P. Viseu
Tecnologia e Gestão Viseu

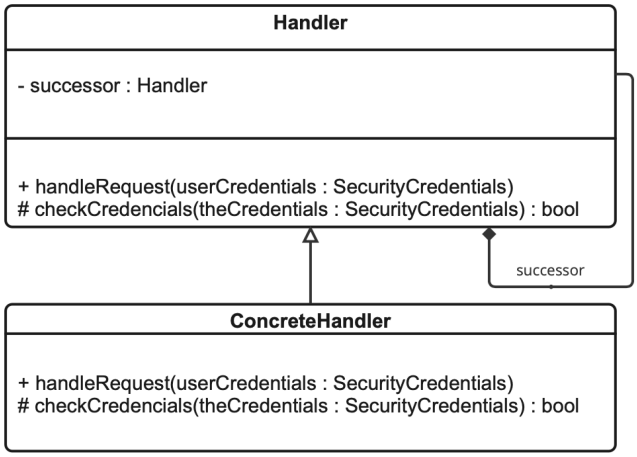
ei

Engenharia de Software II

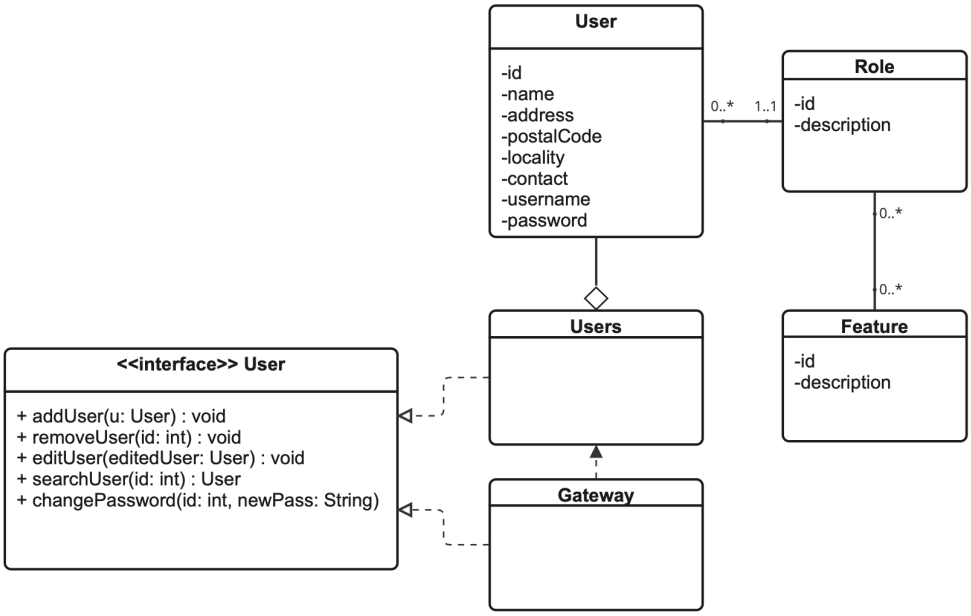
Exame Tipo

Nome: _____ Número: _____

Considere o padrão *Chain of Responsibility*



aplicado à implementação do trabalho prático da UC



de forma a responder às questões seguintes.

P. Viseu
Tecnologia e Gestão Viseu

Para responder às questões de escolha múltipla, deverá usar a seguinte grelha.

I	II	III	IV	V	VI

A duração da prova é de 120 minutos
É descontado ¼ da cotação das respostas incorretas (apenas escolha múltipla).

I) [Psicologia de Testes] Durante o processo de implementação do padrão anterior, a equipa de desenvolvimento tem de proceder à identificação de erros no código. Essa tarefa permitirá encontrar

1. todos os erros da aplicação;
2. o maior número de erros possível;
3. todos os erros passíveis de ser encontrados com a abordagem black-box e os restantes serão encontrados utilizando uma abordagem white-box;
4. todas as afirmações anteriores estão erradas.

II) [Princípios] Alguns erros de desenvolvimento da aplicação expectáveis podem originar

1. da má interpretação da especificação do padrão;
2. da incorreta especificação do padrão na literatura;
3. da escolha incorreta do padrão;
4. todas as afirmações anteriores estão erradas.

III) [Princípios] O desenvolvimento de testes automatizados para o padrão anterior pode trazer ganhos consideráveis no futuro, uma vez que

1. todos os testes podem ser utilizados quando o padrão for utilizado em outros cenários;
2. os testes podem ser utilizados como testes de regressão para validar a funcionalidade das versões posteriores da aplicação;
3. vão permitir testar a performance da aplicação;
4. todas as afirmações anteriores estão erradas.

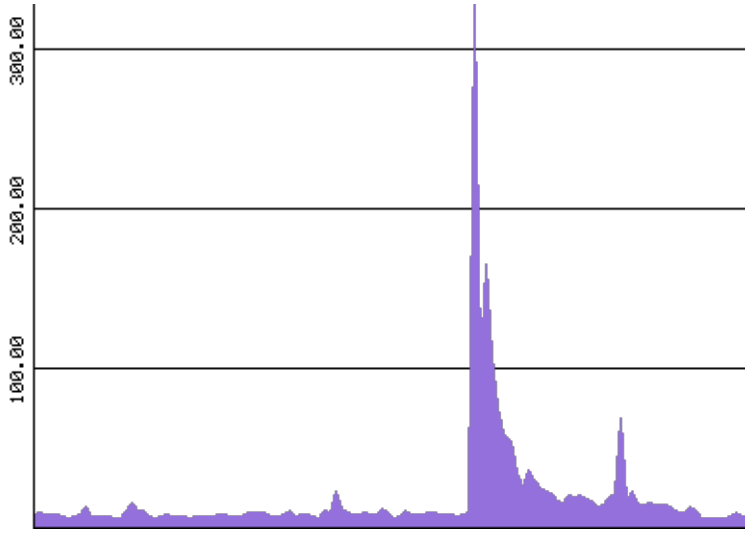
IV) [Inspeções de código] Numa sessão *walkthrough*, o *tester*

1. procede à leitura do código e pede ao programador para explicar as instruções;
2. lidera a sessão, distribuí os materiais pela equipa, regista todos os erros encontrados e assegura a sua posterior correção;
3. identifica um conjunto de testes que são executados de forma a avaliar o comportamento e a evolução do estado da aplicação (variáveis, etc.);
4. todas as afirmações anteriores estão erradas.

V) [Testes de Integração] Ao acrescentar um módulo para armazenar os objetos do tipo *User* numa base de dados, é possível executar testes de integração

1. *top-down* recorrendo a um *driver* para substituir o módulo que comunica com a base de dados;
2. *top-down* recorrendo a um *stub* para substituir o módulo que comunica com a base de dados;
3. *bottom-up* recorrendo a um *driver* para substituir o módulo que comunica com a base de dados;
4. todas as afirmações anteriores estão erradas.

VI) [Testes de Sistema] O gráfico seguinte mostra os pedidos/seg (*requests/sec*) realizados durante o processo de testes da aplicação. Assumindo que a capacidade máxima do sistema é de 200 pedidos/seg



1. o *workload* permite avaliar a disponibilidade da aplicação;
2. o *workload* permite avaliar a robustez da aplicação;
3. o *workload* pode ser utilizado para executar um teste de volume;
4. todas as afirmações anteriores estão erradas.

VII) Indique os *inputs* necessários para testar o método *changePassword()*, recorrendo às técnicas de particionamento de equivalência e análise de valores limite

VIII) Indique os *inputs* necessários para testar o método *addUser()*

IX) Utilizando pseudo-código ou código, apresente o algoritmo do método *handleRequest* (máx: 5 linhas)

X) Descreva os testes *white-box* necessários para testar o código anterior, fazendo a cobertura das decisões.

XI) Utilizando pseudo-código ou código, apresente o algoritmo do *objeto duplo* descrito na questão V) para obter os dados de um utilizador (estrutura da classe e implementação do método).