

acqui comeca o menu principal

```
:menuprincipal
echo.
echo -----Programa-----
echo.
echo M - Manual de uso
echo C - Verificar changelog
echo.
echo |
echo 1 - Selecionar Unidade
echo 2 - Listar Diretorio
echo 3 - Listar Diretorio de Arquivos
echo 4 - Ir para Diretorio Mae
echo 5 - Subir um Diretorio
echo 6 - Abrir Diretorio Especifico
echo.
echo -----Copia-----
echo.
echo 7 - Realizar Copia de Arquivos
echo.
SET /p menu=0 que deseja fazer?
```

```
if %menu% EQU M (goto manual)
if %menu% EQU m (goto manual)
if %menu% EQU C (goto changelog)
if %menu% EQU c (goto changelog)
if %menu% EQU 1 (goto unidade)
if %menu% EQU 2 (goto dirlist)
if %menu% EQU 3 (goto dirarv)
if %menu% EQU 4 (goto dirmae)
if %menu% EQU 5 (goto dirup)
if %menu% EQU 6 (goto opendir)
if %menu% EQU 7 (goto menucopia) else (goto menuprincipal)
```

```
:manual
echo.
echo Esse programa serve para copiar arquivos de determinado tipo entre quaisquer diretorios.
echo Voce pode usar ele para, por exemplo, copiar todas as fotos de um HD sem nenhum esforco bruto.
echo Para seu bom funcionamento, e importante inserir os diretorios de forma correta (igual mostrado
echo no CMD).
```

```
:opendir
echo.
set /p dir=Qual diretorio deseja abrir?
echo.
cd %dir%
dir
pause
goto menuprincipal
```

:: aqui comeca o menu de copia

```
:menucopia
echo.
SET /p origem=Qual diretorio deseja copiar (insira o diretorio completo)?
SET /p destino=Para qual diretorio ou unidade (insira o diretorio completo)?
SET /p sub=Deseja adicionar outros parametros? Se nao, deixe com um espaco.
echo.
echo 1 - Tudo
echo 2 - Imagens (png,jpg,jpeg,gif,ico,svg,bmp)
echo 3 - Videos (mp4,avi,mkv,wmv,vma,mpg,mpeg,asf)
echo 4 - Musicas (mp3,wav,flac,aac)
echo 5 - Textos (txt,docx,pdf,doc,docm)
echo 6 - Office (doc,docx,docm,xlsx,xlsm,xltx,pptx,ppsx,potx,accdb,mdb)
echo 6.1 - Word (doc,docx,docm)
echo 6.2 - Excel (xlsx,xlsm,xltx)
echo 6.3 - Power Point (pptx,ppsx,potx)
echo 6.4 - Access (accdb,mdb)
echo 7 - Web (html,htm,php,js,aspx,css,cpp)
echo 8 - Design (psd,indd,pdf,svg,cdr,ai,aep,aepx,ppj)
echo 9 - Sistema (dll,reg,jar)
echo 10 - Compactados (zip,rar)
echo 11 - Outro (inserir)
echo.
SET /p proc=0 que deseja copiar?
```

```
if %proc% EQU 1 (goto all)
if %proc% EQU 2 (goto imagem)
if %proc% EQU 3 (goto video)
if %proc% EQU 4 (goto musica)
if %proc% EQU 5 (goto texto)
if %proc% EQU 6 (goto office)
if %proc% EQU 6.1 (goto word)
if %proc% EQU 6.2 (goto excel)
if %proc% EQU 6.3 (goto ppt)
if %proc% EQU 6.4 (goto access)
if %proc% EQU 7 (goto web)
if %proc% EQU 8 (goto design)
if %proc% EQU 9 (goto sistema)
if %proc% EQU 10 (goto compactados)
if %proc% EQU 11 (goto outro)
if %proc% EQU 0 (goto changelog)
:changelog
echo.
echo Nao existem mudancas, afinal essa e a versao 1.0.
echo.
pause
goto menuprincipal
```

```
set /p unidade=Qual unidade deseja abrir?
echo.
%unidade%
echo.
echo Trocado para unidade %unidade%.
goto menuprincipal
```

# PROGRAMAÇÃO BÁSICA

NATANAEL ANTONIOLI

# 4

# LINGUAGEM BATCH



FÁBRICA DE  
NOOBS

# Programação Básica – Linguagem Batch

Por: Fábrica de Noobs

## *Índice*

- 1) Prefácio
- 2) Introdução
- 3) Programas Recomendados
- 4) Navegação em Shell
  - a. dir
  - b. cd
- 5) Atributos
  - a. help
- 6) Exibição de Mensagens
  - a. echo
  - b. pause
  - c. cls
  - d. Comentários
- 7) Etiquetas e Menus
  - a. goto
  - b. choice
- 8) Comandos Diversos
- 9) Variáveis
  - a. set
  - b. Operadores
- 10) Tomadas de Decisão
  - a. if
  - b. else
  - c. Comparadores
- 11) Conclusão

## 1) Prefácio

Enfim começamos mais um curso. Dessa vez, o tema será linguagem Batch. É uma linguagem pouco explorada, mas que está presente em qualquer edição no Windows.

Sendo assim, a linguagem Batch pode ser útil para qualquer um que deseja trabalhar melhor com o sistema operacional. Utilizando ela, pode-se criar rotinas automatizadas, criar ferramentas que facilitam o uso do Prompt de Comando e até trojans.

## 2) Introdução

Talvez, mesmo sem intenção, você já tenha entrado em contato com o Prompt de Comando (ou CMD) do Windows. Ele funciona a partir de uma sintaxe, onde pode-se inserir comandos para executar determinadas tarefas.

Outra opção, ao invés de digitar diretamente no CMD, é criar arquivos com a extensão .bat, de forma que ao serem executados, eles rodam os comandos inseridos diretamente no Prompt.

Além dos comandos básicos, aqueles destinados a realizar tarefas do Windows, temos também vários outros elementos (como variáveis) que podem ser inseridos num arquivo bat, caracterizando assim a Linguagem Batch.

Apesar de não ser muito eficiente no quesito visual, pode ser uma ferramenta poderosa para automatizar processos do Windows – ou prejudica-los. Se você se interessa por criação de trojans, é um bom começo.

## 3) Programas Recomendados

Primordialmente, você precisará de uma versão do Windows para executar arquivos em Batch. Já para sua criação, qualquer editor de texto (como o Bloco de Notas) é válido, mas recomendo o uso de uma IDE.

- **Batch Compiler:** um dos melhores programas do gênero, uma vez que não insere nenhum tipo de marca d'água no produto final. Pode ser baixado em <http://sourceforge.net/projects/batchcompiler/>.
- **Batch to Exe Converter:** caso crie um arquivo em batch e queira transformá-lo em executável (isso pode ser útil para evitar que outros bisbilhotem seu código), é uma opção interessante, uma vez que oferece vários recursos adicionais como o uso de ícones. Pode ser baixado em <http://www.f2ko.de/en/b2e.php>.

## 4) Navegação em Shell

Por ser a linguagem de programação utilizada no CMD, é natural que muitas de suas operações envolvam arquivos e diretórios de pastas (todas as pastas presentes no computador). Sendo assim, é importante que primeiro possamos aprender a manejar tais diretórios. Isso é chamado de navegação em shell.

Aqui, usaremos dois comandos principais. O `dir`, usado para listar todos os arquivos de um diretório, e o `cd`, usado para abri-lo.

Primeiro, inicie o CMD. O que você encontrar será mais ou menos assim :

```
Microsoft Windows [versão 10.0.10240]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Natanael>
```

Digite `dir`, tecele `enter` e observe o que acontece.

```
C:\Users\Natanael>dir
O volume na unidade C não tem nome.
O Número de Série do Volume é 60EE-F599

Pasta de C:\Users\Natanael

27/12/2015  16:58    <DIR>          .
27/12/2015  16:58    <DIR>          ..
26/12/2015  19:42    <DIR>          .oracle_jre_usage
22/12/2015  15:01    <DIR>          .sqlmap
23/12/2015  12:42    <DIR>          .VirtualBox
17/12/2015  17:43    <DIR>          Contacts
16/12/2015  23:52           358.932 Contatos.accdt
27/12/2015  12:31    <DIR>          Creative Cloud Files
16/12/2015  23:05           4.120.576 Database1.accdb
16/12/2015  23:52           524.288 Database2.accdb
17/12/2015   00:18           1.572.864 Database3.accdb
30/12/2015  13:00    <DIR>          Desktop
28/12/2015  11:57    <DIR>          Documents
30/12/2015  12:37    <DIR>          Downloads
22/12/2015  14:45           24.270 Estudos
17/12/2015  17:43    <DIR>          Favorites
22/12/2015  14:30           24.270 H
17/12/2015  17:43    <DIR>          Links
17/12/2015  17:43    <DIR>          Music
14/12/2015  01:09    <DIR>          OneDrive
17/12/2015  17:43    <DIR>          Pictures
17/12/2015  17:43    <DIR>          Saved Games
```

O que temos é a lista de todos arquivos do diretório em que estávamos, ou seja, do diretório `C:\Users\Natanael`. Observe que alguns itens possuem a sintaxe `<DIR>` na frente. Isso indica que eles são pastas, que podem ser acessadas. Já os que não possuem, são arquivos.

Vamos supor que queiramos acessar a pasta Desktop (essa pasta existe dentro do Windows, e corresponde a todos os arquivos presentes na sua Área de Trabalho). Para isso, utilizamos a seguinte sintaxe

`cd diretório`

Em diretório, devemos inserir a pasta que desejamos acessar. No nosso caso, seria assim:

```
C:\Users\Natanael>cd Desktop
C:\Users\Natanael\Desktop>
```

Feito isso, passamos para a pasta Desktop. A partir daí, podemos rodar outro dir e ir verificando o conteúdo dos diretórios ao mesmo tempo em que navegamos.

Vamos supor que em nosso Desktop exista uma página chamada arquivos, e dentro dela, outra pasta chamada pessoal, de forma que queremos acessar é última. Se quisermos acessar uma pasta dentro de outra pasta diretamente, sem necessariamente passar por todas as suas subpastas, podemos inserir o comando cd seguido do diretório alvo. No caso, arquivos\pessoal:

```
C:\Users\Natanael>cd Desktop
C:\Users\Natanael\Desktop>cd arquivos\pessoal
C:\Users\Natanael\Desktop\arquivos\pessoal>
```

Imagine que queremos apenas voltar uma pasta atrás. No caso, indo para a pasta arquivos. Podemos fazer isso com a seguinte sintaxe:

`cd ..`

```
C:\Users\Natanael\Desktop\arquivos\pessoal>cd ..
C:\Users\Natanael\Desktop\arquivos>
```

Agora, queremos voltar diretamente para o primeiro diretório exibido na listagem, que no caso é o diretório C:\. Podemos fazê-lo com:

`cd \`

Isso nos levará diretamente ao diretório-mãe.

```
C:\Users\Natanael\Desktop\arquivos>cd \  
C:\>
```

Mas e se quisermos ir para outra unidade? Por exemplo, um disco externo cuja letra é D? O fazemos com `letra:.`. Se a unidade especificada existir, o CMD irá trocar para ela imediatamente, como no exemplo.

```
C:\>d:  
D:\>
```

Basicamente, esses são os comandos da navegação em shell no Windows. Utilizando-os, pode-se trocar para qualquer pasta do computador.

## 5) Atributos

No capítulo anterior, aprendemos a sintaxe do comando `dir`. Porém, esse comando, assim como qualquer outro, pode ser incrementado com atributos. É praticamente impossível saber de cor todos eles. Justamente por essa razão, existe o comando `help`, onde podemos obter uma lista de todos os atributos existentes. Por exemplo, se quisermos ajuda com o comando `dir`, podemos digitar:

```
help dir
```

```
DIR [unidade:][caminho][arquivo] [/A[:atributos]] [/B] [/C] [/D] [/L] [/N]
  [/O[:ordem_de_classificação]] [/P] [/Q] [/R] [/S] [/T[:campo_de_tempo]]
  [/W] [/X] [/4]

[unidade:][caminho][nome_de_arquivo]
  Especifica a unidade, o diretório e/ou arquivos a serem listados.

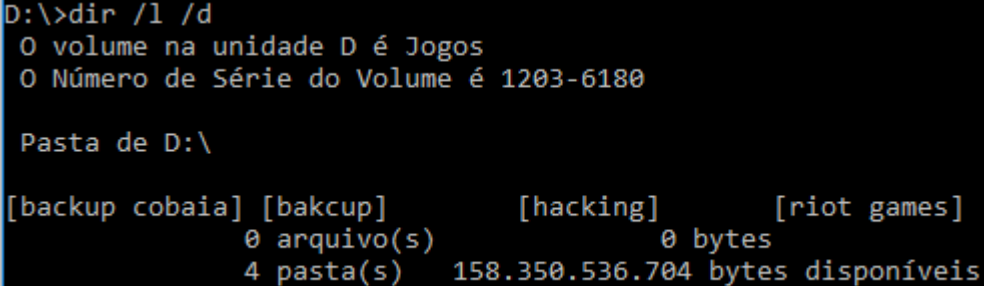
/A      Exibe arquivos com atributos especificados.
atributos  D Diretórios                R Arquivos somente leitura
           H Arquivos ocultos          A Arquivos prontos para arquivamento
           S Arquivos de sistema       I Arquivos sem conteúdo indexado
           L Pontos de nova análise - Prefixo significando negação

/B      Usa formatação básica (sem informações de cabeçalho ou resumo).
/C      Exibe o separador de milhar em tamanhos de arquivos. É o
padrão. Use /-C para desabilitar a exibição do separador.
/D      O mesmo que amplo, mas os arquivos são classificados na lista
por coluna.
/L      Usa letras minúsculas.
/N      Novo formato de lista longo onde os nomes de arquivos estão
à extrema direita.
/O      Lista por arquivos na ordem classificada.
ordem_de_classificação
         N Por nome (alfabético)       S Por tamanho (menor primeiro)
         E Por extensão (alfabética)  D Por data/hora(mais antiga primeiro)
         G Grupo de diretórios primeiro - Prefixo para inverter a ordem

/P      Pausa após cada tela de informações.
Pressione qualquer tecla para continuar. . .
```

O resultado é esse, e a sintaxe serve para qualquer outro comando do CMD. Você pode inserir esses atributos da forma como é especificado na ajuda. Por exemplo

```
dir /l /d
```



```
D:\>dir /l /d
O volume na unidade D é Jogos
O Número de Série do Volume é 1203-6180

Pasta de D:\

[backup cobaia] [bakcup]          [hacking]          [riot games]
                0 arquivo(s)          0 bytes
                4 pasta(s) 158.350.536.704 bytes disponíveis
```

Isso irá incrementar o comando, conforme os atributos selecionados.

## 6) Exibição de Mensagens

A partir desse momento, já passamos a criar nosso scripts para depois executá-los. Para rodar um script em Batch, crie o código no editor de texto preferido, salve-o com a extensão .bat e o execute.

A Linguagem Batch é baseada na inserção de comandos no CMD. Assim, se criarmos um arquivo em Batch contendo apenas o código dir, ele irá rodar o comando ao ser executado, e mostrará na tela qual foi o comando inserido. Porém, podemos fazer com que apareça apenas o resultado desse comando, e não sua inserção, fazemos isso adicionando o seguinte comando no início do arquivo:

```
@echo off
```

Com isso, reduzimos a poluição visual presente na tela.

Para exibir uma mensagem, usamos o comando echo, damos um espaço e em seguida colocamos a mensagem.

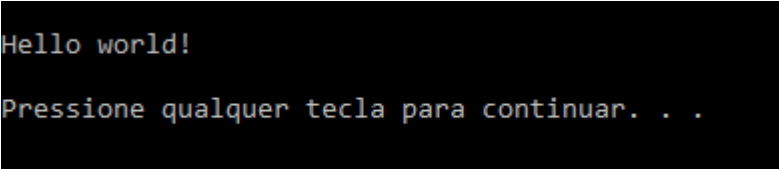
Já para inserirmos uma linha em branco, usamos echo..

Esses são os únicos recursos visuais que temos. Sendo assim, é preciso alguma criatividade para criar interfaces em Batch.

Após inserir um comando que não envolva uma pausa automática, o CMD irá fechar. Para impedir que isso aconteça sem a confirmação do usuário (ou para evitar que o script simplesmente continue após executar determinada ação), podemos usar o comando pause. Assim, o tradicional “Hello world!” em Batch seria escrito dessa forma:

```
@echo off
echo.
echo Hello world!
echo.
pause
```

Aqui, usamos @echo off para deixar a tela mais limpa, depois usamos echo para exibir a mensagem, colocamos duas linhas em branco para deixar mais estético, e pausamos o script. O resultado é esse:



```
Hello world!
Pressione qualquer tecla para continuar. . .
```

Ao pressionarmos qualquer tecla, o CMD irá fechar, uma vez que não há mais nenhuma linha de código abaixo do pause.

Se desejarmos apagar tudo que já foi escrito na tela, podemos usar o comando cls.

Podemos dar um título para nossa janela, utilizando o comando title:

```
title Título
```

Também podemos fazer comentários em Batch. Iniciamos o comentário utilizando :: .

```
:: comentario qualquer
```

## 7) Etiquetas e Menus

Podemos “etiquetar” partes do nosso código com um determinado nome, e fazer que o programa pule para a parte especificada. Sendo assim, criamos uma etiqueta usando:

```
:etiqueta
```

O nome colocado depois dos dois pontos é o nome da etiqueta. Para fazer com que, em determinado momento, o script pule para tal etiqueta, basta usar:

```
goto etiqueta
```

Dois etiquetas não podem ter o mesmo nome, e você pode criá-las em qualquer lugar da página. Veja o exemplo abaixo. Não se preocupe em entender com o menu foi construído, isso será explicado adiante. Foque apenas no uso das etiquetas.



```
@echo off

:menu
echo 1 - Calculadora
echo 2 - Bloco de Notas

choice /c 12 /m "O que deseja executar?"

IF errorlevel==2 goto bloco
IF errorlevel==1 goto calc

:calc
start calc
goto menu

:bloco
start notepad
goto menu
```

Note que temos 3 etiquetas: menu, calc e bloco na parte marcada em vermelho. Se escolhermos a opção 1. Iremos para a parte marcada em amarelo, e depois voltamos para o vermelho. Já na opção 2, vamos para o azul, e depois voltamos novamente ao vermelho.

Perceba que, caso não colocássemos goto menu no final de cada parte, o script simplesmente continuaria para a próxima linha, invalidando a escolha feita.

Vamos agora criar nosso próprio menu de opções. Existem duas formas de cria-lo. A primeira delas envolve o uso do comando choice, que é a que mostrarei agora.

Esse comando funciona de uma forma um pouco diferente dos demais. Primeiro, analise o menu abaixo:

```
echo A - Calculadora
echo B - Bloco de Notas
echo C - Paint
```

```
choice /c ABC /m "O que deseja executar?"
```

Primeiro, criamos um menu com 3 opções de programas. Em seguida, rodamos o comando choice. Perceba que usamos os atributos /c e /m.

O atributo /m serve para inserirmos uma pergunta no contexto do menu. Basta utilizá-lo seguido pela pergunta entre aspas.

O atributo /c serve para especificar as teclas que podem ser pressionadas na execução do menu. Assim, ABC significa que se teclas A, B e C forem pressionadas, irão ativar alguma opção do menu. Caso uma tecla diferente seja pressionada, o CMD emitirá um som.

Agora, precisamos construir a parte do código que será responsável por ativar cada tecla. Veja como ela fica:

```
if errorlevel==3 goto paint
if errorlevel==2 goto blocodenotas
if errorlevel==1 goto calc
```

A construção dessa parte pode ser um tanto complicada. Como pode perceber, usamos if errorlevel== para realizar a comparação. Cada um dos números (3,2,1) representa as teclas pressionadas. Parece um tanto complicado associar cada número com sua tecla. Veja uma ideia para facilitar isso:

```
1 2 3
choice /c ABC /m "O que deseja executar?"
```

Apenas considere um numeral ordinal para cada tecla inserida. Assim, opção 2 irá ser ativada se B for pressionado, fazendo com que o Bloco de Notas se abra.

Outra maneira é considerar que a segunda parte deve estar na ordem inversa do menu.

Nunca se esqueça de colocar os valores do menu sempre em ordem decrescente. Faça o mesmo para quantas opções houverem.

A parte em roxo representa a ação a ser tomada por cada opção. No caso, ir para a parte do código responsável pela Calculadora, pelo Bloco de Notas ou pelo Paint.

Poderíamos ter usado outro comando (desde que coubesse em uma linha), mas é recomendado usar as etiquetas. Assim ficaria o código final:

```

@echo off

:menu

echo A - Calculadora
echo B - Bloco de Notas
echo C - Paint

choice /c ABC /m "O que deseja executar?"

if errorlevel==3 goto paint
if errorlevel==2 goto blocodenotas
if errorlevel==1 goto calc

:blocodenotas
start notepad
pause
goto menu

:calc
start calc
pause
goto menu

:paint
start mspaint
pause
goto menu

```

Particularmente, considero infinitamente mais fácil construir menus utilizando variáveis.

## 8) Comandos Diversos

Como informarmos anteriormente, a linguagem Batch pode ser extremamente útil para automatizar tarefas. Sendo assim, esse capítulo é destinado a mostrar alguns comandos úteis para tais tarefas. Nas próximas páginas há uma lista completa (em português – você não acha isso em qualquer lugar) dos comandos em batch.

Caso deseja aprender mais sobre a sintaxe de algum comando, rode um help no CMD.

## Lista de Comandos para Batch/MS-DOS - Fábrica de Noobs

<b>Comando</b>	<b>Função</b>
addusers	Adiciona usuários com base em uma lista de um arquivo CSV
assoc	Permite alterar a relação entre extensões e tipos de arquivo
attrib	Altera atributos de arquivo
bcdboot	Cria ou repara uma partição do sistema
bcdedit	Gerencia os dados da partição de boot
bitsadimin	Gerencia contas de usuário
bootcfg	Altera configurações de boot
cacls	Altera permissões de arquivo
call	Inicia outro programa em Batch
certreq	Requisita certificado de segurança
certutil	Utilidade para certificados de segurança
cd	Altera o diretório atual
change	Altera propriedades da seção atual no CMD
chkdsk	Verifica e repara problemas de disco
chkntfs	Verifica o sistema de arquivos NTFS
choice	Aceita input do teclado (construção de menus)
cipher	Criptografa arquivos ou diretórios
clip	Copia linhas do CMD para a área de transferência do Windows
cls	Limpa o CMD
cmd	Inicia uma nova janela do CMD
cmdkey	Gerencia nomes de usuário e senhas armazenados
color	Muda a cor da janela do CMD
comp	Compara o conteúdo de dois arquivos
compact	Compacta arquivos ou pastas em uma partição NTFS
compress	Compacta arquivos
convert	Converte um disco FAT para NTFS
copy	Copia arquivos
date	Exibe ou altera a data do sistema
defrag	Desfragmenta um disco rígido

del	Deleta um ou mais arquivos
dir	Exibe detalhes do diretório
diskpart	Exibe o uso do disco
doskey	Edita linha de comando, chama comandos ou altera macros
driverquery	Lista drivers instalados
echo	Exibe uma mensagem na tela
endlocal	Termina a localização das alterações de ambiente em um arquivo em lotes
erase	Deleta um ou mais arquivos
eventcreate	Adiciona uma mensagem no log de eventos do Windows.
exit	Sai do CMD
expand	Descomprime arquivos CAB
extract	Descomprime arquivos CAB
fc	Compara dois arquivos
find	Procura por uma string de texto em um arquivo
findstr	Procura por strings em arquivos
for	Comando de loop
forfiles	Processa vários arquivos em Batch
format	Formata um disco
fsutil	Utilidades de arquivo e volume
ftp	Protocolo para transferência de arquivos
ftype	Associações entre extensões de arquivos
getmac	Exibe o endereço MAC
goto	Direciona a linha de código para uma etiqueta
gpresult	Exibe as informações do Conjunto de Políticas Resultante
gpupdate	Atualiza as informações do Conjunto de Políticas Resultante
help	Ajuda com comandos
hostname	Exibe o nome do hosting do computador
icacls	Altera permissões de arquivo ou pasta
iexpress	Cria um arquivo auto-extraível
if	Expressa condição
ipconfig	Configura IP

label	Altera o rótulo de um disco
lodctr	Atualiza os contadores do registro relacionados ao desempenho
logman	Gerencia logs
logoff	Realiza logoff de um usuário
makecab	Cria arquivos CAB
md	Cria novas pastas
mklink	Cria um link de diretório simbólico
mode	Configura um dispositivo do sistema
more	Exibe as informações tela a tela
mountvol	Cria, exclui ou lista um ponto de montagem de volume
move	Move arquivos de uma pasta para outra
msg	Envia uma mensagem
msiexec	Windows Installer
msinfo32	Informações do sistema
net	Gerencia recursos de rede
nertsh	Configura interface de rede, firewall e relacionados
nbtstat	Exibe as estatísticas de protocolo e as conexões TCP/IP atuais que usam NBT
netstat	Exibe as estatísticas de protocolo e as conexões TCP/IP atuais
nltest	Network Location Test
nslookup	Propriedades de rede/servidor
openfiles	Abre arquivos
path	Exibe o caminho do diretório de programas executáveis
pathping	Traça rota de pacotes de rede
pause	Suspende a linha de comando até pressionar uma tecla
perfmon	Exibe monitor de desempenho
ping	Testa conexão com a internet
powercfg	Configura opções de energia
print	Imprime arquivos de texto
push	Abre um diretório salvo pelo comando pushd
pusdh	Salva diretório
rasdial	Gerencia conexões RAS

rasphone	Gerencia conexões RAS
recover	Recupera arquivos de um disco com defeito
reg	Opções de registro
regedit	Importa ou exporta arquivos de registro
regsvr32	Registra ou remove uma DLL
regine	Altera permissões de registro
rem	Adiciona comentários em arquivos Batch
ren	Renomeia pastas ou arquivos
replace	Substitui um arquivo por outro
rd	Deleta pasta
robocopy	Ferramenta para cópia de diretórios e pastas
route	Gerenciar rotas de rede
runas	Executa um programa de outra conta de usuário
rundll32	Roda um comando de DLL
sc	Service Control
schtasks	Permite que um administrador gerencie tarefas agendadas
set	Operações de variáveis
setlocal	Controla a visibilidade de variáveis
sfc	Sistema de checagem de arquivos
shift	Altera a posição de parâmetros em arquivo Batch
shortcut	Cria atalhos
shutdown	Desliga ou reinicia o computador
start	Inicia um programa
subst	Associa um caminho com um rótulo de disco
systeminfo	Exibe informações do sistema
tasklist	Exibe uma lista de processos em execução
taskkill	Finaliza um processo em execução
telnet	Comunica-se com outro computador usando protocolo Telnet
time	Exibe a hora atual
timeout	Define um tempo limite para aguardar
title	Define um título para a janela do CMD

tracert	Exibe a rota para um servidor externo
tree	Exibe diretórios em árvore
tsdiscon	Desconecta uma sessão remota
tskill	Finaliza um processo em execução
type	Exibe o conteúdo de um arquivo de texto
tzutil	Altera o fuso-horário
ver	Exibe informações sobre a versão do sistema
verify	Verifica se os arquivos foram salvos
vol	Exibe informações sobre o disco atual
w32tm	Serviço de horário
waitfor	Espera ou envia um sinal
wevtutil	Limpa log de eventos
where	Localiza e exibe arquivos em um diretório
whoami	Exibe o nome de usuário atual
winrm	Gerenciamento remoto
wmic	Comandos WMT
wuauclt	Windows Update
xcopy	Copia arquivos ou diretórios
::	Inserir comentários



## 9) Variáveis

Um dos principais elementos de uma linguagem de programação são as variáveis. E em Batch não é diferente.

Para criarmos uma nova variável usamos o comando `set`. Isso pode ser feito de 3 maneiras:

Se queremos apenas fazer com que a variável assuma um valor predefinido, que não envolva uma operação matemática ou um valor digitado pelo usuário, fazemos:

```
set a=valor
```

Dessa forma, a variável que chamamos de `a` irá ser igualda a um valor qualquer, de texto ou numérico. No nosso caso, `valor`.

Em Batch, podemos realizar as 4 operações fundamentais, com os seguintes sinais, podendo também usar os parênteses.

Soma	+
Subtração	-
Multiplificação	*
Divisão	/

Se quisermos que uma variável assuma o resultado de uma operação matemática, usamos o comando `/a` depois de `set`. Por exemplo:

```
set /a var=2+2
```

Assim, a variável `var` passa a valer 4.

Já se desejarmos fazer com que a variável assuma um valor a ser inserido pelo usuário, podemos usar o comando `/p` depois de `set`. Por exemplo:

```
set /p nome=Insira um valor
```

Aqui, a variável `nome` irá receber o valor que o usuário digitar na tela. “Insira um valor” será a mensagem que vai aparecer informando ao usuário de que ele deve digitar algo na tela.

Sempre que queremos nos referir a uma variável, seja para usá-la em uma frase, ou fazer comparações, a escrevemos entre dois sinais de porcentagem. Por exemplo:

echo o valor inserido é igual a %valor%

Observe o código abaixo, que pede para que um valor seja inserido e avisa qual ele é.

```
@echo off

set /p a=Insira um valor
echo 0 valor inserido foi %a%.

pause
```

Seu resultado no CMD:

```
Insira um valor5
0 valor inserido foi 5.
Pressione qualquer tecla para continuar. . .
```

Agora, observe outro código, o de uma calculadora em Batch.

```
@echo off

:menu

@echo off
title = Calculadora

set /p a=Insira o primeiro valor.
set /p b=Insira o segundo valor.

echo A - Soma
echo B - Subtracao
echo C - Multiplicacao
echo D - Divisao

choice /c ABCD /m "O que deseja calcular?"

if errorlevel==4 goto div
if errorlevel==3 goto mult
if errolevel==2 goto sub
if errorlevel==1 goto soma

:soma
set /a result=%a% + %b%
goto rel

:sub
set /a result=%a% - %b%
goto rel

:mult
set /a result=%a% * %b%
```

```

goto rel

:div
set /a result=%A% / %b%
goto rel

:rel
echo.
echo 0 resultado e igual a %result%
echo.
pause
goto sair

:sair
exit

```

Aqui, primeiro pedimos, usando set /p, os dois valores. Depois, criamos um menu com 4 opções. Uma vez escolhido a opção, redirecionamos para a parte do código responsável por realizar determinada operação. Em seguida, exibimos o resultado e encerramos o programa. Veja como ele fica no CMD:

```

Insira o primeiro valor.10
Insira o segundo valor.14
A - Soma
B - Subtracao
C - Multiplicacao
D - Divisao
O que deseja calcular? [A,B,C,D]?C

O resultado e igual a 140

Pressione qualquer tecla para continuar. . .

```

## 10) Tomadas de Decisão

Podemos também comparar variáveis. Fazemos isso utilizando o comando if, na seguinte estrutura:

```

set /p a=Insira o numero 2.

if %a% EQU 2 (echo Certo) else (echo Errado)

```

Primeiro, colocamos o comando if, seguido da variável entre aspas. Depois, inserimos um comparador. Em Batch, sua sintaxe é:

Igual	EQU
Diferente	NEQ
Menor que	LSS
Menor que ou igual	LEQ
Maior que	GTR
Maior que ou igual	GEQ

Depois de inserir o comparador, inserimos o segundo valor a ser comparado (que pode ser um número ou uma variável). Em seguida, abrimos parênteses, e dentro deles, colocamos a decisão para o “se”. Depois, (na mesma linha) inserimos else, abrimos outros parênteses, e colocamos a decisão para o “se não”.

Podemos colocar quantos if quisermos, e o uso do else é opcional. Se todas as condições possíveis forem colocadas nos if's, não é necessário construir uma condição else.

Veja o script abaixo:

```
@echo off

:menu

set /p a=Insira uma nota.

if %a% GTR 10 (echo A nota inserida e maior que 10. Nota invalida.
               goto :menu)
if %a% LSS 0 (echo A nota inserida e negativa. Nota invalida.
              goto :menu)
if %a% GEQ 6 (echo Nota azul
              pause)
if %a% LSS 6 (echo Nota vermelha
              pause)
```

Aqui, fazemos sucessivas comparações para determinar se uma nota inserida é inválida (maior que 10), negativa (menor que 0), azul (maior ou igual a 6) ou vermelha (menor que 6).

Observe que, dentro de cada parênteses, construímos um pequeno script para cada condição. Poderíamos também usar apenas o comando goto e direcionar para outra parte do código.

Agora, observe esse fragmento:

```

echo 10 - Compactados (zip,rar)
echo 11 - Outro (inserir)
echo.
SET /p proc=0 que deseja copiar?

if %proc% EQU 1 (goto all)
if %proc% EQU 2 (goto imagem)
if %proc% EQU 3 (goto video)
if %proc% EQU 4 (goto musica)
if %proc% EQU 5 (goto texto)
if %proc% EQU 6 (goto office)
    if %proc% EQU 6.1 (goto word)
    if %proc% EQU 6.2 (goto excel)
    if %proc% EQU 6.3 (goto powerpoint)
    if %proc% EQU 6.4 (goto access)
if %proc% EQU 7 (goto web)
if %proc% EQU 8 (goto design)
if %proc% EQU 9 (goto sistema)
if %proc% EQU 10 (goto compactados)
if %proc% EQU 11 (goto outro) else (goto menuprincipal)

:: aqui comecam os demais arquivos

:all
xcopy "%origem%\*" "%destino%" %sub%
pause
goto menuprincipal

:texto
xcopy "%origem%\*.txt" "%destino%" %sub%
xcopy "%origem%\*.docx" "%destino%" %sub%
xcopy "%origem%\*.pdf" "%destino%" %sub%
pause

```

Aqui, construímos um menu utilizando variáveis (e não o comando choice, o que também é possível) e direcionamos cada opção escolhida para uma parte do código. Isso ajuda a manter o script mais organizado.

## 11) Conclusão

Sendo assim, terminamos nosso curso de Batch. Agora, você já está apto a construir programas e arquivos em lot capazes de executar tarefas do Windows. Isso inclui ferramentas para backup, automatização de processos ou até trojans. Explore a sintaxe dos demais comandos, faça experiências e logo perceberá que Batch é uma linguagem poderosa.