

**INSTALAÇÃO E CONFIGURAÇÃO DE SERVIDORES**  
**TÉCNICO INTEGRADO EM INFORMÁTICA PARA INTERNET**  
**PROFESSOR IURI SOUZA**

**ROTEIRO DE AULA PRÁTICA 01**

**INSTALAÇÃO E CONFIGURAÇÃO DO APACHE**

**SUMÁRIO**

<b>INSTRUÇÕES PARA A ATIVIDADE [leia com atenção]:</b>	<b>2</b>
<b>1. PRÉ-REQUISITOS</b>	<b>3</b>
1.1 SISTEMA	3
1.2 TESTE DE REDE	3
1.2.1 Adaptadores de rede do virtualbox	3
<b>2 . APACHE</b>	<b>5</b>
2.1 Atualização do Sistema	5
2.2 Instalação do Apache	5
2.3 Verificação da Instalação	5
2.3.1 Ambientação configuração	6
2.3.2 Ambientação conteúdo	6
2.4 Criando um Diretório para o Seu Site	6
2.4.1 Criando um Arquivo Index	6
2.4.2 Configurando o Virtual Host (arquivo .conf)	7
2.5 Ativando o Virtual Host	7
2.5.1 VirtualHost	7
2.5.2 Reiniciando o Apache	7
2.5.3 Acessando Seu Site	8
<b>3. Atividade Proposta</b>	<b>8</b>
<b>4. Considerações Adicionais</b>	<b>9</b>
4.1 Observações:	9
<b>5. Considerações finais</b>	<b>9</b>
<b>6 . ANEXO: Criando um repositório clone do GitHub</b>	<b>10</b>
6.2 Pré-requisitos:	10
6.2.1 Git instalado	10
6.2.2 Uma conta GitHub	10
6.3 Passos	10
6.4 Clone o repositório:	11
6.5 O que acontece depois de clonar:	11

## **INSTRUÇÕES PARA A ATIVIDADE [leia com atenção]:**

- Siga todos os passos descritos em ordem.
- Pode ser feito em dupla (respeitem as duplas da VM)
- Esta é uma atividade de orientação portanto não precisará ser entregue. Porém, sua resolução será alvo de discussão em sala de aula e os conceitos aqui abordados serão cobrados em nossas avaliações.
- A execução da VM é de total responsabilidade da dupla.
- Problemas com a VM devem ser relatados imediatamente ao professor (uso errado de comandos não é problema com a VM)

## **INTRODUÇÃO**

Este tutorial guiará você passo a passo na instalação e configuração básica do servidor web Apache (pacote apache2) no Debian 12. Ao final, você terá um ambiente pronto para hospedar seus primeiros sites.

Para que esse tutorial alcance seu objetivo, assume-se que os alunos já realizaram os estudos dirigidos dos capítulos 1, 2 e 3 da Red Hat Academy e já possuem a familiaridade básica com o prompt de comando e os comandos no shell, seja em modo gráfico ou texto.

Ao longo deste documento algumas perguntas serão realizadas como forma de conduzir a análise e percepção dos alunos. O intuito não é decorar o procedimento mas sim, entender sua lógica. Estas perguntas devem ser alvo de discussão entre a dupla e não precisam ser entregues ao professor.

**OBSERVAÇÃO:** Nas instruções de comandos no shell Bash, sempre que o símbolo **\$** aparecer, antecedendo o comando, isso apenas indicará que a execução deverá ser realizada em modo **usuário comum**. Dessa forma, o caractere **\$** não deve ser inserido no terminal.

## 1. PRÉ-REQUISITOS

### 1.1 SISTEMA

- Um servidor (no caso será a nossa VM) com o sistema operacional Debian 12 instalado.
- Um usuário com privilégios de sudo.

### 1.2 TESTE DE REDE

Vamos realizar um teste de conectividade de rede entre sua máquina virtual (VM) e sua máquina física. Para este teste iremos utilizar o comando **ping**. Sintaxe do comando ping:

```
$ ping <host de destino>
```

onde o **<host de destino>** pode ser um endereço IP ou um nome/domínio que define o host.

- a) Verifique o IP da sua máquina física (via cmd) e em seguida o da sua VM (via terminal). Apenas analisando os IP é possível afirmar se as máquinas irão se comunicar? Justifique.

#### **verificar IP no Debian**

```
$ ip address
```

#### **verificar IP no Windows**

```
ipconfig
```

- b) Realize o teste de ping e analise seu resultado. Ele confirma a sua resposta do item a?

### 1.2.1 Adaptadores de rede do virtualbox.

Acesse as configurações de rede da sua VM no virtualbox e selecione a opção 'Rede', como mostra a figura abaixo.

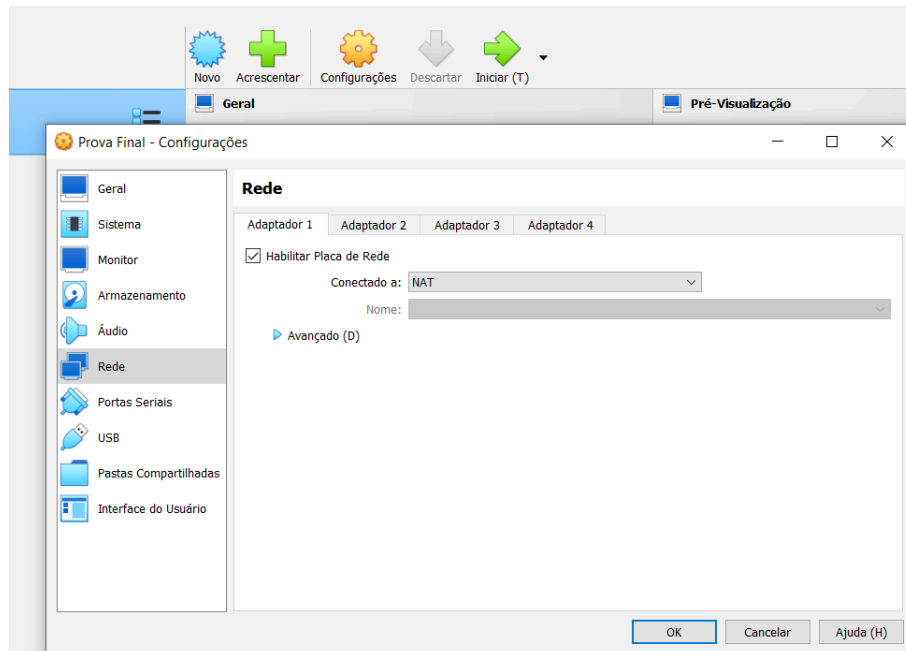


Fig 1 . Configuração de Rede do VirtualBox.

Mude a opção conectado a **NAT** para conectado a **Placa em modo Bridge**

**Obs.: Uma vez conectado em modo Bridge, pede-se que se conecte à rede do laboratório (wLabrede5).**

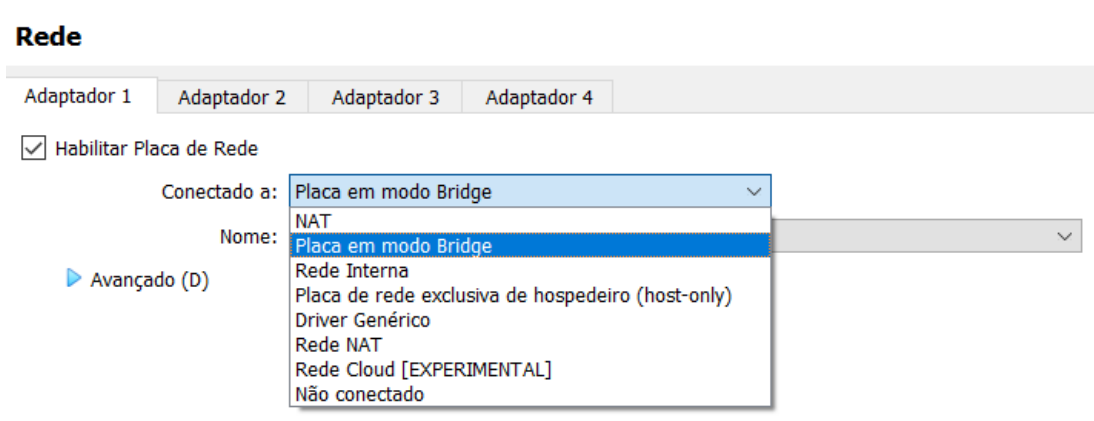


Fig 2. Opção de adaptadores de rede no virtualbox.

**ATENÇÃO:** Após habilitar a placa em modo bridge, certifique-se que na opção Nome estará selecionada a placa de rede em uso na sua máquina física!

Agora refaça o procedimento I e II. Compare o resultado entre os testes de conectividade.

CONCLUSÃO:

## 2 . APACHE

## 2.1 Atualização do Sistema

Antes de iniciar a instalação, é fundamental garantir que seu sistema esteja atualizado:

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

## 2.2 Instalação do Apache

Para instalar o Apache, execute o seguinte comando:

```
$ sudo apt install apache2
```

**Obs.: Durante a instalação, o Apache será iniciado automaticamente.**

## 2.3 Verificação da Instalação

Para verificar se o Apache está funcionando corretamente, abra um navegador e digite o endereço IP do seu servidor. Se tudo estiver configurado corretamente, você verá a página padrão do Apache.

Acesse o diretório de configuração do apache em **/etc/apache2** e verifique todo o seu conteúdo.

### *2.3.1 Ambientação configuração*

Identifique o arquivos de configuração de portas **ports.conf** e os diretórios de hospedagem **sites-available** e **sites-enabled**

- a) Qual(is) porta(s) estão habilitadas? Identifique o padrão de liberação de portas
- b) Qual o conteúdo dos diretórios **sites-available** e **sites-enabled**? Qual o seu padrão de organização?

### *2.3.2 Ambientação conteúdo*

Acesso o diretório de conteúdo em **/var/www**

Procure o arquivo do site que está sendo mantido pelo servidor.

- a) Qual a localização e nome desse arquivo? Existe algum padrão para isso?
- b) Volte ao arquivo de configuração deste site (arquivo **.conf**) e verifique qual parâmetro é responsável por localizar esse arquivo para o servidor apache.

## **2.4 Criando um Diretório para o Seu Site**

Por padrão, o diretório raiz dos documentos do Apache é **/var/www/html**. Para criar um diretório para o seu site, você pode usar o seguinte comando:

```
$ sudo mkdir /var/www/html/meusites
```

### *2.4.1 Criando um Arquivo Index*

Crie um arquivo **index.html** dentro do diretório do seu site com o conteúdo que você deseja exibir:

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

Edite o conteúdo HTML desejado dentro do arquivo e salve.

#### *2.4.2 Configurando o Virtual Host (arquivo .conf)*

Se você deseja hospedar múltiplos sites, você precisará configurar um virtualhost. Crie um arquivo de configuração dentro do diretório **/etc/apache2/sites-available/**:

```
$ sudo nano /etc/apache2/sites-available/meusites.conf
```

Configure seu arquivo com o seguinte conteúdo, adaptando os caminhos e nomes de domínio de acordo com suas necessidades:

```
<VirtualHost *:80>
    ServerName meusites.com
    ServerAlias www.meusites.com
    DocumentRoot /var/www/html/meusites

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

## **2.5 Ativando o Virtual Host**

### *2.5.1 VirtualHost*

Habilite o virtual host que você acabou de criar:

```
$ sudo a2ensite meusites.conf
```

### *2.5.2 Reiniciando o Apache*

Para aplicar as novas configurações, reinicie o serviço Apache:

```
$ sudo systemctl restart apache2
```

### *2.5.3 Acessando Seu Site*

Agora você pode acessar seu site digitando o nome de domínio ou o endereço IP do seu servidor seguido do nome do diretório do seu site no navegador. Por exemplo: [http://seu\\_ip/meusites](http://seu_ip/meusites)

### 3. Atividade Proposta

A dupla deverá hospedar em seu servidor um projeto desenvolvido ao longo do curso. Para isso, deve-se escolher, inicialmente, algum projeto feito sem o uso de nenhum framework<sup>1</sup>, ou seja, um projeto que teve seus arquivos .html, .css e .js desenvolvimento nativamente. Recomenda-se também não fazer uso de banco de dados e arquivos PHP por enquanto.

Para criar um fluxo de trabalho otimizado, recomenda-se seguir o tutorial em anexo *Criando um repositório clone do GitHub*.

### 4. Considerações Adicionais

- Configurações Avançadas: O Apache oferece uma ampla gama de configurações. Consulte a documentação oficial para mais detalhes.
- Segurança: É fundamental manter seu servidor atualizado e aplicar as melhores práticas de segurança.
- PHP: Para executar scripts PHP, você precisará instalar o módulo PHP e configurá-lo corretamente.
- SSL/TLS: Para habilitar o HTTPS, você precisará obter um certificado SSL e configurá-lo no Apache.

#### 4.1 Observações:

- Nome de domínio: Para acessar seu site por um nome de domínio personalizado, você precisará configurar os registros DNS corretamente.
- Virtual Hosts: A configuração de virtual hosts permite que você hospede múltiplos sites em um único servidor.

---

<sup>1</sup> O uso de frameworks será evitado agora apenas por questões didáticas para simplificar o processo de configuração do servidor.



## 5. Considerações finais

Com este tutorial, você deu o primeiro passo para configurar seu próprio servidor web Apache no Debian 12. Para obter mais informações e aprofundar seus conhecimentos, consulte a documentação oficial do Apache.

## 6 . ANEXO: Criando um repositório clone do GitHub

### 6.1 O que é um clone?

Clonar um repositório significa criar uma cópia exata de um projeto que está hospedado no GitHub em seu computador local. Isso permite que você faça alterações no código, experimente novas funcionalidades e colabore com outros desenvolvedores. Em nosso caso, o repositório clonado criará um caminho para importação dos arquivos que irão compor os sites hospedados.

### 6.2 Pré-requisitos:

#### *6.2.1 Git instalado*

O **git** é a ferramenta essencial para interagir com repositórios. Para verificar se o Git está instalado, execute o seguinte comando no terminal:

```
$ git --version
```

Se o Git não estiver instalado, você pode instalá-lo usando o apt:

```
$ sudo apt install git
```

#### *6.2.2 Uma conta GitHub*

Você precisará ter uma conta no [GitHub](https://github.com) e ter criado o repositório que deseja clonar.

## 6.3 Passos

### 1. Obtenha a URL do repositório:

- Acesse o seu repositório no GitHub.
- Clique no botão "Code" (Código).
- Copie a URL HTTPS do repositório.

### 2. Abra o terminal:

- No seu sistema Debian 12, abra um terminal.

### 3. Navegue até o diretório desejado:

Use o comando **cd** para navegar até o diretório onde você deseja clonar o repositório. Por exemplo:

```
$ cd ~/projetos
```

## 6.4 Clone o repositório:

Execute o comando **git clone** seguido da URL que você copiou no passo 1. Por exemplo:

```
$ git clone https://github.com/seu_usuario/seu_repositorio.git
```

Obs.: Substitua **seu\_usuario** e **seu\_repositorio** pelos valores correspondentes ao seu repositório.

## 6.5 O que acontece depois de clonar:

- Uma nova pasta será criada no diretório especificado, com o mesmo nome do repositório.
- Essa pasta conterá todos os arquivos e o histórico de versões do projeto.
- Você poderá fazer alterações nos arquivos, adicionar novos arquivos e commitar as suas mudanças.

## 6.6. Dicas adicionais:

- **Autenticação:** Se você estiver usando autenticação por SSH, você precisará adicionar sua chave SSH ao GitHub.
- **Problemas comuns:**
  - **Permissões:** Certifique-se de que você tem permissão de escrita no diretório onde está clonando o repositório.
  - **Conexão:** Verifique sua conexão com a internet.
  - **URL incorreta:** Verifique se a URL do repositório está correta.

Após clonar o repositório, você pode começar a trabalhar no seu projeto localmente. Para aprender mais sobre como usar o Git, consulte a [documentação oficial do Git](#).