

TCP/IP: PRINCIPAIS CARACTERÍSTICAS, FUNCIONAMENTO E VULNERABILIDADE.

MARCELO NAKAGAWA E MARCOS GASPAR



4

LISTA DE FIGURAS

Figura 4.1 – Cabeçalho IPv4.....	10
Figura 4.2 – Resolução ARP.....	13
Figura 4.3 – Tabela ARP.....	14
Figura 4.4 – Formato geral da mensagem ARP.....	15
Figura 4.5 – Encapsulamento da mensagem ICMP.....	17
Figura 4.6 – Formato geral da mensagem ICMP.....	17
Figura 4.7 – Cabeçalho UDP.....	19
Figura 4.8 – Cabeçalho TCP.....	20
Figura 4.9 – Diagrama de Estados (simplificado) do TCP.....	22
Figura 4.10 – Cenário de referência.....	24
Figura 4.11 – Sockets e estados das conexões.....	27
Figura 4.12 – Download da versão corrente do VirtualBox.....	29
Figura 4.13 – Verificação da integridade do arquivo.....	29
Figura 4.14 – Hash do arquivo de instalação do VirtualBox.....	30
Figura 4.15 – Janela Extensões do VirtualBox.....	31
Figura 4.16 – Download da imagem (ISO) do Debian.....	32
Figura 4.17 – Ferramenta WinMD5.....	33
Figura 4.18 – Conformação dos hashes.....	33
Figura 4.19 – Nome e S.O. da nova VM.....	34
Figura 4.20 – Quantidade de RAM para a VM.....	35
Figura 4.21 – Dashboard do VirtualBox com a VM Debian1.....	36
Figura 4.22 – Painel de configurações da VM.....	36
Figura 4.23 – Seleção do disco de boot.....	37
Figura 4.24 – Seleção do tipo de instalação do Debian.....	38
Figura 4.25 – Seleção do idioma.....	38
Figura 4.26 – Seleção da localidade.....	39
Figura 4.27 – Seleção do teclado.....	39
Figura 4.28 – Carga de componentes adicionais.....	40
Figura 4.29 – Hostname da VM.....	40
Figura 4.30 – Nome de domínio da VM.....	41
Figura 4.31 – Senha do root.....	41
Figura 4.32 – Nome do usuário desprivilegiado	42
Figura 4.33 – Relógio do sistema.....	42
Figura 4.34 – Opção pelo particionamento assistido.....	43
Figura 4.35 – Esquema de particionamento.....	43
Figura 4.36 – Disco a ser particionado.....	44
Figura 4.37 – Efetivação das mudanças do disco	44
Figura 4.38 – Instalação básica do sistema	45
Figura 4.39 – Mídias complementares	45
Figura 4.40 – Download e instalação de pacotes.....	46
Figura 4.41 – Softwares a instalar.....	46
Figura 4.42 – Instalação do GRUB em disco rígido	47
Figura 4.43 – Instalação do GRUB em /dev/das	47
Figura 4.44 – Reinicialização após final da instalação	48
Figura 4.45 – Instalação dos adicionais de convidados	49
Figura 4.46 – Finalização da instalação dos adicionais de convidados	50
Figura 4.47 – Nome da interface	51

Figura 4.48 – Ajustes no arquivo sysctl.conf	52
Figura 4.49 – Verificação do IPv4	52



LISTA DE QUADROS

Quadro 4.1 – Modelo OSI vs. TCP/IP	8
Quadro 4.2 – IPv4 vs IPv6.....	10
Quadro 4.3 – Classes e tipos das mensagens ICMP	18
Quadro 4.4 – Well-known ports	26
Quadro 4.5 – Opções para criação do disco rígido da VM	35

EMANIP

LISTA DE COMANDOS

Lista de comandos 4.1 – Instalação do adicional de convidados	49
Lista de comandos 4.2 – Instalação do NET-TOOLS e Wireshark.....	50
Lista de comandos 4.3 – Desativação do IPv6.....	51

EXEMPLO

SUMÁRIO

4	TCP/IP: PRINCIPAIS CARACTERÍSTICAS, FUNCIONAMENTO E VULNERABILIDADE	7
4.1	O início.....	7
4.2	Arquitetura do TCP/IP	8
4.2.1	Visão geral do TCP/IP	8
4.2.2	TCP/IP: camada de acesso à rede	9
4.2.3	TCP/IP: camada de Internet	9
4.3	O protocolo ARP	12
4.4	O protocolo ICMP	16
4.5	O protocolo IGMP	18
4.6	TCP/IP: camada de transporte	18
4.7	TCP/IP: camada de aplicação	24
4.8	Hands on – iniciando a criação do ambiente	27
4.9	Download e instalação do VirtualBox.....	28
4.10	Download e verificação da imagem do Debian.....	31
4.11	Provisionamento da VM do Debian.....	33
4.12	Instalação do Debian na VM.....	37
4.13	Ajustes iniciais no ambiente.....	48
4.13.1	Instalação do adicional de convidados	48
4.13.2	Instalação dos pacote net-tools e wireshark	50
4.13.3	Desabilitando o suporte ao IPv6	51
	REFERÊNCIAS	53
	GLOSSÁRIO.....	54

4 TCP/IP: PRINCIPAIS CARACTERÍSTICAS, FUNCIONAMENTO E VULNERABILIDADE

4.1 O início

Em 1969, a Advanced Research Projects Agency (Arpa) financiou um projeto de pesquisa e desenvolvimento para criar uma rede experimental comutada por pacotes, chamada Arpanet, que foi construída para estudar técnicas que possibilitassem a comunicação de dados de forma confiável, robusta e independente de fabricante. O sucesso da Arpanet foi tão expressivo que muitas organizações ligadas a ela começaram a utilizar-se de seus recursos em suas comunicações (de dados) diárias. Em 1975, ela deixou de ser uma rede experimental, passando a ser uma rede operacional administrada pela Defense Communications Agency (DCA).

O TCP/IP foi adotado como padrão militar em 1983, e todos os *hosts* conectados à rede foram obrigados a aderir a este novo conjunto de protocolos de comunicação. Para facilitar a conversão, a Darpa contratou a Bolt, Beranek e Newman (BBN Technologies) para implementar o TCP/IP no Berkeley Unix (BSD), sendo estabelecido, desta forma, o vínculo entre o Unix e o TCP/IP – aproximadamente na época em que o TCP/IP foi adotado como padrão, o termo Internet passou a ser utilizado (HUNT, 2010).

Leia mais em: <http://www.internet-guide.co.uk/arpa-darpa.html>

E em: <http://www.internet-guide.co.uk/BoltBeranekandNewmantechologies.html>

De acordo com o Internet-Guide, a suíte de protocolos TCP/IP é baseada em um modelo (de rede) com quatro camadas, não devendo ser confundido com o de sete camadas do modelo OSI – embora haja similaridades entre estes dois modelos, eles são diferentes.

Leia mais em: <http://www.internet-guide.co.uk/tcp-ip.html>

4.2 Arquitetura do TCP/IP

4.2.1 Visão geral do TCP/IP

A suíte de protocolos TCP/IP foi desenvolvida tomando como referência o modelo do Departamento de Defesa (DoD) e antecedeu a publicação do modelo OSI. Entretanto, é importante familiarizar-se também com este por ser frequentemente utilizado para comparar o TCP/IP às outras suítes de protocolo.

Diferentemente do modelo OSI – composto por sete camadas –, o modelo do DoD e, portanto, o TCP/IP têm apenas quatro camadas:

- **Acesso à rede:** também algumas vezes referenciada simplesmente como camada de REDE, abrange as camadas FÍSICA e de ENLACE do modelo OSI.
- **Internet:** equivale à camada de REDES do modelo OSI.
- **Transporte:** equivale à camada de TRANSPORTE do modelo OSI.
- **Aplicação:** abrange as camadas de SESSÃO, APRESENTAÇÃO e APLICAÇÃO do modelo OSI.

O quadro “Modelo OSI vs. TCP/IP” estabelece a comparação entre o modelo OSI e a pilha TCP/IP.

Modelo OSI	TCP/IP
APLICAÇÃO	APLICAÇÃO
APRESENTAÇÃO	
SESSÃO	
TRANSPORTE	TRANSPORTE
REDE	INTERNET
ENLACE	ACESSO À REDE
FÍSICA	

Quadro 4.1 – Modelo OSI vs. TCP/IP

Fonte: Elaborado pelo autor (2020)

4.2.2 TCP/IP: camada de acesso à rede.

É a camada baixa da hierarquia da suíte TCP/IP e abrange as camadas física e de enlace do modelo OSI. Os protocolos nesta camada fornecem os meios para que o sistema entregue dados a outros dispositivos diretamente conectados à rede.

Esta camada define como usar a rede para transmissão de datagramas IP. Diferentemente dos protocolos de nível superior, os protocolos de camada de acesso à rede devem conhecer os detalhes básicos da rede, tais como: sua estrutura de pacotes e endereçamento, entre outros, a fim de ajustar adequadamente os dados a serem transmitidos às características de rede (HUNT, 2010).

Algumas das funções desempenhadas por esta camada incluem o encapsulamento do datagrama IP nos *frames* transmitidos pela rede e o mapeamento dos endereços lógicos (IP) para os endereços físicos (MAC) utilizados pela rede (HUNT, 2010).

4.2.3 TCP/IP: camada de Internet.

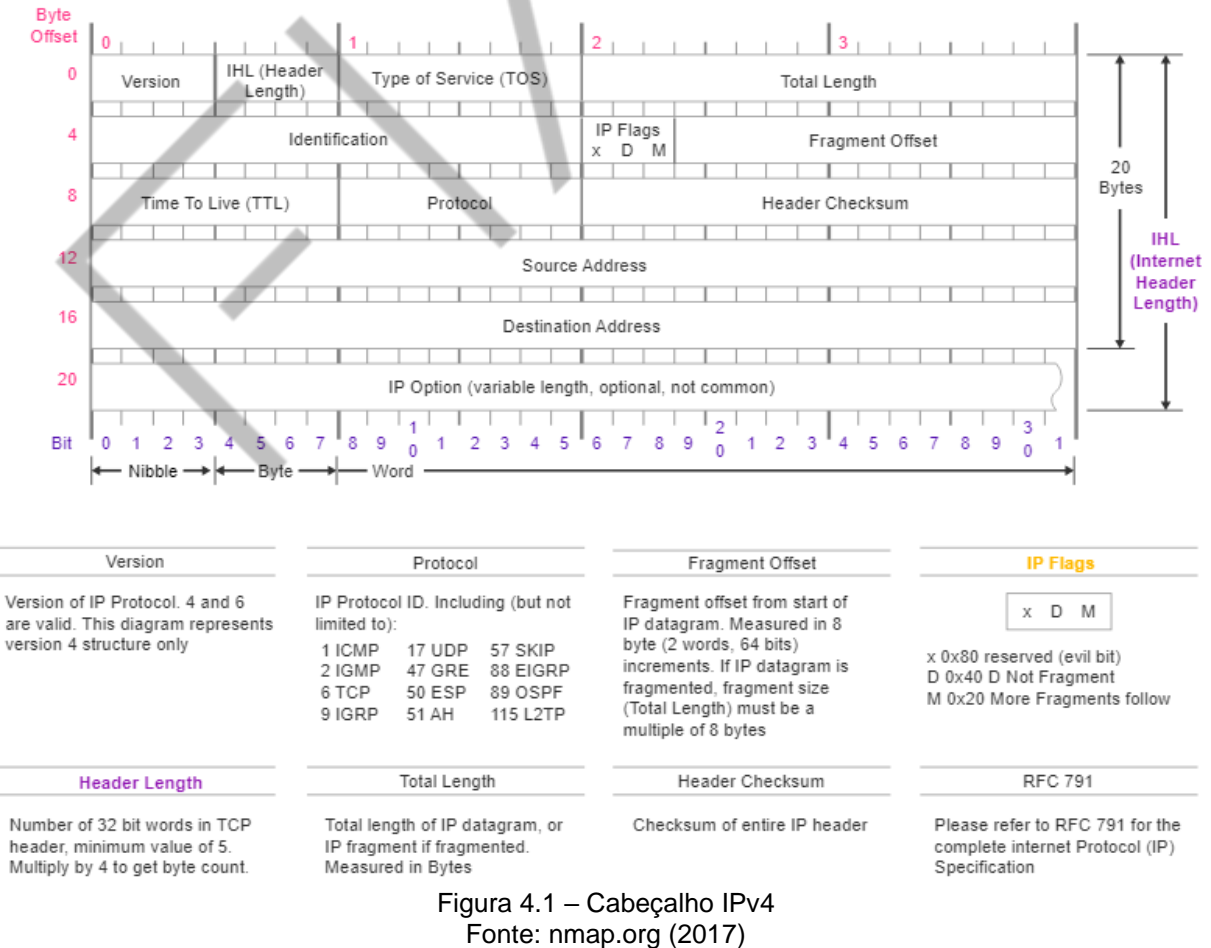
Camada equivalente à camada de REDES do modelo OSI e, como neste, a responsável pelo roteamento de pacotes entre diferentes redes, tendo como principais protocolos o IP, ARP, ICMP e IGMP.

O **protocolo IP** é o responsável pelo endereçamento lógico das redes e *hosts* e pode se apresentar em duas versões: v4 e v6, as quais são significativamente diferentes e também incompatíveis entre si. O Quadro IPv4 vs IPv6 ilustra algumas das principais diferenças entre as duas versões.

Descrição	IPv4	IPv6
Tamanho do endereço	32 bits	128 bits
Representação do endereço	Quatro grupos de 8 bits cada separados por "." e escritos com dígitos decimais.	Oito grupos de 16 bits separados por ":" e escritos com dígitos hexadecimais (0-F).
Tipos de endereços	Unicast, <i>multicast</i> , broadcast.	Unicast, <i>multicast</i> , anycast.
Cabeçalho do pacote	20 bytes	40 bytes
Configuração	Manual ou DHCP	Autoconfiguração disponível
Suporte ao IPSec	Opcional	Nativo

Quadro 4.2 – IPv4 vs IPv6
Fonte: Elaborado pelo autor (2020)

O conhecimento do cabeçalho do protocolo IP é fundamental, entre outras coisas, para a análise do tráfego das redes. Ainda permanecendo o IPv4 como a versão mais utilizada nas redes locais, faz-se necessário discutir, ao menos, os campos de seu cabeçalho de maior relevância para a CyberSecurity.



Onde:

- **IP version:** campo de 4 bits contendo a versão do protocolo IP.
- **IHL Length:** campo de 4 bits contendo o tamanho do cabeçalho IP (em quantidade de palavras de 32 bits – 4 bytes), essencialmente dedicado a especificar onde, exatamente, termina o cabeçalho e onde se iniciam os dados do datagrama IPv4. Um cabeçalho IPv4 mínimo possui 20 bytes de comprimento, portanto, o valor (decimal) mínimo deste campo será 5.
- **Total Length:** campo de 16 bits contendo o tamanho total do datagrama (em bytes). O tamanho mínimo de um datagrama é de 20 bytes e o máximo é 64 KB. Por vezes, as sub-redes impõem restrições ao tamanho, caso em que os datagramas devem ser "fragmentados";
- **Identification:** campo de 16 bits utilizado principalmente para assinalar fragmentos identificativos do datagrama IP original.
- **IP Flags:** campo de 3 bits usado para controlar ou identificar fragmentos, sendo (do bit mais significativo ao menos significativo):
 - bit 0: Reservado; deve ser zero.
 - bit 1: Don't Fragment (DF).
 - bit 2: More Fragments (MF).
 - se a flag DF estiver habilitada e a fragmentação for necessária para o encaminhamento, o pacote será descartado, podendo, assim, ser habilitada para o envio de pacotes para *hosts* incapazes de lidar com a fragmentação. Também poderá ser usada para o Path MTU Discovery – seja automaticamente pelo software IP do *host*, seja manualmente por intermédio de ferramentas de diagnóstico, como *ping* ou *traceroute*. Para pacotes não fragmentados, a *flag* MF permanece desabilitada. Para pacotes fragmentados, todos os fragmentos, exceto o último, terão a flag MF habilitada. O último frag-

mento tem um campo Fragment Offset diferente de zero, distinguindo-o de um pacote não fragmentado.

- **Fragment Offset:** campo de 13 bits, tendo como unidade de medida blocos com 8 bytes. Este campo especifica o deslocamento de um fragmento específico em relação ao início do datagrama IP original, não fragmentado. O primeiro fragmento tem deslocamento zero, permitindo deslocamento máximo de $(2^{13} - 1) \times 8 = 65,528$ bytes, o que excederia o tamanho máximo do pacote IP de 65,535 bytes com o comprimento do cabeçalho incluído ($65,528 + 20 = 65,548$ bytes).
- **Time do Live:** campo de 8 bits destinado a limitar o tempo de vida dos pacotes. Ao chegar a um roteador, o valor do campo será decrementado de um: se seu valor ainda for superior a zero, o pacote será encaminhado ao próximo roteador; se seu valor for igual a zero, o pacote será descartado pelo roteador e uma mensagem de *time exceeded* será enviada ao remetente do pacote.
- **Protocol:** campo de 8 bits que especifica o protocolo a ser utilizado pelo *payload* de um datagrama IP, sendo os mais comuns o ICMP (1) e o TCP (6).
- **Checksum:** campo que verifica a consistência (apenas) do cabeçalho.
- **Source address / Destination address:** campos com tamanho de 32 bits cada (IPv4); respectivamente, referentes aos endereços de origem e destino do pacote.

4.3 O protocolo ARP

O protocolo IP fornece o endereçamento necessário à interconexão lógica das redes (camada 3 do modelo OSI). Entretanto, a interconexão física das redes locais que utilizam a suíte TCP/IP dá-se segundo o padrão ETHERNET, contemplando as especificações para operação na camada 1 (FÍSICA do modelo OSI) e subcamada MAC da camada 2 (ENLACE de DADOS do modelo OSI). Ocorre, entretanto, que o

endereço IPv4 (lógico) é composto por 32 bits, enquanto o endereço ETHERNET (físico) por 48 bits.

O protocolo ARP (Address Resolution Protocol) foi concebido para resolver este problema, criando dinamicamente uma tabela de correspondência entre os endereços (lógicos e físicos) que se autorrenova periodicamente, normalmente, a cada 10 ou 15 minutos.

Deve ser observado também que embora o protocolo ARP tenha sido originalmente desenvolvido para uso com o padrão ETHERNET, este não é o único a utilizar o ARP. A Figura “Resolução ARP” ilustra de forma simplificada o processo de resolução de endereços do protocolo ARP.

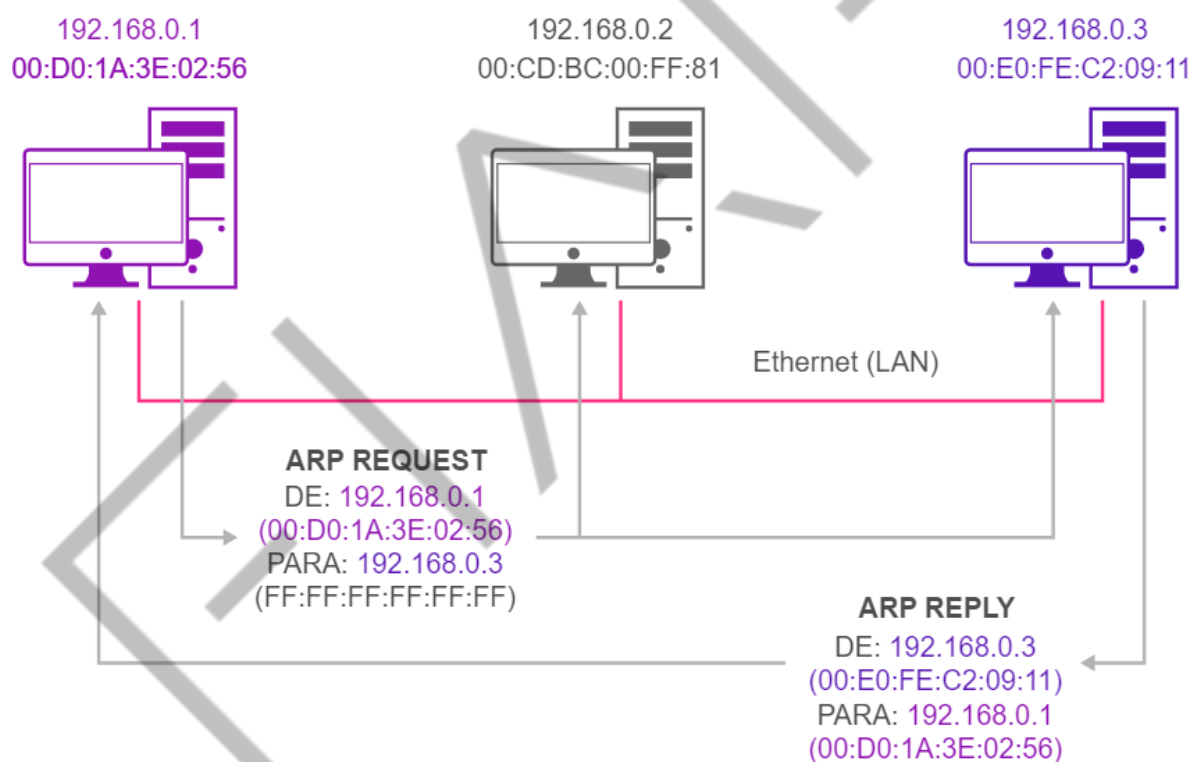
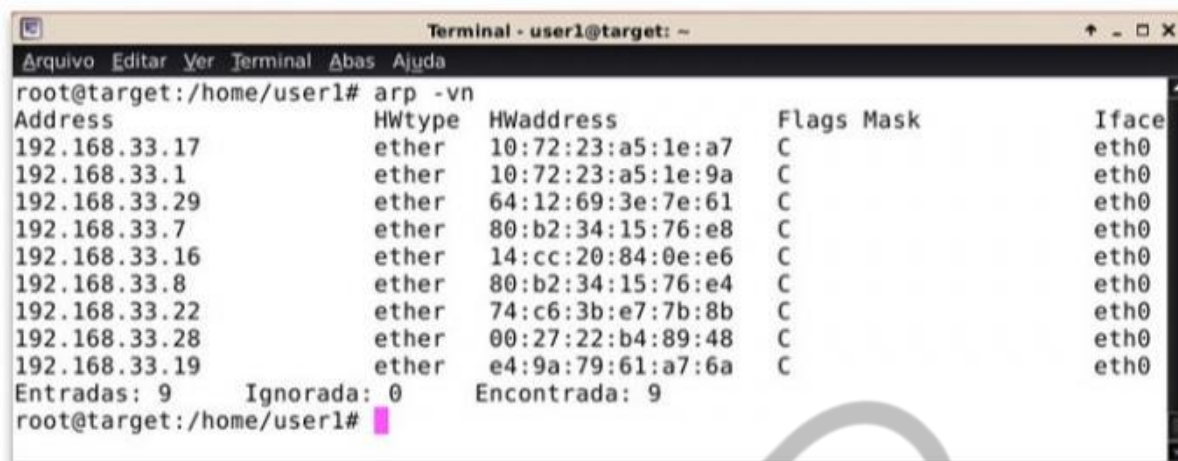


Figura 4.2 – Resolução ARP
Fonte: Elaborado pelo autor (2020)

Observa-se na figura que a resolução do endereço se inicia com um ARP REQUEST, anunciando em broadcast o endereço IP desejado e obtendo como resposta um ARP REPLY do *host* que detém o IP procurado. Entretanto, qualquer máquina da rede poderia usar a informação contida na mensagem ARP para alterar as entradas da sua tabela ARP. A figura a seguir ilustra a tabela ARP de um *host* Linux.



```

Terminal - user1@target: ~
Arquivo Editar Ver Terminal Abas Ajuda
root@target:/home/user1# arp -vn
Address      HWtype  HWaddress    Flags Mask    Iface
192.168.33.17 ether    10:72:23:a5:1e:a7 C          C        eth0
192.168.33.1 ether    10:72:23:a5:1e:9a C          C        eth0
192.168.33.29 ether    64:12:69:3e:7e:61 C          C        eth0
192.168.33.7 ether    80:b2:34:15:76:e8 C          C        eth0
192.168.33.16 ether    14:cc:20:84:0e:e6 C          C        eth0
192.168.33.8 ether    80:b2:34:15:76:e4 C          C        eth0
192.168.33.22 ether    74:c6:3b:e7:7b:8b C          C        eth0
192.168.33.28 ether    00:27:22:b4:89:48 C          C        eth0
192.168.33.19 ether    e4:9a:79:61:a7:6a C          C        eth0
Entradas: 9      Ignorada: 0      Encontrada: 9
root@target:/home/user1#

```

Figura 4.3 – Tabela ARP
Fonte: Elaborado pelo autor (2020)

Ainda na Figura “Tabela ARP”, é possível observar a correspondência entre os endereços IP (lógicos) e os respectivos MAC (físicos) dos *hosts* na rede conhecidos pelo *host* de referência.

O formato da mensagem ARP foi concebido para suportar endereços de camada 2 e de camada 3 de vários tamanhos. A Figura “Formato geral da mensagem ARP” mostra a implementação mais comum, com 32 bits para os endereços de camada 3 (IP) e 48 bits para os endereços de camada 2 (hardware), respectivamente correspondendo aos tamanhos dos endereços IP (v4) e MAC (Ethernet - IEEE 802).

Após a mensagem ARP haver sido montada, ela é enviada à camada de enlace de dados, para transmissão, tornando-se o *payload* do *frame* Ethernet efetivamente enviado à rede. O tamanho total da mensagem ARP é variável, uma vez que os tamanhos dos campos de endereço podem variar. Normalmente, porém, essas mensagens são bem pequenas, por exemplo, tendo apenas 28 bytes em uma rede que encapsula datagramas IPv4 nos endereços MAC (IEEE 802).

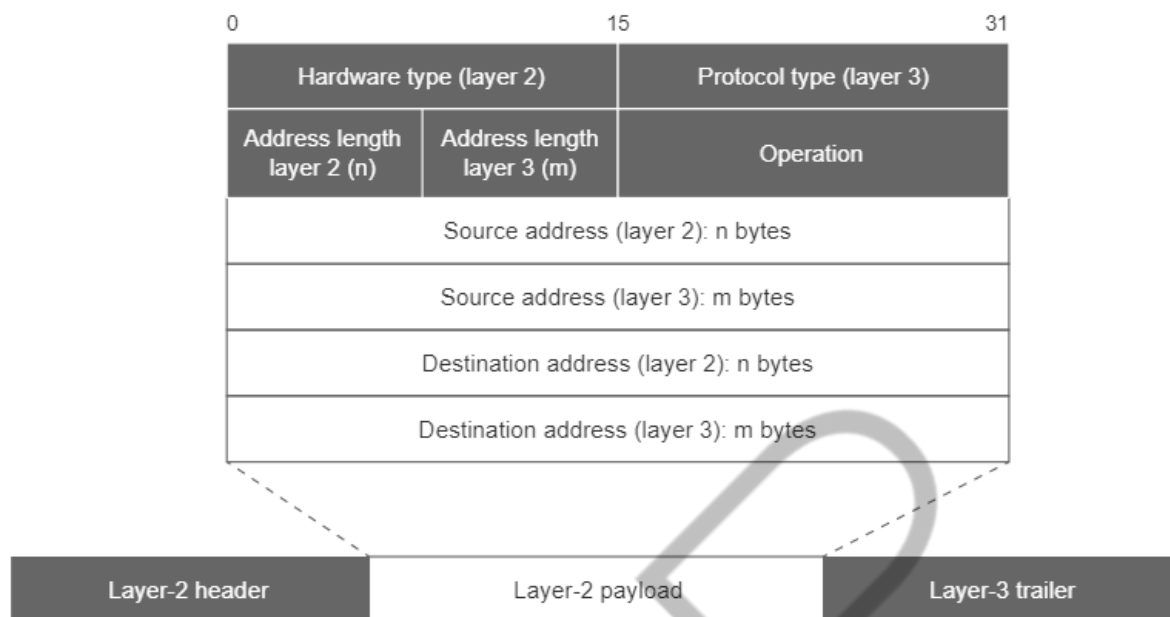


Figura 4.4 – Formato geral da mensagem ARP
Fonte: Cisco (2016)

Da Figura “Formato geral da mensagem ARP”, tem-se:

- **Hardware Type:** especifica o tipo de hardware utilizado pela rede local que transmite a mensagem ARP, identificando, assim, também o tipo de endereçamento utilizado. Alguns dos valores mais comumente encontrados neste campo são: Ethernet (10 Mbps) (valor campo=1), redes IEEE802 (valor do campo=6) e Frame Relay (valor do campo=15).
- **Protocol Type:** este campo é o complemento do campo Hardware Type, especificando o tipo de endereços de camada 3 usados pela mensagem. Para endereços IPv4, esse valor é 2048 (0800 hex), o que corresponde ao código EtherType para o IP.
- **Address length layer 2:** especifica o tamanho dos endereços de hardware nesta mensagem. Para Ethernet ou outras redes usando endereços MAC IEEE 802, o valor é 6.
- **Address length layer 3:** também é o complemento do campo precedente, especificando o tamanho dos endereços de camada 3 nesta mensagem. Para endereços IPv4, o valor é 4.

- **Operation:** este campo especifica a natureza da mensagem ARP sendo transmitida. Os valores 1 e 2 indicam mensagens regulares, respectivamente: ARQ REQUEST e ARP REPLY; porém, vários outros valores podem ser definidos a fim de suportar outros protocolos que também utilizem o formato do *frame* ARP, como, por exemplo, o protocolo RARP.
- **Source address layer 2:** endereço de camada 2 utilizado pelo dispositivo que envia a mensagem ARP.
- **Source address layer 3:** endereço de camada 3 utilizado pelo dispositivo que envia a mensagem ARP.
- **Destination address layer 2:** endereço de camada 2 utilizado pelo dispositivo de destino da mensagem ARP.
- **Destination address layer 3:** endereço de camada 3 utilizado pelo dispositivo de destino da mensagem ARP.

4.4 O protocolo ICMP

O ICMP (Internet Control Message Protocol) é um protocolo integrante do Protocolo IP, definido pela RFC 792. As mensagens ICMP são usadas para troca de diferentes tipos de informações entre dispositivos IP em uma inter-rede. Originalmente criadas para relatar um pequeno conjunto de condições de erro, as mensagens ICMP são empregadas agora para implementar uma ampla gama de relatórios de erros, *feedbacks* e recursos de teste.

Em seu nível mais alto, as mensagens ICMP são divididas em duas classes:

(a) mensagens de erro, utilizadas para notificação ao dispositivo de origem do pacote sobre a ocorrência de um erro.

(b) mensagens informativas (ou de consulta), utilizadas para troca de informações entre os dispositivos para que eles implementem funcionalidades relacionadas ao protocolo IP, ou ainda para execução de testes.

A Figura “Encapsulamento da mensagem ICMP” ilustra o encapsulamento da mensagem ICMP pelo datagrama IP.

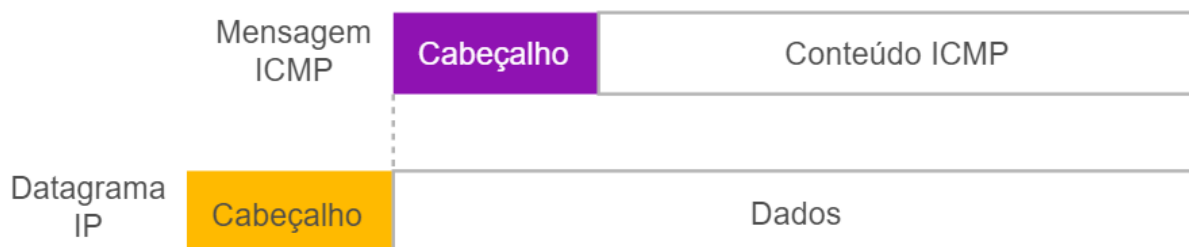


Figura 4.5 – Encapsulamento da mensagem ICMP

Fonte: Elaborado pelo autor (2020)

Complementando, a Figura “Formato geral da mensagem ICMP” ilustra o formato geral da mensagem ICMP e seus campos, a partir da qual se pode observar também onde as mensagens (de erro ou de informação) são transportadas.

O PING é uma das ferramentas mais populares a utilizar o protocolo ICMP e, portanto, não devem ser confundidos um com o outro.

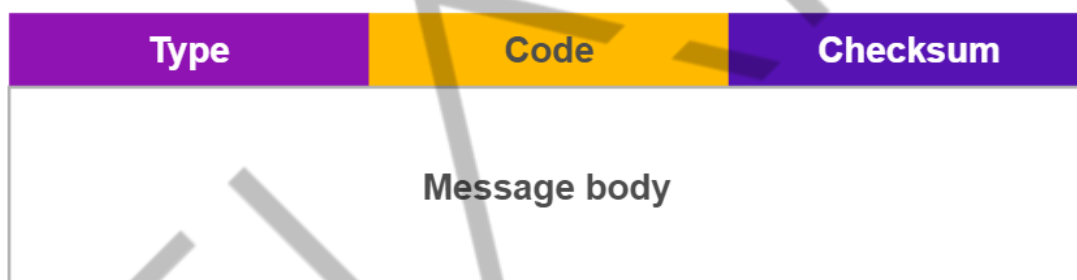


Figura 4.6 – Formato geral da mensagem ICMP

Fonte: Elaborado pelo autor (2020)

Da Figura “Formato geral da mensagem ICMP”, tem-se:

- **Type:** campo com 8 bits que identifica o tipo de mensagem ICMP.
- **Code:** campo com 8 bits que identifica o subtipo da mensagem ICMP especificada pelo valor do campo TYPE. Portanto, até 256 subtipos diferentes podem ser definidos por mensagem ICMP.
- **Checksum:** campo com 16 bits para verificação da consistência da mensagem ICMP como um todo.
- **Message body:** campo com tamanho variável destinado às mensagens de erro ou informação.

O quadro a seguir exibe alguns exemplos das classes e tipos mais comuns das mensagens ICMP.

Classe de Mensagem	Identificador do tipo	Nome da mensagem	Descrição resumida
ICMPv4 Mensagens de erro	3	<i>Destination Unreachable</i>	Indica que o datagrama não pode ser entregue ao destino. O código de erro (Code Value) fornece informações sobre a natureza deste
	5	<i>Redirect</i>	Permite que um roteador informe ao <i>host</i> uma rota melhor para envio de seus datagramas.
	11	<i>Time Exceeded</i>	Enviado quando um datagrama foi descartado antes de ser entregue — Time To Live excedido.
ICMPv4 Mensagens informativas	0	<i>Echo Reply</i>	Enviado em resposta a uma mensagem <i>Echo Request</i> — usada para testes de conectividade.
	8	<i>Echo (Request)</i>	Enviado por um dispositivo para testar a conectividade com outro dispositivo na inter-rede.
	13	<i>Timestamp (Request)</i>	Mensagem enviada por um dispositivo solicitando a outro que lhe envie um <i>timestamp</i> para cálculo do tempo de propagação e sincronização de relógios.
	14	<i>Timestamp Reply</i>	Mensagem enviada em resposta a uma mensagem de requisição de <i>timestamp</i> para cálculo do tempo de propagação e sincronização de relógios.

Quadro 4.3 – Classes e tipos das mensagens ICMP
Fonte: Elaborado pelo autor (2020)

4.5 O protocolo IGMP

Simplificadamente, o IGMP (Internet Group Management Protocol) é um protocolo que também faz parte do protocolo IP, usado para requisitar a participação em um grupo *multicast*. A participação neste tipo de grupo é dinâmica, o que significa que membros podem entrar ou sair do grupo a qualquer tempo. Um *host* pode ser membro de um ou mais grupos *multicast*, simultaneamente. Por motivos de segurança, geralmente o IGMP é desabilitado, pois pode favorecer alguns tipos de ataques.

4.6 TCP/IP: camada de transporte.

A camada de transporte é a responsável pela comunicação entre processos finais de uma mensagem inteira, embora cada pacote desta mensagem seja individualmente tratado. É a camada responsável também pela integridade e ordem de entrega dos pacotes, bem como pelo fluxo de dados. (FOROUZAN, 2008).

Como os computadores geralmente executam diversas tarefas concorrentemente, ou seja, há vários processos ativos, a entrega fim-a-fim não se limita apenas à entrega da mensagem de um computador a outro, mas também de um processo específico no computador de origem a um processo específico no computador de destino. Os protocolos da camada de transporte podem ser divididos em duas categorias: orientados à conexão (*connection oriented*) ou sem conexão (*connection-less*).

O **UDP** (*User Datagram Protocol*) é um protocolo simples, com baixo *overhead*, utilizado para transporte de dados **sem conexão**, portanto, de maneira não confiável. A Figura “Cabeçalho UDP” ilustra o cabeçalho do datagrama UDP.

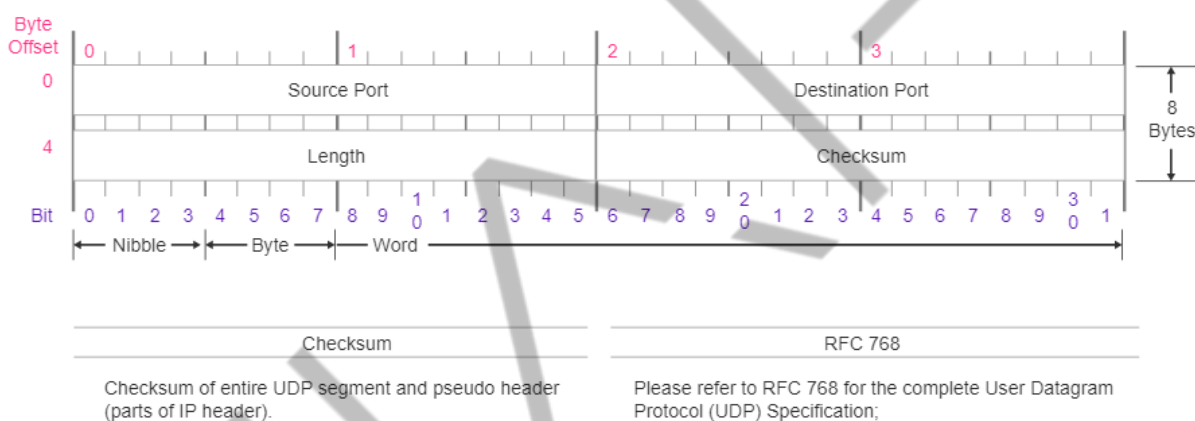


Figura 4.7 – Cabeçalho UDP
Fonte: nmap.org (2020)

Da Figura “Cabeçalho UDP”, tem-se:

- **Source port / Destination port:** campo de 16 bits que especifica o número da porta utilizado pelo processo, respectivamente, no *host* de origem e destino. Cada *host* possui 65536 (2^{16}) portas disponíveis, numeradas de 0 a 65535.
- **Length:** é um número de 16 bits que define o comprimento total do datagrama (cabeçalho + dados), desta forma, variando entre 0 e 65535 bytes. Entretanto, considerando-se que este é também o tamanho máximo do datagrama IP que o encapsulará, o datagrama UDP terá de ser menor do que o referido valor.
- **Checksum:** campo de 16 bits utilizado para detecção de erros na transmissão do datagrama UDP.

O **TCP** (*Transmission Control Protocol*) também é um protocolo de comunicação entre processos finais, porém, **orientado à conexão**. Assim sendo, ele estabelece uma conexão virtual entre dois processos que o utilizem, implementando ainda mecanismos para controle de fluxo e de erros (na camada de transporte) e proporcionando, desta maneira, a transmissão confiável dos dados. A Figura “Cabeçalho TCP” ilustra o cabeçalho do datagrama TCP.

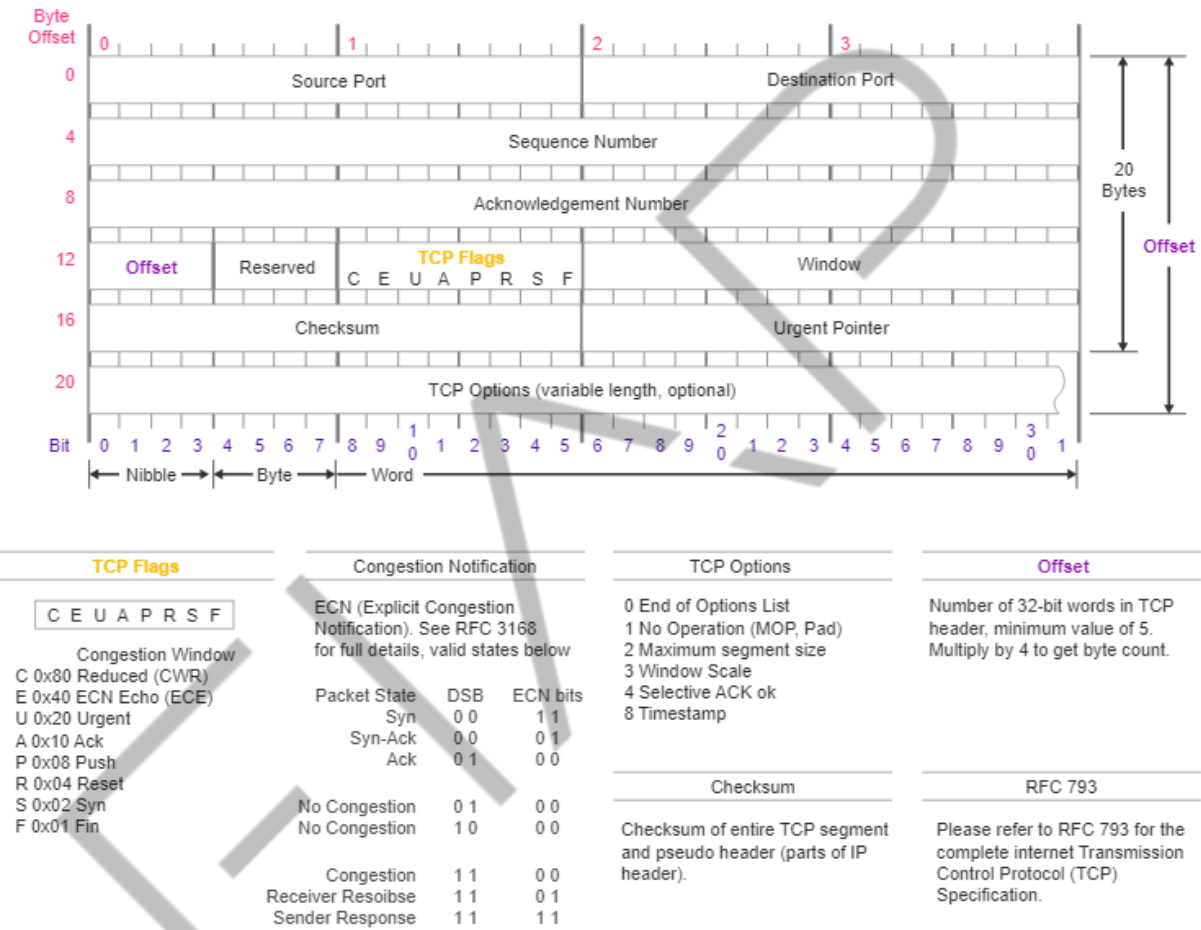


Figura 4.8 – Cabeçalho TCP
Fonte: nmap.org (2020)

Dos vários campos do cabeçalho TCP, apresentam especial relevância no contexto estudado:

- **Source port / Destination port:** campos de 16 bits, os quais especificam os números das portas utilizadas pelos processos, respectivamente, no *host* de origem e *host* de destino. Cada *host* possui 65536 (2^{16}) portas disponíveis, numeradas de 0 a 65535.

- **Sequence number:** número de 32 bits que identifica o primeiro byte de dados de cada segmento enviado. Transmissor e receptor têm fluxos de dados diferentes, portanto, números de sequência próprios, os quais não só permitem a correta remontagem do fluxo no destino, como também possibilitam verificar a eventual falta de algum segmento.
- **Acknowledgment Number:** campo de 32 bits que identifica os bytes que foram recebidos e tratados com sucesso pelo receptor. Se a flag ACK estiver habilitada, o valor deste campo é o próximo número de sequência que o transmissor espera. O primeiro ACK enviado de cada ponta reconhece o número de sequência inicial do outro (INITIAL SEQUENCE NUMBER – ISN).
- **Offset:** campo de 4 bits que especifica o tamanho do cabeçalho TCP em palavras de 32 bits. O tamanho mínimo do cabeçalho é de 5 palavras e o máximo é de 15 palavras, dando, assim, o tamanho mínimo de 20 bytes e máximo de 60 bytes e permitindo até 40 bytes de opções no cabeçalho. O nome deste campo decorre do fato de seu valor indicar o deslocamento entre o início do segmento TCP e os dados reais.
- **TCP Flags:** campo referente às flags de controle, das quais destacam-se no contexto abordado:
 - **syn:** (1 bit) sincronização dos números de sequência. Somente o primeiro pacote enviado de cada ponta deve ter esta flag habilitada.
 - **fin:** (1 bit) último pacote enviado pelo transmissor.
 - **psh:** (1 bit) solicita envio imediato dos dados ao receptor.
 - **rst:** (1 bit) solicita o *reset* da conexão.
 - **ack:** (1 bit) indica que o campo *Acknowledgement* é significativo. Todos os pacotes enviados pelo cliente após o pacote SYN inicial deverão habilitar esta flag.
 - **urg:** (1 bit) indica que o campo URGENT POINTER tem valor significativo.

- **Window:** campo de 16 bits com o tamanho da janela de recepção (geralmente em bytes).

Aliado ao conhecimento dos principais campos do cabeçalho do protocolo TCP, torna-se ainda altamente relevante o conhecimento em relação à sua operação, que pode ser dividida em três fases. O estabelecimento das conexões deverá ser devidamente executado por meio de um *handshake* de múltiplas fases antes de entrar na fase de transferência de dados. Finalizada a transmissão dos dados, a conexão deverá ser adequadamente finalizada, sendo fechados os circuitos virtuais estabelecidos e liberados os recursos alocados.

As conexões TCP são gerenciadas pelo sistema operacional por meio de uma interface de programação que representa o *endpoint* local para comunicações (*socket*), passando este *endpoint* por uma série de mudanças de estado enquanto a conexão é mantida. A Figura “Diagrama de Estados (simplificado) do TCP” representa um Diagrama de Estados (simplificado) do protocolo TCP.

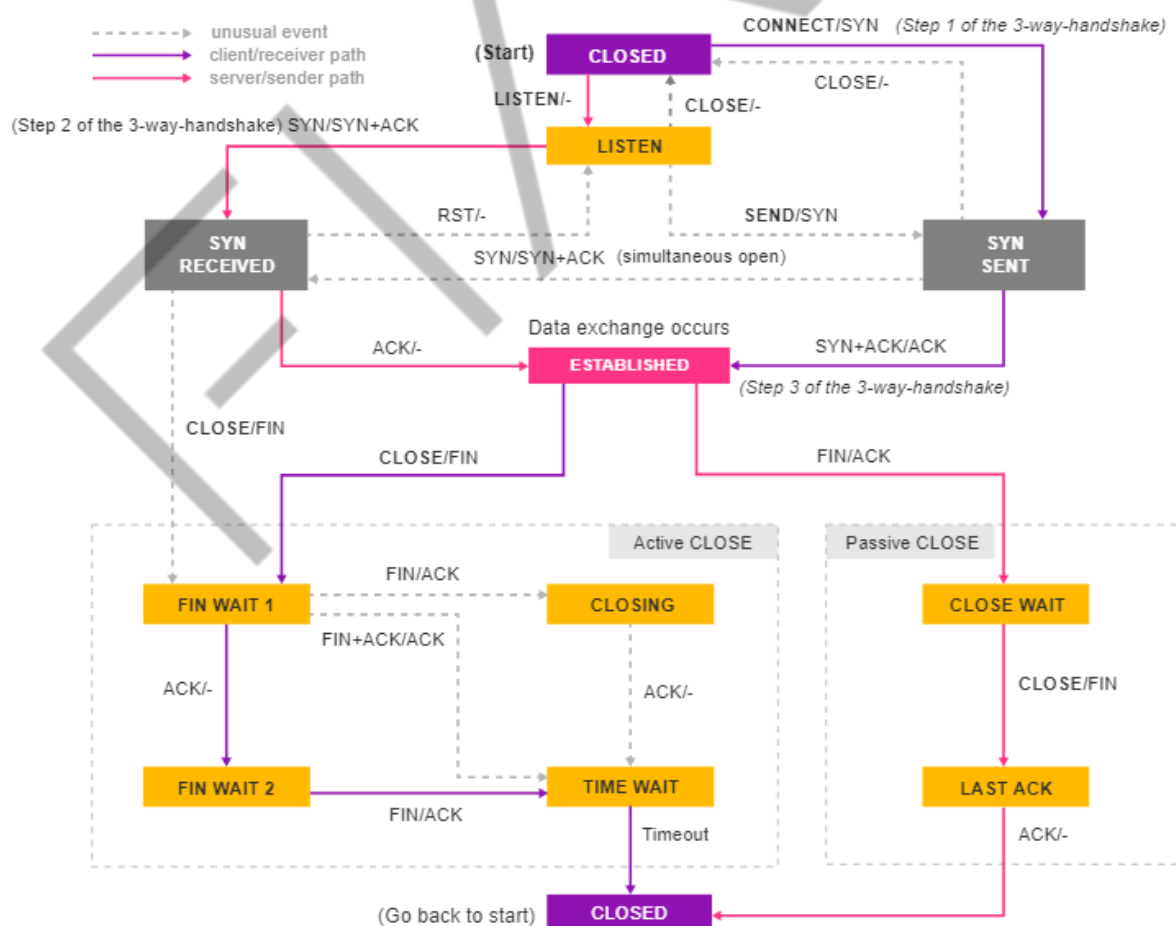


Figura 4.9 – Diagrama de Estados (simplificado) do TCP
Fonte: commons.wikimedia.org (2020)

Da Figura “Diagrama de Estados (simplificado) do TCP”, tem-se os seguintes estados:

- **CLOSED**: também algumas vezes referenciado como estado "ficcional", representa a condição em que não há conexão entre dispositivos, seja por ainda não haver sido estabelecida, seja por já haver sido finalizada ou destruída.
- **LISTEN**: um dispositivo (geralmente um servidor) está aguardando por uma mensagem de sincronização (SYN) enviada por algum cliente, não tendo, ainda, enviado sua própria mensagem SYN a este cliente.
- **SYN-SENT**: o dispositivo (normalmente um cliente) enviou uma mensagem de sincronização (SYN) e está aguardando uma SYN correspondente por parte do outro dispositivo (geralmente um servidor).
- **SYN-RECEIVED**: ambos os dispositivos receberam uma solicitação de conexão (SYN) de seus parceiros, tendo sido já enviados os respectivos SYN; esperando agora por um ACK para seu SYN para finalizar a configuração da conexão.
- **ESTABLISHED**: o "estado estável" de uma conexão TCP estabelecida, quando os dados podem ser trocados livremente uma vez que ambos os dispositivos na conexão se encontram neste estado, assim permanecendo até que a conexão seja finalizada por algum motivo.
- **CLOSE WAIT**: um dos dispositivos recebeu do outro um pedido de encerramento da conexão (FIN), devendo agora esperar até que a aplicação local confirme esta solicitação e gere uma solicitação correspondente.
- **LAST ACK**: o dispositivo que recebeu o pedido de encerramento e já o confirmou enviou seu próprio FIN e está aguardando um ACK para sua requisição.

- **FIN WAIT 1:** um dispositivo neste estado está aguardando um ACK para um FIN que enviou ou aguarda por uma solicitação de finalização de conexão originada pelo outro dispositivo.
- **FIN WAIT 2:** um dispositivo neste estado já recebeu um ACK para sua solicitação de finalização da conexão e agora está aguardando pelo FIN correspondente a ser enviado pelo outro dispositivo.
- **CLOSING:** o dispositivo recebeu um FIN do outro dispositivo e enviou a ele um ACK, mas ainda não recebeu um ACK para sua própria mensagem FIN.

Leia mais em:

<http://tcpipguide.com/free/t_TCPOperationalOverviewandtheTCPFiniteStateMachineF-2.htm>

4.7 TCP/IP: camada de aplicação.

É a camada onde as requisições por dados ou serviços são recebidas e processadas por aplicações “ouvindo” em suas respectivas portas (BLANK, 2004).

É importante salientar que a camada de APLICAÇÃO **não é** onde residem aplicações como um processador de textos ou uma planilha eletrônica, mas, sim, onde aplicações, como um navegador web, por exemplo, fazem uso dos serviços oferecidos por um **protocolo da camada de aplicação** para atender às requisições do usuário, no caso do navegador web, geralmente o protocolo **HTTP**.

Considere-se agora o cenário representado na Figura “Cenário de referência”, onde um único servidor oferece múltiplos serviços ao usuário.

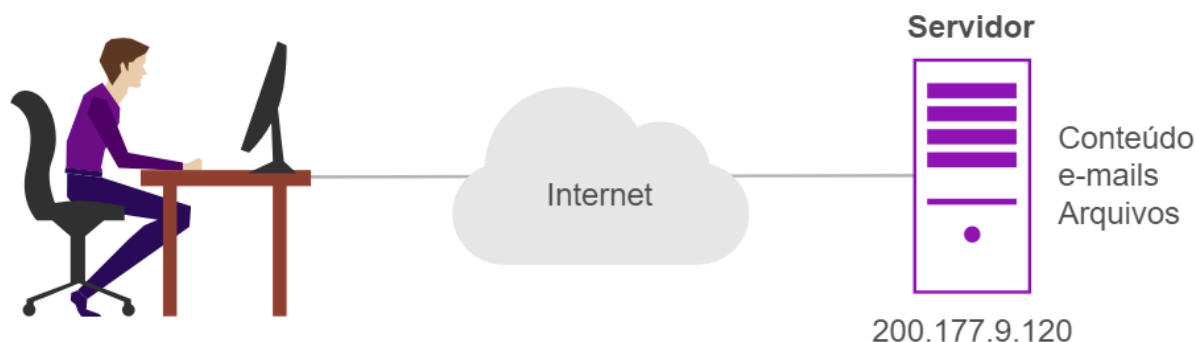


Figura 4.10 – Cenário de referência
Fonte: Elaborado pelo autor (2020)

Isso é possível, pois as aplicações instaladas no servidor – que disponibilizam cada um destes diferentes serviços – “ouvem” as requisições do usuário em portas particulares e padronizadas que apontam para cada um destes serviços individualmente.

Blank (2004) define as portas como endereços numerados entre 1 e 65536, aos quais as requisições são enviadas e onde as aplicações ficam “ouvindo”.

A identificação de cada serviço é feita com base em portas específicas, mais o protocolo da camada de TRANSPORTE (TCP ou UDP) a ser utilizado, sendo tais informações **padronizadas** pela Internet Assigned Number Authority (IANA).

A IANA atribui os números das portas com base em diferentes critérios, definindo três faixas:

- **System Ports:** compreendendo as primeiras 1024 portas – numeradas de 0 a 1023.
- **User Ports:** numeradas de 1024 a 49151.
- **Dynamic and/or Private Ports:** numeradas de 49152 a 65535, sendo as diferenças para o uso destas faixas descritas na RFC6335.

As *system ports* são mais geralmente referenciadas como *well-known ports* (portas bem-conhecidas) e destinadas à identificação de serviços em redes IP privadas ou na Internet; por sua vez, as *user ports* – também referenciadas como *registered ports* (portas registradas) – têm relação com as portas registradas pela IANA para conveniência da comunidade Internet, como, por exemplo, por empresas e outros usuários.

O Quadro “Well-known ports” oferece exemplos das *well-known ports* mais amplamente utilizadas.

Protocolo		Porta	Descrição
Aplicação	Transporte		
SSH	TCP	22	Acesso remoto seguro (criptografado).
TELNET	TCP	23	Acesso remoto sem criptografia.
SMTP	TCP	25	Transferência de e-mails entre servidores ou entre o usuário e um servidor de e-mail.
DNS	UDP / TCP	53	Resolução de nomes. / Troca de zona entre servidores DNS.
DHCP	UDP	67/68	Endereçamento dinâmico de <i>hosts</i> .
HTTP	TCP	80	Navegação <i>web</i> .
POP	TCP	110	Recebimento de e-mails.
NTP	UDP	123	Sincronização de relógios dos <i>hosts</i> em uma rede.

Quadro 4.4: Well-known ports
Fonte: Elaborado pelo autor (2020)

Com base no até aqui exposto, é possível observar que a identificação do processo de uma aplicação, ou, simplificada, a referência ao serviço desejado, é feita com base na combinação: <IP_do_servidor>:<porta_do_serviço>, a qual se denomina *socket*.

Considerando-se ainda a Figura “Cenário de referência”, se o usuário solicitasse ao servidor uma página web, seu *browser* encaminharia a este servidor uma requisição usando a notação 200.177.9.120:80; caso o usuário solicitasse o *download* de um arquivo, seu cliente FTP utilizaria a notação 200.177.9.120:21, ressaltando-se que o próprio aplicativo cliente do serviço (no caso, o *browser* ou cliente FTP do usuário) automaticamente seleciona o protocolo de camada 4 (TRANSPORTE) adequado ao serviço: TCP ou UDP.

A Figura “Sockets e estados das conexões” ilustra uma saída da ferramenta *netstat* na qual se pode observar os *sockets* ativos e os estados das conexões, conforme referenciado na descrição dos estados das conexões.

Arquivo Editor Ver Terminal Abas Ajuda

```
root@target:/home/user1# netstat -nat
```

Conexões Internet Ativas (servidores e estabelecidas)

Proto	Recv-Q	Send-Q	Endereço Local	Endereço Remoto	Estado
tcp	0	0	0.0.0.0:22	0.0.0.0:*	OUÇA
tcp	0	0	0.0.0.0:3000	0.0.0.0:*	OUÇA
tcp	0	0	127.0.0.1:25	0.0.0.0:*	OUÇA
tcp	0	0	0.0.0.0:48835	0.0.0.0:*	OUÇA
tcp	0	0	0.0.0.0:111	0.0.0.0:*	OUÇA
tcp	0	0	192.168.33.11:39475	34.192.94.126:443	ESTABELECIDA
tcp	0	0	192.168.33.11:35790	23.77.33.34:443	ESTABELECIDA
tcp	0	0	192.168.33.11:49880	54.172.42.156:443	TIME_WAIT
tcp	0	0	192.168.33.11:48271	208.84.244.103:443	ESTABELECIDA
tcp	0	0	192.168.33.11:50485	208.84.244.40:443	TIME_WAIT
tcp	0	0	192.168.33.11:50763	156.154.202.36:443	TIME_WAIT
tcp	0	0	192.168.33.11:36849	34.232.123.151:443	TIME_WAIT
tcp	0	0	192.168.33.11:47048	192.16.58.186:443	ESTABELECIDA
tcp	0	0	192.168.33.11:35210	151.101.94.2:443	ESTABELECIDA
tcp	0	0	192.168.33.11:45829	52.200.18.175:443	TIME_WAIT
tcp	0	0	192.168.33.11:57291	192.16.58.8:80	TIME_WAIT
tcp	0	0	192.168.33.11:33780	184.172.206.44:443	TIME_WAIT
tcp	0	0	192.168.33.11:50789	156.154.202.36:443	TIME_WAIT
tcp	0	0	192.168.33.11:44526	23.77.34.15:443	ESTABELECIDA
tcp	0	0	192.168.33.11:51615	8.43.72.42:443	TIME_WAIT
tcp	0	0	192.168.33.11:40400	74.119.119.70:443	ESTABELECIDA

SOCKETS

ESTADOS DAS CONEXÕES

Figura 4.11 – Sockets e estados das conexões
Fonte: Elaborado pelo autor (2020)

4.8 Hands on – iniciando a criação do ambiente

Nesta seção, será iniciada a construção do ambiente a ser utilizado para desenvolvimento prático do conteúdo abordado, o qual terá como base máquinas virtuais, daqui para a frente referenciadas simplesmente como VMs. Resumidamente, uma máquina virtual é um computador completo virtualizado dentro de outro computador, o qual compartilha os recursos da máquina hospedeira (*host*) para poder operar.

Em última análise, as máquinas virtuais consistem de arquivos, que, em nosso caso, serão executados dentro do VirtualBox, aplicação (*hypervisor* tipo 2) especificamente destinada à criação e ao gerenciamento de máquinas virtuais, o qual será instalado sobre o sistema operacional original do computador.

O uso de VMs permite ao usuário executar implementações, testes, análises e quaisquer outras tarefas que possam ser desenvolvidas em máquinas reais, entretanto, com relevantes diferenciais em relação a elas, como, por exemplo:

(a) caso algum problema venha a ocorrer durante a execução de uma implementação, o usuário poderá retornar a VM a um estado anterior, previamente definido (*snapshot*), mais rápida e facilmente do que faria em máquina real;

(b) VMs podem oferecer ambiente adequado a testes de segurança potencialmente perigosos, como, por exemplo, análise de *malwares*;

(c) por consistirem em arquivos, os backups de VMs preservarão não apenas os dados, mas, também, o ambiente completo e já configurado pelo usuário.

O VirtualBox é uma solução completa para virtualização, desenvolvida para hardware x86, mantido pela Oracle. É importante salientar que, como serão criadas máquinas virtuais que compartilharão dos recursos da máquina real – especialmente memória física (RAM), espaço para armazenamento em disco e capacidade de processamento –, as características da máquina real serão determinantes para o provisionamento destes recursos para as VMs. Deve-se ressaltar, ainda, a necessidade do devido suporte à virtualização via hardware por parte do processador da máquina central, e ativação deste no BIOS do sistema (máquina real).

Recomenda-se como plataforma (mínima) adequada aos estudos: CPU Intel i3; 4 GB RAM, HD com 40 GB em disco dedicados aos cenários.

4.9 Download e instalação do VirtualBox

O licenciamento do VirtualBox é baseado no modelo GNU GPLv.2, devendo os downloads, tanto do aplicativo quanto do manual do usuário, entre outros itens relacionados, serem feitos diretamente a partir do site do fabricante, em <http://www.virtualbox.org>.

A Oracle oferece versões específicas para diversas plataformas: Windows, OS X, Linux e Solaris. A abordagem prática do conteúdo estudado será iniciada pela instalação do VirtualBox, considerando-se um *host* executando o sistema operacional Windows10 (se possível, mas não imprescindível, versão Pro).

Os procedimentos para tal consistem em:

- a. Download da versão corrente do **binário** do **VirtualBox** para Windows, a partir de <https://www.virtualbox.org/wiki/Downloads>.



Figura 4.12 – Download da versão corrente do VirtualBox
Fonte: Virtual Box (2020)

Finalizado o download, e antes do início da instalação, a integridade do arquivo deverá ser verificada, clicando sobre ele com o botão direito, e optando por: CRC SHA > SHA-256.

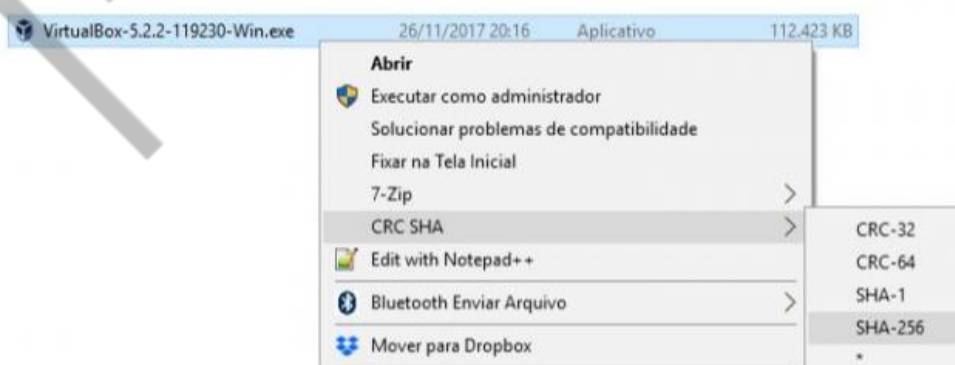


Figura 4.13 – Verificação da integridade do arquivo
Fonte: Elaborado pelo autor (2020)

Este comando retornará o *hash* (SHA 256) do arquivo baixado, o qual deverá ser comparado ao *hash* fornecido em link existente na mesma página do site a partir

da qual o download foi executado. A instalação deverá ser iniciada apenas se eles coincidirem, assegurando, assim, a integridade do arquivo. Caso os *hashes* venham a divergir, mesmo que parcialmente, novo arquivo deverá ser baixado.

Para mais informações: <https://rilcoinblog.com/2017/09/09/cryptocurrency-hash-and-the-difference-between-sha-256-and-script/>

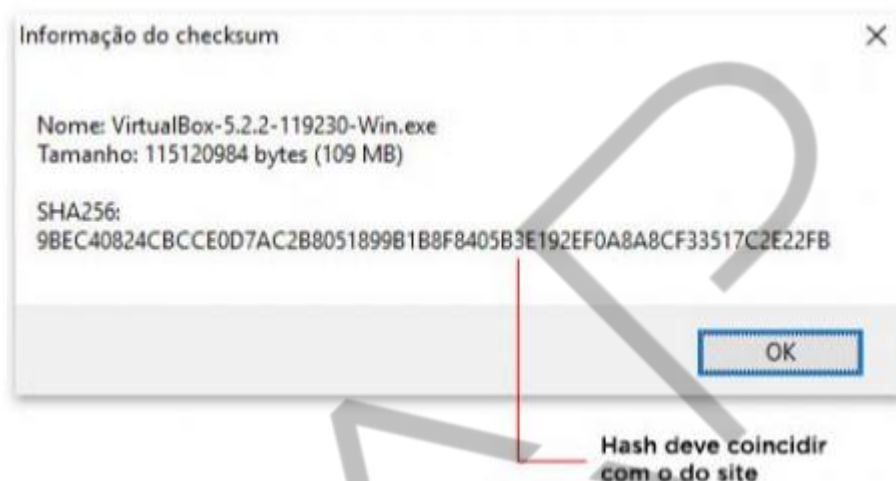


Figura 4.14 – Hash do arquivo de instalação do VirtualBox
Fonte: Elaborado pelo autor (2020)

- b. Confirmada a integridade do arquivo, a instalação poderá, então, ser iniciada pelo administrador do sistema, aceitando-se todas as opções-padrão propostas pelo instalador.
- c. Finalizada a instalação do aplicativo, deverá, então, ser instalado o **Extension Pack**, pacote necessário à correta configuração do vídeo e uso de dispositivos USB, entre outras funcionalidades. Para tal, sugere-se o download do pacote por meio do respectivo link, disponível na mesma página de download do binário. Finalizado o download, o Virtualbox deverá ser **(re)iniciado pelo administrador**. E no menu principal, seguir o caminho: **Arquivo -> Preferências -> Extensões**, clicando-se, então, no ícone “Adicionar pacote”. O processo de instalação do Extension Pack é simples, requerendo apenas que o usuário aceite os termos da licença. Após sua correta instalação, o VirtualBox estará pronto para criação das VMs, **não** necessitando mais ser executado como administrador.

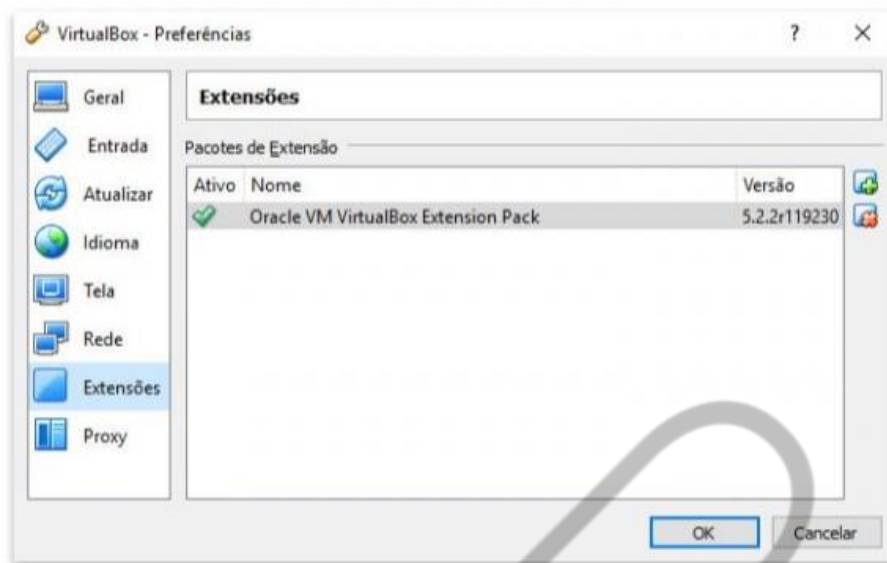


Figura 4.15 – Janela Extensões do VirtualBox
Fonte: Elaborado pelo autor (2020)

4.10 Download e verificação da imagem do Debian

O Debian é um software livre, uma das distribuições Linux mais tradicionais e amplamente utilizadas, tanto em ambientes domésticos quanto corporativos, dadas sua robustez, confiabilidade e expressiva quantidade de pacotes (quando da elaboração deste documento, eram mais de 51 mil).

Leia mais sobre a definição de software livre em: <https://www.debian.org/intro/free>

É relevante conhecer tal conceito, uma vez que muitas das ferramentas utilizadas em CyberSecurity têm esta vertente, cabendo destacar aqui que o termo *free software* deve ser entendido como software livre, e não como software gratuito.

A criação da VM do Debian também será iniciada pelo download de uma imagem (ISO) de instalação, do tipo *netinstall*, ou seja, uma imagem enxuta, com o estritamente necessário para inicializar a VM e o processo de instalação; baixando tudo mais que necessitar a partir da Internet. Reforça-se aqui mais uma vez que, a exemplo do anteriormente feito:

- a. O download da imagem de instalação deverá ser feito a partir do site oficial do projeto, em <https://www.debian.org/>; onde poderá ser encontrado um link para download da versão corrente (imagem única para sistemas de 32 ou 64 bits);

- b. Finalizado o download, a integridade da imagem deverá ser verificada por meio da extração de seu *hash*, e comparação com o *hash* disponível no site do projeto (desta vez, se usado o MD5 em vez do SHA256). Quando da produção deste documento, a versão corrente era a 103 (codinome Buster).



Figura 4.16 – Download da imagem (ISO) do Debian
Fonte: Debian (2020)

Entretanto, sugere-se, desta vez, que a integridade da imagem baixada seja verificada pelo WinMD5, uma ferramenta alternativa à anteriormente usada, que permite comparar de maneira mais fácil o *hash* do arquivo baixado com o *hash* (MD5) disponível no site do projeto.

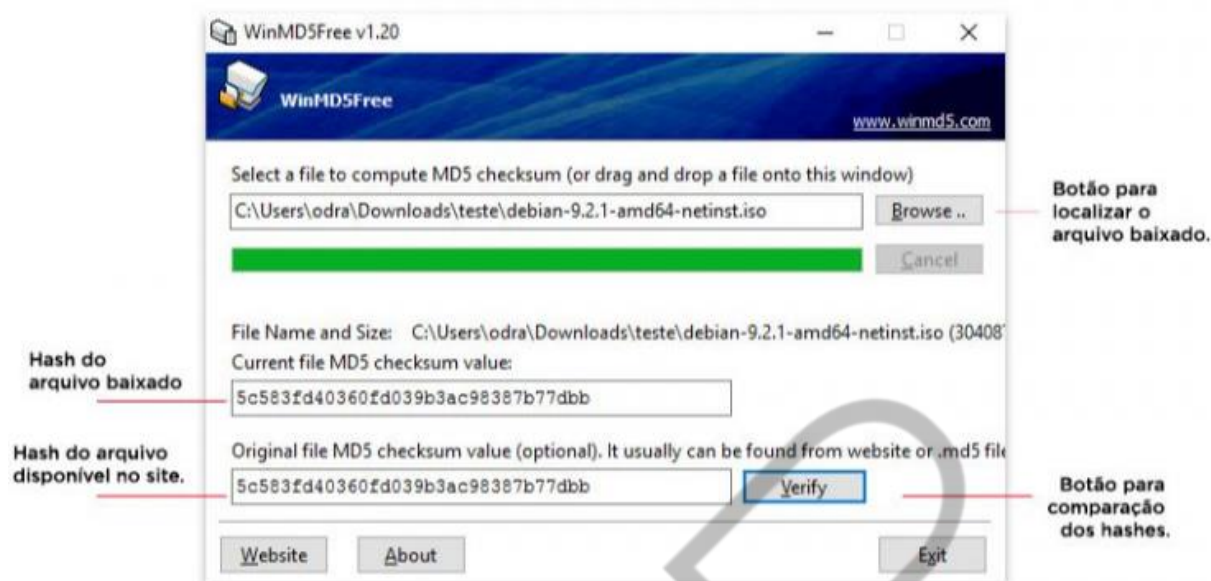


Figura 4.17 – Ferramenta WinMD5
Fonte: Elaborado pelo autor (2020)

Se o arquivo baixado estiver íntegro, o aplicativo emitirá a confirmação.

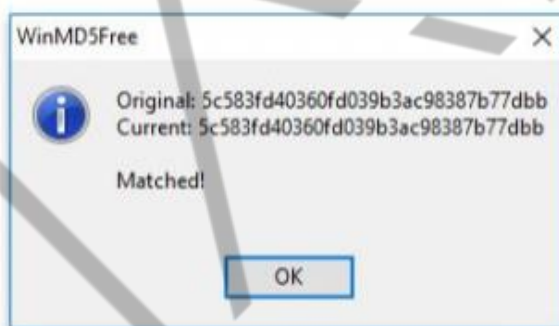


Figura 4.18 – Conformação dos hashes
Fonte: Elaborado pelo autor (2020)

Dica: pode-se também calcular o *hash* MD5 do arquivo baixado e depois copiá-lo da ferramenta (WinMD5), colando-o na barra de pesquisa do navegador *web*. Se o *hash* calculado corresponder ao do arquivo original, a pesquisa retornará à página do site do projeto que contém este último (entre outras).

4.11 Provisionamento da VM do Debian

Finalizado o download e a devida certificação da imagem (ISO) do Debian, é hora de iniciar a criação da VM. Para tal, o VirtualBox deverá ser iniciado (por usuário regular), clicando-se, então, no botão “Novo” na barra do menu principal, em seu

dashboard. A caixa de diálogo “Criar Máquina Virtual” surgirá solicitando os primeiros parâmetros para a criação da VM e sugerindo a instalação do Windows 7.

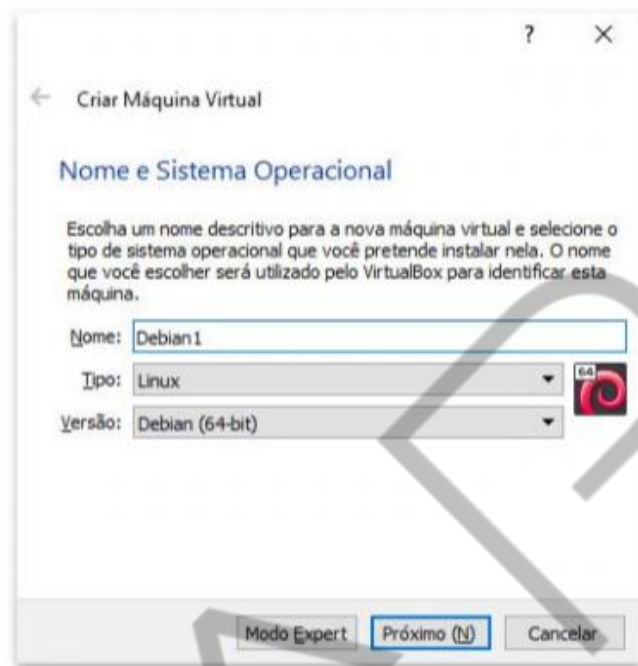


Figura 4.19 – Nome e S.O. da nova VM
Fonte: Elaborado pelo autor (2020)

Preencha o campo “Nome” (Figura “Nome e S.O. da nova VM”) com Debian1 e note que o ícone nesta caixa de diálogo, inicialmente com o logo do Windows 7, muda automaticamente para o logo do Debian, sendo os campos “Tipo” e “versão” também automaticamente preenchidos pelo próprio VirtualBox. Aceite as sugestões oferecidas nestes campos e clique no botão “Próximo (N)”.

A caixa “Tamanho da Memória” surgirá (Figura 4.20) sugerindo a alocação de 1024 MB de memória para a nova VM. Se o sistema hospedeiro tiver esta quantidade de memória disponível, a sugestão poderá ser aceita, caso contrário, reduzir para 768 MB. Após determinada a quantidade de memória para a VM, clicar em “Próximo (N)”.

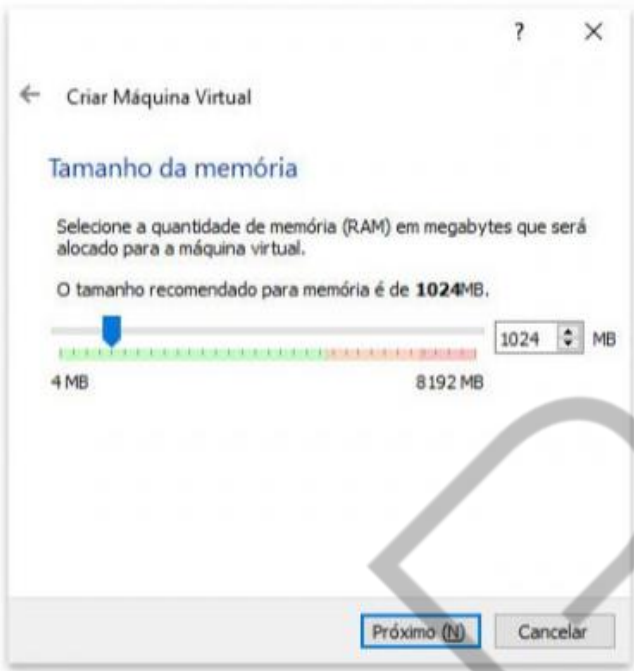


Figura 4.20 – Quantidade de RAM para a VM
Fonte: Elaborado pelo autor (2020)

A partir da caixa de diálogo “Disco rígido”, forneça as opções do Quadro “Opções para a criação do disco rígido da VM”.

Caixa de diálogo	Opção a adotar	Valor do campo
Disco Rígido	Criar um novo disco virtual agora	default
Tipo de arquivo de disco rígido	VDI (VirtualBox Disk Image)	default
Armazenamento em disco rígido físico	Dinamicamente alocado	default
Localização e tamanho do arquivo	Nome do arquivo:	Debian1
	Tamanho da imagem de disco virtual:	default

Quadro 4.5 – Opções para a criação do disco rígido da VM
Fonte: Elaborado pelo autor (2020)

Para finalizar o processo, clique no botão “CRIAR” e a VM será criada. A Figura “Dashboard do VirtualBox com a VM Debian1” ilustra o resumo das características da VM.

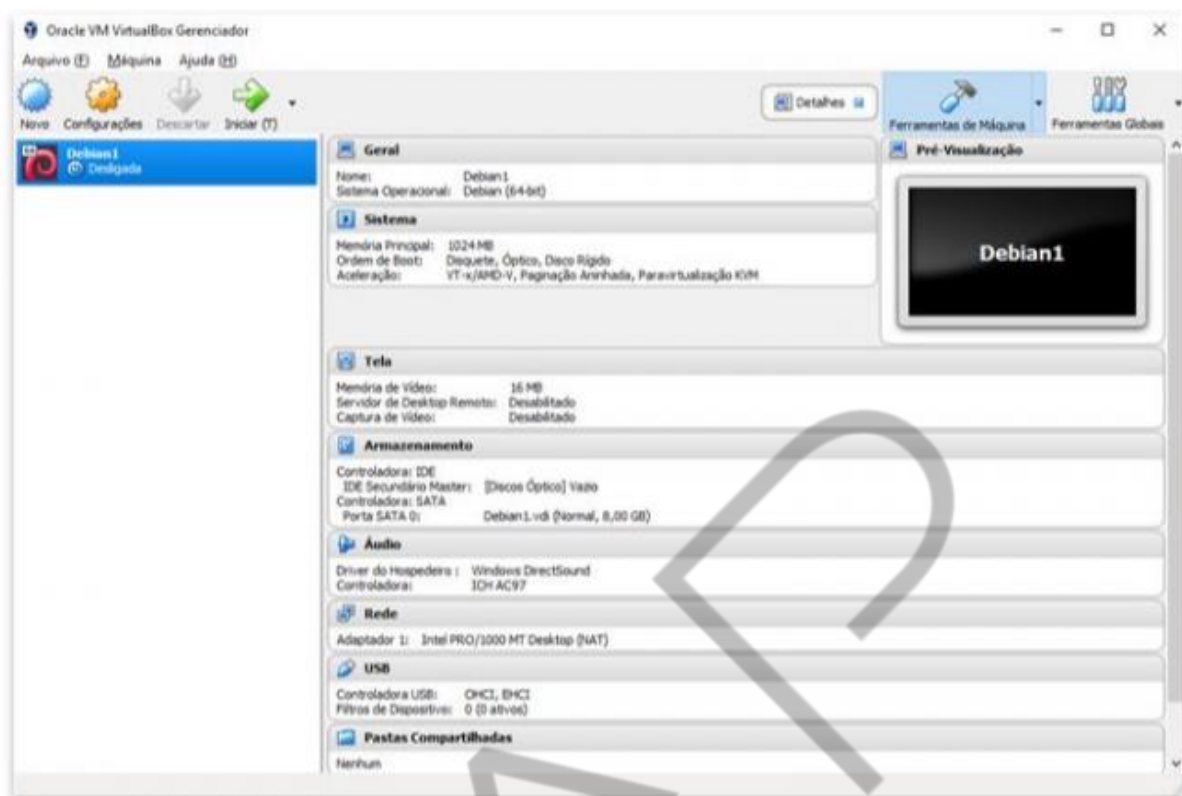


Figura 4.21 – Dashboard do VirtualBox com a VM Debian1

Fonte: Elaborado pelo autor (2020)

A partir do botão “Configurações” da VM, abrir o painel de configurações, procedendo-se aos seguintes ajustes.

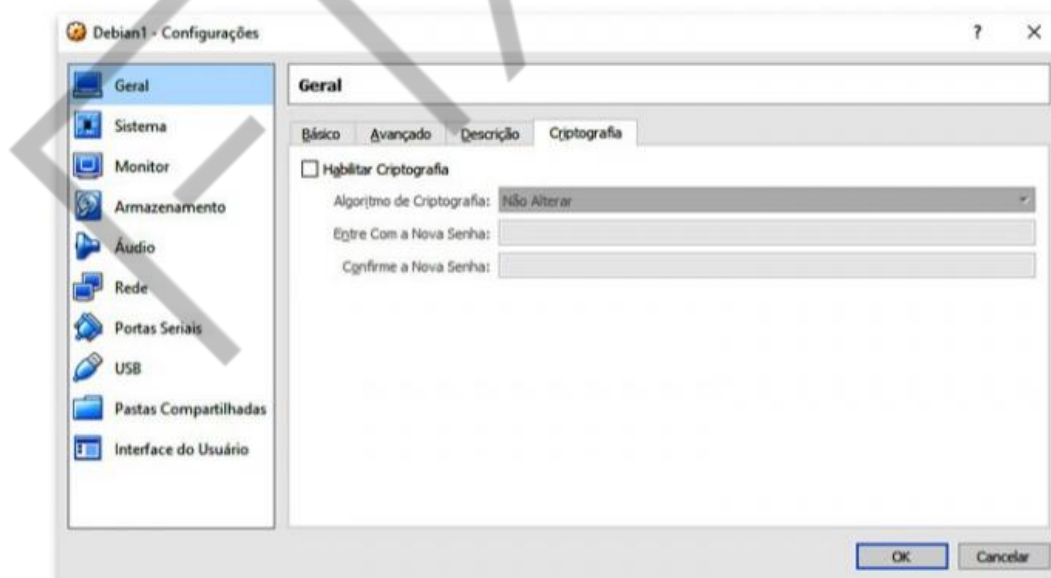


Figura 4.22 – Painel de configurações da VM

Fonte: Elaborado pelo autor (2020)

Ajustes finais da VM:

1) **Geral**: sem alterações.

- 2) **Sistema:** garantir que a opção VT-x/AMD-V esteja habilitada.
- 3) **Monitor:** sem alterações.
- 4) **Armazenamento:** sem alterações.
- 5) **Áudio:** desmarcar a caixa de opção “Habilitar áudio”.
- 6) **Rede:** garantir que a opção vigente é NAT.
- 7) **Portas seriais:** sem alterações.
- 8) **USB:** marcar opção “Habilitar Controladora USB” e seleccionar opção “Controladora USB 2.0 (EHCI).
- 9) **Pastas compartilhadas:** sem alterações.
- 10) **Interface do usuário:** sem alterações.

Feito isto, a instalação do sistema operacional poderá ser iniciada.

4.12 Instalação do Debian na VM

Após certificar-se que há conexão com a Internet, clicar no botão “INICIAR” do *dashboard* do VirtualBox para abertura da janela para seleção da imagem (ISO) a ser instalada, referenciada pelo VirtualBox como disco rígido de *boot* (Figura “Seleção do disco de boot”).

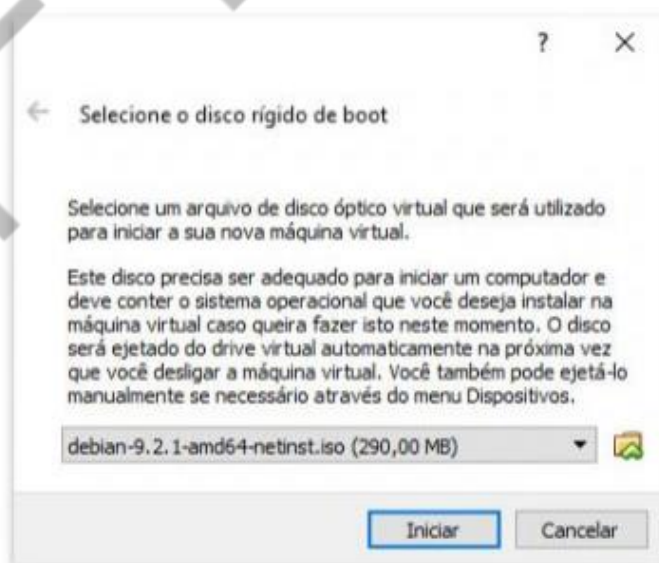


Figura 4.23 – Seleção do disco de boot
Fonte: Elaborado pelo autor (2020)

1. Na janela inicial do instalador do Debian, escolher a opção “INSTALL”.

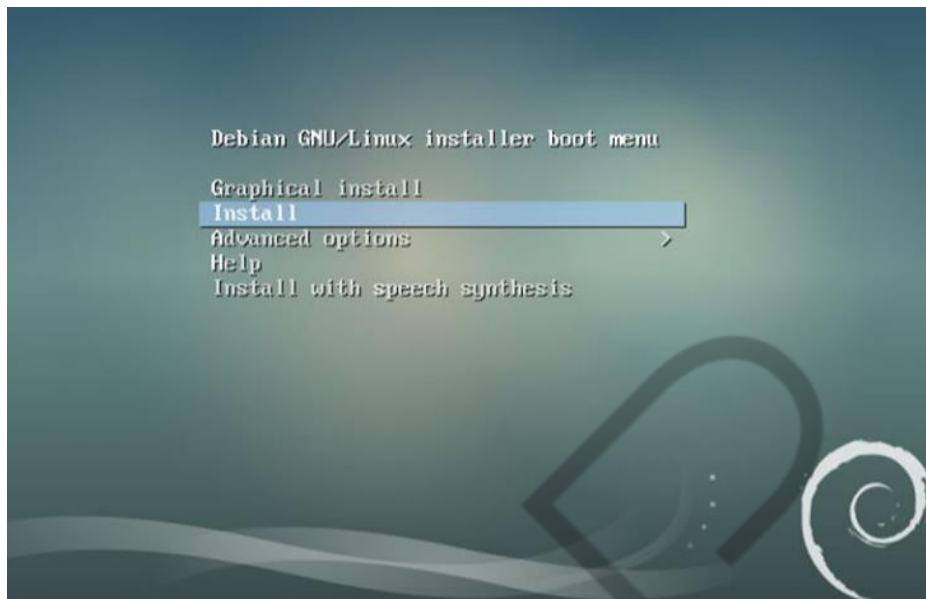


Figura 4.24 – Seleção do tipo de instalação do Debian
Fonte: Elaborado pelo autor (2020)

2. Na janela seguinte, especificar o idioma como “PORTUGUÊS do BRASIL”.

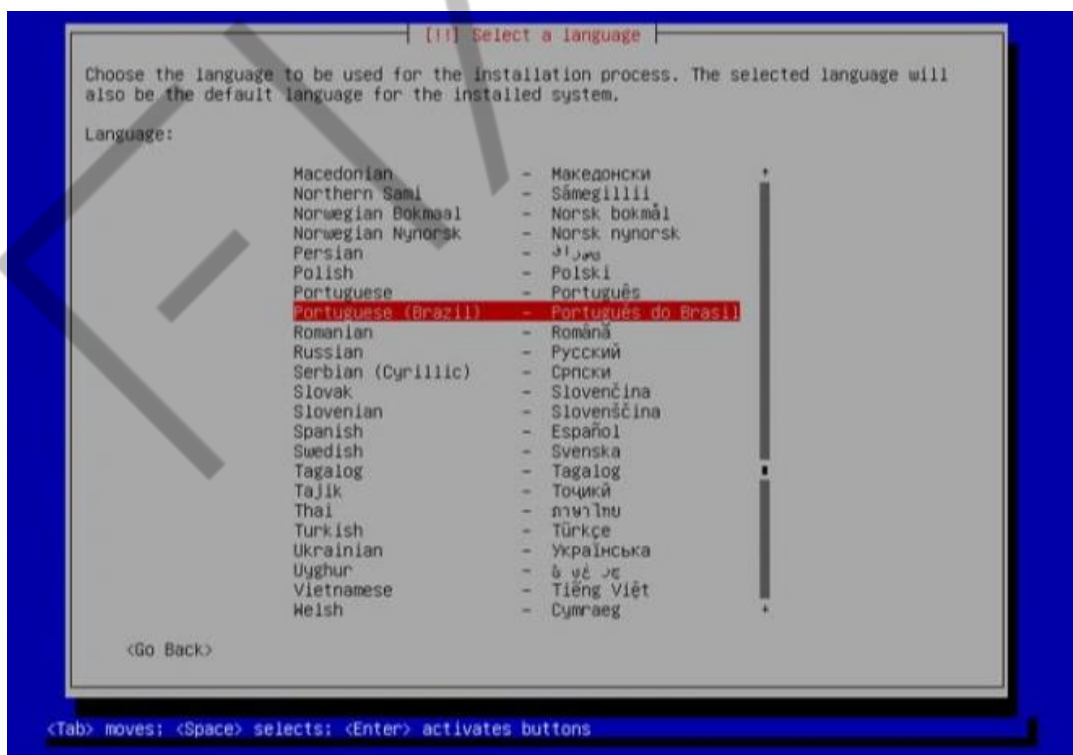


Figura 4.25 – Seleção do idioma
Fonte: Elaborado pelo autor (2020)

3. Definir a localidade desejada como “BRASIL”.

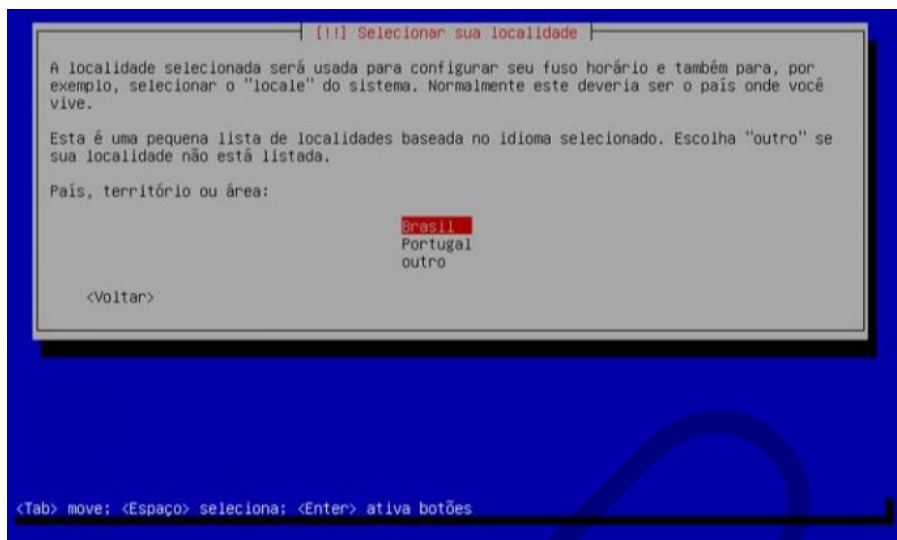


Figura 4.26 – Seleção da localidade
Fonte: Elaborado pelo autor (2017)

4. Configurar o teclado como “Português Brasileiro”.

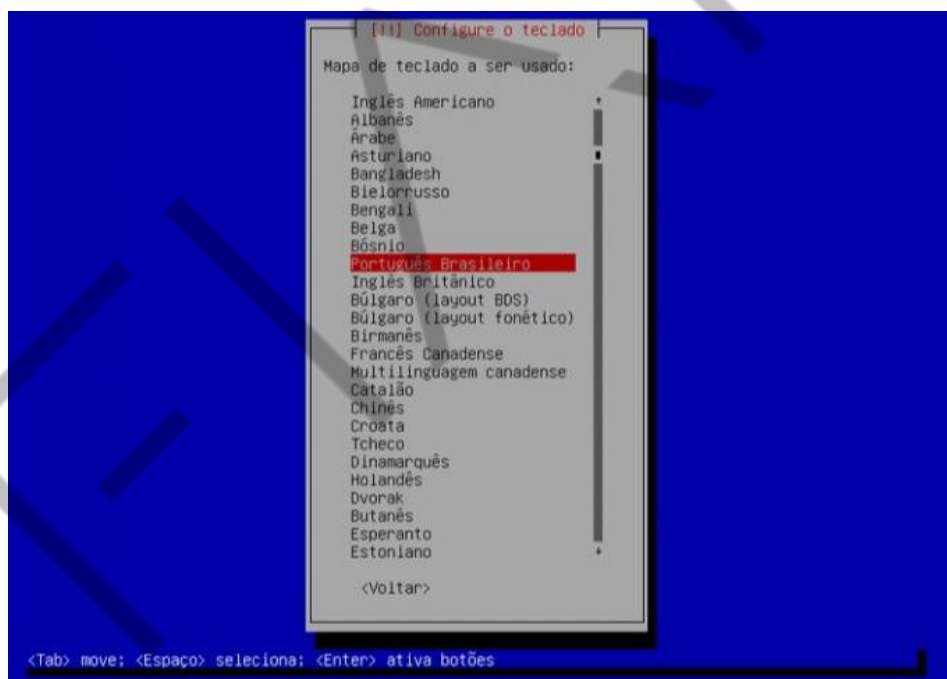


Figura 4.27 – Seleção do teclado
Fonte: Elaborado pelo autor (2020)

5. Aguardar o download dos componentes adicionais.

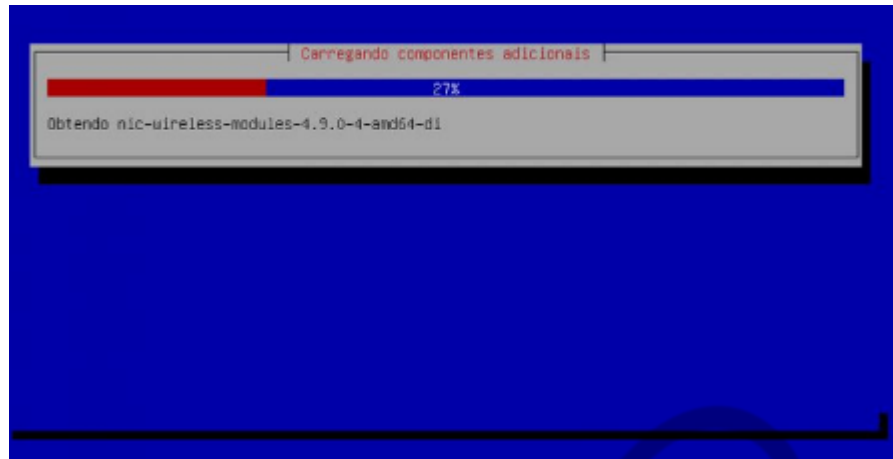


Figura 4.28 – Carga de componentes adicionais
Fonte: Elaborado pelo autor (2020)

6. Fornecer o *hostname* da VM: debian01.

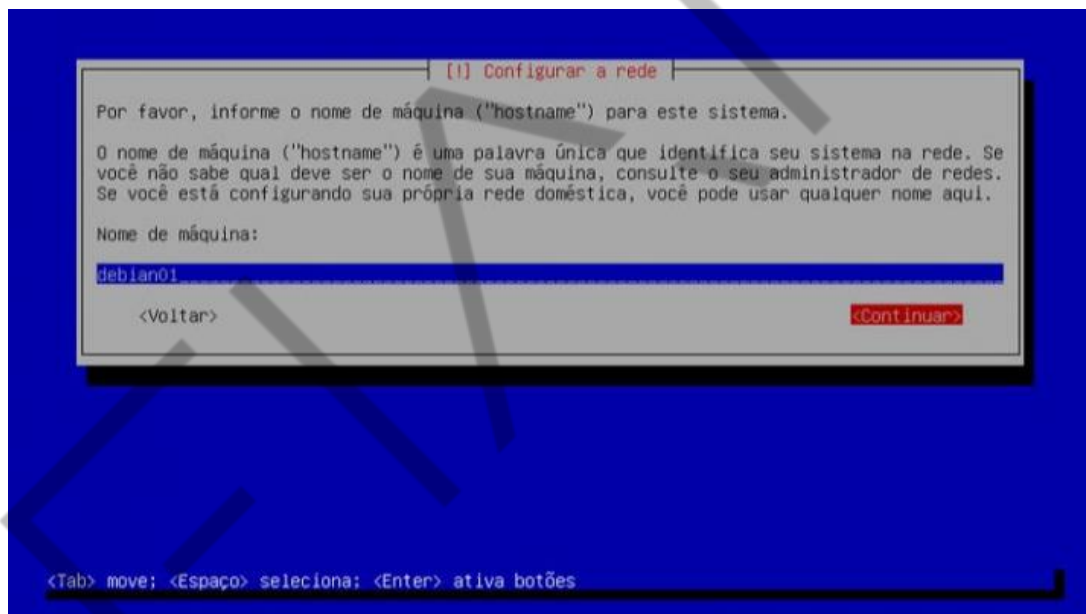


Figura 4.29 – Hostname da VM
Fonte: Elaborado pelo autor (2020)

7. Fornecer o domínio da VM: localdomain.

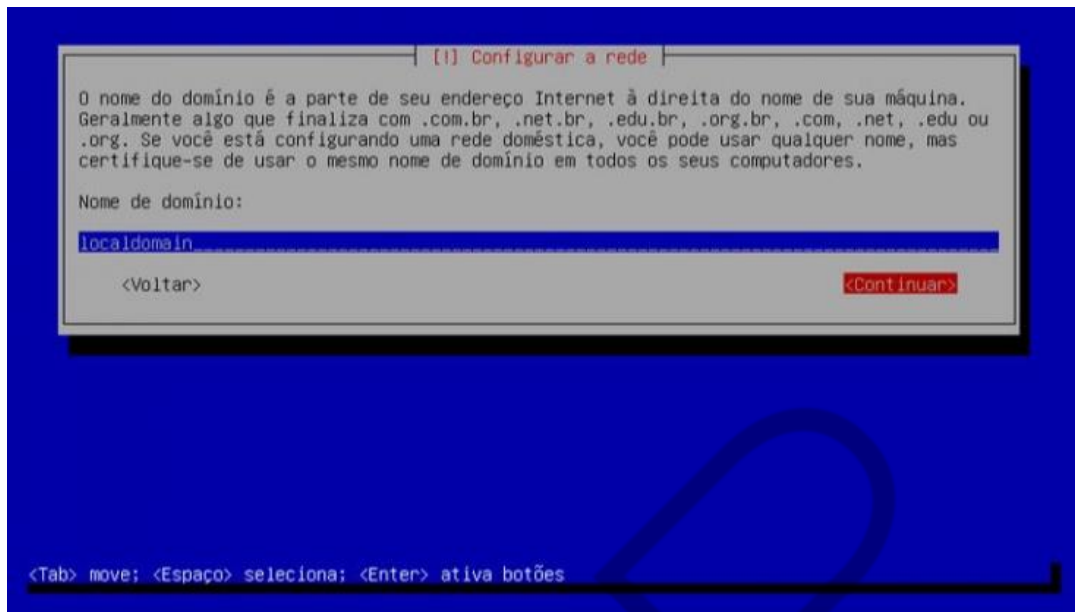


Figura 4.30 – Nome de domínio da VM

Fonte: Elaborado pelo autor (2020)

8. Fornecer a senha do *root*, confirmando esta senha na próxima janela.

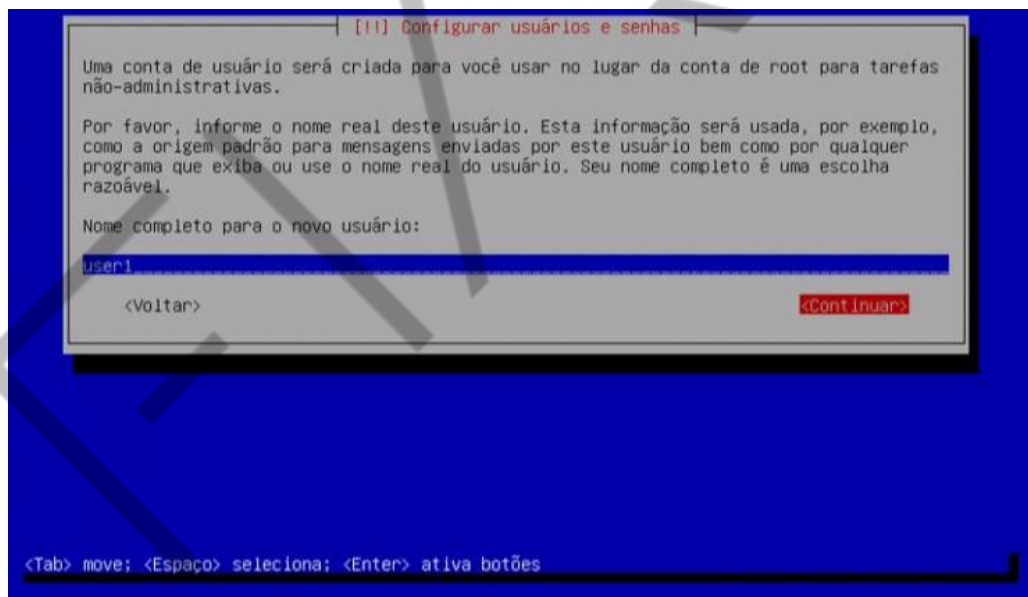


Figura 4.31 – Senha do root

Fonte: Elaborado pelo autor (2020)

9. Fornecer o nome de um usuário desprivilegiado, confirmando na janela subsequente, ao ser solicitado o nome de usuário.

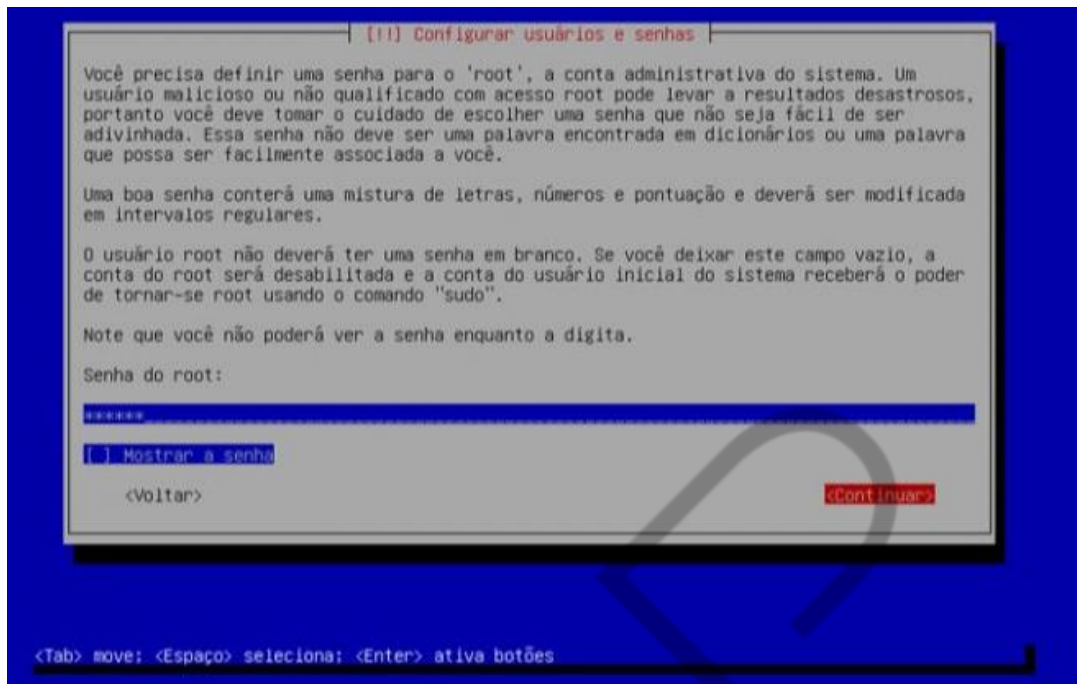


Figura 4.32 – Nome do usuário desprivilegiado
Fonte: Elaborado pelo autor (2020)

10. Nas duas janelas subsequentes, fornecer e confirmar a senha do usuário desprivilegiado.

11. Ao configurar o relógio do sistema, verificar o fuso horário desejado.

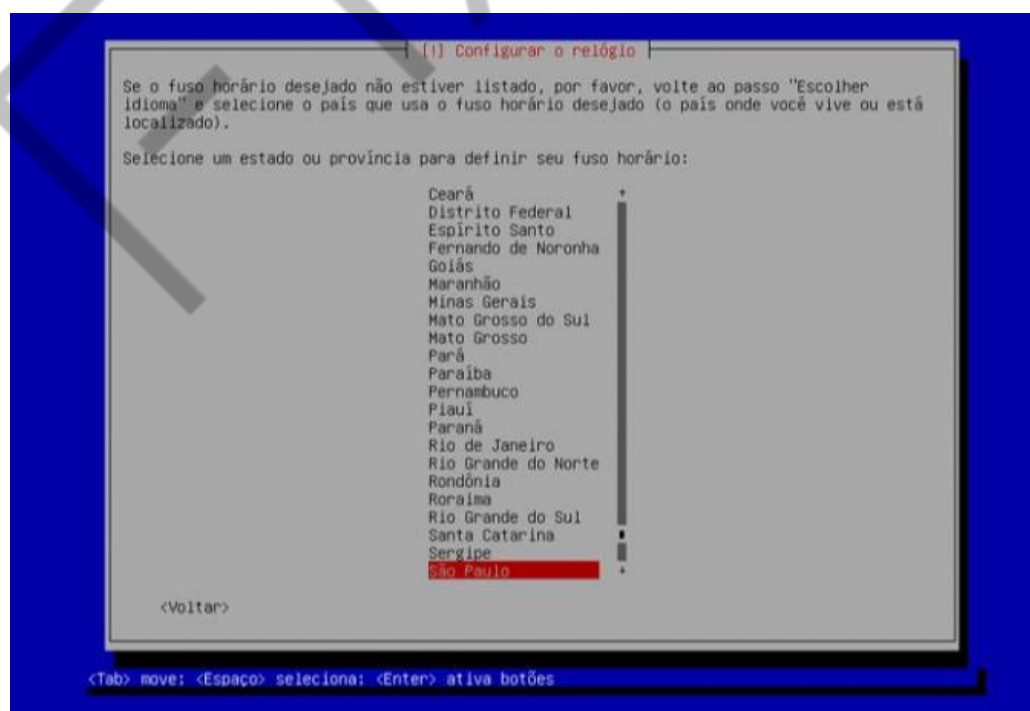


Figura 4.33 – Relógio do sistema
Fonte: Elaborado pelo autor (2020)

12. Aguardar o download de componentes adicionais e selecionar o método de particionamento assistido do disco.

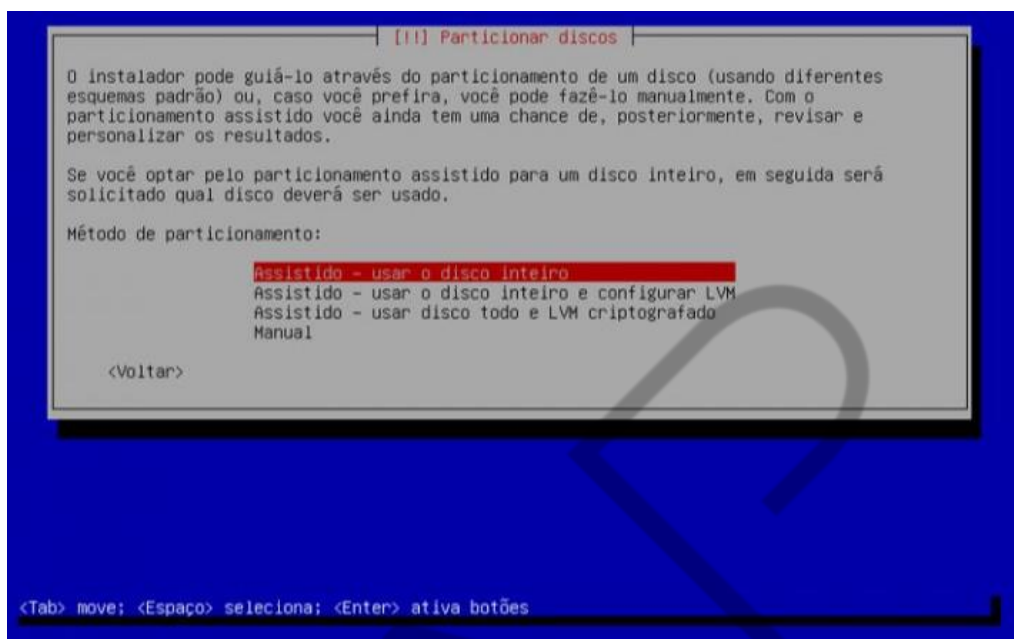


Figura 4.34 – Opção pelo particionamento assistido
Fonte: Elaborado pelo autor (2020)

13. Aceitar a sugestão do sistema para o esquema de particionamento a utilizar e prosseguir com a instalação.

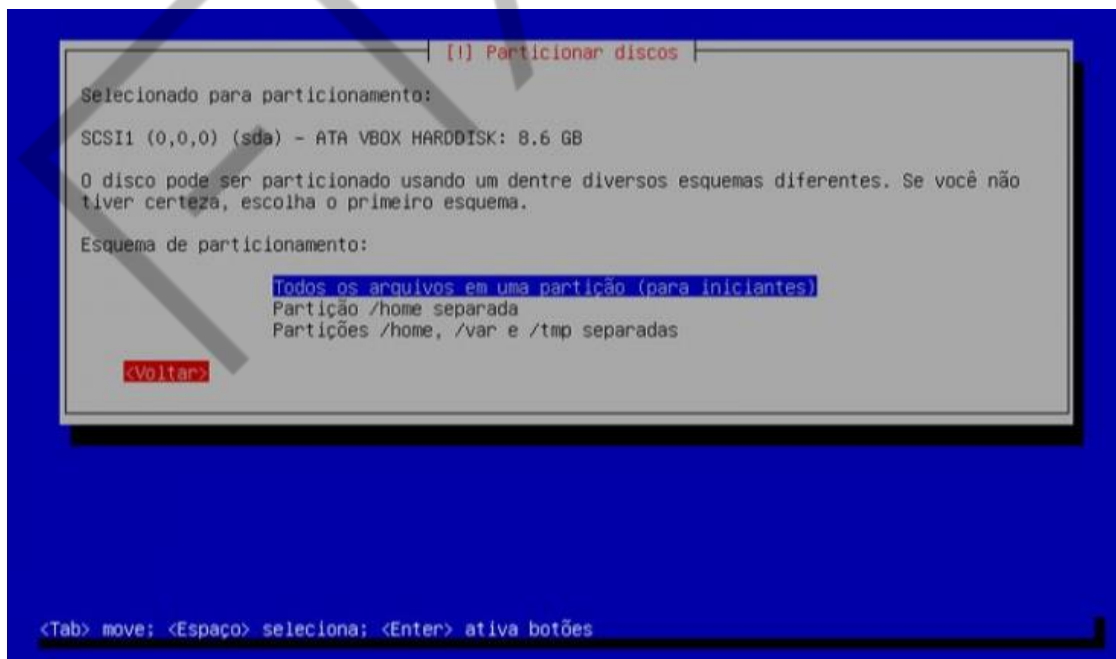


Figura 4.35 – Esquema de particionamento
Fonte: Elaborado pelo autor (2020)

14. Na janela seguinte, aceitar o disco sugerido pelo sistema para ser particionado (sda) e prosseguir com a instalação.

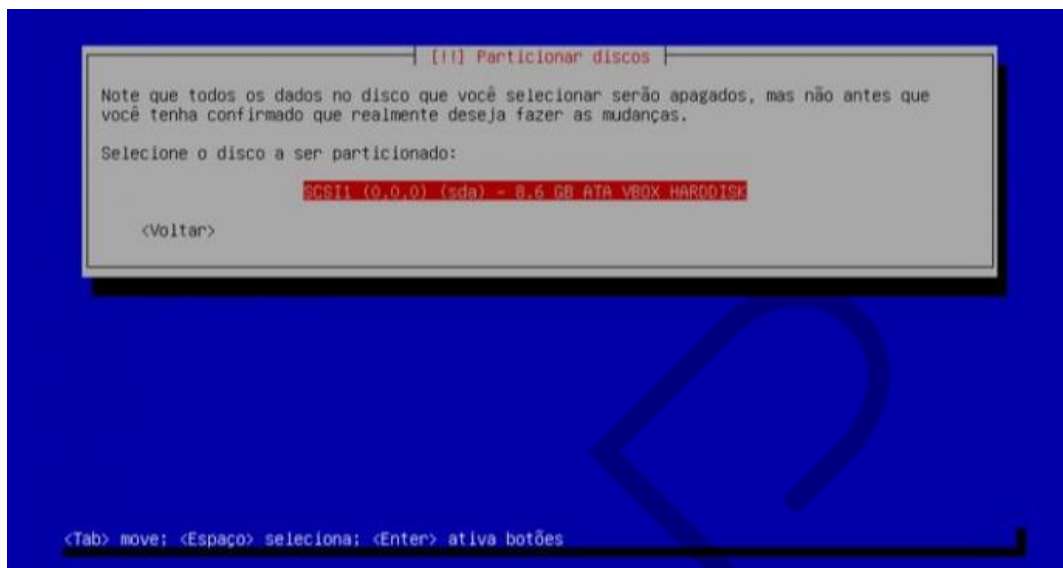


Figura 4.36 – Disco a ser particionado
Fonte: Elaborado pelo autor (2020)

15. Na tela subsequente, optar por: Finalizar o particionamento e escrever as mudanças no disco. Avançar para a próxima tela, nesta respondendo <SIM> para efetivar as mudanças no disco, e continuar.

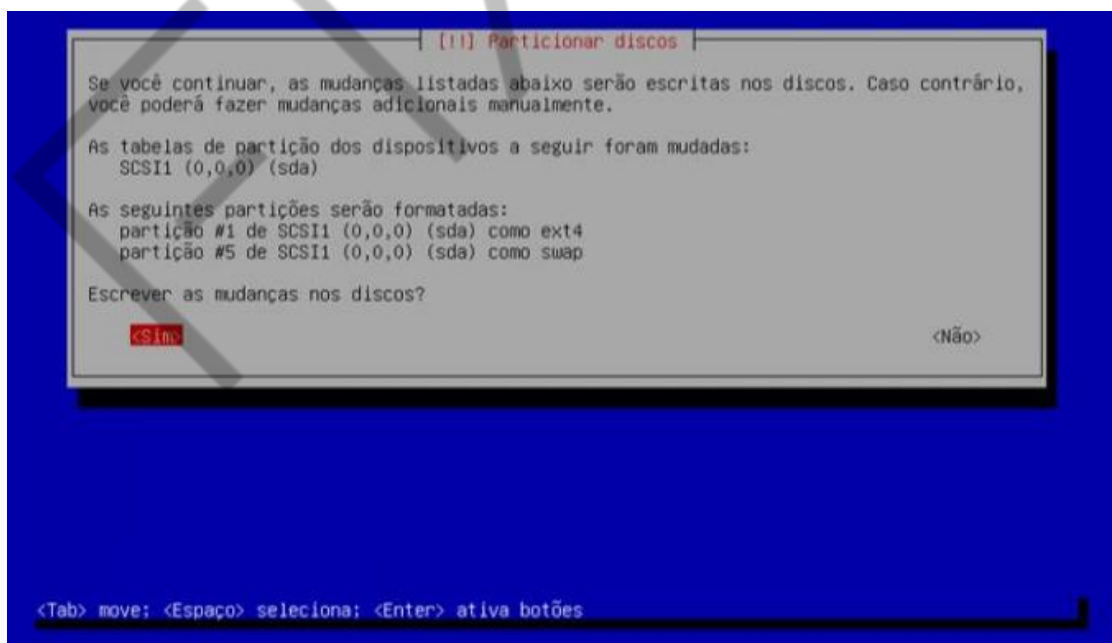


Figura 4.37 – Efetivação das mudanças do disco
Fonte: Elaborado pelo autor (2020)

16. Aguardar a instalação do sistema básico.

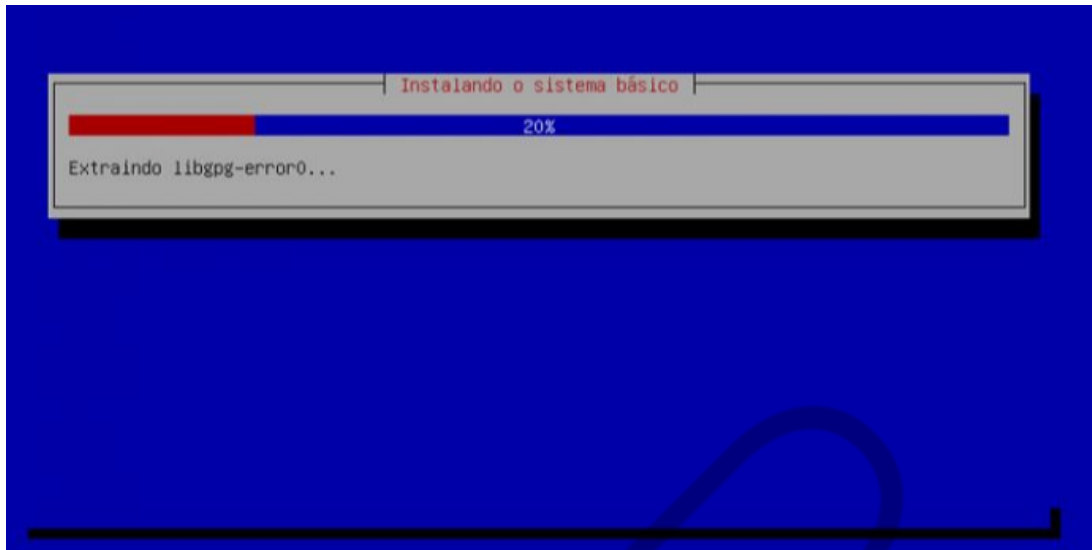


Figura 4.38 – Instalação básica do sistema
Fonte: Elaborado pelo autor (2020)

17. Responder NÃO quando questionado sobre a utilização de outra mídia.

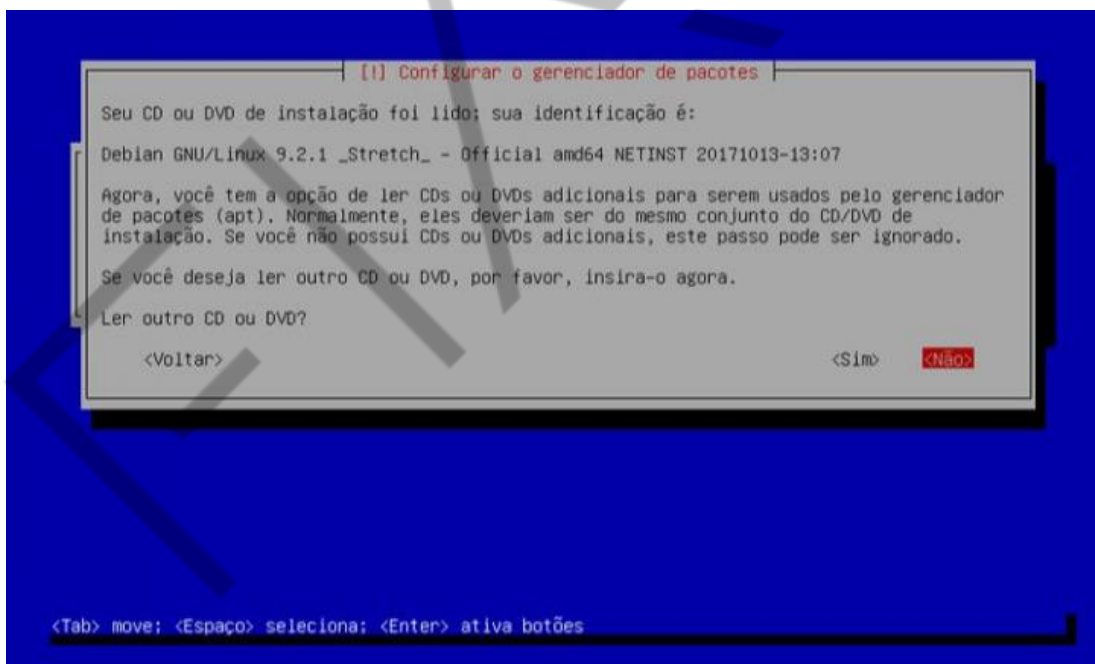


Figura 4.39 – Mídias complementares
Fonte: Elaborado pelo autor (2020)

18. Nas telas subsequentes, responder:

- País do espelho do repositório Debian: Brasil.
- Espelho do repositório Debian: opção-padrão.
- Informação sobre o proxy HTTP: deixar em branco.

19. Na janela “Configurando o apt”, aguardar o download e a instalação dos pacotes que compõem a distribuição.

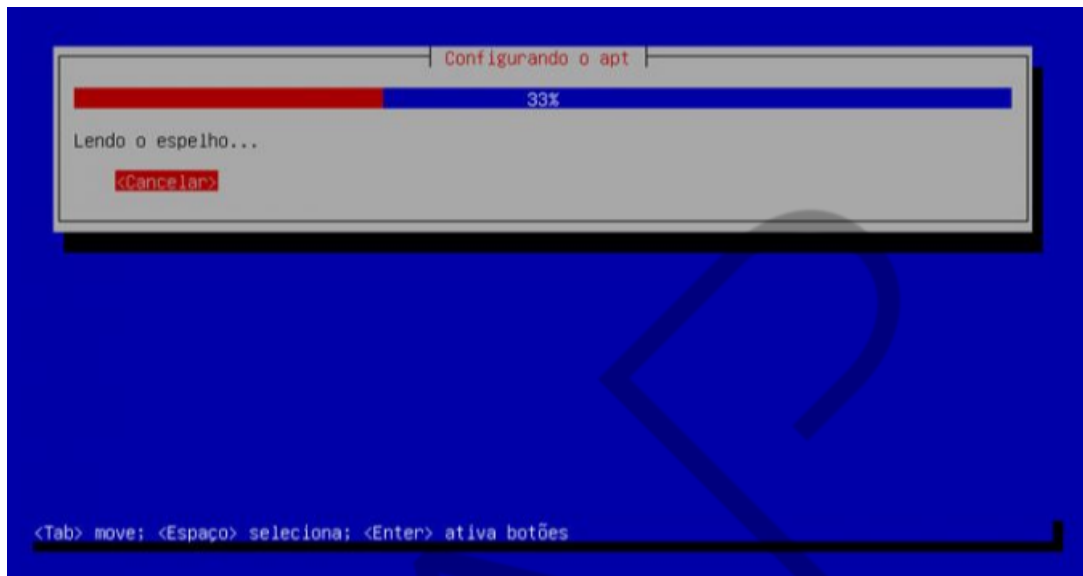


Figura 4.40 – Download e instalação de pacotes
Fonte: Elaborado pelo autor (2020)

20. Na janela “Configurando popularity-contest”, responder <NÃO>, selecionado, na janela subsequente, os softwares a serem instalados conforme a figura.

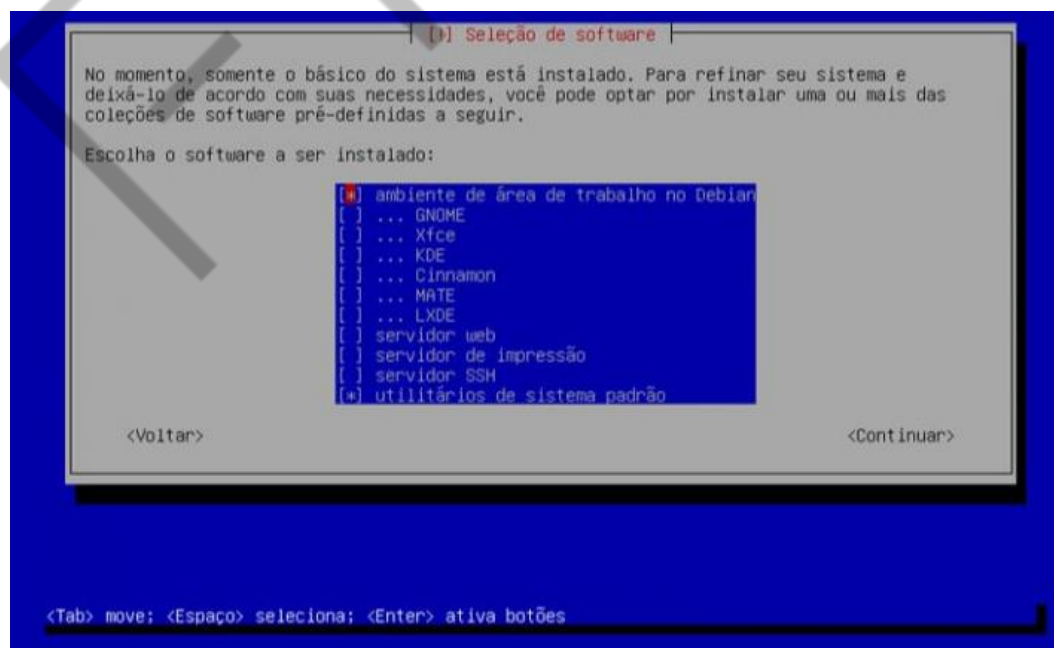


Figura 4.41 – Softwares a instalar
Fonte: Elaborado pelo autor (2020)

21. Ao ser solicitado pelo instalador, responder <SIM> para que o GRUB seja instalado no registro-mestre de inicialização.

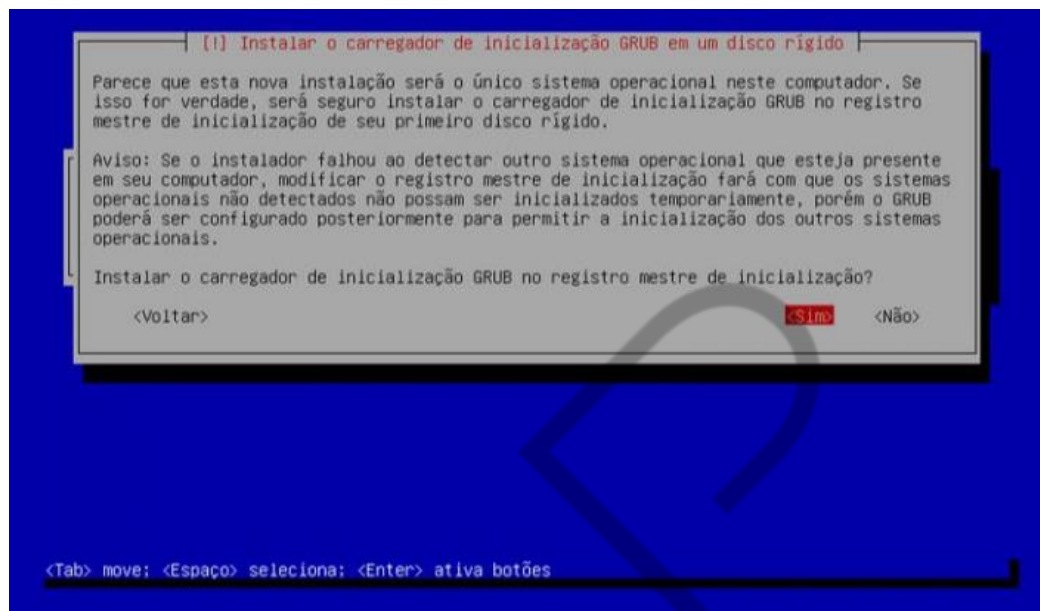


Figura 4.42 – Instalação do GRUB em disco rígido
Fonte: Elaborado pelo autor (2020)

22. Ao ser solicitado pelo instalador, responder </dev/sda> como dispositivo para a instalação do carregador de inicialização.

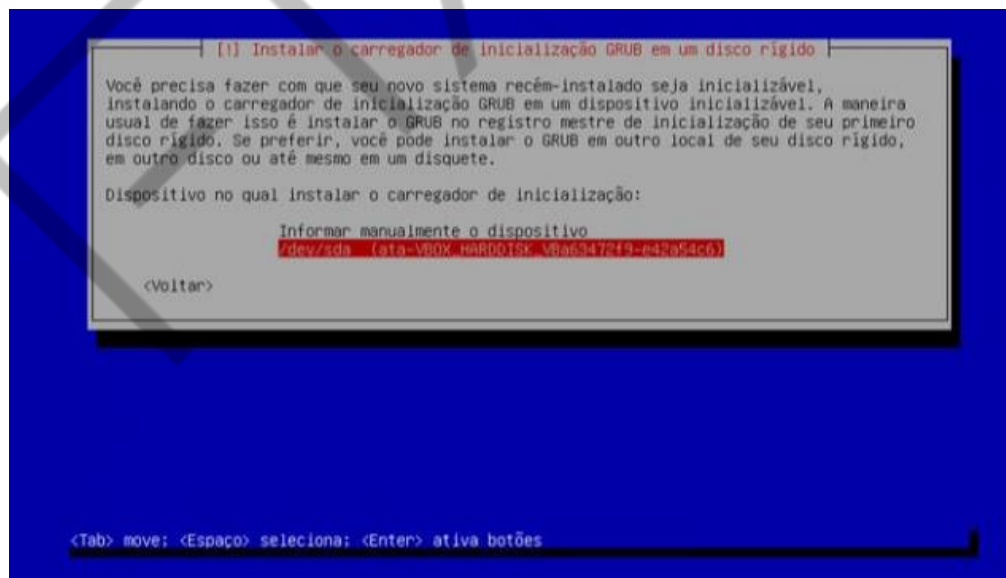


Figura 4.43 – Instalação do GRUB em /dev/sda
Fonte: Elaborado pelo autor (2020)

23. Finalizada a instalação, reinicializar a VM.

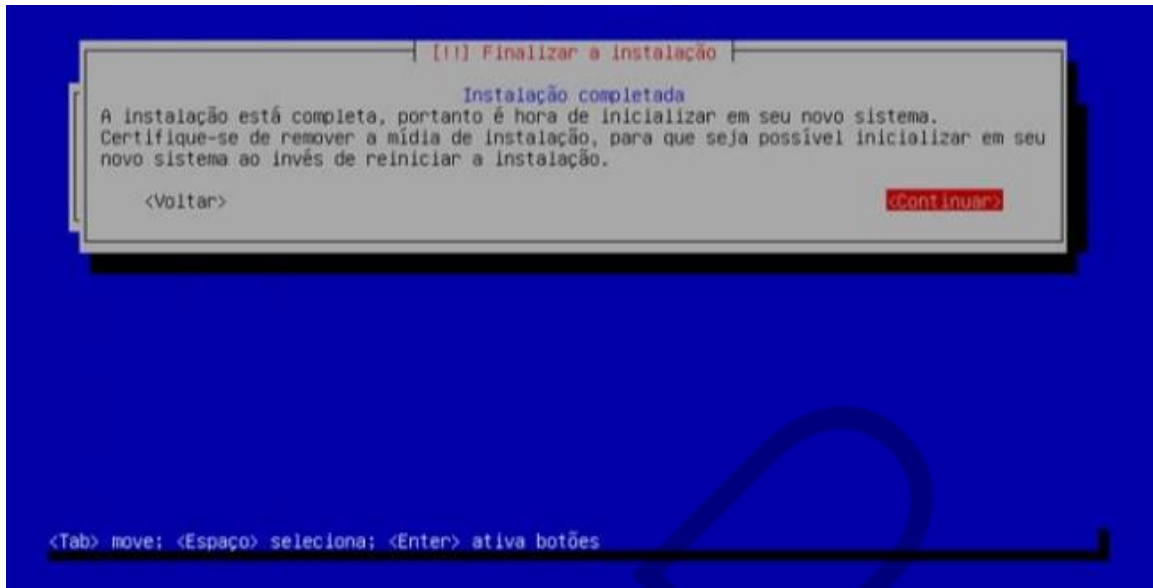


Figura 4.44 – Reinicialização após final da instalação
Fonte: Elaborado pelo autor (2020)

4.13 Ajustes iniciais no ambiente

Estando a VM do Debian pronta para utilização, o usuário deverá (re)inicializá-la e logar no sistema, para que, então, sejam executados os ajustes iniciais do ambiente, os quais consistem:

1. Da instalação do adicional de convidados do VirtualBox;
2. Da instalação do pacote net-tools;
3. Da instalação do Wireshark;
4. Desabilitar suporte ao IPv6.

4.13.1 Instalação do adicional de convidados

- a. Iniciar a VM e, após logar no ambiente, abrir um terminal e tornar-se *root*.
- b. A instalação do adicional de convidados inicia-se pela opção Dispositivos-> Inserir imagem de CD dos adicionais de convidados, na barra de menu.

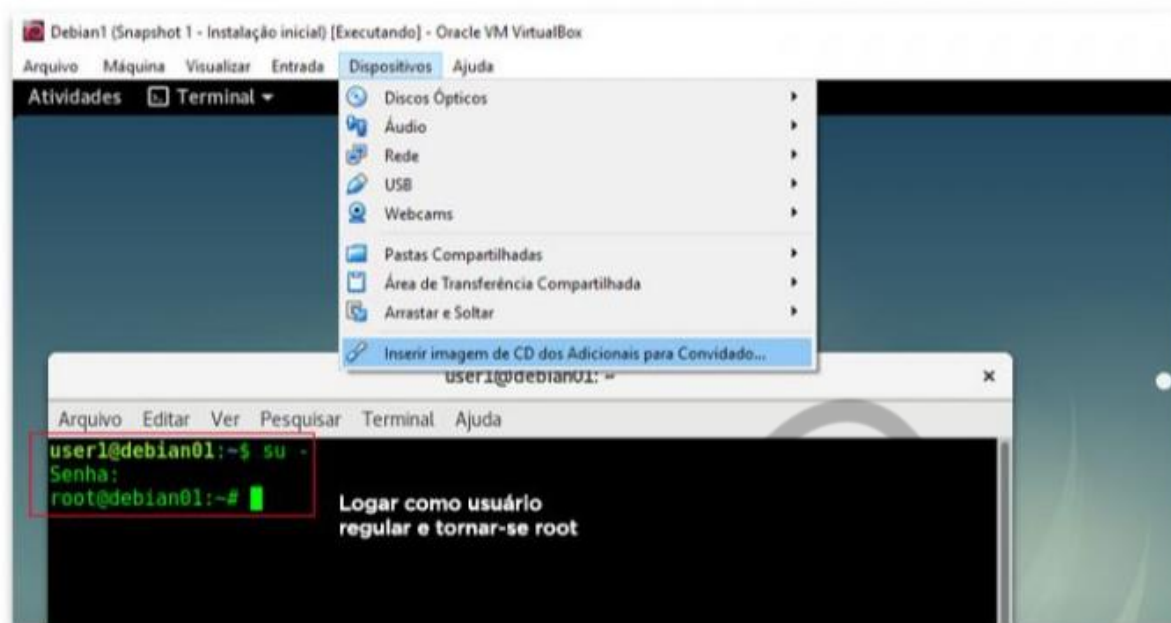


Figura 4.45 – Instalação dos adicionais de convidados
 Fonte: Elaborado pelo autor (2020)

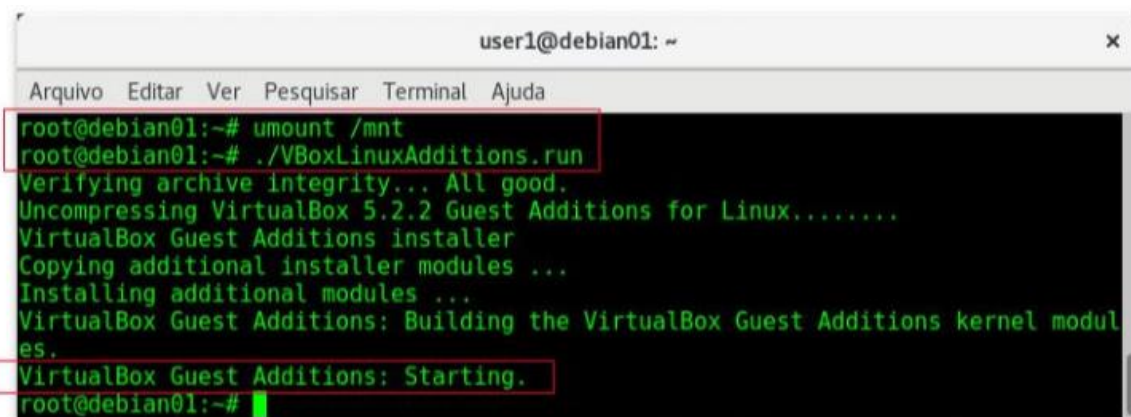
- c. Se uma caixa de mensagem surgir avisando que há um software com início automático, clicar no botão “CANCELAR”.
- d. No terminal (como *root*), executar os comandos especificados (Lista de comandos 1) para a instalação dos pacotes adicionais necessários à instalação do adicional de convidados:

```

root@debian1:~# apt-get update
root@debian1:~# apt-get install -y build-essential
root@debian1:~# apt-get install -y linux-headers-$(uname -r)
root@debian1:~# mount /dev/sr0 /mnt
root@debian1:~# cp /mnt/VBoxLinuxAdditions.run .
root@debian1:~# umount /mnt
root@debian1:~# ./VBoxLinuxAdditions.run
  
```

Lista de comandos 4.1 – Instalação do adicional de convidados
 Fonte: Elaborado pelo autor (2020)

- e. Ao final do processo, o adicional de convidados será automaticamente iniciado.



```
user1@debian01: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
root@debian01:~# umount /mnt  
root@debian01:~# ./VBoxLinuxAdditions.run  
Verifying archive integrity... All good.  
Uncompressing VirtualBox 5.2.2 Guest Additions for Linux.....  
VirtualBox Guest Additions installer  
Copying additional installer modules ...  
Installing additional modules ...  
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel modules.  
VirtualBox Guest Additions: Starting.  
root@debian01:~#
```

Figura 4.46 – Finalização da instalação dos adicionais de convidados
Fonte: Elaborado pelo autor (2020)

4.13.2 Instalação do pacote net-tools e wireshark

O pacote net-tools contém importantes ferramentas para controle do subsistema de rede do *kernel* do Linux, incluindo: *arp*, *ifconfig*, *netstat*, *rarp*, *nameif* e *route*, entre outros. Por sua vez, o Wireshark é um dos analisadores de protocolos mais amplamente utilizados para análise do tráfego em redes. Ainda no terminal, como *root*, executar os comandos especificados na Listagem de comandos 2, observando que, ao ser questionado pelo instalador do Wireshark se usuários desprivilegiados poderão proceder à captura de pacotes, manter a opção-padrão: **<NO>**.

```
root@debian1:~# apt-get install -y net-tools wireshark  
root@debian1:~# which ifconfig  
root@debian1:~# which wireshark
```

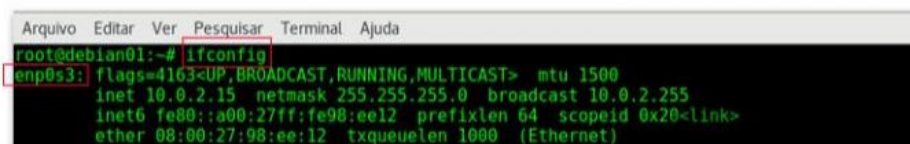
Lista de comandos 4.2 – Instalação do NET-TOOLS e Wireshark
Fonte: Elaborado pelo autor (2020)

Os comandos *which* especificados são apenas para verificação da correta instalação dos pacotes e retornarão, como resposta, o caminho absoluto dos respectivos binários.

4.13.3 Desabilitando o suporte ao IPv6

Não sendo inicialmente necessário, o suporte ao IPv6 será desabilitado. Para tal, ainda no terminal, como *root*:

- a. Identificar o nome da interface.



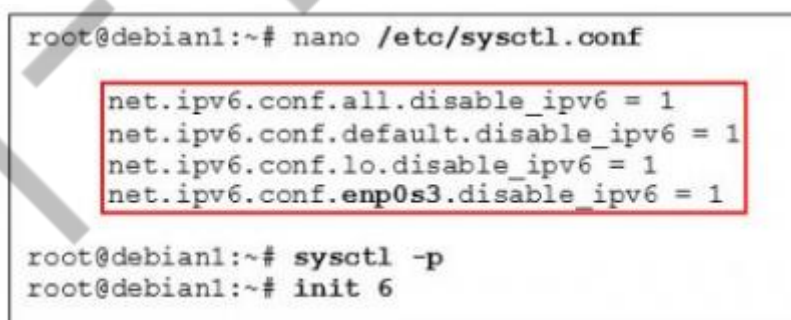
```

Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
root@debian01:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe98:ee12  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:98:ee:12  txqueuelen 1000  (Ethernet)
  
```

Figura 4.47 – Nome da interface
Fonte: Elaborado pelo autor (2020)

Da Figura “Nome da interface”, tem-se como resposta ao comando *ifconfig*:

- **Nome da interface:** **enp0s3**
 - Endereço IPv4: 10.0.2.15
 - Endereço IPv6: fe80: : a00: 27ff: fe98: ee12
 - MAC Address: 08:00:27:98:ee:12
- b. Identificada a interface, as instruções em destaque na Listagem de comandos 3 deverão ser acrescentadas ao **final do arquivo /etc/sysctl.conf**.



```

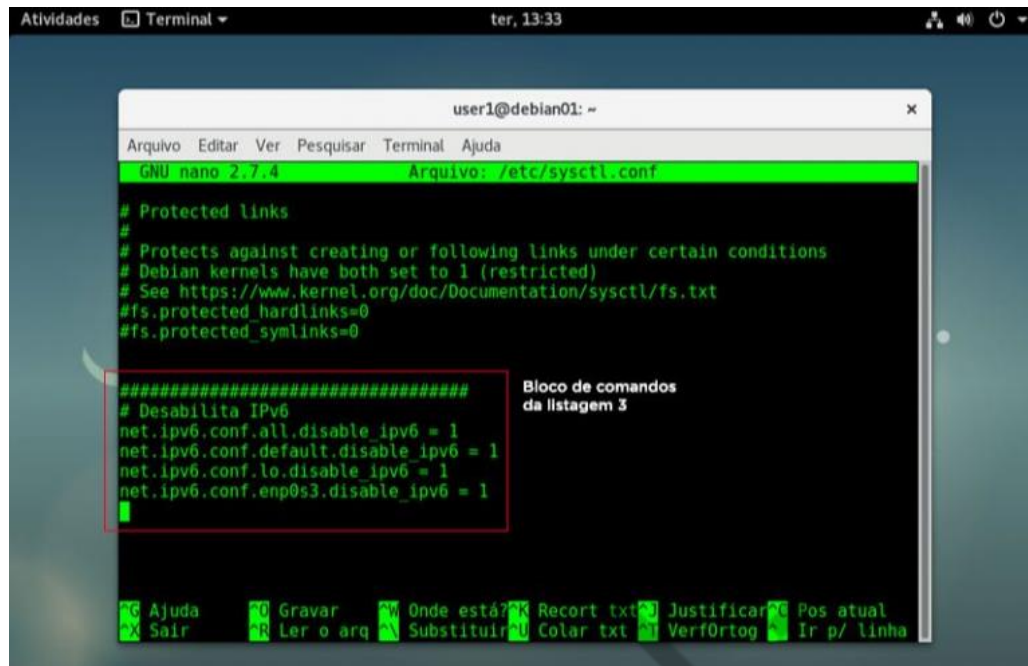
root@debian1:~# nano /etc/sysctl.conf

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
net.ipv6.conf.enp0s3.disable_ipv6 = 1

root@debian1:~# sysctl -p
root@debian1:~# init 6
  
```

Lista de comandos 4.3 – Desativação do IPv6
Fonte: Elaborado pelo autor (2020)

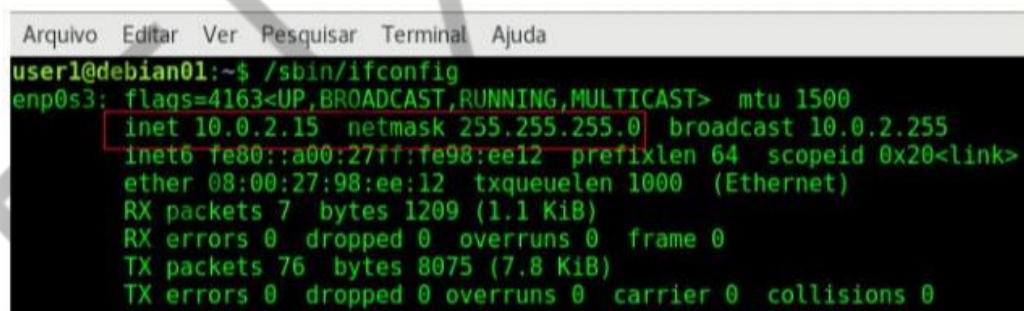
A Figura “Ajustes no arquivo sysctl.conf” ilustra o final do arquivo após finalizada a edição.



```
user1@debian01: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
GNU nano 2.7.4 Arquivo: /etc/sysctl.conf  
  
# Protected links  
#  
# Protects against creating or following links under certain conditions  
# Debian kernels have both set to 1 (restricted)  
# See https://www.kernel.org/doc/Documentation/sysctl/fs.txt  
#fs.protected_hardlinks=0  
#fs.protected_symlinks=0  
  
#####  
# Desabilita IPv6  
net.ipv6.conf.all.disable_ipv6 = 1  
net.ipv6.conf.default.disable_ipv6 = 1  
net.ipv6.conf.lo.disable_ipv6 = 1  
net.ipv6.conf.enp0s3.disable_ipv6 = 1  
[  
  
Bloco de comandos da listagem 3  
  
Ajuda Gravar Onde está? Recort txt Justificar Pos atual  
Sair Ler o arq Substituir Colar txt VerfOrtog Ir p/ linha
```

Figura 4.48 – Ajustes no arquivo sysctl.conf
Fonte: Elaborado pelo autor (2020)

- c. Após reiniciar a VM, abrir o terminal e digitar o comando `/sbin/ifconfig` e verificar se somente um endereço IPv4 é atribuído ao *host*.



```
user1@debian01:~$ /sbin/ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
inet6 fe80::a00:27ff:fe98:ee12 prefixlen 64 scopeid 0x20<link>  
ether 08:00:27:98:ee:12 txqueuelen 1000 (Ethernet)  
RX packets 7 bytes 1209 (1.1 KiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 76 bytes 8075 (7.8 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 4.49 – Verificação do IPv4
Fonte: Elaborado pelo autor (2020)

REFERÊNCIAS

BLANK, A.G. **TCP/IP Foundations**. Alameda (CA): Sybex Inc., 2004.

FOROUZAN, B.A. **Comunicação de dados e redes de computadores**. 4. ed. São Paulo: McGraw-Hill, 2008.

HUNT, C. **TCP/IP Network Administration**. 3. ed. Sebastopol: O'Reilly Media, 2010.

IANA. **Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry**. ago. 2011. Disponível em: <<https://tools.ietf.org/html/rfc6335#page-16>>. Acesso em: 16 out. 2017.

_____. **Service Name and Transport Protocol Port Number Registry**. 16 nov. 2017. Disponível em: <<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>>. Acesso em: 18 nov. 2017.

IPV6. **Endereçamento**. 15 mai. 2012. Disponível em: <<http://ipv6.br/post/enderecamento/>>. Acesso em: 16 out. 2017.

_____. **Cabeçalho**. 15 mai. 2012. Disponível em: <<http://ipv6.br/post/cabecalho/>>. Acesso em: 16 out. 2017.

PETERSON, L. L.; DAVIE, R. S. **Redes de Computadores – Uma abordagem de sistemas**. 3. ed. Rio de Janeiro: Elsevier, 2004.

TCP/IP GUIDE. **IP Datagram General Format**. 2017. Disponível em: <http://www.tcpipguide.com/free/t_IPDatagramGeneralFormat.htm>. Acesso em: 4 out. 2017.

_____. **TCP Message (Segment) Format**. 2017. Disponível em: <http://www.tcpipguide.com/free/t_TCPIPMessagesegmentFormat-3.htm>. Acesso em: 4 out. 2017.

_____. **TCP/IP Address Resolution Protocol (ARP)**. 2017. Disponível em: <http://www.tcpipguide.com/free/t_TCPIPAddressResolutionProtocolARP.htm>. Acesso em: 6 out. 2017.

GLOSSÁRIO

RFC	Acrônimo de Request for Comments, são documentos técnicos desenvolvidos e mantidos pelo IETF (Internet Engineering Task Force), instituição responsável pelos padrões a serem adotados em toda a Internet.
Overhead	De maneira geral, refere-se a qualquer recurso excessivamente demandado para uma tarefa, como, por exemplo, uso excessivo de processamento, memória ou armazenamento.
Frame	Unidade de dados utilizada pela camada de ENLACE do modelo OSI.
Datagrama IP	Unidade básica de dados do protocolo IP, podendo ser dividida em duas áreas: uma de cabeçalho e outra de dados.
Roteamento	Mecanismo que permite a comunicação entre dispositivos residentes em redes distintas.
Payload	É a parte dos dados transmitidos que é o objetivo fundamental da transmissão, já excluídas as informações como cabeçalhos ou metadados.
Processo	É um aplicativo em tempo de execução no <i>host</i> .

Segmento	Unidade básica para transferência de dados utilizada pelas entidades da camada de trans-
-----------------	--

	porte.
Handshake	Sequência de negociação entre <i>hosts</i> distintos a fim de estabelecer uma conexão.
Path MTU Discovery	Técnica padronizada em redes de computadores para determinar o tamanho da Unidade de Transmissão Máxima (MTU) no caminho de rede entre dois <i>hosts</i> IP, geralmente com objetivo de evitar a fragmentação do datagrama IP. O PMTUD foi originalmente destinado a roteadores IPv4, no entanto, todos os sistemas operacionais modernos usam isso em <i>endpoints</i> . No IPv6, esta função foi explicitamente delegada aos <i>endpoints</i> de uma sessão de comunicação.