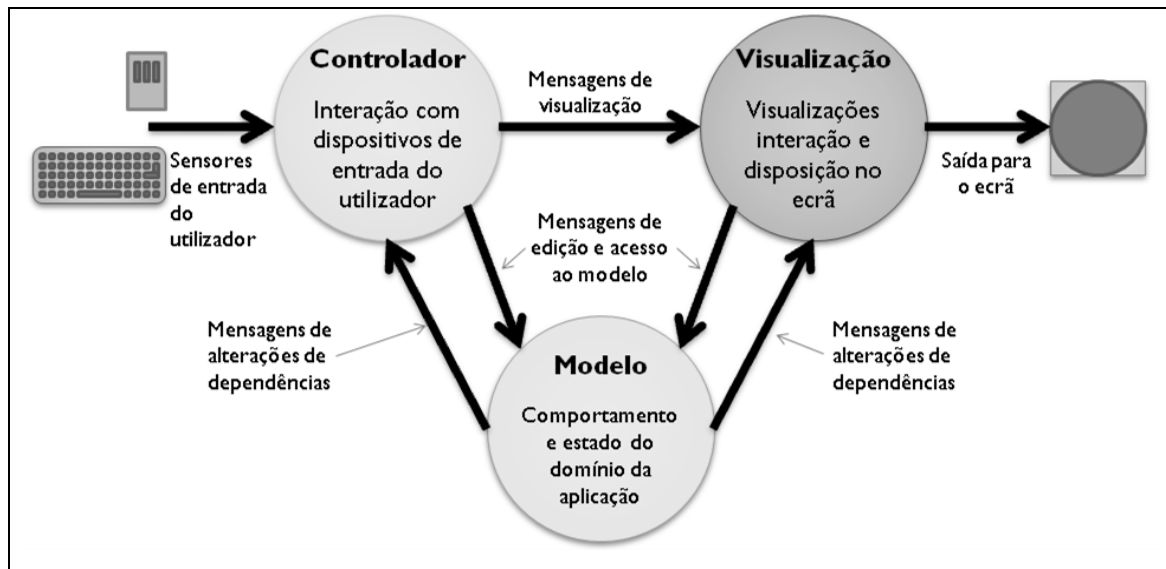
**Em duas páginas**, escreva um texto reflexivo, estruturado **explicitamente** pelas seguintes alíneas:

- a) Relativamente ao uso de eventos e delegados no seu projeto:

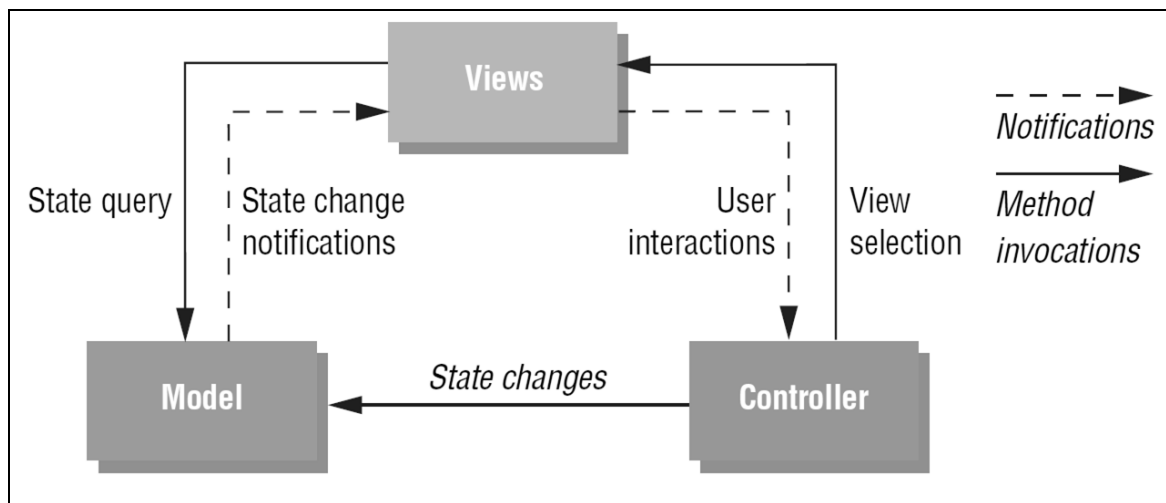
- i. Dê um exemplo concreto, no projeto da sua equipa, de um par evento-delegado que tenha sido criado por vós. Descreva o que comunicava o evento e o contexto em que foi utilizado (se não tiverem criado nenhum par evento-delegado, explicita esse facto).
  - ii. Explique concretamente como o uso deste par evento-delegado contribuiu para o acoplamento fraco entre componentes no projeto, dando exemplos do projeto. Caso não tenha usado eventos/delegados no projeto, use como exemplo uma hipotética evolução do projeto.
- b) Relativamente à estrutura MVC do seu projeto:
- i. Descreva a estruturação MVC implementada no seu projeto, indicando (não é preciso ser exaustivamente) que classes tinham os componentes Model, View e Controller e quais as principais responsabilidades de cada um.
  - ii. Explique como é que essa estruturação ajudou ou dificultou a organização e manutenção do código, fornecendo exemplos específicos de benefícios ou desafios encontrados.
- c) Dê exemplos de testes unitários e de outro tipo (por ex., testes de integração) que a sua equipa tenha efetuado. Explique qual foi o objetivo desses testes, quais os resultados e como afetaram o desenvolvimento do projeto.

## Recursos

### KRASNER & POPE (1988)



### CURRY & GRACE (2008)



### APOIOS DE SINTAXE

```
throw new ClasseDaExcecao (ParametroDaExcecao);  
catch (ClasseDaExcecao VariavelDaExcecao) { }  
public delegate TipoDevolvido  
    NomeDoTipoDeDelegado (Parametro1, Parametro2);  
NomeDoTipoDeDelegado delegado = new  
    NomeDoTipoDeDelegado (MetodoAtribuido);  
public event NomeDoTipoDeDelegado NomeDoEvento;  
NomeDoEvento += delegado;  
NomeDoEvento += new NomeDoTipoDeDelegado (MetodoAtribuido);
```

## **Cr terios de avalia o e cota o**

1.ª parte: 4 valores, distribu dos da seguinte forma:

1. a) 0,3 valores  
b) 0,3 valores
2. a) 0,6 valores  
b) 0,6 valores  
c) 0,2 valores
3. a) 0,5 valores  
b) 0,5 valores
4. a) 0,5 valores  
b) 0,5 valores

2.ª parte: 8 valores, distribu dos da seguinte forma:

- a) i) 1,5 valores  
ii) 1,5 valores
- b) i) 1,5 valores  
ii) 1,5 valores
- c) 2 valores

**Total:** 12 valores