

## Começando com o Vim

Existem diversos editores de texto para terminais, cada um com suas singularidades e modos de utilização.

No mundo Linux, um bem conhecido é o Vim, um editor de texto simples e poderoso. Mas como podemos usá-lo para editar nossos arquivos?

O Vim é um editor de texto que funciona via terminal, isto é, sem interface gráfica como o bloco de notas, nem conta com o auxílio do mouse. E, como praticamente tudo no terminal, ele funciona por meio de comandos.

Como queremos configurar um serviço de DNS, ou seja, queremos configurar o serviço que traduz os nomes das URIs em endereços IPs. Logo, vamos abrir o arquivo `named.conf` que contém as informações sobre nossos domínios.

Esse arquivo fica dentro da pasta `etc`. Para abri-lo com o vim podemos entrar na pasta e depois abrir o arquivo, ou então, simplesmente passar o caminho até ele:

```
vim /etc/named.conf
```

```
[root@servidor01 ~]# vim /etc/named.conf
```

Caso o arquivo não exista, ele é criado pelo Vim. Nós podemos também apenas digitar o comando `vim` que é aberto um editor em branco para a gente. Neste caso, quando formos salvar o arquivo precisamos dizer qual o seu nome.

## Os modos do Vim

### Modo normal

O Vim tem alguns modos. Quando abrimos um arquivo, estamos no modo normal. Neste modo podemos passar por todo o texto utilizando as setas do teclado.

Se, por exemplo, eu quero apagar o endereço IP da linha `listen-on` e colocar outro endereço.

```
options {  
    listen-on port 53 { 127.0.0.1; };
```

Podemos ir até o local desejado e apagar os caracteres desejados pressionando a tecla `x`.

```
options {  
    listen-on port 53 { ; };
```

Mas, agora queremos inserir outro endereço, como podemos fazer isso?

### Modo de inserção

Bem, se queremos inserir algo em um arquivo, basta pressionar a tecla `i` e digitar o novo texto:

```
options {  
    listen-on port 53 { 192.168.0.0; };
```

Quando apertamos a tecla `i`, estamos entrando no modo de inserção. Conseguimos saber o modo que estamos olhando no rodapé da tela:

## -- INSERÇÃO --

Neste modo conseguimos inserir e apagar textos com a tecla `del` e com `backspace`. Quebrar para outras linhas com o `Enter`, como em um editor de textos normal, ou até mesmo percorrer todo o texto. Se quiser voltar para o modo normal, basta apertar `Esc`.

### Dando mais um passo com o editor

Bem, além dessa configuração, eu preciso falar para meu [serviço de DNS](#) qual o nome da zona e qual o arquivo que ele utilizará.

Ele utiliza esta zona para poder traduzir os nomes no nosso domínio em endereços IPs. No final do arquivo temos uma configuração padrão utilizada pelo serviço. Podemos usar de exemplo essa configuração para fazer a nossa.

```
zone "." IN {  
    type hint;  
    file "named.ca";  
};
```

Como vamos usar essa parte do código para fazer nossa configuração, o jeito mais fácil é copiar o trecho e realizar as alterações. Só que, como podemos copiar alguma coisa com o Vim?

### Copiando e colando no Vim

No Vim podemos utilizar a tecla `y` para copiar uma linha e para colar podemos utilizar a tecla `p` (put).

Então vamos ter que copiar linha por linha?

Não necessariamente. Apesar do Vim ser um editor de textos pelo terminal, existe nele um modo chamado visual. Nesse modo conseguimos selecionar uma área do texto para copiá-lo, recortá-lo ou excluí-lo. Para acessar o modo visual é basta apertarmos a tecla `v` (visual).

```
zone "." IN {  
    type hint;  
    file "named.ca";  
};
```

Selecionamos a área desejada, agora basta utilizar as teclas `y` para copiar essa área e podemos utilizar o `p` para colar.

Basta alterar algumas coisas e pronto! :) Podemos salvar e sair do arquivo. Mas como eu consigo sair desse arquivo? Não podemos dar um `Ctrl + s` para salvar o arquivo, nem um `Alt + F4` para sair.

Bem, queremos salvar esse arquivo. Ou seja, queremos escrever as mudanças que fizemos. Para isso, no modo normal, podemos utilizar o comando `:w` (write) que escreve (salva) as mudanças no arquivo. Já para sair do arquivo podemos utilizar o comando `:q` (quit).

## Para saber mais

Além do Vim, existem outros editores de texto para terminal como o Nano, Emacs, entre muitos outros.

O Vim é case-sensitive, ou seja, ele diferencia as letras maiúsculas de minúsculas. Por isso, ao digitarmos nossos comandos, é importante sempre notar se ele é feito com letras maiúsculas ou minúsculas.

O comando `:w` salva o arquivo, já o comando `:q` sai do arquivo. Logo, se desejamos salvar e sair do arquivo podemos unir esses dois comandos e digitar `:wq`, ou simplesmente `:x`. Se quiser saber mais sobre o poder do Vim, você pode digitar o comando `:help` no modo normal do editor, ou ainda no terminal utilizar o comando `vimtutor`:

```
yuri@yuri-laptop:~$ vimtutor
```

O `vimtutor` nada mais é do que um tutorial sobre o vim. Que nos mostra como utilizar alguns comandos.

Essa foi apenas uma pequena introdução sobre o editor. [O Vim vai muito além disso](#), por ser um editor de texto muito versátil, muitas pessoas utilizam como uma IDE para desenvolvimento! Ele possui vários plugins e diversos outros comandos que auxiliam o desenvolvedor.