

**Relatório Asist**

**Unidade Curricular:** ASIST

**Trabalho Elaborado por:**

José Teixeira – 1200941

Jorge Lima – 1210819

João Veríssimo- 1201806

Ano letivo

2023/2024

**Índice**

[**1 - US 910 (José Teixeira)** 2](#_Toc155108002)

[**1.1 - Objetivo Final – Conexão a servidor via SSH sem recurso a credenciais de Login** 2](#_Toc155108003)

[**1º passo - Gerar um Par de Chaves RSA** 2](#_Toc155108004)

[**2º Passo - Copiar a Chave Pública para o Servidor** 2](#_Toc155108005)

[**3 º Passo - Confirmar ficheiro de configuração sshd** 3](#_Toc155108006)

[**4º Passo - Conexão** 3](#_Toc155108007)

[**2 - US 930 (José Teixeira)** 5](#_Toc155108008)

[**2.1 - Objetivo Final - Reposição de Backup e Confirmação** 5](#_Toc155108009)

[**1º Passo – Criação de Script** 5](#_Toc155108010)

[**3- US 680 (José Teixeira)** 8](#_Toc155108011)

[**3.1. Avaliação e Planeamento Inicial** 8](#_Toc155108012)

[**3.2. Estratégias de Resiliência** 8](#_Toc155108013)

[**3.3. Estratégias de Resposta a Incidentes** 8](#_Toc155108014)

[**3.4. Testes e Simulações** 8](#_Toc155108015)

[**3.5. Formação e Consciencialização** 9](#_Toc155108016)

[**3.6. Monitorização Contínua** 9](#_Toc155108017)

[**3.7. Coordenação com Terceiros** 9](#_Toc155108018)

[**Conclusão** 9](#_Toc155108019)

[**4- US 690 (Jorge Lima)** 10](#_Toc155108020)

[**4.1. Maximum Tolerable Downtime** 10](#_Toc155108021)

[**4.2. Significados importantes** 10](#_Toc155108022)

[**4.3. Conclusão** 10](#_Toc155108023)

[**5- US 890 (Jorge Lima)** 12](#_Toc155108024)

[**5.1. Gestão de acessos** 12](#_Toc155108025)

[**5.2. Parâmetros de autorização** 12](#_Toc155108026)

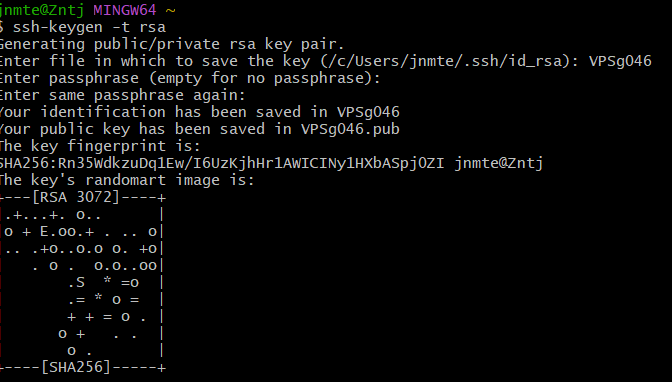
[**5.3. Regras de utilização** 12](#_Toc155108027)

# **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”

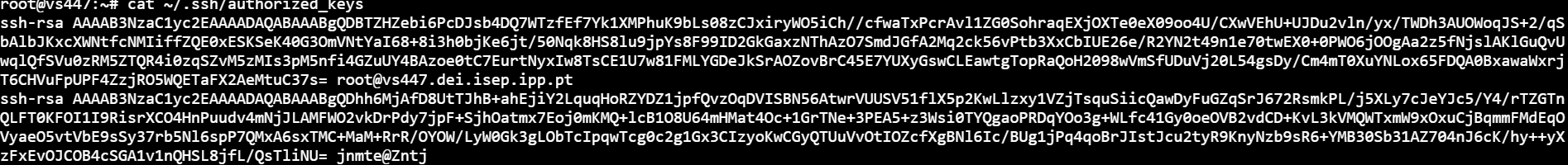
* ssh-keygen: Este é o programa principal usado para criar chaves de autenticação para SSH. Está incluído na maioria dos sistemas Unix e Linux como parte do pacote OpenSSH.
* -t rsa: Este argumento especifica o tipo de chave a ser gerada. No caso, rsa refere-se ao algoritmo RSA (Rivest-Shamir-Adleman),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

# **4- US 690 (Jorge Lima)**

## **4.1. Maximum Tolerable Downtime**

O MTD (Maximum Tolerable Downtime) representa o período máximo durante o qual uma empresa ou organização pode permanecer sem acesso a um sistema ou serviço crítico, antes de sofrer danos significativos nas suas operações e é um parâmetro crucial na avaliação da necessidade de investir em medidas de proteção e resiliência a falhas.

Os MTDs podem variar conforme a prioridade atribuída aos softwares utilizados numa organização, possibilitando diferentes MTDs para diversos departamentos, sistemas ou softwares da empresa.

Uma imagem com texto, captura de ecrã, Tipo de letra, diagrama

Descrição gerada automaticamente

## **4.2. Significados importantes**

1. RTO – Recovery Time Objective: Trata-se de uma técnica para estimar o tempo máximo necessário para que um sistema retorne à sua operação normal após a ocorrência de um problema ou erro. O cálculo desse tempo leva em consideração as prioridades específicas do negócio.
2. WRT – Work Recovery Time: Este é o tempo máximo necessário para realizar testes abrangentes em todos os sistemas, como bases de dados, aplicações e demais componentes. Cada vez mais, busca-se reduzir a dependência de esforço humano nesta fase, tornando a automação um objetivo fundamental e central.
3. RPO – Recovery Point Objective: Refere-se ao ponto no tempo até o qual uma organização está disposta a aceitar a perda de dados em caso de interrupção ou falha no sistema, ou seja, o RPO determina a quantidade máxima de dados que uma organização está disposta a perder durante uma interrupção.

## **4.3. Conclusão**

Para garantir um MTD de 20 minutos, torna-se imperativo implementar medidas de proteção e resiliência a falhas em todas as camadas da infraestrutura. Como por exemplo estas soluções:

* Utilizar soluções de backup e recuperação de desastres é essencial para assegurar a proteção dos dados críticos e possibilitar uma recuperação rápida em caso de falhas.
* Implementação de sistemas de alta disponibilidade, como clusters de servidores, visa garantir a continuidade operacional mesmo diante de falhas em componentes específicos.
* A realização regular de testes de failover para verificar a eficácia dos sistemas de backup e assegurar uma transição suave.
* Medidas de proteção contra falhas humanas, como treino adequado e políticas de mudança controladas, são fundamentais para minimizar a possibilidade de erros humanos que possam comprometer o sistema.
* Manter os sistemas atualizados com as últimas correções de segurança e patches de software é crucial para reduzir a probabilidade de falhas relacionadas a vulnerabilidades conhecidas.

# **5- US 890 (Jorge Lima)**

## **5.1. Gestão de acessos**

Cada utilizador possui um conjunto distinto de critérios para Confidencialidade, Integridade e Disponibilidade (CIA). Esses critérios são fundamentais para guiar as estratégias de segurança da informação.

Desde o início do projeto foram implementados diversos mecanismos de segurança alinhados a esses critérios. Isso inclui a configuração do módulo PAM que condiciona o acesso à solução, o histórico de logins, a encriptação de passwords e a execução regular de backups. Essas medidas visam atender aos requisitos específicos de cada utilizador, contribuindo para a robustez e integridade do sistema.

Para além destas regras foi adicionado o parâmetro *Role* onde cada utilizador só tem acesso às funcionalidades da sua função.

## **5.3. Regras de utilização**

Já para as políticas de acesso, estabelece-se algumas regras para a utilização dos recursos na organização, de entre elas destacam-se, por exemplo:

* Não utilizar contas de outros funcionários da empresa
* Após encontrar um problema técnico informar um administrador imediatamente para que este possa contactar a equipa de desenvolvimento do projeto
* Caso aconteçam erros de conexão com os servidores, contactar a equipa de manutenção.

# **6- US 920 (Jorge Lima)**

## **6.1. Atualizações**

O primeiro passo é verificar se existem atualizações necessárias e se o serviço NFS está instalado.

sudo apt update

sudo apt upgrade

sudo apt install nfs-kernel-server

## **6.2. Partilha do diretório**

É necessário criar o diretório a ser partilhado

sudo mkdir -p /shareFiles

E depois é necessário editar as permissões no ficheiro ‘/etc/exports’

/shareFiles \*(rw,sync,no\_subtree\_check)

Descrição das opções utilizadas:

* “\*”: Permite acesso a qualquer host.
* “rw”: Permite leitura e escrita.
* “sync”: Garante a gravação síncrona nos discos.
* “no\_subtree\_check”: Melhora o desempenho.

## **6.2. Reiniciar o serviço NFS**

Após criar o diretório e definir as permissões é necessário reiniciar o serviço NFS

sudo systemctl restart nfs-kernel-server

# **7-US840(João Veríssimo)**

## **7.1- Objetivo Final – Cópia de segurança da Base de Dados com o nome formatado**

Para realizar esta US optei por criar uma *Action* num repositório GIT destinado apenas a guardar os backups da base de dados do nosso projeto:

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

# **8-US850(João Veríssimo)**

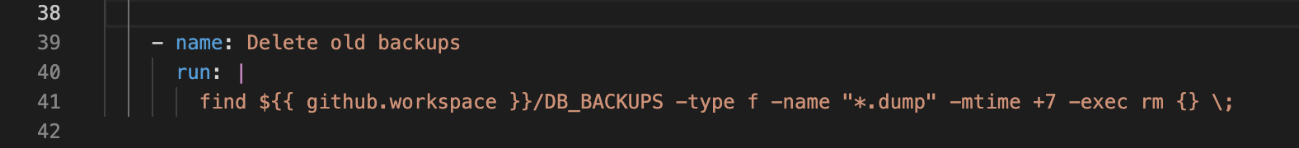
## **8.1-Objetivo Final- Gerir as cópias de segurança conforme pedido pelo cliente.**

**Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente**Para implementar esta US criei a seguinte script, que vai ao diretório onde os backups são guardados e retira de lá sempre o mais recente mediante os critérios, substituindo na pasta onde são guardados os backups pretendidos.

# **9-US870(João Veríssimo)**

## **9.1-Objetivo Final- Limitar o tempo de vida dos backups a 7 dias.**

Para implementar esta US alterei a US 840, uma vez que os backups que se pretende manter estão a ser guardados noutro diretório.

Com este comando, todos os backups que estão na pasta e que tenham um tempo de vida superior a 7 dias serão apagados.