Entramos en la clase número 59. Estamos lejos de nuestra primera clase. ¿Recuerdan cuando entraron sin saber nada de nada? ¿Recuerdan cuando llegaron y no sabían qué era un cifrado simétrico, una compuerta o qué era el registro de Windows? Ya estamos terminando este año 2014, y me parece que fuera hace muy poco que empecé el curso en Taringa! O que eran 100 alumnos registrados y estaba subiendo mi clase número 5 para que pueda estar online. Hoy ya estoy trabajando en la nueva página, con más de 3000 alumnos registrados y muchos logros más que no los tenía previstos. Gracias enormemente a todos por la oportunidad de vivir esto. Es, sobre todo, increible. El mundo de la seguridad informática esta brindándonos más que nunca la posibilidad de integrarnos como una familia y en la calidez de un abrazo de sabiduría nos entregamos los unos a los otros libremente y sin límites. Se convierte de a poco en una forma de vida. Espero que siempre sigan adelante conmigo o con otras personas :)

Ya estuvimos viendo qué corcho era la consola de comandos de Windows y cuáles eran sus comandos, que sin duda llevan mucho tiempo siendo desarrollados. Hoy veremos **Batch**.

"¿Batch? ¿Johan Sebastian? ¡Como el tipo de música clásica!"

No, Manolo por favor no digas esas cosas. **Batch**. Se dice tal y como suena "batch". Se trata de un **tipo de archivo** que tiene **texto** dentro.

Lo que pasa es que el texto **no** es del todo **normal**, sino que es una serie de **comandos** que se **ejecutan**.

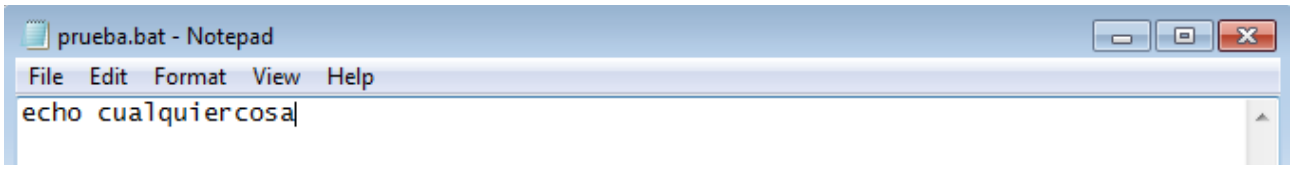
"Yo sé, yo sé. Seguro son comandos como los que vimos."

Claro, **exactamente**. Se ejecutarán de manera **estructurada** -es decir, del primero al último en orden-, como vimos que se trataba en C, pero con **comandos** del **símbolo de sistema**.
La **extensión** del archivo es **.bat**. La idea de este archivo es que podamos **automatizar tareas** y que no tengamos que hacer una y otra vez lo mismo. Por ésto, no sólo tenemos los comandos normales, también tenemos otros que son para usar aquí **exclusivamente**.

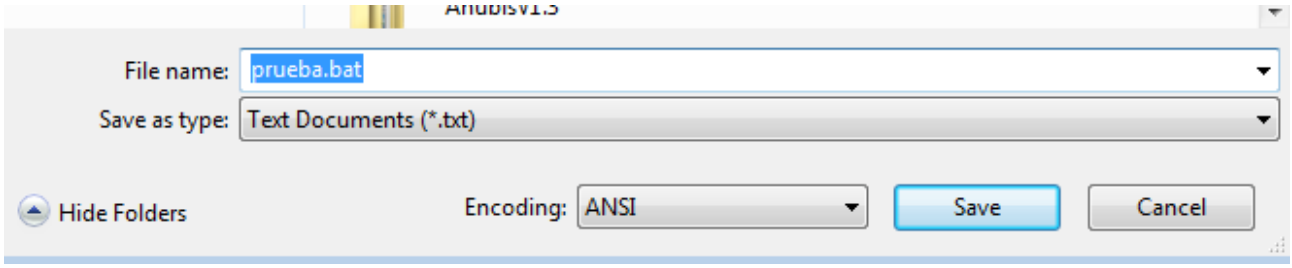
"No otra vez la lista de comandos..."

Vamos Manolo. Necesito que veamos esto y luego haremos **laboratorios** para que practiques :)
Pero antes que eso, vamos a ver **cómo** es que hacemos un **archivo** de estas características. Será fácil para grandes futuros hackers como ustedes.

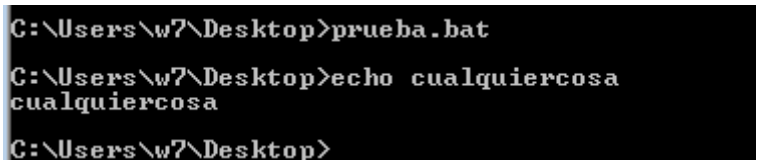
Abramos un **notepad** y escribamos dentro algún comando fácil. Por ejemplo, aquello que imprime algo en pantalla se hace con el comando ***echo***. Recuerden que ésto se puede hacer sin el batch, directamente en el símbolo de sistema. Escribamos: "**echo cualquiercosa**".



Y ahora en File -> Save As, lo guardamos con el nombre que ustedes quieran y la extensión ".bat".

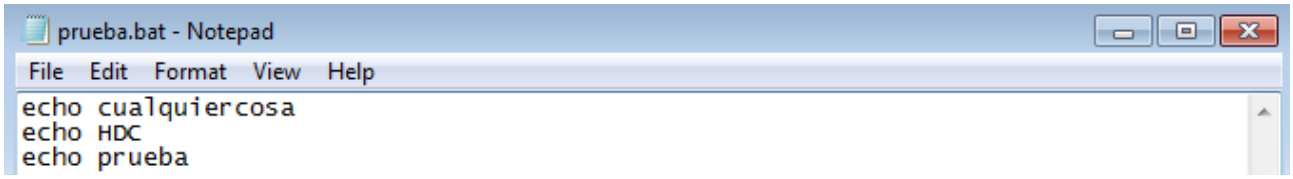


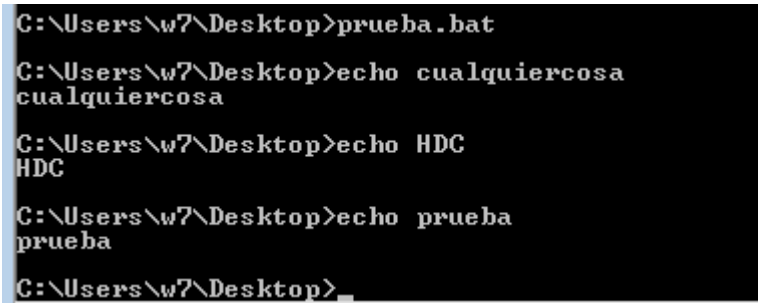
Ahora abran una consola de comandos (ejecutar -> cmd, ya creo que lo saben hacer de memoria :)), vamos hasta donde guardamos el archivo y lo **ejecutamos** simplemente poniendo su **nombre** completo con la **extensión**. Si el nombre lleva **espacios**, hay que encerrar todo entre **comillas**.
Vean que para poder ejecutar el código **no necesita compilación** como el lenguaje C.



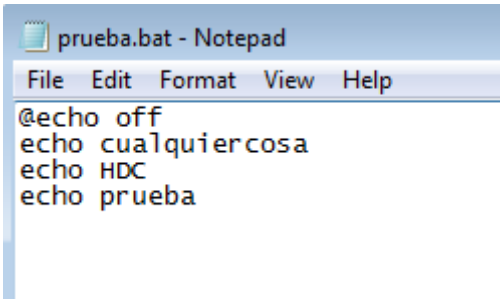
Si ven allí, es tan igual a hacerlo de manera normal que el ejecutable simplemente pone el comando y presiona el enter por nosotros (a una **velocidad inhumana**). Si ponemos más líneas, lo podemos

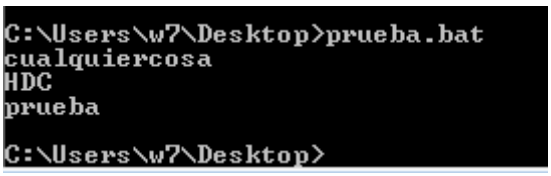
ver mejor. Si cerraron el notepad de donde lo estaban editando al batch, simplemente click derecho -> editar.





Pero nosotros no queremos que aparezcan los "echo cualquiercosa" como que realmente estamos ejecutando el comando. Sino que simplemente aparezcan los resultados de todo eso, como cuando ejecutábamos código C. Claramente no queremos que muestre el código mientras va realizando todo sino sería mucha confusión e información innecesaria. Vamos a usar ***@echo off*** al principio del archivo.

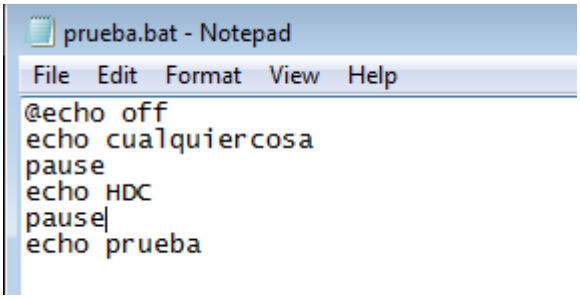


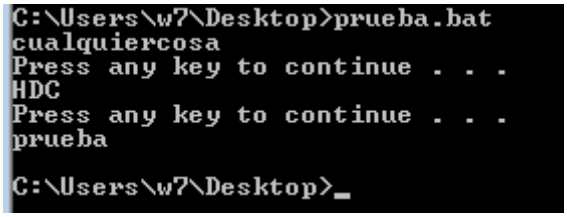


Ya tenemos algo un poco más lindo de ver y ejecutar.

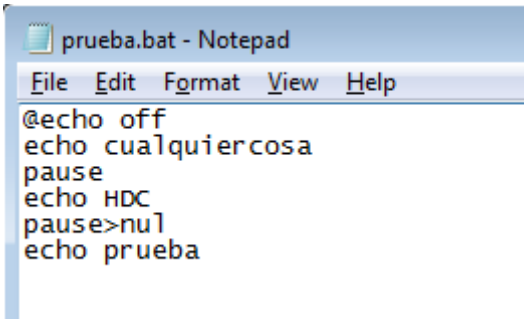
"Es decir. Todo va muy rápido y yo no voy a poder ver nada. Dime que hay algo así como una pausa."

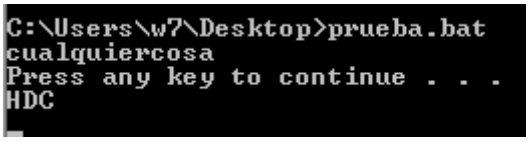
Claro. Exactamente eso, el comando ***pause***. Donde lo pongas, cuando la ejecución llegue a esa línea el programa **esperará** a que presiones una **tecla** para **continuar**. Ya verás que en el momento, te darás cuenta.
Yo lo pongo justo debajo del primero echo y del segundo echo.



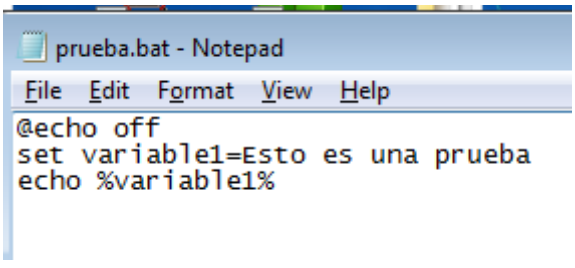


También se puede lograr hacer el pause pero que la salida no salga por aquí. Ésto se hace con ***pause>nul*** (es como querer meter la salida de ésto en un archivo pero con "nul" lo mandamos a la nada misma).





Pero ésto es obviamente lo básico del tema. Necesitamos guardar información en algún lugar. Es decir, variables. Nos ayudará ***set***.

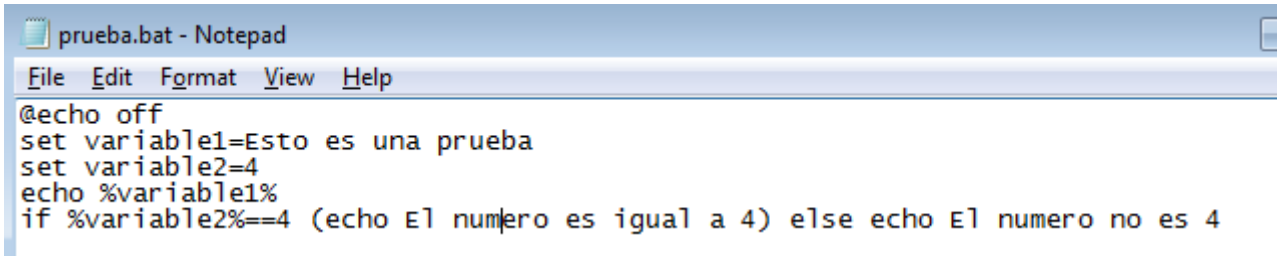


Allí ven cómo se hace para declarar una variable con ***set nombreVariable*** = ***valor***, y cómo se hace para usarla. Simplemente con el nombre entre **%** ya le damos a entender que corresponde a una variable.

"A todo esto me da que Batch es un lenguaje de programación. Soy un genio. Manolo domina 2 lenguajes."

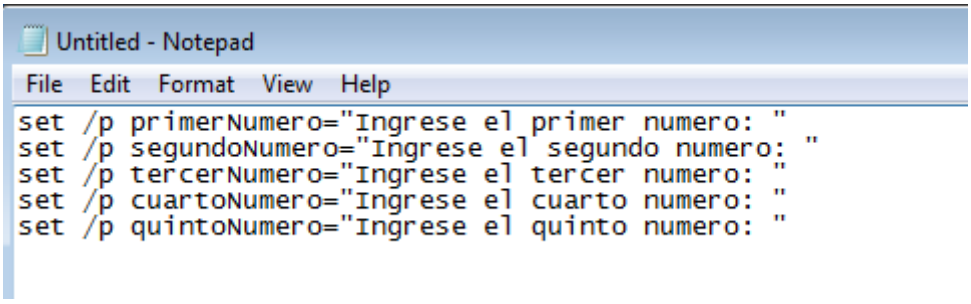
Ni uno, ni dos. Primero que C viste lo básico y segundo que batch no es un lenguaje de programación sino un lenguaje de **script**. Es decir que no se debe compilar y simplemente el sistema lo interpreta y ejecuta.

Pero igualmente, tiene cosas parecidas. Una de las cosas que comparte es la posibilidad de hacer una decisión dependiendo de que se cumpla una **condición** o no: el famoso ***if***.
En este caso, la sintaxis es ***if (condicion) (si es correcto hacer esto) (else) (si es incorrecto hacer esto)***.

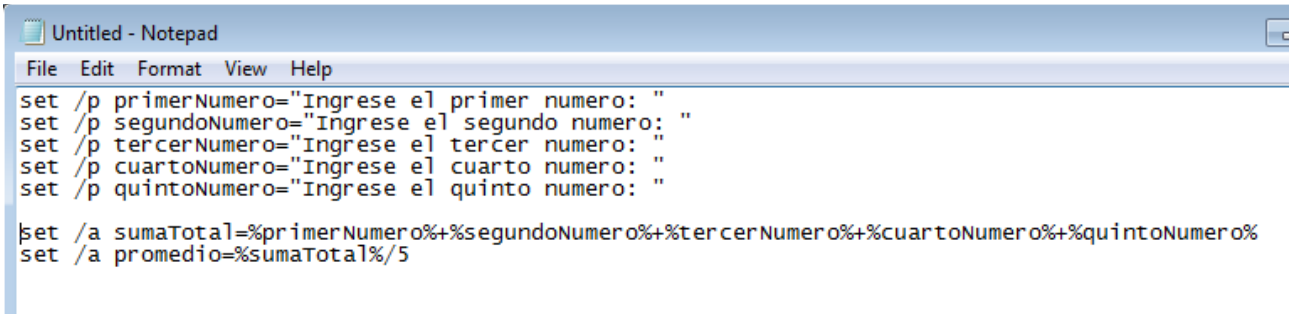


Vamos a hacer algo un poco más útil. Voy a crear un archivo de texto, que contenga el resultado de 10 notas sumadas, el promedio, en otra línea y si el promedio es mayor a 7 que diga que está aprobado.

Primero, vamos a hacer que el usuario ponga 5 números entre 1 y 10, y que los guarde en variables. Para ésto usamos ***set /p nombreVariable="Comentario"***. El **/p** indica que el programa debe esperar el ingreso del valor; el "=" dice que la línea terminó pero que luego de eso se puede poner un comentario encerrado entre **comillas dobles**.



Luego, tenemos el cálculo. Esto lo hacemos con ***set /a nombreVariable = cálculo***. En este caso, saco primero la suma total con las variables y el símbolo "+", y luego el promedio dividiendo ese resultado por la cantidad de notas agregadas (que ya sabemos que es 5).



Recuerden que para usar las variables, deben hacerlo entre signos de porcentaje.
Si quieren pueden ir guardando y probando línea por línea para saber si anda todo correctamente.
Ahora debemos imprimirlo en un archivo de texto. Para ésto usamos el ">**.** El símbolo que pusimos allí, guarda la salida (es decir el resultado del comando) en algun lado. Nosotros elegimos un archivo de texto. A parte, con un sólo ">" crea el archivo si es que no existe, o lo reescribe si tenía algo dentro; pero con doble símbolo ">>" sólo agrega la línea al final del archivo -sino bastante molesto sería para este fin-.

```
echo La suma de notas totales es: %sumaTotal% > notas.txt
echo El promedio de notas es: %promedio% >> notas.txt
```

Estamos por buen camino. Nos toca el **if** que felicitará si es que el promedio es igual a 10, pero sino no hace nada. En ese caso, el resultado es **nul**.

```
if %promedio%==10 (echo ¡Felicidades por el promedio! >> notas.txt) else nul
```

Ya casi estamos terminando el programa. Al principio debemos colocar el **@echo off** para que no molesten los verdaderos comandos. Y también podríamos agregar **cls** -recuerden que es el comando que limpia la pantalla- cada vez que queramos para hacerlo de manera prolija.

```
@echo off

rem programa NOTAS, presentacion
echo
echo
echo
echo
echo Bienvenidos al programas NOTAS.
pause>nul

rem ingreso de numeros

cls
set /p primerNumero="Ingrese el primer numero: "
cls
set /p segundoNumero="Ingrese el segundo numero: "
cls
set /p tercerNumero="Ingrese el tercer numero: "
cls
set /p cuartoNumero="Ingrese el cuarto numero: "
cls
set /p quintoNumero="Ingrese el quinto numero: "
cls

rem calculo de total y promedio

set /a sumaTotal=%primerNumero%+%segundoNumero%+%tercerNumero%+%cuartoNumero%+%quintoNumero%
```

A parte, los **echo** sin nada nos dan líneas vacías (por si queremos hacer el salto de línea) y el **pause>nul** le da al usuario tiempo de que vea lo que hicimos para él.

notas.bat - Notepad

File Edit Format View Help

```
cls
set /p quintoNumero="Ingrese el quinto numero: "
cls

rem calculo de total y promedio

set /a sumaTotal=%primerNumero%+%segundoNumero%+%tercerNumero%+%cuartoNumero%+%quintoNumero%
set /a promedio=%sumaTotal%/5

rem Puesta en notepad

echo La suma de notas totales es: %sumaTotal% > notas.txt
echo El promedio de notas es: %promedio% >> notas.txt

rem si promedio es igual a 10, recompensa

if %promedio%==10 (echo ¡Felicidades por el promedio! >> notas.txt) else nul

echo
echo
echo
echo
echo El programa termino
pause>nul
cls

exit
```

"Eh. Perdón Roadd, pero creo que hay algo que no vimos y no me estás explicando."

Ah, cierto. Mis disculpas ^^! ***rem*** es el comando para sólo hacer una línea de **comentario**. Así ésto no saldrá en el programa. Es información de ayuda para aquel que desarrolla o mira el código nada más, porque a la hora de ejecución del batch esta línea se **saltea**.
Es una **buena práctica**, comentar las líneas que queramos.

Vamos a ver el último comando que nos interesa en Batch (hay más y pueden buscar. Si quieren tienen HackXCrack que si mal no recuerdo tienen un tutorial más extenso) y es el comando ***goto***. Lo explico: este comando hace que el programa no vaya directamente a la línea siguiente sino que salta para ejecutar una línea que esta instrucción indica. Lo haremos con un nombre que inventamos, utilizándolo así:

Untitled - Notepad

File Edit Format View Help

```
@echo off

set /p numero="Ingrese un numero binario"
if %numero%==1 (goto etiqueta) else goto etiqueta2

etiqueta:
echo el numero es 1
pause>nul
cls
exit

etiqueta2:
echo el numero es 0
pause>nul
cls
exit
```

Fíjense que las **etiquetas** (así se llama al lugar donde saltamos) termina con "**:**" y que lo escribimos

igual en ambos casos.
goto está en varios lenguajes de programación -de otras formas- pero no se considera una buena práctica, pero aquí en batch puede ser útil, así que abusen de él mientras pueden.

Bueno, me gustaría que realizen alguna **tarea** que les dejo para que practiquen.

1) Existe un tipo de malware que se denomina **bomba fork.** Ésta se encarga de que se ejecuten programas continuamente sin descanso hasta agotar la memoria del sistema (puede ser que tarde bastante dependiendo de qué programas usen y cuánta memoria tengan). Desarrollarlo en la menor cantidad de líneas posibles.

2) Realizar el cambio en el registro que haga que la **bomba fork** se ejecute al **inicio** de Windows. No importa si hay que ejecutarlo como administrador. No buscamos la perfección aún.

3) Realizar un segundo programa que se ejecute al **inicio** y guarde información del **sistema** y de **red** local a un archivo de texto. Una vez que lo hace, que el recolector de información se **autoelimine**.

Pueden seguirme en Twitter: @RoaddHDC

Cualquier cosa pueden mandarme mail a: r0add@hotmail.com

Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente:
1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw

Roadd.
