



evolution

DEVELOPER.NET MANUAL DE REFERENCIA

Software for a real
omnichannel experience

1 ÍNDICE

1	ÍNDICE.....	2
2	Objetivo de este manual	5
3	Más información?.....	5
4	El entorno de desarrollo	5
4.1	Entorno integrado de desarrollo de Developer.net.....	5
4.1.1	Conceptos Generales.....	6
4.1.2	Área de trabajo	6
4.2	Creación de proyectos	13
4.2.1	Agregar elementos a un proyecto	14
4.3	Despliegue de la aplicación.....	15
5	Scripts de agente	17
5.1	Proyectos de script de agente	17
5.2	Plantillas de proyecto	18
5.3	Plantillas de elementos.....	18
5.4	Edición de una página del proyecto.....	20
5.5	Orígenes de datos proporcionados por Developer.NET	21
5.5.1	Ejemplo de uso	23
6	Scripts DBR	25
6.1	Proyectos de script DBR.....	25
6.2	Plantillas de proyecto	25
6.3	Plantillas de elementos.....	26
6.4	Edición de un script.....	26
6.4.1	Diagrama	27
6.4.2	Bloques	28
6.4.3	Tareas	29
6.4.4	Conectores.....	30
6.4.5	Comentarios	31
6.4.6	Editor de audio	31

6.5	Referencia de los elementos DBR-Script	32
6.5.1	Atributos comunes	32
6.5.2	Bloques	33
6.5.3	Propiedades	33
6.5.4	Colecciones	38
6.5.5	Tareas para el manejo de las colas	40
6.5.6	Tareas para el manejo de llamadas	49
6.5.7	Tareas de acceso a datos	62
6.5.8	Tareas propias del lenguaje	67
6.5.9	Tareas personalizadas.....	75
7	Visual Evolution Library	76
7.1	GOTO_Button	76
7.2	CALLDISPOSITION_Button	76
7.3	CALLDISPOSITION_RadioButtonList.....	77
7.4	DATEPICKER	77
7.5	Controles de datos de cliente	78
7.5.1	CUSTOMERDATA_Label	79
7.5.2	CUSTOMERDATA_LabelGroup	79
7.5.3	CUSTOMERDATA_TextBox.....	80
7.5.4	CUSTOMERDATA_DATAPICKER	81
7.5.5	CUSTOMERDATA_DropDownList.....	81
7.5.6	CUSTOMERDATA_TZDropDownList.....	82
7.5.7	SAVECUSTOMERDATA_Button	82
7.6	Controles de tipo Documento	82
7.6.1	Document_OpenButton	83
7.6.2	Document_TreatedButton	84
7.6.3	Document_Summary	84
7.7	Controles de Twitter	85
7.7.1	Twitter_Editor.....	85

7.7.2	Twitter	86
7.8	Controles de Email	87
7.8.1	Email	87
7.8.2	Email_Editor	89
7.9	Controles de WebChat	90
7.9.1	Chat	90
7.9.2	Chat_Viewer	91
8	EvolutionLibrary	92
8.1	Evolution.EvolutionLibrary.EvoAPI	92
8.1.1	Propiedades	92
8.1.2	Métodos	93
8.1.3	Ejemplos de uso	94
9	Plantillas de documentos	95
9.1.1	Datos de la tabla Clientes	95
9.1.2	Datos de la tabla tbDatosCliente	95
9.1.3	Datos de la tabla Campanya	95
9.1.4	Datos del Agente	95
9.1.5	Datos de la Transacción	95
9.1.6	Datos de la tabla tbContactos	95
9.1.7	Datos de la tabla tbContactosDoc	95
9.1.8	Ejemplo de plantilla	96

2 OBJETIVO DE ESTE MANUAL

Este manual describe las funcionalidades de Evolution Developer.NET, así como recomendaciones y sugerencias para el desarrollo de argumentarios y scripts DBR.

3 MÁS INFORMACIÓN?

Para obtener más información acerca de Evolution, entre al Centro de Recursos en <http://www.evolutioncallcenter.com>, donde encontrará más tutoriales, manuales, y el foro de acceso público de Evolution.

4 EL ENTORNO DE DESARROLLO

Developer.NET constituye un completo entorno de desarrollo de aplicaciones para Call Centers, que permite que los usuarios construyan aplicaciones para los agentes y scripts DBR.

Este entorno está orientado al desarrollo visual, y no requiere conocimientos de programación avanzados. Facilita el despliegue de aplicaciones en un ambiente de call center.

Las características más importantes de Developer.NET son:

- Basado en Microsoft Visual Studio 2010
- Explorador de soluciones y proyectos
- Opciones avanzadas de edición como coloreado basado en la sintaxis, drag & drop, etc.
- Entorno de desarrollo visual. Permite obtener una visión del resultado final antes de desplegar la aplicación para los agentes.
- Despliegue de aplicaciones integrado con Evolution
- Integración con gestores de código fuente como Microsoft Visual SourceSafe.
- Acceso a datos simplificado
- Controles visuales específicos
- Plantillas de proyectos y de archivos

4.1 ENTORNO INTEGRADO DE DESARROLLO DE DEVELOPER.NET

El entorno de desarrollo está basado en Visual Studio 2010. Con carácter general, puede consultar las funciones de edición disponibles consultando la página de información del producto de Microsoft en el siguiente enlace:

<http://www.microsoft.com/spain/visualstudio/>

En los apartados que siguen, comentaremos las funcionalidades más utilizadas y específicas para poder desarrollar aplicaciones con Developer.NET

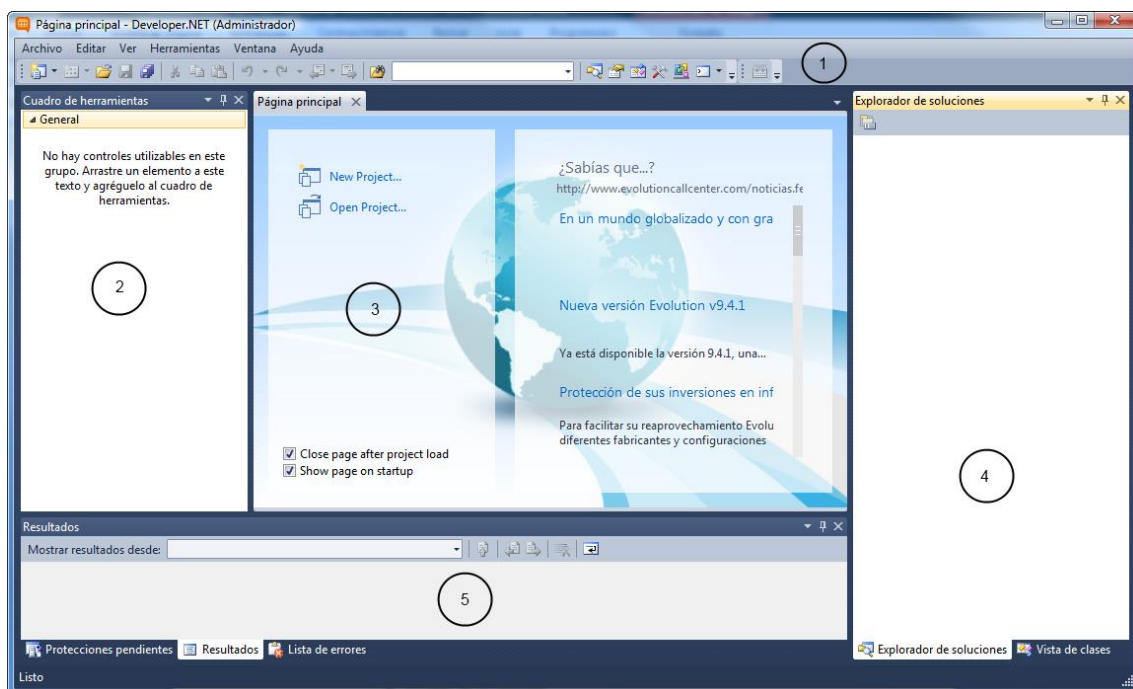
4.1.1 CONCEPTOS GENERALES

Developer.NET utiliza las mismas estructuras que Visual Studio: soluciones y proyectos. Una solución puede contener varios proyectos y cada proyecto está compuesto por un conjunto de elementos.

Los proyectos de Developer.NET se crean en el directorio Developer.NET dentro de la carpeta de Documentos del usuario de Windows.

4.1.2 ÁREA DE TRABAJO

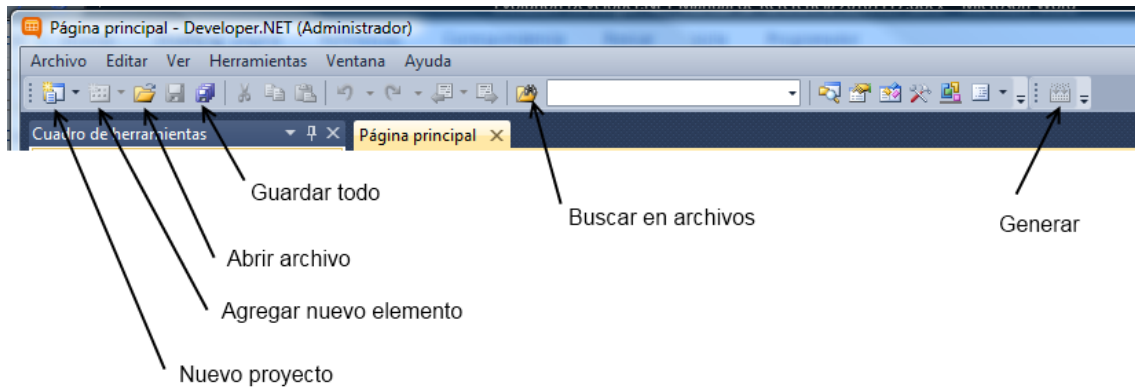
Una vez instalado, el entorno Developer.NET está disponible en la carpeta Evolution de la lista de programas. Tras una “splash screen” nos aparece la página principal de Developer.NET.



Developer.NET presenta un entorno de desarrollo organizado en diferentes áreas de trabajo:

- ❑ 1-Menús y Barras de Herramientas.
- ❑ 2-Cuadro de herramientas.
- ❑ 3-Área de edición.
- ❑ 4-Explorador de soluciones.
- ❑ 5-Área de resultados.

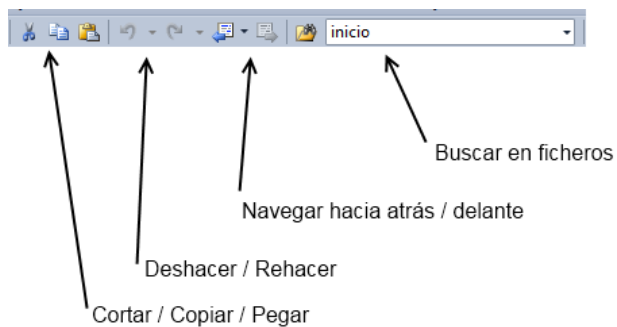
4.1.2.1 MENÚ Y BARRAS DE HERRAMIENTAS



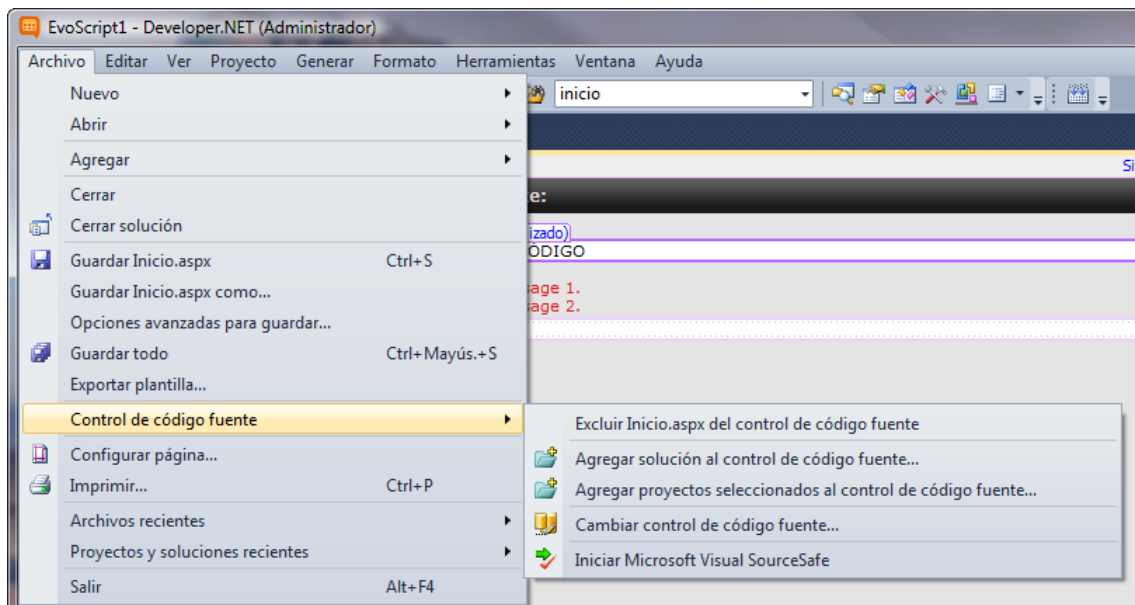
PRINCIPALES FUNCIONES DE LA BARRA DE HERRAMIENTAS

Nuevo Proyecto	Crea un nuevo proyecto a partir de la plantilla de proyecto que escojamos.
Agregar nuevo elemento	Agrega un nuevo elemento al proyecto actual. Los elementos disponibles varían en función del tipo de proyecto.
Abrir archivo	Abre el archivo especificado
Guardar todo	Salva el contenido de las páginas que hayamos modificado
Buscar en archivo	Permite buscar cadenas de caracteres en los archivos de la solución.
Generar	Despliega la aplicación.

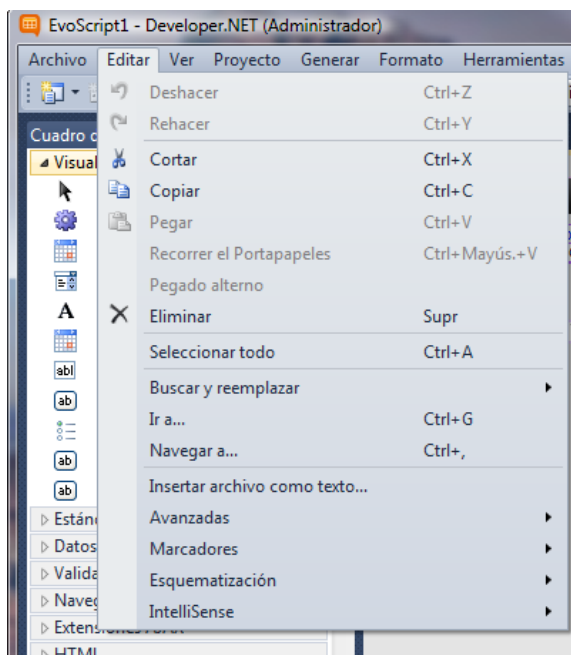
Botones para la edición:



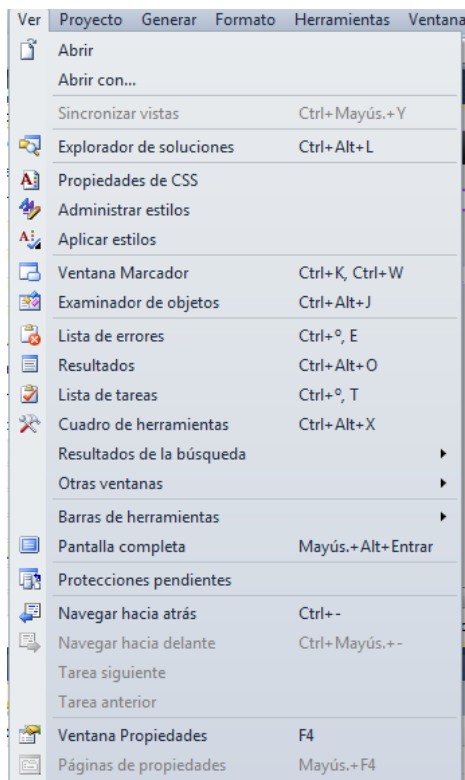
Opción de menú **Archivo**: opciones para abrir y guardar archivos, proyectos y soluciones. Control de código fuente.



Opción de menú **Editar**: opciones de edición básicas y avanzadas.



Opción de menú **Ver**:



Opción de menú **Proyecto/Propiedades**:

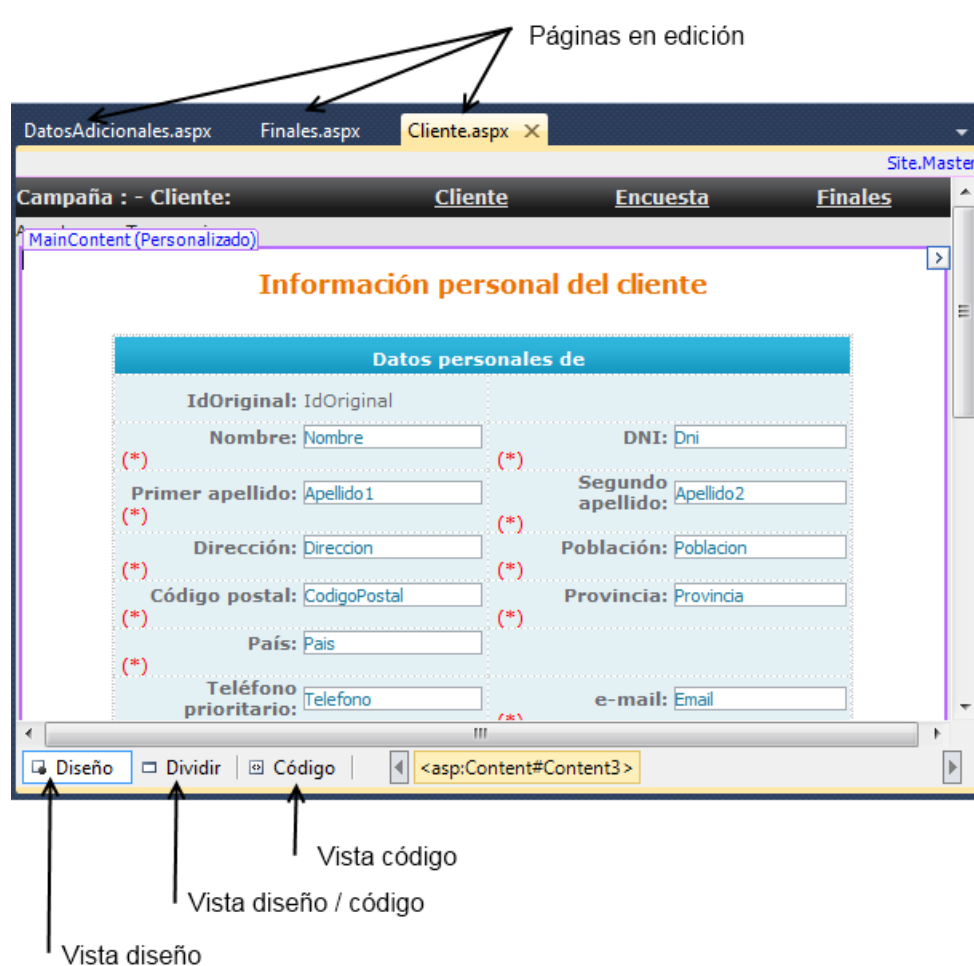
Propiedades comunes de los proyectos	
Tipo de despliegue	DeveloperService -> utiliza el servicio de despliegue de aplicaciones de un servidor Evolution para desplegar la aplicación. FileCopy -> copia los archivos de aplicación en el directorio especificado.
Despliegue a través de servicio web	URL del servicio web de despliegue de Evolution. Por defecto, la URL apunta a localhost.
Despliegue por copia de ficheros	Directorio donde se copiarán los archivos de la aplicación. Por defecto, es el directorio local.

4.1.2.2 CUADRO DE HERRAMIENTAS

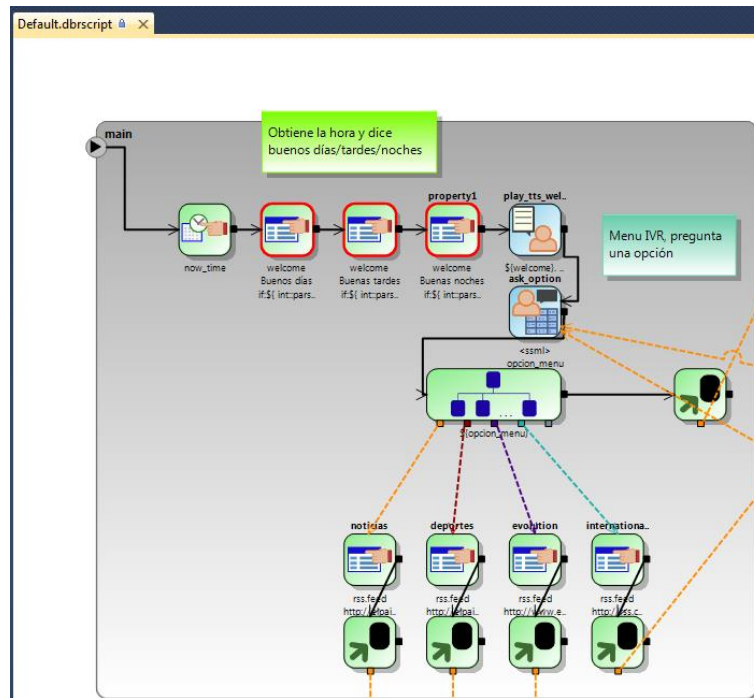
El cuadro de herramientas varía en función del archivo que se esté editando. Las herramientas están agrupadas por fichas. Pueden ser arrastradas y soltadas sobre el archivo que estemos editando, y una vez configuradas sus propiedades más relevantes ya tendremos la funcionalidad implementada en dicho archivo.

4.1.2.3 ÁREA DE EDICIÓN

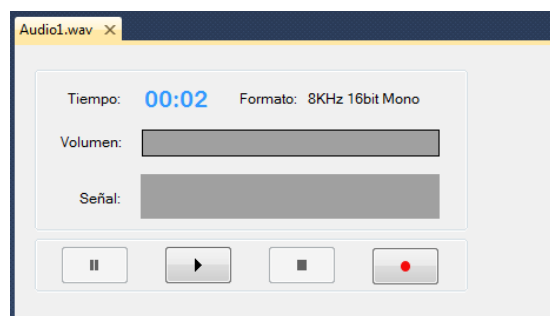
En esta zona se editan los diferentes archivos que forman el proyecto. Los archivos se abren en distintas pestañas y con el editor más apropiado.



Editor de páginas web

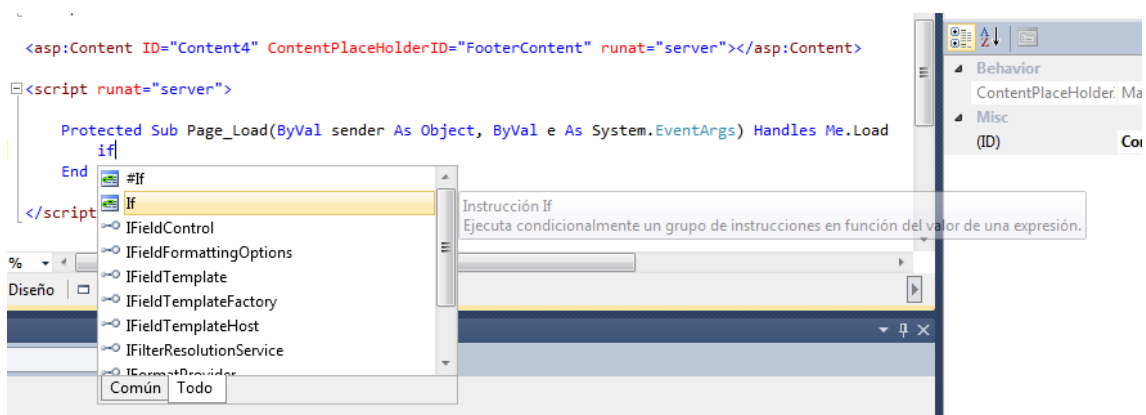


Editor Script DBR



Editor WAV

INTELLISENSE

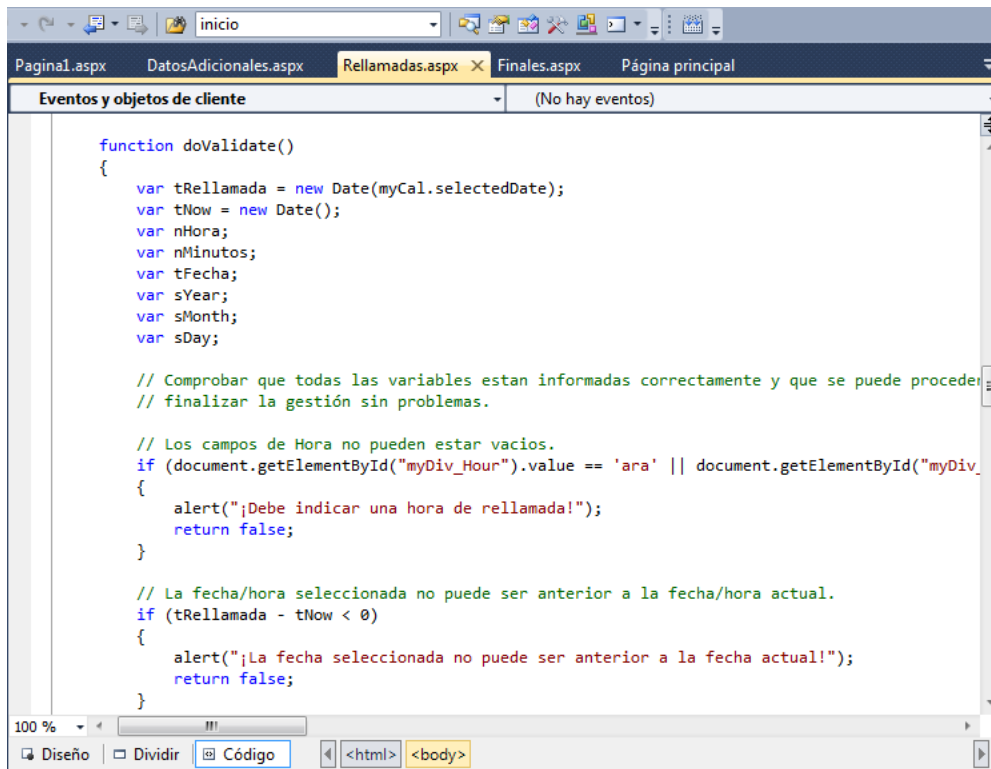


Durante la edición de código usted puede beneficiarse de la característica de Intellisense de Developer.NET. Puede hacer búsquedas en los elementos del lenguaje, insertar elementos del lenguaje,

consultar parámetros e incluso hacer que Intellisense complete una instrucción por usted. Para más información de Intellisense, consulte la web de Visual Studio 2010.

SYNTAX COLORING

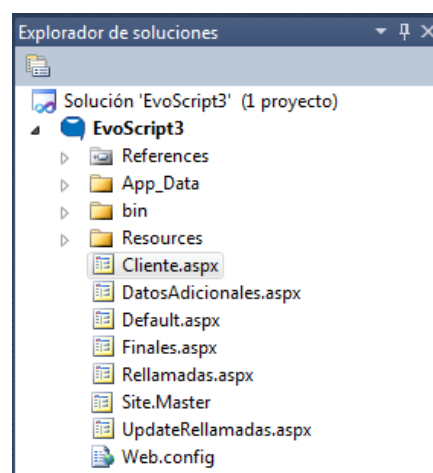
En la vista de código usted podrá visualizar el código de las páginas de Developer.NET diferenciadas por color en función de la sintaxis, lo que proporciona un alto grado de legibilidad.



4.1.2.4 EXPLORADOR DE SOLUCIONES

El Explorador de soluciones permite ver elementos y realizar tareas de administración de elementos en una solución o en un proyecto. Una única solución y sus proyectos se muestran mediante una presentación jerárquica. Se pueden realizar diferentes operaciones de administración de los elementos tales como:

- Copiar, pegar o renombrar elementos
- Añadir, crear y excluir elementos
- Si se está trabajando con un gestor de código fuente, realizar operaciones asociadas con dicho gestor.
- Acceder a las propiedades de un elemento
- Etc.



4.1.2.5 ÁREA DE RESULTADOS

En el área de resultados se muestran diferentes tipos de información, típicamente como resultado de la ejecución de un comando de Developer.NET:

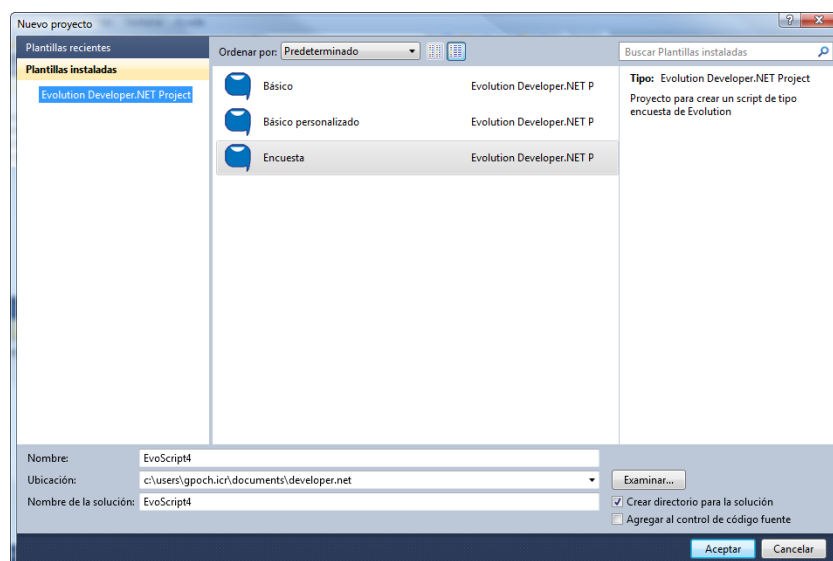
- Búsquedas
- Despliegue de aplicaciones Evolution
- Mensajes de error, advertencias o informativos
- Mensajes del gestor de código fuente
- Lista de tareas
- Marcadores
- Etc.

	Descripción	Archivo	Línea	Columna	Proyecto
1	Validación (XHTML 1.0 Transitional): el atributo 'align' se considera obsoleto. Se recomienda utilizar una construcción más reciente.	Cliente.aspx	19	23	EvoScript3
2	Validación (XHTML 1.0 Transitional): el atributo 'align' se considera obsoleto. Se recomienda utilizar una construcción más reciente.	Finales.aspx	12	23	EvoScript3
3	Validación (XHTML 1.0 Transitional): el atributo 'height' se considera obsoleto. Se recomienda utilizar una construcción más reciente.	Finales.aspx	25	21	EvoScript3

4.2 CREACIÓN DE PROYECTOS

Normalmente iniciaremos el desarrollo de la aplicación mediante la creación de un nuevo proyecto. El nuevo proyecto quedará incluido en una solución con el mismo nombre.

Cuando creamos un nuevo proyecto, obtenemos una pantalla similar a esta:



Se nos propone un nombre de proyecto y de solución que se creará a partir de una de las plantillas de proyecto existentes.

Los proyectos pueden exportarse a plantillas de proyecto. Las plantillas así creadas son plantillas de usuario que pueden ser utilizadas a su vez para crear nuevas aplicaciones. Esto permite que el usuario cree aplicaciones basándose en patrones.

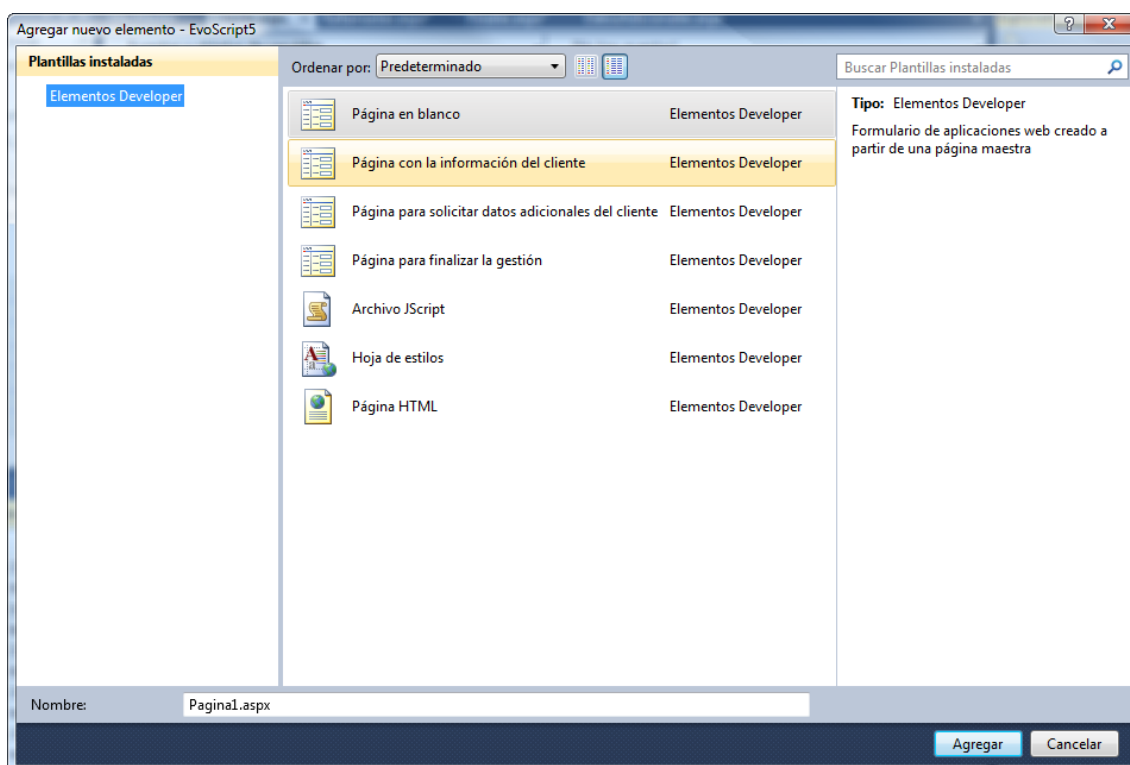
Para exportar un proyecto a plantilla de proyecto elija la opción Archivo/Exportar plantilla. Aparecerá un asistente que le permitirá crear una plantilla basándose en las páginas que contiene actualmente su proyecto.

4.2.1 AGREGAR ELEMENTOS A UN PROYECTO

Podemos añadir un nuevo elemento al proyecto de múltiples formas:

- Seleccionando el icono “Agregar nuevo elemento” de la barra de tareas de Developer.NET
- Desde el explorador de soluciones, seleccionar el proyecto y pulsar Ctrl-Mayúsculas-A.
- Desde el explorador de soluciones, seleccionar el proyecto, botón derecho, Agregar/Nuevo elemento.

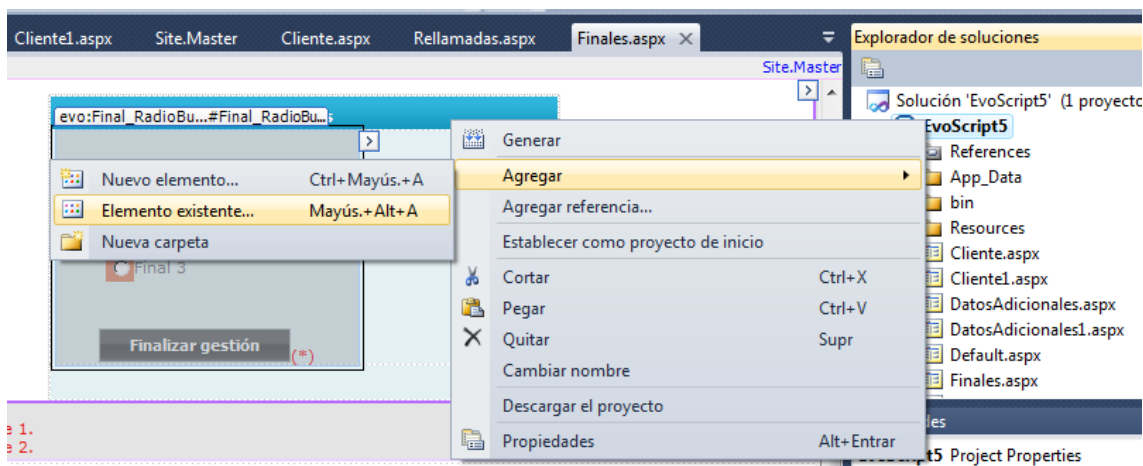
Se nos mostrará un diálogo con los diferentes tipos de elementos que podemos agregar, los elementos que se nos muestran varían en función del tipo de proyecto:



También podemos crear nuestras propias plantillas de elementos a partir de un elemento del proyecto actual. Para ello, utilice la opción Archivo/Exportar plantilla. Aparecerá un asistente. Elija la opción “Plantilla de elemento”.

4.2.1.1 AÑADIR OTROS TIPOS DE FICHERO

Se pueden añadir otros tipos de ficheros al proyecto desde el explorador de soluciones mediante la opción “Agregar elemento existente” del menú contextual de proyecto.



Esto puede ser útil, por ejemplo, para agregar recursos de los que ya dispongamos y los necesitemos utilizar en el proyecto actual: librerías JScript, imágenes, archivos multimedia, etc.

4.3 DESPLIEGUE DE LA APLICACIÓN

Una vez definido el contenido del proyecto, podemos desplegarlo en el servidor de aplicaciones de Evolution.

Existen dos tipos de despliegue, que se pueden seleccionar en las propiedades del proyecto:

- Despliegue con **Developer Service**: despliega la aplicación en el servidor especificado, da de alta el script en Manager y en caso de scripts de agente define el directorio de la aplicación como aplicación web de IIS. Requiere un usuario evolution con el rol de ‘Publicar scripts desde el entorno de desarrollo’ o ‘control total’.
- Despliegue con **FileCopy**: copia los archivos de la aplicación en la ubicación especificada.

Los scripts de agente se despliegan en el servidor de aplicaciones de Evolution bajo el directorio Evolution\ScriptServer\Scripts del sitio web por defecto.

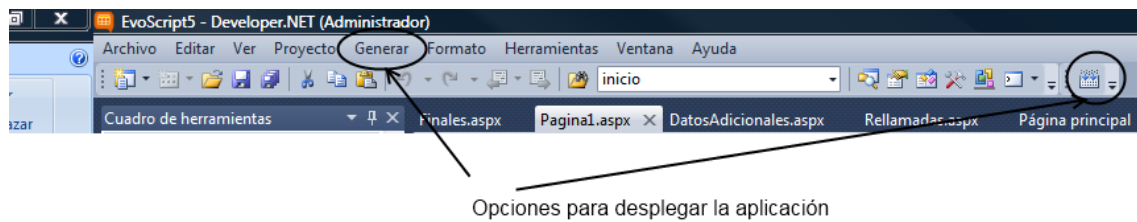
Mientras que los scripts DBR lo hacen en Evolution\DBRScripts.

Normalmente estos directorios están bajo el directorio físico C:\inetpub\wwwroot\Evolution del servidor de Evolution.

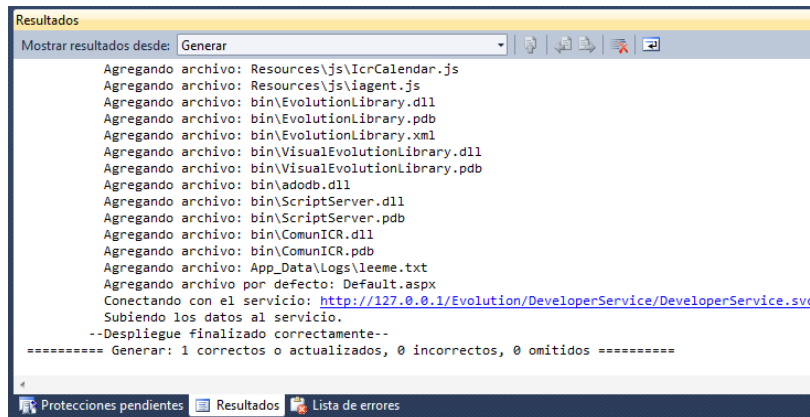
Recuerde:

- Si despliega un script con FileCopy, deberá darlo de alta en Manager y en los scripts de agente también deberá crear la aplicación manualmente en IIS.
- Para usar un script de agente debe asociarse a una o varias campañas mediante la opción Administrar/Campañas de Manager.
- Para usar un script DBR deberá asociarlo a alguna estrategia DBR de tipo script.

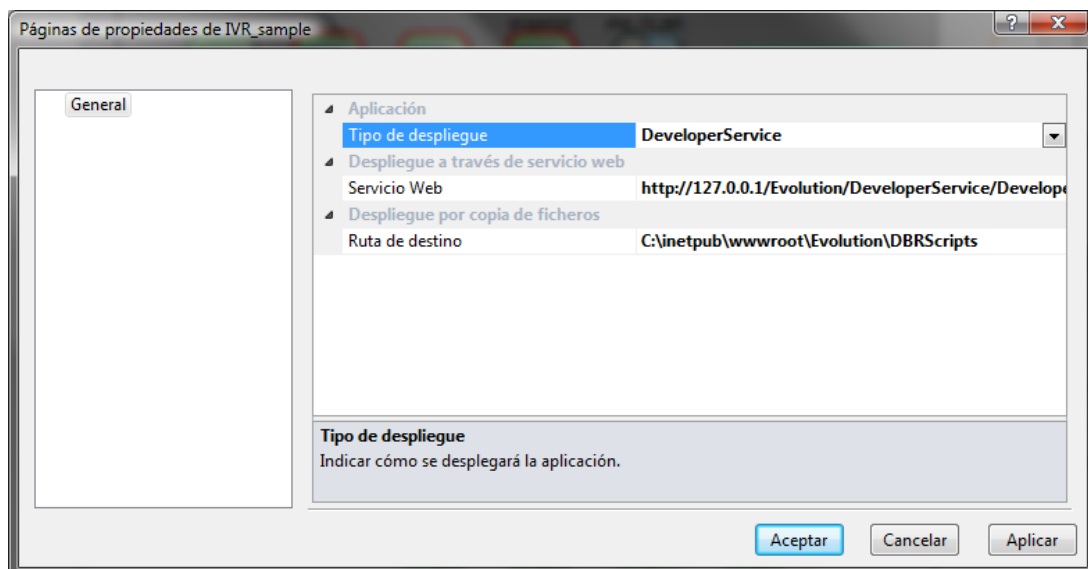
Para desplegar la aplicación puede pulsar el icono de Generar o bien la opción de menú Generar:



Podemos consultar el resultado de la operación de despliegue en la ventana de resultados:



El proyecto se despliega en el directorio o servidor especificado en las propiedades del proyecto:



Por defecto, se despliega en la máquina local donde se ha instalado Developer.NET. Esto permite probar la aplicación en un entorno de desarrollo antes de ponerlo en producción.

5 SCRIPTS DE AGENTE

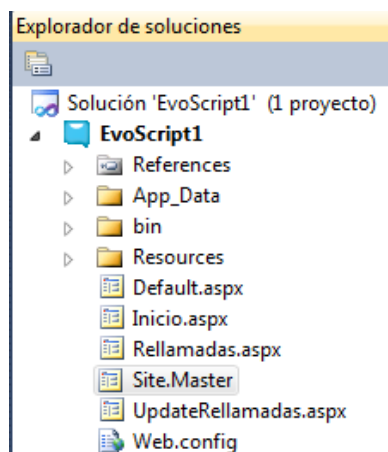
Las aplicaciones generadas son web-based y utilizan la tecnología ASP.NET. Esto significa que podemos utilizar en nuestras aplicaciones toda la potencia que nos ofrece el framework .NET.

Las aplicaciones forman un conjunto de páginas por las que puede ir navegando el agente a lo largo de la conversación con el cliente. Típicamente en estas páginas aparece un guion con texto y acciones a seguir según las contestaciones obtenidas.

5.1 PROYECTOS DE SCRIPT DE AGENTE

Típicamente un proyecto de script de agente tendrá los siguientes elementos:

- **References:** referencias a distintas DLL's utilizadas en el proyecto. Incluye referencias a los elementos de la Visual Evolution Library y a EvolutionLibrary que comentaremos en apartados siguientes.
- **App_Data:** carpeta destinada a contener los logs de la aplicación de agente.
- **Bin:** binarios de algunas DLL's utilizadas en el proyecto.
- **Resources:** hojas de estilo, gráficos, archivos Javascript, templates...
- **Páginas del proyecto.** Todo proyecto de Evolution (Evolution Script Project) contiene una página Default.aspx y todas las páginas necesarias por las que navegará el agente. También se incluye una página maestra Site.Master que puede ser utilizada en las nuevas páginas que vayamos creando.



Las páginas de argumentario son páginas ASP.NET y pueden contener código Visual Basic .NET para incorporar la lógica de aplicación.

Propiedades específicas de un proyecto de script de agente

Buscador personalizado	Indica qué páginas se mostrarán para identificar clientes. No es modificable, depende de si se ha partido de la plantilla con búsqueda personalizada o no. Si -> Se navegará a la página BuscarCliente.aspx del proyecto actual, que podemos personalizar a nuestra conveniencia. No -> Se navegará a una página de sistema.
Página inicial	Seleccione la página a la que navegará el agente al iniciar una transacción con Evolution.

5.2 PLANTILLAS DE PROYECTO

Por defecto, cuando se instala Developer.NET disponemos de tres plantillas:

Plantillas de proyecto		
	Básico	Contiene una página inicial (Inicio.aspx) de la que podemos partir para construir una aplicación desde cero. Además, se añade una página para programar rellamadas, una página Master y todas las referencias necesarias para poder utilizar los controles y objetos de la Visual Evolution Library y Evolution Library.
	Básico personalizado	Añade a lo anterior todas las páginas de identificación de clientes. Esto nos permite personalizar la identificación de clientes en Evolution con nuestra propia lógica de negocio y/o diseño web.
	Encuesta	Se trata de una aplicación completa de referencia para crear aplicaciones de tipo encuesta (estudios de mercado, ventas, etc.) Contiene una página en donde se muestran los datos del cliente (Cliente.aspx), un formulario donde se piden datos de la encuesta (DatosAdicionales.aspx) y una página de finales de Evolution (Finales.aspx).

5.3 PLANTILLAS DE ELEMENTOS

Con Developer.NET se instalan las siguientes plantillas de páginas ASP.NET que contienen los controles integrados con Evolution y que facilitan las tareas más habituales en los argumentarios: presentar información del cliente, tomar datos, finalizar la transacción...

Plantillas de página		
	Página en blanco	Una página en blanco.
	Página con la información del cliente	Página que muestra la información del cliente y permite modificarla.
	Página con control de chat integrado	Página que interactuar con chats.

	Página con la información del Email	Página que muestra emails y permite responder.
	Página con la información del Tweet	Página que muestra 'tweets' y permite responder.
	Página para solicitar datos adicionales del cliente	Página para mostrar y modificar información adicional de cliente (campos libres).
	Página para finalizar la gestión	Página que nos muestra finales para finalizar la transacción en Evolution

También están disponibles otras plantillas más genéricas:

Otros tipos de página		
	Archivo de texto	Fichero de texto
	Archivo JScript	Fichero con código JScript.
	Hoja de estilos	Fichero de hoja de estilos
	Página HTML	Fichero de tipo HTML

5.3.1.1 PÁGINA MAESTRA SITE.MASTER

Las plantillas de página de Evolution hacen referencia a una página maestra (Master Page) llamada Site.Master. Esta página, incluida en las plantillas de proyecto de Developer.NET contiene una cabecera con información de la campaña, el cliente, etc. una zona reservada para contenido y un pie de página. Usted puede modificar esta página maestra a su conveniencia o crear otras páginas maestras.

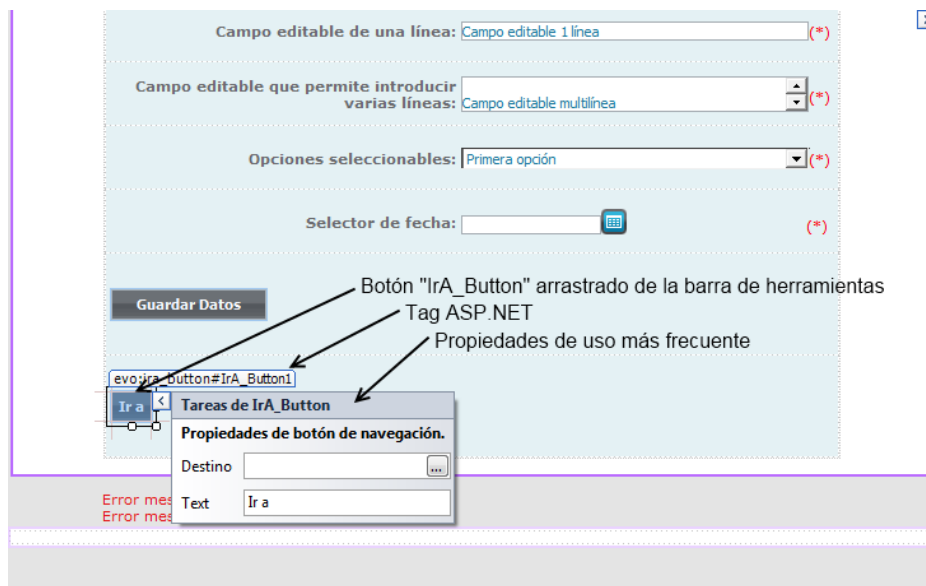
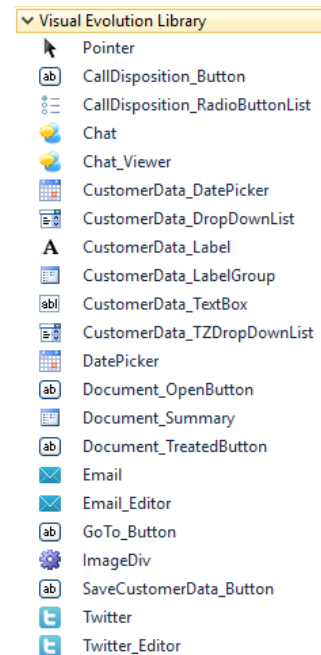
5.4 EDICIÓN DE UNA PÁGINA DEL PROYECTO

Si hacemos doble click en una página del explorador de soluciones, podremos editarla en el editor de páginas de Developer.NET.

Normalmente la edición de una página conlleva incluir controles de interfaz gráfica de usuario y su personalización a nuestras necesidades. Típicamente, arrastraremos del cuadro de herramientas de Developer.NET los controles necesarios, haremos los ajustes necesarios en la interfaz de usuario e incluiremos un botón para navegar a la siguiente página o finalizar la transacción con Evolution.

La ficha de Visual Evolution Library contiene los controles visuales más utilizados en las aplicaciones de Evolution. En el capítulo [Visual Evolution Library](#) se detalla la funcionalidad de cada uno de los controles de esta ficha.

Por ejemplo, si arrastramos el botón “GoTo_Button” de la ficha “Visual Evolution Library” en modo diseño obtendremos lo siguiente:



Vemos el aspecto del botón tal y como lo verá el usuario. Podemos modificar ahora las propiedades más usadas de este control para, en este caso, definir el texto del botón y la página a la que se navegará si se pulsa este botón.

La vista de código nos muestra los tags ASP.NET generados para dicho control.

```
<button type="button" id="IrA_Button1" runat="server" Text="Ir a" />
```

Los controles de la Visual Evolution Library están diseñados para que un usuario sin conocimientos de programación específicos pueda crear y editar formularios de forma rápida y sencilla. Es posible incluir dentro del formulario otros tipos de controles, incluyendo los controles estándar de ASP.NET.

Normalmente no será necesario, pero si usted desea generar argumentarios complejos, puede incluir código ASP.NET en las páginas de Developer.NET. Las páginas están diseñadas para poder ser programadas con Visual Basic.NET.

Las páginas de Developer.NET son páginas ASP.NET por lo que pueden contener:

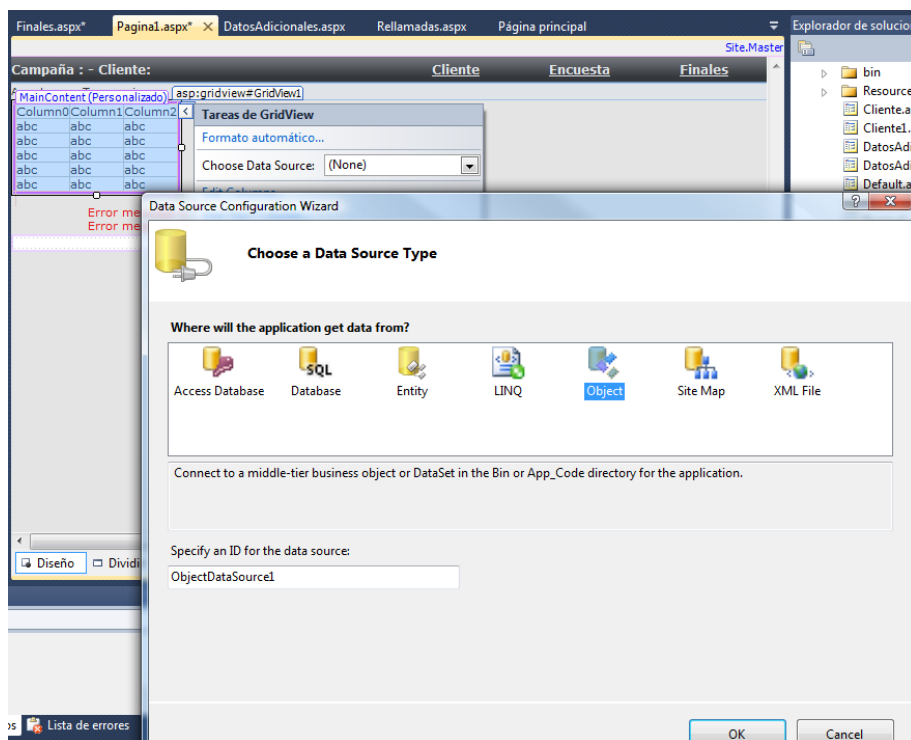
- Código de servidor en Visual Basic .NET
- Código script de cliente (Javascript, VBScript)
- HTML
- Estilos
- Controles de ASP.NET
- Controles de usuario ASP.NET
- Utilización de recursos y clases del framework .NET
- Etc.

Los programadores avanzados pueden utilizar Developer.NET para implementar funcionalidad avanzada en las páginas de argumentario.

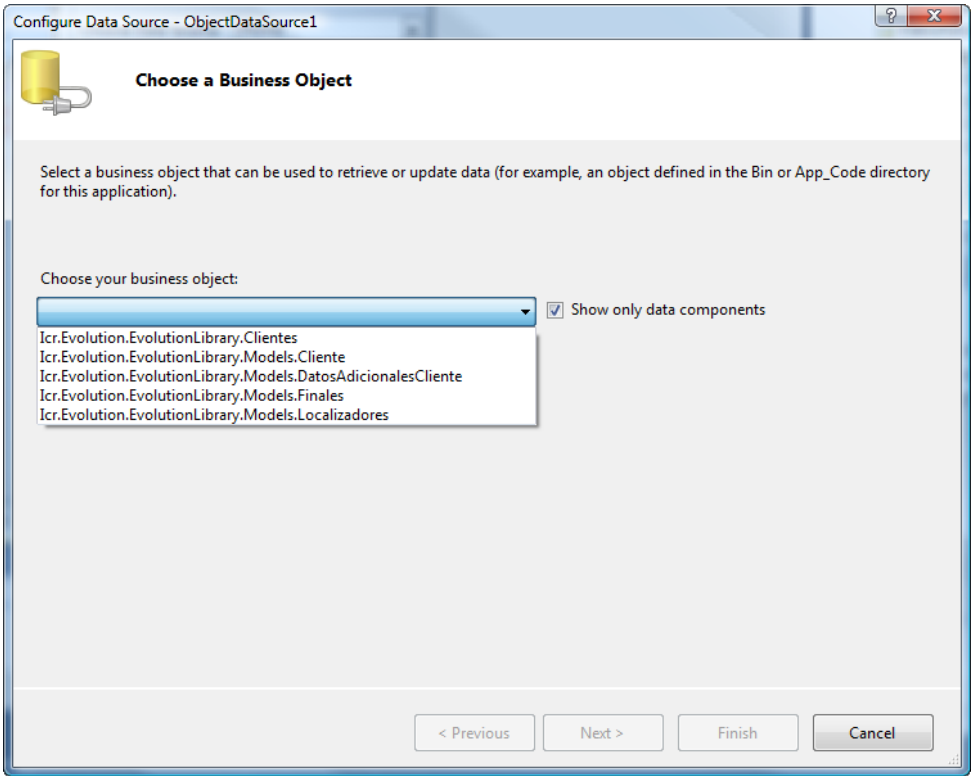
5.5 ORÍGENES DE DATOS PROPORCIONADOS POR DEVELOPER.NET

Cuando se instala Developer.NET se crea unos orígenes de datos de tipo objeto que pueden ser utilizados por el usuario para mostrar información de Evolution en controles .NET que permitan especificar orígenes de datos.

Por ejemplo, podemos poblar un control ASP.NET de tipo GridView para mostrar información de clientes.



Podemos ver la lista de fuentes de tipo Object Data Source disponibles con Developer.NET:



Orígenes de datos de tipo objeto de datos

Cientes	Los clientes de la campaña actual, los clientes de la campaña actual con un localizador (teléfono) determinado, los clientes de una campaña X o los clientes de una campaña X que tengan un localizador determinado.
Cliente	El cliente actual
DatosAdicionalesCliente	Los datos adicionales del cliente actual
Finales	Los finales de la campaña actual
Localizadores	Los localizadores (teléfonos) del cliente actual

5.5.1 EJEMPLO DE USO

Podemos poblar una lista desplegable de ASP.NET indicando como DataSource un ObjectDataSource creado desde Developer.NET. Partimos de la plantilla de página para finalizar la gestión, eliminando el control Final_RadioButtonList.

- 1) Seleccionar la carpeta "Datos" del cuadro de herramientas.
- 2) Arrastrar y soltar ObjectDataSource a la página
- 3) Configurar el ObjectDataSource con Icr.Evolution.EvolutionLibrary.Models.Finales.
- 4) En el método Select, escoger GetFinalesCampanyaActual(String filtro)
- 5) En el DefaultValue del parámetro filtro, poner 100-. Con ello especificamos que únicamente queremos visualizar los finales con id>=100.

Debemos tener un control en la página con este código:

```
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
    DeleteMethod="Delete_NoImplementado"
    InsertMethod="Insert_NoImplementado"
    OldValuesParameterFormatString="original_{0}"
    SelectMethod="GetFinalesCampanyaActual"
    TypeName="Icr.Evolution.EvolutionLibrary.Models.Finales"
    UpdateMethod="Update_NoImplementado">
    <DeleteParameters>
        <asp:Parameter Name="o" Type="Object" />
    </DeleteParameters>
    <InsertParameters>
        <asp:Parameter Name="idCampanya" Type="Int64" />
        <asp:Parameter Name="idFinal" Type="Int64" />
        <asp:Parameter Name="descripcion" Type="String" />
        <asp:Parameter Name="claseFinal" Type="Int32" />
        <asp:Parameter Name="contactado" Type="Int32" />
        <asp:Parameter Name="planificacionAgente"
Type="Int32" />
        <asp:Parameter Name="minutos" Type="Int64" />
        <asp:Parameter Name="maxContactos" Type="Int64" />
        <asp:Parameter Name="gestionLocalizadores"
Type="Int32" />
    </InsertParameters>
    <SelectParameters>
        <asp:Parameter DefaultValue="100-" Name="filtro"
Type="String" />
    </SelectParameters>
    <UpdateParameters>
        <asp:Parameter Name="idCampanya" Type="Int64" />
        <asp:Parameter Name="idFinal" Type="Int64" />
        <asp:Parameter Name="descripcion" Type="String" />
        <asp:Parameter Name="claseFinal" Type="Int32" />
        <asp:Parameter Name="contactado" Type="Int32" />
        <asp:Parameter Name="planificacionAgente"
Type="Int32" />
        <asp:Parameter Name="minutos" Type="Int64" />
        <asp:Parameter Name="maxContactos" Type="Int64" />
        <asp:Parameter Name="gestionLocalizadores"
Type="Int32" />
    </UpdateParameters>
</asp:ObjectDataSource>
```

- 6) Seleccionar la carpeta "Estándar" del cuadro de herramientas.
- 7) Arrastrar y soltar el control DropDownList de ASP.NET
- 8) Elegir como DataSource del control del punto 7 el ObjectDataSource que hemos creado.
- 9) Elegir Descripción con dato a mostrar y IdFinal como dato para el valor de la lista desplegable.

El control creado tendrá el siguiente código:

```
<asp:DropDownList ID="DropDownList1" runat="server"
    Height="16px"
    Width="146px" DataSourceID="ObjectDataSource1"
    DataTextField="Descripcion"
    DataValueField="IdFinal">
</asp:DropDownList>
```

Hasta ahora hemos creado un desplegable que se llena dinámicamente con los finales de negocio de la campaña. Ahora faltaría añadir un botón que permitiera finalizar el argumentario con el final elegido.

- 1) Arrastrar de la Visual Evolution Library un control de tipo Final_Button
- 2) Crear un método que cambie la propiedad idFinal del control del punto anterior en función del valor seleccionado en la lista desplegable.

```
<asp:DropDownList ID="DropDownList1" runat="server"

OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged" Height="16px"
    Width="146px" DataSourceID="ObjectDataSource1"
    DataTextField="Descripcion"
    DataValueField="IdFinal">
</asp:DropDownList>

Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles DropDownList1.SelectedIndexChanged
    Final_Button1.idFinal = DropDownList1.SelectedValue
End Sub
```

Finalizar / Replanificar llamada

The screenshot shows a web application interface with two main sections. The top section, titled 'Observaciones', contains a text area where the word 'Owner' has been entered. The bottom section, titled 'Finales', contains a label 'Final elegido:', a dropdown menu with 'Volver a llamar' selected, and a button labeled 'Finalizar gestión'.

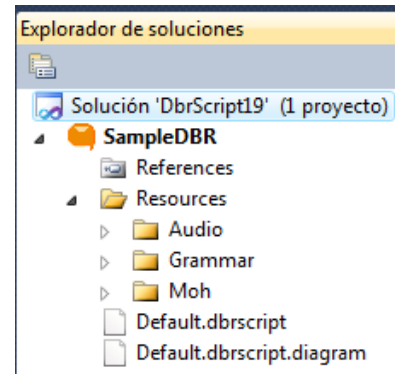
6 SCRIPTS DBR

Los scripts DBR permiten controlar el flujo de las llamadas que entren en el sistema a través de una estrategia configuradas para usar dicho script.

6.1 PROYECTOS DE SCRIPT DBR

Típicamente un proyecto de script DBR tendrá los siguientes elementos:

- **References:** referencias a distintas DLL's utilizadas en el proyecto. *Normalmente estará vacío en este tipo de proyectos.*
- **Resources:**
 - **Audio:** carpeta para los audios que deseamos reproducir.
 - **Grammar:** carpeta para las gramáticas (archivos SRGS) usadas por las tareas de reconocimiento de voz.
 - **Moh:** carpeta para las clases (carpetas con archivos de audio) de música en espera (*Music On Hold*).
- **Default.dbrscript:** archivo XML que contiene la información del script. *No debe ser renombrado.*
- **Default.dbrscript.diagram:** archivo XML que guarda la representación gráfica del script. *No debe ser renombrado.*



Si incluye archivos de audio y usa el despliegue a través del DeveloperService, debe configurarlo adecuadamente, en la documentación de instalación (Capítulo *Dynamic Business Router*, apartado *Developer Service*) se puede consultar como configurarlo para que al generar el proyecto los archivos de audio se guarden en las carpetas correctas y puedan ser reproducidos.

6.2 PLANTILLAS DE PROYECTO

Por defecto, cuando se instala Developer.NET disponemos la siguiente plantilla:

Plantillas de proyecto		
	Básico	Contiene el bloque 'main' y una tarea 'Echo' que guarda un mensaje de inicio de script en las trazas.
	Queue Básico	Similar a usar una estrategia estática, encola la llamada en la campaña configurada en la estrategia, y mientras la llamada no es atendida por un agente se reproduce música en espera. Si pasada una hora la llamada no ha sido atendida, se cortará.

6.3 PLANTILLAS DE ELEMENTOS

Con Developer.NET se instalan las siguientes plantillas de elementos:

Plantillas de página



Archivo de audio (WAV)

Archivo de audio en formato 'wav' compatible con Asterisk.

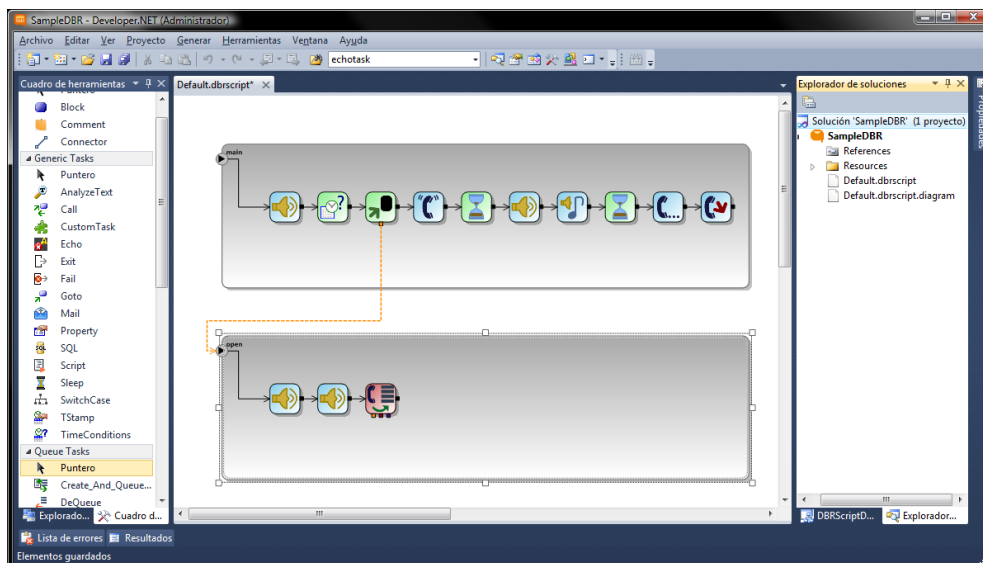


Grammar File (SRGS)

Archivo con los patrones gramaticales que espera identificar la tarea de reconocimiento de voz.

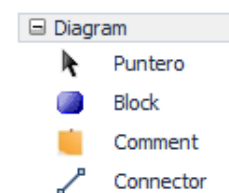
6.4 EDICIÓN DE UN SCRIPT

A continuación, se describe cómo usar Developer.NET para editar un script, para ver una descripción detallada de los Scripts DBR y las tareas consulte en el manual de 'Integración de aplicaciones en Evolution' el capítulo 'Definición de estrategias con DBRScript'.

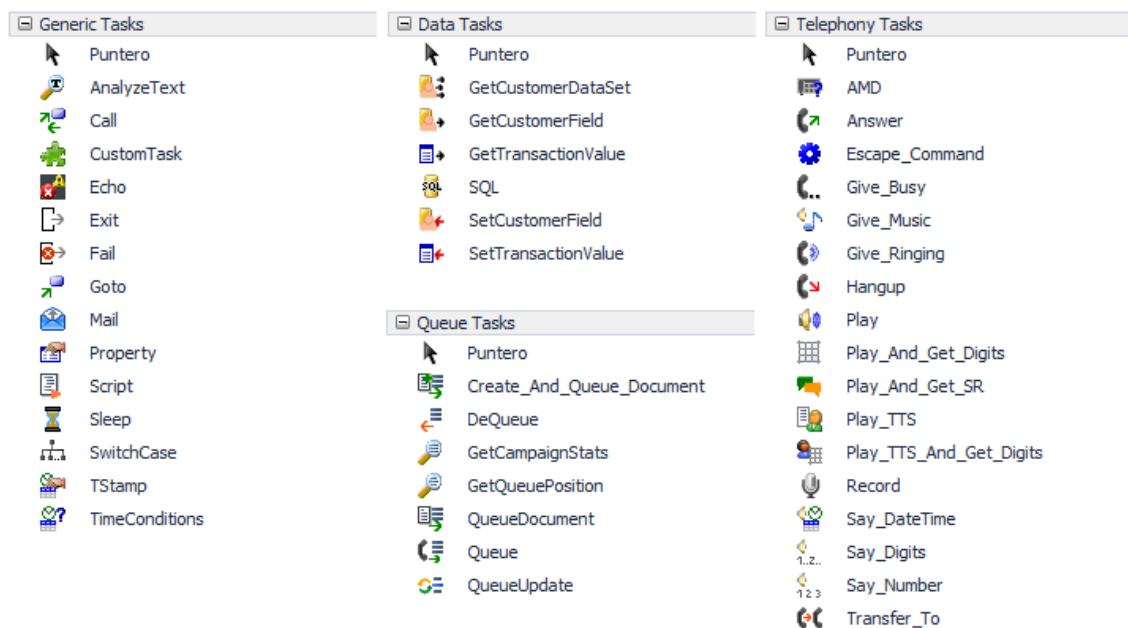


Empezaremos por abrir el archivo *Default.dbrscript*, esto abrirá el diagrama, donde podrá agregar elementos arrastrándolos desde el cuadro de herramientas, en el que se han agrupado dichos elementos en cuatro fichas:

- Diagram: agrupa los elementos que no son tareas de scripting:
 - o Block: elementos que contienen las tareas de los scripts.
 - o Comment: permite añadir cuadros con comentarios.
 - o Connector: permite conectar las tareas para crear el flujo del script.



- Data Tasks: (tareas para la manipulación de datos) acceso a los datos del cliente, ejecución de comandos SQL.
- Generic Tasks: (tareas genéricas) control del flujo de ejecución, análisis y de textos, envío de emails, horarios y calendarios, acceso a librerías .NET, etc.
- Queue Tasks: (tareas de gestión de colas) gestión de las colas de llamadas e interacciones de las campañas, acceso a datos de las colas, etc.
- Telephony Tasks: (tareas de telefonía) control de las llamadas, emisión de mensajes de audio y música en espera, recepción de tonos DTMF, etc.



6.4.1 DIAGRAMA

El diagrama es la representación gráfica del script y contiene uno o más bloques, que a su vez son los que contienen las tareas.

Podemos aumentar o reducir el tamaño de los elementos del diagrama si mantenemos pulsada la tecla 'Control' y usamos la rueda del ratón, hacia arriba aumenta el tamaño y hacia abajo lo reduce.

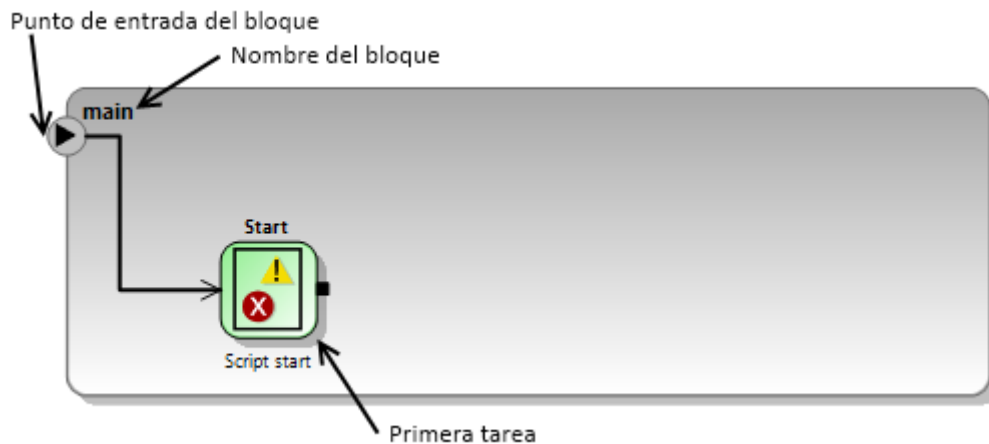
Si hacemos click sobre una zona vacía del diagrama (donde no haya bloques, tareas, comentarios ni conectores), podremos ver las propiedades del mismo:

- Grid Enabled: permite habilitar la rejilla de diseño para que así las tareas se alineen fácilmente.
- Show Task Info: permite mostrar la etiqueta y la información más significativa de las tareas en el diagrama.

6.4.2 BLOQUES

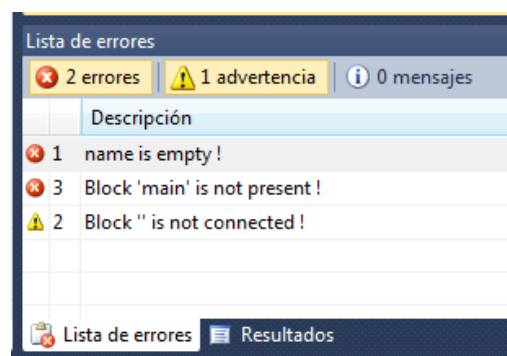
Un script requiere que exista al menos un elemento de tipo bloque (Block) con nombre 'main'. Cada bloque tiene un punto de entrada que marca el inicio del bloque y que debe estar conectado a la primera tarea que se ejecutará, dentro del bloque y externamente debe formar parte del flujo del script (por ejemplo, conectado a una tarea Goto).

Los scripts se inician siempre en el punto de entrada del bloque 'main', por lo que este bloque no necesita ningún tipo de conexión externa para que sea invocado.



Al arrastrar una tarea dentro de un bloque cuyo punto de entrada no está conectado, automáticamente se conectará la tarea con dicho punto de entrada. Dicha conexión puede ser modificada en cualquier momento, ya sea eliminándola o volviendo a conectar el punto de entrada con otra tarea. Sólo podrá haber una tarea conectada con el punto de entrada del bloque en un momento dado.

Las propiedades del bloque le permiten establecer el nombre (name), el color de fondo y si es interrumpible (interruptible). Cuando un bloque se configura para que no sea interrumpible, veremos que se resalta en rojo su borde. En caso de que dos bloques tengan el mismo nombre o no exista un bloque con nombre 'main' se generará un aviso en la ventana de lista de errores. Por ejemplo si se borra el nombre al bloque 'main' del ejemplo, al guardar se nos informa de los siguientes errores.

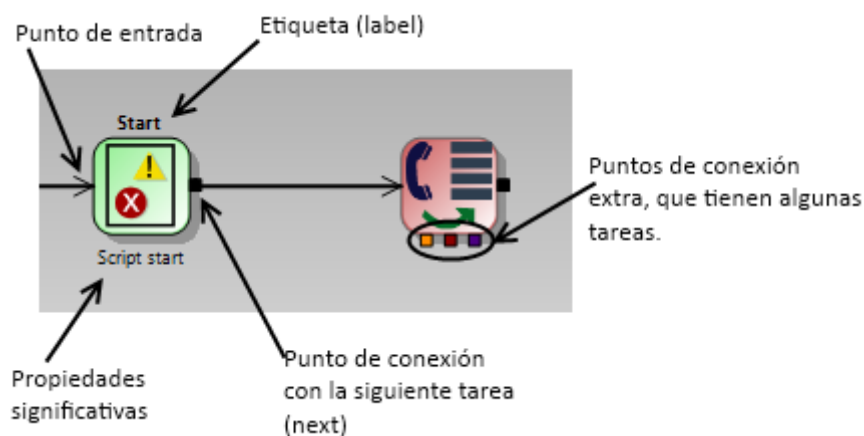


6.4.3 TAREAS

Puede arrastrar cualquier tarea sobre un bloque. Una vez ha soltado la tarea en el bloque deberá conectarla al flujo del script y configurarla adecuadamente.

Una tarea está integrada en el flujo del script si está conectada a otra tarea o al punto de entrada del bloque. Para conectar una tarea con otra lo haremos conectando alguno de sus puntos de conexión con otra tarea.

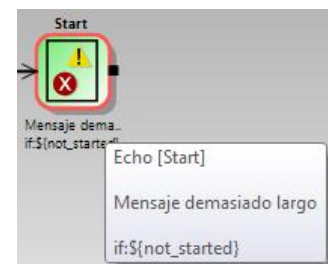
Por defecto todas las tareas tienen un punto (next) que se conecta a la tarea que se ejecutará a continuación de la actual, pero hay algunas tareas como: Goto, Call, Switch... que tienen otros puntos de conexión en la parte inferior y de distintos colores que permiten realizar saltos a otros bloques o tareas si se cumplen ciertas condiciones, en caso contrario saltarán a la siguiente tarea como lo hacen todas las demás.



Para conectar dos tareas pulse sobre el punto de conexión deseado y arrastrelo hasta la tarea de destino.

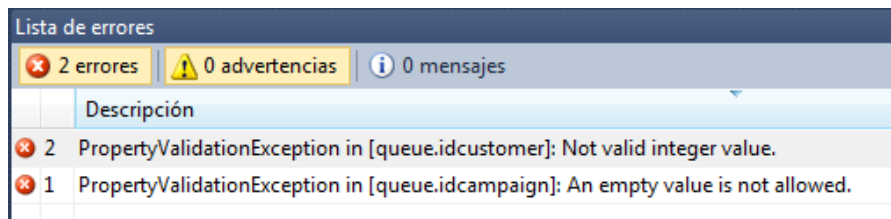
Una vez ha establecido las propiedades de la tarea que ha arrastrado en el diagrama podrá ver la etiqueta (label) y sus propiedades más significativas. Si los textos son muy grandes se truncarán, pero en el tooltip que aparece cuando deja el ratón sobre la tarea podrá ver dicha información.

Si las propiedades *if_condition* o *unless* no tiene su valor por defecto veremos que la tarea nos muestra el valor que tienen asociado y además remarcan la tarea con un borde rojo, indicándonos que esta tarea sólo se ejecutará si se cumple la condición establecida.



Algunas de las propiedades tienen un formato concreto (por ejemplo un número) o son obligatorias, y no pueden estar vacías, en estos casos al guardar el diagrama se nos muestra un mensaje de error y se nos da la oportunidad de guardar igualmente.

En la siguiente imagen vemos que pasa si a una tarea *Queue* no informamos *idcampaign* y escribimos un texto en el campo *idcustomer* que es numérico:

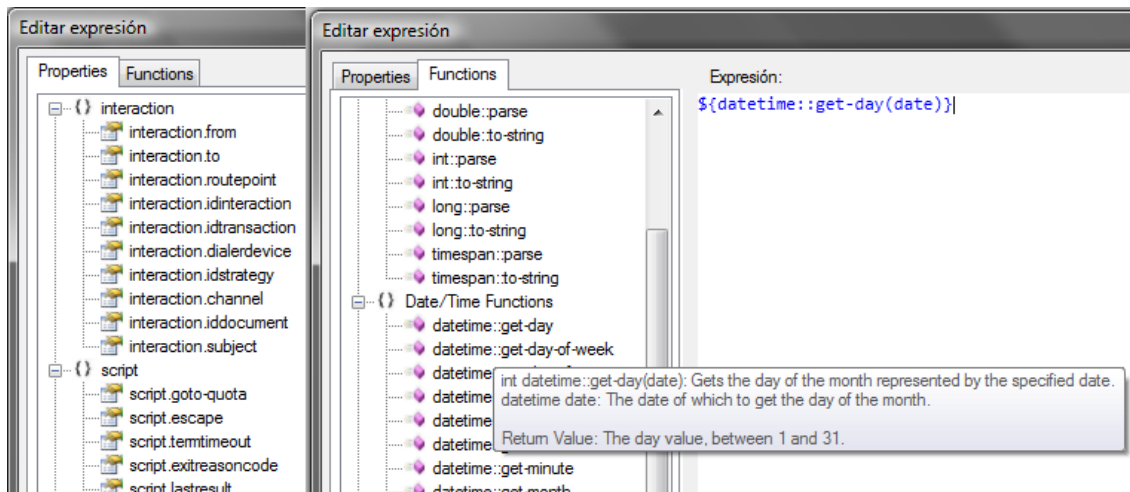


Al hacer doble click sobre cada uno de los mensajes de error se seleccionará la tarea que tiene el error.

6.4.3.1 EDITOR DE EXPRESIONES

La mayoría de las propiedades de una tarea admiten expresiones, es decir que pueden usarse variables y funciones que el entorno pone a disposición del usuario.

Para que editar estas expresiones resulte más fácil se ha creado un editor de expresiones que nos presenta un listado de las propiedades disponibles en el script y de las funciones más comunes.



Al hacer doble click sobre cualquier elemento se incorporará al texto que estamos editando, y si dejamos el ratón sobre ellos tendremos una explicación de su funcionalidad en el ventana emergente.

6.4.4 CONECTORES

Los conectores sirven para establecer el flujo del script, enlazan puntos de entrada de bloques con tareas, tareas con tareas y tareas con bloques.

Para crearlos basta con ir al punto desde el que queremos que parta el conector, veremos como el cursor del ratón cambia, hacemos click y sin soltar el botón arrastramos el ratón hasta el destino.



Cuando se llega a un destino válido el icono vuelve a cambiar y si soltamos se creará la conexión.



Los conectores deben cumplir ciertas restricciones:

- Un punto de entrada de un bloque sólo se puede conectar a una tarea.
- El punto de conexión (next) a la derecha de las tareas sólo se puede conectar con una tarea del mismo bloque, siempre que al conectarlo no se forme un bucle cerrado.

- Los conectores extra (inferiores) no siempre permiten conectarse a otras tareas, en las tareas: Call, Queue, QueueDocument y Create_And_Queue_Document, sólo se pueden conectar con puntos de entrada de bloques.

Al crear los conectores dispone de ciertos atajos que le facilitaran la creación del script:

- **Conectar una tarea con el punto de entrada de un bloque:** no es necesario que arrastre el destino del conector hasta el punto de entrada de un bloque, si lo suelta sobre el bloque la conexión se hará automáticamente.
- **Conectar dos tareas:** si al arrastrar una tarea desde el cuadro de herramientas la suelta sobre otra que está en el diagrama se conectarán automáticamente, si tiene el grid activado depende de en que punto suelte la nueva tarea puede conseguir que se alineen automáticamente. Es recomendable soltarla cerca del punto de conexión *next*.
- **Si pulsa sobre el *Connector* del cuadro de herramientas y hace click sobre el bloque** es como si hubiese iniciado un conector desde el punto de entrada de un bloque.
- **Si pulsa sobre el *Connector* del cuadro de herramientas y hace click sobre una tarea** es como si hubiese iniciado un conector desde el punto de conexión *next*.

6.4.5 COMENTARIOS

Se pueden incorporar mensajes de texto allí donde queramos.

Son elementos del entorno de desarrollo, ya que no tendrán presencia en el script generado.

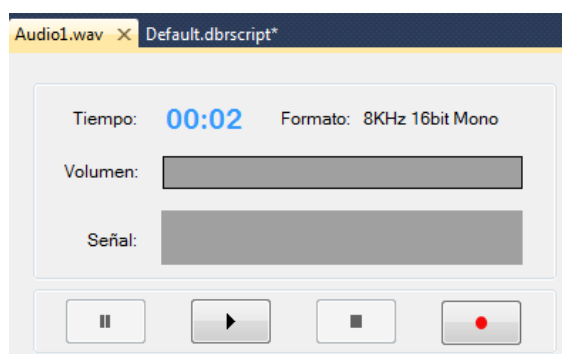
Puede personalizar el color de los comentarios.

Aquí puede escribir sus anotaciones...

6.4.6 EDITOR DE AUDIO

Cuando añadimos una plantilla de elemento de tipo 'Archivo de audio (WAV)' o editamos un fichero de este tipo, se abrirá el siguiente editor, que permite reproducir el archivo o grabar uno nuevo.

Estos audios se graban en el formato requerido por Asterisk: WAV de 8KHz y 16bit Mono.



6.5 REFERENCIA DE LOS ELEMENTOS DBR-SCRIPT

A continuación, se describen los elementos de tipo “tarea” disponibles en el ambiente Developer.NET para el desarrollo de proyectos tipo “DBR script”.

El manual “Integración de aplicaciones con Evolution, Manual de Referencia” contiene información extendida sobre los elementos de programación DBR-Script, que puede serle útil.

Recomendamos que estudie dicha información técnica adicional.

6.5.1 ATRIBUTOS COMUNES

Los siguientes atributos son comunes y aplican a diferentes elementos:

Atributo	Tipo	Descripción	Requerido
label	String	Etiqueta que se puede asignar a la tarea, debe ser única en el bloque en el que se define la tarea. <i>Por defecto no hay ninguna etiqueta asignada.</i>	NO
Failonerror	Bool	Determina si ante un fallo en la tarea el flujo continua o no. <i>Por defecto es TRUE.</i>	NO
Verbose	Bool	Determina si la tarea debe dar información detallada en los ficheros de log. <i>Por defecto es FALSE.</i>	NO
if_condition	Bool	Si es TRUE la tarea será ejecutada, en otro caso no se ejecuta. <i>Por defecto es TRUE.</i>	NO
unless	Bool	Al contrario que if_condition. Si es FALSE la tarea seá ejecutada en otro caso no se ejecuta. <i>Por defecto es FALSE.</i>	NO

6.5.2 BLOQUES

Los bloques son elementos que pueden contener conjuntos de tareas a ejecutar secuencialmente.

Atributos:

Atributo	Tipo	Descripción	Requerido
name	String	Nombre del bloque.	Si
interruptible	Bool	Define si ante un evento (llamada abandonada, enrutamiento de la llamada al agente, timeout en cola...) el flujo dentro de este bloque se interrumpirá. Por defecto es True.	No

6.5.3 PROPIEDADES



Las propiedades son variables que pueden ser usadas en el transcurso de la ejecución del script.

Atributos:

Atributo	Tipo	Descripción	Requerido
name	String	Nombre de propiedad	Si
value	String	Nuevo valor para la propiedad	Si

Se entiende que una propiedad es válida si su nombre es una cadena que contiene los siguientes caracteres:

- Letras (A-Z, a-z),
- Dígitos (0-9),
- Underscore (_),
- Guión (-),
- Punto (.),

Adicionalmente una propiedad valida debe empezar con una letra o un *underscore* debe acabar con una letra, dígito o un *underscore*.

6.5.3.1 ACCESO Y MANIPULACIÓN

Las propiedades pueden referirse con el operador \${}

Ejemplo:

```
<echo message="${property_name}"/>
```

El manejo de las propiedades está basado en el de nant documentado en la siguiente ubicación:

<http://nant.sourceforge.net/release/0.91/help/fundamentals/expressions.html>

6.5.3.2 PROPIEDADES PREESTABLECIDAS

En la ejecución de cada script hay una serie de propiedades referentes a diferentes entidades.

Para la entidad interacción que se refiere a la llamada o contacto multicanal que se está tratando, se podrá acceder a los siguientes atributos:

Nombre	Tipo	Descripción	Readonly
<code>interaction.channel</code>	Numeric	Define código del canal. Posibles valores son: Telefonico = 0, Email = 1, Fax = 2, Web = 3, Sms = 4, Twitter = 5, Unknown = 98, Other = 99	SI
<code>interaction.dialerdevice</code>	Numeric	Dispositivo de marcación asociado.	SI
<code>interaction.from</code>	String	Origen de la llamada o interacción multicanal.	SI
<code>interaction.iddocument</code>	Numeric	Id del documento asociado a la interacción multicanal.	Si
<code>interaction.idinteraction</code>	String	Identificador de la llamada o interacción multicanal.	Si
<code>interaction.idoriginaltransaction</code>	Numeric	identificador de la transacción que genero está transacción, por ejemplo, por una transferencia.	Sí
<code>interaction.idstrategy</code>	Numeric	Identificador de la estrategia actual.	SI

<code>interaction.idtransaction</code>	Numeric	Identificador de la transacción o interacción multicanal.	Si
<code>interaction.routepoint</code>	String	Punto de entrada de la llamada o interacción multicanal.	SI
<code>interaction.subject</code>	String	Asunto del documento asociado a la interacción multicanal.	Si
<code>interaction.to</code>	String	Destino de la llamada o interacción multicanal.	SI
<code>interaction.transaction.close.reason</code>	Numeric	Permite indicar el final que se aplicará a la transacción abierta por el script. Si no se indica una campaña en la estrategia no se creará una transacción automáticamente y no se aplicará este final hasta que no se haga un Queue. Si se encola una interacción en una campaña distinta a la indicada en la estrategia se finaliza la transacción actual con el final indicado y se recetea su valor.	NO
<code>interaction.transaction.close.interval</code>	Numeric	Permite indicar el intervalo de puntualidad en minutos. Este parámetro se aplicará si la transacción se cierra en el script. Para aplicar este parámetro el sujeto debe estar identificado.	NO
<code>interaction.transaction.close.quota</code>	Numeric	Permite indicar el incremento de cuota a aplicar sobre el segmento y la campaña del cliente. Este parámetro se aplicará si la transacción se cierra en el script. Para aplicar este parámetro el sujeto debe estar identificado.	NO

<code>interaction.transaction.close.scheduleon</code>	String	Permite indicar la fecha de planificación del cliente. El formato de fecha y hora debe ser YYYY-MM-DDThh:mm:ss. Este parámetro se aplica si la transacción se cierra en el script. Para aplicar este parámetro el sujeto debe estar identificado.	NO
<code>interaction.userdata</code>	String	Userdata recibida con la llamada.	Si

Para la entidad script que se refiere al motor de ejecución, se podrá acceder a los siguientes atributos:

Name	Tipo	Descripción	Readonly
<code>script.commandtimeout</code>	Numeric	Número de segundos de duración máxima de espera por defecto para las tareas escape_command. Su valor por defecto es 1200, 20 minutos.	No
<code>script.connstring</code>	String	Define la cadena de conexión ODBC por defecto en los accesos a base de datos. Por defecto su valor es 'DSN=EVOLUTIONDB;UID=NCAGENTE;PWD=AGTE1cr@;'.	No
<code>script.culture</code>	String	Define el idioma por defecto del script. Posibles valores son "es-ES", "es-MX", "en-US", etc. Por defecto su valor es 'es-ES'.	No
<code>script.escape</code>	Digit	Escape digit por defecto para task que lo usen y no hayan definido uno propio Su valor por defecto es "".	NO
<code>script.exitreasoncode</code>	Numeric	Código que indica el motivo de finalización del script. Posibles valores son: ScriptExit = 0, RouteEnd = 1, Abandoned = 2, ScriptHangup = 3, Transferred = 4, ScriptTimeout = 100, SystemTimeout = 101, Exception = 102.	SI
<code>script.goto-quota</code>	Numeric	Número máximo de "goto", para evitar bucles infinitos. Esta propiedad NO es readonly para permitir que el propio script la aumente si lo considera necesario.	NO

		Su valor por defecto es 100.	
<code>script.lastresult</code>	Bool	True si la última tarea se ha ejecutado con error. False si falló. Útil para detectar que ha habido un error en una task que se usó FailOnError=False.	SI
<code>script.rec.silencetimeout</code>	Numeric	Número de segundos de silencio tras el cual se detendrá la grabación. Su valor por defecto es 3.	No
<code>script.rec.timeout</code>	Numeric	Número de segundos tras los cuales se detendrá la grabación. Su valor por defecto es 30.	No
<code>script.sr.silencetimeout</code>	Numeric	Número de segundos de silencio tras el cual se detendrá el reconocimiento de voz. Su valor por defecto es 2.	No
<code>script.sr.timeout</code>	Numeric	Número de segundos de timeout por defecto para las task que realicen reconocimiento de voz. Su valor por defecto es 5.	No
<code>script.termtimeout</code>	Numeric	Numero de segundos de timeout por defecto para las task que esperen input por DTMF, 0 significa que se tomará el valor por defecto del Switch. Su valor por defecto es 0.	NO

Para la entidad estrategia, que se refiere a la estrategia de enrutado, se podrá acceder a los siguientes atributos:

Name	Tipo	Descripción	Readonly
<code>strategy.handicap</code>	Numeric	Tiempo en segundos para adelantar/retrasar la posición de la llamada en la cola. Su valor por defecto es 0.	SI
<code>strategy.idagent</code>	Numeric	Identificador del agente Evolution. Su valor por defecto es 0.	SI
<code>strategy.idcampaign</code>	Numeric	Identificador de la campaña de Evolution. Su valor por defecto es 0.	SI
<code>strategy.priority</code>	Numeric	Prioridad en la cola. Su valor por defecto es 0.	SI

6.5.4 COLECCIONES

Las colecciones son tipos complejos que pueden ser usadas definidos y/o usados en el transcurso de la ejecución del script.

6.5.4.1 COLECCIONES PREESTABLECIDAS

Nombre	Tipo	Descripción	Readonly
<code>interaction.iddatarecordset</code>	Ref	Referencia a la colección de data records informada desde la API del conector DBR.	Si
<code>interaction.idskillset</code>	Ref	Referencia a la colección de skills informada desde la API del conector DBR.	Si

6.5.4.2 DATA_RECORD_SET

Para definir una colección de datos se puede hacer uso del elemento “data_record_set” que acepta los siguientes atributos:

Atributos:

Atributo	Tipo	Descripción	Requerido
id	String	Identificador de la colección	Si

Atributos de los elementos internos data_record [0..*] :

Atributo	Tipo	Descripción	Requerido
key	String	Clave del elemento data_record	Si
value	String	Valor del elemento data_record	Si

6.5.4.3 SKILL_SET

Para definir una colección de skills se puede hacer uso del elemento “skill_set” que acepta los siguientes atributos:

Atributos:

Atributo	Tipo	Descripción	Requerido
id	String	Identificador de la colección	Si

Atributos de los elementos internos skill [0..*] :

Atributo	Tipo	Descripción	Requerido
id	String	Identificador del elemento skill	Si
value	String	Valor del elemento skill	Si

6.5.5 TAREAS PARA EL MANEJO DE LAS COLAS

6.5.5.1 QUEUE



Tarea que realiza el encolado de la interacción correspondiente a la llamada que se está tratando. Sólo disponible para interacciones de canal telefónico, para interacciones multicanal ver `queue_document`.

Por defecto la transacción vinculada con la llamada se informa en la propiedad `interaction.idtransaction`.

Si al definir una estrategia de tipo DBR Script se selecciona una campaña, una interacción se puede encolar en Evolution tanto en la campaña seleccionada en la estrategia `${strategy.idcampaign}` como en otra campaña DBR.

En caso de que encolemos a otra campaña, se cerrará la transacción (que había sido creada de manera automática cuando la interacción entró en el script) con el final establecido en la propiedad `${interaction.transaction.closereason}`, y se creará una nueva transacción vinculada a la campaña donde se ha encolado la interacción. Además el valor de `${interaction.transaction.closereason}` se reinicia.

Si no se establece el valor de `${interaction.transaction.closereason}` el sistema determinará el final de manera automática. (abandonada, otras causas, ...)

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>idcampaign</code>	Integer	Identificador de la campaña en la que se quiere encolar la interacción.	SI
<code>idagent</code>	Integer	Identificador del agente que deberá atender la interacción. <i>Por defecto es 0, ningún agente asignado.</i>	NO
<code>idcustomer</code>	Integer	Identificador del sujeto de la interacción. <i>Por defecto es 0, sujeto anónimo.</i>	NO
<code>handicap</code>	Integer	Tiempo, en segundos, que se desea desplazar el momento real de entrada de la interacción en la cola. Admite valores positivos y negativos, los positivos retrasan la fecha de entrada en la cola mientras que los negativos la adelantan. Por ejemplo, si quisiéramos que una interacción entrara en la cola como si llevase 2 minutos esperando indicaríamos un hándicap = -120. <i>Por defecto es 0.</i>	NO
<code>priority</code>	Integer	Indica el orden de prioridad de la interacción, admite valores entre 0 y 100, siendo los valores	NO

		más bajos los más prioritarios, es decir el 0 es la máxima prioridad y el 100 la mínima. <i>Por defecto es 0.</i>	
skill_set_refid	String	Referencia a un skillset declarado previamente. Si no se informa se usara la colección "interaction.skillsetrefid". Si la referencia existe se realizará un "merge" con la colección definida como elemento interno skill_set.	NO
data_record_set_refid	String	Referencia a un datarecordset declarado previamente. Si no se informa se usara la colección "interaction.datarecordsetrefid". Si la referencia existe se realizará un "merge" con la colección definida como elemento interno data_record_set.	NO
on_queue_closed	Block-name	Block que se ejecutará si la cola está cerrada. <i>Por defecto es "", ningún bloque.</i>	NO
on_queue_saturated_full	Block-name	Block que se ejecutará si la cola está llena. <i>Por defecto es "", ningún bloque.</i>	NO
on_queue_saturated_timeout	Block-name	Block que se ejecutará si la interacción supera el tiempo de espera en la cola. <i>Por defecto es "", ningún bloque.</i>	NO

6.5.5.2 QUEUEDOCUMENT



Tarea que realiza el encolado de una interacción multicanal. Por defecto la transacción vinculada con la llamada se informa en la propiedad interaction.idtransaction.

Atributos:

Atributo	Tipo	Descripción	Requerido
idcampaign	Integer	Identificador de la campaña en la que se quiere encolar la interacción.	SI

idagent	Integer	Identificador del agente que deberá atender la interacción. <i>Por defecto es 0, ningún agente asignado.</i>	NO
idcustomer	Integer	Identificador del sujeto vinculado a la interacción. <i>Por defecto es 0, sujeto anónimo.</i>	NO
handicap	Integer	Tiempo, en segundos, que se desea desplazar el momento real de entrada de la interacción en la cola. Admite valores positivos y negativos, los positivos retrasan la fecha de entrada en la cola mientras que los negativos la adelantan. Por ejemplo, si quisiéramos que una interacción entrara en la cola como si llevase 2 minutos esperando indicaríamos un hándicap = -120. <i>Por defecto es 0.</i>	NO
priority	Integer	Indica el orden de prioridad de la interacción, admite valores entre 0 y 100, siendo los valores más bajos los más prioritarios, es decir el 0 es la máxima prioridad y el 100 la mínima. <i>Por defecto es 0.</i>	NO
skill_set_refid	String	Referencia a un skillset declarado previamente. Si no se informa se usará la colección "interacction.skillsetrefid". Si la referencia existe se realizará un "merge" con la colección definida como elemento interno skill_set.	NO
data_record_set_refid	String	Referencia a un datarecordset declarado previamente. Si no se informa se usará la colección "interacction.datarecordsetrefid". Si la referencia existe se realizará un "merge" con la colección definida como elemento interno data_record_set.	NO
on_queue_closed	Block-name	Block que se ejecutará si la cola está cerrada. <i>Por defecto es "", ningún bloque.</i>	NO
on_queue_saturated_full	Block-name	Block que se ejecutará si la cola está llena. <i>Por defecto es "", ningún bloque.</i>	NO

6.5.5.3 CREATEANDQUEUEDOCUMENT



Tarea que da de alta un documento Evolution y realiza el encolado de una interacción multicanal vinculada al documento recientemente creado. Tanto el documento como la interacción recién creados no estarán vinculados a la interacción en curso.

Atributos:

Atributo	Tipo	Descripción	Requerido
from	String	Identificador del origen del documento. <i>Por defecto es el valor asociado a la interacción en curso.</i>	NO
to	String	Identificador del destino del documento. <i>Por defecto es el valor asociado a la interacción en curso.</i>	NO
channel	String	Identificador del canal del documento: 1-Email, 2-Fax, 3-Web, 4-SMS, 98-Unknown, 99-Other. <i>Por defecto es Unknown.</i>	NO
subject	String	Asunto o resumen del documento, máximo 255 caracteres. <i>Por defecto es el valor asociado a la interacción en curso.</i>	NO
mimetype	String	Formato del documento. <i>Por defecto es el valor asociado a la interacción en curso.</i>	NO
filename	String	Ruta completa al fichero con el contenido del documento (debe ser accesible desde el servidor). También se guardará la propiedad 'filename' con el nombre del fichero, salvo que dicha propiedad ya haya sido especificada en la colección atributo document_property_set de esta misma tarea.	NO

idcampaign	Integer	Identificador de la campaña en la que se quiere encolar el documento.	SI
idagent	Integer	Identificador del agente que deberá atender el documento. <i>Por defecto es 0, ningún agente asignado.</i>	NO
idcustomer	Integer	Identificador del sujeto vinculado al documento. <i>Por defecto es 0, sujeto anónimo.</i>	NO
handicap	Integer	Tiempo, en segundos, que se desea desplazar el momento real de entrada del documento en la cola. Admite valores positivos y negativos, los positivos retrasan la fecha de entrada en la cola mientras que los negativos la adelantan. Por ejemplo, si quisiéramos que un documento entrara en la cola como si llevase 2 minutos esperando indicaríamos un hándicap = -120. <i>Por defecto es 0.</i>	NO
priority	Integer	Indica el orden de prioridad del documento, admite valores entre 0 y 100, siendo los valores más bajos los más prioritarios, es decir el 0 es la máxima prioridad y el 100 la mínima. <i>Por defecto es 0.</i>	NO
skill_set_refid	String	Referencia a un skillset declarado previamente. Si no se informa se usará la colección "interaction.skillsetrefid". Si la referencia existe se realizará un "merge" con la colección definida como elemento interno skill_set.	NO
data_record_set_refid	String	Referencia a un datarecordset declarado previamente. Si no se informa se usará la colección "interaction.datarecordsetrefid". Si la referencia existe se realizará un "merge" con la colección definida como elemento interno data_record_set.	NO
iddocument_property	String	Propiedad donde se recibe el Id del nuevo documento creado. <i>Por defecto no se informa ninguna propiedad.</i>	NO

idtransaction_property	String	Propiedad donde se recibe el Id de la nueva transacción creada. <i>Por defecto no se informa ninguna propiedad.</i>	NO
on_queue_closed	Block-name	Block que se ejecutará si la cola está cerrada. <i>Por defecto es "", ningún bloque.</i>	NO
on_queue_saturated_full	Block-name	Block que se ejecutará si la cola está llena. <i>Por defecto es "", ningún bloque.</i>	NO

6.5.5.4 QUEUEUPDATE



Tarea que realiza la actualización de los parámetros de cola de una interacción.

Atributos:

Atributo	Tipo	Descripción	Requerido
idagent	Integer	Identificador del agente que deberá atender la interacción. <i>Por defecto es 0, ningún agente asignado.</i>	NO
handicap	Integer	Tiempo, en segundos, que se desea desplazar el momento real de entrada de la interacción en la cola. Admite valores positivos y negativos, los positivos retrasan la fecha de entrada en la cola mientras que los negativos la adelantan. Por ejemplo, si quisiéramos que una interacción entrara en la cola como si llevase 2 minutos esperando indicaríamos un hándicap = -120. <i>Por defecto es 0.</i>	NO
priority	Integer	Indica el orden de prioridad de la interacción, admite valores entre 0 y 100, siendo los valores más bajos los más prioritarios, es decir el 0 es la máxima prioridad y el 100 la mínima. <i>Por defecto es 0.</i>	NO

<code>skill_set_refid</code>	String	Referencia a un skillset declarado previamente. Si no se informa se usara la colección “ <code>interaccion.skillsetrefid</code> ”. Si la referencia existe se realizará un “merge” con la colección definida como elemento interno <code>skill_set</code> .	NO
------------------------------	--------	---	----

6.5.5.5 DEQUEUE



Tarea que elimina la interacción en curso de la cola de la campaña.

En un momento dado la interacción en curso solamente puede estar en una cola, y tener una única transacción, así que la cola está implícita.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>dequeue_reason</code>	Integer	Identificador del final que se dará a la transacción desencolada.	SI

6.5.5.6 GETQUEUEPOSITION



Tarea que actualiza propiedades con información de la posición en cola.

Por defecto la información se recibe en las propiedades:

- `queueposition.pos` posición que ocupa la interacción en la cola.
- `queueposition.ewt` Propiedad donde se recibe el tiempo (en segundos) de espera estimado que aún le queda a la interacción antes de ser atendida. Existen momentos en los que el sistema no puede determinar este valor (por ejemplo, cuando no hay agentes conectados), en cuyo caso devuelve -1.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>idcampaign</code>	Integer	Identificador de la campaña en la que se quiere recoger la información de la posición de la	SI

		interacción.	
<code>pos_property</code>	String	Propiedad donde se recibe la posición que ocupa la interacción en la cola. <i>Por defecto no se informa ninguna propiedad.</i>	NO
<code>ewt_property</code>	String	Propiedad donde se recibe el tiempo (en segundos) de espera estimado que aún le queda a la interacción antes de ser atendida. Existen momentos en los que el sistema no puede determinar este valor (por ejemplo, cuando no hay agentes conectados), en cuyo caso devuelve -1. <i>Por defecto no se informa ninguna propiedad.</i>	NO

6.5.5.7 GETCAMPAIGNSTATS



Tarea que actualiza propiedades con información estadística de la campaña.

Por defecto la información se recibe en las propiedades:

- `campaignstats.queuedinteractions` Número de interacciones en cola.
- `campaignstats.mwt` (Maximum Wait Time) Tiempo (en segundos) que lleva en la cola la interacción más antigua.
- `campaignstats.ewt` (Expected Wait Time) Tiempo (en segundos) de espera estimado para las nuevas interacciones que puedan entrar en la cola. Existen momentos en los que el sistema no puede determinar este valor (por ejemplo, cuando no hay agentes conectados), en cuyo caso devuelve -1, para indicar que es un valor desconocido.
- `campaignstats.insertwhenclosed` Indica si la campaña está configurada para encolar interacciones en la cola si está cerrada.
- `campaignstats.activeagents` Número de agentes que están en conectados a la campaña en alguno de los siguientes estados: Disponible, Vista previa, Llamada, Hold o Tiempo administrativo.
- `campaignstats.connectedagents` Número de agentes conectados a la campaña.
- `campaignstats.restrictedinteractions` Número de interacciones encoladas que tienen algún requerimiento de agente o skill.
- `campaignstats.restrictedinteractionsagent` Número de interacciones encoladas que requieren un agente en concreto.
- `campaignstats.restrictedinteractionsskill` Número de interacciones encoladas que requieren algún skill.

Atributos:

Atributo	Tipo	Descripción	Requerido
idcampaign	Integer	Identificador de la campaña de la que se quiere recoger la información.	SI
queuedinteractions_property	String	Propiedad donde se recibe el número de interacciones en cola. <i>Por defecto no se informa ninguna propiedad.</i>	NO
mwt_property	String	Propiedad donde se recibe el tiempo (en segundos) que lleva en la cola la interacción más antigua. <i>Por defecto no se informa ninguna propiedad.</i>	NO
ewt_property	String	Propiedad donde se recibe el tiempo (en segundos) de espera estimado para las nuevas interacciones que puedan entrar en la cola. Existen momentos en los que el sistema no puede determinar este valor (por ejemplo, cuando no hay agentes conectados), en cuyo caso devuelve -1, para indicar que es un valor desconocido. <i>Por defecto no se informa ninguna propiedad.</i>	NO
insertwhenclosed_property	String	Propiedad donde se recibe si la campaña está configurada para encolar interacciones en la cola si está cerrada. <i>Por defecto no se informa ninguna propiedad.</i>	NO
activeagents_property	String	Propiedad donde se recibe el número de agentes que están en conectados a la campaña en alguno de los siguientes estados: Disponible, Vista previa, Llamada, Hold o Tiempo administrativo. <i>Por defecto no se informa ninguna propiedad.</i>	NO
connectedagents_property	String	Propiedad donde se recibe el número de agentes conectados a la campaña. <i>Por defecto no se informa ninguna propiedad.</i>	NO
restrictedinteracti	String	Propiedad donde se recibe el número de	NO

<code>ons_property</code>		interacciones encoladas que tienen algún requerimiento de agente o skill. <i>Por defecto no se informa ninguna propiedad.</i>	
<code>restrictedinteractionsagent_property</code>	String	Propiedad donde se recibe el número de interacciones encoladas que requieren un agente en concreto. <i>Por defecto no se informa ninguna propiedad.</i>	NO
<code>restrictedinteractionskill_property</code>	String	Propiedad donde se recibe el número de interacciones encoladas que requieren algún skill. <i>Por defecto no se informa ninguna propiedad.</i>	

6.5.6 TAREAS PARA EL MANEJO DE LLAMADAS

6.5.6.1 ANSWER



Tarea que señala que la llamada ha sido contestada. Sólo disponible para interacciones de canal telefónico.
No se requiere atributos.

6.5.6.2 TRANSFER_TO



Tarea que transfiere la llamada a la extensión indicada, la transferencia se realizará a ciegas. Sólo disponible para interacciones de canal telefónico.

Tras esta tarea el script finalizará ejecutando el bloque “on_exit” con “script.exitreasoncode” = 4 (Transferred).

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>to</code>	String	Destino de la llamada.	SI

6.5.6.3 HANGUP



Tarea que cuelga la llamada. Sólo disponible para interacciones de canal telefónico. Tras esta tarea el script finalizará ejecutando el bloque “on_exit” con “script.exitreasoncode” = 3 (ScriptHangup).

Atributos:

Atributo	Tipo	Descripción	Requerido
hangup_reason	Integer	Código de hangup que se indicará al switch. Por defecto se usará el código por defecto del switch.	NO

6.5.6.4 PLAY



Tarea que emite la locución indicada por filename al interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
filename	String	Archivo de audio que se emitirá.	SI
escape	String	Caracteres de escape, si se pulsan se termina la función. <i>Por defecto no hay carácter de escape.</i>	NO

El formato de filename es URI y se permiten los siguientes usos:

- URI relativa a la ubicación del script, por ejemplo resources/audio/menu1.wav. Recordar que para el uso de URIs relativas para Asterisk es necesario que las locuciones se sitúen bajo la ubicación relativa al script resources/audio/ y se sincronicen.
- URI con protocolo local:// para establecer una ruta absoluta del servidor Evolution.
- URI absoluta referente al sistema de ficheros del SWITCH, por ejemplo /menu1.wav reproduciría /menu1.wav del SWITCH.
- Para Asterisk, URI con protocolo sounds:// ubicación relativa al directorio sounds de Asterisk.

Recordar que, para Asterisk, con URIs que refieran al sistema de fichero de Asterisk (absolutas o por protocolo sounds://) los nombres de fichero deben indicarse sin extensión.

Para Asterisk, los ficheros deben tener un formato aceptado por Asterisk, recomendamos que se usen locuciones en formato WAV PCM de 16bits y 8KHz.

6.5.6.5 PLAY_TTS



Tarea que sintetiza texto a voz y la emite al interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>text</code>	String	Texto que se sintetizará a voz y se emitirá.	NO
<code>culture</code>	String	Cultura para filtrar el locutor/a que se desea usar para sintetizar el texto. Posibles valores son "es-ES", "es-MX", "en-US", etc. <i>Por defecto es "es-ES".</i>	NO
<code>gender</code>	String	Genero para filtrar el locutor/a que se desea usar para sintetizar el texto. <i>Por defecto es vacío y no se filtra por este campo para elegir el locutor/a.</i>	NO
<code>voice_name</code>	String	Nombre de la voz para filtrar el locutor/a que se desea usar para sintetizar el texto. <i>Por defecto es vacío y no se filtra por este campo para elegir el locutor/a.</i>	NO
<code>escape</code>	String	Caracteres de escape, si se pulsan se termina la función. <i>Por defecto no hay carácter de escape.</i>	NO
<code>ssml</code>	String	Elemento opcional que puede referir a un archivo de recurso que contenga definiciones de gramáticas en lenguaje SSML. Este lenguaje es un estándar que permite realizar síntesis de voz avanzada. Puede encontrar más información en: http://msdn.microsoft.com/en-us/library/hh378377.aspx o http://www.w3.org/TR/speech-synthesis/ En el caso que se informe ssml se ignorarán los elementos text, cultura, gender y voice_name. <i>Por defecto vacío.</i>	NO

6.5.6.6 SAY_DIGITS



Tarea que emite los dígitos indicados al interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
digits	String	Dígitos que se emitirán.	SI
escape	String	Caracteres de escape, si se pulsan se termina la función. <i>Por defecto no hay carácter de escape.</i>	NO

6.5.6.7 SAY_NUMBER



Tarea que emite el número indicado al interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
number	String	Número que se emitirá.	SI
escape	String	Caracteres de escape, si se pulsan se termina la función. <i>Por defecto no hay carácter de escape.</i>	NO

6.5.6.8 SAY_DATETIME



Tarea que emite una fecha y/o hora al interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
date	String	Fecha que se emitirá. Debe cumplir el formato ISO básico para fechas yyyyMMdd. Si no se informa no se emitirá la fecha.	NO
time	String	Tiempo que se emitirá. Debe cumplir el formato ISO para horario hhmmss, hhmm, hh:mm:ss o hh:mm. Si no se informa no se emitirá el tiempo.	NO
escape	String	Caracteres de escape, si se pulsan se termina la función. <i>Por defecto no hay carácter de escape.</i>	NO

6.5.6.9 GIVE_MUSIC



Tarea que emite música en espera al interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
on	Bool	True activa la música en espera. False la detiene. Por defecto es true.	NO
class	String	Clase de música en espera. <i>Por defecto se reproduce la música en espera por defecto del switch.</i>	NO

El formato del atributo class es URI y se permiten los siguientes usos:

- URI relativa a la ubicación del script, esta URI debe referenciar una carpeta estará bajo `resources/moh/` y que contiene los ficheros de audio que componen la música en espera, por ejemplo, si `class=beethovenclass` referirá a `resources/moh/beethovenclass`.
Recordar que para el uso de URIs relativas para Asterisk es necesario que las carpetas que contienen los archivos de audio para música en espera se sitúen bajo la ubicación relativa al script `resources/moh/` y se sincronicen.
- Para Asterisk, URI con protocolo `moh://` se refiere a una clase de música en espera ya definida en Asterisk.

Para Asterisk, los ficheros deben tener un formato aceptado por Asterisk, recomendamos que se usen locuciones en formato WAV PCM de 16bits y 8KHz.

6.5.6.10 GIVE_RINGING



Tarea da tono de llamada al interlocutor. Sólo disponible para interacciones de canal telefónico.
No se requiere atributos.

6.5.6.11 GIVE_BUSY



Tarea da tono de ocupado al interlocutor. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>timeout</code>	Integer	Segundos durante los cuales se da tono de ocupado. <i>Por defecto la tarea da tono durante 0 segundos.</i>	NO

6.5.6.12 PLAY_AND_GET_DIGITS



Tarea que emite la locución indicada por filename si se indica y recoge los dígitos DTMF pulsados por el interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>filename</code>	String	Archivo de audio que se emitirá. <i>Si no se indica fichero no se emitirá audio.</i>	NO
<code>received_digits_property</code>	String	Propiedad que recibirá los dígitos. <i>Por defecto no se informa ninguna propiedad.</i>	NO
<code>max_digits</code>	Integer	Número de dígitos que se espera recibir. <i>Por defecto no se esperará recibir una cantidad fija de dígitos.</i>	NO
<code>attempts</code>	Integer	Número de veces que se repetirá el mensaje si no se entra ningún dígito. <i>Por defecto 1.</i>	NO
<code>timeout</code>	Integer	Tiempo a esperar la entrada de dígitos tras oír la locución. <i>Por defecto se usará la propiedad <code>script.termtimeout</code>.</i>	NO

El formato de filename es URI y se permiten los siguientes usos:

- URI relativa a la ubicación del script, por ejemplo `resources/audio/menu1.wav`.
Recordar que para el uso de URIs relativas para Asterisk es necesario que las locuciones se sitúen bajo la ubicación relativa al script `resources/audio/` y se sincronicen.
- URI con protocolo `local://` para establecer una ruta absoluta del servidor Evolution.
- URI absoluta referente al sistema de ficheros del SWITCH, por ejemplo `/menu1.wav` reproduciría `/menu1.wav` del SWITCH.
- Para Asterisk, URI con protocolo `sounds://` ubicación relativa al directorio `sounds` de Asterisk.

Recordar que, para Asterisk, con URIs que refieran al sistema de fichero de Asterisk (absolutas o por protocolo `sounds://`) los nombres de fichero deben indicarse sin extensión.

Para Asterisk, los ficheros deben tener un formato aceptado por Asterisk, recomendamos que se usen locuciones en formato WAV PCM de 16bits y 8KHz.

6.5.6.13 PLAY_TTS_AND_GET_DIGITS



Tarea que sintetiza texto a voz y la emite al interlocutor de la llamada si se indica y recoge los dígitos DTMF pulsados por el interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>text</code>	String	Texto que se sintetizará a voz y se emitirá.	NO
<code>culture</code>	String	Cultura para filtrar el locutor/a que se desea usar para sintetizar el texto. Posibles valores son “es-ES”, “es-MX”, “en-US”, etc. <i>Por defecto es “es-ES”.</i>	NO
<code>gender</code>	String	Genero para filtrar el locutor/a que se desea usar para sintetizar el texto. <i>Por defecto es vacío y no se filtra por este campo para elegir el locutor/a.</i>	NO
<code>voice_name</code>	String	Nombre de la voz para filtrar el locutor/a que se desea usar para sintetizar el texto. <i>Por defecto es vacío y no se filtra por este campo para elegir el locutor/a.</i>	NO
<code>received_digits_property</code>	String	Propiedad que recibirá los dígitos. <i>Por defecto no se informa ninguna propiedad.</i>	NO
<code>max_digits</code>	Integer	Número de dígitos que se espera recibir. <i>Por defecto no se esperará recibir una cantidad fija de dígitos.</i>	NO
<code>attempts</code>	Integer	Número de veces que se repetirá el mensaje si no se entra ningún dígito. <i>Por defecto 1.</i>	NO
<code>timeout</code>	Integer	Tiempo a esperar la entrada de dígitos tras oír la locución. <i>Por defecto se usará la propiedad <code>script.termtimeout</code>.</i>	NO

ssml	String	Elemento opcional que puede referir a un archivo de recurso que contenga definiciones de gramáticas en lenguaje SSML. Este lenguaje es un estándar que permite realizar síntesis de voz avanzada. Puede encontrar más información en: http://msdn.microsoft.com/en-us/library/hh378377.aspx o http://www.w3.org/TR/speech-synthesis/ En el caso que se informe ssml se ignorarán los elementos text, cultura, gender y voice_name. <i>Por defecto vacío.</i>	NO
-------------	--------	--	----

6.5.6.14 PLAY_AND_GET_SR



Tarea que emite la locución indicada por filename si se indica y realiza reconocimiento del habla y/o los dígitos DTMF pulsados por el interlocutor de la llamada. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
filename	String	Archivo de audio que se emitirá. <i>Si no se indica fichero no se emitirá audio.</i>	NO
digits_mask	String	Mascara de dígitos que se espera recibir para iniciar el reconocimiento, sin tener que esperar al silence_timeout. Por ejemplo "123" espera recibir los dígitos 1, 2 o 3. <i>Por defecto no se esperará recibir ningún dígito.</i>	NO
attempts	Integer	Número de veces que se repetirá el mensaje si no se reconoce una respuesta o no se entra ningún dígito de la máscara. <i>Por defecto 1.</i>	NO
attempts_threshold_confidence	Double	Si la confianza (confidence) en la respuesta de voz recibida no supera este umbral se continuará con el proceso de reintentos. <i>Por defecto 0,7.</i>	NO

beep	Bool	Determina si se emitirá un 'beep' antes de empezar el periodo de reconocimiento de voz. <i>Por defecto true.</i>	NO
timeout	Integer	Tiempo a esperar la entrada de dígitos o voz tras oír la locución y/o beep. <i>Por defecto se usará la propiedad script.sr.timeout.</i>	NO
silence_timeout	Integer	Tiempo máximo de silencio en el periodo de reconocimiento de voz o dígitos. <i>Por defecto se usará la propiedad script.sr.silencetimeout.</i>	NO
culture	String	Cultura de la respuesta que se espera reconocer. Posibles valores son "es-ES", "es-MX", "en-US", etc. <i>Por defecto es "es-ES".</i>	NO
grammar_filename	String	Nombre del fichero donde se define la gramática para el reconocimiento de voz. <i>Por defecto no se define gramática.</i>	NO
grammar_rule_id	String	Id de la regla de gramática del fichero definido por grammar_filename. <i>Por defecto no se define.</i>	NO
simple_choice_grammar	String	Alternativa simple para definir una gramática en base a una lista de frases separadas por comas. Por ejemplo: "si,no". <i>Por defecto no se define.</i>	NO
received_digits_property	String	Propiedad que recibirá los dígitos. (En configuraciones Asterisk se recogerá un dígito) <i>Por defecto no se informa ninguna propiedad.</i>	NO
recognized_text_property	String	Propiedad donde se informará el texto reconocido, si la cadena está vacía significa que no se ha reconocido la respuesta. <i>Por defecto no se define.</i>	NO
recognized_text_confidence_property	String	Propiedad donde se informará la confianza en el texto reconocido. <i>Por defecto no se define.</i>	NO

semantic_result_refid	Ref	Referencia a la colección donde se devolverán los resultados. Se puede acceder a estos resultados mediante la función <code>key-value-set::get-value(id, parametro_de_resultado)</code> . <i>Por defecto no se define.</i>	NO
------------------------------	-----	--	----

Puede encontrar más información acerca de las gramáticas en <http://msdn.microsoft.com/en-us/library/hh378341> y en <http://www.w3.org/TR/speech-grammar/>.

En el caso que se informe el atributo “grammar_filename”, no se tendrá en cuenta la gramática definida en el atributo “simple_choice_grammar”.

6.5.6.15 ESCAPE_COMMAND



Tarea que envía el comando en bruto al switch.

Atributos:

Atributo	Tipo	Descripción	Requerido
command	String	Comando que se enviará al switch.	SI
result_refid	Ref	Referencia a la colección donde se devolverán los resultados. Se puede acceder a estos resultados mediante la función <code>key-value-set::get-value(id, parametro_de_resultado)</code> .	SI
command_timeout	Integer	Tiempo total máximo de espera a la finalización del comando de escape en segundos. <i>Por defecto se usará la propiedad script.commandtimeout, por defecto el valor de esta propiedad son 20min.</i>	NO

Recordar que, para Asterisk, los comandos que pueden usarse son los correspondientes a la interface AGI. Puede encontrar más información acerca de AGI en <http://www.voip-info.org/wiki/view/Asterisk+AGI>.

Al estar codificando en XML los caracteres comilla doble dentro de los atributos deben escaparse con "

Ejemplo 1:

```
<escape_command command="STREAM FILE &quot;hello-world&quot; &quot;&quot;"
result_refid="res1"/>

<echo message="res1-result: ${key-value-set::get-value('res1','result')}" />
```

6.5.6.16 AMD



La tarea AMD (Answering Machine Detection) intenta detectar si el interlocutor es un contestador automático en el inicio de llamadas salientes. Esta tarea está especialmente pensada para ser usada al principio del flujo de una llamada saliente con marcación automática. Sólo disponible para interacciones de canal telefónico.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>initial_silence_ms</code>	Integer	Define el máximo tiempo silencio antes del saludo (greeting) en milisegundos. Si se excede se considera que el interlocutor es un contestador automático. <i>Por defecto 2500 milisegundos.</i>	NO
<code>greeting_ms</code>	Integer	Define el tiempo máximo del saludo. Si se excede se considera que el interlocutor es un contestador automático. <i>Por defecto 1500 milisegundos.</i>	NO
<code>after_greeting_silence_ms</code>	Integer	Define el tiempo máximo de silencio tras el saludo. Si se excede se considera que el interlocutor es un humano. <i>Por defecto 800 milisegundos.</i>	NO
<code>total_analysis_time_ms</code>	Integer	Define el tiempo máximo durante el cual la task puede decidir entre humano o contestador automático. <i>Por defecto 5000 milisegundos.</i>	NO
<code>minimum_word_length_ms</code>	Integer	Define cuál es la duración de voz mínima para considerar que es una palabra. <i>Por defecto 100 milisegundos.</i>	NO

<code>between_words_silence_ms</code>	Integer	Define cuál es la duración de voz entre palabras para considerar que el audio que procede es una nueva palabra. <i>Por defecto 50 milisegundos.</i>	NO
<code>maximum_number_of_words</code>	Integer	Define el máximo número de palabras contenidas en el saludo. Si se excede se considera que el interlocutor es un contestador automático. <i>Por defecto 3 palabras.</i>	NO
<code>silence_threshold</code>	Integer	Define el nivel de sonido por debajo del cual se considera que es silencio y por encima del cual se considera que es voz. <i>Por defecto 512.</i>	NO
<code>maximum_word_length_ms</code>	Integer	Define cuál es la duración de voz máxima para considerar que es una palabra. <i>Por defecto 5000 milisegundos.</i>	NO
<code>hangup_if_machine</code>	Boolean	Si el valor de este atributo es 'true' se cuelga la llamada automáticamente y Evolution la finaliza con final "Contestador Automático". <i>Por defecto false.</i>	NO
<code>amd_result_property</code>	String	Propiedad donde se informará el resultado de la detección. Los posibles valores son: HUMAN, MACHINE, NOTSURE. <i>Por defecto no se informa ninguna propiedad.</i>	NO

6.5.6.17 RECORD



Tarea que graba el audio de la llamada en formato WAV.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>filename</code>	String	Nombre del fichero donde se almacenará la grabación.	SI

<code>beep</code>	Bool	Determina si se emitirá un 'beep' antes de empezar el periodo de reconocimiento de voz. <i>Por defecto true.</i>	NO
<code>escape</code>	String	Caracteres de escape. Si se pulsa uno de ellos, se interrumpe la grabación. <i>Por defecto no hay carácter de escape.</i>	NO
<code>timeout</code>	Integer	Transcurrido el número de segundos especificado, la grabación finaliza. <i>Por defecto se usará la propiedad script.rec.timeout.</i>	NO
<code>silence_timeout</code>	Integer	Si se detecta un silencio de duración el número de segundos especificado, la grabación finaliza. <i>Por defecto se usará la propiedad script.rec.silencetimeout.</i>	NO

El formato de filename es URI y se permiten los siguientes usos:

- URI relativa a la ubicación del script, por ejemplo resources/rec1.wav.
- URI con protocolo local:// para establecer una ruta absoluta del servidor Evolution, por ejemplo local://c:\rec1.wav
- URI absoluta referente al sistema de ficheros del SWITCH, por ejemplo /rec1.wav (con extensión)
- Para Asterisk, URI con protocolo sounds:// ubicación relativa al directorio sounds de Asterisk, por ejemplo sounds://rec1.wav (con extensión)

6.5.7 TAREAS DE ACCESO A DATOS

6.5.7.1 SQL



Tarea que ejecuta sentencias SQL contra una base de datos especificada.

Por defecto la información se recibe en la propiedad:

- `sql.affected_rows` Número de filas afectadas.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>connstring</code>	String	Cadena de conexión ODBC usada para acceder a la base de datos. <i>Por defecto se usará la propiedad <code>script.connstring</code>.</i>	NO
<code>cmdtimeout</code>	Integer	Timeout para la ejecución de los comandos SQL.	NO
<code>statement</code>	String	Comando SQL a ejecutar. Permite indicar múltiples sentencias SQL separadas por punto y coma (;)	SI
<code>affected_rows_property</code>	String	Propiedad que recibirá el número de filas afectadas.	NO
<code>result_refid</code>	Ref	Referencia a la colección donde se recibirán los resultados. Se puede acceder a estos resultados mediante las funciones: - <code>data-table-type::get-value(id, fila, col)</code>. - <code>data-table-type::get-row-count(id)</code>.	NO

6.5.7.2 GETCUSTOMERFIELD



Tarea que obtiene un dato de cliente de base de datos.

Por defecto la información se recibe en la propiedad:

- `getcustomerfield.field` Dato del cliente solicitado.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>source</code>	String	Admite los valores 0 para obtener un dato genérico del cliente (tabla CLIENTES) y 1 para obtener un dato adicional del cliente (tabla <code>tbDatosCliente</code>).	SI

key	String	Identificador del dato que se quiere obtener. Corresponde con un campo de la tabla CLIENTES o un dato adicional contenido en tbDatosCliente.	SI
defaultvalue	String	Valor por defecto en caso de no encontrar el dato adicional del cliente.	NO
idsujeto	String	Identificador Evolution del cliente. Si no se informa, se hará uso del identificador de cliente de la transacción en curso.	NO
field_property	String	Propiedad donde se informará el dato del cliente. <i>Por defecto no se informa ninguna propiedad.</i>	NO

6.5.7.3 SETCUSTOMERFIELD



Tarea que informa un dato de cliente en base de datos.

Atributos:

Atributo	Tipo	Descripción	Requerido
source	String	Admite los valores 0 para guardar un dato genérico del cliente (tabla CLIENTES) y 1 para guardar un dato adicional del cliente (tabla tbDatosCliente).	SI
key	String	Identificador del dato que se quiere guardar. Corresponde con un campo de la tabla CLIENTES o un dato adicional contenido en tbDatosCliente.	SI
value	String	Valor que se quiere guardar.	SI
idsujeto	String	Identificador Evolution del cliente. Si no se informa, se hará uso del identificador de cliente de la transacción en curso.	NO

6.5.7.4 GETCUSTOMERDATASET



Tarea que obtiene de base de datos un conjunto de datos de un cliente.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>source</code>	String	Admite los valores 0 para obtener los datos genéricos del cliente (tabla CLIENTES) y 1 para obtener los datos adicionales del cliente (tabla tbDatosCliente).	SI
<code>idsujeto</code>	String	Identificador Evolution del cliente. Si no se informa, se hará uso del identificador de cliente de la transacción en curso.	NO
<code>result_refid</code>	Ref	Referencia a la colección donde se recibirán los resultados. Se puede acceder a estos resultados mediante las funciones: - <code>data-table-type::get-value(id, fila, col)</code>. - <code>data-table-type::get-row-count(id)</code>.	NO

6.5.7.5 GETTRANSACTIONVALUE



Tarea que obtiene un dato de la transacción del repositorio de Evolution Server.

El tiempo de vida de estos datos es la transacción y no persisten en base de datos.

En transferencias desde softphone integrado los datos de transacción se copian de la transacción original a la transacción transferida.

Por defecto la información se recibe en la propiedad:

- `gettransactionvalue.value` Dato de la transacción asociada.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>key</code>	String	Identificador del dato que se quiere guardar. Corresponde con un clave de transacción.	SI

defaultvalue	String	Valor por defecto en caso de no encontrar la clave de la transacción.	NO
transaction	int	Transacción sobre la que se quiere obtener el valor. Posibles valores: 0-Current 1-Original Valor por defecto 0-Current.	NO
value_property	String	Propiedad donde se informará el dato de la transacción. <i>Por defecto no se informa ninguna propiedad.</i>	NO

6.5.7.6 SETTRANSACTIONVALUE



Tarea que informa un dato de la transacción del repositorio de Evolution Server.

El tiempo de vida de estos datos es la transacción y no persisten en base de datos.

En transferencias desde softphone integrado los datos de transacción se copian de la transacción original a la transacción transferida.

Atributos:

Atributo	Tipo	Descripción	Requerido
key	String	Identificador del dato que se quiere guardar. Corresponde con un clave de transacción.	SI
value	String	Valor que se quiere guardar.	SI
transaction	int	Transacción sobre la que se quiere asignar el valor. Posibles valores: 0-Current 1-Original Valor por defecto 0-Current.	NO

6.5.8 TAREAS PROPIAS DEL LENGUAJE

6.5.8.1 GOTO



Tarea que permite ir a otro bloque y/o tarea para continuar desde ahí el flujo del script.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>block</code>	String	Nombre del siguiente bloque a ejecutar. Si no se informa será el actual.	NO
<code>task_label</code>	String	Id de la tarea a ejecutar dentro del bloque elegido. Si no se informa se ejecutará la primera tarea del bloque elegido.	NO

6.5.8.2 ECHO



Tarea que permite escribir un mensaje en el log de salida.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>message</code>	String	Mensaje a escribir en el log de salida.	SI

6.5.8.3 SLEEP



Tarea que permite pausar el flujo del script por un tiempo determinado.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>hours</code>	Integer	Horas a añadir al tiempo que pausa. <i>Por defecto 0.</i>	NO

minutes	Integer	Minutos a añadir al tiempo que pausa. <i>Por defecto 0.</i>	NO
seconds	Integer	Segundos a añadir al tiempo que pausa. <i>Por defecto 0.</i>	NO
milliseconds	Integer	Milisegundos a añadir al tiempo que pausa. <i>Por defecto 0.</i>	NO
interruptible	Bool	Define si la pausa puede ser interrumpida por un evento, como un hangup o RouteEnd. <i>Por defecto TRUE, despierta cuando vence el tiempo o ante un evento.</i>	NO

6.5.8.4 EXIT



Tarea que finaliza la ejecución del script sin errores.

Tras esta tarea el script finalizará ejecutando el bloque "on_exit" con "script.exitreasoncode" = 0 (ScriptExit).

No se requiere atributos.

6.5.8.5 FAIL



Tarea que fuerza la finalización de la ejecución del script con errores.

Tras esta tarea el script finalizará ejecutando el bloque "on_exit" con "script.exitreasoncode" = 102 (Exception).

Atributos:

Atributo	Tipo	Descripción	Requerido
message	String	Texto del mensaje descriptivo del motivo por el que se ha finalizado la ejecución del script.	NO

6.5.8.6 MAIL



Tarea que permite enviar un mail.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>from</code>	String	Dirección de origen del email.	SI
<code>tolist</code>	String	Lista de destinatarios del email, separados por punto y coma.	NO
<code>cclist</code>	String	Lista de destinatarios de una copia del email, separados por punto y coma.	NO
<code>bcclist</code>	String	Lista de destinatarios de una copia oculta del email, separados por punto y coma.	NO
<code>subject</code>	String	Asunto del email.	NO
<code>attachment</code>	String	Lista de archivos adjuntos, separados por punto y coma. Cada adjunto debe incluir la ruta completa, que debe ser accesible desde el servidor Evolution.	NO
<code>message</code>	String	Cuerpo del email.	NO
<code>format</code>	string	Formato del mensaje. Texto ('text', valor por defecto) o HTML ('html').	NO
<code>mailhost</code>	String	Nombre del servidor de mail.	NO
<code>port</code>	Integer	Puerto del servidor de mail.	NO
<code>enablessl</code>	Bool	Indica si se usará SSL	NO
<code>user</code>	String	Usuario del email.	NO
<code>password</code>	String	Contraseña del usuario del email.	NO

Más info: <http://nant.sourceforge.net/release/0.85-rc2/help/tasks/mail.html>

6.5.8.7 TSTAMP



Tarea que permite obtener la fecha/hora actual.

Por defecto la información se recibe en la propiedad:

- `tstamp.date` Sólo fecha actual.
- `tstamp.time` Sólo hora actual.
- `tstamp.now` Sólo fecha/hora actual

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>pattern</code>	String	El patrón del formato de fecha/hora	NO
<code>property</code>	String	Propiedad que recibirá la fecha/hora.	NO

Mas info: <http://nant.sourceforge.net/release/0.90/help/tasks/tstamp.html>

6.5.8.8 CALL



Tarea que permite invocar la ejecución de otro bloque, a diferencia de Goto, al finalizar ese bloque se continuarán ejecutando las siguientes tareas a Call.

Atributos:

Atributo	Tipo	Descripción	Requerido
<code>block</code>	String	Bloque a ejecutar.	SI

6.5.8.9 SWITCHCASE



Tarea que permite realizar Goto a otros bloques y/o tareas en función de un valor de entrada.

Atributos:

Atributo	Tipo	Descripción	Requerido
value	String	Valor a evaluar.	SI

Elementos internos case [0..*]:

Son elementos que representan cada uno de los casos que se tratarán.

Atributos de los elementos internos case:

Atributo	Tipo	Descripción	Requerido
value	String	Valor con el que se comparará el valor de entrada de la tarea.	SI
block	String	Nombre del bloque a ejecutar. Si no se informa será el actual.	NO
task_label	String	Id de la tarea a ejecutar dentro del bloque elegido. Si no se informa se ejecutará la primera tarea del bloque elegido.	NO

En caso de existir varios nodos case que coincidan con el valor de entrada se aceptará el primer case que haya sido definido.

6.5.8.10 SCRIPT



Tarea que permite invocar la ejecución de código de script programable.

Atributos:

Atributo	Tipo	Descripción	Requerido
language	String	El lenguaje del bloque de script. Posibles valores son "VB", "C#".	SI

mainclass	String	El nombre de la clase que contiene el método estático ScriptMain, que es el que será ejecutado.	NO
Prefix	String	El prefijo namespace para cualquier función definida en el script. Por defecto será 'script'.	NO
references	String	Colección de elementos donde se deben indicar los ensamblados necesarios para la ejecución del script.	NO
Imports	String	Colección de elementos donde se deben indicar los namespaces necesarios para la ejecución del script.	NO
code	String	Es obligatorio y contiene el código del script que se quiere ejecutar.	SI

6.5.8.11 TIMECONDITIONS



Tarea que permite evaluar si el momento actual cumple condiciones temporales.

Atributos:

Atributo	Tipo	Descripción	Requerido
property	String	Propiedad que recibirá el resultado de evaluar las condiciones temporales, true o false.	NO

Elementos internos timeconditon [0..*]:

Son elementos que representan una serie de condiciones.

Atributos de los elementos internos timeconditon:

Atributo	Tipo	Descripción	Requerido
time_start	String	Tiempo inicial de la condición. Debe cumplir el formato ISO para horario hhmmss, hhmm, hh:mm:ss o hh:mm. Si no se informa no se tendrá en cuenta.	NO
time_finish	String	Tiempo final de la condición. Debe cumplir el formato ISO para horario hhmmss, hhmm, hh:mm:ss o hh:mm. Si no se informa no se tendrá en cuenta.	NO
week_day	Integer	Día de la semana de la condición. Del 1 al 7, siendo 1 Lunes y 7 Domingo. Si no se informa no se tendrá en cuenta.	NO
week_day_start	Integer	Día de la semana inicial de la condición. Del 1 al 7, siendo 1 Lunes y 7 Domingo. Si no se informa no se tendrá en cuenta.	NO
week_day_finish	Integer	Día de la semana final de la condición. Del 1 al 7, siendo 1 Lunes y 7 Domingo. Si no se informa no se tendrá en cuenta.	NO
month_day	Integer	Día del mes de la condición. Del 1 al 31. Si no se informa no se tendrá en cuenta.	NO
month_day_start	Integer	Día del mes inicial de la condición. Del 1 al 31. Si no se informa no se tendrá en cuenta.	NO
month_day_finish	Integer	Día del mes final de la condición. Del 1 al 31. Si no se informa no se tendrá en cuenta.	NO
month	Integer	Mes de la condición. Del 1 al 12, siendo 1 Enero y 12 Diciembre. Si no se informa no se tendrá en cuenta.	NO

month_start	Integer	Mes inicial de la condición. Del 1 al 12, siendo 1 Enero y 12 Diciembre. Si no se informa no se tendrá en cuenta.	NO
month_finish	Integer	Mes final de la condición. Del 1 al 12, siendo 1 Enero y 12 Diciembre. Si no se informa no se tendrá en cuenta.	NO

Para que “timeconditions” evalúe a true, uno de los elementos “timecondition” debe evaluar a true.
Para que un elemento “timecondition” evalúe a true el momento actual debe cumplir con todas las condiciones definidas por los parámetros declarados.

Adicionalmente a la propiedad almacenada en el parámetro property, el resultado también se guardara en la propiedad “timeconditions.result”, con valor true o false.

6.5.8.12 ANALYZETEXT



Tarea que permite evaluar si un texto cumple las condiciones indicadas.

Atributos:

Atributo	Tipo	Descripción	Requerido
property	String	Propiedad donde se recibe el resultado del análisis del texto, true o false.	NO
input	String	Texto a analizar.	SI
contains	String	Texto que ha de contener el input. Si no se informa no se tendrá en cuenta.	NO
startswith	String	Texto por el que ha de empezar el input. Si no se informa no se tendrá en cuenta.	NO
endswith	String	Texto por el que ha de terminar el input. Si no se informa no se tendrá en cuenta.	NO
equals	String	Texto que ha de ser idéntico al del input. Si no se informa no se tendrá en cuenta.	NO
regexpattern	String	Patrón Regex con el que se evaluará el input. Si no se informa no se tendrá en cuenta.	NO

<code>regexoptions</code>	String	Opciones del patrón Regex con el que se evaluará el input. Tipo: <code>System.Text.RegularExpressions.RegexOptions</code> Si no se informa no se tendrá en cuenta.	NO
---------------------------	--------	---	----

Para que “analyzetext” evalúe a true se debe cumplir con todas las condiciones de los atributos declarados (AND).

Cada atributo puede contener una lista de elementos separados por coma. Para que un atributo evalúe a true es necesario con que se cumpla con uno de los elementos (OR).

Adicionalmente a la propiedad almacenada en el parámetro property, el resultado también se guardara en la propiedad “analyzetext.result”, con valor true o false.

6.5.9 TAREAS PERSONALIZADAS



En escenarios especiales se puede extender el lenguaje DBRScript con tareas personalizadas. Gracias a esta característica se pueden cubrir escenarios específicos, por ejemplo: acceso a bases de datos, web services...

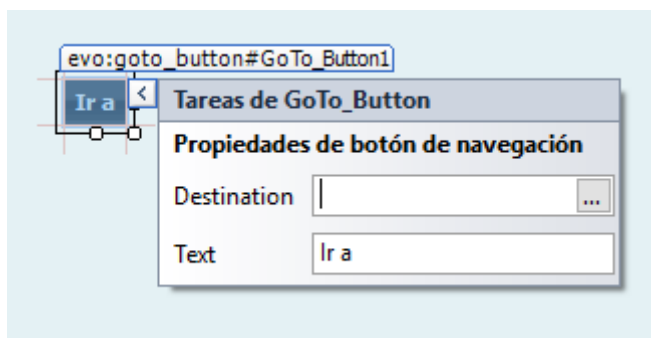
7 VISUAL EVOLUTION LIBRARY

La librería Visual Evolution Library contiene una serie de controles de usuario para ubicar en las páginas de Developer.NET. Simplemente hay que arrastrar y soltar el control en la ubicación correspondiente de la página y rellenar las propiedades adecuadamente para personalizar el control según nuestras necesidades.

En general los controles muestran un “Smart Tag” o diálogo donde se pueden configurar las propiedades más utilizadas del control. Adicionalmente, podemos acceder a todas las propiedades del control mediante la ventana de propiedades del mismo (pulsando Alt-Enter).

7.1 GOTO_BUTTON

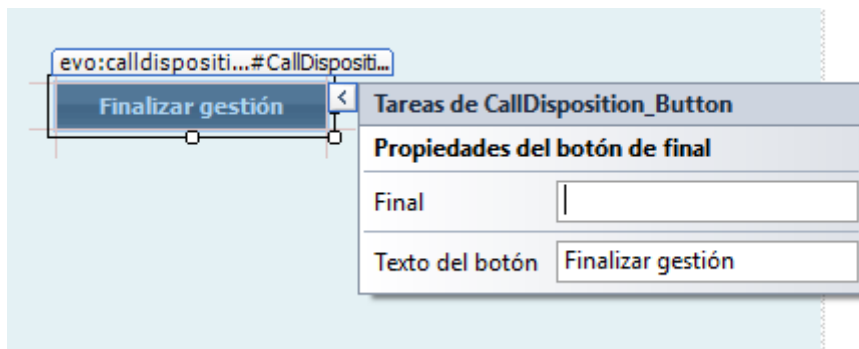
Se utiliza para que el agente pueda navegar a la página de argumentario especificada. También se puede navegar a una URL personalizada, por ejemplo, para integrar una aplicación externa en el argumentario de agente.



El texto del botón puede modificarse mediante la propiedad Text.

7.2 CALLDISPOSITION_BUTTON

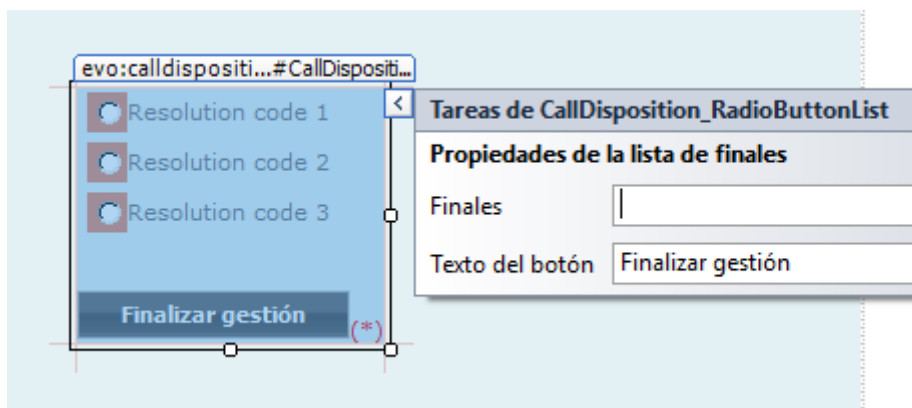
Se utiliza para finalizar la transacción de Evolution con el final especificado (CallDispositionId). Recuerde que toda transacción en Evolution es finalizada por el argumentario. Este control nos facilita la tarea al incluir toda la lógica necesaria para realizar el fin de transacción Evolution.



El texto del botón puede modificarse mediante la propiedad Text. Este control expone el evento PreClick que permite incorporar lógica de negocio que se ejecutará antes de finalizar la transacción.

7.3 CALLDISPOSITION_RADIOBUTTONLIST

Funcionalmente es similar al control anterior, solo que, en lugar de finalizar con un final determinado, presenta una lista (RadioButtons) de finales, que el agente podrá escoger para finalizar la transacción de Evolution.

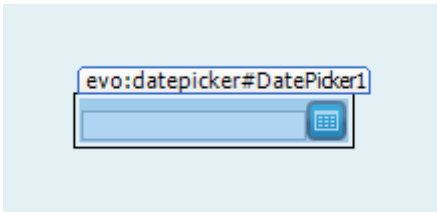


La propiedad Finales contiene una expresión con los finales que queremos presentar. Podemos especificar valores separados por comas y rangos. Por ejemplo: 1,2,3,100-105

El texto del botón puede modificarse mediante la propiedad Text. Este control expone el evento PreClick que permite incorporar lógica de negocio que se ejecutará antes de finalizar la transacción.

7.4 DATEPICKER

Se trata de un control que nos permite introducir fechas en el formato que especifiquemos. El control incluye un calendario que muestra el mes entero.



El formato de la fecha se especifica en la propiedad Formato de fecha. El significado del string de especificación del formato es el siguiente:

%a	abbreviated weekday name
%A	full weekday name
%b	abbreviated month name
%B	full month name
%C	century number
%d	the day of the month (00 .. 31)
%e	the day of the month (0 .. 31)
%H	hour (00 .. 23)
%I	hour (01 .. 12)
%j	day of the year (000 .. 366)
%k	hour (0 .. 23)
%l	hour (1 .. 12)
%m	month (01 .. 12)
%M	minute (00 .. 59)
%n	a newline character
%p	"PM" or "AM"
%P	"pm" or "am"
%S	second (00 .. 59)
%s	number of seconds since Epoch (since Jan 01 1970 00:00:00 UTC)
%t	a tab character
%U, %W, %V	the week number
%u	the day of the week (1 .. 7, 1 = MON)
%w	the day of the week (0 .. 6, 0 = SUN)
%y	year without the century (00 .. 99)
%Y	year including the century (ex. 1979)
%%	a literal % character

7.5 CONTROLES DE DATOS DE CLIENTE

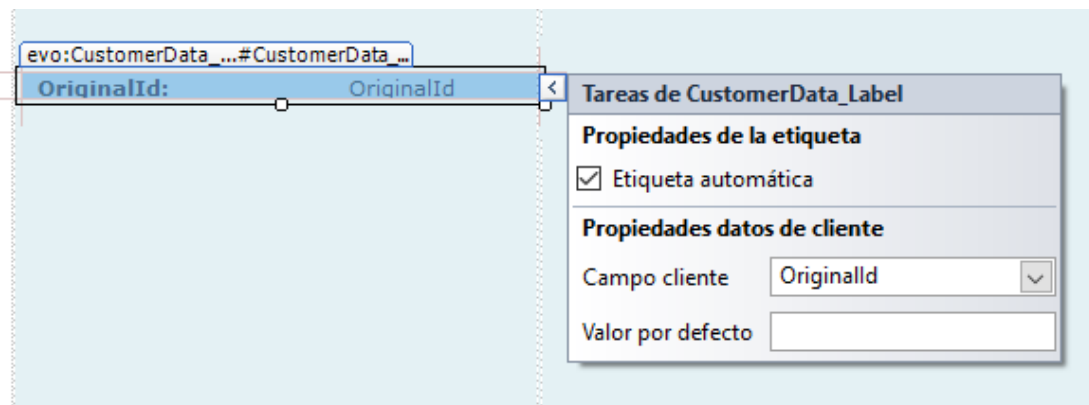
Presentamos ahora un grupo de controles que nos permiten visualizar y/o modificar datos de clientes de Evolution. Pueden ser **datos de clientes** propiamente dichos, es decir, aquellos que se corresponden con la tabla de CLIENTES del modelo de datos de Evolution o un **dato adicional de cliente**. Los datos de cliente son campos fijos, con un nombre y tipo de dato específicos. En cambio, en el caso de los datos adicionales de cliente, el nombre del dato y la información que contiene son totalmente personalizables. Típicamente, contendrán información específica del negocio.

Recuerde que los datos adicionales de cliente se pueden cargar en la campaña con el resto de datos de clientes o bien pueden ser campos que se cumplimentarán cuando se presente el argumentario en la pantalla del agente (o ambas cosas a la vez).

Por ejemplo, en la plantilla de proyecto de encuestas, se hace uso de los datos adicionales de cliente para almacenar las respuestas a la encuesta.

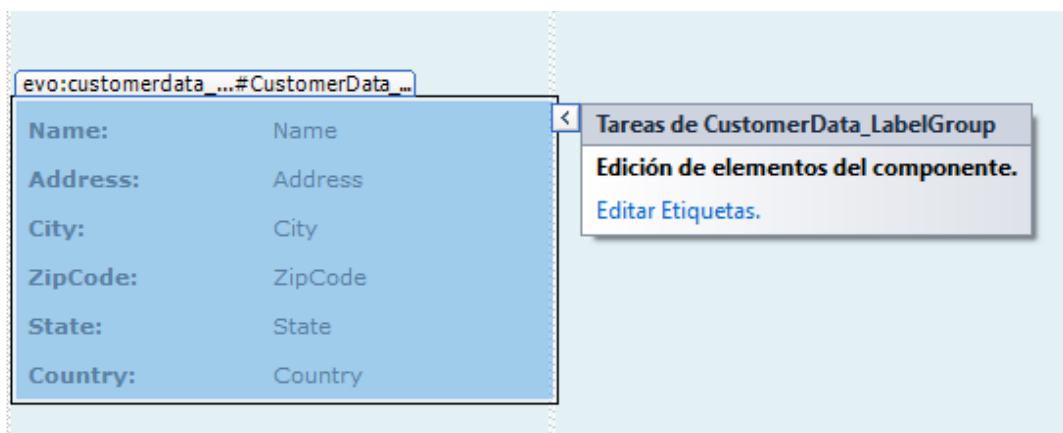
7.5.1 CUSTOMERDATA_LABEL

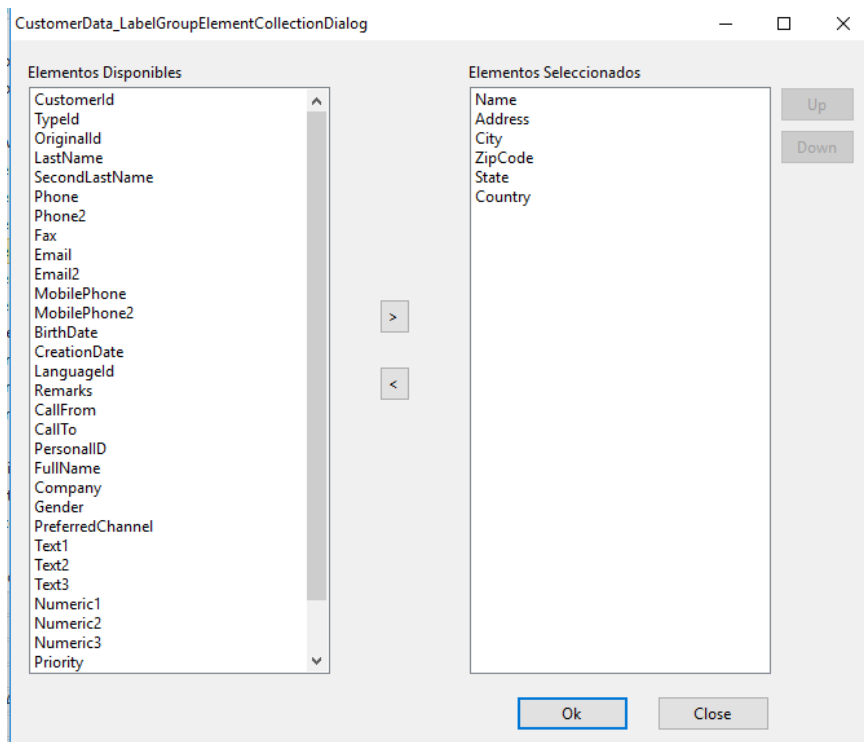
Este control permite mostrar un dato de cliente o dato adicional de cliente. El control incluye una etiqueta (nombre del campo) automática o configurable y el campo propiamente dicho. Disponemos de un desplegable para seleccionar el campo de cliente o, si se trata de un dato adicional, podemos especificar el nombre de campo que deseemos (por defecto, Clave). Asimismo, podemos especificar un valor por defecto. Es el valor que se mostrará cuando el valor del campo en base de datos no exista o sea NULL.



7.5.2 CUSTOMERDATA_LABELGROUP

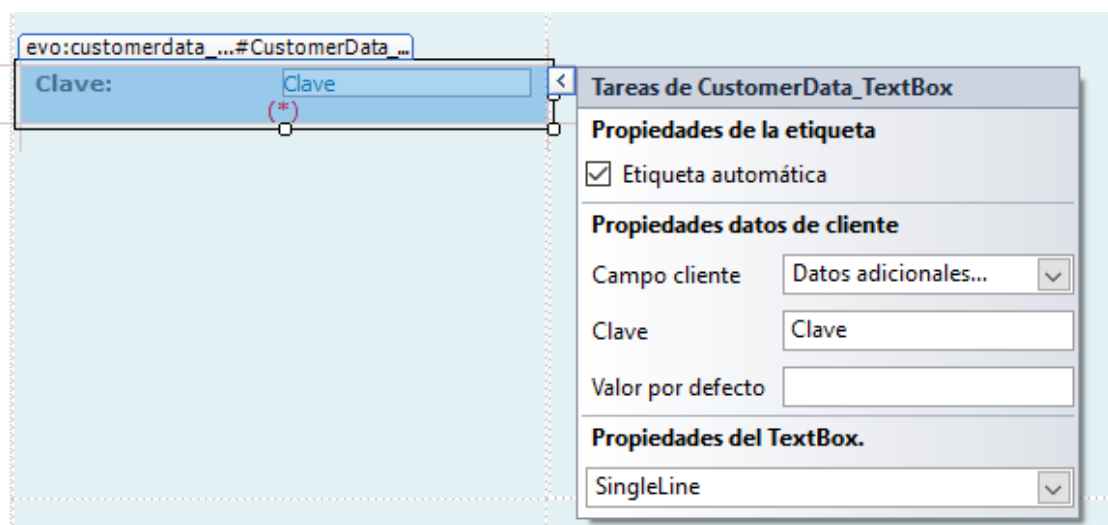
Este control permite mostrar uno o varios datos de cliente de forma simultánea. Podemos seleccionar entre los diferentes campos que conforman la tabla de CLIENTE, qué campos deseamos mostrar.





7.5.3 CUSTOMERDATA_TEXTBOX

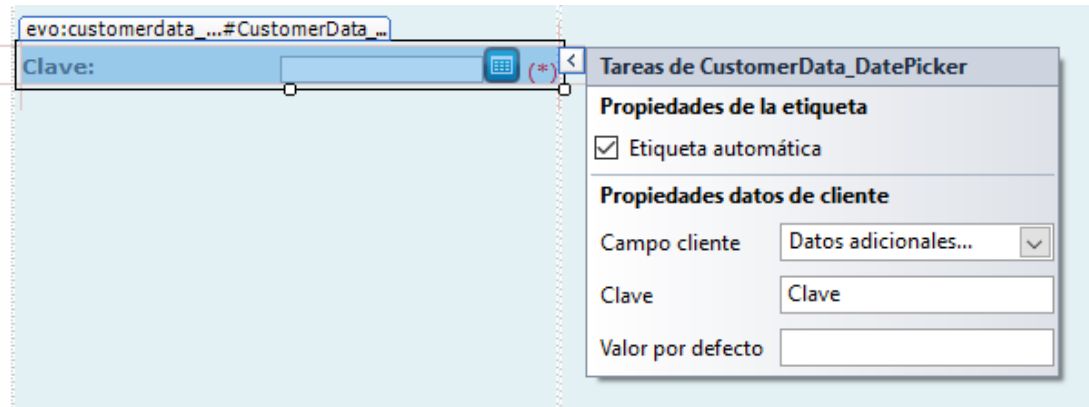
Este control permite editar un dato de cliente o dato adicional de cliente. El control incluye una etiqueta (nombre del campo) automática o configurable y el campo propiamente dicho. Disponemos de un desplegable para seleccionar el campo de cliente o, si se trata de un dato adicional, podemos especificar el nombre de campo que deseemos (por defecto, Clave). Asimismo, podemos especificar un valor por defecto. Es el valor que se mostrará cuando el valor del campo en base de datos no exista o sea NULL.



El campo de edición propiamente dicho puede ser de una sola línea, de múltiples líneas o de tipo password.

7.5.4 CUSTOMERDATA_DATAPICKER

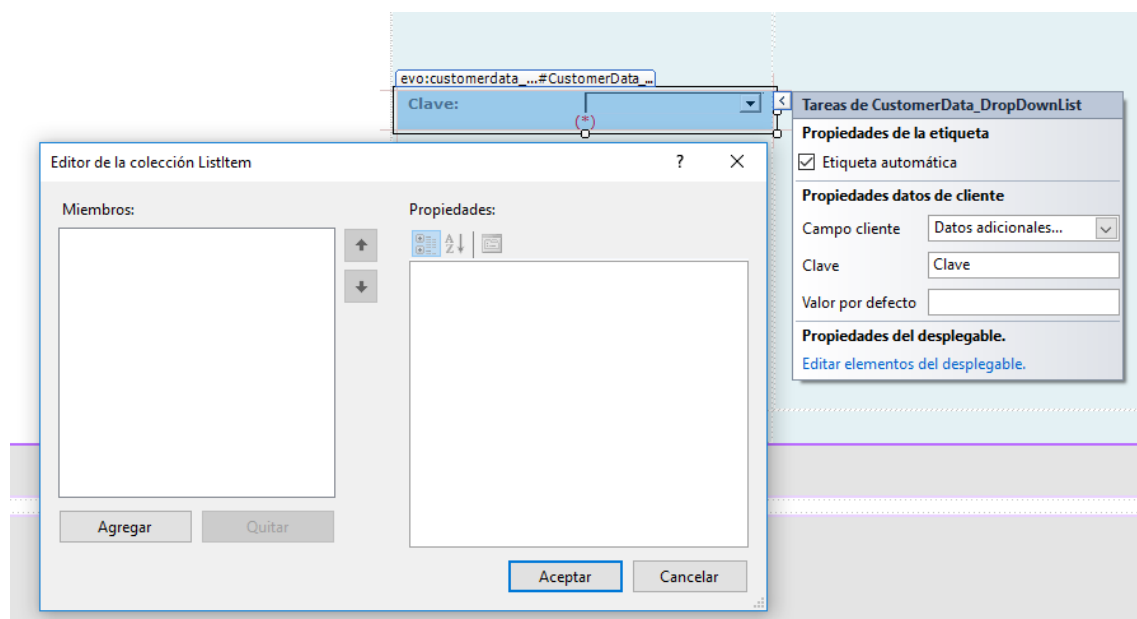
Este control permite editar un dato de cliente o dato adicional de cliente de tipo fecha.



El formato de fecha sigue el mismo formato que el SelectorFecha comentado anteriormente.

7.5.5 CUSTOMERDATA_DROPDOWNLIST

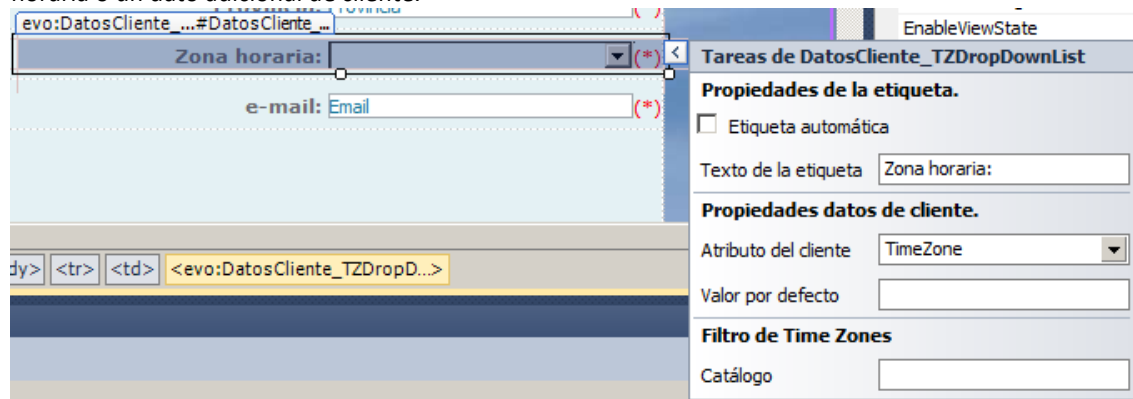
Es un control destinado a ofrecer un desplegable con diferentes valores para cumplimentar el dato de cliente o el dato adicional de cliente.



Típicamente este control lo utilizaremos para que el agente elija un valor entre una lista de valores posibles. La lista de valores a escoger la podemos configurar editando la lista de elementos del desplegable.

7.5.6 CUSTOMERDATA_TZDROPDOWNLIST

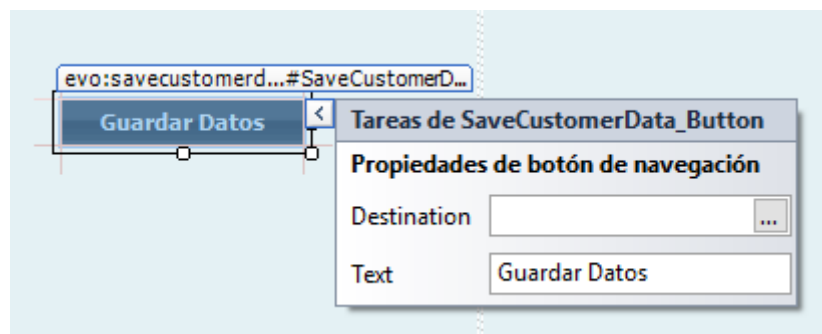
Es un control destinado a ofrecer un desplegable con diferentes TimeZones para cumplimentar la zona horaria o un dato adicional de cliente.



Adicionalmente se podría definir el parámetro CatalogoTimeZones para elegir la lista de zonas horarias a mostrar correspondientes a la tabla TZ_ZoneName.

7.5.7 SAVECUSTOMERDATA_BUTTON

Este control permite persistir en base de datos los valores que se han editado utilizando los controles anteriormente especificados.



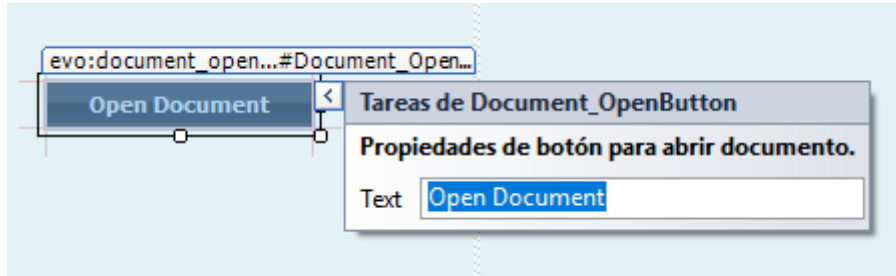
Destino es la página a la cual se navegará tras haber guardado los datos. El control permite especificar el evento Preclick para incluir lógica de negocio previa al guardado de datos.

7.6 CONTROLES DE TIPO DOCUMENTO

Los siguientes controles permiten manipular documentos genéricos.

7.6.1 DOCUMENT_OPENBUTTON

Este control permite abrir o descargar documentos asociados al contacto inicial.



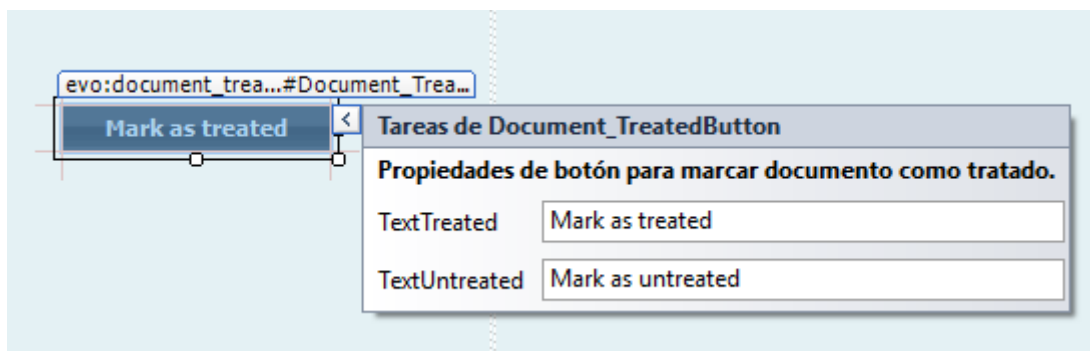
El control permite especificar el evento Preclick para incluir lógica de negocio previa a abrir o descargar el fichero.

Si no hay documento asociado el botón se mostrará deshabilitado.

7.6.2 DOCUMENT_TREATEDBUTTON

Este control permite marcar como tratado el contacto inicial y el documento asociado cuando no está tratado. Y permite marcar como no tratado el contacto inicial y el documento asociado cuando está tratado.

Cuando hay contacto inicial, al terminar la gestión si se ha marcado como tratado se finalizará el registro normalmente, pero si no ha sido tratado volverá a la cola para que sea tratado por otro agente.

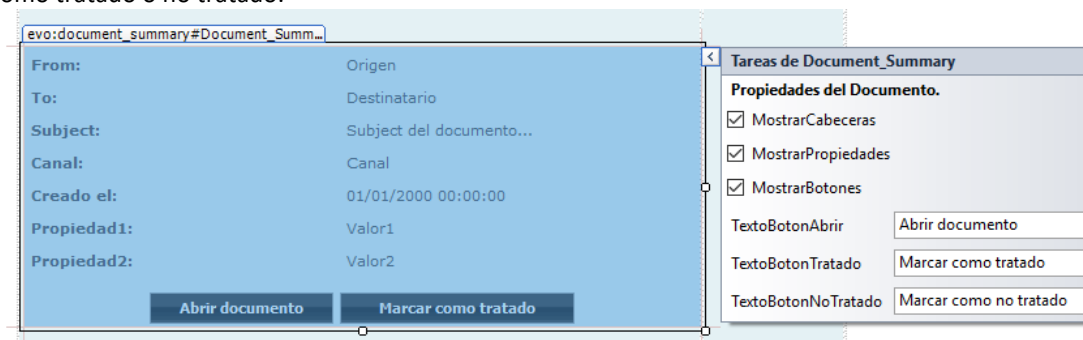


El control permite especificar el evento Preclick para incluir lógica de negocio previa a marcar como tratado.

Si no hay contacto inicial el botón se mostrará deshabilitado.

7.6.3 DOCUMENT_SUMMARY

Este control muestra información del documento, permite abrir o descargar el documento y marcarlo como tratado o no tratado.



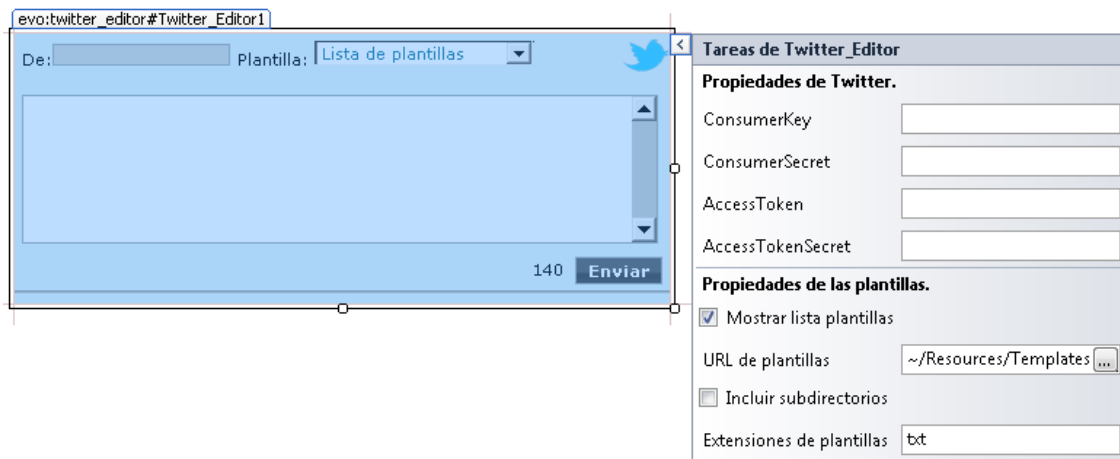
7.7 CONTROLES DE TWITTER

Presentamos ahora un grupo de controles que nos permiten interactuar con Twitter. Estos controles se comunican con Twitter a través de las API's publicadas por Twitter.

7.7.1 TWITTER_EDITOR

Este control, que permite editar y publicar una actualización de estado en Twitter, se utiliza para que el agente pueda emitir tweets desde el argumentario.

Permite incorporar un listado de plantillas de respuestas contenidas en una carpeta del argumentario que permitirán al agente añadir respuestas predefinidas con rapidez y facilidad.



El control ha de ser informado del usuario a través del cual va a publicar tweets a través de las credenciales que identifican una aplicación Twitter y su usuario vinculado.

Este control también permite habilitar/deshabilitar el desplegable de plantillas de respuestas predefinidas. Este desplegable muestra todas las plantillas contenidas en una carpeta determinada que cumplan con la extensión o conjunto de extensiones indicada para que, al elegir una opción el agente, automáticamente se añada la respuesta.

Nota: Este control está regido por las limitaciones propias de la API Twitter pública y, por tanto, se ajusta a las restricciones impuestas por Twitter en cuanto a su política de límites de uso diario/horario (<http://support.twitter.com/articles/344781-sobre-los-limites-en-twitter-actualizaciones-dms-api-seguidores>).

7.7.2 TWITTER

Este control permite mostrar, visualizar y responder vía Twitter a documentos de tipo tweet encolados mediante Evolution DBR.

Este control muestra información del documento de tipo Twitter, permite mostrar las cabeceras y propiedades del documento y marcarlo como tratado o no tratado.

The screenshot displays the Twitter control interface. The main area shows a tweet from @evolutionICR with a placeholder text. Below the tweet is a 'De:' field and a 'Plantilla:' dropdown menu. To the right is a sidebar titled 'Tareas de Twitter' containing sections for document properties, Twitter authentication, and templates.

Tareas de Twitter

Propiedades del Documento.

- ☐ Mostrar cabeceras del documento
- ☐ Mostrar propiedades del documento
- ☐ Mostrar Botón de Marcar como tratado
- ☒ Permitir respuestas

Propiedades de autenticación Twitter.

ConsumerKey:

ConsumerSecret:

AccessToken:

AccessTokenSecret:

Propiedades de Plantillas.

- ☒ Mostrar lista de plantillas

URL de plantillas: ...

☐ Incluir subdirectorios

Extensiones de plantillas:

Si se activa la opción PermitirRespuestas, este control permite responder, retwittear o enviar un mensaje directo a través de Twitter al usuario emisor del tweet.

El control ha de ser informado del usuario a través del cual va a publicar tweets a través de las credenciales que identifican una aplicación Twitter y su usuario vinculado.

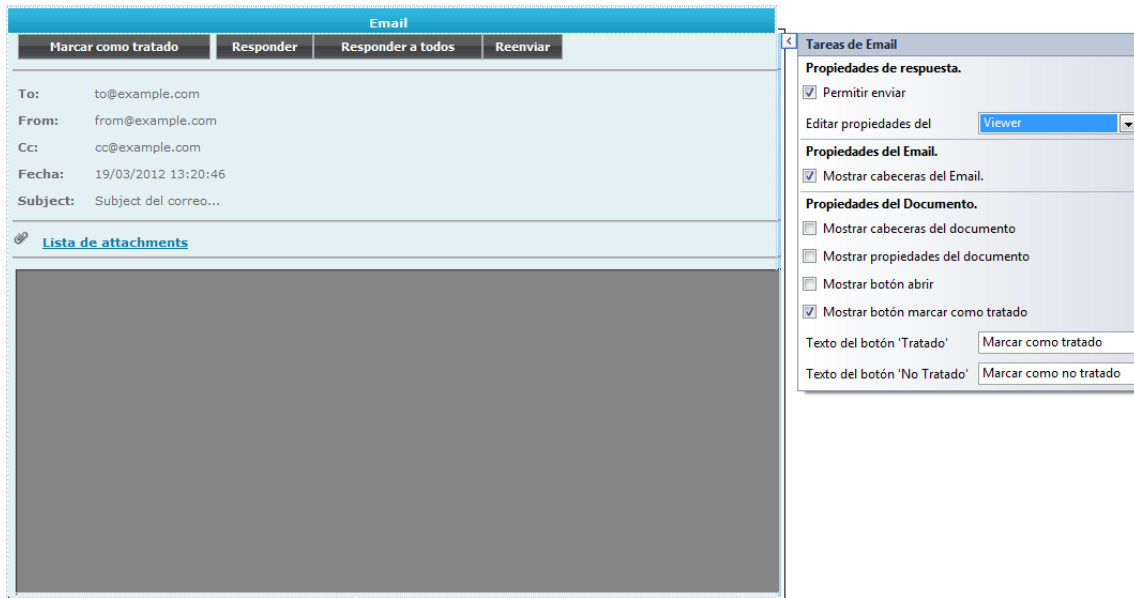
Este control también permite habilitar/deshabilitar el desplegable de plantillas de respuestas predefinidas. Este desplegable muestra todas las plantillas contenidas en una carpeta determinada que cumplan con la extensión o conjunto de extensiones indicada para que, al elegir una opción el agente, automáticamente se añada la respuesta.

7.8 CONTROLES DE EMAIL

El siguiente grupo de controles nos permiten ver y enviar emails.

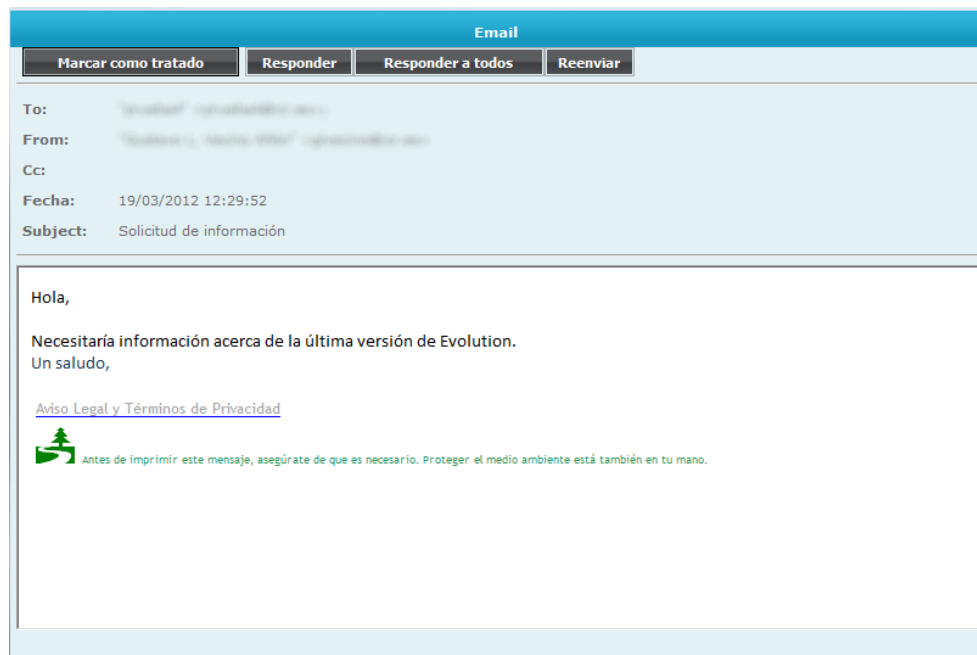
7.8.1 EMAIL

Este control permite al agente visualizar, editar y enviar emails desde el argumentario.



Visor de emails en modo diseño

El programador puede configurar el control para que en tiempo de ejecución se muestren las cabeceras y propiedades del documento, así como marcarlo como tratado o no tratado, e incluso se puede abrir el email con una herramienta externa.



Visor de emails en iAgent

Si se habilitan la opción de responder, el agente podrá contestar el email.

Editor en modo diseño

Las propiedades agrupadas en la categoría Evolution permiten configurar el control. Aunque los Smart Tags dan acceso a las principales propiedades.

Si se desea se puede incorporar un listado de plantillas contenidas en una carpeta del argumentario para que el agente incorpore con facilidad textos predefinidos en el contenido del email. Al seleccionar una plantilla el nombre del fichero que contiene dicha plantilla se usará como asunto.

Se puede especificar un archivo con la firma, que se incorporará automáticamente a la respuesta.

También se pueden añadir a la respuesta archivos adjuntos, ya sea de una lista de archivos que se encuentran en una carpeta del argumentario, u otros que considere el agente. Se puede deshabilitar de manera independiente cualquiera de los dos tipos de adjuntos.

Hay una serie de campos como son el remitente, la copia, la copia oculta o el asunto que es posible prefijar y además se puede permitir que el agente los modifique o no.

Para que se pueda enviar el correo es imprescindible que se haya informado del Servidor SMTP correctamente (y si fuese necesario el usuario y contraseña). Además, los campos De (From), Para (To) y Asunto (Subject) no pueden estar vacíos.

El mail que se generará será de tipo HTML y el agente dispondrá de un editor WYSIWYG para escribir la respuesta.

Editor en iAgent

El programador tendrá acceso a las distintas propiedades de un objeto de tipo Email_Editor. Por ejemplo:

- Templates: da acceso a las plantillas.
- Attachments: a los adjuntos.
- Headers: a las cabeceras De, Para, Asunto ...
- SmtptServer: a la configuración del servidor de correo SMTP.
- Send(): enviar el mail.
- LastErrorMessage: último error generado.

Puede aprovecharse del IntelliSense del control para más información.

7.8.2 EMAIL_EDITOR

Este control permite al agente editar y enviar emails desde el argumentario. Tiene las mismas características que el Editor del control Email.

7.9 CONTROLES DE WEBCHAT

Presentamos ahora un grupo de controles que nos facilitan interactuar con Evolution WebChat permitiendo participar en chats o visualizar sus conversaciones.

Los controles publican unas propiedades, agrupadas en la categoría Evolution, que permiten su configuración. No obstante, mediante los Smart Tags se tendrá acceso a las principales propiedades.

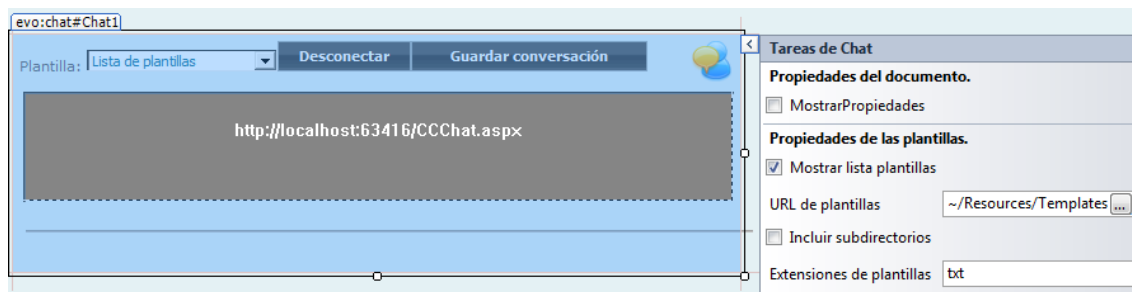
El programador tendrá acceso a todas las propiedades de los controles de tipo Chat para hacer uso de ellas en tiempo de ejecución. Puede aprovecharse del IntelliSense del control para más información.

7.9.1 CHAT

Este control permite a un agente participar, desde un argumentario, en una sesión de chat iniciada por Evolution WebChat

Permite incorporar un listado de plantillas de respuestas contenidas en una carpeta del argumentario que permitirán al agente añadir respuestas predefinidas con rapidez y facilidad. Este desplegable muestra todas las plantillas contenidas en una carpeta determinada que cumplan con la extensión o conjunto de extensiones indicada para que, al elegir una opción el agente, automáticamente se cargue la respuesta.

El control también permite mostrar/ocultar las propiedades del documento de tipo Chat almacenado en Evolution.



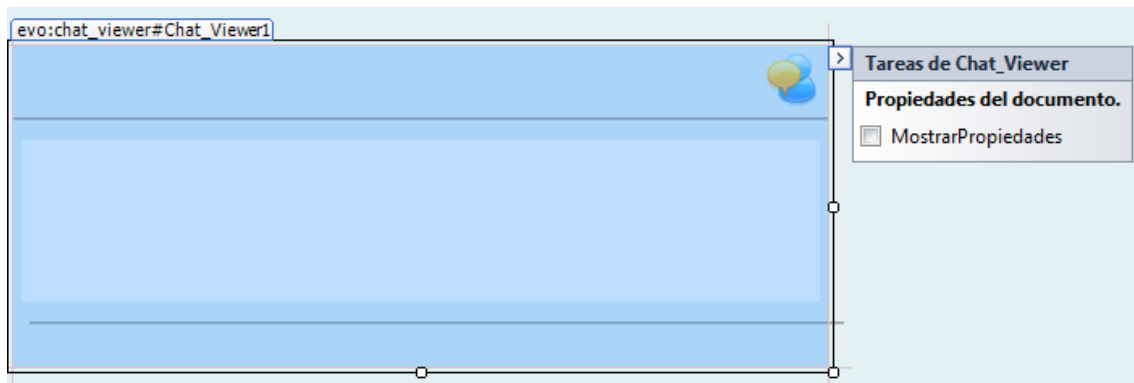
Este control dispone de un botón Desconectar que permite al agente desconectarse del chat notificándose al cliente. Mediante el botón Guardar conversación, también se permite al agente guardar, a demanda, la conversación mantenida con el cliente.

7.9.2 CHAT_VIEWER

Este control permite mostrar las conversaciones mantenidas en los chats registrados a través de Evolution WebChat.

El control también permite mostrar/ocultar las propiedades del documento de tipo Chat almacenado en Evolution.

Este control publica la propiedad `IdDocument` que permite al programador informar en tiempo de ejecución el documento Chat almacenado en Evolution del cual queremos obtener la conversación.

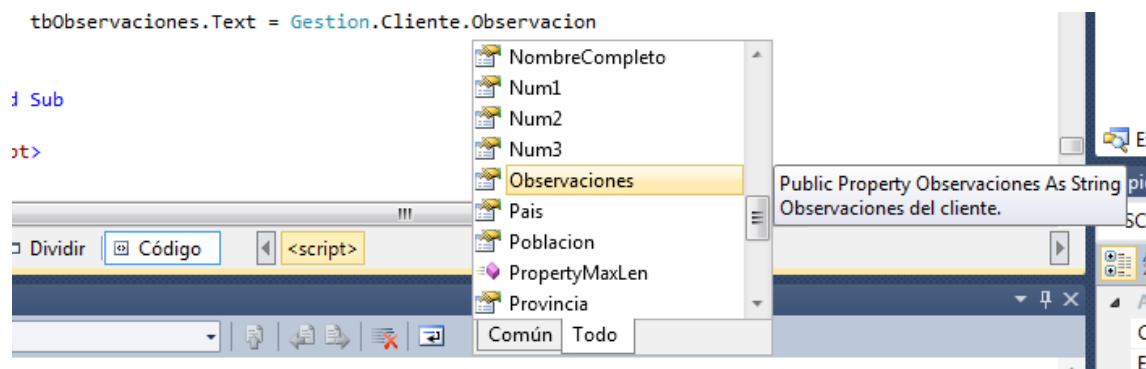


8 EVOLUTIONLIBRARY

La Evolution Library es una librería que contiene una serie de objetos, métodos y propiedades que pueden ser útiles en los argumentarios que podemos construir con Developer.NET.

La librería se encuentra en el espacio de nombres ICR.Evolution.EvolutionLibrary. La forma más habitual de acceder a estos métodos será mediante el objeto API. Este objeto, de tipo ICR.Evolution.EvolutionLibrary.API, ya viene inicializado y puede utilizarse directamente.

Durante la edición de código y gracias a la característica Intellisense de Developer.NET, puede acceder a los métodos y propiedades disponibles junto con una descripción de los mismos.



A continuación, se listan las propiedades y métodos más interesantes del objeto API.

8.1 EVOLUTION.EVOLUTIONLIBRARY.EVOAPI

Información de la gestión en curso.

8.1.1 PROPIEDADES

Propiedad	Observaciones
IsLoaded	Indica si la sesión se encuentra cargada. Valida que los objetos 'Customer', 'Campaign' y 'Transaction' se encuentren cargados.
Customer	Datos del cliente de la gestión actual
Agent	Datos del agente de la gestión actual
Campaign	Datos de la campaña actual

Transaction	Datos de la transacción actual (ANI, DNIS, IdTransaccion, LocalizadorActual, SentidoContacto, Transferencia). Cuando una transacción corresponde a una llamada entrante de campaña o está asociada a un documento podemos consultar la información del contacto en EvoAPI.Transaction.OriginatingContact. Si se trata de un documento, podremos acceder a la información del mismo en EvoAPI.Transaction.OriginatingContact.Document.
--------------------	--

8.1.2 MÉTODOS

Método	Observaciones
LoadCustomer	Carga los datos del cliente indicado. Parámetros: customerId: Identificador del cliente.
ApplyCallDisposition	Finaliza la gestión redirigiendo a una página específica. Parámetros: callDispositionId: Identificador del final. scheduledDateTime: Día y hora del próximo contacto. iInterval: Intervalo en minutos para establecer el próximo contacto a partir de scheduledDateTime. iQuota: Incremento en el valor de cuota del segmento y la campaña.

8.1.3 EJEMPLOS DE USO

Mostrar el nombre de la campaña y del cliente actual:

```
<b>Campaña: <%= EvoAPI.Campaign.Name %> - Cliente: <%= EvoAPI.Customer.Name  
& " " & EvoAPI.Customer.SecondName & " " & EvoAPI.Customer.LastName%></b>
```

Uso de ApplyCallDisposition para finalizar una gestión con el final correspondiente:

```
<%@ Page Title="" Language="VB" AutoEventWireup="true" %>  
<%  
    Dim idFinal As Integer = Convert.ToInt32(Request("idFinal"))  
    Dim nCuota As Integer? = Nothing  
    If Not String.IsNullOrEmpty(Request("nCuota")) Then  
        nCuota = Convert.ToInt32(Request("nCuota"))  
    End If  
    Dim nIntervalo As Integer? = Nothing  
    If Not String.IsNullOrEmpty(Request("nIntervalo")) Then  
        nIntervalo = Convert.ToInt32(Request("nIntervalo"))  
    End If  
    Dim tHoraProxContacto As Date? = Nothing  
    If Not String.IsNullOrEmpty(Request("tHoraProxContacto")) Then  
        tHoraProxContacto = Request("tHoraProxContacto").ToDateTimeRellamada  
    End If  
  
    'Llamamos a ApplyCallDisposition pasando los parámetros que hemos ido  
    arrastrando  
    EvoAPI.ApplyCallDisposition(idFinal, tHoraProxContacto, nIntervalo, nCuota)  
>%>
```

9 PLANTILLAS DE DOCUMENTOS

Algunos controles permiten incorporar plantillas, dichas plantillas se deben almacenar en archivos de texto (txt, html, ...),

Existen una serie de etiquetas que son reemplazadas por su valor correspondiente en la gestión en curso y así generar el texto de la plantilla de manera dinámica.

9.1.1 DATOS DE LA TABLA CLIENTES

#CLIENTE_ID_SUJETO#, #CLIENTE_ID_ORIGINAL#, #CLIENTE_NOMBRE#, #CLIENTE_APELLIDO1#,
#CLIENTE_APELLIDO2#, #CLIENTE_DIRECCION#, #CLIENTE_POBLACION#, #CLIENTE_CODIGO_POSTAL#,
#CLIENTE_PROVINCIA#, #CLIENTE_PAIS#, #CLIENTE_TELEFONO#, #CLIENTE_TELEFONO2#,
#CLIENTE_FAX#, #CLIENTE_EMAIL#, #CLIENTE_EMAIL2#, #CLIENTE_MOVIL#, #CLIENTE_MOVIL2#,
#CLIENTE_FECHA_NACIMIENTO#, #CLIENTE_OBSERVACIONES#, #CLIENTE_DNI#,
#CLIENTE_NOMBRE_COMPLETO#, #CLIENTE_EMPRESA#, #CLIENTE_SEXO#, #CLIENTE_TEXTO1#,
#CLIENTE_TEXTO2#, #CLIENTE_TEXTO3#, #CLIENTE_NUM1#, #CLIENTE_NUM2#, #CLIENTE_NUM3#

9.1.2 DATOS DE LA TABLA TB DATOS CLIENTE

Esta es una tabla que almacena para cada cliente una especie de diccionario de datos, con sus claves y valores asociados. En este caso la etiqueta permite especificar una clave y será sustituida por el valor. Es importante que la clave se escriba tal y como está en la base de datos, respetando mayúsculas y minúsculas.

#CLIENTE_DATO_(clave)#

Ej: #CLIENTE_DATO_cuenta de twitter#

9.1.3 DATOS DE LA TABLA CAMPANYA

#CAMPANYA_NOMBRE#, #CAMPANYA_DESCRIPCION#

9.1.4 DATOS DEL AGENTE

#AGENTE_NOMBRE#, #AGENTE_APELLIDO1#, #AGENTE_APELLIDO2#

9.1.5 DATOS DE LA TRANSACCIÓN

#TRANSACCION_ANI#, #TRANSACCION_DNIS#, #TRANSACCION_LOCALIZADOR_ACTUAL#

9.1.6 DATOS DE LA TABLA TB CONTACTOS

#CONTACTO_ORIGEN#, #CONTACTO_DESTINO#

9.1.7 DATOS DE LA TABLA TB CONTACTOS DOC

#DOCUMENT_SUBJECT#

9.1.8 EJEMPLO DE PLANTILLA

El siguiente ejemplo ilustra el contenido de un archivo que podría usarse para generar una plantilla de respuesta de Email:

Estimado/a #CLIENTE_NOMBRE# #CLIENTE_APELLIDO1#:

En respuesta a su solicitud de información acerca de nuestro nuevo servicio
"#CAMPANYA_DESCRIPCION#" le podemos decir que ...

...

Atentamente #AGENTE_NOMBRE# #AGENTE_APELLIDO1#

Esta plantilla generaría un texto como el que sigue:

Estimado/a Juan López:

En respuesta a su solicitud de información acerca de nuestro nuevo servicio "Atención en Carretera
Plus" le podemos decir que ...

...

Atentamente Ana Martínez