

Comandos Úteis para usar com REGEX

grep (global regular expression print)

Busca textos ou padrões dentro de arquivos ou saídas de outros comandos.

`grep -E "[0-9]{3}-[0-9]{4}" arquivo.txt` → busca por uma expressão regular no arquivo "arquivo.txt".

sed (substituição)

Ele lê cada linha de um arquivo ou entrada, aplica comandos (como substituições, exclusões ou inserções) e **exibe a saída modificada** (por padrão, sem alterar o arquivo original).

Estrutura básica

`sed <comando> arquivo.txt`

Substituição com s

`sed 's/padrão/substituição/opções' arquivo.txt`

s = substituição.

padrao = texto que você quer encontrar.

substituto = texto que vai substituir o padrão encontrado.

opções = modificadores para controlar como a substituição é feita.

arquivo.txt = arquivo onde o sed vai operar.

`sed 's/gato/cachorro/' arquivo.txt` → Substitui **apenas a primeira ocorrência de "gato"** em **cada linha** do arquivo por "cachorro".

`sed 's/gato/cachorro/g' arquivo.txt` → O **g** no final significa **global**, ou seja, substitui **todas as ocorrências de "gato" em cada linha** por "cachorro". Se a linha tiver 3 vezes "gato", todas vão virar "cachorro".

`sed 's/gato/cachorro/i' arquivo.txt` → O **i** aqui quer dizer **ignorar maiúsculas e minúsculas** ao procurar "gato". Assim, ele vai substituir também "Gato", "GATO", "gAto", etc., por "cachorro". **Só substitui a primeira ocorrência por linha**, porque não tem o g.

`sed 's/gato/cachorro/2' arquivo.txt` → O número **2** indica para substituir **apenas a segunda ocorrência de "gato" em cada linha**.

Deletar linhas com d

`sed '/padrão/d' arquivo.txt`

`sed '/^#/d' arquivo.txt` → Remove todas as **linhas que começam com o caractere #**. Isso normalmente são linhas de comentário em arquivos de configuração.

`sed '/^$/d' arquivo.txt` → Remove todas as linhas vazias (linhas sem nenhum caractere, nem espaço), pois `^` significa início da linha e `$` significa fim da linha. Juntos, isso significa "uma linha que não tem absolutamente nada entre o início e o fim", ou seja, uma linha em branco.

Inserir ou adicionar linhas com `i` e `a`

`sed '/Erro/i --- Início de Erro ---' arquivo.txt` → Sempre que encontrar uma linha com a palavra **"Erro"**, insere uma linha **--- Início de Erro ---** antes.

`sed '/Erro/a --- Fim de Erro ---' arquivo.txt` → Insere **--- Fim de Erro ---** depois de cada linha que contenha **"Erro"**.

Editar o arquivo diretamente com `-i` (modo inplace)

- O `-i` significa **"in-place"**, ou seja, edita diretamente no arquivo, sem precisar de redirecionamento (`> arquivo_novo.txt`).
- Sem o `-i`, o `sed` apenas mostra a alteração na tela.
- Com o `-i`, ele faz a alteração dentro do próprio arquivo.

`sed -i 's/Erro/Sucesso/g' arquivo.txt` → Substitui todas as ocorrências de **"Erro"** por **"Sucesso"** dentro do arquivo **arquivo.txt**.

`sed -i.bak 's/Erro/Sucesso/g' arquivo.txt` → Cria uma cópia de backup com a extensão **.bak** antes de modificar. O arquivo original fica salvo como **arquivo.txt.bak**. Faz a substituição de **"Erro"** → **"Sucesso"** no arquivo **arquivo.txt**.