



TUTORIAL APLICACIONES EVOLUTION CON ASP.NET

1	Introducción.....	2
2	Requisitos previos.....	2
3	Habilitar el argumentario.....	2
4	Ejemplo de argumentario desarrollado en ASP.NET	4
4.1	El código fuente	4
4.2	Descripción del argumentario.....	5
4.2.1	Información recibida por el argumentario.....	5
4.2.2	Información obtenida de la base de datos	5
4.2.3	Información enviada a Evolution	6

1 INTRODUCCIÓN

Este documento ilustra la facilidad que ofrece Evolution para integrar argumentarios desarrollados en distintos entornos, en este caso en ASP.NET. Para ello hemos desarrollado un ejemplo simple facilitando el código de la aplicación.

En el manual de referencia existe más información acerca de las facilidades que ofrece Evolution para el desarrollo de argumentarios.

2 REQUISITOS PREVIOS

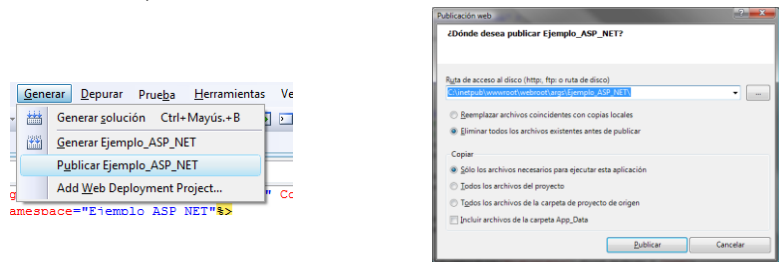
Es necesario tener correctamente instalado Evolution e iAgent, para probar el argumentario.

MS Visual Studio 2008 © para revisar el código.

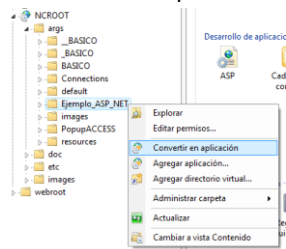
3 HABILITAR EL ARGUMENTARIO

Una vez obtenido el código, para probarlo debemos:

- 1- Publicarlo en el IIS desde el Visual Studio (por ejemplo en `C:\inetpub\wwwroot\webroot\args\Ejemplo_ASP_NET`, que es donde se despliegan los argumentarios de ejemplo de Evolution, pero cualquier otra ubicación es válida).



- 2- Con el administrador del IIS convertirlo en aplicación.



Ejemplo con IIS 7

- 3- Necesitaremos que exista un origen de datos ODBC: EVOLUTIONDB que apunte a la base de datos de Evolution (ya existirá si estamos en la misma máquina donde se encuentra instalado Evolution).

Si ya hemos desplegado en el directorio recomendado en *http://(IP del equipo)/NCROOT/args/Ejemplo_ASP_NET* podemos ir a esa página en el navegador web y veremos la información del cliente 0:



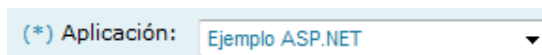
Una vez hemos comprobado que la aplicación se ve en el navegador, deberemos darla de alta como argumentario. Para ello en *Manager > Administración > Argumentarios > Nuevo* damos de alta el nuevo argumentario:

Nombre: Ejemplo ASP.NET

URL: *http://(IP del equipo)/NCROOT/args/Ejemplo_ASP_NET/Default.aspx* (en nuestro ejemplo)

Es necesario introducir la URL completa (*http...Default.aspx*) porque así se pasarán una serie de parámetros en dicha URL, más adelante veremos como se utilizarán dichos parámetros.

Ahora lo asociaremos a la campaña Telemarketing (es una de las dos que Evolution instala como ejemplo). *Manager > Administración > Campañas > Aplicación = Ejemplo ASP.NET*

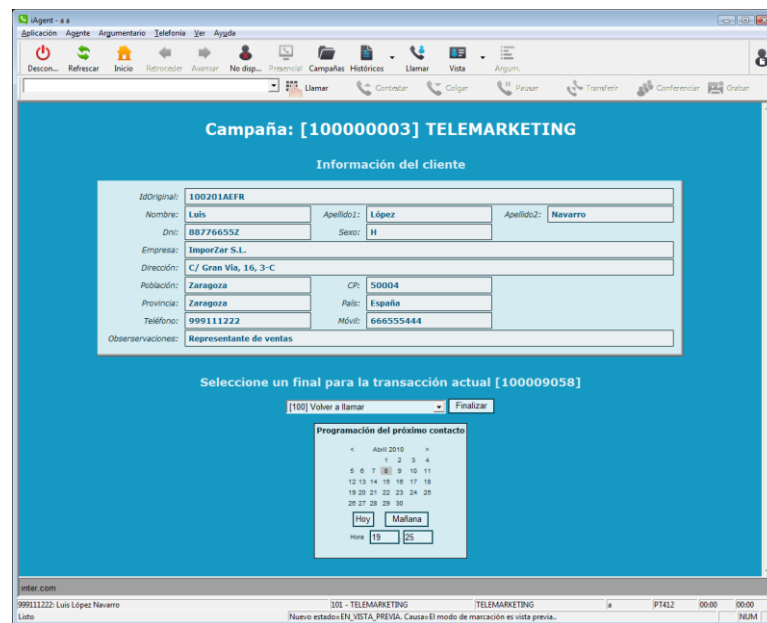


Si se ha hecho todo correctamente, al iniciar una aplicación de agente que se conecte a la campaña Telemarketing debería de utilizar este argumentario.

4 EJEMPLO DE ARGUMENTARIO DESARROLLADO EN ASP.NET

El ejemplo desarrollado es un argumentario simple para una campaña saliente, que identificará al cliente correspondiente a la transacción actual, mostrando parte de la información almacenada en la base de datos referente a dicho cliente.

También veremos un listado de los finales correspondientes a la campaña, y podremos finalizar la gestión del cliente. Recordemos que cada transacción realizada por los agentes debe ser finalizada, con un identificador (obtenido de la lista de finales) y en caso de que sea un final de tipo replanificado por agente, también se requiere una fecha para volver a contactar con el cliente.



4.1 EL CÓDIGO FUENTE

Una vez descomprimido el archivo con el código fuente, en la carpeta del proyecto nos quedarán los siguientes archivos significativos:

- *Default.aspx*: página web del argumentario.
- *Default.aspx.cs*: código de dicha página.
- *Sql.cs*: código de acceso a base de datos.

La página web principal de la aplicación (*Default.aspx*) se ha desarrollado de manera simple, pensando en desarrollos MVC (que no se ha incluido para no dificultar el acceso a quien no lo utilice), pero si el desarrollador deseara usar WebControls sería perfectamente lícito.

4.2 DESCRIPCIÓN DEL ARGUMENTARIO

El argumentario nos informará de la campaña en la que estamos (ID y Nombre)

Informará también de los datos del cliente al que se va a llamar (ID, Nombre...)

Ofrece un listado de los finales disponibles para esta campaña y podremos finalizar la transacción actual.

4.2.1 INFORMACIÓN RECIBIDA POR EL ARGUMENTARIO

En la URL se recibe diversa información relacionada con la gestión que está en curso, en nuestro caso accederemos a los identificadores de Campaña, Sujeto y Transacción (idCampanya, idSujeto, idTrans). Cargamos estos valores en *Default.aspx.cs*.

```
Request.QueryString["idSujeto"]
```

```
Request.QueryString["idCampanya"]
```

```
Request.QueryString["idTrans"]
```

El manual de referencia contiene más información acerca de los parámetros recibidos en la URL.

4.2.2 INFORMACIÓN OBTENIDA DE LA BASE DE DATOS

Con la información que hemos recibido podemos acudir a la base de datos de EvolutionDb y obtener los datos que necesitamos, en *Sql.cs*. Para realizar el acceso a los datos se ha definido una constante que representa la cadena de conexión, normalmente no será necesario modificarla, siempre que exista un origen de datos ODBC: EVOLUTIONDB que apunte a la base de datos de Evolution.

```
ConnectionString = "DSN=EVOLUTIONDB;UID=NCADMIN;PWD=ADMN1cr@"
```

- Nombre de la campaña: *Sql.ObtenerNombreCampanya*, nos devolverá un *string* con el nombre.

```
ExecuteSql("SELECT Nombre FROM CAMPANYA WHERE idCampanya=" + idCampanya);
```

- Datos del cliente: *Sql.ObtenerDatosCliente*, nos devolverá un *DataTable* con una fila que contiene los datos del cliente (ver en el modelo de datos los campos de la tabla *Clientes*).

```
ExecuteSql("SELECT * FROM CLIENTES WHERE idSujeto=" + idSujeto)
```

- Datos de finales para la campaña: *Sql.ObtenerListadoFinales*, nos devolverá un *DataTable* con los registros correspondientes a los finales de la campaña en la que estamos (ver en el modelo de datos los campos de la tabla *Finales*).

```
ExecuteSql("SELECT * FROM FINALES WHERE idCampanya=" + idCampanya + " ORDER BY idFinal")
```

Esta información se obtiene y se va utilizando en la propia página *Default.aspx*.

Recordemos que este es un ejemplo, se pueden obtener los datos como se desee (Linq a SQL, EntityFramework...) y de donde se desee, no tiene por qué usarse únicamente la base de datos de Evolution, además si se necesita incluir datos o tablas propias sería recomendable no alterar el modelo de datos de Evolution, para evitar conflictos con futuras actualizaciones del mismo.

4.2.3 INFORMACIÓN ENVIADA A EVOLUTION

Para informar a Evolution de que hemos decidido finalizar la transacción con un final determinado (seleccionar final en el desplegable y pulsar finalizar) utilizaremos la interfaz COM que ofrece iAgent.

En JavaScript (*Default.aspx* > *FinalGestion*, ya que este código debe ejecutarse en el cliente) crearemos un objeto ActiveX que nos permitirá invocar a la función *FinalGestion* de iAgent, con lo que podremos finalizar la gestión en curso.

```
function FinalGestion() {
    var iagent = null;
    // Creamos el objeto para acceder al iAgentScript
    try {
        iagent = new ActiveXObject("iagent.agentscript");
    }
    catch (e) {
        alert("[inicializar] " + e);
        iagent = null;
        return false;
    }
    ...

    return (iagent.FinalGestion( idFinal, FechaProxContacto, 0, 0 ));
    ...
}
```

El manual de referencia contiene más información acerca de la interfaz de *iAgent Scripting*.