

Atividade Prática 20: Servidor WEB

Implementação de Hosts Virtuais com Nginx

Tema: Nginx (Servidor Web de Alta Performance) - Configuração de Servidor Blocks (Virtual Hosts)

Objetivo: Instalar e configurar o Servidor Web **Nginx**, criando **dois Virtual Hosts** (chamados **Server Blocks** no Nginx) para hospedar dois sites no mesmo servidor, validando a alta performance e a diferença de estrutura em relação ao Apache.

Materiais por Dupla

- 2 Computadores com **Ubuntu 24.04.3 LTS** em rede (Servidor e Cliente).
 - IP fixo configurado no Servidor (Ex: 192.168.1.100).
-

Fase 1: Instalação e Controle Básico (PC Servidor)

1. Instalação do Nginx:

Use o APT para instalar o pacote nginx:

```
sudo apt update
```

```
sudo apt install nginx
```

2. Configuração do Firewall (CRÍTICO):

- Verifique os perfis do Nginx no UFW: sudo ufw app list (A saída deve mostrar Nginx Full, etc.).
- **Liberar Porta 80 (HTTP):** Use o perfil Nginx HTTP para liberar a porta 80.

```
sudo ufw allow 'Nginx HTTP'
```

Verificação: Confirme o status do Nginx:

```
sudo systemctl status nginx
```

verifique também a Porta 80

```
sudo ss -tulpn | grep 80
```

Fase 2: Preparação da Estrutura (Document Roots)

1. **Escolha de Nomes:** Usem os mesmos nomes de domínio da atividade Apache

Ex: senac_jf.com e senac_903.com

2. **Criação dos Diretórios:** Crie os diretórios que servirão como raízes para os sites.

```
sudo mkdir -p /var/www/senac_jf.com/html
```

```
sudo mkdir -p /var/www/senac_903.com/html
```

3. Permissões:

Defina a propriedade para o usuário web padrão (www-data)

```
sudo chown -R www-data:www-data /var/www/senac_jf.com/
```

```
sudo chown -R www-data:www-data /var/www/senac_903.com/
```

4. Criação das Páginas de Teste:

Crie um arquivo index.html em cada diretório com mensagens distintas.

Fase 3: Configuração dos Server Blocks (Virtual Hosts)

1. Cópia do Template:

Copie o template de configuração padrão do Nginx

(localizado em /etc/nginx/sites-available/default) para o primeiro site:

```
sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/senac_jf.com
```

2. Edição do Server Block:

Abra e edite o novo arquivo (sudo nano /etc/nginx/sites-available/senac_jf.com).

Ajustes Importantes (Devem ser repetidos para ambos os sites):

- **root:** Mude para o caminho correto (Ex: /var/www/senac_jf.com/html;).
- **server_name:** Adicione o nome de domínio (Ex: server_name senac_jf.com;).

3. Ativação (A Lógica do Link Simbólico):

Ative os hosts criando um link simbólico da pasta [sites-available](#) para a pasta [sites-enabled](#).

Exemplo para o primeiro site:

```
sudo ln -s /etc/nginx/sites-available/senac_jf.com /etc/nginx/sites-enabled/
```

4. Teste de Sintaxe e Reinício:

- Verifique a sintaxe do Nginx ([Importante](#)): sudo nginx -t
- Reinicie o Nginx: sudo systemctl restart nginx

Fase 4: Interoperabilidade e Teste Final

1. Resolução de Nomes (PC Cliente):

No **PC Cliente**, edite o arquivo **hosts** para mapear os nomes para o IP do Servidor (Ex: 192.168.1.100 senac_jf.com).

2. Teste de Validação:

No PC Cliente, acesse no navegador: http://senac_jf.com e http://senac_903.com.

Conclusão e Relatório (Ajustada para o Aluno)

1. **Comprovação do Server Block:** Comprove com *print screens* que o servidor Nginx entregou as **duas páginas diferentes** a partir do mesmo IP, validando o Server Block.
2. **Validação de Serviço e Segurança:** Inclua a saída do comando **sudo nginx -t** (Teste de Sintaxe) e **sudo ufw status** para provar que a configuração está correta e o firewall está ativo.
3. **Reflexão de Arquitetura (Comparativo Servidor Web vs. Servidor de Aplicação):** Em um parágrafo, o aluno deve diferenciar as funções, respondendo às seguintes questões:
 - **Servidor Web (Nginx/Apache):** Qual é o seu papel principal? (Foque na funcionalidade de servir arquivos estáticos, caching e Proxy).
 - **Servidor de Aplicação (Ex: Node.js, Tomcat):** Qual é o seu papel? (Foque em processamento de código dinâmico, lógica de negócios e dados).
 - **Função Estratégica do Nginx:** Justifique como o Nginx é usado para proteger e balancear o tráfego que é, na verdade, processado por um Servidor de Aplicação (Backend).