



LPIC-1

Fernando Ruan

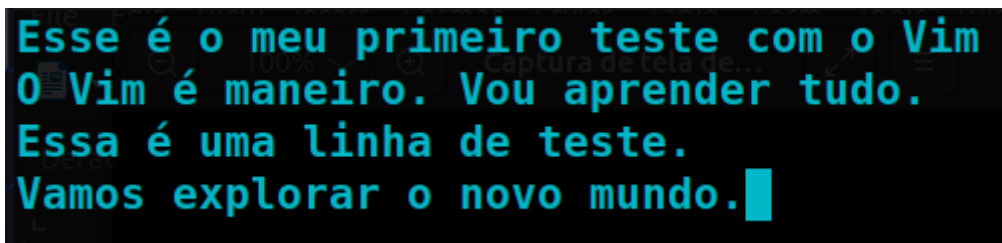
Olá, pessoal, tudo bem? Estou montando esse caderno de estudos onde estou registrando, por completo, toda a minha aventura e descobertas seguindo a trilha de certificações Linux da *Linux Professional Institute*. É um grande prazer poder está aqui contribuindo com a jornada de cada um. Um grande abraço :)

De: Fernando Ruan.

Para começar, vamos abrir o nosso Vim. Usando o comando **vim 1.txt**. Após isso, aperte **i** para entrar no modo de inserção e digitar um pequeno texto:

“Esse é o meu primeiro teste com o Vim.
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo”

Use o enter para dá quebras de linha no momento.



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

É assim que deve ficar para você. Após terminar isso, aperte o botão **ESC** para sair do modo de inserção. Inclusive, perceba que quando apertar o **ESC** ele volta o cursor para o último caractere que foi printado na tela. Apenas no modo de inserção é possível avançar uma posição a mais por conta que o cursor sempre imprime na sua posição e avança. Iremos falar mais sobre isso depois.



```
-- INSERÇÃO --
```

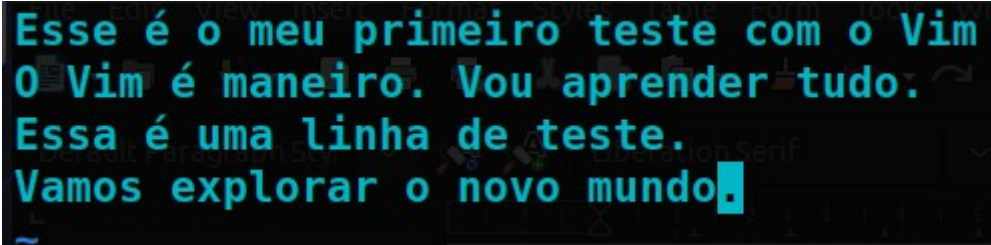
Perceba que quando no modo de inserção, aparece essa mensagem logo abaixo

A nossa primeira etapa de aprendizagem vai ser aprender a conseguir se mover sem precisar usar tanto as teclas do teclado. Para isso, o Vim disponibiliza algumas opções como:

h → move o cursor para a esquerda
j → move o cursor para baixo
k → move o cursor para cima

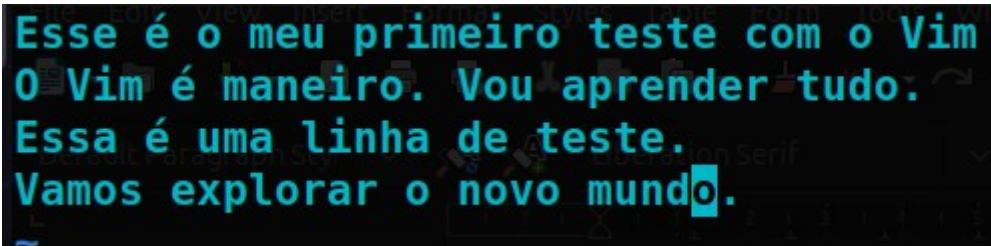
l → move o cursor para a direita

Vamos testar. Usando a tecla **h** veremos se a mesma vai mesmo para a esquerda. Primeiro, vamos ver onde está o nosso cursor após termos dado um **ESC**.



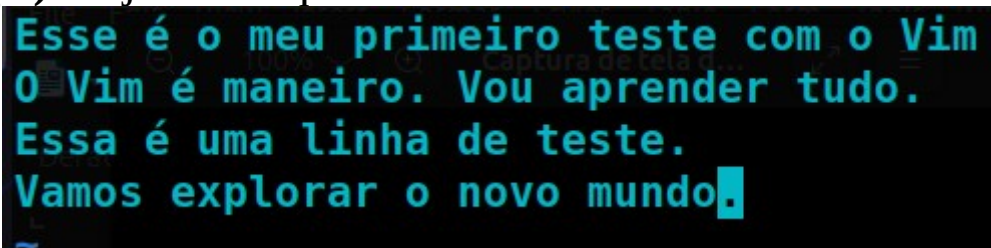
```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Ou seja, ele está no “.”. Agora vamos testar a movimentação com o **h**.



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Perceba que ele foi para a esquerda, como informei. Agora, e se eu quiser voltar para a posição de onde eu sai? Basta, eu pressionar a tecla **l (L)**. Vejamos o que acontece.



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Ou seja, voltamos para onde viemos. E se eu quiser subir para a linha de cima? Eu posso pressionar a tecla **k**. Vamos verificar seu funcionamento.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Perceba que ela subiu e fez o cursor apontar para o último caracter da linha superior. Isso acontece por conta que não temos mais caracteres na linha debaixo. Por isso, o cursor subiu para cima ocupando a última posição disponível. Ok, e se eu quiser descer novamente? Basta eu pressionar a tecla **j**. Analisando isso...

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

E estamos nós aqui novamente :)

Certo, mas, se mover um em um caracter seja para cima/baixo ou esquerda/direita é cansativo. E se quisermos uma posição mais específica como avançar para a primeira letra da palavra anterior ou para a primeira palavra da próxima palavra ou para a última letra da palavra atual, início da linha ou final da linha???

Para isso, temos a nossa disposição:

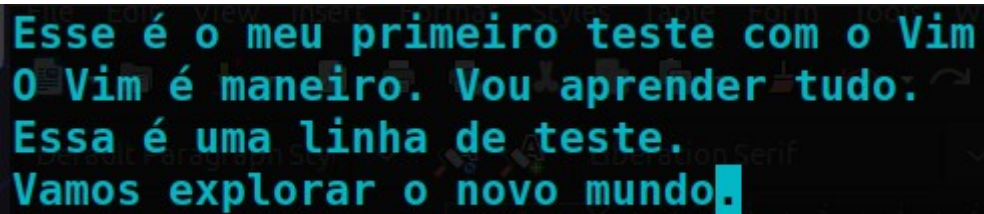
w → **avança o cursor para a primeira letra da próxima palavra.**

b → **retrocede o cursor para a primeira letra da palavra anterior.**

e → **avança o cursor para a última letra da palavra atual**

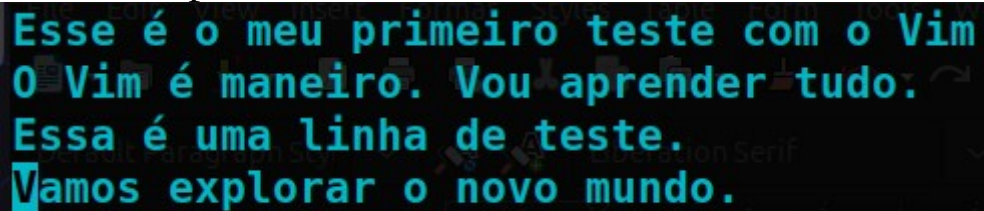
0 → **retrocede o cursor para o início da linha atual**

\$ → **avança o cursor para o final da linha**



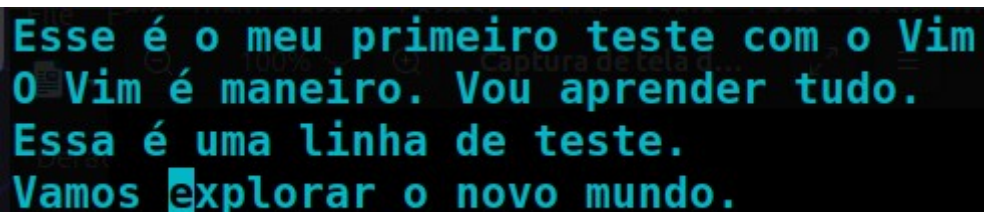
```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Perceba que o nosso cursor está no final da linha. Vamos começar posicionando ele para o início dessa linha. Para isso, qual tecla mesmo devemos pressionar? A tecla **O**.



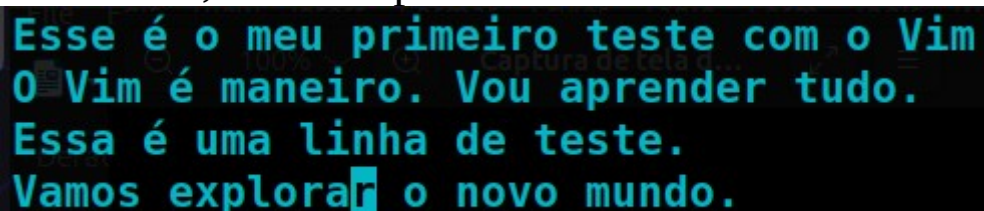
```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Agora estamos no início da linha. Se eu quiser ir para o início da próxima palavra, me posicionando na primeira letra, o que devo fazer? Pressionar a tecla **w**.



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Ok. E se o que eu desejo, na verdade, seja a letra final dessa palavra? Para isso, basta eu pressionar a tecla **e**.



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Agora, nesta posição, se eu quiser voltar para a primeira letra da próxima palavra eu devo pressionar a tecla **b**. Entretanto, se eu estou em uma palavra e ao final dela, qual é o início de palavra anterior mais próximo??? A da atual. Logo, o resultado será:

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Exatamente esse. Pressionando novamente a tecla **b**.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Qual era a palavra mais próxima anterior ao meu cursor? Sim! Era a palavra “Vamos”. Por isso, meu cursor foi colocado nessa posição. E se quisermos nos posicionar no final da linha? Basta, pressionarmos o **\$** usando **shift + 4** (isso por conta que o **\$** está acima do 4 do teclado brasileiro).

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
```

Perceba que meu cursor se posicionou imediatamente no último caracter que foi digitado. Isso é um ponto importante de se notar por conta que vamos começar a brincar com edição de texto. Afinal, como podemos editar no Vim? Aprendemos a nos locomover de forma um pouco mais eficiente. Como editar? Para isso, o Vim nos disponibiliza vários comandos em teclas. Lá vai:

i → entra no modo de inserção **ANTES** do cursor. O primeiro caracter será impresso na esquerda do cursor.

I → entra no modo inserção no **INÍCIO** da linha.

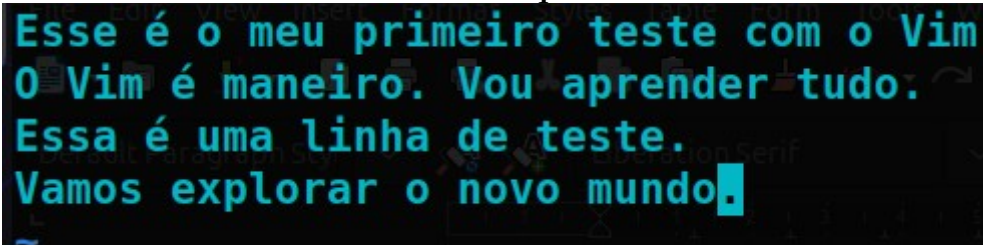
A → entra no modo inserção no **FINAL** da linha.

a → entra no modo de inserção movendo o cursor um pouco a esquerda (INSERÇÃO A ESQUERDA DO CURSOR).

O → entra no modo de inserção adicionando uma nova linha acima da posição do cursor.

o → entra no modo de inserção adicionando uma nova linha abaixo do cursor.

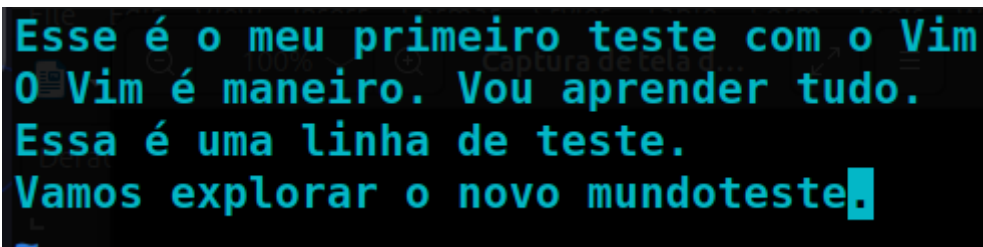
Vamos testar esses novos recursos que temos?



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundo.
~
```

Primeiro de tudo, perceba que estamos com o cursor em “.”.

Vamos adicionar uma palavra. Que tal “**teste**”. Ao pressionar a tecla “**i**”. A primeira coisa que irá acontecer é que irá aparecer a palavra “**inserção**” indicando que entramos nesse modo. Lembre-se que para sair do modo de inserção. Você precisa pressionar **ESC**.



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundoteste.
~
```

Perceba que o meu cursor ficou onde? No mesmo lugar onde ele estava e tudo foi sendo inserido sempre a esquerda do cursor.

Certo, e se eu quiser digitar algo após o “**m**” de “**mundoteste**”.

Para eu voltar para o início da palavra anterior (que no nosso caso agora é o “**m**” da mesma palavra pela forma que o comando usa).

Qual tecla devemos usar? Usaremos a tecla **b**. (**lembrando de pressionar ESC para sair do modo de inserção primeiro**).


```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo mundoteste.
```

Perceba que estamos na nossa letra alvo. Desejo digitar “oi” após a letra “m”. Para que eu possa fazer isso, eu devo usar a tecla “a”.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo m undoteste.
```

Podem notar como o meu cursor se posicionou para a direita de “m”? Pois é, quando eu digo que quando pressiona “a” ele vai printar a esquerda, na verdade, ele move o cursor um passo a direita e volta a printar a esquerda do cursor. Vamos testar digitando “oi”.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
Vamos explorar o novo moi undoteste.
```

Perceba que é praticamente o mesmo comportamento que notamos quando pressionamos “i” com a diferença de apenas um único passo a direita. O cursor também se mantém posicionado em cima da letra “u” que tínhamos como alvo do cursor antes do teste, após pressionarmos a tecla “a”.

Certo, e se quisermos começar a inserir no início da linha. Como mencionei, usaremos a tecla “I” para realização desse trabalho. Printaremos “**testando**” no início dessa linha.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
testandoVamos explorar o novo moiundoteste.
```

Quando voltamos ao início, eu já irei acreditar que sabem sair do modo de inserção usando a tecla **ESC**. O cursor fica exatamente na letra “**V**” de “**Vamos**”. Ao adicionar “**testando**” ele seguirá o mesmo modo que vimos em “**i**” e “**a**” inserindo a esquerda do cursor, mesmo com ou sem deslocamento do mesmo. E se quisermos inserir no final da linha de cima?

Primeiro, como subir para a linha de cima, lembra? Basta pressionar a tecla “**k**”.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
testandoVamos explorar o novo moiundoteste.
```

Ao apertar o **ESC**, o cursor pode volta uma posição. Estavámos na tecla “**V**” e agora estamos em “**o**”. Perfeito. Vamos para cima agora pressionando a tecla “**k**”.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
testandoVamos explorar o novo moiundoteste.
```

Agora que subimos. Iremos pressionar a tecla “**A**” com shift (shift + a). Para colocarmos no modo de inserção exatamente no final da linha.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.
testandoVamos explorar o novo moiundoteste.
```

Perceba que, como mencionei, o cursor sempre fica, no modo de inserção, a uma posição após o último caracter nesse modo. Isso explica o retorno quando saímos do modo de inserção. Vamos adicionar a palavra **“newtest”**.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Perfeito. E se eu quiser entrar no modo de inserção para adicionar uma nova linha acima dessa e digitar **“Nova linha”**. Para isso, usaremos a tecla **“O”**.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova linha
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

E agora digitaremos a nossa linha de interesse

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova linha
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

E se abaixo dessa linha eu desejar colocar uma nova linha como **“Nova linha 2”**? Dessa vez usaremos a tecla **“o”** minúsculo, sem o uso do shift.


```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova linha
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Agora estou na linha logo abaixo. Vamos digitar o que queremos.

```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova linha
Nova linha 2
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Perfeito. Com isso, terminamos essa parte. Agora, e se eu quiser apagar um caracter ou uma palavra inteira ou, de uma posição específica apagar até o final da linha ou a linha completa???

Novos comandos chegando, calma!

x → **apaga o caracter sob o cursor**

dw → **apaga da posição atual até o fim da palavra**

d\$ → **apaga da posição atual até o final da linha**

dd → **apaga a linha inteira.**

Detalhes importantes é que quando copiamos ou deletamos algo, na verdade, o Vim armazena o que capturamos usando um buffer. Por conta disso, podemos realizar as ações como colar novamente um texto que foi retirado. Então, a opção correta aqui não é “deletar” e sim “recortar”.

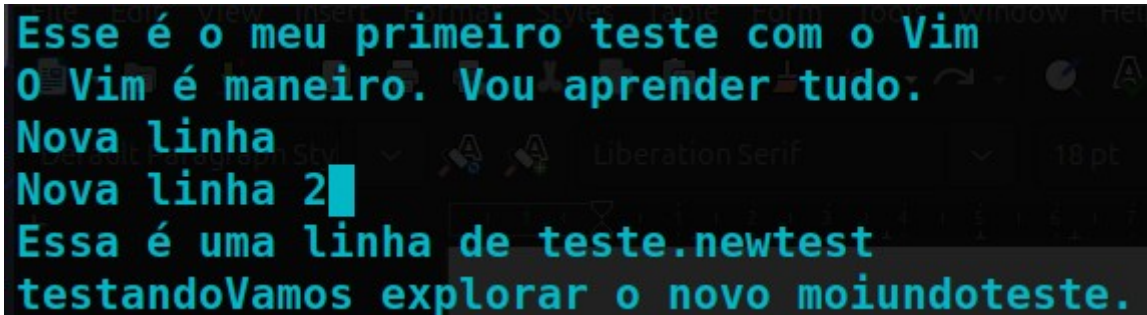
Outro detalhe importante. Para voltar uma ação anterior ou realmente “colar” o que estava no buffer, podemos usar os seguintes comandos:

p → **cola o conteúdo após o cursor ou linha**

P → cola o conteúdo antes do cursor ou linha

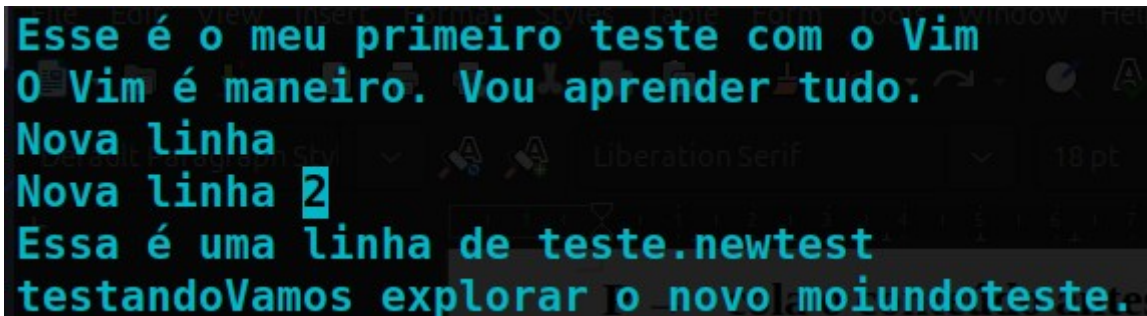
u → volta atrás uma ação que fizeste

Acredito que ter isso em mente irá nos ajudar nessa próxima etapa. Vamos lá.



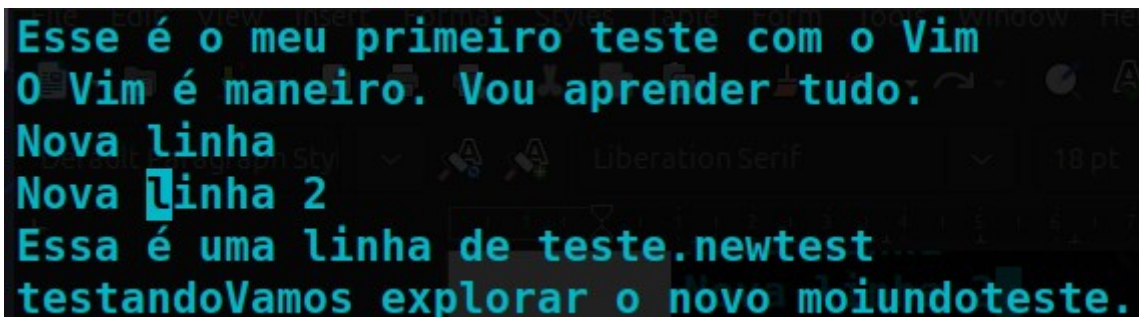
```
Esse é o meu primeiro teste com o Vim
0 Vim é maneiro. Vou aprender tudo.
Nova linha
Nova linha 2
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Na posição onde eu estou, desejo apagar o “in” de “linha”. Como voltamos mesmo? Pressionaremos a tecla “b” duas vezes para retornar para a tecla “I”? Não. Por qual motivo? Ao sair do modo de inserção, o cursor volta um passo para trás. Vejam novamente o fenômeno.



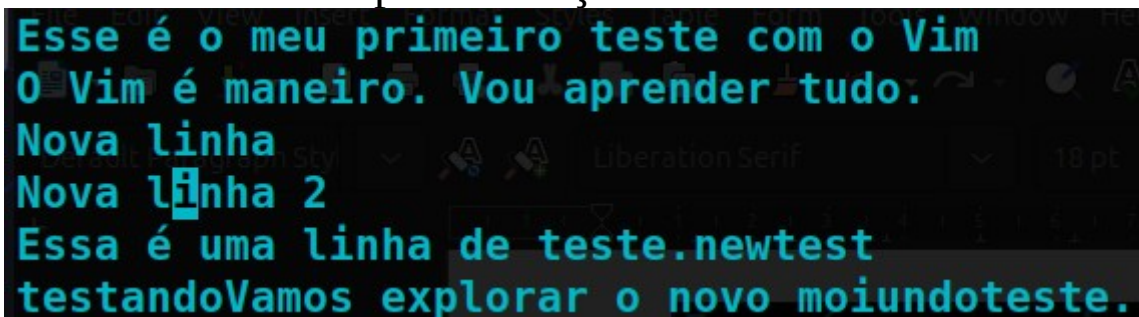
```
Esse é o meu primeiro teste com o Vim
0 Vim é maneiro. Vou aprender tudo.
Nova linha
Nova linha 2
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Agora, iremos pressionar “b” apenas uma única vez. Com isso, estaremos posicionados bem acima da letra “I”. Mas, não é esta o nosso alvo e sim quem está do lado direito dela. O que devemos fazer para nos locomovermos para a direita mesmo, não vale teclas! Usaremos a tecla “I”, que spoiler não?



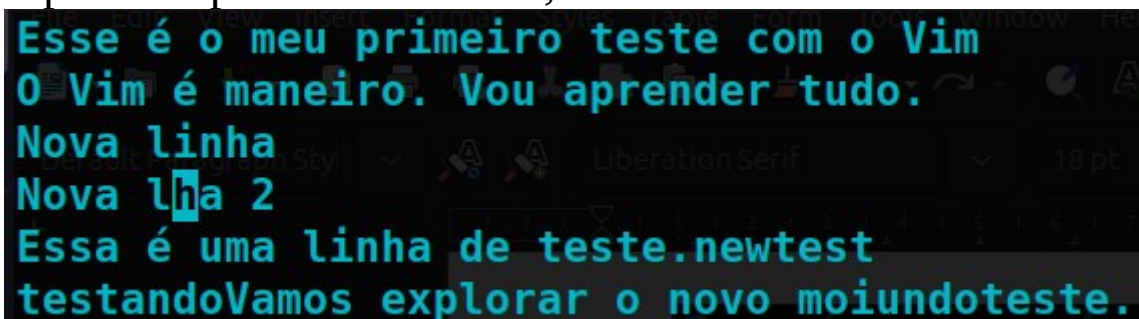
```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova linha
Nova linha 2
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Nessa imagem acima podemos perceber o retorno com “b”. Agora vamos usar a tecla “l” para alcançar o nosso alvo “i”.



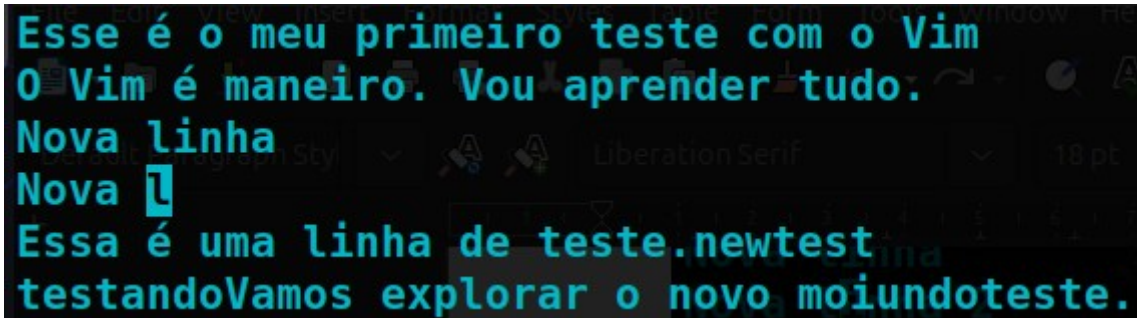
```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova linha
Nova linha 2
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Como o nosso alvo na mira. Para retirarmos um único caracter, este apontado pelo nosso cursor, usaremos a tecla “x” duas vezes.



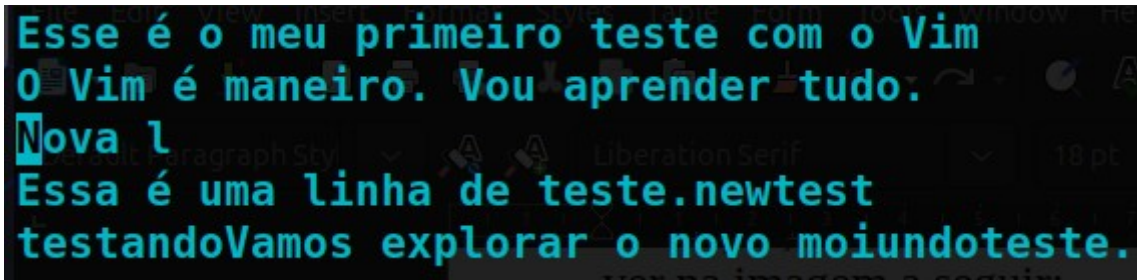
```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova linha
Nova linha 2
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Perceba que conseguimos remover os dois caracteres de nossa palavra alvo tranquilamente. E se quisermos dessa posição, ou seja, com o “h” incluso apagar tudo até o final dessa linha. Para isso usaremos **d\$** . Lembra do “\$”? Usamos ele para mover o nosso cursor para o final da linha. O que deve sobrar? Ao fazer essa ação, tudo, desde o cursor será totalmente apagado. Podemos ver na imagem a seguir:



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova linha
Nova l
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

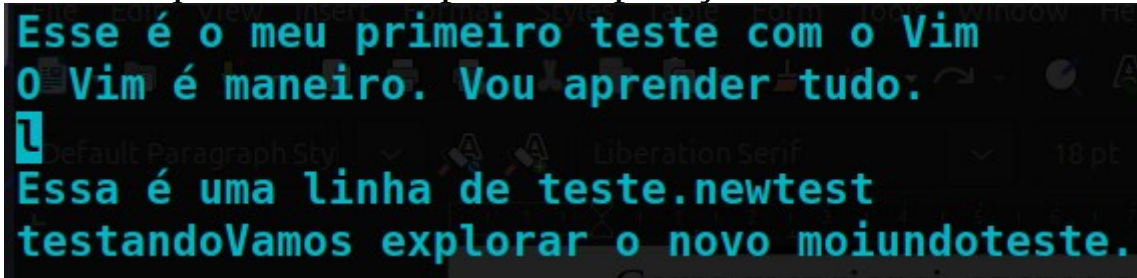
Como mencionei, apenas o “l” vai ficar presente agora. E se eu quiser apagar toda a linha de cima? Vamos lá. Aperte “k” e dê “dd”.



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Nova l
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Lembre-se que a linha foi removida, mas se encontra no buffer e podemos usar teclas como “P” ou “p” para recuperar o conteúdo do jeito que desejarmos. Já iremos verificar sobre elas.

Como agora, o nosso cursor está posicionado na palavra “Nova” vamos apagar ela inteira usando **dw**, que é a opção que apaga a palavra em que estamos a partir da posição de nosso cursor.



```
Esse é o meu primeiro teste com o Vim
O Vim é maneiro. Vou aprender tudo.
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Perceba que a palavra “Nova” sumiu completamente do mapa. Ok, vamos testar sobre o que acabei de dizer sobre a possibilidade de apagarmos a partir de uma posição em específico? Para retomarmos o que tínhamos antes, vamos apertar a tecla “u”.

```
Esse é o meu primeiro teste com o Vim
0 Vim é maneiro. Vou aprender tudo.
Nova l
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Agora, vamos apertar a tecla “l” para mover o cursor um passo a direita e, em seguida, vamos apertar novamente **dw**.

```
Esse é o meu primeiro teste com o Vim
0 Vim é maneiro. Vou aprender tudo.
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
```

Essa é a nova saída que teremos. Percebeu a diferença?

Agora vamos recortar linhas consecutivas e posicionar elas acima da primeira linha atual. Como fazer isso? Digite simplesmente a quantidade de linhas que deseja (contando a partir da atual) e aperte **dd** após. Confie. Não muda nada após digitar, por exemplo, o número 3, mas digite o **dd** após isso e verá a mágica acontecer.

```
Esse é o meu primeiro teste com o Vim
0 Vim é maneiro. Vou aprender tudo.
```

Agora pressionaremos a tecla “k” Para subir para a linha acima e ATENÇÃO! Quando o cursor estiver na posição que desejarmos, se quiseres que ele printe acima, precisas usar “P”, maiúsculo, se quiser após “p”. Começaremos com o primeiro.

```
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
Esse é o meu primeiro teste com o Vim
0 Vim é maneiro. Vou aprender tudo.
```

Perceba como ele colocou tudo acima de onde nosso cursor ficou após pressionarmos a tecla “k” para subir. Iremos desfazer com “u” e agora iremos pressionar “p” e ver o que acontece.


```
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Ele coloca logo abaixo. Essa é a diferença principal entre usar o “P”, que coloca antes do cursor e o “p” que coloca após o cursor.

Existem outros comandos bem úteis. Por exemplo, podemos não querer recortar uma linha e sim apenas copiar ela. Para isso, podemos usar essas opções no Vim:

yy → copia a linha inteira

y\$ → copia da posição do cursor até o final da linha

yw → copia da posição do cursor até o final da palavra

```
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

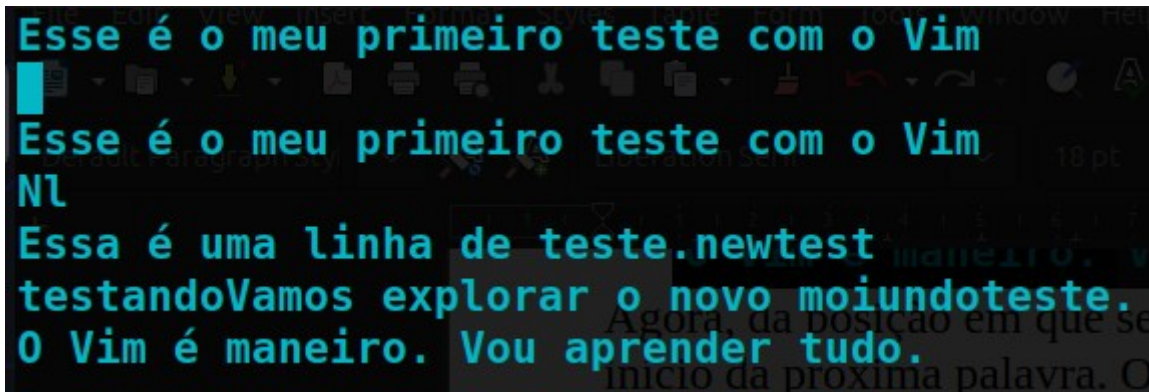
Vamos copiar essa linha inteira e colocar ela acima. Como fazer isso? **yy + P**.

```
Esse é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Agora, da posição em que seu cursor ficou. Vamos avançar para o início da próxima palavra. O que devemos fazer? Pressionar a tecla “w”. Lembra? Ela é quem nos leva para a primeira letra da próxima palavra. Agora pressione **y\$** e depois vamos adicionar

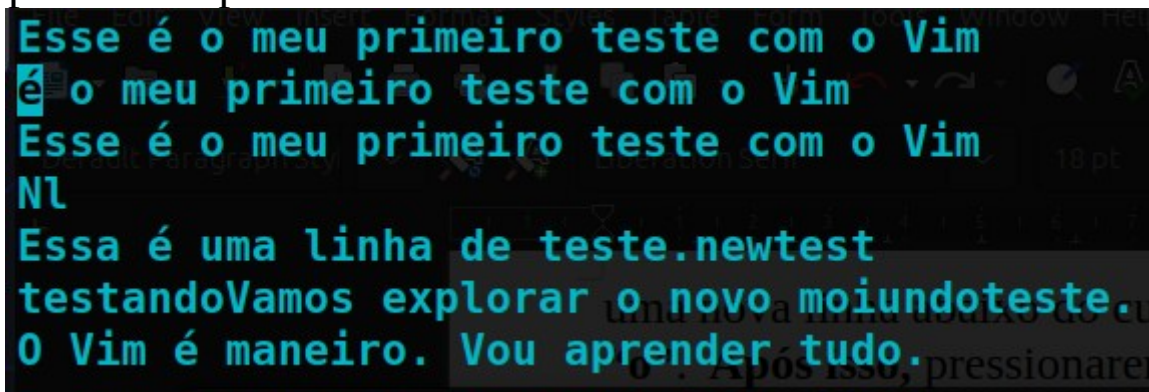
uma nova linha abaixo do cursor. Como fazemos isso? Usaremos **“o”**. **Após isso**, pressionaremos **“p”** para adicionar as informações copiadas.

Uma informação interessante é que podemos usar **CTRL + r** para desfazer uma alteração desfeita com **“u”**



```
Esse é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
NL
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Eu retirei a linha que tinha copiado com **“u”** e agora usarei **CTRL + r** para a recuperar novamente.



```
Esse é o meu primeiro teste com o Vim
e o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
NL
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Perfeito. Agora, e se quisermos pesquisar para saber onde uma palavra está. Nós temos técnicas a nossa disposição

/palavra → busca pela palavra sentido início/fim

?palavra → busca a palavra no sentido fim/início

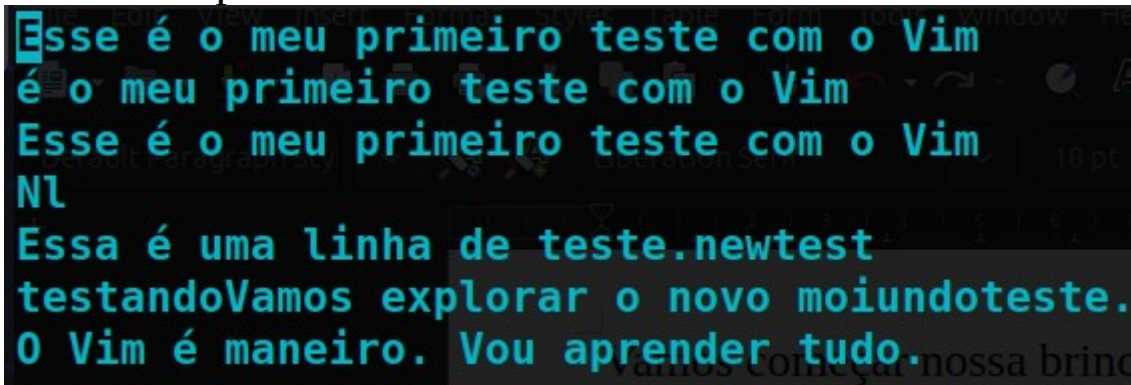
n → pula para o próximo resultado

N → volta para o resultado anterior.

Vamos começar nossa brincadeira pesquisando por “Esse” com /Esse

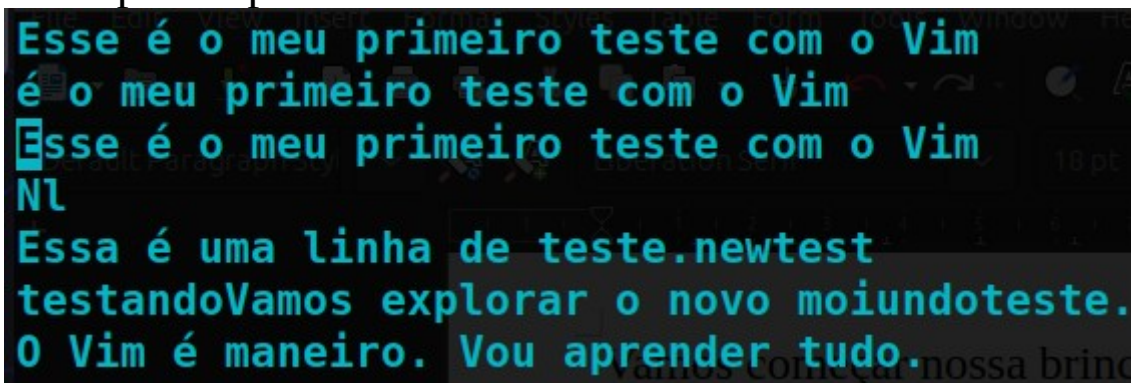
Depois confirme com **ENTER**

Perceba que ele continua na posição em que está. Vamos verificar que funcionou pressionando a tecla “N”.



A screenshot of a Vim editor window with a dark background and light blue text. The text is as follows:
Esse é o meu primeiro teste com o Vim
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.

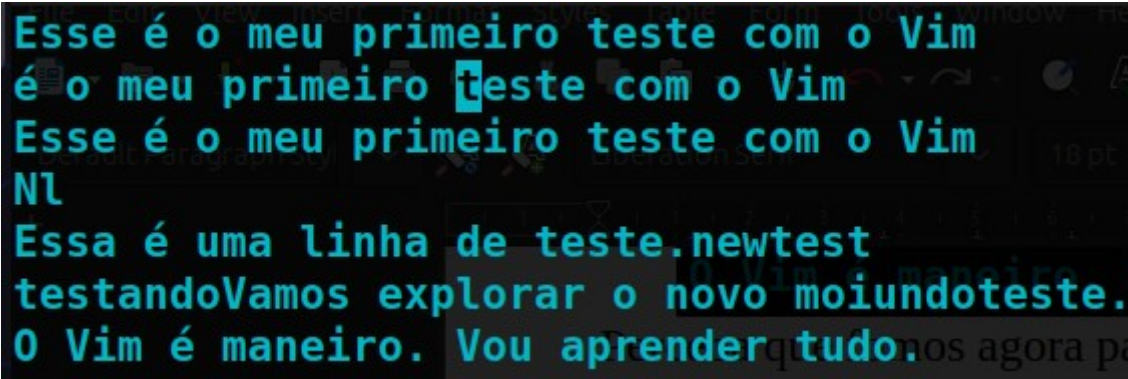
Perceba que fomos agora para o primeiro caso da ocorrência desta palavra. Para testarmos, vejamos se conseguimos retornar com a tecla “n” para o próximo caracter.



A screenshot of a Vim editor window with a dark background and light blue text. The text is as follows:
Esse é o meu primeiro teste com o Vim
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.

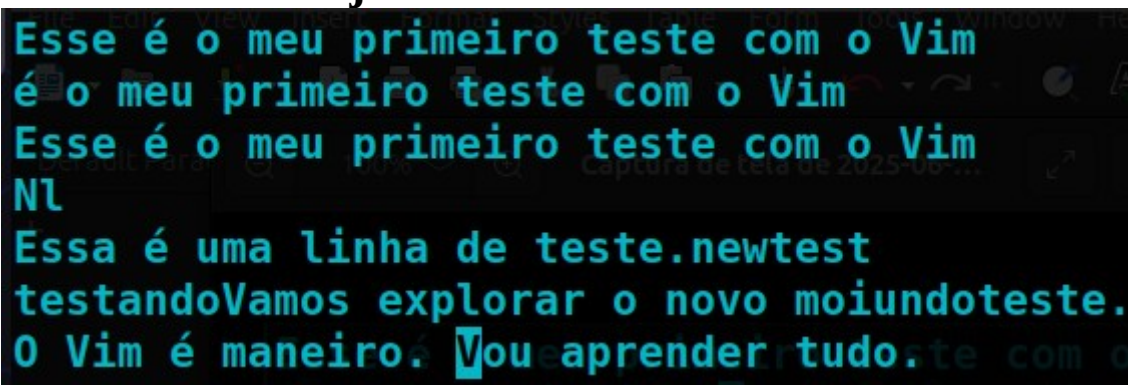
Bingo! Vamos agora realizar um teste de busca com a palavra “teste” por onde iremos querer começar a partir da última ocorrência da mesma. **?teste**

Perceba que o cursor irá se posicionar no segundo “teste” que podemos observar. Isso acontece por conta que a busca considera a pesquisa a partir da posição em que você está. Perceba que meu cursor está no início da terceira linha. Quando eu rodo **?teste** ele deve capturar o quê? Logo a palavra anterior, a primeira que tem essa ocorrência.



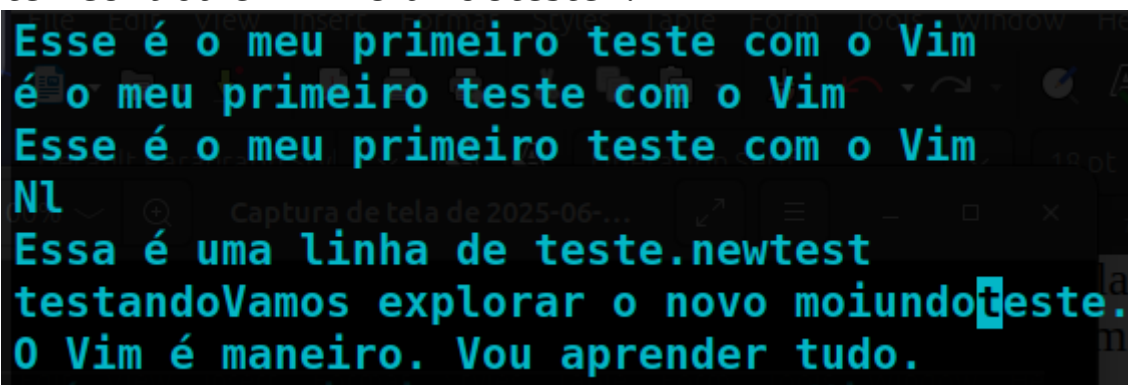
```
Esse é o meu primeiro teste com o Vim
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
NL
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Como podem observar pela imagem acima, confirmando o nosso raciocínio. Vamos fazer um experimento? Vá até a última linha pressionando a tecla “j”.



```
Esse é o meu primeiro teste com o Vim
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
NL
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Agora vamos lançar nosso teste novamente procurando por **?teste** Qual deve ser o resultado? O cursor deve apontar para a substring “teste” contida em “moiundoteste”.



```
Esse é o meu primeiro teste com o Vim
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
NL
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Bingo! O que confirma nosso raciocínio. A mesma coisa funciona quando usamos **/teste** só que agora o alvo é a próxima ocorrência, descendo o arquivo. Para confirmar isso, vamos voltar para o início dessa linha pressionando “0”, lembra?

```
Esse é o meu primeiro teste com o Vim
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Agora vamos verificar se podemos retornar ao “teste” de “moindoteste” usando /teste

```
Esse é o meu primeiro teste com o Vim
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundotesteste.
O Vim é maneiro. Vou aprender tudo.
```

Bingo! Deciframos esse funcionamento. :)

Agora, sabemos como procurar. E se quisermos encontrar e substituir? Para isso podemos fazer o seguinte:

:s/antigo/novo → troca apenas a primeira ocorrência (da linha atual)

:s/antigo/novo/g → troca todas as ocorrências (da linha atual)

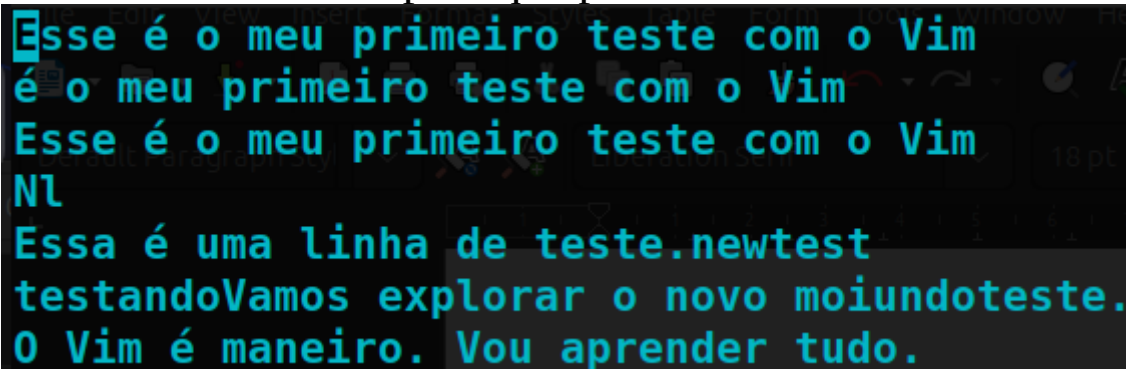
:s/antigo/novo/gc → troca todas, mas confirma antes, cada uma, com você. (da linha atual)

:1,10s/antigo/novo/g → troca todas as ocorrências da primeira até a décima linha

:%s/antigo/novo/g → troca todas as ocorrências do texto de forma global (Esse % significa todas as linhas, é o nosso range que fica ao lado desse comando).

Vamos começar nossa brincadeira trocando a primeira ocorrência da palavra “Esse” por “exec”. Vamos lá. Ainda na primeira linha.

Antes de tudo, preciso que entendas que o efeito da pesquisa só irá acontecer a partir da posição em que você está. Vou explicar executando e mostrando para que possas entender claramente.

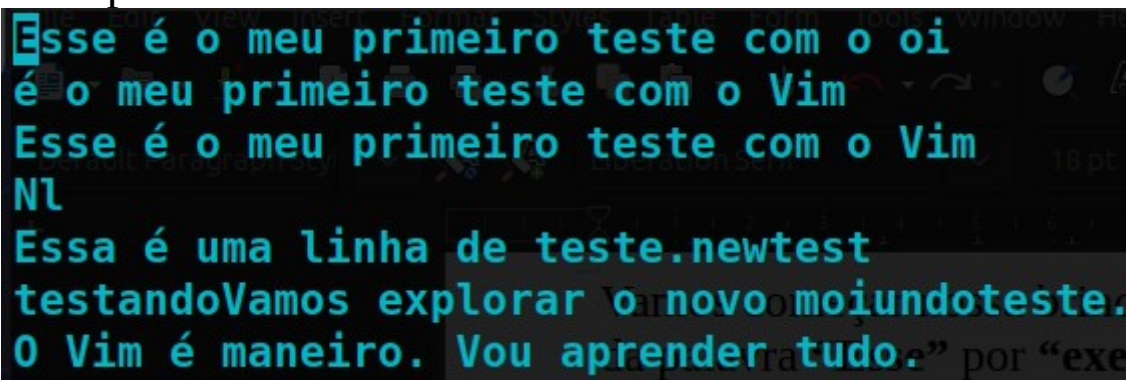


```
Esse é o meu primeiro teste com o Vim
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Desejo trocar a primeira ocorrência da palavra “Vim” por “oi” e desejo manter o raio de alcance apenas na linha atual. Que comando devo executar?

:s/Vim/oi

Com esse comando, sem especificar um range antes do “s” o Vim entende apenas na linha atual.



```
Esse é o meu primeiro teste com o oi
é o meu primeiro teste com o Vim
Esse é o meu primeiro teste com o Vim
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Perceba que mudamos. Para ter certeza, vamos desfazer essa alteração. Como fazer isso, lembra? Tecla “u”. Após isso, tentaremos agora **:s/Vim/oi/g** para forçar uma mudança global e analisar se as demais linhas são ou não são afetadas.

Vamos analisar o resultado na imagem abaixo:


```
Esse é o meu primeiro teste com o oi  
é o meu primeiro teste com o Vim  
Esse é o meu primeiro teste com o Vim  
Nl  
Essa é uma linha de teste.newtest  
testandoVamos explorar o novo moiundote.  
O Vim é maneiro. Vou aprender tudo.
```

Perceba que mesmo colocando o global, apenas a linha em que estou é afetada e isso acontece por conta que eu não especifiquei um range de ação. Para confirmar isso, vamos aumentar o alcance das linhas.

Para ver as linhas, usaremos:

:set number ou :set nu

Para retirar as mesmas:

:set nonumber ou :set nonu

```
1 Esse é o meu primeiro teste com o oi  
2 é o meu primeiro teste com o Vim  
3 Esse é o meu primeiro teste com o Vim  
4 Nl  
5 Essa é uma linha de teste.newtest  
6 testandoVamos explorar o novo moiundote.  
7 O Vim é maneiro. Vou aprender tudo.
```

Agora podemos ver as linhas do lado. Quero pegar da linha 1 até a linha 3 e trocar “Vim” por “oi”.

:1,3s/Vim/oi/g

Quero que seja global para que ele não trocar apenas a primeira ocorrência, queremos testar o poder de alcance.

Vamos rodar esse comando e verificar o que teremos como resultado.


```

1 Esse é o meu primeiro teste com o oi
2 é o meu primeiro teste com o oi
3 Esse é o meu primeiro teste com o oi
4 NL
5 Essa é uma linha de teste.newtest
6 testandoVamos explorar o novo moiundote
7 O Vim é maneiro. Vou aprender tudo.

```

Perceba que alteramos o que queríamos nas linhas alvos, de forma global, e que o cursor ficou posicionado no início da nova última linha alvo.

E se eu quiser Trocar todos os “teste” por “Linux”.

:%s/teste/Linux/g

Com isso, podemos ter um raio de alcance maior. A opção “c” seria apenas para controle nesses casos, para que confirme uma alteração antes de qualquer coisa.

Vamos aplicar o comando: :%s/teste/Linux/gc

Assim testamos ambas.

```

1 Esse é o meu primeiro teste com o oi
2 é o meu primeiro teste com o oi
3 Esse é o meu primeiro teste com o oi
4 NL
5 Essa é uma linha de teste.newtest
6 testandoVamos explorar o novo moiundote.
7 O Vim é maneiro. Vou aprender tudo.

```

Primeira coisa que irá acontecer, a primeira ocorrência será marcada e uma pergunta aparecerá logo abaixo

```

substituir por Linux (y/n/a/q/l/^E/^Y)?

```

O modo de confirmação na substituição oferta várias opções.

y → **confirma**

n → **nega**

a → **confirma todas**

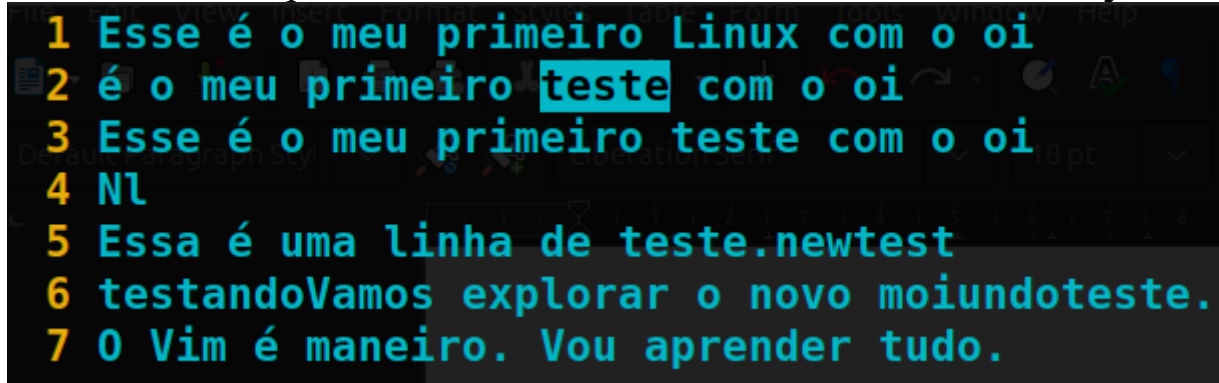
q → sai sem alterar

l → altera esta e sai

^E → rola para cima (Esse ^ é o CTRL)

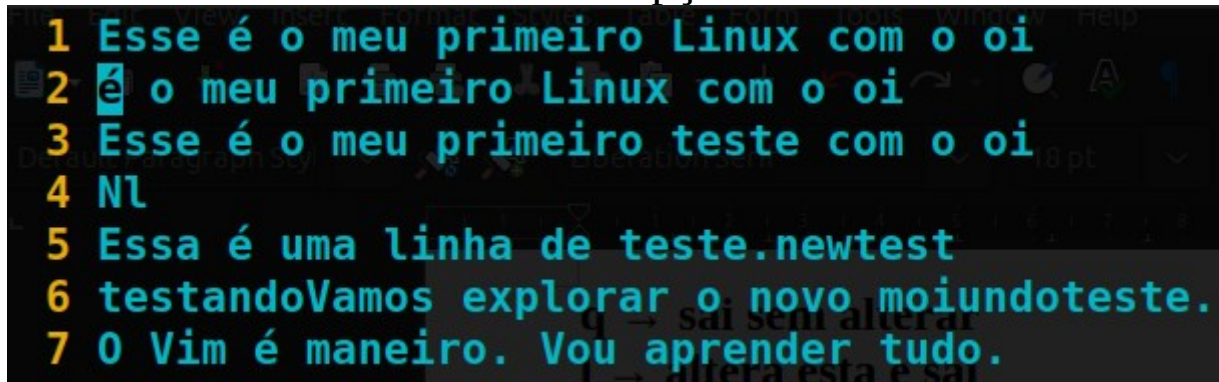
^Y → rola para baixo

No nosso caso, queremos substituir essa. Pressionaremos “y”.



```
1 Esse é o meu primeiro Linux com o oi
2 é o meu primeiro teste com o oi
3 Esse é o meu primeiro teste com o oi
4 Nl
5 Essa é uma linha de teste.newtest
6 testandoVamos explorar o novo moiundoteste.
7 0 Vim é maneiro. Vou aprender tudo.
```

Trocamos o segundo. E se quisermos substituir este e imediatamente sair? “l” é a nossa opção.



```
1 Esse é o meu primeiro Linux com o oi
2 é o meu primeiro Linux com o oi
3 Esse é o meu primeiro teste com o oi
4 Nl
5 Essa é uma linha de teste.newtest
6 testandoVamos explorar o novo moiundoteste.
7 0 Vim é maneiro. Vou aprender tudo.
```

Com isso, ele altera e imediatamente sai do modo de substituição.

Irei retirar os números do lado com **:set nonumber**

Vamos aprender agora a como fazer manipulações usando o modo visual do Vim. Para isso, precisamos aprender novos comandos

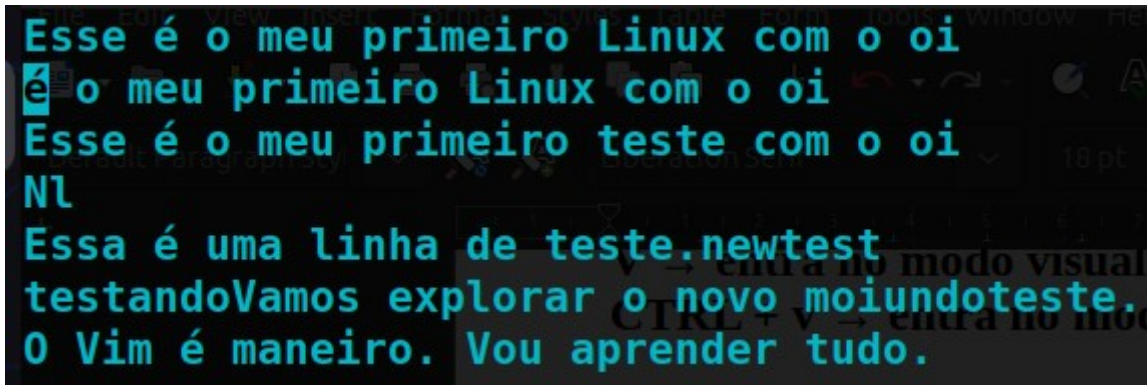
v → entra no modo visual capturando caracter por caracter

V → entra no modo visual capturando linhas inteiras

CTRL + v → entra no modo visual capturando blocos inteiros

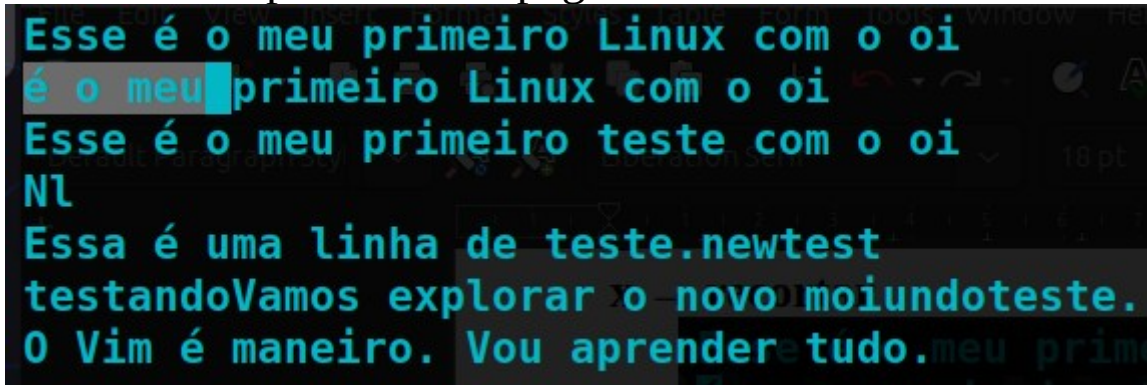
No modo visual, podemos também apagar, deletar, recortar, colar de volta usando comandos similares aos conhecidos.

y → copiar
d → deletar
x → recortar



```
Esse é o meu primeiro Linux com o oi
é o meu primeiro Linux com o oi
Esse é o meu primeiro teste com o oi
NL
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Vamos começar no modo visual de captura de caracteres.
Vamos usando as teclas que conhecemos, **h**, **j**, **k** e **l** para avançar até o “**meu**” e depois vamos apagar com “**d**”.



```
Esse é o meu primeiro Linux com o oi
é o meu primeiro Linux com o oi
Esse é o meu primeiro teste com o oi
NL
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Um fato interessante é que, nesse modo temos um controle maior sobre os caracteres, linhas e blocos inteiros.
Após isso, vamos inserir uma nova linha acima com “**O**” e digitar “**p**” para colar lá.
Lembra o que fazem essas opções?

“**o**” → insere uma nova linha abaixo e entra no modo inserção
“**O**” → também insere uma nova linha só que acima do cursor e entra no modo de inserção.


```
Esse é o meu primeiro Linux com o oi
é o meu
primeiro Linux com o oi
Esse é o meu primeiro teste com o oi
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

E se eu quiser, usando o modo visual, copiar esta linha do meu cursor e as duas abaixo acima da primeira?

V lembre do **j** para descer

```
Esse é o meu primeiro Linux com o oi
é o meu
primeiro Linux com o oi
Esse é o meu primeiro teste com o oi
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

Agora é só apertar “y”.

```
Esse é o meu primeiro Linux com o oi
é o meu
primeiro Linux com o oi
Esse é o meu primeiro teste com o oi
Nl
Essa é uma linha de teste.newtest
testandoVamos explorar o novo moiundoteste.
O Vim é maneiro. Vou aprender tudo.
```

O cursor vai voltar para onde você estava antes. Aí usamos a tecla “k” para subir e logo a tecla “P” que cola acima de onde estamos. Como irei mostrar nesta próxima imagem:

```
é o meu  
primeiro Linux com o oi  
Esse é o meu primeiro teste com o oi  
Esse é o meu primeiro Linux com o oi  
é o meu  
primeiro Linux com o oi  
Esse é o meu primeiro teste com o oi  
NL  
Essa é uma linha de teste.newtest  
testandoVamos explorar o novo moiundoteste.  
O Vim é maneiro. Vou aprender tudo.
```

Sucesso absoluto. Se bem que ainda prefiro a forma antiga que eu ensinei por aqui.

Agora vamos criar um novo arquivo chamado **2.txt** e adicionar o seguinte dentro dele

Com isso,
começaremos a fazer
a mágica da múltipla
edição de linhas.

```
Item A: Teste 1  
Item B: Teste 2  
Item C: Teste 3  
Item D: Teste 4
```

Como fazer isso? Vamos posicionar o nosso cursor no início do arquivo. Lembrando que podes usar **0** e **\$** para **acelerar a captura de tudo**.

0 → início da linha

\$ → final da linha

Após capturar tudo o
que precisamos,
faremos a mágica
acontecer. Pressione
“**I**” para entrar no

```
Item A: Teste 1  
Item B: Teste 2  
Item C: Teste 3  
Item D: Teste 4
```

modo de inserção no início da linha e digite “**test**”. Após isso, veja

a mágica pressionando “ESC”. O mesmo vale para a tecla “A” que adiciona ao final da linha. :)

O vim está se tornando cada vez mais interessante, não? Podemos agora fazer edições em múltiplas linhas, no início, final. É isso aí!!!

```
testItem A: Teste 1
testItem B: Teste 2
testItem C: Teste 3
testItem D: Teste 4
```

Vamos aprender agora a marcar e saltar posições dentro do Vim. Como assim? Se quisermos marcar um caracter onde o nosso cursor estava e depois saltar seja para o mesmo ou para o início da linha em que ele estava, por exemplo, temos comandos específicos para resolver esses casos.

m{letra} → Podemos marcar um caracter como “ma”

{letra} → volta para o início da linha em que consta o caracter marcado

`{letra} → volta exatamente para o caracter marcado

Vamos testar nossas novas capacidades?

Perceba que estou na primeira linha. Desejo marcar a terceira linha com **mb** e a quarta linha

```
testItem A: Teste 1
testItem B: Teste 2
testItem C: Teste 3
testItem D: Teste 4
```

com **mc**. Não irá aparecer nada de especial. Apenas se posicione e faça as marcações. Após isso pressione **‘b** para retornar a **terceira linha** depois pressione **‘c** para retornar a última linha.

Ao final, você deve se encontrar nessa posição final. E se quisermos retomar a posição exata do caracter marcado e não ao início da linha marcada?

```
testItem A: Teste 1
testItem B: Teste 2
testItem C: Teste 3
testItem D: Teste 4
```

Só pressionar `b e `c para retornar ao caracter marcado. :)

Alguns comandos úteis

G → vai para a última linha, ideal para saber quantas linhas tem um arquivo.

gg → vai para a primeira linha do arquivo.

:{numero} → vai para a linha que colocastes.

**:{range}d → deleta o intervalo de linhas passado como
:1,10d → deleta**

:{range}y → copia o intervalo de linhas passado.

:%d → apaga o arquivo inteiro

:%y → copia o arquivo inteiro. (apenas reutilizando o que já aprendemos).

REGISTERS

Muita gente não sabe, mas o Vim possui registradores que podem guardar informações que desejarmos seja como palavras, linhas inteiras e mais. Para acessar um registrador do Vim, precisamos chamar o mesmo usando “**(aspas duplas)**”. Por exemplo:

“ayy → no registrador “a” eu irei copiar a linha inteira

“ayw → no registrador “a” eu irei copiar a palavra

“byw → no registrador “b” eu irei copiar a palavra

**“ap → irei colar o conteúdo que estava em meu registrador
“a”**

**“bp → irei colar o conteúdo que estava em meu registrador
“b”.**