

## Guião de Demonstração – Grupo A63

### Instruções de instalação e configuração:

- Abrir um terminal na pasta onde irão ficar os ficheiros do projeto.
  - Executar nesse terminal o comando “git clone <https://github.com/tecnico-distsys/A63-Komparator.git>”
  - Executar nesse terminal o comando “mvn clean install -DskipTests”
- 

## 2ª Entrega

### Instruções para correr testes da 2ª Entrega:

- Seguir Instruções de instalação e configuração em cima descritas.
  - Abrir um terminal na pasta “supplier-ws” e executar comando “mvn compile exec:java”
  - Abrir outro terminal na pasta “supplier-ws” e executar comando “mvn exec:java -Dws.i=2”
  - Abrir outro terminal na pasta “supplier-ws” e executar comando “mvn exec:java -Dws.i=3”
  - Abrir um terminal na pasta “mediator-ws” e executar comando “mvn compile exec:java”
  - Abrir um terminal na pasta “mediator-ws-cli” e executar comando “mvn verify”
- 

## SOAP Handlers

### Instruções para testar funcionamento dos SOAP handlers criados:

- Seguir Instruções de instalação e configuração em cima descritas.
  - Abrir um terminal na pasta “supplier-ws” e executar o comando “mvn compile exec:java”
  - Abrir um terminal na pasta “supplier-ws-cli” e executar o comando “mvn compile exec:java”
  - É possível verificar o bom funcionamento tanto do envio e receção de mensagens, que são imprimidas em cada terminal, como dos LoggingHandler, TemporalValidationHandler e SignatureHandler
- 

## Segurança

### (S1) Instruções para testar funcionamento normal:

- Seguir Instruções de instalação e configuração em cima descritas.
- Abrir um terminal na pasta “supplier-ws” e executar comando “mvn compile exec:java”
- Abrir outro terminal na pasta “supplier-ws” e executar comando “mvn exec:java -Dws.i=2”
- Abrir um terminal na pasta “mediator-ws” e executar comando “mvn compile exec:java”
- Abrir um terminal na pasta “mediator-ws-cli” e executar comando “mvn -Dtest=BuyCartIT test”
- É possível verificar que os processos correm de forma segura ao analisar as mensagens SOAP que aparecem nos terminais das pastas “mediator-ws” e “mediator-ws-cli”, na qual se verifica que o cartão de crédito está corretamente encriptado (devido ao bom funcionamento do CipherCCHandler)
- É também possível verificar que existem *signatures* nas mensagens entre os terminais das pastas “supplier-ws” e “mediator-ws” (correspondente à relação entre Supplier-ws e Supplier-ws-cli), evidenciando o bom funcionamento do SignatureHandler.

**(S2) Instruções para testar resistência a um ataque:**

- Seguir Instruções de instalação e configuração em cima descritas.
  - Abrir um terminal na pasta “supplier-ws” e executar comando “mvn compile exec:java”
  - Abrir um terminal na pasta “supplier-ws-cli” e executar comando “mvn -Dtest=GetProductIT test”
  - Um dos testes em GetProductIT fará com que o AttackHandler altere o ID do produto “XPTO” para “HACKED”. Na recepção da mensagem o SignatureHandler verifica que esta foi alterada e por isso é inválida, lançando uma RuntimeException (esperada no teste que a provoca), mantendo assim a integridade das mensagens. Isto é observável ao verificar que no terminal da pasta “supplier-ws-cli” todos os testes passam com sucesso
- 

**Replicação****(R1) Instruções para testar funcionamento normal:**

- //TODO Auto-generated instruction description

**(R2) Instruções para testar tolerância a falta:**

- //TODO Auto-generated instruction description