

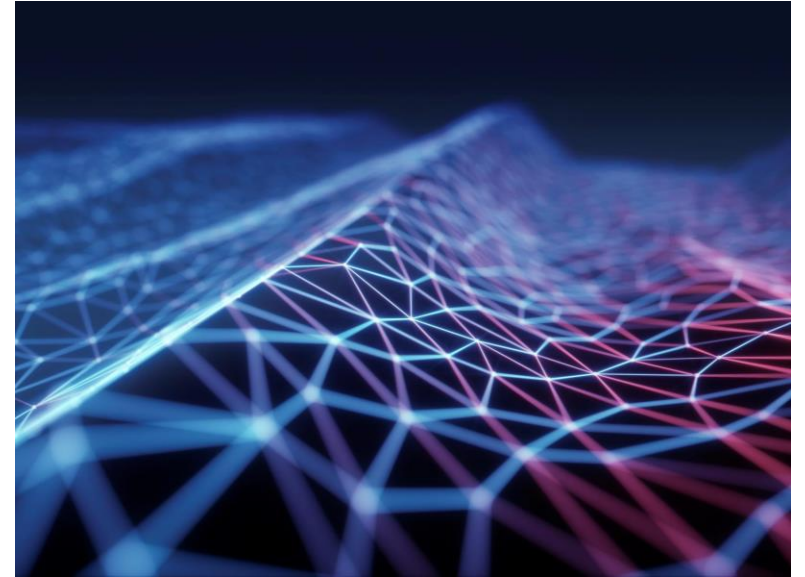
# 01 - SISTEMAS OPERACIONAIS - PRÁTICA

PROFESSORES:

OSMAR DE OLIVEIRA BRAZ JUNIOR

RICHARD HENRIQUE DE SOUZA

JORGE WERNER



# <TECH\_WEEK>

## CONECTE-SE AO FUTURO




29, 30 E 31 DE AGOSTO  
TEMA: INTELIGÊNCIA ARTIFICIAL

- + PALESTRAS
- + ATIVIDADES PRÁTICAS
- + PARTICIPAÇÃO DE GRANDES EMPRESAS
- + OPORTUNIDADES DE NETWORKING
- + APRENDIZADO E ATUALIZAÇÃO





# EXTENSÃO

- Olá, estudante!
- Já parou para pensar que a sua atuação profissional pode gerar impacto social e qualidade de vida para comunidades inteiras, desde a graduação?! Vem com a gente nesta missão incrível e verdadeiramente transformadora! Inscreva-se e participe dos nossos projetos de extensão 2023/2!
- Confira o catálogo de projetos e edital disponíveis no site da sua IES. Podem se inscrever estudantes de graduação nas modalidades presencial, live e semipresencial.
-  Para se inscrever, siga o caminho:
  - Ulife > Vida Acadêmica > Pesquisa & Extensão > Inscrições Extensão
  -  Período de inscrição: 21/8 a 28/8
-  Ah, e se liga: a inscrição não garante a reserva da vaga. A reserva será feita após o período de inscrição e considerará a ordem dos inscritos e o número de vagas. Estão aptos para as vagas alunos(as) com o contrato de matrícula/rematrícula concluído

INSCRIÇÕES ABERTAS PARA OS

# PROJETOS DE EXTENSÃO

USE SEUS CONHECIMENTOS PARA  
GERAR **IMPACTOS SOCIAIS**

INSCRIÇÕES DE  
21 A 28 DE AGOSTO

Para se inscrever, basta fazer login no Ulife e seguir o caminho:  
**Vida Acadêmica > Pesquisa e Extensão  
> Inscrições Extensão.**

Garanta sua vaga!



ecossistema  
ânima

# LÍDER

- Eleições líder de turma – Projeto Liderando

# OBJETIVOS

- Conhecer o ambiente Windows
- Conceituar virtualização
- Preparar uma máquina virtual para executar os comandos Linux
- Executar comandos Linux e Windows
- Comparar comandos.

# CONCEITO



○ que é um Sistema Operacional?

- É um software cuja função é controlar e gerenciar o funcionamento de um computador e seus diversos recursos.

# FUNÇÕES BÁSICAS

- Um sistema operacional possui as seguintes funções:
  - gerenciamento de processos;
  - gerenciamento de memória;
  - gerenciamento do sistema de arquivos;
  - gerenciamento da entrada e saída de dados.

# SISTEMAS OPERACIONAIS



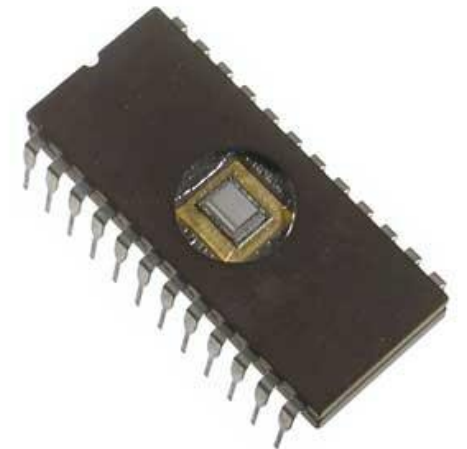
Mac





# LOCALIZAÇÃO DO SO

- No disco rígido ou
- Armazenado em um chip do tipo ROM

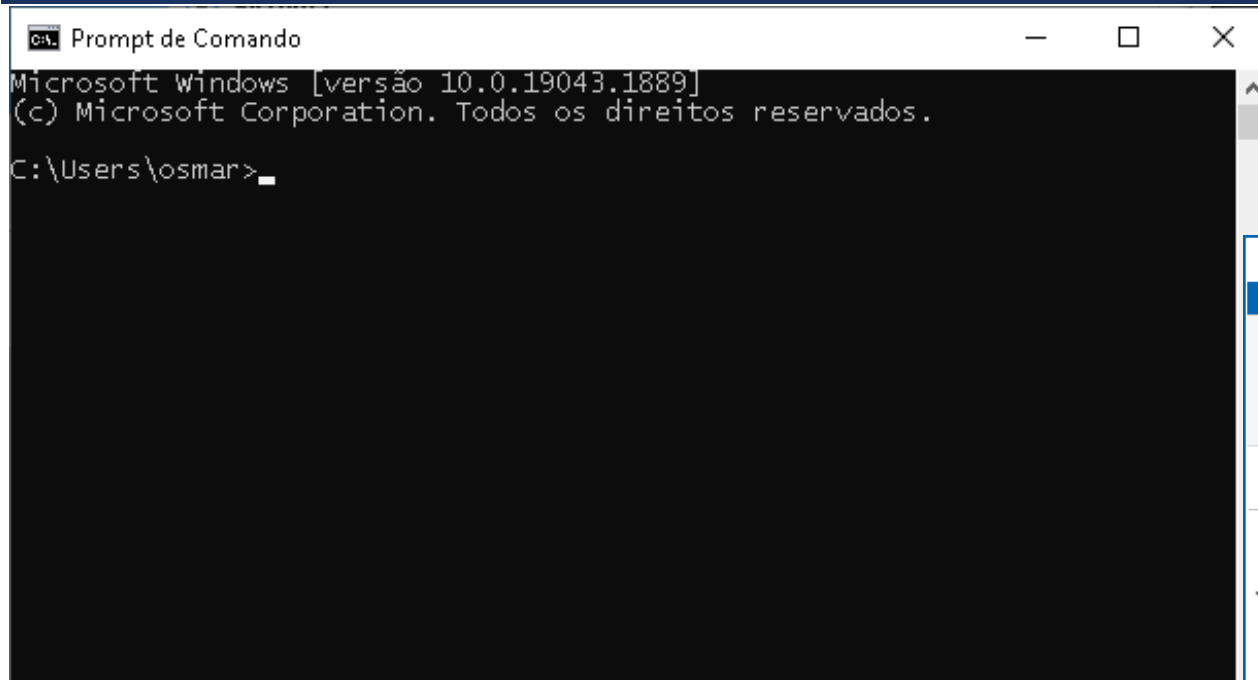


## TIPOS DE INTERFACE DE ACESSO AO SO

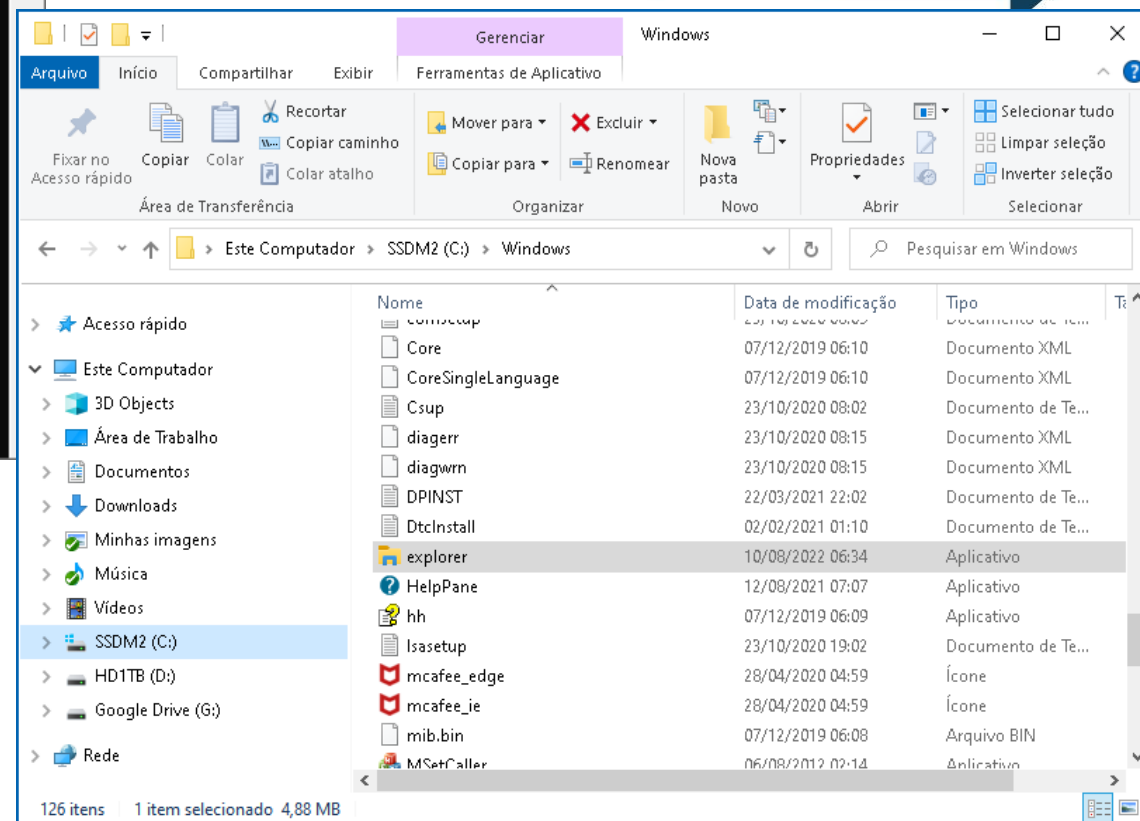
- **CLI:** Command-Line Interface – Interface de Linha de Comandos
  - O SO é controlado por meio de comandos emitidos em um prompt.
- **GUI:** Graphical User Interface – Interface Gráfica
  - O SO é controlado por meio de elementos gráficos acessados com um dispositivo apontador, como um mouse ou via touch.
- **Terminal:** Software que emula um terminal de hardware, geralmente com uma interface de linha de comandos, que pode ser executada em uma interface gráfica.



# TIPOS DE INTERFACE DE ACESSO AO SO

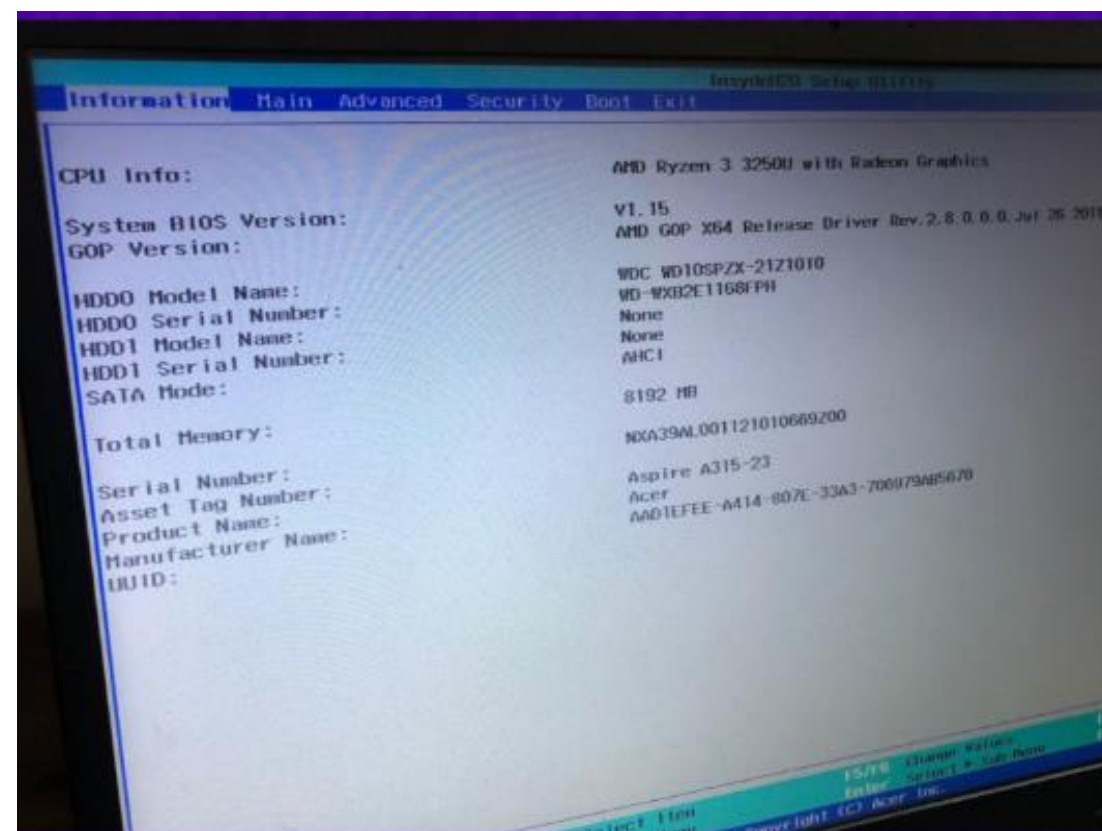


```
Prompt de Comando
Microsoft Windows [versão 10.0.19043.1889]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Users\osmar>
```



# FIRMWARE

- Firmware ou software embarcado é um conjunto de instruções programadas diretamente no hardware, que contém parâmetros específicos para a operação de um determinado dispositivo.
- Por exemplo,
  - A BIOS (Sistema Básico de Entrada e Saída) e a UEFI (Interface Extensível Unificada de Firmware) são exemplos de firmware, pois comportam instruções voltadas para a operação do hardware de um computador.



## ATIVIDADES

- Quais seriam as principais dificuldades que um programador teria no desenvolvimento de uma aplicação em um ambiente sem um sistema operacional?
- Explique a seguinte sentença: “O Sistema Operacional protege o usuário da máquina e a máquina do usuário”.



# SIMULAÇÕES DE SO

- O Projeto PCjs é uma coleção de simulações de computador escritas em JavaScript.
- É um projeto de código aberto mantido no GitHub e hospedado em [pcjs.org](https://www.pcjs.org).
- Os objetivos do projeto são criar simulações rápidas e completas de hardware de computador clássico, ajudar as pessoas a entender como essas primeiras máquinas funcionavam, facilitar a experimentação com diferentes configurações de máquina e fornecer uma plataforma para executar e analisar software de computador antigo.
- <https://www.pcjs.org/>

# SIMULAÇÕES DE SO



---

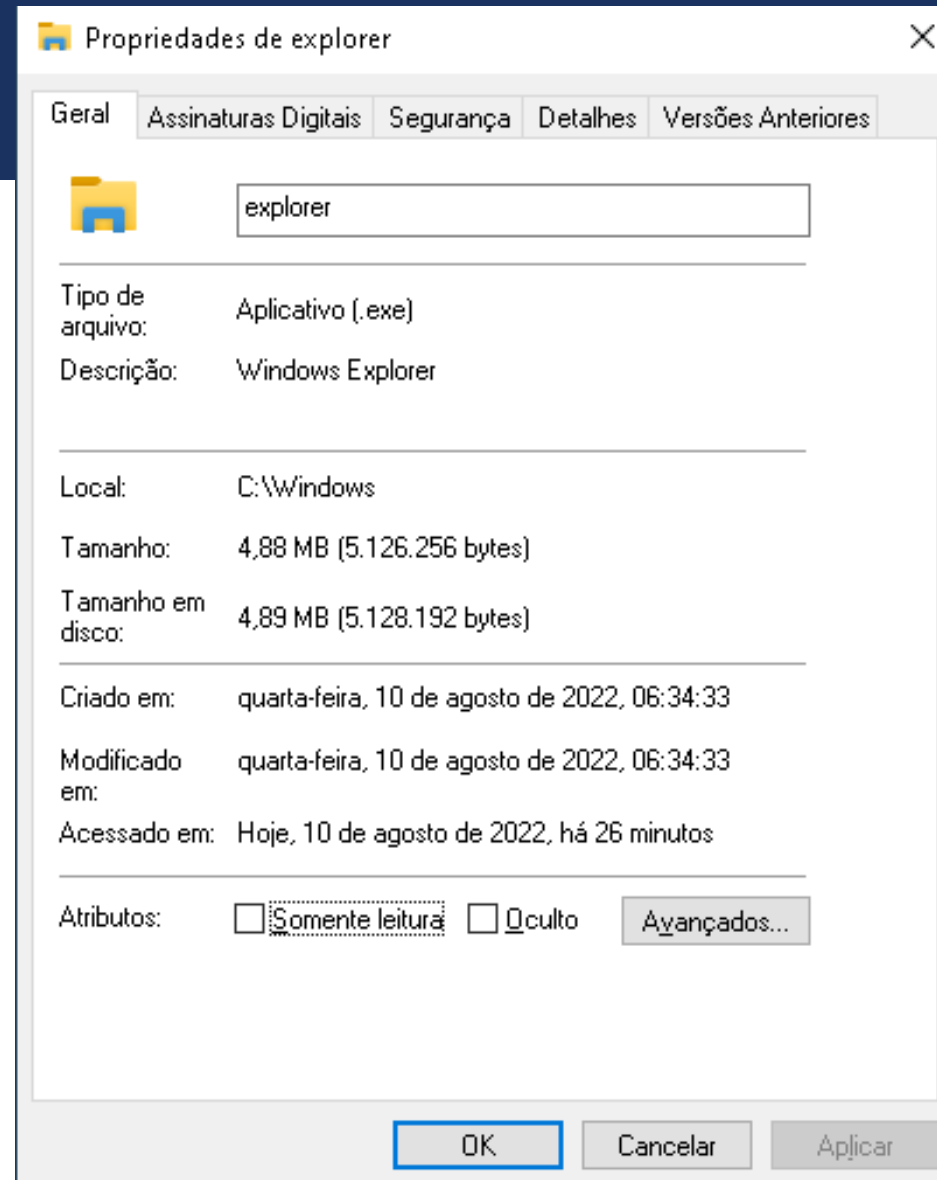
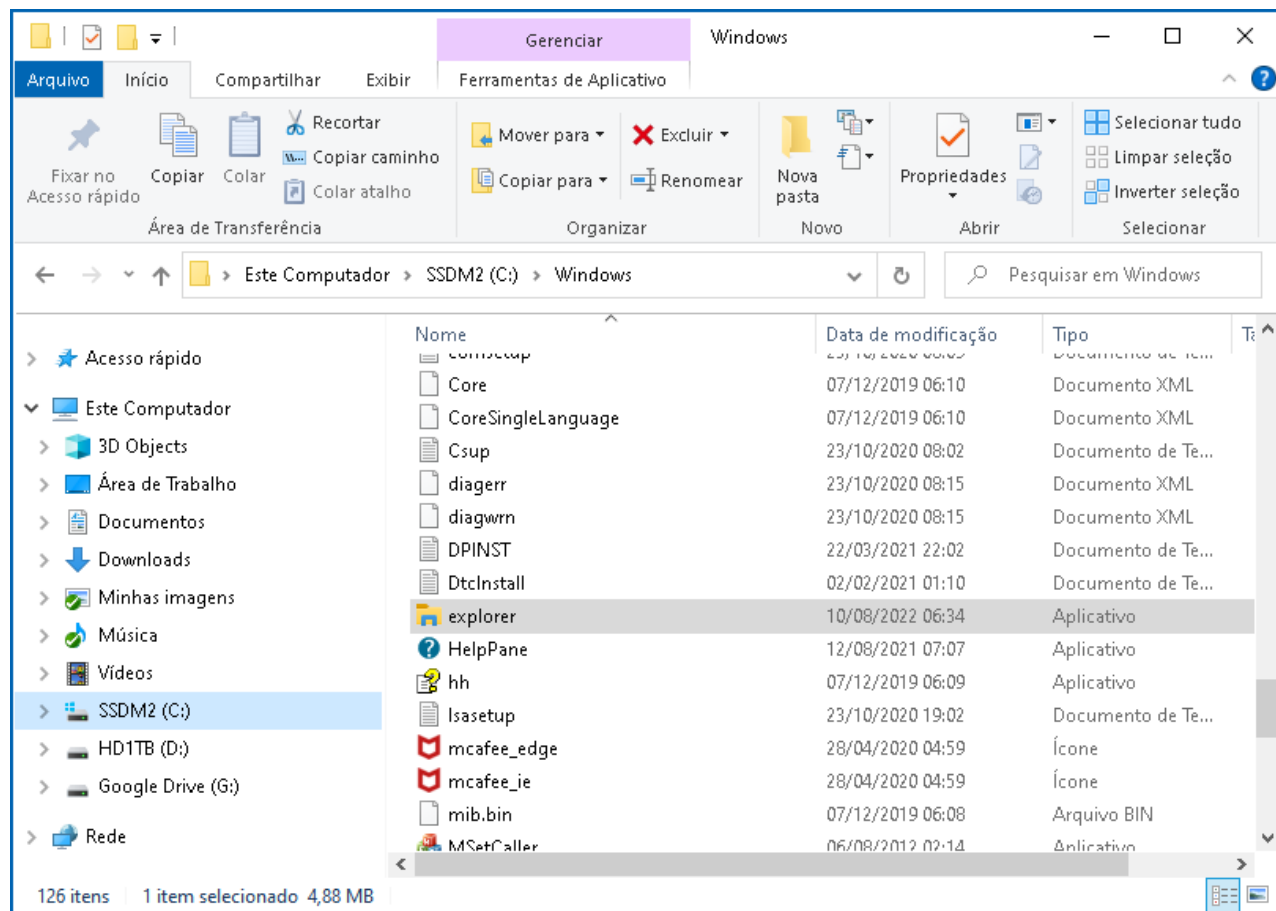
WINDOWS



# WINDOWS

- Microsoft Windows é uma família sistemas operacionais criados pela Microsoft, empresa fundada por Bill Gates.
- O Windows é um produto comercial, com preços diferenciados para cada uma de suas versões.
- O Windows é um sistema operacional de interface gráfica multitarefa. Isso significa que podemos trabalhar com vários programas simultaneamente.

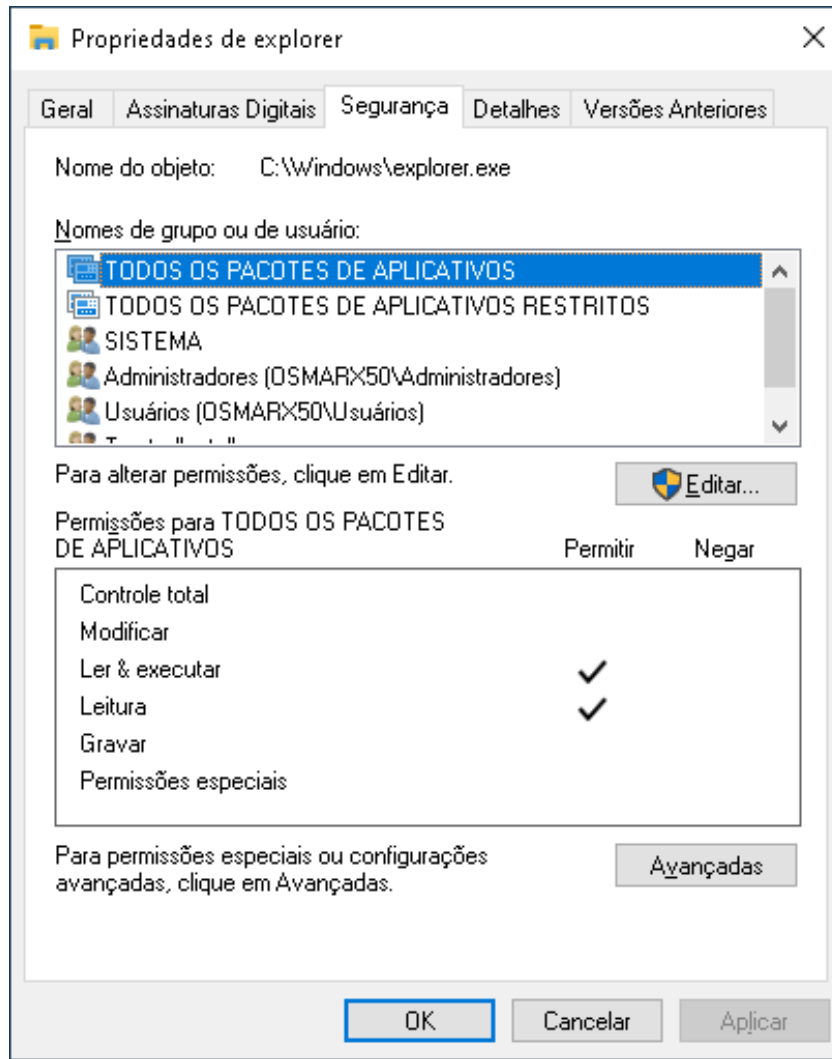
# WINDOWS GERENCIAMENTO DE ARQUIVOS





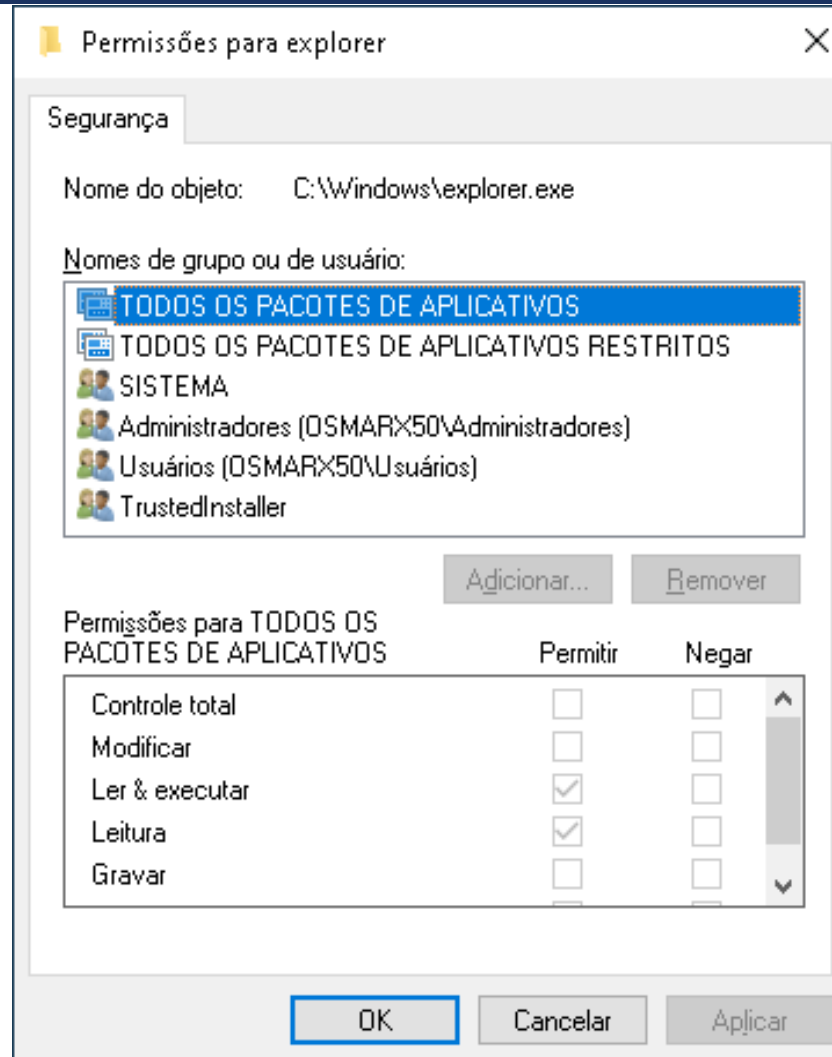
# WINDOWS

## GERENCIAMENTO DE ARQUIVOS - SEGURANÇA



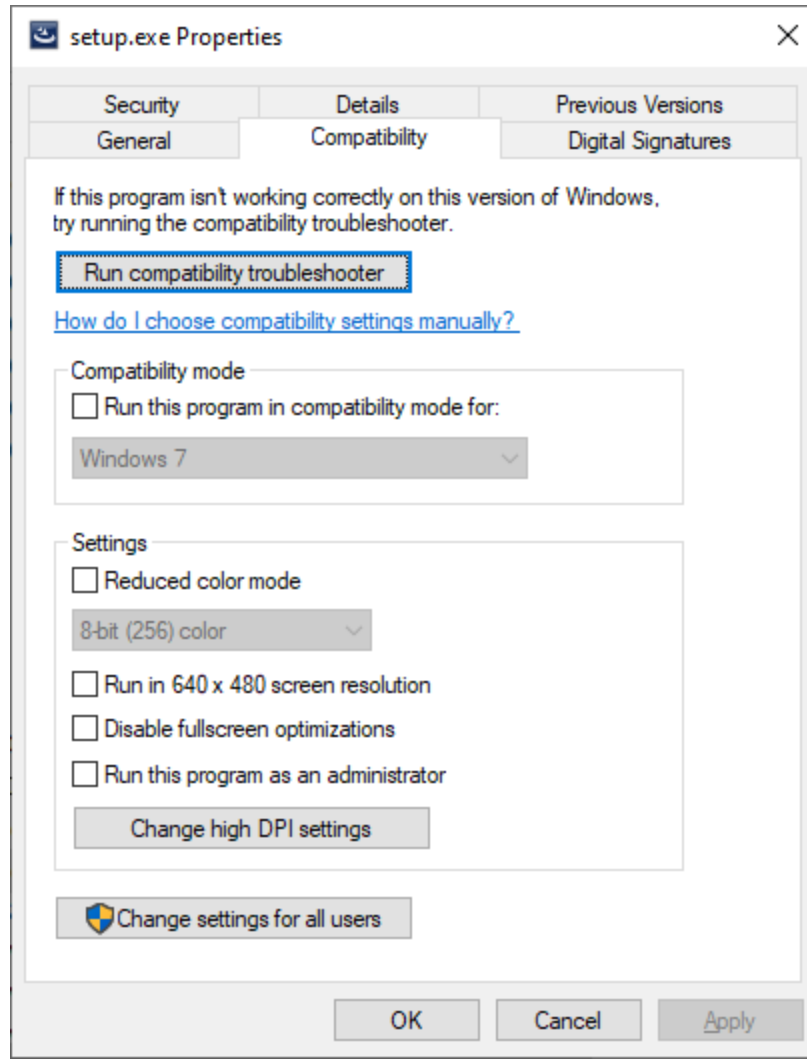
# WINDOWS

## GERENCIAMENTO DE ARQUIVOS - SEGURANÇA



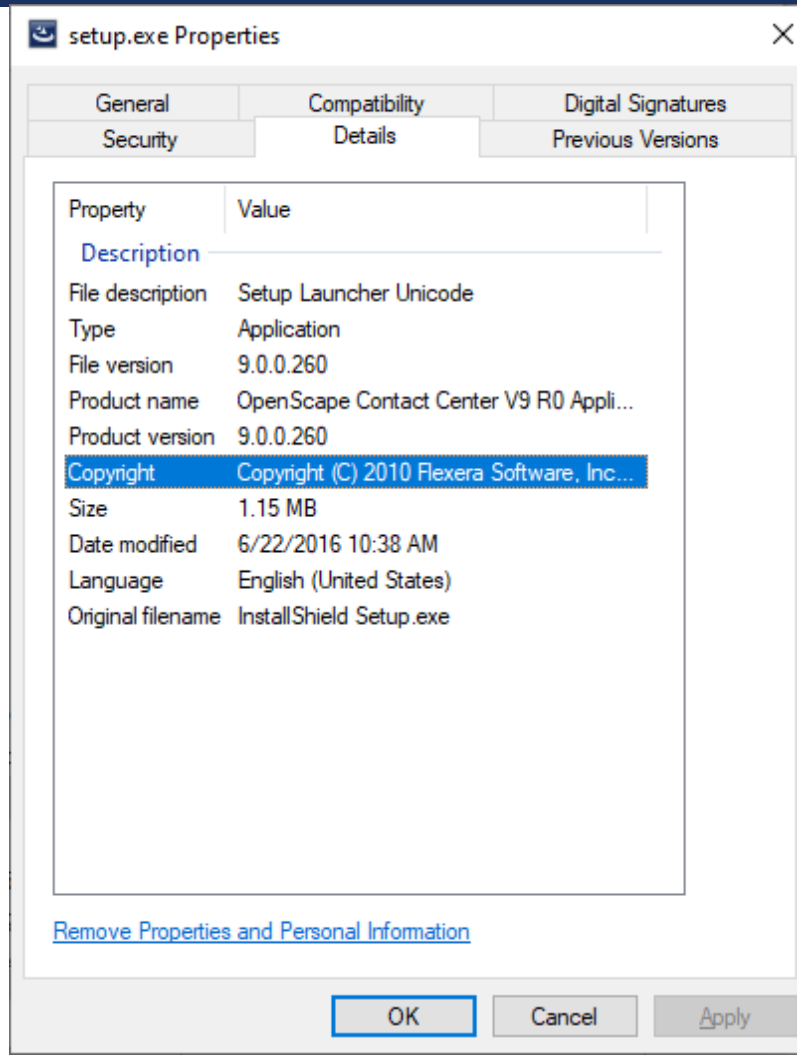
# WINDOWS

## GERENCIAMENTO DE ARQUIVOS – COMPATIBILIDADE



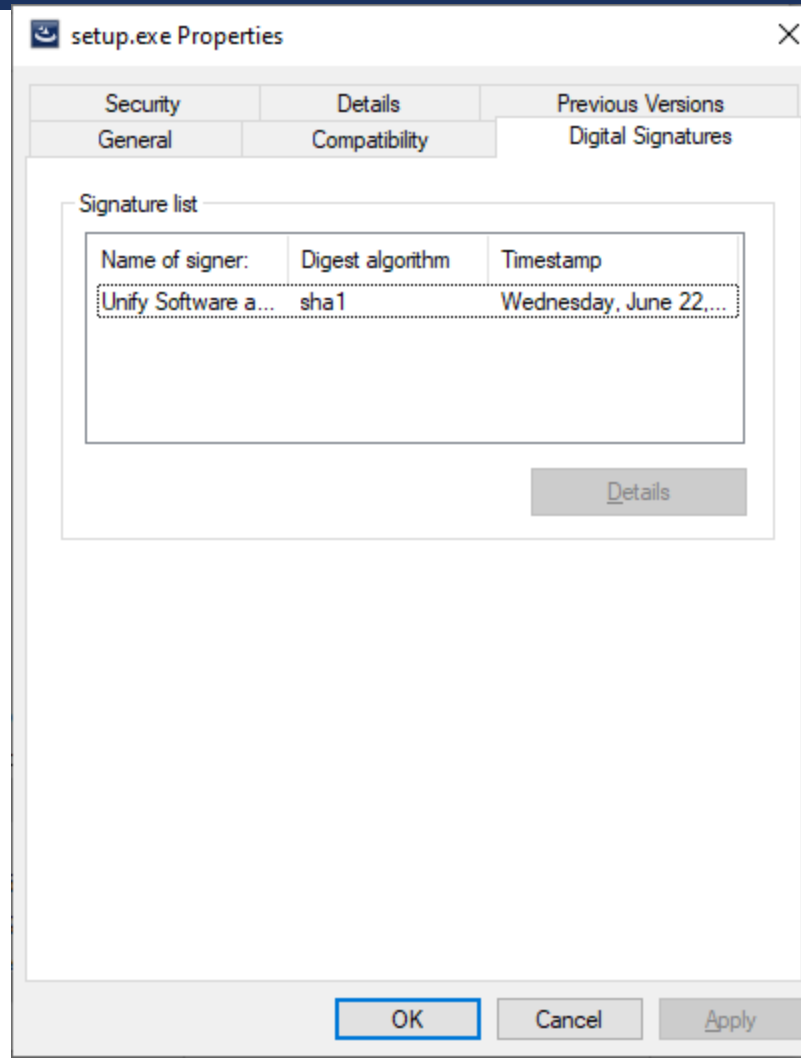
# WINDOWS

## GERENCIAMENTO DE ARQUIVOS – DETALHES



# WINDOWS

## GERENCIAMENTO DE ARQUIVOS – ASSINATURA DIGITAL





# WINDOWS POLÍTICAS LOCAIS

- Interface para configuração de Políticas Locais para:
  - Configuração para o Computador
  - Configuração para o Usuário
- Nota: Windows permite aplicação de Políticas por Grupo

# WINDOWS POLÍTICAS LOCAIS

The screenshot displays the Windows Local Security Policy console. The left pane shows a tree view of security policies, with 'System Audit Policies - Local Computer' expanded. The right pane shows a list of audit events and their status.

Subcategory	Status
Audit Events	
Audit IPsec Driver	Not Configured
Audit Other System Events	Not Configured
Audit Security State Change	Not Configured
Audit Security System Extension	Not Configured
Audit System Integrity	Not Configured

Below the main console, a file explorer window is open, showing a folder named 'logs' with a search bar. A context menu is open over a file, showing 'Scripts (Logon/Logoff)' and 'Logon' options, each with a 'More Actions' link.

At the bottom of the screen, the Windows taskbar is visible, showing the Start button, search icon, and several application icons. The system tray on the right shows the language (ENG), time (11:14 AM), and date (11/15/2021). An 'Activate Windows' watermark is present in the bottom right corner.

# WINDOWS

## MMC – MICROSOFT MANAGEMENT CONSOLE

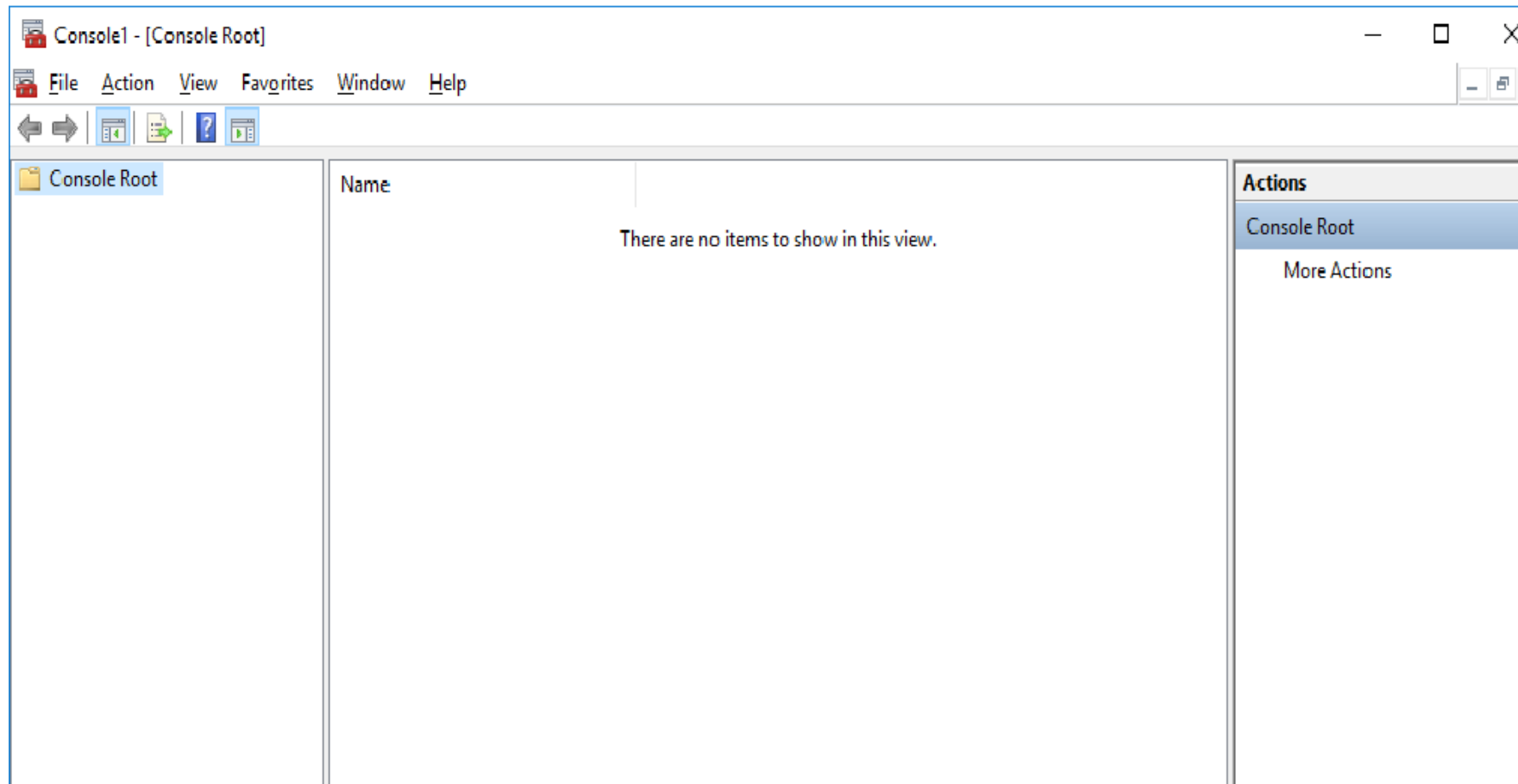
- Console para criar, salvar abrir ferramentas administrativas, chamadas de consoles, que gerenciam os componentes de:
  - hardware,
  - software
  - rede
- do seu sistema operacional Microsoft Windows.

# WINDOWS

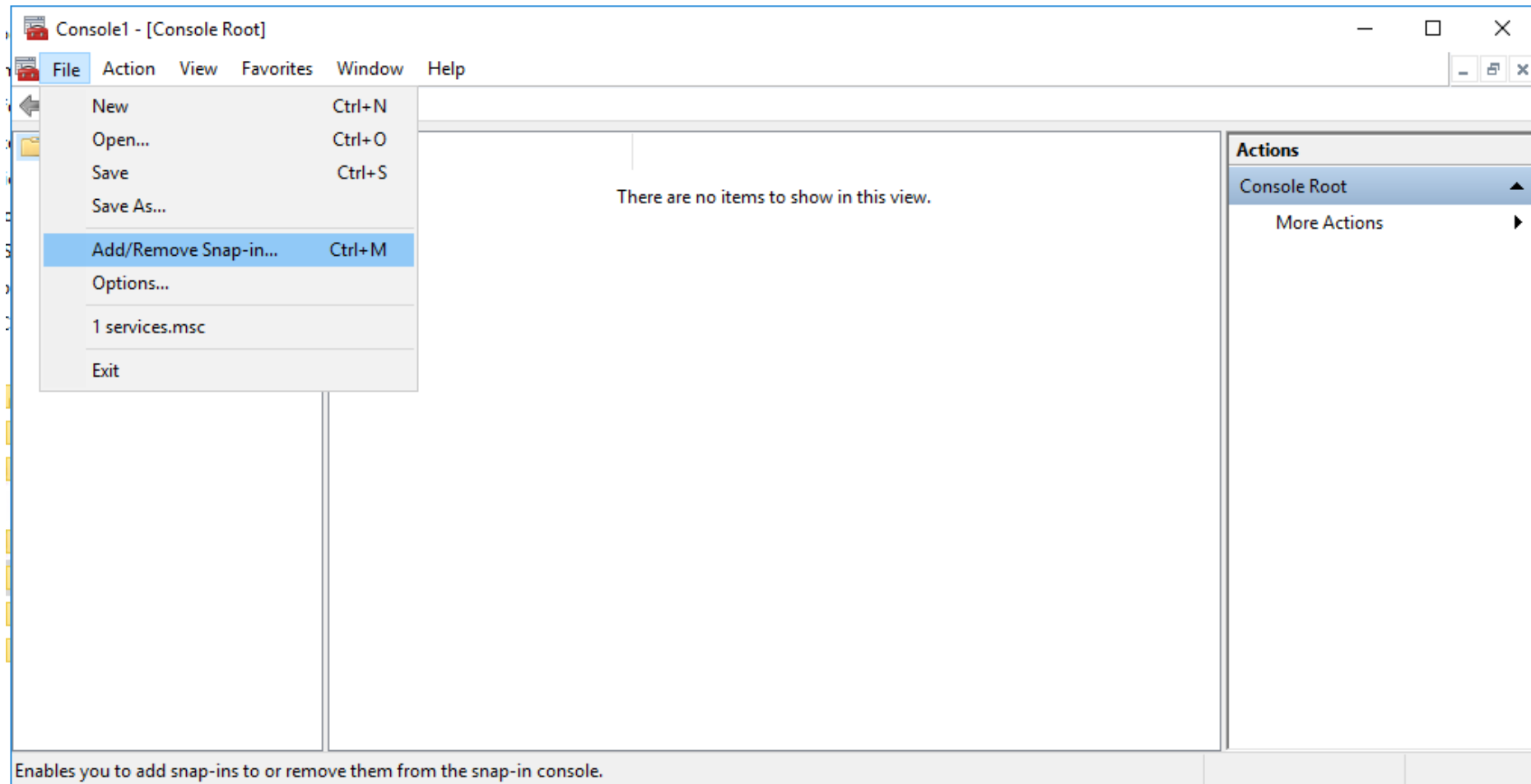
## MMC – MICROSOFT MANAGEMENT CONSOLE

- Snap-in
  - Ferramenta hospedada no MMC
  - Permite o gerenciamento de vários serviços usando uma única interface
  - Permite personalização do console

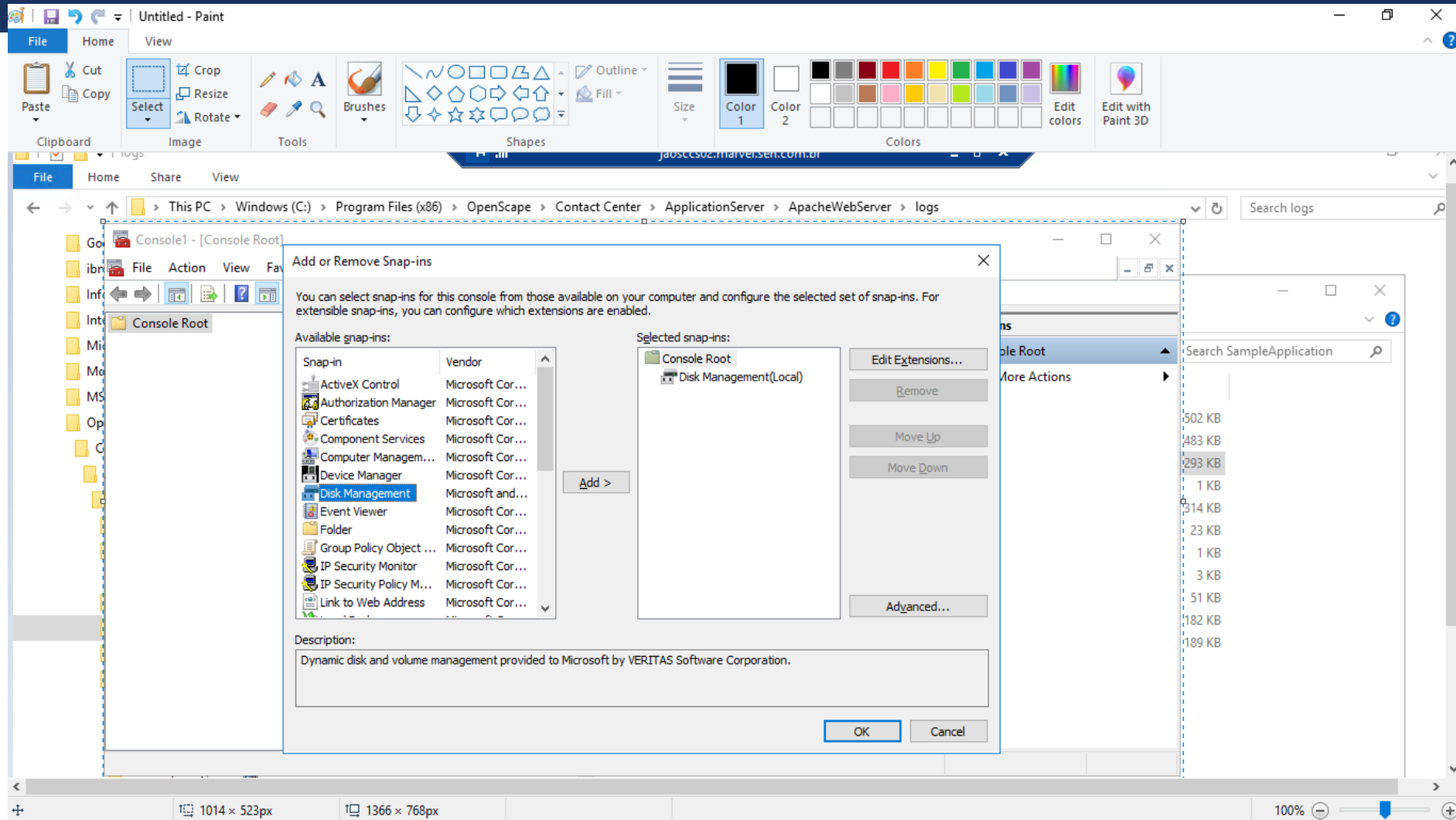
# WINDOWS MMC – MICROSOFT MANAGEMENT CONSOLE



# WINDOWS MMC – MICROSOFT MANAGEMENT CONSOLE



# WINDOWS MMC – MICROSOFT MANAGEMENT CONSOLE



# WINDOWS MMC – MICROSOFT MANAGEMENT CONSOLE

The screenshot displays the Windows Management Console (MMC) interface for Disk Management. The window title is "Console1 - [Console Root\Disk Management(Local)]". The menu bar includes File, Action, View, Favorites, Window, and Help. The left pane shows the tree view with "Console Root" and "Disk Management(Local)". The main pane displays a table of volumes and a detailed view of Disk 0.

Volume	Layout	Type	File System	Status
[Blue bar]	Simple	Basic		Healthy (Recovery Partition)
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)
Windows (C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)

Disk 0	System Reserved	Windows (C:)	Recovery Partition
Basic 160.00 GB Online	499 MB NTFS Healthy (System, /	157.91 GB NTFS Healthy (Boot, Page File, Crash Dump,	1.60 GB Healthy (Recovery Part

Legend: ■ Unallocated ■ Primary partition



# PREPARAÇÃO DO AMBIENTE LINUX

- Utilizaremos máquinas virtuais para realizar as nossas experiências com linux
- Permite instalar os ambientes em qualquer computador compatível com o virtualizador.
- Isola o ambiente de experimento do computador real.

# VIRTUALIZAÇÃO

- Consiste em simular um hardware ou um software sobre o qual roda um outro software
- Com isso podem-se rodar vários sistemas operacionais sobre uma única máquina
- “É como se fossem um ou mais computadores distintos dentro de um só”

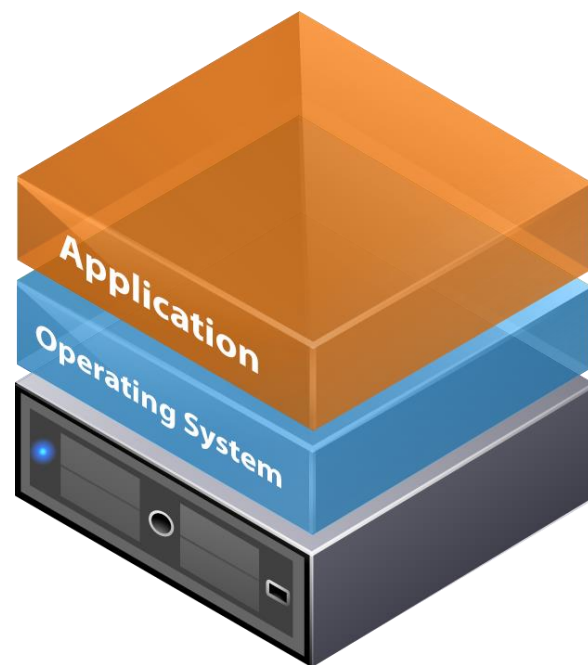
# MÁQUINA VIRTUAL

- “Uma réplica eficiente e isolada de uma máquina real”
- A IBM define como uma máquina virtual como uma cópia totalmente protegida e isolada de um sistema físico.
- É o ambiente simulado
- O gerenciamento é feito pelo virtualizador que controla o fluxo de informação entre os Sistemas Operacionais hóspedes e o hardware físico como:
  - CPU
  - Armazenagem em disco
  - Memória
  - Placas de interface de rede

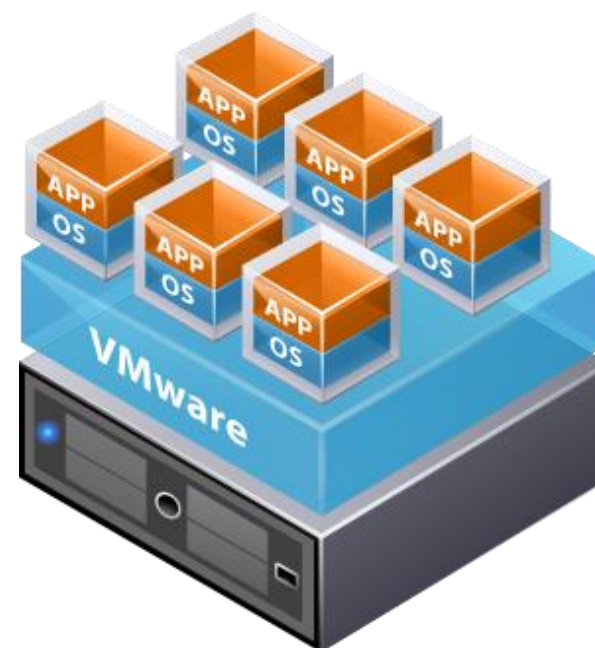
# MÁQUINA VIRTUAL

- Cria uma “**camada**” para compatibilizar diferentes plataformas e um mesmo Hardware.
- Esta camada é chamada de **virtualização**.
- Software são utilizados para fazer os recursos de hardware parecerem reais.

# VIRTUALIZAÇÃO



Arquitetura Tradicional



Arquitetura Virtual

# VIRTUALIZAÇÃO

Aplicações

Sistema Operacional

Hardware Virtual

Aplicações

Sistema Operacional

Hardware Virtual

**Software de Virtualização**

**Sistema Operacional**

**Hardware Real**

# MÁQUINA VIRTUAL

- **Vantagens:**
  - Execução de outros sistemas operacionais dentro de uma máquina física;
  - Ambiente para testes;
  - Acessar dados que possam conter vírus;
  - Testar um sistema operacional diferente ao que você utiliza;
  - Executar aplicações que não são compatíveis com seu sistema operacional;
  - Diminuição de custos com hardware;
  - Desenvolver softwares utilizando linguagens compatíveis apenas com sistemas operacionais específicos, entre outros.

# ISOLAMENTO

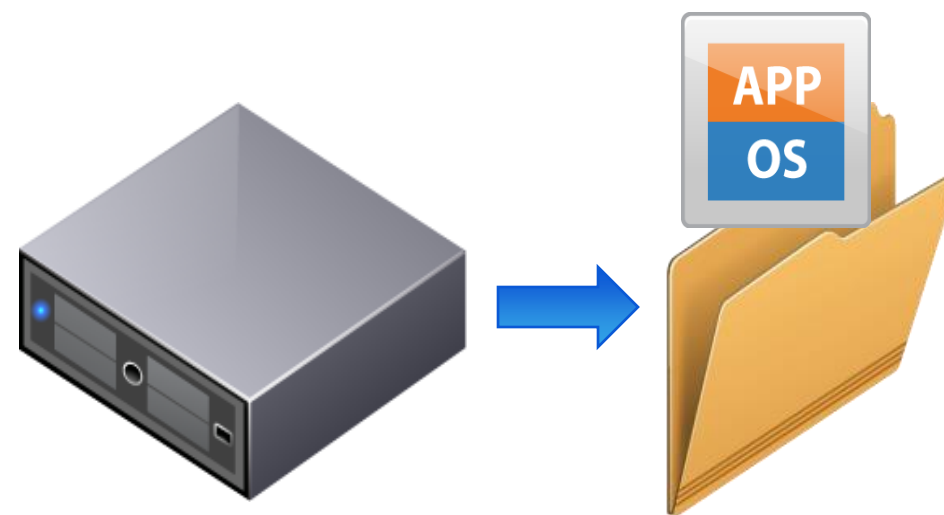
- Isolamento de falha e segurança no nível do hardware
- Controle de recursos preserva desempenho





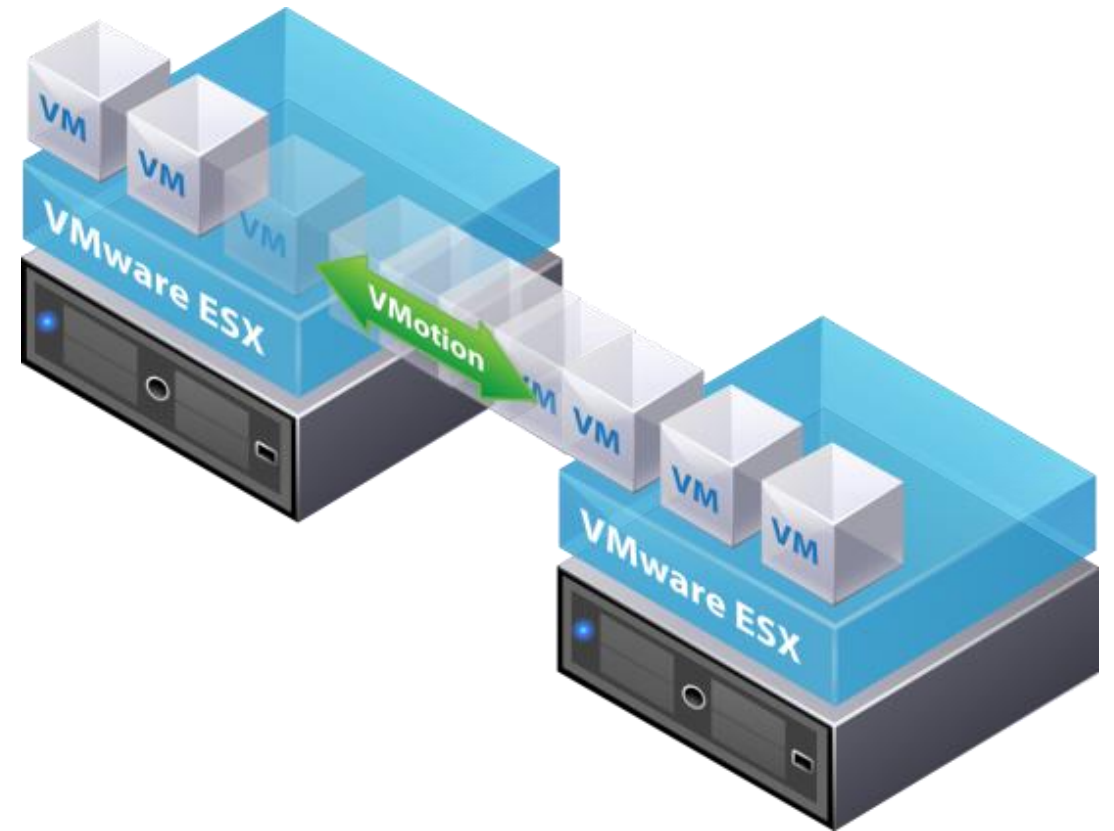
# ENCAPSULAMENTO

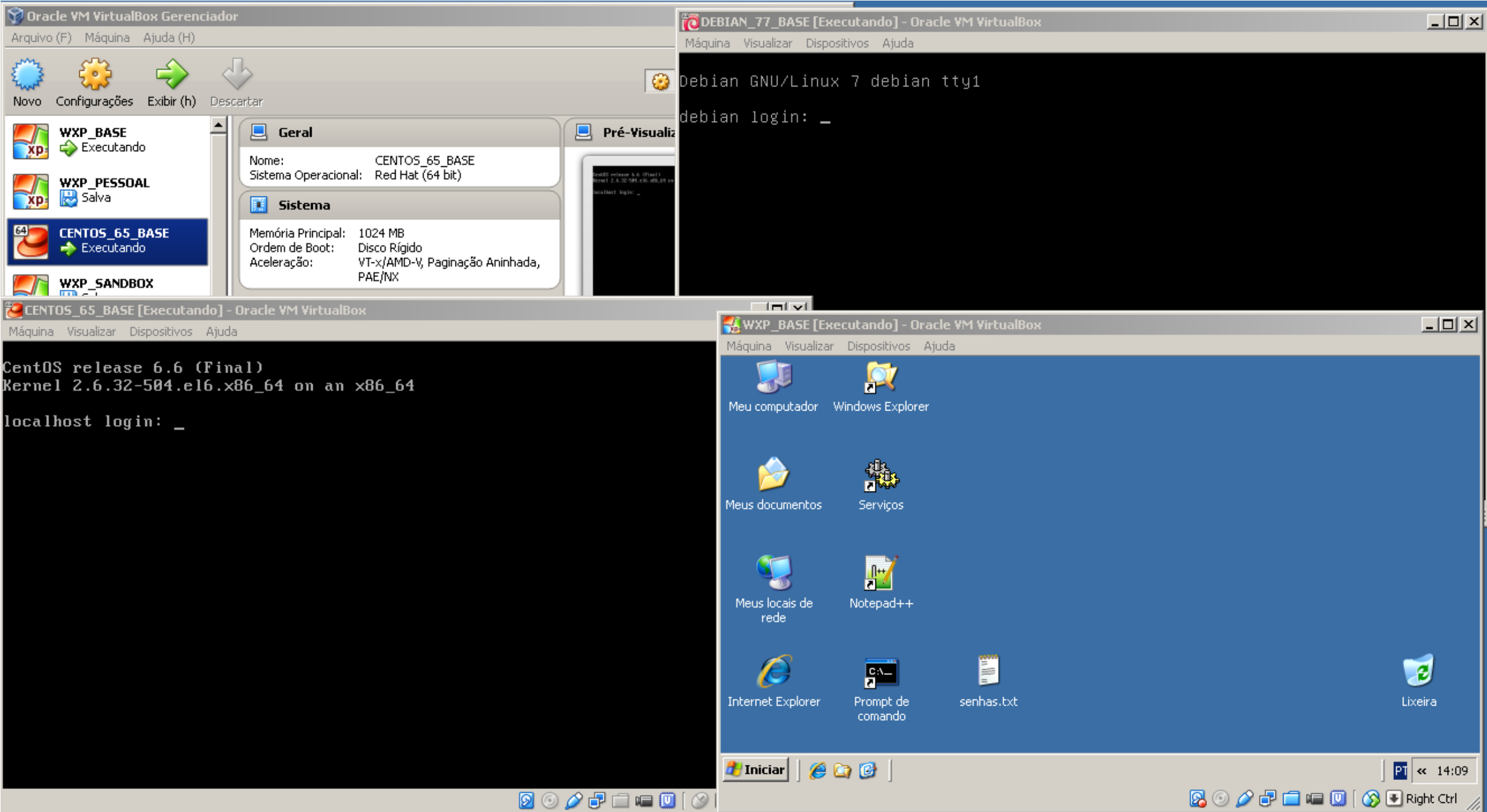
- Estado completo da máquina virtual pode ser salvo em arquivos
- Move e copia máquinas virtuais de forma tão fácil como mover e copiar arquivos



# INDEPENDÊNCIA DO HARDWARE

- Instalar ou migrar qualquer máquina virtual para qualquer servidor similar ou diferente





# VIRTUALIZAÇÃO

## ■ Tipo I

- Um software de virtualização se comunica diretamente com o hardware. Também chamado de **nativo** ou **bare-metal**. As máquinas virtuais rodam diretamente sobre ele.

## ■ Tipos II

- O software de virtualização roda sobre um sistema operacional(**hosted**). A máquina virtual(**guest**) roda sobre esta camada de software.

# VIRTUALBOX

- **VirtualBox** é um software de virtualização desenvolvido pela empresa **Innotek** depois comprado pela Sun Microsystems que posteriormente foi comprada pela **Oracle** que, como o **VMware Workstation**, visa criar ambientes para instalação de sistemas distintos.
- Ele permite a instalação e utilização de um sistema operacional dentro de outro, assim como seus respectivos softwares, **como** dois ou mais computadores independentes, mas compartilhando fisicamente o mesmo hardware.

# MÁQUINA VIRTUAL - MITOS

- A máquina virtualizada não oferece riscos para a hospedeira?
- Os recursos da hospedeira são afetados pelas máquinas virtuais?
- A gestão de aplicativos usando máquinas virtuais fica facilitada nos ambientes corporativos?
- Virtualização viabiliza a criação de contingência com menores custos?



# ARQUIVO ISO



- **Permite abrir arquivos ISO para sua execução ou instalação.**
- Um arquivo ISO é uma cópia idêntica (imagem) de arquivos gravados num CD, DVD ou BD. Essas imagens são compostas pelo conteúdo total dos dados contidos num disco óptico, incluindo sistema de boot, número de setores gravados, sistema operacional e sistema de arquivos.
- Os softwares que criam e fazem a gravação de um arquivo ISO permitem também registrar essas imagens em outros dispositivos de armazenamento, como pendrives ou hard disks, para fins como backup ou duplicação de conteúdo em larga escala.
- Uma imagem ISO é especialmente útil para replicar mídias utilizadas na distribuição de software, áudio ou vídeo.
- Além disso, administradores de TI também utilizam o formato ISO para a criação de imagens dos programas instalados em computadores e servidores, facilitando a recuperação do ambiente em caso de desastres.

# VIRTUALBOX

- Possui versões para Windows, Linux e OS X.
- Versão atual 4.3.20
- O download pode ser feito em:
  - <https://www.virtualbox.org/wiki/Downloads>
- A instalação é bastante simples **NNF**.



# MÁQUINA VIRTUAL

- 1) Baixar um virtualizador (<https://www.virtualbox.org/>)
- 2) Instalar o virtualizar (sem grandes segredos)



- 1) Baixar a ISO de um SO, nesse exemplo estamos considerando o Ubuntu (<https://ubuntu.com/download/desktop>)
- 2) Instalar o Ubuntu a partir da ISO baixada conforme orientações a seguir.

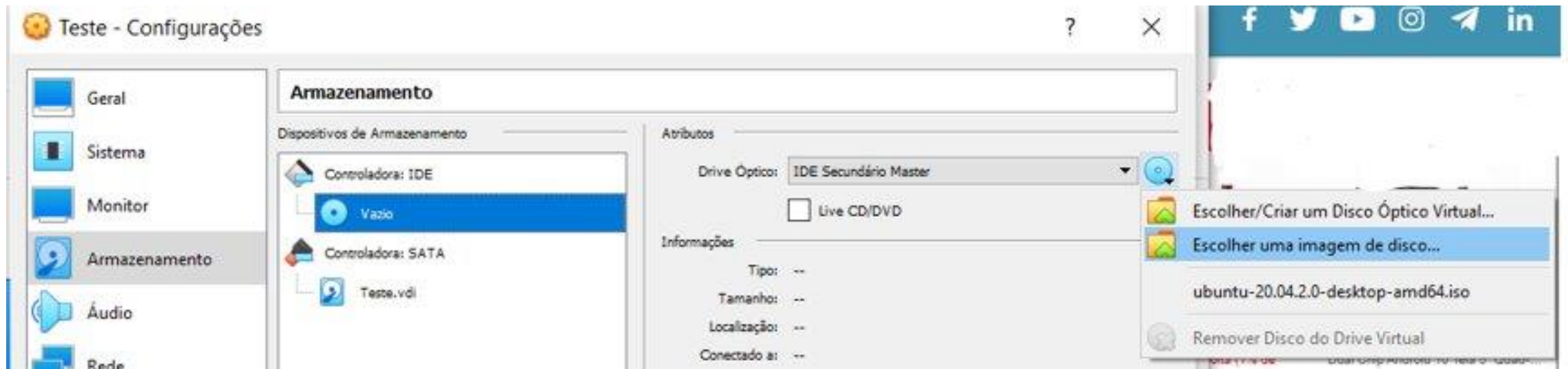
# MÁQUINA VIRTUAL


- 5) Com o Vbox instalado e iniciado, clicar em “Novo”
- 6) Nomear esta instalação, indicando onde ficará salva, sinalizar que será do tipo “Linux”, na versão “Ubuntu” 64 bits)
- 7) Na alocação de memória, no modo Live, funcionará com 1024Mb, mas ficará mais confortável ajustar para 2048Mb
- 8) Em relação ao disco, escolher a opção de “Criar um novo disco virtual agora”
- 9) O tipo de disco será VDI

# MÁQUINA VIRTUAL

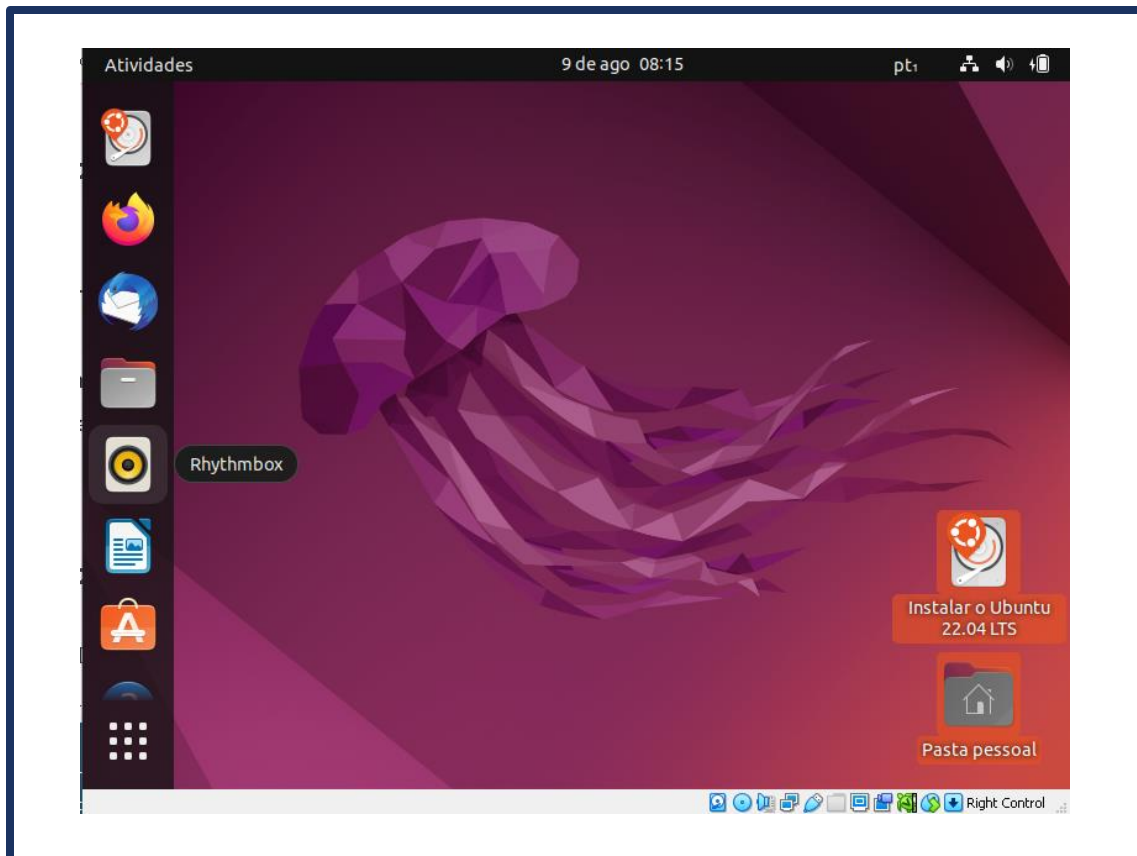
- 10) Escolher “Dinamicamente alocado”
- 11) Ajustar o tamanho do disco para 10Gb (suficiente para o teste)
- 12) Finalizado estes passos, devemos ir no menu configurações para indicar onde esta a ISO
- 13) No menu lateral esquerdo, escolher o item “Armazenamento”
- 14) E na controladora IDE, clicar em vazio para apontar o caminho da ISO, conforme figura a seguir.

# MÁQUINA VIRTUAL



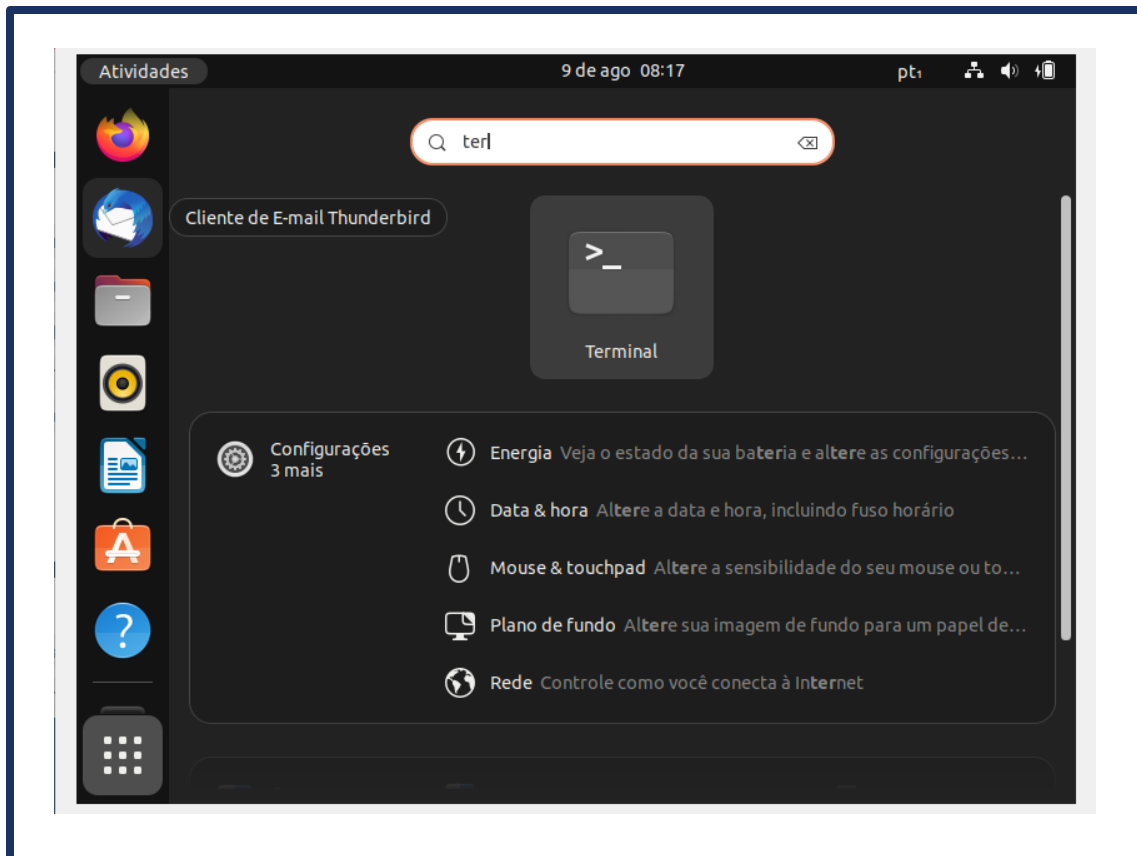
- 15) Indicar a ISO baixada e dar ok nesta tela
- 16) Iniciar o virtualizador  Iniciar (T)
- 17) Escolher a opção “Try or Install Ubuntu” para usar o SO diretamente em memória
- 18) Escolher a opção "Experimentar o Ubuntu"


# MÁQUINA VIRTUAL



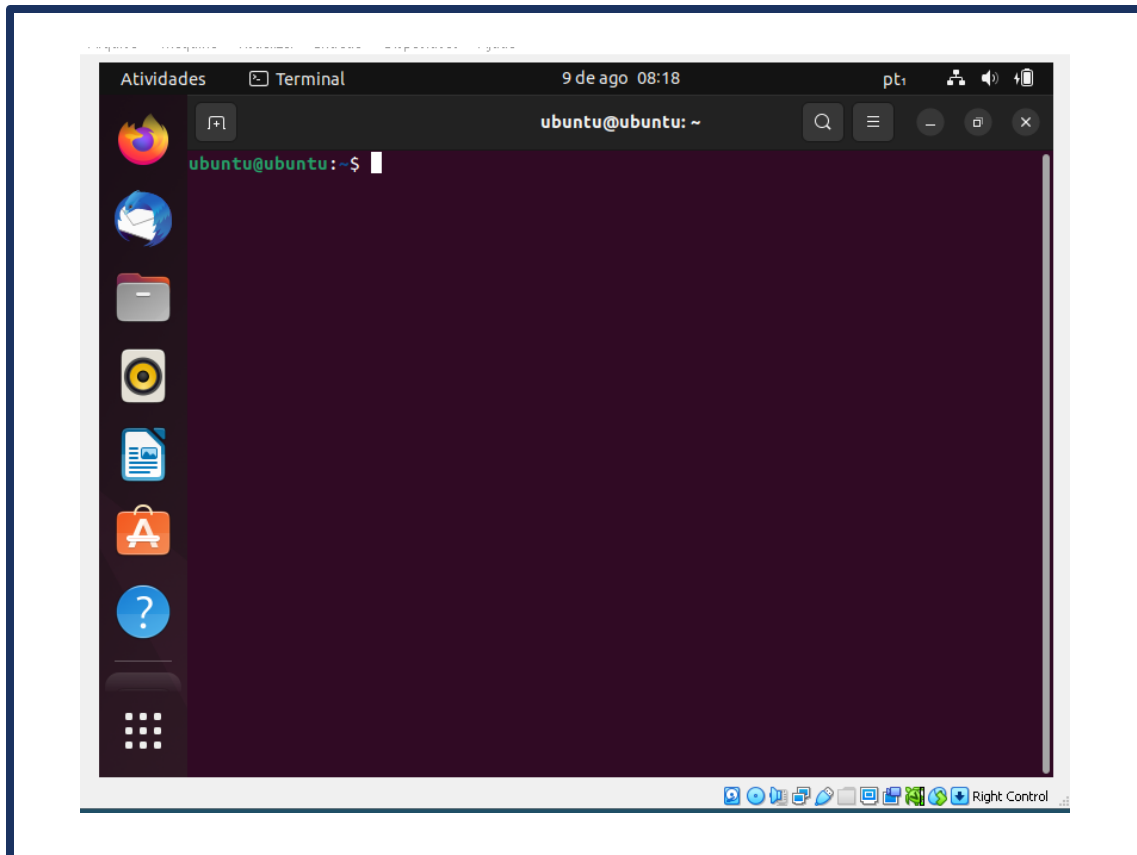
- Modo gráfico do Linux

# MÁQUINA VIRTUAL



- Acessar o modo texto do Linux
- Clique em 
- Digite **terminal**

# MÁQUINA VIRTUAL



- Modo texto para digitação dos comandos no Linux

# EMULADOR LINUX

- Rode o linux no seu navegador
- <https://bellard.org/jslinux/>

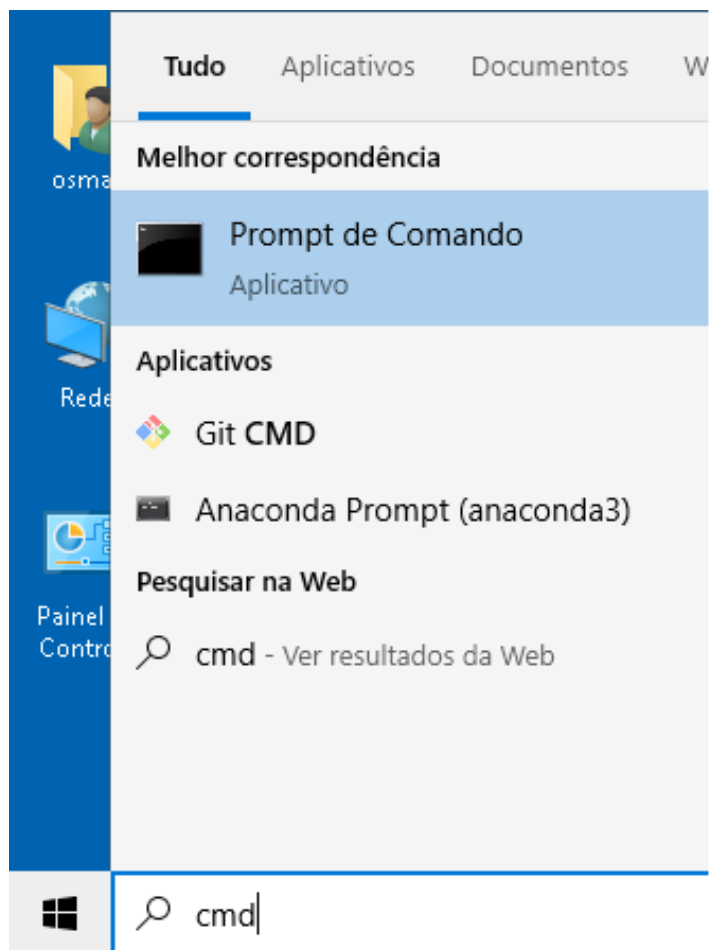
CPU	OS	User Interface	VEsync access	Startup Link	TEMU Config
x86	Alpine Linux 3.12.0	Console	Yes	<a href="#">click here</a>	<a href="#">url</a>
x86	Alpine Linux 3.12.0	X Window	Yes	<a href="#">click here</a>	<a href="#">url</a>
x86	Windows 2000	Graphical	No	<a href="#">click here</a>	<a href="#">url</a>
x86	FreeDOS	VGA Text	No	<a href="#">click here</a>	<a href="#">url</a>
riscv64	Buildroot (Linux)	Console	Yes	<a href="#">click here</a>	<a href="#">url</a>
riscv64	Buildroot (Linux)	X Window	Yes	<a href="#">click here</a>	<a href="#">url</a>
riscv64	Fedora 33 (Linux)	Console	Yes	<a href="#">click here</a>	<a href="#">url</a>
riscv64	Fedora 33 (Linux)	X Window	Yes	<a href="#">click here</a>	<a href="#">url</a>



## COMPARANDO COM O WINDOWS

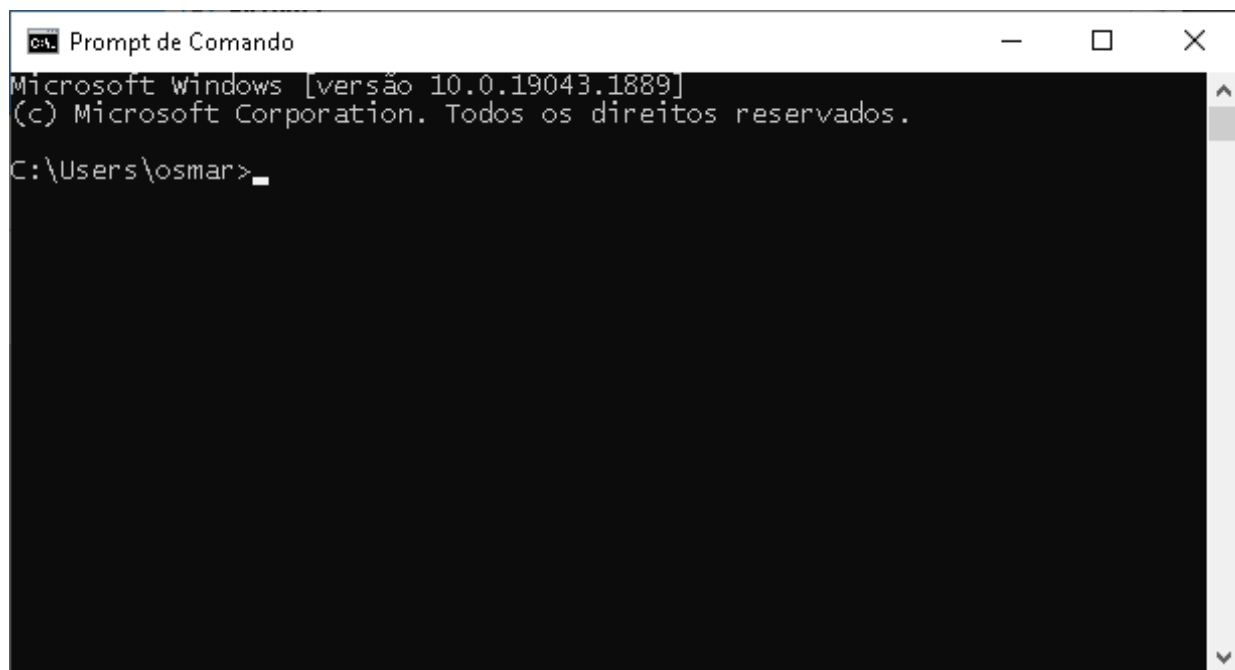
Vamos comparar os ambientes e comandos com o Windows.

# ACESSO MODO TEXTO WINDOWS



- Acessar o modo texto do windows
- Clique em iniciar e pesquisar
- Digite **cmd** (prompt de comando)

# ACESSO MODO TEXTO WINDOWS



A screenshot of the Windows Command Prompt window. The title bar reads "Prompt de Comando". The main content area shows the following text: "Microsoft Windows [versão 10.0.19043.1889]", "(c) Microsoft Corporation. Todos os direitos reservados.", and "C:\Users\osmar>". The window has a standard Windows interface with minimize, maximize, and close buttons in the title bar.

- Modo texto para digitação dos comandos no windows
- O terminal no Windows é conhecido como "**Prompt de Comando**",
  - PowerShell
  - Windows Terminal

# ATIVIDADE - I: MUNDO REAL X VIRTUAL

- 1) Na opinião do grupo, qual seria a grande vantagem de usar os sistemas operacionais virtualizados?
- 2) O ambiente virtualizado ofereceu acesso a todos os recursos computacionais da máquina hospedeira? Como você alocou os recursos do HD?  
(o espaço físico na VM ficou alocado)
- 3) No quesito segurança, um ataque na máquina hospedeira causaria problemas na máquina virtualizada?
- 4) No quesito segurança, de que maneira a fragilidade das proteções da máquina virtualizada impactaria a hospedeira?

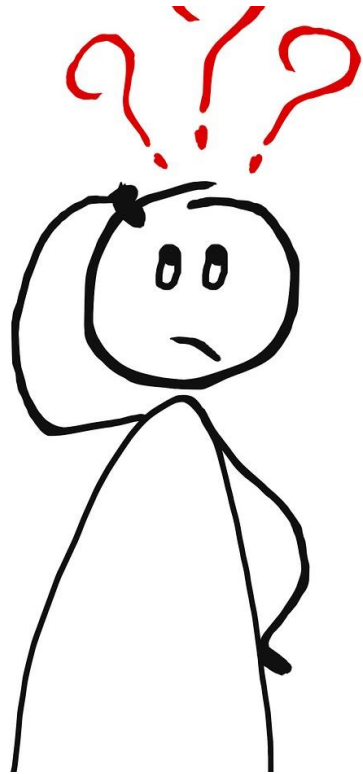
## ATIVIDADE – I : MUNDO REAL X VIRTUAL

- 5) No ambiente empresarial, em que cenários práticos seriam úteis a utilização dos Sistemas Operacionais virtualizados?
- 6) E para fins de estudos (universidade) e uso domiciliar, é possível destacar situações onde a virtualização seria adequada?
- 7) A questão do backup fica facilitada com o uso de Sistemas Virtualizados?
- 8) Qual a opinião do grupo em relação a recuperação de desastres em ambientes virtualizados. Indique vantagens e desvantagens.

# COMANDOS LINUX

- **Laboratório:Acionando Comandos do SO**

## UMA DÚVIDA...



- Esta mesma filosofia de linguagens de comando permanece ativa com as interfaces gráficas mais atualizadas?

# EXECUTANDO COMANDOS

- Digitar algum comando e pressionar Enter.
- O resultado do comando, na maioria das vezes, será exibido na tela.
- Nesse momento, várias coisas podem acontecer:
  - O comando é executado corretamente e uma mensagem de sucesso aparece.
  - O comando é executado corretamente porém nada é exibido (raro de acontecer).
  - O comando é executado e um erro é exibido.
  - O comando é executado, um erro acontece, porém nada é exibido na tela (muito raro de acontecer).



## CASE-SENSITIVE

A ≠ a

- Comandos são case-sensitive (sensíveis à caixa), ou seja, há diferença entre digitar um comando em maiúsculas e minúsculas.
  - Depende do SO
- Na grande maioria das vezes, tudo deve ser digitado com letra minúscula

# COMANDOS BÁSICOS LINUX X WINDOWS

- Alguns comandos são iguais (mesmo léxico) e função. Outros mudam o comando mas com mesma função.
- Digite no terminal do linux e windows o comando:
  - **dir**
  - **date**
  - **time**

# COMANDOS BÁSICOS LINUX X WINDOWS

- O linux é case-sensitive?
- O windows é case-sensitive?

# COMANDOS BÁSICOS LINUX

**Terminal**

**Prompt**

```
usuario@usuario-VirtualBox: ~  
usuario@usuario-VirtualBox:~$ echo $0  
bash  
usuario@usuario-VirtualBox:~$ whoami  
usuario  
usuario@usuario-VirtualBox:~$ hostname  
usuario-VirtualBox  
usuario@usuario-VirtualBox:~$
```

**Comandos**

# COMANDOS BÁSICOS LINUX

- **whoami:** Quem é o usuário
- **ls:** Lista todos os arquivos do diretório
- **clear:** Limpa a tela
- **df:** Mostra a quantidade de espaço usada no disco rígido
- **top:** Mostra o uso da memória(ctrl+c sair da exibição)
- **cd:** Acessa uma determinada pasta (diretório)
  - cd / - Leva você ao diretório raiz.
  - cd .. - Leva você até um nível de diretório.
  - cd - - Leva você ao diretório root
- **cat:** Abre um arquivo

# COMANDOS BÁSICOS LINUX

- **pwd:** (print working directory) exibe o caminho completo do diretório de trabalho atual.
- **cp:** (copy) permite copiar um arquivo. Você deve especificar tanto o arquivo que deseja copiar quanto o local onde deseja copiá-lo - por exemplo, `cp xyz / home / myfiles` copia o arquivo "xyz" para o diretório "/ home / myfiles".
- **mv:** (move) permite mover arquivos. Você também pode renomear arquivos, movendo-os para o diretório em que estão atualmente, mas com um novo nome. O uso é o mesmo que `cp-f` ou exemplo `mv xyz / home / myfiles` moveria o arquivo "xyz" para o diretório "/ home / myfiles".
- **rm:** (remove) remove o arquivo especificado.
- **mkdir:** ("make dir") - Cria um diretório vazio.
- **rmdir:** ("remove directory") - Remove um diretório vazio.
- **rm -r:** ("remove recursively") - Remove um diretório junto com seu conteúdo

# COMANDOS BÁSICOS LINUX

- Informações no prompt:
  - Nome do usuário
  - Hostname (nome do computador)
  - Diretório de trabalho atual
- **Detalhes:**
  - Dois pontos (:) no prompt é um separador para distinguir entre o nome do host e a localização atual.
  - Til (~) significa o diretório **Home** do usuário atual.
  - O prompt termina com o símbolo de dólar (\$). Para o usuário root, ele termina com símbolo de libra ou hash (#).

```
usuario@usuario-VirtualBox:~$ whoami
usuario
usuario@usuario-VirtualBox:~$ sudo su
[sudo] password for usuario:
root@usuario-VirtualBox:/home/usuario# whoami
root
root@usuario-VirtualBox:/home/usuario# pwd
/home/usuario
root@usuario-VirtualBox:/home/usuario# exit
exit
usuario@usuario-VirtualBox:~$ S
```

home

root

usuário

hostname

diretório atual

# COMANDOS BÁSICOS LINUX

- **free:** Exibe a quantidade de espaço livre disponível no sistema.
- **touch:** cria novos arquivos em branco através de uma linha de comando.
- **history:** exibe todos os seus comandos anteriores, até o limite do histórico.



# COMANDOS BÁSICOS LINUX - EDITOR DE TEXTOS VI

```
VIM - Vi IMproved

version 8.2.3995
by Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable

Help poor children in Uganda!
type  :help iccf<Enter>      for information

type  :q<Enter>             to exit
type  :help<Enter> or <F1>  for on-line help
type  :help version8<Enter> for version info

Running in Vi compatible mode
type  :set nocp<Enter>     for Vim defaults
type  :help cp-default<Enter> for info on this
```

- **vi**: Abre o editor vi (lê-se **viai**) para editar/criar arquivos
- **Entrar e Sair**
  - `vi arquivo.txt <ENTER>`
  - `:q <ENTER>`
- **Entrar, Inserir Texto, Salvar e Sair**
  - `vi arquivo.txt <ENTER>`
  - `i`
  - Minha primeira frase no VI.
  - `<ESC>`
  - `:wq <ENTER>`
- **Entrar, Alterar Texto e Sair sem Salvar**
  - `vi arquivo.txt <ENTER>`
  - `dd`
  - `:q! <ENTER>`

# COMANDOS BÁSICOS LINUX - EDITOR DE TEXTOS NANO

```
GNU nano 6.2      arquivo.txt *
SS
[ New File ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

**^= CTRL**

- **nano:** Abre o editor nano para editar/criar arquivos
- **Entrar e Sair**
  - nano arquivo.txt <ENTER>
  - ^x <ENTER>
- **Entrar, Inserir Texto, Salvar e Sair**
  - nano arquivo.txt <ENTER>
  - Minha primeira frase no VI.
  - <ESC>
  - ^o <ENTER>
  - ^x <ENTER>
- **Entrar, Alterar Texto e Sair sem Salvar**
  - vi arquivo.txt <ENTER>
  - dd
  - ^x <ENTER>
  - n
  - <ENTER>

## ATIVIDADE-II

As aspas não fazem parte do nome!

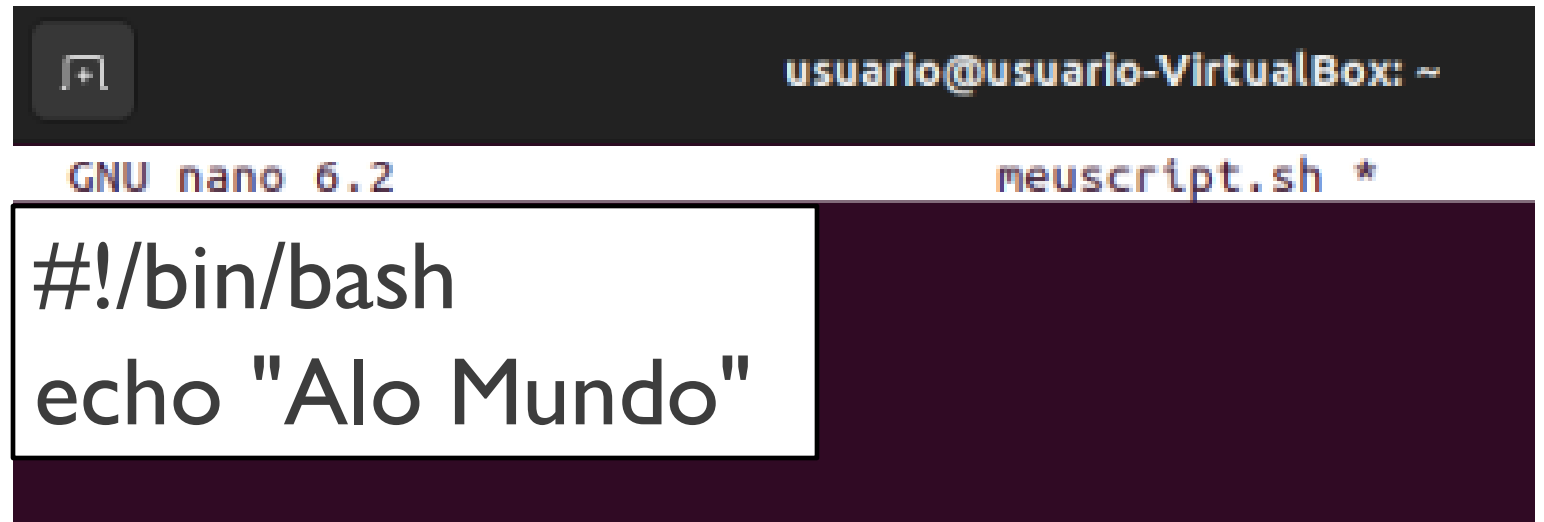
- 1) Logar no Linux com usuário, abrir o Shell (prompt) e verificar em qual diretório você se encontra.
- 2) Criar os seguintes diretórios dentro do diretório atual: "aula" e "trabalho". E dentro do diretório "aula" criar o diretório "ambiente".
- 3) Entrar na pasta aula.
- 4) Criar um arquivo chamado "numero.txt", usando um dos editores contendo o seguinte conteúdo:  
1  
3  
5  
7
- 5) Duplicar o arquivo "numeros.txt" para "numeros1.txt" e "numeros2.num".

## ATIVIDADE-II

- 6) Copiar os arquivos com a extensão ".txt" para a pasta "trabalho".
- 7) Exibir todos os arquivos com seus detalhes (permissões de acesso, data, hora de criação, tamanho)
- 8) Crie a pasta backup no diretório corrente e copie o conteúdo da pasta "trabalho" para ele.
- 9) Deletar os arquivos com a extensão ".txt" do diretório corrente.
- 10) Apagar a pasta "ambiente" dentro da pasta "aula"
- 11) Entrar na pasta "trabalho" e renomear o arquivo "numeros2.num" para "sequencia.txt"
- 12) Listar todos os arquivos da pasta "/bin" e guardar essa lista em um arquivo chamado "listabin.txt"

# SCRIPT EM LINUX

- Vamos criar o arquivo "meuscript.sh" para conter um pequeno bash script.
- Digite o comando: `$nano meuscript.sh`

A terminal window with a dark background. The top bar shows a window icon, the user 'usuario@usuario-VirtualBox', and a tilde '~'. Below the bar, the text 'GNU nano 6.2' is on the left and 'meuscript.sh \*' is on the right. The main area of the terminal is dark purple, with a white rectangular box highlighting the text: '#!/bin/bash' followed by 'echo "Alo Mundo"' on the next line.

```
usuario@usuario-VirtualBox: ~  
GNU nano 6.2 meuscript.sh *  
#!/bin/bash  
echo "Alo Mundo"
```

- Para executar o script digite:  
`$/meuscript.sh`

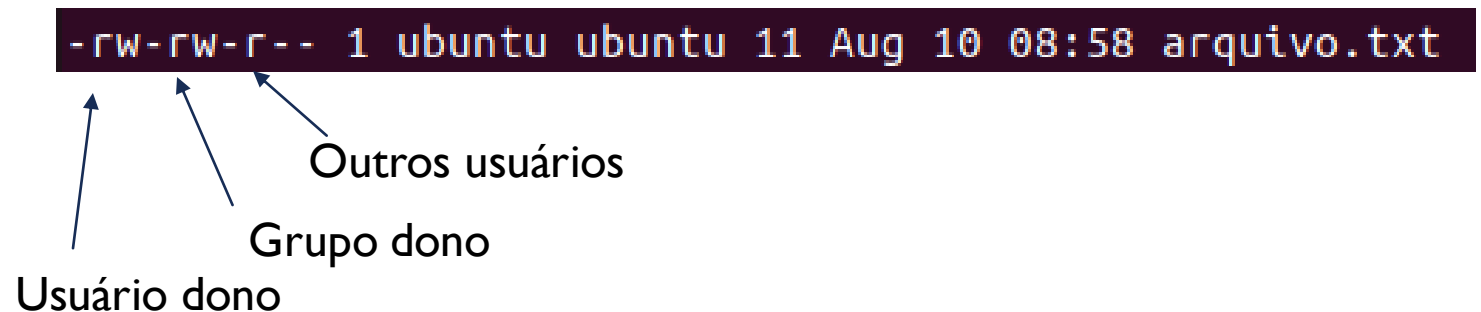
# PERMISSÕES DE ARQUIVOS LINUX

- No Linux, quando um arquivo ou diretório é criado, algumas permissões são atribuídas a ele automaticamente.
- Essas permissões são divididas em três grupos:
  - O usuário que criou o arquivo (usuário dono),
  - o grupo dono do arquivo (que pode conter vários usuários),
  - e as dos demais usuários que não pertencem ao grupo dono.
- Essas permissões podem ser de leitura (r, read), escrita (w, write) e execução (x, execute).
- A permissão de leitura (r) permite visualizar o conteúdo de um arquivo ou diretório, já a permissão de escrita (w) serve para alterar o conteúdo de um arquivo e diretório.
  - A opção de execução serve para tornar o arquivo um executável, ou, em caso de diretórios, para poder acessá-lo e usar comandos, como o cd.

# PERMISSÕES DE ARQUIVOS LINUX

- Para visualizar a permissão de um diretório, podemos ir até o local onde esse diretório se encontra.
- A lista retornada pelo comando ls é uma lista normal.
- **ls -l**: visualizar os arquivos e suas permissões temos que dizer para o ls que queremos uma lista longa (-l).

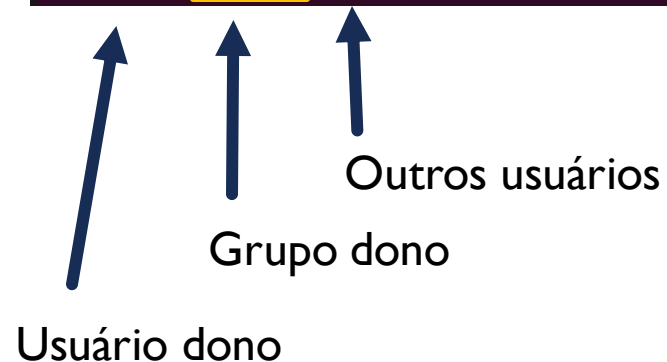
```
ubuntu@ubuntu:~$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu 11 Aug 10 08:58 arquivo.txt
drwxr-xr-x 2 ubuntu ubuntu 60 Aug 10 08:53 Desktop
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Documents
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Downloads
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Music
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Public
drwx----- 3 ubuntu ubuntu 60 Aug 10 08:54 snap
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Videos
ubuntu@ubuntu:~$
```



# PERMISSÕES DE ARQUIVOS LINUX

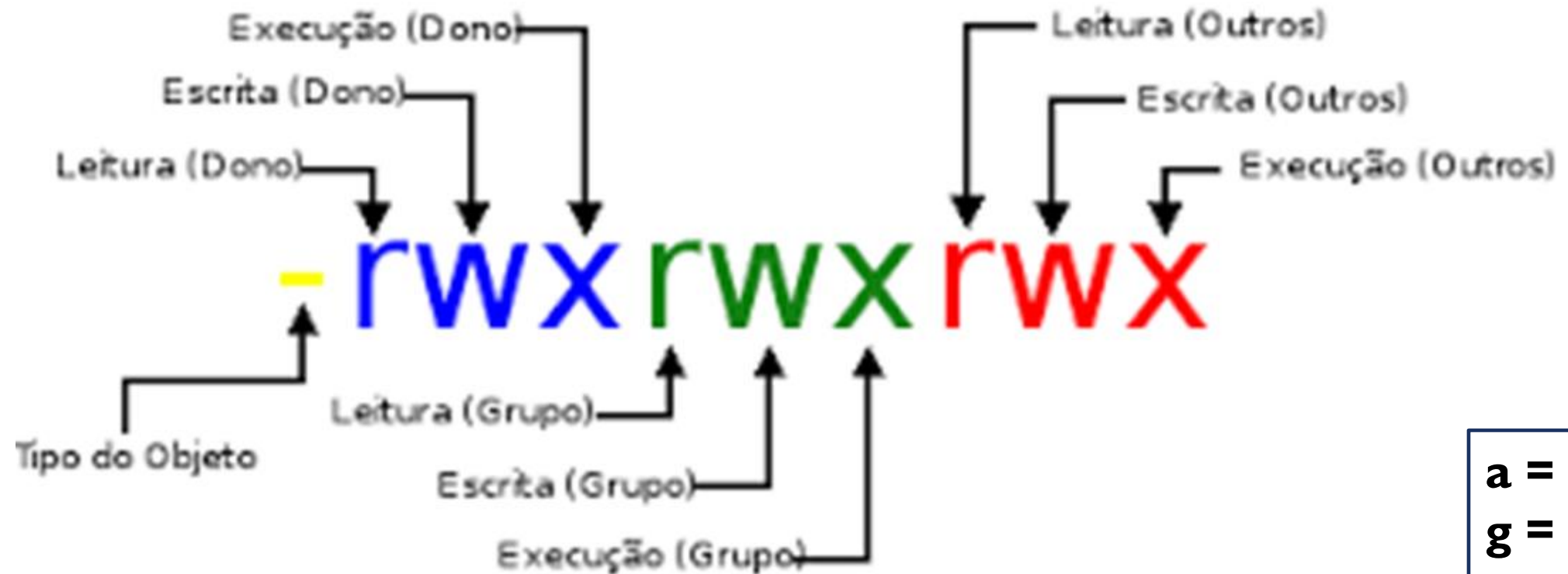
- A primeira coluna é dividida em dez caracteres. O primeiro nos mostra se o objeto listado é um arquivo (-) ou um diretório (d).
- Caso tenham alguma permissão, é mostrado a letra correspondente a permissão, senão um traço (-) é apresentado.
- Os nove caracteres restantes nos mostram as permissões do usuário dono, grupo dono e outros usuários respectivamente:

```
-rw-rw-r-- 1 ubuntu ubuntu 11 Aug 10 08:58 arquivo.txt
```





# PERMISSÕES DE ARQUIVOS LINUX



**a = all**  
**g = group**  
**u = user**  
**o=other**

# PERMISSÕES DE ARQUIVOS LINUX

Permissão do arquivo	Comando chmod aplicado	Nova permissão	Significado
<b>rw-r--r--</b>	<b>chmod a+x arq1</b>	<b>rwX-r-Xr-X</b>	Permissão de execução adicionada a todos
<b>rw-r--r--</b>	<b>chmod g=u arq1</b>	<b>rw-rw-r--</b>	Permissão do grupo configurada igual à do proprietário.
<b>r--r--r--</b>	<b>chmod ug=rw arq1</b>	<b>rw-rw-r--</b>	Permissões do proprietário e do grupo ajustadas em leitura e escrita.
<b>rw-----</b>	<b>chmod u+x,g=rx arq1</b>	<b>rwXr-X---</b>	Permissão de execução adicionada ao proprietário, e leitura e execução ajustadas ao grupo.
<b>rwXrw-rw-</b>	<b>chmod o-w arq1</b>	<b>rwXrw-r--</b>	Permissão de gravação retirada dos outros.
<b>r--r--r--</b>	<b>chmod a+wx arq1</b>	<b>rwXrwXrwX</b>	Permissões de gravação e execução adicionadas a todos

**a = all**  
**u = user**  
**g = group**  
**o=other**

# PERMISSÕES DE ARQUIVOS LINUX - MODO OCTAL

- O modo **octal** recebe este nome, pois utilizamos oito números, de 0 à 7, cada um desses números correspondem a uma letra, ou a um conjunto de letras, no modo simbólico:
  - Representa a opção de execução (x) no modo simbólico;
  - A opção de escrita (w);
  - A opção de leitura (r).
- Quando utilizamos o modo **octal**, podemos passar o modo de permissões de cada grupo de usuários. A ordem é sempre: **usuário dono, grupo dono e outros usuários**.
- Exemplo, colocar a permissão de leitura para o usuário dono (4), para o grupo dono a de escrita (2) e a de execução para os demais usuários (1):
  - **chmod 421 arquivo.txt**
- Equivale a
  - **chmod u=r,g=w,o=x arquivo.txt**

# PERMISSÕES DE ARQUIVOS LINUX

Octal	Binário	Letras	Descrição
0	000	---	Sem acesso
1	001	--x	Somente Execução
2	010	-w-	Somente Escrita
3	011	-wr	Somente Escrita e execução
4	100	r--	Somente leitura
5	101	r-x	Somente leitura e execução
6	110	rw-	Somente leitura e escrita
7	111	rwX	Leitura, escrita e execução

# PERMISSÕES DE ARQUIVOS LINUX

- Para alterar as permissões de um arquivo ou diretório, temos que alterar o modo (**chmod**, *change mode*) que elas estão organizadas.
- Queremos que os **outros usuários** (o) tenham acesso de escrita(w) no diretório, assim conseguirão editar o arquivo. Então podemos dizer para o terminal:
- **chmod o=w arquivo.txt**
- **Chmod 2 arquivo.txt**

Outros usuários

```
ubuntu@ubuntu:~$ chmod o=w arquivo.txt
ubuntu@ubuntu:~$ ls -l
total 4
-rw-rw--w- 1 ubuntu ubuntu 11 Aug 10 08:58 arquivo.txt
drwxr-xr-x 2 ubuntu ubuntu 60 Aug 10 08:53 Desktop
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Documents
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Downloads
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Music
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Public
drwx----- 3 ubuntu ubuntu 60 Aug 10 08:54 snap
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 Aug 10 08:55 Videos
ubuntu@ubuntu:~$
```

# PERMISSÕES DE ARQUIVOS LINUX

- Se precisar alterar a permissão de todos os arquivos, diretórios e subdiretórios, precisamos fazer um por um?
  - Não precisamos!
- O `chmod` tem uma opção que faz isso para a gente. Essa opção muda a permissão dos arquivos recursivamente. Isto é, ela muda a permissão de todos os arquivos em todo o diretório e subdiretórios.
- O `chmod` mudar a permissão de maneira recursiva (-R):
  - **`chmod -R o=rx Videos/`**
  - **`Chmod -R 5 Videos/`** (101 binário)

## ATIVIDADE - III

- 1) Ir para a raiz do disco e localizar o arquivo "sequencia.txt"
- 2) Mudar a permissão de acesso do arquivo "sequencia.txt" para `-rwxr-xr-x`
- 3) Mudar a permissão de acesso do arquivo "numeros.txt" para `-rw-r--r--`

# INSTALAÇÃO DE APLICATIVOS LINUX

- Usando o apt-get
  - `sudo apt-get install <nome_do_programa>`
    - \$ indica que você deve usar o usuário comum para fazer essa operação.
    - sudo serve para pedir permissões de administrador temporariamente.
    - apt do inglês, *Advanced Package Tool*, em português, Ferramenta de Empacotamento Avançada; é a ferramenta que nos ajuda na instalação, atualização e desinstalação de programas, entre outras funções.
    - install é o comando de instalar, indicando ao apt-get o que fazer.
    - digite o nome do programa sem os sinais < e >.



## ATIVIDADE - IV

- 1) `sudo apt-get install sl`  
Digite `sl`
- 2) `sudo apt-get install cmatrix`  
Digite: `cmatrix`
- 3) `sudo apt-get install cowsay`  
Digite: `cowsay`

# CRIAÇÃO DE PROCESSOS

- Um processo pode ser criado de diversas maneiras. Seguem as três principais formas de criação de processos:
  - Logon interativo;
  - Via linguagem de comando;
  - Via rotina do Sistema Operacional.

# CRIAÇÃO DE PROCESSOS (LOGON INTERATIVO)

- No logon interativo o usuário, por intermédio de um terminal, fornece ao sistema um nome de identificação (username ou logon) e uma senha (password);
- O sistema operacional autentica estas informações verificando se estão corretamente cadastradas no arquivo de usuários;
- O arquivo de usuários é um arquivo do sistema operacional onde são armazenados todos os usuários autorizados a ter acesso ao sistema;
- Privilégios, quotas, permissões e o logoff.

# GERENCIANDO PROCESSOS

- O LINUX é um sistema operacional multitarefa, executa diversos processos simultaneamente. Um processo é um programa em execução. Cada processo possui um identificador e um processo pode chamar outro processo.
- Os principais comandos para se trabalhar com processos são os seguintes:
  - ps
  - kill
  - top

# GERENCIANDO PROCESSOS

- `ps` – mostra os processos ativos no sistema.
  - `-a` – mostra todos os processos.
  - `-u` – inclui na lista os usuários e hora do início.
  - `-x` – inclui processos não associados a um terminal.
  - `-p PID` - exibe o processo cujo número é PID
  - `-l` - exibe linhas detalhadas
  - `-e` exibe todos os processos ativos
  - Ex: `ps -aux` – Exibe todos os processos, inclusive os não associados a terminais, mostrando os usuários que iniciaram.
- `kill` – permite que o superusuário ou o dono dos processos possam eliminar processos ativos.
  - Ex: `kill -9 1023` – elimina o processo com PID 1023.

# GERENCIANDO PROCESSOS

- top – mostra uma lista atualizada periodicamente dos processos ativos do sistema.
- Digite: \$top
  - 1 - linha tempo do sistema, tempo de atividade e sessões do usuário
  - 2 - Tarefas em execução
  - 3- Utilização da CPU
  - 4 -Utilização da memória

```
top - 14:55:20 up 49 min, 1 user, load average: 0,01, 0,06, 0,17
Tasks: 179 total, 1 running, 177 sleeping, 1 stopped, 0 zombie
%Cpu(s):  1,1 us,  2,1 sy,  0,0 ni, 96,1 id,  0,0 wa,  0,0 hi,  0,7 si,  0,0 st
MiB Mem : 1976,0 total, 287,9 free, 641,0 used, 1047,1 buff/cache
MiB Swap: 2048,0 total, 1891,1 free, 156,9 used. 1165,1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
976	usuario	20	0	4013064	232536	75760	S	5,6	11,5	1:21.04	gnome-+
2201	usuario	20	0	562356	40004	28352	S	0,7	2,0	0:09.99	gnome-+
7567	root	20	0	0	0	0	I	0,3	0,0	0:01.53	kworke+
<b>9189</b>	<b>usuario</b>	<b>20</b>	<b>0</b>	<b>13092</b>	<b>3952</b>	<b>3340</b>	<b>R</b>	<b>0,3</b>	<b>0,2</b>	<b>0:00.02</b>	<b>top</b>
1	root	20	0	167968	10600	5680	S	0,0	0,5	0:43.09	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthrea+
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_pa+
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworke+
9	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mn_per+
10	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_ta+
11	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_ta+
12	root	20	0	0	0	0	S	0,0	0,0	0:04.69	ksofti+
13	root	20	0	0	0	0	I	0,0	0,0	0:01.11	rcu_sc+
14	root	rt	0	0	0	0	S	0,0	0,0	0:00.05	migrat+
15	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_i+
17	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtm+
19	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	inet_f+
20	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd
21	root	20	0	0	0	0	S	0,0	0,0	0:00.00	khunqt+

# CRIAÇÃO DE PROCESSOS (ROTINA DO SO)

```
procedure TForm1.CriaProcesso(Sender: TObject);
var
  status: boolean;
  si: STARTUPINFO;
  pi: PROCESS_INFORMATION;
  Handle: THandle;
  NomeExe: PChar;
begin
  NomeExe := PChar('\WINNT\notepad.exe');
  FillChar(si, SizeOf(si), 0);
  si.cb := SizeOf(si);
  status := CreateProcess(NomeExe, nil, nil, nil, TRUE,
    NORMAL_PRIORITY_CLASS, nil, nil, si, pi);
  if (not status) then MessageBox (Handle, 'Erro na criação do
    processo', nil, MB_OK);
end;
```

- Um processo pode ser criado a partir de qualquer programa executável com o uso de rotinas do sistema operacional.

# GERENCIANDO PROCESSOS

- jobs – exibe os processos parados ou em execução que se encontram em segundo plano.
  - jobs -l

```
usuario@usuario-VirtualBox:~$ jobs -l
[3] 3460 Stopped (tty output) ftp www.ftp.com
[4] 4114 Running firefox &
[5] 6424 Stopped (tty output) cmatrix
[6]- 7532 Stopped (tty output) ftp www.ftp.com
[7]+ 7539 Stopped (tty output) vi
usuario@usuario-VirtualBox:~$
```

Se existir  
algum  
processo em  
execução.



# GERENCIANDO PROCESSOS

- Para colocar um processo para executar em segundo plano, basta digitar o comando seguido do símbolo **&**.

- `vi&`
- `ftp www.ftp.com &`

Quais são os processos em execução?

- **Ctrl+Z**, suspende o processo que está executando em primeiro plano
- **Ctrl+C**, aborta um processo
- **&**, executa um comando em coloca em segundo plano (*background*)

```
usuario@usuario-VirtualBox:~$ jobs -l
usuario@usuario-VirtualBox:~$ vi &
[1] 9087
usuario@usuario-VirtualBox:~$ ftp www.ftp.com &
[2] 9088

[1]+  Stopped                  vi
usuario@usuario-VirtualBox:~$
```

# GERENCIANDO PROCESSOS

- Execute o comando para matar o primeiro processo da lista.

```
usuario@usuario-VirtualBox:~$ jobs -l
[1]-  9087 Stopped (tty output)      vi
[2]+  9088 Stopped (tty output)      ftp www.ftp.com
usuario@usuario-VirtualBox:~$ kill -9 9087
usuario@usuario-VirtualBox:~$ jobs -l
[1]-  9087 Killed                    vi
[2]+  9088 Stopped (tty output)      ftp www.ftp.com
usuario@usuario-VirtualBox:~$
```

# ATIVIDADES: MONITORANDO PROCESSOS NO SISTEMA OPERACIONAL LINUX

- ps: é a principal ferramenta do administrador de sistemas para o monitoramento dos processos. Podemos usá-lo para exibir o PID, o UID, a prioridade e o terminar de controle dos processos.
- Podemos obter uma visão geral dos processos em execução no sistema
- Digite: \$ ps -aux
- 1. Identifique qual o primeiro processo pai inicializado, qual seu usuário e PID: \_\_\_\_\_
- 2. Qual o processo que está utilizando mais memória no momento:  
\_\_\_\_\_
- 3. O campo STAT identifica o estado de cada processo, S(Sem atividade), (R (executável), D(espera), T(interrompido). Qual o estado do primeiro e último processo que estão rodando em sua máquina: \_\_\_\_\_

# GERENCIANDO PROCESSOS

- `bg` – permite reativar um processo que está suspenso e coloca para executar em segundo plano
  - `bg %JobId`
  - Ex. `bg 1`
- `fg` – ao contrário do comando `bg`, o `fg` (foreground) coloca o job em primeiro plano.
  - `fg %JobId`
  - Ex. `fg 2`

# GERENCIANDO PROCESSOS

- Campo STAT:
  - S : sleeping (sem atividade);
  - R : running (executando);
  - D : waiting (aguarda um dispositivo de I/O);
  - T : gestopt (suspensão, parado);
  - Z : zombie(orfão);

```
usuario@usuario-VirtualBox:~$ ps -aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.5 167968 10596 ?        Ss   14:05   0:02 /sbin/init s
root           2  0.0  0.0     0     0 ?        S    14:05   0:00 [kthreadd]
root           3  0.0  0.0     0     0 ?        I<   14:05   0:00 [rcu_gp]
root           4  0.0  0.0     0     0 ?        I<   14:05   0:00 [rcu_par_gp]
root           5  0.0  0.0     0     0 ?        I<   14:05   0:00 [netns]
root           7  0.0  0.0     0     0 ?        I<   14:05   0:00 [kworker/0:6
root           9  0.0  0.0     0     0 ?        I<   14:05   0:00 [mm_percpu_w
root          10  0.0  0.0     0     0 ?        S    14:05   0:00 [rcu_tasks_r
root          11  0.0  0.0     0     0 ?        S    14:05   0:00 [rcu_tasks_t
root          12  0.1  0.0     0     0 ?        S    14:05   0:04 [ksoftirqd/0
root          13  0.0  0.0     0     0 ?        I    14:05   0:01 [rcu_sched]
root          14  0.0  0.0     0     0 ?        S    14:05   0:00 [migration/0
root          15  0.0  0.0     0     0 ?        S    14:05   0:00 [idle_inject
root          17  0.0  0.0     0     0 ?        S    14:05   0:00 [cpuhp/0]
root          18  0.0  0.0     0     0 ?        S    14:05   0:00 [kdevtmpfs]
root          19  0.0  0.0     0     0 ?        I<   14:05   0:00 [inet_frag_w
root          20  0.0  0.0     0     0 ?        S    14:05   0:00 [kauditd]
root          21  0.0  0.0     0     0 ?        S    14:05   0:00 [khungtaskd]
root          22  0.0  0.0     0     0 ?        S    14:05   0:00 [oom_reaper]
root          23  0.0  0.0     0     0 ?        I<   14:05   0:00 [writeback]
root          24  0.0  0.0     0     0 ?        S    14:05   0:00 [kcompactd0]
root          25  0.0  0.0     0     0 ?        SN   14:05   0:00 [ksmd]
root          26  0.0  0.0     0     0 ?        SN   14:05   0:00 [khugepaged]
root          72  0.0  0.0     0     0 ?        I<   14:05   0:00 [kintegrityc
root          73  0.0  0.0     0     0 ?        I<   14:05   0:00 [kblockd]
root          74  0.0  0.0     0     0 ?        I<   14:05   0:00 [blkcg_punt_
```

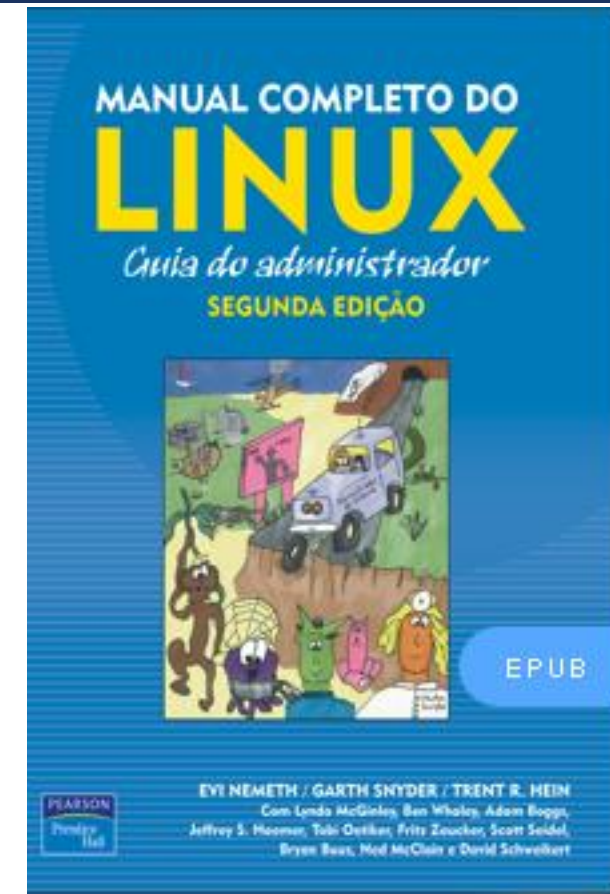
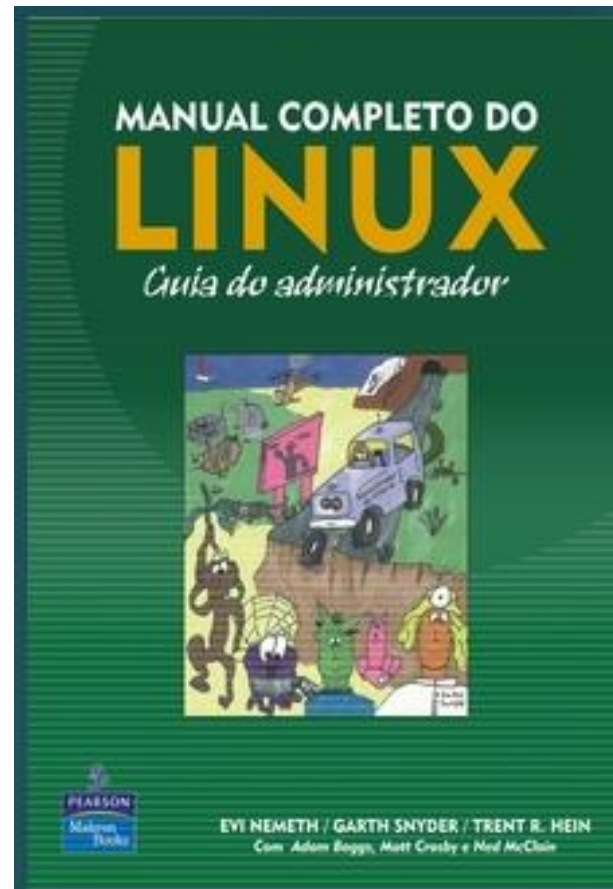
# ATIVIDADES: MONITORANDO PROCESSOS NO SISTEMA OPERACIONAL LINUX

- O comando top fornece um sumário atualizado dos processos ativos e o emprego de seus recursos.
- Digite %top
- I. Quantos processos estão dormindo, rodando e parados esperando o processador? \_\_\_\_\_
- Suponha que um usuário tenha iniciado um processo de longa duração que está consumindo uma fração significativa da máquina.
- II. Como você reconheceria um processo que está exaurindo recursos?

# CONCLUSÃO

- Conhecemos a virtualização
- Conhecemos os comandos básicos do linux e alguns do windows.
- Acessem os links das referências para conhecer outros comandos.
- A necessidade vai dizer quais serão importantes.

# LIVROS COMPLEMENTARES



<https://plataforma.bvirtual.com.br/Acervo/Publicacao/286>

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/787>

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/268>



# REFERÊNCIAS

- COMER, Douglas E. Redes de Computadores e Internet. Porto Alegre: Bookman, 2016. <https://integrada.minhabiblioteca.com.br/reader/books/9788582603734/>
- TANENBAUM, Andrew. Redes de Computadores. 5.ed. São Paulo: Campus, 2011. <https://plataforma.bvirtual.com.br/Acervo/Publicacao/2610>
- MORAES, Alexandre Fernandes de; Redes de computadores. -- 1. ed. -- São Paulo : Érica, 2014. <https://integrada.minhabiblioteca.com.br/reader/books/9788536532981/>

FIM UNIDADE I