

# Servidor Web - Apache

Prof. Thiago Guimarães Tavares - [thiagogmta@gmail.com](mailto:thiagogmta@gmail.com) IFTO Campus Palmas

## Objetivos do capítulo

Ao final deste capítulo você deverá ser capaz de:

- Entender o que é e para que serve um servidor Web Apache.
- Ser capaz de realizar as configurações iniciais para o funcionamento do servidor como: Iniciar, parar e reiniciar o serviço; Realizar alteração de diretórios e portas de funcionamento do servidor.
- Realizar a configuração de hosts virtuais.

## Apresentação

### O que é?

Um servidor web é responsável pela gestão e armazenamento de diversos sites e serviços disponíveis na internet. Desde pequenos sites a grandes canais de entretenimento, redes sociais e mecanismos de busca, são hospedados em servidores web (MORIMOTO, 2013).

### Para que serve?

Um servidor web conta com uma gama de módulos que vão fornecer as funcionalidades necessárias para o funcionamento dos serviços ali hospedados. A exemplo temos uma aplicação comumente conhecida como LAMP que combina alguns dos módulos essenciais para hospedagem de sites. O LAMP é acrônimo para Linux + Apache + MySQL + PHP. (MORIMOTO, 2013).

### Como funciona?

Quando o usuário digita o endereço de um determinado site o servidor DNS converte esse nome em um endereço IP. Esse endereço é correspondente ao endereço do servidor que hospeda o site. Ao chegar no servidor a requisição é tratada de forma a retornar para o browser do usuário a página desejada. (MORIMOTO, 2013). Nesta sessão sobre servidores web nos atentaremos a respeito da configuração do Apache.

## Apache: Considerações Iniciais

**Instalação** O Apache2 pode ser instalado através do seguinte comando:

```
$ sudo apt-get install apache2
```

No netkit o apache já vem instalado por padrão. Ao instalar o apache2, também é instalado o suporte ao SSL automaticamente.

Iniciar, parar e reiniciar o servidor:

```
$ /etc/init.d/apache2 start  
$ /etc/init.d/apache2 stop  
$ /etc/init.d/apache2 restart
```

## Arquivos de configuração

Todos os arquivos de configuração do servidor Web Apache estão disponíveis em `/etc/apache2`. Segue sua descrição:

- `apache2.conf` - Arquivo de configuração principal do apach
- `mods-available` - Módulos disponíveis no servidor
- `mods-enabled` - Módulos ativos no servidor
- `ports.conf` - Arquivo de configuração para alteração de portas no servidor
- `sites-available` - Diretório que armazena os hosts virtuais no servidor
- `sites-enabled` - Diretório que armazena os hosts virtuais ativos no servidor

## Comandos

Existem alguns comandos que podem ser utilizados para habilitar ou desabilitar um módulo ou um host virtual:

- `a2ensite` - Habilita um host virtual
- `a2dissite` - Desabilita um host virtual
- `a2enmod` - Habilita um módulo específico
- `a2dismod` - Desabilita um módulo específico

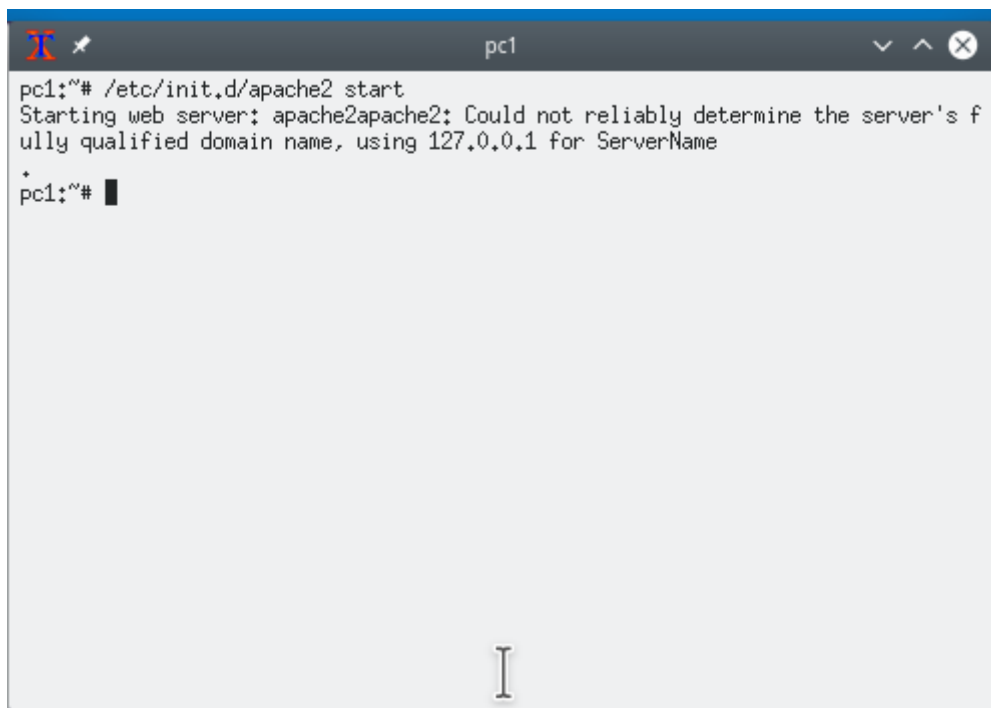
Ao trabalharmos com hosts virtuais esses comandos serão melhor detalhados.

# Minha primeira hospedagem

Por padrão nosso servidor já vem com uma pequena página de testes que pode ser visualizada no navegador assim que o servidor for iniciado. Vamos dar uma olhada. No seu servidor apache inicie o serviço com o seguinte comando:

```
$ /etc/init.d/apache2 start
```

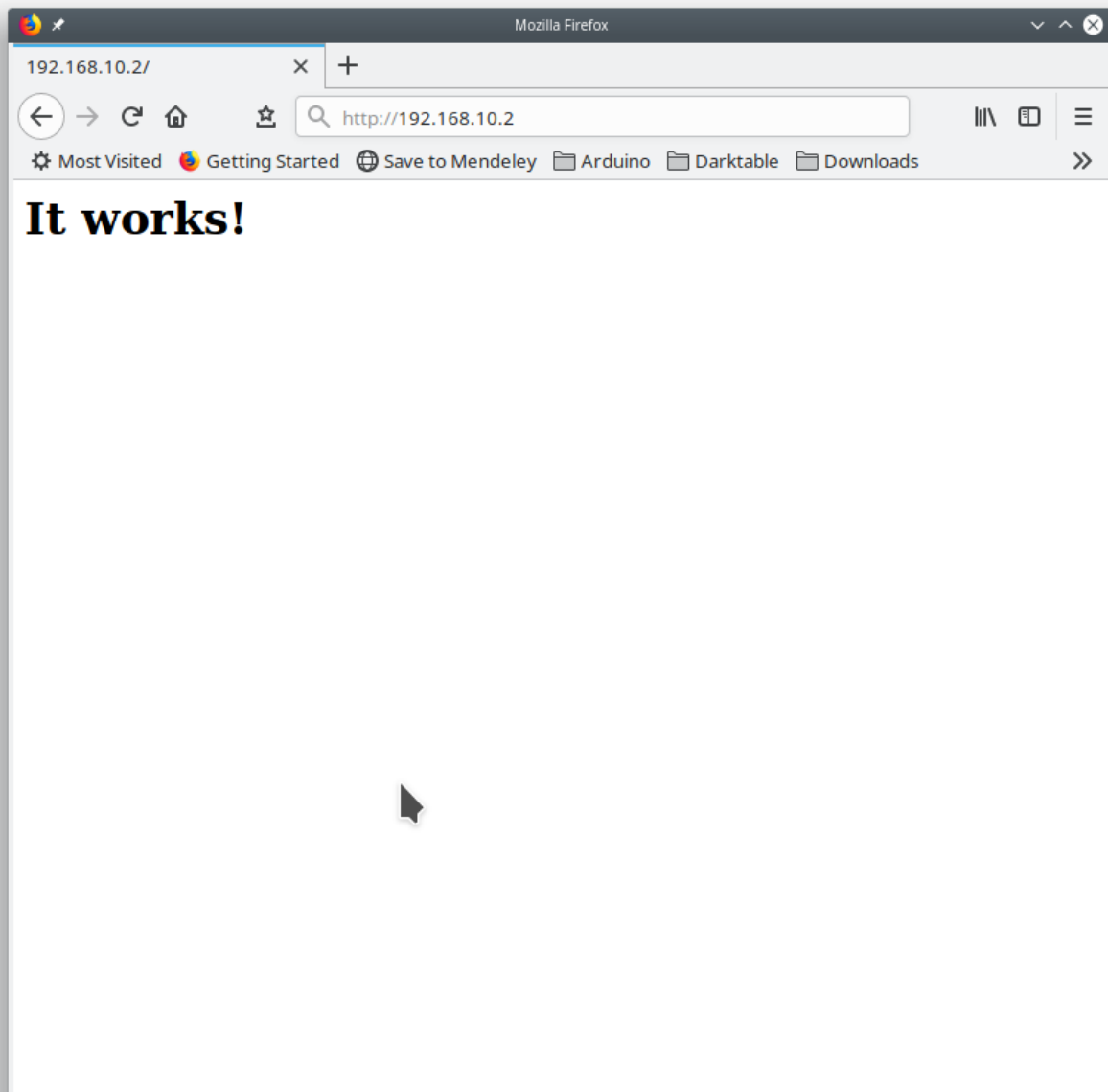
Lembrando que nenhuma configuração foi feita no servidor, apenas inicializado o serviço, você deverá visualizar uma tela semelhante a esta:

A terminal window titled 'pc1' with a blue header bar. The terminal shows the command 'pc1:~# /etc/init.d/apache2 start' being entered. The output is 'Starting web server: apache2' followed by a warning message: 'apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName'. The prompt 'pc1:~#' is shown again with a cursor. A vertical cursor is also visible at the bottom center of the terminal area.

```
pc1:~# /etc/init.d/apache2 start
Starting web server: apache2
apache2: Could not reliably determine the server's fully
qualified domain name, using 127.0.0.1 for ServerName
pc1:~#
```

*Figure 1. apache2 start*

Podemos testar o funcionamento do nosso servidor utilizando qualquer navegador de internet de nosso host anfitrião (Pc Real) bastando apenas inserir no navegador de internet o endereço IP da nossa conexão **tap**.



*Figure 2. It Works!*

O apache apresenta essa tela de testes por padrão.

Faremos uma pequena modificação a nível de teste. Vamos criar um novo usuário em nosso servidor, criar um diretório para armazenar uma página teste dentro do /home desse usuário e apontar nosso servidor Apache para esse diretório. Are you ready?

Para criar um novo usuário utilizamos o seguinte comando:

```
$ adduser pedro
```

Serão solicitadas informações do usuário como nome e telefone. Preencha as que forem pertinentes e confirme com Y ao final. Ao criar um novo usuário é criado também um novo diretório em /home nesse caso: /home/pedro. Dentro desse diretório criaremos uma pasta chamada "www" para armazenar nossos arquivos da nova página teste e um arquivo chamado index.html que será nossa página inicial.

```
$ cd /home/pedro
$ mkdir www
$ cd www
$ touch index.html
```

O index.html receberá o seguinte conteúdo:

```
<html>
  <head>
    <title>Site do Pedro</title>
    <meta charset="utf8">
  </head>
  <body>
    <h1>Bem vindo ao site do Pedro!</h1>
    <hr/>
    <p> Olá seja bem vindo! </p>
  </body>
</html>
```

Pronto! Agora precisamos apontar nosso diretório default do servidor apache para esse novo diretório. Para tanto devemos editar o arquivo **default** em */etc/apache2/sites-available*. Altere as linhas:

```
DocumentRoot e Directory
```

Para ambas as linhas, substituir o caminho padrão (/var/www) pelo novo caminho: /home/pedro/www. Reinicie o servidor e atualize a página do navegador. Você deverá obter um resultado semelhante a este:

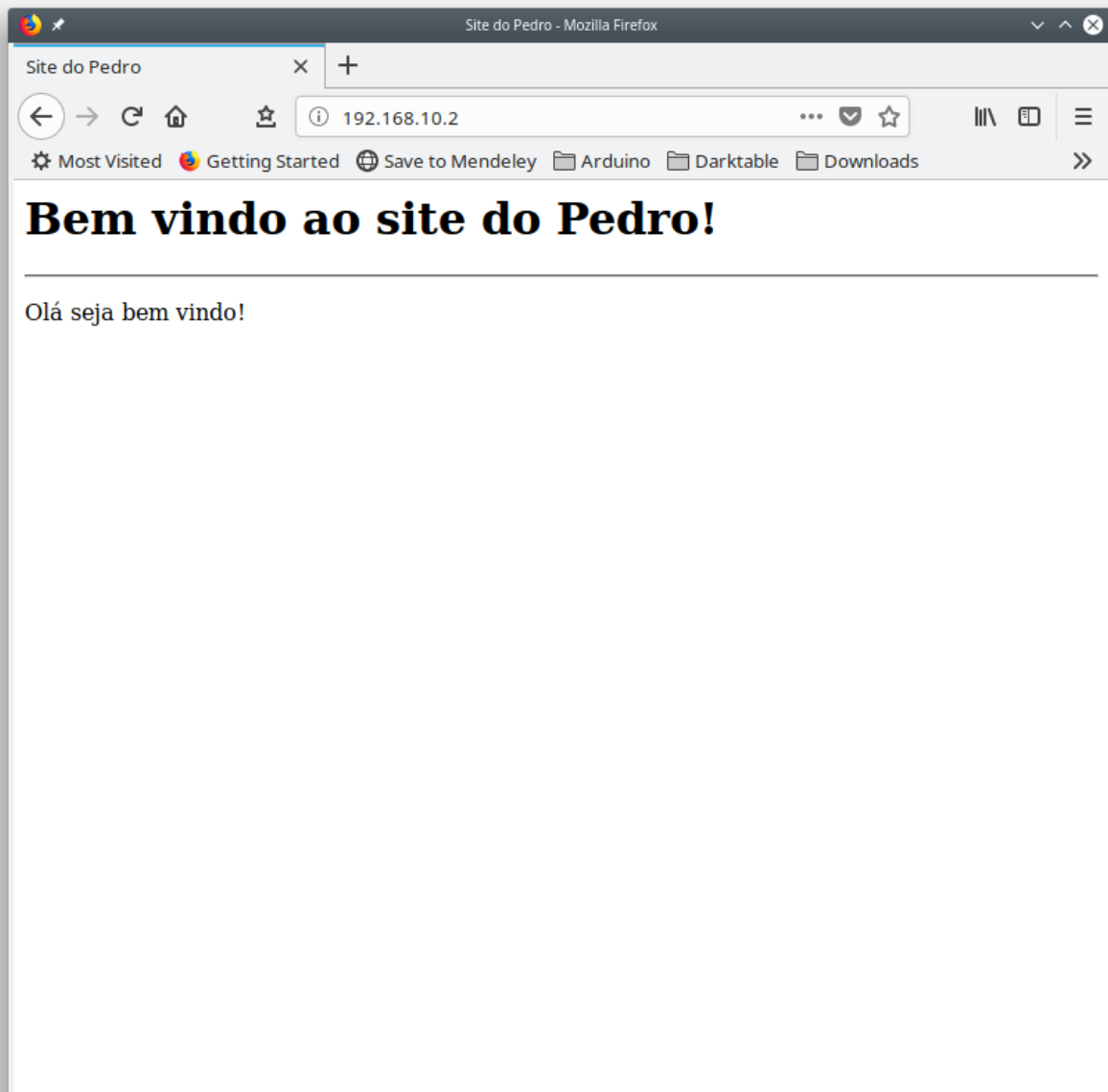


Figure 3. Site do Pedro

### Alterando a porta padrão do Apache

Por padrão o apache funciona através da porta 80. Caso necessário essa configuração pode ser alterada. Para alterar a porta padrão do apache deve-se alterar o arquivo **ports.conf** localizado em */etc/apache2*. Ao acessar esse arquivo altere as linhas:

*O número 80 informa a porta de funcionamento. Caso queira que o apache opere em outra porta basta alterar essas duas linhas.*

```
NameVirtualHost *:80
Listen 80
```

Posteriormente deve-se alterar o arquivo responsável pelo host, no nosso caso o arquivo **default** localizado em */etc/apache2/sites-available*. Nesse arquivo localize a linha

Altere a porta conforme feito no arquivo `ports.conf`

```
<VirtualHost *:80>
```

#### IMPORTANT

Não se esqueça de reiniciar o servidor!

Para verificar as novas configurações acesse pelo navegador da seguinte maneira:

*Lembre-se que 192.168.10.2 é nosso endereço ip entre o computador e o netkit. A informação 8080 diz respeito a nova porta que foi inserida (como exemplo).*

```
http://192.168.10.2:8080
```

## Hosts Virtuais

### Definição:

Nosso servidor web é capaz de suportar vários sites hospedados fazendo distinção entre eles de acordo com as requisições dos usuários. A medida que o usuário envia um pacote de solicitação de acesso à um determinado site hospedado no servidor, este verifica em sua base se tem alguma hospedagem com aquele domínio. Em caso positivo o servidor irá responder a solicitação apontando para os arquivos do site correspondente à solicitação. Para que isso funcione é necessário que cada domínio tenha seu arquivo de configuração de forma a apontar para os arquivos daquele site. A imagem a baixo representa o funcionamento.

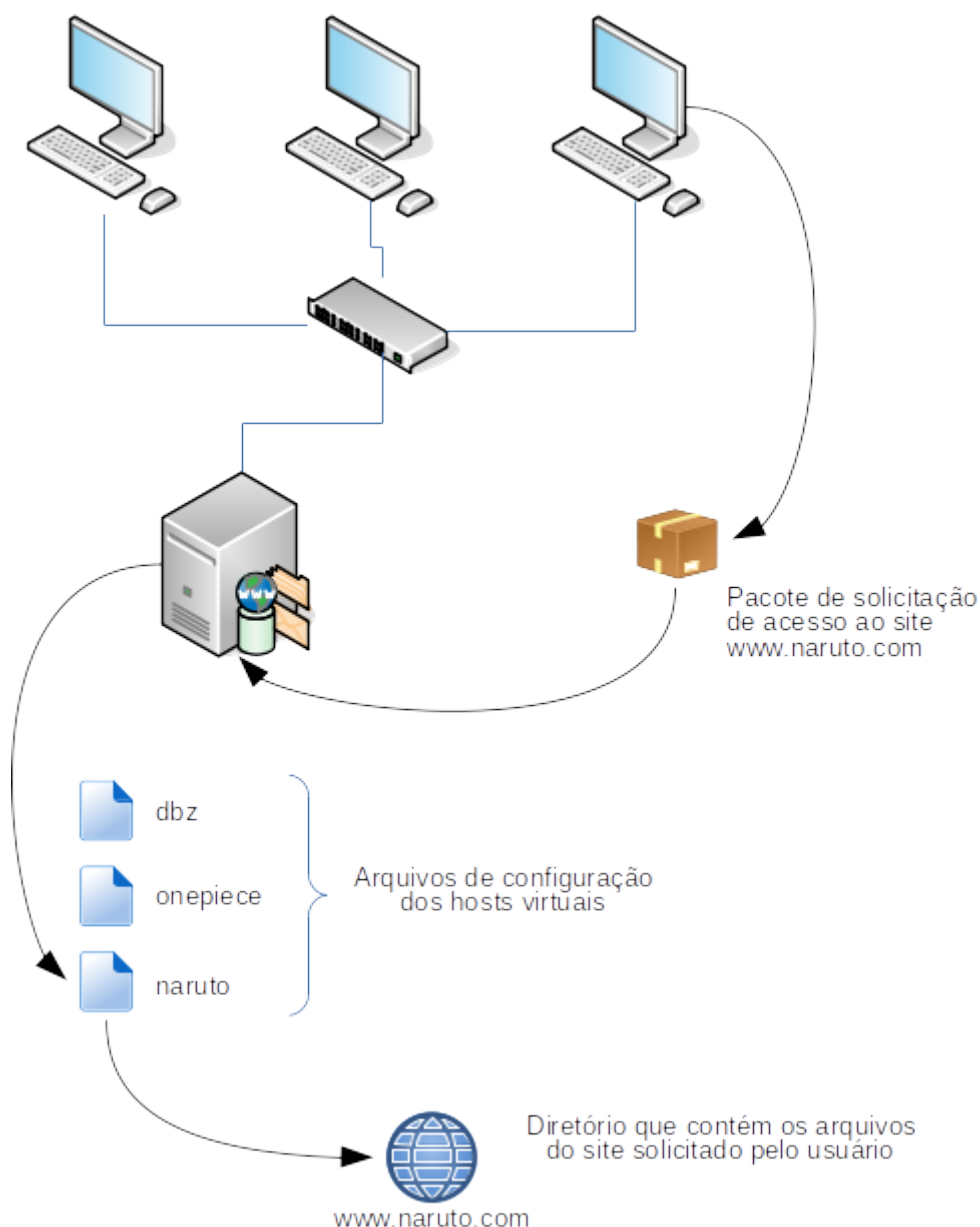


Figure 4. Representação do funcionamento dos hosts virtuais no apache

### Criando estrutura

Vamos levar em consideração que nosso servidor irá hospedar os seguintes três sites:

- [www.dbz.com](http://www.dbz.com)
- [www.naruto.com](http://www.naruto.com)
- [www.onepiece.com](http://www.onepiece.com)

Para melhor organização vamos criar um usuário para cada host virtual e armazenar o conteúdo de cada site dentro da pasta do usuário. Imagine que cada usuário poderá posteriormente ter acesso ao seu diretório com seu usuário e senha para fazer upload ou alterar os arquivos do seu próprio site. Os passos a seguir foram executados no servidor web apache com o usuário root.



*Criando usuários (preencha as informações pertinentes a cada usuário e confirme com y).*

```
$ adduser dbz
$ adduser naruto
$ adduser onepiece
```

Usuários criados, vamos criar o diretório `www` dentro do `/home` de cada usuário, criar um arquivo `index.html` para cada um e alterar as configurações de propriedade dos arquivos:

*Criando estrutura de diretórios e arquivos `index.html`*

```
$ mkdir /home/dbz/www
$ mkdir /home/naruto/www
$ mkdir /home/onepiece/www

$ touch /home/dbz/www/index.html
$ touch /home/naruto/www/index.html
$ touch /home/onepiece/www/index.html
```

Vamos agora alterar as configurações de propriedade dos arquivos.

#### NOTE

Como os diretórios e arquivos foram criados com o usuário `root`. O usuário `root` é dessa forma proprietário dos diretórios e arquivos. Vamos alterar as configurações de propriedade dos arquivos para que cada usuário tenha permissão de escrita em seu próprio diretório `www`.

*Alterando propriedades*

```
$ chown -R dbz:dbz /home/dbz/www
$ chown -R naruto:naruto /home/naruto/www
$ chown -R onepiece:onepiece /home/onepiece/www
```

Estrutura pronta, vamos apenas alterar o conteúdo do arquivo `index.html` de cada site:

*`index.html`: DBZ*

```
<html>
  <head>
    <title>Dragon Ball Z</title>
    <meta charset="utf8">
  </head>
  <body>
    <h1>Posso pressentir o perigo e o caos...</h1>
    <hr/>
    <p> "Você não é derrotado quando perde, mais sim quando você desiste."
(VEGETA). </p>
  </body>
</html>
```

*index.html: NARUTO*

```
<html>
  <head>
    <title>Naruto</title>
    <meta charset="utf8">
  </head>
  <body>
    <h1>Vila Oculta da Folha</h1>
    <hr/>
    <p> "Se você não gosta do seu destino, não aceite. Em vez disso, tenha a
coragem de o mudar do jeito que você quer que seja.." (UZUMAKI NARUTO). </p>
  </body>
</html>
```

*index.html: ONEPIECE*

```
<html>
  <head>
    <title>Naruto</title>
    <meta charset="utf8">
  </head>
  <body>
    <h1>Akuma No Mi</h1>
    <hr/>
    <p> "Não importa o que o mundo diz de mim, o que importa é que eu nunca fiz
nada que contrariasse os meus princípios e nunca farei." (RORONOA ZORO). </p>
  </body>
</html>
```

## Criando arquivos de configuração dos Hosts Virtuais

Os arquivos de virtual host devem ficar armazenados no diretório */etc/apache2/sites-available* e especifica como o servidor apache irá responder às solicitações daquele domínio em específico. Nesse diretório encontra-se o arquivo **default** que pode ser utilizado de base para os demais arquivos de hosts virtuais ou podemos criar um arquivo em branco e inserir as configurações mínimas necessárias. Essa segunda opção é o que faremos.

*Criando arquivos dos Hosts Virtuais*

```
$ cd /etc/apache2/sites-available
$ touch dbz
$ touch naruto
$ touch onepiece
```

Faremos a configuração do Virtual Host **dbz**. Para os demais Hosts Virtuais, aproveite esta mesma configuração alterando apenas o necessário:

## Configuração do host dbz

```
<VirtualHost *:80>
    ServerAdmin admin@dbz.com ①
    ServerName dbz.com ②
    ServerAlias www.dbz.com ③
    DocumentRoot /home/dbz/www ④
    ErrorLog /etc/apache2/error.log ⑤
    CustomLog /etc/apache2/access.log combined ⑥
</VirtualHost>
```

- ① E-mail do administrador do site.
- ② Domínio base do virtual host.
- ③ Outros nomes que podem corresponder ao Virtual Host.
- ④ Localização dos arquivos do domínio em questão
- ⑤ Caminho para o arquivo de log de erro
- ⑥ Caminho para o arquivo de log de requisições ao servidor

### IMPORTANT


Para os demais Hosts Virtuais a estrutura é a mesma.

Nosso host virtual foi criado entretanto ainda não está ativo. Para ativar nosso host em questão basta utilizar o comando a seguir:

### Ativando o site dbz

```
$ a2ensite dbz
```

Deve retrornar uma mensagem parecida com esta:

A terminal window titled 'pc1' with a dark header bar. The terminal shows the command 'a2ensite dbz' being executed in the directory 'pc1:/etc/apache2/sites-available#'. The output indicates that the site 'dbz' is being enabled and provides instructions to reload the Apache2 service. The prompt returns to 'pc1:/etc/apache2/sites-available#' with a cursor.

```
pc1:/etc/apache2/sites-available# a2ensite dbz
Enabling site dbz.
Run '/etc/init.d/apache2 reload' to activate new configuration!
pc1:/etc/apache2/sites-available#
```

Figure 5. *a2ensite*

A mensagem informa que o servidor deve ser atualizado. Note que o arquivo de configuração do novo virtual host também irá aparecer agora no diretório */etc/apache2/sites-enabled*. Isso significa que o site está ativo.

Reinicie o servidor!

## Testando os hosts virtuais

### Configuração dos clientes

Como não dispomos de um servidor DNS para fazer a conversão de nomes (até este ponto, pois isso será feito no próximo capítulo) iremos modificar temporariamente o arquivo **hosts** para realizar testes locais. Esse passo deve ser feito nos computadores clientes (pc2, pc3 e pc4). Para tanto acesse o arquivo */etc/hosts* e adicione as seguintes linhas:

*/etc/hosts*

```
10.1.1.1 www.dbz.com
10.1.1.1 www.naruto.com
10.1.1.1 www.onepiece.com
```

### Teste

Nossos testes de hosts virtuais serão executados apenas em nosso ambiente de testes Netkit através de nossos clientes. Como o netkit não trabalha com interface gráfica iremos utilizar um navegador de internet em modo texto nos clientes para validar nosso resultado. O netkit já possui o **Links** que

é um navegador de internet em modo texto. O Links já oferece funcionalidade suficiente para testarmos nosso ambiente. Basta apenas, no terminal do pc2 por exemplo, executar o comando:

Acesso ao site dbz pelo Pc2

```
$ links www.dbz.com
```

A imagem a seguir ilustra o acesso aos três sites criados através dos pcs 2, 3 e 4 utilizando o comando **links**.

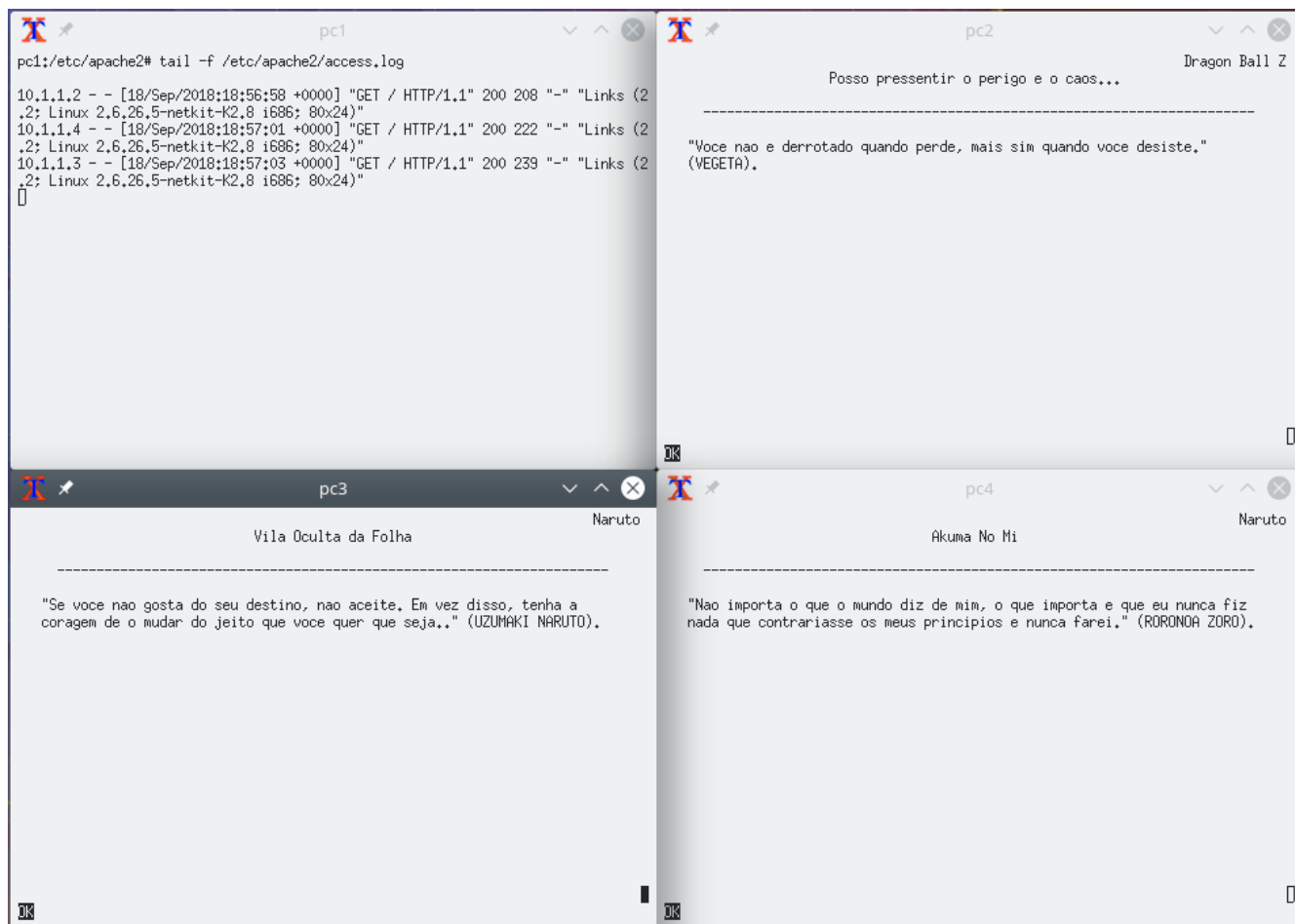


Figure 6. Teste de acesso aos Hosts Virtuais

#### NOTE

No terminal do pc1 foi executado o comando: `tail -f /etc/apache2/access.log`. Esse comando analisa as ultimas entradas no arquivo `access.log` que por sua vez acusa acessos ao nosso servidor. Observando atentamente podemos perceber que o nosso servidor recebeu três solicitações provenientes dos endereços: 10.1.1.2, 10.1.1.3 e 10.1.1.4 pertencentes aos pcs 2, 3 e 4.

## Problematização

Seguem dois problemas para testar o que foi absorvido até o momento. O primeiro problema aborda o tópico inicial sobre o servidor web apache, capítulo *Minha Primeira Hospedagem*. O segundo problema aborda o conteúdo apresentado no capítulo *Hosts Virtuais*.

### 1. Minha Primeira Hospedagem

- a. Crie um usuário com seu nome no servidor
- b. Dentro do diretório do seu usuário crie o diretório www
- c. Crie um arquivo index.html com o seguinte conteúdo:

```
h1 - Seu nome
h2 - Nome da Disciplina
h2 - Nome do Professor
p - Um parágrafo sobre o que acha do curso
p - Um parágrafo sobre seu desenvolvimento na disciplina
p - Um parágrafo sobre o que pretende fazer ao final do curso
* Uma imagem qualquer
* Um link para o site do IFTO
h3 - Data
```

## 2. Hosts Virtuais

- a. Crie os três usuários dos hosts virtuais apresentados neste material.
- b. Crie a estrutura de diretórios e os arquivos html de cada host.
- c. Crie as entradas no servidor apache para cada host.
- d. Faça o teste de acesso a cada um dos sites, monitorando o arquivo access.log conforme apresentado no exemplo.