



Menu ▾

 Cadastro  Login

CMDLETS: Gerenciando Serviços com PowerShell no Windows 10

AGRADEÇA AO AUTOR

COMPARTILHE!

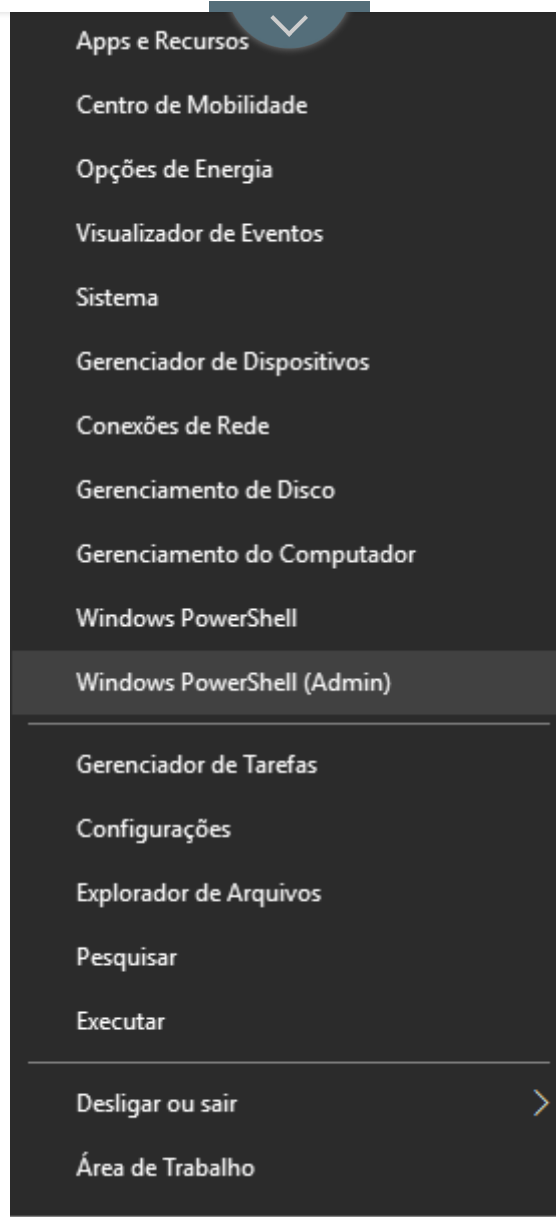


Opa, pessoal!

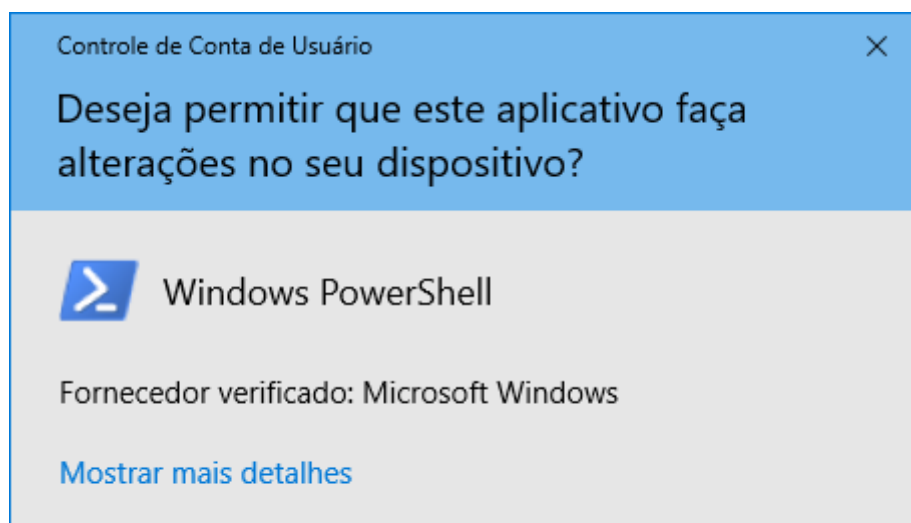
Continuando com os tutoriais sobre CMDLETS do PowerShell para gerenciamento do sistema, estarei demonstrando aqui como gerenciar os serviços do Windows 10.

Primeiramente iremos executar o PowerShell com permissões elevadas para que o cmdlet seja executado com êxito.

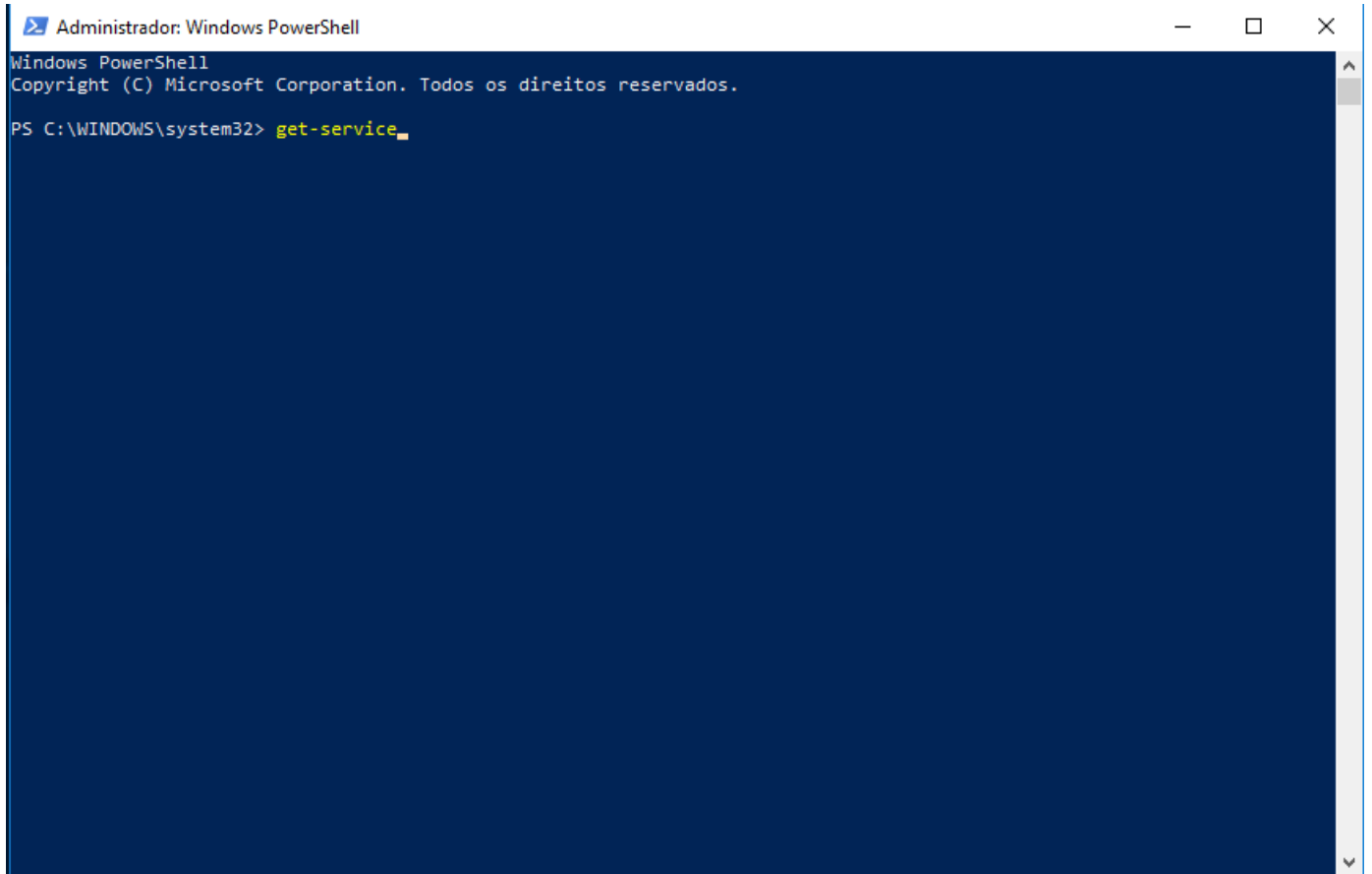
Para isso, iremos clicar com o botão direito do mouse na logo do menu do Windows, situado no canto inferior esquerdo da área de trabalho e depois clicar com o botão esquerdo em “Windows PowerShell (Admin)”.



Como estamos querendo executar com permissões administrativas, irá aparecer a janela de Controle de Conta de Usuário onde iremos clicar em “Sim” para permitir que o Windows PowerShell faça alterações no sistema.



O comando para listar todos os serviços que estão configurados no sistema é "get-service", o qual iremos digitar e em seguida pressionar a tecla <enter>.

A screenshot of a Windows PowerShell window titled "Administrador: Windows PowerShell". The window has a dark blue background. The text inside shows the PowerShell prompt "PS C:\WINDOWS\system32>" followed by the command "get-service" which is highlighted in green. Above the prompt, the text "Windows PowerShell" and "Copyright (C) Microsoft Corporation. Todos os direitos reservados." are visible. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

PS C:\WINDOWS\system32> get-service
```

Reparem que aparecerá a lista de todos os serviços configurados no Windows e que podem estar sendo executados ou, no momento, estarem parados.

```
Administrador: Windows PowerShell

Running WdiServiceHost Host do Serviço de Diagnóstico
Stopped WdiSystemHost Host do Sistema de Diagnósticos
Running WdNisSvc Serviço de Inspeção de Rede do Wind...
Stopped WebClient Cliente da Web
Stopped Wecsvc Coletor de Eventos do Windows
Stopped WEPHOSTSVC Serviço de Host do Provedor de Crip...
Stopped wercplsupport Suporte do Pannel de Controle Relat...
Stopped WerSvc Serviço de Relatórios de Erro do Wi...
Stopped WFDSCongMgrSvc Serviço do Gerenciador de Conexões ...
Stopped WiaRpc Eventos de Aquisição de Imagens Est...
Running WinDefend Serviço Windows Defender Antivirus
Running WinHttpAutoProx... Serviço de Descoberta Automática de...
Running Winmgmt Testador de instrumentação de geren...
Stopped WinRM Windows Remote Management (WS-Manag...
Stopped wisvc Serviço do Participante do Programa...
Stopped WlanSvc Configuração Automática de WLAN
Stopped wldsvc Assistente de Conexão de Conta da M...
Stopped wlpasvc Serviço Assistente de Perfil Local
Stopped wmiApSrv Adaptador de Desempenho WMI
Stopped WMPNetworkSvc Serviço de Compartilhamento de Rede...
Stopped workfolderssvc Pastas de Trabalho
Stopped WpcMonSvc Controles dos País
Stopped WPDBusEnum Serviço Enumerador de Dispositivos ...
Running WpnService Serviço do Sistema de Notificações ...
Running WpnUserService_... WpnUserService_1f66f
Running wscsvc Central de Segurança
Running WSearch Windows Search
Running wuauserv Windows Update
Stopped WwanSvc Configuração Automática de WWAN
Stopped xbgm Xbox Game Monitoring
Stopped XblAuthManager Gerenciador de Autenticação Xbox Live
Stopped XblGameSave Salvar Jogos no Xbox Live
Stopped XboxGipSvc Xbox Accessory Management Service
Stopped XboxNetApiSvc Serviço de Rede Xbox Live

PS C:\WINDOWS\system32>
```

Supondo, como exemplo, que temos um serviço em específico que queremos deixá-lo parado. Neste meu caso, é o SQLEXPRESS.

Para para o serviços, preciso saber o nome correto deste processo. Então irei filtrar digitando o seguinte comando: "get-service *sql*" e em seguida pressionarei a tecla <enter>. Reparem que, para pesquisar por um trecho do nome, colocamos este pedaço entre asteriscos (*).

```
Administrador: Windows PowerShell

Running wscsvc Central de Segurança
Running WSearch Windows Search
Running wuauclt Windows Update
Stopped WwanSvc Configuração Automática de WWAN
Stopped xbgm Xbox Game Monitoring
Stopped XblAuthManager Gerenciador de Autenticação Xbox Live
Stopped XblGameSave Salvar Jogos no Xbox Live
Stopped XboxGipSvc Xbox Accessory Management Service
Stopped XboxNetApiSvc Serviço de Rede Xbox Live

PS C:\WINDOWS\system32> get-service *sql*

Status Name DisplayName
-----
Running MSSQL$SQLEXPRESS SQL Server (SQLEXPRESS)
Stopped SQLAgent$SQLEXP... SQL Server Agent (SQLEXPRESS)
Stopped SQLBrowser SQL Server Browser
Running SQLTELEMETRY$SQ... SQL Server CEIP service (SQLEXPRESS)
Running SQLWriter Gravador VSS do SQL Server

PS C:\WINDOWS\system32>
```

Como resultado, foram filtrados somente os serviços que tem SQL no nome e que estão executados ou parados.

Se quiser ainda filtrar os serviços SQL e mostrar só os que estão sendo executados, iremos utilizar o comando “get-service *sql* | where { \$_.status -eq ‘running’ }”, seguido de pressionar a tecla <enter>. O pipeline (|) serve para concatenar comandos.

```
Administrador: Windows PowerShell

Running wscsvc Central de Segurança
Running WSearch Windows Search
Running wuauserv Windows Update
Stopped WwanSvc Configuração Automática de WWAN
Stopped xbgm Xbox Game Monitoring
Stopped XblAuthManager Gerenciador de Autenticação Xbox Live
Stopped XblGameSave Salvar Jogos no Xbox Live
Stopped XboxGipSvc Xbox Accessory Management Service
Stopped XboxNetApiSvc Serviço de Rede Xbox Live

PS C:\WINDOWS\system32> get-service *sql*

Status Name DisplayName
-----
Running MSSQL$SQLEXPRESS SQL Server (SQLEXPRESS)
Stopped SQLAgent$SQLEXP... SQL Server Agent (SQLEXPRESS)
Stopped SQLBrowser SQL Server Browser
Running SQLTELEMETRY$SQ... SQL Server CEIP service (SQLEXPRESS)
Running SQLWriter Gravador VSS do SQL Server

PS C:\WINDOWS\system32> get-service *sql* | where { $_.status -eq 'running' }

Status Name DisplayName
-----
Running MSSQL$SQLEXPRESS SQL Server (SQLEXPRESS)
Running SQLTELEMETRY$SQ... SQL Server CEIP service (SQLEXPRESS)
Running SQLWriter Gravador VSS do SQL Server

PS C:\WINDOWS\system32>
```

Como queremos parar o primeiro serviço listado, que é o MSSQL\$SQLEXPRESS, iremos digitar o comando "stop-service 'MSSQL\$SQLEXPRESS'" seguido de pressionar a tecla <enter>. Como neste caso temos um caractere especial no nome, que é o cifrão (\$), teremos que colocá-lo entre apóstrofe. Se quiséssemos parar o SQLWriter, poderíamos digitar o comando "get-service SQLWriter " <enter>.

Depois que o comando foi executado com êxito, retornará ao prompt do PowerShell como é mostrada na imagem abaixo.

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> get-service *sql* | where { $_.status -eq 'running' }

Status      Name                DisplayName
-----
Running     MSSQL$SQLEXPRESS    SQL Server (SQLEXPRESS)
Running     SQLTELEMETRY$SQ...  SQL Server CEIP service (SQLEXPRESS)
Running     SQLWriter           Gravador VSS do SQL Server

PS C:\WINDOWS\system32> stop-service 'MSSQL$SQLEXPRESS'
PS C:\WINDOWS\system32>
```

Se, por exemplo, quisermos saber quais serviços SQL estão parados, digitaremos o comando “get-service *sql* | where { \$_.status -eq 'stopped' }” e ao final do comando sempre pressionando a tecla <enter>.

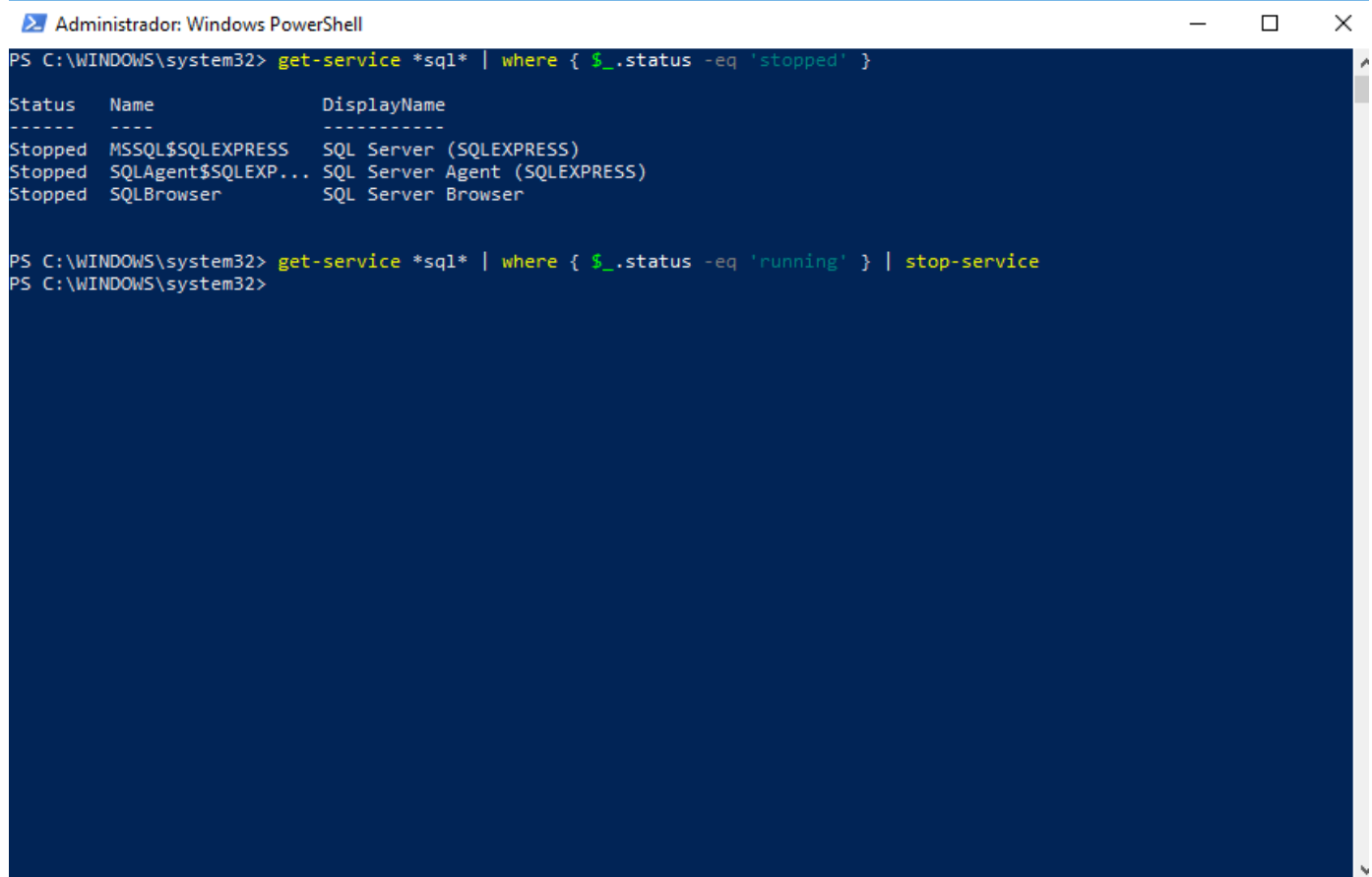
```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> get-service *sql* | where { $_.status -eq 'stopped' }

Status      Name                DisplayName
-----
Stopped     MSSQL$SQLEXPRESS    SQL Server (SQLEXPRESS)
Stopped     SQLAgent$SQLEXP...  SQL Server Agent (SQLEXPRESS)
Stopped     SQLBrowser          SQL Server Browser

PS C:\WINDOWS\system32>
```

Se quisermos filtrar um serviço em específico que está sendo executado e na mesma linha parar este serviço, usaremos o pipeline (|) para concatenar estes comandos. Com o SQL como exemplo ficaria da seguinte forma: “get-service *sql* | where { \$_.status -eq 'running' } | stop-service” seguido de pressionar <enter>.

Os comando sendo executados com êxito, voltará para o prompt de comando.



```
PS C:\WINDOWS\system32> get-service *sql* | where { $_.status -eq 'stopped' }

Status      Name                DisplayName
-----
Stopped     MSSQL$SQLEXPRESS    SQL Server (SQLEXPRESS)
Stopped     SQLAgent$SQLEXP...  SQL Server Agent (SQLEXPRESS)
Stopped     SQLBrowser          SQL Server Browser

PS C:\WINDOWS\system32> get-service *sql* | where { $_.status -eq 'running' } | stop-service
PS C:\WINDOWS\system32>
```

Se quisermos iniciar o serviço do SQL novamente, digitaremos o comando “start-service ‘MSSQL\$SQLEXPRESS’” seguido de pressionar a tecla <enter>. Iremos notar que aparecerá a mensagem que o serviço solicitado está sendo iniciado.


```
Administrador: Windows PowerShell

PS C:\WINDOWS\system32> get-service *sql*

Status      Name                DisplayName
-----
Stopped    MSSQL$SQLEXPRESS    SQL Server (SQLEXPRESS)
Stopped    SQLAgent$SQLEXP...  SQL Server Agent (SQLEXPRESS)
Stopped    SQLBrowser          SQL Server Browser
Stopped    SQLTELEMETRY$SQ...  SQL Server CEIP service (SQLEXPRESS)
Stopped    SQLWriter           Gravador VSS do SQL Server

PS C:\WINDOWS\system32> get-service sqlwriter | select-object *

Name                : sqlwriter
RequiredServices    : {}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : False
DisplayName          : Gravador VSS do SQL Server
DependentServices   : {}
MachineName         : .
ServiceName          : sqlwriter
ServicesDependedOn  : {}
ServiceHandle        : SafeServiceHandle
Status              : Stopped
ServiceType          : Win32OwnProcess
StartType            : Automatic
Site                :
Container           :

PS C:\WINDOWS\system32> start-service 'MSSQL$SQLEXPRESS'
AVISO: Aguardando que o serviço 'SQL Server (SQLEXPRESS) (MSSQL$SQLEXPRESS)' seja iniciado...
AVISO: Aguardando que o serviço 'SQL Server (SQLEXPRESS) (MSSQL$SQLEXPRESS)' seja iniciado...
PS C:\WINDOWS\system32>
```

Se quisermos saber maiores informações sobre um serviço em específico, que neste exemplo é o SQLWriter, iremos digitar o seguinte comando: “get-service sqlwriter | select-object *” seguido de <enter>. A instrução após o pipeline trará maiores informações sobre o serviço solicitado.

```
Administrador: Windows PowerShell

PS C:\WINDOWS\system32> get-service *sql*

Status      Name                DisplayName
-----
Stopped    MSSQL$SQLEXPRESS    SQL Server (SQLEXPRESS)
Stopped    SQLAgent$SQLEXP...  SQL Server Agent (SQLEXPRESS)
Stopped    SQLBrowser          SQL Server Browser
Stopped    SQLTELEMETRY$SQ...  SQL Server CEIP service (SQLEXPRESS)
Stopped    SQLWriter           Gravador VSS do SQL Server

PS C:\WINDOWS\system32> get-service sqlwriter | select-object *

Name                : sqlwriter
RequiredServices    : {}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : False
DisplayName          : Gravador VSS do SQL Server
DependentServices   : {}
MachineName         : .
ServiceName          : sqlwriter
ServicesDependedOn  : {}
ServiceHandle        : SafeServiceHandle
Status              : Stopped
ServiceType          : Win32OwnProcess
StartType            : Automatic
Site                :
Container           :

PS C:\WINDOWS\system32>
```

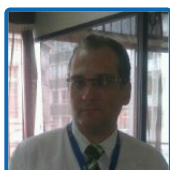
Acho que é isso pessoal.

Nos próximos artigos trarei outros comandos bastante utilizados no PowerShell.

Espero que seja útil para quem curte (ou necessita) gerenciar via linha de comando. Até a próxima! 😊

AGRADEÇA AO AUTOR

COMPARTILHE!



Luciano Gusso



[Mais artigos deste autor »](#)

CONTEÚDOS RELACIONADOS

Anúncio

A melhor escola
de TI do país

Saiba Mais



Simplificando a Cloud para iniciantes e leigos

profissionaisti.com.br



71,5% das violações de segurança na AWS...

profissionaisti.com.br



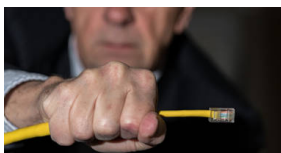
Blockchain: a tecnologia que está mudando o...

profissionaisti.com.br



O que é o Kubernetes e sua importância

profissionaisti.com.br



A Síndrome de Estocolmo no mundo corporativo

profissionaisti.com.br



Gestão de Indicadores/Métricas...

profissionaisti.com.br



Cursos de TI GRATUITOS do ITA

profissionaisti.com.br



Serverless: entenda como e porque aprove

profissionaisti.com.br



IBM adquire Reo US\$ 34 bilhões:

profissionaisti.com.br

Anúncio

1. Serviços de atendimento
2. Cmdlet powershell
3. Exportação serviços

1 COMENTÁRIOS

Vinicius Giacomini

6 de dezembro de 2018 às 9:35

1

Bom dia!

Excelente artigo técnico! Bastante detalhado e com boa didática. Muito obrigado pelas informações, são úteis para nosso aprendizado em PowerShell.



MAIS DE 100 JOGOS POR UM BAIXO C
Lançamentos exclusivos de XBI

