

HACK XCRACK
HACK XCRACK

echo

Manual de Batch

BY ALAN++



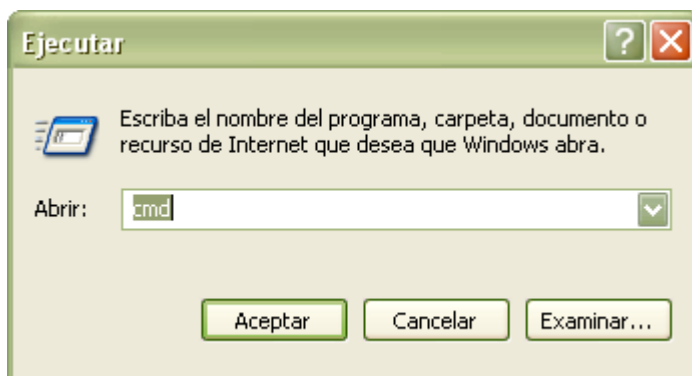
1ª Parte

Bueno gente la idea de este manual es que alguien que sepa al menos como prender la computadora sea capaz de crear un pequeño programita constructivo... y al terminar que ese programita se transforme en destructivo...

Pero empecemos por el principio... mmm... mejor no, hay un montón de manuales (o la misma Internet) que explica las nociones de la programación en batch, de que se trata y un montón de cosas aburridas, solo basta mencionar que trabajar con batch es trabajar con una shell de sistema, para conocer a este bichito tan raro solo hace falta ir a

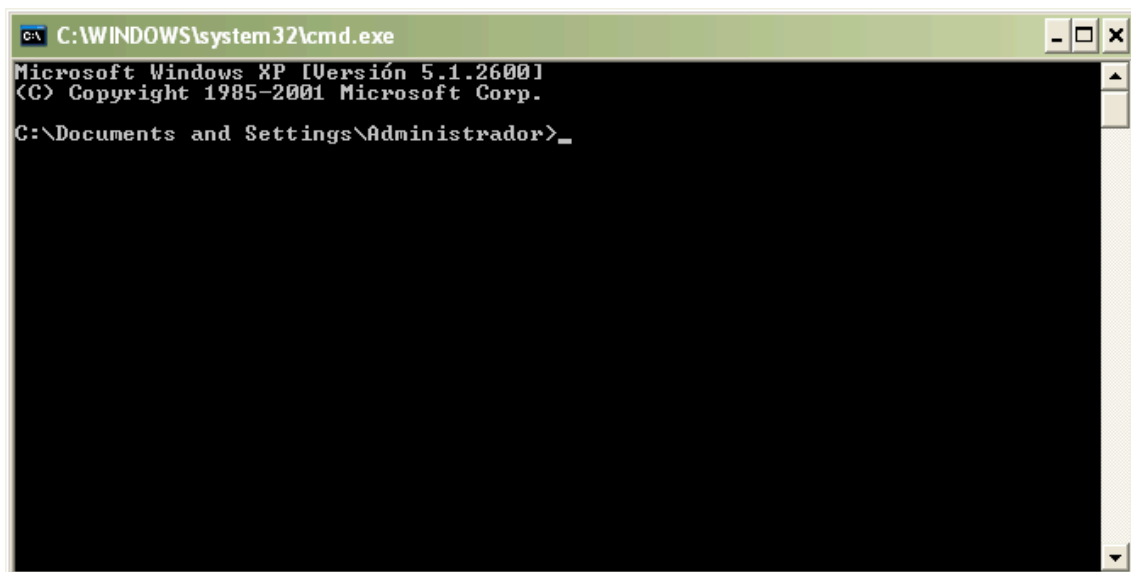


Inicio → Ejecutar → cmd (apretar enter) // Alternativa: Tecla Windos + R



Nos va a abrir la típica ventanita negra, esa ventanita es la shell (se cierra escribiendo "exit" y apretando enter), desde allí podemos hacer lo que queramos como si estuviéramos en el mismísimo windows, claro que

no con unas ventanas tan lindas (ni sus lindos iconos) pero la realidad es que la shell es muy útil, completa y la forma de "navegar" es mas eficiente... una vez que uno sabe como funciona.



Bueno no perdamos mas tiempo, ya tenemos una noción básica de lo que es la línea de comandos, ahora hace falta explicar que es un batch:

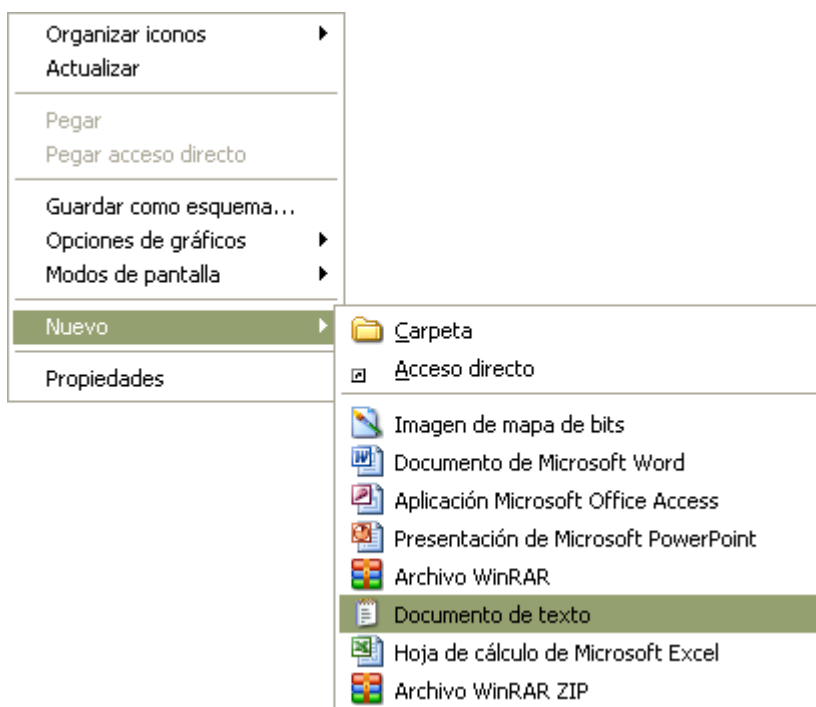
Un batch, o bat, es una serie de instrucciones que nosotros le vamos a dar a la computadora que ésta va a realizar con mucho gusto.

Como la ejecución es rápida no vamos a tardar en darnos cuenta si nos equivocamos así que pasaremos a aprender a programar cosillas nefastas

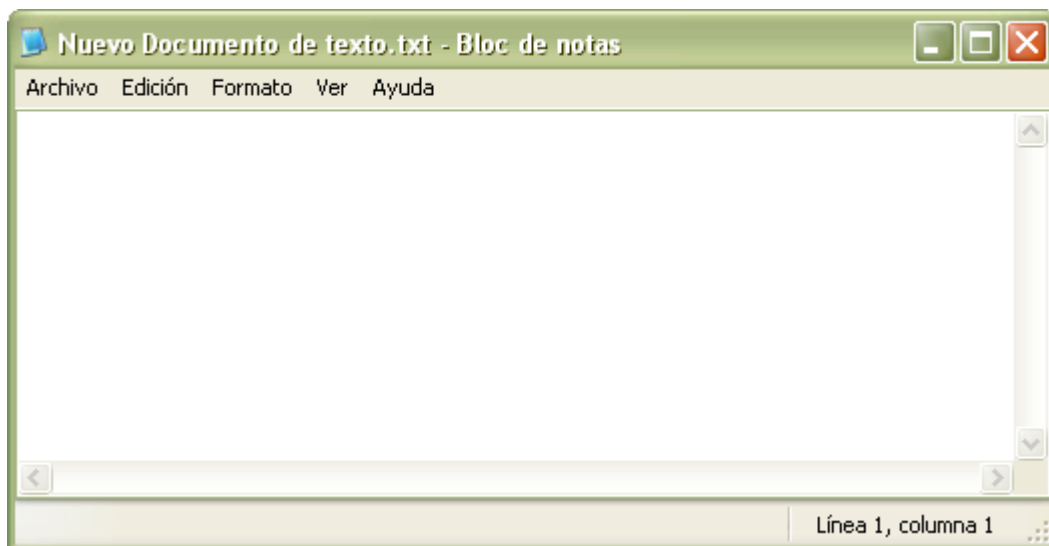
¿Cómo creamos un batch?

La respuesta es muy sencilla, creamos un documento de texto plano... ¿que como lo hacemos?

(digamos que estamos en el escritorio)
Botón derecho → Nuevo → Documento de texto



Como verán el documento de texto es una hoja en blanco donde ni siquiera se le puede dar formato (formato = forma) mas que a la fuente (fuente = letra), no es como el Word con todos esos botones y colores.



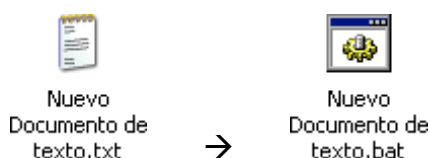
Una vez que creamos el “documento de texto” con una rutina adentro (tranquilo, ya vamos a llegar a la rutina) cambiamos la extensión del archivo a .bat...

¿Cómo cambiar la extensión del archivo?

Muy fácil, si el documento de texto se llama “nuevo documento de texto.txt”, apretamos sobre él...

Botón derecho → propiedades → ...

...donde dice “nuevo documento de texto.txt” lo cambiamos por “nuevo documento de texto.bat”, el truco es cambiar el archivo “xxxxx.txt” a “xxxxx.bat”, cuando lo hagas una ventanita te va a preguntar si estas seguro, como seguro no hay nada en la vida clickea “si”, vas a ver como cambia el icono de una hoja a una ventanita



Nota Importante: Si el Windows no esta configurado “apt” (a prueba de tontos) deberías ver la extensión sin problemas, pero en general Windows viene predeterminado para ser “apt”, para solucionarlo van a...

Inicio → panel de control → opciones de carpetas

Hacemos clic en la solapa “ver” y destildamos “ocultar las extensiones de archivo para tipos de archivos conocidos”, clickeamos aplicar y después aceptar, a partir de ahora vamos a ver todas las extensiones de los archivos...

Excel → .xls

Word → .doc

Música → .wma, .mp3

Videos → .avi, .mpeg

Batch → .bat (este nos interesa)

Texto plano → .txt (también este)

Entre muchas otras extensiones

Rutinas

Al fin llegamos a lo mas divertido... y también a lo mas largo y complicado, la rutina es aquello que yo deseo que haga el programa, hay que tener algo muy en claro... batch (o bat) son secuenciales, esto significa que hacen la rutina 1, después la 2, la tres y así sucesivamente... en criollo... digamos que mi rutina diaria es...

1. Despertarme
2. Desayunar
3. Trabajar
4. Entrar en hackxcrack
5. Almorzar
6. Estudiar
7. Cenar
8. Irme a dormir hasta despertarme

La computadora entiende una orden debajo de la otra por eso hay que tener cuidado de el orden en el cual escribo, digamos que yo pusiera...

1. Despertarme
2. Desayunar Trabajar
3. Entrar en hackxcrack
4. Irme a dormir hasta despertarme
5. Almorzar
6. Estudiar
7. Cenar

En este caso cuando llegue a la orden "irme a dormir hasta despertarme" me saltaría almorzar, estudiar y cenar, ya que estoy durmiendo y no puedo hacer las órdenes restantes... ¿se entiende?

Muchas veces vamos a tener pequeños errores que solo se van a deber a que la rutina es incorrecta o no se encamina por donde nosotros queremos, en estos casos mi recomendación (después amplio un truco) es que se preparen una taza de café y se alisten a pasar varios minutos observando porque demonios no quiere funcionar.

@echo off

Se que se ve raro pero es la línea principal de cualquier programa en batch, ¿Qué es lo que hace?

El “echo off” oculta los procesos que se llevan a cabo en el .bat y que nosotros no tenemos intención que se vean, y el @ se encarga de ocultar esa primera orden, el “echo off”, como ya dijimos antes, trabajar con el .bat es como trabajar en una shell (¿se

acuerdan? La ventanita negra), en realidad el .bat corre en la shell, por eso una vez que sepamos como funciona el .bat nos va a ser mas fácil entender también la shell, cuando sientan que a este tutorial ya lo saben y no poseen dudas quiten el “echo off” y van a ver la maquina detrás de la ilusión, por ahora es mejor ver solo al hombre volando

echo

Es la palabra que hay que poner antes de una frase para que esta aparezca en la shell (no se preocupen en un par de líneas van a comprender mejor)

pause

Detiene al programa con el mensaje “presione una tecla para continuar...”, no tiene mucha ciencia, detiene la rutina hasta que apretemos una tecla cualquiera, es muy importante dada la velocidad de este tipo de procesos, de lo contrario se iniciaría, ejecutaría y acabaría sin que nosotros nos diéramos cuenta

Ahora si, con estos tres primeros comandos vamos a crear nuestro primer programa en batch... ¡mua ja ja!

Abrimos un documento de texto que vamos a llamar “hola”, va a quedar “hola.txt”, lo abrimos y adentro escribimos

```
@echo off
Echo Mi primer práctico en hackxcrack
Pause
```

Lo guardamos y convertimos en batch cambiando la extensión de .txt a .bat, es decir que con el cambio va a pasar a llamarse hola.bat, cuando el icono de la libretita se convierte en ventanita hacemos doble click sobre el...

Vamos a ver una ventana negra que dice

*Mi primer practico en hackxcrack
presione una tecla para continuar...*

Cuando apretamos cualquier tecla el programa se cierra, si sucede esto significa que... ¡tu primer programa ha sido todo un éxito!

haciendo tarde y no queremos que te canses con repeticiones que en el peor de los casos puedes chequear volviendo unas paginas para atrás

Bien, de ahora en mas voy a obviar la transformación de .txt a .bat, yo les voy a dar la receta para completar el .txt y el resultado que van a obtener cuando ejecuten el .bat, ya se esta

El comando echo tiene varias funciones muy importantes, ya conocemos dos pero hay una tercera que nos va a servir de mucho

echo.

Exacto, con solo un punto al final, la función de este "echo." es crear un espacio entre líneas, es decir, ingresamos en el .txt lo siguiente...

```
@echo off
Echo.
Echo Un espacio
Echo.
Echo Otro espacio
Echo.
Pause
```

Cuando ejecutemos el .bat (ejecutar = doble click sobre la ventanita con extensión .bat) vamos a ver en la shell

Un espacio

Otro espacio

presione una tecla para continuar...

Si lo ves significa que tu segundo batch esta echo, y además aprendiste a usar el "echo."

pause>nul

Es igual que "pause" sin embargo este no escribe "presione una tecla para continuar..." aunque si queda a la espera de que presiones una tecla cualquiera, probémoslo...

Ingresamos en el txt. lo siguiente...

```
@echo off
Echo Esto ya resulta fácil
Pause>nul
```

Cuando ejecutemos el .bat vamos a ver...

Esto ya resulta fácil

¿Entienden? No esta ese mensaje de “presione una tecla para continuar...” y sin embargo espera a que presionemos una tecla cualquiera

Ahora comencemos, digamos que queremos hacer un programa para ofender a alguien... (el soft malicioso no tiene porque ser dañino, ja ja ja)... mas o menos va a verse así

*Quieres saber que pienso de vos?
presione una tecla para continuar...*

Acá el usuario va a tener que presionar una tecla

*Te odio...
Sabes que mas?, apreta una tecla*

El usuario tiene que apretar una tecla

Te detesto... pero pensándolo bien...

Quizás...

Quizás...

Apreta la “s” para saberlo

El usuario tiene que apretar una tecla (es indiferente que tecla sea)

Me prestas \$5?

La “p” es para putearme...

El usuario tiene que apretar una tecla (es indiferente que tecla sea) y termina. La solución está a continuación... pero intentalo, no es tan difícil...

```
@echo off
Echo.
Echo Queres saber que pienso de vos?
Pause
Echo.
Echo Te odio...
Echo Sabes que mas?, apreta una tecla
Echo.
Pause>nul
Echo Te detesto... pero pensándolo bien...
Echo.
Echo Quizás...
```



```
Echo.  
Echo Quizás...  
Echo.  
Echo Apreta la "s" para saberlo  
Pause>nul  
Echo.  
Echo Me prestas $5?  
Echo.  
Echo La "p" es para putearme...  
Pause>nul
```

Cls

Sirve para limpiar la pantalla, en el caso anterior, las líneas se agregaban, con este comando las líneas visualizadas hasta el momento desaparecen... probémoslo

```
@echo off  
Echo Primera línea  
Echo.  
Pause  
Cls  
Echo Segunda línea  
Echo.  
Pause
```

Como van a comprobar primero aparece...

Primer línea

presione una tecla para continuar...

Cuando nosotros presionemos una tecla nos va a borrar todo lo escrito y continua con el programa

Segunda línea

presione una tecla para continuar...

Title

Poniendo en nuestro código la palabra title seguido por un texto lo que vamos a lograr es que cambie el nombre de la ventanita, en su defecto aparece la ruta donde se desarrolla el batch (concepto que después ampliaremos)

★ judgeNow!



... continuará

Color

Esta función cambia el color de las letras y del fondo, la elección es hexadecimal (es decir 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f) y va de a pares...

Ejemplo: titulo de la ventana "hackxcrack" con color al mejor estilo Windows registro de error fatal (letras blancas, fondo azul)

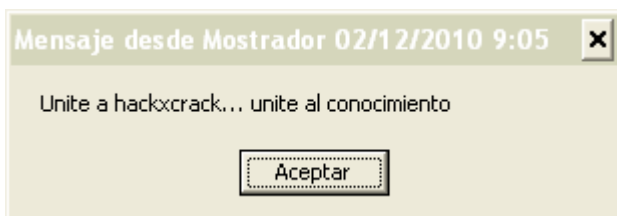
```
@echo off
Title hackxcrack
Color 1f
Echo Error fatal
Echo.
Echo Por favor reinicie su pc
Pause>nul
```

Msg *

El comando "msg *" es para dejar un mensaje desde el mostrador, una ventanita la cual nos da las únicas opciones de aceptar o cerrar

```
@echo off
Msg * Unite a hackxcrack... unite al conocimiento
```

Al ejecutarlo nos va a aparecer una ventanita con la misma leyenda, ya que no necesitamos que el programa se detenga no hace falta ingresar un "pause"



Start

Ahora empezamos con uno de los comandos mas importantes, start nos va a servir (por el momento) para abrir paginas de Internet, ejecutar programas y abrir archivos...

Digamos que queremos abrir una pagina de Internet, mas precisamente hackxcrack, la orden en el .txt seria

```
@echo off
Start www.hackxcrack.es
```

Una vez mas aclaro que como no me interesa ver la shell (el cuadradito negro que aprendimos a cambiar de color) así que no pongo el pause...

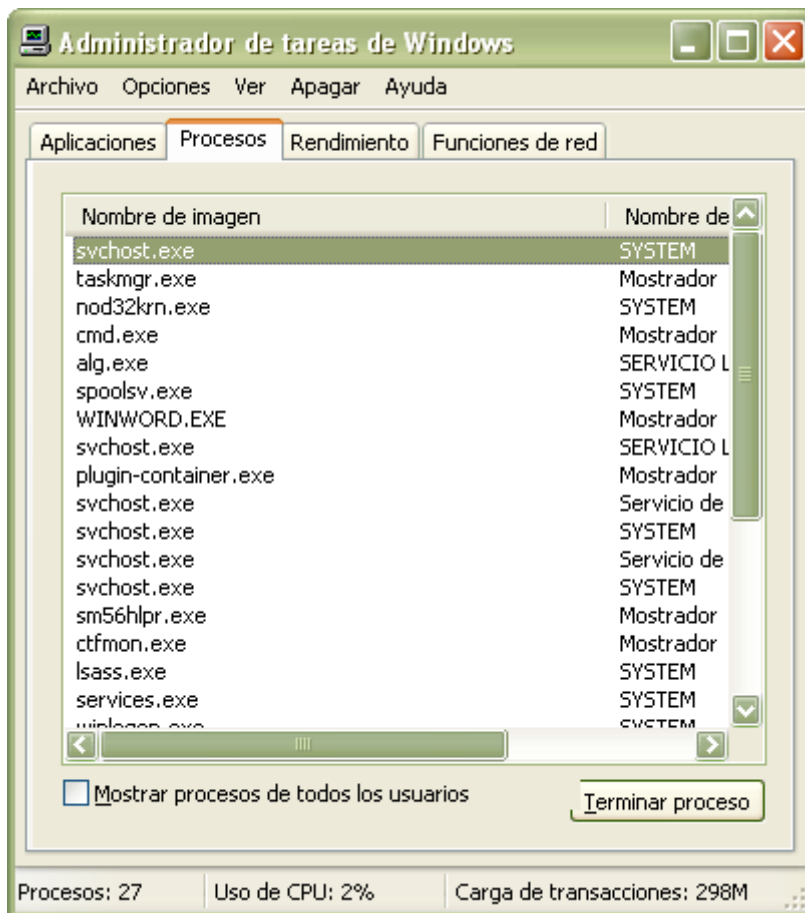
Ahora un .txt que habrá mi navegador de Internet

```
@echo off
Start firefox.exe
```

o "iexplorer.exe" dependiendo de cual tengan

¿Se dan cuenta? Con una simple orden podemos ejecutar un programa sin siquiera saber en que lugar de la computadora esta... para conocer el comando de cada software hay varias recetas, a mi criterio la mas simple es hacer correr el programa después...

Alt+Ctrl+Supr → Procesos



Ahí vamos a ver todo los procesos con su sintaxis para ejecutarlos con el comando start, es solo cuestión de localizar cual es

Otros ejemplos:

Pdf → acro32.exe, *Calculadora* → calc.exe, *Nod32* → nod32krn.exe, etc

Set

Este individuo es quien realiza las funciones de un programa en batch y quien además permite que el usuario interactúe con el... sin embargo nunca viene solo, siempre lo acompañan sus dos amigos `"/a"` y `"/p"`, todo esto es algo más abstracto, sin embargo ya a estas alturas no nos vamos a detener con una simple montaña

"Set" significa que a una variable le voy a entregar determinado valor

"Set/p" significa que le voy a pedir al usuario que ingrese el valor de una variable

"Set/a" significa que voy a hacer cálculos con variables

¿Qué es una variable?, es una concatenación de letras y/o dígitos al cual le asigno un valor... en criollo, letras juntas que representan algo (¿verdad que sonaba más complicado?), las variables en batch no deben estar definidas al comenzar el programa (algunos programas si lo requieren), se puede agregar una variable donde lo desee siempre y cuando sea anterior al momento de utilizarla

Digamos que queremos que la variable `"letras"` valga 6

```
@echo off
Set letras=6
```

Ahora le vamos no solo a pedir darle un valor a una variable, sino que además vamos a pedirle que nos muestre el valor de la misma

```
@echo off
Set letras=6
Echo %letras%
Pause>nul
```

Algo curioso que tener en cuenta, es necesario para llamar a la variable encerrar su nombre entre %, de lo contrario en este caso nos mostraría la palabra `"letras"`

En la pantalla solo vamos a ver

6

Cuando apretemos una tecla se cierra

Set/p

Ahora vamos a interactuar con el usuario pidiéndole a él que ingrese el valor de una variable, ¿Cómo lo hacemos?, de la siguiente forma, pero hay que tener en cuenta el formato en que escribimos el set

Set/p vil=Que nombre es el tuyo?=
vil

Separemoslo para comprenderlo de la mejor manera...

“Set/p” nos indica que la variable va a pedirle al usuario su valor

“vil” elegido al azar por mi es la variable la cual guardara el dato ingresado

“=” significa el final del nombre de la variable y el comienzo de el texto que va a aparecer en pantalla

“Que nombre es el tuyo?=” es la pregunta o texto que va a leer el usuario antes de ingresar el dato que pasara a tomar la variable, el = final es solo para la percepción del usuario, queda mas estético pero no tiene utilidad el agregarlo o no

El .txt seria de esta forma:

```
@echo off
Set/p vil=Que nombre es el tuyo?=
Echo.
Echo Tu nombre es %vil%, la verdad no es nada lindo
Pause
```

Cuando lo convirtamos en .bat vamos a ver...

Que nombre es el tuyo?=-

Y en vez de continuar (como hasta ahora lo hacían todos los comandos), va a detenerse hasta que introduzcamos un dato... el set/p como pueden ver ahora hace las veces de pause también

Al ingresar un texto (por ejemplo “Alan”) y apretar enter vamos a ver...

Que nombre es el tuyo?=-

*Tu nombre es Alan, la verdad no es nada lindo
presione una tecla para continuar...*

De la misma forma podemos pedirle al usuario que ingrese valores para distintas variables, el nombre de estas se lo damos nosotros a nuestro antojo, pero siempre es mejor un nombre que se relacione para minimizar errores

Set/a

Este nuevo amigo nuestro se encarga de la aritmética (cuentas) cuando yo quiero sumar, restar, multiplicar o dividir...

Nota importante: batch no trabaja con decimales por lo cual al intentar ingresar un valor con decimales va a decirnos “Falta el operador.” por no mandarnos a freír churros, a su vez cuando el resultado de con decimales solo se limita a redondear el numero y darnos un entero

La forma de utilizarlo es la siguiente, digamos que queremos sumar dos números y mostrarlo por pantalla

El .txt seria...

```
@echo off
Set/a suma=5+3
Echo %suma%
Echo.
Pause
```

Esto nos va a devolver por pantalla al ejecutar el batch

8

presione una tecla para continuar...

Pero esto no es todo y lo gracioso es que nosotros le podemos pedir al usuario de nuestro batch (combinando el “set/p” y el “set/a”) que inserte los datos a sumar, multiplicar, restar o dividir... hagamos todo juntos así los confundo un poco...

.txt

```
@echo off
Set/p ero=Ingrese el primer numero=
Cls
Set/p ndo=Ingrese el segundo numero=
Cls
Set/a suma=%ero%+%ndo%
Set/a rest=%ero%-%ndo%
Set/a mult=%ero%*%ndo%
Set/a divi=%ero%/%ndo%
Echo los números ingresados son %ero% y %ndo%
Echo.
Echo La suma entre ambos da %suma%
Echo.
Echo La resta del primero por el segundo da %rest%
Echo.
Echo La multiplicación da %mult%
Echo.
Echo La división del primero por el segunda da %divi%
Echo.
Pause
```

Lo que veríamos en la shell seria

Ingrese el primer numero=

Nosotros al ingresar el primer numero y dar enter (supongamos que ingreso el 60) va a borrar la primera frase (gracias al cls) y nos va a aparecer en pantalla

Ingrese el segundo numero=

El usuario va a elegir un segundo numero (en este caso 10) y al apretar enter el otro cls nos va a borrar la segunda frase y vamos a ver...

Los números ingresados son 60 y 10

La suma entre ambos da 70

La resta del primero por el segundo da 50

La multiplicación da 600

La división del primero por el segundo da 6

presione una tecla para continuar...

If

Mucha gente se pregunta que hubiera pasado si..., este "if" significa un si condicional en ingles, una elección a través de diversos parámetros, esto esta sonando muy complicado... el if sirve para elegir un camino en caso de que ocurra algo... así quedo bonito

El "if" funciona con su pandilla

LSS (menor / <)

GTR (mayor / >)

LEQ (menor o igual <=)

GEQ (mayor o igual >=)

EQU o == (igual / =)

NEQ (distinto / !=)

*En la segunda parte de cada paréntesis están los operadores usados en la mayoría de lenguajes un poco más avanzados, así ya os van sonando.

¿Cómo funciona?

Nosotros le vamos a preguntar si un valor es igual, menor, mayor, etc, y este nos va a responder en dos forma, que el valor cumpla con el requisito (correcto) o que no lo haga (incorrecto)

"If (variable, nombre o numero)(condición... ==, >, <)(variable, nombre o numero) (accion "dirección, valor, etc") else accion (dirección, valor, etc)"

Ufff!!!, se ve complicado pero con un ejemplo, algo sencillo (aclaro que el valor "correcto" e "incorrecto" pueden llevar el nombre que quieran)

```
@echo off
If 1==1 (echo correcto) else echo incorrecto
Pause>nul
```

¿se entiende la sintaxis?, este ejemplo siempre nos va a mostrar correcto ya que siempre 1 va a ser = a 1, si planteáramos el caso de que 2 sea igual a 1 nos va a mostrar incorrecto

Pero no solo podemos darle un valor fijo, podemos pedirle al usuario que inserte uno e inclusive ambos valores... ¿Cómo? Con el set/p

Nota Importante: a medida que vamos avanzando no solo vamos a aprender nuevos comandos sino que también vamos incitar a que interactúen entre ellos, el siguiente es un buen ejemplo

Vamos a realizar un pequeño juego en el cual nuestro usuario va a tener que adivinar en que numero estamos pensando

```
@echo off
Echo Adivina en que numero estoy pensando
Set/p num=Es mayor que 0 y menor que 10...=
Echo.
If %num%==6 (echo Genio... como lo sabias?) else echo No... no adivinaste
Echo.
Pause
```

En la pantalla vamos a ver

*Adivina en que numero estoy pensando
Es mayor que 0 y menor que 10...=*

En caso de escribir 6 nos va a aparecer la leyenda

Sos Genio... como lo sabias?

presione una tecla para continuar...

En caso de que el numero sea otro vamos a leer...

No... no adivinaste

presione una tecla para continuar...

¡Pufff! Cuantas cosas vamos aprendiendo, quiero dejar en claro desde ahora y antes de entrar en terreno mas turbulento que el limite te lo vas a poner vos y tu

imaginación, muchas cosas se te van a ocurrir así como a mi mientras busco el ejemplo mas sencillo para dejar en claro cada punto...

Continuemos

Goto

Ahora comienzan los comandos, por así llamarlos, mas peligrosos, en realidad aquellos que nos permiten hacer alguna que otra cosilla perversa, el goto es una de ellas por una sencilla razón y es que...

“goto” seria la traducción literal de “ir a”, lo que hace este comando es enviarnos a una parte determinada del código que estamos creando, la peculiaridad del comando es que nos va a enviar al lugar donde le demos la orden, solo que con dos puntitos antes

Goto aca → :aca

¿difícil?... hagamos un programita sencillo

```
@echo off
Echo Comenzamos
Goto segu
:terc
Echo Tercera línea
Goto Cuar
:segu
Echo Segunda línea
Goto terc
:cuar
Echo Cuarta línea
Pause
```

Cuando lo pasemos a batch vamos a ver en la shell

Comenzamos

Segunda línea

Tercera línea

Cuarta línea

presione una tecla para continuar...

A que ahora parece una tontería la explicación que voy a dar

Goto segu → :segu

Goto terc → :terc

Goto cuar → :cuar

así de simple, hay que tener bien en cuenta el orden y el no abusar (por nuestra propia salud) de este comando, pero además ahora vamos a tener que ingresar un comando de cierre, como ya hemos visto el batch sigue la rutina de arriba hacia abajo, a menos que el "goto" nos envíe a una nueva dirección, sin embargo a veces es necesario que se termine en cierto punto el programa, para esto (y es una sana costumbre usarlo siempre) existe el "exit"