

Módulo 08 – Linux – Atividade 05

1. Pesquise editores de terminal linux, cite e descreva características dos 3 mais abordados.
2. Liste e explique os estados de conteúdo no vim.
3. Liste os principais comandos utilizados para manipulação de tarefas com o vim.
4. Crie um arquivo com o editor vim: "meu_texto_vim.txt", escreva uma receita de qualquer bolo e formate o conteúdo usando os comandos necessários. Ao final, salve o arquivo. Quando fala-se em formatar, é praticar os comandos.

Q1) Os três principais editores de terminal linux são:

Vi/Vim (Vi Improved) – É muito popular, tem uma curva de aprendizagem íngreme devido ao seus modos múltiplos, mas uma vez que se aprende a usar os extensos mapeamentos de teclado e comandos que o Vim oferece, pode ser muito mais produtivo. Por exemplo, sem necessidade de percorrer até ao fim do documento, pode simplesmente pressionar G. Possui mapeamentos chave que também podem ser personalizados e expandidos, além da funcionalidade "macros de gravação" que permite automatizar a sequência de toques de teclas. Vim tem também muitos plugins comunitários que aumentam a funcionalidade ou mesmo embelezam o seu ambiente de trabalho. Oferece ainda uma aplicação interativa chamada "vimtutor" que o levará através de todos os seus comandos e características básicas.

Emacs - É um editor de texto, usado por programadores e usuários que necessitam desenvolver documentos técnicos, em diversos sistemas operacionais. Emacs vs Vim geram muitas discussões na comunidade Linux. O Emacs foi criado em 1976. O Emacs, tal como o vim, pode ser difícil no início, mas irá proporcionar-lhe o ambiente de trabalho mais rápido e produtivo. Para além de ser capaz de fazer tudo o que Vim pode, Emacs é muito extensível. Vem com o seu próprio gestor de pacotes para baixar extensões. Com o Emacs, pode-se navegar na Internet, verificar o seu e-mail, ler pdf, ouvir música, partilhar arquivos, ver filmes, entre outras coisas. O Emacs é sem dúvida um dos mais poderosos editores de texto desenvolvido. Muitas vezes é chamado de sistema operacional em vez de editor de texto e não é difícil de ver porquê.

Nano - É um editor de texto para sistemas Unix-like ou ambientes usando **command line interface (CLI)**. Ele emula o editor de texto Pico, parte do cliente de email Pine, e também provê funcionalidade adicional. Em contraste com Pico, nano é licenciado sob o GNU General Public License (GPL). Lançado como free software por Chris Allegretta in 1999, atualmente nano é parte do GNU Project. Nano, ao contrário de Emacs e Vim, é o mais iniciante possível. O Nano vem pré-instalado na maioria das distros e é normalmente o primeiro editor de texto-cliente que a maioria das pessoas usam. É simples, mínimo, intuitivo, e perfeito para edição de configuração casual.

Q2) Comentando sobre quatro modos do Vim: **Normal, Visual, Insert e Command**:

O **normal mode**, este é o modo ativado assim que o Vim é aberto. Nele, o usuário se depara com os efeitos dos comandos e, normalmente, a surpresa desagrada. Por exemplo, ao digitar um caractere alfanumérico, como o j, a ação equivale a teclar "enter" para mover o cursor uma linha abaixo. Se digitarmos :wq o editor vai salvar o conteúdo e encerrar o processo. Caso escreva uma linha e, de repente, digite dd, a linha em questão será deletada.

Quando passamos para o **modo insert**, toda a confusão inicial é dispensada quando o usuário digita i. Ele serve, justamente, para que o usuário redija o seu texto normalmente. Para sair do insert, basta digitar a tecla esc — e o Vim voltará a operar no modo normal.

o **visual mode** se assemelha bastante ao mecanismo de clicar e arrastar com o mouse. Conforme o usuário seleciona trechos de um conteúdo, ele consegue aplicar comandos restritos à seleção (copiar, substituir, deletar etc.). Para acessar o modo visual, basta pressionar a tecla **v** e usar o cursor.

O **command mode** nos permite aplicar vários comandos de maneira mais fácil que no modo normal. Supondo que você tenha escrito um programa com mais de mil linhas de código e descobriu que determinado termo foi empregado incorretamente em várias partes do conteúdo. Um simples comando pode resolver o problema rapidamente. Para acessar o modo comando, abra o Vim no modo normal e tecle “:”.

Q3) Para listar os principais comandos, listando inicialmente comandos de movimentação:

- **0**: move o cursor para o início da linha;
- **\$**: move o cursor para o fim da linha;
- **k**: move o cursor linha acima;
- **j**: move o cursor linha abaixo;
- **h**: move o cursor um caractere à esquerda;
- **l**: move o cursor um caractere à direita;
- **`.`**: move o cursor para o último elemento editado;
- **G**: move o cursor para o fim do arquivo como um todo;
- **gg**: move o cursor para o início do arquivo;
- **w**: move o cursor uma palavra adiante;
- **b**: move o cursor uma palavra atrás.

Se desejamos mover mais de uma linha ou palavra, basta digitar o valor junto do atalho. Exemplo: **15w** para mover quinze palavras à frente, **8b** para mover oito palavras atrás etc.

Shortcuts para deletar elementos:

- **x**: deleta o caractere apontado pelo cursor;
- **u**: restaura ação mais recente;
- **dw**: deleta uma palavra completa;
- **d0**: deleta até o início da linha;
- **d\$**: deleta até o fim da linha;
- **dgg**: deleta até o início do arquivo;
- **dG**: deleta até o fim do arquivo;
- **d)**: deleta até o fim de uma sentença;
- **dd**: deleta a linha inteira.

Copiar e colar:

- **v**: destaca um caractere por vez;
- **V**: destaca uma linha por vez;
- **Ctrl+v**: destaque por colunas;
- **p**: cola o texto no fim da linha atual;
- **P**: cola texto na posição atual do cursor;
- **y**: puxa o texto para a área de transferência.

Localizando e substituindo elementos:

- **/:** busca pelo termo digitado logo após a barra. Exemplo: /uma ameaça assombrava Gotham City;
- **n:** move o cursor para a próxima instância do texto da última pesquisa;
- **N:** move o cursor para a instância anterior do texto da última pesquisa;
- **?:** funciona como a barra, mas retrocedendo a pesquisa.

Para sair usei Esc, o que leva para o modo normal, pressionei dois pontos (:) que ativa o command mode, pressionei “x” e dei Enter ele grava e sai.

Q4)

Para criar usei o vim “meu_texto_vim.txt”, abrindo o arquivo para edição, como fui adicionar conteúdo copiado da internet, digitei primeiro “v” para trocar para o “visual mode”. Utilizei comandos de movimentação, comandos de copiar e colar (visual mode), pois peguei uma receita pronta de bolo de laranja (<https://www.tudogostoso.com.br/receita/13953-bolo-de-laranja.html>).

Para sair usei Esc, o que leva para o modo normal, pressionei dois pontos (:) que ativa o command mode, pressionei “x” e dei Enter ele grava e sai.

```

letonio@letonio-Inspiron-15-3567: ~/Documentos/alphaedte...
letonio@letonio-Inspiron-15-3567:~/Documentos/alphaedtech/hardSkills/repositorioHard/Alpha-edtech-cycle01/Module08-Linux/Aula05$ vim meu_texto_vim.txt

```

```

BOLO DE LARANJA

INGREDIENTES

-4 ovos
-2 xícaras (chá) de açúcar
-1 xícara (chá) de óleo
-suco de 2 laranjas
-casca de 1 laranja
-2 xícaras (chá) de farinha de trigo
-1 colher (sopa) de fermento-

MODO DE PREPARO

1.Bata no liquidificador os ovos, o açúcar, o óleo, o suco e a casca da laranja.
2.Passe para uma tigela e acrescente a farinha de trigo e o fermento.
3.Leve para assar em uma forma com furo central, untada e enfarinhada, por mais ou menos 30 minutos.
4.Desenforme o bolo e molhe com suco de laranja.

```