****

**Objetivos**

Nos exercícios a seguir, vamos executar dois scripts do PowerShell. Esses mesmos problemas já foram resolvidos no Bash, o objetivo é entender como podemos usar uma ferramenta diferente para resolver o mesmo problema.

**O que iremos fazer?**

Primeiro um script simples de seleção de nomes, depois um mais avançado de backup de arquivos.

**Instruções**

**Exercício 1**

Individualmente, fazemos um script no Power Shell que deve implementar o seguinte fluxo:



Para conseguir isso, devemos executar as seguintes etapas:

1. Primeiro crie um arquivo, podemos usar o Vscode, Bloco de Notas ou Notepad++ e inserir uma lista de 20 nomes. Finalmente, salvamos o arquivo com o nome ‘lista\_nomes.txt’.
2. Agora vamos criar o scriptnomes.ps1  
   Nota: a extensão padrão para os scripts em PowerShell é o **.ps1**

| **Nota:** Antes de começar a escrever o script, podemos experimentar o comando **‘foreach’**, fazendo um script menor que contenha as seguintes linhas:   | foreach($nome in $(Get-Content -Path .\lista\_nomes.txt)) {  Write-Output "O nome é $nome"  } | | --- |   Como resultado, devemos percorrer o arquivo e mostrar os nomes linha por linha, colocando a mensagem "O nome é" antes de cada item do arquivo. |
| --- | --- |

1. Sabendo disso, agora o que vamos fazer é adicionar a lógica de controle, para isso devemos usar uma instrução **'if-else'**, onde vamos comparar o valor da variável '$nome’ com o texto "Pedro" , essa parte deve ser codificada da seguinte forma:

| if ($nome -eq "Pedro") {  Write-Output "Encontrou $nome"  } else {  $outrosNomes++  } |
| --- |

Com essas duas partes, procedemos à conclusão do exercício, não esquecendo de declarar a variável 'outrosNomes' no script final.

4) Agora executamos o comando para testar o script

| powershell -executionpolicy bypass -File .\scriptnomes.ps1 |
| --- |

**Exercício 2**

Backup simples de arquivos

Na pasta do seu usuário crie duas subpastas **teste** e **backup**

Para fazer isso podemos usar o Powershell, **entrar na sua pasta de usuário** e digitar os comandos:

| mkdir teste  mkdir backup |
| --- |

**Nota: Não sabe o seu usuário no windows ? digite o seguinte comando no powershell:**

| Get-WMIObject -class Win32\_ComputerSystem | select username |
| --- |

1. Vamos criar um arquivo no vscode ou notepad++ com o nome scriptbackup.ps1

**Nota: Não se esqueça de alterar o “Emp1” dos comandos para o seu nome de usuário do windows.**

1. Vamos definir a pasta de origem - o caractere "\*" significa que iremos pegar tudo, arquivos e pastas. **$sourcePath = "C:\Users\Emp1\Documents\teste\\*"**
2. Agora definiremos a pasta de destino do Backup. **$destinyPath = "C:\Users\Emp1\Documents\backup\"**
3. Vamos pegar a data atual no formato dia-mês-ano para fazer o arquiv de log **$date = Get-Date -Format d-M-yyy**
4. Por fim fazemos o backup copiando os arquivos da origem para o destino - o parâmetro "Recurse" diz que vai copiar todas as subpastas - O parâmetro "Verbose" diz que as ações serão mostradas. Para gravar as ações realizadas em um arquivo de log, vamos direcionar a saída do comando para o arquivo que possui um nome composto por "log" mais a data

**Copy-Item $sourcePath $destinyPath -Recurse -Verbose > "$destinyPath\log-$date.txt"**

O arquivo deverá ficar da seguinte forma **(lembrando que o “Emp1” deve ser alterado para o seu nome de usuário do windows)**:

| #define a pasta de origem - o caractere "\*" significa que iremos pegar tudo, arquivos e pastas.  $sourcePath = "C:\Users\Emp1\Documents\teste\\*"    #define a pasta do drive de destino.  $destinyPath = "C:\Users\Emp1\Documents\backup\"    #pega a data atual no formato dia-mês-ano para fazer o arquivo de log  $date = Get-Date -Format d-M-yyy    #Copia da origem para o destino - o parâmetro "Recurse" diz que vai copiar todas as subpastas  #O parâmetro "Verbose" diz que as ações serão mostradas  #para gravar as ações realizadas em um arquivo de log, vamos direcionar a saída do comando para o arquivo que possui um nome composto por "log" mais a data  Copy-Item $sourcePath $destinyPath -Recurse -Verbose > "$destinyPath\log-$date.txt" |
| --- |

Salve e feche o arquivo.

Agora execute o seguinte comando no Powershell dentro da sua pasta de usuário:

| powershell -executionpolicy bypass -File .\scriptbackup.ps1 |
| --- |