

# Relatório de Atividades

## Trabalho de Instalação Parte 1

Bryan De Lima Naneti Barbosa - 202121026 - 10A  
Gustavo Soares Silva - 202120103 - 10A  
Rafael Brunini Pereira - 202120488 - 10A

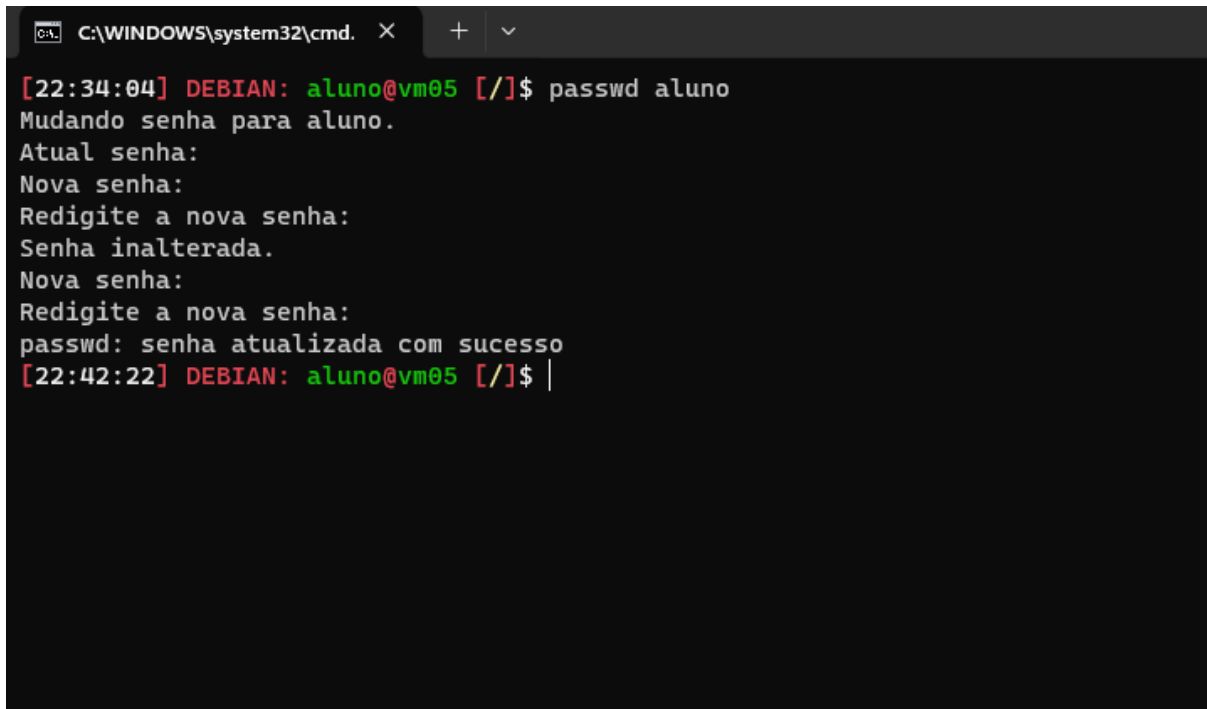
Vms Utilizadas :  
Vm05(192.168.1.5)  
Vm06(192.168.1.6)

<b>Alteração de senhas das VM's.....</b>	<b>3</b>
Etapas executadas:.....	3
1. Uso do comando passwd aluno.....	3
<b>Configuração do serviço de servidor web.....</b>	<b>4</b>
Etapas executadas:.....	4
Passo-a-Passo:.....	4
1. Instalação do servidor web Apache:.....	4
2. Verificação do status do Apache:.....	5
3. Criação da página web:.....	5
4. Configuração do suporte a HTTPS:.....	6
Problemas/Dificuldades encontradas e Soluções Adotadas:.....	7
Solução que arrumamos para enviar os arquivos:.....	8
<b>Configuração do serviço de sincronização de hora utilizando o protocolo NTP.....</b>	<b>8</b>
Passo-a-passo:.....	8
1. Instalar o Chrony na Vm06(Servidor) com o comando abaixo:.....	8
2. Mudar o arquivo /etc/chrony/chrony.conf com o comando sudo nano.....	9
3. Reiniciar o Chrony com o comando abaixo.....	9
4. Verificação da data e hora da VM6.....	9
5. Instalação do pacote "chrony" na VM5.....	10
6. Mudar o arquivo /etc/chrony/chrony.conf com o comando sudo nano.....	10
7. Verificação da data e hora da VM5.....	11
Problemas/Dificuldades encontradas e Soluções Adotadas:.....	11

# Alteração de senhas das VM's

Etapas executadas:

## 1. Uso do comando passwd aluno



```
C:\WINDOWS\system32\cmd. X + v
[22:34:04] DEBIAN: aluno@vm05 [/]$ passwd aluno
Mudando senha para aluno.
Atual senha:
Nova senha:
Redigite a nova senha:
Senha inalterada.
Nova senha:
Redigite a nova senha:
passwd: senha atualizada com sucesso
[22:42:22] DEBIAN: aluno@vm05 [/]$ |
```

# Configuração do serviço de servidor web

## Etapas executadas:

Instalação e configuração do servidor web Apache, criação de uma página web e configuração do suporte a HTTPS.

## Passo-a-Passo:

### 1. Instalação do servidor web Apache:

Executamos o seguinte comando para instalar o Apache:

```
sudo apt update
```

```
sudo apt-get install apache2
```

```
[22:49:17] DEBIAN: aluno@vm05 [/var]$ sudo apt install apache2
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
Os seguintes pacotes foram instalados automaticamente e já não são necessários:
  libevent-core-2.1-7 libevent-pthreads-2.1-7 libopts25 sntp
Utilize 'sudo apt autoremove' para os remover.
Pacotes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
Os NOVOS pacotes a seguir serão instalados:
  apache2
0 pacotes atualizados, 1 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 0 B/278 kB de arquivos.
Depois desta operação, 641 kB adicionais de espaço em disco serão usados.
A seleccionar pacote anteriormente não seleccionado apache2.
(Lendo banco de dados ... 34619 ficheiros e directórios actualmente instalados.)
A preparar para desempacotar .../apache2-2.4.56-1-deb11u2_amd64.deb ...
A descompactar apache2 (2.4.56-1-deb11u2) ...
Configurando apache2 (2.4.56-1-deb11u2) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
[22:49:34] DEBIAN: aluno@vm05 [/var]$
```

## 2. Verificação do status do Apache:

Executamos o seguinte comando para verificar se o Apache está em execução:

```
sudo systemctl status apache2
```

```
[22:51:13] DEBIAN: aluno@vm05 [/var/www/html]$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-10-19 22:49:32 -03; 2min 31s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 58769 (apache2)
    Tasks: 55 (limit: 1115)
   Memory: 8.8M
      CPU: 65ms
   CGroup: /system.slice/apache2.service
           └─58769 /usr/sbin/apache2 -k start
           └─58771 /usr/sbin/apache2 -k start
           └─58772 /usr/sbin/apache2 -k start
```

Verificamos que o status estava "active (running)".

## 3. Criação da página web:

Navegamos até o diretório padrão do Apache:

cd /var/www/html e editamos o arquivo para a página web:

```
sudo nano /var/www/html/index.html
```

```
C:\WINDOWS\system32\cmd. x + v
[22:55:18] DEBIAN: aluno@vm05 [/var/www/html]$ sudo nano index.html|
```

Dentro do arquivo, adicionamos o seguinte conteúdo HTML, com os nomes dos integrantes do grupo e o relatório de atividades desenvolvidas:

```
GNU nano 5.4
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <title>Grupo TCP</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      margin: 0;
      padding: 0;
    }
    header {
      background-color: #333;
      color: #fff;
      text-align: center;
      padding: 10px;
    }
    h1 {
      margin-top: 20px;
      text-align: center;
    }
    p {
      margin-left: 20px;
    }
    h3 {
      margin-left: 20px;
    }
    a {
      display: block;
      margin: 10px 0 0 20px;
      text-decoration: none;
      color: #0077bb;
    }
    a:hover {
      text-decoration: underline;
    }
  </style>
</head>
<body>
  <header>
    <h1>Grupo TCP</h1>
  </header>
  <h1>Integrantes:</h1>
  <p>Bryan Naneti - Turma 10A - Matrícula: 202121026</p>
  <p>Gustavo Soares - Turma 10A - Matrícula: 202120103</p>
  <p>Rafael Brunini - Turma 10A - Matrícula: 202120488</p>

  <h3>Arquivos:</h3>
  <a href="/trabalhos/trabalhoPratico1.pkt">Trabalho Prático 1 - Packet Tracer</a>
</body>
</html>
```

## 4. Configuração do suporte a HTTPS:

Geramos um certificado autoassinado usando o seguinte comando:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

```
C:\WINDOWS\system32\cmd. X + v
[23:02:13] DEBIAN: aluno@vm05 [/]$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
Generating a RSA private key
.....+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:Minas Gerais
Locality Name (eg, city) []:Lavras
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UFLA
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:192.168.1.5
Email Address []:
[23:03:03] DEBIAN: aluno@vm05 [/]$
```

\* Preenchemos as informações solicitadas para o certificado.

Criamos um arquivo de configuração adicional para o Apache:

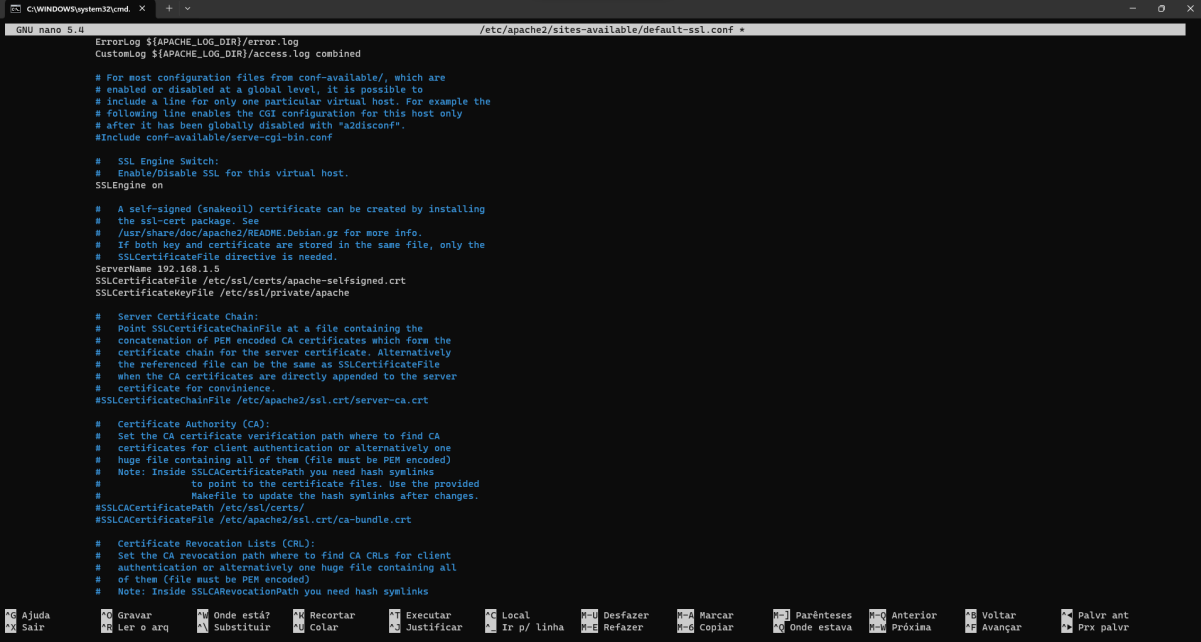
```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Dentro do arquivo, apagamos as linhas do "SSLCertificateFile" e "SSLCertificateKeyFile" e adicionamos:

```
ServerName 192.168.1.5
```

```
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/apache
```



```
GNU nano 5.4 /etc/apache2/sites-available/default-ssl.conf
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

#
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

#
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
ServerName 192.168.1.5
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache

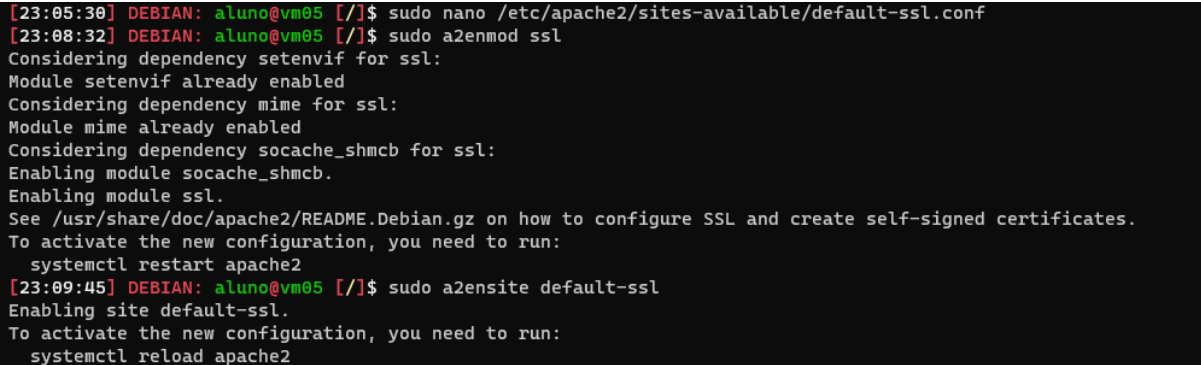
#
# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#
# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCertificatePath you need hash symlinks
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
#SSLCertificatePath /etc/ssl/certs/
#SSLCertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

#
# Certificate Revocation Lists (CRL):
# Set the CA revocation path where to find CA CRLs for client
# authentication or alternatively one huge file containing all
# of them (file must be PEM encoded)
# Note: Inside SSLCARevocationPath you need hash symlinks
```

Por fim: Ativamos o site SSL e reiniciamos o Apache:

Usando os Comandos :

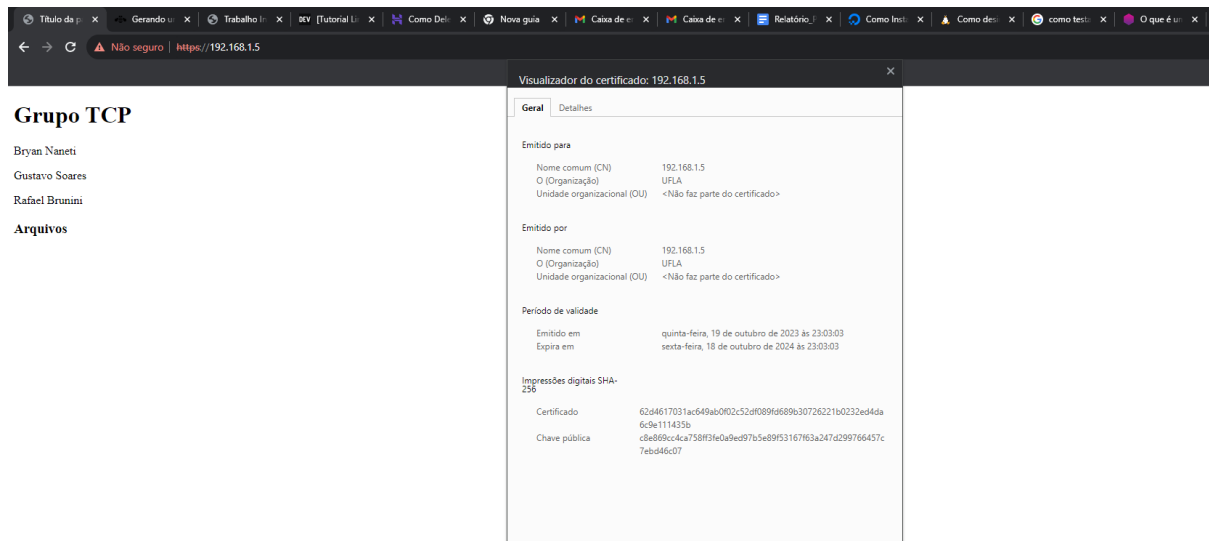


```
[23:05:30] DEBIAN: aluno@vm05 [/]$ sudo nano /etc/apache2/sites-available/default-ssl.conf
[23:08:32] DEBIAN: aluno@vm05 [/]$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
[23:09:45] DEBIAN: aluno@vm05 [/]$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

## Problemas/Dificuldades encontradas e Soluções Adotadas:

Durante o processo de configuração não conseguimos um certificado tsl/ssl uma vez que as organizações responsáveis pela emissão deste certificado não conseguiriam acesso a nosso servidor uma vez que ele é privado e necessita de uma conexão por vpn para

acesso. Para solucionar este problema implementamos um certificado autoassinado através do openssl. Para verificar a funcionalidade do ssl conectamos na máquina utilizando o https e depois verificamos se o navegador reconhecia o certificado como é demonstrado na captura de tela abaixo.



Solução que arrumamos para enviar os arquivos:

Para isso foi necessário alterar a propriedade de usuário do diretório pelo comando:  
**chown aluno /var/www/html/trabalhos**

Depois utilizamos o scp para enviar o arquivo por ssh:

```
PS C:\Users\gusta> scp "C:\Users\gusta\OneDrive\Documentos\AreaTrabalhoEscolar\UFLA\5º Período\redes\trabalhoPratico1.pk
t" aluno@192.168.1.5:/var/www/html/trabalhos
aluno@192.168.1.5's password:
trabalhoPratico1.pkt                                     100% 72KB 498.4KB/s 00:00
PS C:\Users\gusta> |
```

## Configuração do serviço de sincronização de hora utilizando o protocolo NTP.

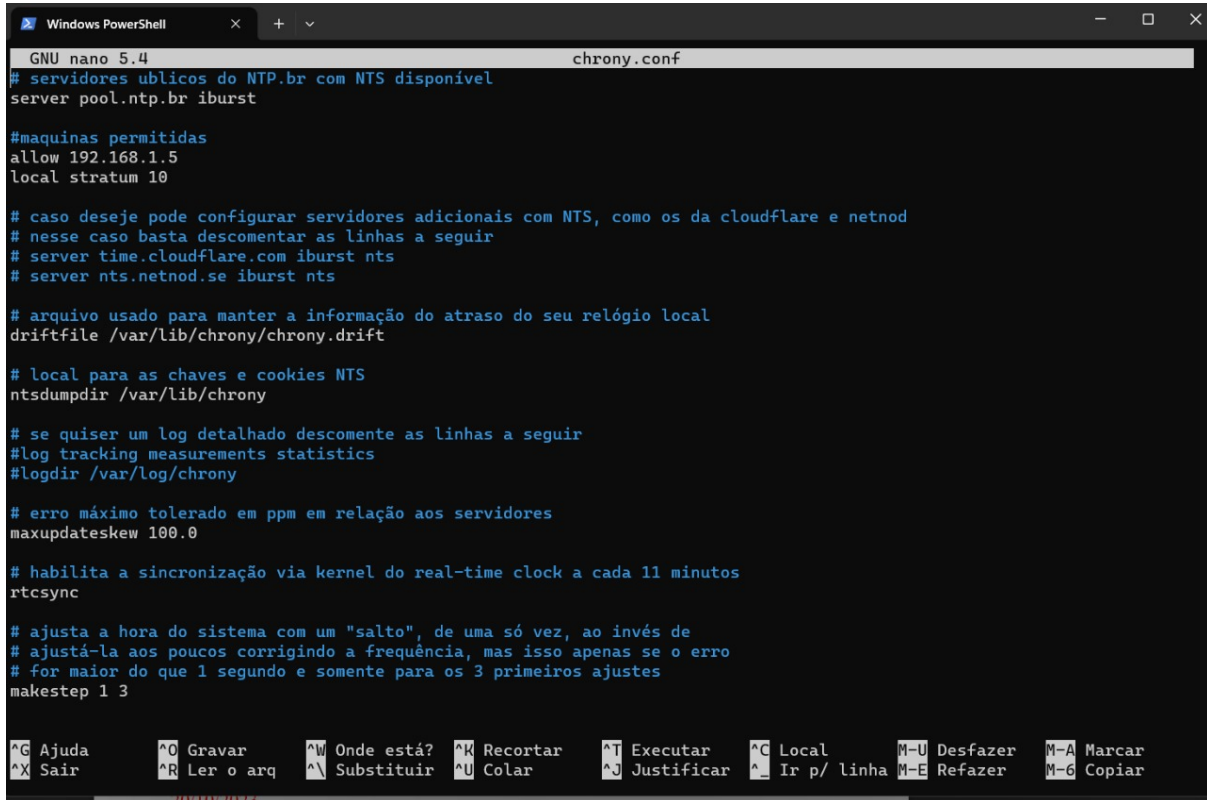
### Passo-a-passo:

1. Instalar o Chrony na Vm06(Servidor) com o comando abaixo:

```
apt-get install chrony
```



## 2. Mudar o arquivo /etc/chrony/chrony.conf com o comando sudo nano



```
GNU nano 5.4 chrony.conf
# servidores publicos do NTP.br com NTS disponivel
server pool.ntp.br iburst

#maquinas permitidas
allow 192.168.1.5
local stratum 10

# caso deseje pode configurar servidores adicionais com NTS, como os da cloudflare e netnod
# nesse caso basta descomentar as linhas a seguir
# server time.cloudflare.com iburst nts
# server nts.netnod.se iburst nts

# arquivo usado para manter a informação do atraso do seu relógio local
driftfile /var/lib/chrony/chrony.drift

# local para as chaves e cookies NTS
ntsdumpdir /var/lib/chrony

# se quiser um log detalhado descomente as linhas a seguir
#log tracking measurements statistics
#logdir /var/log/chrony

# erro máximo tolerado em ppm em relação aos servidores
maxupdateskew 100.0

# habilita a sincronização via kernel do real-time clock a cada 11 minutos
rtcsync

# ajusta a hora do sistema com um "salto", de uma só vez, ao invés de
# ajustá-la aos poucos corrigindo a frequência, mas isso apenas se o erro
# for maior do que 1 segundo e somente para os 3 primeiros ajustes
makestep 1 3

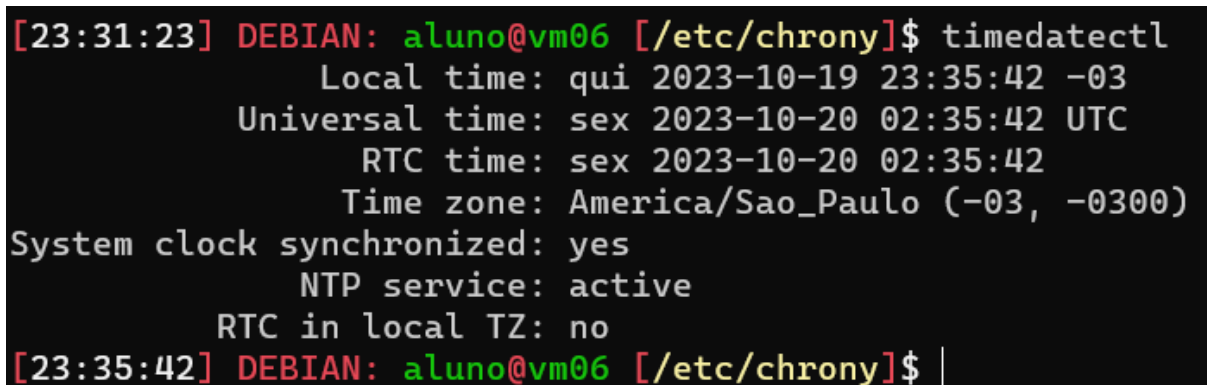
^G Ajuda      ^O Gravar    ^W Onde está? ^K Recortar   ^T Executar   ^C Local     M-U Desfazer  M-A Marcar
^X Sair      ^R Ler o arq ^N Substituir ^U Colar      ^J Justificar ^_ Ir p/ linha M-E Refazer  M-G Copiar
```

## 3. Reiniciar o Chrony com o comando abaixo

```
sudo systemctl chrony restart
```

## 4. Verificação da data e hora da VM6

Utilizamos o comando "timedatectl" para verificar se a data e hora da VM6 estão corretas.



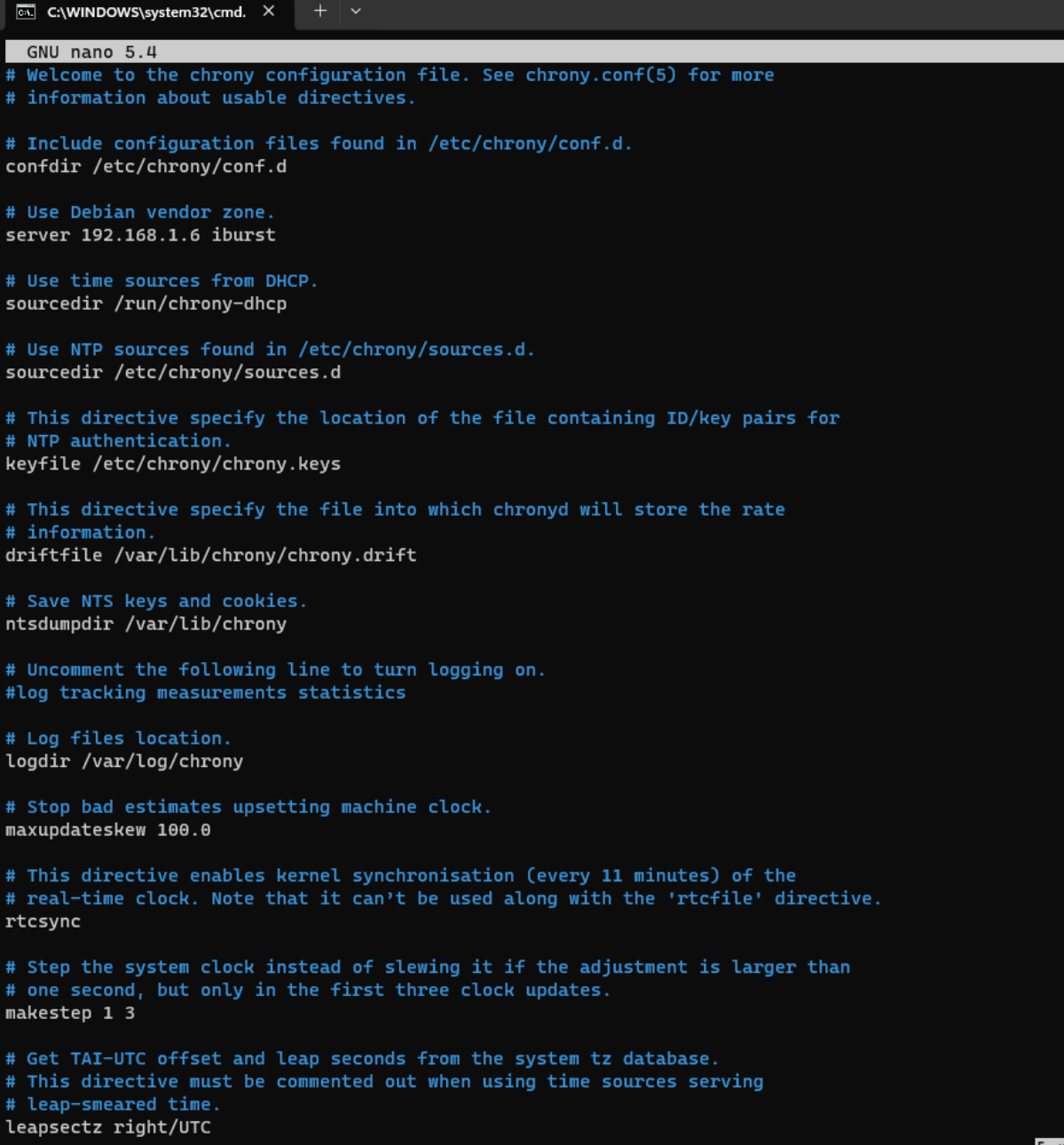
```
[23:31:23] DEBIAN: aluno@vm06 [/etc/chrony]$ timedatectl
Local time: qui 2023-10-19 23:35:42 -03
Universal time: sex 2023-10-20 02:35:42 UTC
RTC time: sex 2023-10-20 02:35:42
Time zone: America/Sao_Paulo (-03, -0300)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
[23:35:42] DEBIAN: aluno@vm06 [/etc/chrony]$ |
```

## 5. Instalação do pacote "chrony" na VM5

Utilizamos o comando `apt-get install chrony` para instalar o pacote necessário na VM5.

## 6. Mudar o arquivo /etc/chrony/chrony.conf com o comando sudo nano

**Colocamos o servidor da VM5 para consumir o servidor de horas da Vm6**

A screenshot of a terminal window with a dark background. The title bar shows 'C:\WINDOWS\system32\cmd.' and a tab icon. The terminal displays the content of the /etc/chrony/chrony.conf file, which is being edited with GNU nano 5.4. The file contains various configuration directives for the Chrony time service, including include paths, server definitions, source directories, keyfile location, driftfile, logging settings, and clock adjustment parameters.

```
GNU nano 5.4
# Welcome to the chrony configuration file. See chrony.conf(5) for more
# information about usable directives.

# Include configuration files found in /etc/chrony/conf.d.
confdir /etc/chrony/conf.d

# Use Debian vendor zone.
server 192.168.1.6 iburst

# Use time sources from DHCP.
sourcedir /run/chrony-dhcp

# Use NTP sources found in /etc/chrony/sources.d.
sourcedir /etc/chrony/sources.d

# This directive specify the location of the file containing ID/key pairs for
# NTP authentication.
keyfile /etc/chrony/chrony.keys

# This directive specify the file into which chronyd will store the rate
# information.
driftfile /var/lib/chrony/chrony.drift

# Save NTS keys and cookies.
ntsdumpdir /var/lib/chrony

# Uncomment the following line to turn logging on.
#log tracking measurements statistics

# Log files location.
logdir /var/log/chrony

# Stop bad estimates upsetting machine clock.
maxupdateskew 100.0

# This directive enables kernel synchronisation (every 11 minutes) of the
# real-time clock. Note that it can't be used along with the 'rtcfile' directive.
rtcsync

# Step the system clock instead of slewing it if the adjustment is larger than
# one second, but only in the first three clock updates.
makestep 1 3

# Get TAI-UTC offset and leap seconds from the system tz database.
# This directive must be commented out when using time sources serving
# leap-smeared time.
leapsectz right/UTC
```

## 7. Verificação da data e hora da VM5

Utilizamos o comando "timedatectl" para verificar se a data e hora da VM5 está corretas.

```
[23:38:29] DEBIAN: aluno@vm05 [/etc/chrony]$ timedatectl
Local time: qui 2023-10-19 23:38:34 -03
Universal time: sex 2023-10-20 02:38:34 UTC
RTC time: sex 2023-10-20 02:38:34
Time zone: America/Sao_Paulo (-03, -0300)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

## Problemas/Dificuldades encontradas e Soluções Adotadas:

Durante a configuração da VM6, que funciona como um serviço cliente do servidor de horas do ntp.br e como servidor de horas para a VM5, tivemos problemas para a configuração dos servidores com que a VM6 iria sincronizar, para contornar esta dificuldade substituímos a sequência de servidores, pelo conjunto de servidores pool.ntp.br, durante a sincronização da VM5 com a VM6, notamos que a sincronização não acontecia, para solucionar este problema, na VM6 no arquivo chrony.conf, foi necessário configurar a permissão para a VM5 acessar a VM6.