Gerenciando processos no Windows

A ferramenta mais utilizada no Windows para gerenciar processos é o “Gerenciador de Tarefas”, que além de processos, monitora e gerencia o desempenho de vários recursos do computador, como memória, armazenamento, processadores, assim como outros recursos de.

É importante ter conhecimento destas informações para entender o funcionamento da do computador e do sistema operacional de forma a ter o controle sobre o sistema computacional e saber, por exemplo, qual processo usa mais CPU, mais memória ou que está sendo executado desnecessariamente ou que está tornando o sistema operacional mais lento.

Acessando o gerenciador de tarefas

Para abrir o “Gerenciador de Tarefas” pode-se utilizar/pressionar as teclas Ctrl+Shift+Esc ao mesmo tempo. Outra opção disponível em todas as versões do Windows é clicar no menu iniciar (ícone da janela) e digitar “Gerenciador de Tarefas”, assim a opção da Figura 1 irá ser exibida bastando então clicar no ícone.

No Windows 10 também é possível acessar clicando com o botão direito na barra de tarefas e selecionando “Gerenciador de Tarefas”. No Windows 11 é possível, além das opções citadas acima, clicar com o botão direito do “mouse” no ícone do menu iniciar/janela e selecionar “Gerenciador de Tarefas”.

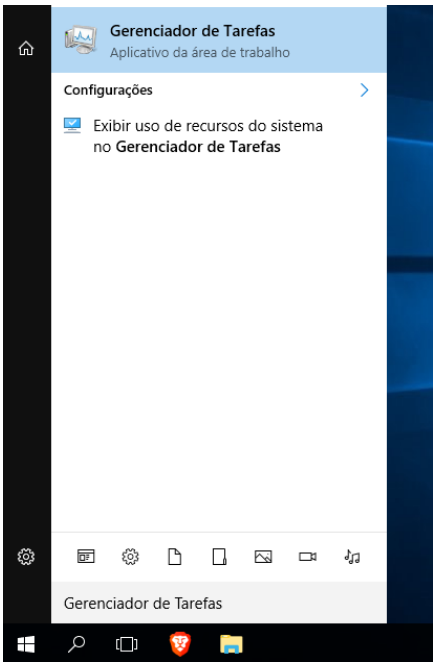


Figura 1. Acesso ao Gerenciador de Tarefas (Task Manager) do Windows.

Caso o Gerenciador de Tarefas seja inicializado no modo simplificado, basta clicar na opção “Mais detalhes” no canto inferior esquerdo (Figura 2).

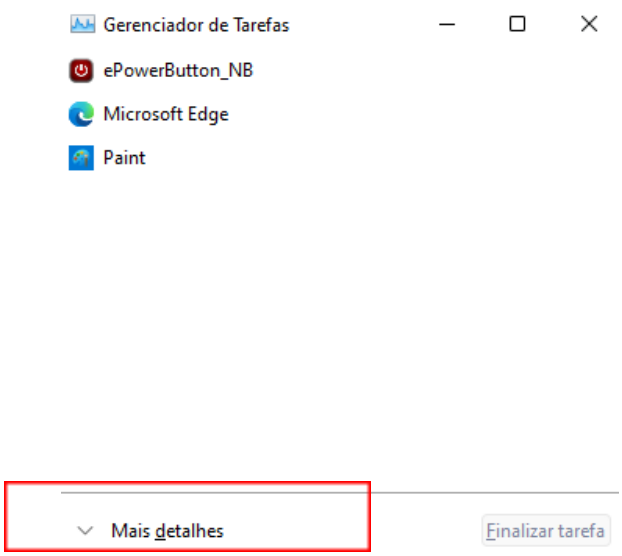


Figura 2. Tela inicial do “Gerenciador de Tarefas”.

.

As opções de monitoramento e gerenciamento do computador são (Figura 3):

- Processos;
- Desempenho;
- Histórico de aplicativos;
- Inicializar;
- Usuários;
- Detalhes;
- Serviços.

Processos

Na guia “Processos”, todos os processos do sistema operacional Windows e aplicações dos usuários no computador são exibidos (Figura 3).

Os processos são divididos em “aplicativos” e “processos em segundo plano”.

Os aplicativos são os programas que possuem janela visível na interface do Windows e os processos em segundo plano (background) são os processos executados sem uma janela visível.

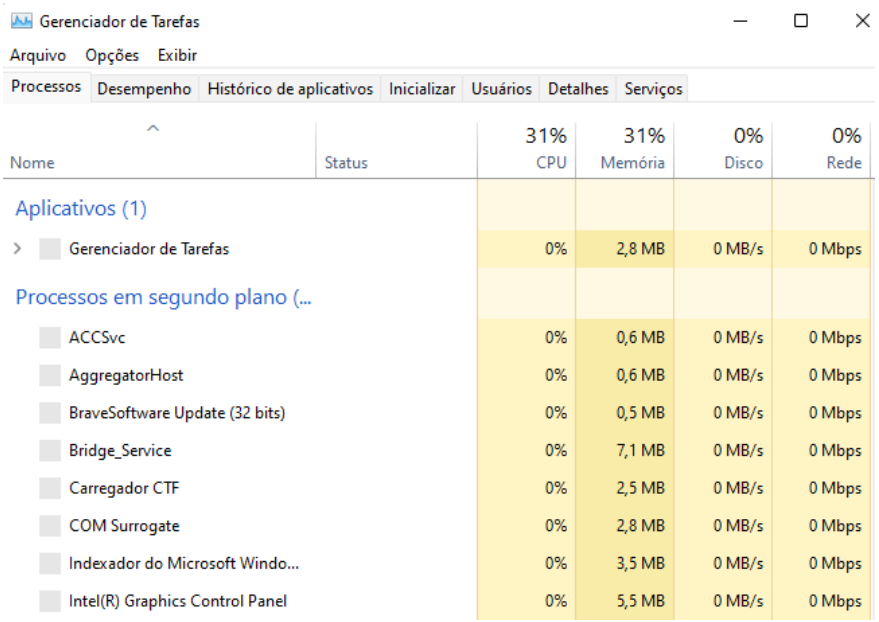


Figura 3. Processos em execução no computador.

Detalhes

A aba “Detalhes” (Figura 4) exibe informações mais detalhadas dos processos, que estão executando no computador (Figura 3).

Em “Detalhes”, são exibidos:

- Nome: exibe o nome do processo;
- PID (Process Identifier - Identificador do Processo): número que identifica o processo no sistema operacional;
- Status: exibe o estado do processo, se está em execução ou suspenso;
- Nome de usuário: o nome do usuário do sistema a qual o processo pertence, ou seja, qual o usuário que iniciou a execução do processo;
- CPU: a quantidade de processamento que o processo está utilizando;
- Memória: quantidade de memória principal que o processo está utilizando;
- Arquitetura: exibe a arquitetura na qual o processo precisa ser executado e está sendo executado. É interessante para saber quais processos utilizam qual arquitetura;
- Descrição: breve descrição do que se trata o processo.

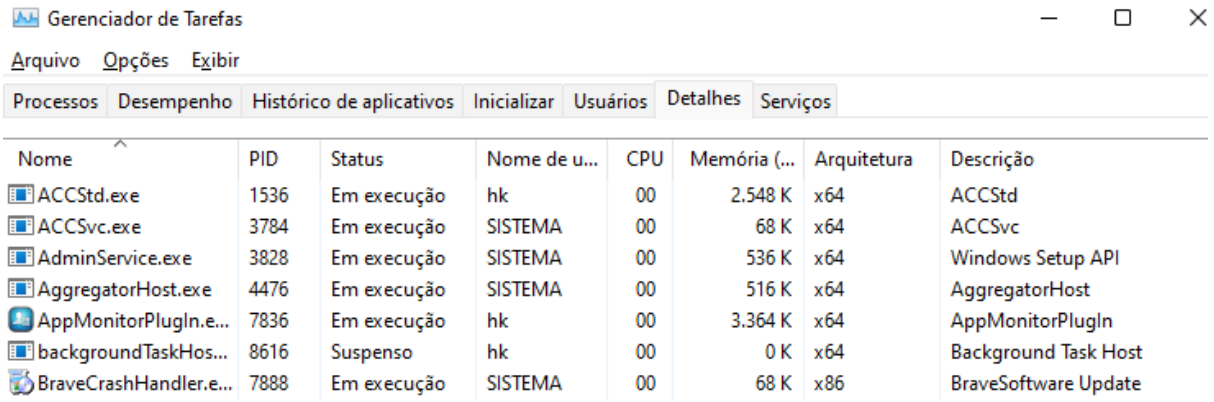


Figura 4. Opção “Detalhes” do Gerenciador de Tarefas.

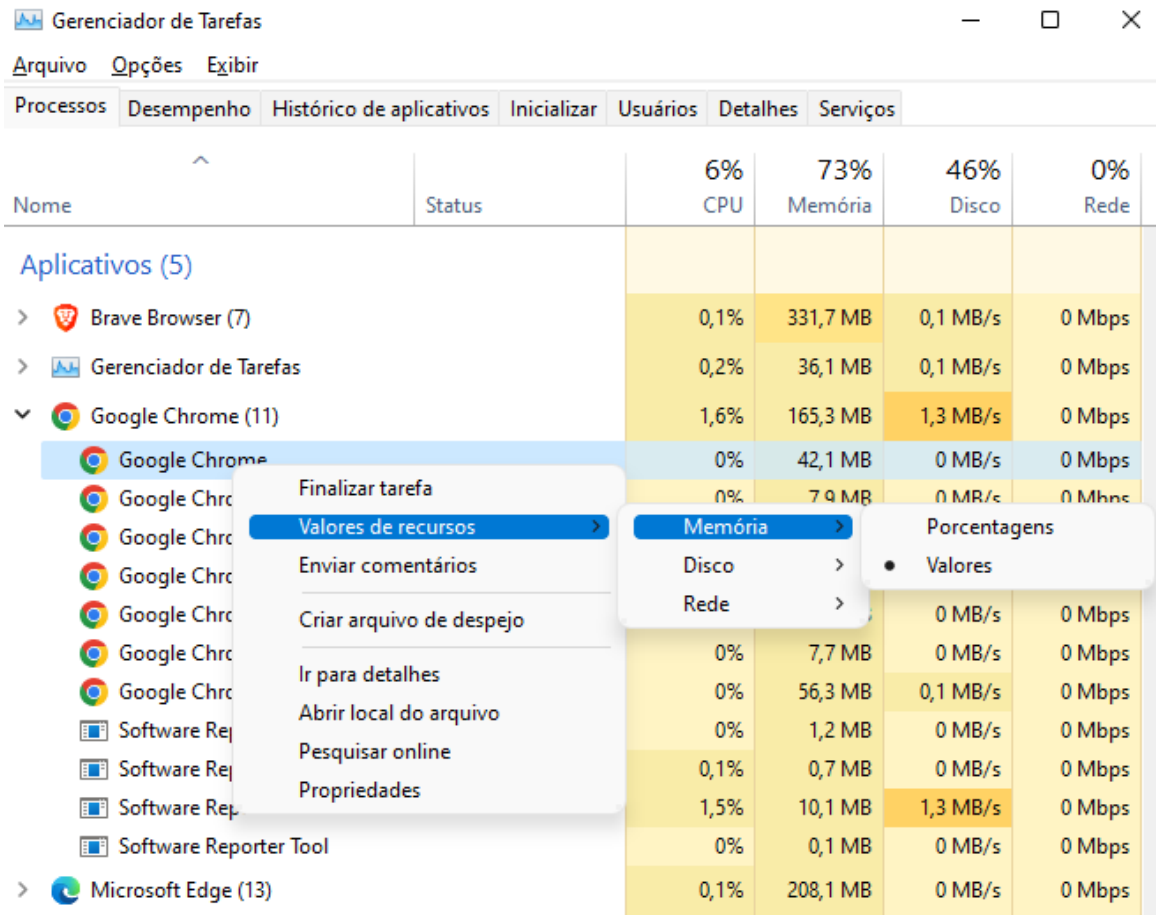


Figura 5. Opções de informação e visualização dos processos via Gerenciador de tarefas.

Finalizando processos no Windows

Quando você executa um programa no computador, o Windows carrega uma “instância” do programa na memória do computador.

Essa instância ou código em execução no processador é chamada de “processo”, no entanto, devido a uma questão técnica, como por exemplo mau funcionamento da aplicação

ou travamento, é permitido ao usuário, pelo sistema operacional, finalizar os processos gerados por ele, ou seja, finalizar processos aos quais possui hierarquia.

Os passos para finalizar processos podem ocorrer de duas formas:

- Gerenciador de tarefas;
- Prompt de comando e ou Power Shell.

Usando o Gerenciador de tarefas para finalizar processos

Como visto anteriormente, a ferramenta Gerenciador de tarefas permite encerrar processos indesejados que estejam na hierarquia do usuário, conforme as figuras a seguir:

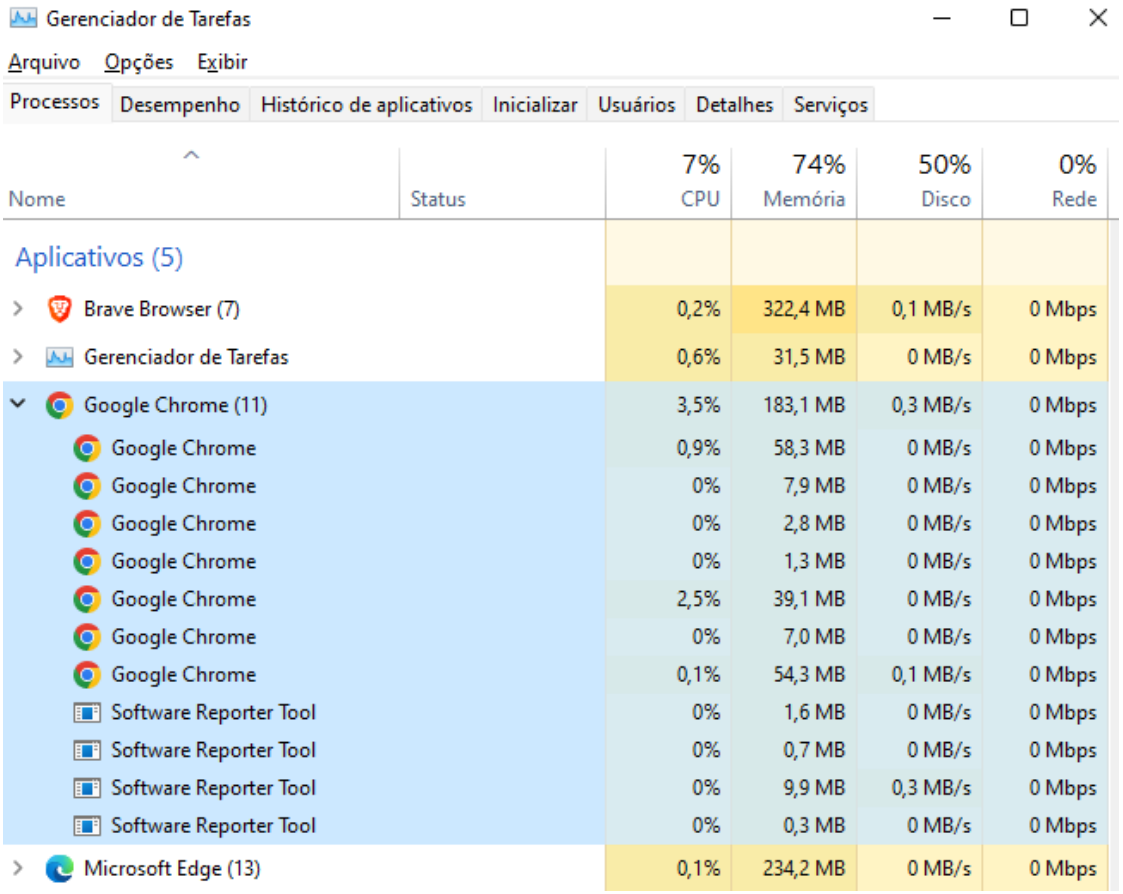


Figura 6. Interface do Gerenciador de tarefas com processos em execução.

Processo-pai e processo-filho

Processo-pai

O processo-pai consiste no processo principal ou primeiro processo gerado pelo usuário, ou seja, é o processo que vai gerar outros sub-processos ou processos-filhos.

Caso o processo-pai seja finalizado, automaticamente irá finalizar todos os processos-filho gerados a partir dele.

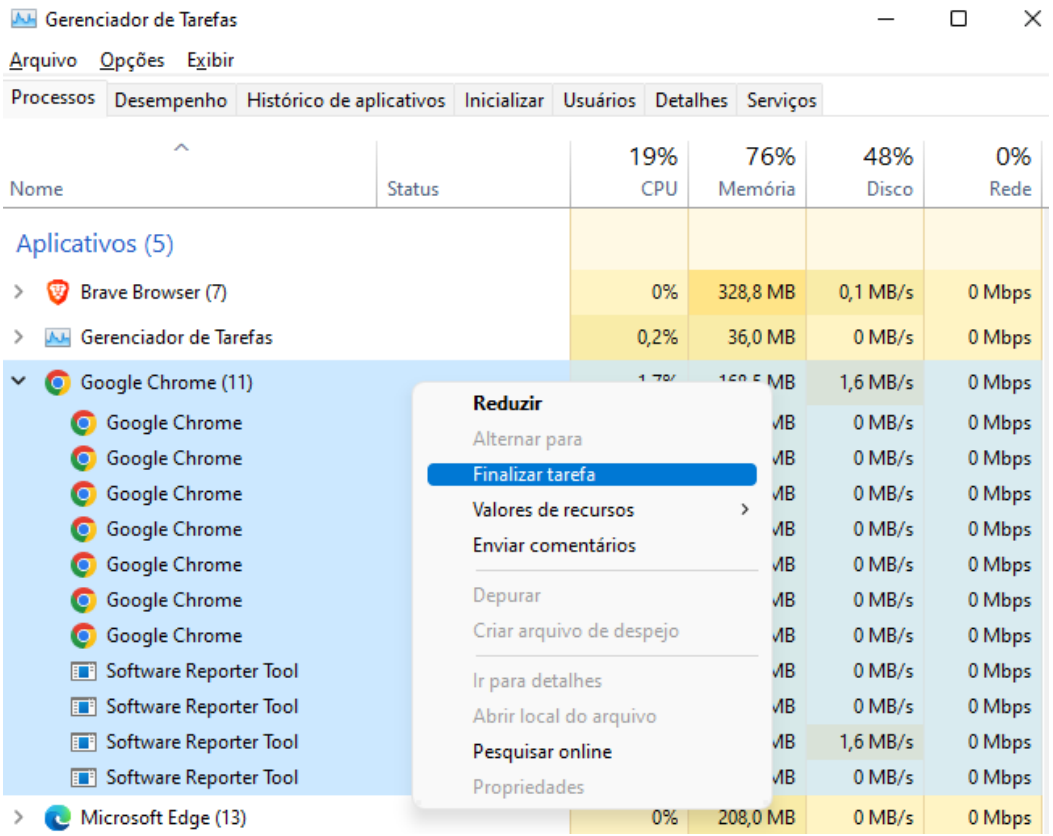


Figura 7. Encerrando processo-pai via Gerenciador de tarefas.

Processo-filho

Processo-filho ou sub-processo é todo processo gerado a partir do processo-pai.

Se um processo-filho for finalizado, não implica na finalização do seu processo-pai ou em outros processos-filho.

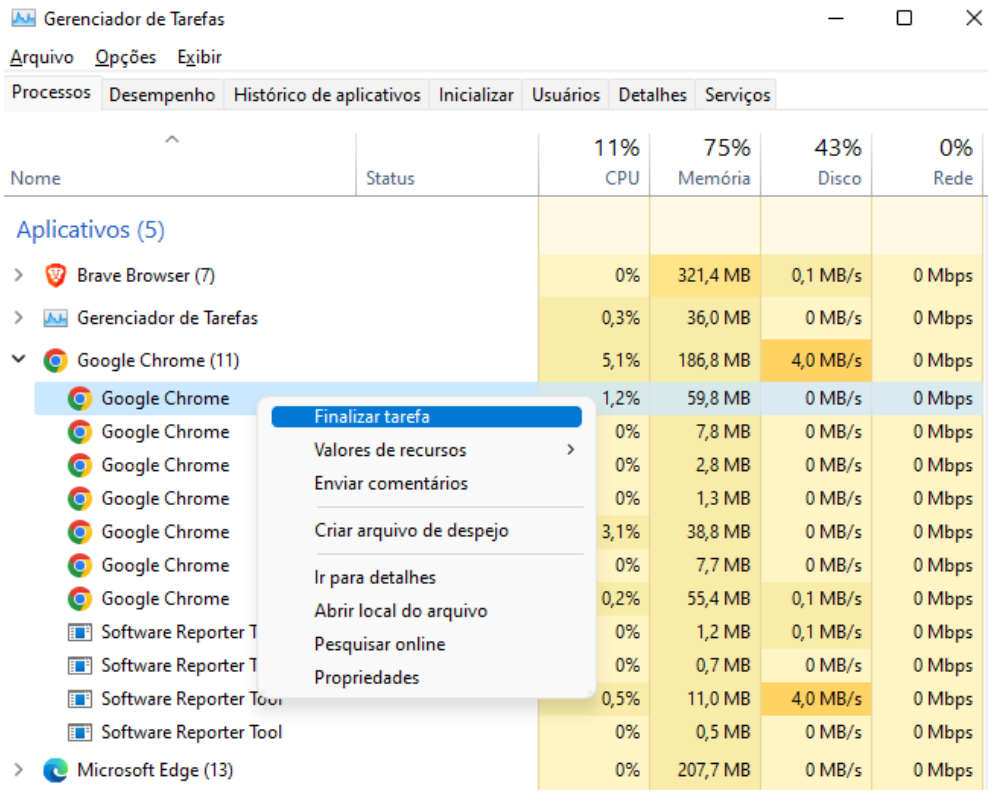


Figura 8. Encerrando processo-filho via Gerenciador de tarefas.

Usando o Prompt de comando e ou Power Shell para finalizar tarefas

Para finalizar processos utilizando o prompt de comando do windows, chamado “taskkill”, que como o nome sugere, permite finalizar processos em execução, basta digitar CMD no menu Iniciar, depois, clicar com o botão direito do mouse e selecionar a opção “Executar como administrador”.

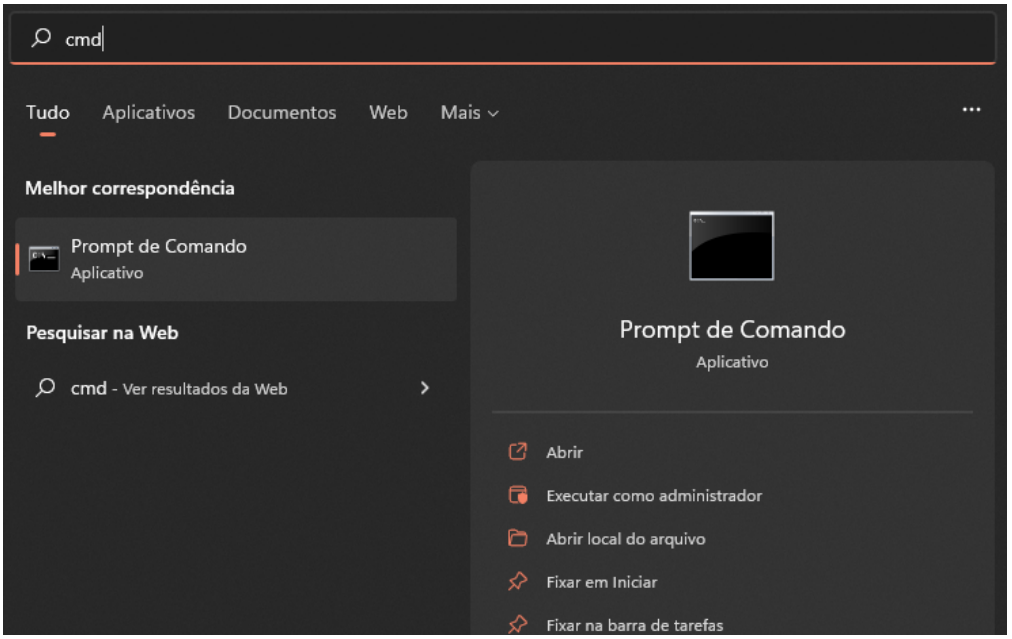


Figura 9. Prompt de comando do Windows “taskkill”.

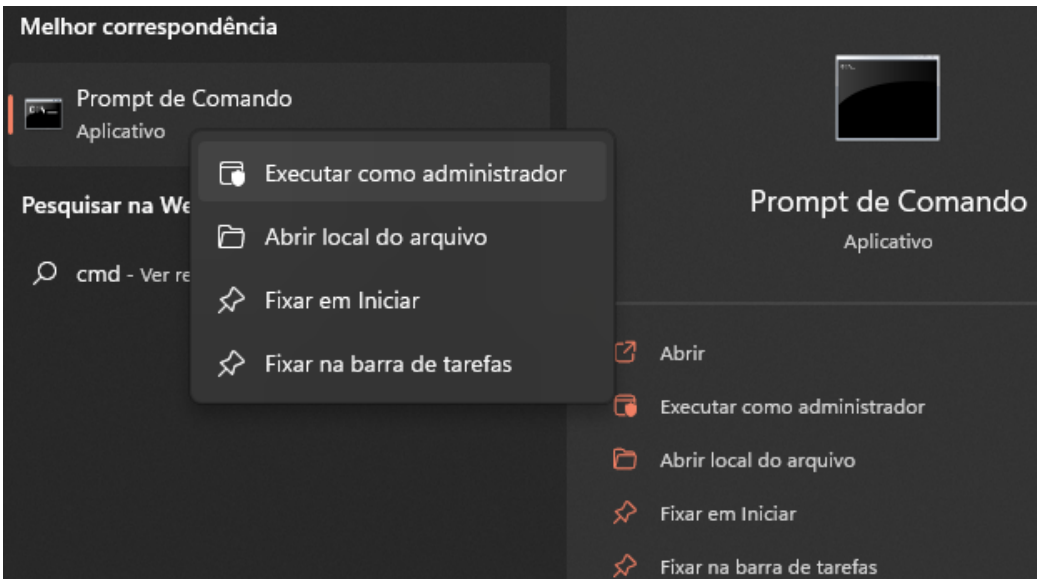


Figura 10. Prompt de comando sendo acionado no modo administrador.

O passo a seguir da execução do prompt é conhecer o nome ou o PID do processo. Desta forma, basta digitar ***tasklist*** para listar todos os processos em execução no computador, assim, os processos são listados com o nome da imagem e identificador (PID).

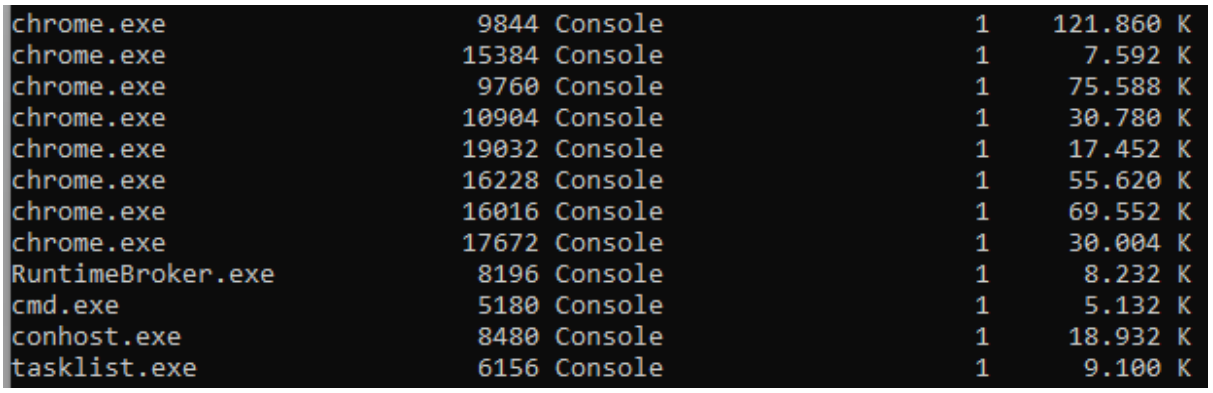
```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.22000.739]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>tasklist

Nome da imagem            Identifi Nome da sessão       Sessão# Uso de memór
========================= ======== ================ =========== ============
System Idle Process             0 Services                   0          8 K
System                          4 Services                   0      2.536 K
Registry                      160 Services                   0     87.092 K
smss.exe                      516 Services                   0      1.120 K
csrss.exe                     776 Services                   0      5.844 K
wininit.exe                   860 Services                   0      6.092 K
csrss.exe                     876 Console                    1      6.256 K
services.exe                  936 Services                   0      8.780 K
winlogon.exe                  968 Console                    1     11.724 K
lsass.exe                     336 Services                   0     35.524 K
svchost.exe                   600 Services                   0     67.148 K
WUDFHost.exe                 1028 Services                   0      5.640 K
fontdrvhost.exe              1080 Console                    1      8.716 K
fontdrvhost.exe              1100 Services                   0      3.440 K
svchost.exe                  1196 Services                   0     29.576 K
WUDFHost.exe                 1236 Services                   0      5.056 K
svchost.exe                  1288 Services                   0     19.592 K
dwm.exe                      1396 Console                    1    126.852 K
svchost.exe                  1520 Services                   0     22.172 K
svchost.exe                  1544 Services                   0     32.612 K
svchost.exe                  1612 Services                   0     19.392 K
svchost.exe                  1672 Services                   0     22.312 K
```

Figura 11. Listando processos com o comando tasklist.

O comando ***taskkill*** pode ser utilizado de duas formas diferentes.

Caso se queira usar o PID do processo, o comando fica:

> taskkill /F /PID pid_processo



chrome.exe	9844	Console	1	121.860 K
chrome.exe	15384	Console	1	7.592 K
chrome.exe	9760	Console	1	75.588 K
chrome.exe	10904	Console	1	30.780 K
chrome.exe	19032	Console	1	17.452 K
chrome.exe	16228	Console	1	55.620 K
chrome.exe	16016	Console	1	69.552 K
chrome.exe	17672	Console	1	30.004 K
RuntimeBroker.exe	8196	Console	1	8.232 K
cmd.exe	5180	Console	1	5.132 K
conhost.exe	8480	Console	1	18.932 K
tasklist.exe	6156	Console	1	9.100 K

Figura 12. Lista de processos usando o comando tasklist.

Exemplo:

taskkill /F /PID 9844

Caso se queira usar o nome do processo, o comando fica:

> taskkill /IM “nome_processo”

ou

> taskkill /IM nome_processo

Obs: usando o nome do processo encerra-se todas as instâncias do processo, ou seja, todos os processos-filhos, no entanto, usando o PID, finaliza-se apenas o processo especificado.

Exemplo:

> taskkill /IM “chrome.exe”

ou

> taskkill /IM chrome.exe

Para encerrar vários processos usando um único comando, pode-se utilizar a opção /IM para finalizar pelo no nome ou /PID pelo número identificador do processo.

Exemplo:

> taskkill /IM “chrome.exe” /IM “brave.exe”

ou

> taskkill /IM chrome.exe /IM brave.exe

Exemplo:

> taskkill /PID 9844 /PID 5180

Usando o Powershell para finalizar processos/tarefas

Para finalizar processos com o Power Shell utiliza-se os mesmos comandos ***tasklist*** e ***taskkills*** com as mesmas opções utilizadas no prompt de comando do Windows.

Para acessar o terminal do Power Shell, basta pressionar as teclas windows+r (tecla da janela do windows +r) e logo após digitar ***powershell*** e enter.

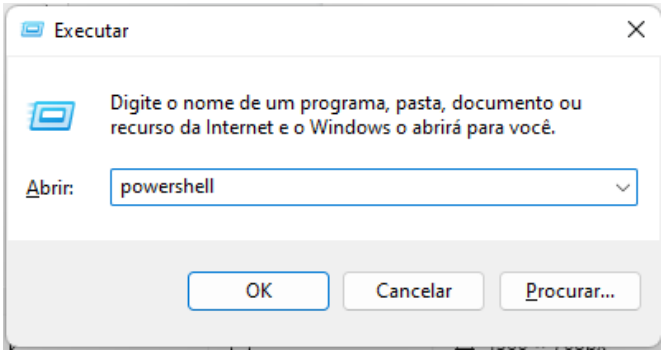


Figura 13. Executando o Power Shell.

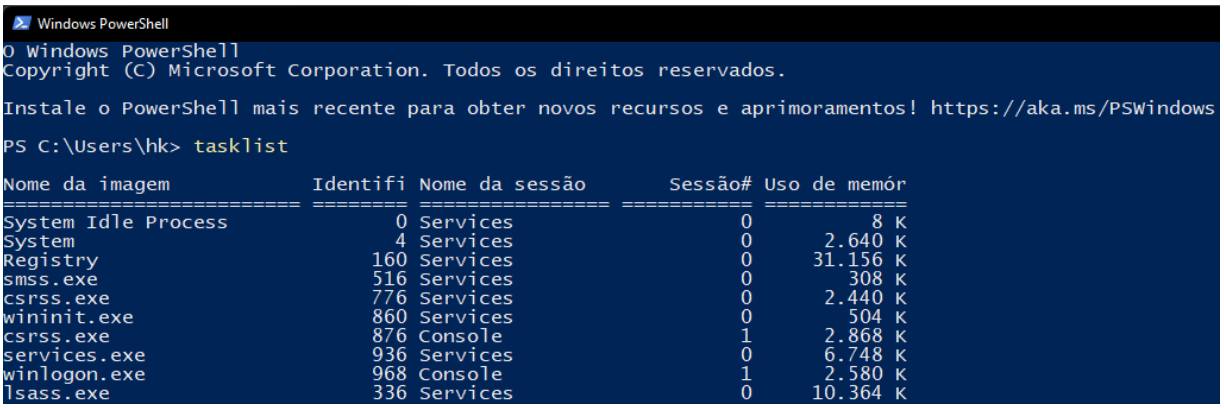


Figura 14. Janela de Power Shell executando o comando tasklist.

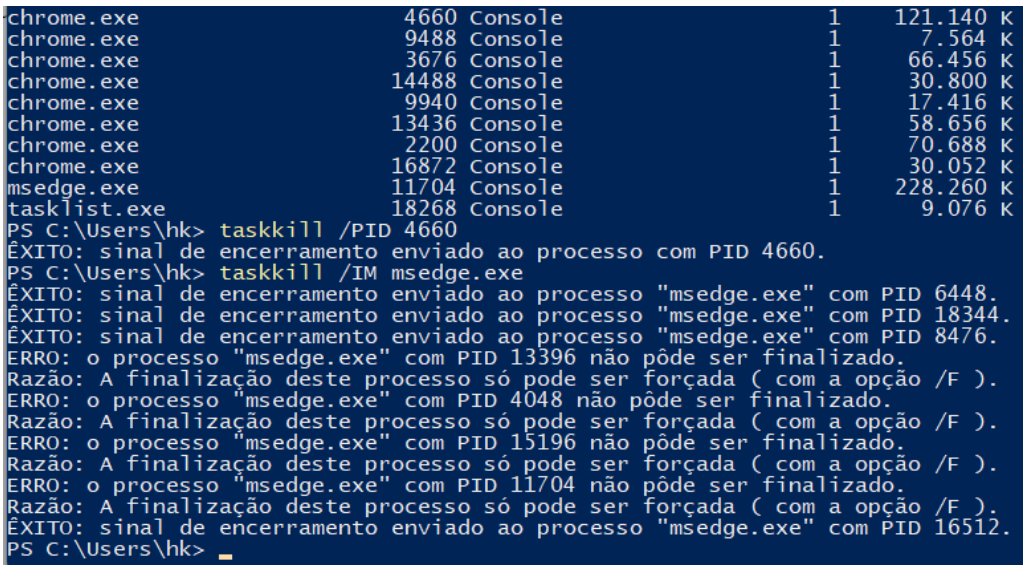


Figura 15. Uso dos comandos taskkill via nome do processo e PID.

Parando processos usando Power Shell

Além de finalizar tarefas, o Power Shell também permite que processos sejam parados “STOP”, que pode ser feito pelo nome do processo ou pelo seu identificador PID.

Exemplo:

> tasklist

chrome.exe 8480 Console 1 129.632 K

msedge.exe 11784 Console 1 30.044 K

> Stop-Process -Name chrome

> Stop-Process -Id 11784

Forçando a finalização de um processo

Caso um processo/tarefa não receba o sinal de finalização, o Power Shell oferece a opção -Force para que sua finalização seja feita de forma forçada, ou seja, obrigatória.

Exemplo:

> taskkill /F /IM chrome.exe

> taskkill /F /PID 9844

ou

> taskkill /IM chrome.exe /F

> taskkill /PID 9844 /F

Gerenciando processos com o Power Shell

Uma forma simples de se obter os processos em execução e seu status no sistema operacional Windows é via utilização do comando ***Get-Process***.

O comando também permite obter informações sobre processos específicos definindo seus nomes de processo ou PID.

Exemplo:

```
> Get-Process -id 0
Handles    NPM(K)    PM(K)    WS(K)    VM(M)    CPU(s)    Id    ProcessName
----------    ----------    --------    ---------    ---------    ----------    --    ------------------
         0             0           0          16           0                0    Idle
```

O Power Shell permite buscar processos por nome usando strings ou parte dos nomes.

Exemplo:

```
> Get-Process -Name ex*
Handles  NPM(K)      PM(K)      WS(K)   VM(M)   CPU(s)     Id   ProcessName
-------      ------          -----          -----       -----       ------          --   -----------
234            7           5572        12484     134       2.98      1684   EXCEL
555           15          34500        12384     134     105.25       728    explorer
```

Também é possível realizar busca por processos com nomes diferentes.

Exemplo:

```
> Get-Process -Name ex*, power*
Handles  NPM(K)      PM(K)      WS(K)   VM(M)   CPU(s)       Id   ProcessName
-------      ------          -----          -----       -----         ------          --   -----------
234            7           5572        12484     134         2.98      1684    EXCEL
555           15          34500        12384     134       105.25       728    explorer
605            9          30668         29800     155         7.11      3052    powershell
```

O Power Shell possibilita obter informações de processos em computadores remotos.

Exemplo:

```
> Get-Process -Name PowerShell -ComputerName localhost, server1, server2
Handles    NPM(K)    PM(K)    WS(K)    VM(M)    CPU(s)        Id    ProcessName
----------    ----------    --------    ---------    ---------    ----------        --    ------------------
258                8    29772     38636       130                3700     powershell
398               24    75988     76800       572                5816     powershell
605                9    30668     29800       155         7.11   3052     powershell
```

Neste contexto, o POwer Shell também permite interromper/parar um processo em um host remoto, no entanto, é necessário utilizar o comando ***Invoke-Command*** seguido da opção ***-ComputerName***.

Exemplo:

```
> Invoke-Command  -ComputerName  Server01  {Stop-Process Powershell}
```

Caso o uso da opção *-ComputerName* retorne erro é necessário adicionar a propriedade MachineName à exibição padrão de Get-Process usando o comando a seguir.

Exemplo:

```
> Get-Process powershell -ComputerName localhost, server1, server2 |
        Format-Table -Property Handles,
        @{Label="NPM(K)";Expression={[int]($_.NPM/1024)}},
        @{Label="PM(K)";Expression={[int]($_.PM/1024)}},
        @{Label="WS(K)";Expression={[int]($_.WS/1024)}},
        @{Label="VM(M)";Expression={[int]($_.VM/1MB)}},
        @{Label="CPU(s)";Expression={if ($_.CPU -ne $()){$_.CPU.ToString("N")}}},
        Id, ProcessName, MachineName -auto
```

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName	MachineName
258	8	29772	38636	130		3700	powershell	server1
605	9	30668	29800	155	7.11	3052	powershell	server2

Interrompendo processos com confirmação

Para forçar a parada/interrupção de um processo com opção de confirmação, para que não se interrompa acidentalmente um processo, pode-se utilizar o parâmetro ***Confirm***. A opção Confirm é bastante útil principalmente quando se usa caractere curinga ao especificar o nome do processo, dessa forma a lista de processos retornados pode ser grande e diversificada, portanto, lista-se muitos processos que podem ter nome semelhantes, dessa forma, quando se utiliza a interrupção de processos em massa pode-se interromper processos que não se deseja parar, neste contexto o Confirm pergunta ao usuário se este deseja interromper os processo um a um.

Exemplo

```
> Stop-Process   -Name   t*,e*   -Confirm
Confirm
Are you sure you want to perform this action?
Performing operation "Stop-Process" on Target "explorer (408)".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help
(default is "Y"): n

Confirm
Are you sure you want to perform this action?
Performing operation "Stop-Process" on Target "taskmgr (4072)".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help
(default is "Y"): n
```