

Batch ou **arquivos de lote** (também conhecidos por **.bat** ou *ponto-bat*) são arquivos de computador utilizados pra automatizar tarefas, é regularmente confundido com o modo de processamento de dados no qual os dados são processados em grupos, ou lotes, através de uma rotina agendada. O batch neste caso, nada mais é um conjunto de comandos que são rodados sequencialmente. Podemos compará-lo grosseiramente aos scripts do Unix. Criado para o MSDOS, o batch foi utilizado para o sistema operacional carregar programas para preparar o ambiente para rodar outros programas. Por exemplo, o EMM386 que habilitava o uso da memória expandida (além dos 640K).

Alguns técnicos de informática utilizam deste recurso para facilitar a instalação do sistema operacional da Microsoft (*Windows* por exemplo) utilizando disquetes de boot.

Um exemplo de um job batch. *Nota: executar os scripts deste artigo no seu sistema Windows pode danificá-lo ou causar perda de dados.*

```
shutdown -s
```

e shutdown -a são formas de desligamento forçados sendo -a o anulador do shutdown -s

Este tipo de arquivo pode ser muito perigoso, pois eles são capazes de destruir o sistema operacional sem o usuário saber deletando a pasta principal, enchendo o HD, desconfigurando a inicialização do sistema, entre outros. Por isso é recomendado que apenas execute jobs batch conhecidos. No windows, estes arquivos podem ser facilmente editados clicando com o botão direito do mouse nele e clicando em *Editar*. Você pode fazer um utilizando o bloco de notas do windows onde cada linha é um comando diferente de MS-DOS e ao final, salve-o sempre com o final do nome *.bat* (*por_exemplo.bat*).

Como estes arquivos não são capazes de interagir com o usuário, não é possível fazer jogos e programas em geral apenas com batch, uma alternativa para isso é a utilização da linguagem de programação Basic, que é bem parecida com jobs batch.

Índice

- 1 Comandos
- 2 chicom
- 3 Condições
 - 3.1 Caracteres
 - 3.2 Numéricas
 - 3.3 Variáveis
 - 3.4 Arquivos
- 4 Exemplos
 - 4.1 X é igual a 1?
 - 4.1.1 Faça o teste
 - 4.2 Mensagem de acordo com a idade
 - 4.2.1 Faça o teste
- 5 Ver também

Comandos

Aqui estão alguns comandos utilizados em jobs batch.

```
ECHO [mensagem ou variável]
```

Escreve na tela

```
ECHO OFF
```

Desactiva as informações e confirmações feitas pelo sistema. Como "C:>", "*Tem certeza que deseja fazer tal?*"

```
ECHO ON
```

Activa as informações e confirmações do sistema.

```
ECHO.
```

Pula uma linha

```
SET variavel=valor
```

Cria ou modifica uma **variável** (espaço reservado temporariamente na memória RAM para guardar algumas informações numéricas, de caracteres, positivo/negativo entre outros. Não podem ter acento e devem começar com letra). Para chamar a variável posteriormente basta colocá-la entre "%".

Por exemplo:

```
SET nome=João  
ECHO O nome dele é %nome%
```

```
:nomedolink
```

Nomeia um ponto do arquivo de lote permitindo avançar ou voltar à execução a partir daquele ponto

```
CLS
```

Esse comando limpa todo o ecrã. **CL**ear **S**creen.

chicom

```
@[Comando]
```

Desactiva as informações e confirmações feitas pelo sistema apenas para esta linha.

```
IF [condição] (  
    [acção]  
) ELSE (  
    [acção2]  
)
```

Este é um comando condicional. "*Se condição faça acção, senão faça acção2*".

Por exemplo:

```
@ECHO OFF  
SET nome=Ana  
IF "%nome%"=="Ana" (  
    ECHO Oi Aninha!  
) ELSE (  
    ECHO Oi %nome%!  
)
```

```
GOTO [ponto]
```

Avança ou volta a execução para um ponto do arquivo de lote.

```
FOR /L %%variavel IN (inicio,incremento,fim) DO comando [parâmetro]
```

Repete um comando várias vezes fazendo incrementações até o fim ser atingido.

/L, é uma opção do comando **FOR** que faz as incrementações.

%%variavel, é o nome da variável que vai ser incrementada.

inicio, é o valor de início da variável que será incrementada.

incremento, número que será somado a cada incrementação.

fim, a repetição irá acabar quando o valor da variável atingir este valor.

comando, um comando que será repetido.

[parâmetro], o parâmetro do comando que será repetido.

Por exemplo:

```
@ECHO OFF  
FOR /L %%a IN (1,1,5) DO ECHO O valor de A é %%a
```

```
PAUSE
```

Faz uma pausa e só volta quando o utilizador premir alguma tecla do teclado.

```
REM [comentário]
```

Este comando não faz absolutamente nada. Ele é utilizado para inserir comentários nos arquivos.

- *Todos aqui listados são comandos de MS-DOS que foram criados especificamente para arquivos de lote. Se você precisar de mais detalhes sobre cada um deles basta digitar no MS-*

*DOS (ou no prompt de comando do Windows) [**comando**]/? Todos comandos de MS-DOS também entram nesta lista incluindo chamadas de outros programas para este sistema.*

Condições

- **Numéricas e Caracteres:**
 - Igual (**EQU** ou **==**)
 - Diferente (**NEQ**)
 - Menor que (**LSS**)
 - Menor ou igual a (**LEQ**)
 - Maior que (**GTR**)
 - Maior ou igual a (**GEQ**)
 - Condição negativa (**NOT**)
- **Variáveis:**
 - Definido (**DEFINED** [*variável sem %*])
 - Condição negativa (**NOT**)
- **Em arquivos:**
 - Existe (**EXIST**)
 - Condição negativa (**NOT**)

Os símbolos acima vêm das seguintes fontes:

```

EQU vem da palavra inglesa EQuals (Igual).
NEQ vem da palavra inglesa Not EQuals (Não-Igual, diferente)
LSS vem da palavra inglesa LeSS (Menor)
LEQ vem da palavra inglesa Less or EQuals (Menor ou Igual)
GTR vem da palavra inglesa GreaTeR (Maior)
GEQ vem da palavra inglesa Greater or EQuals (Maior ou Igual)
NOT é uma palavra inglesa que significa Não
DEFINED é uma palavra inglesa que significa Definido
EXIST é uma palavra inglesa que significa Existe

```

Caracteres

As condições numéricas e de caracteres são iguais, porém avaliam de maneira diferente. Por exemplo, se compararmos "Ana" com *Danielle* na condição **LEQ** (*Menor ou igual*), será avaliado o tamanho . "Ana" tem 3 bytes e "Danielle" tem 8. Então "Danielle" é maior que "Ana" e a condição é positiva.

Veja como ficaria:

```

@ECHO OFF
SET nome1=Ana
SET nome2=Danielle
REM O resultado da condição abaixo será positivo
IF "%nome1%" LEQ "%nome2%" ( ECHO Positivo ) ELSE ( ECHO Negativo )

```

Numéricas

Nas condições numéricas não há mistério, as duas únicas diferenças entre as condições numéricas e de caracteres é que nas numéricas são avaliados os números, não o tamanho, e que você não usa aspas. Vamos supor que a *Ana* tem 23 anos e que a *Danielle* tem 19. Se compararmos as suas idades respectivamente usando **LSS** (*Menor que*) o resultado será negativo.

Veja como ficaria:

```
@ECHO OFF
SET ana=23
SET danielle=19
REM O resultado da condição abaixo será negativo
IF %ana% LSS %danielle% ( ECHO Positivo ) ELSE ( ECHO Negativo )
```

Variáveis

A condição **DEFINED** verifica se uma variável já foi definida. Por exemplo, se a variável "x" estiver definida, então o resultado será positivo, senão negativo.

Veja como ficaria:

```
@ECHO OFF
IF DEFINED x ( ECHO Positivo ) ELSE ( ECHO Negativo )
```

Arquivos

A condição **EXIST** verifica se existe algum arquivo. Por exemplo, se existir o arquivo "*exemplo.bat*" o resultado será positivo, senão negativo.

Veja como ficaria:

```
@ECHO OFF
IF EXIST exemplo.bat ( ECHO Positivo ) ELSE ( ECHO Negativo )
```

Esta condição não para por aí. No sistema MS-DOS, você pode fazer uma "*pesquisa*" na pasta em que está por um arquivo digitando parte dele. Você usa "*" para indicar a(s) parte(s) que você não sabe. Por exemplo, se existir algum arquivo com a extensão ".bat" na pasta atual o resultado será positivo, senão negativo.

Veja como ficaria:

```
@ECHO OFF
IF EXIST *.bat ( ECHO Positivo ) ELSE ( ECHO Negativo )
```

Exemplos

X é igual a 1?

```
@ECHO OFF
CLS
SET X=1
IF "%X%" == "1" GOTO ok
ECHO X não é igual a 1, X é igual a %X%
GOTO saida
:ok
ECHO X é igual a 1
:saida
```

Na primeira linha. O "@" indica que não é para aparecer o comando que segue, no caso, "ECHO" (Comando que escreve na janela). Nesta linha, eu estou grosseiramente mandando o MS-DOS *calar a boca* (não falar nada sem eu mandar), pois se **ECHO** faz o computador "falar", **ECHO OFF** faz o computador "*desligar a boca*" e com o @ no começo da linha, ele nem vai informar o

utilizador que está parando de falar.

Na segunda linha, o comando *CLS* faz o MS-DOS apagar todo o ecrã.

Na terceira linha, como já foi explicado, faz a variável *X=1*.

Na quarta linha, é uma linha condicional. Se "*X*" for igual à 1 vá para "ok".

Na quinta linha, é óbvio que *X* não vai ser igual a 1, então o MS-DOS vai "falar" *X não é igual a 1, X é igua a [valor de x]*.

Na sexta linha, ele irá para o ponto "saida" porque o resto é feito se *X* for igual a 1.

Na sétima linha, está sendo definido o ponto "ok", aquele ponto citado na quarta linha.

Na oitava linha, *X* será igua a 1, então o MS-DOS vai dizer *X é igual a 1*.

E na nona linha, é definido o ponto-final do processamento. Que é chamado na sexta linha, nada mais que isso, então o MS-DOS entende que o job batch concluiu.

Faça o teste

1. Abra o *Bloco de notas do Windows*;
2. Copie o exemplo acima e cole nele;
3. Salva-o como *exemplo.bat*;
4. Dê dois-clicks nele observe o que acontecerá;
5. Agora experimente modificar a terceira linha por *SET X=qualquer coisa*;
6. Execute e veja o que acontece.

Mensagem de acordo com a idade

```
@ECHO OFF
SET idade=15
IF %idade% LSS 10 (
    ECHO Bom dia menino!
    ECHO.
    ECHO Você não vai pra escola?
) ELSE (
    IF %idade% LSS 18 (
        ECHO Bom dia garoto!
        ECHO.
        ECHO Você não vai pro colégio?
    ) ELSE (
        IF %idade% LEQ 64 (
            ECHO Bom dia!
            ECHO.
            ECHO Você não vai trabalhar hoje?
        ) ELSE (
            ECHO Bom dia!
            ECHO.
            ECHO Tudo bem?
        )
    )
)
```

Esta é uma forma mais avançada de usar o comando condicional **IF**. Isso é quase idêntico a Algoritmo. Ele avalia a variável *%idade%*. Se *%idade%* for menor que 10 então escreva tal; Senão, se *%idade%* for menor que 18 então escreva tal; Senão se *%idade%* for menor ou igual a 64 então escreva tal; Senão obviamente *%idade%* é maior que 64, então imprima tal.

Faça o teste

1. Abra o *Bloco de notas do Windows*;
2. Copie o exemplo acima e cole nele
3. Salva-o como *exemplo2.bat*;
4. Dê dois-clicks nele observe o que acontecerá;
5. Agora experimente modificar a segunda linha por *SET idade=n* (troque **n** por algum número);

6. Execute e veja o que acontece.