

# Vários sites num servidor Nginx

O Nginx é um servidor HTTP (Web) gratuito, open-source e com alta performance. O Nginx foi criado em 2005 e tinha como principal objetivo ser um servidor estável, simples de configurar e que necessitasse de poucos recursos ao nível de hardware. Atualmente o Nginx é um dos servidores Web mais usados para publicação de sites na Internet.

## Vários sites num servidor Nginx

Existem diversas configurações que permitem ter vários sites disponíveis num único servidor Web. A forma mais simples é criando Server Blocks (idênticos aos VirtualHosts no Apache). A criação de Server Blocks permite que, numa única máquina, possamos ter vários sites, sendo que cada um pode estar associado a um nome ou a um endereço IP.

Pretende-se que os alunos tenham 3 sites criados: a.ipg.pt b.ipg.pt c.ipg.pt Cada site deverá apresentar informação sobre que site é. Por exemplo, acedendo a "a.ipg.pt" deverá indicar em h1, "a.ipg.pt".

## Questões:

- Indique os ficheiros de log para cada um dos sites
- Faça upload dos ficheiros de configuração de cada site
- Indique qual o comando para ver se a configuração do Nginx tem erros

A entrega deste trabalho pode ser feita no Classroom, mas depois pretende-se que seja feito um pequeno tutorial no Medium.

## Criação dos sites

Para criar os sites, vamos criar um diretório para cada um deles com um **index.html** muito simples, apenas para ver se funcionam. Irei criar os diretórios em **/var/www**. Para criar os diretórios, irei usar o comando **mkdir**.

```
mkdir /var/www/a.ipg.pt
mkdir /var/www/b.ipg.pt
mkdir /var/www/c.ipg.pt
```

Para cada pasta irei criar um **index.html** com o seguinte conteúdo (alterando o título para o nome da respetiva pasta):

```
<!DOCTYPE html>
<html>
  <head>
    <title>a.ipg.pt</title>
    <style>
      body {
        background-color: #f5f5f5;
        font-family: Arial, sans-serif;
```

```
    }
    h1 {
        color: #333;
        text-align: center;
        margin-top: 50px;
    }
</style>
</head>
<body>
    <h1>Bem-vindo ao a.ipg.pt</h1>
</body>
</html>
```

Agora, passo extremamente importante, dar as permissões adequadas às pastas e aos ficheiros.

```
chmod -R 755 /var/www
chmod 644 /var/www/*/index.html
# Este "*" refere-se a todos os diretórios dentro de /var/www
```

## Configuração do Nginx

Agora que já temos os sites criados, vamos configurar o Nginx para que ele saiba onde estão os ficheiros e como servir os mesmos.

Para isso, vamos criar um ficheiro de configuração para indicar os servidores (de Server Block) disponíveis. O ficheiro irá estar na pasta **/etc/nginx/conf.d** e chamar-se-á **javali.conf**. O nginx irá perceber que este ficheiro é de configuração pois o seu ficheiro de configuração base irá chamar todos os ficheiro dentro de **/etc/nginx/conf.d** que acabem em **.conf**. Cada server deve ser criado independentemente dentro do mesmo ficheiro (ou separado).

```
cd /etc/nginx/conf.d
nano javali.conf
```

```
server {
    listen      80;
    server_name a.ipg.pt;
    # server_name b.ipg.pt;
    # server_name c.ipg.pt;

    location / {
        root    /var/www/a.ipg.pt;
        # root    /var/www/b.ipg.pt;
        # root    /var/www/c.ipg.pt;
        index   index.html;
    }
}
```

```
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root /usr/share/nginx/html;
}

access_log /var/log/nginx/a.ipg.pt.access.log;
# access_log /var/log/nginx/b.ipg.pt.access.log;
# access_log /var/log/nginx/c.ipg.pt.access.log
error_log /var/log/nginx/a.ipg.pt.error.log;
# error_log /var/log/nginx/b.ipg.pt.error.log;
# error_log /var/log/nginx/c.ipg.pt.error.log
}
```

## Configurações do Firewall e do "SELinux Policy Management tool"

No nosso caso, o firewall já se encontra aberto para a porta 80. No entanto, caso o firewall não esteja aberto, podemos fazer o seguinte:

```
firewall-cmd --permanent --add-port=80/tcp
firewall-cmd --reload
```

Para o **SELinux Policy Management tool**, vamos permitir que o Nginx possa aceder aos ficheiros que criámos.

```
# Adiciona a política
semanage fcontext -a -t httpd_sys_content_t '/var/www(/.*)?'
# Aplica a política
restorecon -Rv /var/www
```

## Ficheiros de log

Para cada um dos sites, criamos um ficheiro de log separado, tanto para o access\_log como para o error\_log. Decidimos arbitrariamente que os ficheiros de log iriam estar na pasta **/var/log/nginx**, conforme o ficheiro de configurações que criamos **/etc/nginx/conf.d/javali.conf**. Apenas temos que criar os diretórios para os ficheiros de log.

```
mkdir -p /var/log/nginx
```

## Testar a configuração

Para testar, o Nginx tem um comando que permite verificar se a configuração está correta. Para isso, basta executar o comando **nginx -t**. Se tudo estiver correcto, iremos reiniciar o serviço do Nginx.

```
nginx -t  
systemctl restart nginx
```

## Configuração do nome de domínio

Para configurar o nome de domínio, vamos editar o ficheiro **/etc/hosts** e adicionar as seguintes linhas:

```
# o "-e" permite-nos usar o "\t" para tabular  
echo -e "127.0.0.1\ta.ipg.pt b.ipg.pt c.ipg.pt" >> /etc/hosts  
echo -e "192.168.16.204\ta.ipg.pt b.ipg.pt c.ipg.pt" >> /etc/hosts
```

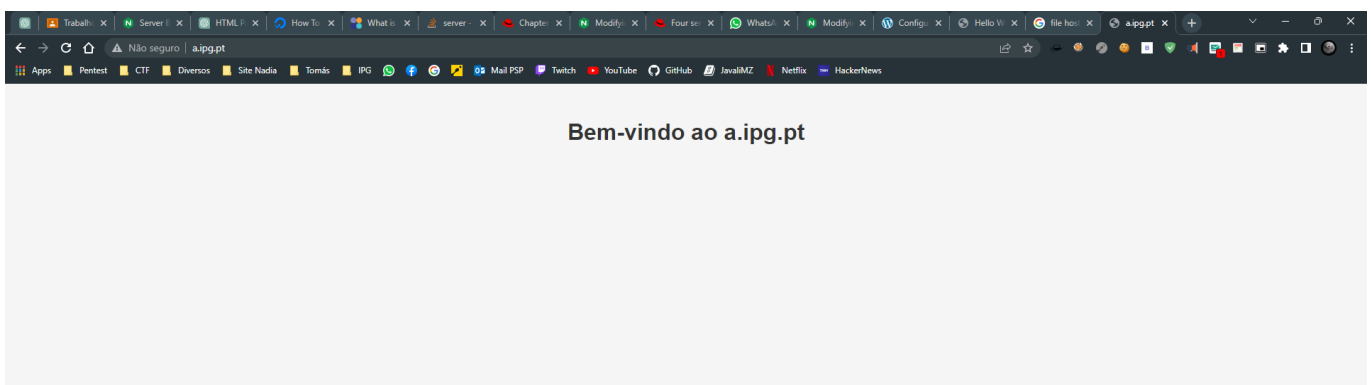
## Verificação do funcionamento

Como estamos num servidor sem interface gráfica, vamos usar o comando **curl** para verificar se os sites estão a funcionar.

```
curl http://a.ipg.pt
```

Por se tratar de uma máquina virtual, para podermos ver o site no nosso browser, podemos ainda alterar o ficheiro hosts do nosso computador (neste caso Windows) e adicionar a seguinte linha:

```
# Ip da máquina virtual  
192.168.56.101 a.ipg.pt b.ipg.pt c.ipg.pt
```



## Documentação e ajudas

[Site oficial do nginx - server\\_blocks](#)

[Site com os comandos das políticas do SELinux](#)

[Outro site referente ao SELinux](#)

Comando semanage