

Projeto Final

O objetivo do projeto final é o de permitir aos alunos aplicarem os conhecimentos obtidos na disciplina, ao testarem uma API. Deve ser tido em conta as fases do **Processo de Teste** na elaboração deste trabalho. Desde a escolha de recursos, tecnologias e ferramentas, tipos de teste e metodologias a aplicar, especificação de Casos de Teste adequados, execução, resultados e discussão dos resultados.

O projeto final deve ser desenvolvido **individualmente**.

O objeto de testes deste trabalho é uma API de nome **Garage**.

Esta API, que se encontra em desenvolvimento, tem como propósito o processamento de informação de uma oficina. Contém dados de clientes, veículos, reparações e faturas bem como as associações entre eles.

A mesma pode ser descarregada da plataforma de e-learning em formato jar e ser executada com o comando **"java -jar garage.jar"**. Para tal deverá ter a versão de Java 21 instalada na sua máquina. Quando a aplicação inicia é criada uma pasta **"tmp"** onde poderão ser encontrados os ficheiros referentes à base de dados H2, utilizada pela API. Estes ficheiros poderão ser apagados caso se pretenda fazer reiniciar a informação da mesma.

A API estará acessível através do endereço <http://localhost:8080/> e a documentação da mesma está disponível em <http://localhost:8080/swagger-ui/index.html>.

Analise com atenção toda a informação documentada disponível, que deverá ser utilizada como ponto de entrada para a especificação dos testes necessários.

O Aluno deverá escolher para testar, **pelo menos um conjunto** de *endpoints*, dos 4 disponíveis: **Vehicle, Service, Invoice e Client**.

Será necessário **entregar até às 00 horas do dia 9 de Junho de 2024** um relatório sucinto com o trabalho desenvolvido, bem como o código produzido no caso de testes automatizados (evidências de testes manuais podem ser anexadas ao relatório ou ficheiro à parte), e preparar uma breve **defesa/apresentação de 10 min** que será realizada na data do exame, **11 de Junho de 2024**.

Relatório

Como guião do relatório, os pontos essenciais mas não limitativos, são:

- Breve apresentação/Descrição do objeto de teste
- Objetivo dos testes
- Estratégia de Testes (cenário e ambiente)
- Casos de Teste (Descrição de cada teste)
 - o Condições iniciais
 - o Passos a executar
 - o Resultados esperados
- Execução e resultados obtidos
- Discussão/Avaliação dos resultados

Conceitos e Boas práticas

Para a execução deste trabalho é necessário o conhecimento de alguns conceitos específicos, para além de todas as metodologias, processos e termos descritos, no decorrer da disciplina.

Cenário de Teste – Comportamento do sistema a ser testado. *Ex: passos para efetuar login*

Caso de Teste – Validação feita a um cenário de teste. *Ex: validar que mensagem de erro aparece caso o utilizador seja invalido num login*

Especificação de Caso de Teste – Deve conter vários parâmetros que permitam a descrição do caso de teste bem como o registo do resultado.

Nome	CT0001 - Palavra-passe Errada
Descrição	Validar login com Palavra-passe inválida
Prioridade	Alta
Condições iniciais	Deverá estar no ecrã de login
Procedimento	1 - Inserir o utilizador válido 2 - Inserir a palavra-passe inválida 3 - Clicar no botão de login
Dados de Entrada	Utilizador: "user" Palavra-passe: "0000"
Resultado Esperado	Login não deve ter sucesso. Mensagem de erro deve ser visível com o texto: "Dados de acesso inválidos"
Sucesso	Passou
Resultado Atual	Como esperado
Ambiente	Ambiente de Teste; Chrome 86.0.4240.75; Windows 10
Técnica	Manual

Boas práticas

- Casos de teste devem ser simples e transparentes
- Devem ser criados com o utilizador final em mente
- Evitar repetição de testes
- Não assumir nada
- Assegurar a cobertura a 100%
- Casos de teste devem poder ser identificados
- Aplicar técnicas de teste
- Não deixar "lixo"
- Devem poder ser repetidos
- Devem poder ser executados de forma independente

Dicas de Automação

- Os métodos **@Before**, **@After** e da família, devem ser utilizados para a criação de condições iniciais e para a limpeza depois da execução dos testes, respetivamente. Se o teste for verificar que é possível editar um utilizador, o mesmo deve ser criado num método *Before* e apagado num método *After*.

Notas sobre método de execução para avaliação

- Os testes serão corridos contra uma cópia “limpa” da API, pelo que caso seja necessário algum procedimento que não esteja automatizado, o mesmo deve ser descrito num ficheiro README.md na raiz do projeto.

- O trabalho deverá ser submetido através da plataforma de e-learning, bastando a submissão por um elemento do grupo

- Caso se pretenda utilizar alguma plataforma de CVS (GitHub, GitLab, etc) para execução e/ou entrega do trabalho, o mesmo deverá ser feito num repositório privado e ao qual deverá ser dado acesso ao email do professor (no caso de entrega apenas). A referência ao projeto deverá ser indicada num documento a entregar através da plataforma de e-learning.

- O ficheiro executável (.java) da API, alvo de testes, não deverá ser entregue juntamente com o trabalho, bem como os ficheiros e pastas referentes à BD ou processos de “build” (pastas .gradle e build)