

ForkExec – T02

Entrega 2 – Guião de Demonstração

Nome	Número
Catarina Pedreira	87524
Miguel Coelho	87687
Ricardo Silva	87700

Git Repository: <https://github.com/tecnico-distsys/T02-ForkExec>

Instalação e configuração:

1. `git clone https://github.com/tecnico-distsys/T02-ForkExec`
2. `cd T02-ForkExec` (para entrar dentro do diretório correto)
3. `mvn clean install -DskipTests` (para compilar e instalar todos os módulos)
4. `mvn exec:java` (em terminais diferentes, executar em cada módulo de servidor para ligar um servidor do hub-ws, um do rst-ws e um do pts-ws)

5. `mvn exec:java -Dws.i=2` (num terminal diferente, executar no pts-ws para ligar a segunda réplica do pts-ws)
6. `mvn exec:java .Dws.i=3` (num terminal diferente, executar no pts-ws para ligar a terceira réplica do pts-ws)
7. `mvn verify` (para correr os testes em todos os módulos. Para correr individualmente executar o comando no módulo desejado)
8. Para mudar o uddi server, mudar o atributo `<uddi.url>` na pasta pom base e na pom da pasta uddi-naming para o url que queremos

Cenário F1

Um PointsClient estabelece uma ligação com o Hub. De seguida activa a sua conta através da função `activateAccount` e depois quer efetuar uma leitura do seu saldo.

O Front End chama, assincronamente, a operação de leitura (no nosso caso, a função `pointsBalance`) para todos os N gestores de réplicas do servidor de Points disponíveis naquele momento - três no caso do nosso projeto. A partir daqui, é chamada a operação de leitura de cada gestor de réplica. Esta operação recebe um `userEmail` e retorna um tuplo `<Tag, Saldo>`.

Os três servidores estão disponíveis para responder. No entanto, quando o Front End obtém respostas da maioria (no nosso caso, dois servidores), prossegue e devolve o valor que acabou de ler, garantido estar consistente.

Cenário F2

Um PointsClient estabelece uma ligação com o Hub. De seguida activa a sua conta através da função `activateAccount` e depois quer efetuar uma escrita no seu saldo (adicionar pontos).

O Front End chama, assincronamente, a operação de escrita (função `write`) para os três gestores de réplica do servidor de Points disponíveis naquele momento. Novamente, é chamada a operação de escrita para cada um dos gestores de réplica que recebe um `userEmail`, e um tuplo `<Tag, ValorAEscrever>` e retorna um ACK (inteiro).

Poucos segundos depois de as operações de escrita terem sido chamadas para cada gestor de réplica, o terceiro falha.

Como o Front End apenas precisa da resposta de uma maioria, espera apenas pela resposta de dois (os outros dois servidores) e prossegue, sem se ter sequer apercebido que o terceiro servidor falhou.

Quando o terceiro servidor voltar a estar ativo, atualiza os seus valores sem necessitar de intervenção do Front End.