

**Relatório Asist**

**Unidade Curricular:** ASIST

**Trabalho Elaborado por:** José Nuno Mota Teixeira 1200941

Ano letivo

2023/2024

**Índice**

[1 - US 910 (José Teixeira) 2](#_Toc154861539)

[1.1 - Objetivo Final – Conexão a servidor via SSH sem recurso a credenciais de Login 2](#_Toc154861540)

[1º passo - Gerar um Par de Chaves RSA 2](#_Toc154861541)

[2º Passo - Copiar a Chave Pública para o Servidor 2](#_Toc154861542)

[3 º Passo - Confirmar ficheiro de configuração sshd 3](#_Toc154861543)

[4º Passo - Conexão 3](#_Toc154861544)

[2 - US 930 (José Teixeira) 5](#_Toc154861545)

[2.1 - Objetivo Final - Reposição de Backup e Confirmação 5](#_Toc154861546)

[1º Passo – Criação de Script 5](#_Toc154861547)

[3- US 680 (José Teixeira) 8](#_Toc154861548)

[3.1. Avaliação e Planeamento Inicial 8](#_Toc154861549)

[3.2. Estratégias de Resiliência 8](#_Toc154861550)

[3.3. Estratégias de Resposta a Incidentes 8](#_Toc154861551)

[3.4. Testes e Simulações 8](#_Toc154861552)

[3.5. Formação e Consciencialização 9](#_Toc154861553)

[3.6. Monitorização Contínua 9](#_Toc154861554)

[3.7. Coordenação com Terceiros 9](#_Toc154861555)

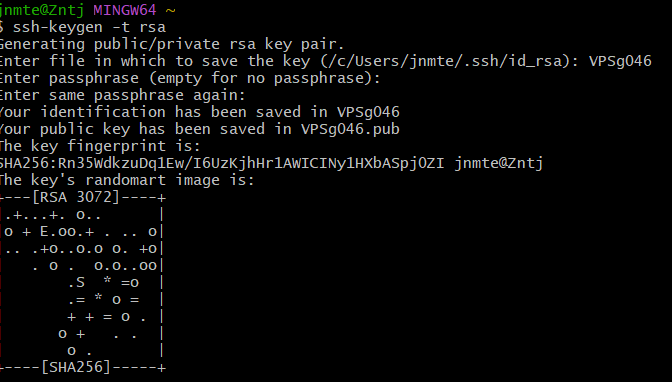
[Conclusão 9](#_Toc154861556)

# **1 - US 910** **(José Teixeira)**

## **1.1 - Objetivo Final – Conexão a servidor via SSH sem recurso a credenciais de Login**

### **1º passo - Gerar um Par de Chaves RSA**

Usar comando “ssh-keygen -t rsa”.



**Explicação do comando** “ssh-keygen -t rsa”

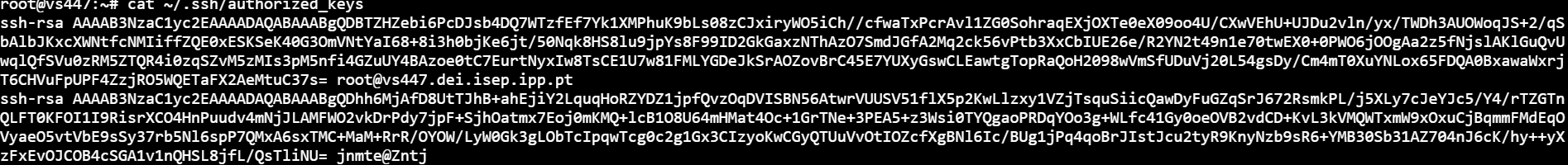
* ssh-keygen: Este é o programa principal usado para criar chaves de autenticação para SSH. Está incluído na maioria dos sistemas Unix e Linux como parte do pacote OpenSSH.
* -t rsa: Este argumento especifica o tipo de chave a ser gerada. No caso, rsa refere-se ao algoritmo RSA (Rivest-Shamir-Adleman), um dos algoritmos de criptografia mais comuns usados para a criação de pares de chaves públicas e privadas.

Se não se especificar um local, este comando usará um local padrão (geralmente ~/.ssh/id\_rsa para a chave privada e ~/.ssh/id\_rsa.pub para a chave pública).

### **2º Passo - Copiar a Chave Pública para o Servidor**

Copiar a chave pública gerada, para o servidor ao qual deseja conectar-se. Este passo pode ser feito de várias formas.

Decidi utilizar uma forma mais manual. Conectei-me ao servidor por ssh com password, colei, ainda, no ficheiro “~/.ssh/authorized\_keys” a chave pública criada anteriormente:

Todo este processo seria possível executar com o comando:

ssh-copy-id [root@vs447.dei.isep.ipp.pt](mailto:root@vs447.dei.isep.ipp.pt)

Não optei por esta solução devido a estar a utilizar uma máquina Windows, a qual cria conflito com a localização da chave pública gerada, por isso decidi utilizar a prática acima indicada.

### **3 º Passo - Confirmar ficheiro de configuração sshd**

Confirmar, se a linha PermitRootLogin no ficheiro de configuração “etc/ssh/sshd\_config” está em “yes” e para uma maior segurança desabilitar a password:

**Uma imagem com texto, Tipo de letra, captura de ecrã, Gráficos

Descrição gerada automaticamente**

### **4º Passo - Conexão**

**Comando:**

ssh -v -i C:/Users/jnmte/VPSg046 [root@vs447.dei.isep.ipp.pt](mailto:root@vs447.dei.isep.ipp.pt)

**Explicação do comando**

-v: Esta opção ativa o modo de depuração (verbose mode) no SSH. Quando usado, o SSH exibe informações detalhadas sobre o processo de conexão. Isso é útil para diagnosticar problemas de conexão, autenticação e configuração.

-i C:/Users/jnmte/VPSg046:

* -i: Esta opção é usada para especificar o caminho da chave privada que será usada para a autenticação com o servidor.

root@vs447.dei.isep.ipp.pt:

* root: Este é o nome de usuário que você está tentando usar para fazer login no servidor remoto. Neste caso, é o utilizador 'root', que é o superusuário em sistemas baseados em Unix/Linux.
* vs447.dei.isep.ipp.pt: Este é o endereço do servidor.

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

# **2 - US 930 (José Teixeira)**

## **2.1 - Objetivo Final - Reposição de Backup e Confirmação**

### **1º Passo – Criação de Script**

Para obter a reposição de Backup e a sua devida confirmação, foi criado um script em que os utilizadores administradores têm permissão de execução.

**Script:** O objetivo deste código é automatizar a restauração de backups de um banco de dados MongoDB e verificar a integridade do backup restaurado. O código permite ao usuário escolher um backup disponível. É de realçar que existem **dependências** para o funcionamento da script.

Devido a ser um sistema Debian (versão 11.3) é necessário seguir os seguintes passos:

* **Adicionar a Chave do Repositório do MongoDB**: Importar a chave pública usada pelo sistema de gestão de pacotes:

**Comando:** “wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add –“

* **Adicione o Repositório do MongoDB**: Criar um arquivo de lista para o MongoDB:

**Comando:** “echo "deb http://repo.mongodb.org/apt/debian buster/mongodb-org/5.0 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list”

* **Atualizar o Repositório de Pacotes:**

**Comando:** “sudo apt update”

* **Instalar o MongoDB:** Agora, instalar o MongoDB. Isso incluirá os comandos mongorestore e mongoexport:

**Comando:** “sudo apt install -y mongodb-org”



O script fornece feedback ao usuário em cada etapa e relata qualquer erro encontrado durante o processo de restauração e verificação. Isso ajuda a garantir que os backups sejam restaurados com sucesso. Para a confirmação do processo de restauro, verifica-se a presença e consistência da coleção "buildings". É também possível ser verificado o número total de coleções repostas.



**Segurança**: Em todos os momentos de utilização direta com a base de dados os dados de login estão guardados no ficheiro config.sh onde as permissões são extremamente restritas de forma a preservar a **confidencialidade, usando “chmod 600 config.sh”**

Existem duas “Strings Connections” para a mesma base de dados, isto deve-se ao “mongorestore” e ao “mongoexport” utilizarem uma sintaxe diferente.

**Explicação detalhada das etapas no script**

* Listagem de Backups:

O script começa por listar todos os backups disponíveis no diretório de backup especificado.

* Seleção de Backup:

O usuário é solicitado a escolher um backup digitando o número correspondente ao backup desejado.

* Restauração do Backup:

O script utiliza o utilitário mongorestore para restaurar o backup selecionado no banco de dados MongoDB especificado nas configurações.

* Verificação de Restauração Bem-Sucedida:

É verificado se a restauração foi concluída com sucesso, verificando o código de saída do comando mongorestore. Se a restauração falhar, o script sai com uma mensagem de erro.

* Verificação da Existência da Coleção "buildings":

O script utiliza o mongoexport para verificar se a coleção "buildings" existe no banco de dados restaurado após a restauração.

# **3- US 680 (José Teixeira)**

## **3.1. Avaliação e Planeamento Inicial**

* **Identificação de Ativos Críticos:** Concentrar-se no módulo de entregas e nas funções de videovigilância.
* **Análise de Riscos:** Identificar ameaças potenciais (como avarias de hardware/software, desastres naturais, ataques cibernéticos) que podem afetar os módulos críticos.
* **Estabelecimento de Objetivos:** Manter a operacionalidade do módulo de entregas e das funções de videovigilância dentro de um limite de 30 minutos de inatividade. As outras funções podem tolerar até 12 horas de interrupção.

## **3.2. Estratégias de Resiliência**

* **Redundância de Sistemas:** Implementar redundância em hardware e software para o módulo de entregas e as funções de videovigilância.
* **Backup e Recuperação de Dados:** Assegurar backups frequentes e testados dos dados críticos.
* **Planos de Contingência para Avarias de Hardware:** Ter hardware de reserva ou acordos com fornecedores para substituições rápidas.

## **3.3. Estratégias de Resposta a Incidentes**

* **Equipa de Resposta a Incidentes:** Formar uma equipa dedicada para responder a interrupções, com foco especial nos módulos críticos.
* **Protocolos de Comunicação:** Estabelecer linhas claras de comunicação interna com os elementos da equipa de resposta e com as partes interessadas (estudantes, funcionários, professores) durante uma interrupção.

## **3.4. Testes e Simulações**

* **Simulações Regulares:** Realizar simulações de desastre para testar a eficácia do plano e a prontidão da equipa.
* **Revisões e Atualizações do Plano:** Avaliar e atualizar o plano regularmente com base nos resultados dos testes e nas mudanças no ambiente de TI (Tecnologia da Informação).

## **3.5. Formação e Consciencialização**

* **Formação de Funcionários:** Garantir que todos os funcionários estejam cientes dos seus papéis em caso de desastre.
* **Consciencialização dos Utilizadores:** Informar os utilizadores (estudantes, funcionários, professores) sobre os procedimentos durante as interrupções.

## **3.6. Monitorização Contínua**

* **Monitorização em Tempo Real:** Utilizar sistemas de monitorização para detetar e responder rapidamente a interrupções.
* **Atualizações de Segurança:** Manter sistemas e software atualizados para prevenir avarias e ataques cibernéticos.

## **3.7. Coordenação com Terceiros**

* **Acordos de Nível de Serviço (SLAs):** Estabelecer SLAs (acordo de nível de serviço) com fornecedores externos para garantir tempos de resposta rápidos.
* **Parcerias Estratégicas:** Formar parcerias com empresas de tecnologia e segurança para apoio adicional.

## **Conclusão**

Este plano de recuperação de desastres deve ser revisto e ajustado regularmente para se adaptar às mudanças nas necessidades da organização e do ambiente tecnológico. O essencial é manter a continuidade das operações críticas minimizando a interrupção e mantendo a segurança e a eficiência.