UNIVERSIDADE da MADEIRA

Nome: **Número:** **Recurso/Melhoria de:**

06/06/2022 - Programação Avançada - Exame de Recurso
Docente: Filipe Quintal, Diogo Freitas Duração: 3h

Qualquer tentativa de fraude implica a anulação do teste;
Utilize uma caligrafia legível. Duração: 3h para a o exame completo 1h30m para metade

1ª Frequência

1) Classifique as afirmações abaixo como verdadeiras (V) ou falsas(F) (0.25 por cada alínea correta, -0.1 por cada alínea incorreta)

a) O paralelismo por bit está inerente ao aumento da capacidade de processamento dos bits processados a cada ciclo de um processador. ___

b) A técnica conhecida como *Load balancing* garante que existe um programa paralelo tem sempre uma execução mais rápida que o mesmo programa implementado de uma forma sequencial. ____

c) Diferentes níveis de paralelismo podem ser utilizados na mesma solução/programa. ___

d) Numa implementação do padrão *Client-Server* as entidades Cliente e Servidor processam a mesma lógica do programa. ___

e) O método wait() aplicado a uma *Thread* em Java espera que esta finalize a sua execução. ___

f) Semáforos e trancas poderão ser utilizadas nas mesmas situações independentemente do problema em questão. ___

g) Ataques a Linhas de comunicação podem por exemplo ameaçar sistemas operativos e utilitários do computador.___

h) Uma troca de mensagem que utilize MAC codes irá obrigatoriamente utilizar mais largura de banda do que mensagens *plain-text*.___

2) Descreva sucintamente a diferença entre paralelismo por memória partilhada e distribuída (0.75 valores).

3) Como podemos diferenciar entre paralelismo por bit e paralelismo por pipelining (0.75 valores).

4) Considere o problema abaixo identificado através de pesudo-código?

a) Identifique os principais problemas com a implementação (0.755 valores).

```
1  T1:
2  lock(S1)
3  while(value%2!=0)
4      value = AskUserForInput()
5
6  lock(S2)
7  print(value);
8  unlock(S1)
```

```
1  T2:
2  lock(S2)
3  lock(S1)
4  timesensitiveFuncion()
5  unlock(S2)
6  unlock(S2)
```

b) Proponha uma possível solução que mantenha a filosofia paralela (0.755 valores).

5) Descreva sucintamente a diferença entre as arquiteturas *Single-Instruction, Multiple-Data e Multiple-Instruction, Multiple-Data*, para a implementação de uma solução paralela (1 valos).

6) Considere o problema : Uma empresa finanças tem de aplicar uma série de cálculos fiscais aos registos de todos os cidadãos de um país (>10M), Estes cálculos são cumulativos, e os valores finais deverão ficar guardados num ficheiro

a) Identifique uma abordagem para a paralelização da lógica proposta. Identifique que padrões arquiteturais poderiam ser utilizados para a sua implementação (1.25 valores)

7) Considere que um *atacker* C consegue escutar mensagens entre duas entidades A e B. Estas mensagens estão a ser enviadas utilizando criptografia de chave pública/privada. As chaves públicas de todas as entidades estão localizadas num repositório comum. Indique se este mecanismo permite:

a) Esconder o conteúdo da mensagem (justifique a sua resposta) (0.75 valores).

b) 'A' verificar que C não intercetou e alterou uma mensagem de 'B' (1 valor).

8) Descreva sucintamente as principais diferenças entre os conceitos de Integridade e Disponibilidade num sistema de computação (1 valor).

UNIVERSIDADE da MADEIRA **Nome:** **Número:** **Recurso/Melhoria de:**

2ª Frequência

9) Classifique as afirmações abaixo como verdadeiras (V) ou falsas(F), (0.25 por cada alínea correta, -0.1 por cada alínea incorreta)

a) A criação de um certificado para uma data entidade A necessita apenas da sua chave privada e algoritmo de encriptação simétrico.

b) Uma assinatura digital permite ao recipiente de uma mensagem verificar que a mesma não foi alterada.

c) O resultado do processo de rasterização é uma série de vértices e primitivas que descrevem a geometria de uma cena.

d) Nas primeiras mensagens do protocolo TLS são definidos os algoritmos de encriptação a utilizar na troca de mensagens

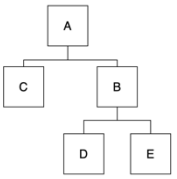
e) A imagem resultante de uma câmara *pinhole* será invertida.

f) Um sistema de coordenadas homogéneo permite realizar operações em matrizes utilizando apenas soma de matrizes.

g) A profundidade do framebuffer define apenas o tamanho (em bits) de uma imagem a apresentar.

h) A organização hierárquica dos certificados significa que o certificado de uma entidade de topo A tem de ser assinado por todas as entidades nos níveis abaixo.

10) Considere a hierarquia à direita, composta pelas entidades A,B,C,D e E. Indique a composição do certificado de B (1.25 valor).



11) Descreva sucintamente o processo de triangulação em computação gráfica. Porquê é que este processo é tão importante em sistemas gráficos modernos (0.75 valores).

12) Descreva sucintamente o modelo de iluminação em computação gráfica, defina 2 tipos de luz (1 valores).

13) Descreva sucintamente o processo de *clipping* e montagem de primitivas parte do pipeline de processamento gráfico (1 valores).

14) Descreva sucintamente os 3 principais passos para a utilização de texturas em computação gráfica (0.75 valores)

15) Considere um ponto x,y,z, especifique uma matriz de transformação para:

a) Mover os pontos por 15 unidades no eixo de x's, -1 unidades no eixo do y's 2 unidades no eixo z's. (1 valor)

b) Redimensionar os pontos por um fator de -1 no eixo dos x's, 10 no eixo dos y's e manter a mesma dimensão nos z's (0.75 valores).

16) Descreva <u>sucintamente</u> o comportamento implementado pelo código abaixo (1.5 valores)

```
49    gl.clearColor(0.75, 0.75, 0.75, 1);
50    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
51    gl.enable(gl.CULL_FACE);
52    var vertexShader = gl.createShader(gl.VERTEX_SHADER);
53    var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
54    gl.shaderSource(vertexShader,vertexShaderText);
55    gl.shaderSource(fragmentShader, fragmentShaderText);
56    gl.compileShader(vertexShader);
57    gl.compileShader(fragmentShader);
58    var program = gl.createProgram();
59    gl.attachShader(program, vertexShader);
60    gl.attachShader(program,fragmentShader);
61    gl.linkProgram(program);
```