



## Contenido

On Time Framework.....	1
Propiedades.....	1
Errores.....	2
Clase Principal (Core).....	3
Métodos Básicos.....	3
Métodos Avanzado.....	6
Funciones.....	8
Básicas.....	8
Contenido.....	10
Validación.....	12
Depuración.....	12
Derechos de Autor y Licencia.....	13



## On Time Framework

On Time, es la plataforma de desarrollo de Mundo Genesys diseñada especialmente para aprovechar las ventajas técnicas y económicas que se obtiene de utilizar las nubes, la versión actual es la 3.0 y se esta libreando la 4.0, su uso es totalmente intuitivo y no se requieren conocimientos técnicos para operarlo, no requiero un servidor de base de datos para operar, es robusto y flexible, si se utiliza correctamente es una plataforma de desarrollo, no puede clasificarse en una familia de software particular o un uso específico, este manual se escribe específicamente par la versión 3,0, la cual se entrega como código abierto, puedes utilizarlo libremente dentro de sus desarrollos indicando lo que incluyes y sus correspondiente derecho de autor, una recomendación si unicamente transfieres nuestro software no cobres por ello, te aseguro que se gana mas regalando lo que es gratuito.

Este manual es incremental y se publica anexo a cada clase, siendo la que se publica la primera en el mismo (no se incluyen los códigos previos debes de descargarlos manualmente).

Solo como aclaración para hablar en los mismos términos, el core es la clase central, el siguiente nivel es el modulo y son clases con una función específica, las características son pequeñas aplicaciones y los desarrollos, estos dos últimos se crean usando los módulos.

**PRECAUCIÓN: NO ALTERES DE FORMA MANUAL LA ESTRUCTURA O CONTENIDO, LOS RESULTADOS SON IMPREVISIBLES.**

### Propiedades

Nombre	Desde	Uso
conected	Core	Tiene el estado de la conexión si lo estas o eres un usuario anónimo
content	Core	Incluye la lista del contenido del contenedor
err	Core	Código del ultimo error presentado, si la ultima llamada no presento error tiene el valor "0"
errtext	Core	Arreglos de las descripción de los errores, clasificado por su código
features	Core	Incluye los módulos instalados en el contenedor
id	Core	Id del usuario conectado, si no esta conectado ninguno tiene el valor "Anonimus"
name	Core	El nombre completo del usuario
nick	Core	Su nombre corto, apodo o como lo conocen socialmente
retval	Core	Variable interna

Nombre	Desde	Uso
safety	Core	Indica todo los módulos a los que tiene acceso el usuario, tanto la suya propia que es la prioritaria como la asignada por medio de grupos
user	Core	Arreglo que obtiene la información publica del usuario, lo que quiere compartir con todos
userp	Core	Arreglo que obtiene la información privada del usuario, lo que no quiere compartir con todos

## Errores

Nombre	Desde	Significado
C0010M001	Core	Fallo la lectura de contenido
C0010M002	Core	Fallo la creación de contenido
C0010M003	Core	Fallo la grabación de contenido
C0010M004	Core	Contenedor faltante
C0010M005	Core	Falta contenido básico del sistema, sistema corrupto
C0010M006	Core	Error de acceso
C0010M007	Core	Registro ya existe
C0010M008	Core	Registro no existe
C0010M009	Core	Registro no disponible
C0010M010	Core	Registro no visible
C0010M011	Core	No hay usuario Conectado
C0010M012	Core	No estas autorizado
C0010M013	Core	Id, no se puede cambiar
C0010M014	Core	Contenedor desconocido
C0010M015	Core	Modulo solo disponible en la versión de pago
C0010M016	Core	Error de tipo de datos
C0010M017	Core	Contenedor existe



Nombre	Desde	Significado
C0010M018	Core	Valor Invalido
C0010M019	Core	Modulo no instalado
C0010M020	Core	Modulo instalado
C0010M021	Core	Hay un usuario Conectado
C0010M022	Core	Registro no esta activo
C0010M023	Core	Registro no autorizado
C0010M024	Core	Estatus desconocido
C0010M025	Core	No es un estatus valido
C0010M026	Core	Dato invalido

## Clase Principal (Core)

Esta es la clase principal, en ella se manejan usuarios, seguridad y errores, para que este lista para trabajar coloca todos los archivos en el lugar donde desees llamarlo y ejecuta desde tu navegador el archivo "OnTimeInstaller.php" se distribuirán el contenido si todo es exitoso se eliminara el archivo, el archivo "DemoCore.php" te mostrara un caso practico del uso de la clase, la clase esta dividida en "para todos" (CoreB) y "para administradores" (CoreA), la básica debe de usarse siempre, la avanzada solo si quien este conectado sea un administrador.

## Métodos Básicos

Nombre	Parámetro		Uso
	Tipo	Contenido	
Check	Salida	Lógico	Este método valida la existencia del contenido mínimo necesario para que el sistema opere, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
Connect	Entrada	El id del usuario que se desea conectar	Este método, se emplea para abrir el sistema para un usuario en particular, si la conexión es exitosa carga su seguridad e información, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Entrada	Clave de acceso de usuario	

Nombre	Parámetro		Uso
	Tipo	Contenido	
	Salida	Lógico, si se pudo conectar o no	
DiscConnect	Salida	Lógico	Borra los datos del usuario y queda como usuario "Anonimus"
FrcPssChg	Entrada	Clave del usuario	Si un administrador cambio el estatus a "force", clave de acceso debe de ser cambiada lo cual la volverá a activar en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Entrada	Clave de acceso actual del usuario	
	Entrada	Nueva clave de acceso del usuario	
	Salida	Lógico	
FrtShwAll	Salida	Arreglo	Muestra la lista de todos los módulos instalados, si no existe devuelve un arreglo vacío
FtrShwPrv	Entrada	Modulo	Muestra la información publica de un modulo, si no existe la información o el modulo regresa un arreglo vacío
	Salida	Información	
FtrShwPbl	Entrada	Modulo	Muestra la información publica de un modulo, si no existe la información o el modulo regresa un arreglo vacío
	Salida	Arreglo	
MyAddPbl	Entrada	Clave	Agrega el contenido a la información publica del usuario conectado bajo el rubro de clave, si la información publica no existe la crea, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Entrada	Contenido	
	Salida	Lógico	
MyAddPrv	Entrada	Clave	Agrega el contenido a la información privada del usuario conectado bajo el rubro de clave, si la información privada no existe la crea, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Entrada	Contenido	
	Salida	Lógico	
MyChgInf	Entrada	Nombre	Al usuario conectado, le asigna el nombre de usuario y el nick, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Entrada	Nick	
	Salida	Lógico	
MyChgPbl	Entrada	Clave	En la información publica del usuario, modifica el contenido almacenado bajo el rubro de clave, verificar el código de error para identificar el motivo
	Entrada	Contenido	
	Salida	Lógico	
MyChgPrv	Entrada	Clave	En la información privada del usuario, modifica el



Nombre	Parámetro		Uso
	Tipo	Contenido	
	Entrada	Contenido	contenido almacenado bajo el rubro de clave, verificar el código de error para identificar el motivo
	Salida	Lógico	
MyDltPbl	Entrada	Clave	De la información publica elimina el contenido relacionado con la clave, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Salida	Lógico	
MyDltPrv	Entrada	Clave	De la información privada elimina el contenido relacionado con la clave, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Salida	Lógico	
MyPssChg	Entrada	Clave de acceso actual	Del usuario actualmente conectado cambia la clave de acceso si la clave suministrada es correcta y la cuenta se encuentra activa, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Entrada	Clave de acceso nueva	
	Salida	Lógico	
PssChk	Entrada	Clave del usuario	Valida si la combinación de clave del usuario y clave de acceso del usuario son correctas y el usuario esta activo en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo
	Entrada	Clave de acceso del usuario	
	Salida	Lógico	
UserDlt	Entrada	Clave del usuario a eliminar	Elimina todo rastro del usuario, tanto físico como lógico, esta acción no puede revertirse, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo, verificar el código de error para identificar si hay error.
	Salida	Lógico	
UserShwPbl	Entrada	Clave del usuario	Busca si existe información publica de un usuario, si no existe el usuario o su información publica devuelve un arreglo vacío, verificar el código de error para identificar si hay error.
	Salida	Arreglo con la información	
UsrShwAll	Salida	Arreglo	Devuelve la lista de todos los usuarios, en caso de obtener una lista en blanco verificar el código de error para identificar el motivo.
UsrShwNln	Salida	Arreglo	Devuelve la lista de usuarios activos, en caso de obtener una lista en blanco verificar el código de error para identificar el motivo.

## Métodos Avanzado

Nombre	Parámetro		Uso
	Tipo	Contenido	
ErrAdd	Entrada	Clave	Agrega la clave y descripción del error, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Entrada	Descripción	
	Salida	Lógico	
ErrChg	Entrada	Clave	Modifica la descripción asociada a la clave del error, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Entrada	Descripción	
	Salida	Lógico	
ErrDlt	Entrada	Clave	Elimina la descripción asociada a la clave del error, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Salida	Lógico	
FtrAddPbl	Entrada	Modulo	Si el modulo existe y el usuario tiene permiso en el mismo, añade una nueva descripción relacionada a la clave en el contenido publico, el valor de retorno contiene el resultado, en caso de ser falso validad el código de error
	Entrada	Clave	
	Entrada	Descripción	
	Salida	Lógico	
FtrAddPrv	Entrada	Modulo	Si el modulo existe y el usuario tiene permiso en el mismo, añade una nueva descripción relacionada a la clave en el contenido privado, el valor de retorno contiene el resultado, en caso de ser falso validad el código de error
	Entrada	Clave	
	Entrada	Descripción	
	Salida	Lógico	
FtrChgPbl	Entrada	Modulo	Si el modulo existe y el usuario tiene permiso en el mismo, cambia la descripción relacionada a la clave en el contenido publico, el valor de retorno contiene el resultado, en caso de ser falso validad el código de error
	Entrada	Clave	
	Entrada	Descripción	
	Salida	Lógico	
FtrChgPrv	Entrada	Modulo	Si el modulo existe y el usuario tiene permiso en el mismo, cambia la descripción relacionada a la clave en el contenido privado, el valor de retorno contiene el resultado, en caso de ser falso validad el código de error
	Entrada	Clave	
	Entrada	Descripción	
	Salida	Lógico	
FtrDltPbl	Entrada	Modulo	Si el modulo existe y el usuario tiene permiso en el mismo, elimina la descripción relacionada a la clave en el contenido publico, el valor de retorno contiene el resultado, en caso de ser falso validad el código de error
	Entrada	Clave	
	Salida	Lógico	
FtrDltPrv	Entrada	Modulo	Si el modulo existe y el usuario tiene permiso en el mismo, elimina la descripción relacionada a la clave en el contenido privado, el valor de retorno contiene el resultado, en caso de ser falso validad el código de error
	Entrada	Clave	
	Salida	Lógico	
FtrShwUsr	Entrada	Clave	Devuelve los usuarios y niveles asociados, a la clave de





Nombre	Parámetro		Uso
	Tipo	Contenido	
	Salida	Arreglo	modulo, en caso de obtener un arreglo vacío, verificar el código de error para identificar el motivo
UserAdd	Entrada	Clave del Usuario	Agrega un usuario al sistema, y le asigna el estatus inicial, en caso de que no se suministre nombre y/o nick se les asigna el valor 'None', en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Entrada	Clave de acceso del Usuario	
	Entrada	Estatus	
	Entrada	Nombre (opcional)	
	Entrada	Nick (opcional)	
	Salida	Lógico	
UserChgStt	Entrada	Clave de usuarios	Devuelve el valor del estatus del usuario, en caso de ser un valor vacío, verificar el código de error para identificar el motivo.
	Salida	Valor	
UserChkStt	Entrada	Clave del usuario	Modifica el valor del estatus del usuario al nuevo valor, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Entrada	Valor del estado	
	Salida	Lógico	
UsrAddFtr	Entrada	Clave	Agrega al usuario clave acceso al modulo con el nivel, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Entrada	Modulo	
	Entrada	Nivel	
	Salida	Lógico	
UsrAddMn	Entrada	Clave	Agrega al usuario clave acceso al modulo interno con el nivel, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Entrada	Nivel	
	Salida	Lógico	
UsrChgFtr	Entrada	Clave	Modifica el nivel de acceso al usuario clave acceso del modulo, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Entrada	Modulo	
	Entrada	Nivel	
	Salida	Lógico	
UsrChgMn	Entrada	Clave	Modifica el nivel de acceso al usuario clave acceso del modulo interno, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Entrada	Nivel	
	Salida	Lógico	
UsrDltFtr	Entrada	Clave	Elimina al usuario clave del modulo, en caso de que el valor devuelto sea falso, verificar el código de error para
	Entrada	Modulo	



Nombre	Parámetro		Uso
	Tipo	Contenido	
	Entrada	Nivel	identificar el motivo.
	Salida	Lofico	
UsrDltMn	Entrada	Clave	Elimina al usuario clave del modulo interno, en caso de que el valor devuelto sea falso, verificar el código de error para identificar el motivo.
	Salida	Lógico	
UsrShwFtr	Entrada	Clave	Devuelve los módulos y niveles asociados, a la clave de usuario, en caso de obtener un arreglo vacío, verificar el código de error para identificar el motivo
	Salida	Arreglo	
UsrShwMn	Salida	Arreglo	Devuelve los usuarios que tienen derechos a realizar modificaciones en el modulo interno asi como su nivel de acceso, en caso de obtener un arreglo vacío, verificar el código de error para identificar el motivo

## Funciones

### Básicas

Nombre	Parámetro		Desde	Uso
	Tipo	Contenido		
ot_connect	Entrada	Lógico conexión (opcional)	Core	Verifica si la conexión esta abierta o cerrada en caso de que se pase como parámetro false, el retorno es congruente con la solicitud, en caso que proceda agrega el error y el retval
	Salida	Lógico		
ot_qexist	Entrada	Contenido	Core	Esta función regresa si existe o no el contenido dentro del contenedor, solo regresa un valor lógico no almacena un error o un código de retorno retval
	Entrada	Contenedor (opcional)		
	Salida	Lógico		
ot_exist	Entrada	Contenido	Core	Esta funciono no solo valida la existencia del contenido en el contenedor, también almacena el resultado en el error y en el valor de retorno
	Entrada	Contenedor (opcional)		
	Salida	Lógico		
not_exist	Entrada	Contenido	Core	Esta funciono no solo valida la no existencia del contenido en el contenedor, también almacena el resultado en el error y
	Entrada	Contenedor (opcional)		



Nombre	Parámetro		Desde	Uso
	Salida	Lógico		en el valor de retorno
ot_can	Entrada	Nivel	Core	Esta función valida que usuario actualmente conectado, en el modulo tiene un nivel menor o igual a nivel requerido, el nivel 0 es administrador, el 1 es puede eliminar, el 2 puede añadir y el 3 puede cambiar, no solo regresa el resultado lo almacena en el error y valor de retorno
	Entrada	Modulo		
	Salida	Lógico		
ot_remove	Entrada	Contenido	Core	Esta función eliminar el contenido dentro del contenedor en caso de que exista, si el contenido es un contenedor elimina todo el contenido, en caso de error lo almacena en su variable y en el valor de retorno
	Entrada	Contenedor (opcional)		
	Salida	Lógico		
ot_create	Entrada	Contenedor	Core	Esta función crea un contenedor, en la raíz o dentro del otro, en caso de presentarse error lo almacena en la variable y el valor de retorno
	Entrada	En (opcional)		
	Salida	Lógico		
ot_destroy	Entrada	Contenedor	Core	Esta función es recursiva y elimina uno a uno contenido y contenedor
	Salida	Null		
ot_getin	Entrada	Contenedor	Core	Esta función devuelve el arreglo del contenido (contenedores) dentro del contenedor, nunca considera la posibilidad de error, devuelve el arreglo vacio
	Salida	Contenido (arreglo)		
ot_now	Entrada	Formato (opcional)	Core	Devuelve en la cadena, la fecha y/o hora del servidor, en el formato especificado
	Salida	Cadena		
ot_RandomString	Entrada	Largo	Core	Devuelve una cadena alfanumérica aleatoria, de largo en caracteres, las cadenas aleatorias siempre comienzan en X
	Salida	Cadena		
ot_clean	Entrada	Cadena	Core	A la cadena le elimina caracteres ocultos para dejar una cadena limpia que se pueda procesar mas fácilmente
	Salida	Limpia		
ot_value	Entrada	Dato	Core	Valida que el dato y el valor sean idénticos, en caso contrario, registra el error, y el valor de retorno queda en falso
	Entrada	Valor		
	Entrada	Error		
	Salida	Lógico		
not_value	Entrada	Dato	Core	Valida que el dato y el valor sean diferentes si son idénticos registra el error y el valor
	Entrada	Valor		

Nombre	Parámetro		Desde	Uso
	Entrada	Error		de retorno queda como falso
	Salida	Lógico		
ot_name	Entrada	Contenedor	Core	De un contenedor en particular si existe devuelve el nombre y el nick del mismo, si no existe devuelve la cadena vacía
	Salida	Nombre		

## Contenido

Nombre	Parámetro		Desde	Uso
	Tipo	Contenido		
ot_add	Entrada	Clave	Core	Busca la clave en el contenido recibido, en caso que no exista lo agrega asociandolo al valor, inmediatamente lo almacena usando el nombre del contenido, dentro de un contenedor que puede omitirse, si existe la clave coloca el error que aplica y falso en el valor de retorno, el resultado es el contenido original adicionando el nuevo si es que procede
	Entrada	Valor		
	Entrada	Contenido (arreglo)		
	Entrada	Nombre del contenido		
	Entrada	Contenedor (opcional)		
	Salida	Resultado (arreglo)		
ot_addchangein	Entrada	Clave	Core	Abre el contenido localizado en el contenedor, busca la clave si no existe agrega la clave asociándola al valor, si existe modifica el valor relacionado, si el contenido no existe lo crea, en todo caso devuelve el arreglo con la nueva información
	Entrada	Valor		
	Entrada	Contenido		
	Entrada	Contenedor (opcional)		
	Salida	Resultado (arreglo)		
ot_addin	Entrada	Clave	Core	Abre el contenido localizado en el contenedor, busca la clave, si no existe la agrega asociada al valor, si el contenido no existe lo crea, si la clave ya existe, mueve el valor del error y del valor del retorno, en ambos casos devuelve como resultado el arreglo
	Entrada	Valor		
	Entrada	Contenido		
	Entrada	Contenedor (opcional)		
	Salida	Resultado (arreglo)		
ot_array	Entrada	Arreglo	Core	Si sobrescribir es falso, abre el contenido dentro del contenedor, agrega el arreglo a la lectura si la clave existe modifica el
	Entrada	Contenido		
	Entrada	Sobrescribir		



Nombre	Parámetro		Desde	Uso
	Tipo	Contenido		
	Entrada	Contenedor		valor, si no existe contenido previo o tiene la opción de sobre escribir el contenido es es el valor del arreglo, unicamente maneja un error si el Arreglo que debe de suministrar no es tal
	Salida	Lógico		
ot_change	Entrada	Clave	Core	Busca la clave en el contenido, si existe modifica su valor asociado, si no existe mueve el error y el valor de retorno a falso, lo almacena inmediatamente en el contenedor con el nombre del contenido, el valor de regreso es el arreglo modificado
	Entrada	Valor		
	Entrada	Contenido (arreglo)		
	Entrada	Nombre del Contenido		
	Entrada	Contenedor Opcional		
	Salida	Regreso (arreglo)		
ot_changein	Entrada	Clave	Core	Si existe abre el contenido dentro del contenedor, dentro del el busca la clave y en caso afirmativo le asigna el valor, grabándolo inmediatamente, si estas condiciones no se cumple graba el error y modifica el valor de retorno, devuelve un arreglo con la información.
	Entrada	Valor		
	Entrada	Contenido		
	Entrada	Contenedor		
	Salida	Resultado (arreglo)		
ot_delete	Entrada	Clave	Core	Busca la clave en el contenido y si existe la elimina del contenido, almacenándolo con el nombre indicado dentro del contenedor, regresa el arreglo resultante, si se presenta un error lo almacena tanto en su variable como en la de retorno
	Entrada	Contenido		
	Entrada	Nombre del contenido		
	Entrada	Contenedor (opcional)		
	Salida	Regreso (arreglo)		
ot_deletein	Entrada	Clave	Core	Abre el contenido en el contenedor, busca la clave y la elimina, almacenado el nuevo contenido, y regresandolo, si se presento un error este se coloca en su variable igual que el valor de retorno
	Entrada	Contenido		
	Entrada	Contenedor (opcional)		
	Salida	Regreso (arreglo)		

### Lectora/Escritura del contenido

Nombre	Parámetro		Desde	Uso
	Tipo	Contenido		
ot_read	Entrada	Contenido	Core	Trata de leer el contenido en el contenedor, en caso de que no exista reconoce el error, también se refleja en el valor de retorno, se asume que el contenido es JSON.
	Entrada	Contenedor (opcional)		
	Salida	Regreso (arreglo)		
ot_readif	Entrada	Contenido	Core	Trata de leer el contenido en el contenedor, en caso de no existir, no marca error lo genera en blanco se asume que el contenido es JSON.
	Entrada	Contenedor (opcional)		
	Salida	Regreso (arreglo)		
ot_write	Entrada	Contenido	Core	Trata de escribir los datos en el contenido dentro del contenedor
	Entrada	Datos		
	Entrada	Contenedor (opcional)		
	Salida	Logico		

## Validación

Nombre	Parámetro		Desde	Uso
	Tipo	Contenido		
ot_record				No aplica
ot_valid				No aplica

## Depuración

Nombre	Parámetro		Desde	Uso
	Tipo	Contenido		
ot_error	Entrada	Valor (opcional)	Core	En caso de que en la ultima llamada no presente error, devuelve como texto el valor, si se presento error busca la definición del mismo y la devuelve si la clave de error no esta registra el texto es Undefined Error
	Salida	Texto		
ot_log				No Aplica



Nombre	Parámetro		Desde	Uso
	Tipo	Contenido		
ot_show	Entrada	Datos	Core	Despliega a pantalla los datos indicando clave y valor, taladra los niveles que se le indique si son otros arreglos
	Entrada	Nivel (opcional)		
	Salida	Nulo		

## Derechos de Autor y Licencia

\*\*\*\*\*

Diseño y programación por Mario Carrocera Menéndez

copyright @ 2003-2021 Mario Carrocera Menéndez

copyright @ 2019-2021 Mundo Genesys de IT (On Time)

Licencia para las versiones anteriores a la 4.0 publicadas como open source en los términos de:

Attribution-NonCommercial-ShareAlike 4.0 International

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Limite de Responsabilidad:

Mario Carrocera Menéndez y/o Mundo Genesys de IT, no son responsables legal o moralmente de daños físicos o lógicos causados por el uso del software, tampoco se comprometen a realizar ninguna modificación al mismo, se suministra de forma gratuita en su estado actual y su venta esta prohibida.

Pongo a tu disposición mis mail de contacto, sin embargo, el recibir una sugerencia incluya código fuente o no, no me comprometo de ninguna forma a incluirlo y/o genera compromisos comerciales o legales con quien me envíe un mail,

Si deseas desarrollos especiales para ti o tu empresa, también te ofrezco mis servicios como desarrollador independiente, siéntete con la total confianza de contactarme

Email:

[marioc@mundogenesys.com](mailto:marioc@mundogenesys.com)



[marioc@mundogenesys.com](mailto:marioc@mundogenesys.com) [mario.carrocera@hotmail.com](mailto:mario.carrocera@hotmail.com)

[mario.carrocera@hotmail.com](mailto:mario.carrocera@hotmail.com)

[marcartools@gmail.com](mailto:marcartools@gmail.com)

Fecha de Ultima Modificación:

2 / 1 / 2021

Release: 3.47

\*\*\*\*\*

On Time  
Mario Carrocera@2003-2021  
marioc@mundogenesys.com