



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

CTeSP de Tecnologias e Programação de Sistemas de Informação

Utilização Unix/Linux em linha de comandos

28/03/2025

Nome do Autor:

Filipe Fontes Jerónimo

Curso:

TPSI

Introdução

Este Trabalho Prático tem como objetivo explorar a utilização do Unix/Linux através da linha de comandos, com ênfase nas operações essenciais para manipulação e gestão de ficheiros, extração de informação, redireccionamento, gestão da máquina e dos utilizadores, assim como a configuração do ambiente de trabalho. O trabalho foi desenvolvido no âmbito do curso de Tecnologias e Programação de Sistemas de Informação (CTeSP), no Instituto Politécnico de Cantanhede.

Ao longo deste relatório, serão abordados diversos tópicos e operações comuns no sistema Unix/Linux, como a criação, manipulação e cópia de ficheiros, a contagem de linhas e palavras em ficheiros de texto, o redireccionamento de saídas e a utilização de pipes, bem como a gestão de utilizadores e a configuração do ambiente de trabalho. Além disso, serão apresentadas as etapas de personalização do ambiente de trabalho, como a criação de aliases e scripts Bash para automação de tarefas.

Este trabalho tem como objetivo não só a compreensão dos comandos básicos, mas também a capacidade de configurar e otimizar o ambiente de trabalho de forma eficiente, melhorando a produtividade e o controlo sobre o sistema. Ao final do trabalho, será demonstrada a execução de todas as tarefas propostas, com a explicação detalhada dos comandos utilizados e capturas de ecrã que ilustram os resultados obtidos.

Índice

Introdução

Lista de Comandos e Opções

Gestão de Diretórios e Ficheiros

Manipulação de Conteúdo

Informação do Sistema e Utilizadores

Gestão de Permissões e Utilizadores

Variáveis e Configuração de Ambiente

Gestão de Permissões

1. Manipulação e Gestão de Ficheiros

1.1 Criar um diretório chamado `"meu_trabalho"` e navegar para ele.

1.2 Criar um ficheiro de texto chamado `"notas.txt"` e adicionar algum conteúdo.

1.3 Copiar o ficheiro `"notas.txt"` para um novo ficheiro chamado `"backup_notas.txt"`.

1.4 Mover `"backup_notas.txt"` para um diretório chamado `"backup"`.

1.5 Listar os ficheiros do diretório `"meu_trabalho"` e do diretório `"backup"`.

1.6 Apagar o ficheiro `"notas.txt"` e depois restaurá-lo a partir do `"backup"`.

2. Manipulação e Extração de Informação em Ficheiros de Texto

2.1 Criar um ficheiro `"dados.txt"` com pelo menos 10 linhas de texto.

2.2 Exibir as primeiras 5 linhas do ficheiro.

2.3 Exibir as últimas 3 linhas do ficheiro.

2.4 Contar o número de linhas, palavras e caracteres do ficheiro.

2.5 Procurar e exibir todas as linhas que contenham uma determinada palavra.

2.6 Ordenar as linhas do ficheiro *alfabeticamente* e guardar o resultado num novo ficheiro `"ordenado.txt"`.

3. Redireccionamento e Pipes

3.1 Redireccionar a saída do comando `"ls -l"` para um ficheiro `"lista_ficheiros.txt"`.

- 3.2 Concatenar os ficheiros "dados.txt" e "ordenado.txt" e guardar o resultado em "final.txt".
- 3.3 Contar quantas vezes uma palavra aparece no ficheiro "final.txt".
- 3.4 Filtrar e exibir apenas as linhas que contêm números em "final.txt".
- 4. Comandos para Gestão da Máquina e Utilizadores
 - 4.1 Exibir o nome do utilizador atualmente autenticado.
 - 4.2 Mostrar informações sobre a utilização do disco.
 - 4.3 Consultar a carga do sistema e os processos em execução.
 - 4.4 Listar os utilizadores atualmente ligados ao sistema.
 - 4.5 Criar um novo utilizador (se permitido) e atribuir-lhe uma password.
- 5. Configuração do Ambiente de Trabalho em Linha de Comandos
 - 5.1 Criar um alias chamado listar que execute "ls -lh".
 - 5.2 Adicionar o diretório "meu_trabalho" à variável "PATH".
 - 5.3 Personalizar o prompt de comandos.
 - 5.4 Criar um script "Bash" que exiba uma mensagem de boas-vindas ao iniciar sessão.

Conclusão

Índice de Imagens

Imagem 1 - abrir o terminal	13
Imagem 2 – terminal aberto	13
Imagem 3 – cd Desktop/	14
Imagem 4 – mkdir meu_trabalho	14
Imagem 5 – ls	14
Imagem 6 – cd meu_trabalho	15
Imagem 7 – echo “Olá Filipe” > notas.txt	16
Imagem 8 – cat notas.txt	16
Imagem 9 - cp notas.txt backup_notas.txt	17
Imagem 10 - cat backup_notas.txt	17
Imagem 11 – mkdir backup	18
Imagem 12 - mv backup_notas.txt backup/	18
Imagem 13 – ls backup/	19
Imagem 14 – ls	20
Imagem 15 – ls backup/	20
Imagem 16 – rm notas.txt	21
Imagem 17 – ls notas.txt	21
Imagem 18 - cp backup/backup_notas.txt notas.txt	22
Imagem 19 – ls -l notas.txt	23
Imagem 20 – cat notas.txt	23
Imagem 21 – cat > dados.txt <<EOF	24
Imagem 22 – head -n 5 dados.txt	25
Imagem 23 – tail -n 3 dados.txt	25
Imagem 24 – wc dados.txt	26
Imagem 25 – nano dados.txt	27
Imagem 26 – editor nano na linha de comandos	27
Imagem 27 – adição da palavra ornitorrinco	28
Imagem 28 – confirmação de edição	28
Imagem 29 – ctrl + x para voltar para o terminal	29
Imagem 30 – cat dados.txt depois da alteração	29
Imagem 31 – grep “Ornitorrinco” dados.txt	30

Imagem 32 – sort dados.txt > ordenado.txt	31
Imagem 33 – cat ordenado.txt	31
Imagem 34 – ls -l > lista_ficheiros.txt	32
Imagem 35 – cat lista_ficheiros.txt	32
Imagem 36 - cat dados.txt ordenado.txt > final.txt	33
Imagem 37 – cat final.txt	33
Imagem 38 – grep -o “Ornitorrinco” final.txt wc -l	34
Imagem 39 - grep -E '[0-9]' final.txt	35
Imagem 40 - whoami	36
Imagem 41 – df -h	37
Imagem 42 – du -sh Desktop/meu_trabalho/	37
Imagem 43 - top	38
Imagem 44 – processos	38
Imagem 45 - uptime	39
Imagem 46 – who	40
Imagem 47 - w	40
Imagem 48 – sudo useradd teste	41
Imagem 49 – sudo passwd teste	41
Imagem 50 – getent passwd teste	42
Imagem 51 – Login teste	42
Imagem 52 – nano ~/.bashrc	43
Imagem 53 - alias listar='ls -lh'	43
Imagem 54 – guardar nano ~/.bashrc	44
Imagem 55 – source ~/.bashrc	44
Imagem 56 - listar	44
Imagem 57 - export PATH="\$HOME/meu_trabalho:\$PATH"	45
Imagem 58 - echo \$PATH	45
Imagem 59 – nano ~/.bashrc	46
Imagem 60 – PS1	47
Imagem 61 – Personalização do prompt concluída	47
Imagem 62 – nano ~/inicio.sh	48
Imagem 63 – echo “Bem-vindo ao ambiente de trabalho, \$(whoami)”	48
Imagem 64 – nano ~/.bashrc	49
Imagem 65 – adição do comando ~/inicio.sh no nano ~/.bashrc	49
Imagem 66 – erro source ~/.bashrc	50

Imagem 67 – chmod +x ~/inicio.sh	50
Imagem 68 – source ~/.bashrc	50

Lista de Comandos e Opções

Gestão de Diretórios e Ficheiros

cd

Muda de diretório.

mkdir

Cria um diretório.

ls

Lista ficheiros e diretórios.

- -l → Exibe os ficheiros em formato de lista detalhada.
- -lh → Exibe a lista detalhada com tamanhos em formato legível (KB, MB, GB).
- > → Redireciona a saída do comando para um ficheiro.

echo

Exibe uma mensagem ou texto no terminal.

- > → Redireciona a saída para um ficheiro, substituindo o conteúdo existente.
- >> → Redireciona a saída para um ficheiro, adicionando sem apagar o conteúdo anterior.
- \$PATH → Variável de ambiente que armazena os diretórios onde os comandos são procurados.

cat

Exibe o conteúdo de um ficheiro.

- > → Redireciona a entrada para um ficheiro, substituindo o conteúdo existente.
- << → Inicia um **here document**, permitindo múltiplas linhas de entrada até um marcador final.

cp

Copia ficheiros ou diretórios.

mv

Move ou renomeia ficheiros e diretórios.

rm

Remove ficheiros ou diretórios.

Manipulação de Conteúdo

head

Exibe as primeiras linhas de um ficheiro.

- -n → especifica o número de linhas a serem exibidas do ficheiro.

tail

Exibe as últimas linhas de um ficheiro.

- -n → especifica o número de linhas a serem exibidas do ficheiro.

wc

Conta linhas, palavras e caracteres de um ficheiro.

nano

Editor de texto no terminal.

grep

Pesquisa padrões em ficheiros ou texto.

- -o → Exibe apenas as partes do texto que correspondem ao padrão.
- | → Encaminha a saída de um comando para a entrada de outro (**pipe**).
- -E → Usa expressões regulares estendidas.
- '[0-9]' → Procura por números dentro do texto.

sort

Ordena linhas de um ficheiro ou entrada.

- > → Redireciona a saída ordenada para um ficheiro.

Informação do Sistema e Utilizadores

whoami

Exibe o utilizador atualmente autenticado.

df

Mostra informações sobre o espaço em disco.

- `-sh` → Exibe o espaço utilizado de forma resumida e legível.

top

Monitora os processos em execução no sistema.

uptime

Exibe há quanto tempo o sistema está ligado.

who

Lista utilizadores atualmente ligados ao sistema.

w

Mostra quem está ligado ao sistema e o que está a fazer.

Gestão de Permissões e Utilizadores

sudo

Executa comandos como superutilizador (root).

- `useradd` → Cria um novo utilizador no sistema.
- `passwd` → Define ou altera a palavra-passe de um utilizador.

getent

Exibe entradas de bases de dados do sistema.

- `passwd` → Lista os utilizadores do sistema.

Variáveis e Configuração de Ambiente

alias

Cria atalhos para comandos.

source

Aplica alterações feitas num ficheiro de configuração sem precisar reiniciar a sessão.

export

Define variáveis de ambiente.

- `PATH` → Atualiza os diretórios onde o sistema procura comandos.

\$

Representa uma variável ou indica que o comando seguinte será executado no terminal.

Gestão de Permissões

chmod

Modifica permissões de ficheiros e diretórios.

1. Manipulação e Gestão de Ficheiros

1.1 Criar um diretório chamado **"meu_trabalho"** e navegar para ele.

Primeiramente, abre-se o Terminal através do atalho **Ctrl + Alt + T** ou pesquisando por **'Terminal'** no menu de aplicações/menu iniciar.

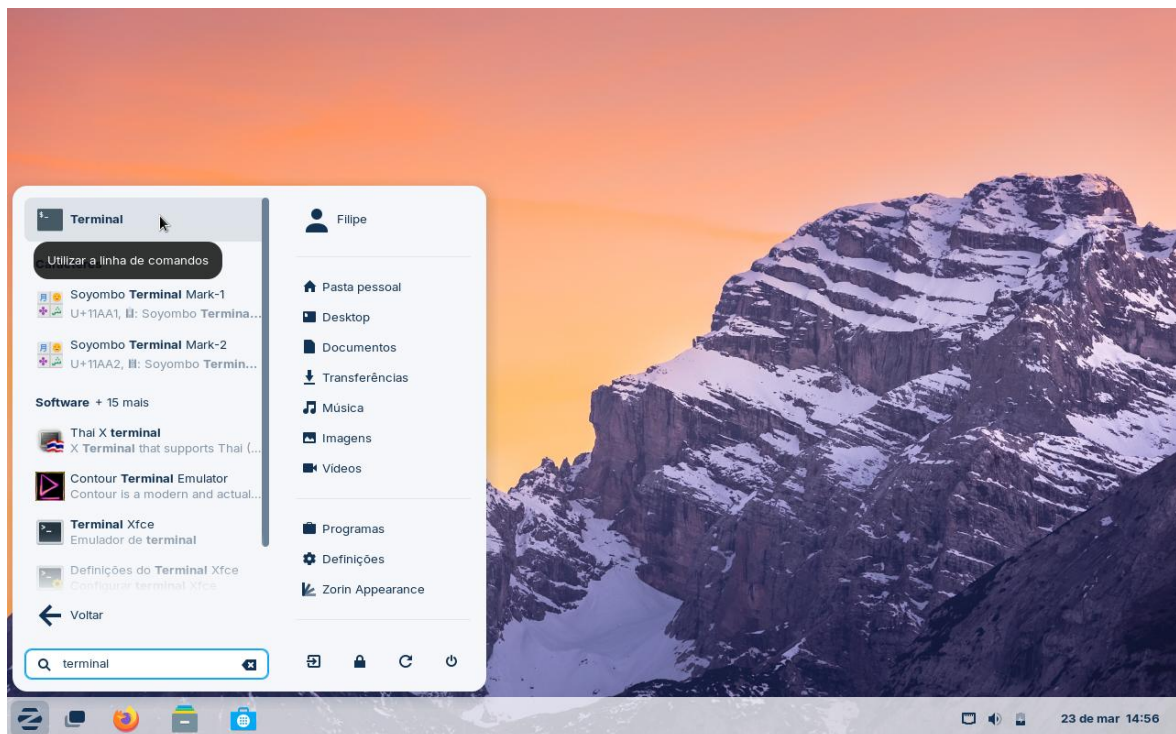


Imagem 1 - abrir o terminal



Imagem 2 – terminal aberto

Utilizei o comando **`cd Desktop/`** para navegar até a pasta Desktop, que está localizada no diretório principal do utilizador.

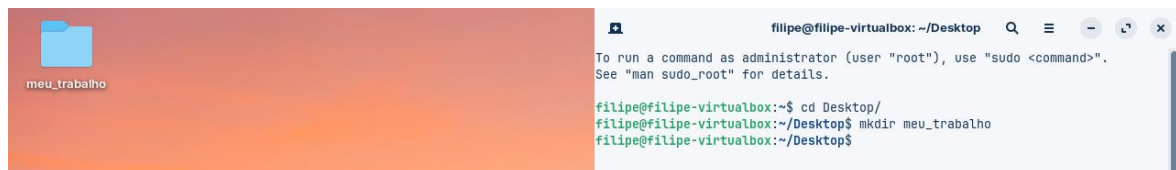


```
filipe@filipe-virtualbox: ~/Desktop
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

filipe@filipe-virtualbox:~$ cd Desktop/
filipe@filipe-virtualbox:~/Desktop$
```

Imagem 3 – `cd Desktop/`

Após isso, utilizei o comando **`mkdir meu_trabalho`** para criar um diretório chamado 'meu_trabalho' dentro da pasta Desktop.

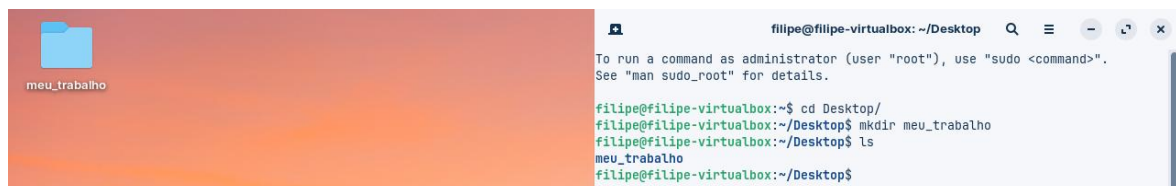


```
filipe@filipe-virtualbox: ~/Desktop
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

filipe@filipe-virtualbox:~$ cd Desktop/
filipe@filipe-virtualbox:~/Desktop$ mkdir meu_trabalho
filipe@filipe-virtualbox:~/Desktop$
```

Imagem 4 – `mkdir meu_trabalho`

Para confirmar a criação do diretório 'meu_trabalho', executei o comando **`ls`**, que exibe a lista de ficheiros e pastas presentes na pasta Desktop.



```
filipe@filipe-virtualbox: ~/Desktop
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

filipe@filipe-virtualbox:~$ cd Desktop/
filipe@filipe-virtualbox:~/Desktop$ mkdir meu_trabalho
filipe@filipe-virtualbox:~/Desktop$ ls
meu_trabalho
filipe@filipe-virtualbox:~/Desktop$
```

Imagem 5 – `ls`

Por fim, executei o comando **`cd meu_trabalho/`** para entrar no diretório 'meu_trabalho', que será o principal local de trabalho para a maior parte do projeto.

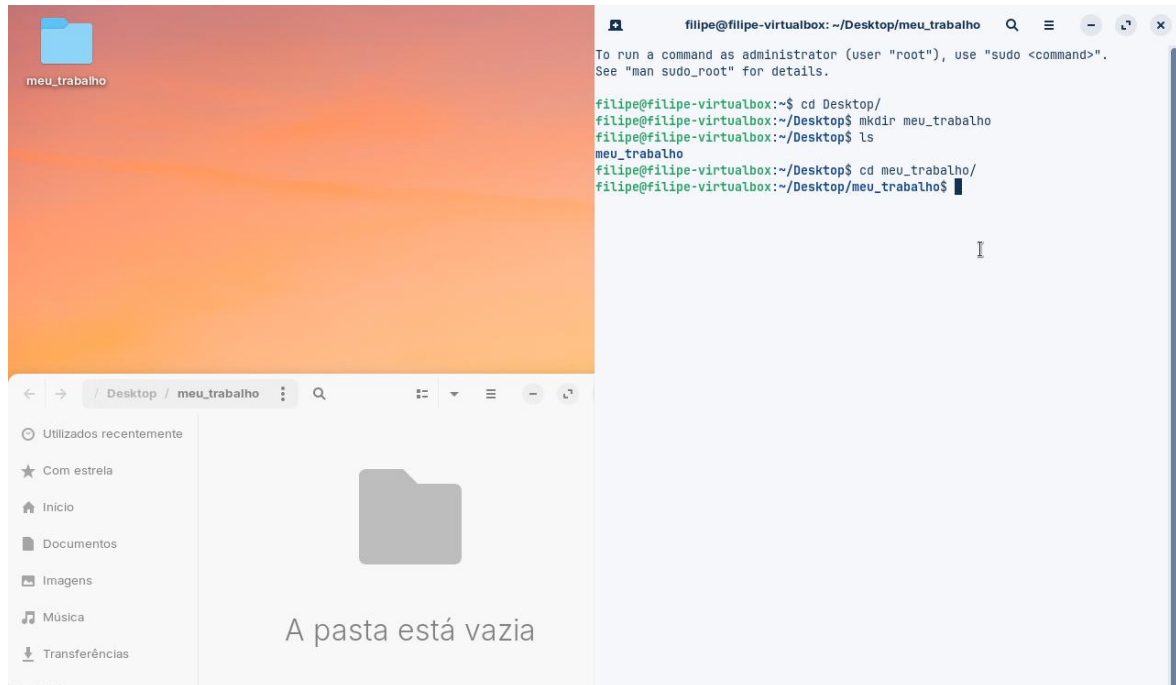


Imagem 6 – `cd meu_trabalho`

1.2 Criar um ficheiro de texto chamado **"notas.txt"** e adicionar algum conteúdo.

Dentro da pasta 'meu_trabalho', criei um ficheiro utilizando o comando **`echo "Olá Filipe" > notas.txt`**.

Este comando escreve a mensagem 'Olá Filipe' no ficheiro **'notas.txt'**. Caso o ficheiro não exista, ele é criado automaticamente; caso já exista, o conteúdo anterior é substituído.

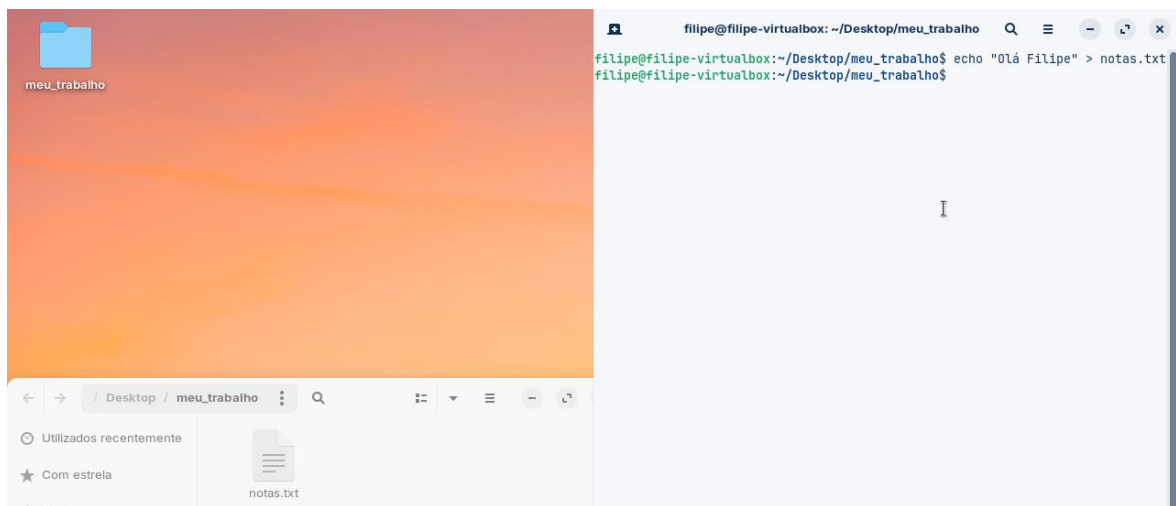


Imagem 7 – `echo "Olá Filipe" > notas.txt`

Para garantir que o ficheiro 'notas.txt' foi criado corretamente e contém o texto desejado, executei o comando **`cat notas.txt`**. Este comando exibe no terminal o conteúdo do ficheiro especificado.

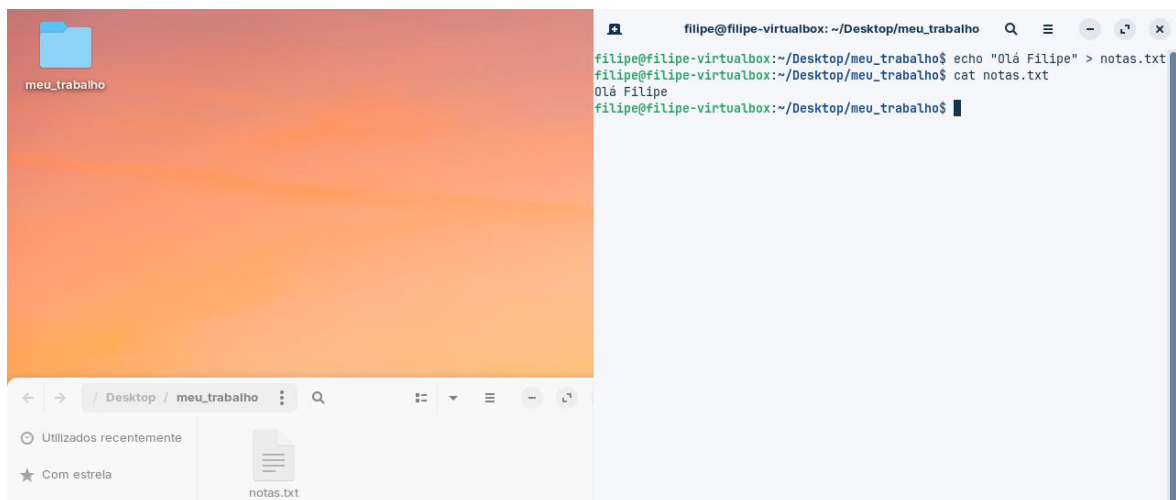


Imagem 8 – `cat notas.txt`

1.3 Copiar o ficheiro “*notas.txt*” para um novo ficheiro chamado “*backup_notas.txt*”.

Para garantir uma cópia de segurança do ficheiro ‘notas.txt’, executei o comando **cp notas.txt backup_notas.txt**. Como ‘backup_notas.txt’ ainda não existia, o comando criou um ficheiro com o mesmo conteúdo de ‘notas.txt’.

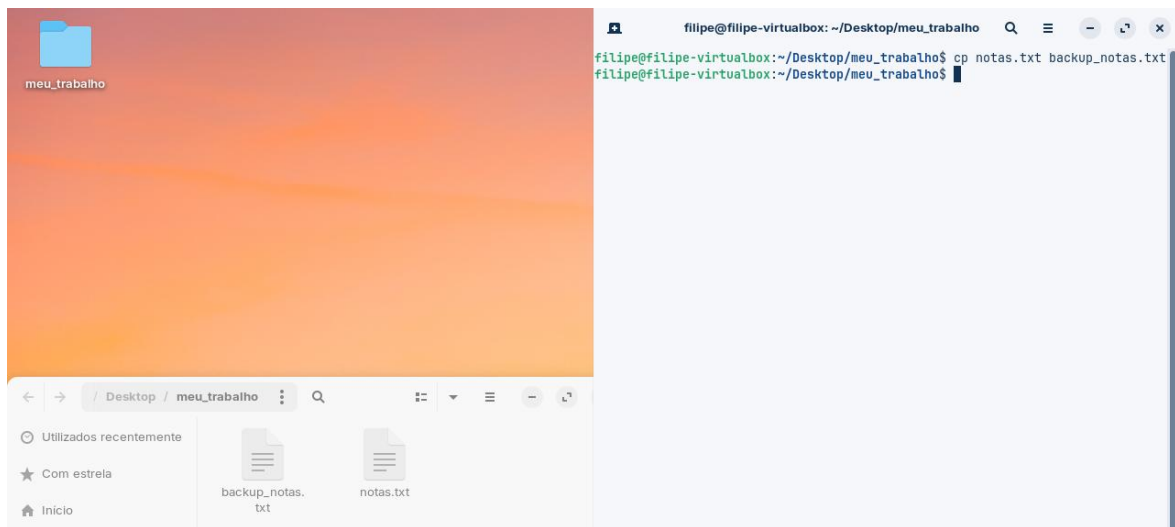


Imagem 9 - `cp notas.txt backup_notas.txt`

Após criar o ficheiro de backup, utilizei o comando **cat backup_notas.txt** para verificar se a cópia foi realizada corretamente e se o conteúdo corresponde ao ficheiro original.

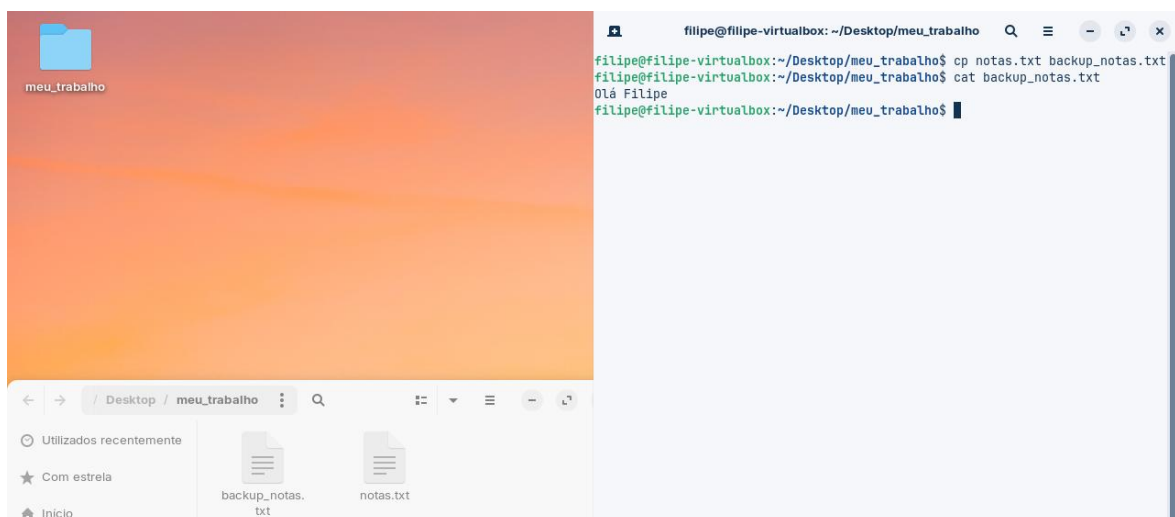


Imagem 10 - `cat backup_notas.txt`

1.4 Mover “*backup_notas.txt*” para um diretório chamado “*backup*”.

Antes de mover o ficheiro ‘*backup_notas.txt*’, foi necessário criar um diretório para armazená-lo. Para isso, utilizei o comando ***mkdir backup***, que criou a pasta ‘*backup*’ dentro do diretório atual.

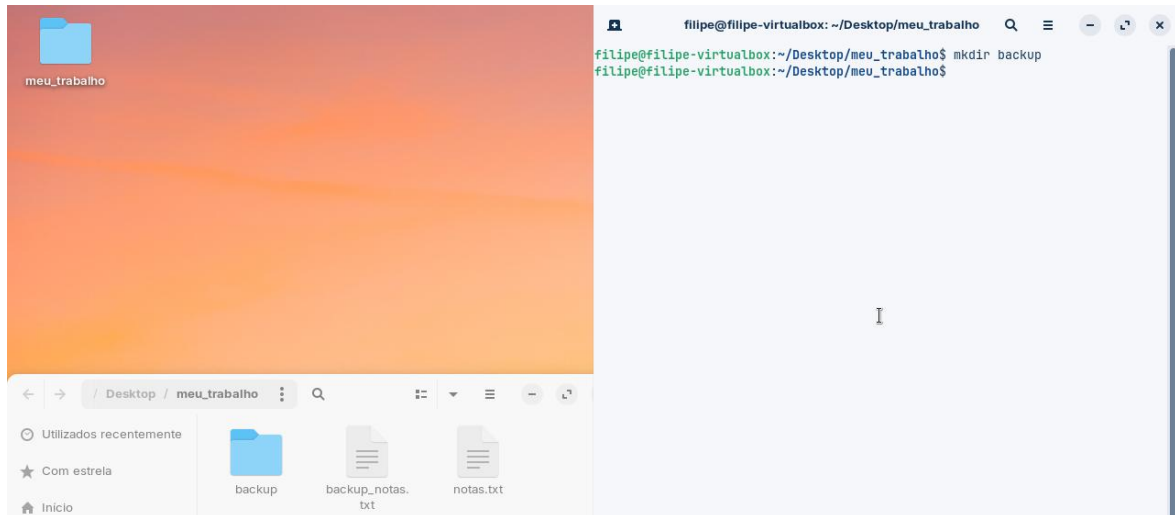


Imagem 11 – *mkdir backup*

Com a pasta ‘*backup*’ criada, executei o comando ***mv backup_notas.txt backup/*** para mover o ficheiro ‘*backup_notas.txt*’ para dentro desse diretório. O comando *mv* (move) permite reorganizar ficheiros e diretórios, alterando a sua localização no sistema.

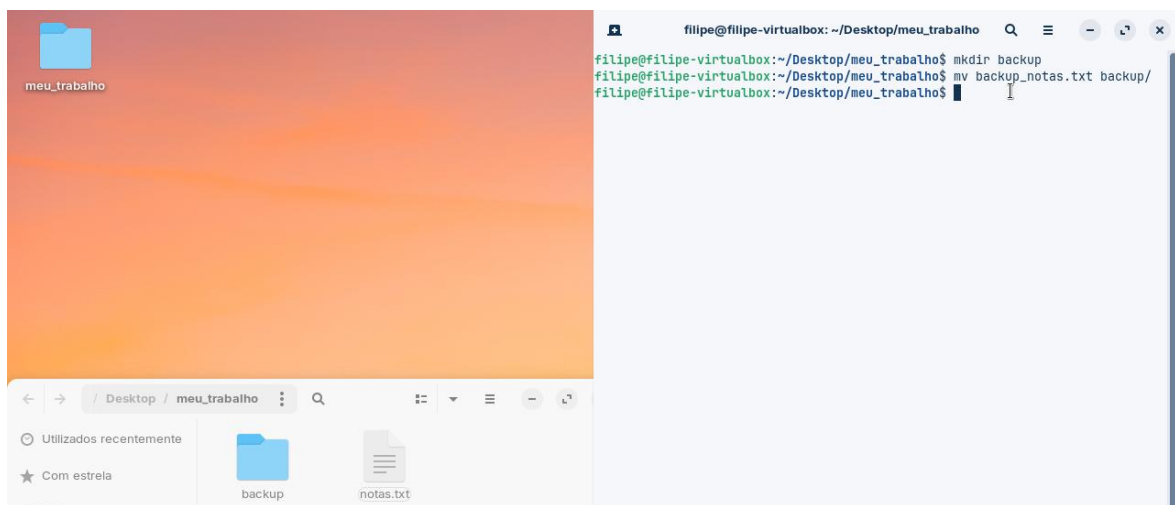


Imagem 12 - *mv backup_notas.txt backup/*

Para confirmar que o ficheiro tinha sido movido com sucesso para a pasta 'backup', utilizei o comando **`ls backup/`**, que lista o conteúdo desse diretório.

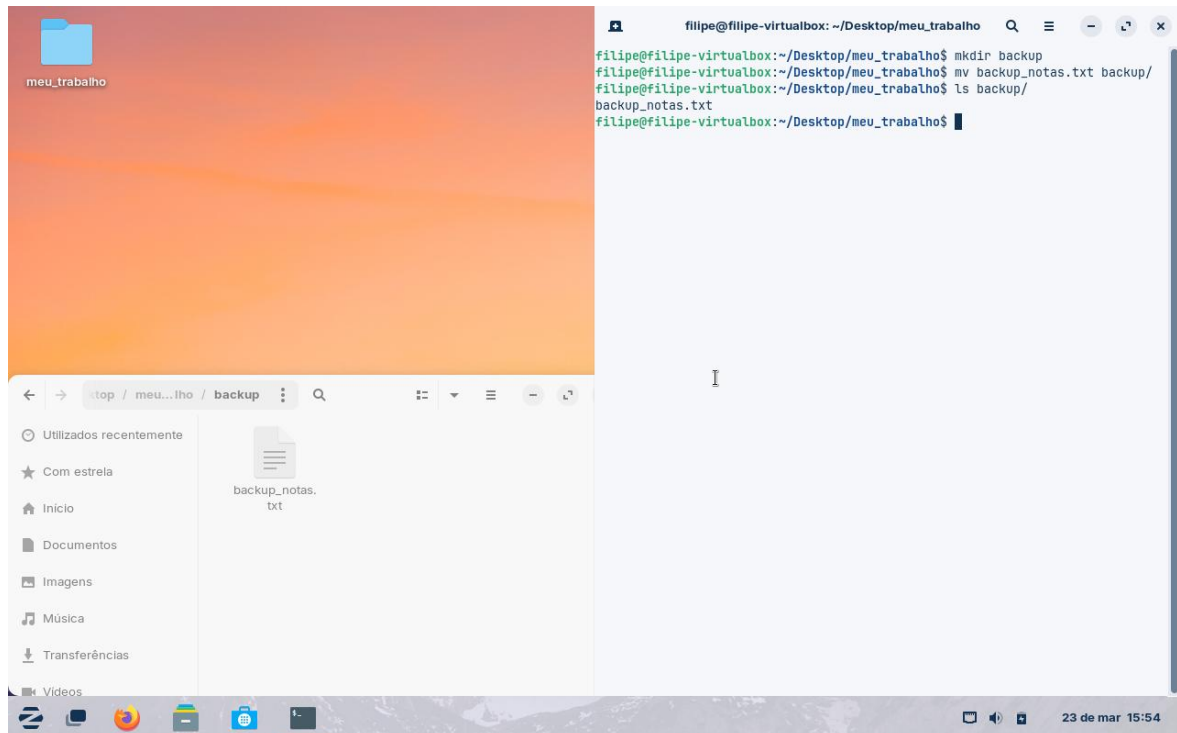


Imagem 13 – ls backup/

1.5 Listar os ficheiros do diretório “meu_trabalho” e do diretório “backup”.

Para listar os ficheiros e pastas dentro do diretório ‘meu_trabalho’, utilizei o comando `ls`. Como resultado, foram exibidos o ficheiro ‘notas.txt’ e a pasta ‘backup’

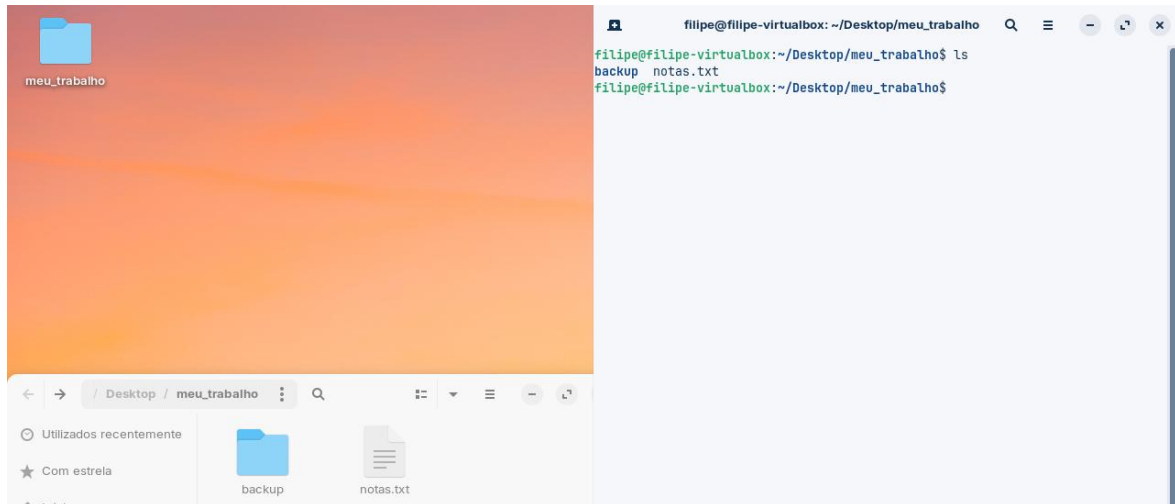


Imagem 14 – ls

Para listar os ficheiros no diretório ‘backup’, utilizei o mesmo comando, mas desta vez indiquei o caminho completo do diretório: `ls backup/`. O resultado mostrou apenas o ficheiro ‘backup_notas.txt’, conforme apresentado no passo anterior 1.4.

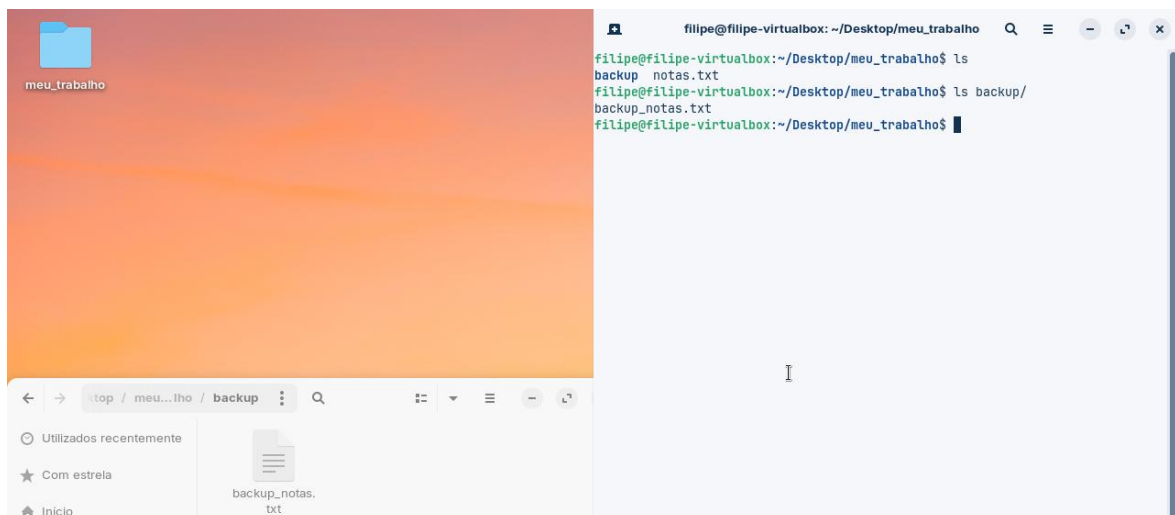


Imagem 15 – ls backup/

1.6 Apagar o ficheiro “*notas.txt*” e depois restaurá-lo a partir do “*backup*”.

Neste passo, utilizei o comando `rm notas.txt` para eliminar o ficheiro ‘notas.txt’ do diretório atual.

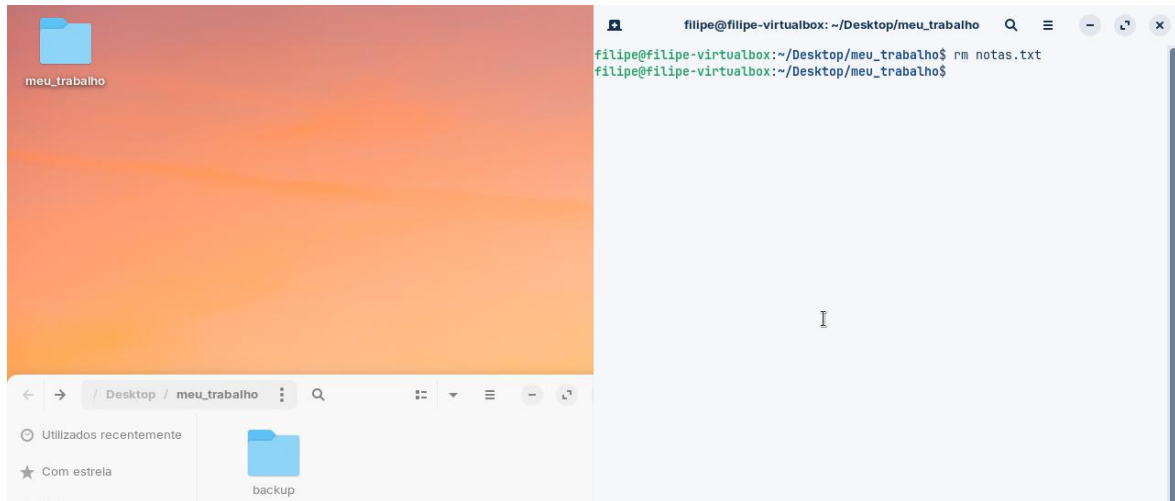


Imagem 16 – `rm notas.txt`

Utilizei o comando `ls notas.txt` para verificar se o ficheiro tinha sido eliminado. Este comando permitiu confirmar a ausência do ficheiro ‘notas.txt’ no diretório atual, uma vez que, após a sua remoção, o sistema não retornou qualquer resultado.

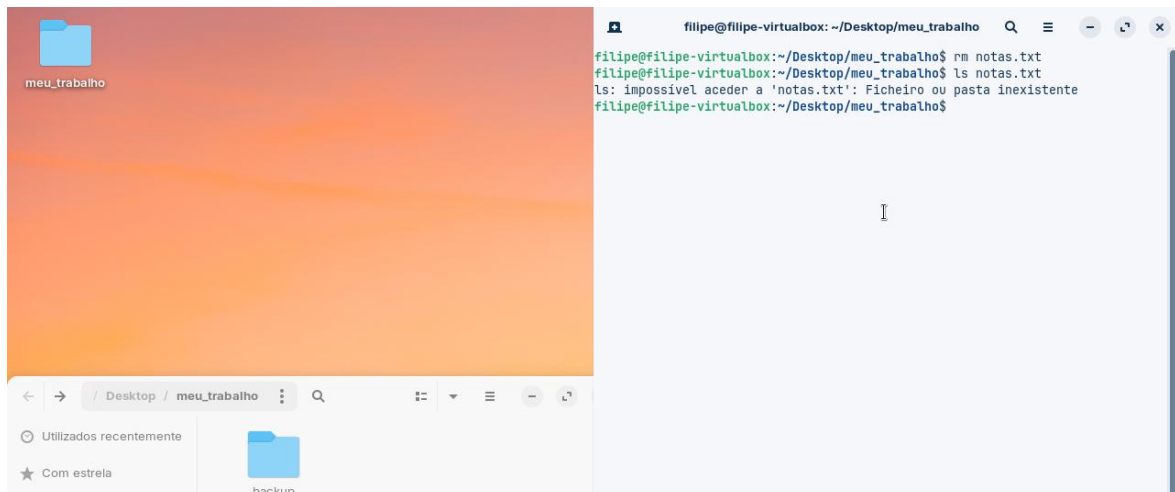


Imagem 17 – `ls notas.txt`

Para realizar o backup do ficheiro que acabei de remover, utilizei o comando **cp**. O ficheiro de backup, criado anteriormente no ponto **1.3**, que contém as informações do ficheiro removido, foi copiado de volta para o diretório atual através do código **cp backup/backup_notas.txt notas.txt**, restaurando assim o ficheiro original com o nome 'notas.txt'.

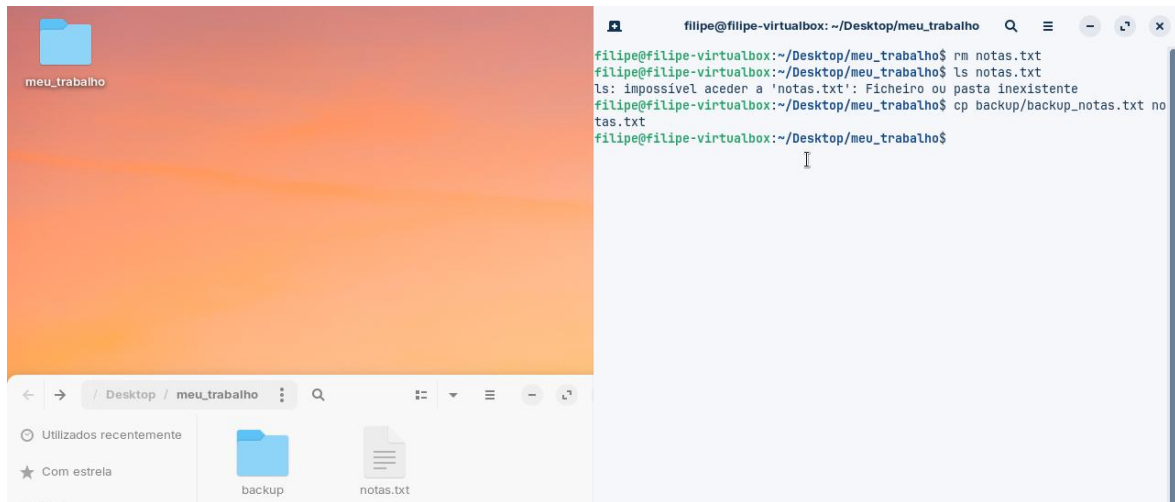


Imagem 18 - `cp backup/backup_notas.txt notas.txt`

Para verificar se o backup foi feito com sucesso, executei o comando **ls -l notas.txt**, que retornou a seguinte saída:

```
-rw-rw-r-- 1 filipe filipe 12 mar 23 16:06 notas.txt
```

Esta saída significa o seguinte:

- **-rw-rw-r--**: São as permissões do ficheiro, onde:
 - O primeiro caractere (-) indica que é um ficheiro regular.
 - **rw-** (para o proprietário) significa que o proprietário tem permissões de leitura e escrita.
 - **rw-** (para o grupo) significa que os membros do grupo têm permissões de leitura e escrita.
 - **r--** (para outros) significa que todos os outros utilizadores têm apenas permissão de leitura.
- **1**: Indica o número de links para o ficheiro (neste caso, apenas um).
- **filipe**: O proprietário do ficheiro.
- **filipe**: O grupo associado ao ficheiro.
- **12**: O tamanho do ficheiro em bytes.

- **mar 23 16:06**: A data e hora em que o ficheiro foi modificado pela última vez.
- **notas.txt**: O nome do ficheiro.

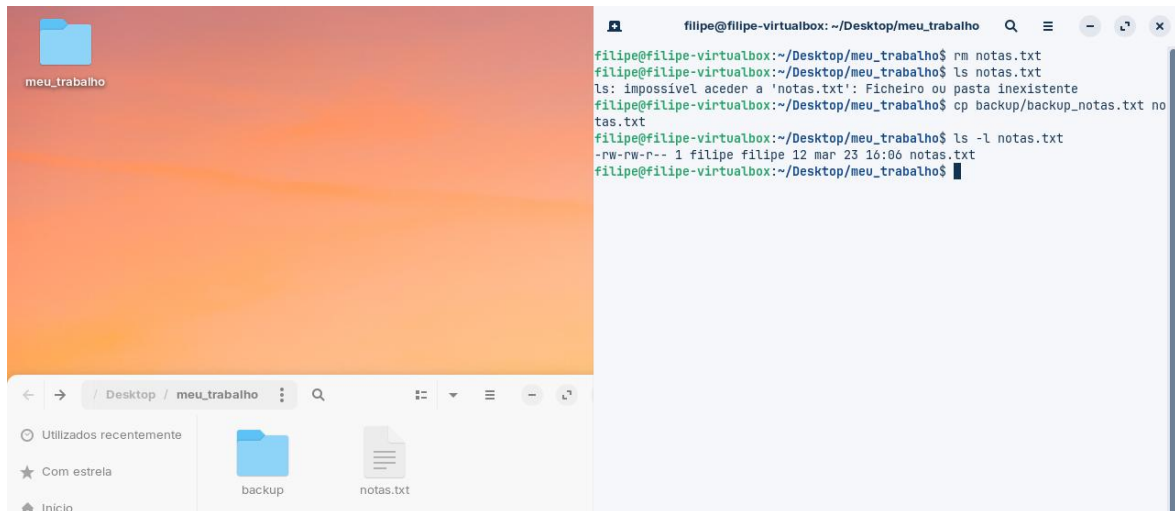


Imagem 19 – `ls -l notas.txt`

Por fim, neste ponto, executei o comando `cat notas.txt` para exibir o conteúdo do ficheiro. Este comando mostrou o conteúdo armazenado em 'notas.txt', permitindo verificar se o ficheiro foi restaurado corretamente com as informações esperadas.

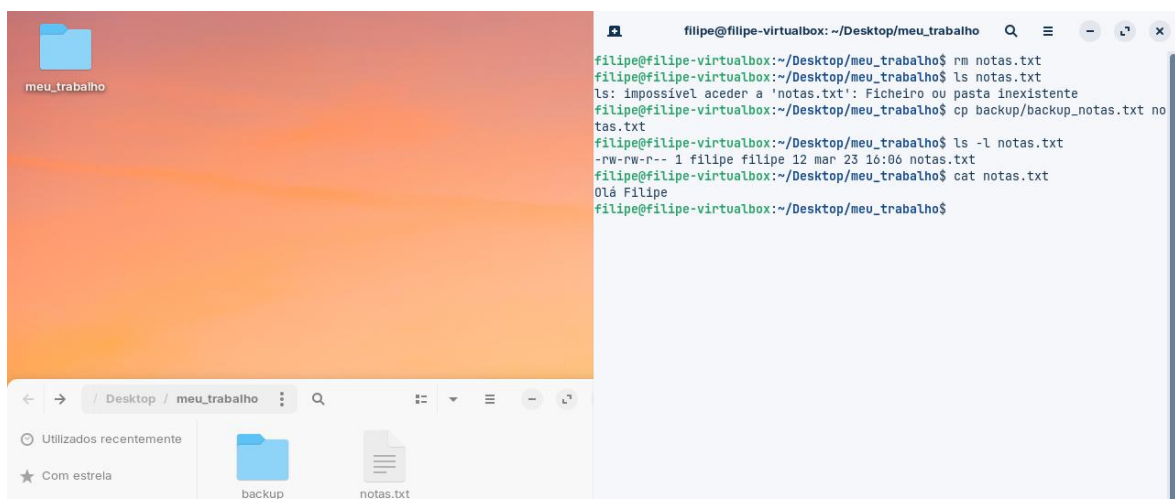


Imagem 20 – `cat notas.txt`

2. Manipulação e Extração de Informação em Ficheiros de Texto

2.1 Criar um ficheiro `"dados.txt"` com pelo menos 10 linhas de texto.

Para criar o ficheiro, utilizei o comando `cat > dados.txt << EOF`, que me permitiu redigir diretamente o conteúdo no ficheiro 'dados.txt'. Depois de inserir as linhas de texto necessárias, finalizei a operação com o comando `EOF`, o que indicou o fim da entrada de dados.

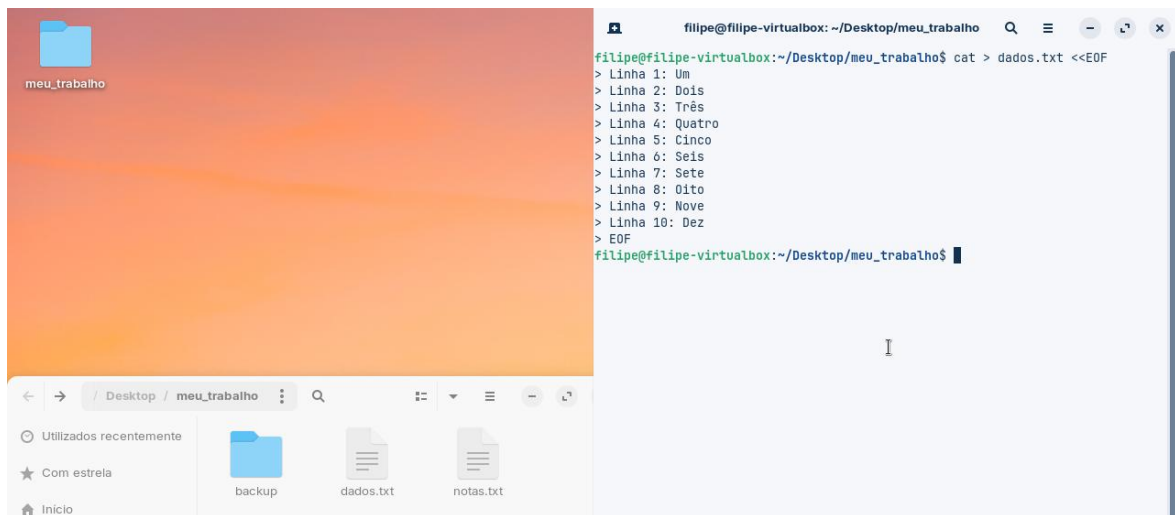


Imagem 21 – `cat > dados.txt << EOF`

2.2 Exibir as primeiras 5 *linhas* do ficheiro.

Após ter criado o ficheiro no passo anterior 2.1, executei o comando **`head -n 5 dados.txt`**, que exibiu as primeiras 5 linhas do ficheiro 'dados.txt'.

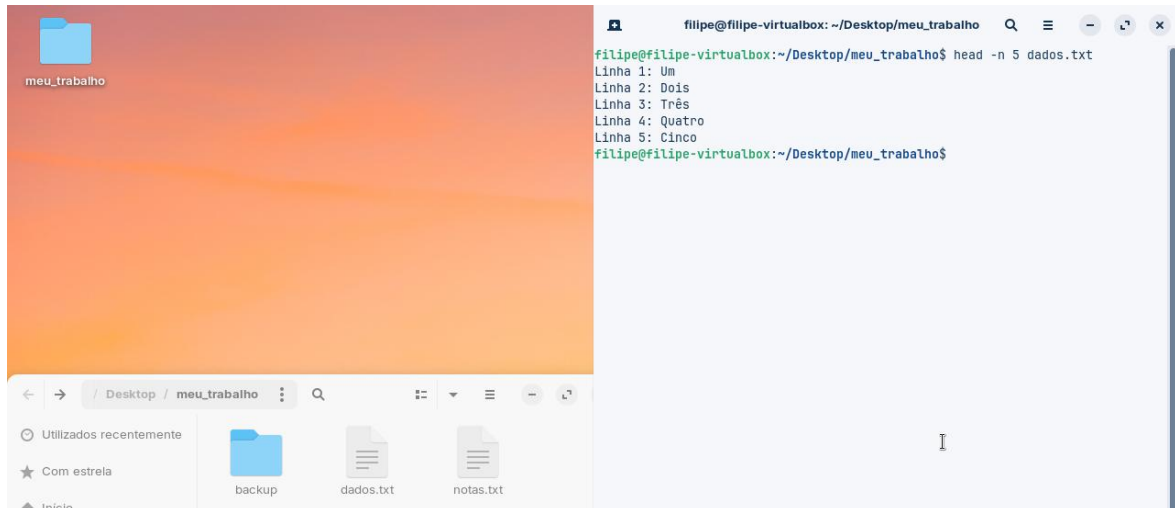


Imagem 22 – `head -n 5 dados.txt`

2.3 Exibir as últimas 3 *linhas* do ficheiro.

Semelhante ao comando anterior, mas com uma abordagem diferente, utilizei o comando **`tail -n 3 dados.txt`**, que exibiu as últimas 3 linhas do ficheiro 'dados.txt'.

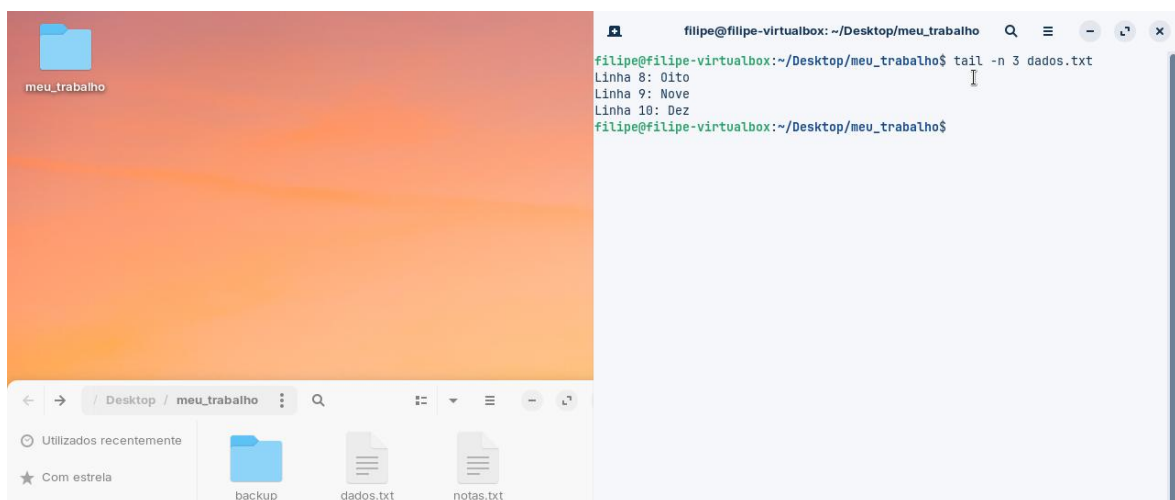


Imagem 23 – `tail -n 3 dados.txt`

2.4 Contar o número de *linhas*, *palavras* e *caracteres* do ficheiro.

Utilizei o comando `wc dados.txt`, que retornou o seguinte resultado:

```
10 30 142 dados.txt
```

Este resultado significa:

- **10**: O número de linhas no ficheiro "dados.txt".
- **30**: O número de palavras no ficheiro.
- **142**: O número de caracteres no ficheiro.

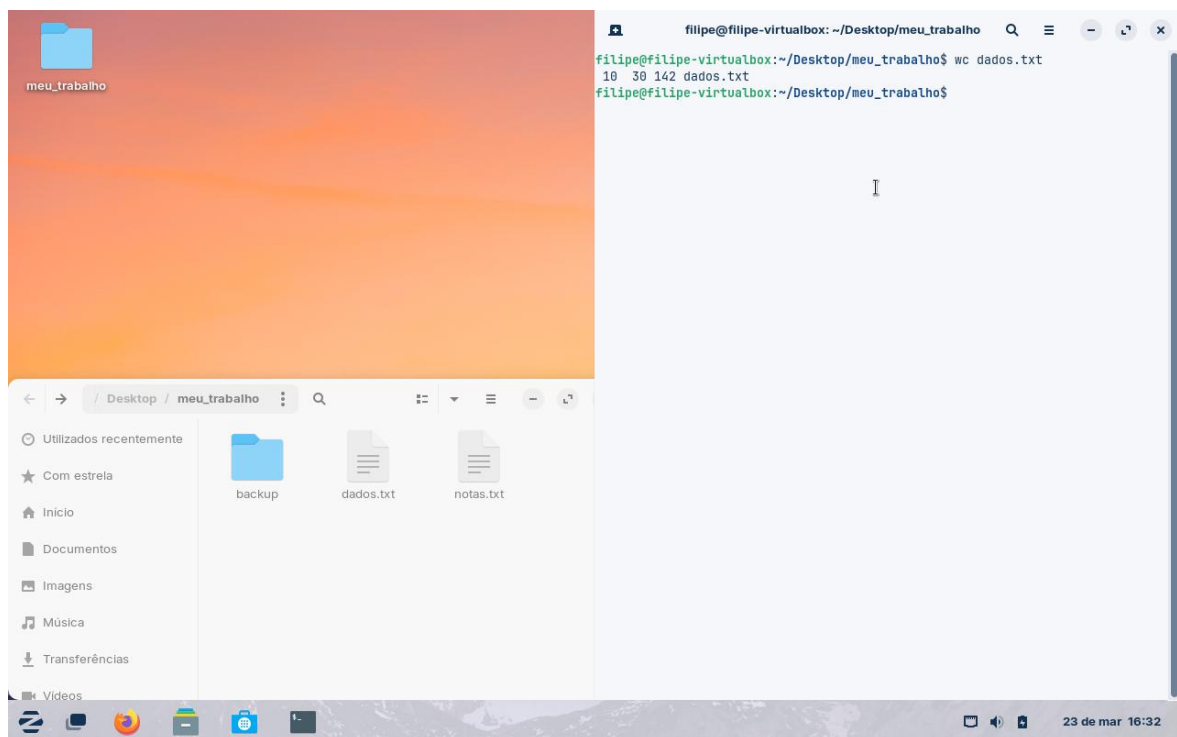


Imagem 24 – `wc dados.txt`

2.5 Procurar e exibir todas as linhas que contenham uma determinada palavra.

Como eu tinha escrito no ficheiro com palavras simples e as linhas não tinham nenhuma palavra repetida, exceto a palavra 'linha', precisei editar o ficheiro para garantir que ele atendesse ao critério. Para isso, utilizei o comando `nano dados.txt`, que abriu o editor de texto *nano* no terminal.

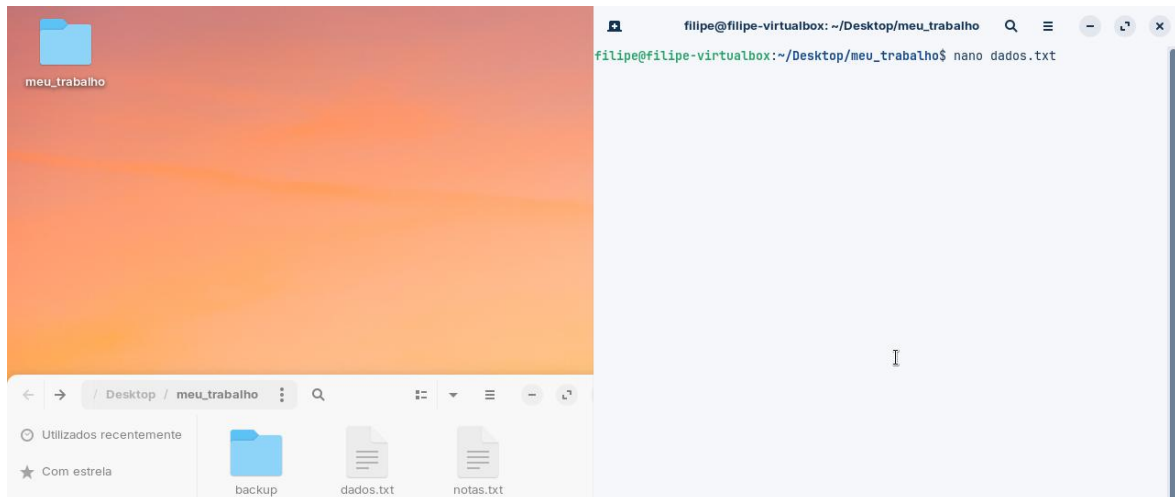


Imagem 25 – nano dados.txt

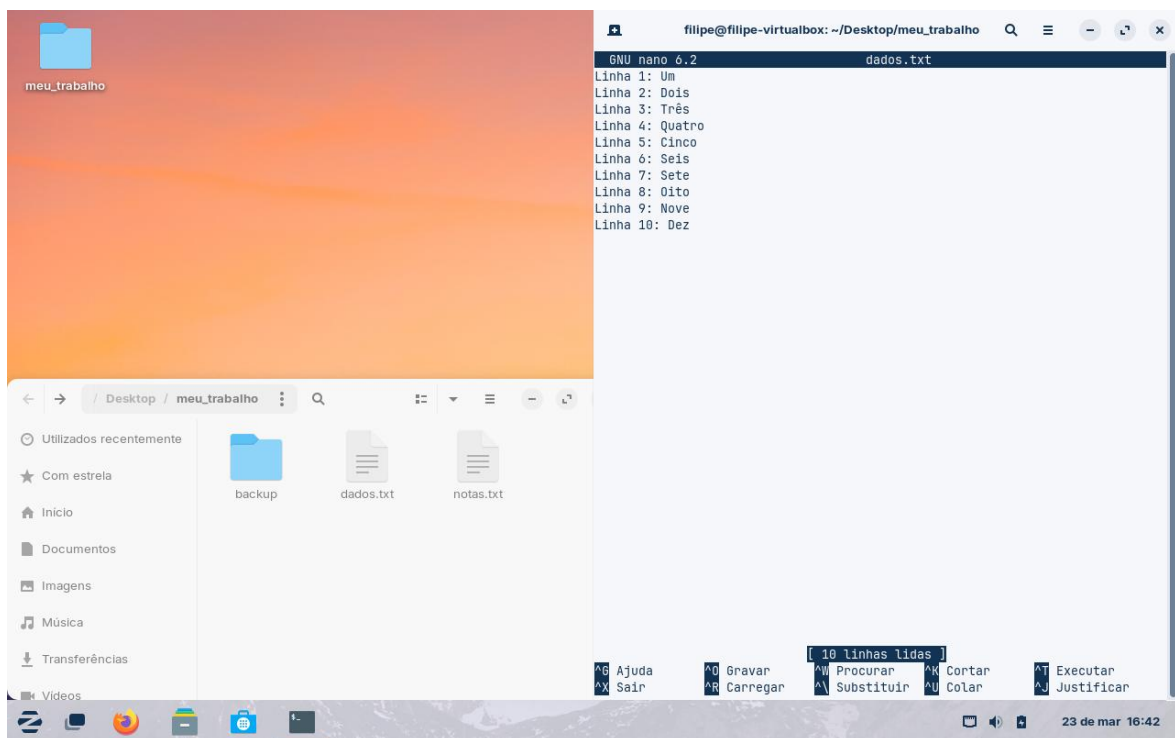


Imagem 26 – editor nano na linha de comandos

Com isso, adicionei a palavra ‘**Ornitorrinco**’ às linhas **1, 4 e 9** do ficheiro. Após inserir as palavras nas linhas desejadas, pressionei **Ctrl + O** para salvar as edições realizadas no ficheiro.

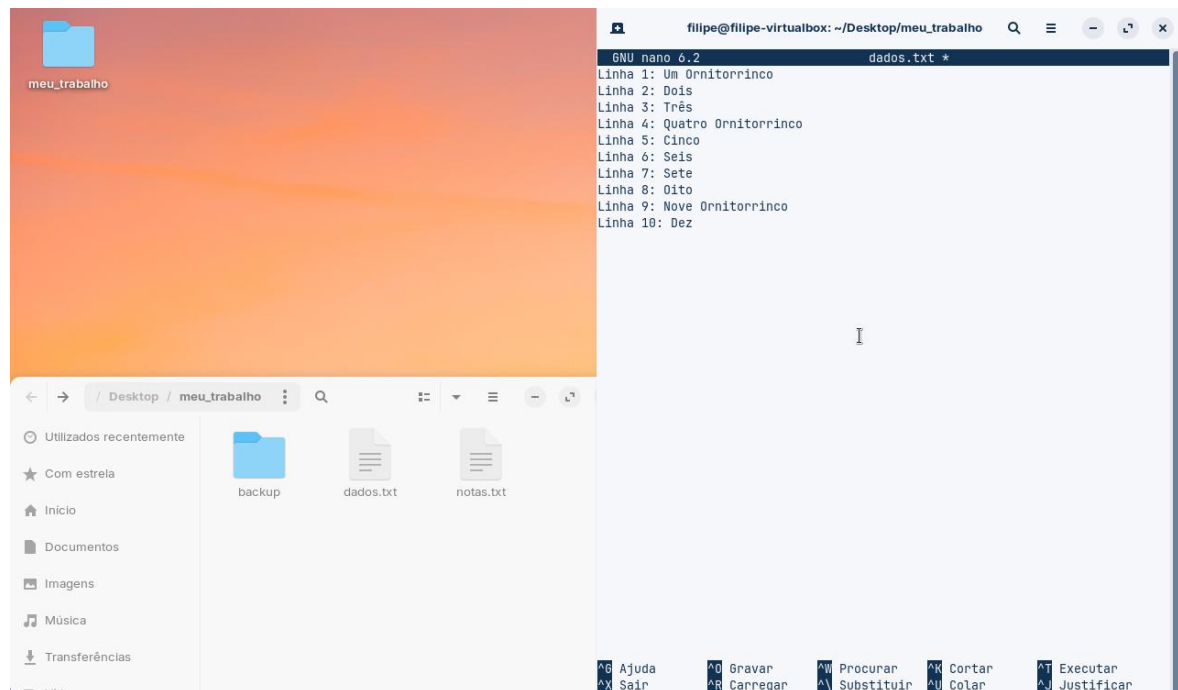


Imagem 27 – adição da palavra ornitorrinco

O editor perguntou qual seria o nome do ficheiro onde guardar as alterações, e eu pressionei apenas **Enter** para confirmar que queria editar o ficheiro existente, em vez de criar um novo. Isso fez com que as alterações fossem salvas no próprio ficheiro ‘**dados.txt**’.

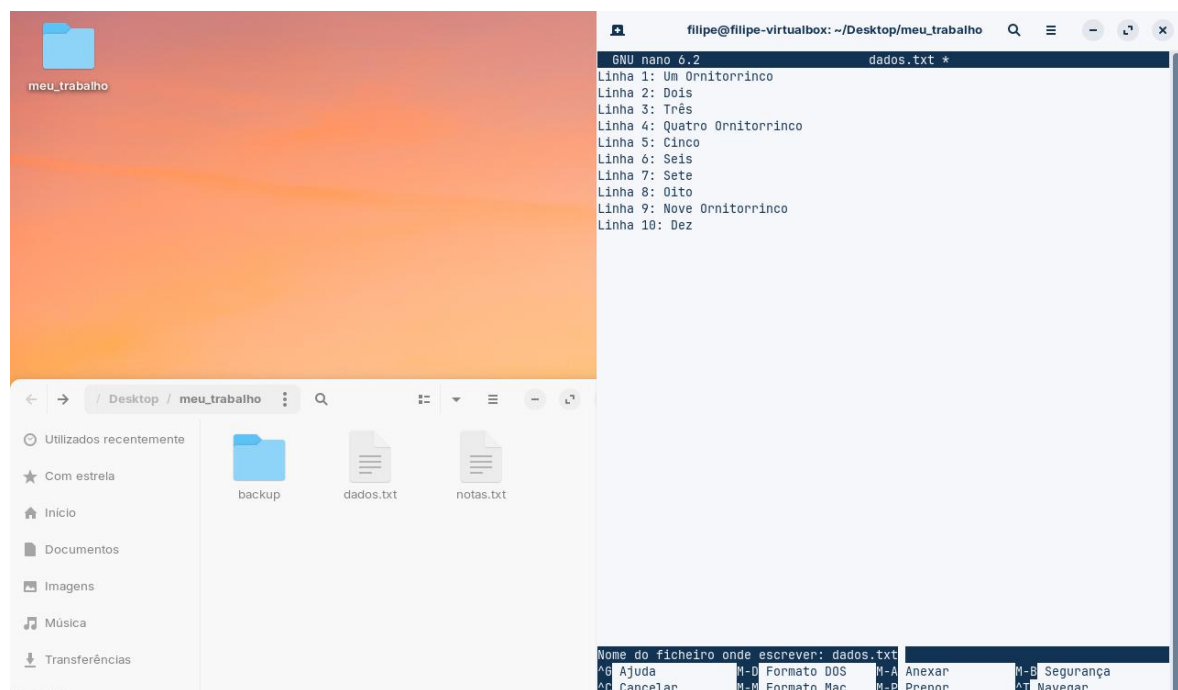


Imagem 28 – confirmação de edição

Após editar e guardar as alterações, pressionei **Ctrl + X** para sair do editor **nano** e voltar ao terminal.

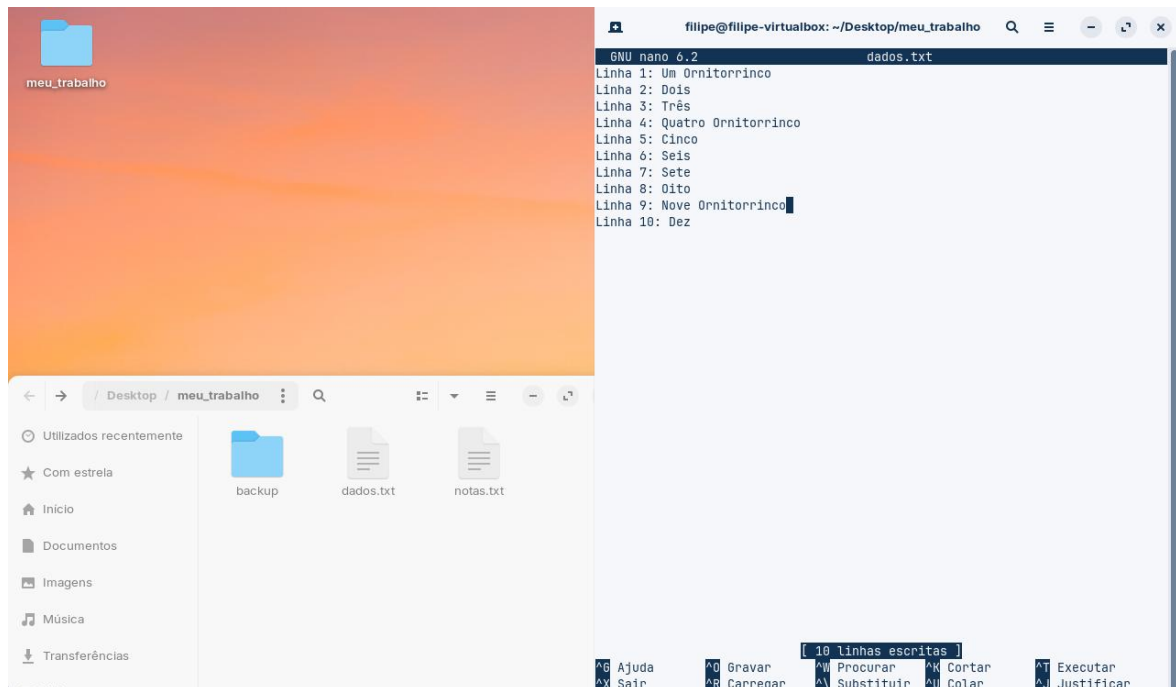


Imagem 29 – ctrl + x para voltar para o terminal

Para verificar se a edição foi concluída e bem-sucedida, executei o comando **cat dados.txt**, que exibiu o conteúdo atualizado do ficheiro 'dados.txt'

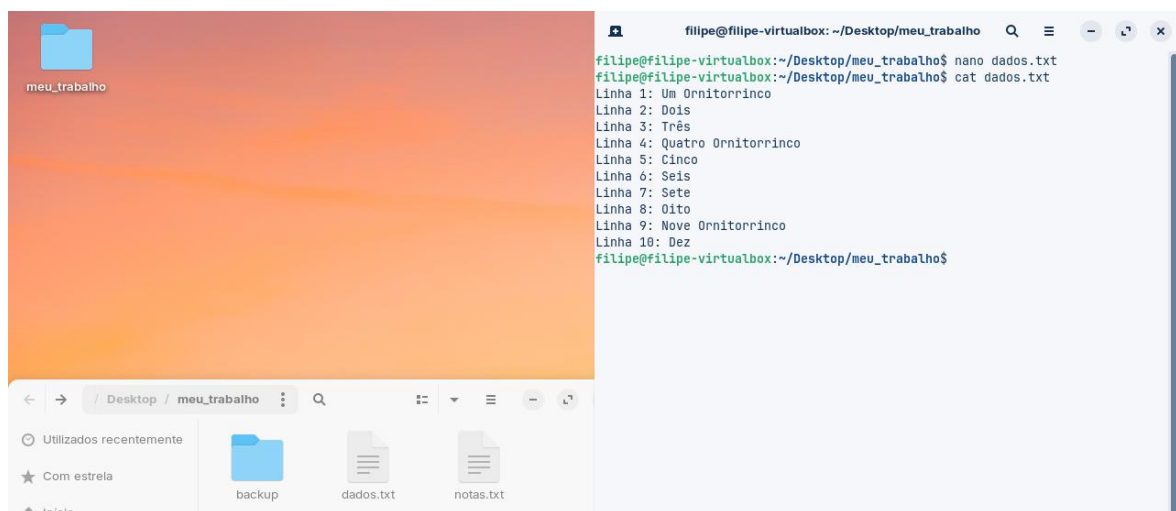


Imagem 30 – cat dados.txt depois da alteração

E, por fim, neste ponto, utilizei o comando **`grep "Ornitorrinco" dados.txt`**, que procurou e exibiu todas as linhas do ficheiro '**`dados.txt`**' que contêm a palavra '**`Ornitorrinco`**'.

Esse comando permitiu verificar se a palavra foi corretamente inserida nas linhas **1, 4 e 9**.

A palavra '**`Ornitorrinco`**' apareceu a **vermelho**, indicando que o comando **`grep`** encontrou as ocorrências da palavra no ficheiro e as destacou, facilitando a confirmação de que a edição foi realizada com sucesso.

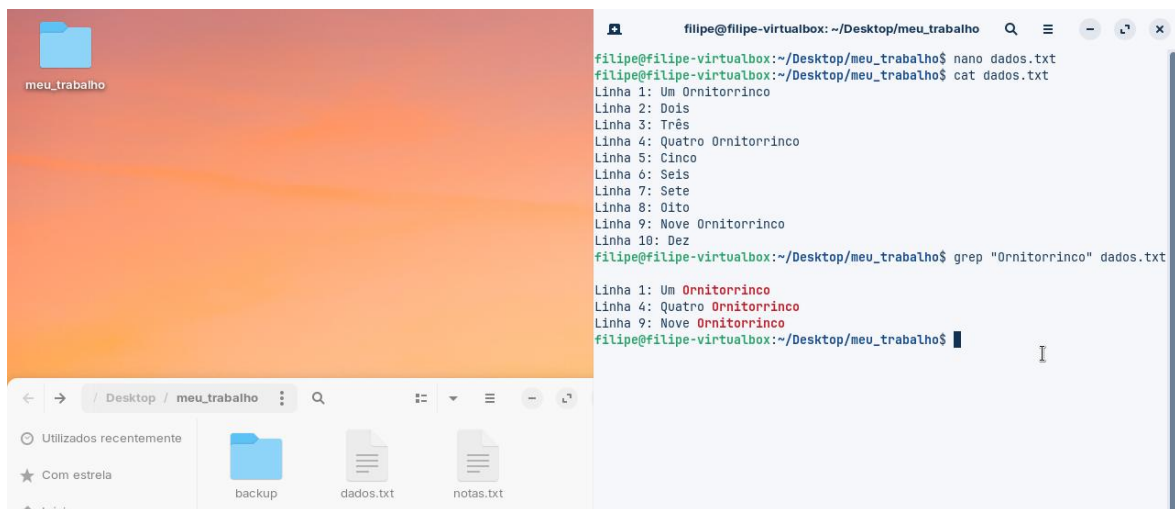


Imagem 31 – `grep "Ornitorrinco" dados.txt`

2.6 Ordenar as linhas do ficheiro *alfabeticamente* e guardar o resultado num novo ficheiro **“ordenado.txt”**.

Utilizei o comando `sort dados.txt > ordenado.txt`, que ordenou as linhas do ficheiro ‘dados.txt’ de forma alfabética e guardou o resultado no novo ficheiro ‘ordenado.txt’.

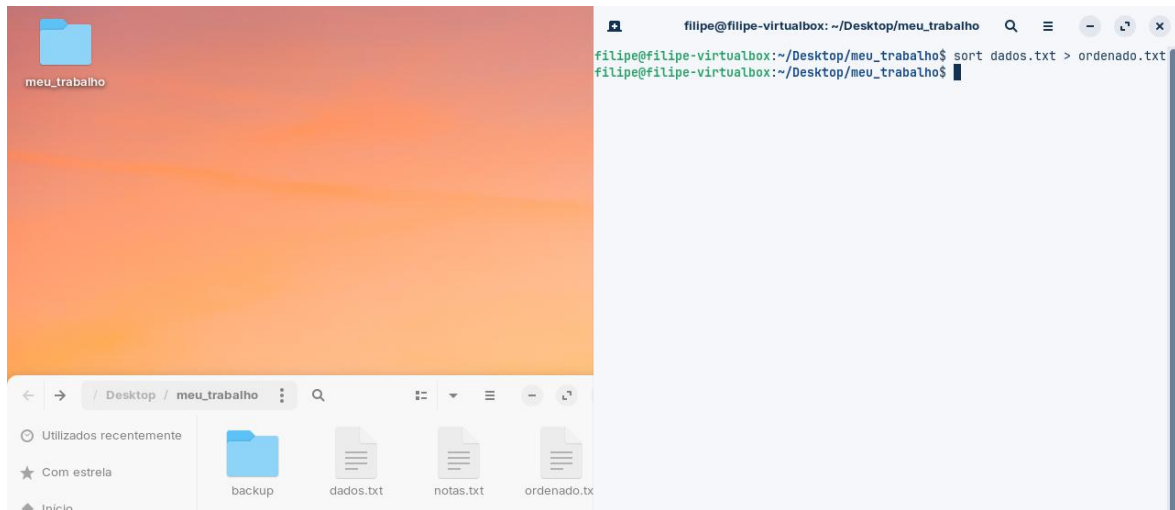


Imagem 32 – `sort dados.txt > ordenado.txt`

A seguir, executei o comando `cat ordenado.txt` para exibir o conteúdo do ficheiro ‘ordenado.txt’ e verificar se as linhas foram ordenadas corretamente de forma alfabética.

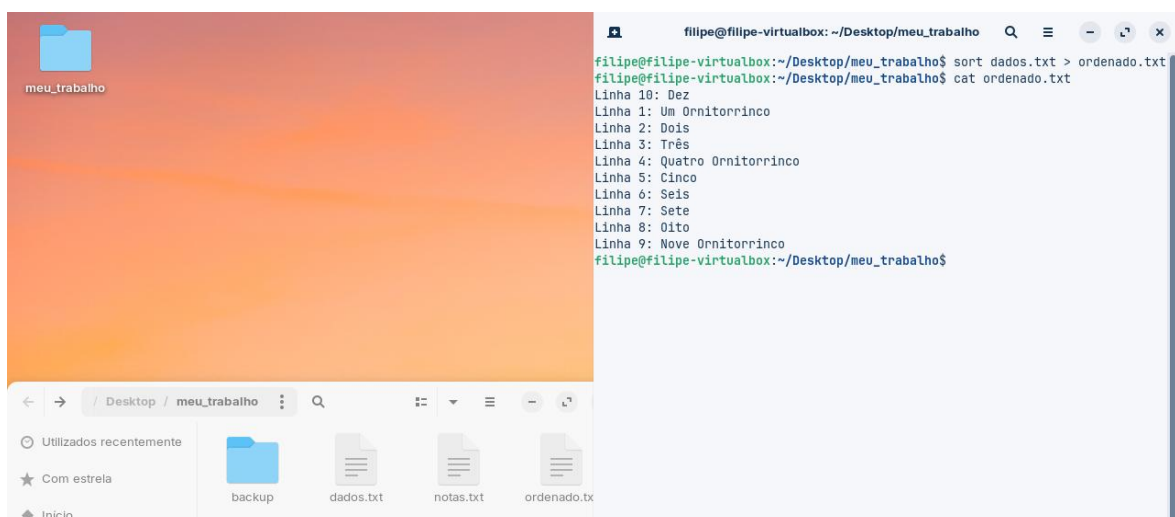


Imagem 33 – `cat ordenado.txt`

3. Redireccionamento e Pipes

3.1 Redirecionar a saída do comando “ls -l” para um ficheiro “lista_ficheiros.txt”.

O comando `ls -l > lista_ficheiros.txt` foi usado para redirecionar a saída do comando `ls -l`, que lista os ficheiros e diretórios com detalhes, para o ficheiro ‘lista_ficheiros.txt’.

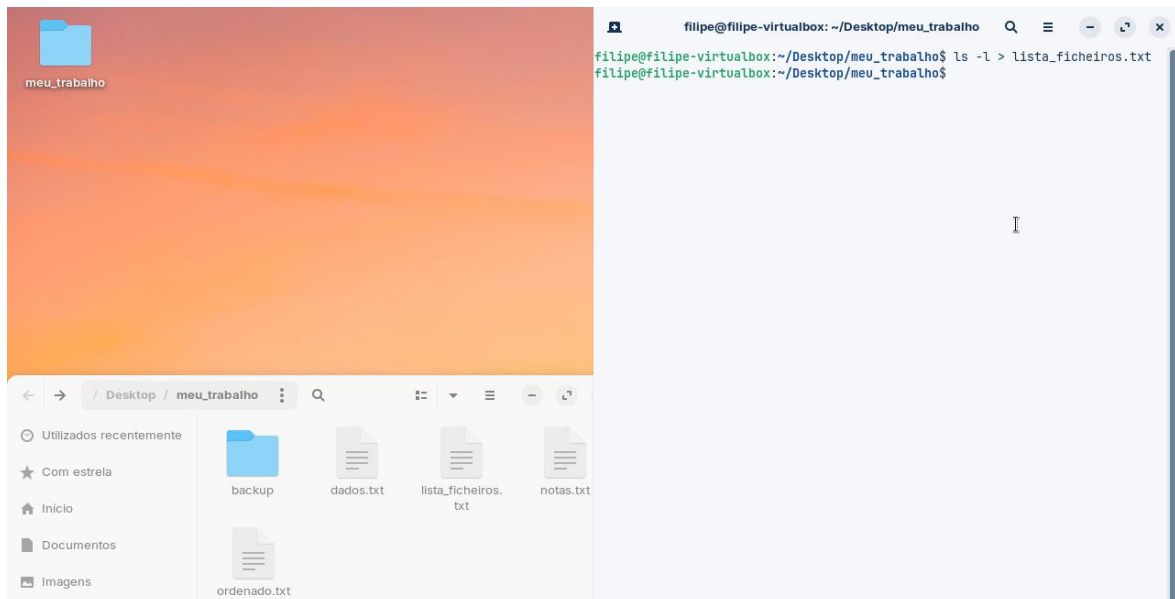


Imagem 34 – `ls -l > lista_ficheiros.txt`

A seguir, executei `cat lista_ficheiros.txt` para verificar a saída redirecionada do comando `ls`.

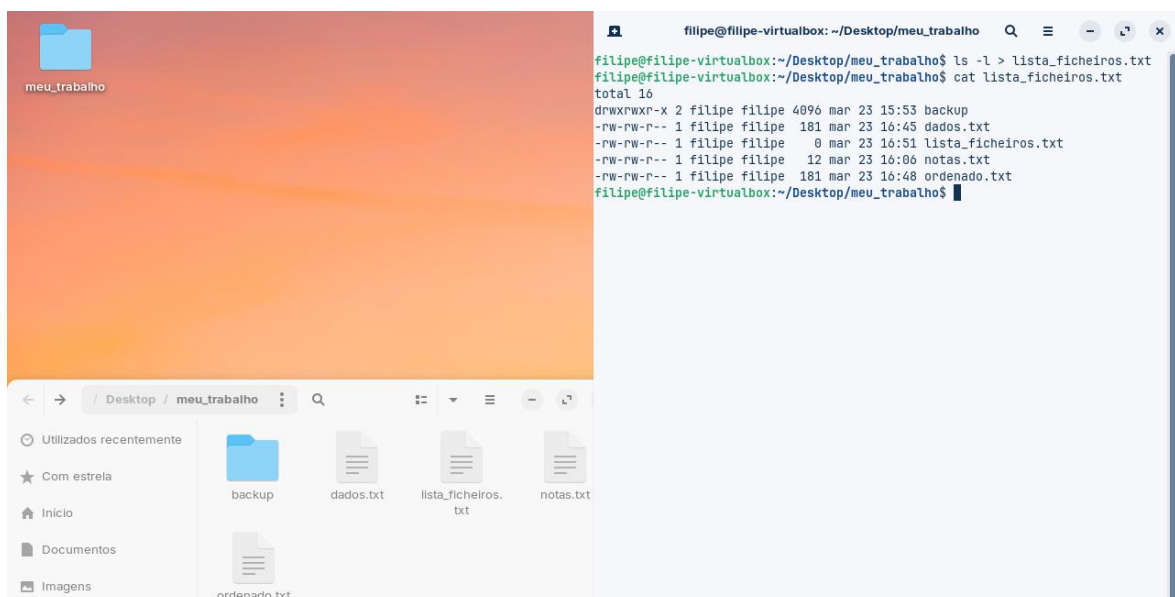


Imagem 35 – `cat lista_ficheiros.txt`

3.2 Concatenar os ficheiros “dados.txt” e “ordenado.txt” e guardar o resultado em “final.txt”.

O comando `cat dados.txt ordenado.txt > final.txt` foi usado para concatenar o conteúdo dos ficheiros ‘dados.txt’ e ‘ordenado.txt’. Através deste comando, o conteúdo de ambos os ficheiros foram combinados em sequência e, em seguida, redirecionado para o ficheiro ‘final.txt’.

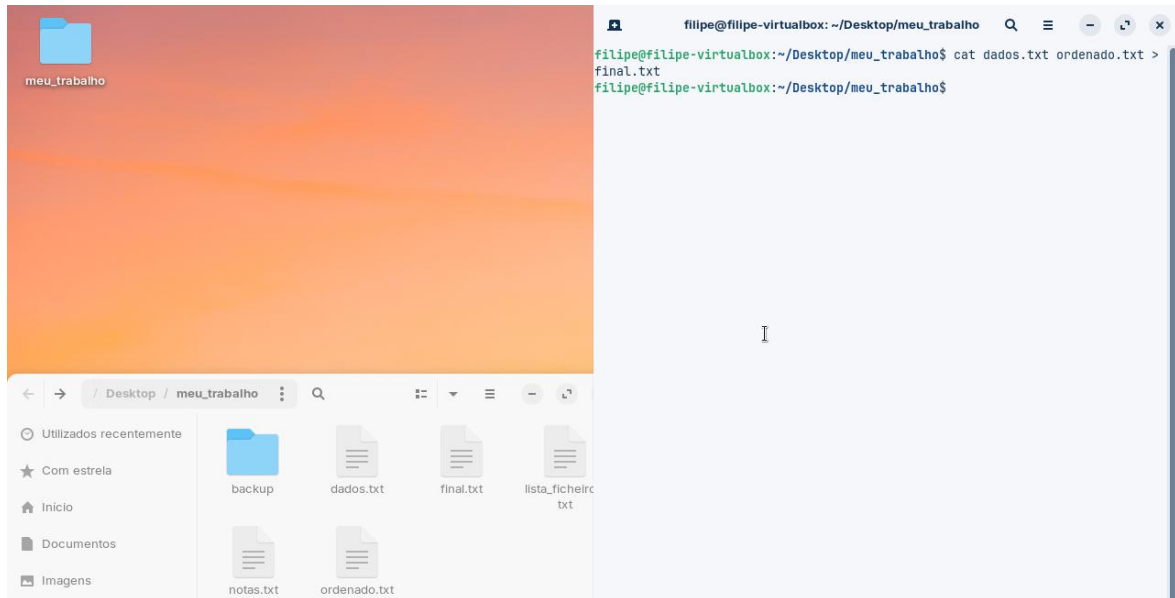


Imagem 36 - `cat dados.txt ordenado.txt > final.txt`

A seguir, executei o comando `cat final.txt` para exibir o conteúdo do ficheiro ‘final.txt’ e verificar se a concatenação dos ficheiros ‘dados.txt’ e ‘ordenado.txt’ foi realizada corretamente.

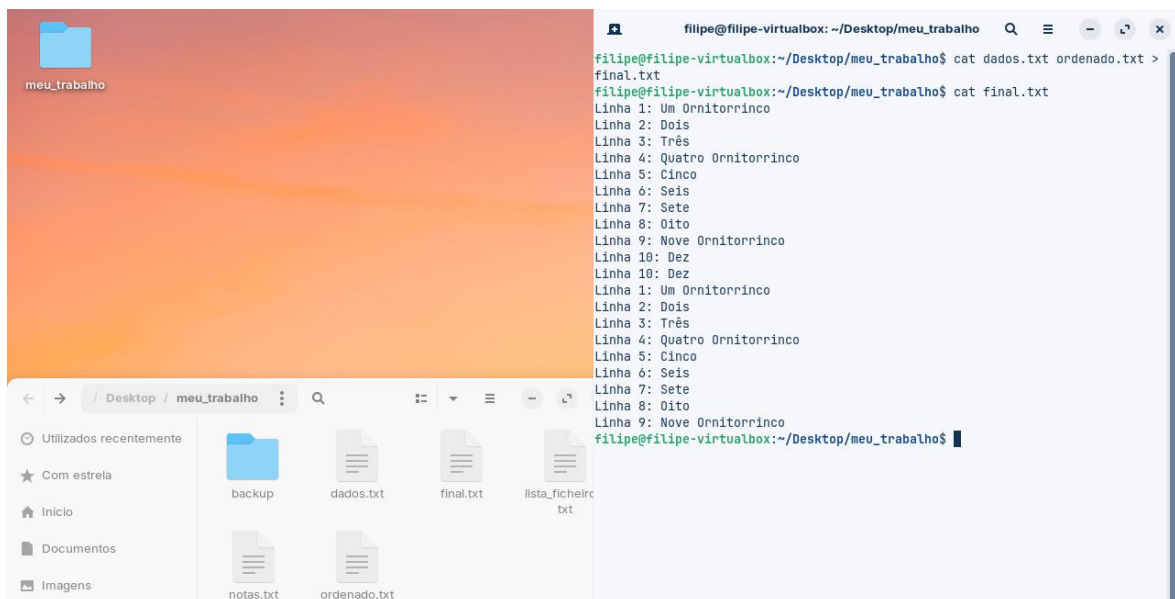


Imagem 37 – `cat final.txt`

3.3 Contar quantas vezes uma palavra aparece no ficheiro

“final.txt”.

Para contar quantas vezes a palavra ‘Ornitorrinco’ aparece no ficheiro ‘final.txt’, utilizei o comando:

```
grep -o "Ornitorrinco" final.txt | wc -l
```

O comando **grep -o "Ornitorrinco" final.txt** procura todas as ocorrências da palavra ‘Ornitorrinco’ no ficheiro ‘final.txt’.

A opção **-o** faz com que o **grep** “imprima” apenas a palavra ‘Ornitorrinco’ sempre que ela aparece, colocando cada ocorrência numa nova linha, mesmo que a palavra se repita na mesma linha do ficheiro. No entanto, essas ocorrências **não são exibidas no terminal** devido à utilização da **pipe (|)**, que redireciona a saída.

Em seguida, o **pipe (|)** redireciona a saída do comando **grep** para o comando **wc -l**, que conta o número de linhas geradas. Como cada ocorrência da palavra ‘Ornitorrinco’ é apresentada numa linha separada, o **wc -l** conta o total de vezes que a palavra aparece no ficheiro.

Este processo permitiu-me obter o número exato de vezes que ‘Ornitorrinco’ ocorre no ficheiro ‘final.txt’.

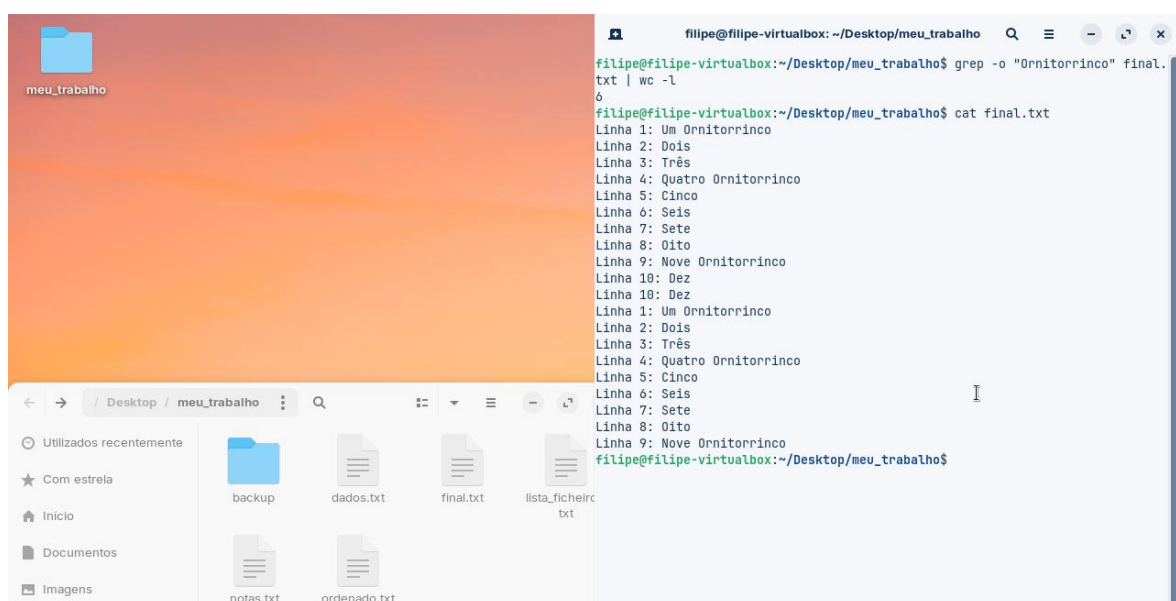


Imagem 38 – **grep -o "Ornitorrinco" final.txt | wc -l**

3.4 Filtrar e exibir apenas as linhas que contêm números em "final.txt".

Para filtrar e exibir apenas as linhas que contêm números no ficheiro "final.txt", utilizei o comando:

```
grep -E '[0-9]' final.txt
```

A opção **-E** ativa as expressões regulares estendidas no **grep**, permitindo o uso de padrões mais complexos. A expressão regular **[0-9]** corresponde a qualquer dígito numérico, ou seja, qualquer linha que contenha pelo menos um número será exibida.

Este comando procura e mostra todas as linhas do ficheiro **'final.txt'** que contêm um ou mais números. O resultado será a exibição das linhas que têm números, permitindo assim filtrar as informações que contêm dígitos.

Neste caso, todas as linhas foram exibidas, uma vez que todas elas contêm pelo menos um número.

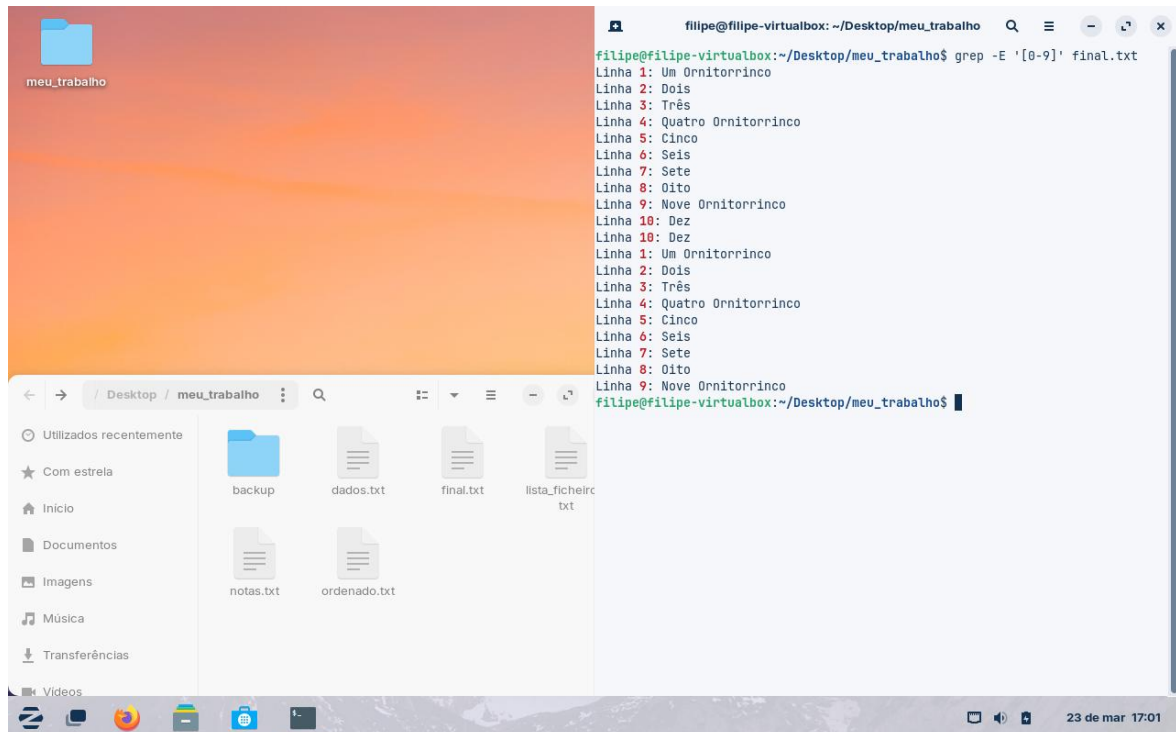


Imagem 39 - `grep -E '[0-9]' final.txt`

4. Comandos para Gestão da Máquina e Utilizadores

4.1 Exibir o nome do utilizador atualmente autenticado.

Para exibir o nome do utilizador atualmente autenticado no sistema, utilizei o comando **whoami**. Este comando retorna o nome do utilizador que iniciou a sessão no terminal, permitindo confirmar qual conta está ativa.

A screenshot of a terminal window titled 'filipe@filipe-virtualbox: ~'. The prompt is 'filipe@filipe-virtualbox:~\$'. The user has entered the command 'whoami'. The output is 'filipe'. The prompt is now 'filipe@filipe-virtualbox:~\$' with a cursor at the end.

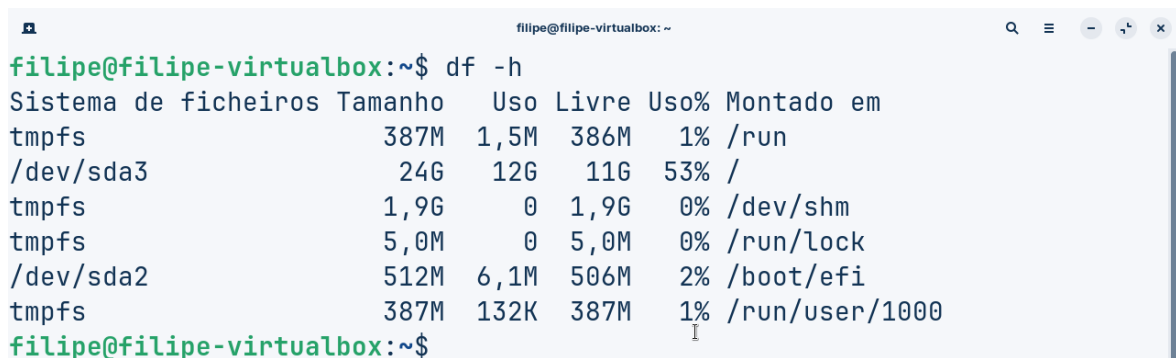
```
filipe@filipe-virtualbox:~$ whoami
filipe
filipe@filipe-virtualbox:~$
```

Imagem 40 - whoami

4.2 Mostrar informações sobre a utilização do disco.

Para visualizar informações sobre a utilização do disco, utilizei o comando **df -h**. O comando **df** (**disk free**) exibe o espaço utilizado e disponível em todos os sistemas de ficheiros montados.

A opção **-h** (**human-readable**) formata os valores para uma leitura mais intuitiva, apresentando as unidades em KB, MB ou GB, conforme necessário.



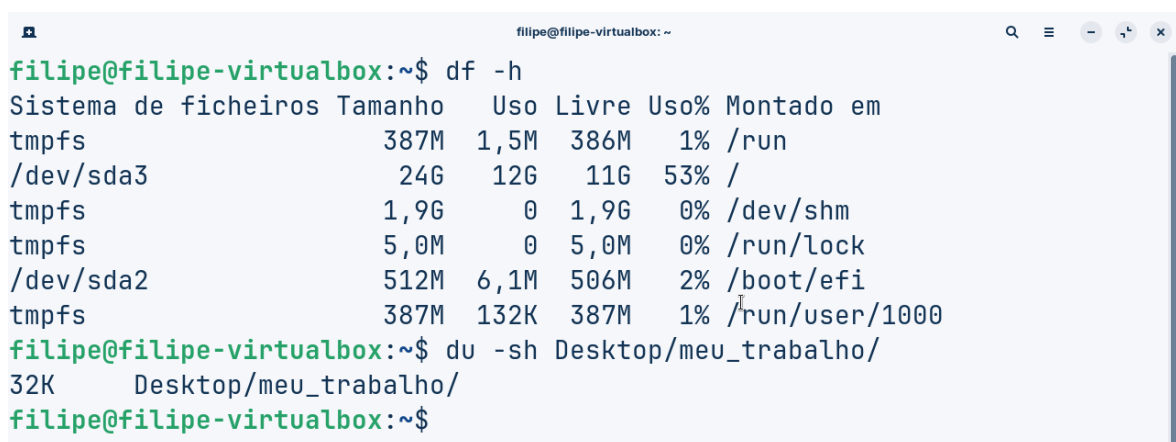
```
filipe@filipe-virtualbox: ~$ df -h
Sistema de ficheiros Tamanho  Uso  Livre  Uso%  Montado em
tmpfs                    387M  1,5M  386M   1%  /run
/dev/sda3                 24G   12G   11G  53%  /
tmpfs                    1,9G    0  1,9G   0%  /dev/shm
tmpfs                    5,0M    0  5,0M   0%  /run/lock
/dev/sda2                512M   6,1M  506M   2%  /boot/efi
tmpfs                    387M  132K  387M   1%  /run/user/1000
filipe@filipe-virtualbox: ~$
```

Imagem 41 – df -h

Achei interessante e encontrei uma forma de verificar o espaço ocupado por um diretório específico através do comando **du -sh Desktop/meu_trabalho/**.

O comando **du** (**disk usage**) exibe o espaço em disco utilizado pelos ficheiros e diretórios.

A opção **-s** (**summary**) faz com que apenas seja apresentado o tamanho total do diretório especificado, sem listar individualmente os subdiretórios



```
filipe@filipe-virtualbox: ~$ df -h
Sistema de ficheiros Tamanho  Uso  Livre  Uso%  Montado em
tmpfs                    387M  1,5M  386M   1%  /run
/dev/sda3                 24G   12G   11G  53%  /
tmpfs                    1,9G    0  1,9G   0%  /dev/shm
tmpfs                    5,0M    0  5,0M   0%  /run/lock
/dev/sda2                512M   6,1M  506M   2%  /boot/efi
tmpfs                    387M  132K  387M   1%  /run/user/1000
filipe@filipe-virtualbox: ~$ du -sh Desktop/meu_trabalho/
32K    Desktop/meu_trabalho/
filipe@filipe-virtualbox: ~$
```

Imagem 42 – du -sh Desktop/meu_trabalho/

4.3 Consultar a carga do sistema e os processos em execução.

Para consultar a carga do sistema e os processos em execução, utilizei o comando `top`.

Este comando exibe em tempo real uma lista dos processos em execução, incluindo informações como o uso de CPU, memória, tempo de execução e ID do processo.

Além disso, mostra a carga do sistema, permitindo analisar o desempenho da máquina. A interface interativa do `top` possibilita a ordenação dos processos e a finalização de tarefas diretamente pelo terminal.



Imagem 43 - top

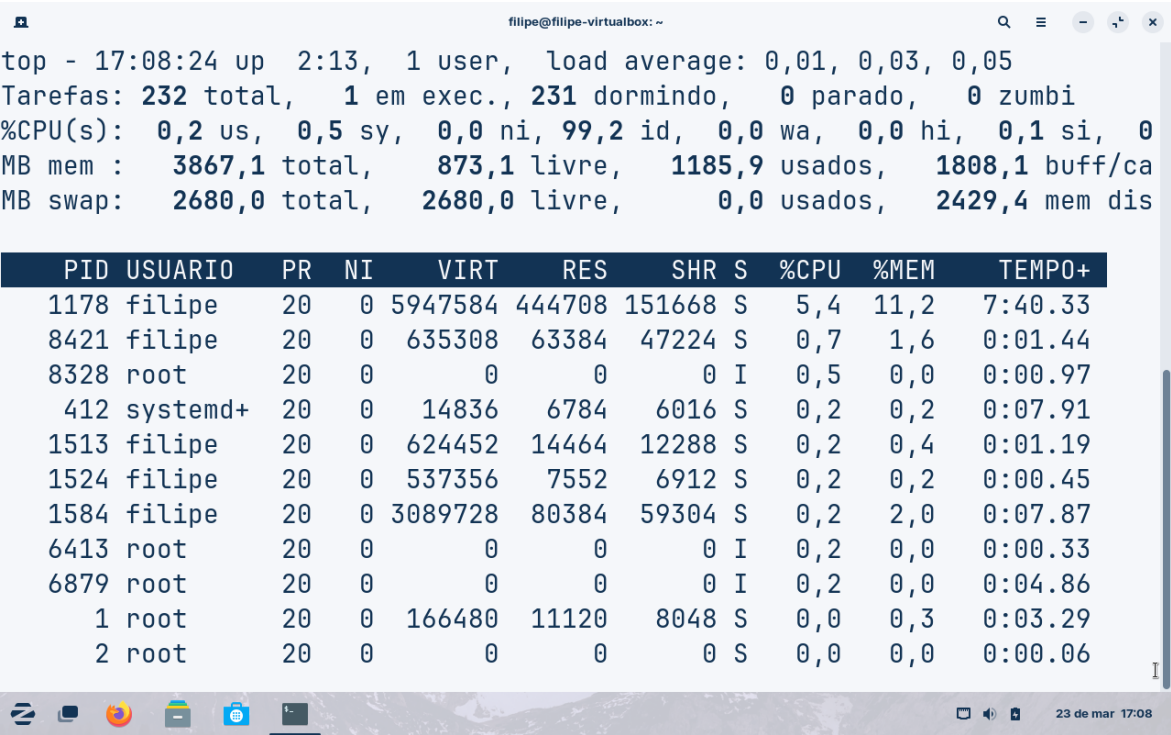


Imagem 44 – processos

Além do comando **top**, utilizei também o **uptime** para obter informações sobre o tempo de atividade do sistema.

Este comando exibe a hora atual, há quanto tempo o sistema está ligado, o número de utilizadores conectados e a carga média do sistema nos últimos 1, 5 e 15 minutos.

É útil para ter uma visão rápida do desempenho e estabilidade da máquina.

A screenshot of a terminal window titled 'filipe@filipe-virtualbox: ~'. The prompt is 'filipe@filipe-virtualbox:~\$'. The command 'uptime' has been entered and executed. The output is '17:09:01 up 2:14, 1 user, load average: 0,06, 0,04, 0,05'. The prompt is now 'filipe@filipe-virtualbox:~\$' followed by a cursor.

```
filipe@filipe-virtualbox:~$ uptime
17:09:01 up 2:14, 1 user, load average: 0,06, 0,04, 0,05
filipe@filipe-virtualbox:~$
```

Imagem 45 - uptime

4.4 Listar os utilizadores atualmente ligados ao sistema.

Para listar os utilizadores atualmente ligados ao sistema, utilizei os comandos **who** e **w**.

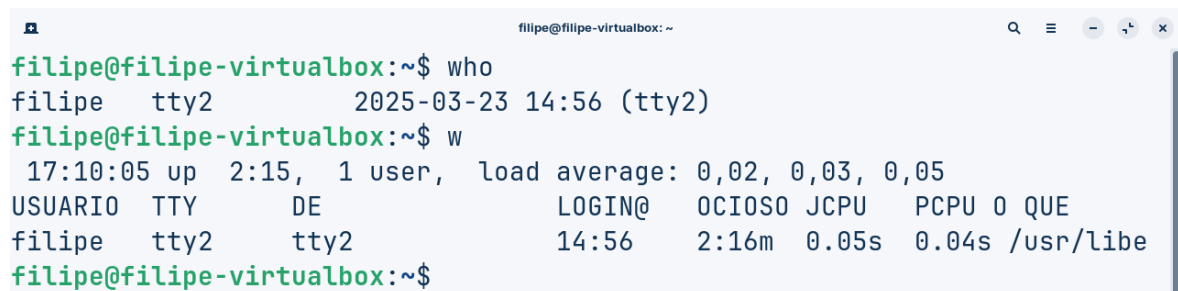
O comando **who** exibe uma lista dos utilizadores com sessão iniciada, mostrando informações como o nome de utilizador, terminal utilizado e hora de login.



```
filipe@filipe-virtualbox: ~  
filipe@filipe-virtualbox:~$ who  
filipe  tty2          2025-03-23 14:56 (tty2)  
filipe@filipe-virtualbox:~$
```

Imagem 46 – who

O comando **w** fornece informações mais detalhadas, incluindo o tempo de atividade do sistema, a carga do processador e o que cada utilizador está a executar no momento.



```
filipe@filipe-virtualbox: ~  
filipe@filipe-virtualbox:~$ who  
filipe  tty2          2025-03-23 14:56 (tty2)  
filipe@filipe-virtualbox:~$ w  
17:10:05 up 2:15, 1 user, load average: 0,02, 0,03, 0,05  
USUARIO  TTY      DE              LOGIN@  OCIOSO JCPU   PCPU  O QUE  
filipe   tty2     tty2           14:56   2:16m  0.05s  0.04s /usr/libe  
filipe@filipe-virtualbox:~$
```

Imagem 47 - w

4.5 Criar um novo utilizador (se permitido) e atribuir-lhe uma password.

Para criar um novo utilizador e atribuir-lhe uma password, utilizei os seguintes comandos:

sudo useradd teste: Este comando cria um novo utilizador chamado teste.

O **useradd** é responsável por adicionar a conta ao sistema, enquanto o **sudo** garante que a operação é realizada com privilégios administrativos.

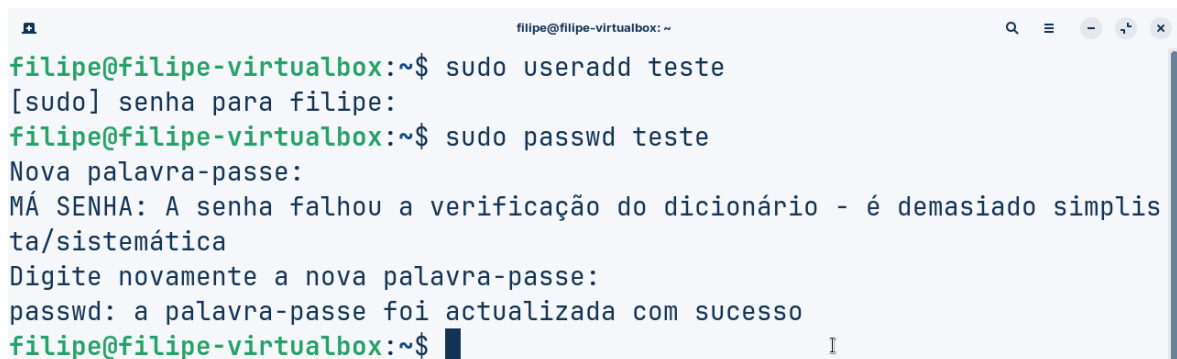


```
filipe@filipe-virtualbox:~$ sudo useradd teste
[sudo] senha para filipe:
filipe@filipe-virtualbox:~$
```

Imagem 48 – `sudo useradd teste`

sudo passwd teste: Após a criação do utilizador, este comando permite definir uma password para a conta **teste**.

O sistema solicitará a introdução da nova password e a sua confirmação.



```
filipe@filipe-virtualbox:~$ sudo useradd teste
[sudo] senha para filipe:
filipe@filipe-virtualbox:~$ sudo passwd teste
Nova palavra-passe:
MÁ SENHA: A senha falhou a verificação do dicionário - é demasiado simplista/sistemática
Digite novamente a nova palavra-passe:
passwd: a palavra-passe foi actualizada com sucesso
filipe@filipe-virtualbox:~$
```

Imagem 49 – `sudo passwd teste`

getent passwd teste: Este comando consulta a base de dados do sistema para verificar se o utilizador **teste** foi corretamente criado.

A saída contém informações como o nome de utilizador, UID (**User ID**), GID (**Group ID**), diretório pessoal e shell atribuído.

```
filipe@filipe-virtualbox: ~$ sudo useradd teste
[sudo] senha para filipe:
filipe@filipe-virtualbox:~$ sudo passwd teste
Nova palavra-passe:
MÁ SENHA: A senha falhou a verificação do dicionário - é demasiado simplis
ta/sistemática
Digite novamente a nova palavra-passe:
passwd: a palavra-passe foi actualizada com sucesso
filipe@filipe-virtualbox:~$ getent passwd teste
teste:x:1001:1001:~/home/teste:/bin/sh
filipe@filipe-virtualbox:~$
```

Imagem 50 – `getent passwd teste`

Segue a imagem do novo utilizador no ecrã de login do sistema.

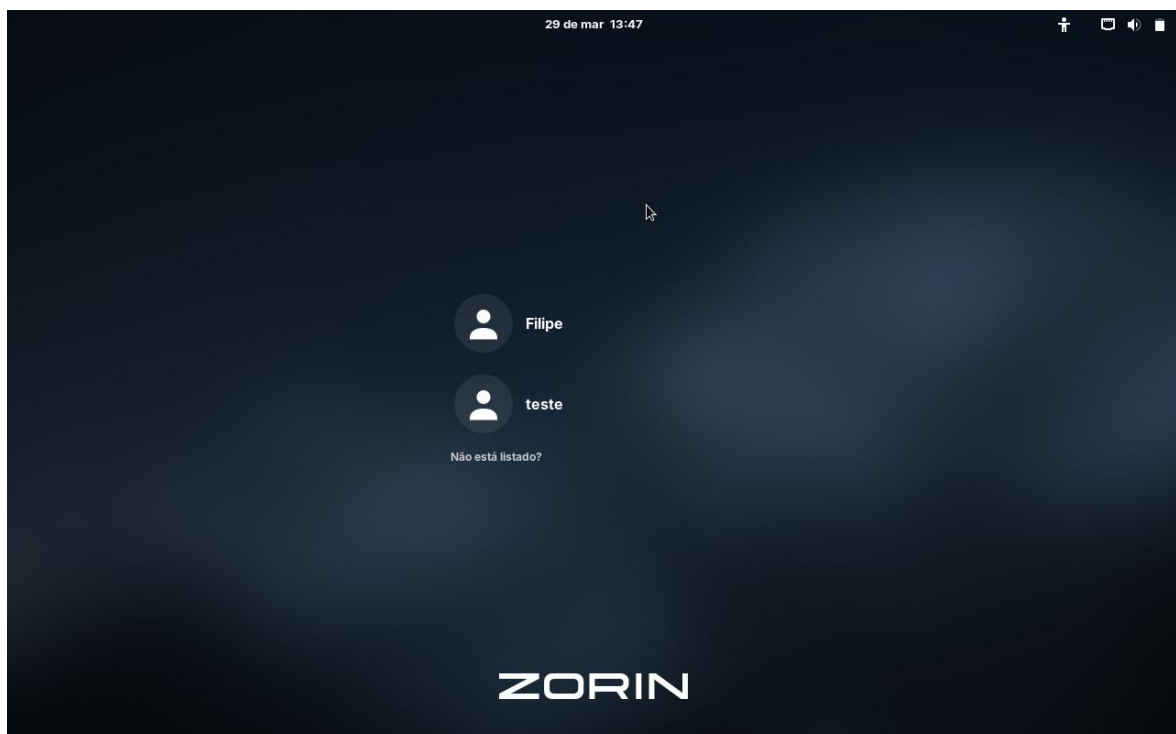


Imagem 51 – Login teste

5. Configuração do Ambiente de Trabalho em Linha de Comandos

5.1 Criar um alias chamado listar que execute “ls -lh”.

Para criar um alias chamado listar que execute `ls -lh`, utilizei o comando `nano ~/.bashrc`.

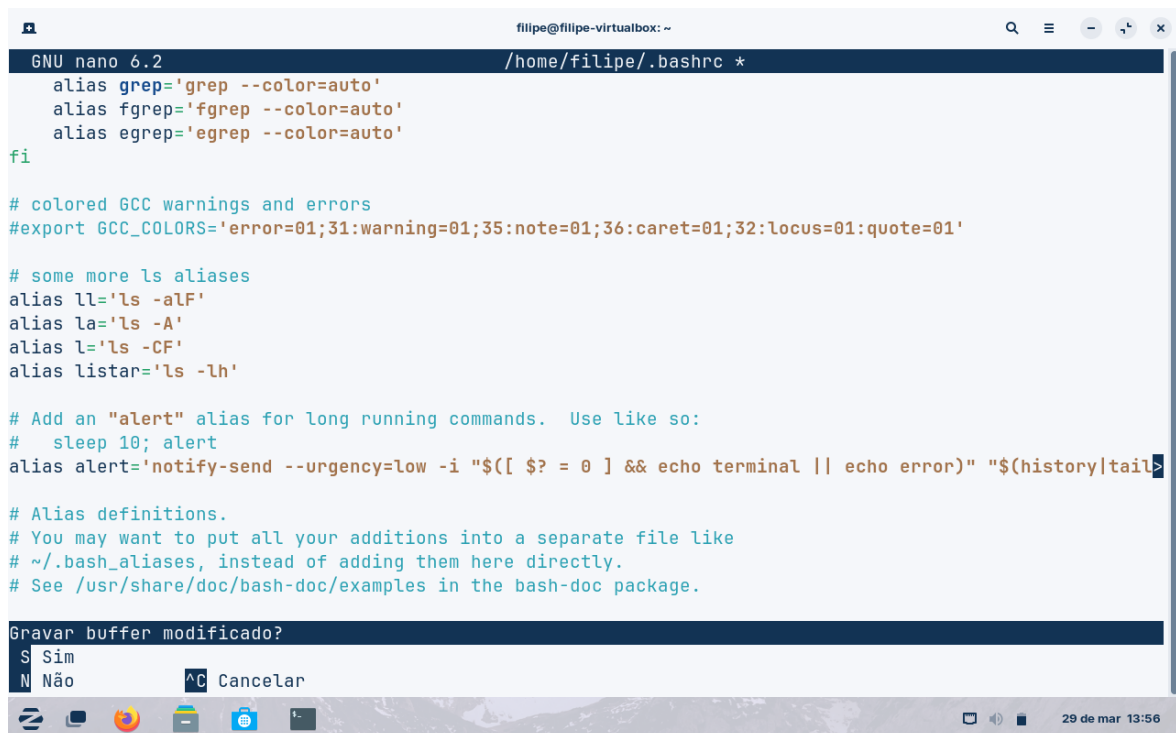
Este comando abre o ficheiro de configuração do **Bash**, onde foi adicionada a linha `alias listar='ls -lh'`.



Imagem 52 – nano ~/.bashrc



Imagem 53 - alias listar='ls -lh'



```
GNU nano 6.2 /home/filipe/.bashrc *
alias grep='grep --color=auto'
alias fgrep='fgrep --color=auto'
alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -aLF'
alias la='ls -A'
alias l='ls -CF'
alias listar='ls -lh'

# Add an "alert" alias for long running commands.  Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error)" "${history|tail>

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

Gravar buffer modificado?
S Sim
N Não ^C Cancelar
```

Imagem 54 – guardar nano ~/.bashrc

Após guardar as alterações e fechar o editor, executei `source ~/.bashrc` para aplicar a nova configuração.



```
filipe@filipe-virtualbox: ~$ nano ~/.bashrc
filipe@filipe-virtualbox: ~$ source ~/.bashrc
filipe@filipe-virtualbox: ~$
```

Imagem 55 – source ~/.bashrc

Com esta configuração, sempre que o comando listar for digitado no terminal, será executado `ls -lh`, facilitando a listagem detalhada dos ficheiros e diretórios."



```
filipe@filipe-virtualbox: ~$ nano ~/.bashrc
filipe@filipe-virtualbox: ~$ source ~/.bashrc
filipe@filipe-virtualbox: ~$ listar
total 32K
drwxr-xr-x 3 filipe filipe 4,0K mar 23 15:44 Desktop
drwxr-xr-x 2 filipe filipe 4,0K mar 8 18:12 Documentos
drwxr-xr-x 2 filipe filipe 4,0K mar 8 18:12 Imagens
drwxr-xr-x 2 filipe filipe 4,0K mar 8 18:12 Modelos
drwxr-xr-x 2 filipe filipe 4,0K mar 8 18:12 Música
drwxr-xr-x 2 filipe filipe 4,0K mar 8 18:12 Público
drwxr-xr-x 2 filipe filipe 4,0K mar 8 18:12 Transferências
drwxr-xr-x 2 filipe filipe 4,0K mar 8 18:12 Vídeos
filipe@filipe-virtualbox: ~$
```

Imagem 56 - listar

5.2 Adicionar o diretório “meu_trabalho” à variável “PATH”.

Para adicionar o diretório **meu_trabalho** à variável **PATH**, utilizei o comando **export** **`PATH="$HOME/meu_trabalho:$PATH"`**.

Este comando atualiza a variável **PATH**, permitindo que os executáveis dentro do diretório **meu_trabalho** sejam executados a partir de qualquer local no terminal.

A terminal window titled 'filipe@filipe-virtualbox: ~' showing the command 'export PATH="\$HOME/meu_trabalho:\$PATH"' being executed. The prompt changes from 'filipe@filipe-virtualbox:~\$' to 'filipe@filipe-virtualbox:~\$' after the command is run.

```
filipe@filipe-virtualbox:~$ export PATH="$HOME/meu_trabalho:$PATH"
filipe@filipe-virtualbox:~$
```

Imagem 57 - `export PATH="$HOME/meu_trabalho:$PATH"`

Para confirmar a alteração, utilizei o comando **echo** **`$PATH`**, que exibe o conteúdo atualizado da variável **PATH**.

A terminal window titled 'filipe@filipe-virtualbox: ~' showing the command 'echo \$PATH' being executed. The output is a long string of directory paths separated by colons.

```
filipe@filipe-virtualbox:~$ export PATH="$HOME/meu_trabalho:$PATH"
filipe@filipe-virtualbox:~$ echo $PATH
/home/filipe/meu_trabalho:/usr/local/sbin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/snap/bin:/snap/bin
filipe@filipe-virtualbox:~$
```

Imagem 58 - `echo $PATH`

5.3 Personalizar o prompt de comandos.

Para personalizar o prompt de comandos, utilizei o comando **nano ~/.bashrc** para editar o ficheiro de configuração do **Bash**.

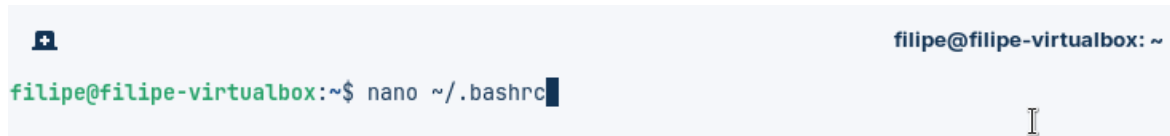


Imagem 59 – nano ~/.bashrc

Foi adicionada a seguinte linha ao ficheiro:

```
PS1='\[\e[40m\]\[\e[97m\]$(uptime -p) Mem: $(free -h | grep Mem | awk "{print\n$3/\n$2 * 100.0}")%\[\e[1;36m\]\t \[\e[1;32m\]@\h \[\e[1;34m\]\w\[\e[0m\] \ $ '
```

Esta configuração altera o prompt para exibir:

- O tempo que o sistema está em funcionamento (uptime -p).
- A percentagem de memória utilizada (free -h | grep Mem | awk "{print \n\$3/\n\$2 * 100.0}").
- A hora atual (\t).
- O nome do anfitrião (@\h).
- O diretório atual (\w).
- Diferentes cores para melhorar a legibilidade.
 - Preto de fundo → \e[40m
 - Branco de texto → \e[97m (utilizado para mostrar o tempo de atividade e a percentagem de memória usada)
 - Ciano claro → \e[1;36m (utilizado para exibir a hora atual)
 - Verde claro → \e[1;32m (utilizado para o nome do anfitrião @\h)
 - Azul claro → \e[1;34m (utilizado para o diretório atual \w)
 - Reset de cores → \e[0m (para garantir que as cores não se propaguem para os comandos seguintes)

```

filipe@filipe-virtualbox: ~
GNU nano 6.2 /home/filipe/.bashrc *
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
alias listar='ls -lh'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error)" "${history|tail -n1|sed -e 's/\s*[0-9]\+\s*//;s/;/&|\\s*alert$//}'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

PS1='\[\e[40m\]\[\e[97m\]$(uptime -p) Mem: $(free -h |grep Mem | awk "{print \$3/\$2 * 100.0}")%\[\e[1;36m\]\t \[\e[1;32m\]@h \[\e[1;34m\]w\[\e[0m\] \$ '

```

Imagem 60 – PS1

Após guardar as alterações e fechar o editor, utilizei o comando `source ~/.bashrc` para aplicar as modificações.

Por fim, executei `cd Desktop/` para navegar até ao diretório **Desktop** e verificar a nova aparência do prompt.

```

filipe@filipe-virtualbox: ~
filipe@filipe-virtualbox:~$ nano ~/.bashrc
filipe@filipe-virtualbox:~$ source ~/.bashrc
up 49 minutes Mem: 34,2105% 16:11:10 @filipe-virtualbox ~ $ cd Desktop/
up 49 minutes Mem: 34,2105% 16:11:20 @filipe-virtualbox ~/Desktop $

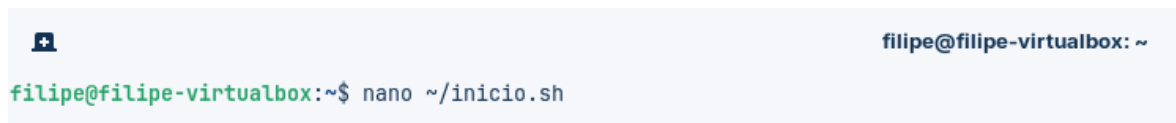
```

Imagem 61 – Personalização do prompt concluída

5.4 Criar um script “Bash” que exiba uma mensagem de boas-vindas ao iniciar sessão.

Para criar um script Bash que exiba uma mensagem de boas-vindas ao iniciar sessão, segui os seguintes passos:

Utilizei o comando `nano ~/inicio.sh` para criar e editar o script.



```
filipe@filipe-virtualbox: ~$ nano ~/inicio.sh
```

Imagem 62 – nano ~/inicio.sh

Dentro do script `inicio.sh`, adicionei a linha:

`echo "Bem-vindo ao ambiente de trabalho, ${whoami}!"`



Imagem 63 – echo “Bem-vindo ao ambiente de trabalho, \${whoami}”

Esta linha exibe uma mensagem de boas-vindas personalizada, usando o comando **whoami** para mostrar o nome do utilizador atual.

Em seguida, editei o ficheiro `~/bashrc` com o comando `nano ~/bashrc` e adicionei a linha:

`~/inicio.sh`



```
filipe@filipe-virtualbox: ~$ nano ~/inicio.sh
filipe@filipe-virtualbox: ~$ nano ~/.bashrc
```

Imagem 64 – nano `~/bashrc`



```
GNU nano 6.2 /home/filipe/.bashrc
# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
alias listar='ls -lh'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error)" "${history|tail -n1|sed -e '\''s/^s*[0-9]\+\s*//;s/[:&]\s*alert$//'\`

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

~/inicio.sh
```

Imagem 65 – adição do comando `~/inicio.sh` no nano `~/bashrc`

Esta linha garante que o script **inicio.sh** seja executado automaticamente sempre que uma nova sessão de terminal for iniciada.

Quando tentei aplicar as alterações com `source ~/.bashrc`, ocorreu um erro devido à falta de permissões de execução para o script.

A terminal window titled 'filipe@filipe-virtualbox: ~' showing the following commands and output: 'nano ~/inicio.sh', 'nano ~/.bashrc', 'source ~/.bashrc', and 'bash: /home/filipe/inicio.sh: Permissão recusada'. The prompt returns to 'filipe@filipe-virtualbox:~\$'.

Imagem 66 – erro `source ~/.bashrc`

Para resolver isso, executei o comando `chmod +x ~/inicio.sh` para tornar o script executável.

A terminal window titled 'filipe@filipe-virtualbox: ~' showing the following commands and output: 'nano ~/inicio.sh', 'nano ~/.bashrc', 'source ~/.bashrc', 'bash: /home/filipe/inicio.sh: Permissão recusada', 'chmod +x ~/inicio.sh', and the prompt returns to 'filipe@filipe-virtualbox:~\$'.

Imagem 67 – `chmod +x ~/inicio.sh`

Após conceder as permissões, executei novamente `source ~/.bashrc` para aplicar as alterações corretamente.

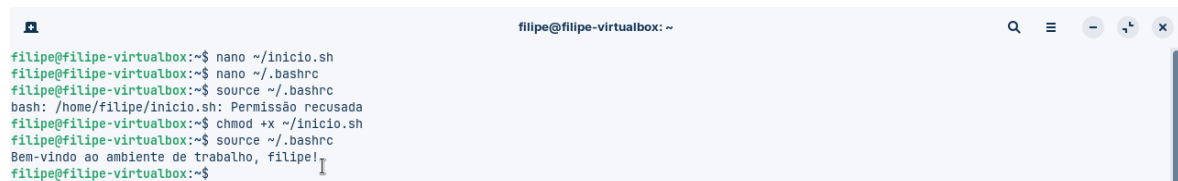
A terminal window titled 'filipe@filipe-virtualbox: ~' showing the following commands and output: 'nano ~/inicio.sh', 'nano ~/.bashrc', 'source ~/.bashrc', 'bash: /home/filipe/inicio.sh: Permissão recusada', 'chmod +x ~/inicio.sh', 'source ~/.bashrc', and ' Bem-vindo ao ambiente de trabalho, filipe!'. The prompt returns to 'filipe@filipe-virtualbox:~\$'.

Imagem 68 – `source ~/.bashrc`

E com isso a partir de agora sempre que iniciar uma sessão de terminal será exibida a mensagem de boas-vindas.

Conclusão

Este trabalho permitiu explorar e aplicar uma série de operações essenciais no ambiente Unix/Linux, com ênfase na utilização da linha de comandos. Durante a realização das tarefas, adquiri conhecimentos práticos sobre a manipulação e gestão de ficheiros, redirecionamento de saídas, extração de informação de ficheiros de texto e gestão da máquina e utilizadores.

Através da execução de comandos como **ls**, **cp**, **mv**, **grep**, **sort**, e **echo**, foi possível compreender o funcionamento básico e avançado do sistema, bem como a importância de utilizar ferramentas de forma eficiente para otimizar o trabalho no terminal. Além disso, a personalização do ambiente de trabalho, através de aliases, variáveis de ambiente e scripts Bash, demonstrou a flexibilidade do Unix/Linux para se adaptar às necessidades de cada utilizador.

A gestão de utilizadores e a consulta de informações do sistema, como a carga de trabalho e a utilização do disco, proporcionaram uma visão mais profunda do estado e desempenho da máquina.