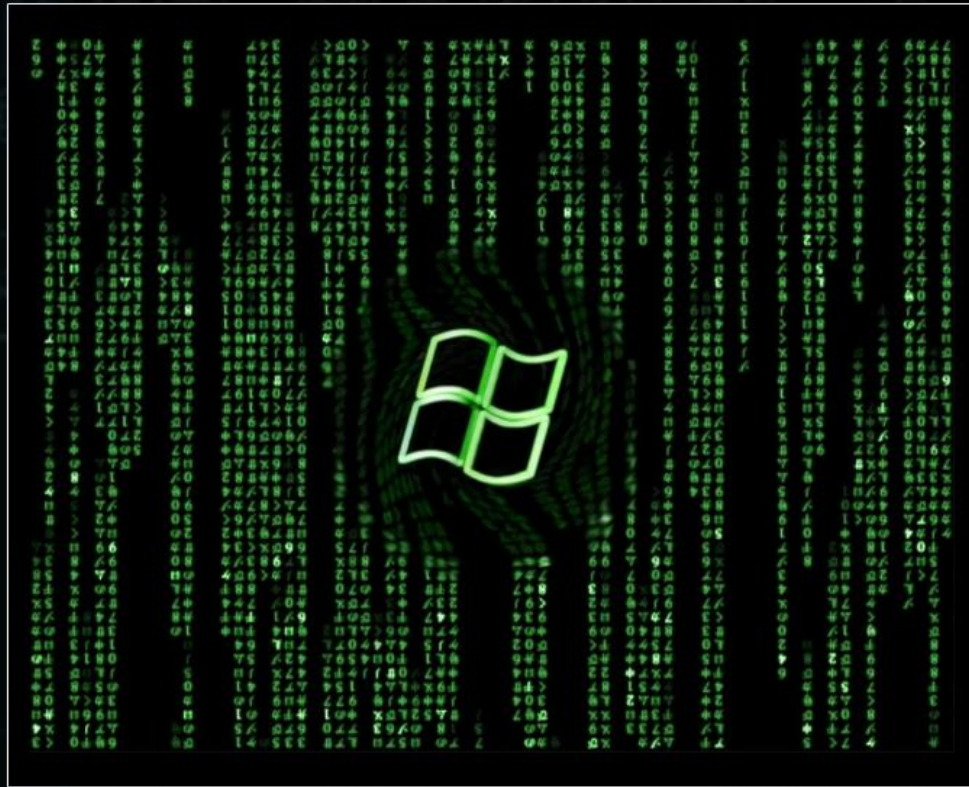


# UNDERCODE

## TALLER DE PROGRAMACION EN BATCH



### TEMAS

- HISTORIA
- ECHO ON/OFF
- ECHO/ECHO.
- GOTO/ELSE
- PAUSE
- Y MAS...!

### TUTOR

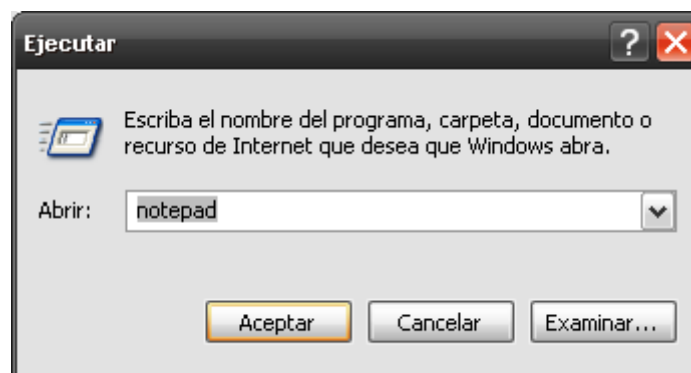
ABNORMALITY

## Historia Del MS-DOS

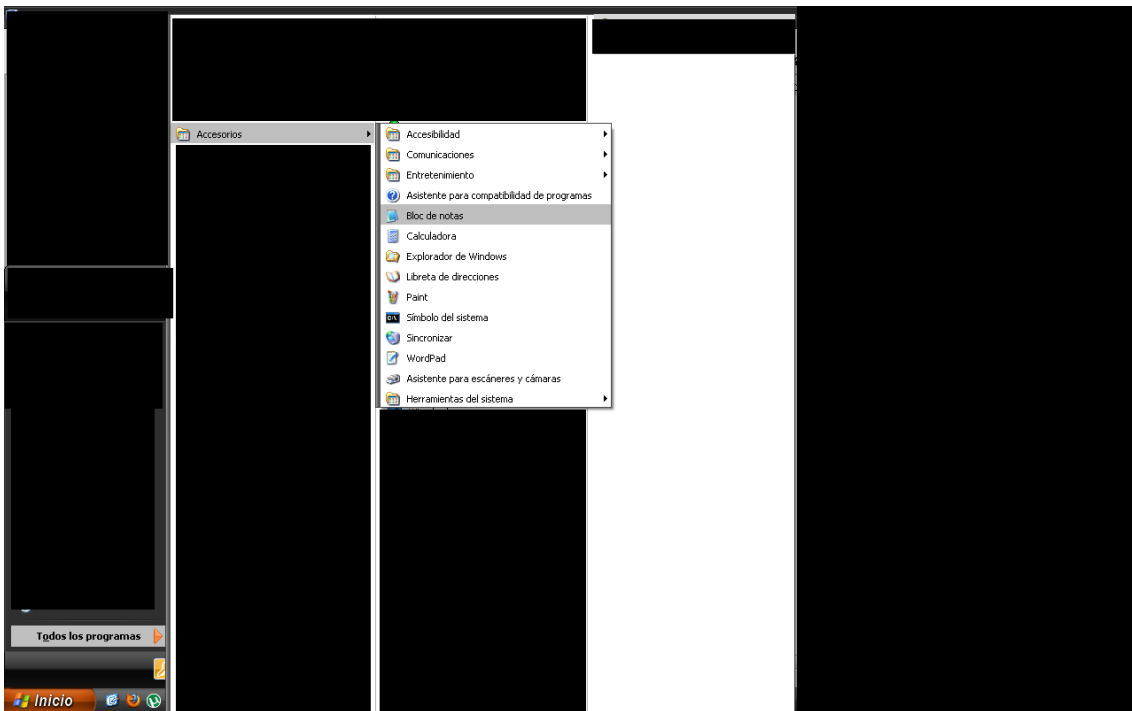
**MS-DOS** (siglas de *MicroSoft Disk Operating System, Sistema operativo de disco de Microsoft*) es un sistema operativo basado en X86. MS-DOS nació en 1981 al ser encargada Microsoft de producir un sistema operativo para la gama de computadores personales IBM PC de IBM. Rápidamente Microsoft compró los derechos de **QDOS** también conocido como 86-DOS, de Seattle Computer Products, y comenzó a trabajar en las modificaciones para poder cumplir con los requerimientos de IBM. La primera edición, MS-DOS 1.0, fue presentada en 1982. La versión incluida en los PC's de IBM fue conocida como PC DOS. Aunque MS-DOS y PC-DOS fueron desarrollados por Microsoft e IBM en paralelo, los dos productos se separaron con el tiempo.

## Como Crear Un Archivo .BAT

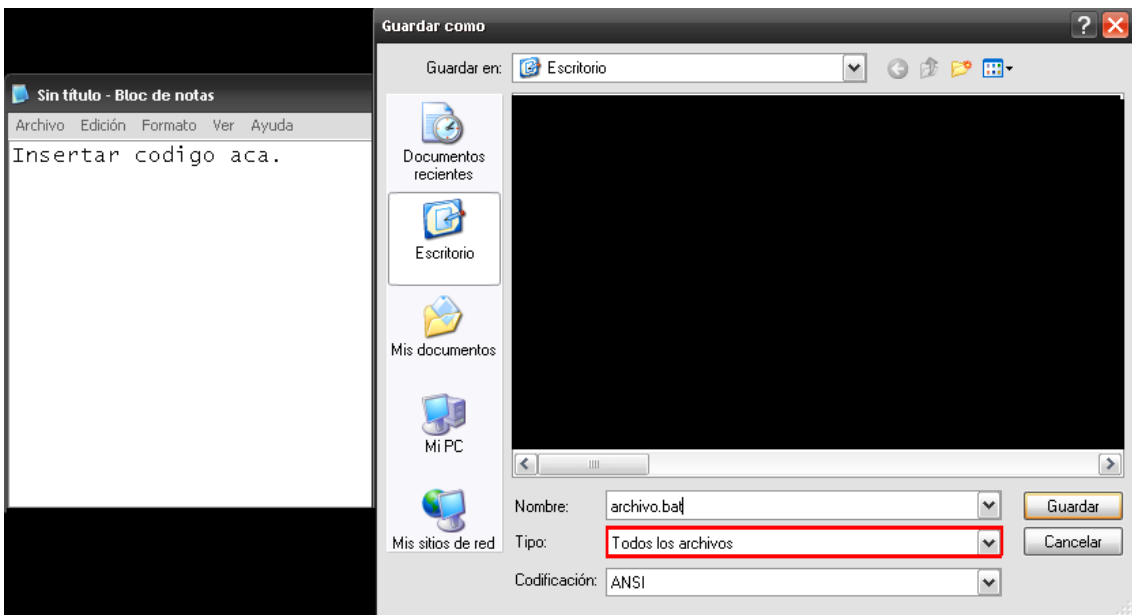
Para crear un archivo bat primero abrimos el notepad (o block de notas) También se puede usar Notepad++ pero lo veremos en la próxima entrega)



Sino también: Inicio>Todos los programas>Accesorios>Bloc de notas



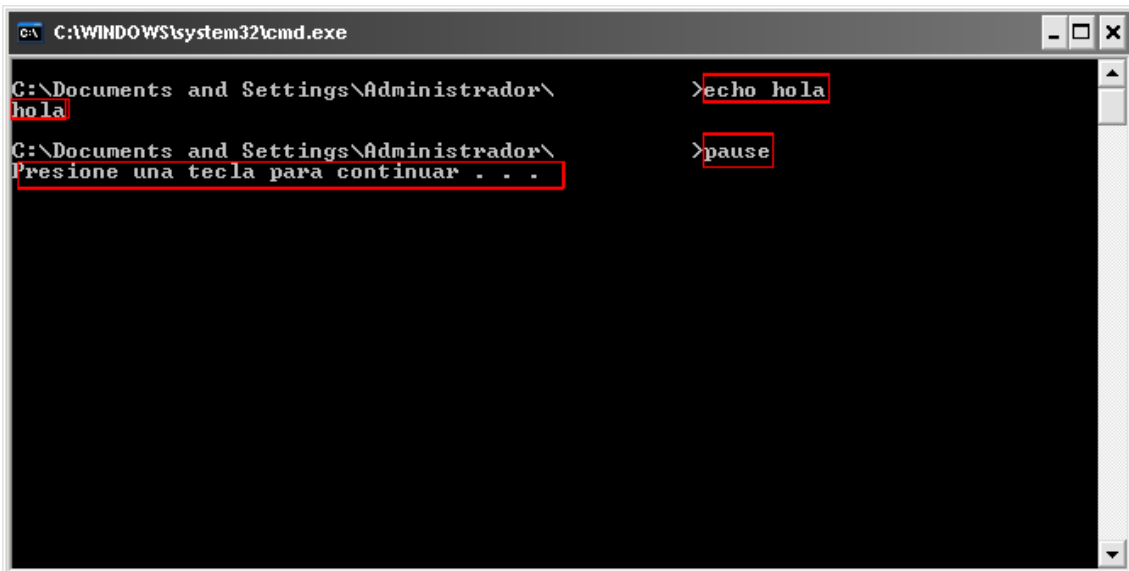
Al abrir el block de notas escribimos el código y lo guardamos con extensión .bat (fijarse que diga todos los archivos)



Y ahí tenemos hecho el archivo.bat.

## @Echo off/on

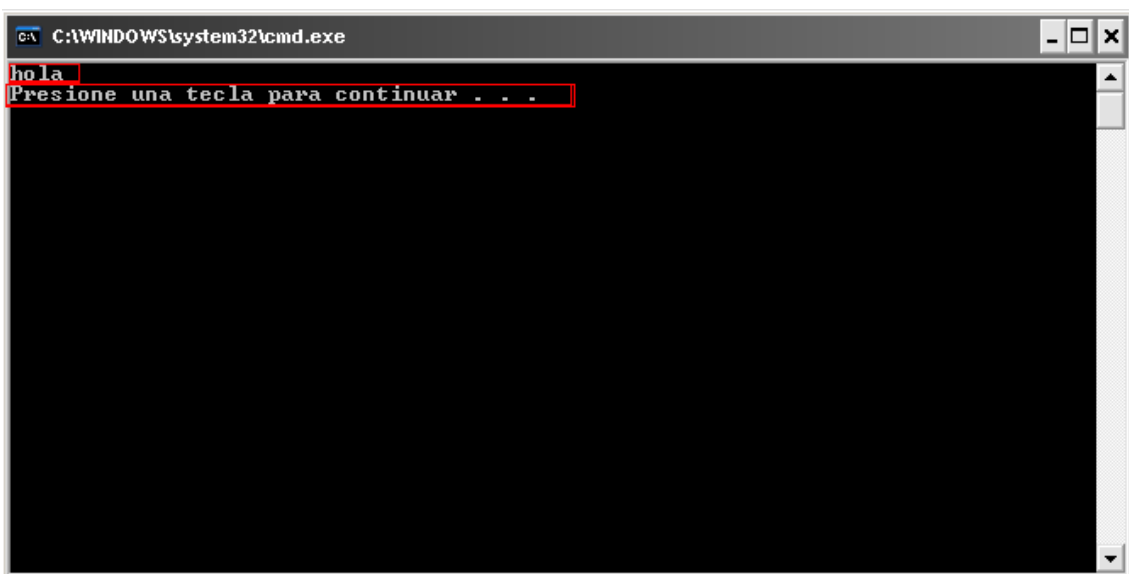
El @echo off/on es para determinar si queremos mostrar los comandos que escribimos (independientemente del output del comando) Para ser mas claros les mostrare dos fotos y va al principio de todo código.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrador\ >echo hola
hola
C:\Documents and Settings\Administrador\ >pause
Presione una tecla para continuar . . .
```

En esta foto se puede ver el input (comando ingresado) que seria echo hola y el output (el resultado del comando) que seria solo mostrar hola. Lo mismo para pause (es el input) y el output es una pausa que se expresa con "Presione una tecla para continuar..."

Esto seria con @echo off



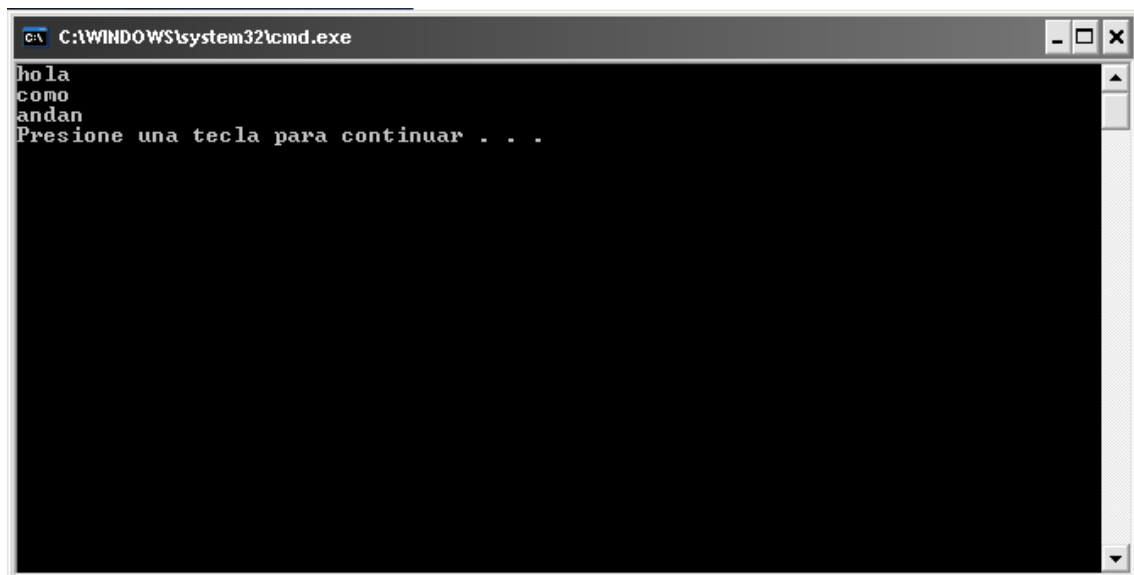
```
C:\WINDOWS\system32\cmd.exe
hola
Presione una tecla para continuar . . .
```

El código que use para ese .Bat fue:

1. @echo off
  2. Echo hola
  3. Pause
  4. Exit.
- 

## Echo/Echo.

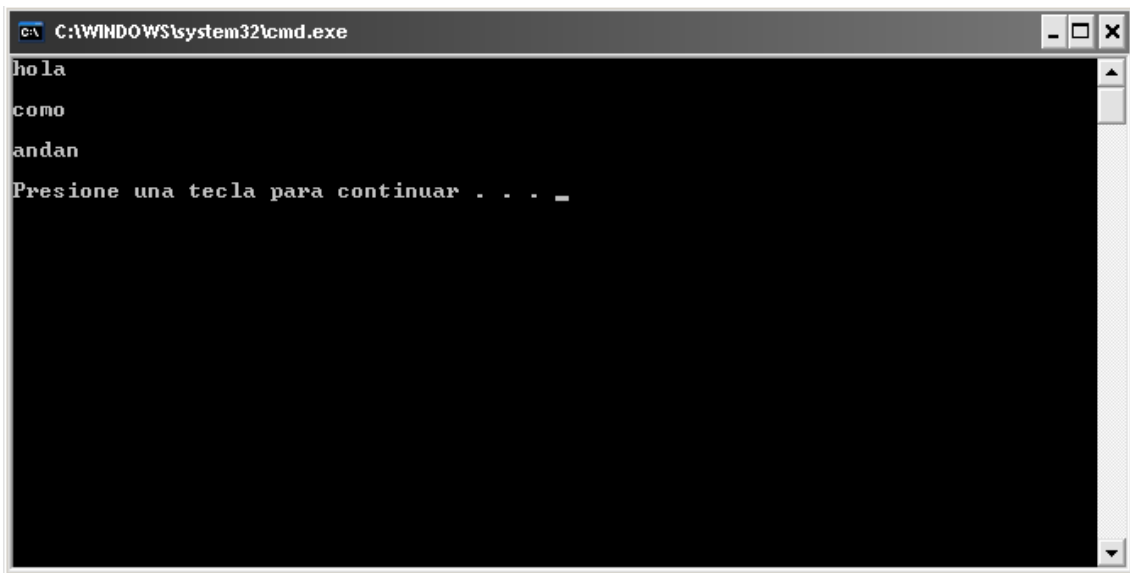
Como ya saben, el Echo imprime lo que escribimos. Si escribimos: Echo hola mundo entonces imprimirá: Hola mundo.



El código fue simplemente:

1. @echo off
  2. Echo hola
  3. Echo como
  4. Echo andan
  5. Pause
  6. Exit
- 

El echo. Sirve para dejar una línea en blanco (en este caso es color negro pero se entiende lo que quiero decir) Acá una foto para ilustrarlo.



```
C:\WINDOWS\system32\cmd.exe
ho la
como
andan
Presione una tecla para continuar . . . _
```

Lo único que hice fue agregar un echo. Entre cada línea.

### Código:


1. `@echo` off
2. `Echo` hola
3. `Echo`.
4. `Echo` como
5. `Echo`.
6. `Echo` andan
7. `Echo`.
8. `Pause`
9. `Exit`

### Goto

Bueno acá veremos los comandos GOTO/ELSE. Vamos a mostrarlo para que vean que tal es.

1. `@echo` off
2. `Echo` hola
3. `:bucle`
4. `Echo` como andas
5. `Goto` bucle

Acá lo que hace el código es imprimir hola, después le ponemos un ":" para definir un label. Entonces cuando ponemos goto bucle es que estamos yendo al label :bucle entonces así imprimiría el "como andas" infinitas veces. Otras funciones de esto se pueden ver también en el comando IF que lo veremos después.

A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window displays a vertical list of the text "ho la" repeated 20 times. The text is in a monospaced font and is left-aligned. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

## Comandos:

1. `@echo off`
2. `:bucle`
3. `Echo hola`
4. `Goto bucle`

## Pause

Bueno el comando PAUSE hace exactamente lo que dice el nombre, hace una pausa que se puede quitar apretando cualquier tecla. Para dejarlo más claro les paso una foto

```
C:\WINDOWS\system32\cmd.exe
hola como andan?
aca esta el pause
Presione una tecla para continuar . . .
ahi aprete cualquier tecla y el codigo sigue
Presione una tecla para continuar . . . _
```

## Código:

1. `@echo` off
2. `echo` hola como andan?
3. `echo` acá esta el `pause`
4. `Pause`
5. `echo` ahí apreté cualquier tecla y el código sigue
6. `pause`

Para hacer que desaparezca el “presione cualquier tecla para continuar” le tenemos que agregar `> nul` al `pause`. Sería algo así:

1. `@echo` off
2. `echo` hola como andan?
3. `echo` acá esta el `pause`
4. `Pause` > `nul`
5. `echo` ahí apreté cualquier tecla y el código sigue
6. `Pause` > `nul`

## DIR

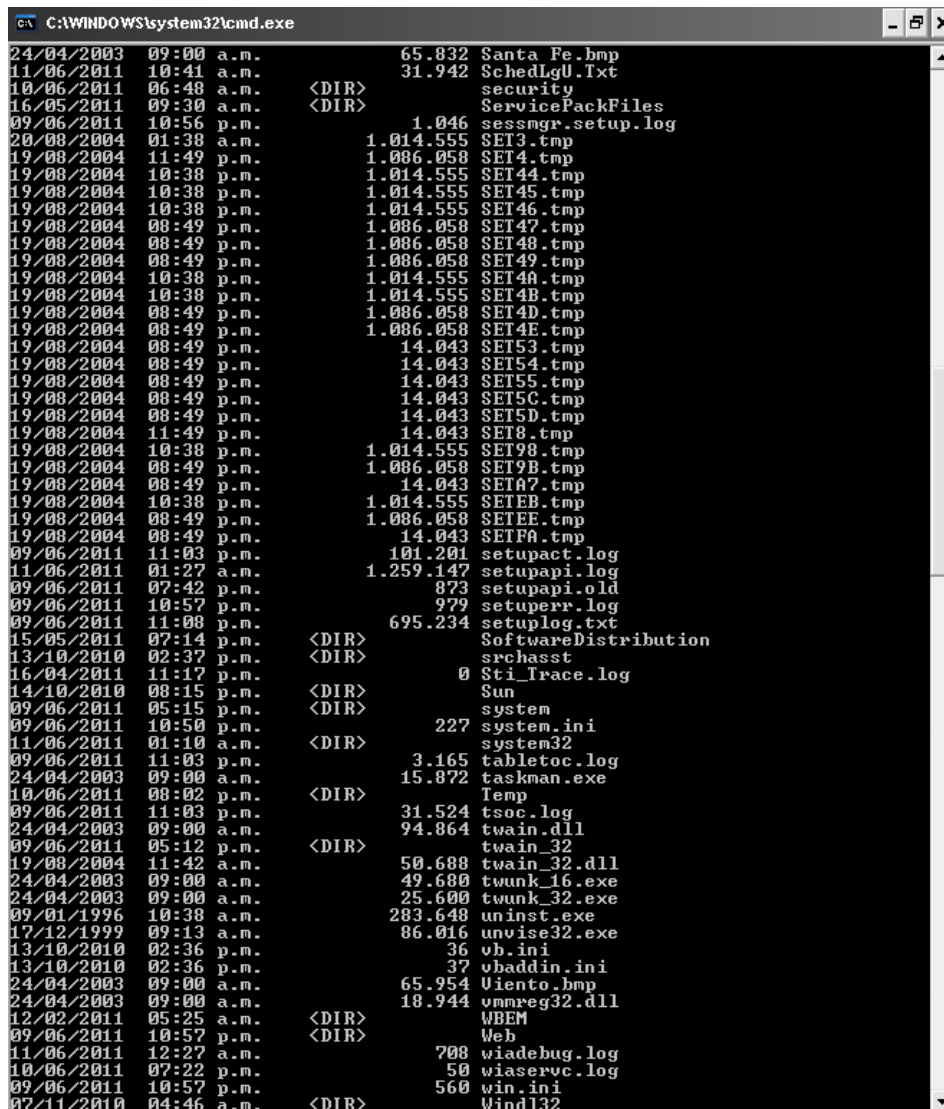
El comando `DIR` enlista todos los archivos que tengamos en el directorio especificado. Por ejemplo para cambiar un directorio hacemos `CD`.

1. `@echo` off
2. `CD` C:\Windows
3. `Pause`
4. `Exit`



Eso haría que en vez de estar en C:\Windows\System32\cmd.exe este en C:\Windows.

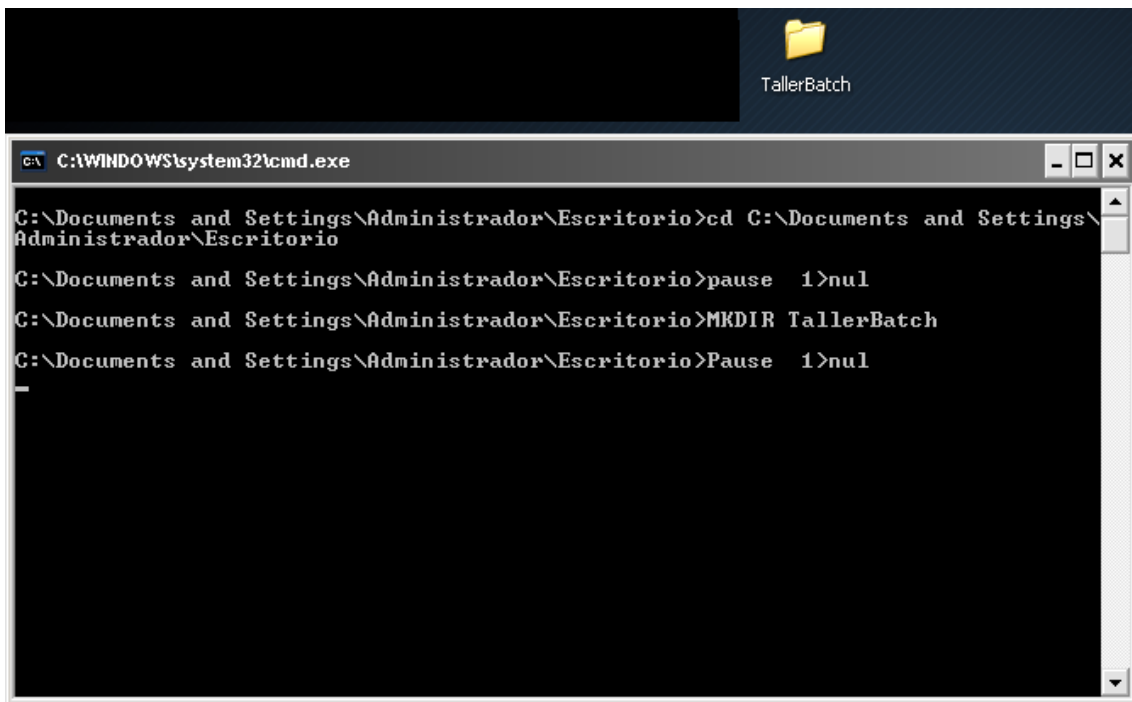
Para enlistar los archivos y directorio damos el comando DIR. Acá se ve ilustrado lo que digo.



```
C:\WINDOWS\system32\cmd.exe
24/04/2003 09:00 a.m.      65.832 Santa Fe.bmp
11/06/2011 10:41 a.m.      31.942 SchedLgU.Txt
10/06/2011 06:48 a.m.      <DIR> security
16/05/2011 09:30 a.m.      <DIR> ServicePackFiles
09/06/2011 10:56 p.m.          1.046 sessmgr.setup.log
20/08/2004 01:38 a.m.      1.014.555 SET3.tmp
19/08/2004 11:49 p.m.      1.086.058 SET4.tmp
19/08/2004 10:38 p.m.      1.014.555 SET44.tmp
19/08/2004 10:38 p.m.      1.014.555 SET45.tmp
19/08/2004 10:38 p.m.      1.014.555 SET46.tmp
19/08/2004 08:49 p.m.      1.086.058 SET47.tmp
19/08/2004 08:49 p.m.      1.086.058 SET48.tmp
19/08/2004 08:49 p.m.      1.086.058 SET49.tmp
19/08/2004 10:38 p.m.      1.014.555 SET4A.tmp
19/08/2004 10:38 p.m.      1.014.555 SET4B.tmp
19/08/2004 08:49 p.m.      1.086.058 SET4D.tmp
19/08/2004 08:49 p.m.      1.086.058 SET4E.tmp
19/08/2004 08:49 p.m.          14.043 SET53.tmp
19/08/2004 08:49 p.m.          14.043 SET54.tmp
19/08/2004 08:49 p.m.          14.043 SET55.tmp
19/08/2004 08:49 p.m.          14.043 SET5C.tmp
19/08/2004 08:49 p.m.          14.043 SET5D.tmp
19/08/2004 11:49 p.m.          14.043 SET8.tmp
19/08/2004 10:38 p.m.      1.014.555 SET98.tmp
19/08/2004 08:49 p.m.      1.086.058 SET9B.tmp
19/08/2004 08:49 p.m.          14.043 SETA7.tmp
19/08/2004 10:38 p.m.      1.014.555 SETEB.tmp
19/08/2004 08:49 p.m.      1.086.058 SETEE.tmp
19/08/2004 08:49 p.m.          14.043 SETFA.tmp
09/06/2011 11:03 p.m.      101.201 setupact.log
11/06/2011 01:27 a.m.      1.259.147 setupapi.log
09/06/2011 07:42 p.m.          873 setupapi.old
09/06/2011 10:57 p.m.          979 setuperr.log
09/06/2011 11:08 p.m.      695.234 setuplog.txt
15/05/2011 07:14 p.m.      <DIR> SoftwareDistribution
13/10/2010 02:37 p.m.      <DIR> srchasst
16/04/2011 11:17 p.m.          0 Sti_Trace.log
14/10/2010 08:15 p.m.      <DIR> Sun
09/06/2011 05:15 p.m.      <DIR> system
09/06/2011 10:50 p.m.          227 system.ini
11/06/2011 01:10 a.m.      <DIR> system32
09/06/2011 11:03 p.m.          3.165 tabletoc.log
24/04/2003 09:00 a.m.      15.872 taskman.exe
10/06/2011 08:02 p.m.      <DIR> Temp
09/06/2011 11:03 p.m.      31.524 tsoc.log
24/04/2003 09:00 a.m.      94.864 twain.dll
09/06/2011 05:12 p.m.      <DIR> twain_32
19/08/2004 11:42 a.m.      50.688 twain_32.dll
24/04/2003 09:00 a.m.      49.680 twunk_16.exe
24/04/2003 09:00 a.m.      25.600 twunk_32.exe
09/01/1996 10:38 a.m.      283.648 uninst.exe
17/12/1999 09:13 a.m.      86.016 unvise32.exe
13/10/2010 02:36 p.m.          36 vb.ini
13/10/2010 02:36 p.m.          37 vbaddin.ini
24/04/2003 09:00 a.m.      65.954 Viento.bmp
24/04/2003 09:00 a.m.      18.944 vmvreg32.dll
12/02/2011 05:25 a.m.      <DIR> Wbem
09/06/2011 10:57 p.m.      <DIR> Web
11/06/2011 12:27 a.m.          708 wiadefbug.log
10/06/2011 07:22 p.m.          50 wiaservc.log
09/06/2011 10:57 p.m.          560 win.ini
07/11/2010 04:46 a.m.      <DIR> Wind132
```

## MKDIR

Lo que hace el comando MKDIR es crear un directorio en una ruta especificada. Ponemos CD y elegimos el directorio (Ej.: CD C:\Documents and Settings\Administrador\Escritorio) y ahí ponemos MKDIR Hola. Eso nos crearía una carpeta nueva llamada Hola. Una foto.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrador\Escritorio>cd C:\Documents and Settings\Administrador\Escritorio
C:\Documents and Settings\Administrador\Escritorio>pause 1>nul
C:\Documents and Settings\Administrador\Escritorio>MKDIR TallerBatch
C:\Documents and Settings\Administrador\Escritorio>Pause 1>nul
_
```

Nota: Acá puse @echo on para que vean los comando tipeados.

1. @echo on
2. CD C:\Documents and Settings\Administrador\Escritorio
3. pause > nul
4. MKDIR TallerBatch
5. Pause > nul
6. Exit

## REM

El comando REM es una abreviación de la palabra en ingles “Remak” que es para marcar algo. Lo que quiere decir es que sirve para hacer comentarios en el código, es para encontrar partes mas fácilmente o para acordarse de ciertas cosas sin que afecte el código. Ejemplo:

1. @echo off
2. REM Acá empieza el bucle
3. :bucle
4. Echo Hola
5. Goto bucle

También para hacer comentarios se usa :: ACUERDENSE QUE SON DOS :: CUANDO SE USA UNO (:) ES PARA HACER UN LABEL.

## Ejemplo:

```
1. @echo off
2. ::Acá empieza el bucle
3. :bucle
4. Start
5. Goto bucle
```

---

## IF Básico/Else

Bueno el comando IF es uno de las importantes y poderosos en batch. Ya que tiene múltiples usos y su versatilidad es bastante grande. Se puede usar para chequear varias cosas, acá como es el básico veremos si existe un archivo con IF EXIST. Por ejemplo la sintaxis es así:

IF EXIST "X" "command" else "command"  
O también "command" & "command" &,etc...

Veamos un ejemplo:

```
1. @echo off
2. IF exist hola.txt echo hola & mkdir existe
3. Pause > nul
4. Exit
```

---

El código esta separado así

(IF exist) (hola.txt) (echo hola) (&) (mkdir existe)

IF exist se fija si existe ese archivo, el echo hola es la parte de "command" que seria que si existe tal archivo hacer tal comando y el & es para no solo hacer echo hola sino para agregar mas comandos como el MKDIR (que lo vimos antes). También se puede hacer esto.

```
1. @echo off
2. IF exist hola.txt Echo existe el archivo Else Echo no existe
3. Pause > nul
4. Exit
```

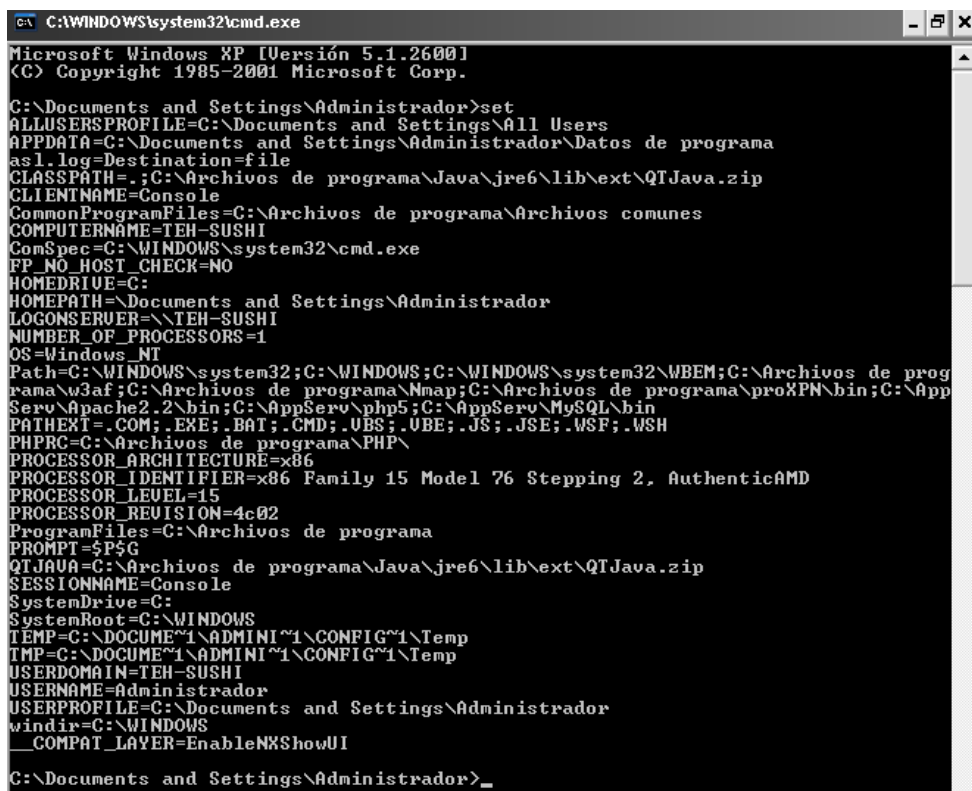
---

Se separaría así

(IF exist) (hola.txt) (Echo existe el archivo) (Else) (Echo no existe)  
Esto se utiliza para fijarse si existe el hola.txt entonces echo Existe el archivo y si no existe el archivo entonces echo No existe.

## SET

No, esto no es un social engineering tool :P El comando SET es para definir variables o para usar PATHs preestablecidos. Para ver los PATHs preestablecidos le damos a set en el CMD.



```
ca C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>set
ALLUSERSPROFILE=C:\Documents and Settings\All Users
APPDATA=C:\Documents and Settings\Administrador\Datos de programa
asl.log=Destination=file
CLASSPATH=.;C:\Archivos de programa\Java\jre6\lib\ext\QTJava.zip
CLIENTNAME=Console
CommonProgramFiles=C:\Archivos de programa\Archivos comunes
COMPUTERNAME=TEH-SUSHI
ComSpec=C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_CHECK=NO
HOMEDRIVE=C:
HOMEPATH=\Documents and Settings\Administrador
LOGONSERUER=\\TEH-SUSHI
NUMBER_OF_PROCESSORS=1
OS=Windows_NT
Path=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\system32\WBEM;C:\Archivos de programa\w3af;C:\Archivos de programa\Nmap;C:\Archivos de programa\proXPN\bin;C:\AppServ\Apache2.2\bin;C:\AppServ\php5;C:\AppServ\MySQL\bin
PATHEXT=.COM;.EXE;.BAT;.CMD;.UBS;.UBE;.JS;.JSE;.WSF;.WSH
PHPRC=C:\Archivos de programa\PHP\
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 15 Model 76 Stepping 2, AuthenticAMD
PROCESSOR_LEVEL=15
PROCESSOR_REVISION=4c02
ProgramFiles=C:\Archivos de programa
PROMPT=$P$G
QTJAVA=C:\Archivos de programa\Java\jre6\lib\ext\QTJava.zip
SESSIONNAME=Console
SystemDrive=C:
SystemRoot=C:\WINDOWS
TEMP=C:\DOCUME~1\ADMINI~1\CONFIG~1\Temp
TMP=C:\DOCUME~1\ADMINI~1\CONFIG~1\Temp
USERDOMAIN=TEH-SUSHI
USERNAME=Administrador
USERPROFILE=C:\Documents and Settings\Administrador
windir=C:\WINDOWS
__COMPAT_LAYER=EnableNXShowUI

C:\Documents and Settings\Administrador>_
```

Entonces esto sirve para usar BATs en otras PCs y para evitar errores.  
Ejemplo: Por ahí alguien tiene su directorio de Windows en D:\Windows entonces si ponemos CD C:\windows no nos funcionara, para evitar eso hacemos CD %windir% entonces nos dirigirá automáticamente en donde tengamos el directorio de Windows puesto.

El set también se puede hacer para establecer variables con esta sintaxis:

Set Nombre= valor

Un ejemplo seria:

1. @echo off
2. Set var1=hola
3. Echo %var1%
4. Pause
5. Exit

Ahí saldría el texto: Hola

## Shutdown

Este es un comando muy básico pero puede llegar a ser muy útil (yo personalmente lo uso todos los días)

Estas son las sintaxis que se pueden usar:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>shutdown /?
Uso: shutdown [-l | -s | -r | -a] [-f] [-m \\equipo] [-t xx] [-c "comentario"] [-d up:xx:yy]

Sin argumentos   Mostrar este mensaje (igual a -?)
-i               Mostrar interfaz GUI, debe ser la primera opción
-l              Cerrar sesión (no se puede usar con la opción -m)
-s              Apagar el equipo
-r              Apagar y reiniciar el equipo
-a             Anular el apagado de equipo
-m \\equipo    Equipo que se apagará/reiniciará/anulara
-t xx          Establecer el tiempo de espera de apagado en xx segundos
-c "comentario" Comentario de apagado (máximo, 127 caracteres)
-f             Fuerza el cierre de aplicaciones sin advertir
-d [u|p]:xx:yy Código de motivo de apagado
                u es el código de usuario
                p es el código de apagado planeado
                xx es el código primario del motivo de apagado
                (entero positivo menor que 256)
                yy es el código secundario del motivo de apagado
                (entero positivo menor que 65536)

C:\Documents and Settings\Administrador>
```

Entonces de ahí elegimos las sintaxis que queramos y las usamos.

Por ejemplo:

**Shutdown -s -f -t 230**

Eso haría que se apague (-s) que fuerce el cierre de aplicaciones (-f) y que tenga un contador de 230 segundos para apagarse (-t 230)

Esto nos viene bien por ejemplo si nos vamos a dormir y dejamos descargando algo y faltan 2 horas para descargarse, ponemos shutdown para que cuando termine de descargarse se apague la PC.

### **Shutdown -s f -t 7200**

El 7200 es porque son segundos (2 horas)