# UNIVERSIDADE FEDERAL DE LAVRAS
# Disciplina: Redes de Computadores - GCC125
# Trabalho de Instalação - Etapa 1
# Grupo N: Bernardo Nunes Leris 14A - Arthur Castro 14A - João Luiz Rodrigues de Araújo 14A

## Primeiros passos:

Para ter acesso às máquinas virtuais foi necessário conectar-nos ao Laboratório pelo OpenVPN. Logo depois, utilizando o servidor ssh, Secure Socket Shell, o qual é um dos protocolos específicos de segurança de troca de arquivos entre cliente e servidor de internet, usando criptografia, que já está presente nas versões mais atuais do Windows, sistema operacional utilizado.

No prompt do comando inicializado como administrador deve-se executar os seguintes comandos:

Para acessar a primeira máquina virtual:

```
ssh aluno@192.168.1.27
```

Para acessar a segunda máquina virtual:

```
ssh aluno@192.168.1.28
```

- Esse comando retornará a exigência de uma senha que possibilitará a **entrada na máquina virtual**. A senha inserida em ambas foi “aluno”.

Por conseguinte, foi realizada a **troca da senha** de acesso aos dois servidores, **192.168.1.27** e **192.168.1.28**, utilizando o usuário “aluno” para **“wagner123”**. Os comandos utilizados para realizar essa mudança foi:

Para indicar qual usuário quer trocar de senha:

```
passwd aluno
```

Para inserção da senha nova:

```
wagner123
```

# Serviço de Sincronização de Hora:

A VM **192.168.1.27** se conectará a um servidor NTP.br e a VM **192.168.1.28** servirá como cliente de hora. Os seguintes comandos foram utilizados:

Para verificar a data e hora na máquina virtual:

```
date
```

Adiante, o comando a seguir **instalará o pacote chrony no sistema**. O chrony é uma implementação do protocolo NTP (Network Time Protocol), que é usado para sincronizar a hora dos computadores com um servidor de hora:

```
sudo apt install chrony
```

Devemos **interromper o serviço Chrony**. Isso é necessário para que você possa editar o arquivo de configuração do chrony sem problemas:

```
sudo systemctl stop chrony
```

Depois, o comando a ser inserido **define a hora do sistema para 00:00:00**. Isso é útil para testar o serviço chrony, pois você pode verificar se o chrony é capaz de sincronizar a hora do sistema com um servidor de hora mesmo quando a hora do sistema está incorreta:

```
sudo timedatectl set-time 00:00:00
```

Podemos realizar a verificação se a **data e o horário** foram atualizados::

```
date
```

Ainda, devemos fazer a **abertura do arquivo de configuração** do chrony no editor de texto nano. Você pode editar o arquivo de configuração para especificar o servidor de hora que o chrony deve usar para sincronizar a hora do sistema:

```
sudo nano /etc/chrony/chrony.conf
```

Depois foi **configurado** conforme recomendação do **NTP.br**:

```
# servidores publicos do NTP.br com NTS disponível
server a.st1.ntp.br iburst nts
server b.st1.ntp.br iburst nts
server c.st1.ntp.br iburst nts
server d.st1.ntp.br iburst nts
```

```
server gps.ntp.br iburst nts

# 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

# diretiva que indica que o offset UTC e leapseconds devem ser lidos
# da base tz (de time zone) do sistema
leapsectz right/UTC
```

Para **verificar** se o chrony está **sincronizado** com o servidor de **hora**, executamos o seguinte comando:

```
sudo systemctl start chrony
```

O próximo comando irá **listar as fontes de tempo** que o chrony está usando para sincronizar a hora do sistema, para que possamos **fazer a verificação**:

```
chronyc sources
```

Ainda, devemos **exibir as informações** sobre o **estado de rastreamento** do chrony:

```
chronyc tracking
```

Após **verificar a data** da máquina virtual com o comando "date" e confirmar que ela estava sincronizada corretamente, tentamos modificar a hora novamente usando o comando `sudo timedatectl set-time 00:00:00`. No entanto, nos deparamos com o **seguinte erro**:

```
Failed to set time: Automatic time synchronization is enabled
```

## Sincronização da hora do VM 192.168.1.28 com o VM 192.168.1.27:

Para começar, acessamos o arquivo de configuração na máquina virtual com o endereço **IP 192.168.1.27** usando o comando `sudo nano /etc/chrony/chrony.conf`. Em seguida, inserimos o seguinte fragmento de código no arquivo, permitindo que a máquina virtual com o endereço IP **192.168.1.28** acesse a primeira máquina como cliente NTP.

```
[...]
# permite o acesso aos seguintes clientes NTP
allow 192.168.1.28
```

Depois disso, **encerramos nossa conexão** com a máquina virtual de endereço IP 192.168.1.27 e, em seguida, **conectamos** à máquina virtual **192.168.1.28**. Uma vez conectados à VM 192.168.1.28, seguimos os mesmos procedimentos para instalar o Chrony:

```
sudo apt install chrony
sudo systemctl stop chrony
```

Fizemos a **modificação da data**, definindo um valor incorreto para fins de teste, e posteriormente verificamos o resultado:

```
sudo timedatectl set-time 00:00:00
date
```

**Acessamos e modificamos** o arquivo de configuração do Chrony usando o comando `sudo nano /etc/chrony/chrony.conf` para que a máquina virtual 192.168.1.28 seja capaz de reconhecer a máquina virtual 192.168.1.27 como um servidor a ser sincronizado:

```
# servidores publicos do NTP.br com NTS disponível
server 192.168.1.27 iburst
[...]
```

Por último, **habilitamos** o serviço do Chrony e verificamos com os seguintes comandos se o relógio da máquina virtual 192.168.1.28 está **sincronizado** com a máquina 192.168.1.27:

```
sudo systemctl start chrony
chronyc sources
chronyc tracking
date
```

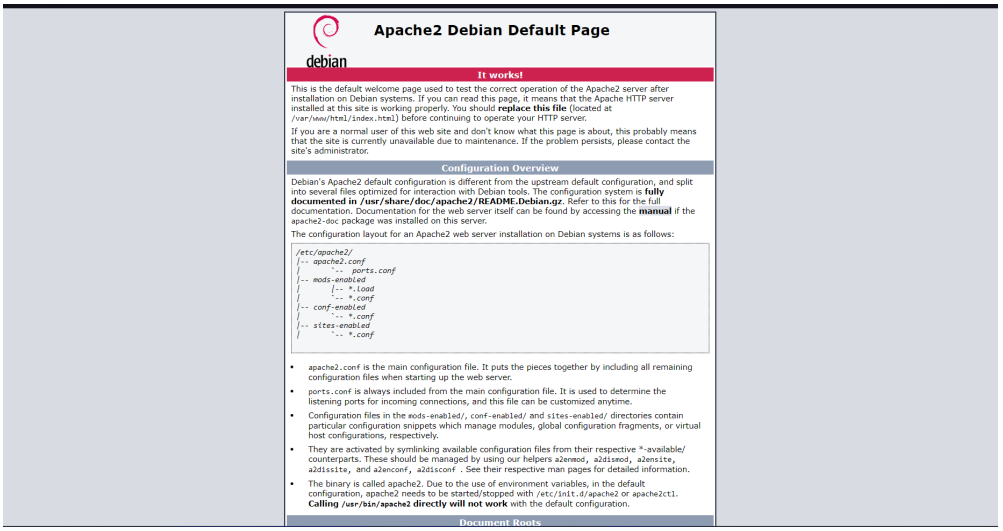
## Instalação do servidor Web:

Primeiramente, foi realizada a **instalação do apache 2** com o comando:

```
sudo apt install apache2
```

A instalação foi **verificada** por meio do acesso de uma máquina conectada a VM por meio de um browser utilizando o url: **http://192.168.1.28**, retornando a seguinte página:

,



## Inserção da página HTML no servidor Web:

Realizamos o **acesso ao diretório do servidor apache**, o qual há os arquivos HTML com o comando abaixo:

```
cd /var/www/html/
```

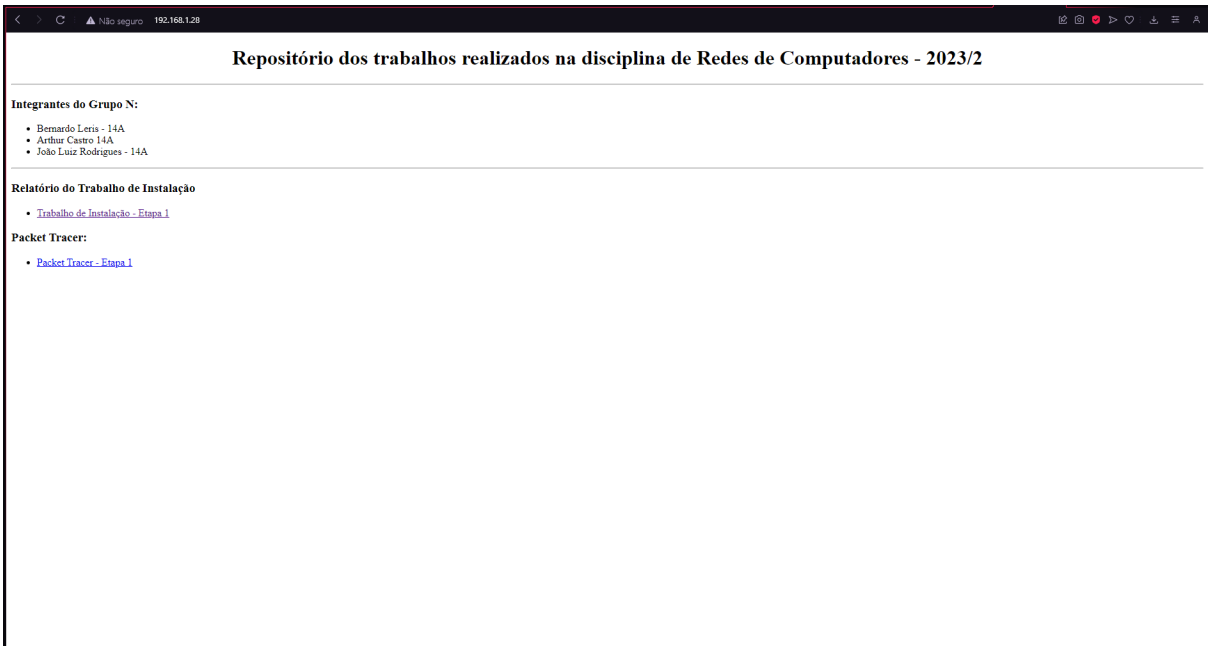
Com o comando a seguir **apagamos os arquivos presentes na pasta**, a fim de preparar o ambiente para os arquivos a serem inseridos:

```
sudo rm *
```

Os arquivos a serem inseridos são **baixados do repositório** https://github.com/bernardoleris/redes_de_computadores.git, por intermédio dos comandos a seguir:

```
sudo
wget
https://raw.githubusercontent.com/bernardoleris/redes_de_computadores/main/index.html
```

Ao acessar novamente o endereço http://192.168.1.28 temos o seguinte retorno:



## Acesso com criptografia HTTP no servidor Web:

Foi instalado o pacote **openssl** na máquina de servidor web **192.168.1.28**, utilizando o código abaixo:

```
sudo apt install openssl
```

Logo, habilitamos o **ssl e o rewrite do Apache**:

```
sudo a2enmod ssl
sudo a2enmod rewrite
```

Utilizamos o editor de arquivo nano para realizar a **configuração do Apache**:

```
sudo nano /etc/apache2/apache2.conf
```

No editor, foi adicionado o trecho de código abaixo:

```
<Directory /var/www/html>
     AllowOverride All
</Directory>
```

Ainda, foi criada a pasta para **armazenar o certificado ssl** que será criado:

```
sudo mkdir /etc/apache2/certificate
cd /etc/apache2/certificate
```

Logo, ao entrar na pasta, criamos uma **chave privada e o certificado ssl**:

```
sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache-certificate.crt -keyout apache-private-key.key
```

Depois disso, foi **inserida** as informações a seguir:

```
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]:Departamento de Ciencia da Computacao
Organizational Unit Name (eg, section) []:Grupo N
Common Name (e.g. server FQDN or YOUR name) []:192.168.1.28
Email Address []:.
```

Utilizaremos o **nano para editar** o arquivo 000-default.conf:

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

O arquivo possui apenas o conteúdo abaixo:

```
<VirtualHost *:80>
 [...]
</VirtualHost>
```

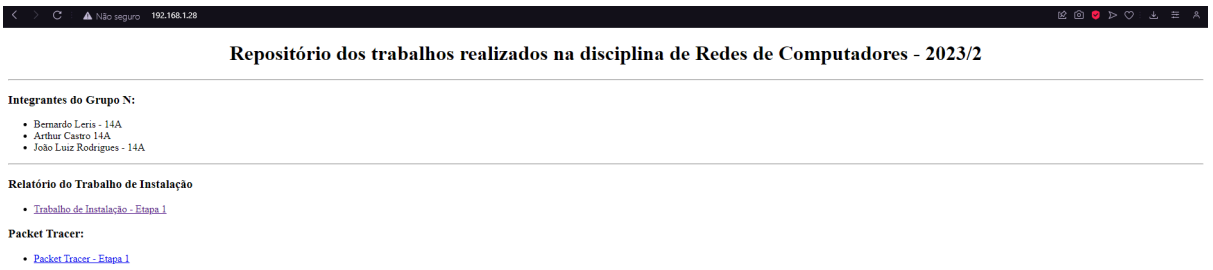
**Adicionamos** nele as configurações a seguir:

```
<VirtualHost *:443>
 [...]
 SSLEngine on
                                                SSLCertificateFile
/etc/apache2/certificate/apache-certificate.crt
 SSLCertificateKeyFile /etc/apache2/certificate/apache.key
</VirtualHost>
```

Após isso, **reiniciamos o Apache**:

```
sudo systemctl restart apache2
```

Para finalizar, realizamos o **teste** com o **acesso sem criptografia**, pela url http://192.168.1.28, o retorno foi:



Primeiramente, ao acessar o url acima, foi gerado um aviso de segurança pelo browser, pelo motivo de que o certificado foi criado pelo próprio servidor, não garantindo uma segurança real aos usuários.

Dessa maneira, temos o servidor Web instalado de forma efetiva, possuindo a criptografia HTTP. Os trabalhos submetidos à avaliação estão disponíveis para serem acessados na página web.