

# Sprint 3 – ASIST

Grupo 28

Tiago Marques – 1201276

Eduardo Silva – 1201371

João Vieira – 1201376

Pedro Alves – 1201381

Pedro Rocha – 1201382

janeiro, 2023

## Índice de conteúdos

US C1 - Como administrador da organização quero um plano de recuperação de desastre que satisfaça o MBCO definido na US B5.....	4
US C2 - Como administrador da organização quero que me seja apresentada de forma justificada a ou as alterações a realizar na infraestrutura por forma a assegurar um MTD (Maximum Tolerable Downtime) de 20 minutos.....	5
US C3 - Como administrador de sistemas quero que seja realizada uma cópia de segurança da(s) DB(s) para um ambiente de Cloud através de um script que a renomeie para o formato _yyyymmdd sendo o nome da base de dados, yyyy o ano de realização da cópia, mm o mês de realização da cópia e dd o dia da realização da cópia. ....	6
US C4 – Como administrador de sistemas quero que utilizando o Backup elaborado na US C3, seja criado um script que faça a gestão dos ficheiros resultantes desse backup, no seguinte calendário. 1 Backup por mês no último ano, 1 backup por semana no último mês, 1 backup por dia na última semana. ....	7
US C5- Como administrador de sistemas quero que o processo da US C3 seja mantido no log do Linux, num contexto adequado, e alertado o administrador no acesso à consola se ocorrer uma falha grave neste processo. ....	10
US C6 - Como administrador de sistemas quero que a cópia de segurança da US C3 tenha um tempo de vida não superior a 7 (sete) dias exceto no indicado na US C4. ....	12
US C7 - Como administrador da organização quero que me seja apresentado um BIA (Business Impact Analysis) da solução final, adaptando se e onde aplicável o(s) risco(s) da US B4.....	13
Determinação dos processos de negócio e a criticidade da sua recuperação .....	13
Análise do Impacto.....	13
US C10 - Como administrador de sistemas quero que o administrador tenha um acesso SSH à máquina virtual, apenas por certificado, sem recurso a password.....	16

## Índice de Tabelas

Tabela 1 - Análise da criticidade .....	13
Tabela 2 - Fundamentação da criticidade .....	15

## Índice de Figuras

Figura 1 - Instalação dos programas de backup da BD .....	6
Figura 2 - Comando para correr o script com ajuda do "Cron".....	6
Figura 3 - Script US C4 .....	7
Figura 4 - Script de verificação da última semana do mês, para guardar o backup do último dia do mês .....	7
Figura 5 - Script para apagar todos os scripts de há 2 anos .....	8
Figura 6 - Definição das "crontables" dos scripts .....	8
Figura 7 - Storage do Azure (antes) .....	8

Figura 8 - Storage do Azure (agora).....	8
Figura 9 - Remoção dos scripts de 2021.....	9
Figura 10 - Definição das "crontables".....	9
Figura 11 - Definição das origens.....	10
Figura 12 - Definição do destino.....	10
Figura 13 - Captação de mensagens que possuam "azclierror".....	10
Figura 14 - Filtragem de erros.....	10
Figura 15 - Montagem da estrutura do log.....	10
Figura 16 - Teste da solução.....	11
Figura 17 - Demonstração do erro de acesso.....	11
Figura 18 - Implementação US C6.....	12
Figura 19 - Comando para correr o script diariamente.....	12
Figura 20 - Criação da chave pública.....	16
Figura 21 - Cópia da chave pública.....	16
Figura 22 - Login via SSH sem password.....	17

## US C1 - Como administrador da organização quero um plano de recuperação de desastre que satisfaça o MBCO definido na US B5.

O **DRP** (*Disaster Recovery Plan*) é um conjunto de ações e procedimentos que garante a recuperação de um sistema ou processo após uma interrupção accidental. Visa minimizar o impacto desses eventos e garantir a continuidade do negócio. Tais interrupções podem ser causadas devido a:

- Falhas de hardware;
- Problemas de rede;
- Ataques cibernéticos;
- Desastres naturais.

Ao analisar a estrutura do sistema, verificamos que todos os módulos são essenciais para o funcionamento da aplicação e, além disso, também percebemos que a integridade das bases de dados também é vital para a funcionamento da nossa aplicação.

Considerando o que foi mencionado acima, o **DRP** que deve ser adotado é:

1. Verificar a inoperabilidade do sistema, acedendo à solução;
2. Adoção de um novo serviço, como por exemplo a criação de um novo server (em caso de falha do corrente), utilizando os *backups* realizados de modo a restaurar a informação no servidor criado;
3. Contactar os responsáveis pelos serviços interrompidos.

De modo a reduzir os impactos causados por um eventual desastre, os seguintes procedimentos devem ser realizados:

- Realizar *backups* regulares dos sistemas e base de dados;
- Monitorização dos equipamentos;
- Redundância de serviços provedores da solução;
- Utilização de *firewalls*.

US C2 - Como administrador da organização quero que me seja apresentada de forma justificada a ou as alterações a realizar na infraestrutura por forma a assegurar um MTD (Maximum Tolerable Downtime) de 20 minutos.

O **MTD** (*Maximum Tolerable Downtime*) refere-se à quantidade máxima de tempo que uma organização suporta para ter os seus sistemas ou processos críticos indisponíveis devido a uma interrupção ou falha. O MTD ajuda as organizações a determinar o nível de resiliência e redundância necessário nos seus sistemas e processos para garantir que possam prosseguir com a sua operação mesmo diante de uma interrupção inesperada.

O MTD pode ser calculado através do seguinte cálculo:

$$MTD = RTO + WRT$$

Em que:

- O **RTO** (*Recovery Time Objective*) refere-se ao tempo que médio necessário para recuperar um sistema ou serviço após um desastre ou falha inesperada;
- O **WRT** (*Work Recovery Time*) refere-se ao tempo necessário para recuperar um ambiente de trabalho e testá-lo após um desastre ou falha inesperada;

Quanto menor for o MTD, melhor a empresa vai estar preparada para prosseguir com as suas atividades diante de um desastre. Sendo assim, as alterações a realizar na infraestrutura para que o MTD seja menor são:

- Adoção da técnica de *mirroring* dos dados para um local remoto, visto que esta técnica é ideal em situações em que se pretenda que exista uma cópia de segurança pois facilita a recuperação dos mesmos e, além disso, o WRT é praticamente nulo (dependente do tipo de sincronização usado), o que assegura um MTD muito baixo;
- Adoção de cópias integrais de segurança, visto que, apesar de influenciar negativamente o RPO (pois a existe a cópia de todos os dados, sem seleção), tem um tempo menor de reposição da informação, o que corresponde à diminuição do valor do WRT, logo o MDT será mais baixo.

Assumimos que, perante estas alterações na infraestrutura, é assegurado que o MTD seja de 20 minutos, visto que existe escassez de dados relativamente a valores reais de RTO e WRT, pelo que sem esses valores não é calculado um valor preciso de MTD.

US C3 - Como administrador de sistemas quero que seja realizada uma cópia de segurança da(s) DB(s) para um ambiente de Cloud através de um script que a renomeie para o formato `_yyyymmdd` sendo o nome da base de dados, `yyyy` o ano de realização da cópia, `mm` o mês de realização da cópia e `dd` o dia da realização da cópia.

Para realizar este script, foi necessário instalar dois programas para fazer o backup das duas bases de dados: para a base de dados *MariaDB*, instalamos o programa *mysqldump* e para a base de dados *MongoDB*, instalamos o programa *mongodump*.

```
GNU nano 5.4 db_backup.sh
#!/bin/bash

./cleanMtd.sh
#DIR='date +%Y%m%d'
DIR=20221127
DEST=$DIR
cd db_backups

mongodump --uri mongodb+srv://Pentax:0la1ola2ola3@mongodb.detw68e.mongodb.net/teste2 --out /home/db_backups \
&& cat /home/db_backups/teste2/packings.bson /home/db_backups/teste2/paths.bson /home/db_backups/teste2/plannings.bson /home/db_backups/teste2/roles.bson /home/db_backups/teste2/trips.bson \
&& mv /home/db_backups/teste2 /home/db_backups/$DEST \
&& az storage blob upload --account-name pentax --account-key hJsbcnLWHC0/dfSuJnz5idr4eqk0831X6ZFrJmWTGrygLlxS1T7qrgGpmxf4cQw/uM13kwQmaRM3+ASTgS9vsw== --container-name databases --file /home/db_backups/$DEST \
|| echo "Error"

cd $DEST

mysqldump -h "vs736.dei.isep.ipp.pt" -u "root" -p"kjCY5/jSy/ll" WMDatabase > WMDatabase_$DIR.sql \
&& az storage blob upload --account-name pentax --account-key hJsbcnLWHC0/dfSuJnz5idr4eqk0831X6ZFrJmWTGrygLlxS1T7qrgGpmxf4cQw/uM13kwQmaRM3+ASTgS9vsw== --container-name databases --file WMDatabase_$DIR.sql \
|| echo "Error"
```

Figura 1 - Instalação dos programas de backup da BD

De seguida, recolhendo a informação de cada base de dados através da *connection string* no caso da base de dados *MongoDB*, e, através das credencias da base de dados *MariaDB* (host, utilizador e password), guardamos um ficheiro `.sql` para a primeira BD e `.bson` para a segunda BD renomeando para o formato pedido, que contém a informação das bases de dados, numa pasta com nome igual à data de criação do backup.

Posteriormente, foi feito *upload* dos dois ficheiros criados para uma *blob* do Azure (para tal, foi necessário instalar o package Azure CLI, onde ficaram então armazenados).

As cópias de segurança são realizadas diariamente, à meia-noite, através do script acima apresentado. Na imagem seguinte, mostramos como corremos o script automaticamente, com a ajuda do *"Cron"*.

```
0 0 * * * cd /home && ./db_backup.sh
```

Figura 2 - Comando para correr o script com ajuda do "Cron"

US C4 – Como administrador de sistemas quero que utilizando o Backup elaborado na US C3, seja criado um script quer faça a gestão dos ficheiros resultantes desse backup, no seguinte calendário. 1 Backup por mês no último ano, 1 backup por semana no último mês, 1 backup por dia na última semana.

Para a resolução da UC4, realizamos 3 scripts - um para a verificação dos scripts da semana anterior, outro para o mês e por fim, um para o ano.

O objetivo da *User Story* é ao mudar de semana, manter todos os *backups* da semana anterior, mas apagando todos de há duas semanas, com exceção do domingo, que vai ser usado para cumprir os requisitos da gestão dos *backups* na mudança de mês. Este *script* não vai intervir para “limpar” a última semana do mês, visto que esta funcionalidade é realizada no *script* da gestão do mês, de maneira a não guardar o domingo, mas sim o último dia do mês.

```
GNU nano 5.4 lastWeekBackups.sh
cd db_backups

DAY_OF_WEEK=$(date +%u)

FIRST_DAY_OF_THE_MONTH=$(date +%Y-%m-01)

LAST_DAY_OF_THE_PREVIOUS_MONTH=$(date -d "$FIRST_DAY_OF_THE_MONTH -1 day" +%s)
echo "O backup do arquivo $arquivo foi realizado no dia $datacriacao logo, é para ser mantido."
LAST_MONDAY_OF_THE_PREVIOUS_MONTH=$(date -d "$LAST_DAY_OF_THE_PREVIOUS_MONTH" -d "last Monday" +%s)
LAST_MONDAY=$(date -d "last Monday" +%F)
LAST_SUNDAY=$(date -d "last Sunday" +%F)

PENULTIMATE_MONDAY=$(date -d "$LAST_MONDAY - 1 week" +%F)
PENULTIMATE_SUNDAY=$(date -d "$LAST_SUNDAY" +%F)
PENULTIMATE_MONDAY_S=$(date -d "$PENULTIMATE_MONDAY" +%s)
PENULTIMATE_SUNDAY_S=$(date -d "$PENULTIMATE_SUNDAY" +%s)

for arquivo in `ls`; do
    datacriacao=$(date -d "$arquivo" +%Y-%m-%d)
    dia_criacao=$(date -d "$datacriacao" +%s)
    dia_semana=$(date -d "$datacriacao" +%u)

    if [ $dia_criacao -ge $LAST_MONDAY_OF_THE_PREVIOUS_MONTH ] && [ $dia_criacao -le $LAST_DAY_OF_THE_PREVIOUS_MONTH ]; then
        echo "O backup do arquivo $arquivo foi realizado no dia $datacriacao logo, é para ser mantido."
    elif [ $dia_criacao -ge $PENULTIMATE_MONDAY_S ] && [ $dia_criacao -le $PENULTIMATE_SUNDAY_S ]; then
        if [ $dia_semana -eq 7 ]; then
            echo "O backup do arquivo $arquivo foi realizado no dia $datacriacao (domingo) logo, é para ser mantido"
        else
            echo "O backup do arquivo $arquivo foi realizado no dia $datacriacao logo, é para ser eliminado."
            az storage blob delete --account-name pentax --account-key h3sbcnLWHCO/df3uJnz5ldr4eq0831X62FrJmWTGrygLLxS1T7qrg0pmxf4cQw/uM13kVQmaRM3+ASTgS9vsw== --container-name $arquivo
            az storage blob delete --account-name pentax --account-key h3sbcnLWHCO/df3uJnz5ldr4eq0831X62FrJmWTGrygLLxS1T7qrg0pmxf4cQw/uM13kVQmaRM3+ASTgS9vsw== --container-name $arquivo
        fi
    fi
done
```

Figura 3 - Script US C4

Quando mudava de mês, no mês anterior era suposto ficar um *backup* por cada semana, então apagamos todos os *backups* de há 2 meses, tirando o último do mês, para assim cumprir com os requisitos da gestão dos *backups* na mudança do ano. Para além disso, este *script* vai verificar a última semana do último mês, para assim guardar o *backup* do último dia do mês.

```
GNU nano 5.4 lastMonthBackups.sh
cd db_backups
CURRENT_MONTH=$(date +%m)

FIRST_DAY_OF_THE_MONTH=$(date +%Y-%m-01)

LAST_DAY_OF_THE_PREVIOUS_MONTH=$(date -d "$FIRST_DAY_OF_THE_MONTH -1 day" +%d)
LAST_DAY_OF_THE_PENULTIMATE_MONTH=$(date -d "$FIRST_DAY_OF_THE_MONTH -1 month -1 day" +%d)
LAST_MONDAY_OF_THE_PREVIOUS_MONTH=$(date -d "$LAST_DAY_OF_THE_PREVIOUS_MONTH" -d "last Monday" +%d)

CURRENT_DATE=$(date +%Y-%m-%d)

if [ $CURRENT_MONTH -eq 1 ]; then
    LAST_MONTH=12
else
    LAST_MONTH=$((CURRENT_MONTH - 1)) #MES PARA DEIXAR 1 por semana
fi

MONTH_TO_DELETE=$((LAST_MONTH - 1)) #MES PARA APAGAR TUDO MENOS ULTIMA SEMANA

CURRENT_YEAR=$(date +%Y)

for arquivo in `ls`; do
    datacriacao=$(date -d "$arquivo" +%Y-%m-%d)
    dia_da_semana=$(date -d "$datacriacao" +%u)
    mes_criacao=$(date -d "$datacriacao" +%m)
    dia_criacao=$(date -d "$datacriacao" +%d)

    if [ $mes_criacao -eq $LAST_MONTH ]; then
        if [ $dia_criacao -ge $LAST_MONDAY_OF_THE_PREVIOUS_MONTH ] && [ $dia_criacao -lt $LAST_DAY_OF_THE_PREVIOUS_MONTH ]; then
            echo "O backup do arquivo $arquivo foi criado no dia $datacriacao logo, é para apagar"
            az storage blob delete --account-name pentax --account-key h3sbcnLWHCO/df3uJnz5ldr4eq0831X62FrJmWTGrygLLxS1T7qrg0pmxf4cQw/uM13kVQmaRM3+ASTgS9vsw== --container-name $arquivo
            az storage blob delete --account-name pentax --account-key h3sbcnLWHCO/df3uJnz5ldr4eq0831X62FrJmWTGrygLLxS1T7qrg0pmxf4cQw/uM13kVQmaRM3+ASTgS9vsw== --container-name $arquivo
        fi
        echo "O backup do arquivo $arquivo foi criado no dia $datacriacao, ou seja, no último dia do mês logo, é para ser mantido"
    elif [ $mes_criacao -eq $MONTH_TO_DELETE ]; then
        if [ $dia_criacao -eq $LAST_DAY_OF_THE_PENULTIMATE_MONTH ]; then
            echo "O arquivo $arquivo foi criado no dia $datacriacao, ou seja, no último dia do mês logo, é para ser mantido"
        else
            echo "O backup do arquivo $arquivo foi criado no dia $datacriacao logo, é para apagar"
            az storage blob delete --account-name pentax --account-key h3sbcnLWHCO/df3uJnz5ldr4eq0831X62FrJmWTGrygLLxS1T7qrg0pmxf4cQw/uM13kVQmaRM3+ASTgS9vsw== --container-name $arquivo
            az storage blob delete --account-name pentax --account-key h3sbcnLWHCO/df3uJnz5ldr4eq0831X62FrJmWTGrygLLxS1T7qrg0pmxf4cQw/uM13kVQmaRM3+ASTgS9vsw== --container-name $arquivo
        fi
    fi
done
```

Figura 4 - Script de verificação da última semana do mês, para guardar o backup do último dia do mês

Para finalizar, quando mudava de ano, o ano anterior tinha de ter 1 *backup* por mês, então apagamos todos os *backups* de há 2 anos, visto que o requisito de ter 1 *backup* por mês já foi cumprido no *script* do mês.



```
GNU nano 5.4 lastYearBackups.sh
#!/bin/bash

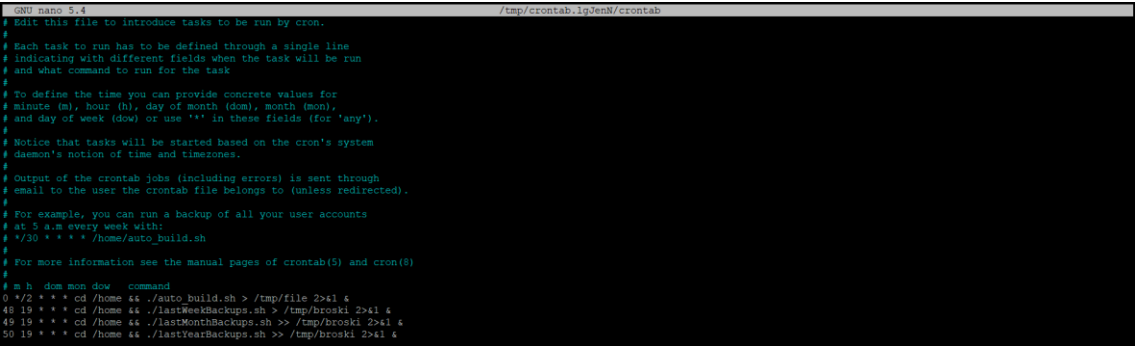
cd db_backups

ANO=$(date +%Y)
#echo $ANO
ANOAFAGAR=$((ANO - 2))
ANOANTERIOR=$((ANO - 1))
#echo "ANOAFAGAR"

for arquivo in `ls`; do
    cd $arquivo
    data_criacao=$(date -d "$arquivo" +%Y-%m-%d)
    ano_criacao=$(date -d "$data_criacao" +%Y)
    if [ $ano_criacao -eq $ANOAFAGAR ]; then
        echo "O arquivo $arquivo foi criado em $ano_criacao logo, é para apagar"
        az storage blob delete --account-name pentax --account-key h3bcnLWBC0/dfSu3hz5idr4eqK0831X68Fr3mW7Gryglxsl7grg9mx4cqW/uM13kwQmaRM3+ASTg9vsw --container-name database
    fi
    cd ..
done
```

Figura 5 - Script para apagar todos os scripts de há 2 anos

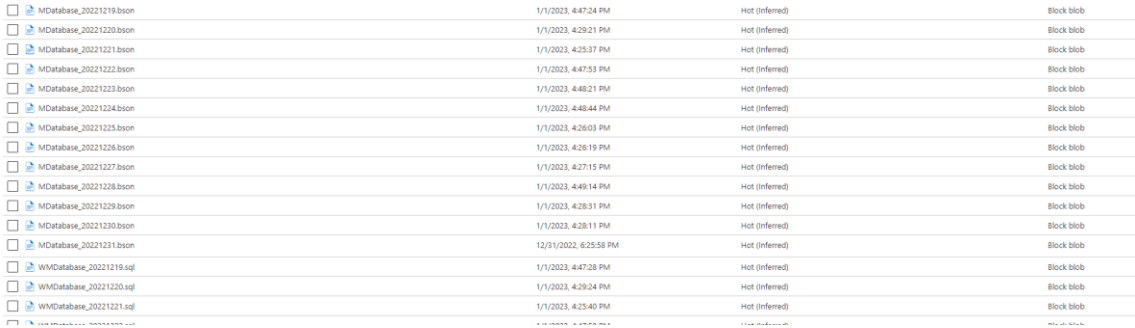
Após a realização dos scripts, definimos as “*crontables*” dos mesmos, primeiramente definindo os 3 *scripts* para uma hora específica, de maneira a poder confirmar os outputs e se estava a remover do servidor *cloud* “*Azure*” como previsto



```
GNU nano 5.4 /tmp/crontab.lq3en8/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m. every week with:
# */5 * * * * /home/auto_build.sh
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
0 */5 * * * cd /home 44 ./auto_build.sh > /tmp/file 2>&1 &
48 19 * * * cd /home 44 ./lastWeekBackups.sh > /tmp/broski 2>&1 &
49 19 * * * cd /home 44 ./lastMonthBackups.sh >> /tmp/broski 2>&1 &
50 19 * * * cd /home 44 ./lastYearBackups.sh >> /tmp/broski 2>&1 &
```

Figura 6 - Definição das “*crontables*” dos scripts

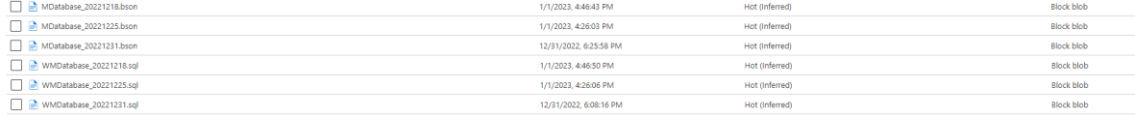
O antes da nossa *storage* do *azure*:



	MDatabase_20221219.bson	1/1/2023, 4:47:24 PM	Hot (Inferred)	Block blob
	MDatabase_20221220.bson	1/1/2023, 4:29:21 PM	Hot (Inferred)	Block blob
	MDatabase_20221221.bson	1/1/2023, 4:25:37 PM	Hot (Inferred)	Block blob
	MDatabase_20221222.bson	1/1/2023, 4:47:53 PM	Hot (Inferred)	Block blob
	MDatabase_20221223.bson	1/1/2023, 4:48:21 PM	Hot (Inferred)	Block blob
	MDatabase_20221224.bson	1/1/2023, 4:48:44 PM	Hot (Inferred)	Block blob
	MDatabase_20221225.bson	1/1/2023, 4:26:03 PM	Hot (Inferred)	Block blob
	MDatabase_20221226.bson	1/1/2023, 4:26:19 PM	Hot (Inferred)	Block blob
	MDatabase_20221227.bson	1/1/2023, 4:27:15 PM	Hot (Inferred)	Block blob
	MDatabase_20221228.bson	1/1/2023, 4:48:14 PM	Hot (Inferred)	Block blob
	MDatabase_20221229.bson	1/1/2023, 4:28:31 PM	Hot (Inferred)	Block blob
	MDatabase_20221230.bson	1/1/2023, 4:28:11 PM	Hot (Inferred)	Block blob
	MDatabase_20221231.bson	12/31/2022, 6:25:58 PM	Hot (Inferred)	Block blob
	WMDatabase_20221219.sql	1/1/2023, 4:47:28 PM	Hot (Inferred)	Block blob
	WMDatabase_20221220.sql	1/1/2023, 4:29:24 PM	Hot (Inferred)	Block blob
	WMDatabase_20221221.sql	1/1/2023, 4:25:40 PM	Hot (Inferred)	Block blob

Figura 7 - *Storage* do *Azure* (antes)

E o depois:



	MDatabase_20221218.bson	1/1/2023, 4:46:43 PM	Hot (Inferred)	Block blob
	MDatabase_20221225.bson	1/1/2023, 4:26:03 PM	Hot (Inferred)	Block blob
	MDatabase_20221231.bson	12/31/2022, 6:25:58 PM	Hot (Inferred)	Block blob
	WMDatabase_20221218.sql	1/1/2023, 4:46:50 PM	Hot (Inferred)	Block blob
	WMDatabase_20221225.sql	1/1/2023, 4:26:06 PM	Hot (Inferred)	Block blob
	WMDatabase_20221231.sql	12/31/2022, 6:08:16 PM	Hot (Inferred)	Block blob

Figura 8 - *Storage* do *Azure* (agora)



Este foi o output gerado pelos *scripts* todos, como podemos confirmar, simulamos hoje (1 de janeiro de 2023) a mudança de semana, mês e ano, então os *backups* de há duas semanas são para ser removidos, para além dos de domingo. Os da semana passada, para ser mantidos.

No caso do script do mês, remove todos os backups de há dois meses, deixando apenas um backup da última semana. Relativamente ao mês anterior, remove todos os backups diários, à exceção dos realizados no domingo.

Para terminar, no caso do ano, vão ser removidos os *scripts* de 2021.

```
GNU nano 5.4 broski
O backup do arquivo 20221219 foi realizado no dia 2022-12-19 logo, é para ser eliminado.
O backup do arquivo 20221220 foi realizado no dia 2022-12-20 logo, é para ser eliminado.
O backup do arquivo 20221221 foi realizado no dia 2022-12-21 logo, é para ser eliminado.
O backup do arquivo 20221222 foi realizado no dia 2022-12-22 logo, é para ser eliminado.
O backup do arquivo 20221223 foi realizado no dia 2022-12-23 logo, é para ser eliminado.
O backup do arquivo 20221224 foi realizado no dia 2022-12-24 logo, é para ser eliminado.
O backup do arquivo 20221225 foi realizado no dia 2022-12-25 (domingo) logo, é para ser mantido
O backup do arquivo 20221226 foi realizado no dia 2022-12-26 logo, é para ser mantido.
O backup do arquivo 20221227 foi realizado no dia 2022-12-27 logo, é para ser mantido.
O backup do arquivo 20221228 foi realizado no dia 2022-12-28 logo, é para ser mantido.
O backup do arquivo 20221229 foi realizado no dia 2022-12-29 logo, é para ser mantido.
O backup do arquivo 20221230 foi realizado no dia 2022-12-30 logo, é para ser mantido.
O backup do arquivo 20221231 foi realizado no dia 2022-12-31 logo, é para ser mantido.
O backup do arquivo 20221227 foi criado no dia 2022-12-27 logo, é para apagar
O backup do arquivo 20221228 foi criado no dia 2022-12-28 logo, é para apagar
O backup do arquivo 20221229 foi criado no dia 2022-12-29 logo, é para apagar
O backup do arquivo 20221230 foi criado no dia 2022-12-30 logo, é para apagar
O backup do arquivo 20221231 foi criado no dia 2022-12-31, ou seja, no último dia do mês logo, é para ser mantido
O arquivo 20210101 foi criado em 2021 logo, é para apagar
```

Figura 9 - Remoção dos *scripts* de 2021

Após finalizar os testes, definimos então corretamente as “*crontables*”:

```
GNU nano 5.4 /tmp/crontab.luvTVR/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
#
# */30 * * * * /home/auto_build.sh
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# h dom mow dow command
0 */2 * * * cd /home && ./auto_build.sh > /tmp/file 2>&1 &
0 0 * * 1 cd /home && ./lastWeekBackups.sh > /tmp/broski 2>&1 &
0 0 1 * * cd /home && ./lastMonthBackups.sh >> /tmp/broski 2>&1 &
0 0 1 1 * cd /home && ./lastYearBackups.sh >> /tmp/broski 2>&1 &
```

Figura 10 - Definição das “*crontables*”

US C5- Como administrador de sistemas quero que o processo da US C3 seja mantido no log do Linux, num contexto adequado, e alertado o administrador no acesso à consola se ocorrer uma falha grave neste processo.

Para a realização desta *User Story*, utilizamos o *syslog-ng*.

Primeiramente foi preciso ativar os *logs*, tanto do *azure*, como do *mysql*. Após isso, com as pastas dos *logs* já definidas, dentro do ficheiro de configuração do *syslog-ng* definimos as origens de quando vem dessas duas mesmas pastas:

```
source s_file_az {  
    file("/var/log/azure/az.log");  
};  
  
source s_file_mysql {  
    file("/var/log/mysql/log");  
};
```

Figura 11 - Definição das origens

Após a definição da origem, vamos então definir o destino, neste caso a pasta “*motd*” que é a pasta que escreve no ecrã pós *login*, de maneira ao utilizador conseguir ver que ocorreram erros:

```
# Destination for db backup logs  
destination d_dbbackup { file("/etc/motd"); };
```

Figura 12 - Definição do destino

Depois de definido a origem e o destino vamos então definir os filtros para definir que erros captar.

No caso dos erros vindos do *azure*, captamos todas as mensagens que tenham “*azclierror*”, todas as mensagens com este excerto de texto, são erros do *azure* que consideramos importante mencionar ao utilizador.

```
filter f_azure_cli_errors {  
    match("azclierror" value("MESSAGE"));  
};
```

Figura 13 - Captação de mensagens que possuam “azclierror”

No caso dos erros vindos do *mysql*, consideramos todos os erros desde o nível “*notice*” até ao nível “*emerg*”, visto que a partir do nível “*notice*” começam a aparecer erros como *password errada*.

```
filter f_notice_or_more_severe { level(notice .. emerg); };
```

Figura 14 - Filtragem de erros

Para terminar, montamos a estrutura do *log*.

```
log { source(s_file_mysql); filter(f_notice_or_more_severe); destination(d_dbbackup); };  
log { source(s_file_az); filter(f_azure_cli_errors); destination(d_dbbackup); };
```

Figura 15 - Montagem da estrutura do log

Para então testar a solução, corremos os seguintes *scripts*:

```
root@vs705:~# az storage blob upload --account-name pentax --account-key h3abcnMHC9/dfSuJnz5ldr4eQK0831X6ZF7JmTGrYglx5177grgopexf4cQw/UM13kQmaR03+ASTg59vsw== --container-name databases --file /home/db_b
ackups/20221231/WMDatabase_20221231.sql --name WMDatabase_20221231.sql
root@vs705:~# mysqldump -h "vs736.dei.isep.ipp.pt" -u "root" -p"kJCY5/jSy/l" WMDatabase > WMDatabase_$(DIR).sql
```

Figura 16 - Teste da solução

No primeiro script, já existe no *azure* um *backup* com o mesmo nome, gerando assim o erro esperado. Já no caso do segundo *script*, colocamos mal a password da base de dados, dando assim erro de acesso negado à mesma.

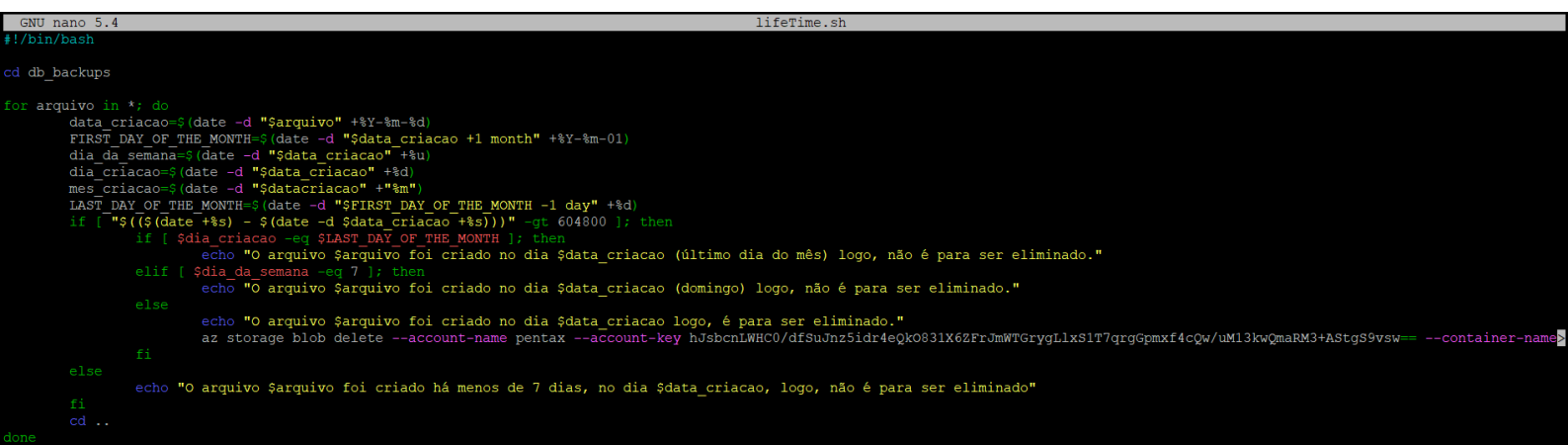
Ambos os erros são listados no menu de *login*.

```
Jan  1 22:51:54 vs705 96601 : 2023-01-01 22:51:53,585 : ERROR : cli.azure.cli.core.azclierror : The specified blob already exists.
Jan  1 22:53:13 vs705 mysqldump: Got error: 1045: "Access denied for user 'root'@'10.9.22.193' (using password: YES)" when trying to connect
Last login: Sun Jan  1 22:20:12 2023 from 10.8.56.199
root@vs705:~#
```

Figura 17 - Demonstração do erro de acesso

US C6 - Como administrador de sistemas quero que a cópia de segurança da US C3 tenha um tempo de vida não superior a 7 (sete) dias exceto no indicado na US C4.

Para a realização desta US, criamos um *script* que compara a data de criação de cada ficheiro de *backup* da base de dados com a data atual e, se tiver sido criado há mais de 7 dias, o ficheiro é eliminado. No entanto, e considerando os *scripts* criados na US C4, os ficheiros de *backup* da base de dados criados a um domingo e um dos *backups* realizados na última semana do respetivo mês são mantidos.



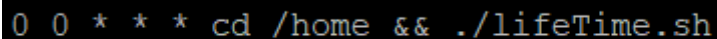
```
GNU nano 5.4 lifeTime.sh
#!/bin/bash

cd db_backups

for arquivo in *; do
    data_criacao=$(date -d "$arquivo" +%Y-%m-%d)
    FIRST_DAY_OF_THE_MONTH=$(date -d "$data_criacao +1 month" +%Y-%m-01)
    dia_da_semana=$(date -d "$data_criacao" +%u)
    dia_criacao=$(date -d "$data_criacao" +%d)
    mes_criacao=$(date -d "$datacriacao" +%m)
    LAST_DAY_OF_THE_MONTH=$(date -d "$FIRST_DAY_OF_THE_MONTH -1 day" +%d)
    if [ "$((($(date +%s) - $(date -d $data_criacao +%s)))" -gt 604800 ]); then
        if [ $dia_criacao -eq $LAST_DAY_OF_THE_MONTH ]; then
            echo "O arquivo $arquivo foi criado no dia $data_criacao (último dia do mês) logo, não é para ser eliminado."
        elif [ $dia_da_semana -eq 7 ]; then
            echo "O arquivo $arquivo foi criado no dia $data_criacao (domingo) logo, não é para ser eliminado."
        else
            echo "O arquivo $arquivo foi criado no dia $data_criacao logo, é para ser eliminado."
            az storage blob delete --account-name pentax --account-key hJsbcnLWHCO/dFSuJnz5idr4eQk0831X6ZFrJmWTGrygLLxS1T7qrgGpmxf4cQw/uM13kwQmaRM3+ASTgS9vsw== --container-name $arquivo
        fi
    else
        echo "O arquivo $arquivo foi criado há menos de 7 dias, no dia $data_criacao, logo, não é para ser eliminado"
    fi
done
```

Figura 18 - Implementação US C6

Este script corre todos os dias, à meia-noite, automaticamente, com a ajuda do “Cron”.



```
0 0 * * * cd /home && ./lifeTime.sh
```

Figura 19 - Comando para correr o script diariamente

## US C7 - Como administrador da organização quero que me seja apresentado um BIA (Business Impact Analysis) da solução final, adaptando se e onde aplicável o(s) risco(s) da US B4

O Business Impact Analysis tem como objetivo identificar e priorizar os processos de negócio, correlacionando-os com o impacto existente no negócio quando um ou mais processos se encontram indisponíveis.

### Determinação dos processos de negócio e a criticidade da sua recuperação

Na US B4, foram tratados os riscos envolvidos na solução preconizada. Como os diferentes módulos do projeto são serviços significantes - p.e., o módulo de logística e o SPA necessitam do módulo de Gestão de Armazéns para o seu funcionamento - considerou-se a Falha dos Servidores do como probabilidade ocasional e impacto catastrófico.

Relativamente à criticidade, a mesma resulta da aferição do impacto na reputação e do impacto no não cumprimento do **RTO** (*Recovery Time Objective*) e o **RPO** (*Recovery Time Objective*).

- O **RTO** corresponde ao limite máximo de tempo aceitável perder no processo de recuperação após disrupção
- O **RPO** corresponde ao limite de tempo de dados que é aceitável perder antes da disrupção.
- O **MTD** (*Maximum Time of Disruption*) corresponde à soma do RTO e WRT, o tempo limite em que é aceitável o processo estar indisponível.

### Análise do Impacto

Como indicado pelo cliente, “são aceites pequenos períodos de indisponibilidade inferiores a 1 hora e que, preferencialmente, o sistema deve ser resiliente o suficiente para suportar o funcionamento parcial (alguns módulos disponíveis)”, desta forma, considerou-se o RTO de criticidade Muito alta >20m visto que na US C2 é solicitado assegurar um MTD de 20 minutos e o MTD tem de ser igual ou superior ao RTO, uma vez que o MTD é o período máximo onde a ocorrência de falhas é tolerável, já no RTO é aconselhável.

Criticidade	RTO	RPO	MTD
Muito alta	>20m e <4h	>0h e <1h	>20m e <6h
Alta	>4h e <12h	>1h e <12h	>4h e <2h
Moderada	>12h e <5d	>12h e <5d	>1d e <7d
Baixa	>5d e <14d	>5d e <7d	>5d e <20d
Muito Baixa	>14d e <30d	>7d e <14d	>14d e <40d

Tabela 1 - Análise da criticidade

Na tabela seguinte, fundamentou-se a criticidade associada a cada componente.

Componente	RTO	RPO	MTD	Criticidade	Fundamentação
SPA (Angular)	1h	30m	1h	Muito alta	O sistema referido tem como principal responsabilidade apresentar toda a interface gráfica ao utilizador da aplicação. Este sistema usa o sistema MDL (Node.js)", de forma a aceder toda a informação relativamente à gestão de logística, o sistema MGA (.NET)" de forma a aceder a toda a informação relativamente à gestão de armazéns e ao sistema Planeamento (SWI-PROLOG) de modo a apresentar a funcionalidade de certos algoritmos da aplicação.
MDL (Node.js)	8h	8h	24h	Alta	Este sistema é utilizado para serem definidas as <i>entities</i> e os <i>value objects</i> relacionados à gestão de logística e dá uso ao sistema "Base de Dados (Mongo DB) de forma a armazená-las e o controlo dos dados é feito pelo módulo de SPA. Em caso de desastre, não existe comunicação com a base de dados nem com SPA tornando as funcionalidades relacionadas à logística indisponíveis.
Planeamento (SWI-PROLOG)	12h	1h	32h	Moderada	O sistema referido tem como principal responsabilidade armazenar e executar todos os algoritmos necessários para a otimização da criação de rotas. Em caso de desastre, a aplicação fica sem o componente de otimização de rotas, o que não impossibilita o funcionamento da aplicação, somente a torna ineficiente.
Base de Dados Mongo DB	8h	8h	24h	Alta	Este sistema é utilizado para controlo da informação inserida, alterada e eliminada do sistema em causa. Em caso de desastre, a aplicação tem a sua funcionalidade extremamente reduzida, já que não se torna possível o armazenamento de dados.

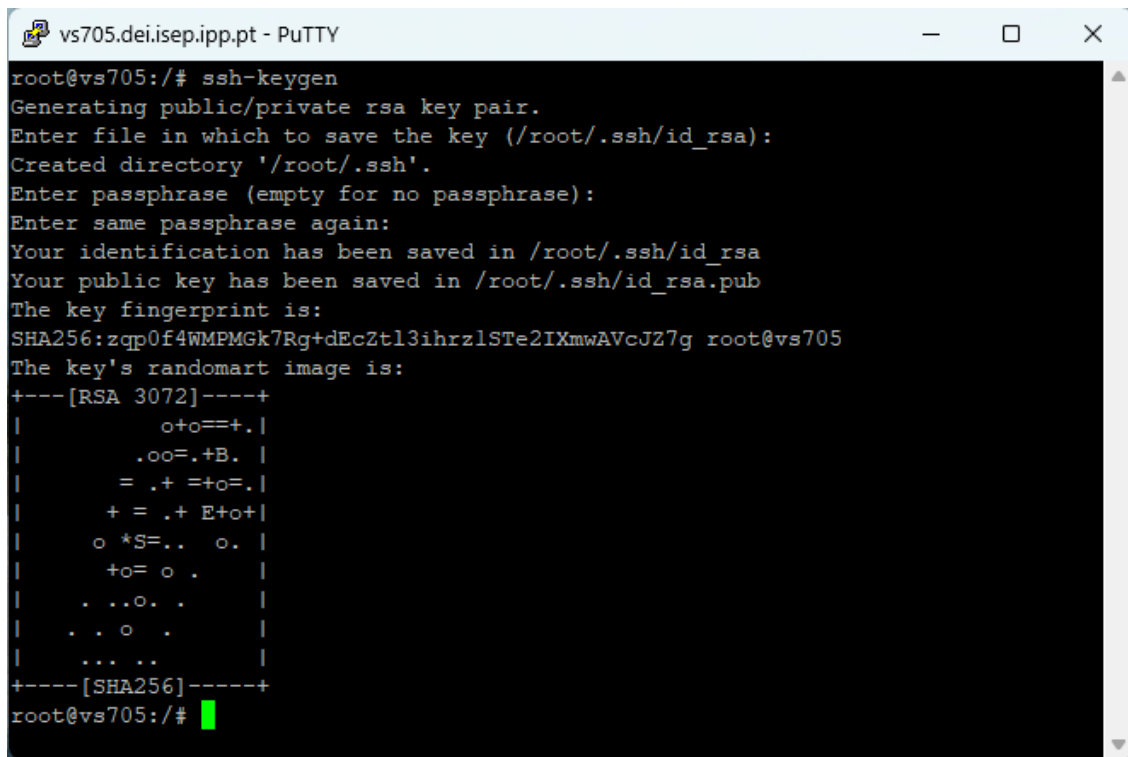
MGA (.NET)	8h	8h	24h	Alta	Este sistema é utilizado para serem definidas as <i>entities</i> e os <i>value objects</i> relacionados à gestão de logística, o controlo dos dados é feito pelo módulo de SPA. Em caso de desastre, não existe comunicação com a base de dados nem com SPA tornando as funcionalidades relacionadas à gestão de armazéns indisponíveis.
------------	----	----	-----	------	--

*Tabela 2 - Fundamentação da criticidade*

## US C10 - Como administrador de sistemas quero que o administrador tenha um acesso SSH à máquina virtual, apenas por certificado, sem recurso a password

O acesso SSH de uma máquina virtual pode ser realizado apenas através de certificados nem a necessidade de uma password. É considerado mais seguro face ao método de utilização de *password* uma vez que faz uso de criptografia de chave pública.

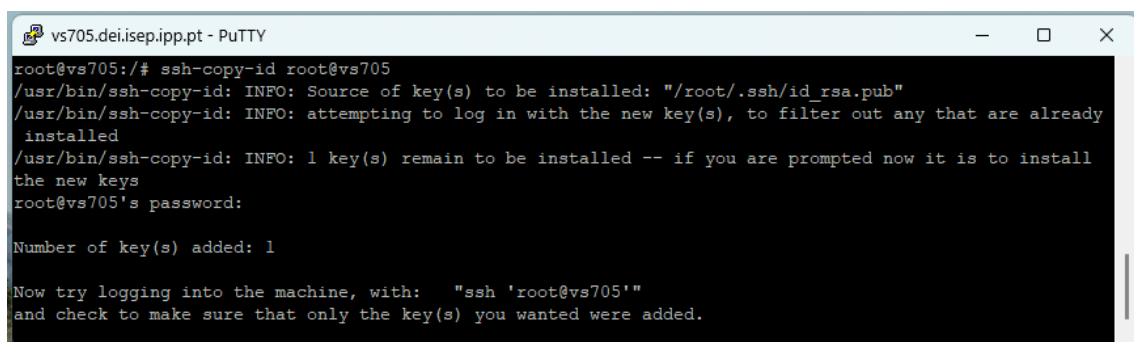
Num primeiro momento foi gerada a chave pública através do comando “**ssh-keygen**”.



```
vs705.dei.isep.ipp.pt - PuTTY
root@vs705:/# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zqp0f4WMPMGk7Rg+dEcZtl3ihrzlSTe2IXmwAVcJZ7g root@vs705
The key's randomart image is:
+---[RSA 3072]-----+
|      o+o==+.|
|      .oo=.+B.|
|      = .+ =+o=.|
|      + = .+ E+o+|
|      o *S=.. o.|
|      +o= o .|
|      . ..O. .|
|      . . o .|
|      ... ..|
+---[SHA256]-----+
root@vs705:/#
```

Figura 20 - Criação da chave pública

De seguida, copiou-se a chave pública para ser habilitada a conexão SSH sem senha através do comando “**ssh-copy-id root@vs705**”.



```
vs705.dei.isep.ipp.pt - PuTTY
root@vs705:/# ssh-copy-id root@vs705
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already
installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install
the new keys
root@vs705's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@vs705'"
and check to make sure that only the key(s) you wanted were added.
```

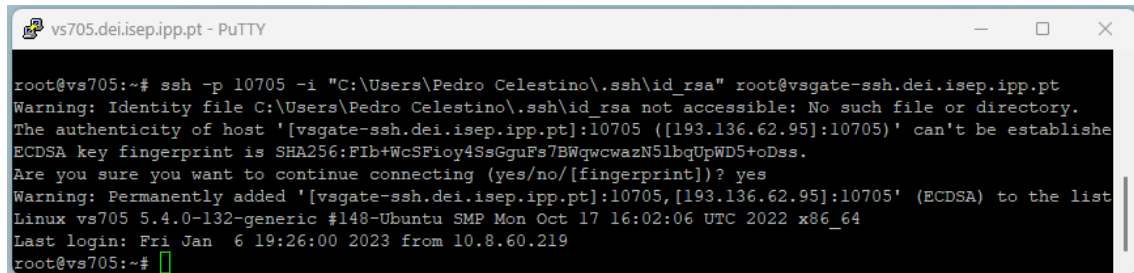
Figura 21 - Cópia da chave pública



Procedeu-se à transferência da private key (id\_rsa) para um local de acesso restrito ao administrador "C:\Users\Pedro Celestino\SSH\ssh" p.e.

Para ser efetuado o login via ssh com certificado usa-se o comando no terminal do Windows:

**ssh -p 10705 -i "C:\Users\Pedro Celestino\SSH\ssh" root@vsgate.ssh.dei.isep.ipp.pt**



```
vs705.dei.isep.ipp.pt - PuTTY
root@vs705:~# ssh -p 10705 -i "C:\Users\Pedro Celestino\SSH\id_rsa" root@vsgate.ssh.dei.isep.ipp.pt
Warning: Identity file C:\Users\Pedro Celestino\SSH\id_rsa not accessible: No such file or directory.
The authenticity of host '[vsgate.ssh.dei.isep.ipp.pt]:10705 ([193.136.62.95]:10705)' can't be established.
ECDSA key fingerprint is SHA256:F1b+WcSFioy4SsGguFs7BWqwcwazN5lbqUpWD5+oDss.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[vsgate.ssh.dei.isep.ipp.pt]:10705, [193.136.62.95]:10705' (ECDSA) to the list of known hosts.
Linux vs705 5.4.0-132-generic #148-Ubuntu SMP Mon Oct 17 16:02:06 UTC 2022 x86_64
Last login: Fri Jan 6 19:26:00 2023 from 10.8.60.219
root@vs705:~#
```

*Figura 22 - Login via SSH sem password*