

U.C. 21179

Laboratório de Desenvolvimento de Software

6 de julho de 2018

INSTRUÇÕES

PARA A RESOLUÇÃO DA PROVA, ACONSELHA-SE QUE LEIA ATENTAMENTE O SEGUINTE:

1. Verifique o exemplar que lhe foi entregue e, no caso de estar incompleto ou com qualquer deficiência, dirija-se ao professor vigilante. A prova termina com a palavra **FIM**.
2. A prova é efetuada sem consulta de outros materiais. No final do enunciado tem material de referência para consulta durante a prova.
3. A prova está dimensionada para um tempo de resolução de uma hora e trinta minutos.
4. A prova é constituída por 2 partes. A cotação de cada uma das partes está indicada, bem como a cotação das alíneas individuais.
5. A resolução deve ser escrita a esferográfica. Utilize, sempre, uma letra legível e não use uma caneta de outra cor que não seja o preto ou o azul. As respostas a lápis não serão consideradas.
6. A interpretação dos enunciados das questões também faz parte da sua resolução, pelo que, se existir alguma ambiguidade, deve indicar claramente como foi resolvida.

Bom Trabalho!

Nome:

N.º de Estudante: B. I. n.º

Turma: Assinatura do Vigilante:

1.ª Parte (4 Valores)

Código de referência

Este bloco de código é um exemplo do uso da Face API para deteção de rostos em imagens.

Adaptado de <https://docs.microsoft.com/en-us/azure/cognitive-services/face/tutorials/faceapiincsharp/tutorial>

```
1 public partial class MainWindow : Window
2     {
3         (...) //Dados sobre as faces detetadas na imagem.
4         Face[] faces;
5         String[] faceDescriptions;
6         (...)
7         // Reage ao clique no botão "Procurar".
8         private void BrowseButton_Click(object sender, RoutedEventArgs e)
9         {
10            // Obter do utilizador o ficheiro de imagem a analisar.
11            var openFileDialog = new Microsoft.Win32.OpenFileDialog();
12            openFileDialog.Filter = "Imagem JPEG (*.jpg)|*.jpg";
13            bool result = openFileDialog.ShowDialog(this);
14
15            // Regressar em caso de cancelamento.
16            if (!result) { return; }
17
18            // Mostrar a imagem.
19            Uri fileUri = new Uri(openFileDialog.FileName);
20            BitmapImage bitmapSource = new BitmapImage();
21            (...)
22            bitmapSource.UriSource = fileUri;
23            (...)
24            FacePhoto.Source = bitmapSource;
25            (...)
26        }
27
28        // Mostra a descrição da face quando o rato se lhe sobrepõe.
29        private void FacePhoto_MouseMove(object sender, MouseEventArgs e)
30        {
31            (...)
32        }
33    }
```

As alíneas seguintes são todas de resposta curta.

1.

a) Indique as linhas do código de referência onde há operações de *input*: _____ 0,5

b) Indique as linhas do código de referência onde há operações de *output*: _____ 0,5

Caso tenha tido dúvidas de interpretação do código que lhe afetaram a resposta, exponha-as aqui:

Nome:

N.º de Estudante:

B. I. n.º

Turma:

Assinatura do Vigilante:

2. Suponha que queria reescrever o código de referência segundo o estilo arquitetónico MVC. Indique como distribuiria pelos componentes as tarefas descritas nos comentários desse código.

Por ex., se considerar que a tarefa "//Dados sobre as faces detetadas na imagem." (linha 03) corresponde ao Controller, escreva "C: 03", se acha que corresponde ao Model, escreva "M: 03", para indicar que corresponde à View, escreva "V: 03".

a) Segundo a abordagem de Krasner & Pope (1988): _____

0,3

b) Segundo a abordagem de Curry & Grace (2008): _____

0,3

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: _____

0,4

3. O código anterior depende de componentes não explicitados. Para que funcione a linha 24 (`FacePhoto.Source = bitmapSource;`), é preciso que (i) esta atribuição lance um evento que origine a atualização do ecrã; ou (ii) o código da propriedade `Source` faça essa atualização do ecrã diretamente; ou (iii) haja um ciclo na aplicação durante o qual se vai usar a propriedade `Source` para atualizar o ecrã.

1

Considere que há pelo menos três componentes: (1) A aplicação global; (2) a classe `FacePhoto`; (3) o componente que gere a visualização. Se sentir necessidade, pode indicar outros.

Desenhe três arquiteturas encaminhamento e filtragem, respetivamente para os casos i, ii, e iii:

Nome:

N.º de Estudante: B. I. n.º

Turma: Assinatura do Vigilante:

4. Considere o código de referência nas linhas 15 e 16. Se `result` for `false`, tal indica que o utilizador clicou no botão “Cancelar”.

a) Se `result` for `false`, considera que tal pode ser um falha, um erro ou uma falta? Ou não? 0,4

Justifique. _____

b) Escolha uma linha que não as 15 e 16 e indique uma falha que possa ocorrer na execução dessa linha. 0,3

c) Escolha uma linha que não as 15 e 16 e aponte uma falta neste código (justifique). 0,3

2.ª Parte (8 Valores)

Em duas páginas, efetue uma descrição do trabalho desenvolvido ao longo dos e-fólios, refletindo sobre os temas que se seguem com base nos princípios teóricos estudados.

a) A separação em módulos (Model-View-Controller ou o estilo arquitetónico efetivamente seguido); 3
i) qual o principal problema que encontrou;
ii) qual a abordagem seguida e como resultou;
iii) em retrospectiva, que mudaria na abordagem a esta separação.

b) O recurso a eventos e interfaces para separação de interesses e independência de dados; 2,5
i) qual o principal problema que encontrou;
ii) qual a abordagem seguida e como resultou;
iii) em retrospectiva, que mudaria na abordagem a esta técnica.

c) O recurso a exceções personalizadas para gestão de erros. 2,5
i) qual o principal problema que encontrou;
ii) qual a abordagem seguida e como resultou;
iii) em retrospectiva, que mudaria na abordagem a esta técnica.

FIM

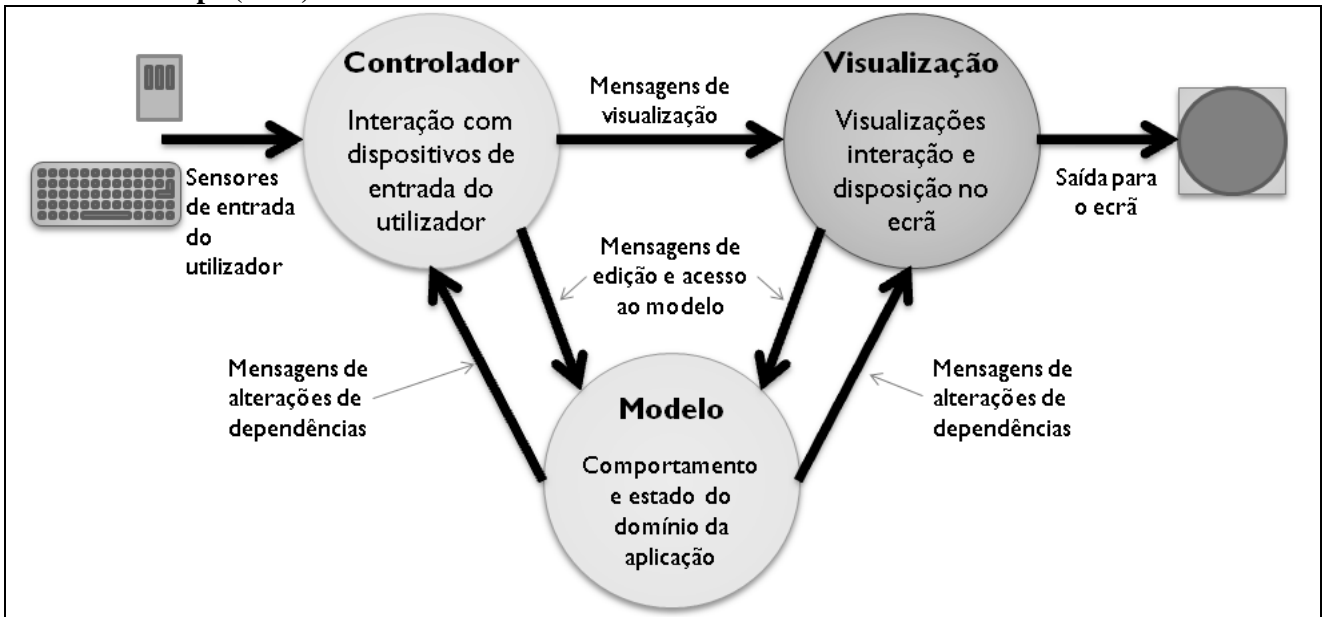
Nome:

N.º de Estudante: B. I. n.º

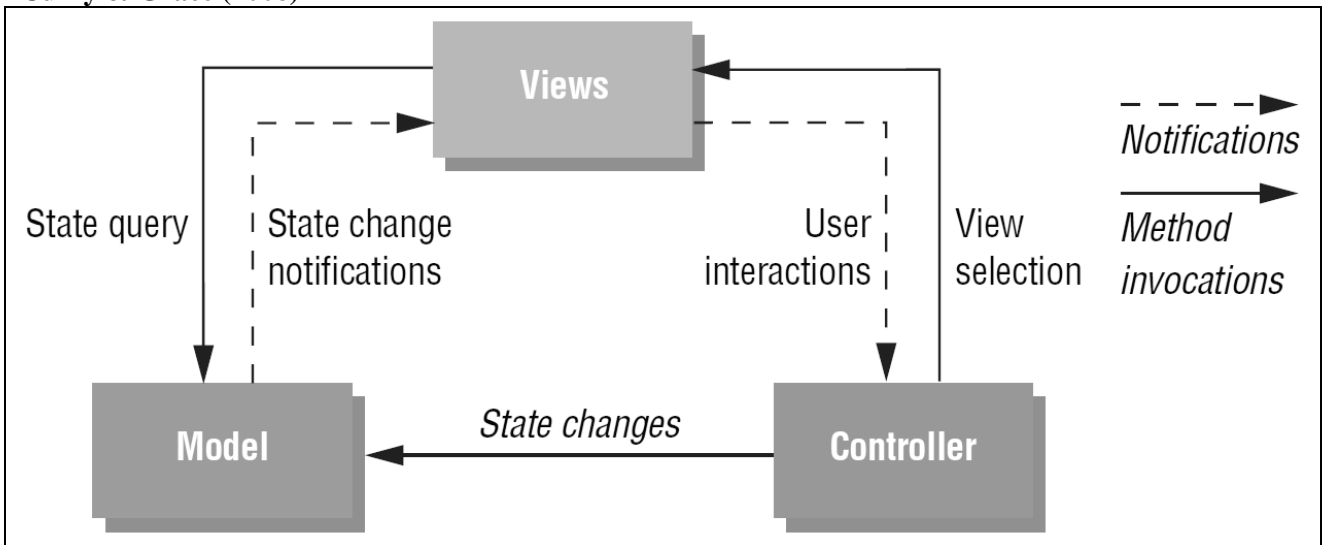
Turma: Assinatura do Vigilante:

Material de referência

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);
```