

ADVENTURE_5

Setup

Para realizar a quinta parte do projeto deve começar por resolver na totalidade o exercício prático da aula-jmeter.

Para ser efetuado por um único elemento do grupo

Obtenha a nova versão do código disponibilizada em <https://github.com/tecnico-softeng/reference.git>. Instale essa versão no ramo master e crie uma cópia para o ramo develop onde o grupo irá realizar as suas atividades de desenvolvimento. Para isso faça

```
$ git fetch reference
$ git checkout master
$ git merge reference/master
$ git push origin master
```

de seguida deve assegurar-se que o código da quarta parte foi entregue através do ramo fourth-deliver, mudando para esse ramo

```
$ git checkout fourth-deliver
```

pode então apagar o ramo develop, mudar para o ramo master, criar um novo ramo develop a partir do master e colocar a nova versão do develop em origin

```
$ git branch -d develop
$ git checkout master
$ git checkout -b develop
$ git push origin develop --force
```

deve agora correr os testes existentes

```
$ mvn clean test
```

Para ser efetuados pelos restantes elementos do grupo, após as alterações anteriores

Após a nova versão do código estar no GitHub do projeto deve actualizar os ramos locais:

```
$ git fetch --all
$ git checkout master
$ git pull
$ git branch -d develop
$ git checkout -b develop origin/develop
```

deve agora correr os testes existentes

```
$ mvn clean test
```

e abrir o projeto no Eclipse.

Enunciado

A quinta parte do projeto corresponde a efetuar sobre o resultado da quarta parte (quarto sprint), a que foi adicionada a ligação remota entre os diversos módulos, testes de integração e de carga, usando a ferramenta JMeter.

Comece por executar o teste de carga que foi fornecido na diretoria com os testes de JMeter (integration) e que exercita a integração dos diversos módulos. Analise-o e proceda à implementação de todas as restantes combinações de testes de carga que correspondem aos testes de sequência em AdventureSequenceTest.java no módulo broker. O teste fornecido é o primeiro dos restantes 8 que deve implementar.

Analise agora os outros 3 testes de carga fornecidos. Esses testes após a inicialização de dados, efetuada sequencialmente, executam os testes de carga simulando diversos utilizadores simultâneos. Comece por alterar os testes por forma a funcionarem corretamente com as novas interfaces utilizador, por exemplo o hotel passa a necessitar de mais informação na altura da sua criação. De seguida, deve, mantendo a carga definida para cada um dos testes, adicionar instâncias dos módulos car e tax, por forma a que as aventuras sejam processadas na sua totalidade, e considerando que todas as aventuras necessitam de reservar quarto e alugar carro.

Os dados a criar devem ser suficientes para que o processamento das aventuras seja confirmado. Note que os dados a criar têm impacto no sucesso e/ou desempenho de alguns cenários. Por exemplo, se todas as aventuras usarem o mesmo IBAN é maior a possibilidade de conflitos de escrita e o processamento da aventura ter uma maior latência, ou vir mesmo a ser cancelado.

O script JMeter e os ficheiros com os dados a utilizar durante os testes deve ser colocado numa pasta JMeter no directório de topo do projeto, onde vão encontrar já dois subdiretórios, um para os testes de carga e o outro para os testes de integração. Entregue um relatório em que analise os resultados e o impacto que a política otimista implementada pela FénixFramework tem na invocação de serviços remotos. O relatório deve ser elaborado utilizando a linguagem markdown e colocado no diretório de topo do projeto, num ficheiro de nome **report.md**.

Cada aluno do grupo deve implementar pelo menos um dos testes de sequência e deve participar com outro colega num dos testes de carga (3 subgrupos x 2 alunos). **É obrigatório indicar no README.md quais os elementos que trabalham em cada um dos scripts JMeter.**

Para a entrega deverão fazer:

```
$ git checkout develop
$ git checkout -b fifth-deliver
$ git tag ADVENTURE_5
$ git push origin --tags fifth-deliver:fifth-deliver
```

Esta tag colocada deve ter uma data anterior à data limite de entrega, **dia 20 de Maio pelas 20:00**.

Durante os laboratórios, os alunos dos grupos serão avaliados com base no trabalho desenvolvido durante cada uma das semanas. Para isso é necessária a criação de um sprint no GitHub, usando a interface de Project, em que cada tarefa deve ser representada por um Note. Neste caso o projeto a criar deve-se chamar Sprint Five. Uma vez criado o projeto, escolhendo como template pré-definido "Kanban (automatic)", deve ser alterado o nome da coluna To Do para Backlog,. Nesta coluna serão colocadas todas as tarefas a realizar, as quais serão definidas durante a reunião do grupo. Cada tarefa é criada como uma Note, contendo a descrição da tarefa a realizar, e deve ser convertida num Issue, por forma a que os commits possam ser-lhe associados. Cada aluno que vá realizar uma tarefa move o respetivo note da coluna de Backlog para a coluna In progress, e atribui-se essa tarefa no Issue respetivo. Quando a tarefa é terminada, o commit deve ser associado ao respetivo Issue, passando este para o estado fechado. Para se associar um commit a um Issue deve colocar-se na mensagem de commit close #n, em que n é o número do Issue. Quando o Issue é fechado o respetivo Note é automaticamente passado da coluna In progress para a coluna Done. **Cada commit deve estar associado a uma única tarefa.**

Página Inicial



Grupos

Avaliação

Bibliografia

Horário

Métodos de Avaliação

Objectivos

Planeamento

Programa

Turnos

Anúncios



Sumários



Notas

Resultado dos QUC

Teóricas

Laboratórios

Horário de Dúvidas

Documentação

Projeto

ADVENTURE_1

ADVENTURE_2

ADVENTURE_3

ADVENTURE_4

ADVENTURE_5

FAQ

Extra-Mile
(Prémio
Novabase)

Época Especial

Notas intercalares