

Sistemas Distribuídos

Guia de Demonstração

1. Obtenção do software

Link para download do **código**: <https://github.com/tecnico-distsys/T01-SAYF>

Download pode ser feito manualmente ou obtido através do comando:

```
git clone https://github.com/tecnico-distsys/T01-SAYF
```

Sugestão: Colocar software no desktop (ambiente de trabalho).

Requisitos para a execução e correto funcionamento do projeto:

- Oracle [JDK 11](#) +
- Apache [Maven](#) 3.6.0+
- Apache [Zookeeper](#)
- Github [Git 2.x](#)
- Biblioteca [zknaming](#) (instalada através do maven - **comando shell**: `mvn install`, com o Zookeeper ligado)

Nota: Todos os requisitos devem estar instalados e configurados para seguir para os próximos passos.

2. Instalação e setup do software

Note-se que os próximos passos serão executados através de janela de comandos - *shell unix*.

1º - Inicializar o servidor Zookeper (`./zkServer.sh start`) .

2º - No diretório `zknaming` deve executar o seguinte comando: `mvn install`

3º - No diretório raíz deve executar o seguinte comando: `mvn install -DskipTests`

3. Inicialização

Para inicializar o projeto **deve** primeiro garantir que o Zookeeper está inicializado como referido na secção anterior.

Comandos para inicializar os diferentes elementos do projeto:

(pressupõe-se que se encontra com uma janela de comandos situada no diretório raíz)

Iniciar depot-server:

- cd depot-server
- mvn exec:java -Dins.i=\$instance - onde instance corresponde ao número da réplica

Iniciar sentry-server:

- cd sentry-server
- mvn exec:java -Dins.i=\$instance - instance é o número da sentry e se pretender executar uma segunda sentry, deve ser alterado o seu nome com a tag:
-Dsentry_loc.name=\$name

Iniciar seeker:

- cd seeker
- mvn exec:java

Iniciar feeder:

- cd feeder
- mvn exec:java

4. Tolerância a faltas

Caso F1 - Rotina de funcionamento normal

Especificações para esta rotina:

- 4 Gestores de réplicas (Depot servers)
- 2 Sentries
- 3 Feeders
- 1 Seeker

Passos para iniciar a rotina:

- ★ Comece por inicializar três gestores de réplicas com os comandos da secção anterior.
- ★ De seguida initialize ambas as sentries, instâncias 1 e 2 onde na 2^a é colocado o nome: ISTALAMEDA

★ Initialize 3 Feeders e 1 Seeker

Note que a ligação aos servidores por partes dos clientes pressupõe de uma escolha aleatória das réplicas disponíveis e registadas no Zookeeper.

Passos para mostrar o funcionamento:

- ★ Insira no feeder 1 os seguintes endereços MAC:
 - 11:22:33:44:66:DD
 - AA:AA:FF:EE:CC:99
 - EE:CC:FF:AA:7F:1D
 - 1A:2D:3F:4E:5A:9E
- ★ Insira no feeder 2 os seguintes endereços MAC:
 - 11:22:44:44:56:ED
 - AA:BB:AA:EE:EC:99
 - EE:FF:FF:AA:7F:1D
 - 1A:2D:9F:4E:5A:9E
- ★ Insira no feeder 3 os seguintes endereços MAC:
 - 11:22:33:44:66:DD
 - AA:AA:FF:EE:CC:99
 - EE:CC:FF:AA:7F:1D
 - 1A:2D:3F:4E:5A:9E
 - 11:22:44:44:56:ED
- ★ Ligue o gestor de réplicas restante (instância 4) e desligue outros dois (2 e 3) pressionando a tecla **ENTER**.
- ★ Irá verificar-se agora, após transmissão de mensagens *Gossip*, que os dados foram sincronizados com o gestor mais novo.
- ★ Desligue o gestor número 1 e ligue o seeker que se irá juntar automaticamente ao gestor 4.
- ★ A inserção dos seguintes comandos no seeker irá permitir validar não só o correto funcionamento da sincronização *Gossip* previamente feita, como o correto funcionamento das funções do depot:
 - trace 1A:
 - track 1A:2D:3F:4E:5A:9E
 - trace 11:22:
 - trace 11:22:44:44:56:ED
 - track :ED
 - trace :9E
- ★ Após a validação dos argumentos, termine a execução do seeker escrevendo: *quit*.
- ★ Desligue os restantes componentes pressionando a tecla **ENTER**.

Caso F2 - Rotina de Tolerância a faltas

Especificações para esta rotina:

- 2 réplicas funcionais (Depot servers)
- 2 réplicas que irão perder a conexão inesperadamente - o programa deverá mostrar a deteção das falhas e a consequente correção.
- 2 Sentries - Pelo menos 1 ligada a uma das réplicas que falham
- 3 Feeders - Pelo menos 1 ligado a uma das réplicas que falham
- 2 Seekers - 1 Ligado a uma das réplicas que falham

Passos para iniciar a rotina:

- ★ Comece por inicializar todas as componentes pela **ordem sequencial**: depot1, depot2, sentry1, feeder1, sentry2, feeder2, feeder3, seeker1, seeker2, depot3, depot4
 - Pode atribuir o nome ISTALAMEDA à sentry2

Mais uma vez, note que a ligação aos servidores por partes dos clientes pressupõe de uma escolha aleatória das réplicas disponíveis e registadas no Zookeeper.

Passos para mostrar o funcionamento e como lida com falhas:

★ Falha no sentry1:

Insira, com o feeder1, os seguintes endereços MAC:

- 11:22:33:44:66:DD
- AA:AA:FF:EE:CC:99

Imagine agora que o sentry1 teve uma falha que o obrigou a parar abruptamente.

Simule essa situação com a combinação de teclas: CTRL + C.

Tente inserir os restantes endereços.

- EE:CC:FF:AA:7F:1D
- 1A:2D:3F:4E:5A:9E

Uma vez que cada ação do feeder é um conjunto de macs, esta situação faz com que nenhum dos macs seja registado, no entanto, a falha é de facto detetada.

Insira os seguintes MACs com sucesso, para utilização nos próximos passos:

- EE:CC:FF:AA:7F:1D
- 1A:2D:3F:4E:5A:9E

★ Falha no depot1 e 2:

Introduza, do mesmo modo, uma falha nestes dois gestores de réplicas. Pode notar que se tentar fazer um comando Trace :9E com o seeker1, este irá produzir um erro em como perdeu a conexão e ligou-se a outro depot diferente. Neste caso, apesar

de não ser um processo automatizado, como seria preferível, a falha é de facto abordada.

★ Leituras inconsistentes:

Esta falha não deve existir, ou seja, quando um cliente vê uma atualização numa réplica 1, ao fazer o mesmo pedido para uma réplica 2, essa atualização não poderá ter desaparecido. Tal pode ser verificado com os seguintes passos:

- Feeder3 adiciona MACs EE:CC:FF:AA:7F:1D e 1A:2D:3F:4E:5A:1D
- Seeker lê esse “trace :1D” do depot a que está ligado.
- Desliga-se esse depot.
- Volta-se a executar o mesmo comando no seeker.
- Deverá ter o mesmo resultado

Com estas demonstrações é possível entender o funcionamento do código, assim como a sua tolerância a um leque de faltas que podem ocorrer, como leituras inconsistentes, propagação de modificações nas réplicas, perdas de mensagens, entre outros.