

Critérios de Correção

# Modelação de Sistemas de Informação | CÓDIGO 21177

## Período de Realização: decorre 07-06-2022, 10:00 TMG com 2,5 horas de duração

## Data de Limite de Entrega: decorre até 12:30 TMG

## Temática / Tema / Conteúdos: Modelação de Sistemas de Informação

## Objetivos: Modelizar requisitos de sistemas de informação com UML "Unified Modeling Language" usando diagramas de utilização, diagramas de classes e CRUD. Especificar o software com diagramas de interação: sequência, estados e atividades; e especificar infraestruturas com os diagramas de arquitetura.

## Trabalho a desenvolver: Resolução de um conjunto de exercícios.

**Critérios de avaliação e cotação:** A cotação deste e-fólio é de 120 pontos = 12 valores, pode encontrar as cotações parciais junto de cada pergunta. A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida. Critérios de avaliação gerais: (i) para a dificuldade de leitura (linhas cruzadas, letras com fontes desadequadas) a penalização é de 20% a 100%; (ii) para erros e omissões a penalização é de 20% a 100%.

**Normas a respeitar:** Deve redigir o seu E-fólio na Folha de Resolução disponibilizada na turma e preencher todos os dados do cabeçalho. Todas as páginas do documento devem ser numeradas. O documento A4 deve ser redigido em Times New Roman, tamanho de letra 12. O espaçamento entre linhas deve corresponder a 1,0 ou 1,5 linhas.

Nomeie o ficheiro com “eFolioG” + <nº estudante> +<nome estudante com o máximo de 3 palavras>. Finalmente deve gerar um PDF do documento. Deve carregar o referido ficheiro para a plataforma no dispositivo E-fólio Global até à data e hora limite de entrega. Evite a entrega próximo da hora limite para se precaver contra eventuais problemas. O ficheiro a enviar não deve exceder 8 MB. Votos de bom trabalho!

Vetor das perguntas: 1 2 3, 4 5 6

Vetor das cotações: 2 2 2, 2 2 2 somando 12 valores

**Critérios gerais:**

* Respostas com atores ou classes com nome “sistema”, "bases de dados" ou "aplicação" são penalizadas;
* Respostas com nomes de atores, use-cases ou classes que não constam no texto são penalizadas;
* Respostas com trocas de atores com use-cases, trocas de atores com classes, ou outras combinações semelhantes, são penalizadas.

**Parte I – Caso Prático Integrado**

Leia com atenção a seguinte introdução. Na sua análise não exceda 5 atores, 7 casos-de-utilização e 7 classes.



O diretor de uma empresa que produz produtos alimentares, pretende melhorar o sistema de informação do stock (aprovisionamento) de matérias primas (MP).

* A empresa tem vários produtos fornecidos por diferentes fornecedores com preços diferentes;
* É importante ter os registos dos fornecedores com o nome, a morada, e os contactos (telefone, telemóvel, e-mail, etc.).
* Os produtos estão agrupados em categorias;
* Pretende-se ter uma imagem clara do stock MP a cada instante, também conhecido por inventário à data (ano-mês-dia);
* Pretende-se ainda ter o registo de todos os movimentos de entrada e saída do stock de matérias primas; a entrada é realizada pelos fornecedores e as saídas são para a produção de produtos alimentares;
* Para cada entrega dos fornecedores é importante saber se já estão pagas ou ainda falta pagar;

Para além do diretor, existem dois assistentes que lançam os dados na aplicação. Um dos assistentes faz o registo das entradas e saídas do stock MP. O outro assistente tem como tarefas o envio dos pedidos de encomendas para os fornecedores e os pagamentos aos fornecedores.

O diretor pretende obter com o novo SI um vasto conjunto de relatórios: um relatório mensal sobre os movimentos, pretende saber o valor em stock (quantidade \* preço), pretende saber quais os fornecedores a quem falta pagar, entre outros.

O diretor pretende ainda que o sistema funcione na 'cloud'. Todos os processos devem ter em consideração o Regulamento Geral de Proteção de Dados Pessoais.

**1.** (2,0 valores) O que entende por requisitos funcionais e não-funcionais. Defina dois requisitos funcionais e dois não-funcionais do caso prático. Utilize a tabela em baixo para responder.

|  |  |
| --- | --- |
| definir requisito funcional |  |
| definir requisito não-funcional |  |
| exemplo de dois requisitos funcionais da narrativa | 12 |
| exemplo de dois requisitos não-funcionais da narrativa | 12 |

**Resposta:**

|  |  |
| --- | --- |
| definir requisito funcional(0,4) | são os requisitos associados às funcionalidades (use-cases) do sistema, i.e., às funcionalidades que o sistema deve suportar;  |
|  |
| definir requisito não-funcional (0,4) | são requisitos que não são suportados pelo sistema, sendo transversais ou complementares ao disto sistema tais como requisitos organizacionais (políticas, processos), externos (ético, legais) ou de produto (eficiência, portabilidade); |  |
|  |
| exemplo de dois requisitos funcionais da narrativa (0,6) | 1) lançar encomendas e pagamentos fornecedores;2) lançar movimentos de entrada e saída; |  |
| exemplo de dois requisitos não-funcionais da narrativa (0,6) | 1) o diretor pretende ainda que o sistema funcione na 'cloud';2) todos os processos devem ter em consideração o Regulamento Geral de Proteção de Dados Pessoais. |  |

**Critérios de correção:**

- 0,4 definição requisitos funcionais

- 0,4 definição requisitos não-funcionais

- 0,6 exemplos de requisitos funcionais

- 0,6 exemplos de requisitos não-funcionais

- penalização de 50% a 100% para requisitos desadequados

**2.** (2,0 valores) Construa a Casos-de-Utilização versus Atores (ver tabela em baixo) e o Diagrama de Casos-de-Utilização do novo sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| Casos-de-uso \ Atores | A1 | A2 | A3 |
| X |   |   |   |
| Y |   |   |   |
| Z |   |   |   |
| W |   |   |   |

**Resposta parcial:**

|  |  |  |  |
| --- | --- | --- | --- |
| **casos-de-uso \ atores** | assistente 1 | assistente 2 | diretor |
| lançar encomendas e pagamentos fornecedores | X |   |   |
| lançar movimentos de entrada e saída |   | X |   |
| gerar relatórios |   |   | X |

**Critérios de correção:**

- 50% para a matriz

- 50% para o diagrama

- penalização de 50% a 100% para atores ou casos de uso desadequados

**3.** (2,0 valores) Construa o Diagrama de Classes do novo sistema com os seguintes elementos:



 Herança Classe Agregação

 Generalização-Especialização Associativa Todo-Parte

**Resposta:**

****

**Critério de correção:**

**-** 70% para as classes

- 30% para as associações

- penalização de 50% a 100% para classes ou associações desadequados

**4.** (2,0 valores) Construa a matriz CRUD (*create, read, update, delete*), com os Casos-de-Utilização versus as Classes encontradas, com o seguinte aspeto:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Casos-de-uso \ Classes | A | B | C | D |
| X |  |  |  |  |
| Y |  |  |  |  |
| Z |  |  |  |  |
| W |  |  |  |  |
| Contadores C R U D |  |  |  |  |

**Resposta:**



**Critério de correção:**

- 70% para a tabela CRUD

- 30% para os contadores CRUD

- penalização 50% a 100% para casos-de-uso, classes ou preenchimento desadequados

- penalização para matrizes sem a forma 1N11

**5.** (2,0 valores) Construa um Diagrama de Sequência de um dos use-case mais complexos do sistema de informação. Tenha em consideração a informação das alíneas anteriores.

**Resposta:**

****

**Critério de correção:**

- 50% classes

- 50% atores e mensagens

- penalização de 50% a 100% para atores, classes ou mensagens desadequadas

- penalização quando não é referido o use-case

- penalização se as classes não coincidirem com as classes do CRUD

**Parte II – Outros**

**6.** (2,0 valores) Na teoria dos números, um número abundante ou excessivo é um número menor do que a soma de seus divisores próprios. O inteiro 12 é o primeiro número abundante. Seus divisores apropriados são 1, 2, 3, 4 e 6 para um total de 16. A quantidade pela qual a soma excede o número é a abundância. O número 12 tem abundância de 4, por exemplo. Os primeiros números abundantes são: 12, 18, 20, 24, 30, 36, 40, 42, 48, 54 ...

Construa um Diagrama de Atividades que verifique se um número é abundante. De seguida construa outro Diagrama de Atividades de gere os 10 primeiros números abundantes.

**Resposta:**

def calc\_divisores(num):

 divisores=[]

 for i in range(1, num//2+1):

 if num % i ==0: divisores.append(i)

 return divisores

def abundantes(num):

 for i in range(1, num):

 if sum(calc\_divisores(i)) > i: print(i)

abundantes(60)



**Critério de correção:**

- 50% para o ciclo e condições

- 50% para os detalhes dos algoritmos: variáveis e afetações

**FIM**