

Professor: Macêdo Firmino
Disciplina: Administração de Sistemas Proprietários
Aula 17: Introdução ao PowerShell.

Olá, meu queridos!! Tudo bem??? Na aula de hoje iremos fazer uma breve introdução ao PowerShell da Microsoft, ensinando os conceitos básicos e alguns comandos. Vamos lá!!! Preparados???

PowerShell

O PowerShell é um shell desenvolvida pela Microsoft. Ela é uma ferramenta que visa permitir automação de tarefas e gerenciamento de sistemas através de comandos e scripts em ambientes Windows. Shell é uma interface que permite aos usuários interagir com o sistema operacional e pode ser tanto no modo *Graphical User Interface* (GUI) quanto em modo *Command-Line Interface* (CLI).

O PowerShell ajuda os administradores do sistema e os usuários avançados a automatizar rapidamente as tarefas que gerenciam processos e sistemas operacionais. Ele foi projetado para melhorar o ambiente de *script* e de linha de comando eliminando problemas do antigos do *prompt* de comandos e adicionando novos recurso.

O prompt de comando usa uma linguagem de script mais simples, baseada em lotes (Batch), que é mais limitada em termos de funcionalidades e capacidades. Enquanto que o PowerShell é uma linguagem de script robusta e orientada a objetos. Ele suporta a manipulação de objetos e oferece recursos avançados de automação.

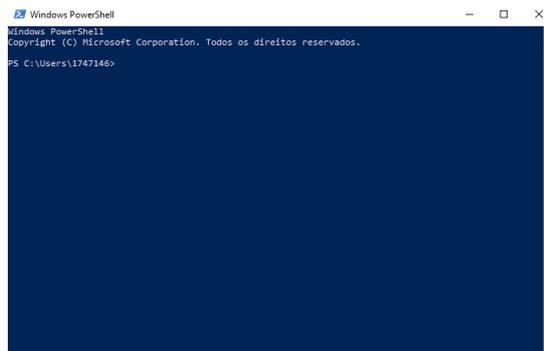
Os comandos do PowerShell (chamados de cmdlet) permitem que você gerencie os computadores da linha de comando. Ele ainda permitem o acesso a vários recursos e configurações do Windows, por exemplo, ao repositórios de dados, como o registro e o repositório de certificados, de forma tão fácil quanto acessar o sistema de arquivos.

O PowerShell é um *software livre* e código-fonte aberto para contribuições da comunidade.

Abrir o PowerShell no Windows

Para trabalharmos com o Powershell podemos utilizar o Shell diretamente (PS) ou o Powershell ISE. Para iniciar PS diretamente:

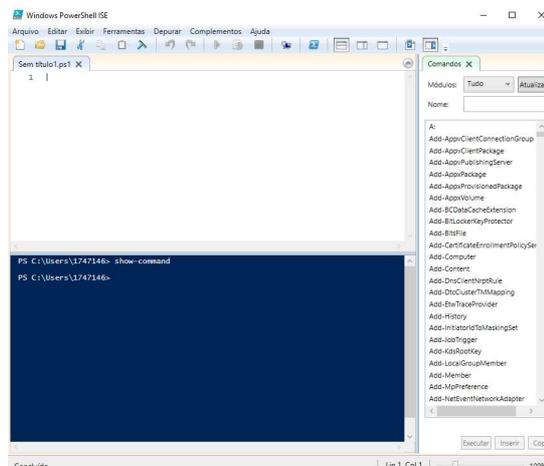
1. Clique em “Iniciar”, digite “PowerShell” e clique em “Windows PowerShell”. Ou no menu Iniciar, clique em “Todos os Programas”, “Acessórios”, clique na pasta “Windows PowerShell” e clique em “Windows PowerShell”.



Você pode optar por carregar o PowerShell ISE (*Integrated Scripting Environment*), um ambiente de programação do PowerShell que facilita o desenvolvimento de *scripts*. Nele você pode executar comandos, gravar, testar e depurar *scripts* em uma interface de usuário gráfica baseada no Windows.

Para iniciar PS ISE:

2. Clique em “Iniciar”, digite “ISE” e clique em “ISE do Windows PowerShell”. Ou no menu “Iniciar”, clique em “Todos os Programas”, “Acessórios”, clique na pasta “Windows PowerShell” e clique em “ISE do Windows PowerShell”.



O console Powershell ISE se divide em 3 painéis

- Script Painel: onde você edita seus scripts.
- Console: exibe a saída do seus comandos e onde podemos executar comandos diretamente.
- Commands: exibe uma lista dos comandos disponíveis e seus parâmetros.

Conhecendo Alguns Comandos

Os CMDLETS são como são conhecidos os comando do PowerShell. Eles não são case sensitive e usam uma convenção simples de nomes, baseada em Verbo-Substantivo. Por exemplo, o verbo **get** que deve ser seguido de um substantivo, por exemplo, **date**. Resultando em **get-date**.

É fácil de entender o que um comando do PowerShell faz lendo o nome dele. O comando para desligar um computador é Stop-Computer. O comando para listar todos os computadores em uma rede é Get-Computer. O comando para obter a data do sistema é Get-Date.

Para ter uma lista completa dos cmdlets existentes use o comando:

```
get-command -commandtype cmdlet
```

Na sequência iremos mostrar alguns comandos.

Show-Command

Na dúvida sobre os parâmetros de um determinado cmdlet, você pode carregar o painel de comandos. Ele é uma janela gráfica muito similar ao painel de comandos do PowerShell ISE e permite a visualização dos parâmetros opcionais e obrigatórios dos cmdlets. Por exemplo:

```
Show-Command Get-Date
```

Help

Caso o comando Show-Command não foi suficiente para esclarecer suas dúvidas, existe ainda comando Get-Help. O mesmo apresenta a sintaxe, parâmetros, aliases e comentários. O comando é:

```
Get-Help <cmdlet>
```

Por exemplo, digamos que eu queira saber mais sobre o comando “Test-Connection” usamos:

```
Get-Help Test-Connection
```

Alias

O PowerShell dá suporte a aliases para se referir aos comandos por 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. Por exemplo, o PowerShell tem uma função interna denominada Clear-Host que limpa a janela de saída. Você pode digitar o alias cls ou clear em um prompt de comando. O PowerShell interpreta esses aliases e executa a função Clear-Host.

Para listar todos os Alias utilize o seguinte comando:

```
get-command -commandtype alias
```

Através dos alias podemos utilizar comandos do Linux e comandos do prompt de comandos do Windows, por exemplo: cd , cat, cp, dir, echo, ls, mount, man, kill, move, mv, ps, rm, entre outros.

Coletando Informações

Agora iremos demonstrar como podemos coletar informações de um computador. É possível especificar um nome ou endereço IP associado a qualquer computador que você pode acessar.

Vamos começar com um comando que coleta informações sobre o BIOS do sistema:

```
Get-CimInstance -ClassName Win32_BIOS
```

As informações de modelo do computador também estão disponíveis usando o comando:

```
Get-CimInstance -ClassName  
Win32_ComputerSystem
```

Por último, para exibir o status de todos os serviços no computador você poderá usar o comando:

```
Get-CimInstance -ClassName  
Win32_Service
```

Trabalhando com Arquivos e Pastas

Com o PowerShell você pode navegar pelas unidades do Windows e manipular os arquivos e pastas. Para listar todos os arquivos e pastas dentro de uma pasta, similar ao uso do comando DIR do Cmd.exe ou do ls em um shell do UNIX, podemos usar o comando:

```
Get-ChildItem -Path Desktop
```

Como ls é um alias para Get-ChildItem, podemos utilizar

```
ls Desktop
```

Utilizamos o parâmetro opcional "Force" para exibir itens ocultos ou do sistema.

Para modificar a sua localização poderá utilizar o comando Set-Location e a pasta de destino. Por exemplo, para entrar na pasta Music, podemos utilizar:

```
Set-Location Music
```

Para copiar arquivos e pastas utilizamos o comando Copy-Item. Por exemplo, o comando a seguir faz backup de C:\boot.ini em C:\boot.bak

```
Copy-Item -Path C:\boot.ini  
-Destination C:\boot.bak
```

Se o arquivo de destino já existir, a tentativa de cópia falhará. Para substituir um destino pré-existente, use o parâmetro Force:

```
Copy-Item -Path C:\boot.ini  
-Destination C:\boot.bak -Force
```

Copiar a pasta funciona da mesma maneira. Este comando copia a pasta c:\temp\test1 na nova pasta c:\temp\DeleteMe recursivamente:

```
Copy-Item C:\temp\test1  
-Recurse C:\temp\DeleteMe
```

Você também pode copiar uma seleção de itens. Por exemplo, o comando a seguir copia todos os arquivos .txt contidos em qualquer lugar em c:\data em c:\temp\text:

```
Copy-Item -Filter *.txt -Path c:\data  
-Recurse -Destination C:\temp\text
```

Para criar uma nova pasta em C:\temp\New, utilize o comando:

```
New-Item -Path 'C:\temp\New'  
-ItemType Directory
```

Para criar um novo arquivo vazio em C:\temp\New\file.txt

```
New-Item -Path 'C:\temp\New\file.txt'  
-ItemType File
```

É possível remover os arquivos e pastas usando o comando Remove-Item. Será solicitando uma confirmação. Por exemplo, se você tentar excluir a pasta C:\temp\DeleteMe:

```
Remove-Item -Path C:\temp\DeleteMe
```

Gerenciando a Rede

Se você desejar obter as configurações de rede atuais dos adaptadores, use o seguinte comando:

```
Get-NetIPConfiguration
```

Para executar um ping simples em um computador poderá usar o comando abaixo:

```
Test-NetConnection portal.ifrn.edu.br
```

O comando para listar os adaptadores de rede presentes no computador, com seus nomes, descrição, status, endereços MAC e taxa de transmissão é:

```
Get-NetAdapter
```

Para fazer uma resolução de domínio e obter informações sobre os servidores DNS associados a um host (similar a nslookup):

```
Resolve-DNSName portal.ifrn.edu.br
```

Para traçar a rota até um determinado host remoto (tracert), utilize o comando:

```
Test-NetConnection portal.ifrn.edu.br  
-TraceRoute
```

O comando utilizado para visualizar as conexões de rede TCP/IP atuais da máquina (similar ao netstat):

```
Get-NetTCPConnection
```

O comando que permite alterar o endereço IP da placa de rede é New-NetIPAddress. Veja o exemplo a seguir:

```
New-NetIPAddress 192.168.1.35  
-InterfaceAlias Ethernet  
-DefaultGateway 192.168.1.1  
-AddressFamily IPV4  
-PrefixLength 24
```

O comando que permite alterar o endereço do Servidor DNS.

```
Set-DnsClientServerAddress  
-InterfaceIndex <interface>  
-ServerAddresses <IP>
```

Se quisermos alterar de IP Fixo para o IP Dinâmico em uma determinada interface de rede, utilizaremos o comando:

```
Set-NetIPInterface  
-InterfaceAlias <interface>  
-Dhcp enabled
```

Por último, se quisermos que o endereço do Servidor DNS fique automático, podemos utilizar o comando:

```
Set-DnsClientServerAddress  
-InterfaceAlias <interface>  
-ResetServerAddresses.
```

Gerenciando Servidores

No Windows Server, o console Gerenciador do Servidor e os cmdlets Windows PowerShell para Gerenciador do Servidor permitem a instalação de funções e recursos em servidores locais ou remotos.

O PowerShell possui nativamente muitos módulos com centenas de cmdlets que podem nos ajudar nestas tarefas diárias de administração. Digite `Get-WindowsFeature` e pressione `Enter` para exibir uma lista de funções e recursos disponíveis e instalados no servidor local.

`Get-WindowsFeature`

Digite `Get-Help install-WindowsFeature` e pressione `Enter` para exibir a sintaxe e os parâmetros aceitos para o cmdlet.

`Get-Help install-WindowsFeature`

O comando `Install-WindowsFeature` é utilizado para instalar as funções e serviços. Por exemplo, para instalar o servidor DNS.

`Install-WindowsFeature DNS`
`-IncludeManagementTools`

No Windows PowerShell, ao contrário do assistente para adicionar funções e recursos, as ferramentas de gerenciamento para uma função não são incluídos por padrão. Para incluir ferramentas de gerenciamento como parte da instalação de uma função, adicione o parâmetro `IncludeManagementTools` ao cmdlet.

Para instalar o DHCP, abra Windows PowerShell privilégios de Administrador e executando o comando a seguir.

`Install-WindowsFeature DHCP`
`-IncludeManagementTools`

Outros Comandos

O comando `Get-Process` retorna informações sobre processos em execução no sistema.

`Get-Process`

O comando `Get-Service` obtém informações sobre os serviços do sistema:

`Get-Service`

Para iniciar, parar e reiniciar um serviço específico podemos utilizar o comando `Start-Service`, `Stop-Service` e `Restart-Service`, respectivamente.

`Start-Service -Name NOME_DO_SERVICO`

`Stop-Service -Name NOME_DO_SERVICO`

`Restart-Service -Name NOME_DO_SERVICO`

Atividades

De acordo com os conhecimentos adquiridos na aula de hoje responda os seguintes questionamentos:

1. Abra o PowerShell e conheça alguns comandos usando o seu Help.
2. Colete informações sobre a BIOS, processador e Sistema Operacional.
3. Determine a data e hora do sistema.
4. Determine a sua pasta corrente e liste todos os arquivos que ela contém.
5. Crie uma pasta no diretório corrente chamada de LADIR. Depois crie um arquivo de texto nessa pasta chamada de teste.txt.
6. Cópie o arquivo teste.txt para a sua Área de Trabalho.
7. Obtenha as configurações de rede atuais dos seus adaptadores.
8. Teste a resolução de nome para o domínio `www.ifrn.edu.br`.
9. Teste sua conexão fazendo um ping para o endereço `www.ifrn.edu.br`.
10. Teste sua conexão fazendo um TraceRoute para o endereço `www.ifrn.edu.br`.