	Nombre Documento	Versión
	Robot NET Manual Técnico	2.0




DIRECTV
Bogotá D. C., Colombia

Departamento de Informática


MANUAL TECNICO DEL SOFTWARE

APLICACIÓN ROBOT NET

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

Contenido

1	INTRODUCCIÓN	6
1.1	Objetivo.....	6
1.2	Alcance	6
1.3	Audiencia	6
1.4	Descripción general.....	6
1.4.1	Descripción de Proceso Técnico.....	7
1.4.2	Despliegue.....	8
1.5	arquitectura general	9
2	BASE DE DATOS.....	10
2.1	Modelo entidad relación.....	10
2.2	TABLAS	11
2.2.1	tbHistorico.....	11
2.2.2	tbHistoricoProvisioning.....	12
2.2.3	tbHistoricoMensajes	12
2.2.4	tbMapURLProvisioning.....	13
2.2.5	tbMapPerfilesInternet.....	13
2.2.6	tbconfiguracion	13
2.2.7	tbParametros	14
2.2.8	tbPrmMensajesProvisioning	14
2.2.9	Log.....	14
2.3	VALORES DE DOMINIO.....	15
2.3.1	Tipos de Eventos	15
2.3.2	Tipos de estado de procesos.....	15
2.4	Store procedures.....	16
2.4.1	stpUpdHistorico	16
2.4.2	stpUpdHistoricoProvisioning.....	16
2.4.3	stpUpdConfiguracion	16

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

2.4.4	stpUpdHistoricoMensajes	17
2.4.5	stpUpdMapURLProvisioning	17
2.4.6	stpUpdLog	17

3 ARQUITECTURA DE APLICACION 18


3.1	Capa de Acceso a Datos	18
3.1.1	Clase CIProveedorBD.....	18
3.1.2	Clase CISQLServer.....	19
3.2	Capa de negocio DTV.Col.RobotInternet.Business	21
3.3	Capa de negocio DTV.Col.RobotInternet.Entities	21
3.3.1	Clases Entidades.....	21
3.3.1.1	Clase clConfiguracion	21
3.3.1.2	Clase clHistorico	23
3.3.1.3	Clase clHistoricoProvisioning	24
3.3.2	Clases Controladoras.....	25
3.3.2.1	Clase ctrConfiguracion	25
3.3.2.2	Clase ctrHistorico	26
3.3.2.3	Clase ctrHistoricoProvisioning.....	27
3.4	Librería DTVCol.RobotInternet.Engine.....	29
3.4.1	Componentes:.....	29
3.5	CAPA DE PRESENTACION	30
3.5.1	Clase clBusinessObject	30
3.5.2	Clase frmAbout	30
3.5.3	Clase frmMain	31
3.5.4	Módulo modMain	32

4 CONFIGURACION DE LA APLICACIÓN 33

4.1	WS Provisioning	33
CustomBinding: Adicional a esto, se debe agregar la configuración para el endpoint "ISO8859Binding". Para esto se agrega las siguientes líneas dentro de la sección <system.serviceModel>:		
4.2	Conexiones a bases de datos transaccionales y de configuración de Robot NET	34
4.3	Otros Parámetros.....	35

5 EJECUCION..... 35


6 EVENTOS 36

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0


6.1	Listado general de eventos	36
-----	----------------------------------	----

7 GLOSARIO..... 37

7.1	Entidad controladora	37
7.2	EVENTO	37
7.3	IBS	37
7.4	Aplicación Satélite.....	37
7.5	Razón.....	38
7.6	Swop.....	38
7.7	Internet Listener	38
7.8	Upgrade/Downgrade	38
7.9	Provisioning.....	38
7.10	Distributor	38
7.11	WS.....	38

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

CONTROL DE VERSIONES			
Versión	Autor(es) Ordenados alfabéticamente	Descripción de la versión	Fecha de Elaboración
Versión 1.0	Yury Carvajal	Documento inicial	2014-02-14
Versión 2.0	Yury Carvajal	Aprovisionamiento de tabletas/prepago. Actualización diagrama de red.	2014-07-14
Versión 3.0	Yury Carvajal	Aprovisionamiento de BTB.	2014-11-21

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

1 INTRODUCCIÓN

1.1 OBJETIVO

El objetivo del documento es describir técnicamente el funcionamiento del sistema **ROBOT NET**.

1.2 ALCANCE

Este documento describe las características técnicas de software **ROBOT NET**.

1.3 AUDIENCIA

Este documento está dirigido a personal involucrado en la administración del aplicativo **ROBOT NET**, así:


- Ingenieros Sistemas & Aplicaciones.
- Desarrolladores.
- Ingenieros de soporte 2° y 3° nivel.

1.4 DESCRIPCIÓN GENERAL

El programa *Robot NET* es una interfaz que se encarga de procesar mensajes generados por diferentes eventos como: activaciones, desconexiones, suspensiones, cancelaciones, [downgrades/upgrades](#), [swops](#) entre otros y enruta los eventos para el sistema de aprovisionamiento de Internet ([Provisioning](#)).

El funcionamiento en general de la aplicación se describe a continuación:


- La aplicación realiza lecturas de la tabla tbHistorico todo el tiempo en busca de eventos que deben ser enviados al sistema Provisioning (estado 0), esta lectura depende de la configuración de los eventos en la tabla tbConfiguracion.
- Los eventos son procesados, validados y enviados al sistema Provisioning.
- El sistema Provisioning procesa la solicitud.
- El evento es actualizado según la respuesta del servicio Provisioning. Si la respuesta no es exitosa se guarda el código y el XML de la respuesta.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

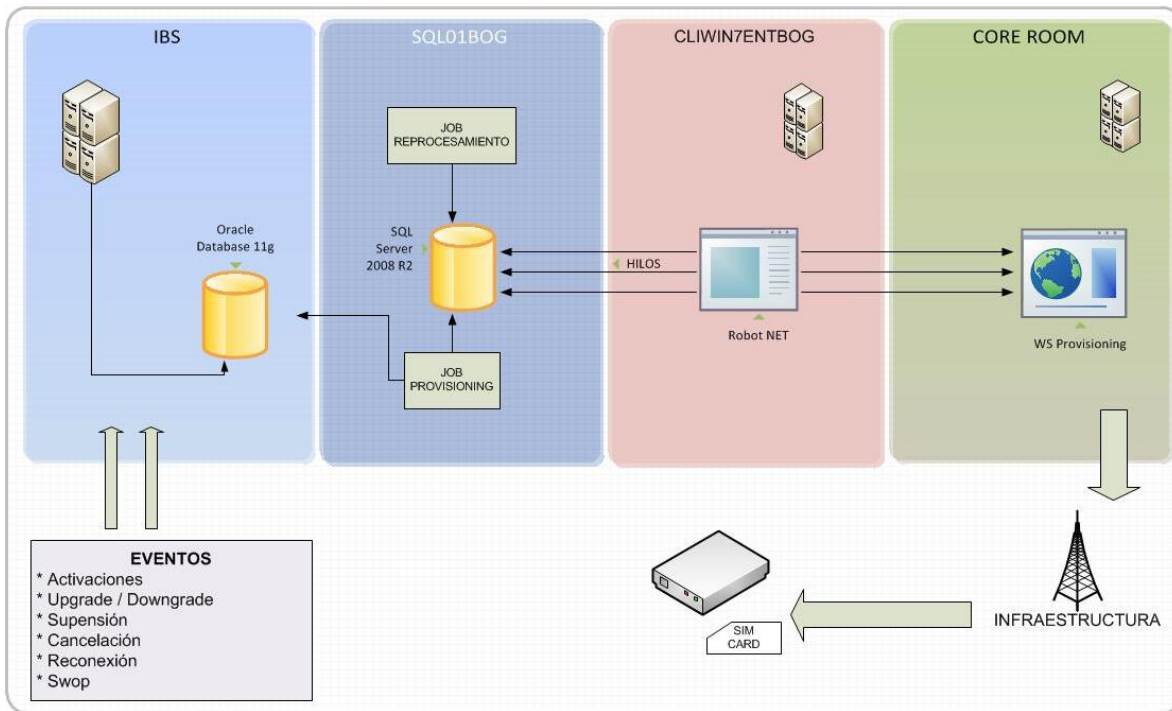
1.4.1 Descripción de Proceso Técnico

El sistema Robot NET se compone de los siguientes procesos:

- Envío de mensajes: Este proceso se encarga de registrar los eventos (activaciones, reconexiones, suspensiones, cancelaciones, downgrades / upgrades y swops) en la base de datos de Robot NET. Este proceso puede ser realizado por alguno de los siguientes sistemas (nunca deben estar en ejecución al mismo tiempo):
 - ETL Provisioning: Es una ETL que procesa cada 5 minutos haciendo lecturas a la base de datos de IBS en busca de eventos y los registra en la base de datos de Robot NET. Actualmente es el medio por el cual se envían mensajes.
 - Internet Listener: Es una interfaz que se encarga de recibir mensajes del Sistema IBS (módulo Distributor) que son generados por diferentes eventos y los registra en la base de datos de Robot NET según un proceso de validación. Actualmente no soporta aprovisionamiento de tabletas, por lo tanto no se usa.
- Procesamiento de mensajes: Este proceso se encarga de enviar al sistema de aprovisionamiento de Internet (Provisioning) los mensajes pendientes por procesar (para esto verifica que se encuentren en estado 0 (cero). [Ver mayor información de Estados de Procesos](#)). El envío de los eventos se realiza mediante Servicios Web que son expuestos por Provisioning. El resultado de este procesamiento es almacenado en la base de datos.
- Reprocesamiento de mensajes: Este proceso es realizado por una ETL que se ejecuta cada 5 minutos, revisando los mensajes que no hayan procesado correctamente. Si el código del error corresponde a un error que debe ser reprocesado, el estado será actualizado a 0 (cero) para que sea procesado nuevamente por el Robot NET.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

A continuación se presenta diagrama que ilustra el funcionamiento de manera general.




1.4.2 Despliegue

En la actualidad (Julio de 2014) la aplicación Robot NET se encuentra instalada en el servidor cliwin7entbog.

Las ETL's para Provisioning y Reprocesamiento y la base de datos de Robot NET se encuentran en el servidor sql01bog.

La aplicación Internet Listener no se encuentra desplegada actualmente.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0


1.5 ARQUITECTURA GENERAL

El sistema **ROBOT INTERNET** se desarrolló con la siguiente Arquitectura:

- Herramienta de Desarrollo y lenguaje de Programación: Microsoft VB .NET utilizando el Framework 4.0 de Microsoft .NET.
- Motor de base de datos: Se utiliza Microsoft SQL Server 2008 R2 para el procesamiento de la información de los eventos.
- Arquitectura de Aplicación: Utiliza modelo de tres capas: capa de negocio, capa de acceso a datos y la capa de presentación (aplicación Windows). Para el acceso a datos se utiliza las librerías de Enterprise Library. Para el monitoreo se utiliza la librería Log4Net.

Las ETL's se desarrollaron bajo la siguiente arquitectura:

- Herramienta de desarrollo SQL Server Business Intelligence Development Studio (Visual Studio 2008).
- Se conecta a las bases Oracle (IBS) y SQL Server (Robot NET) mediante Proveedores Oledb.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

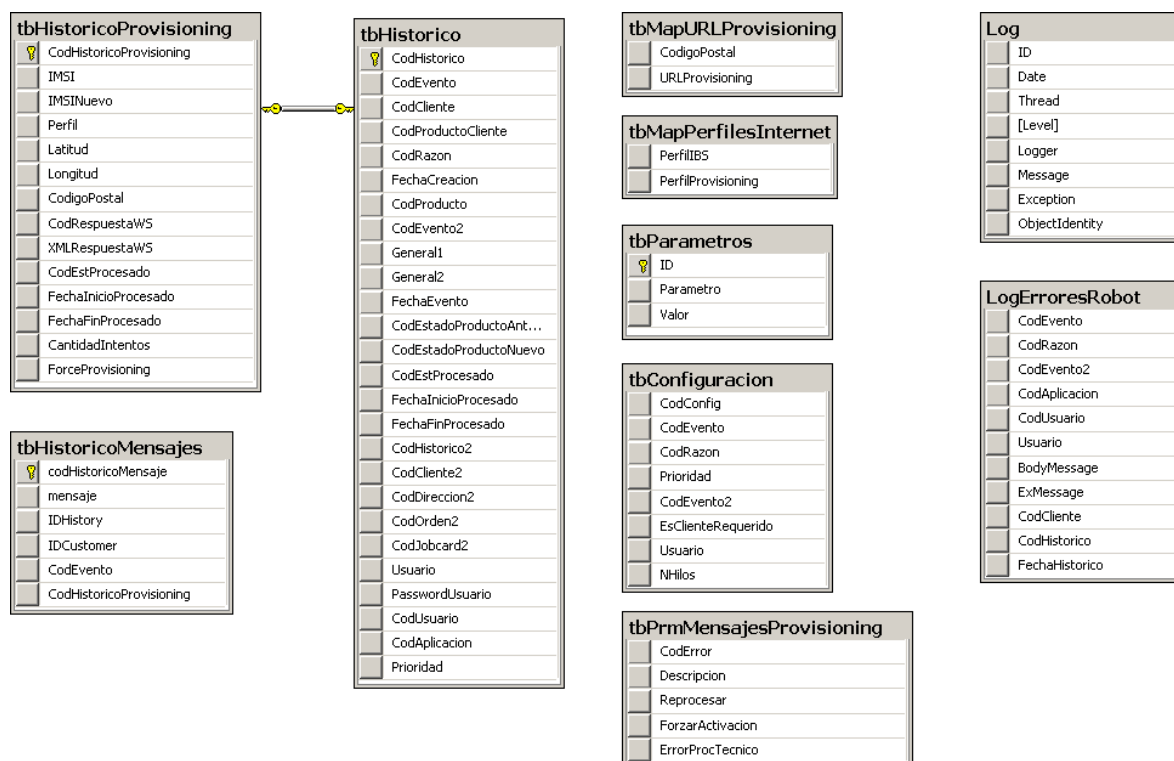
2 BASE DE DATOS


La aplicación Robot NET utiliza motor de base de datos SQL Server 2008 R2.

Las tablas de la base de datos que empiezan por el prefijo “tb”. Son las utilizadas por el sistema Robot Internet para el registro de los eventos. Las demás tablas son “dummies” o tablas temporales.

La base de datos de Robot Internet se llama bdIBSBatchLogInternet.

2.1 MODELO ENTIDAD RELACIÓN



	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

2.2 TABLAS

2.2.1 *tbHistorico*


Esta tabla contiene la información general de los eventos. **Es el registro base** que le indica al Robot Internet que hay un evento a ser procesado, el campo CodEstProcesado indica a la aplicación Robot Internet el estado del procesamiento del registro.

El Robot Internet inicia la lectura de los registros en orden de prioridad desde la fecha más antigua sobre la tabla tbHistorico cuyo **estado** sea cero.

Esta tabla corresponde a la tabla suHistor de la base de datos del sistema IBS.

A continuación se describen los campos de la tabla tbHistórico, los campos que no tienen descripción no se utilizan.

Campo	Descripción
CodHistorico	Campo que es la llave primaria, identifica el registro de evento de manera única.
CodEvento	Código de Tipo de Evento.
CodCliente	Código del Cliente. Es generado por el sistema IBS.
CodProductoCliente	
CodRazon	Código que identifica el Tipo de Razón por el cual se realiza el evento. Cada Tipo de Evento tiene sus razones propias.
FechaCreacion	Fecha de creación del registro en la tabla tbHistorico.
CodProducto	
CodEvento2	Código del proceso que crea el registro. Hace referencia a una función de negocio que se desea rastrear.
General1	
General2	
FechaEvento	Fecha de creación del registro en la tabla tbHistorico.
CodEstadoProductoAnterior	
CodEstadoProductoNuevo	
CodEstProcesado	Código que indica el estado de proceso.
FechaInicioProcesado	Fecha y hora en que se inicia el procesamiento del evento por parte de la aplicación Robot Internet. Inicialmente este campo está en Null.
FechaFinProcesado	Fecha y hora en que se finaliza el procesamiento del evento por parte de la aplicación Robot Internet. Inicialmente este campo está en Null.
CodHistorico2	
CodCliente2	
CodDireccion2	
CodOrden2	
CodJobcard2	
Usuario	Usuario de la aplicación IBS. Usuario con el que se ejecuta el proceso en IBS.
PasswordUsuario	
CodUsuario	
CodAplicacion	
Prioridad	Orden en que se deben ejecutar los eventos.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

2.2.2 *tbHistoricoProvisioning*


Esta tabla contiene información relacionada con eventos de Internet que deben ser enviados al sistema Provisioning.

Campo	Descripción
CodHistoricoProvisioning	Consecutivo de identificación del registro en la tabla. Identifica el registro de manera única.
IMSI	Serial de la Simcard.
Perfil	Velocidad del producto de Internet. Debe ser un valor que registre la tabla tbMapPerfilesInternet en el campo PerfilIBS.
Latitud	Actualmente no se usa, se debe enviar 0.
Longitud	Actualmente no se usa, se debe enviar 0.
CodigoPostal	Identifica la URL del WS de Provisioning a la cual debe ser enviado el evento. Debe ser un valor que registre la tabla tbMapURLProvisioning.
CodRespuestaWS	Código de la respuesta del WS Provisioning.
XMLRespuestaWS	XML de la respuesta del WS Provisioning.
CodEstProcesado	Código de estado del procesamiento del evento
FechaInicioProcesado	Fecha y hora de inicio del procesamiento del evento
FechaFinProcesado	Fecha y hora de finalización del procesamiento del evento
CantidadIntentos	Cantidad de intentos de procesamiento del evento.
ForceProvisioning	Indica si se debe realizar validaciones previamente a la activación. Normalmente se envía en true, pero para tabletas se envía siempre el false.

2.2.3 *tbHistoricoMensajes*

Tabla que contiene cada mensaje que el sistema Listener Internet recibe de IBS (Ver Internet Listener MANUAL TÉCNICO).

Código	Descripción
CodHistoricoMensaje	Consecutivo de identificación del registro en la tabla. Identifica el registro de manera única.
Mensaje	Ruta del mensaje en formato XML establecido previamente por el parámetro PathFilesLog del sistema Listener Internet.
IDHistory	Identifica el histórico que genera el sistema IBS.
IDCustomer	Identifica el cliente en IBS.
CodEvento	Código del evento en IBS.
CodHistoricoProvisioning	Código del histórico de la tabla tbHistoricoProvisioning.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

2.2.4 *tbMapURLProvisioning*

Contiene mapeo de código postal en IBS y URL del WS Provisioning. Si no se encuentra una URL para un código postal, el evento quedará en estado -1. Esta tabla también se encuentra en la base de datos de Robot IVR “bdIBSBatchLog_IVRHSM”.

Código	Descripción
CodigoPostal	Código postal del cliente correspondiente en el sistema IBS.
URLProvisioning	URL del WS Provisioning al cual debe enviarse el evento.

2.2.5 *tbMapPerfilesInternet*


Contiene mapeo de correspondencia de IBS de la velocidad del producto de internet con el perfil de internet de Provisioning.

Código	Descripción
PerfilIBS	Velocidad del producto de internet correspondiente en el sistema IBS.
PerfilProvisioning	Perfil del WS Provisioning.

2.2.6 *tbconfiguracion*

Esta tabla indica los eventos que debe procesar el sistema Robot Internet. Permite configurar los hilos que genera la aplicación Robot Internet.

Código	Descripción
CodConfig	Consecutivo de identificación del registro en la tabla. Identifica el registro de manera única.
CodEvento	Código de evento que se desea procesar.
CodRazon	Código de Razón que se desea procesar.
Prioridad	Prioridad que se asigna para el procesamiento de registros.
CodEvento2	
EsClienteRequerido	Identifica si se van a procesar eventos donde el cliente es requerido (Si o No).
Usuario	
NHilos	Cantidad de hilos que se desean establecer.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

2.2.7 *tbParametros*

Contiene parámetros generales en el sistema Robot Internet.

Código	Descripción
ID	Consecutivo de identificación del registro en la tabla. Identifica el registro de manera única.
Parametro	Nombre del parámetro.
Valor	Valor del parámetro.

2.2.8 *tbPrmMensajesProvisioning*

Contiene valores de parametrización de los mensajes del WS Provisioning. Esta tabla también se encuentra en la base de datos de Robot IVR “bdIBSBatchLog_IVRHSM”.


Código	Descripción
CodError	Código del mensaje del WS Provisioning.
Descripción	Descripción del mensaje.
Reprocesar	Indica si los eventos que tenga este código de respuesta de deben reprocesar.
ForzarActivacion	Indica si HSMobile forzará la activación, es decir, activará el producto si sale el error.
ErrorProcTecnico	Indica si es posible que el error sea causado por alguna mala práctica en el procedimiento técnico, en este caso se hará saber al técnico por medio HSMobile para que revise su instalación. En caso contrario, el error es del sistema y se le pide al técnico notificar el error a TST para que se le de solución de forma adecuada.

2.2.9 *Log*

Esta tabla es para instrumentación de la aplicación. Hace referencia a información de eventos y errores.

Situaciones en que el Robot NET deje de operar –caídas súbitas / abruptas- no permiten -obviamente- registrar los errores.

Código	Descripción
ID	Consecutivo de identificación del registro en la tabla. Identifica el registro de manera única.
Date	Fecha de registro del log.
Thread	Nombre del hilo que genera el log.
Level	Tipo de log (ERROR, INFO).
Logger	Aplicación que genera el log. En este momento podría ser el sistema Robot Internet o el sistema Listener Internet.
Mensaje	Mensaje de error o información
Exception	En caso de ser una excepción durante la ejecución del programa se registra la traza completa en este campo.
ObjectIdentity	Hace referencia al histórico que genera el log.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0


2.3 VALORES DE DOMINIO

2.3.1 *Tipos de Eventos*

Código	Descripción
1013	Activación
1014	Suspensión
1015	Reconexión
1016	Upgrade/Downgrade
1017	Swop
1018	Cancelación

2.3.2 *Tipos de estado de procesos*

Código	Descripción
0	El registro ha sido insertado en la tabla padre (tbHistorico) y en sus respectivas tablas hijas. Está pendiente de ser procesado por la aplicación Robot Internet.
1	El registro ha sido <i>procesado exitosamente</i> por la aplicación robot, fue enviado exitosamente al sistema Provisioning. <i>Es un estado de salida</i> . En ocasiones se presenta que a pesar que quede en estado 1 el registro haya presentado algún inconveniente en su procesamiento. Se deberá identificar si esta situación es un problema del programa Robot o del sistema Provisioning.
99	El registro está en proceso por parte de la aplicación Robot Internet. Este estado garantiza que ningún otro hilo toma el mismo registro. Existen ocasiones que hay transacciones que se quedan indefinidamente en este estado debido a que no se logra completar la transacción, el Robot Internet se cae, etc. También se pueden dar casos en que el registro de la

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

Código	Descripción
	tabla padre se actualiza pero el registro de la “tabla hijo” queda en estado 99.
10	El registro está en espera de otro mensaje de IBS para completar la información necesaria para su ejecución en el sistema Provisioning. Este tipo de registro es utilizado para los swops, ya que maneja dos eventos.
< 0	Son códigos que indican que se ha presentado algún error en el proceso que ejecutó la aplicación Robot Internet con el sistema Provisioning. Son estados de Salida.

2.4 STORE PROCEDURES

Por cada entidad existente en el Robot NET existe un store procedure con el cual se realiza el mantenimiento sobre la misma: Creación, modificación o consultas.

De manera general los store Procedures realizan las operaciones de mantenimiento con Tipo de Operación así:

- Inserción: 0 – cero.
- Modificación : 1
- Borrado : 2
- Consulta un registro : 3
- Consulta todos los registros : 4

2.4.1 *stpUpdHistorico*


Realiza el mantenimiento sobre la tabla [tbHistorico](#). Este store procedure además soporta el Tipo de Operación = 4.

2.4.2 *stpUpdHistoricoProvisioning*

Realiza el mantenimiento sobre la tabla [tbHistoricoProvisioning](#).

2.4.3 *stpUpdConfiguracion*

Realiza el mantenimiento sobre la tabla [tbConfiguracion](#).

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

2.4.4 *stpUpdHistoricoMensajes*


Realiza el mantenimiento sobre la tabla tbHistoricoMensajes.

2.4.5 *stpUpdMapURLProvisioning*

Realiza el mantenimiento sobre la tabla tbMapURLProvisioning. Este procedimiento almacenado también se encuentra en la base de datos de Robot IVR “bdIBSBatchLog_IVRHSM”.

2.4.6 *stpUpdLog*

Realiza el mantenimiento sobre la tabla Log.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

3 ARQUITECTURA DE APLICACION

El proyecto utiliza el concepto de “Entidad Controladora”.

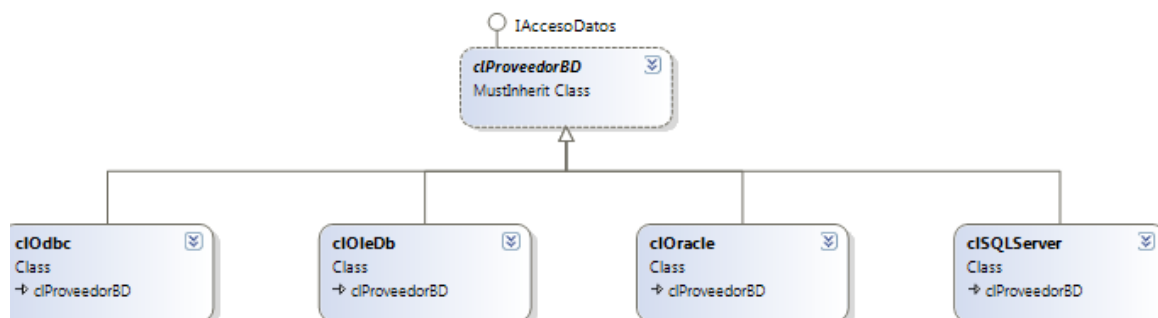
El proyecto utiliza arquitectura de tres capas, las cuales se describen a continuación:

3.1 CAPA DE ACCESO A DATOS

El proyecto hace uso de la librería DTVCol.RobotInternet.DataAccess para interactuar con el repositorio. Para la conexión con la base de datos se utiliza Enterprise Library.

El proyecto tiene una interfaz de acceso a datos llamada IAccesoDatos la cual ejecuta Sentencias SQL, Store Procedures, permite conectarse, pasar parámetros, retorna conjuntos de datos a través de Datasets.


A continuación se presenta diagrama general del acceso a datos.

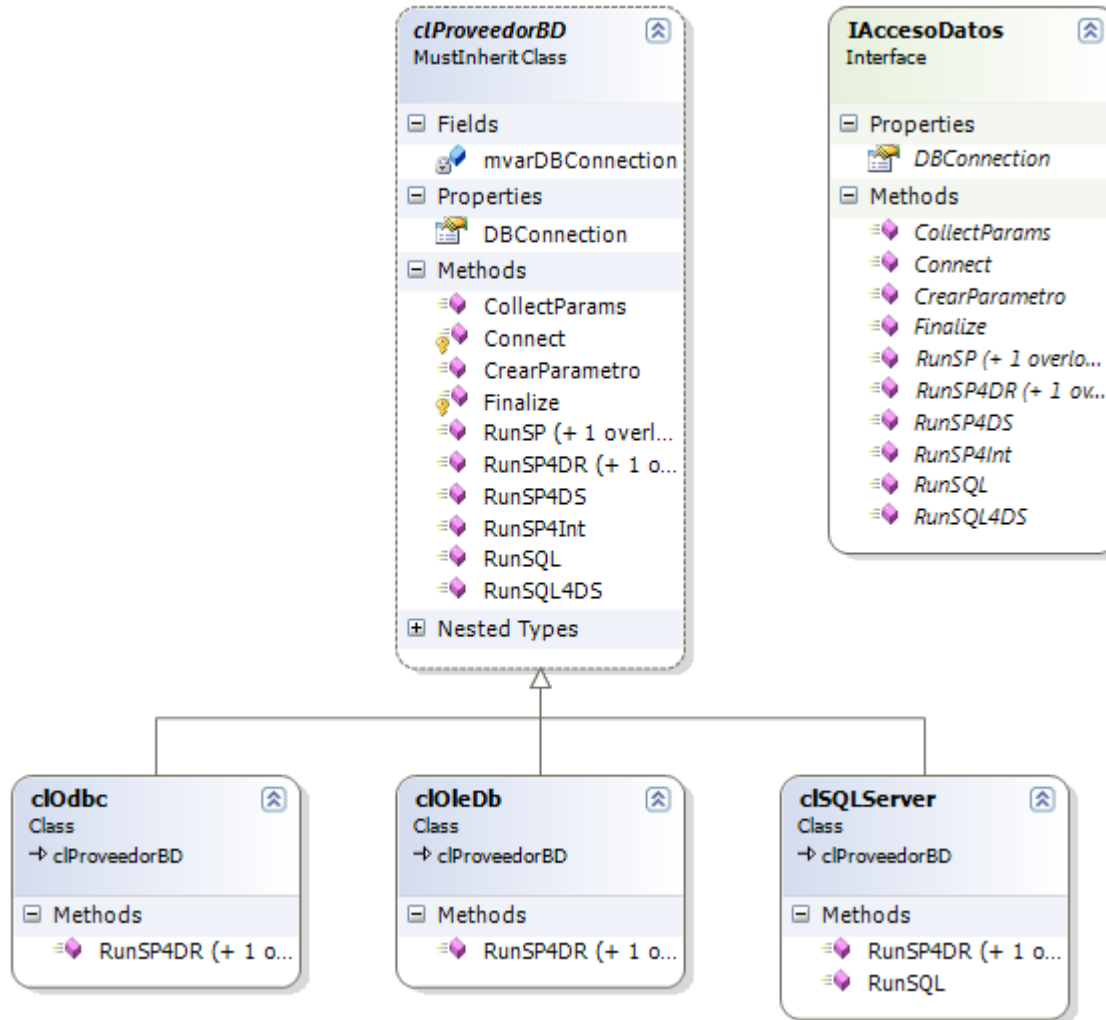


Clases

3.1.1 Clase ClProveedorBD

Es la Clase base a partir de la cual se realiza el acceso a la base de datos de la aplicación Robot Internet. De esta Clase se deriva la clase clSqlServer que es la clase utilizada para el motor específico que se usa en la aplicación: Sql Server.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0




3.1.2 Clase *clSQLServer*

Clase derivada de **clProveedorDb**, contiene los métodos para realizar las operaciones de acceso a la base de datos.

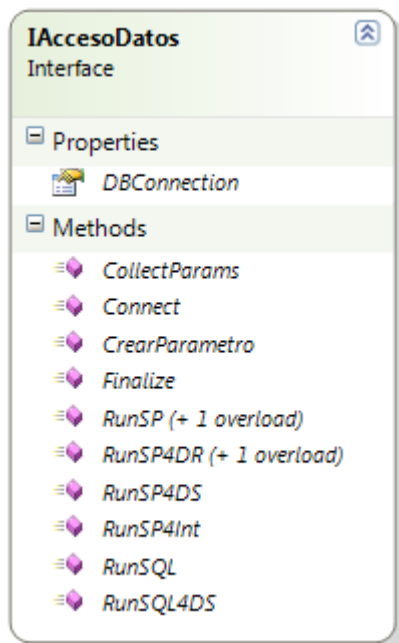
Métodos


- **RunSP4DR:** Ejecuta Store Procedure sobre la base de datos.
- **RunSQL:** Ejecuta Sentencia SQL sobre la base de datos.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

2

2.1.6 Interface IAccesoDatos



	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

3.2 CAPA DE NEGOCIO DTV.COL.ROBOTINTERNET.BUSINESS

Esta librería contiene fachada mediante la cual se usan los servicios de la librería DTVCol.RobotInternet.Engine.

3.3 CAPA DE NEGOCIO DTV.COL.ROBOTINTERNET.ENTITIES

Esta librería contiene librería que realiza lógica para definición de entidades y controladoras que utiliza la aplicación.


En esta librería se implementa el concepto de “Entidad Controladora” mediante el cual se tienen entidades que realizan mapeos hacia las tablas de la base de datos.

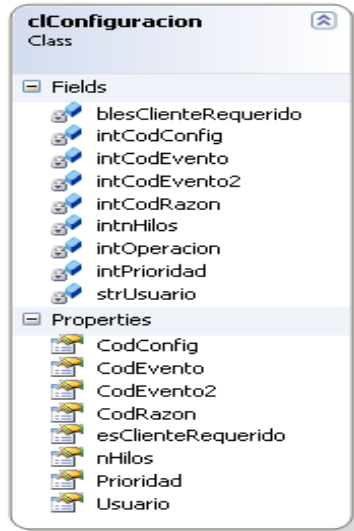
3.3.1 *Clases Entidades*


Estas clases en sus propiedades contienen la información que contiene la tabla correspondiente, por cada campo que tiene la tabla en la base de datos existe una propiedad en la clase correspondiente.

3.3.1.1 *Clase clConfiguracion*

Esta entidad realiza el mapeo sobre la [tabla tbConfiguracion](#).

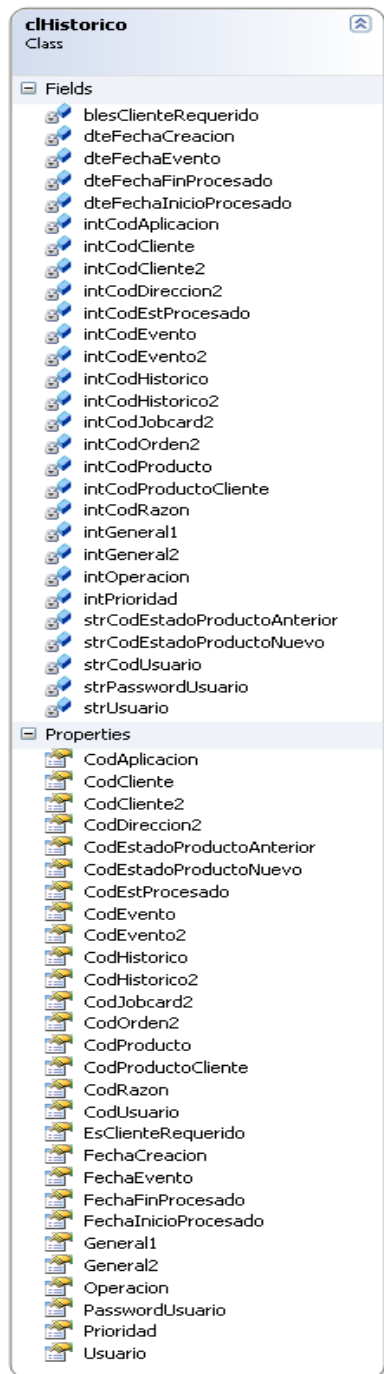
	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0




	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

3.3.1.2 Clase *clHistorico*

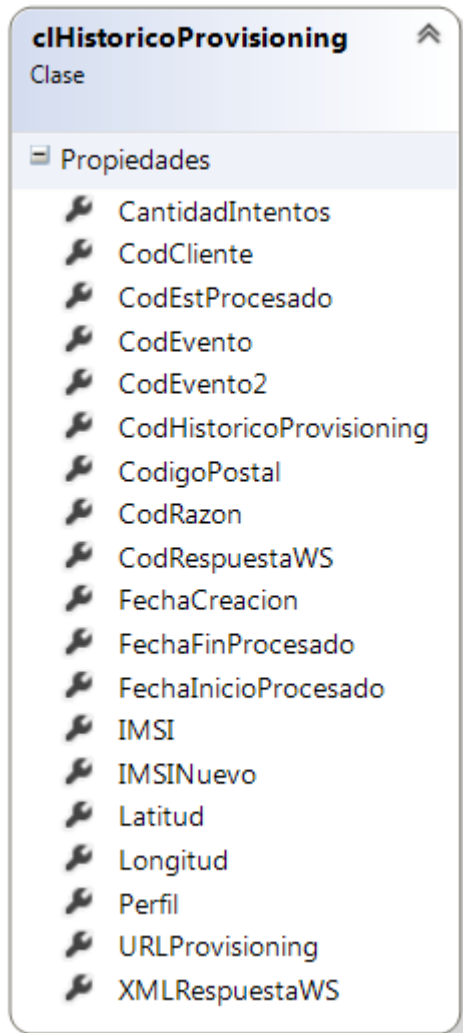
Esta entidad realiza el mapeo sobre la [tabla tbHistorico](#).




	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

3.3.1.3 Clase *clHistoricoProvisioning*

Esta entidad realiza el mapeo sobre la tabla [tbHistoricoProvisioning](#).



	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

3.3.2 Clases Controladoras

3.3.2.1 Clase ctrConfiguracion

Hereda de ctrMaster, tiene propiedad de tipo [clConfiguracion](#)

Constructor


Asigna a la propiedad [strNombreStp](#) el valor “stpUpdConfiguracion”.

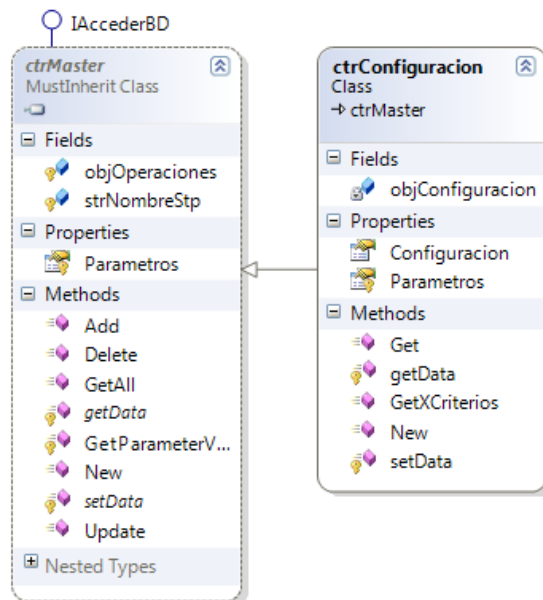
Propiedades

- **Configuracion:** Es de tipo [clConfiguracion](#) mediante esta propiedad se realiza el mapeo a la tabla [tbConfiguracion](#).
- **Parametros:** Recibe Tipo de Operación, en esta propiedad se realiza el mapeo de parámetros para ejecutar el store procedure [stpUpdConfiguracion](#). Para realizar la creación de parámetros llama al método “CrearParametro” de la clase [clProveedorDb](#).

Métodos

- **Get:** Retorna instancia de objeto de tipo [clConfiguracion](#), para ello:
 - Ejecuta el método GetData el cual retorna un DataSet.
 - A partir del DataSet se realiza el mapeo (cargue) del objeto de la entidad correspondiente [clConfiguracion](#) para el caso.
- **GetXCriterios:** Tiene como parámetro un numero para ejecutar una operación del Store Procedure, retorna DataSet.
- **setData:** Escribe sobre la base de datos. Robot Internet escribe cuando realiza actualización de estado. Se le pasa el string de conexión.
- **getData:** Ejecuta store procedure con la colección de parámetros que se ha asignado. Se le pasa el string de conexión.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0



3.3.2.2 Clase ctrHistorico

Hereda de ctrMaster, tiene propiedad de tipo [clHistorico](#).

Constructor


Asigna a la propiedad strNombreStp el valor “[stpUpdHistorico](#)”.

Propiedades

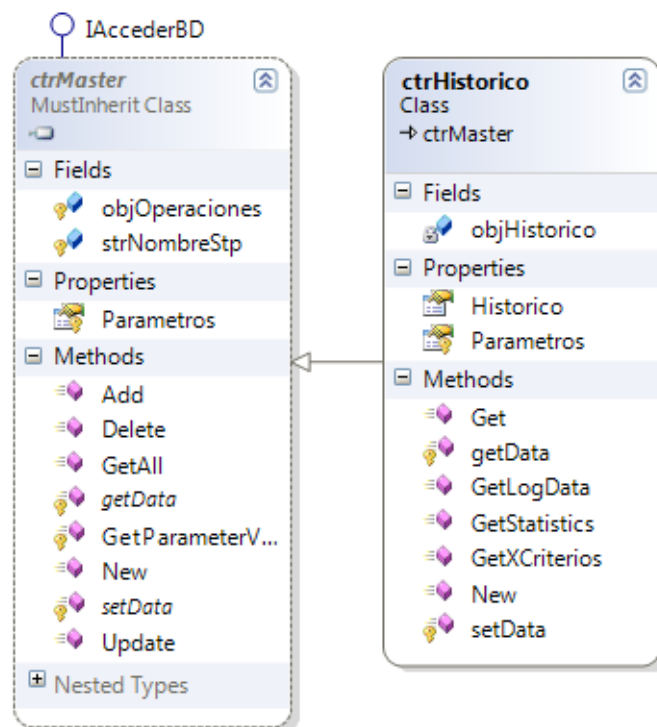
- **Configuracion:** Es de tipo [clHistorico](#) mediante esta propiedad se realiza el mapeo a la tabla [tbHistorico](#).
- **Parametros:** Recibe Tipo de Operación, en esta propiedad se realiza el mapeo de parámetros para ejecutar el store procedure [stpUpdHistorico](#). Para realizar la creación de parámetros llama al método “CrearParametro” de la clase [clProveedorDb](#).

Métodos

- **Get:** Retorna instancia de objeto de tipo [clHistorico](#), para ello:
- Ejecuta el método GetData el cual retorna un DataSet.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

- A partir del DataSet se realiza el mapeo (cargue) del objeto de la entidad correspondiente **clHistorico** para el caso.
- **GetXCriterios:** Tiene como parámetro un numero para ejecutar una operación del Store Procedure, retorna DataSet.
- **setData:** Escribe sobre la base de datos. Robot Internet escribe cuando realiza actualización de estado. Se le pasa el string de conexión.
- **getData:** Ejecuta store procedure con la colección de parámetros que se ha asignado. Se le pasa el string de conexión.




3.3.2.3 Clase *ctrHistoricoProvisioning*

Hereda de **ctrMaster**, tiene propiedad de tipo **clHistoricoProvisioning**.

Constructor

Asigna a la propiedad **strNombreStp** el valor "**stpUpdHistoricoProvisioning**".

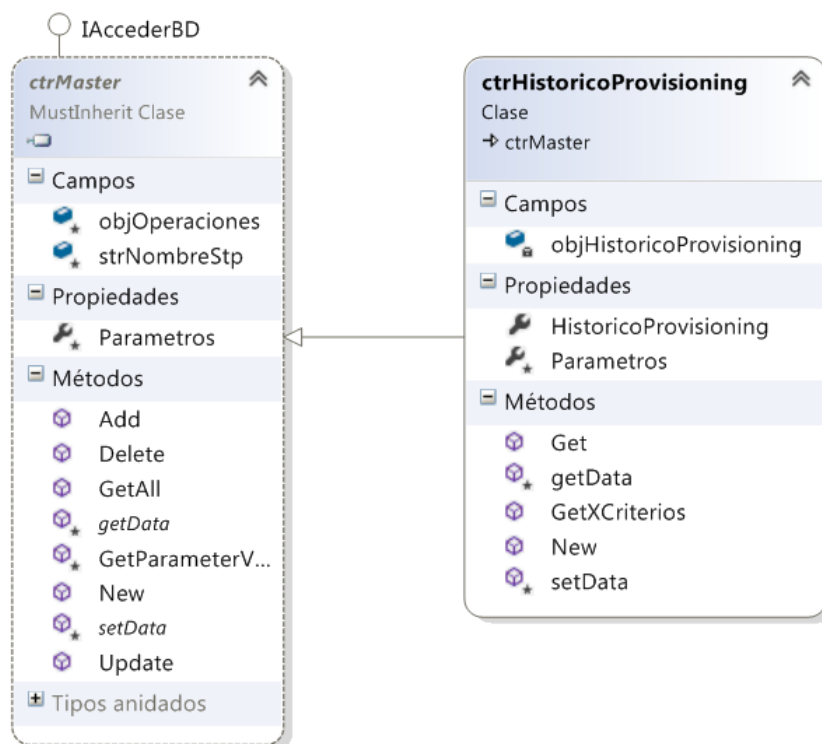
	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0


Propiedades

- **HistoricoProvisioning:** Es de tipo `clHistoricoProvisioning`.
- **Parametros:** Recibe Tipo de Operación, en esta propiedad se realiza el mapeo de parámetros para ejecutar el store procedure `stpUpdHistoricoProvisioning`. Para realizar la creación de parámetros llama al método “CrearParametro” de la clase `clProveedorDb`.

Métodos

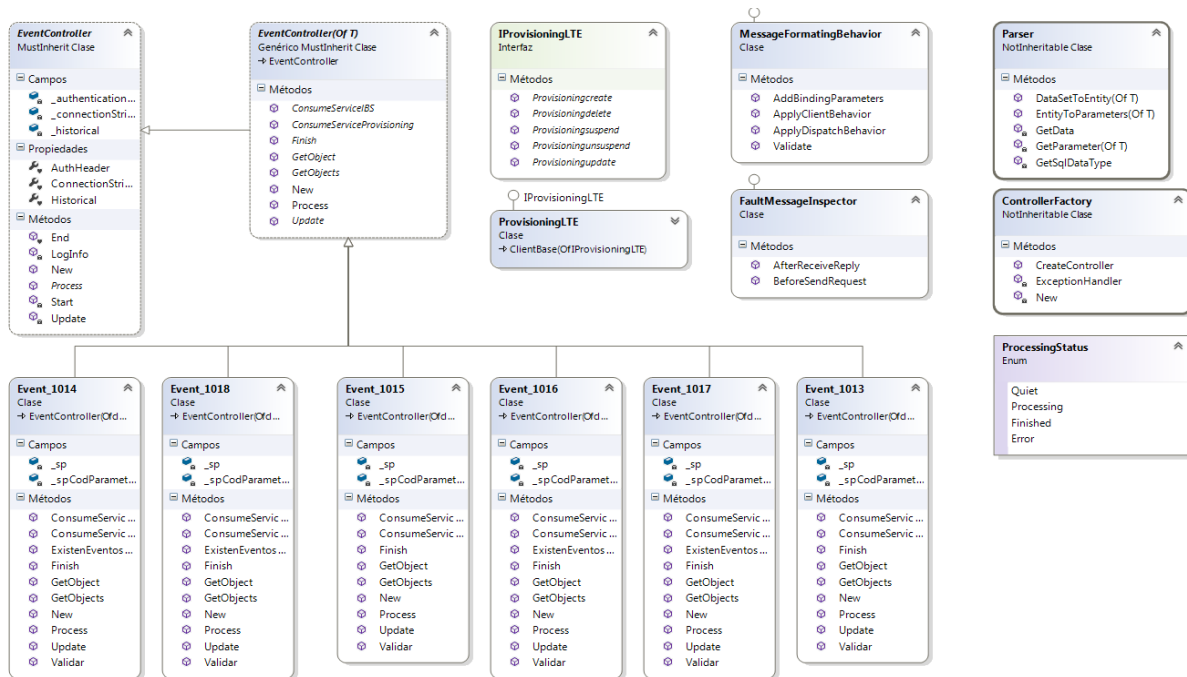
- **Get:** Retorna instancia de objeto de tipo `clHistoricoProvisioning`, para ello:
 - Ejecuta el método `GetData` el cual retorna un `DataSet`.
 - A partir del `DataSet` se realiza el mapeo (cargue) del objeto de la entidad correspondiente `clHistoricoProvisioning` para el caso.
- **GetXCriterios:** Tiene como parámetro un numero para ejecutar una operación del Store Procedure, retorna `DataSet`.
- **setData:** Escribe sobre la base de datos. Robot Internet escribe cuando realiza actualización de estado. Se le pasa el string de conexión.
- **getData:** Ejecuta store procedure con la colección de parámetros que se ha asignado. Se le pasa el string de conexión.



	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0


3.4 LIBRERÍA DTVCOL.ROBOTINTERNET.ENGINE

Esta librería contiene librería que realiza lógica para procesamiento de los mensajes. Consta de la siguiente estructura:



3.4.1 Componentes:

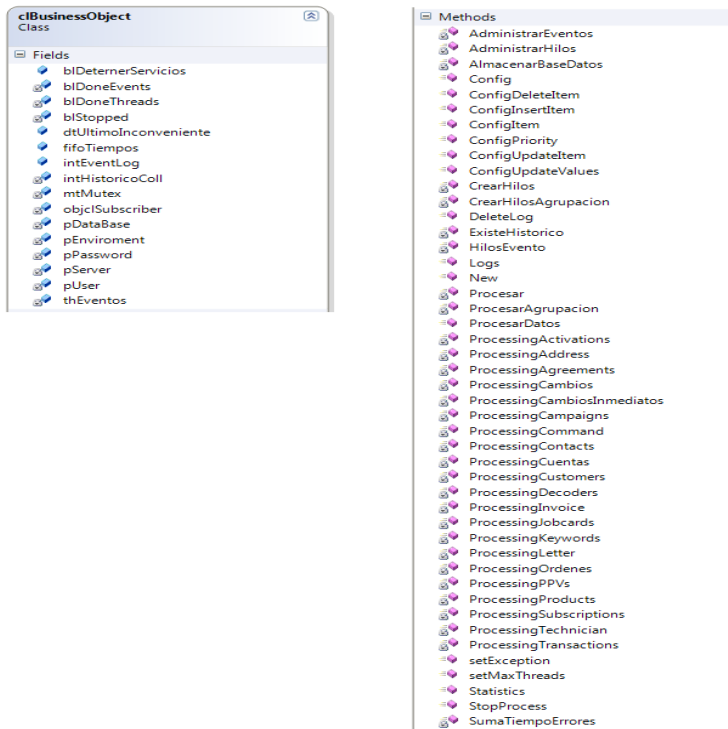
Nombre	Tipo	Descripción
ControllerFactory	Clase	Se encarga de inicializar las clases con que se ejecutan los eventos.
EventController	Clase	Clase abstracta que maneja los parámetros comunes para todos los eventos de la cual heredan todos los que se van a procesar.
Event_1013	Clase	Procesamiento del evento 1013 [Herencia de EventController].
Event_1014	Clase	Procesamiento del evento 1014 [Herencia de EventController].
Event_1015	Clase	Procesamiento del evento 1015 [Herencia de EventController].
Event_1016	Clase	Procesamiento del evento 1016 [Herencia de EventController].
Event_1017	Clase	Procesamiento del evento 1017 [Herencia de EventController].
Event_1018	Clase	Procesamiento del evento 1018 [Herencia de EventController].
Parser	Clase	Se encarga de transformar las entidades a objetos que la base de datos comprenda
ProcessingStatus	Enumeracion	Contiene los estados de procesamiento de los registros en las tablas de Robot
IProvisioningLTE	Interfaz	Se encarga de implementar los métodos hacia el WS Provisioning.
ProvisioningLTE	Clase	Se encarga de implementar los métodos hacia el WS Provisioning.
MessageFormatingBehavior	Clase	Maneja el comportamiento de los mensajes que retorna el WS Provisioning.
FaultMessageInspector	Clase	Transforma los mensajes XML de error en excepciones que puedan ser controladas por el proceso.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

3.5 CAPA DE PRESENTACION

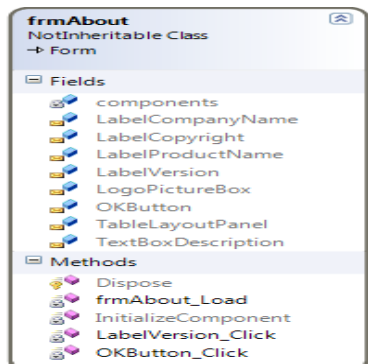
3.5.1 Clase *clBusinessObject*


Esta clase realiza la configuración de hilos del programa. Adicionalmente esta clase es la encargada de identificar los tipos de eventos y su respectivo procesamiento.



3.5.2 Clase *frmAbout*

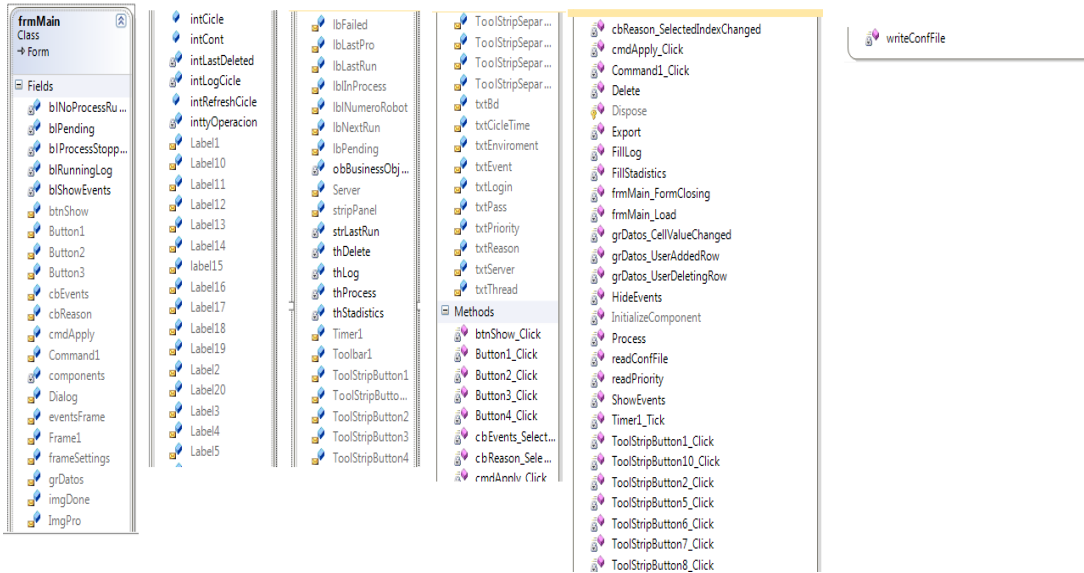
Formulario que contiene información del programa Robot NET respecto a aspectos como: Versión, Descripción de la aplicación.




	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

3.5.3 Clase frmMain

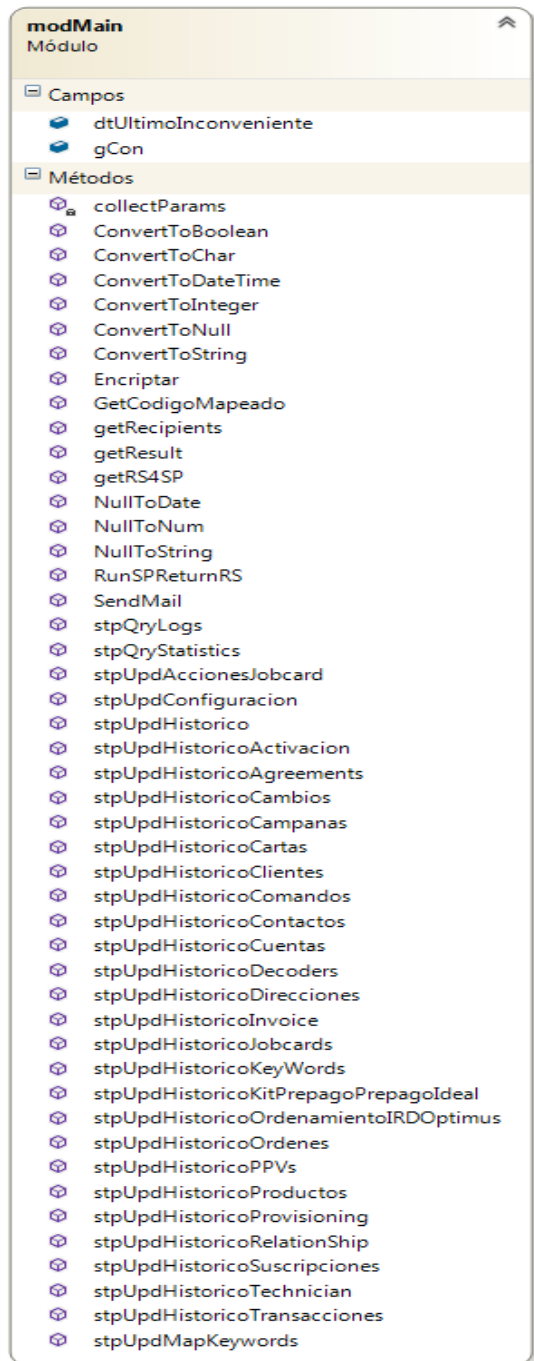
Contiene formulario que define la Interfaz de usuario con que el usuario de la aplicación Robot Internet interactúa. Este formulario permite configurar el número de hilos que la aplicación Robot Internet procesa.




	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

3.5.4 *Módulo modMain*

Contiene definición de métodos comunes que son definidos en la capa de presentación.



	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

4 CONFIGURACION DE LA APLICACIÓN

En el archivo de configuración de la aplicación App.Config se realizan las siguientes configuraciones:


4.1 WS PROVISIONING

EndPoint: Información del Web Service Provisioning. En este endpoint **NO** se define la URL del servicio (Ver tabla [tbMapURLProvisioning](#)).

```
<endpoint binding="customBinding" bindingConfiguration="ISO8859Binding"
          contract="DTVCol.RobotInternet.Engine.Provisioning.IProvisioningLTE"
          name="IProvisioningLTE" />
```

CustomBinding: Adicional a esto, se debe agregar la configuración para el endpoint “ISO8859Binding”. Para esto se agrega las siguientes líneas dentro de la sección <system.serviceModel>:

```
<extensions>
  <bindingElementExtensions>
    <add name="customTextMessageEncoding"
        type="Microsoft.Samples.CustomTextMessageEncoder.CustomTextMessageEncodingElement,
        CustomTextMessageEncoder, Version=4.0.0.0, Culture=neutral, PublicKeyToken=null"/>
  </bindingElementExtensions>
</extensions>
<bindings>
  <customBinding>
    <binding name="ISO8859Binding">
      <customTextMessageEncoding encoding="ISO-8859-1" messageVersion="Soap11"/>
      <httpTransport/>
    </binding>
  </customBinding>
</bindings>
<client>
  <endpoint name="IProvisioningLTE" binding="customBinding"
    bindingConfiguration="ISO8859Binding"
    contract="DTVCOL.IVR.TecnicosPospago.Services.Provisioning.ServiceContracts.IProvis
    ioningLTE"/>
</client>
```

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

4.2 CONEXIONES A BASES DE DATOS TRANSACCIONALES Y DE CONFIGURACIÓN DE ROBOT NET


String de Conexiones db SQL: Información de configuración de las fuentes de datos que se utilizan en el proyecto así:

- cnStrbdIBSBatchLog: Base de datos donde se registran los eventos.
- cnStrbdIBSBatchLogConfiguracion: Base de datos que tiene la configuración de los registros que se van a procesar en el sistema Robot Internet.

Nota: Los string de conexión mencionados pueden hacer referencia a la misma base de datos.

Ejemplo

```
<connectionStrings>
  <add name="cnStrbdIBSBatchLogConfiguracion" connectionString="Data Source=ITFAB-
92\STEFANINI;Initial Catalog=bdIBSBatchLogConfig;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False"
providerName="System.Data.SqlClient" />
  <add name="cnStrbdIBSBatchLog" connectionString="Data Source=ITFAB-
92\STEFANINI;Initial Catalog=bdIBSBatchLog;User ID=sa; Password=sa;Connect Timeout=240"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0


4.3 OTROS PARÁMETROS

Nombre parámetro	Descripción	Observaciones
<code>RefreshCicleTime</code>	Tiempo entre eventos de timer de la aplicación.	
<code>ThreadSleep</code>	Tiempo que se establece para enviar comando de sleep al Thread	
<code>log4net.AppName</code>	Nombre de la aplicación (ejem:Robot Internet) esto es con el fin de tener una correlacion con el archivo de configuración de Log4Net	Debe ser correlacionado con el nodo <code><logger name="</code> Del archivo de configuración de Log4Net
<code>log4net.Internal.Debug</code>	Indica si log4net generara excepciones en depuración	NA

5 EJECUCION

Procedimiento


- Se coloca el ejecutable de la aplicación y se configura el archivo App.config.
- La aplicación abre una ventana donde se muestra la configuración de hilos por evento
- En la ventana principal se muestran los botones de iniciar, detener y cerrar el robot Internet.
- Cuando se oprime el botón Iniciar muestra la cantidad de registros que el sistema va procesando.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

6 EVENTOS

6.1 LISTADO GENERAL DE EVENTOS

Transacción	Modulo	Código de Evento IBS	Robot Internet		Observaciones
			Código de Evento	CodRazon	
Activación	Producto	5229	1013	10	Créate
Enter serial number	Device	148	1013		Créate
Desconexión	Producto	133	1014		Suspend
Suspensión	Producto	5245	1014		Suspend
Desconexión automática	Producto	1133	1014		Suspend
Reconexión	Producto	171	1015		Unsuspend
Upgrade / Downgrade de producto	Producto	108	1016		Update
Swop	Device	172	1018	30 (Swap Desconectar)	Delete
		173	1013	40 (Swap Conectar)	Create
Cancelación	Producto	142	1014	20	Suspend
Cambio de estado manual Simcard	Device	156	1018		Delete

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

7 GLOSARIO

7.1 ENTIDAD CONTROLADORA

La arquitectura definida para la aplicación Robot Internet hace uso de este concepto, mediante el cual la entidad controladora realiza los llamados al Store Procedure y retorna la información a la entidad que realice el llamado.

7.2 EVENTO

Un evento es una acción que ocurre sobre el sistema.

Ejemplos:

- Activación
- Swop

7.3 IBS


CRM que utiliza DIRECTV. Su motor de base de datos es Oracle. Está desarrollado en Visual Basic con Visual Studio 2010. Provee Web Services utilizando Windows Communication Foundation (WCF). Tiene base de clientes activa e inactiva de DIRECTV. IBS funciona por [eventos](#) y razones.

El acceso a IBS por política institucional está restringido a personal interno de la Compañía, los socios de negocios como los Dealers realizan registro y acceso a la información mediante [aplicaciones satélites](#).

7.4 APLICACIÓN SATÉLITE

Aplicación que alimenta o sirve de interfaz con el sistema IBS. Algunas son:

- Smart Dealer.
- IVR
- Recarga.
- Recaudos.
- HS Mobile
- Procesos masivos.

	Nombre Documento	Versión
	Robot NET Manual Técnico	3.0

7.5 RAZÓN

Establece o identifica en IBS el por qué se realiza un **evento** o acción. Las razones están clasificadas por Eventos, cada evento tiene sus razones específicas.

7.6 SWOP

Es el proceso de cambio de decodificador, bien sea por daño o por cambio de tecnología.

7.7 INTERNET LISTENER

El programa Internet Listener es una aplicación Windows que recibe mensajes de los eventos de IBS y los registra en la base de datos Sql Server a partir del cual el programa *Robot Internet* realiza su procesamiento.

7.8 UPGRADE/DOWNGRADE

Se refiere al cambio de velocidad en un producto de Internet (Upgrade: aumento de velocidad, Downgrade: disminución de velocidad).

7.9 PROVISIONING

Sistema de aprovisionamiento de Internet.

7.10 DISTRIBUTOR

Módulo del sistema para mensajería de transacciones.

7.11 WS

Web Service.