

# Guia de demonstração

## 1. Instalação

Ir á página do repositório Github <https://github.com/tecnico-distsys/A58-SD18Proj> , abrir a página das Releases, e descarregar o ficheiro zip ou tar.gz do código do projeto da tag SD\_P3.

Obter código do laboratório de criptografia, com os módulos kerby-ws, kerby-ws-cli e kerby-lib. Abrir o terminal e executar o comando “mvn clean install -DskipTests” para cada pasta raiz dos módulos. Isto permite guardar no repositório local m2 do computador a versão mais recente.

Abrir outro terminal na raiz do projeto A58-SD18Proj, e executar “mvn clean install -DskipTests”.

## 2. Configuração e demonstração do funcionamento normal com testPing

Abrir duas janelas terminal na pasta raid do projeto, A58-SD18Proj

Inserir o comando “cd binas-ws-cli” numa janela, e na outra inserir “cd binas-ws”

Inserir o comando “mvn compile exec:java” primeiro no terminal respetivo ao servidor (binas-ws) e de seguida fazer o mesmo para o outro terminal.

Em ambos os terminais é possível observar as mensagens SOAP que são enviadas e recebidas.

### 3.1. Demonstração do controlo de acessos

Para observar o correto funcionamento do controlo de acessos, é necessário correr os testes BinasAuthorizationIT na pasta dos testes.

Para correr o teste, devemos:

- Entrar no módulo binas-ws no terminal e inserir o comando “mvn clean compile exec:java”
- Entrar no módulo binas-ws-cli no terminal e inserir o comando “mvn test -Dtest=BinasAuthorizationIT”

O método de teste setCreditOfAnotherUser demonstra o servidor a rejeitar o pedido como pretendido.

### 3.2. Demonstração da verificação de integridade

Para observar o correto funcionamento da verificação de integridade, é necessário correr os métodos de teste `success` e `failIntegrity` presentes no ficheiro de testes `MessageIntegrityIT` na pasta dos testes.

Para correr o teste, devemos:

- Entrar no módulo `binas-ws` no terminal e inserir o comando `"mvn clean compile exec:java"`
- Entrar no módulo `binas-ws-cli` no terminal e inserir o comando `"mvn test -Dtest=MessageIntegrityIT"`

O método de teste `failIntegrity`, permite demonstrar o servidor a rejeitar o pedido, dado que após ter gerado o `digest`, é indicado ao `MACHandler` para inserir um elemento de texto adicional ao corpo da mensagem apenas com o objetivo de fazer falhar a verificação de integridade feita no Servidor.