

DISCIPLINA SISTEMAS OPERACIONAIS

AULA 1

Visão Geral de Sistemas Operacionais Practices

BY SPTECH SCHOOL

2022

Laboratório - PowerShell

i) Práticas

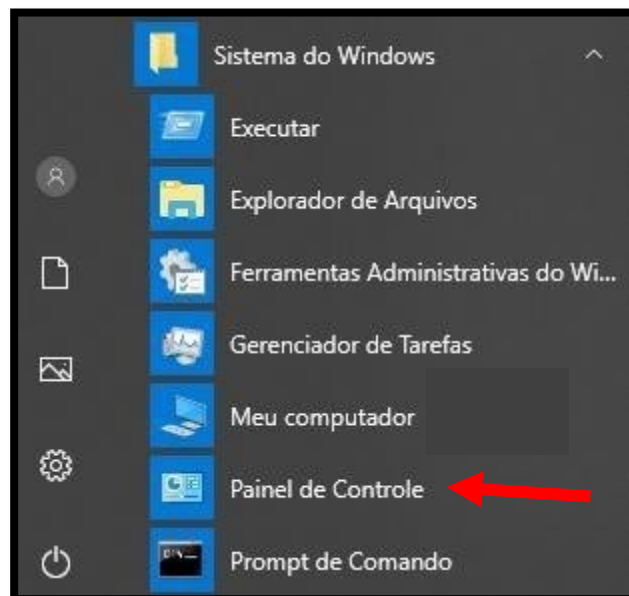
i1- Análise das características do Sistema Operacional e o recurso do computador

MENU INICIAR

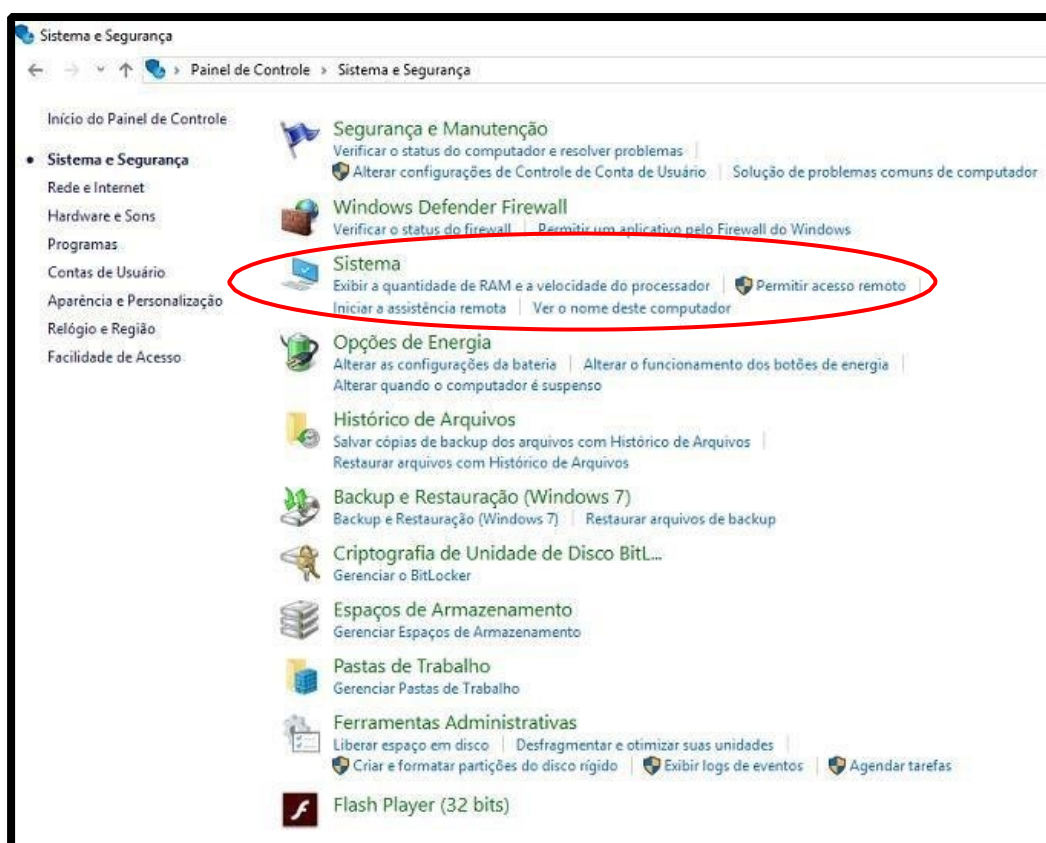
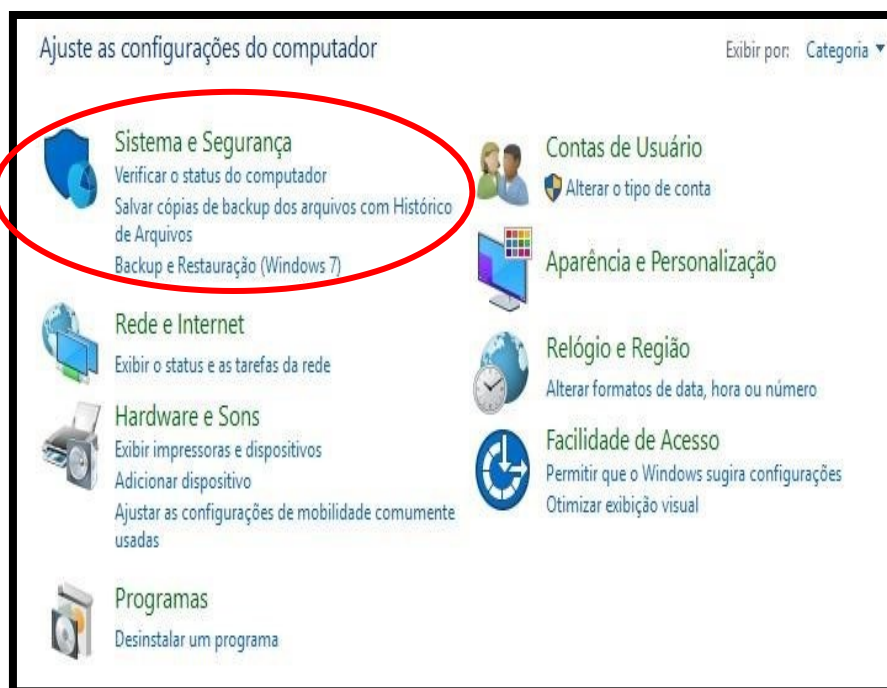
PAINEL DE CONTROLE

SISTEMA

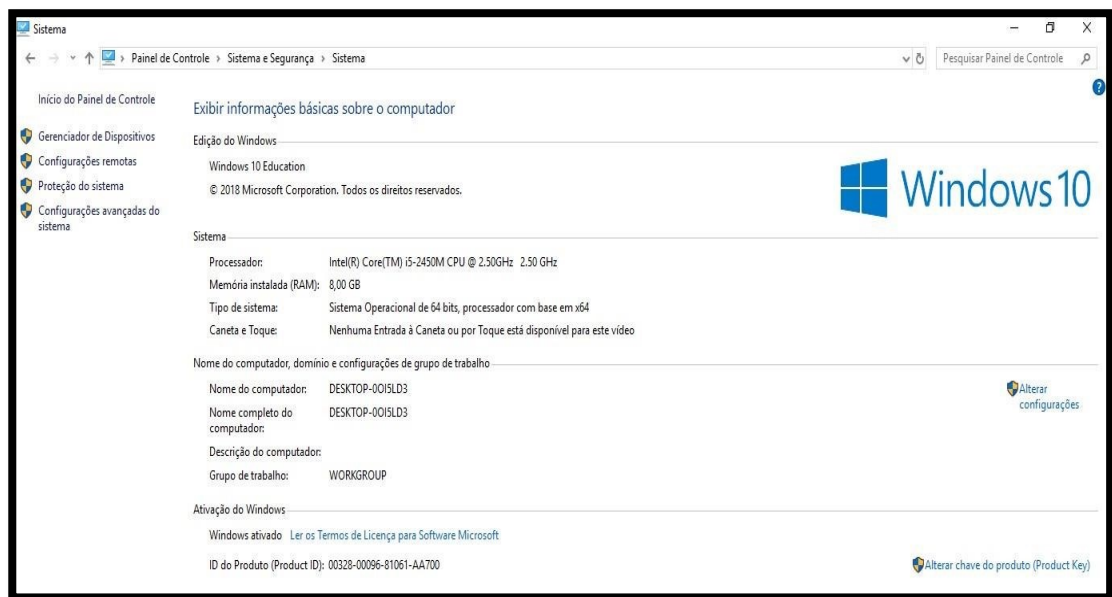
SISTEMA DO WINDOWS



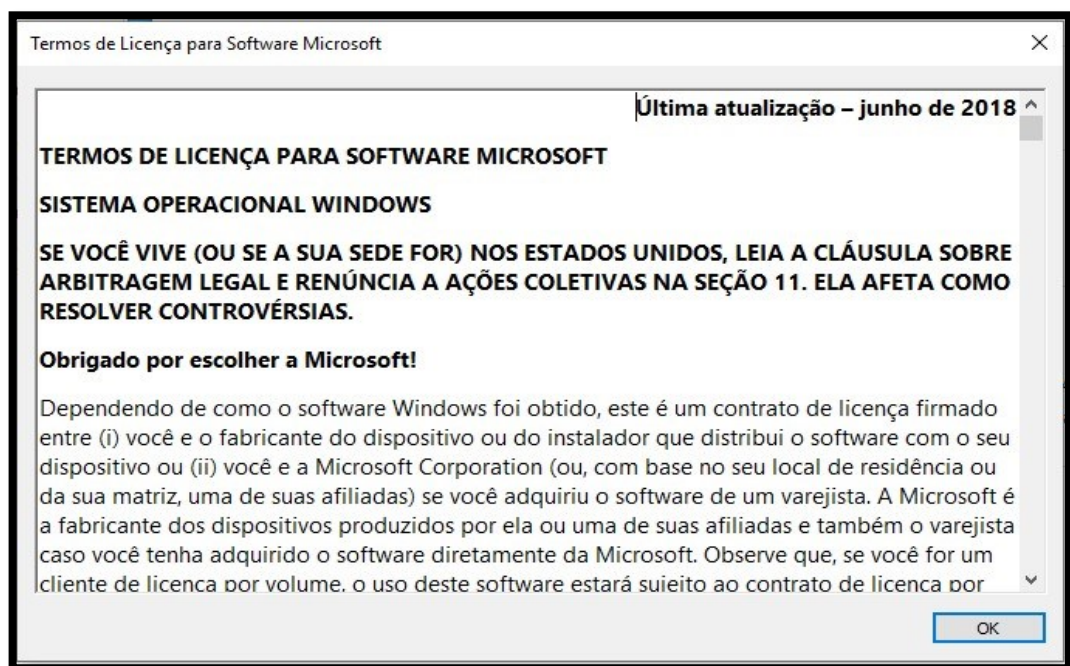
EM ALGUNS CASOS PODE IR DIRETAMENTE PAINEL DE CONTROLE 
SISTEMA



Verifique a versão do Sistema Operacional, a memória, e o processador



Agora verifique a Licença de uso do Sistema Operacional e a chave ID do produto SO Microsoft

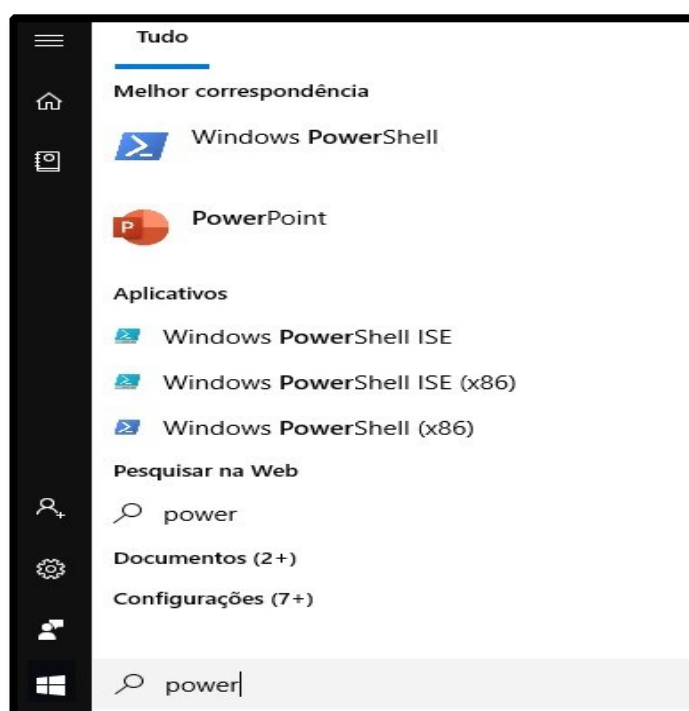


Em alguns casos, a licença pode estar por volume e o contrato se encontra com a empresa.

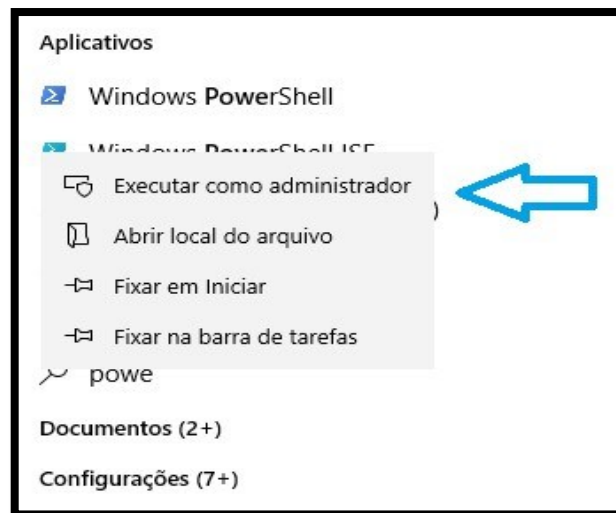
Agora vamos fazer isso por outra forma. Por meio do Windows Shell. Lembrando que “O PowerShell ajuda os administradores do sistema e os usuários avançados a automatizar rapidamente as tarefas que gerenciam processos e sistemas operacionais (Linux, macOS e Windows)”.

TOME NOTA

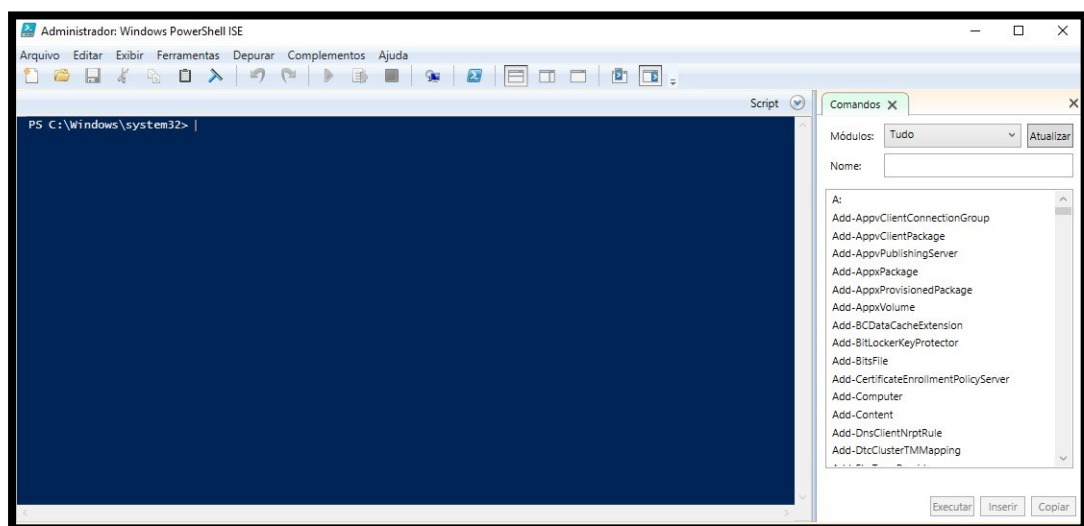
O **Windows PowerShell (WPS)** é o shell de linha de comando do Windows. Shell é uma interface que permite aos usuários interagir com o sistema operacional e pode ser tanto no modo gráfico *Graphical User Interface (GUI)* quanto em modo texto *Command-Line Interface (CLI)*. O **WPS** inclui um prompt interativo e um ambiente para criação de scripts para administração do sistema e automação. Compilado sobre o **CLR (Common Language Runtime)** do .NET Framework permite que profissionais de TI e desenvolvedores controlem e automatizem a administração do Windows e aplicativos. O **WPS** usa linguagem de script expressiva, com expressões regulares e permite o uso do .NET Framework, **Windows Management Instrumentation (WMI)**, dentre outras funcionalidades. O **WPS** introduz o conceito de **cmdlet** (pronuncia-se "**command-let**"), uma ferramenta de linha de comando simples, de função única e compilada no *shell*. Também pode usar o **ISE (Integrated Scripting Environment)**, onde pode-se executar comandos, gravar, testar e depurar scripts em uma interface de usuário gráfica baseada no Windows.



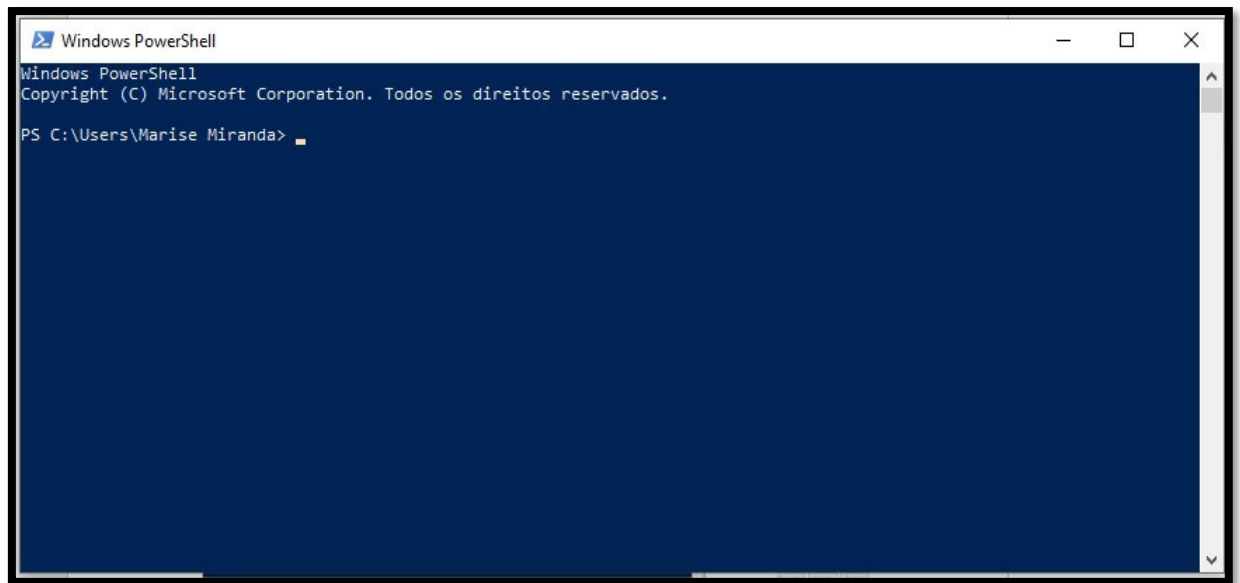
Selecione o WPS ISE com o botão direito



Se executar o WPSISE como administrador, você entrará no diretório system32, caso não seja como admin, o sistema WPSISE aparecerá com o diretório do usuário.



Feche esse WPS ISE e abra o WPS



A diferença está em quem acessa o Sistema – System32 e o outro o Usuário.

O que é o System 32?

É o local do Sistema de diretórios do Windows em que se encontram todas as DLLs necessárias para o sistema operacional e os aplicativos funcionem corretamente.



Satya Nadella - CIO da Microsoft

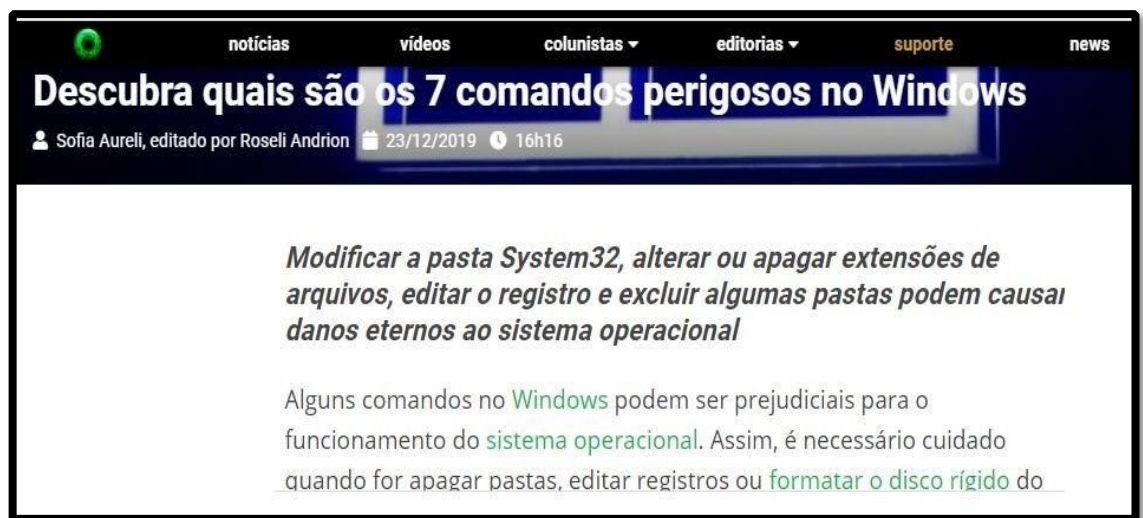
"No pode deletar o pasta da System32"

Recomendação:

Acesse a matéria do Olhar Digital de dezembro de 2019 que traz alertas sobre os perigos de modificar a pasta System32 do Sistema Operacional Windows.

Clique no link abaixo:

https://olhardigital.com.br/dicas_e_tutoriais/noticia/descubra-quais-sao-os-7-comandos-perigosos-no-windows/94726



Voltando ao WPS (Windows Power Shell) que é uma DLL de mecanismo de script que é inserida em vários hosts. Os hosts mais comuns que você iniciará são a linha de comando interativa **powershell.exe** e o Ambiente de Script Integrado **powershell_ise.exe**.

pwsh.exe para o Windows

pwsh para o macOS e o Linux



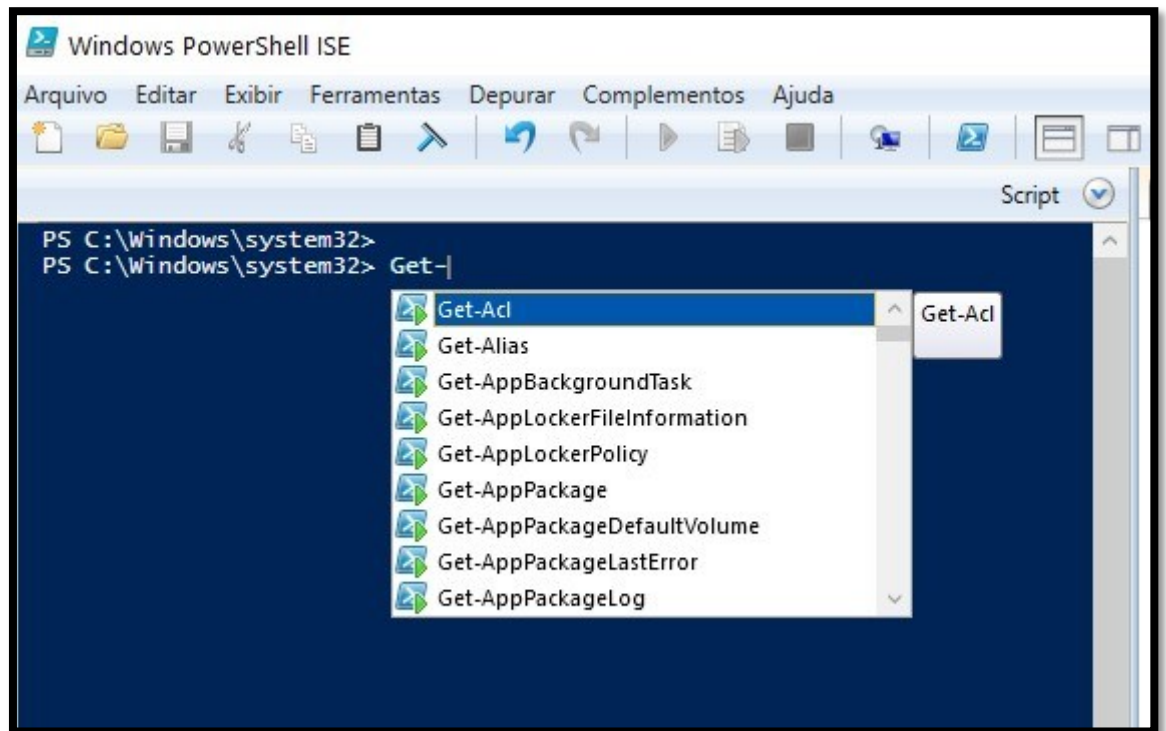
DLL – Dynamic link library (Biblioteca de vínculo dinâmico)

Para os sistemas operacionais Microsoft, grande parte da funcionalidade do sistema operacional é fornecida por bibliotecas de link dinâmico (DLL). O uso de DLLs ajuda a promover a modularização do código, a reutilização do código, o uso eficiente da memória e o espaço reduzido do disco (Microsoft em <https://support.microsoft.com/pt-br/help/815065/what-is-a-dll>, acessado no dia 20/12/2019). Para saber mais acesse os links a seguir <https://windows10dll.nirsoft.net/index.html> <https://dll.website/> <https://epson.com.br/peps-drivers-apis-impressoras-fiscais>

Vamos a alguns comandos do WPS usando o Power Shell ISE

Abra o ambiente WPS ISE

Digite **Get-**



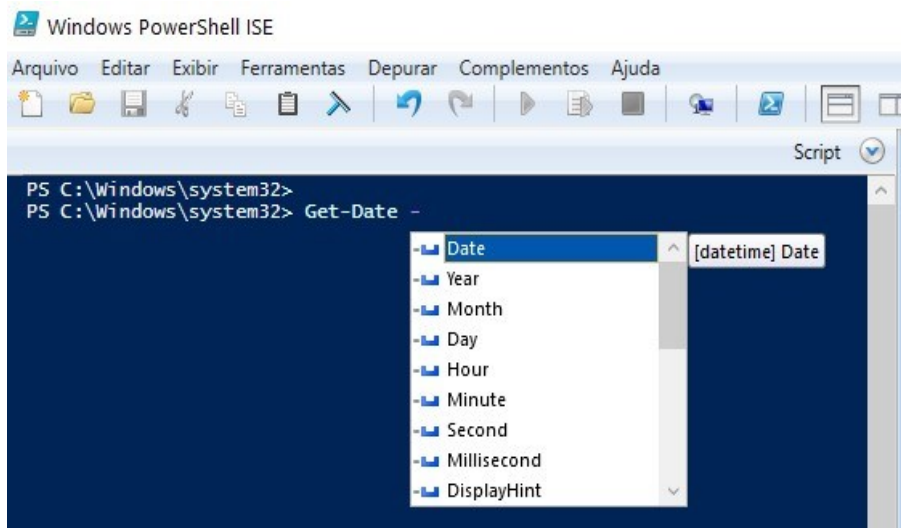
P WPS ISE utiliza a tecnologia IntelliSense completando os comandos, parâmetros e as variáveis criadas.

Vamos utilizar o cmdlet **Get-Date**

terça-feira, 28 de dezembro de 2019 16:59:58

Então esse comando captura a data corrente da máquina local

Vamos a outros parâmetros.



Get-Date -Date"arguments"

Exemplo: Get-Date - Date "05/02/2020" Retorna:

quarta-feira, 5 de fevereiro de 2020 00:00:00

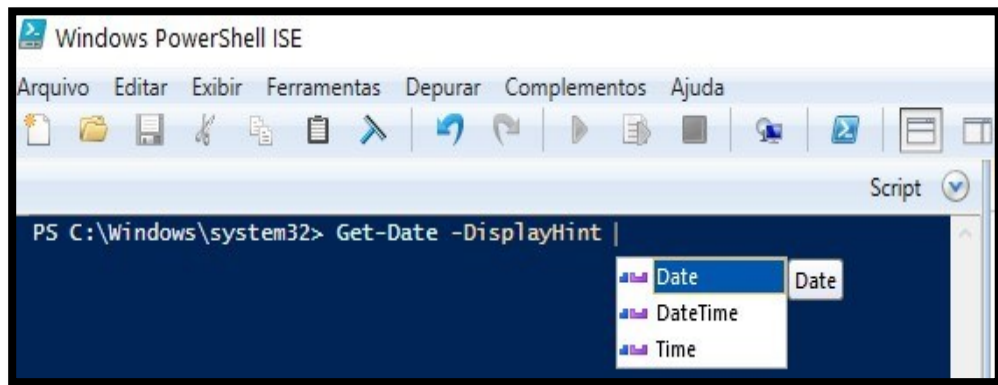
Faça o teste mudando a string do argumento para o ano de 2019

Agora, faça o teste em cada um destes cmdlets

Get-Date -DisplayHint

DisplayHint é o modo de exibição, veja a sequência seguinte

Aperte a tecla ESPAÇO para que o parâmetro correto apareça



Verifique o que sucede com os três parâmetros anteriores, Date, DateTime e Time.

Complete a Analise o que retorna de cada comando:

`Get-Date -DisplayHint Date`

`Get-Date -DisplayHint DateTime`

`Get-Date -DisplayHint Time`

Seja capaz de digitar sem ajuda da ferramenta, faça o teste no WPS.

Observe que os comandos funcionam apenas não temos o auxílio do *IntelliSense*.

Você já se fez uma pergunta? Para que servem esses comandos?

Resposta: **Captura de dados/informações**

Continuando com a prática

`Get-Date -Format "dddd MM/dd/yyyy HH:mm K"`

Qual a diferença dos parâmetros do cmdlet anterior e com os seguintes?

`Get-Date -Format "dddd MM/dd/yyyy"`

`Get-Date -Format "dddd MM/dd"`

```
Get-Date-Format"dddd"
```

ATENÇÃO: Parâmetro `k` na string do cmdlet significa Deslocamento de **fuso horário** da Universal Time Coordinate (UTC), atrasado quando negativo e adiantado quando positivo.

Resposta:

A diferença entres os 4 comandos é o formato de saída dos parâmetros na string de data completa

```
terça-feira 01/28/2020 18:08 -03:00
```

```
terça-feira 01/28/2020
```

```
terça-feira 01/28
```

```
terça-feira
```

Para saber mais detalhes dos cmdlets date consulte

<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-date?view=powershell-7>

Vamos escolher outro parâmetro de data.

`Get-Date -Year 1999`

O que este parâmetro faz: Busca a data (dia e mês atual) mas do ano de 1999

Retorna resposta:

`quinta-feira, 28 de janeiro de 1999 18:18:37`

Verifique qual foi o dia da semana da sua **data de nascimento**

`Get-Date -Year xxxx -Month x -Day xx`

Resposta:

Selecione alguns alunos para verificar o dia da semana, no teste com a data de nascimento do professor retornou quarta feira.

O Aluno pode confrontar a data com o calendário do computador



Vamos verificar a chamada dentro do WPS ISE de um método para checar se o fuso horário da máquina está ajustado para o horário de verão.

O DST (DaylightSavingTime) é método instanciado em DateTime, retornando uma booleana *true* ou *false* se essa instância está dentro do horário de verão em relação ao fuso horário do computador.

`IsDaylightSavingTime`

DaylightSavingTime – horário de verão

```
PS C:\Windows\system32> $DST = Get-Date  
PS C:\Windows\system32> $DST.IsDaylightSavingTime() False
```

No caso do teste aqui retornou false

Vamos agora ver um pouco do poder do WPSISE

Digite

Get-Command -Verb Get

Veja a lista de cmdlets, alias e funções

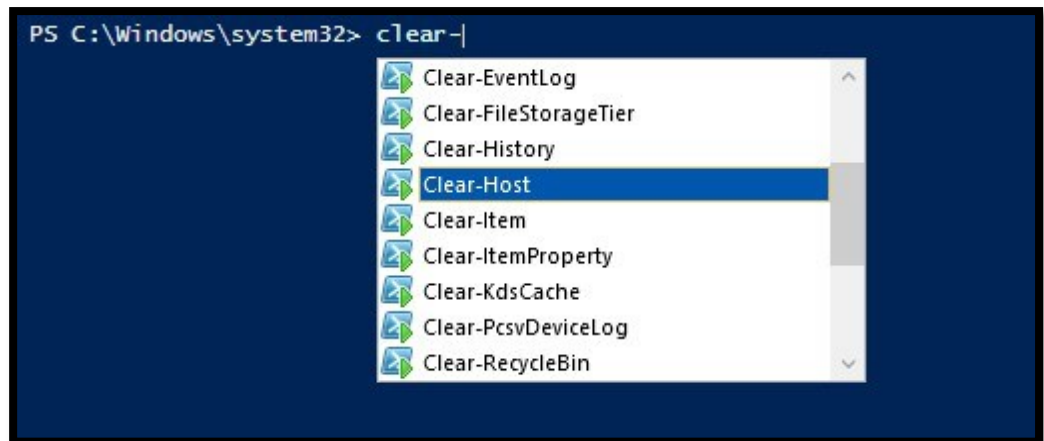
Observe a lista de funções, alias e cmdlets

TOME NOTA

O *PowerShell* dá suporte aos *aliases* para fazer referência aos comandos através de nomes alternativos. Usar alias permite que usuários com experiência em outros *shells* utilizem nomes de comando comuns que já conhecem para operações semelhantes no *PowerShell*. O alias associa um novo nome a outro comando.

<https://docs.microsoft.com/pt-br/powershell/scripting/learn/using-familiar-command-names?view=powershell-5.1>

Uma prática simples é o comando “clear screen”, limpar a tela da digitação, digitando apenas seu alias “cls”. A mesma ação com outro comando pode ser somente o “clear”. A Microsoft chama de clear-host a função com alias cls ou clear.



Selecione Clear-Host e dê ←

Faça o mesmo, utilizando “**cls**” ou “**clear**” que o comando será executado
Prática:

Faça o teste com os comandos a seguir utilizando o Get-Alias “ “

cat	cd	chdir	clear	cls	copy	del	diff	dir	echo
erase	h	history	kill	lp	ls	move	popd	ps	pushd
pwd	r	ren	rm	rmdir	sleep	sort	tee	tipo	write

Por meio do comando cd, entre no diretório

users public cd.. (um diretório acima)

PS C:\Windows\system32> cd ..

cd.. (novamente, vá saindo do diretório
Windows)

Se vc der o comando dir, verá todos os diretórios e os detalhes de cada diretório com as respectivas permissões.

```
PS C:\Windows> cd ..
PS C:\> dir

Diretório: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          10/01/2020    08:14           Intel
d-----          15/09/2018    04:33          PerfLogs
d-r---          13/01/2020    18:20       Program Files
d-r---          13/01/2020    15:30  Program Files (x86)
d-----          21/01/2020    11:42          TOTVS
d-r---          10/01/2020    08:06          Users
d-----          14/01/2020    14:59         Windows
```

Entrar no diretório Users

```
PS C:\> cd Users
PS C:\Users> dir

Diretório: C:\Users

Mode                LastWriteTime         Length Name
----                -
27/01/2020   16:19 d-r---         Marise Miranda
10/01/2020   10:47         Public
```

Entrar no diretório Public

```
PS C:\Users> cd Public
PS C:\Users\Public> dir

Diretório: C:\Users\Public

Mode                LastWriteTime         Length Name
----                -
d-r--- d-r-- 10/01/2020    09:42         Documents
-      -      15/09/2018    04:33         Downloads
d-r---      15/09/2018    04:33         Music
d-r---      15/09/2018    04:33         Pictures
d-r---      15/09/2018    04:33         Videos
```

Entrar no diretório Documents

```
PS C:\Users\Public> cd Documents
PS C:\Users\Public\Documents>
```

Agora, criar um arquivo texto vazio

Vamos chamar de file.txt

```
file.txt
```

Para criar um arquivo com WPS usamos o cmdlet New -Item --- o caminho (Path) e o tipo de arquivo

```
PS C:\Users\Public\Documents>
```

```
New-Item -Path '\Users\Public\Documents\file.txt' -ItemType File
```

Ou, simplesmente

New-Item file.txt

Crie um outro arquivo chamado file1.txt

Escolha

Neste caso, você já se encontra no diretório do qual quer criar o arquivo

Verifique que foi criado um arquivo texto vazio

Crie um outro arquivo chamado file1 no mesmo diretório

```
Diretório: C:\Users\Public\Documents

Mode                LastWriteTime         Length Name
----                -
-a----          29/01/2020   17:02             19 file.txt
-a----          29/01/2020   17:02             15 file1.txt
```

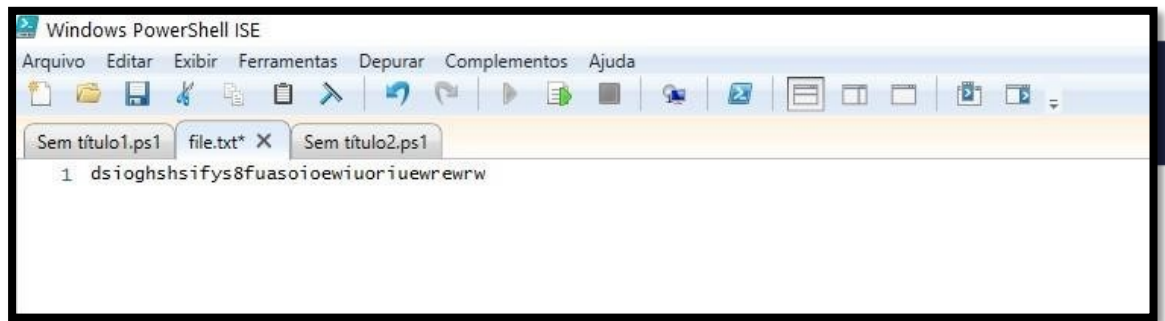
Só que você deve estar vendo o tamanho do arquivo com zero, porque eles foram criados vazio, veja abaixo.

```
Diretório: C:\Users\Public\Documents

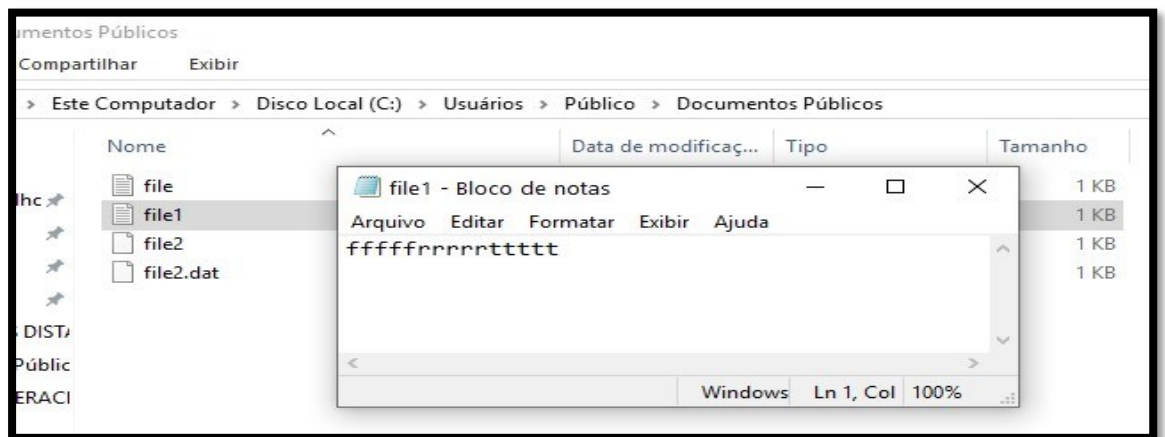
Mode                LastWriteTime         Length Name
----                -
-a----          29/01/2020   17:01             0 fil2
-a----          29/01/2020   11:42             0 file.txt
-a----          29/01/2020   16:52             0 file1.txt
-a----          29/01/2020   17:00             0 type
```

Para incluir alguma informação nos arquivos criado você pode fazer de duas maneiras:

- 1) Abrir o editor de scripts do WPSISE e abrir o diretório Public/Documents os arquivos file e file1. Digite algo e salve. Veja a tela abaixo:



- 2) De outra maneira, você poderá acessar via interface do Windows na pasta Public/ Docs e abrir com o bloco de notas ou notepad o arquivo e digitar alguma informação. Sempre salve o conteúdo do arquivo.



Estamos incluindo informação texto nos dois arquivos para que possamos concatenar os conteúdos.

Se você for no WPSISE e verificar por meio do comando dir, os tamanhos dos arquivos file e file1. Devem estar diferentes de vazio.

Agora vamos concatenar os conteúdos em um arquivo chamado file 2.txt

Estamos concatenando os conteúdos de file e file1 no arquivo texto file2

O que acontece?

Resposta: **Parece que o comando está em loop**

Vamos parar a função, com o CTRL C

Agora dê um comando dir

```
PS C:\Users\Public\Documents> dir

Diretório: C:\Users\Public\Documents

Mode                LastWriteTime         Length Name ----
-a----- 29/01/2020 17:02          19 file.txt
-a----- 29/01/2020 17:02          15 file1.txt
-a----- 29/01/2020 17:56       3065732 file2.txt
```

Veja o tamanho do arquivo file 2. Abra esse arquivo via janela de script do WPS. Veja que houve várias repetições da concatenação.

Este é um problema do WPSISE, é um problema de recursividade, que podemos resolver da seguinte forma, não atribuir txt no tipo de arquivo, pois ele se auto regeira dentro do *.txt de forma recursiva.

Para corrigir o problema, vamos fazer de modo correto, usando o mesmo nome de arquivo file 2. Mas vamos remover o arquivo file2.txt com o comando rm (remove)

```
PS C:\Users\Public\Documents> rm file2.txt
```

Dê um dir e você verá que o file2.txt não está mais no diretório.

```
PS C:\Users\Public\Documents> dir
Diretório: C:\Users\Public\Documents
Mode                LastWriteTime         Length Name ----
-a----- 29/01/2020 17:02          19 file.txt
-a----- 29/01/2020 17:02          15 file1.txt
```

Então vamos concatenar os conteúdos dos arquivos texto file e file1

Podemos fazer de duas formas, mas é preciso entender como o problema da recursividade é contornado.

```
PS C:\Users\Public\Documents> cat >file2 *.txt
```

Foi gerado um novo arquivo file2, sem tipo, com o conteúdo de file1 e file, porém com a natureza de texto.

```

PS C:\Users\Public\Documents> cat >file2 *.txt

PS C:\Users\Public\Documents> dir
Diretório:
C:\Users\Public\Documents

Mode LastWriteTime Length Name ----
-----
-a---- 29/01/2020 17:02 19 file.txt
-a---- 29/01/2020 17:02 15 file1.txt
-a---- 29/01/2020 17:58 78 file2

```

Podemos gerar a concatenação de file e file1 usando uma outra extensão para file 2, por exemplo. dat (tipo de arquivo dado - *data*)

```

PS C:\Users\Public\Documents> cat >file2.dat *.txt

```

```

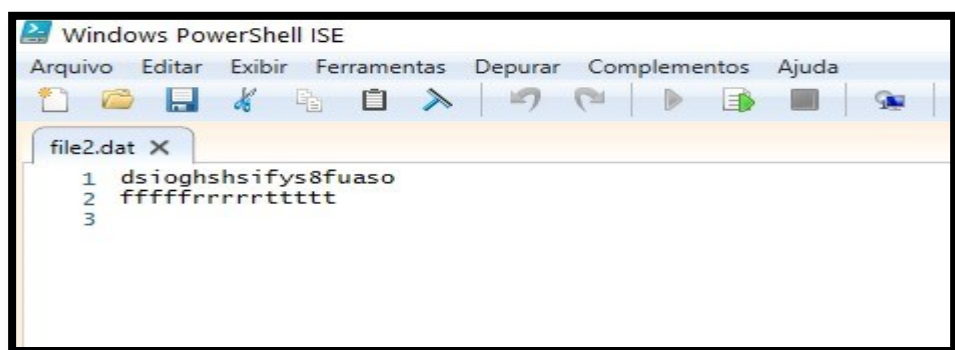
PS C:\Users\Public\Documents> dir

Diretório: C:\Users\Public\Documents

Mode LastWriteTime Length Name ----
---
-a---- 29/01/2020 17:02 19 file.txt
-a---- 29/01/2020 17:02 15 file1.txt
-a---- 29/01/2020 17:57 78 file2.dat

```

Se você abrir o file 2 no script do WPSISE verá apenas duas linhas com os conteúdos dos dois arquivos file e file 1 gravados em file2



Agora podemos nos aprofundar nos Sistemas Operacionais. Pois manipulamos arquivos e comandos sem auxílio de interfaces e janelas.

Isto nos remete a uma ideia de **camadas de abstração**. Os recursos podem ser abstraídos sem que o usuário menos especializado necessite

de entender, reconhecer e aplicar as instruções necessárias a operação do seu computador e recursos como arquivos, telas e pastas.

REFERÊNCIA BIBLIOGRÁFICA:

TANENBAUM, A. Sistema Operacionais Modernos. Tradução Jorge Ritter. 2ª Edição, São Paulo. Pearson Education do Brasil, 2009.

MACHADO, F. B. Arquitetura de Sistemas Operacionais, 4ª Ed, Rio de Janeiro. LTC, 2007.

SILBERSCHATZ, A. Sistemas Operacionais: Conceitos. 5ª Ed. São Paulo. Prentice Hall, 2000.

- REDHAT. Disponível em www.redhat.com/topics/middleware. Acessado em 19/12/2019.

- FERRARI, F. O Shell. Disponível em <http://www.ferrari.pro.br/home/documents/FFerrari-O-Shell-Unix.pdf>. Acessado em 19/12/2019.

<http://www.agasus.com.br/4-grandes-motivos-para-atualizar-hardware-e-sistemas-operacionais-da-empresa/>

DONDA, D. Windows Power Shell 3.0. Um Guia de Windows PowerShell desenvolvido especificamente para profissionais de infraestrutura. Todo o conteúdo está sob licença da Creative Commons Attribution 3.0 Unported License <http://bit.ly/ZnVDOD>. Disponível em <http://professorramos.com/Materiais/Documentos/PowerShell%20para%20IT%20Pro-%20Book.pdf>. Acessado em 19/12/2019.

LICENÇA MICROSOFT EDUCATION: Instituições de ensino credenciadas, como escolas de ensino fundamental e médio, universidades, faculdades públicas e privadas e faculdades comunitárias estaduais, poderão efetuar o download e reproduzir os Documentos para serem distribuídos em sala de aula. A distribuição fora de sala de aula exigirá permissão por escrito.