

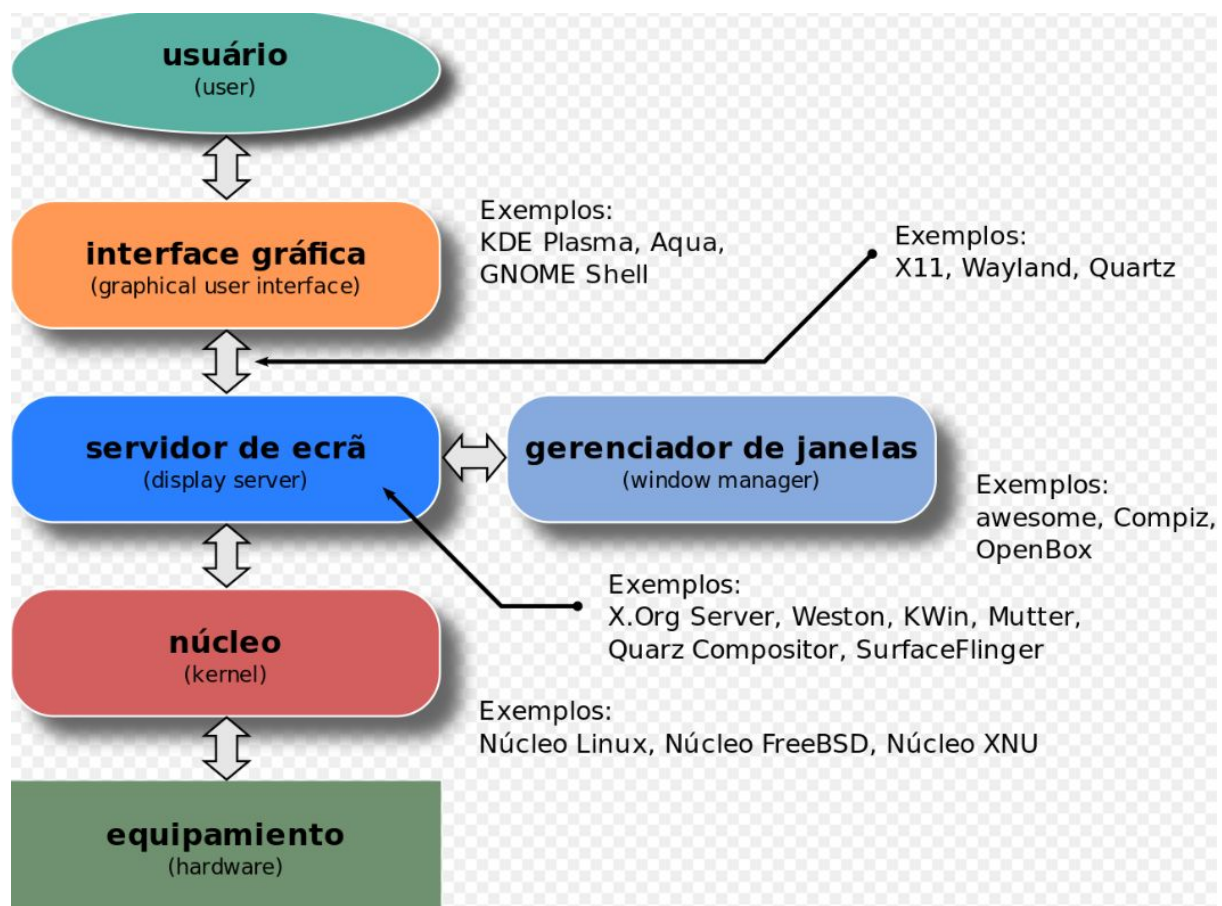
**Aluno : Paulo Eduardo Gobor**

## **C8: Sistemas Operacionais**

**Pesquisa sobre os sistemas operacionais Windows, Linux, MacOS e Android, apresentando:**

**°Interfaces de interação com usuários para as aplicações (comando, gráfica)**

### **GUI**



Na computação, uma interface gráfica do usuário ( GUI , às vezes pronunciada pegajosa ) é um tipo de interface do usuário que permite aos usuários interagir com dispositivos eletrônicos com imagens, em vez de comandos de texto. As GUIs podem ser usadas em computadores, dispositivos portáteis como MP3 players, media players portáteis ou dispositivos de jogos, eletrodomésticos e equipamentos de escritório. Uma GUI representa as informações e ações disponíveis para um usuário por meio de ícones gráficos e

indicadores visuais, como notação secundária, em oposição a interfaces baseadas em texto, rótulos de comando digitados ou navegação de texto. As ações geralmente são executadas através da manipulação direta dos elementos gráficos.

O termo GUI é historicamente restrito ao escopo de telas bidimensionais com resoluções capazes de descrever informações genéricas, na tradição da pesquisa em ciência da computação no Xerox PARC (Centro de Pesquisa Palo Alto). O termo GUI anterior pode ter sido aplicável a outros tipos de interfaces de alta resolução que não são genéricos, como videogames ou não se restringem a telas planas, como displays volumétricos.

A interação é feita geralmente com um mouse ou um teclado, com os quais o usuário é capaz de selecionar símbolos e manipulá-los de forma a obter algum resultado prático.

Esses símbolos são designados de widgets e são agrupados em kits.

Ambiente gráfico é um software feito para facilitar e tornar prática a utilização do computador por meio de representações visuais do sistema operacional.

Uma interface gráfica do utilizador usa uma combinação de tecnologias e dispositivos para fornecer uma plataforma com a qual o utilizador pode interagir.

Em computadores pessoais, a combinação mais conhecida é o WIMP, que consiste de janelas, ícones, menus e ponteiros. Nesse sistema, utiliza-se um dispositivo de ponteiro como o rato para controlar a posição do cursor e apresentar informação organizada em janelas e representada por meio de ícones. Os comandos disponíveis são compilados por menus e acionados mediante dispositivo de ponteiro. Um gerenciador de janela facilita a interação entre janelas, aplicações e o sistema de janelas, este, responsável por lidar com os dispositivos de hardware como o dispositivo de ponteiro e o hardware gráfico.

A simulação proporcionada pelos gerenciadores de janelas, incluindo a interação entre janelas e outros elementos gráficos, produz um ambiente de desktop.

Dispositivos móveis como PDAs e smartphones também usam elementos do WIMP mas com outros tipos de metáforas, devido às limitações de recurso do próprio dispositivo.

Para Windows temos apenas o ambiente gráfico padrão, nas versões Windows Vista e Windows 7 temos a chamada Windows Aero.

No **GNU/Linux** temos vários ambientes gráficos, entre eles, o KDE, Gnome, BlackBox, Xfce, LXDE, etc.. Há também a opção de não precisar usar ambientes gráficos. Para prover a funcionalidade do ambiente gráfico existem programas como X.org, XFree86.

Podemos usar como exemplo o projeto KDE, que é responsável por gerenciar várias aplicações diferentes, desde aplicações como calculadora, calendário, o nosso querido Kdenlive, utilizado para editar os vídeos do canal e é claro a interface gráfica KDE Plasma, ou somente “Plasma”, como os próprios desenvolvedores do KDE convencionaram chamar. Sendo assim, o desktop environment KDE é composto do Plasma juntamente com outras ferramentas desenvolvidas pelo projeto KDE.

Apesar de usarmos o KDE como exemplo, o mesmo vale para, virtualmente, qualquer outro projeto. Além disso, você pode utilizar uma interface gráfica, mas com aplicações de projetos diferentes. Uma prova disso, é utilizar uma distro como o Linux Mint, com a interface Cinnamon, mas com aplicações GNOME, como o Monitor do Sistema, ou seja, você usa parte de um desktop environment em uma distro cujo ambiente gráfico corresponde a outro diferente.

No MAC O OS X, cujo X é o número romano de 10 e é uma parte proeminente de sua identidade de marca, baseia-se nas tecnologias desenvolvidas na NeXT entre a segunda metade da década de 1980 e na compra da empresa pela Apple no final de 1996. O 'X'

também é usado para enfatizar a relação entre OS X e UNIX. As versões 10.5 "Leopard" em execução nos processadores Intel, 10.6 "Snow Leopard", 10.7 "Lion", 10.8 "Mountain Lion" e 10.9 "Mavericks" obtiveram a certificação UNIX 03. O iOS, que roda no iPhone, iPod Touch, iPad e Apple TV de segunda e terceira geração, compartilha o núcleo de Darwin e muitas estruturas com o OS X. Uma variante sem nome da v10.4 alimentou a Apple TV de primeira geração.

O Mac OS X 10.7 "Lion" foi a primeira versão do OS X a abandonar o suporte a processadores Intel de 32 bits e rodar exclusivamente em CPUs Intel de 64 bits.

## **Android**

Para o usuário, a visualização e a interação estão limitadas à interface do usuário. O Android oferece diversos componentes de IU incorporados, como objetos de layout e controles de IU estruturados que permitem criar a interface gráfica do usuário do aplicativo. O Android também disponibiliza outros módulos de IU para interfaces especiais como caixas de diálogo, notificações e menus.

Os aplicativos Android usam uma variedade de recursos de plataforma padrão para projetar e programar interfaces de usuário. Os aplicativos Android usam classes na linguagem Java para implementar vários aspectos da funcionalidade da Interface Gráfica do Usuário (GUI). Os desenvolvedores podem usar essas classes para criar um comportamento padrão em seus aplicativos. Além dos arquivos de classe Java, os aplicativos Android usam dados XML para declarar layouts de tela.

Cada tela em um aplicativo Android é normalmente representada no código de programação do aplicativo como uma Atividade. Isso envolve estender a classe Activity nos arquivos de classe de um aplicativo, instruindo o dispositivo do usuário a tratar o lançamento da Activity como uma nova tela na interface do usuário. As atividades podem conter vários elementos visuais e interativos e podem implementar métodos padrão para lidar com usuários pressionando botões e itens de menu. Os desenvolvedores também podem passar dados para o Activities quando iniciá-los, usando métodos da classe Intent. A plataforma Android fornece aos desenvolvedores um conjunto padrão de classes de interação para lidar com a entrada do usuário. Os desenvolvedores podem incluí-los no código Java ou XML. Cada atividade em um aplicativo também pode usar um menu de opções, que aparece quando os usuários selecionam o botão de menu em seus dispositivos. Dentro de uma Atividade, os desenvolvedores podem implementar funções de ouvinte e manipulador para interação do usuário com elementos específicos do aplicativo, como o menu de contexto que pode aparecer quando os usuários "pressionam" um item. Ao criar um botão em uma Atividade, o desenvolvedor pode fornecer um ouvinte "onClick" para detectar e gerenciar a interação do usuário com ele.

## **Linha de Comando**

O Prompt de Comando é um aplicativo de linha de comando disponível na maioria dos sistemas operacionais Windows, e é usado para executar os comandos inseridos. Esses comandos têm como função automatizar tarefas por meio de scripts e arquivos em lotes, executando desde funções administrativas avançadas até a solução de certos tipos de

problemas do sistema operacional. O Prompt de Comando também é conhecido como shell de comando, ou prompt do cmd, ou ainda cmd.exe.

#### Diferença do CMD e interface gráfica

Para saber o que é o prompt de comando do Windows, é necessário primeiramente entender a diferença entre ele e a interface gráfica. No Windows, quando você liga o computador, verá a interface gráfica. Normalmente, inserimos os comandos no Windows através de cliques do mouse. Por exemplo, ao clicar no ícone do navegador da Internet na área de trabalho, você dá uma ordem ao PC. Já o CMD é uma ferramenta do PC através da qual podemos inserir pedidos usando comandos de texto.

#### Diferença do CMD e do MS-DOS

Abreviação de Microsoft Disk Operating System, o MS-DOS é um sistema operacional de linha de comando não gráfico, derivado do 86-DOS, criado para computadores compatíveis com IBM. Hoje, o MS-DOS não é mais usado; no entanto, o shell de comando, mais conhecido como linha de comando do Windows, ainda é usado por muitos usuários.

O MS-DOS originalmente escrito por Tim Paterson e lançado pela Microsoft em agosto de 1981, foi atualizado pela última vez em 1994, quando o MS-DOS 6.22 foi lançado. O MS-DOS permitia ao usuário navegar, abrir e manipular arquivos em seu computador a partir de uma linha de comando, em vez de uma GUI como o Windows.

O prompt de comando do Windows é o intérprete de comando de todas as versões do Windows após 2000 (Windows 2000, XP, Server 2003, Vista, 7, 8, 8.1 e 10), e funciona como um programa dentro do sistema Windows. Nas versões anteriores (nos anos 90), como Windows 98, 95, 3.1, 3.11, etc, não havia um prompt de comando, pois, para iniciá-las, por não serem sistemas operacionais, o PC precisava primeiro operar o MS-DOS (command.com). O MS-DOS foi o software básico nos anos 90, e os comandos são os atuais no prompt de comando.

Muitas vezes, o prompt de comando é incorretamente referido como "prompt do DOS", ou como o próprio MS-DOS. O prompt de comando é um programa do Windows que emula muitas das habilidades de linha de comando disponíveis no MS-DOS, mas não é o MS-DOS.

O uso atual do prompt de comando do Windows concentra-se em usuários avançados, e com conhecimento de comandos. É reduzido a pouco mais do que utilitários para reparar problemas quando o computador é iniciado de forma anormal, ou quando há falhas na conexão com a Internet e é necessário alterar as configurações.

#### Diferença entre o CMD e o PowerShell

O Windows PowerShell desenvolvido em novembro de 2006, é um shell de linha de comando para gerenciamento de sistemas muito mais completo que o antigo cmd.exe.

O PowerShell permite a automação de rotinas de administração do Windows e aplicativos através de Scripts com acesso a todas as APIs .NET disponíveis no sistema e de outras APIs do Windows.

Graças ao CMD, é possível iniciar comandos básicos e preparar scripts relativamente simples. Mas quando se exige mais do sistema, ou quando é necessário fazer desenvolvimentos reais, ou ainda quando é necessário acessar determinadas áreas do

sistema em busca de informações é onde o CMD mostra suas deficiências, sendo impossível atingir um nível profundo do sistema operacional. Isso quer dizer que o prompt de comando pode ser usado para ações básicas e scripts simples.

Já o Powerhell é uma ferramenta mais moderna, que surgiu no Windows Vista e foi mantida atualizada; na verdade, ela tornou-se cada vez mais importante e não há dúvida de que no futuro substituirá completamente o prompt de comando.

A principal diferença entre os dois é que o PowerShell permite programação orientada a objetos, ou seja, programação complexa, além de trabalhar não apenas com o sistema operacional Windows no nível do usuário, mas também com programas da Microsoft como SQL Server, Exchange ou IIS, isto é, um escopo completo que permite automatizar tarefas também no nível do servidor.

Graficamente, o prompt de comando é uma janela com fundo preto e um cursor branco piscando. Normalmente, ele é iniciado, geralmente mostra os seguintes símbolos (ou uma rota que começa e termina com estes): C: >, seguido de user e o nome de usuário .

## Linux

No Linux O intérprete de comandos é a interface entre o usuário e o sistema operacional.

A interface Shell funciona como o intermediário entre o sistema operacional e o usuário graças às linhas de comando escritas por ele. A sua função é ler a linha de comando, interpretar seu significado, executar o comando e devolver o resultado pelas saídas. Na verdade, a interface Shell é um arquivo executável, encarregado de interpretar comandos, transmiti-los ao sistema e devolver resultados.

Existem vários tipos de Shell, sendo os mais comuns o sh (chamado Bourne shell), o bash (Bourne again shell), o csh (C Shell), o Tcsh (Tenex C shell), o ksh (Korn shell) e o zsh (Zero shell). Normalmente, seus nomes correspondem ao nome do executável.

Cada usuário tem um Shell padrão, que será ativado na abertura de um indicador do comando. O Shell padrão é definido no arquivo de configuração /etc/passwd no último campo da linha que corresponde ao usuário. Você pode alterar o Shell durante uma sessão. Para isso, basta rodar o arquivo executável correspondente, por exemplo: /bin/bash.

O Shell se inicia ao ler sua configuração global (em um arquivo do diretório /etc/). Em seguida, ele lê a configuração própria ao usuário (em arquivo oculto, cujo nome começa por um ponto e se situa no diretório básico do usuário: /home/user\_name/.configuration\_file). Depois, aparece o seguinte indicador de comando (prompt): máquina:/diretório/atual\$.

Por padrão, na maior parte dos Shells, o prompt é composto pelo nome da máquina, seguido de dois pontos (:), do diretório atual e de um caractere que indica o tipo de usuário conectado: \$ indica que se trata de um usuário normal e # do administrador, chamado root.

Durante a execução de um comando, é criado um processo que abrirá três fluxos:

- `stdin`, chamado entrada padrão, no qual o processo lê os dados de entrada. Por padrão, o `stdin` se refere ao teclado e é identificado pelo número 0;
- `stdout`, chamado saída padrão, no qual o processo escreve os dados de saída. Por padrão, o `stdout` se refere à tela e é identificado pelo número 1;
- `stderr`, chamado erro padrão, no qual o processo escreve as mensagens de erro. Por padrão, o `stderr` se refere à tela e é identificado pelo número 2.

Por padrão, quando se executa um programa, os dados são lidos a partir do teclado e o programa envia a sua saída e os seus erros para a tela. No entanto, também é possível ler os dados a partir de qualquer dispositivo de entrada, ou mesmo a partir de um arquivo, e enviar a saída para um dispositivo de visualização, arquivo etc.

## Mac

O Terminal (oficialmente chamado de Terminal.app) é, estritamente falando, um *emulador* e aceita a maioria dos comandos do UNIX (OS X é um sistema baseado no UNIX, diferente do Windows, que é baseado no NT). Diferente do OS X, que tem uma interface gráfica de usuário (GUI), o Terminal lança mão de uma interface textual e todos os comandos devem ser digitados -

Você pode usar o ambiente da linha de comando interativamente, digitando um comando e aguardando um resultado ou usar o shell para compor scripts para execução sem interação direta.

- No app Terminal do Mac, digite o nome de caminho completo do arquivo executável da ferramenta, seguido dos argumentos necessários, e pressione Retorno.

Se o comando estiver localizado em uma das pastas conhecidas do shell, você pode omitir as informações de caminho ao digitar o nome do comando. A lista de pastas conhecidas fica armazenada na variável de ambiente `PATH` do shell e inclui as pastas que contêm a maioria das ferramentas de linha de comando.

## Android

O Android Debug Bridge (`adb`) é uma ferramenta de linha de comando versátil que permite a comunicação com um dispositivo. O comando `adb` facilita uma variedade de ações do dispositivo, como instalar e depurar apps, e fornece acesso a um shell Unix que pode ser usado para executar diversos comandos em um dispositivo. Ele é um programa cliente-servidor com três componentes:

- **Um cliente**, que envia comandos. O cliente é executado no computador de desenvolvimento. Você pode emitir um comando `adb` para invocar o cliente de um terminal de linha de comando.
- **Um daemon (`adbd`)**, que executa comandos em um dispositivo. O daemon é executado como um processo em segundo plano em cada dispositivo.

- **Um servidor**, que gerencia a comunicação entre o cliente e o daemon. O servidor é executado como um processo em segundo plano na máquina de desenvolvimento.

adb está incluído no pacote Android SDK Platform-Tools. Você pode fazer o download desse pacote com o SDK Manager, que o instala em `android_sdk/platform-tools/`. Ou, se quiser o pacote autônomo Android SDK Platform-Tools,

Quando você inicia um cliente do adb, ele primeiro verifica se há um processo de servidor do adb em execução. Se não houver, ele inicia esse processo. Quando o servidor é iniciado, ele é vinculado à porta TCP 5037 local e escuta comandos enviados de clientes do adb. Todos os clientes do adb usam a porta 5037 para se comunicar com o servidor do adb.

Em seguida, o servidor configura as conexões para todos os dispositivos em execução. Ele localiza os emuladores por meio da varredura de portas de número ímpar no intervalo de 5555 a 5585, o intervalo usado pelos primeiros 16 emuladores. Onde o servidor encontrar um daemon do adb (adbd), ele irá configurar uma conexão com a porta em questão. Observe que cada emulador usa um par de portas sequenciais: uma porta com numeração par para conexões de console e uma porta ímpar para conexões adb.

Depois que o servidor estiver configurado conexões com todos os dispositivos, você poderá usar os comandos adb para acessá-los. Como o servidor gerencia conexões com dispositivos e manipula comandos de vários clientes adb, você pode controlar qualquer dispositivo de qualquer cliente (ou de um script).

Para usar o adb com um dispositivo conectado via USB, você precisa ativar a opção Depuração USB nas configurações do sistema do dispositivo, em Opções do desenvolvedor.

No Android 4.2 e versões posteriores, a tela "Opções do desenvolvedor" normalmente fica oculta por padrão. Para exibi-la, acesse Config. > Sobre o dispositivo e toque em Número da versão sete vezes. Retorne à tela anterior para encontrar as Opções do desenvolvedor na parte inferior.

Em alguns dispositivos, a tela "Opções do desenvolvedor" pode ter uma localização ou um nome diferente.

Agora você pode conectar seu dispositivo a uma porta USB. Verifique se o dispositivo está conectado executando `adb devices` no diretório `android_sdk/platform-tools/`. Se eles estiverem conectados, o nome do dispositivo estará listado como "device"

Após configurado adp ja podera ser feito qualquer atividade por linha de comando ,

instalar app, Copiar arquivos de um dispositivo, Configurar encaminhamento de portas, Emitir comandos do Shell

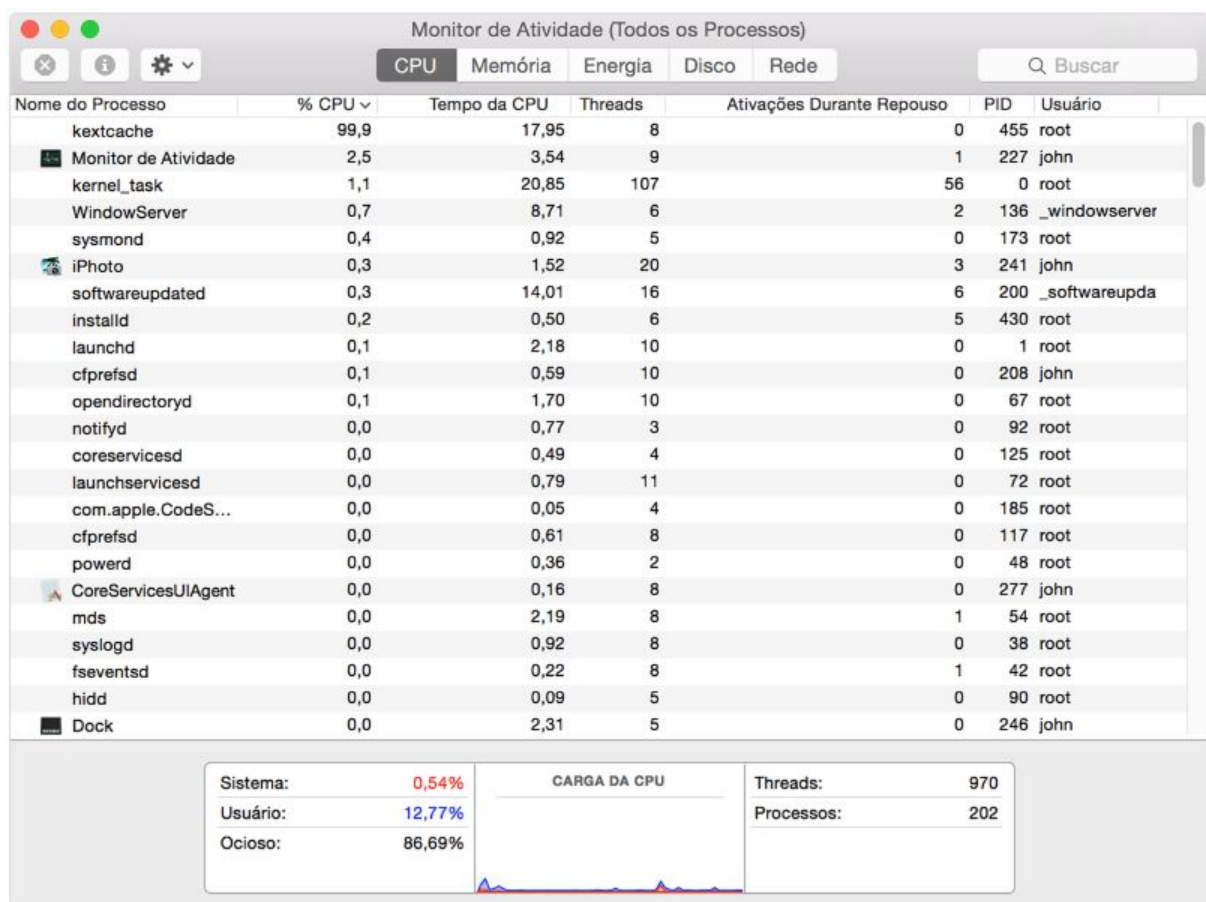
Você pode usar o comando shell para emitir comandos do dispositivo por meio do adb ou para iniciar um shell interativo. Para emitir um único comando, use o comando shell da seguinte maneira: `adb [-d | -e | -s serial_number] shell`

O Android fornece a maioria das ferramentas de linha de comando Unix comuns. Para ver uma lista de ferramentas disponíveis, use o seguinte comando: `adb shell ls /system/bin`

## °Acessórios e ferramentas para visualização do desempenho (CPU, Memória e Disco)

### MacOS -

#### CPU



Mais informações estão disponíveis na parte inferior do painel CPU:

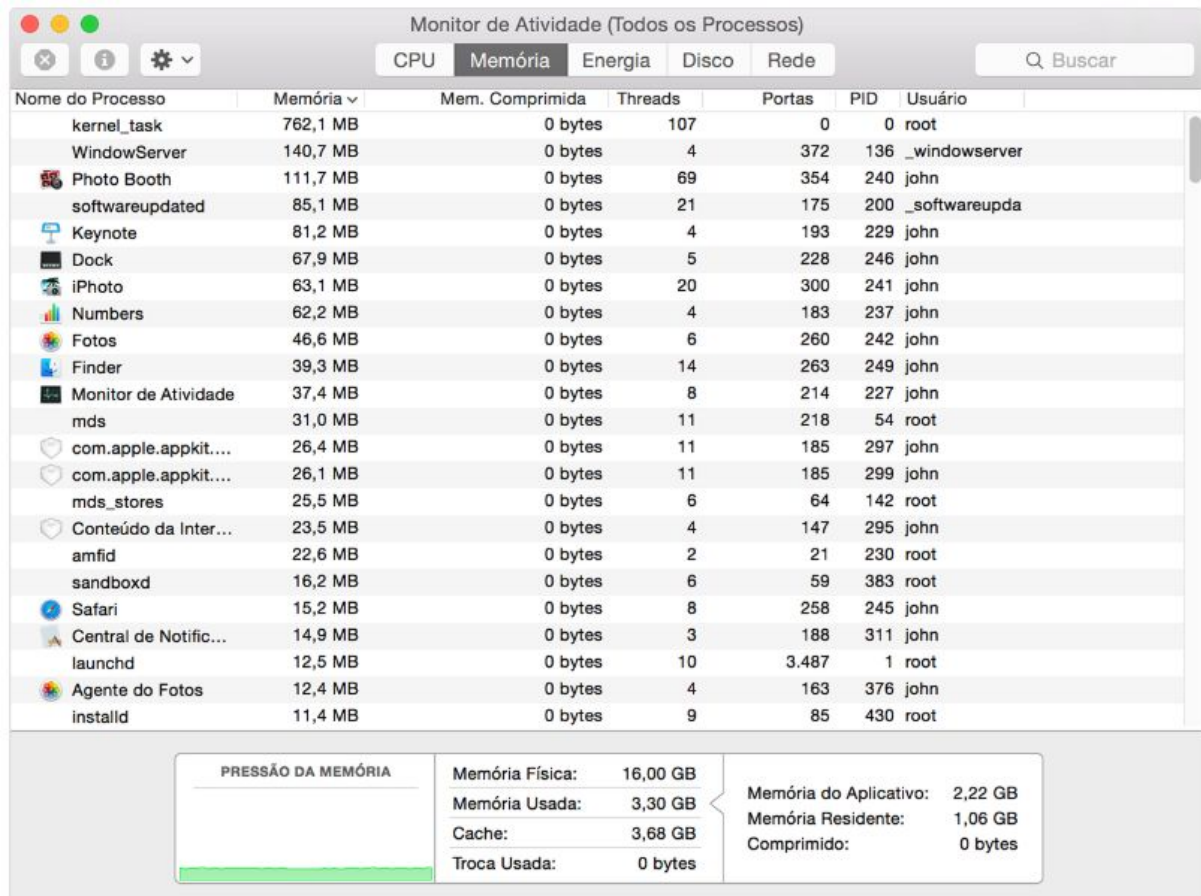
- Sistema: a porcentagem da capacidade da CPU usada atualmente pelos processos do sistema, que são processos que pertencem ao macOS.
- Usuário: a porcentagem da capacidade da CPU usada atualmente por apps abertos ou pelos processos abertos por esses apps.
- Ocioso: a porcentagem da capacidade da CPU que não é utilizada.
- Carga da CPU: a porcentagem da capacidade da CPU usada atualmente por todos os processos de Sistema e Usuário. O gráfico move-se da direita para a esquerda e é atualizado nos intervalos definidos em Visualizar > Frequência de Atualização. A



cor azul mostra a porcentagem da capacidade total da CPU usada atualmente pelos processos do usuário. A cor vermelha mostra a porcentagem da capacidade total da CPU usada atualmente pelos processos do sistema.

- Threads: o número total de threads usados por todos os processos combinados.
- Processos: o número total de processos executados atualmente.

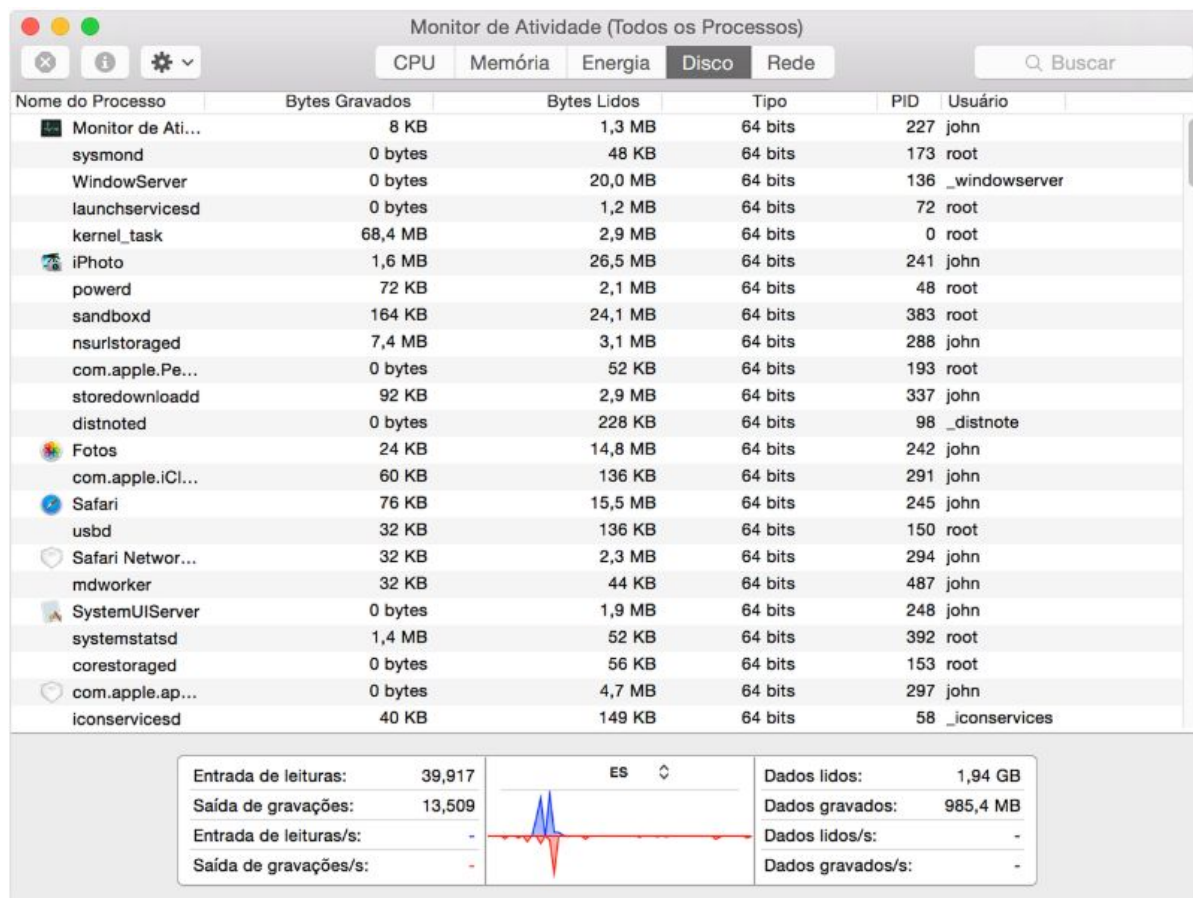
## Memória -



- **Pressão da Memória:** este gráfico ajuda a ilustrar a disponibilidade dos recursos da memória. O gráfico move-se da direita para a esquerda e é atualizado nos intervalos definidos em Visualizar > Frequência de Atualização.
- **Memória Física:** a quantidade de RAM instalada no Mac.
- **Memória Usada:** a quantidade total de memória usada atualmente por todos os apps e processos do macOS.
- **Troca Usada:** o espaço usado na unidade de inicialização pelo gerenciamento de memória do macOS. É normal encontrar alguma atividade aqui. Enquanto a pressão da memória não estiver no estado vermelho, o macOS terá recursos de memória disponíveis.
- **Arquivos em Cache:** memória usada recentemente por apps e que agora está disponível para utilização por outros apps.

## Disco

O painel Disco mostra o volume de dados que cada processo leu e gravou no disco. Ele também mostra as entradas de leituras e as saídas de gravações (ES), que é o número de vezes que o Mac acessa o disco para ler e gravar dados.



## Linux

Certamente o comando **top** é um dos mais utilizados, já que fornece uma visão contínua da atividade do processador em tempo real por uma lista dos processos mais utilizados pela CPU, como também fornece uma interface para manipulá-los. Também mostra um resumo informativo da situação do sistema, assim como uma lista de processos ou “threads” utilizados pelo Kernel do Linux. É possível ordenar as tarefas por uso de CPU, uso de memória e tempo de execução.

```

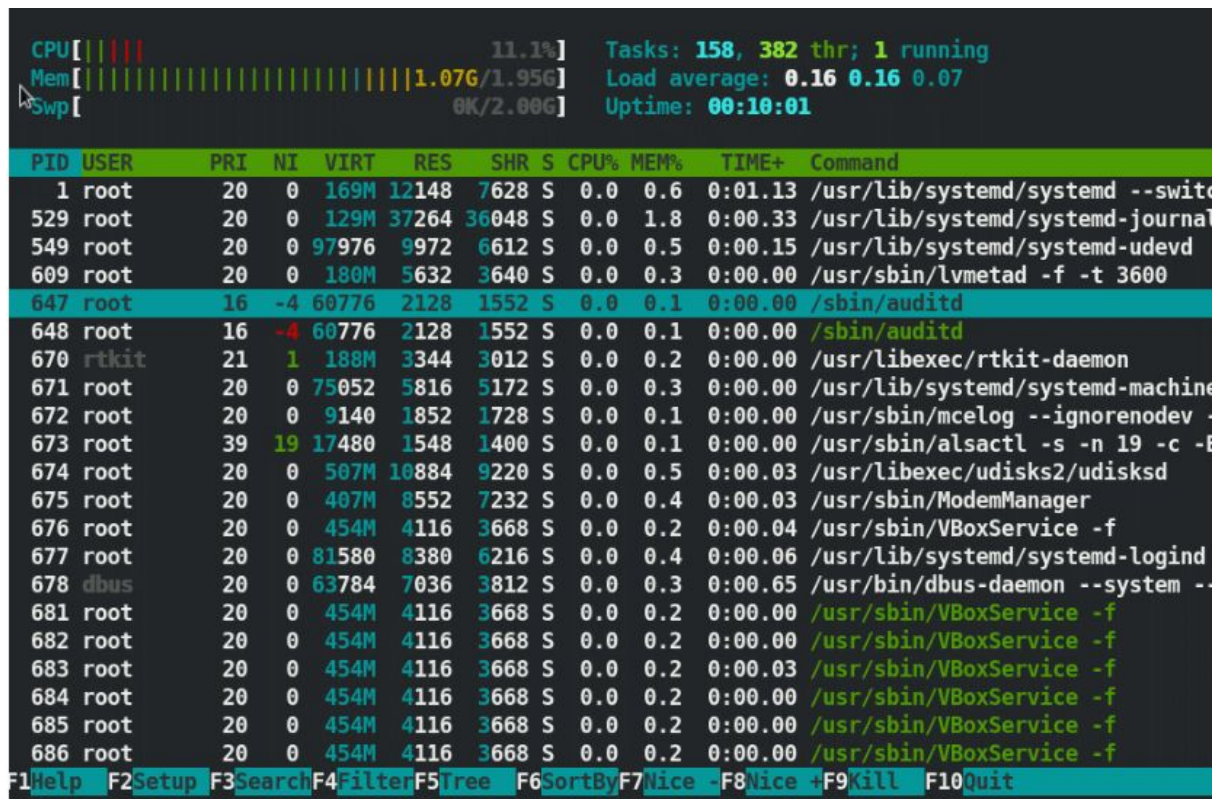
top - 15:55:41 up 13 min, 1 user, load average: 0,49, 0,18, 0,08
Tasks: 209 total, 1 running, 173 sleeping, 0 stopped, 0 zombie
%Cpu(s): 25,2 us, 1,3 sy, 0,0 ni, 70,8 id, 0,0 wa, 1,3 hi, 1,3 si, 0,0 st
KiB Mem : 2040308 total, 416660 free, 1087696 used, 535952 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 787528 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	173504	12148	7628	S	0,0	0,6	0:01.13	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
8	root	20	0	0	0	0	S	0,0	0,0	0:00.06	ksoftirqd/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
12	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/0
13	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
16	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_kthre
17	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	oom_reaper
20	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kcompactd0
21	root	25	5	0	0	0	S	0,0	0,0	0:00.00	ksmd
22	root	39	19	0	0	0	S	0,0	0,0	0:00.00	khugepaged
30	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdogd
33	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kswapd0
89	root	20	0	0	0	0	S	0,0	0,0	0:00.00	scsi_eh_0
91	root	20	0	0	0	0	S	0,0	0,0	0:00.00	scsi_eh_1
93	root	20	0	0	0	0	S	0,0	0,0	0:00.00	scsi_eh_2
444	root	20	0	0	0	0	S	0,0	0,0	0:00.00	jbd2/dm-0-8
529	root	20	0	141188	38048	36824	S	0,0	1,9	0:00.40	systemd-journal
549	root	20	0	97976	9972	6612	S	0,0	0,5	0:00.16	systemd-udev
609	root	20	0	185136	5632	3640	S	0,0	0,3	0:00.00	lvmetad
623	root	20	0	0	0	0	S	0,0	0,0	0:00.00	jbd2/sda1-8
647	root	16	-4	60776	2128	1552	S	0,0	0,1	0:00.00	auditd
670	rtkit	21	1	192976	3344	3012	S	0,0	0,2	0:00.00	rtkit-daemon

Em contrapartida, o comando **htop** possui várias funcionalidades a mais em comparação ao comando **top** e também o considero mais “simpático”. Sobretudo possui uma interface intuitiva, com gráfico de uso em tempo real das CPU’s, memória e SWAP, além de várias opções como ordenar as informações ou até matar processos. Este comando não vem instalado por padrão na maioria das distribuições.





O comando **netstat** (**net**working **stat**us) mostra informações como conexões de rede, tabelas de roteamento e estatísticas de interfaces de rede. O comando **netstat** puramente, sem nenhum parâmetro, mostra todas as conexões (não somente rede, mas também locais),

```
[root@localhost ~]# netstat
Conexões Internet Ativas (sem os servidores)
Proto Recv-Q Send-Q Endereço Local      Endereço Remoto      Estado
Domain sockets UNIX ativos (sem os servidores)
Proto RefCnt Flags   Type       State        I-Node  Caminho
unix  8      [ ]     DGRAM      12031        /run/systemd/journal/socket
unix  28     [ ]     DGRAM      12050        /run/systemd/journal/dev-log
unix  2      [ ]     DGRAM      24131        /run/user/42/systemd/notify
unix  2      [ ]     DGRAM      33222        /run/user/1000/systemd/notify
unix  2      [ ]     DGRAM      20422        /var/run/chrony/chronyd.sock
unix  3      [ ]     DGRAM      12012        /run/systemd/notify
unix  3      [ ]     STREAM     CONECTADO    38930
unix  2      [ ]     DGRAM      36883
unix  3      [ ]     STREAM     CONECTADO    35751 /run/systemd/journal/stdout
unix  3      [ ]     STREAM     CONECTADO    24867
unix  3      [ ]     STREAM     CONECTADO    22955
unix  3      [ ]     STREAM     CONECTADO    39958 /run/systemd/journal/stdout
unix  3      [ ]     STREAM     CONECTADO    36475 /run/systemd/journal/stdout
unix  2      [ ]     DGRAM      20988
unix  3      [ ]     STREAM     CONECTADO    20187
unix  3      [ ]     STREAM     CONECTADO    40591
unix  3      [ ]     STREAM     CONECTADO    35168 /run/systemd/journal/stdout
unix  3      [ ]     STREAM     CONECTADO    38986
unix  3      [ ]     STREAM     CONECTADO    37183
unix  3      [ ]     STREAM     CONECTADO    35538 /run/systemd/journal/stdout
unix  3      [ ]     STREAM     CONECTADO    24901 /run/systemd/journal/stdout
unix  3      [ ]     DGRAM      24133
unix  3      [ ]     STREAM     CONECTADO    39959 /run/systemd/journal/stdout
unix  3      [ ]     STREAM     CONECTADO    36448
unix  3      [ ]     STREAM     CONECTADO    20560
unix  2      [ ]     DGRAM      19859
```

O comando **vmstat** (virtual **m**emory **s**tatus) gera um relatório com informações estatísticas de memória. Além disso, possui informações sobre processos, memória, paginação, gravação (block I/O), discos e atividade da CPU. O comando executado sem parâmetros nos traz as informações médias desde a última reinicialização do sistema mas é possível monitorar o **vmstat** por intervalos, passando este intervalo como parâmetro.

```
[root@localhost ~]# vmstat 1 10
procs -----memory----- --swap-- -----io----- -system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
3 0 524 212176 45544 692544 0 0 145 24 1990 263 2 2 95 0 0
1 0 524 212076 45544 692544 0 0 0 0 2020 579 23 3 74 0 0
0 0 524 212044 45544 692544 0 0 0 0 1765 522 24 4 72 0 0
0 0 524 212044 45544 692544 0 0 0 0 1945 208 3 2 95 0 0
0 0 524 211888 45544 692548 0 0 0 0 1972 1271 27 8 65 0 0
0 0 524 211888 45544 692548 0 0 0 0 1875 237 4 2 94 0 0
1 0 524 211952 45552 692548 0 0 0 16 1866 561 16 3 81 0 0
0 0 524 211920 45552 692548 0 0 0 0 1695 749 30 5 65 0 0
0 0 524 211888 45552 692548 0 0 0 0 1921 216 4 2 94 0 0
0 0 524 211888 45552 692548 0 0 0 0 2235 257 2 2 96 0 0
```

O comando **ps** (**p**rocesses) mostra o status dos processos atuais. Entretanto este comando (sem nenhum parâmetro) mostra os processos que estão rodando com o seu ID de usuário e estão conectados a um terminal (TTY),

```
[root@localhost ~]# ps
  PID TTY          TIME CMD
 2659 pts/0    00:00:00 su
 2668 pts/0    00:00:00 bash
 3868 pts/0    00:00:00 ps
[root@localhost ~]#
```

Com o **ps -aux** é possível visualizar quais processos estão utilizando mais CPU e memória RAM.

```
[root@localhost ~]# ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.5	239172	12204	?	Ss	15:42	0:01	/usr/lib/systemd/systemd --switched-root
root	2	0.0	0.0	0	0	?	S	15:42	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	15:42	0:00	[rcu_gp]
root	5	0.0	0.0	0	0	?	I<	15:42	0:00	[kworker/0:0H]
root	7	0.0	0.0	0	0	?	I<	15:42	0:00	[mm_percpu_wq]
root	8	0.0	0.0	0	0	?	S	15:42	0:00	[ksoftirqd/0]
root	9	0.0	0.0	0	0	?	I	15:42	0:00	[rcu_sched]
root	10	0.0	0.0	0	0	?	I	15:42	0:00	[rcu_bh]
root	11	0.0	0.0	0	0	?	S	15:42	0:00	[migration/0]
root	12	0.0	0.0	0	0	?	S	15:42	0:00	[watchdog/0]
root	13	0.0	0.0	0	0	?	S	15:42	0:00	[cpuhp/0]
root	14	0.0	0.0	0	0	?	S	15:42	0:00	[kdevtmpfs]
root	15	0.0	0.0	0	0	?	I<	15:42	0:00	[netns]
root	16	0.0	0.0	0	0	?	S	15:42	0:00	[rcu_tasks_kthre]
root	17	0.0	0.0	0	0	?	S	15:42	0:00	[kauditd]
root	18	0.0	0.0	0	0	?	S	15:42	0:00	[oom_reaper]
root	19	0.0	0.0	0	0	?	I<	15:42	0:00	[writeback]
root	20	0.0	0.0	0	0	?	S	15:42	0:00	[kcompactd0]
root	21	0.0	0.0	0	0	?	SN	15:42	0:00	[ksmd]
root	22	0.0	0.0	0	0	?	SN	15:42	0:00	[khugepaged]
root	23	0.0	0.0	0	0	?	I<	15:42	0:00	[crypto]
root	24	0.0	0.0	0	0	?	I<	15:42	0:00	[kintegrityd]
root	25	0.0	0.0	0	0	?	I<	15:42	0:00	[kblockd]
root	26	0.0	0.0	0	0	?	I<	15:42	0:00	[ata_sff]
root	27	0.0	0.0	0	0	?	I<	15:42	0:00	[md]
root	28	0.0	0.0	0	0	?	I<	15:42	0:00	[edac-poller]
root	29	0.0	0.0	0	0	?	I<	15:42	0:00	[devfreq_wq]
root	30	0.0	0.0	0	0	?	S	15:42	0:00	[watchdogd]
root	31	0.0	0.0	0	0	?	I	15:42	0:00	[kworker/0:1]

## Windows

Podemos ter acesso a informações de desempenho do uso da Cpu Disco e Memória através do Gerenciador de Tarefas

O Gerenciador de Tarefas é um recurso disponível no Windows que mostra quais programas estão sendo executados no seu computador. Ele permite que um usuário visualize cada uma das tarefas atualmente em execução no computador, os processos e o desempenho geral do PC.


Como uma ferramenta de monitoramento, o Gerenciador de Tarefas exibe dados básicos de desempenho de CPU, memória, etc. Os profissionais de TI geralmente usam o Gerenciador de Tarefas para identificar rapidamente o que pode estar afetando o desempenho do sistema.


Uma das coisas mais comuns feitas no Gerenciador de Tarefas é usar “Finalizar Tarefa” para impedir que um programa seja executado. Se um programa não estiver mais respondendo, você poderá optar por Finalizar tarefa no Gerenciador de tarefas para fechar o programa sem precisar reiniciar o computador.


Com o Gerenciador de Tarefas do Windows é possível verificar os programas abertos como também os programas que estão sendo executado em segundo plano.


O Gerenciador de Tarefas pode ser usado para encerrar programas em execução, bem como para ver o uso de recursos de hardware do computador, quais programas e serviços estão sendo iniciados e muito mais.





**CPU**  
12% 2,31 GHz


**Memória**  
4,3/7,9 GB (54%)

**Disco 0 (E:)**  
HDD  
0%

**Disco 1 (C:)**  
SSD  
0%

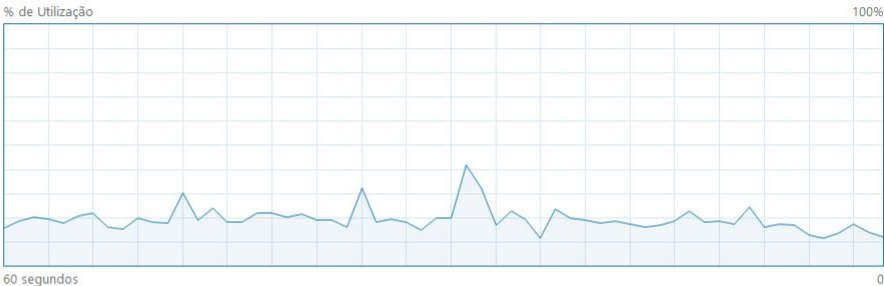
**Wi-Fi**  
Wi-Fi  
S: 0 R: 0 Kbps

**GPU 0**  
Intel(R) HD Graphics  
1%

**GPU 1**  
NVIDIA GeForce 940  
0%


## CPU


Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz





Utilização	Velocidade	Velocidade base:	2,90 GHz
12%	2,31 GHz	Sockets:	1
		Núcleos:	2
Processos	Threads	Identificadores	Processadores lógicos:
183	1814	72273	4
			Virtualização: Habilitado
Tempo de atividade		Cache L1:	128 KB
0:22:06:27		Cache L2:	512 KB
		Cache L3:	4,0 MB


Menos detalhes | Abrir Monitor de Recursos


**CPU**  
12% 2,32 GHz


**Memória**  
4,3/7,9 GB (54%)

**Disco 0 (E:)**  
HDD  
0%

**Disco 1 (C:)**  
SSD  
1%

**Wi-Fi**  
Wi-Fi  
S: 0 R: 0 Kbps

**GPU 0**  
Intel(R) HD Graphics  
1%

**GPU 1**  
NVIDIA GeForce 940  
0%

## Memória

8,0 GB



Em uso (Compactada)	Disponível	Velocidade:	2133 MHz
4,2 GB (99,8 MB)	3,6 GB	Slots usados:	1 de 2
Confirmado	Em cache	Fator forma:	SODIMM
5,2/13,9 GB	1,9 GB	Reservada para hardware:	102 MB
Reserva de memória paginável	Reserva de memória não paginável		
269 MB	220 MB		

Menos detalhes | Abrir Monitor de Recursos



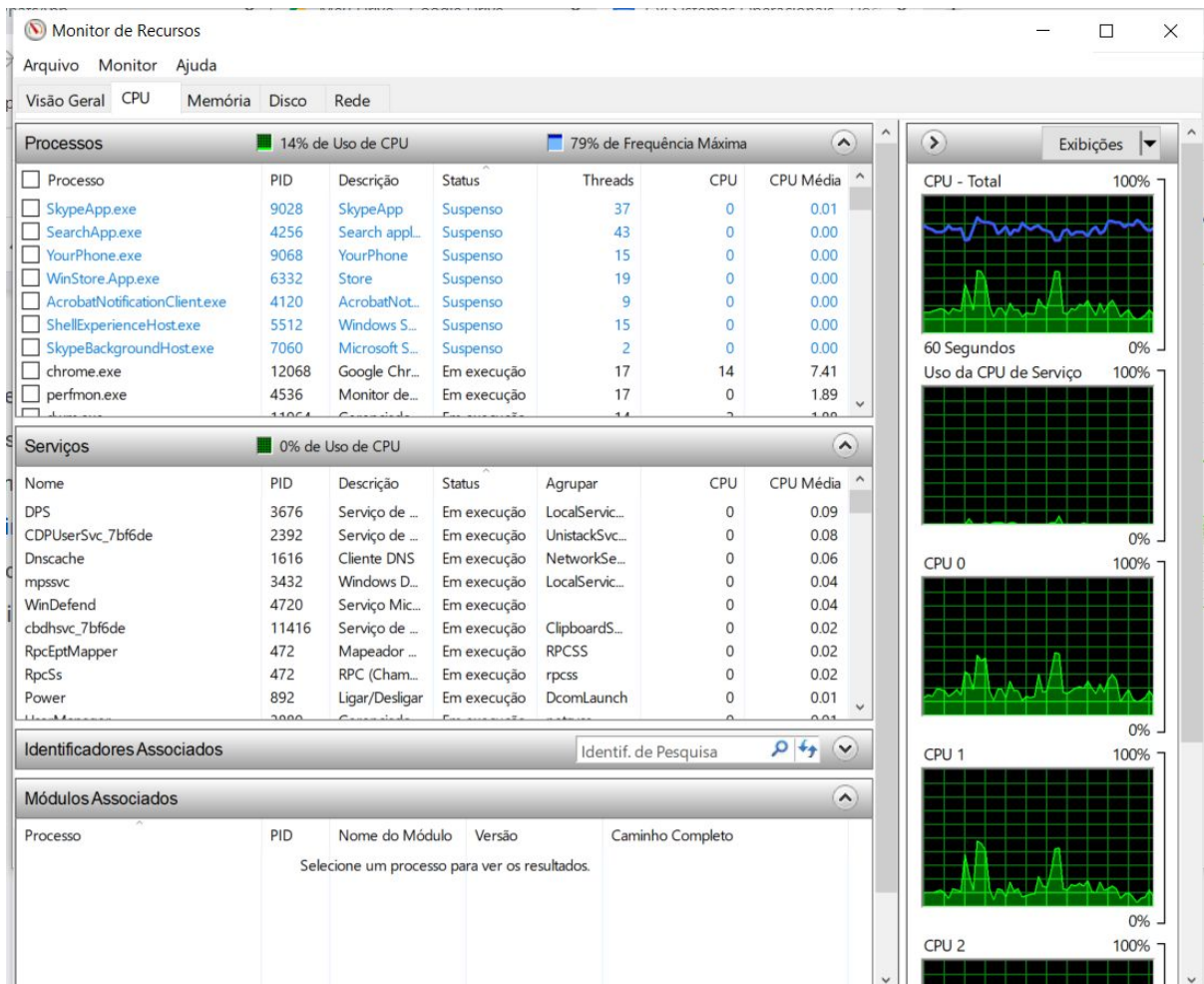
Caso precise de informações mais detalhadas podemos utilizar o Monitor de Recursos do Windows

Monitor de recursos do Windows é um recurso do sistema que foi incluído no Windows Vista e versões posteriores do Windows que permite aos usuários verificar a alocação de recursos em um computador, sendo adequado para usuários avançados que precisam de mais detalhes sobre a execução de processos e serviços.

O Monitor de Recursos exibe dados em tempo real do PC e permite que os administradores e outros usuários verifiquem como os recursos do sistema estão sendo usados. É um utilitário gratuito que é muito útil para monitorar problemas de desempenho do seu computador.

O *Monitor de Recursos* Também pode ser iniciado digitando “resmon” na caixa de pesquisa do Windows. É possível exibir várias informações, como o uso da CPU, disco e memória. O gráfico que é exibido torna mais fácil para aqueles que gerenciam um sistema.





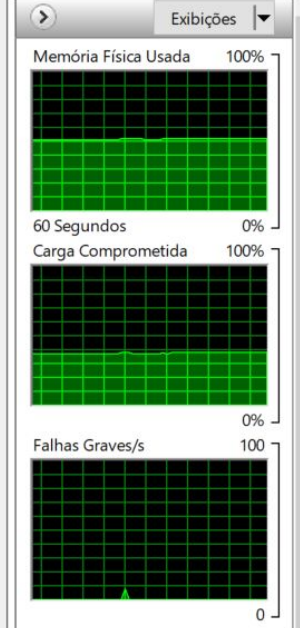
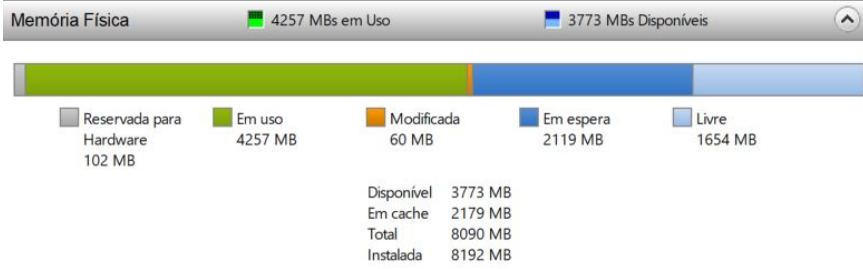
Monitor de Recursos

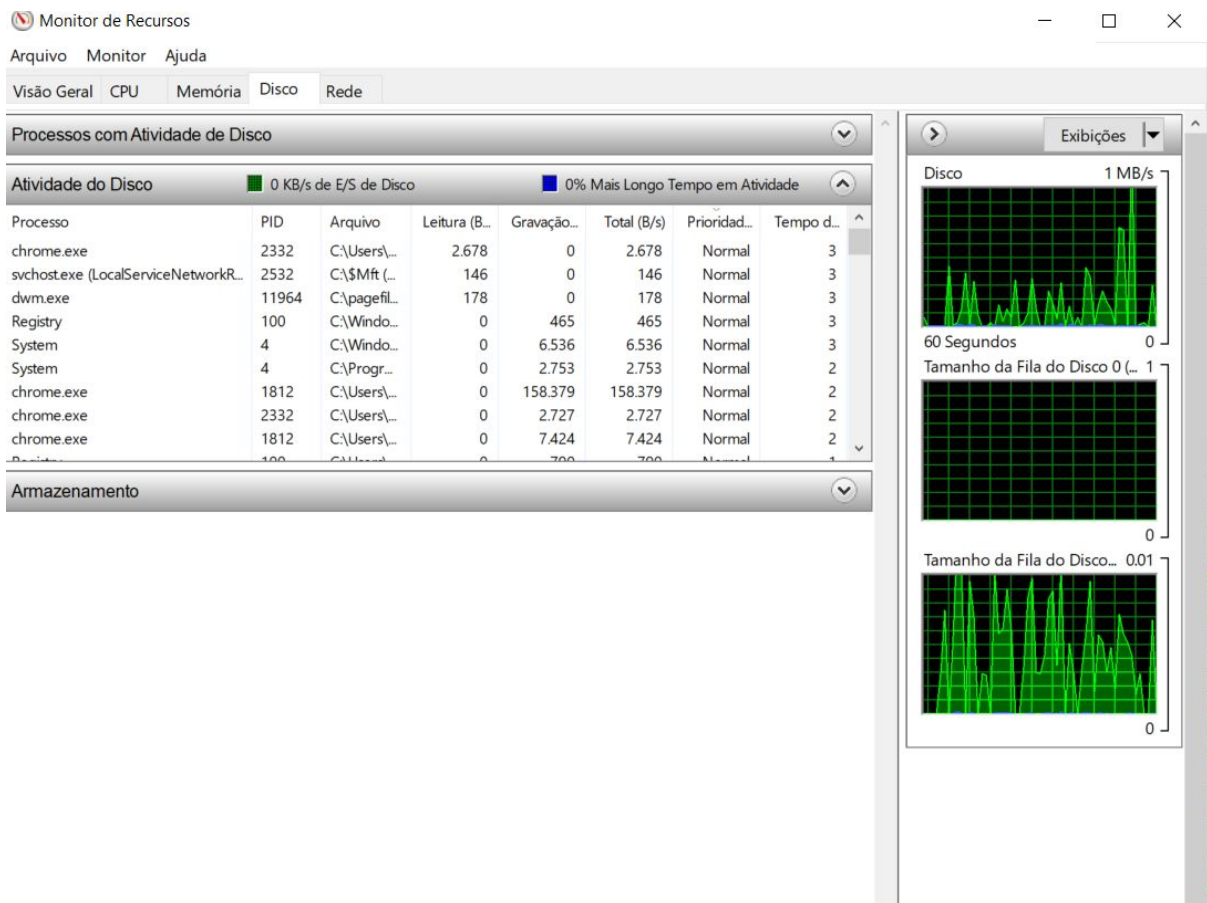
Arquivo Monitor Ajuda

Visão Geral CPU Memória Disco Rede

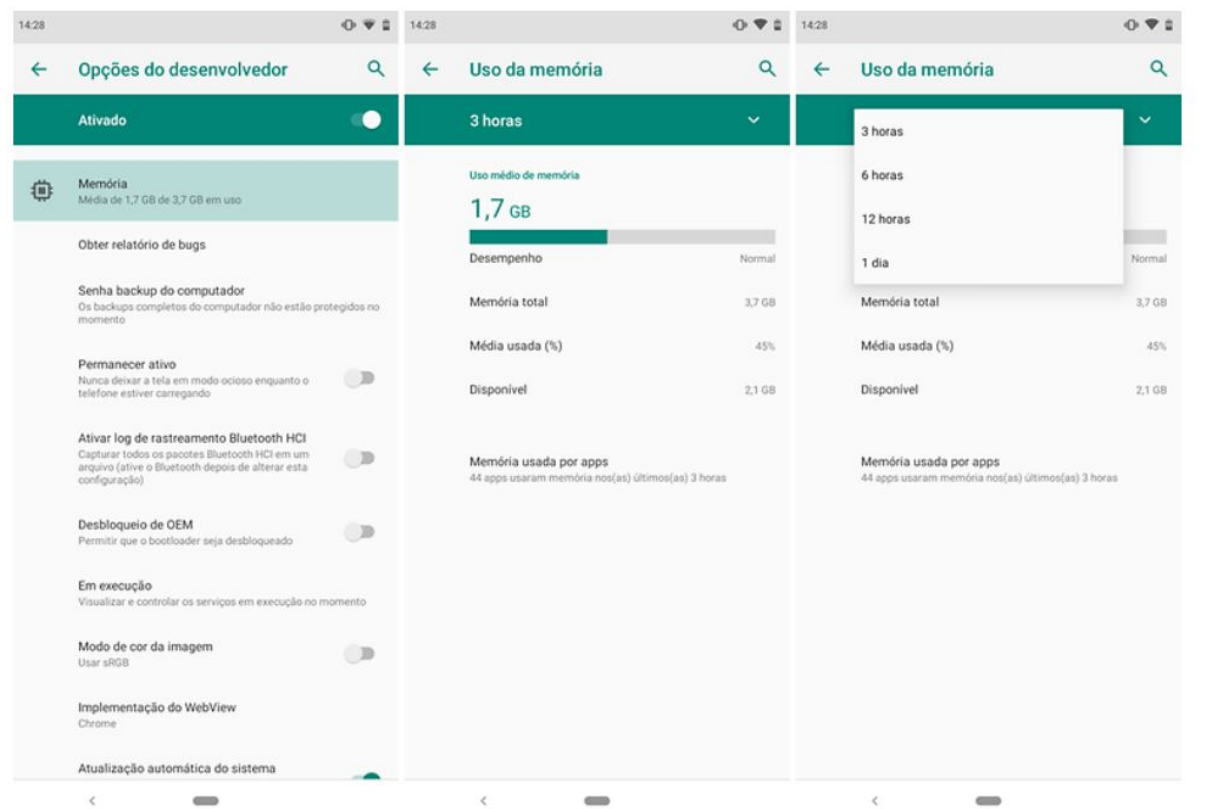
Processos 52% de Memória Física Usada

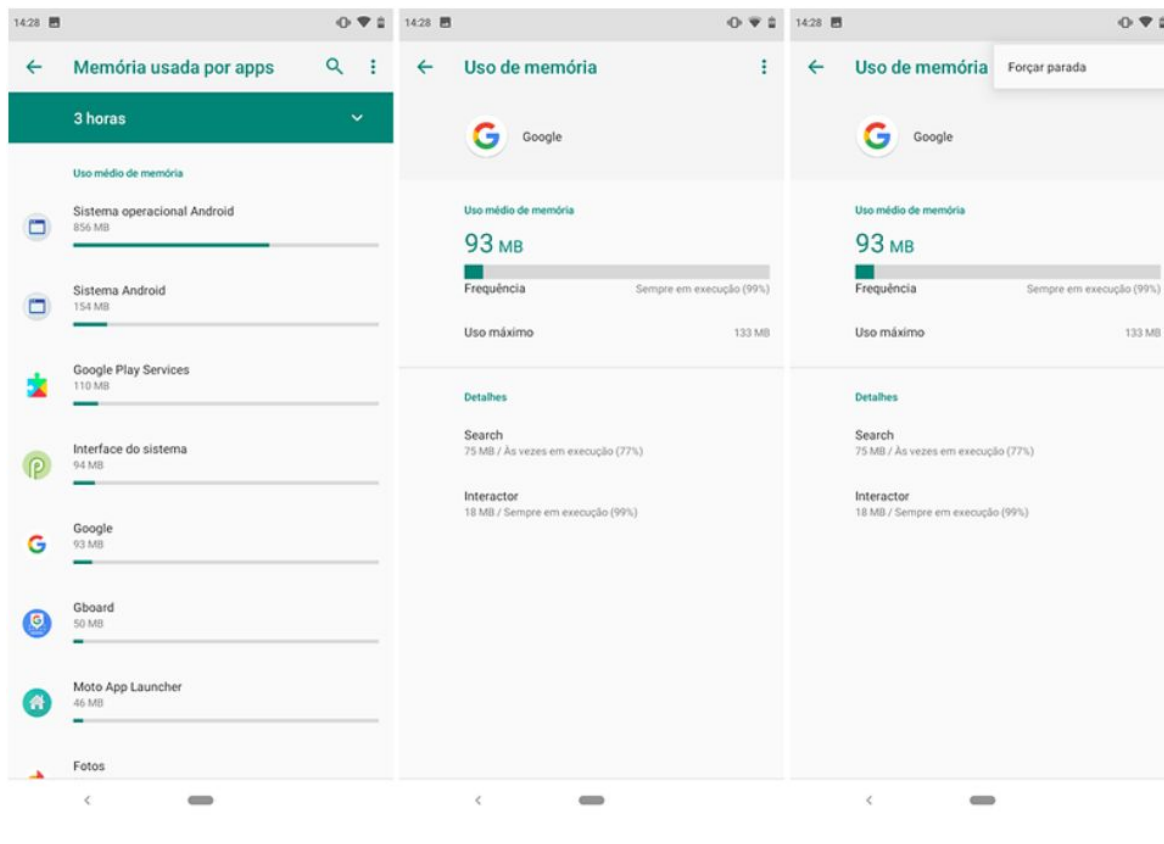
Processo	PID	Falhas Grav...	Confirmar (...)	Conjunto d...	Compartilhã...	Privado (KB)
chrome.exe	10124	0	432.440	424.828	56.036	368.792
chrome.exe	1812	0	183.380	240.312	78.396	161.916
Memory Compression	2920	0	528	161.420	0	161.420
chrome.exe	10200	0	171.612	188.584	55.024	133.560
SkypeApp.exe	9028	0	195.016	176.328	42.956	133.372
MsMpEng.exe	4720	0	362.712	184.792	56.616	128.176
chrome.exe	12068	0	132.444	168.632	58.008	110.624
chrome.exe	8908	0	129.452	160.584	61.912	98.672
chrome.exe	9708	0	175.080	137.176	56.004	81.172





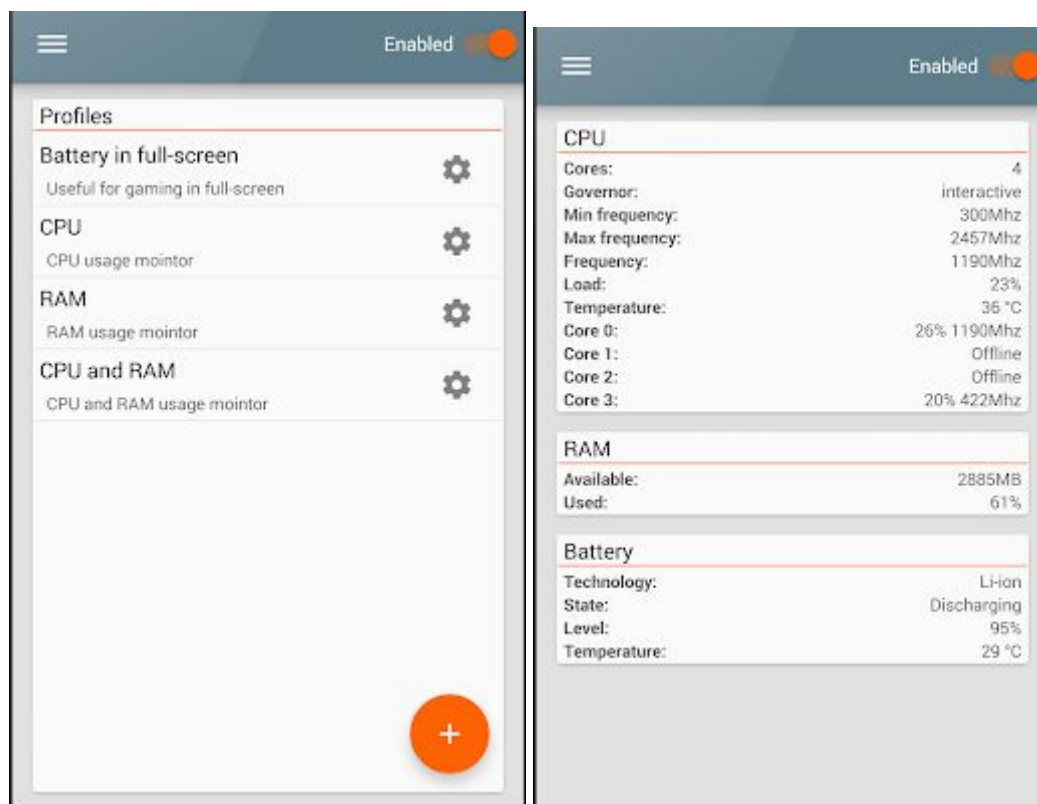
## Android





Aqui temos as estatísticas detalhadas sobre o uso da memória nas últimas horas. Dependendo do smartphone e da interface do usuário, este painel estará escondido em algum sub-menu. Para ver a quantidade de RAM usada, vá até às configurações do seu smartphones. Procure pelo menu de aplicativos e selecione a guia "ativos". Desde o Android Marshmallow e até o 10, a seção que mostra a quantidade de RAM em uso está no menu "Memória", na lista de configurações principais. Já no Android Pie esse menu fica escondido nas Opções do Desenvolvedor.

Podemos fazer uso também de aplicativos um exemplo seria o TinyCore aonde conseguimos verificar o desempenho da CPU e a Memória.



Desta forma conseguimos ter uma visualização do uso da memória e da CPU, esse é um dos vários aplicativos que a play store pode ter.

## °Formas de eliminar processos em execução.

Windows - Clique com o botão direito do mouse em uma área vazia da Barra de Tarefas e selecione Iniciar Gerenciador de Tarefas no menu.

Na guia Processos, clique em Mostrar processos de todos os usuários.

Selecione o processo que deseja eliminar e clique no botão Finalizar processo.

Ou através da linha de comando no CMD: 'taskkill /f /pid xxxxx' (xxxxx é o PID referente a tarefa dentro do gerenciador de tarefas do Windows)

Linux - Uma das maneiras de eliminar é através da linha de comando *kill -9 PID*. Assim você força o processo a ser interrompido a qualquer custo.

Se quiser matar mais de um processo ao mesmo tempo, basta listar os PIDs separando-os com um espaço, logo depois do comando kill. Exemplo: *kill 3657 6785 3456*.

MacOS - Através do Activity Monitor, lista os processos ativos, apenas selecionando o processo e Clicar no botão "Encerrar processo" no canto superior esquerdo da janela.

também por linha de comando no Terminal, o comando 'top' faz a listagens dos processos, escolha o processo , anote o PID, comando para finalizar o processo seria "kill 3703".

Android - para finalizar um aplicativo ou processo a melhor forma seria ir em Configurações->Aplicativos escolher o aplicativo. e ir em Force Stop(Forçar parada).

## °Sistemas de Arquivos utilizados

SISTEMA OPERACIONAL	SISTEMA DE ARQUIVOS SUPORTADOS
Linux	EXT3, EXT4, XFS, JFS
MacOS	HFS
Windows	FAT, HPFS, NTFS
Android	EXT4 e F2FS.

- **EXT3** (*third extended filesystem*) – foi adotado como padrão Linux a partir de 2001. Introduziu o registro (*journal*) que melhora a confiabilidade e permite recuperar o sistema em caso de desligamento não programado. EXT3 suporta 16TB (1 terabyte corresponde a 240 bytes) de tamanho máximo no sistema de arquivos, e 2TB de tamanho máximo de um arquivo. Um diretório pode ter, no máximo, 32.000 subdiretórios.
- **EXT4** (*fourth extended filesystem*) – passou a ser o padrão Linux a partir de 2008. EXT4 suporta 1EB (1 exabyte corresponde a 260 bytes) de tamanho máximo de sistema de arquivos e 16TB de tamanho máximo de arquivos. É possível ter um número ilimitado de subdiretórios
- **XFS** (*Extended Filesystem*) – usado como padrão por algumas distribuição Linux desde 2014. XFS é um sistema de arquivos desenvolvido em 64 bits, compatível com sistemas de 32 bits. Ele suporta até 16 EB de tamanho total do sistema de arquivos e até 8 EB de tamanho máximo para um arquivo individual. É considerado um sistema de arquivos de alto desempenho.
- **JFS** (*Journaled File System*) – é um sistema de arquivos de 64 bits com *journaling* desenvolvido pela **IBM**.
- **HFS** (*Hierarchical File System*) – é um sistema de arquivos proprietário da **Apple**.
- **FAT** (*File Allocation Table*) – é um sistema desenvolvido para o **MS-DOS** e usado em versões do **Microsoft Windows** até o **Windows 95**. É suportado praticamente por todos os sistemas operacionais existentes. Existem 3 versões do sistema: FAT (12 bits, usado pelos disquetes), FAT16 (para OS 16 bits ou 32 bits) e FAT32 (só para SO a 32 bits)
- **NTFS** (*New Technology File System*) é o sistema de arquivos padrão do sistema operacional **Microsoft Windows**. São algumas características deste tipo de sistema: aceita volumes de até 2 TB; o tamanho do arquivo é limitado apenas pelo tamanho do volume; é um sistema de arquivos muito mais seguro que o FAT; NTFS podem se recuperar de um erro mais facilmente.
- **HPFS** – é o sistema de arquivos utilizado pelo **OS/2** da **IBM**, com recursos que se aproximam muito dos permitidos pelo NTFS.
- **F2FS** -- A sigla significa em inglês “Flash-Friendly File System”. Trata-se do sistema de arquivos criado por Kim Jaegeuk na Samsung para o núcleo Linux (em que se baseia o Android). Ele foi criado de forma específica para ter em conta os dispositivos de armazenamento Flash, ou seja, a forma de memória da

grande maioria dos smartphones. Os nossos celulares trazem uma memória de estado sólido, ou SSD, além de cartões SD, em ambos os casos com tecnologia flash.