



EDICOM BUSINESS INTEGRATOR

Título del documento:	EDICOM BUSINESS INTEGRATOR
Nombre del fichero:	ES_Edicom Business Integrator.odt
Versión:	1.0
Estado:	Versión Preliminar.
Fecha:	4/10/13
Autor:	jvalero

Revisión, Aprobación

Revisado por:		Fecha:
Aprobado por:		Fecha:

Historial de cambios

Versión	Fecha	Descripción de la acción	Páginas
1.0	4/10/13	Versión inicial	Todas
1.1	14/10/13	Se documentan las funciones FileSystemObject	
1.2	24/10/13	Se documentan las funciones XMLDOM	
1.3	24/10/13	Se añade información sobre las variables que hay disponibles durante una transformación	
1.4	28/10/13	En la función EBIPublishAS2, se documenta la posibilidad de utilizar patrones por el subject del mensaje	
1.5	29/10/13	Se añaden las funciones restful subscription y confirmsubscription	
1.6	19/11/13	Se corrige un problema en la documentación de la función ConcatFiles	
1.7	23/12/13	Se añade información relativa a los ficheros de log que generan los adaptadores	
1.8	13/01/14	Se modifica la documentación de la funciones EBIPublish, ExecuteFile	
1.9	14/01/14	En las funciones de VFS se incluye la opción de setUserDirIsRoot	
1.10	20/01/14	Se añade la función EBIRemotePublish	
1.11	10/03/14	Nuevos parámetros de configuración del EBIBROKER	
1.12	10/03/14	Se añaden la funciones AddTextToFile y AddTrace	
1.13	14/03/14	Se añade la función EBIExportHeadersReport	
1.14	24/04/14	Se añade la documentación de las planificaciones a intervalos exactos	
1.15	13/05/14	Nueva función EBIEdiwinBulkImportI	

1.16	19/05/2014	Se añade la función splitLOT	
1.17	22/05/2014	Se añade la función EBIPutRepositoryItem	
1.18	04/06/2014	Nuevo parámetro a nivel de aplicación para poder trocear las publicaciones acorde al interfaz de publicación.	
1.19	13/06/2014	El adaptador soporta que se le indique la contraseña encriptada.	
1.20	05/09/2014	Se añade un nuevo parámetro a las funciones de envío a AS2, para confirmar o no la entrega del mensaje	
1.21	17/02/2015	Se añade la función EBISubsAndTransformAndBulkImportD	
1.22	23/02/2015	Se añaden las funciones EBIEdiwinDeleteVolumeDocument y EBIEdiwinDeleteDocument	
1.23	08/04/2015	Nuevos parámetros en el adaptador SAP	

Índice de contenido

1Enterprise Application Integration (EAI).....	11
1.1Introducción.....	11
1.2Diferencias y similitudes entre EAI tradicional y la Integración B2B	12
1.3Niveles de Integración de Aplicaciones.....	12
1.4EDICOM BUSINESS INTEGRATOR (EBI): una herramienta orientada a integración de procesos empresariales.....	16
2El EBI Broker.....	18
2.1Configuración.....	18
2.2Soporte de balanceo de carga.....	22
2.3Acceso vía webServices.....	22
2.3.1EBIBrokerWS.....	22
2.3.2EBIBrokerWSV2.....	30
2.4Acceso vía restful.....	39
2.4.1Publicación.....	40
2.4.2Suscripción.....	43
2.4.3Confirmación.....	44
2.4.4Transformación y publicación.....	44
2.4.5Ejecución de plantillas.....	45
2.4.6Exportación e Importación de la configuración.....	46
2.5Acceso vía FTP/S y SFTP.....	47
2.5.1Sistema virtual de carpetas.....	47
2.5.2Configuración	49
2.6Ficheros de log	50
3Adaptadores.....	52
3.1Funciones de propósito general	54
3.1.1FileSystemObject.....	54
FileSystemObject.....	55
File.....	58
Folder.....	59
TextStream.....	60
Ejemplos.....	61
3.1.2XMLDOM.....	62
XMLDOM.....	62
XmlNodeList.....	64
XmlNode.....	64
XmlAttr.....	68
XmlText.....	68
XmlElement.....	68
XmlError.....	69
XmlNodeNamedNodeMap.....	70
Ejemplos.....	70
3.1.3Connection.....	71
Connection.....	72
Recordset.....	73
Ejemplos.....	75
3.1.4Variables disponibles.....	75
3.1.5Funciones de propósito general.....	76
AddError.....	76

AddFile.....	76
AddFiles.....	76
AddTextToFile.....	76
AddTrace.....	76
B64Encode.....	77
B64Decode.....	77
CompressB64Zlib.....	77
CompressB64ZlibFile.....	77
CompressZip.....	77
ConcatFiles.....	77
ConnectHTTP.....	78
CopyTextFile.....	79
CreateFolder.....	79
DeCompressB64Zlib.....	79
DeCompressZip.....	79
DeleteCAB.....	80
DeleteChar.....	80
DeleteFile.....	80
DeleteFileIfEmpty.....	80
DeleteFileOld.....	80
DeleteLine.....	80
DynamicClientSOAP.....	81
ExecAndWait.....	81
ExecuteFile.....	81
FileExists.....	82
FormatFloat.....	82
FTPGet.....	82
FTPPut.....	83
getDateFormat.....	83
getError.....	84
getLogFileName.....	84
getScriptName.....	84
getStartDate.....	84
isError.....	84
GunzipFile.....	84
MoveTextFile.....	85
MoveFile.....	85
Pos.....	85
ReplaceChar.....	85
ReplaceStr.....	86
SendMail.....	86
Sleep.....	86
SplitDOC.....	87
SplitLOT.....	87
Sync.....	87
UncompressFile.....	87
WriteLog.....	88
3.2Adaptador EDICOM.....	88
3.2.1Configuración.....	88
3.2.2Funciones.....	89

DsXml.....	89
EBIAcquireApplication.....	90
EBIBeginTrans.....	90
EBIChangeDocumentState.....	90
EBIChangeDocumentStateV2.....	90
EBIChangeMessageReference.....	91
EBIChangeMessageReferenceV2.....	91
EBICommitTrans.....	91
EBIConfigApp.....	91
EBIConfigDomain.....	92
EBIConfirmedPublications.....	92
EBIConfirmedPublicationsv2.....	92
EBIConfirmSubscription.....	92
EBIConfirmSubscriptions.....	92
EBICriticalSectionAcquire.....	93
EBICriticalSectionRelease.....	93
EBICriticalSectionUpdateTimeStamp.....	93
EBIDocumentDeleteOld.....	93
EBIDocumentList.....	93
EBIDocumentListV2.....	94
EBIEquiv.....	95
EBIExecuteTemplate.....	95
EBIGetCounterValue.....	96
EBIGetMessage.....	96
EBIGetRepositoryItem.....	96
EBIItemCount.....	96
EBIItemList.....	96
EBIItemListElement.....	97
EBIList2Array.....	97
EBIMessageByReference.....	98
EBIMessageID.....	98
EBIMessageReference.....	98
EBINextCounterValue.....	98
EBINotifiedPublication.....	98
EBINotifiedPublicationV2.....	98
EBINotifiedRejectedPublication.....	99
EBINotifiedRejectedPublicationV2.....	99
EBIPOPAndPublish.....	99
EBIPublish.....	100
EBIPublishDB.....	101
EBIPublishString.....	102
EBIPublishVFS.....	103
EBIPublishXSLT.....	104
EBIPutRepositoryItem.....	105
EBIRejectedPublications.....	105
EBIRejectedPublicationsV2.....	105
EBIRejectSubscription.....	106
EBIRejectSubscriptionV2.....	106
EBIReleaseApplication.....	106
EBIRemotePublish.....	106

EBIRemoteSQL.....	108
EBIRollbackTrans.....	108
EBISendCMIS.....	109
EBISubsAndFTP.....	110
EBISubsAndMail.....	110
EBISubsAndOpen.....	111
EBISubsAndPrint.....	111
EBISubsAndReport.....	112
EBISubsAndSave.....	112
EBISubsAndTransform.....	113
EBISubsAndTransformAndFTP.....	114
EBISubsAndTransformAndMail.....	115
EBISubsAndTransformVFS.....	115
EBISubscriptions.....	117
EBISubscriptionsList.....	117
EBISubscriptionsPending.....	118
EBISubscriptionsVFS.....	118
EBITransformation.....	119
EBITransformationAndPublish.....	120
EBITransformationMakeProfile.....	121
EBIUpdateAcquireApplication.....	121
EBIWaitUntilSubscriptionsGhatered.....	122
LockFiles.....	122
UnLockFiles.....	122
3.2.3Ficheros de Log.....	122
3.3Adaptador EDIWIN.....	123
3.3.1Configuración.....	123
3.3.2Variables disponibles.....	124
3.3.3Funciones.....	125
EBIBulkSubsAndImport.....	125
EBICfdiGetAsyncResult.....	125
EBICfdiProcessBlock.....	126
EBICfdiProcessBlockAsync.....	126
EBICfdiSignGetCfdiAndNom151.....	127
EBICreateEDIBinari.....	127
EBIEdiwinBulkImportl.....	128
EBIEdiwinChangeSituation.....	128
EBIEdiwinConnect.....	129
EBIEdiwinDeleteDocument.....	129
EBIEdiwinDeleteVolumeDocument.....	129
EBIEdiwinDeposit.....	130
EBIEdiwinImport.....	130
EBIEdiwinImportLib.....	130
EBIEdiwinMant.....	131
EBIEdiwinReceive.....	131
EBIEdiwinSyncroAddressBook.....	132
EBIEdiwinTratar.....	132
EBIExtractEDIBinari.....	132
EBIExportHeadersReport.....	133
EBIPublishAS2.....	133

EBIPublishEOS.....	135
EBIReceiveAS2.....	136
EBIRunComplement.....	136
EBIRunIECTE.....	136
EBISendAS2.....	137
EBISubsAndImport.....	137
EBISubsAndSendAS2.....	138
EBISubsAndSendEOS.....	139
EBISubsAndTransformAndBulkImportD.....	140
EBISubsAndTransformAndImport.....	141
EBISubsAndTransformAndSendAS2.....	142
EBISubsAndTransformAndSendEOS.....	143
EBITransformAndImport.....	144
getEBIRunComplementResult.....	145
3.3.4Ficheros de Log.....	145
3.4Adaptador SAP.....	146
3.4.1Arquitectura básica.....	146
EBI hacia SAP.....	146
SAP hacia EBI.....	148
3.4.2Arquitectura del adaptador SAP.....	149
3.4.3Instalación.....	150
3.4.4Configuración.....	151
3.4.5Creación del destino RFC.....	153
3.4.6Configuración de la puerta ALE.....	156
3.4.7Funciones.....	158
EBIImportToSAP.....	158
EBISubsAndImportToSAP.....	159
EBISubsAndTransformAndImportToSAP.....	159
3.4.8Ficheros de Log.....	161
4Administración EBI.....	163
4.1Introducción.....	163
4.2EBI Manager.....	163
4.2.1Como acceder al Panel de gestión EBI Manager.....	163
4.2.2Ámbito; Dominio y Aplicación.....	165
4.3Menú principal de Navegación.....	165
4.4Reglas.....	166
4.4.1Introducción.....	166
Menú.....	168
Propiedades de una regla.....	168
Propiedades de un Proceso.....	168
4.5Scripts.....	171
4.5.1Introducción.....	171
4.5.2Diálogo Scripts.....	171
Columnas.....	171
Opciones.....	172
4.5.3Insertar Script.....	172
Editar un script.....	173
Duplicar Script.....	174
4.6Procesos activos.....	174

4.6.1	Introducción.....	175
4.6.2	Diálogo Procesos activos.....	175
4.6.3	Columnas.....	175
4.6.4	Opciones.....	176
4.7	Trabajos.....	176
4.7.1	Introducción.....	176
4.7.2	Diálogo trabajos.....	176
	Columnas.....	176
	Opciones.....	177
4.7.3	Filtrar.....	177
4.8	Esquemas.....	179
4.8.1	Introducción.....	179
4.8.2	Diálogo Esquemas.....	179
	Columnas.....	180
	Opciones.....	180
4.8.3	Nuevo esquema.....	180
4.9	Publicaciones.....	181
4.9.1	Introducción.....	181
4.9.2	Diálogo Publicaciones.....	181
	Columnas.....	182
	Opciones.....	182
4.9.3	Nueva Publicación.....	182
4.10	Suscripciones.....	182
4.10.1	Introducción.....	183
4.10.2	Diálogo Suscripciones.....	183
	Columnas.....	183
	Opciones.....	183
4.10.3	Insertar nueva Suscripción.....	183
4.11	Mensajes.....	184
4.11.1	Introducción.....	184
4.11.2	Diálogo mensajes.....	184
	Columnas.....	185
	Opciones.....	186
4.11.3	Filtrar mensajes.....	187
4.12	Objetos.....	188
4.12.1	Introducción.....	188
4.12.2	Diálogo Objetos.....	188
	Columnas.....	188
4.13	Usuarios.....	189
4.13.1	Introducción.....	189
4.13.2	Diálogo Usuarios.....	189
	Columnas.....	189
	Opciones.....	190
4.13.3	Diálogo Usuario.....	190
	Pestaña Datos de usuario.....	190
	Pestaña Configuración.....	191
	Pestaña Conexiones.....	191
	Pestaña Planificaciones.....	191
	Pestaña Permisos.....	191

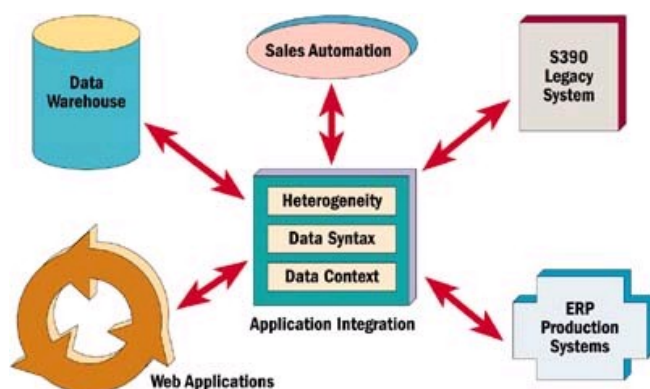
4.14 Alarmas.....	193
4.14.1 Introducción.....	193
4.14.2 Diálogo Alarmas.....	194
Columnas.....	194
Opciones.....	194
4.14.3 Configuración de Alarmas.....	194
Datos de la alarma.....	195
Configuración SMTP.....	195
Condiciones de disparo.....	196

1 ENTERPRISE APPLICATION INTEGRATION (EAI).

1.1 INTRODUCCIÓN

Muchas empresas se encuentran ante la necesidad de integrar diferentes sistemas informáticos y se enfrentan a la problemática de realizar dicha integración ocasionando el mínimo impacto en las aplicaciones existentes. Hasta ahora, la mayoría de las veces los sistemas eran sólo internos pero con el auge del B2B cada vez son más las situaciones en las que deben integrar sus sistemas internos con otros externos correspondientes a sus interlocutores comerciales. Uno de los objetivos más buscados en este tipo de integraciones es mantener la mayor independencia posible de cada uno de los aplicativos, es decir, se buscan integraciones donde las aplicaciones queden débilmente acopladas, de forma que se puedan hacer cambios en una aplicación sin necesidad de modificar el resto.

Para ayudar a cumplir este deseo aparentemente imposible ha surgido en los últimos años una nueva tecnología dentro de los sistemas de información llamada *Integración de Aplicaciones* (Application Integration). El concepto que subyace tras la Integración de Aplicaciones no es más que la aplicación sistemática de ciencias, metodologías y tecnologías en la tarea de integrar sistemas de información.



Necesidad de escalabilidad.

Las soluciones de integración punto a punto no pueden escalarse para las necesidades masivas de integración actuales.

Por ello las empresas están apostando por una infraestructura común de compartir información (EAI)

Una solución EAI tiene las siguientes misiones:

- Liberar a las aplicaciones de negocio de cualquier dependencia de la infraestructura de comunicaciones.
- Proporcionar una conversión de datos segura y fiable.
- Ofrecer escalabilidad sin pérdida de funcionalidad.

Se puede dividir la integración de aplicaciones en dos grandes “macro-problemas”:

- Integración de aplicaciones Intra-Empresa (EAI tradicional, Enterprise Application Integration)
- Integración de aplicaciones Inter-Empresa, cuando la integración de sistemas se realiza entre dos o más organizaciones diferentes (B2B Integration).

Las tecnologías que tradicionalmente se definían como EAI están evolucionando para contemplar la integración de sistemas B2B, de forma que un breve espacio de tiempo tan sólo se hablará del concepto genérico “Integración de Aplicaciones”.

1.2 DIFERENCIAS Y SIMILITUDES ENTRE EAI TRADICIONAL Y LA INTEGRACIÓN B2B

Aunque el EAI tradicional y la Integración B2B comparten muchos conceptos y tecnologías, cada uno de ellos tiene algunas características específicas. El concepto EAI normalmente está relacionado con la integración de aplicaciones y fuentes de datos de una empresa en vistas a resolver un problema local. Por ello, el EAI tradicional carece de características propias de la integración B2B como son: gestión de interlocutores, gestión de perfiles, sofisticados sistemas de seguridad y soporte exhaustivo de los estándares de B2B como EDI (Electronic Data Interchange), XML (Extensible Markup Language), X12, ODETTE, etc.

En contraste, la integración de aplicaciones B2B es una integración de sistemas entre diferentes organizaciones, que permite soportar cualquier requerimiento del negocio, tal como compartir la información con los socios comerciales de la cadena de suministro, o la colaboración en el diseño de un producto. Mientras que el software de integración B2B, proporciona muchas características de las que carecen herramientas EAI, tradicionalmente no han poseído las características necesarias para una integración profunda y no intrusiva con las aplicaciones empresariales que necesitan participar en los intercambios con los socios comerciales.

En la mayoría de las integraciones B2B, se debería aplicar como primer paso las soluciones EAI tradicionales. La lógica sugiere, que las aplicaciones internas deben estar integradas antes de que las podamos externalizar. Por ello, EAI y B2B son dos conceptos claramente integrados que se benefician el uno del otro.

1.3 NIVELES DE INTEGRACIÓN DE APLICACIONES

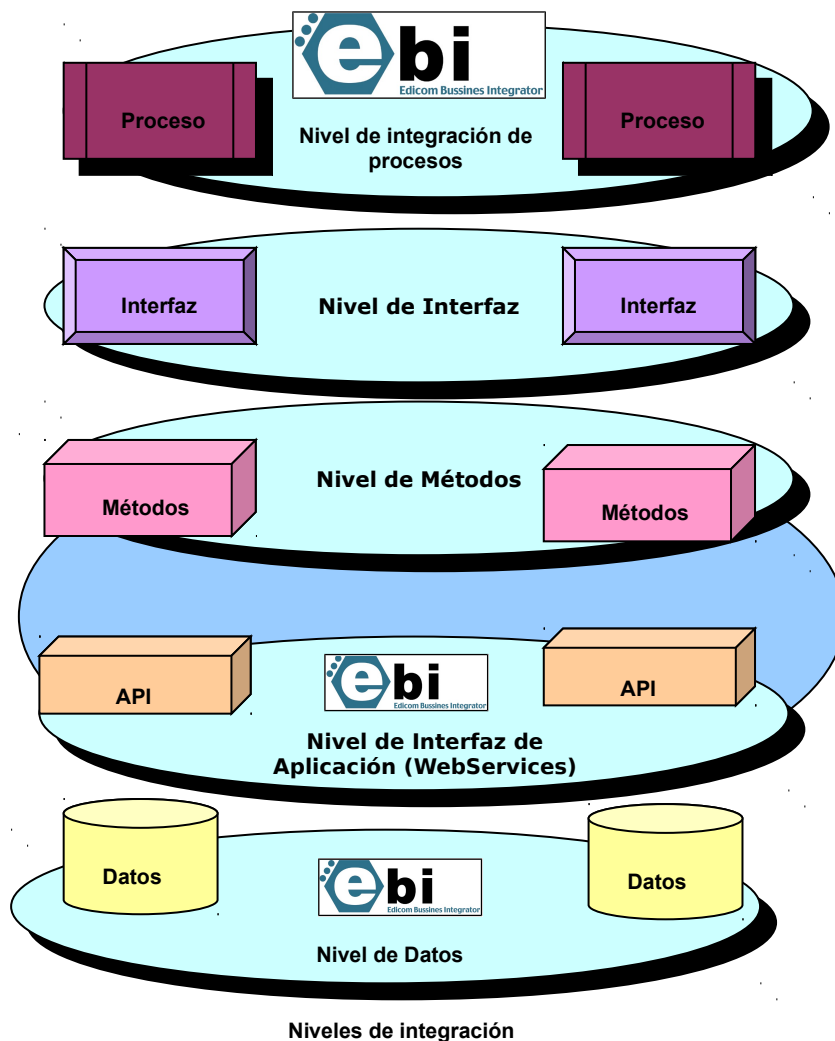
Para realizar correctamente una integración, las empresas deben conocer, tanto los procesos de negocio, como los datos implicados. En ese sentido, deben seleccionar qué procesos y qué datos requieren integración.

Se pueden distinguir distintos niveles en la integración de aplicaciones:

- Nivel de datos (Data level)
- Nivel de interfaz de aplicación, API (Application Interface Level)
- Nivel de métodos (Method Level)

- Nivel de interfaz de usuario (User Interface Level)
- Nivel de integración de procesos

El *Nivel de Datos* lo forman los procesos, técnicas y tecnologías aplicados en la extracción de información de una base de datos, transformarla si es necesario y



El Nivel de Datos lo forman los procesos, técnicas y tecnologías aplicados en la extracción de información de una base de datos, transformarla si es necesario y almacenarla en otra base de datos. El coste es la principal ventaja de este tipo de integraciones, ya que al no alterar la aplicación origen, no se incurrirá en grandes costes de desarrollar, probar y distribuir una nueva aplicación. Además, la tecnología que permite mover datos entre bases de datos y reformatearlos es relativamente económica comparada con otras tecnologías de integración de aplicaciones.

El Nivel de Interfaz de Aplicación se refiere a la disponibilidad de interfaces expuestos por las aplicaciones estándar empaquetadas o a medida. Estos interfaces permiten acceder tanto a los procesos de negocio como a los datos. Utilizando estos interfaces, los desarrolladores deberían ser capaces de integrar cualquier aplicación. La única limitación de esta estrategia serán las propias limitaciones de los interfaces expuestos por las aplicaciones.

Normalmente esta estrategia es la utilizada para integrar con aplicaciones empaquetadas tales como SAP, PEOPLESOFT y BAAN. El problema es cada una de ellas exponen los interfaces de una forma diferente, de manera que para intercambiar información se deben utilizar los interfaces de la aplicación origen para extraer la información, transformarla para que sea entendible por la aplicación destino y transmitirla a través de los interfaces específicos de esta última.

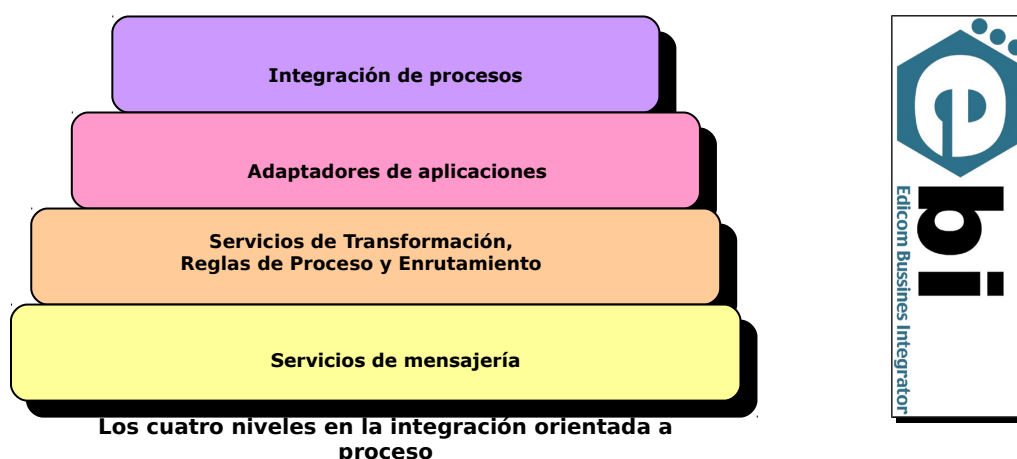
El Nivel de integración orientado a Métodos consiste en compartir métodos o procedimientos que implementan la lógica de negocio. Por ejemplo, se puede tener un método que sepa validar el crédito de un cliente, que sea compartido por diversas aplicaciones de dentro y fuera de la empresa. Estos métodos sólo se escribirían una vez y se podrían utilizar en muchas aplicaciones. Normalmente estos métodos compartidos suelen residir en servidores de aplicaciones o en monitores transaccionales. Mediante este tipo de integración se consiguen aplicaciones fuertemente acopladas.

El Nivel de Interfaz es un método de integración muy arcaico pero en ocasiones es el único aplicable. Consiste en dotar de un interfaz único a un conjunto de aplicaciones que no pueden ser integradas de ninguna otra forma. Un ejemplo de este tipo de integración es la orientada a Portal muy popular debido al auge de Internet. Gracias a este enfoque los arquitectos de aplicaciones pueden mostrar información procedente de varias aplicaciones locales o externas a través de un mismo interfaz de usuario (Web Browser).

El Nivel de Integración Orientado a Proceso es un sofisticado sistema de gestión que ofrece de forma centralizada un nivel de abstracción orientado a funciones de negocio (facturación, logística, etc.) que oculta por debajo diversos mecanismos de movimiento y transformación de datos tradicionales de más bajo nivel (middleware). La integración de Procesos define procesos comunes del negocio mediante secuencias, jerarquías, eventos, lógica de ejecución y movimiento de información entre aplicaciones que residen en la propia organización (EAI) y fuera de ella (B2B).

La integración de aplicaciones orientada a proceso es la meta final de la tecnología de integración de aplicaciones ya que no sólo permite compartir información con facilidad sino además permite compartir dicha información con herramientas visuales fáciles de usar.

Normalmente un sistema que soporte integración de aplicaciones orientada a proceso opera a cuatro niveles:



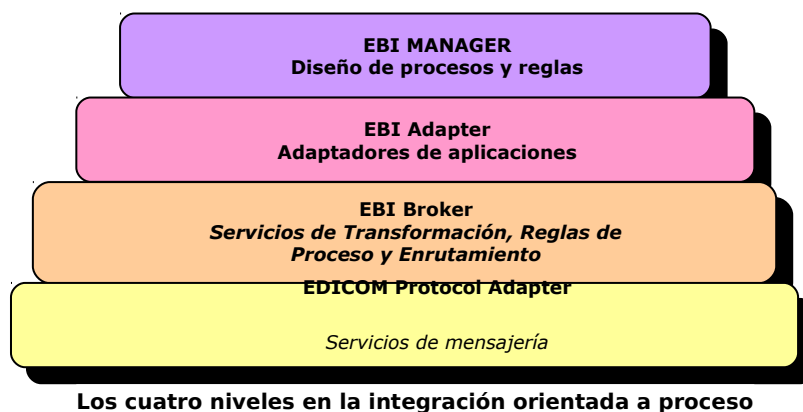
- El nivel de integración de procesos en sí mismo. Básicamente se definen procesos utilizando herramientas de modelización para programar el flujo de proceso que posteriormente ejecutan los motores correspondientes.

- En el nivel de adaptadores de aplicaciones se deben proveer de las máximas facilidades para conectar las aplicaciones existentes a la herramienta de integración general.
- El nivel de transformación, enrutamiento y reglas de proceso donde se realiza el formateo y el movimiento de la información a otras aplicaciones.
- El nivel de mensajería es el responsable de mover la información entre todas las aplicaciones conectadas. Normalmente se utiliza para esta labor un middleware orientado a mensaje con estándares tales como XML o EDI.

1.4 EDICOM BUSINESS INTEGRATOR (EBI): UNA HERRAMIENTA ORIENTADA A INTEGRACIÓN DE PROCESOS EMPRESARIALES

La visión de EDICOM de la problemática de la Integración de Aplicaciones es la de ofrecer un paquete de software (EDICOM Business Integrator, EBI) que cubra íntegramente las necesidades de una correcta integración a Nivel de Procesos Empresariales según la taxonomía expuesta anteriormente.

Por ello, EBI incluye herramientas que operan en los cuatro niveles exigidos:



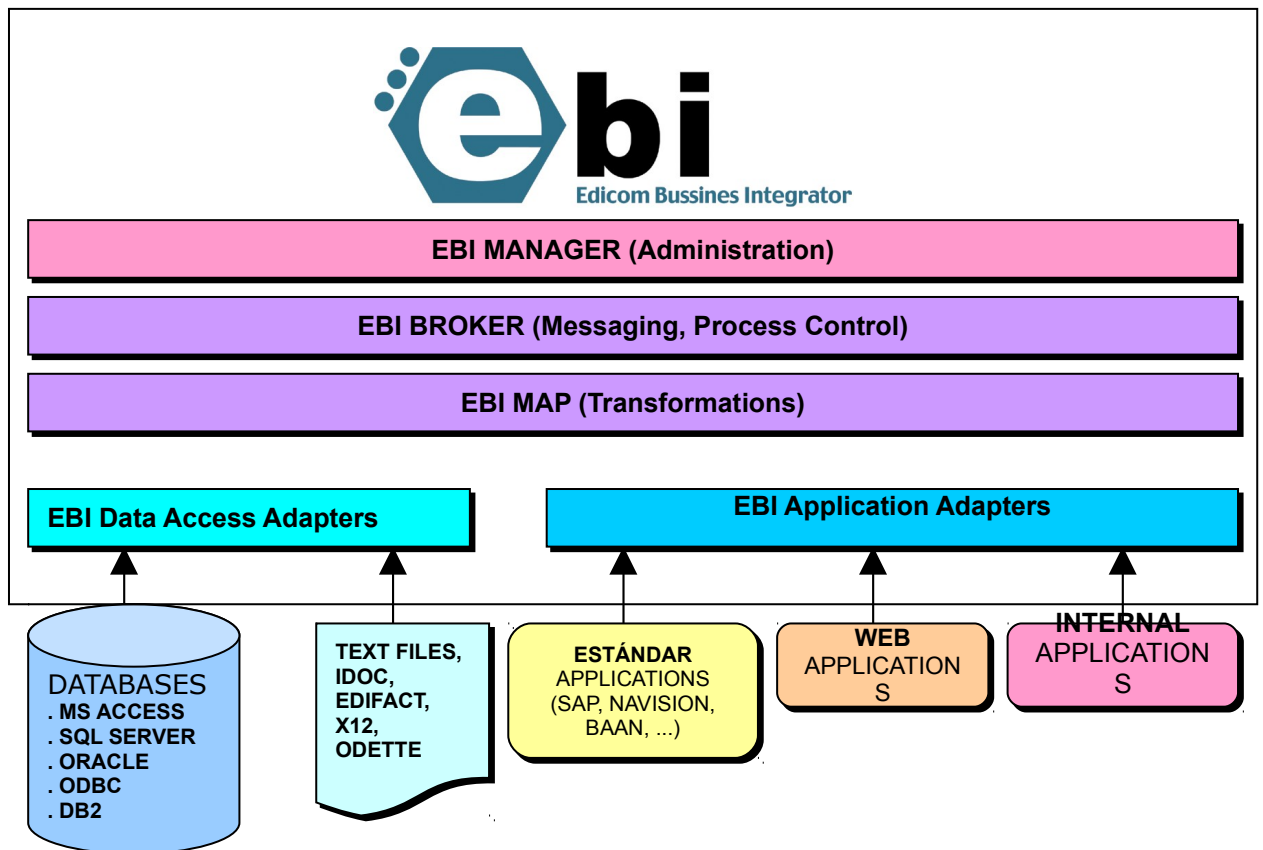
El nivel de integración de procesos se resuelve mediante la herramienta “EBI MANAGER”.

En el nivel de adaptadores de aplicaciones se proveen módulos que facilitan la conexión de las aplicaciones y fuentes de datos existentes con el EDICOM BUSINESS INTEGRATOR.

El nivel de transformación, enrutamiento y procesamiento de reglas de negocio es realizado por el “EBI Broker”

El nivel de mensajería se implementa en EBI mediante adaptadores de protocolos “EDICOM Protocol Adapter” gestionados por el EBI Broker. Estos adaptadores son los responsables de mover la información, entre todas las aplicaciones conectadas.

Normalmente se utiliza para esta labor mensajes con estándares tales como XML o EDI y protocolos como Web Services, HTTP, SMTP/POP, FTP, etc.

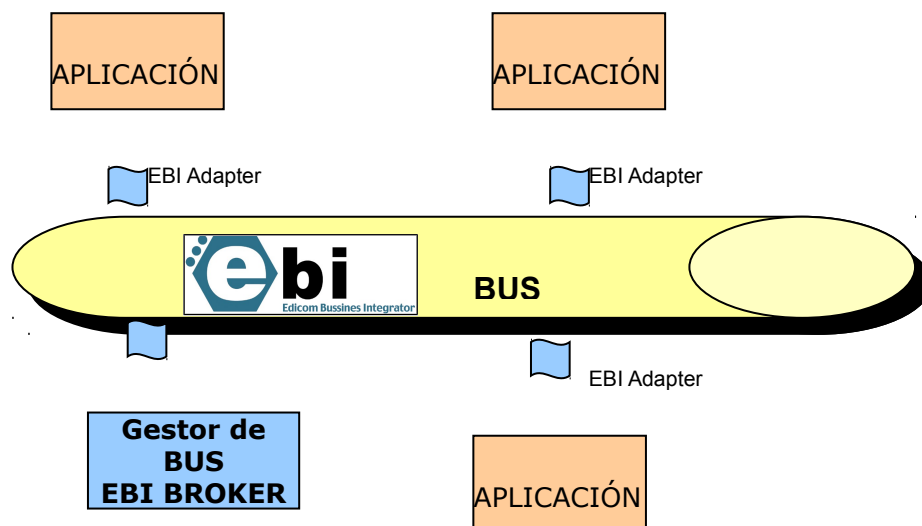


2 EL EBI BROKER

El EBI BROKER es el encargado de crear, controlar y gestionar el acceso al BUS de integración EAI por el cual fluirán los mensajes. A este BUS, es al que se conectarán los diferentes adaptadores, para que las aplicaciones sean capaces de publicar y anunciar eventos o cambios, y a su vez sean capaces de responder o leer los cambios realizados por otras aplicaciones del BUS.

Las tareas que debe asumir el EBI BROKER son las siguientes:

1. Realizar las transformaciones necesarias en los mensajes que publica la aplicación origen, para que estos puedan ser interpretados correctamente por las aplicaciones destino (esquemas, listas y transformaciones).
2. Enrutamiento de mensajes, para que los mensajes publicados lleguen a todas las aplicaciones suscritas a dichos mensajes (publicaciones y suscripciones).
3. Procesamiento de las reglas de negocio (reglas y scripts).
4. Gestión y mantenimiento del repositorio, en el que se depositarán de forma temporal los mensajes que viajan por el BUS.
5. Envío de alarmas.



ARQUITECTURA DE INTEGRACIÓN MEDIANTE BUS EN EBI

2.1 CONFIGURACIÓN

El EBI BROKER, es una aplicación Java que debe ejecutarse dentro de un servidor de aplicaciones, del estilo de Jboss, Tomcat, WebSphere, etc.

Para su funcionamiento, es necesario la versión de Java 1.6 o superior, y el servidor de aplicaciones debe definir la variable de contexto

```
edicom.path.conf
```

la cual indicará la ruta base donde, se encontrarán los ficheros de configuración del EBI BROKER.

Los parámetros de configuración del EBI BROKER se encuentran en el fichero ebibroker.properties, que se haya en la ruta

```
edicom.path.conf/ebibroker/ebibroker.properties
```

PROPIEDAD	DESCRIPCIÓN	VALOR POR DEFECTO
ebibroker.language	Idioma por defecto del EBI BROKER	es
ebibroker.country	País por defecto del EBI BROKER	ES
ebibroker.strongEncryptor	Clave que se empleará para encriptar los identificadores de sesión	
ebibroker.maxConcurrentSessions	Número máximo de sesiones concurrentes, por origen de datos. Si su valor es cero, no habrá restricción	0
ebibroker.maxThreadsConcurrentAccessControl	Número máximo de hilos estimados con el que se inicializa el ConcurrentHashMap que controlará las sesiones	16
ebibroker.enablesystemActivity	Habilita el registro de actividad del EBI BROKER	true
ebibroker.servicesenabled	Indica si están activas las tareas de mantenimiento	false
ebibroker.serviceslist	Orígenes de datos sobre los que se va realizar las tareas de mantenimiento	
ebibroker.whitelist	Lista de orígenes de datos, para los que esta permitido el acceso al EBI-BROKER	
ebibroker.whitelistRegExp	Indica si la lista <i>whitelist</i> , contiene expresiones regulares para su definición	false
ebibroker.publishBackup	Indica si se deben copiar las publicaciones en la tabla de BACKUP, para su almacenamiento posterior.	false
ebibroker.utc	Indica si las fechas se almacenan en la base de datos en formato UTC	false
ebibroker.defaultUserTimeZone	Zona horaria por defecto en la cual opera el EBI BROKER	Europe/Madrid
ebibroker.logsdir	Ruta donde se ubicarán los ficheros de sesión y de errores que genere el EBI BROKER	
ebibroker.logsFrom	Origen que enviará los mensajes de error	
ebibroker.logsTo	Destinatario de los mensajes de error	
ebibroker.logsSMTPHost	Servidor SMTP donde se enviarán los mensajes de error que se configuren en el fichero lo4j.xml	
ebibroker.remotesqlmaxrows	Número máximo de filas que devuelve la función EBIRemoteSQL	5000
ebibroker.autoregister	Indica si se activa el autoregistro de adaptadores	false
ebibroker.controlMultipleExecu	Se habilita que el EBI, controle la disminución de instancias,	false

tionProcess	en los procesos con múltiples instancias.	
ebibroker.enableRemoteRepository	Se habilita el repositorio remoto del ebiMapES, para almacenar los scripts y esquemas del EBI BROKER	false
ebibroker.processTimeStampTimeout	Milisegundos para considerar un proceso de EBI muerto	150000
ebibroker.profiles	Habilita la cache de métodos en los servicios Habilitada = cache-enabled Deshabilitada = vacío	
ebistats.consolidationDay	Se consolidan las estadística de mas de X días, eliminando el detalle de la hora .	0 deshabilitada
ebistats.consolidationMonth	Se consolidan las estadística de mas de X meses eliminando el detalle del día y hora	0 deshabilitada
ebibroker.sysuser	Usuario de sistema del EBI BROKER	
ebibroker.syspassword	Contraseña codificada en MD5 del usuario de sistema	
ebibroker.anonymoususer	Usuario anónimo del EBI BROKER	
ebibroker.anonymouspassword	Contraseña codificada en MD5 del usuario anónimo	
ebimap.address	URL de conexión al servicio de ebiMapES	
ebirepository.address	URL de conexión al repositorio de almacenamiento de scripts, transformaciones y esquemas	
ebiftpserver.enabled	Habilita el servicio de FTP/S de acceso EBI BROKER	true
ebiftpserver.port	Puerto de escucha del servidor FTP/S	21
ebiftpserver.keystoreFile	Ruta donde se encuentra el almacén de certificados para realizar las conexiones FTPS	
ebiftpserver.sslProtocol	Protocolo SSL a utilizar para las conexiones FTPS	TLS
ebiftpserver.clientAuthentication	Se debe realizar autenticación de cliente. Valores permitidos NEED, WANT y NONE	NONE
ebiftpserver.keyPassword	Contraseña del certificado se que empleará en la conexión FTPS	
ebiftpserver.keystorePassword	Contraseña del almacén de certificados	
ebiftpserver.implicitSsl	True si el servicio de utilizar SSL implicito	false
ebiftpserver.idleTimeout	Número de segundos, antes que el cliente sea desactivado por inactividad. Si el valor es 0, esta propiedad estará desactivada.	300
ebiftpserver.passivePorts	Rango de puertos del modo FTP pasivo	
ebiftpserver.passiveExternalAddress	Dirección, en la cual el servidor pedirá ser escuchado como respuesta al modo pasivo. Esta propiedad será útil, cuando el servicio este detrás de un firewall	
ebisshserver.enabled	Habilita el servicio de SFTP de acceso al EBI BROKER	true
ebisshserver.port	Puerto de escucha del servidor SFTP	
ebisshserver.filekeypair	Fichero con la clave que se empleará para cifrar la conexiones SFTP	
ebisshserver.filekeypairpassword	Contraseña para acceder a la clave	
ebimanagerws.securityaccess	Utilizando lenguaje SpEL (Spring Expression Language), se pueden indicar las restricciones de acceso del cliente	

	webServices de administración del EBI BROKER. Posibles valores son permitAll: permite el acceso a todo el mundo hasIpAddress('192.168.0.0/16') or hasIpAddress('172.16.0.0/16'): solo permite el acceso desde la IP's 192.168 y 172.16	
hibernate.cache.use_second_level_cache	Habilita las caches de segundo nivel de hibernate	false
hibernate.cache.use_structured_entries	Fuerza a hibernate a guardar la caches en formato estructurado para que sean mas legibles	false
hibernate.cache.use_query_cache	Habilita las caches de las consultas	true
hibernate.generate_statistics	Habilita las estadísticas de hibernate para realizar tuning	false
hibernatesearch.enabled	Esta opción no esta operativa así que su valor debe ser false	false
ebibroker.repositoryid	Identificador del repositorio en el cual se encuentran los objetos del EBI BROKER	
ebibroker.applicationid	Identificador de la aplicación en la cual se encuentran los objetos del EBI BROKER	
ebibroker.subscriptionsToBlock	Número de intentos de suscripción de un mensaje, antes de pasar a estado bloqueado.	5
ebibroker.logsdeleted	Número de horas de permanencia de los logs en el EBI BROKER.	0 desactivado
ebibroker.documentdeleted	Número de minutos de permanencia de los documentos en estado final, no rechazados, en el EBI BROKER	0 desactivado
ebibroker.documentrejecteddeleted	Número de minutos de permanencia de los documentos en estado rechazado, en el EBI BROKER	0 desactivado
ebibroker.aliasedidb	Alias de la cadena de conexión de EDIWIN, que se pasará al servicio de transformación EBIMAPES	
ebibroker.useredidb	Usuario de conexión, a la base de datos de EDIWIN, que se pasará al servicio de transformación EBIMAPES	
ebibroker.passwordedidb	Contraseña de conexión, a la base de datos de EDIWIN, que se pasará al servicio de transformación EBIMAPES	
ebibroker.connectionedidb	Cadena de conexión, a la base de datos de EDWIN, que se pasará al servicio de transformación EBIMAPES	
ebibroker.connectiondb	Cadena de conexión en formato JDBC, a la base de datos del EBI BROKER	
ebibroker.userdb	Usuario de conexión a la base de datos del EBI BROKER	
ebibroker.passworddb	Contraseña de conexión, a la base de datos del EBI BROKER	
ebibroker.driverdb	Driver JDBC, que se empleara para la conexión a la base de datos. SQLSERVER=com.microsoft.sqlserver.jdbc.SQLServerDriver MYSQL=com.mysql.jdbc.Driver ORACLE=oracle.jdbc.OracleDriver	
ebibroker.dialectdb	Dialecto SQL que debe emplear hiberante para conectarse a la base de datos SQLSERVER=org.hibernate.dialect.SQLServer2005Dialect MYSQL=org.hibernate.dialect.MySQLInnoDBDialect	

	ORACLE=org.hibernate.dialect.Oracle10gDialect	
--	---	--

2.2 SOPORTE DE BALANCEO DE CARGA

El software del EBI BROKER, esta diseñado para que soporte entornos de balanceo de carga. Un entorno de balanceo de carga consiste en tener un dispositivo, hardware o software, que distribuye la carga entre los distintos EBI BROKER que estén dados de alta en el dispositivo de balanceo.

Desde el punto de vista de los adaptadores, solo existirá un único EBI BROKER, y las peticiones que estos hagan, se realizarán contra la dirección de balanceo.

Si no se dispone del dispositivo, hardware o software, que distribuye la carga entre los distintos EBI BROKERS, existe la posibilidad de indicar a los distintos adaptadores, una lista con la IP's en las cuales está instalado el servicio del EBI BROKER, realizando los adaptadores un balanceo interno por software, a los distintos EBI BROKER que tienen en su configuración.

2.3 ACCESO VÍA WEBSERVICES

Existén dos API webServices para acceder al EBI BROKER, el *EBIBrokerWS*, y el webServices *EBIBrokerWSV2*

2.3.1 EBIBROKERWS

El webService EBIBrokerWS, permite el acceso al EBIBROKER, poder realizar operaciones envío y recepción de mensajes.

La wsdl de acceso al servicio se encuentra disponible en

<http://xxx.yvy.zz/services/EBIBrokerWS?wsdl>

Funciones del webservice

Función	Descripción	ver.
TEBIBrokerInterface PublishWS	Esta función permite publicar mensajes en el EBI-BROKER vía web service, utilizando SOAP Attachment, para enviar los fichero a publicar	
TEBIBrokerInterface_ConfirmedPublicationsWS	Devuelve las publicaciones que han sido recuperadas por el destinatario	
TEBIBrokerInterface_NotifiedPublicationWS	Emite un acuse de la publicación recuperada por el destinatario	
TEBIBrokerInterface_RejectedPublicationsWS	Devuelve las publicaciones que han sido rechazadas por el destinatario	
TEBIBrokerInterface_NotifiedRejectedPublicationWS	Emite un acuse de la publicación rechazada	
TEBIBrokerInterface_SubscriptionWS	Permite recuperar suscripciones,	

	aplicándoles o no una transformación	
TEBIBrokerInterface_ConfirmSubscriptionWS	Confirma de forma positiva la entrega de los mensajes que se han recuperado	
TEBIBrokerInterface_RejectSubscriptionWS	Rechaza la suscripción recogida del sistema (es una confirmación negativa).	

TEBIBrokerInterface_PublishWS

Esta función permitirá publicar mensajes en el EBIBROKER

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ueb="UEBIBrokerInterface">
  <soapenv:Header/>
  <soapenv:Body>
    <ueb:TEBIBrokerInterface_PublishWS>
      <aClientId>?</aClientId>
      <aUser>?</aUser>
      <aPassword>?</aPassword>
      <aDomain>?</aDomain>
      <aApplication>?</aApplication>
      <aSchema>?</aSchema>
      <aDestination>?</aDestination>
      <aReference>?</aReference>
      <aMessage>?</aMessage>
      <aMessageFormat>?</aMessageFormat>
      <aDuplicates>?</aDuplicates>
    </ueb:TEBIBrokerInterface_PublishWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros:

aClientId	identificador del cliente
aUser	usuario que realiza la publicación
aPassword	contraseña del usuario que realiza la publicación
aDomain	dominio donde se va a publicar el documento
aApplication	aplicación en la que se quiere publicar el documento
aSchema	nombre del esquema con el que se publicará el documento
aDestination	destino de la publicación del documento
aReference	referencia con la que se publicará el documento
aMessage	Content-ID del adjunto SOAP con el mensaje, o mensaje codificado en base 64.
aMessageFormat	Indica el formato con el que se esta enviado el mensaje: 1: Codificado en base64 en la variable aMessage. 10: Como SOAP Attachement. Content-Type: application/x-gzip Content-Transfer-Encoding: base64
aDuplicates	indica que acción se debe tomar ante documentos duplicados 0: dupIgnore. La publicación siempre se genera, incluso si ya existe una ya duplicada en el sistema. 1: dupAccept. En caso de ya existir una publicación anterior similar (con el mismo HASH), no se genera la nueva publicación, ni sus suscripciones correspondientes. 2: dupError. En caso de ya existir una publicación anterior devuelve un código de error 3: dupOverride. En caso de ya existir una publicación, esta y sus suscripciones se borran, y se crea una nueva con el mensaje que se esta publicando.

Por publicación duplicada, se entiende, aquellos documentos con el mismo HASH, dentro del mismo dominio y aplicación.

Si la respuesta satisfactoria: Identificador de la publicación que se ha realizado en el EBIBROKER

Si se produce un error, se generará un código de error, que indicará el tipo de error que se ha producido.

Código	Error
-1	Error general durante la publicación
-2	Publicación no autorizada
-3	Aplicación destino no encontrada
-4	Documento incorrecto acorde a la validación sintáctica del esquema
-5	La aplicación destino no acepta la suscripción
-7	Usuario no autorizado a la suscripción
-8	Publicación duplicada
-9	El dominio no permite publicaciones BROADCAST
-10	El documento no se ha insertado correctamente en la base de datos
-11	Problema generando las suscripciones
-12	Error registrando la sesión
-16	Tamaño de la publicación excedido

TEBIBrokerInterface_ConfirmedPublicationsWS

Recupera las publicaciones que han sido recuperadas por el destinatario

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ueb="UEBIBrokerInterface">
  <soapenv:Header/>
  <soapenv:Body>
    <ueb:TEBIBrokerInterface_ConfirmedPublicationsWS>
      <aClientId?></aClientId>
      <aUser?></aUser>
      <aPassword?></aPassword>
      <aDomain?></aDomain>
      <aApplication?></aApplication>
    </ueb:TEBIBrokerInterface_ConfirmedPublicationsWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros

aClientId	Identificador del cliente
aUser	usuario que realiza la petición
aPassword	contraseña del usuario que realiza la petición
aDomain	dominio en el cual el documento fue publicado
aApplication	aplicación en la que la cual el documento fue publicado

Si la respuesta es satisfactoria se obtendrá la lista con los identificadores de los mensajes que han sido satisfactoriamente recuperados.

Si produce un error, se devolverá una excepción del tipo `EBIBrokerServerException` la cual indicará el código y el motivo del error.

Código	Error
-1	Error general durante la publicación
-12	Usuario no válido

TEBIBrokerInterface_NotifiedPublicationWS

Emite un acuse de entrega de la confirmación de la publicación

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ueb="UEBIBrokerInterface">
  <soapenv:Header/>
  <soapenv:Body>
    <ueb:TEBIBrokerInterface_NotifiedRejectedPublicationWS>
      <aClientId?></aClientId>
      <aUser?></aUser>
      <aPassword?></aPassword>
      <aDomain?></aDomain>
      <aApplication?></aApplication>
      <aMessageIds>
        <!--Zero or more repetitions:-->
        <item?></item>
      </aMessageIds>
    </ueb:TEBIBrokerInterface_NotifiedRejectedPublicationWS>
  </soapenv:Body>
</soapenv:Envelope>
```

aClientId	Identificador del cliente
aUser	usuario que realiza la petición
aPassword	contraseña del usuario que realiza la petición
aDomain	dominio donde se encuentra la publicación confirmada
aApplication	aplicación donde se encuentra la publicación confirmada
aMessageIds	lista con los identificadores de la publicaciones confirmadas, a emitir su acuse de entrega

True si la operación se realiza con éxito

False si se produce un error, o se devolverá una excepción del tipo `EBIBrokerServerException` la cual indicará el código y el motivo del error.

Código	Error
-1	Error general durante la operación
-12	Usuario no válido

TEBIBrokerInterface_RejectedPublicationsWS

Recupera las publicaciones que han sido rechazadas por el destinatario

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ueb="UEBIBrokerInterface">
  <soapenv:Header/>
  <soapenv:Body>
    <ueb:TEBIBrokerInterface_RejectedPublicationsWS>
      <aClientId?</aClientId>
      <aUser?</aUser>
      <aPassword?</aPassword>
      <aDomain?</aDomain>
      <aApplication?</aApplication>
    </ueb:TEBIBrokerInterface_RejectedPublicationsWS>
  </soapenv:Body>
</soapenv:Envelope>
```

aClientId	Identificador del cliente
aUser	usuario que realiza la petición
aPassword	contraseña del usuario que realiza la petición
aDomain	dominio en el cual el documento fue publicado
aApplication	aplicación en la que la cual el documento fue publicado

Si la respuesta es satisfactoria, se obtendrá la lista con los identificadores de los mensajes que han sido rechazados.

Si produce un error, se devolverá una excepción del tipo `EBIBrokerServerException` la cual indicará el código y el motivo del error.

Código	Error
-1	<i>Error general durante la operación</i>
-12	Usuario no válido

TEBIBrokerInterface_NotifiedRejectedPublicationWS

Emite un acuse de entrega del rechazo de la publicación

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ueb="UEBIBrokerInterface">
  <soapenv:Header/>
  <soapenv:Body>
    <ueb:TEBIBrokerInterface_NotifiedRejectedPublicationWS>
      <aClientId?</aClientId>
      <aUser?</aUser>
      <aPassword?</aPassword>
      <aDomain?</aDomain>
      <aApplication?</aApplication>
      <aMessageIds>
        <item?</item>
      </aMessageIds>
    </ueb:TEBIBrokerInterface_NotifiedRejectedPublicationWS>
  </soapenv:Body>
</soapenv:Envelope>
```

aClientId	Identificador del cliente
aUser	usuario que realiza la petición

aPassword	contraseña del usuario que realiza la petición
aDomain	dominio donde se encuentra la publicación rechazada
aApplication	aplicación donde se encuentra la publicación rechazada
aMessageIds	lista con los identificadores de la publicaciones rechazadas, a emitir su acuse de entrega

True si la operación se realiza con éxito

False si se produce un error, o se devolverá una excepción del tipo `EBIBrokerServerException` la cual indicará el código y el motivo del error.

Código	Error
-1	<i>Error general durante la publicación</i>
-12	Usuario no válido

TEBIBrokerInterface_SubscriptionWS

Recoge los mensajes que estén en estado suscrito

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ueb="UEBIBrokerInterface">
  <soapenv:Header/>
  <soapenv:Body>
    <ueb:TEBIBrokerInterface_SubscriptionWS>
      <aClientId>?</aClientId>
      <aUser>?</aUser>
      <aPassword>?</aPassword>
      <aDomain>?</aDomain>
      <aApplication>?</aApplication>
      <aSchema>?</aSchema>
      <aTransformation>?</aTransformation>
      <aReference>?</aReference>
      <aMessage>?</aMessage>
      <aMessageFormat>?</aMessageFormat>
    </ueb:TEBIBrokerInterface_SubscriptionWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros:

aClientId	identificador del cliente
aUser	usuario que realiza la suscripción
aPassword	contraseña del usuario que realiza la suscripción
aDomain	dominio donde se encuentra la suscripción a recoger
aApplication	aplicación donde se encuentra la suscripción
aSchema	esquema del cual se quiere recoger la suscripción
aTransformation	transformación que se desea aplicar a la suscripción, antes de recogerla. Este valor es opcional, sino se indica transformación se recogerá la suscripción tal cual se haya publicado.
aReference	referencia del mensaje que se quiere recuperar, sino se indica ninguna, se recogerá la primera suscripción que este disponible.

aMessage	este parámetro es de salida, y en el caso que el parámetro aMessageFormat se igual a 1, viene la suscripción o el resultado de la transformación (solo el primer fichero) codificado en base 64,
aMessageFormat	indica el formato en el que la respuesta será devuelta al cliente
	1: Codificada en base 64 en la variable aMessage. Si se aplica transformación, solo se devolverá el primer documento resultado de la transformación.
	10: Como adjuntos a la respuesta (SOAP Attachment)

Si la respuesta es satisfactoria, devolverá como respuesta, el identificador de la suscripción recuperada, así como la suscripción propiamente dicha, bien como adjuntos (SOAP Attachments) o bien codificada en base 64 en la variable aMessage. Si no hay suscripciones, pendiente de recoger, devolverá un identificador vacío.

Si se produce algún error devolverá el código de error, indicando el motivo.

Código	Error
-1	Error general durante la suscripción
-2	Publicación no autorizada a recoger la suscripción
-3	Transformación no encontrada
-4	Error realizando la transformación
-7	Usuario no autorizado a la suscripción
-12	Problema generando las suscripciones

TEBIBrokerInterface_ConfirmSubscriptionWS

Confirma de forma positiva la entrega de las suscripciones recogidas.

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ueb="UEBIBrokerInterface">
  <soapenv:Header/>
  <soapenv:Body>
    <ueb:TEBIBrokerInterface_ConfirmSubscriptionWS>
      <aClientId?</aClientId>
      <aUser?</aUser>
      <aPassword?</aPassword>
      <aDomain?</aDomain>
      <aApplication?</aApplication>
      <aMessageID?</aMessageID>
    </ueb:TEBIBrokerInterface_ConfirmSubscriptionWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros:

aClientId	identificador del cliente
aUser	usuario que realiza la confirmación
aPassword	contraseña del usuario que realiza la confirmación
aDomain	dominio donde se encuentra la suscripción a confirmar
aApplication	aplicación donde se encuentra la suscripción a confirmar.

aMessageID	identificador del mensaje que se quiere confirmar. Si se desean confirmar varios mensajes, se indicarán los identificadores de mensaje separados por ' '
-------------------	--

Si la respuesta es satisfactoria, devolverá 1

Si se produce algún error devolverá el código de error, indicando el motivo.

Código	Error
-1	Error general durante la confirmación
-2	Suscripción no encontrada
-3	Estado del documento, no valido para la confirmación. Los documentos, solo pueden ser confirmados, cuando se encuentran en estado suscrito, pendiente de confirmación o bloqueado
-4	Documento ya en estado confirmado
-12	Error registrando la sesión

TEBIBrokerInterface_RejectSubscriptionWS

Rechaza la suscripción recogida del sistema (es una confirmación negativa).

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ueb="UEBIBrokerInterface">
  <soapenv:Header/>
  <soapenv:Body>
    <ueb:TEBIBrokerInterface_RejectSubscriptionWS>
      <aClientId?</aClientId>
      <aUser?</aUser>
      <aPassword?</aPassword>
      <aDomain?</aDomain>
      <aApplication?</aApplication>
      <aMessageID?</aMessageID>
      <aInfo?</aInfo>
    </ueb:TEBIBrokerInterface_RejectSubscriptionWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros:

aClientId	identificador del cliente
aUser	usuario que realiza el rechazo
aPassword	contraseña del usuario que realiza el rechazo
aDomain	dominio donde se encuentra la suscripción a rechazar
aApplication	aplicación donde se encuentra la suscripción a rechazar.
aMessageID	identificador del mensaje que se quiere rechazar. Si se desean rechazar varios mensajes, se indicarán los identificadores de mensaje separados por ' '.
aInfo	Información adicional del motivo del rechazo.

Si la respuesta es satisfactoria, devolverá 1

Si se produce algún error devolverá el código de error, indicando el motivo.

Código	Error
--------	-------

-1	Error general durante el proceso de rechazo
-2	Suscripción no encontrada
-3	Estado del documento, no valido para ser rechazado. Los documentos, solo pueden ser rechazados, cuando se encuentran en estado suscrito, pendiente de confirmación o bloqueado
-4	Documento ya en estado rechazado
-12	Error registrando la sesión

2.3.2 EBIBROKERWSV2

El webService EBIBrokerWSV2, permite el acceso al EBIBROKER, poder realizar operaciones envío y recepción de mensajes.

La wsdl de acceso al servicio se encuentra disponible en

<http://xxx.yyy.zz/services/EBIBrokerWSV2?wsdl>

Funciones del webservice

Función	Descripción	ver.
publishWS	Esta función permite publicar mensajes en el EBI-BROKER vía web service, utilizando SOAP Attachment, para enviar los fichero a publicar	
transformAndPublishWS	Esta función publica mensajes en el EBIBROKER, aplicándoles previamente la transformación que se indica como parámetro. La función devolverá el resultado de la transformación	
confirmedPublicationsWS	Devuelve las publicaciones que han sido recuperadas por el destinatario	
notifiedPublicationWS	Emite un acuse de la publicación recuperada por el destinatario	
rejectedPublicationsWS	Devuelve las publicaciones que han sido rechazadas por el destinatario	
notifiedRejectedPublicationWS	Emite un acuse de la publicación rechazada	
subscriptionWS	Permite recuperar suscripciones, aplicándoles o no una transformación	
confirmSubscriptionWS	Confirma de forma positiva la entrega de los mensajes que se han recuperado	
rejectSubscriptionWS	Rechaza la suscripción recogida del sistema (es una confirmación negativa).	

TipoS complejos que devolverán alguna de las funciones webServices del servicio EBIBrokerWSV2

- **EBIDocumentWS**, representa un mensaje en el EBIBROKER

```
<xs:complexType name="ebiDocumentWS">
  <xs:sequence>
    <xs:element minOccurs="0" name="application" type="xs:string"/>
    <xs:element minOccurs="0" name="applicationdst" type="xs:string"/>
    <xs:element minOccurs="0" name="applicationsub" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:element minOccurs="0" name="countsub" type="xs:int"/>
<xs:element minOccurs="0" name="date" type="xs:dateTime"/>
<xs:element minOccurs="0" name="domain" type="xs:string"/>
<xs:element minOccurs="0" name="id" type="xs:string"/>
<xs:element minOccurs="0" name="idsub" type="xs:string"/>
<xs:element minOccurs="0" name="message" type="xs:string"/>
<xs:element maxOccurs="unbounded" minOccurs="0" name="messages"
nillable="true" type="tns:ebiMessageServerWS"/>
<xs:element minOccurs="0" name="reference" type="xs:string"/>
<xs:element minOccurs="0" name="size" type="xs:int"/>
<xs:element minOccurs="0" name="type" type="tns:ebiDocumentTypeWS"/>
</xs:sequence>
</xs:complexType>

```

- **EBIMessageServerWS**, representa el contenido de un mensaje. Un mismo mensaje, puede tener varias repeticiones de esta estructura.

```

<xs:complexType name="ebiMessageServerWS">
  <xs:sequence>
    <xs:element minOccurs="0" name="message" type="xs:base64Binary"/>
    <xs:element minOccurs="0" name="name" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

- **EBIDocumentTypeWS**, contiene un enumerado, con los diferentes estados de un documento.

```

<xs:simpleType name="ebiDocumentTypeWS">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PUBLISHED"/>
    <xs:enumeration value="SUBSCRIPTIONED"/>
    <xs:enumeration value="CONFIRMED"/>
    <xs:enumeration value="BLOCKED"/>
    <xs:enumeration value="PUBLISHPARTLYCONFIRMED"/>
    <xs:enumeration value="PUBLISHCONFIRMED"/>
    <xs:enumeration value="PUBLISHNOTIFIED"/>
    <xs:enumeration value="PENDINGCONFIRMATION"/>
    <xs:enumeration value="INEBIMAPFORMAT"/>
    <xs:enumeration value="REJECTED"/>
    <xs:enumeration value="PUBLISHREJECTED"/>
    <xs:enumeration value="PUBLISHREJECTEDNOTIFIED"/>
    <xs:enumeration value="PUBLISHPARTLYREJECTED"/>
    <xs:enumeration value="PUBLISHDUPLICATED"/>
    <xs:enumeration value="DELETED"/>
    <xs:enumeration value="DELETEDONLYDOC"/>
    <xs:enumeration value="COPIED"/>
    <xs:enumeration value="NOCOPIED"/>
    <xs:enumeration value="UNKNOWN"/>
  </xs:restriction>
</xs:simpleType>

```

publishWS

Esta función permitirá publicar mensajes en el EBIBROKER

Llamada:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:publishWS>
      <clientId>?</clientId>
      <user>?</user>
      <password>?</password>
      <domain>?</domain>
    </com:publishWS>
  </soapenv:Body>
</soapenv:Envelope>

```

```

<application>?</application>
<schema>?</schema>
<destination>?</destination>
<reference>?</reference>
<messages>
  <message>cid:272360061242</message>
  <name>?</name>
</messages>
<duplicates>?</duplicates>
</com:publishWS>
</soapenv:Body>
</soapenv:Envelope>

```

Parámetros:

clientId	identificador del cliente
user	usuario que realiza la publicación
password	contraseña del usuario que realiza la publicación
domain	dominio donde se va a publicar el documento
application	aplicación en la que se quiere publicar el documento
schema	nombre del esquema con el que se publicará el documento
destination	destino de la publicación del documento
reference	referencia con la que se publicará el documento
messages	Esta estructura contendrá los ficheros que representarán la publicación
message	contenido del fichero, bien el content-id del adjunto SOAP, bien el contenido codificado en base 64.
name	nombre del fichero
duplicates	Indica que acción se debe tomar ante documentos duplicados
	0: duplgnore. La publicación siempre se genera, incluso si ya existe una ya duplicada en el sistema.
	1: dupAccept. En caso de ya existir una publicación anterior similar (con el mismo HASH), no se genera la nueva publicación, ni sus suscripciones correspondientes.
	2: dupError. En caso de ya existir una publicación anterior devuelve un código de error
	3: dupOverride. En caso de ya existir una publicación, esta y sus suscripciones se borran, y se crea una nueva con el mensaje que se esta publicando.

Por publicación duplicada, se entiende, aquellos documentos con el mismo HASH, dentro del mismo dominio y aplicación.

Si la respuesta es satisfactoria, se obtendrá el identificador de la publicación que se ha realizado en el EBIBROKER

Si se produce un error, se generará un código de error, que indicará el tipo de error que se ha producido.

Código	Error
-1	Error general durante la publicación
-2	Publicación no autorizada
-3	Aplicación destino no encontrada
-4	Documento incorrecto acorde a la validación sintáctica del esquema
-5	La aplicación destino no acepta la suscripción
-7	Usuario no autorizado a la suscripción
-8	Publicación duplicada

-9	El dominio no permite publicaciones BROADCAST
-10	El documento no se ha insertado correctamente en la base de datos
-11	Problema generando las suscripciones
-12	Error registrando la sesión
-16	Tamaño de la publicación excedido

transformAnPublishWS

Esta función permitirá publicar mensajes en el EBIBROKER, aplicando la transformación que se indica como parámetro, previo a la publicación del mensaje.

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:transformAndPublishWS>
      <clientId?></clientId>
      <user?></user>
      <password?></password>
      <domain?></domain>
      <application?></application>
      <transformation?></transformation>
      <!--Zero or more repetitions:-->
      <profile>
        <key?></key>
        <value?></value>
      </profile>
      <filter?></filter>
      <schema?></schema>
      <destination?></destination>
      <reference?></reference>
      <!--Zero or more repetitions:-->
      <messages>
        <message>cid:1052811169699</message>
        <name?></name>
      </messages>
      <duplicates?></duplicates>
      <mask?></mask>
    </com:transformAndPublishWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros:

clientId	identificador del cliente
user	usuario que realiza la transformación y publicación
password	contraseña del usuario que realiza la transformación y publicación
domain	dominio donde se va a publicar el documento
application	aplicación en la que se quiere publicar el documento
transformation	transformación que se va a aplicar a los documentos enviados, previo a su publicación.
profile	en el perfil de ejecución, se puede pasar parámetros, para la ejecución de la transformación
key	clave del parámetro
value	valor del parámetro
schema	nombre del esquema con el que se publicará el documento
destination	destino de la publicación del documento
reference	referencia con la que se publicará el documento

messages	Esta estructura contendrá los ficheros sobre los que se va aplicar la transformación
message	contenido del fichero, bien el content-id del adjunto SOAP, bien el contenido codificado en base 64.
name	nombre del fichero
duplicates	Indica que acción se debe tomar ante documentos duplicados
	0: dupIgnore. La publicación siempre se genera, incluso si ya existe una ya duplicada en el sistema.
	1: dupAccept. En caso de ya existir una publicación anterior similar (con el mismo HASH), no se genera la nueva publicación, ni sus suscripciones correspondientes.
	2: dupError. En caso de ya existir una publicación anterior devuelve un código de error
	3: dupOverride. En caso de ya existir una publicación, esta y sus suscripciones se borran, y se crea una nueva con el mensaje que se esta publicando.
mask	máscara de que se aplicará a los ficheros resultado de la transformación, y cuyo resultado serán los que se publiquen en el EBIBROKER.

Por publicación duplicada, se entiende, aquellos documentos con el mismo HASH, dentro del mismo dominio y aplicación.

Si la respuesta es satisfactoria devolverá los ficheros resultado de la transformación que se va publicar. Estos ficheros de devolverán una lista contiene la estructura **EBIMessageServerWS**.

Si se produce un error, se generará una excepción que contendrá el código de error.

Código	Error
-1	Error general durante la publicación
-2	Publicación no autorizada
-3	Aplicación destino no encontrada
-4	Documento incorrecto acorde a la validación sintáctica del esquema
-5	La aplicación destino no acepta la suscripción
-7	Usuario no autorizado a la suscripción
-8	Publicación duplicada
-9	El dominio no permite publicaciones BROADCAST
-10	El documento no se ha insertado correctamente en la base de datos
-11	Problema generando las suscripciones
-12	Error registrando la sesión
-14	Error durante la transformación
-16	Tamaño de la publicación excedido

confirmedPublicationsWS

Recupera las publicaciones que han sido recuperadas por el destinatario

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:confirmedPublicationsWS>
      <clientId>?</clientId>
```

```

    <user>?</user>
    <password>?</password>
    <domain>?</domain>
    <application>?</application>
  </com:confirmedPublicationsWS>
</soapenv:Body>
</soapenv:Envelope>

```

Parámetros:

clientId	Identificador del cliente
user	usuario que realiza la petición
password	contraseña del usuario que realiza la petición
domain	dominio en el cual el documento fue publicado
application	aplicación en la que la cual el documento fue publicado

Si la respuesta es satisfactoria, devolverá la lista de objetos **EBIDocumentWS** con los mensajes que han sido satisfactoriamente recuperados.

Si produce un error, se devolverá una excepción del tipo **EBIBrokerServerException** la cual indicará el código y el motivo del error.

Código	Error
-1	Error general durante la operación
-12	Usuario no válido

notifiedPublicationWS

Emite un acuse de entrega de la confirmación de la publicación

Llamada:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:notifiedPublicationWS>
      <clientId>?</clientId>
      <user>?</user>
      <password>?</password>
      <domain>?</domain>
      <application>?</application>
      <!--Zero or more repetitions:-->
      <messageIds>?</messageIds>
    </com:notifiedPublicationWS>
  </soapenv:Body>
</soapenv:Envelope>

```

Parámetros:

clientId	Identificador del cliente
user	usuario que realiza la petición
password	contraseña del usuario que realiza la petición
domain	dominio donde se encuentra la publicación confirmada
application	aplicación donde se encuentra la publicación confirmada
messageIds	lista con los identificadores de la publicaciones confirmadas, a emitir su acuse de entrega

Si la respuesta satisfactoria devolverá true

Si se produce un error devolverá false, o se devolverá una excepción del tipo `EBIBrokerServerException` la cual indicará el código y el motivo del error.

Código	Error
-1	Error general durante la operación
-12	Usuario no válido

rejectedPublicationsWS

Recupera las publicaciones que han sido rechazadas por el destinatario

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:rejectedPublicationsWS>
      <clientId?></clientId>
      <user?></user>
      <password?></password>
      <domain?></domain>
      <application?></application>
    </com:rejectedPublicationsWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros:

clientId	Identificador del cliente
user	usuario que realiza la petición
password	contraseña del usuario que realiza la petición
domain	dominio en el cual el documento fue publicado
application	aplicación en la que la cual el documento fue publicado

Si la respuesta es satisfactoria, devolverá la lista de objetos *EBIDocumentWS* con los mensajes que han sido satisfactoriamente recuperados.

Si produce un error, se devolverá una excepción del tipo `EBIBrokerServerException` la cual indicará el código y el motivo del error.

Código	Error
-1	Error general durante la publicación
-12	Usuario no válido

notifiedRejectedPublicationWS

Emite un acuse de entrega del rechazo de la publicación

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:notifiedRejectedPublicationWS>
      <clientId?></clientId>
      <user?></user>
      <password?></password>
      <domain?></domain>
    </com:notifiedRejectedPublicationWS>
  </soapenv:Body>
</soapenv:Envelope>
```

```

<application>?</application>
<!--Zero or more repetitions:-->
<messageIds>?</messageIds>
</com:notifiedRejectedPublicationWS>
</soapenv:Body>
</soapenv:Envelope>

```

Parámetros:

clientId	Identificador del cliente
user	usuario que realiza la petición
password	contraseña del usuario que realiza la petición
domain	dominio donde se encuentra la publicación rechazada
application	aplicación donde se encuentra la publicación rechazada
messageIds	lista con los identificadores de la publicaciones rechazadas, a emitir su acuse de entrega

Si la operación se realiza con éxito, devolverá true

Si se produce un error devolverá false, o se generará una excepción del tipo `EBIBrokerServerException` la cual indicará el código y el motivo del error.

Código	Error
-1	Error general durante la operación
-12	Usuario no válido

subscriptionWS

Recoge los mensajes que estén en estado suscrito

Llamada:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:subscriptionWS>
      <clientId>?</clientId>
      <user>?</user>
      <password>?</password>
      <domain>?</domain>
      <application>?</application>
      <schema>?</schema>
      <transformation>?</transformation>
      <reference>?</reference>
    </com:subscriptionWS>
  </soapenv:Body>
</soapenv:Envelope>

```

Parámetros:

clientId	identificador del cliente
user	usuario que realiza la suscripción
password	contraseña del usuario que realiza la suscripción
domain	dominio donde se encuentra la suscripción a recoger
application	aplicación donde se encuentra la suscripción
schema	esquema del cual se quiere recoger la suscripción
transformation	transformación que se desea aplicar a la suscripción, antes de recogerla. Este valor es opcional, sino se indica transformación se recogerá la suscripción tal cual se haya

	publicado.
reference	referencia del mensaje que se quiere recuperar, sino se indica ninguna, se recogerá la primera suscripción que este disponible.

Si la respuesta es satisfactoria, se devolverá, el objeto **EBIDocumentWS**, que contendrá la suscripción recuperada del EBIBROKER. Este objeto a su vez, contendrá un lista de **EBIMessageServerWS**, con el contenido de los ficheros que había en la suscripción.

Si se produce algún error devolverá el código de error, indicando el motivo.

Código	Error
-1	Error general durante la suscripción
-2	Publicación no autorizada a recoger la suscripción
-3	Transformación no encontrada
-4	Error realizando la transformación
-7	Usuario no autorizado a la suscripción
-12	Problema generando las suscripciones

confirmSubscriptionWS

Confirma de forma positiva la entrega de las suscripciones recogidas.

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:confirmSubscriptionWS>
      <clientId?></clientId>
      <user?></user>
      <password?></password>
      <domain?></domain>
      <application?></application>
      <!--Zero or more repetitions:-->
      <messageIds?></messageIds>
    </com:confirmSubscriptionWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros:

clientId	identificador del cliente.
user	usuario que realiza la confirmación.
password	contraseña del usuario que realiza la confirmación.
domain	dominio donde se encuentra la suscripción a confirmar.
application	aplicación donde se encuentra la suscripción a confirmar.
messageID	identificador del mensaje que se quiere confirmar. Si se desean confirmar varios mensajes, se repetirá esta sección con nuevos identificadores.

Si la respuesta es satisfactoria, devolverá 1

Si se produce algún error devolverá el código de error, indicando el motivo.

Código	Error
--------	-------

-1	Error general durante la confirmación
-2	Suscripción no encontrada
-3	Estado del documento, no valido para la confirmación. Los documentos, solo pueden ser confirmados, cuando se encuentran en estado suscrito, pendiente de confirmación o bloqueado
-4	Documento ya en estado confirmado
-12	Error registrando la sesión

rejectSubscriptionWS

Rechaza la suscripción recogida del sistema (es una confirmación negativa).

Llamada:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:com="http://com.edicom.broker.adapter.services">
  <soapenv:Header/>
  <soapenv:Body>
    <com:rejectSubscriptionWS>
      <clientId>?</clientId>
      <user>?</user>
      <password>?</password>
      <domain>?</domain>
      <application>?</application>
      <!--Zero or more repetitions:-->
      <messageIds>?</messageIds>
      <info>?</info>
    </com:rejectSubscriptionWS>
  </soapenv:Body>
</soapenv:Envelope>
```

Parámetros:

clientId	identificador del cliente
user	usuario que realiza el rechazo
password	contraseña del usuario que realiza el rechazo
domain	dominio donde se encuentra la suscripción a rechazar
application	aplicación donde se encuentra la suscripción a rechazar.
messageID	identificador del mensaje que se quiere rechazar. Si se desean rechazar varios mensajes, se repetirá esta sección con nuevos identificadores.
info	Información adicional del motivo del rechazo.

Si la respuesta es satisfactoria, devolverá 1

Si se produce algún error devolverá el código de error, indicando el motivo.

Código	Error
-1	Error general durante el proceso de rechazo
-2	Suscripción no encontrada
-3	Estado del documento, no valido para ser rechazado. Los documentos, solo pueden ser rechazados, cuando se encuentran en estado suscrito, pendiente de confirmación o bloqueado
-4	Documento ya en estado rechazado
-12	Error registrando la sesión

2.4 ACCESO VÍA RESTFUL

Todos los mecanismos de acceso al EBI BROKER vía restful, utilizarán autenticación básica http, como medio de validación

2.4.1 PUBLICACIÓN

DATA POST

El EBI BROKER, ofrece la posibilidad de publicar mensajes, a través del protocolo HTTP/S, mediante un POST .

La llamada HTTP/S, que se debe realizar para poder publicar mensajes, será la siguiente

```
http://xxx.yyy.zz/services/EBIBrokerRESTful/publishpost/{domain}/{application}/{schema}/{destination}/{reference}/{duplicates}/
```

Siendo los parámetros de la llamada HTTP, los siguientes

domain Dominio en el que se publica el documento

application Aplicación en la que se publica el documento

schema Tipo de mensaje asociado al documento

destination Referencia del documento que se publica

reference Referencia del documento que se publica

duplicates Indica la acción a tomar con las publicaciones duplicadas

0: *dupIgnore*. Se generará una nueva publicación aunque esta ya este duplicada

1: *dupAccept*. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH

2: *dupError*. Se devuelve un código de error

3: *dupOverride*. Se elimina la publicación/suscripción original, y se creará una nueva publicación/suscripción nuevas. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación

Si la publicación es correcta, se devuelve el código de publicación asignado por el EBI BROKER, en caso contrario devolverá el código indicando cual ha sido el problema. Códigos de error:

Código	Error
> 0	Identificador de la publicación en el Broker.
-1	Error general durante la publicación
-2	Publicación no autorizada
-3	Aplicación destino no encontrada
-4	Documento incorrecto acorde a la validación sintáctica del esquema

-5	La aplicación destino no acepta la suscripción
-7	Usuario no autorizado a la suscripción
-8	Publicación duplicada
-9	El dominio no permite publicaciones BROADCAST
-10	El documento no se ha insertado correctamente en la base de datos
-11	Problema generando las suscripciones
-12	Error registrando la sesión
-16	Tamaño de la publicación excedido

MULTIPART/FORM DATA POST

El EBI BROKER, ofrece la posibilidad de publicar mensajes, a través del protocolo HTTP/S, mediante un MULTIPART/FORM-DATA POST .

La llamada HTTP/S, que se debe realizar para poder publicar mensajes, será la siguiente

```
https://xxx.xxx.xxx:yyy/services/EBIBrokerRESTful/publishform/
```

y el MULTIPART/FORM-DATA

```
<SCRIPT language="JavaScript">
function OnSubmitForm()
{
    document.myform.action
="https://xxx.xxx.xxx:yyy/services/EBIBrokerRESTful/publish/";
    return true;
}
</SCRIPT>

<FORM name="myform"
onSubmit="return OnSubmitForm();"
enctype="multipart/form-data"
method="post">
<table>
  <thead bgcolor="lightGray" align="center">
    <tr>
      <td>Par&aacute;metros de publicaci&oacute;n</td>
    </tr>
  </thead>
  <tbody align="left">
    <tr>
      <td>Id. cliente</td>
      <td><INPUT type="clientId" name="clientId" size="60"><BR></td>
    </tr>
    <tr>
      <td>Dominio</td>
      <td><INPUT type="domain" name="domain" size="60"><BR></td>
    </tr>
    <tr>
      <td>Aplicaci&oacute;n</td>
      <td><INPUT type="application" name="application" size="60"><BR></td>
    </tr>
    <tr>
      <td>Esquema</td>
      <td><INPUT type="schema" name="schema" size="60"><BR></td>
    </tr>
    <tr>
      <td>Destino</td>
      <td><INPUT type="destination" name="destination" size="60"><BR></td>
    </tr>
    <tr>
      <td>Referencia</td>
      <td><INPUT type="reference" name="reference" size="60"><BR></td>
    </tr>
  </tbody>
</table>
```

```

<tr>
  <td>Duplicados</td>
  <td><SELECT name="duplicates">
    <option>0</option>
    <option>1</option>
    <option>2</option>
  </SELECT><BR></td>
</tr>
<tr>
  <td>Fichero</td>
  <td><INPUT type="file" name="files" size="60"><BR></td>
</tr>
<tr>
  <td>Fichero</td>
  <td><INPUT type="file" name="files" size="60"><BR></td>
</tr>
<tr>
  <td>Fichero</td>
  <td><INPUT type="file" name="files" size="60"><BR></td>
</tr>
<tr>
  <td>Fichero</td>
  <td><INPUT type="file" name="files" size="60"><BR></td>
</tr>
<tr>
  <td>Fichero</td>
  <td><INPUT type="file" name="files" size="60"><BR></td>
</tr>
<tr>
  <td>Fichero</td>
  <td><INPUT type="file" name="files" size="60"><BR></td>
</tr>
</tbody>
</table>
<hr>

```

Siendo los parámetros de la llamada HTTP, los siguientes

ClientId Identificador del cliente HTTP

Domain Dominio en el que se publica el documento

Application Aplicación en la que se publica el documento

Schema Tipo de mensaje asociado al documento

Destination Aplicación destinataria de la publicación

Reference Referencia del documento que se publica

Duplicates Indica la acción a tomar con las publicaciones duplicadas

0: duplgnore. Se generará una nueva publicación aunque esta ya este duplicada

1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH

2: dupError. Se devuelve un código de error

3: dupOverride. Se elimina la publicación/suscripción original, y se creará una nueva publicación/suscripción nuevas. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro del mismo dominio y aplicación

Files Ficheros a publicar

Si la publicación es correcta, se devuelve el código de publicación asignado por el EBI BROKER, en caso contrario devolverá el código indicando cual ha sido el problema. Códigos de error:

Código	Error
> 0	<i>Identificador de la publicación en el Broker.</i>
-1	Error general durante la publicación
-2	Publicación no autorizada
-3	Aplicación destino no encontrada
-4	Documento incorrecto acorde a la validación sintáctica del esquema
-5	La aplicación destino no acepta la suscripción
-7	Usuario no autorizado a la suscripción
-8	Publicación duplicada
-9	El dominio no permite publicaciones BROADCAST
-10	El documento no se ha insertado correctamente en la base de datos
-11	Problema generando las suscripciones
-12	Error registrando la sesión
-16	Tamaño de la publicación excedido

2.4.2 SUSCRIPCIÓN

El EBI BROKER, ofrece la posibilidad de recoger mensajes, a través del protocolo HTTP/S, mediante un POST . Si existen suscripciones pendientes de recoger, en el dominio, aplicación y esquema que se recibe como parámetro, la función devolverá un fichero zip, con el contenido de UNA suscripción, siendo el nombre del fichero zip, el identificador de la suscripción recuperada, el cual podrá ser utilizado posteriormente para emitir el acuse de entrega.

La llamada HTTP/S, que se debe realizar para poder recoger mensajes, será la siguiente

```
http://xxx.yyy.zz/services/EBIBrokerRESTful/subscription/{domain}/{application}/{schema}/{confirmation}/{reference:.*}
```

Siendo los parámetros de la llamada HTTP, los siguientes

domain Dominio en el que se encuentra el documento

application Aplicación en la que se encuentra el mensaje a recuperar

schema Tipo de mensaje asociado al documento

confirmation True si se quiere confirmar el mensaje de forma implícita cuando se entregue en el fichero de respuesta, o false, si el mensaje se deja en el estado pendiente de confirmar, esperando un acuse de entrega

reference Lista de referencias a recuperar

2.4.3 CONFIRMACIÓN

El EBI BROKER, ofrece la posibilidad de confirmar los mensajes recogidos, a través del protocolo HTTP/S, mediante un POST .

La llamada HTTP/S, que se debe realizar para confirmar mensajes, será la siguiente

```
http://xxx.yyy.zz/services/EBIBrokerRESTful/confirmsubscription/{domain}/{application}/{messageid}
```

Siendo los parámetros de la llamada HTTP, los siguientes

domain Dominio en el que se encuentra el documento

application Aplicación en la que se encuentra el documento

messagid Identificador del mensaje a confirmar

Si la confirmación es correcta devolverá 1 como resultado de la función, si se produce algún error estos son los códigos de error que genera la función:

Código	Error
-1	Error general durante la confirmación
-2	Subscripción no encontrada
-3	Estado del documento no valido para ser confirmado. Los documentos solo se pueden confirmar cuando estén en estado Suscrito, Pendiente de Confirmación ó Bloqueado
-4	Documento ya en estado confirmado

2.4.4 TRANSFORMACIÓN Y PUBLICACIÓN

La transformación y publicación de mensajes, aplicará la transformación que se indique como parámetro a los ficheros enviados, y el resultado de la misma será lo que finalmente se publique en el EBI-BROKER

La llamada HTTP/S, que se debe realizar para poder transformar y publicar mensajes, será la siguiente:

```
http://xxx.xxx.xxx:yy/ebiBroker/EBIBrokerRESTful/transformandpublish/{domain}/{application}/{transformation}/{profile}/{filter}/{schema}/{destination}/{reference}/{duplicates}
```

Siendo los parámetros de la llamada HTTP, los siguientes

domain Dominio

application Aplicación

transformation Transformación que se aplicará a los datos antes de ser publicados

profile Perfil de ejecución de la transformación. Para indicar la lista de la valores del perfil de ejecución se realizará utilizando nomenclatura JSON

```
[{"key": "variable1", "value": "data1"}, {"key": "variable2", "value": "data2"}, {"key": "variable3", "value": "data3"}, {"key": "variable4", "value": "data4"}]
```

filter Filtro que se aplicará a la ejecución de la transformación. Si no se desea aplicar filtro, se deberá pasar un espacio en blanco.

schema Esquema de publicación

destination Destino de la publicación

reference Referencia de publicación del mensaje

duplicates Control de duplicados de la publicación

0: dupIgnore

1: dupAccept

2: dupError

mask Máscara de publicación que se aplicará a los ficheros generados por la transformación, publicando solo aquellos que cumplan dicha máscara

Los ficheros a enviar, que serán posteriormente transformados y su resultado publicado, se deberán enviar en un fichero en formato multipart/related

```
--MIME_boundaryB0R9532143182121
Content-ID: <cabfac7.txt>
Content-Disposition: attachment; filename="cabfac7.txt"
Content-Type: text/plain; charset=us-ascii

.....

--MIME_boundaryB0R9532143182121
Content-ID: <linfac7.txt>
Content-Disposition: attachment; filename="linfac7.txt"
Content-Type: text/plain; charset=us-ascii

.....

--MIME_boundaryB0R9532143182121--
```

Como respuesta del POST se devolverá un fichero en formato multipart/related, con los ficheros resultados de la transformación

```
--uuid:6ee21c7f-c9c7-4cf5-b897-f13fe9f3e91e

Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <edi.edi>

UNH+EW1+INVOIC:D:.....
```

Se si produce un error durante la operación generará una excepción del tipo *EBIBrokerServerException*

2.4.5 EJECUCIÓN DE PLANTILLAS

Esta función restful, permitirá ejecutar plantillas almacenadas en el EBI-BROKER

La llamada HTTP/S, que se debe realizar para poder ejecutar plantillas , será la siguiente:

```
http://xxx.xxx.xxx:yy/ebiBroker/EBIBrokerRESTful/executetemplate/{template}/{params:*}
```

Siendo los parámetros de la llamada HTTP, los siguientes

template Nombre de la plantilla a ejecutar.

params Lista con los parámetros que utiliza la plantilla

2.4.6 EXPORTACIÓN E IMPORTACIÓN DE LA CONFIGURACIÓN

La importación y exportación de la configuración de las aplicaciones en el EBI-BROKER, se podrá realizar mediante las funciones restful applicationConfigurationSet y applicationConfigurationGet. La información que se podrá importar y exportar serán la reglas, script, publicaciones, suscripciones, esquemas y transformaciones.

La llamada HTTP/S, que se debe realizar para exportar la configuración de una aplicación , será la siguiente:

```
http://xxx.xxx.xxx:yy/ebiBroker/EBIBrokerRESTful/applicationConfigurationGet/{domain}/{application}
```

Siendo los parámetros de la llamada HTTP, los siguientes

domain Dominio donde esta la aplicación de la cual se quiere exportar la configuración

application Aplicación de la que se quiere exportar la configuración.

Como resultado de la llamada restful, devolverá un fichero ZIP, con el contenido de la exportación de la configuración.

La llamada HTTP/S, que se debe realizar para importar la configuración de una aplicación , será la siguiente:

```
http://xxx.xxx.xxx:yy/ebiBroker/EBIBrokerRESTful/applicationConfigurationSet/{domain}/{application}
```

Siendo los parámetros de la llamada HTTP, los siguientes

domain Dominio donde esta la aplicación en la que se quiere importar la configuración

application Aplicación en la que se quiere importar la configuración.

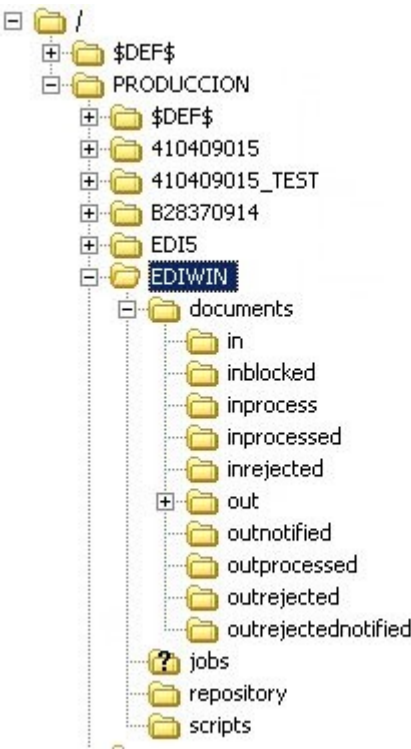
configfile Adjunto a la petición restful, deberá ir el fichero ZIP, que contiene la configuración que se desea importar.

2.5 ACCESO VÍA FTP/S Y SFTP

El EBI BROKER dispone de un acceso nativo vía FTP/S y SFTP, que permitirá tanto la administración del EBI, así como la publicación y recepción de documentos.

2.5.1 SISTEMA VIRTUAL DE CARPETAS

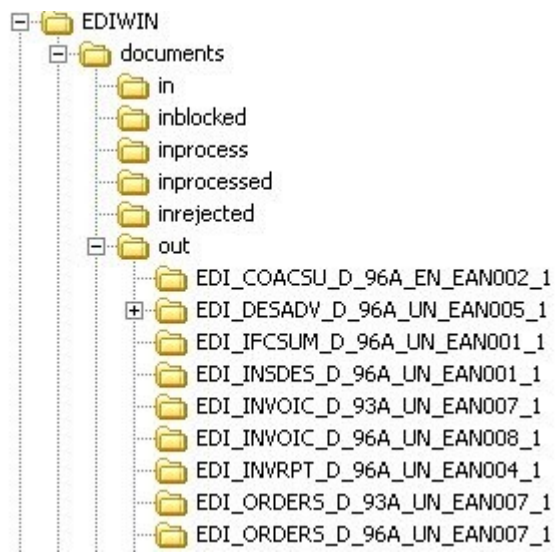
Se ha diseñado un estructura virtual de carpeta con el objetivo de poder administrar y publicar/recibir documentos en el EBI BROKER




La estructura virtual de carpetas esta estructurada de la siguiente forma:

```
/domain/  
/domain/application/  
/domain/application/scripts/  
/domain/application/repository/  
/domain/application/jobs/  
/domain/application/documents/in  
/domain/application/documents/inprocess  
/domain/application/documents/inprocessed  
/domain/application/documents/inblocked  
/domain/application/documents/inrejected  
/domain/application/documents/out  
/domain/application/documents/outprocessed  
/domain/application/documents/outnotified  
/domain/application/documents/outrejected  
/domain/application/documents/outrejectednotified
```

domain	Lista de dominios para los que el usuario tiene permisos de acceso
application	Lista de aplicaciones para las que el usuario tiene permisos de acceso
scripts	Si el usuario tiene permisos de acceso a esta carpeta, le presentará los scripts

	<p>existentes, dentro de la aplicación seleccionada.</p> <p>En este directorio, se podrán crear/modificar/borrar scripts, siempre y cuando el usuario disponga de los suficientes permisos.</p>
repository	<p>Si el usuario tiene permisos de acceso a esta carpeta, le mostrará los objetos del repositorio asociados a la aplicación seleccionada.</p> <p>En este directorio, se podrán crear/modificar/borrar objetos del repositorio, siempre y cuando el usuario disponga de los suficientes permisos.</p>
jobs	<p>Si se disponen de permisos de acceso a esta carpeta, mostrará los trabajos realizados por la aplicación seleccionada. Esta carpeta es de solo lectura.</p>
documents/in	<p>Esta carpeta representa los documentos que se encuentran en estado suscrito. Esta carpeta es de solo lectura.</p> <p>La transferencia de un documento de esta carpeta implicará que el documento pasa al estado <i>pendiente de confirmar</i>, y si la transferencia del documento se realiza con éxito, pasará a estado <i>confirmado</i>, desapareciendo el documento de esta carpeta.</p>
documents/inprocess	<p>Esta carpeta representa los documentos que se encuentran en estado pendiente de confirmar. Esta carpeta es de solo lectura.</p>
documents/inprocessed	<p>Esta carpeta representa los documentos que se encuentran en estado confirmado. Esta carpeta es de solo lectura.</p>
documents/inblocked	<p>Esta carpeta representa los documentos que se encuentran en estado bloqueado. Esta carpeta es de solo lectura.</p>
documents/inrejected	<p>Esta carpeta representa los documentos que se encuentran en estado rechazado. Esta carpeta es de solo lectura.</p>
documents/out	<p>Esta carpeta es la carpeta de salida, sobre la que se realizará la publicación de los documentos.</p> <p>Cuando se acceda a esta carpeta el sistema mostrará una nueva estructura de carpetas que representará los esquemas para los que el usuario tiene autorizada la publicación.</p>  <p>El diagrama muestra una estructura de carpetas. En la raíz, hay una carpeta llamada 'EDIWIN'. Dentro de 'EDIWIN', hay una carpeta llamada 'documents'. Dentro de 'documents', hay varias carpetas: 'in', 'inblocked', 'inprocess', 'inprocessed', 'inrejected' y 'out'. La carpeta 'out' está seleccionada y se muestra una subestructura de carpetas dentro de ella. Estas carpetas son: 'EDI_COACSU_D_96A_EN_EAN002_1', 'EDI_DESADV_D_96A_UN_EAN005_1' (con un signo más a su lado), 'EDI_IFCSUM_D_96A_UN_EAN001_1', 'EDI_INSDES_D_96A_UN_EAN001_1', 'EDI_INVOIC_D_93A_UN_EAN007_1', 'EDI_INVOIC_D_96A_UN_EAN008_1', 'EDI_INVRPT_D_96A_UN_EAN004_1', 'EDI_ORDERS_D_93A_UN_EAN007_1' y 'EDI_ORDERS_D_96A_UN_EAN007_1'.</p> <p>Cuando acceda al esquema con el cual se quiere realizar la publicación, el sistema mostrará aquellas aplicaciones que estén suscritas a dicho esquema.</p>

	 <p>Una vez seleccionada la aplicación destino, será solo en esta carpeta en la que se dispondrá de permisos de escritura, y la transferencia de un fichero a esta carpeta, implicará que se realice una publicación de del fichero transferido, con origen la aplicación seleccionada, con esquema el esquema seleccionado y como aplicación destino, la aplicación seleccionada dentro del esquema.</p>
documents/outprocessed	<p>Esta carpeta representa los documentos que se encuentran en estado publicación confirmada. Esta carpeta es de solo lectura.</p> <p>La transferencia de un fichero de esta carpeta, implicará que el documento cambie de situación de publicación confirmada a publicación notificada</p>
documents/outnotified	<p>Esta carpeta representa los documentos que se encuentran en estado publicación notificada. Esta carpeta es de solo lectura.</p>
documents/outrejected	<p>Esta carpeta representa los documentos que se encuentran en estado publicación rechazada. Esta carpeta es de solo lectura.</p> <p>La transferencia de un fichero de esta carpeta, implicará que el documento cambie de situación de publicación rechazada a publicación rechazada notificada</p>
documents/outrejectednotified	<p>Esta carpeta representa los documentos que se encuentran en estado publicación rechazada notificada. Esta carpeta es de solo lectura.</p>

No se permite la creación de carpetas dentro de ninguna de la estructuras virtuales del servidor.

Cuando se muestre el contenido de las carpetas de documentos, el nombre del fichero que mostrará el servicio FTP/S o SFTP, tendrá el siguiente formato:

- Si es una publicación

```
identificador_del_documento-aplicación_destino-esquema-referencia
```

1. Si es una suscripción

```
identificador_del_documento-aplicación_origen-esquema-referencia
```

2.5.2 CONFIGURACIÓN



El acceso por FTP/S y SFTP, se debe autorizar de forma explícita, en las opciones de configuración del usuario

```
FTP_ACCESS          1
FTP_HOMEDIRECTORY   /DOMINIO/APLICACION/documents
SSH_ACCESS          1
SSH_HOMEDIRECTORY   /DOMINIO/APLICACION/documents
```

Datos Usuario	Configuración	Conexiones	Planificaciones	Permisos
Insertar  Borrar 				
Opción				
FTP_ACCESS				1
FTP_HOMEDIRECTORY				/DOMINIO/APLICACION/documents
GRANT_SYSTEMAPP				1
SSH_ACCESS				1
SSH_HOMEDIRECTORY				/DOMINIO/APLICACION/documents

Una vez autorizado el acceso, y fijado el directorio base, se debe comprobar que el usuario tiene permisos de acceso a la zona de documentos, publicaciones, suscripciones y esquemas de la aplicación, en la que se ha establecido el directorio base, y permisos de publicación y suscripción sobre la aplicación.

Para ello desde la pestaña de PERMISOS, se selecciona el dominio y aplicación correspondientes y se comprueba que tenga permisos necesarios, y en el caso que no los tenga se le habilitan.

Datos Usuario	Configuración	Conexiones	Planificaciones	Permisos
Nombre				
Acceso				
				
Dominios				
Dominio				
Aplicaciones				
Aplicación				
Reglas				
Publicaciones				
Suscripciones				
Esquemas				
Repositorio				
Scripts				
Alarmas				
Procesos activos				
Trabajos				
Estadísticas				
Mensajes				
				
Ejecución				
Publicación				
Suscripción				
Transformación				
Suscripción y Transformación				
Ejecutar script				
Borrar documento				

2.6 FICHEROS DE LOG

El EBI BROKER generará dos ficheros de log, uno de sesión y otro de errores en la ruta que indique la variable de configuración **ebibroker.logsdir**

En nivel de detalle del log de sesión se configura en el fichero **log4j.xml**, que se encontrará ubicado donde diga la variable de contexto, del servidor de aplicaciones, **edicom.path.conf**.

```
<logger name="com.edicom" additivity="false">
  <level value="INFO"/>
  <appender-ref ref="LOGASYNC" />
  <appender-ref ref="LOGFILE"/>
</logger>
```

El nivel de detalle del log admite dos valores, INFO, valor por defecto, TRACE, que generará un log de sesión con información MUY exhaustiva de la actividad del EBI BROKER.

Se producirá un rotación diaria tanto de los logs de sesión, como de errores.

```
2013-09-22 11:52:41,309 INFO [http-bio-8080-exec-1] > [EBI] [EBIADAPTEREDIWINJAVA]
[EBIADAPTEREDIWIN ASPEBI DEVELOPER] [SUPERVISOR] [192.168.0.1] [PRODUCTION] [TEST]
EBIBrokerServerImpl.isAlive ()

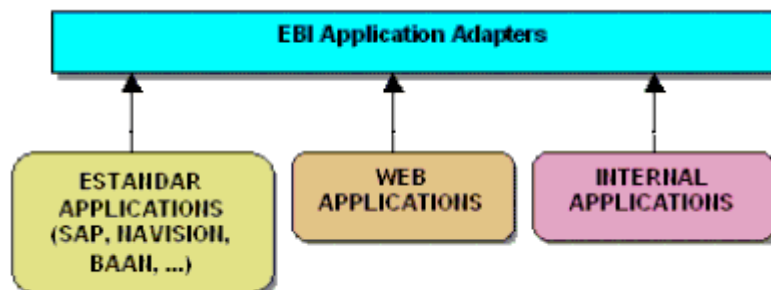
2013-09-22 11:52:41,316 INFO [http-bio-8080-exec-1] < [EBI] [EBIADAPTEREDIWINJAVA]
[EBIADAPTEREDIWIN ASPEBI DEVELOPER] [SUPERVISOR] [192.168.0.1] [PRODUCTION] [TEST]
EBIBrokerServerImpl.isAlive: true
```

La información que se puede observar en cada línea de log por orden de aparición es la siguiente:

- Fecha y hora de la petición.
- Nivel de traza INFO o TRACE
- Tipo de conexión que ha realizado la petición http o NioProcessor (ftp o sftp), e hilo de ejecución asignado.
- Origen de datos que ha realizado la petición EBI
- Tipo de adaptador que ha realizado la petición
- Nombre del adaptador que ha realizado la petición
- Usuario que hace la petición
- IP desde donde se realiza la petición
- Dominio que realiza la petición
- Aplicación que realiza la petición
- Y por ultimo, función invocada con sus parámetros de entrada o salida, en función que sea la petición o la respuesta.

3 ADAPTADORES

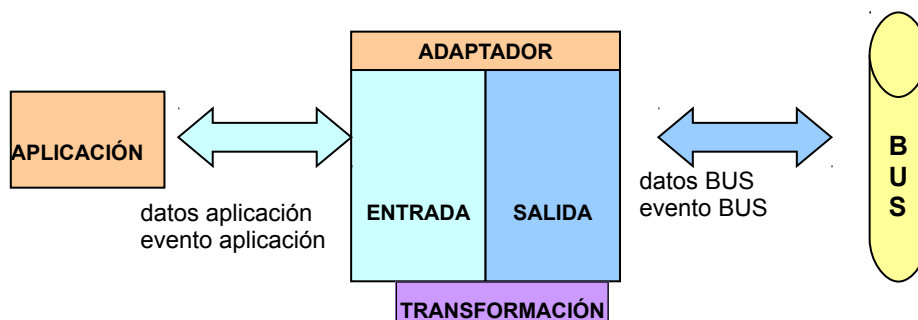
El concepto más importante que subyace en la integración mediante BUS es el de adaptador. Cada aplicación o componente a integrar debe contar con un adaptador que lo conecte al BUS.



Los adaptadores deben permitir mover información entre diferentes fuentes sin necesidad de crear código nuevo. Estos adaptadores esconden la complejidad de la integración transformando y formateando la información en la forma que lo requieren las diferentes aplicaciones.

Los adaptadores que en EBI se llaman "EBI Application Adapter" permiten:

- Transformar los datos de una aplicación o componente al formato que se desee
- Introducir y recoger los datos del BUS a través de la comunicación con el EBI BROKER mediante webServices.



ARQUITECTURA DE EDICOM APPLICATION ADAPTER

La parte "entrada" del adaptador se comunica con la aplicación interna a través del protocolo convenido (FTP, archivos compartidos, MQSERIES, SMTP/POP, HTTP, etc.) y en el formato preferido por la aplicación interna (XML, EDIFACT, TEXTO, BD, etc.).

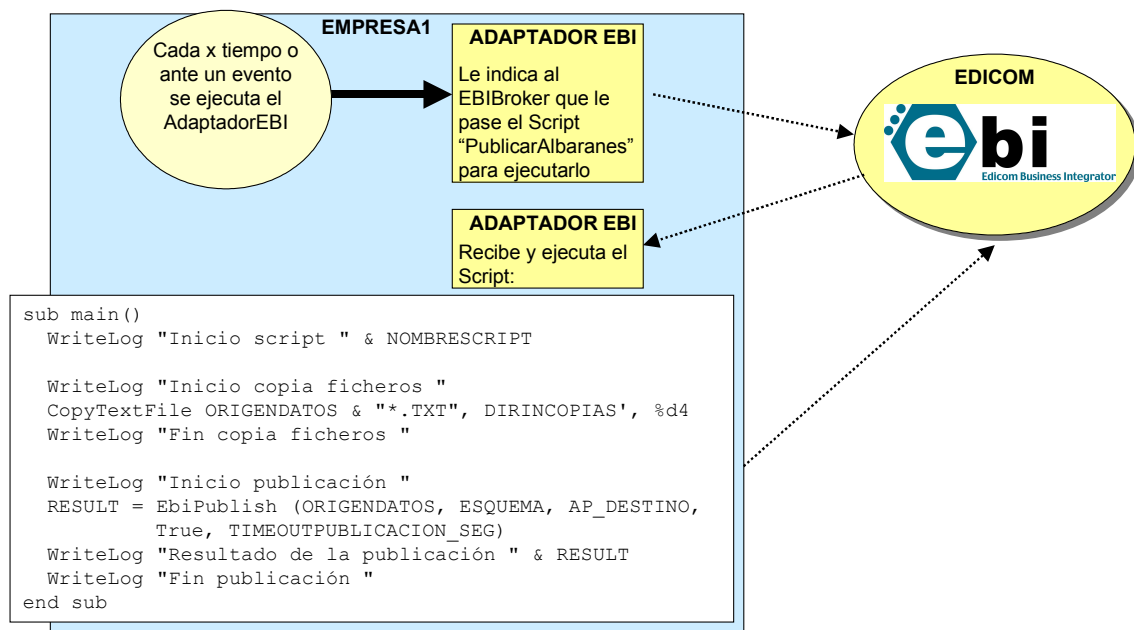
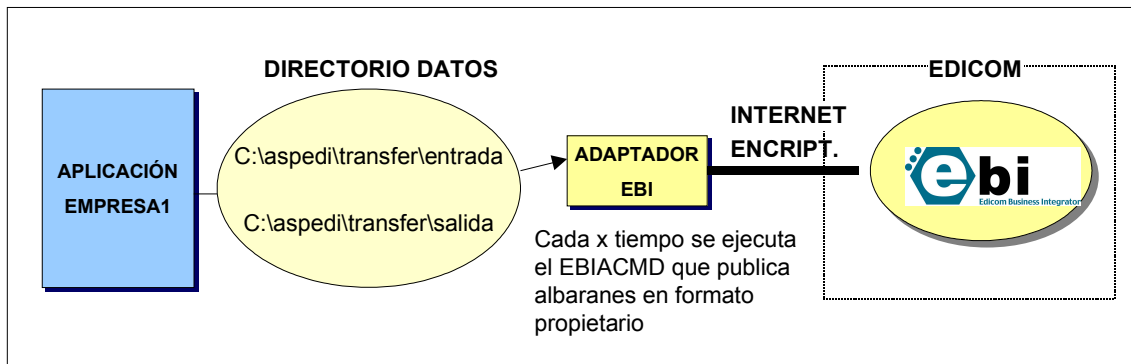
La parte de "salida" del adaptador se comunica con el EBI Broker a través de webServices y deposita y recoge los datos del BUS en el formato acordado como estándar.

La secuencia de transmisión de mensajes entre dos aplicaciones empleando el BUS, vendrá determinada por

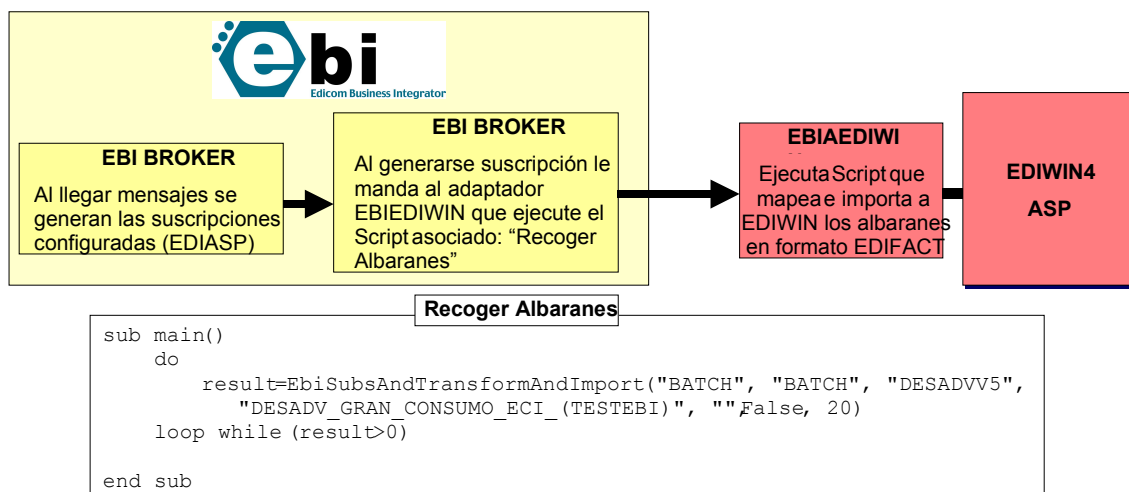
- El modelo de procesos de negocio asociado a la aplicación, activado por eventos, o por planificación. Estos modelos son directamente procesables por el módulo de procesamiento de reglas del EBI BROKER, que se encargará de su ejecución y monitorización
- El código script asociado al proceso. El código script empleado es JavaScript, enriquecido con algunas funciones adicionales para facilitar la integración con el EAI.

Una secuencia de transmisión sería la siguiente:

1. Ante un evento de la aplicación interna, o una petición del EBI BROKER, se lanza una transacción sobre el adaptador que reacciona tomando los datos de la aplicación origen, los transforma en el formato del BUS y se los envía al EBI BROKER.



2. El EBI BROKER identifica el creador y tipo del mensaje y se encarga de distribuirlo a las aplicaciones suscritas a dicho mensaje. En caso de comunicaciones síncronas, a continuación lanza un evento sobre la parte de “salida” del adaptador de dicha aplicación destino para que procese el mensaje recibido.
3. Ante el evento lanzado por el EBI BROKER, la parte de “salida” del adaptador de una aplicación, recoge los datos del BUS y los transforma en el formato requerido por la aplicación destino, avisando posteriormente a esta si es necesario.



3.1 FUNCIONES DE PROPÓSITO GENERAL

Las funciones que se describen a continuación estarán disponibles para todos los adaptadores que se describen a continuación.

El objetivo de estas funciones, es ampliar la potencia del adaptador, para que de forma nativa y sin necesidad de importar librerías, se puedan manejar ficheros, **FileSystemObject**, estructuras XML, **XMLDOM** y conexiones a base de datos, **Connection**.

3.1.1 FILESYSTEMOBJECT

El conjunto de objetos que se describirán a continuación, tienen como objetivo facilitar el acceso al sistema de ficheros del cliente, y por realizar operaciones con los ficheros y directorios. Los objetos que nos facilitarán esta tarea son :

- **FileSystemObject.** Este objeto permitirá acceder al sistema de ficheros, y realizar operaciones sobre los ficheros y carpetas. Este objeto será el único publico, y como resultado de alguna de sus llamadas, devolverá algunos de los siguientes objetos.
- **File.** Permitirá realizar operaciones con los ficheros que devuelvan las funciones del objeto FileSystemObject.

- **Folder.** Permitirá realizar operaciones sobre los directorios.
- **TextStream.** Este objeto permitirá manipular el contenido de los ficheros.

FileSystemObject

BuildPath

Añade el nombre que recibe como parámetro a la ruta existente

```
public String FileSystemObject.BuildPath(String path, String name)
```

path	Ruta origen
name	Nombre de ruta que se quiere concatenar a la ruta origen
Devuelve el resultado de concatenar ambas rutas	

CopyFile

Copia el fichero origen o grupo de ficheros, a la ruta o fichero destino que se le indica

```
public void FileSystemObject.CopyFile(String source, String destination, boolean overwrite)
```

source	Fichero origen o ruta con mascara, del fichero/s que se quieren copiar al destino
destination	Fichero o ruta destino en la que se quieren copiar los ficheros
overwrite	Indica si se quiere sobrescribir el destino en el caso que exista. OPCIONAL

CopyFolder

Copia la ruta origen que se recibe como parámetro, al destino que se indica.

```
public void FileSystemObject.CopyFolder(String source, String destination, boolean overwrite)
```

source	Ruta origen que se desea copiar al destino
destination	Ruta destino donde se copiar el contenido del source
overwrite	Indica si se quiere sobrescribir el destino en el caso que exista. OPCIONAL

CreateFolder

En el caso que no exista, crea el directorio que recibe como parámetro.

```
public Folder FileSystemObject.CreateFolder(String foldername)
```

foldername	Ruta que se quiere crear
Devuelve un objeto del tipo <i>Folder</i> , para permitir operar sobre el directorio	

CreateTextFile

Crea y abre el fichero de texto que recibe como parámetro.

```
public TextStream FileSystemObject.CreateTextFile(String filename, boolean overwrite, boolean unicode)
```

filename	Nombre del fichero con ruta completa, que se desea crear o abrir
overwrite	Indica si se debe sobrescribir el fichero en el caso que exista
unicode	Indica si el fichero esta en formato UNICODE o no
Devuelve un objeto del tipo <i>TextStream</i> , para permitir operar sobre el fichero. En el caso que el fichero exista, y el valor de <i>overwrite</i> , sea <i>false</i> , devolverá <i>null</i>	

```
public TextStream FileSystemObject.CreateTextFile(String filename, boolean overwrite, String charsetName)
```

filename	Nombre del fichero con ruta completa, que se desea crear o abrir
overwrite	Indica si se debe sobrescribir el fichero en el caso que exista
charsetName	Codificación del fichero (UTF-8, ISO-XXXX, etc)
Devuelve un objeto del tipo <i>TextStream</i> , para permitir operar sobre el fichero. En el caso que el fichero exista, y el valor de <i>overwrite</i> , sea <i>false</i> , devolverá <i>null</i>	

DeleteFile

Borra el fichero que recibe como parámetro.

```
public boolean FileSystemObject.DeleteFile(String filespec)
```

filespec	Nombre del fichero con ruta completa, que se desea borrar
Devuelve <i>true</i> si la operación ha tenido éxito, <i>false</i> en caso contrario	

DeleteFolder

Borra de forma recursiva el directorio que recibe como parámetro.

```
public boolean FileSystemObject.DeleteFolder(String folderspec)
```

folderpec	Nombre del directorio que se desea borrar
Devuelve <i>true</i> si la operación ha tenido éxito, <i>false</i> en caso contrario	

FileExists

Comprueba si el fichero que se recibe como parámetro existe

```
public boolean FileSystemObject.FileExists(String filespec)
```

filespec	Nombre del fichero con ruta completa, que se desea comprobar
Devuelve <i>true</i> si el fichero existe, <i>false</i> en caso contrario	

FolderExists

Comprueba si el directorio que recibe como parámetro existe

```
public boolean FileSystemObject.FolderExists(String folderspec)
```

folderpec	Nombre del directorio que se desea comprobar
Devuelve <i>true</i> si el directorio existe, <i>false</i> en caso contrario	

GetBaseName

Recupera el nombre base, sin la ruta ni la extensión

```
public String FileSystemObject.GetBaseName(String filename)
```

filename	Nombre del fichero a recuperar el nombre base
Devuelve el nombre base del directorio	
<pre>a/b/c.txt --> c a.txt --> a a/b/c --> c a/b/c/ --> ""</pre>	

GetExtensionName

Recupera la extensión del fichero

```
public String FileSystemObject.GetExtensionName(String filename)
```

filename	Nombre del fichero a recuperar la extensión
Devuelve el nombre base del directorio	
<pre>foo.txt --> ".txt" a/b/c.jpg --> ".jpg" a/b.txt/c --> "" a/b/c --> ""</pre>	

GetFile

Recupera un objeto del tipo File, asociado al fichero que recibe como parámetro

```
public File FileSystemObject.GetFile(String filename)
```

filename	Nombre del fichero para el que se quiere recuperar el objeto File
Objeto del tipo File, asociado al fichero que se recibe como parámetro	

GetFileName

Recupera el nombre del fichero, sin ruta

```
public String FileSystemObject.GetFileName(String filename)
```

filename	Nombre del fichero del cual se quiere recupera el nombre
Nombre del fichero sin ruta	
<pre>a/b/c.txt --> c.txt a.txt --> a.txt a/b/c --> c a/b/c/ --> ""</pre>	

getFileSeparator

Recupera el separador de ficheros, en función de la plataforma en la que se este ejecutando el script.

```
public String FileSystemObject.getFileSeparator()
```

GetFolder

Recupera un objeto del tipo Folder, asociado al directorio que recibe como parámetro

```
public Folder FileSystemObject.GetFolder(String folder)
```

folder	Nombre del directorio para el que se quiere recuperar el objeto File
Objeto del tipo <i>Folder</i> , asociado al directorio que se recibe como parámetro	

GetParentFolderName

Recupera la ruta padre, del directorio que recibe como parámetro.

```
public String FileSystemObject.GetParentFolderName(String folder)
```

folder	Nombre del directorio para el que se quiere recuperar el padre
Directorio padre, del directorio que se recibe como parámetro	

GetTempName

Recupera el nombre del directorio temporal

```
public String FileSystemObject.GetTempName()
```

MoveFile

Mueve el fichero que recibe como parámetro al fichero o ruta que se le indique.

```
public void FileSystemObject.MoveFile(String source, String destination)
```

source	Nombre del fichero que se desea mover
destination	Nombre del fichero destino o ruta, donde se desea mover.

MoveFolder

Mueve el contenido del directorio origen al destino, creando este si no existe

```
public void FileSystemObject.MoveFolder(String source, String destination)
```

source	Nombre del directorio origen que se desea mover
destination	Nombre del directorio donde se desea mover.

OpenTextFile

Abre el fichero que recibe como parámetro, devolviendo un objeto del tipo *TextStream*, para poder acceder a el.

```
public TextStream FileSystemObject.OpenTextFile( String filename,
int iomode,
String charsetName)
```

filename	Nombre del fichero que se desea abrir
iomode	Modo en el que se abrirá el fichero 1 → Solo lectura 2 → Se abre en modo escritura. Si un fichero, como el mismo nombre existe, este se sobrescribirá 8 → Se abre el fichero en modo escritura, escribiendo al final del fichero.
charsetName	Codificación del fichero UTF-8, ISO-XXXX, etc. OPCIONAL. Sino se indica utilizará la codificación por defecto del sistema operativo.
Objeto del tipo <i>TextStream</i> para acceder al fichero	

File exists

Devuelve true, si el fichero existe, false en caso contrario

```
public boolean exists()
```

getDateCreated

Devuelve la fecha de creación del fichero

```
public Date getDateCreated()
```

getDateLastAccessed

Devuelve la última fecha de acceso al fichero

```
public Date getDateLastAccessed()
```

getDateLastModified

Devuelve la última fecha de modificación del fichero

```
public Date getDateLastModified()
```

getName

Devuelve el nombre del fichero con extensión

```
public String getName()
```

getParentFolder

Ruta del directorio padre donde se encuentra ubicado el fichero

```
public String getParentFolder()
```

getPath

Ruta donde se encuentra ubicado el fichero

```
public String getPath()
```

getSize

Devuelve el tamaño del fichero en bytes.

```
public int getSize()
```

Delete

Borra el fichero asociado al objeto, devolviendo true si ha podido realizar la operación con éxito.

```
public boolean Delete()
```

OpenAsTextStream

Abre el fichero como un fichero de texto, devolviendo el objeto *TextStream*, para su manipulación

```
public TextStream OpenAsTextStream(int iomode, String charsetName)
```

iomode	Modo en el que se abrirá el fichero 1 → Solo lectura 2 → Se abre en modo escritura. Si un fichero, como el mismo nombre existe, este se sobrescribirá 8 → Se abre el fichero en modo escritura, escribiendo al final del fichero.
charsetName	Codificación del fichero UTF-8, ISO-XXXX, etc. OPCIONAL. Sino se indica utilizará la codificación por defecto del sistema operativo.
Objeto del tipo <i>TextStream</i> para acceder al fichero	

Folder**Delete**

Borra el directorio al que hace referencia el objeto *Folder*, devolviendo true si la operación tiene éxito, false, caso contrario.

```
public boolean Delete()
```

getDateCreated

Recupera la fecha de creación del directorio

```
public Date getDateCreated()
```

getDateLastAccessed

Recupera la fecha en la que se accedió al directorio por última vez

```
public Date getDateLastAccessed()
```

getDateLastModified

Recupera la fecha en la que se modifico el directorio por última vez.

```
public Date getDateLastModified()
```

getFiles

Recupera la lista de ficheros que hay en el directorio. El resultado de la función es un array, de objetos *File*.

```
public File[] getFiles()
```

getName

Devuelve el nombre del directorio

```
public String getName()
```

getParentFolder

Devuelve el directorio padre del directorio actual.

```
public String getName()
```

getPath

Recupera la ruta completa del directorio actual.

```
public String getPath()
```

getSize

Recupera de forma recursiva, el tamaño en bytes, del directorio actual, incluyendo el tamaño de los subdirectorios.

```
public int getSize()
```

getSubFolders

Devuelve un array de objetos de tipo *Folder*, con los subdirectorios que hay dentro del directorio actual.

```
public Folder[] getSubFolders()
```

IsRootFolder

Comprueba si el directorio es el directorio raíz o no

```
public boolean IsRootFolder()
```

setName

Cambio el nombre al directorio actual.

```
public void setName(String s)
```

TextStream

getAtEndOfLine

Devuelve true, si el curso se encuentra al final de línea, false en caso contrario.

```
public boolean getAtEndOfLine()
```

getAtEndOfStream

Devuelve *true*, si el cursor se encuentra al final del fichero, *false* en caso contrario

```
public boolean getAtEndOfStream()
```

getColumn

Devuelve el número de columna, en la que se encuentra posicionado el cursor que recorre el fichero.

```
public int getColumn()
```

getLine

Devuelve el número de línea en la que se encuentra posiciona el cursor que recorre el fichero.

```
public int getLine()
```

Read

Lee el número de caracteres que se indica como parámetro, a partir de la posición actual del fichero.

```
public String Read(int chars)
```

ReadAll

Devuelve como resultado de la llamada a la función, todo el contenido del fichero, desde la posición actual, hasta el final.

```
public String ReadAll()
```

ReadLine

Lee la línea actual del fichero de texto, devolviéndola como resultado de la función, y posiciona el cursor en la línea siguiente.

```
public String ReadLine()
```

Seek

Coloca el cursor del fichero de texto, en la posición que recibe como parámetro.

```
public void Seek(long pos)
```

Skip

Mueve hacia delante, la posición del cursor que recorre el fichero de texto, el número de posiciones que se indica como parámetro.

```
public void Skip(int chars)
```

SkipLine

Mueve la posición del cursor que recorre el fichero de texto, a la siguiente línea.

```
public void SkipLine()
```

Write

Escribe en la posición en la que se encuentre el cursor en el fichero, el texto que recibe como parámetro.

```
public void Write(String text)
```

WriteBlankLines

Añade al fichero el número de líneas en blanco que se le indica como parámetro.

```
public void WriteBlankLines(int lines)
```

WriteLine

Escribe en el fichero la línea que recibe como parámetro.

```
public void WriteLine(String text)
```

Si se invoca sin parámetro, escribe un salto de línea

```
public void WriteLine()
```

Ejemplos

Recupera los ficheros que hay dentro de un directorio.

```
var carpeta = FileSystemObject.GetFolder("/tmp/");

WriteLog ("Carpeta:" + carpeta);

var ficheros = carpeta.GetFiles();

for (var i = 0; i < ficheros.length; i++){
    fichero = ficheros[i];
    WriteLog ("Fichero:" + fichero.getName());
}
```

Abre un fichero en modo solo lectura, y lee su contenido

```
var ForReading = 1;
var file = FileSystemObject.OpenTextFile(anomfich, ForReading, "ISO-8859-15");
var text = file.ReadAll();
file.Close();
WriteLog ("Texto : " + text);
```

Abre un fichero y lo recorre línea a línea leyendo su contenido

```
if (FileSystemObject.FileExists(ORIGENDATOS + "FACBM.TXT")) {
    var txtStream = FileSystemObject.OpenTextFile(ORIGENDATOS + "FACBM.TXT",1);
    while (!txtStream.eof){
        var Linea = txtStream.ReadLine();
        WriteLog("valor de la linea " + Linea );
    }
}
```

3.1.2 XMLDOM

El conjunto de objetos XMLDOM, tienen como objetivo poder crear y leer estructuras XML. Los objetos que facilitarán esta tarea son :

- XMLDOM. Objeto principal, que permitirá la carga, creación y manipulación de las estructuras XML
- XmlNodeList. Objeto que almacena la lista de nodos de una estructura XML.
- XmlNode. Objeto que representa un nodo de la estructura XML.
- XmlAttr. Objeto que representa un atributo de un nodo.
- XmlText. Objeto que representa un nodo de tipo texto.
- XmlComment. Objeto que representa un comentario xml.
- XmlElement. Objeto que representa un elemento dentro de la estructura XML
- XmlError. Objeto que contiene los errores que se pueden producir, durante el procesamiento de estructuras XML
- XmlNamedNodeMap

XMLDOM

createCDATASection

Crea una estructura CDATA, para añadirla al documento XML. La información contenida dentro de la estructura CDATA, no será interpretada por el analizador XML.

```
public XmlCDATASection createCDATASection(String data)
```

data	Contenido que se insertará dentro de la sección CDATA
	<code><![CDATA[..... data]]></code>
Objeto XmlCDATASection creado	

createComment

Crea un objeto comentario, para añadir a la estructura xml

```
public XmlComment createComment(String data)
```

data	Contenido del comentario que se ha creado
Objeto XmlComment creado	

createElement

Crea un elemento xml, con el tag que se indica como parámetro.

```
public XmlElement createElement(String tagName)
```

name	Nombre del nuevo elemento
Objeto XmlElement creado	

createTextNode

Crea un nodo de texto, con la cadena que recibe como parámetro.

```
public XmlText createTextNode(String data)
```

data	Contenido del nodo de texto
Objeto XmlText creado	

getElementsByTagName

Recupera la lista de nodos, en el objeto XmlNodeList, cuyo nombre coincide con el que se pasa como parámetro

```
public XmlNodeList getElementsByTagName(String tagname)
```

tagname	Nombre de los nodos a recuperar
Lista con los nodos que cumplen el criterio de búsqueda.	

getXmlError

Recupera en un objeto XmlError, los errores que se hayan podido producir durante la carga o generación de un xml

```
public XmlError getXmlError()
```

load

Carga un fichero XML, en el objeto XMLDOM

```
public boolean load(String location)
```

location	Nombre del fichero con ruta completa, donde se encuentra la estructura XML a cargar
True si sea conseguido cargar la estructura con éxito, false en caso contrario.	

loadXML

Carga una cadena de texto XML en el objeto XMLDOM

```
public boolean loadXML(String xml)
```

xml	Cadena de texto, en formato XML que se quiere cargar en el objeto XMLDOM
True si sea conseguido cargar la estructura con éxito, false en caso contrario.	

reset

Borra la estructura XML almacenada en el objeto XMLDOM

```
public void reset()
```

save

Almacena al estructura XML, en la fichero indicado en el parámetro

```
public void save(String location)
```

XmlNodeList

Esta objeto almacena una lista de nodos (*XmlNode*)de la estructura XML

item

Recupera el nodo cuyo índice recibe como parámetro.

```
public XmlNode item(int index)
```

index	Índice del nodo XML que se quiere recuperar
Objeto XmlNode que esta en el índice indicado.	

getLength

Devuelve el número de nodos en la lista

```
public int getLength()
```

XmlNode**appendChild**

Añade el nodo hijo que recibe como parámetro al nodo actual.

```
public XmlNode appendChild(XmlNode newChild)
```

newChild	Nodo que se quiere añadir al nodo que invoca a la función
Objeto XmlNode que se ha añadido	

getAttributes

Devuelve un objeto *XmlNamedNodeMap*, con los atributos del nodo

```
public XmlNamedNodeMap getAttributes()
```

getChildNodes

Devuelve un objeto *XmlNodeList*, con la lista de nodos hijos, del nodo actual

```
public XmlNodeList getChildNodes()
```

getFirstChild

Recupera el primer nodo hijo, del nodo actual

```
public XmlNode getFirstChild()
```

getLastChild

Recupera el último nodo hijo, del nodo actual

```
public XmlNode getLastChild()
```


getNextSibling

Devuelve el siguiente nodo hermano, al nodo actual

```
public XmlNode getNextSibling()
```

getNodeName

Devuelve el nombre del nodo, dependiendo del tipo de nodo que sea

```
public String getNodeName()
```

getNodeType

Tipo de nodo de la estructura XML

```
public short getNodeType()
```

Tipos de nodo

```

ELEMENT_NODE           = 1
ATTRIBUTE_NODE         = 2
TEXT_NODE              = 3
CDATA_SECTION_NODE     = 4
ENTITY_REFERENCE_NODE  = 5
ENTITY_NODE            = 6
PROCESSING_INSTRUCTION_NODE = 7
COMMENT_NODE           = 8
DOCUMENT_NODE          = 9
DOCUMENT_TYPE_NODE     = 10
DOCUMENT_FRAGMENT_NODE = 11
NOTATION_NODE          = 12

```

getNodeValue

Valor del nodo, que dependerá del tipo que sea

```
public String getNodeValue()
```

getParentNode

Recupera el nodo padre, del nodo actual.

```
public XmlNode getParentNode()
```

getPreviousSibling

Recupera el hermano previo al nodo actual

```
public XmlNode getPreviousSibling()
```

getText

Recupera el texto contenido en el nodo actual

```
public String getText()
```

getXml

Recupera la estructura XML del nodo actual

```
public String getXml()
```

hasChildNodes

Devolverá true o false, si el nodo actual tiene nodos hijo o no

```
public boolean hasChildNodes()
```

insertBefore

Añade el nodo hijo que recibe como parámetro, antes del nodo que se indica como *refChild*. Si no existe el nodo referencia, se añadirá al final de los nodos hijo.

```
public XmlNode insertBefore(XmlNode newChild, XmlNode refChild)
```

newChild	Nodo que se quiere añadir como hijo, antes del nodo refChild
refChild	Nodo referencia, para insertar el nuevo nodo hijo. Si este nodo no existe, se añadirá al final de los nodos hijos.
Objeto XmlNode que se ha añadido	

removeChild

Elimina el nodo hijo que recibe como referencia

```
public XmlNode removeChild(XmlNode oldChild)
```

replaceChild

Sustituye en los nodos hijos del nodo actual, el nodo oldChild, por el nodo newChild

```
public XmlNode replaceChild(XmlNode newChild, XmlNode oldChild)
```

newChild	Nodo que se quiere
oldChild	Nodo referencia, para insertar el nuevo nodo hijo. Si este nodo no existe, se añadirá al final de los nodos hijos.
Objeto XmlNode que se ha sustituido	

selectNodes

Devuelve un objeto XmlNodeList, con los objetos que cumplen el criterio de búsqueda indicado en la función

```
public XmlNodeList selectNodes(String expression)
```

expression	Expresión para localizar nodos en la estructura XML	
	Expresión	Descripción
	nodename	Selecciona todos los nodos con el nombre "nodename"
	/	Realiza la selección desde el nodo raíz.
	//	Selecciona los nodos, desde el nodo actual, sin importar donde estén.
	.	Selecciona el nodo actual.
	..	Selecciona el nodo padre al actual
	@	Selecciona atributos
Objeto <i>XmlNodeList</i> , con los nodos que cumplan el criterio de búsqueda		

Ejemplo

Para una estructura XML como la siguiente:

```
<?xml version="1.0"?>
<Employees>
  <Employee emplid="1111" type="admin">
    <firstname>John</firstname>
    <lastname>Watson</lastname>
    <age>30</age>
    <email>johnwatson@sh.com</email>
  </Employee>
  <Employee emplid="2222" type="admin">
    <firstname>Sherlock</firstname>
```

```

    <lastname>Homes</lastname>
    <age>32</age>
    <email>sherlock@sh.com</email>
  </Employee>
  <Employee emplid="3333" type="user">
    <firstname>Jim</firstname>
    <lastname>Moriarty</lastname>
    <age>52</age>
    <email>jim@sh.com</email>
  </Employee>
  <Employee emplid="4444" type="user">
    <firstname>Mycroft</firstname>
    <lastname>Holmes</lastname>
    <age>41</age>
    <email>mycroft@sh.com</email>
  </Employee>
</Employees>

```

Expresión	Resultado
employee	Recupera todos los nodos con el nombre "employee"
employees/employee	Recupera todos los nodos "employee" que son hijos del nodo "employees"
//employee	Selecciona todos los elementos "employee" sin importar donde se encuentre en el documento
/employees/employee[1]	Selecciona el primer nodo "employee" hijo del nodo "employees"
/employees/employee[last()]	Selecciona el último nodo "employee" hijo del nodo "employees"
/employees/employee[last()-1]	Selecciona el penúltimo nodo "employee" hijo del nodo "employees"
//employee[@type='admin']	Selecciona todos los nodos "employee", que tenga en el atributo "type" el valor "admin"

selectSingleNode

Recupera un objeto del tipo *XmlNode*, acorde al criterio de búsqueda que se pasa a la función.

```
public XmlNode selectSingleNode(String expression)
```

expression	Expresión para localizar el nodo en la estructura XML
Expresión	Descripción
nodename	Selecciona todos los nodos con el nombre "nodename"
/	Realiza la selección desde el nodo raíz.
//	Selecciona los nodos, desde el nodo actual, sin importar donde estén.
.	Selecciona el nodo actual.
..	Selecciona el nodo padre al actual
@	Selecciona atributos

Objeto *XmlNode*, con el nodo que cumple el criterio de búsqueda

setNodeValue

Dependiendo del tipo de nodo, fija el valor del mismo

```
public void setNodeValue(String nodeValue)
```

nodeValue	Nuevo valor del nodo XML
-----------	--------------------------

XmlAttribute

Este objeto extiende todas las propiedades del objeto *XmlNode* y añade las siguientes

getName

Devuelve el nombre del atributo

```
public String getName()
```

getSpecified

Indica si el atributo esta especificado, para nodo actual.

```
public boolean getSpecified()
```

getValue

Recupera el valor del atributo

```
public String getValue()
```

setValue

Fija el valor del atributo

```
public void setValue(String value)
```

XmlText

Este objeto hereda todas las propiedades del objeto *XmlNode*, añadiendo las siguientes:

splitText

Parte el nodo actual, en dos nodos, tomando como punto de corte, el offset, que se indica como parámetro. Ambos nodos se mantendrán como hermanos. Cuando el offset indicado, sea mayor, que la longitud del primer nodo, el nuevo nodo, no contendrá datos.

```
public XmlText splitText(int offset)
```

offset	Offset para indicar el punto de corte.
Nuevo nodo, que contiene la información después del offset indicado.	

XmlElement

Este objeto hereda todas las propiedades del objeto *XmlNode*, añadiendo las siguientes:

getAttribute

Recupera el valor del atributo que se indica como parámetro

```
public String getAttribute(String name)
```

name	Nombre del atributo a recuperar
Recupera el valor del atributo indicado en el parámetro	

getAttributeNode

Recupera el objeto XmlAttr indicado como parámetro

```
public XmlAttr getAttributeNode(String name)
```

name	Nombre del atributo a recuperar
Recupera el objeto XmlAttr que contiene el atributo indicado en el parámetro	

getElementsByTagName

Recupera la lista de nodos hijo, del nodo actual, que se indica en el parámetro

```
public XmlNodeList getElementsByTagName(String name)
```

name	Nombre los nodos hijos a recuperar
Objeto <i>XmlNodeList</i> con los nodos hijos que si indican en el parámetro name	

getTagName

Devuelve el nombre del nodo actual

```
public String getTagName()
```

removeAttribute

Elimina el atributo que se indica como parámetro, del nodo actual.

```
public void removeAttribute(String name)
```

removeAttributeNode

Elimina el atributo que recibe como parámetro en el nodo actual.

```
public XmlAttr removeAttributeNode(XmlAttr oldAttr)
```

oldAttr	Atributo que se desea eliminar
Objeto <i>XmlAttr</i> con el atributo eliminado	

setAttribute

Añade un atributo al nodo actual

```
public void setAttribute(String name, String value)
```

name	Nombre del atributo
value	Valor del atributo

setAttributeNode

Añade el atributo que recibe como parámetro al nodo actual.

```
public XmlAttr setAttributeNode(XmlAttr newAttr)
```

NewAttr	Nuevo atributo que se desea añadir
Objeto <i>XmlAttr</i> que se ha añadido al nodo actual	

XmlError

getErrorCode

Recupera el código de error

```
public int getErrorCode()
```

getLine

Devuelve la línea de la estructura xml, donde se ha producido el error

```
public int getLine()
```

getLinePos

Devuelve la posición dentro de la línea, donde se ha producido el error

```
public int getLinePos()
```

getReason

Devuelve un texto, describiendo el motivo del error

```
public String getReason()
```

XmlNamedNodeMap

Este objeto, almacena los atributos de un nodo, que podrán ser accedidos por el nombre del atributo.

getLength

Devuelve el número de atributos que hay almacenados en el objeto.

```
public int getLength()
```

getNamedItem

Recupera el atributo que se le indica como parámetro.

```
public XmlNode getNamedItem(String name)
```

item

Recupera el atributo cuyo índice se indica en el parámetro.

```
public XmlNode item(int index)
```

removeNamedItem

Elimina el atributo que se indica como parámetro.

```
public XmlNode removeNamedItem(String name)
```

setNamedItem

Añade el atributo que se recibe como parámetro.

```
public XmlNode setNamedItem(XmlNode arg)
```

Ejemplos

Carga un XML desde un fichero, localiza los nodos con el tag, "V_MSG_ERROR", y los recorre escribiendo su contenido

```
WriteLog ("Start" );
var result = XMLDOM.load ("ORDERS_IN.xml");
WriteLog ("result : " + result );

var nodeList = XMLDOM.getElementsByTagName("V_MSG_ERROR");

if (nodeList != null){
    var i = nodeList.getLength();
    for (j=0;j<i;j++) {
        var node = nodeList.item(j);
        var nodeaux = node.getFirstChild();
        while (nodeaux!=null) {
            if (nodeaux.getNodeType() == 1) {
                WriteLog("node getNodeName " + nodeaux.getNodeName() );
            }
            nodeaux = nodeaux.getNextSibling();
        }
    }
}
```

```

        WriteLog("node getNodeTypes " + nodeaux.getNodeTypes() );
        WriteLog("node getText " + nodeaux.getText() );
    }
    nodeaux = nodeaux.getNextSibling();
}
}
WriteLog("Find " + i );
}
else {
    WriteLog("NodeList null ");
}
}

```

Crea un XML desde el principio,

```

WriteLog ("Start" );
XMLDOM.setAsync(false);
var result = XMLDOM.loadXML("<?xml version = \"1.0\"?>");

var nodoBook = XMLDOM.createElement("Procedure");
nodoBook.setAttribute ("name", "APPS.PK_PARAMETROS.F_RECUPERA_PARAMETRO");
nodoBook.setAttribute ("loaded", "FALSE");
XMLDOM.appendChild(nodoBook);

var nodoComment = XMLDOM.createComment("TEST XML");
nodoBook.appendChild(nodoComment);

var nodoCodClave = XMLDOM.createElement("P_Cod_Clave");
nodoCodClave.setAttribute ("direction", "in");
nodoCodClave.setAttribute ("class", "int");
nodoCodClave.setAttribute ("type", "");
nodoCodClave.text="Cod_Clave";

nodoBook.appendChild(nodoCodClave);

var nodoCodEntr = XMLDOM.createElement("P_Cod_Entr");
nodoCodEntr.setAttribute ("direction", "in");
nodoCodEntr.setAttribute ("class", "int");
nodoCodEntr.text="Cod_Entr";
nodoBook.appendChild(nodoCodEntr);

var nodoTipoFecha = XMLDOM.createElement("P_TIPO_FECHA");
nodoTipoFecha.setAttribute ("direction", "in");
nodoTipoFecha.setAttribute ("class", "int");
nodoTipoFecha.text="TIPO_FECHA";
nodoBook.appendChild(nodoTipoFecha);

var nodoPReturn = XMLDOM.createElement("P_Return");
nodoPReturn.setAttribute ("direction", "return");
nodoBook.appendChild(nodoPReturn);

WriteLog (XMLDOM.getXml());

```

generando el siguiente XML

```

<Procedure loaded="FALSE" name="APPS.PK_PARAMETROS.F_RECUPERA_PARAMETRO">
  <!--TEST XML-->
  <P_Cod_Clave class="int" direction="in" type="">Cod_Clave</P_Cod_Clave>
  <P_Cod_Entr class="int" direction="in">Cod_Entr</P_Cod_Entr>
  <P_TIPO_FECHA class="int" direction="in">TIPO_FECHA</P_TIPO_FECHA>
  <P_Return direction="return"/>
</Procedure>

```

3.1.3 CONNECTION

Las funciones del tipo *Connection*, permitirán a los scripts ejecutados en los adaptadores, realizar operaciones de SELECT/UPDATE/INSERT/DELETE, contra base de datos. Las base de datos contra las que puede operar el adaptador son SQL SERVER, ORACLE, DB2 y MySQL.

Los objetos disponibles para realizar las operaciones de *Connection* son:

- *Connection*. Este es el objeto principal, que permitirá establecer la conexión con la base de datos, y realizar las operaciones de acceso a la información de la base de datos.
- *RecordSet*. Si la operación de base de datos, devuelve resultados, a través de este objeto se podrá recorrer y recuperar la información.

Connection BeginTrans

Inicia una transacción de base de datos

```
Connection.BeginTrans()
```

Close

Cierra la conexión actual con la base de datos

```
Connection.Close()
```

CommitTrans

Cierra la transacción activa de base de datos

```
Connection.CommitTrans()
```

Execute

Ejecuta el comando SQL que recibe como parámetro

```
RecordSet Connection.Execute(String sql,
                             int resultSetType,
                             int resultSetConcurrency)
```

sql	Sentencia SQL a ejecutar
resultSetType	<p>Modo en el que se puede recorrer el resultado de la base de datos. OPCIONAL valor por defecto (TYPE_SCROLL_INSENSITIVE) TYPE_FORWARD_ONLY = 1003; Solo se puede mover el cursor hacia delante. TYPE_SCROLL_INSENSITIVE = 1004; Se puede mover por el resultado hacia delante y hacia atrás, no siendo sensible a los cambios que pudiera haber en la base de datos tras la consulta. TYPE_SCROLL_SENSITIVE = 1005; Se puede mover por el resultado hacia delante y hacia atrás, siendo sensible a los posibles cambios que pudiera haber en la base de datos.</p>
resultSetConcurrency	<p>OPCIONAL valor por defecto (CONCUR_READ_ONLY) CONCUR_READ_ONLY = 1007; El resultado devuelto es de solo lectura CONCUR_UPDATABLE = 1008; El resultado devuelto puede modificarse</p>
Objeto Recordset con el resultado de la operación SQL ejecutada. Este valor de vuelta solo tendrá sentido si la operación es del tipo SELECT	

ExecuteStoreProc

Esta función permite ejecutar procedimientos almacenados en el servidor de base de datos

```
RecordSet Connection.ExecuteStoreProc(String storeProc, Object[] params )
```

storeProc	Nombre del procedimiento almacenado a ejecutar.
params	Array con los parámetros del procedimiento almacenado.
Objeto Recordset con el resultado del procedimiento almacenado.	

Open

Abre una conexión contra la base de datos que recibe como parámetro

```
Connection.Open(String url, String username, String password)
```

url	Cadena conexión en formato JDBC, para acceder a la base de datos
username	Usuario de conexión a la base de datos
password	Contraseña de conexión a la base de datos

RollbackTrans

Deshace la transacción activa de base de datos

```
Connection.RollbackTrans()
```

State

Consulta el estado de la conexión con la base de datos

```
Connection.State()
```

Devuelve el estado de la conexión

1 → Abierta

0 → Cerrada

Recordset

AbsolutePosition

Devuelve la posición actual del cursor

```
public int AbsolutePosition()
```

AbsolutePosition (int pos)

Fija la posición de cursor en el conjunto de datos

```
public void AbsolutePosition(int pos)
```

pos	Nueva posición del cursor
------------	---------------------------

ActiveConnection

Devuelve la cadena de JDBC de conexión, asociada al RecordSet.

```
public String ActiveConnection()
```

AddNew

Añade una nueva fila, con los valores que recibe como parámetro.

```
public void AddNew(Object fields[], Object values[])
```

fields	Array con las columnas que se van a recibir valores.
values	Array con los nuevos valores de las columnas.

CancelUpdate

Cancela todas las actualizaciones pendientes de realizarse.

```
public void CancelUpdate()
```

Close

Cierra el RecordSet activo

```
public void Close()
```

Delete

Borra la fila actual del RecordSet

```
public void Delete()
```

getBlob

Recupera el BLOB de la columna que recibe como parámetro como un conjunto de bytes

```
public byte[] getBlob(String columnName)
```

columnName	Nombre de la columna a recuperar
Array de bytes con el contenido del BLOB	

GetRows

Devuelve en un array de dos dimensiones (fila, columna), el conjunto de datos del RecordSet

```
public Object[][] GetRows(int rows)
```

rows	Número de filas a recuperar. Si el valor es -1, se recuperarán todas las filas del RecordSet
Array de dos dimensiones (fila/columna) con el contenido del RecordSet	

getValue

Recupera el valor de la columna que recibe como parámetro.

```
public Object getValue(String columnName)
```

columnName	Nombre de la columna de la que se quiere recuperar el resultado
Contenido de la columna solicitada	

Move

Mueve el cursor de datos el número de filas que se le indica como parámetro.

```
public void Move(int numRows)
```

numRows	Número de filas a mover el cursor de datos
----------------	--

MoveFirst

Mueve el cursor a la primera fila.

```
public void MoveFirst()
```

MoveLast

Mueve el cursor a la última fila

```
public void MoveLast()
```

MoveNext

Mueve el cursor a la siguiente fila del RecordSet

```
public void MoveNext()
```

MovePrevious

Mueve el cursor a la final anterior del RecordSet

```
public void MovePrevious()
```

RecordCount

Devuelve el número de filas del RecordSet

```
public int RecordCount()
```

Requery

Lanza la consulta activa en el RecordSet, de nuevo contra la base de datos

```
public void Requery()
```

Update

Actualiza en la base de datos, los cambios pendientes en el RecordSet

```
public void Update()
```

Ejemplos

Ejecución de una consulta y recorrido de su resultado

```
WriteLog ("Connection : ");

/*PARAMETROS DE CONEXION AL DB2*/
url = "jdbc:db2://bcbeta122:50002/bcddocta";
user = "db2instx";
pwd = "db2instx";
sql = "SELECT ID_SELLO_DIGITAL FROM EDOCTA.SELLO_DIGITAL_EDO_CTA WHERE
ID_ESTATUS_SELLADO=1";

Connection.Open(url,user,pwd);

TYPE_FORWARD_ONLY = 1003;
CONCUR_READ_ONLY = 1007;
WriteLog ("Ejecutando : " + sql);
rs = Connection.Execute(sql,TYPE_FORWARD_ONLY, CONCUR_READ_ONLY);
while (!rs.EOF) {
    WriteLog ("ID " + rs.getValue("ID_SELLO_DIGITAL"));
    rs.MoveNext();
}

WriteLog ("Connection fin ");
```

Ejecución de un procedimiento almacenado y recorrido de su resultado

```
/*ORACLE*/
url = "jdbc:oracle:thin:@192.168.116.128:1521:xe";
user = "xdev";
pwd = "xdev";

Connection.Open(url,user,pwd);
rs = Connection.ExecuteStoreProc("ObtenerLote ()",[]);

while (!rs.EOF) {
    for (i=0; i<rs.Fields.getCount(); i=i+1) {
        field = rs.Fields.getItem(i);
        WriteLog (field.Name() + " -- " + field.getValue());
    }
    rs.MoveNext();
}
```

3.1.4 VARIABLES DISPONIBLES

Application. Aplicación que esta ejecutando el script.

ComputerName. Nombre de la máquina que esta ejecutando el script.

DIRTEMP. Directorio temporal asociado a la ejecución del script. Cuando el script, finalice este directorio será borrado por el adaptador.

ExecError. En caso que se produzca algún error durante la ejecución del script, contendrá el mensaje de error.

LogName. Nombre con ruta, donde se escribirá el log de ejecución del script en curso.

ScriptName. Nombre del script en ejecución.

TIMEOUT. Tiempo máximo en segundos, de ejecución del script.

3.1.5 FUNCIONES DE PROPÓSITO GENERAL.

AddError

Añade un texto de error en log de ejecución del script

```
public void AddError(String text)
```

AddFile

Añade el fichero source al fichero destination

```
public boolean AddFile(String source, String destination, boolean deletesource)
```

source	Ruta completa donde se encuentra el fichero origen
destination	Ruta completa donde se encuentra el fichero destino
deletesource	Indica si se debe borrar el fichero origen
true si la operación tiene éxito, false en caso contrario	

AddFiles

Añade los ficheros que cumple la máscara origen al fichero destino

```
public boolean AddFiles(    String maskSource,
                          String destination,
                          boolean deleteSource)
```

maskSource	Mascara con los ficheros origen
destination	Fichero destino donde se concatenaran los ficheros origen
deleteSource	Indica si se deben borrar o no los fichero origen
true si la operación tiene éxito, false en caso contrario	

AddTextToFile

Añade el texto que recibe como parámetro al fichero file

```
public void AddTextToFile(String text, String file)
```

text	Texto a concatenar al fichero
file	Fichero al que se quiere concatenar el texto

AddTrace

Añade un texto de traza en log del script

```
public void AddTrace(String text)
```

text	Texto añadir en el log de traza
-------------	---------------------------------

B64Encode

Esta función codificará en BASE64 la información recibida en la variable text

```
public String B64Encode (String text, String charSetName)
```

text	Texto a codificar en base 64
charSetName	Codificación de la cadena de texto, UTF-8, ISO-8859-1, etc. OPCIONAL
Cadena de texto codificada en base64	

B64Decode

```
public String B64Decode (String text, String charSetName)
```

text	Texto a decodificado en base 64
charSetName	Codificación de la cadena de texto, UTF-8, ISO-8859-1, etc., que se generará como resultado de decodificación OPCIONAL
Cadena de texto decodificada	

CompressB64Zlib

Esta función comprimirá la información en ZLIB y la codificará en base 64

```
public String CompressB64Zlib (String text)
```

text	Texto a codificado en base 64 y comprimir en Zlib
Cadena de texto codificada en base64 y comprimida en Zlib	

CompressB64ZlibFile

Esta función comprimirá el fichero que recibe como parámetro en ZLIB y lo codificará en BASE64

```
public String CompressB64ZlibFile (String filename)
```

filename	Nombre del fichero a comprimir en ZLIB y codificar en base64
Cadena de texto codificada en base64 y comprimida en Zlib	

CompressZip

Esta función comprimirá el fichero o directorio que recibe como parámetro, en un fichero ZIP

```
public int CompressZip(String fileName, String zipFileName, boolean recursive, boolean deleteSource)
```

filename	Fichero o ruta con los archivos a comprimir
zipFileName	Nombre del fichero donde se comprimirán los ficheros
recursive	Indica si se deben comprimir de forma recursiva todos los directorios o no
deleteSource	Indica si se debe borrar o no el origen de datos
Número de ficheros comprimidos	

ConcatFiles

Esta función concatenará los ficheros que cumplan la mascara maskSource, en el fichero destino destination.

```
public boolean ConcatFiles(String maskSource, String destination, boolean deleteSource, int blockSize)
```

maskSource	Mascara del directorio origen con los ficheros a concatenar
destination	Fichero destino con ruta completa donde se concatenarán los ficheros

deleteSource	Indica si se deben borrar o no los ficheros origen
blockSize	Tamaño de bloque, para realizar la concatenación. Este parámetro, ha de buscar un compromiso entre memoria y velocidad. A mayor tamaño, mayor velocidad, pero consume mas recursos de memoria
true o false en función del éxito o fracaso de la operación	

ConnectHTTP

Esta función establece una conexión HTTP, con la url y los ficheros que se le indican en los parámetros

```

method          = "GET";
urlAddress      = "http://127.0.0.1:8080/services/test/;
requestContent  = "";
requestContentType = "application/text";
requestFile     = "";
responseFile    = "/tmp/response.txt";
user            = "TEST";
password        = "PWDTEST";
sslProtocol     = "TLS";
timeout         = 0;
proxyHost       = "";
proxyPort       = 0;
proxyUser       = "";
proxyPassword   = "";
result = ConnectHTTP( method,
                      urlAddress,
                      requestContent,
                      requestFile,
                      requestContentType,
                      responseFile,
                      user,
                      password,
                      sslProtocol,
                      timeout,
                      proxyHost,
                      proxyPort,
                      proxyUser,
                      proxyPassword);

WriteLog (result)

```

```

public boolean ConnectHTTP( String method,
                           String urlAddress,
                           String requestContent,
                           String requestFile,
                           String requestContentType,
                           String responseFile,
                           String user,
                           String password,
                           String sslProtocol,
                           int timeout,
                           String proxyHost,
                           int proxyPort,
                           String proxyUser,
                           String proxyPassword)

```

method	PUT, POST, GET, HEAD, DELETE
urlAddress	Dirección url, donde se encuentra el servicio
requestContent	Texto con la petición
requestFile	Fichero con la petición
requestContentType	Conten-Type de la petición realizada
responseFile	Fichero donde se escribirá la respuesta en el caso que lo haya (métodos, GET y HEAD)
user	Usuario
password	Contraseña

sslProtocol	Tipo de protocolo SSL -> TLS
timeout	Tiempo máximo de conexión
proxyHost	En caso de utilizar proxy, máquina donde esta el proxy
proxPort	Puerto del servicio de proxy
proxyUser	Usuario del servicio de proxy
proxyPassword	Contraseña del servicio de proxy
Devuelve true o false en función si la operación ha tenido éxito o no	

CopyTextFile

Copia los ficheros de texto que se le indican como origen al destino, si el fichero destino ya existe, se le añadirá el contenido del fichero origen.

```
public boolean CopyTextFile( String source,
                             String destination,
                             String param,
                             String newExtension,
                             boolean deleteDestination,
                             boolean failIfError)
```

source	Máscara con los ficheros origen a copiar
destination	Ruta o fichero destino donde se copiarán los ficheros
param	Sufijo que se le añadirá a los ficheros copiados. OPCIONAL %d1 MmddHHmm %d2 ddHHmm %d3 Hhmm %d4 MmddHHmmss %d5 yyMMdd %d6 Hhmmss %d7 ddMMyy %d8 ddMMyyyy %d9 yyyyMMddHHmmss
newExtension	Nueva extensión de los ficheros copiados. OPCIONAL
deleteDestination	Indica si se debe borrar el destino antes de realizar la copia o no. OPCIONAL
failIfError	Indica si se debe generar un error si falla el proceso de copia. OPCIONAL
Devuelve true o false en función si la operación ha tenido éxito o no	

CreateFolder

Crea el directorio que recibe como parámetro

```
public boolean CreateFolder(String folder)
```

folder	Ruta completa del directorio que se desea crear
Devuelve true o false en función si la operación ha tenido éxito o no	

DeCompressB64Zlib

Esta función decodificará la información comprimida en ZLIB y codificada en base 64

```
public String DeCompressB64Zlib (String text)
```

text	Cadena a comprimida en ZLIB y codificar en BASE 64
Información descomprimida, y decodificada	

DeCompressZip

Descomprime el fichero en formato zip, en la ruta que se le indica

```
public boolean DeCompressZip(String zipFileName,
                             String dirDest,
                             boolean deleteSource,
                             boolean withPaths)
```

zipFileName	Fichero comprimido en formato zip a descomprimir
dirDest	Ruta destino donde se descomprimirá el contenido del fichero zip
deleteSource	Se indica si se debe borrar el fichero origen o no
withPaths	Se indica si se debe realizar la descompresión del fichero con ruta o no
Devuelve true o false en función si la operación ha tenido éxito o no	

DeleteCAB

Este procedimiento eliminará la línea de cabecera y pie de los ficheros de tipo LOT

```
CAB00068EDI ---> Elimina esta fila
AEHORDERSD 93AUNEAN007
BGM220                                148
DT013720050211                        102
DT064 20050211                        102
.....
.....
CLT ---> Elimina esta fila
```

```
public boolean DeleteCAB(String maskSource)
```

maskSource	Mascara con los ficheros a eliminar la cabecera
Devuelve true o false en función si la operación ha tenido éxito o no	

DeleteChar

Borrar los caracteres de los ficheros que cumple la mascara de búsqueda

```
public boolean DeleteChar ( String maskSource,
                           String[] characters)
```

maskSource	Mascara con los ficheros a eliminar la cabecera
characters	Lista de caracteres que se desea borrar, puede ser una letra o su código (#nnn)
Devuelve true o false en función si la operación ha tenido éxito o no	

DeleteFile

Borra los ficheros que cumplen la máscara que se recibe como parámetro

```
public void DeleteFile(String fileMask)
```

fileMask	Mascara con los ficheros a eliminar
-----------------	-------------------------------------

DeleteFileIfEmpty

Borra los ficheros que cumplen la mascara y cuyo tamaño es cero

```
public void DeleteFileIfEmpty(String fileMask)
```

fileMask	Mascara con los ficheros a eliminar
-----------------	-------------------------------------

DeleteFileOld

Borra los ficheros que cumplen la mascara de búsqueda y con una antigüedad mayor que los días que se indican como parámetro

```
public void DeleteFileOld(String fileMask, int daysold, boolean deletetoler)
```

fileMask	Mascara con los ficheros a eliminar
daysold	Días de antigüedad de los ficheros a eliminar
deletetoler	Indica si se deben borrar los directorios o no. OPCIONAL

DeleteLine

Borra las líneas de los ficheros que cumple la mascara de búsqueda


```
public boolean DeleteLine ( String      maskSource,
                           String      criteria,
                           boolean     trimLines,
                           boolean     emptyLines,
                           String      charsetName)
```

maskSource	Mascara para localizar los ficheros que se desan procesar
criteria	Criterio para eliminar las lineas del fichero
trimLines	Quita los caracteres iniciales en blanco de la linea
emptyLines	Elimina las lineas vacías o solo con blancos(32)
charsetName	Nombre del juego de caracteres del fichero origen. OPCIONAL
Devuelve true o false en función si la operación ha tenido éxito o no	

DynamicClientSOAP

Esta función creará un cliente dinámico de webservices, en función wsdl que se recibe como parámetro.

```
public Object[] DynamicClientSOAP( String wsdl,
                                   String username,
                                   String password,
                                   String operationName,
                                   Object[] paramValues)
```

wsdl	URL que contiene la wsdl sobre la que se va operar
username	Usuario para realizar la autentificación básica HTTP. OPCIONAL
password	Contraseña para realizar la autentificación básico HTTP. OPCIONAL
operationName	Nombre del método de la wsdl que se desea ejecutar
paramValues	Parámetros que se pasan a la operación web services
Lista con los valores devueltos con la función	

ExecAndWait

Esta función ejecuta el comando que recibe como parámetro, esperando o no la finalización del mismo, el máximo de milisegundos que se le indica como parámetro.

```
public int ExecAndWait( String application_with_params,
                       boolean waitexecution,
                       int timeout)
```

application_with_params	Aplicación con sus parámetros a ejecutar
waitexecution	True si se quiere que el script, espere a su finalización, false en caso contrario
timeout	Tiempo máximo de espera en milisegundos, a que finalice la ejecución de la aplicación
0 → Finalización normal de la aplicación -1 → Error general -2 → Vence el tiempo máximo de ejecución	

ExecuteFile

Lanza la ejecución de la aplicación que recibe como parámetro.

```
public void ExecuteFile(String fileName)
public void ExecuteFile(String fileName, int mode)
```

fileName	Nombre de la aplicación a ejecutar
mode	Modo de ejecución de la tarea. Existe tres modos de ejecución 0 : Lanza la ejecución del programa y el script continua su ejecución 1 : Lanza la ejecución del programa y espera a que finalice. Valor por defecto. 2 : Lanza la ejecución del programa y espera su finalización cuando aparezca la instrucción del script SYNC

FileExists

Comprueba si el fichero que se recibe como parámetro existe o no

```
public boolean FileExists(String filename)
```

filename	Nombre del fichero con ruta completa
true si existe el fichero false en caso contrario	

FormatFloat

Esta función permite formatear el valor numérico que se recibe a una cadena de texto

```
public String FormatFloat(String pattern, double value)
```

pattern	Patrón que se emplea para formatear el valor numérico 0 Fuerza que se muestre un dígito o 0 # Muestra de forma opcional un dígito , Muestra el separador de miles . Muestra el separador de decimales
value	Valor numérico a formatear
true si existe el fichero false en caso contrario	

FTPGet

Este procedimiento recogerá el/los ficheros que se encuentren en *source* en el servidor de FTP que se recibe como parámetro, y los depositara en *destination*

```
public int FTPGet (      String      host,
                        int          port,
                        String       user,
                        String       password,
                        String       transferType,
                        boolean      passive,
                        String       sourceDir,
                        String       sourceFile,
                        String       destination,
                        boolean      canOverwrite,
                        boolean      delete,
                        int          retries,
                        String       systemKey)
```

host	Máquina en la que se encuentra el servicio de FTP
port	Puerto en el que este escuchando el servicio de FTP
users	Usuario FTP
password	Contraseña
transferType	Tipo de transferencia ASCII o BINARY
passive	Tipo de transferencia, pasiva o activa
sourceDir	Ruta origen de los ficheros a recuperar
sourceFile	Fichero a recuperar
destination	Ruta local en la que se depositarán los ficheros recuperados
canOverwrite	Si en el fichero a recuperar, ya se encuentra en el origen indica si se debe sobrescribir.
delete	Indica si debe borrar los fichero del servidor FTP, una vez descargados
retries	Indica los reintentos que se realizarán en caso de algún fallo realizando el FTP GET. OPCIONAL.
systemKey	Tipo de sistema del servidor FTP. OPCIONAL "UNIX";"VMS";"WINDOWS";"OS/2";"OS/400";"AS/400"; "MVS";"TYPE: L8";"NETWARE";

> 0 Número de documentos recuperados del servidor de FTP

- 1 Error haciendo el GET al servidor de FTP
En caso de producir un error durante el envío de fichero, en la variable Error habrá una descripción del error que se ha producido
- 2 Error estableciendo la conexión con servidor FTP
- 3 Directorio destino no existe
- 4 Directorio origen no existe

FTPPut

Este procedimiento enviará el/los ficheros que se encuentren en *source* al servidor de FTP que se recibe como parámetro

```
public int FTPPut ( String host,
                  int port,
                  String user,
                  String password,
                  String transferType,
                  boolean passive,
                  String source,
                  String destinationPath,
                  String destinationName,
                  boolean append,
                  boolean deleteSource,
                  int retries,
                  boolean renameUpload)
```

host	Máquina en la que se encuentra el servicio de FTP
port	Puerto en el que este escuchando el servicio de FTP
users	Usuario FTP
password	Contraseña
transferType	Tipo de transferencia ASCII o BINARY
passive	Tipo de transferencia, pasiva o activa
source	Origen de los datos a transferir. Bien puede ser un directorio, una ruta con mascara o un fichero
destinationPath	Ruta del servidor FTP en la que se depositarán los ficheros
destinationName	Nombre del fichero destino, en caso de estar vacío se depositarán en el servidor FTP, con el mismo nombre que tienen en el origen
append	Si en el fichero a enviar, ya se encuentra en el servidor FTP indica si se debe sobrescribir o se añade al que ya existe.
deleteSource	Si la transferencia ha tenido éxito se borra el mensaje del origen
retries	Indica el número de reintentos del FTPPut, en caso de error. OPCIONAL
renameUpload	Indica si durante la subida del fichero al servidor FTP, se debe renombrar para evitar ser procesado por SII. OPCIONAL
>0 Número de documentos subidos al servidor de FTP <ul style="list-style-type: none"> -1 Error haciendo el PUT al servidor de FTP En caso de producir un error durante el envío de fichero, en la variable Error habrá una descripción del error que se ha producido -2 Error estableciendo la conexión con servidor FTP -3 Directorio destino no existe 	

getDateFormat

Esta función recupera la fecha actual utilizando el formato especificado en el parámetro *format*, y la zona horaria indicada en el parámetro *tiemzone*

```
public String getDateFormat(String format, String timezone)
```

format	Formato en el que se recuperará la fecha actual			
	Letra	Fecha o Hora	Presentación	Ejemplos
	G	Marcador de Era	Texto	
	y	Año	Año	
				AD
			1996; 96	

	M	Mes in año	Mes	July; Jul; 07	
	w	Semana en el año		Número 27	
	W	Semana en el mes		Número 2	
	D	Día en el año	Número 189		
	d	Día en el mes	Número 10		
	F	Día de la semana en el mes		Número 2	
	E	Día en la semana	Texto	Jueves; Jue	
	a	Am/pm	Texto	PM	
	H	Hora en el día (0-23)		Número 0	
	k	Hora en el día (1-24)		Número 24	
	K	Hora en am/pm (0-11)		Número 0	
	h	Hora en am/pm (1-12)		Número 12	
	m	Minuto en la hora		Número 30	
	s	Segundo en el minuto		Número 55	
	S	Milisegundo	Número 978		
	z	Zona horaria	Zona horaria general		Pacific Standard Time;
	PST; GMT-08:00				
	Z	Zona horaria	RFC 822	-0800	
timezone	Zona horaria que se aplicará a la fecha recuperada GMT+1, GMT-1, etc...				
Fecha actual de la zona horaria indicada					

getError

Si se ha producido un error durante la ejecución del script, este se podrá consultar a través de esta función.

```
public String getError()
```

getLogFileName

Esta función devuelve el nombre del fichero, en el cual el script esta escribiendo su log de trabajo

```
public String getLogFileName()
```

getScriptName

Recupera el nombre del script que esta en ejecución.

```
public String getScriptName()
```

getStartDate

Devuelve la hora de inicio de ejecución del script actual

```
public Date getStartDate()
```

isError

Indica si se ha producido algún error durante la ejecución del script en curso.

```
public boolean isError()
```

GunzipFile

Esta función descomprimirá el fichero en formato GZIP, que recibe como parámetro, escribiendo en el fichero fileout

```
public boolean GunzipFile (String filein,
                          String fileout)
```

filein	Fichero con ruta completa con la información comprimida en gzip, que se desea descomprimir
fileout	Fichero resultado con la información descomprimida
true o false en función del éxito o fracaso de la operación	

MoveTextFile

Mueve los ficheros de texto que se le indican como origen al destino, si el fichero destino ya existe, se le añadirá el contenido del fichero origen.

```
public boolean MoveTextFile(String source,
                           String destination,
                           String param,
                           String newExtension,
                           boolean deleteDestination,
                           boolean failIfError)
```

source	Máscara con los ficheros origen a mover
destination	Ruta o fichero destino donde se moverán los ficheros
param	Sufijo que se le añadirá a los ficheros copiados. OPCIONAL %d1 MmddHHmm %d2 ddHHmm %d3 Hhmm %d4 MmddHHmmss %d5 yyMMdd %d6 Hhmmss %d7 ddMMyy %d8 ddMMyyyy %d9 yyyyMMddHHmmss
newExtension	Nueva extensión de los ficheros movidos. OPCIONAL
deleteDestination	Indica si se debe borrar el destino antes de mover el fichero. OPCIONAL
failIfError	Indica si se debe generar un error en caso de fallo durante el proceso de mover el fichero
Devuelve true o false en función si la operación ha tenido éxito o no	

MoveFile

Mueve el fichero que se le indica en srcFile a destFile

```
public boolean MoveFile(String srcFile, String destFile)
```

srcFile	Fichero origen a mover
destFile	Fichero destino
Devuelve true o false en función si la operación ha tenido éxito o no	

Pos

Devuelve el índice, de la primera ocurrencia de la cadena *searchvalue*, en la cadena *strSource*, empezando en la posición *start*

```
public int Pos(String strSource, String searchvalue, int start)
```

strSource	Cadena de texto sobre la que se realizará la búsqueda
searchvalue	Cadena de texto que se quiere buscar
start	Posición desde la que se iniciará la búsqueda
Devuelve el índice de la primera ocurrencia de la cadena buscada. Sino se encuentra ningún la cadena devolverá -1	

ReplaceChar

Sustituye los caracteres de los ficheros que cumple la mascara de búsqueda

```
public boolean ReplaceChar ( String maskSource,
                           String[] oldChar,
                           String[] newChar)
```

maskSource	Mascara para localizar los ficheros que se desean procesar
oldChar	Lista de caracteres origen. Se puede indicar el código ASCII del carácter de la forma

	#nnn
newChar	Lista de caracteres por los que se reemplazarán los caracteres origen. Se puede indicar el código ASCII del carácter de la forma #nnn
true si la operación tiene éxito false en caso contrario	

ReplaceStr

Sustituye las líneas de los ficheros que cumple la mascara de búsqueda

```
public boolean ReplaceStr ( String      maskSource,
                          boolean     doubleQuote,
                          boolean     recursive,
                          boolean     caseSensitive,
                          String[]    oldStr,
                          String[]    newStr,
                          String      encoding) {
```

maskSource	Mascara para localizar los ficheros que se desean procesar
doubleQuote	Indica si se deben sustituir las comillas simples por comillas dobles
recursive	Recorre de forma recursiva la estructura de directorios
caseSensitive	Indica si el remplazado debe ser sensible a mayúsculas y minúsculas o no
oldStr	Lista de cadenas origen. Se pueden indicar caracteres de la cadena en formato ASCII #nnnn
newStr	Lista de cadenas por las que se reemplazarán las cadenas origen. Se pueden indicar caracteres de la cadena en formato ASCII #nnnn
encoding	Codificación de los ficheros sobre los que se quiere realizar el ReplaceStr
true si la operación tiene éxito false en caso contrario	

SendMail

Este procedimiento enviará un mail con los valores que recibe como parámetro

```
public void SendMail( String server,
                     String fromUser,
                     String toUser,
                     String subject,
                     String text,
                     String attachments,
                     String smtpUser,
                     String smtpPassword)
```

server	Servidor de SMTP al cual se envía el mail
fromUser	Usuario que envía el mail
toUser	Usuario destinatario del mail
subject	Asunto del mail
text	Texto del mail
attachments	Ficheros adjuntos a enviar, separados por ';' También existe la posibilidad del que se pase un mascara sobre la cual iterar y recuperar los ficheros a enviar
smtpUser	Usuario ESMTP
smtpPassword	Contraseña ESMTP

Sleep

Detiene la ejecución del script los milisegundos que se indica

```
public void Sleep(int milliseconds)
```

milliseconds	Milisegundos que se detiene el script
---------------------	---------------------------------------

SplitIDOC

Esta función troceará el IDOC que recibe como parámetro en las partes que se le indique en función de los valores de *maxFileSize* y *maxDocsPerFile*.

```
public boolean SplitIDOC (    String idocFile,
                           String dirDest,
                           int maxFileSize,
                           int maxDocsPerFile,
                           String charsetName)
```

idocFile	Nombre de fichero con ruta completa que contiene el IDOC a trocear
dirDest	Directorio destino donde se depositarán los IDOC's troceados
maxFileSize	Tamaño máximo en bytes de la partes en las que se generarán los IDOCS
maxDocsPerFile	Número máximo de IDOC's por parte generada
charsetName	Juego de caracteres del IDOC origen
true si la operación tiene éxito false en caso contrario	

SplitLOT

Esta función troceara el LOT que recibe como parámetro en las partes que se le indique en función de los valores de *maxFileSize* y *maxDocsPerFile*

```
public boolean SplitLOT (    String lotFile,
                           String dirDest,
                           int maxFileSize,
                           int maxDocsPerFile,
                           String documentLabel,
                           String recordSep,
                           String escapeChar,
                           String charsetName)
```

lotFile	Nombre de fichero con ruta completa que contien el LOT a trocear
dirDest	Directorio destino donde se depositarán los LOT's troceados
maxFileSize	Tamaño máximo en bytes de la partes en las que se generarán los LOT's
maxDocsPerFile	Número máximo de LOT's por parte generada
documentLabel	Etiqueta para identificar el documento
recordSep	Carácter separador de registros
escapeChar	Carácter de escape
charsetName	Juego de caracteres del LOT origen
true si la operación tiene éxito false en caso contrario	

Sync

Espera la finalización de las aplicaciones que se han lanzado a ejecución, mediante la instrucción *ExecuteFile*, en modo 2

```
public void Sync()
```

UncompressFile

Esta función descomprimirá ficehros comprimidos con la función de unix compress escribiendo en el fichero fileout el resultado de la descompresión. El algoritmo que utiliza la función compress de unix es LZW

```
public boolean UncompressFile (    String filein,
                                String fileout)
```

filein	Fichero con ruta completa con la información comprimida en LZW, que se desea descomprimir
fileout	Fichero resultado con la información descomprimida
true si la operación tiene éxito false en caso contrario	

WriteLog

Escribe un texto informativo en el log de ejecución del script

```
public void WriteLog (String text)
```

3.2 ADAPTADOR EDICOM

El Adaptador EDICOM, es un conector de propósito general pensado para conectar las aplicaciones clientes al BUS. Para facilitar la conexión entre la aplicación y el EBIBROKER, el código script que soporta el adaptador, *Javascript*, que ha sido enriquecido con un conjunto de funciones de específicas

3.2.1 CONFIGURACIÓN

La configuración del adaptador se realiza en el fichero de propiedades ***ebiadapter.properties***. Este fichero se encontrará ubicado en la ruta que indique la variable de sistema ***user.dir***, que se indicará en el arranque del adaptador.

PROPIEDAD	DESCRIPCIÓN	VALOR POR DEFECTO
ebiadapter.multilInstance	Indica si el adaptador debe conectarse a distintos orígenes de datos	false
ebiadapter.group	Origen de datos asociado al usuario de conexión	
ebiadapter.user	Usuario de conexión	
ebiadapter.password	Contraseña	
ebiadapter.passwordE	Contraseña encriptada	
ebiadapter.type	Tipo de adaptador	EBIAEDICOM
ebiadapter.name	Nombre con el que se registrará el adaptador en el EBI BROKER	
ebiadapter.domain	Dominio en el que se registra el adaptador	
ebiadapter.application	Aplicación en la que se registra el adaptador	
ebiadapter.url1	URL de conexión al EBI BROKER	
ebiadapter.language	Idioma del adaptador	es
ebiadapter.country	País del adaptador	ES
ebiadapter.deleteDirLogs	Número de días de permanencia de los logs en el directorio	0
ebiadapter.tr	Intervalo en minutos, de reinicio del adaptador, si no hay ningún proceso en ejecución. 0 deshabilitado	0
ebiadapter.tc	Intervalo en minutos de reinicio del adaptador, independientemente, si hay algo en ejecución o no. 0 deshabilitado.	0
ebiadapter.proxyServer	Servidor de PROXY	
ebiadapter.proxyServerPort	Puerto en el que se encuentra el servidor de PROXY	

ebiadapter.proxyUserName	Usuario PROXY	
ebiadapter.proxyPassword	Contraseña del PROXY	
ebiadapter.proxyServerType	Tipo de servidor PROXY	

El adaptador puede funcionar en modo comando, en el cual se le indicará el script que debe ejecutar, y en modo servicio, el cual se conectará al EBI BROKER, y se quedará a la espera de las planificaciones, que le indique el EBI BROKER.

Para indicar si se arranca de un modo u otro, se realizará a través de la variable de sistema **mode**, que se indicará durante el arranque de la aplicación.

Modo comando

```
java -Xmx512m -Xms128m -Duser.language=es -Duser.dir=... -Duser.log=...
-Djava.io.tmpdir=... -Dmode=CMD -Dscript="$scriptname" -jar ebiaedicom.jar
```

Modo servicio

```
java -Xmx512m -Xms128m -Duser.language=es -Duser.dir=... -Duser.log=...
-Djava.io.tmpdir=... -Dmode=SERVER -jar ebiaedicom.jar
```

PARÁMETRO	DESCRIPCIÓN	VALOR POR DEFECTO
java	Máquina virtual de Java (versión 1.6 o superior) que ejecuta el adaptador	
-Xmx	Tamaño máximo de la memoria	512m
-Xms	Tamaño mínimo de la memoria	128m
-Duser.language	Idioma en el que funciona el adaptador	es
-Duser.dir	Directorio base del adaptador, donde se encontrarán los ficheros de configuración	\$PWD ó %cd%
-Djava.io.tmpdir	Directorio donde se dejarán los ficheros temporales que utilice el adaptador	\$PWD/tmp ó %cd%\tmp
-Duser.log	Directorio donde se depositarán los ficheros de log	\$PWD/log/ ó %cd%\log\
-Dmode	Modo de arranque del adaptador CMD → Modo comando SERVER → Modo servicio	
-Dscript	Nombre del script que va ejecutar el adaptador (solo aplicable en el modo comando)	
-Dproperties	Nombre del fichero de propiedades con el que arrancará el adaptador	ebiadapter.properties
-jar ebiaedicom.jar	Jar que contiene el adaptador	

3.2.2 FUNCIONES

DsXml

Tras la ejecución de la función EBIRemoteSQL, contendrá el XML con el resultado de la consulta realizada.

```
public String DsXml()
```

EBIAcquireApplication

Esta función intenta adquirir una aplicación, para el acceso exclusivo a la misma. Si tiene éxito, devolverá 1

```
public int EBIAcquireApplication (int timeout)
```

timeout	Tiempo máximo que puede estar la aplicación bloqueada sin actividad (segundos)
-1 Error 0 No se puede adquirir la aplicación 1 Aplicación adquirida con éxito	

EBIBeginTrans

Esta función iniciará una transacción sobre los documentos almacenados en el EBI BROKER

```
public String EBIBeginTrans (      int timeoutSec,  
                                String description)
```

timeoutSec	Tiempo máximo de duración de la transacción. Si es cero, indicará que no existe tiempo máximo de duración
description	Descripción asociada a la transacción
Identificador de la transacción -1 Error	

EBIChangeDocumentState

Esta función cambiará el estado del documento que recibe por referencia

```
public boolean EBIChangeDocumentState (      String messageIds,  
                                           int state,  
                                           String transactionId)
```

messageIds	Identificador/es del mensaje/s a notificar. Si son varios los mensajes a notificar estos irán separados por ' '
state	Nuevo estado del documento. <div> Publicado = 1 Suscrito = 2 Confirmado = 3 Bloqueado = 4 PublicaconConfirmadaParcialmente = 5 PublicacionConfirmada = 6 PublicacionNotificada = 7 PendienteConfirmacion = 8 Rechazado = 10; PublicacionRechazada = 11; PublicacionRechazadaNotificada = 12; PublicacionParcialmenteRechazada = 13; </div>
transactionId	Identificador de transacción asociada al cambio de estado del documento. OPCIONAL
True o false, en función del éxito de la operación	

EBIChangeDocumentStateV2

Esta función cambiará el estado del documento que recibe por referencia

```
public boolean EBIChangeDocumentStateV2 (      String[] messageIds,  
                                           int state,  
                                           String transactionId)
```

messageIds	Array con los identificadores de documento qu se desean cambiar
state	Nuevo estado del documento. <div> Publicado = 1 Suscrito = 2 Confirmado = 3 Bloqueado = 4 </div>

	<div> PublicaconConfirmadaParcialmente = 5 PublicacionConfirmada = 6 PublicacionNotificada = 7 PendienteConfirmacion = 8 Rechazado = 10; PublicacionRechazada = 11; PublicacionRechazadaNotificada = 12; PublicacionParcialmenteRechazada = 13; </div>
transactionId	Identificador de transacción asociada al cambio de estado del documento. OPCIONAL
True o false, en función del éxito de la operación	

EBIChangeMessageReference

Esta función cambiará la referencia de los documentos que reciben como parámetros

```
public boolean EBIChangeMessageReference (
    String messageIds,
    String reference,
    String transactionId) throws EBIBrokerServerException
```

messageIds	Identificador/es del mensaje/s a notificar. Si son varios los mensajes a notificar estos irán separados por " "
reference	Nuevo valor de la referencia
transactionId	Identificador de transacción asociada al cambio de referencia del documento. OPCIONAL
True o false, en función del éxito de la operación	

EBIChangeMessageReferenceV2

Esta función cambiará la referencia de los documentos que reciben como parámetros

```
public boolean EBIChangeMessageReferenceV2 (
    String[] messageIds,
    String reference,
    String transactionId) throws EBIBrokerServerException
```

messageIds	<u>Identificadores</u> a cambiar la referencia
reference	Nuevo valor de la referencia
transactionId	Identificador de transacción asociada al cambio de referencia del documento. OPCIONAL
True o false, en función del éxito de la operación	

EBICommitTrans

Esta función realiza un commit de la transacción que recibe como parámetro

```
public boolean EBICommitTrans (String transactionId)
```

transactionId	Identificador de transacción
True o false, en función del éxito de la operación	

EBIConfigApp

Esta función recuperará el valor del parámetro de configuración que recibe en la llamada de la función

```
public String EBIConfigApp ( String application,
    String qualifier)
```

application	Aplicación de la cual se quiere recuperar el calificador
qualifier	Calificador del cual se quiere recuperar el parámetro
Valor del calificador que se recibe como parámetro	

EBIConfigDomain

Esta función recupera el valor del calificador del dominio que recibe como parámetro

```
public String EBIConfigDomain (String qualifier)
```

qualifier	Calificador del cual se quiere recuperar el parámetro
Valor del calificador que se recibe como parámetro	

EBIConfirmedPublications

Esta función devolverá en una cadena los identificadores de todas la publicaciones que estén en estado confirmado. Los identificadores de la publicaciones irán separados por el símbolo '|'

```
public String EBIConfirmedPublications (    String domain,
                                           String application,
                                           int extendedInfo)
```

domain	Aplicación de la cual se quiere recuperar el calificador. OPCIONAL
application	Calificador del cual se quiere recuperar el parámetro. OPCIONAL
extendedInfo	Indica si se quiere devolver información extendida sobre sobre los mensajes confirmados. OPCIONAL <div> 0 --> sin información extendida 1 --> con información extendida [MENSAJE]~[APPSUB]~[APPDST]~[FECHA]~[REFERENCIA]~[TIPO]~[COUNTSU B]~[TAMANYO] </div>
Cadena con todas las aplicaciones en estado confirmado, o -1 en caso de error. Estas irán separadas por el símbolo ' '	

EBIConfirmedPublicationsv2

Esta función devolverá un array de objetos del tipo *EBIDocumentWS*, con las publicaciones que estén en estado confirmado

```
public String EBIConfirmedPublicationsV2 (
    String domain,
    String application,
    int extendedInfo) throws EBIBrokerServerException
```

domain	Aplicación de la cual se quiere recuperar el calificador. OPCIONAL
application	Calificador del cual se quiere recuperar el parámetro. OPCIONAL
extendedInfo	Indica si se quiere devolver información extendida sobre sobre los mensajes confirmados. OPCIONAL <div> 0 --> sin información extendida 1 --> con información extendida </div>
Un array de objetos del tipo <i>EBIDocumentWS</i>	

EBIConfirmSubscription

Confirma el identificador de mensaje que recibe como parámetro

```
public boolean EBIConfirmSubscription(String idMessage)
```

idMessage	Identificador del mensaje a confirmar
True o false en función que todo haya funcionado correctamente o no	

EBIConfirmSubscriptions

Confirma los mensajes del directorio que recibe como parámetro. Será muy importante que estos ficheros, conserven el nombre original ya que en este esta el identificador de documento que se quiere confirmar.

Si en el directorio que se le pasa como parámetro, se encuentra el fichero .ids. \$ids\$eso será porque se ha realizado una suscripción conservando el nombre del fichero por lo tanto se procesarán las confirmaciones de este fichero.

```
public boolean EBICConfirmSubscriptions(String directory)
```

directory	Directorio del cual se quieren confirmar los mensajes
True o false en función que todo se haya confirmado correctamente o no	

EBICriticalSectionAcquire

Esta función adquiere la sección crítica, por un tiempo máximo timeOut (en segundos)

```
public int EBICriticalSectionAcquire(    String id,
                                       String description,
                                       int timeOut)
```

id	Identificador único de la sección crítica
description	Descripción de la sección crítica
timeOut	Tiempo máximo de adquisición del semáforo
Si adquisición del semáforo tiene éxito devolverá 1, si no puede conseguir el bloqueo devolverá 0, y -1 en caso de error	

EBICriticalSectionRelease

Esta función que libera la sección crítica que recibe como parámetro

```
public int EBICriticalSectionRelease(String id)
```

id	Identificador de la sección crítica
0 si se consigue liberar el semáforo, -1 en caso de error	

EBICriticalSectionUpdateTimeStamp

Este procedimiento, actualiza la marca de tiempo de la sección crítica, para no ser eliminada de forma automática por el sistema

```
public int EBICriticalSectionUpdateTimeStamp(String id, int timeOut)
```

id	Identificador de la sección crítica
timeOut	Tiempo en segundos, en los que se aumenta la marca
0 si se consigue liberar el semáforo, -1 en caso de error	

EBIDocumentDeleteOld

Esta función eliminará todos los documentos del esquema schema, que están el EBIBROKER mas de minuteOld minutos

```
public int EBIDocumentDeleteOld(    String schema,
                                    int minuteOld,
                                    boolean onlyFinnalState){
```

schema	Esquema del que se quieren eliminar los documentos
minuteOld	Número de minutos de permanencia en el EBI BROKER
onlyFinnalState	Solo se eliminan los documentos que están en estado final
Número de documentos borrados, -1 en caso de error	

EBIDocumentList

Esta función devolverá una lista con los ID's de los documentos que cumplan la condición de búsqueda. Si se dejan el parámetro de búsqueda vacío, se interpretará

como que se quieren recuperar todos los documentos, sin aplicar ese criterio de búsqueda.

```
public String EBIDocumentList (    String schema,
                                   String messageIds,
                                   String reference,
                                   int    type)
```

schema	Esquema de los mensajes
messageIds	Identificador del documento a recuperar. Se pueden indicar varios identificadores de mensajes, separados por ' '
reference	Referencia de los mensajes
type	Tipo del mensaje a recuperar. (1) Mensaje publicado en el EBI BROKER (2) Mensaje suscrito en el EBI BROKER (pendiente de ser recogido por su destinatario) (3) El destinatario del mensaje ha recogido el mensaje (4) El mensaje se ha sido bloqueado por el EBI BROKER (5) alguna de las suscripciones asociadas a la publicación ha sido recogida por el destinatario (6) Todas las suscripciones asociadas a la publicación han sido recogidas por sus destinatarios (7) El cliente que ha realizado la publicación ha sido notificado de la entrega correcta de las suscripciones (8) El mensaje ha si recogido por el receptor y esta pendiente su confirmación (9) Indica que el mensaje esta siendo procesado por el EBIMapFormat (10) El destinatario del mensaje a rechazado el mensaje (11) Publicación rechazada por el destinatario del mensaje (12) El publicador del mensaje ha sido notificado que sus publicación ha sido rechazada (13) La publicación ha sido parcialmente rechazada (14) Indica que la publicación esta duplicada y por lo tanto no se ha generado la suscripción
Lista con los mensajes recuperados de la base de datos, formato SDF (separadas por ,)	

EBIDocumentListV2

Esta función devolverá una lista con los objetos EBIDocumentWS, que cumplan la condición de búsqueda. Si se dejan el parámetro de búsqueda vacío, se interpretará como que se quieren recuperar todos los documentos, sin aplicar ese criterio de búsqueda.

```
public EBIDocumentWS[] EBIDocumentListV2 ( String    schema,
                                              String[]  messageIds,
                                              String    reference,
                                              int       type)
```

schema	Esquema de los mensajes
messageIds	Array con los identificadores de documentos a recuperar
reference	Referencia de los mensajes
type	Tipo del mensaje a recuperar (1) Mensaje publicado en el EBI BROKER (2) Mensaje suscrito en el EBI BROKER (pendiente de ser recogido por su destinatario) (3) El destinatario del mensaje ha recogido el mensaje (4) El mensaje se ha sido bloqueado por el EBI BROKER (5) alguna de las suscripciones asociadas a la publicación ha sido recogida por el destinatario (6) Todas las suscripciones asociadas a la publicación han sido recogidas por sus destinatarios (7) El cliente que ha realizado la publicación ha sido notificado de la entrega correcta de las suscripciones (8) El mensaje ha si recogido por el receptor y esta pendiente su confirmación (9) Indica que el mensaje esta siendo procesado por el EBIMapFormat (10) El destinatario del mensaje a rechazado el mensaje (11) Publicación rechazada por el destinatario del mensaje (12) El publicador del mensaje ha sido notificado que sus publicación ha sido rechazada

	(13) La publicación ha sido parcialmente rechazada (14) Indica que la publicación esta duplicada y por lo tanto no se ha generado la suscripción
Array de objetos <i>EBIDocumentWS</i> , que cumplan el criterio de búsqueda	

Ejemplo

```
WriteLog ("EBIDocumentListV2 : ");
var miArray = new Array();
miArray = EBIDocumentListV2("FACTURA1_TESTA", [], "201012161627", 1);

WriteLog ("EBIDocumentListV2 size: " + miArray.length);

for (i=0;i<miArray.length;i++){
    WriteLog (miArray[i].getId());
    WriteLog (miArray[i].getDomain());
    WriteLog (miArray[i].getApplication());
    WriteLog (miArray[i].getMessage());
    WriteLog (miArray[i].getDate());
    WriteLog (miArray[i].getSize());
}
```

EBIEquiv

Convierte en valor «value» mediante la lista de equivalencia «list». El tipo de conversión «type» indica si la conversión es normal(0) o inversa(1). Ejemplo: Si tenemos una lista LISTA1 con los valores 1111|2222

```
EBIEquiv("1111",LISTA1,0) Devuelve 2222.
EBIEquiv("2222",LISTA1,1) Devuelve 1111.
```

```
public String EBIEquiv(String value, String list, int type)
```

value	Valor a buscar en la lista y convertir
list	Lista de equivalencia con el valor a buscar
type	Tipo de conversión a realizar en la lista de equivalencia
resultado de la conversión del valor en la lista	

```
public String EBIEquiv(String value, String list, String fieldKey, String fieldResult)
```

value	Valor a buscar en la lista y convertir
list	Lista de equivalencia con el valor a buscar
fieldKey	Campo por el cual se realiza la búsqueda. Existe la posibilidad realizar la búsqueda por varios campos. Tanto estos como sus valores, irán separados por el carácter #
fieldResult	Campo resultado que se devuelve
resultado de la conversión del valor en la lista	

EBIExecuteTemplate

Esta función ejecutará la PLANTILLA indicada en <templateName> previa adaptación (macro sustitución) los parámetros <parameters>.

```
public boolean EBIExecuteTemplate(String templateName, String parameters)
```

templateName	Es el nombre de la plantilla a ejecutar. Las plantillas se almacenan en el repositorio del EBI BROKER. Una plantilla, es una secuencia de sentencias SQL, con una serie de variables que se macro sustituirán por parámetros.
parameters	Es una lista de parámetros con formato. Param1=valor1~ ...~Paramn=Valorn
True o false en función del éxito de la operación	

EBIGetCounterValue

Devuelve el valor del contador que se recibe como parámetro y que esta almacenado en el repositorio del ebiMapES.

```
public int EBIGetCounterValue(String)
```

counter	Nombre del contador, del cual se quiere recuperar su valor.
Valor de contador ó -1 en caso que no exista o error.	

EBIGetMessage

Esta función obtendrá el mensaje, asociado al identificador de mensaje, que recibe como parámetro

```
public int EBIGetMessage ( String messageId,
                          String folder,
                          int messageType){
```

messageId	Identificador del mensaje a recuperar
folder	Directorio destino, en el cual se depositará el mensaje
messageType	En el caso de ser una suscripción, se puede decidir que mensaje, se desea recuperar 0 : Mensaje origen 1 : Mensaje destino (resultado de la transformación) 2 : Ambos, mensaje origen, y resultado de la transformación
Como resultado devolverá el número de mensajes recuperados -1 --> Error general -2 --> Mensaje no encontrado -3 --> Directorio destino no existe -4 --> Tipo no soportado	

EBIGetRepositoyItem

Recupera el objeto del repositorio que se indica como parámetro

```
public String EBIGetRepositoyItem ( String directory,
                                   String name,
                                   String type)
```

directory	Directorio donde se recuperará el objeto del repositorio
name	Nombre del objeto a recuperar
type	Tipo del objeto a recuperar
Nombre con ruta completa donde se ha recuperado el objeto o -1 en caso de error	

EBIItemCount

Esta función devuelve el número de elementos que hay en la lista *EBIItemList*.

```
public int EBIItemCount()
```

Número de elementos de la lista *EBIItemList*

EBIItemList

Tras la ejecución de *EBISuscriptions*, *EBISusAndTransform*, *EBISubsAndPrint*, *EBIDocumentList*, ó *EBIGetMessage*, *EBIItemList*, almacenará una lista, con los mensajes recuperados

Esta lista almacenará los valores de:

```
0 --> [IDENTIFICADOR DEL MENSAJE]
1 --> [RUTAS, CON LOS FICHEROS DESCARGADOS]
2 --> [DOMINIO]
3 --> [APLICACIÓN]
4 --> [MENSAJE]
```



```

5 --> [APPSUB]
6 --> [APPDST]
7 --> [FECHA]
8 --> [REFERENCIA]
9 --> [TIPO]
10 --> [COUNTSUB]
11 --> [TAMANYO]

```

```

var item =new Array();
for (i=0;i<EBIItemCount();i++){
    item = EBIItemList(i);
    for (j=0;j<item.length;j++){
        WriteLog ("EBIItemList result: " + item[j]);
    }
}

```

```
public String[] EBIItemList (int index)
```

index	Índice del elemento a recuperar
Array con los valores recuperados	

EBIItemListElement

Tras la ejecución de EBISuscriptions, EBISusAndTransform, EBISubsAndPrint, EBIDocumentList, ó EBIGetMessage, EBIItemList, almacenará una lista, con los mensajes recuperados

```
public String EBIItemListElement (int index, int element)
```

index	Índice del elemento a recuperar
element	Elemento de que se quiere consultar 0 --> [IDENTIFICADOR DEL MENSAJE] 1 --> [RUTAS, CON LOS FICHEROS DESCARGADOS]ç 2 --> [DOMINIO] 3 --> [APLICACIÓN] 4 --> [MENSAJE] 5 --> [APPSUB] 6 --> [APPDST] 7 --> [FECHA] 8 --> [REFERENCIA] 9 --> [TIPO] 10 --> [COUNTSUB] 11 --> [TAMANYO]
Valor de elemento consultado	

EBIList2Array

Esta función convierte una lista EBI, un Array multidimensional de *JavaScript* para que luego este pueda ser recorrido desde este lenguaje de programación

```

result = EBIList2Array("comprador");
for (i=0;i<result.length;i++){
    for (j=0;j<result[i].length;j++){
        WriteLog ("EBIList2Array result: " + result[i][j]);
    }
}

```

```
public String[][] EBIList2Array (String list)
```

list	Lista de equivalencia a convertir en un array de JavaScript
Array con la lista de equivalencia	

EBIMessageByReference

Esta función recuperará el documento que se identifica por la referencia y la aplicación origen que recibe como parámetro.

```
public EBIDocumentWS EBIMessageByReference(
    String reference,
    String applicationSource)
```

reference	Referencia del documento
applicationSource	Aplicación origen del documento
Objeto del tipo EBIDocumentWS, que cumple el criterio de búsqueda	

EBIMessageID

Esta función obtendrá el identificador del mensaje, de la referencia que recibe como parámetro

```
public String EBIMessageID(String reference, String applicationSource)
```

reference	Referencia del mensaje, del cual se quiere recuperar el identificador
applicationSource	Aplicación origen que publicó la referencia. OPCIONAL.
Identificador del mensaje, o -1 en caso de error	

EBIMessageReference

Esta función obtendrá la referencia del identificador de mensaje que recibe como parámetro

```
public String EBIMessageReference( String messageId,
    String domain,
    String application)
```

messageID	Referencia del mensaje, del cual se quiere recuperar el identificador
domain	Aplicación origen que publicó la referencia. OPCIONAL.
application	Aplicación de los documentos a obtener la referencia. OPCIONAL.
Referencia del mensaje, o -1 en caso de error	

EBINextCounterValue

Devuelve el valor del contador incrementando en una unidad su valor. Si el contador no existe, crea un nuevo contador con valor 1.

```
public int EBINextCounterValue(String counter)
```

counter	Nombre del contador a incrementar
domain	Aplicación origen que publicó la referencia. OPCIONAL.
application	Aplicación de los documentos a obtener la referencia
El valor del contador incrementado, -1 en caso de error.	

EBINotifiedPublication

Esta función actualizará el estado de la publicación a "Publicación notificada"

```
public boolean EBINotifiedPublication (String messageIds)
```

messageIds	Identificador/es del mensaje/s a notificar. Si son varios los mensajes a notificar estos irán separados por ' '
True o false en función del éxito de la operación	

EBINotifiedPublicationV2

Esta función actualizará el estado de la publicación a "Publicación notificada"

```
public boolean EBINotifiedPublicationV2 (String[] messageIds)
```

messageIds	Array con los identificadores de los mensajes a notificar
True o false en función del éxito de la operación	

EBINotifiedRejectedPublication

Esta función actualizará el estado de la publicación a "Publicación rechazada notificada"

```
public boolean EBINotifiedRejectedPublication (String messageIds)
```

messageIds	Identificador/es del mensaje/s a notificar el rechazo. Si son varios los mensajes a rechazar estos irán separados por ' '
True o false en función del éxito de la operación	

EBINotifiedRejectedPublicationV2

Esta función actualizará el estado de la publicación a "Publicación rechazada notificada"

```
public boolean EBINotifiedRejectedPublicationV2 (String[] messageIds)
```

messageIds	Array con los identificadores de los mensajes a notificar el rechazo.
True o false en función del éxito de la operación	

EBIPOPAndPublish

Esta función se descargará un mensaje de un origen POP3 o IMAP, publicándolo ese mensaje en el EBI. Si la publicación del mensaje tiene éxito, el mensaje se borrará de la cuenta

```
public String EBIPOPAndPublish(
    String host,
    int port,
    String provider,
    String username,
    String password,
    int maxmessages,
    boolean parse,
    String schema,
    String destination,
    int duplicate )
```

host	Dirección donde se encuentra el servidor pop
port	Puerto donde se encuentra el servidor pop
provider	Protocolo usado (pop3, pop3s, imap, imaps).
username	Nombre de usuario para la conexión al servidor de correo
password	Contraseña para la conexión al servidor de correo
maxmessages	Número máximo de mensajes a descargar
parse	Indica si se desea procesar el contenido del mensaje, extrayendo las partes del documento.
schema	Esquema con el que se publicarán los mensajes en el EBI BROKER
destination	Destino de los mensajes
duplicate	Indica la acción a tomar con las publicaciones duplicadas 0: dupIgnore. Se generará una nueva publicación aunque esta ya este duplicada 1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH 2: dupError. Se devuelve un código de error. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación
Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los	

identificadores de los documentos publicados separados por '|'

- 1 = Error
- 2 = Publicación no autorizada
- 3 = Destino de la publicación desconocido
- 4 = Documento incorrecto según la validación sintáctica de su interfaz
- 5 = Aplicación destino no acepta la suscripción
- 7 = Usuario no autorizado a publicar
- 8 = Publicación duplicada
- 10= Esquema no encontrado
- 20= Error recuperando los mensajes del servidor POP
- 21= No se ha localizado carpeta INBOX

EBIPublish

Realiza la publicación del mensaje o máscara que recibe como parámetro en el EBI BROKER

Se puede configurar a nivel de aplicación, si los mensajes que se envían al EBI-BROKER, deben ser troceados por el esquema, previamente a su publicación, generando tantas publicaciones como mensajes genere el recorrido del mensaje enviado.

MENSAJE	CALIFICADOR	VALOR1	VALOR2	VALOR3
EDI_INVOIC_D_96A_UN_EAN008_1	SPLIT_MESSAGE	TRUE		

```
public String EBIPublish(
    String message,
    String schema,
    String destination,
    boolean wait,
    boolean delete,
    int timeout,
    String copyDestination,
    String param,
    String reference,
    int duplicate,
    boolean ebiMapFormat,
    boolean keepFileName,
    boolean publishEmptyFiles,
    String ebiMapFormatFilter)
```

message	(1) Ruta donde se encuentran los ficheros a publicar. En este caso el nombre de los ficheros se extrae del interfaz (2) Nombre del fichero a publicar (3) Mascara con los ficheros a publicar
schema	Esquema con el que se publicarán los mensajes en el EBI BROKER
destination	Aplicación destino de los mensajes
wait	Indica si se debe esperar o no la recolección de todos los mensajes publicados
delete	Indica si se deben borrar o no los documentos publicados
timeOut	Tiempo máximo de espera en segundos.
copyDestination	Directorio en el cual se quiere realizar una copia de los ficheros
param	Formato con el cual se quiere hacer la copia
reference	Referencia con la que se publicará el documento
duplicate	Indica la acción a tomar con las publicaciones duplicadas 0: dupIgnore. Se generará una nueva publicación aunque esta ya este duplicada 1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH 2: dupError. Se devuelve un código de error. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro

	de el mismo dominio y aplicación
ebiMapFormat	Indica si el IEAD debe recorrer sus ficheros y generar el formato interno de EBI, para realizar la publicación de los mismos
keepFileName	Parámetro obsoleto. Valor por defecto true.
publishEmptyFiles	Indica si se permite la publicación de ficheros vacíos o no
ebiMapFormatFilter	En el caso de estar activado el EBIMapFormat, el esquema realizará el recorrido, aplicando el filtro que se recibe en este parámetro
<p>Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los identificadores de los documentos publicados separados por ' '</p> <p>-1 = Error -2 = Publicación no autorizada -3 = Destino de la publicación desconocido -4 = Documento incorrecto según la validación sintáctica de su interfaz -5 = Aplicación destino no acepta la suscripción -6 = Vence el tiempo de espera de recogida de las publicaciones -7 = Usuario no autorizado a publicar -8 = Publicación duplicada -10 = Esquema no encontrado -11 = Error realizando las copias de los ficheros -12 = Problema generando las suscripciones -16 = Tamaño de la publicación excedido -21 = El directorio origen de la publicación no existe o no está accesible -22 Aplicación destino se encuentra en la lista de aplicaciones denegadas. -23 Aplicación origen no se encuentra en la lista de aplicaciones permitidas -24 Aplicación origen se encuentra en la lista de aplicaciones denegadas. -25 Error troceando el documento</p>	

EBIPublishDB

Realiza la publicación de la base de datos que representa el esquema que recibe como parámetro. La publicación de la base de datos implica:

1. Recorrer la base de datos acorde al IEAD que se recibe como parámetro.
2. Durante el recorrido se generará el mensaje EBIMapFormat con el contenido de la base de datos
3. Se realiza la publicación de este mensaje
4. Si la publicación tiene éxito, se realizarán las operaciones finales en la base de datos (borrado o renombrado)

```
public String EBIPublishDB( String schema,
                           String destination,
                           boolean wait,
                           boolean delete,
                           String update,
                           int timeout,
                           String reference,
                           int duplicate,
                           String filter)
```

schema	Esquema que representa la estructura de base de datos a publicar
destination	Aplicación destino del mensaje
wait	Indica si se debe esperar o no la recolección de todos los mensajes publicados
delete	Indica si se deben borrar o no los registros origen
update	Indica si se debe actualizar alguna campo del los registro destino
timeout	Tiempo máximo de espera en segundos.
reference	Referencia con la que se publicará el documento

duplicate	Indica la acción a tomar con las publicaciones duplicadas 0: dupIgnore. Se generará una nueva publicación aunque esta ya este duplicada 1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH 2: dupError. Se devuelve un código de error. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación
ebiMapFormat	Indica si el IEAD debe recorrer sus ficheros y generar el formato interno de EBI, para realizar la publicación de los mismos
keepFileName	Parámetro obsoleto. Valor por defecto true.
publishEmptyFiles	Indica si se permite la publicación de ficheros vacíos o no
ebiMapFormatFilter	En el caso de estar activado el EBIMapFormat, el esquema realizará el recorrido, aplicando el filtro que se recibe en este parámetro
Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los identificadores de los documentos publicados separados por ' ' -1 = Error -2 = Publicación no autorizada -3 = Destino de la publicación desconocido -4 = Documento incorrecto según la validación sintáctica de su interfaz -5 = Aplicación destino no acepta la suscripción -7 = Usuario no autorizado a publicar -8 = Publicación duplicada -10= Esquema no encontrado -11= Fallo recuperando el esquema	

EBIPublishString

Realiza la publicación de la cadena message que recibe como parámetro

```
public String EBIPublishString (    String message,
                                String schema,
                                String destination,
                                boolean wait,
                                int    timeOut,
                                String reference,
                                int    duplicate)
```

message	Cadena que se quiere publicar en el EBI BROKER
schema	Tipo del mensaje que corresponde con el documento
destination	Aplicación destino del mensaje
wait	Indica si se debe esperar o no la recolección de todos los mensajes publicados
timeOut	Tiempo máximo de espera en segundos.
reference	Referencia con la que se publicará el documento
duplicate	Indica la acción a tomar con las publicaciones duplicadas 0: dupIgnore. Se generará una nueva publicación aunque esta ya este duplicada 1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH 2: dupError. Se devuelve un código de error. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación
Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los identificadores de los documentos publicados separados por ' ' -1 = Error -2 = Publicación no autorizada -3 = Destino de la publicación desconocido -4 = Documento incorrecto según la validación sintáctica de su interfaz -5 = Aplicación destino no acepta la suscripción -7 = Usuario no autorizado a publicar -8 = Publicación duplicada	

-10= Esquema no encontrado

EBIPublishVFS

Realiza la publicación del mensaje o máscara que recibe como parámetro en el repositorio de la base de datos.

El parámetro url, representa un sistema virtual de ficheros, pudiendo ser este sistema de fichero uno de los siguientes: FTP, fichero locales y SFTP

```
public String EBIPublishVFS( String url,
                             String schema,
                             String destination,
                             boolean wait, //
                             boolean delete,
                             int timeout,
                             String urlCopyDestination,
                             String param,
                             String reference,
                             int duplicate,
                             String[] vfsOptions)
```

url	<p>(1) Ruta virtual donde se encuentran los ficheros a publicar. En este caso el nombre de los ficheros se extrae del interfaz</p> <p>(2) Ruta virtual con nombre del fichero a publicar</p> <p>(3) Ruta virtual con máscara con los ficheros a publicar</p> <p>Local Files Proporciona acceso a los ficheros, en sistema local de archivos</p> <pre>URI Format [file:///] absolute-path file:///home/someuser/somedir file:///C:/Documents and Settings</pre> <p>HTTP and HTTPS Provee acceso a los fichero a través de un servidor HTTP</p> <pre>URI Format http://[username [: password]@] hostname [: port][absolute-path] https://[username [: password]@] hostname [: port] [absolute-path] http://somehost:8080/downloads/somefile.jar http://myusername@somehost/index.html</pre> <p>FTP Proporciona acceso la los ficheros en un servidor FTP.</p> <pre>URI Format ftp://[username [: password]@] hostname [: port][absolute-path] ftp://myusername:mypassword@somehost/pub/downloads/somefile.t gz</pre> <p>SFTP Proporciona acceso a los ficheros en un servidor SFTP</p> <pre>URI Format sftp://[username [: password]@] hostname [: port][absolute-path] sftp://myusername:mypassword@somehost/pub/downloads/somefile. tgz</pre>
schema	Tipo del mensaje que corresponde con el documento
destination	Aplicación destino del mensaje
wait	Indica si se debe esperar o no la recolección de todos los mensajes publicados
delete	Indica si se deben borrar o no los documentos publicados
timeOut	Tiempo máximo de espera en segundos.
urlCopyDestination	Directorio en el cual se quiere realizar una copia de los ficheros
param	Formato con el cual se quiere hacer la copia
reference	Referencia con la que se publicará el documento
duplicate	<p>Indica la acción a tomar con las publicaciones duplicadas</p> <p>0: dupIgnore. Se generará un nueva publicación aunque esta ya este duplicada</p> <p>1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador del la publicación que tiene el mismo HASH</p>

	2: dupError. Se devuelve un código de error. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación
vfsOptions	<p>Opciones de configuración asociadas al sistema de ficheros</p> <pre> setPassiveMode --> true/false setSys --> Para fijar el sistema de ficheros FTP SYST_UNIX UNIX SYST_NT WINDOWS SYST_OS2 OS/2 SYST_OS400 OS/400 SYST_VMS VMS SYST_MVS MVS SYST_NETWARE NETWARE setTimeout --> Timeout en milisegundos setProxyHost --> Servidor Proxy setProxyPort --> Puerto Proxy setProxyType --> Tipo de proxy HTTP vs SOCKS setUsername --> Usuario proxy setPassword --> Contraseña proxy setDomain --> Dominio proxy setUserDirIsRoot --> Determina si el directorio raíz es el del usuario o el del sistema. Valor por defecto true </pre>
<p>Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los identificadores de los documentos publicados separados por ' '</p> <p>-1 = Error -2 = Publicación no autorizada -3 = Destino de la publicación desconocido -4 = Documento incorrecto según la validación sintáctica de su interfaz -5 = Aplicación destino no acepta la suscripción -7 = Usuario no autorizado a publicar -8 = Publicación duplicada -10= Esquema no encontrado</p>	

EBIPublishXSLT

Realiza la publicación del mensaje, que cumple la máscara del esquema que recibe como parámetro, aplicándole previamente la transformación XSLT que se indica en el parámetro *xsltTransform*. La función localizará en la ruta *directory*, los ficheros que indique el esquema, y a estos se les aplicará la transformación XSLT (que se ejecutará en el proceso en local) y el resultado de esta se publicará en el EBI BROKER.

```

public String EBIPublishXSLT(
    String directory,
    String schema,
    String xsltTransform,
    String destination,
    boolean wait,
    boolean delete,
    int timeout,
    String directoryCopy,
    String param,
    String reference,
    int duplicate)

```

directory	Directorio origen de datos
schema	Esquema con el que se publicará el resultado de la transformación XSLT, y que contiene los ficheros origen
xsltTransform	Transformación XSLT a aplicar
destination	Aplicación destino del mensaje
wait	Indica si se debe esperar o no la recolección de todos los mensajes publicados
delete	Indica si se deben borrar o no los documentos publicados
timeOut	Tiempo máximo de espera en segundos.
directoryCopy	Directorio en el cual se quiere realizar una copia de los ficheros

param	Formato con el cual se quiere hacer la copia
reference	Referencia con la que se publicará el documento
duplicate	Indica la acción a tomar con las publicaciones duplicadas 0: dupIgnore. Se generará una nueva publicación aunque esta ya este duplicada 1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH 2: dupError. Se devuelve un código de error. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación

Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los identificadores de los documentos publicados separados por '|'

- 1 = Error
- 2 = Publicación no autorizada
- 3 = Destino de la publicación desconocido
- 4 = Documento incorrecto según la validación sintáctica de su interfaz
- 5 = Aplicación destino no acepta la suscripción
- 7 = Usuario no autorizado a publicar
- 8 = Publicación duplicada
- 10= Esquema no encontrado
- 11= Error realizando las copias de los ficheros
- 15= Error recuperando el esquema del repositorio

- 16= Error generando el ebiMapFormat

- 17= Error recuperando la transformación XSLT

- 18= Error realizando la transformación

EBIPutRepositoryItem

Esta función permitirá subir/actualizar objetos al repositorio del EBIMAP. Los objetos se subirán a al dominio/aplicación que esta ejecutando el script.

```
public int EBIPutRepositoryItem (    String filename,
                                   String type,
                                   String description)
```

filename	Fichero con ruta completa del objeto que se desea subir al repositorio
type	Tipo del objeto a subir, si se deja vacío se tomará la extensión del fichero
description	Descripción asociada a la subida del objeto al repositorio
Si es mayor 0, representará el identificador del objeto insertado/actualizado en el repositorio -1 Error general -2 No existe el fichero pasado como parámetro	

EBIRejectedPublications

Esta función devolverá en una cadena los identificadores de todas la publicaciones que estén en estado rechazado. Los identificadores de la publicaciones irán separados por el símbolo '|'

```
public String EBIRejectedPublications (    String domain,
                                           String application)
```

domain	Dominio de las publicaciones rechazadas a devolver. OPCIONAL
application	Aplicación de la publicaciones rechazadas a devolver. OPCIONAL
Cadena con todas las aplicaciones en estado rechazado. Estas irán separadas por el símbolo ' ', -1 si hay error	

EBIRejectedPublicationsV2

Esta función devolverá en una lista con todas la publicaciones que esten en estado rechazado

```
public EBIDocumentWS[] EBIRejectedPublicationsV2 ( String domain,
                                                    String application)
```

domain	Dominio de las publicaciones rechazadas a devolver. OPCIONAL
---------------	--

application	Aplicación de la publicaciones rechazadas a devolver. OPCIONAL
Array de objetos del tipo <i>EBIDocumentWS</i> , con las publicaciones en estado rechazado	

EBIRejectSubscription

Rechazada el identificador de mensaje que recibe como parámetro

```
public int EBIRejectSubscription ( String messageIds,
                                String info)
```

messageIds	Identificador del mensaje a rechazar. Se pueden rechazar, varias suscripciones a la vez si estas se envían al EBI BROKER separadas por
info	Información adicional indicando el motivo de rechazo
1 → Mensaje rechazado correctamente -1 → Error -2 → Suscripción no encontrada -3 → Estado del documento no valido para ser rechazado. Los documentos solo se pueden rechazar cuando están en estado Suscrito ó Pendiente de Confirmación -4 → Documento ya en estado rechazado	

EBIRejectSubscriptionV2

Rechazada el identificador de mensaje que recibe como parámetro

```
public int EBIRejectSubscriptionV2 (String[] messageIds,
                                   String info)
```

messageIds	Array con los identificadores de mensaje a rechazar.
info	Información adicional indicando el motivo de rechazo
1 → Mensaje rechazado correctamente -1 → Error -2 → Suscripción no encontrada -3 → Estado del documento no valido para ser rechazado. Los documentos solo se pueden rechazar cuando están en estado Suscrito ó Pendiente de Confirmación -4 → Documento ya en estado rechazado	

EBIReleaseApplication

Esta función libera una aplicación de su acceso exclusivo

```
public boolean EBIReleaseApplication()
```

True o false en función del éxito de la operación

EBIRemotePublish

Realiza la publicación del mensaje o máscara que recibe como parámetro, en el EBI BROKER **remoto**, que se le indica en el parámetro url.

El parámetro source, representa un sistema virtual de ficheros, pudiendo ser este sistema de ficheros uno de los siguientes: FTP, fichero locales y SFTP

```
public String EBIRemotePublish(String url,
                               String user,
                               String password,
                               String domain,
                               String application,
                               String source,
                               String schema,
                               String destination,
                               boolean wait,
                               boolean delete,
                               int timeout,
                               String copyDestination,
                               String param,
                               String reference,
                               int duplicate,
                               String[] options)
```

)	
url	URL donde se encuentra el EBIBROKER remoto en el que se quiere realizar la publicación
user	Usuario para validarse en el EBIBROKER remoto
password	Contraseña para validarse en el EBIBROKER remoto
domain	Domino remoto en el que se realizará la publicación
application	Aplicación remota en la que se realizará la publicación
source	<p>(1) Ruta virtual donde se encuentran los ficheros a publicar. En este caso el nombre de los ficheros se extrae del interfaz</p> <p>(2) Ruta virtual con nombre del fichero a publicar</p> <p>(3) Ruta virtual con máscara con los ficheros a publicar</p> <p>Local Files Proporciona acceso a los ficheros, en sistema local de archivos</p> <pre>URI Format [file://] absolute-path file:///home/someuser/somedir file:///C:/Documents and Settings</pre> <p>FTP Proporciona acceso a los ficheros en un servidor FTP.</p> <pre>URI Format ftp://[username [: password]@] hostname [: port][absolute-path] ftp://myusername:mypassword@somehost/pub/downloads/somefile.t gz</pre> <p>SFTP Proporciona acceso a los ficheros en un servidor SFTP</p> <pre>URI Format sftp://[username [: password]@] hostname [: port][absolute-path] sftp://myusername:mypassword@somehost/pub/downloads/somefile. tgz</pre>
schema	Tipo del mensaje que corresponde con el documento
destination	Aplicación destino del mensaje
wait	Indica si se debe esperar o no la recolección de todos los mensajes publicados
delete	Indica si se deben borrar o no los documentos publicados
timeOut	Tiempo máximo de espera en segundos.
copyDestination	Directorio en el cual se quiere realizar una copia de los ficheros
param	Formato con el cual se quiere hacer la copia
reference	Referencia con la que se publicará el documento
duplicate	<p>Indica la acción a tomar con las publicaciones duplicadas</p> <p>0: dupIgnore. Se generará una nueva publicación aunque esta ya este duplicada</p> <p>1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH</p> <p>2: dupError. Se devuelve un código de error.</p> <p>Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación</p>
options	<p>Opciones de configuración asociadas al sistema de ficheros</p> <pre>setPassiveMode --> true/false setSys --> Para fijar el sistema de ficheros FTP SYST_UNIX UNIX SYST_NT WINDOWS SYST_OS2 OS/2 SYST_OS400 OS/400 SYST_VMS VMS SYST_MVS MVS SYST_NETWORKWARE NETWORKWARE setTimeout --> Timeout en milisegundos setProxyHost --> Servidor Proxy setProxyPort --> Puerto Proxy setProxyType --> Tipo de proxy HTTP vs SOCKS setUsername --> Usuario proxy</pre>

```

setPassword --> Contraseña proxy
setDomain --> Dominio proxy
setUserDirIsRoot --> Determina si el directorio raiz es el
del usuario o el del sistema. Valor por defecto true

```

Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los identificadores de los documentos publicados separados por '|'

-1 = Error
 -2 = Publicación no autorizada
 -3 = Destino de la publicación desconocido
 -4 = Documento incorrecto según la validación sintáctica de su interfaz
 -5 = Aplicación destino no acepta la suscripción
 -7 = Usuario no autorizado a publicar
 -8 = Publicación duplicada
 -10= Esquema no encontrado
 -50= Error registrando la sesión en el EBIBROKER remoto

EBIRemoteSQL

El objetivo de esta función es ejecutar sentencias SQL de forma remota en el EBI BROKER.

Para garantizar la seguridad de ejecución de la consulta SQL, si el *connectString* es vacío, lo que quiere decir que la consulta va destinada a la base de datos del EBI BROKER, y esta deberá ir tanto en formato normal, es decir, en una cadena de texto, como encriptada. Si existe alguna discrepancia entre ellas, no se ejecutará la sentencia SQL.

Si la sentencia SQL es del tipo SELECT, el resultado de la misma se podrá consultar en formato XML o en formato plano en la función *DsXml()*.

```

public int EBIRemoteSQL(String connectionString,
                        String user,
                        String passwordE,
                        String sql,
                        String sqlE,
                        String parameters,
                        boolean plainformat)

```

connectString	Cadena de conexión
user	Usuario de conexión a la BD
passwordE	Contraseña encriptada
sql	Sentencia SQL a ejecutar
sqlE	Sentencia SQL encriptada, que validará la sentencia SQL anterior. Este parámetro solo tiene sentido si la cadena de conexión es vacía, ya que la consulta se realizará sobre la base de datos del EBI BROKER
parameters	Parámetros de la sentencia SQL. Estos irán separados por ' '
plainFormat	Este parámetro indica si el resultado que se almacena en la variable DsXML, se quiere en formato XML o formato plano, es decir, las columnas separadas por ; y la filas por retorno de carro
0 si la ejecución es correcta, -1 en caso de error	

EBIRollbackTrans

Esta función realiza un rollback de la transacción que recibe como parámetro

```

public boolean EBIRollbackTrans (String transactionId)

```

transactionId	Identificador de la transacción
True o false en función del éxito o fracaso de la operación	

EBISendCMIS

Esta función enviará el fichero que recibe como parámetro, al servidor CMIS que se indica en la URL

```
EBISendCMIS ( "http://repo.opencmis.org/inmemory/atom/",
    "",
    "",
    "Apache Chemistry OpenCMIS InMemory Repository",
    "Bandeja entrada EDI",
    "/tmp/composer1234.pdf",
    "fact:pedidoId=PEDIDO_ID
    fact:numFactura=NUMFACTURA
    fact:importe=IMPORTE
    fact:divisa=DIVISA
    fact:abono=ABONO
    fact:proveedor=PROVEEDOR
    fact:cifProveedor=CIFPROVEEDOR
    fact:fechafact=FECHFACTURA
    comun:nombreempresa=EMPRESARECEPTORA",
    "application/pdf"
    "none")
```

```
public boolean EBISendCMIS ( String url,
    String user,
    String password,
    String repositoryName,
    String folder,
    String filename,
    String properties,
    String mimetype,
    String versioning,
    String cmisServer)
```

url	Url donde se encuentra ubicado el servidor CMIS
user	Usuario de acceso al servidor CMIS
password	Contraseña de acceso al servidor CMIS
repositoryName	Nombre del repositorio del servidor CMIS donde se van a depositar los ficheros
folder	Carpeta del repositorio donde se van dejar los ficheros
filename	Fichero con ruta completa que se desea enviar al servidor CMIS
properties	Propiedades del fichero que se va enviar al servidor CMIS. Los valores de la propiedad, se debe indicar de tipo son, añadiendo un carácter al final de la cadena, para indicar el tipo ' B ' --> boolean ' L ' --> long ' I ' --> integer ' D ' --> date yyMMdd ' F ' --> float ' S ' --> String
mimetype	Tipo de documento que se enviar al servidor CMIS application/pdf, text/xml, text/plain, application/doc,
versioning	Tipo de versionado de la información que se envía NONE("none"), MAJOR("major"), MINOR("minor"), CHECKEDOUT("checkedout")
cmisServer	Tipo de servidor CMIS ALFRESCO APACHE_CHEMISTRY SHARE_POINT
True si la operación tiene éxito o false en caso contrario	

messagelds	Array con los identificadores de los mensajes a recuperar
transactionId	Transacción a la que pertenecen la suscripciones que se están recogiendo
references	Array con las referencias de las suscripciones a recuperar
server	Servidor de SMTP al cual se envía el mail
fromUser	Usuario que envía el mail
toUser	Usuario destinatario del mail
subject	Asunto del mail
text	Texto del mail
smtpUser	Usuario ESMTP
smtpPassword	Contraseña ESMTP
Si > 0 número de suscripciones recibidas y enviadas por mail -1 Error -2 Suscripción no autorizada -7 Usuario no autorizado a suscribir	

EBISubsAndOpen

Recoge todos los mensajes que estén suscritos en el EBI BROKER y los abre con el programa que tenga asociado el sistema operativo

```
public int EBISubsAndOpen (String schema,
                           int maxSubscriptions,
                           String reference)
```

schema	Esquema del mensaje del cual se quieren recoger las suscripciones
maxSubscriptions	Número máximo de suscripciones a recuperar. OPCIONAL
reference	Indica la referencia de los documentos a recoger. Esta puede ir separada por ' '. OPCIONAL
Si > 0 número de suscripciones recibidas -1 Error -2 Suscripción no autorizada -12 Fallo abriendo de los mensajes -13 Fallo durante la confirmación de mensajes -14 La JVM no soporta operaciones del tipo desktop	

EBISubsAndPrint

Recoge todos los mensajes que estén en el EBI BROKER, en estado suscrito y los imprime en función del tipo que se reciba como parámetro

```
public int EBISubsAndPrint( String type,
                           String schema,
                           int maxSubscriptions,
                           String printer,
                           String reference,
                           boolean showPrinterDialog,
                           int copies,
                           String transactionId)
```

type	Tipo de suscripción a imprimir (PDF, JPEG, BMP, WMF, etc...) Por el momento solo se soporta PDF, PDFEXT, PDFEXT2 Y WMF
schema	Esquema del mensaje del cual se quieren recoger las suscripciones
maxSubscriptions	Número máximo de suscripciones a recibir
printer	Nombre de la impresora por la cual se quiere realizar la impresión
stretchDraw	Indica si la imagen se debe ajustar al tamaño del papel
reference	Indica la referencia de los documentos a imprimir. Esta puede ir separada por ' '
showPrinterDialog	Indica si se debe mostrar o no el cuadro de dialogo de la impresión
copies	Se puede especificar el número de copias del documento a imprimir

Si > 0 número de suscripciones recibidas

- 1 Error
- 2 Suscripción no autorizada
- 3 Fallo confirmando la suscripción
- 12 Fallo durante la impresión del PDF
- 13 Formato no soportado
- 14 Impresora no seleccionada
- 15 Error confirmando las suscripciones
- 16 Fallo durante la impresión del WMF

EBISubsAndReport

Recoge todos los mensajes que estén el repositorio y les aplica el informe que se recibe como parámetro

```
public int EBISubsAndReport( String    directory,
                             String    schema,
                             String    report,
                             String    type,
                             boolean    mergeResults,
                             String    dynamicFilter,
                             int        maxSubscriptions,
                             String    transactionId,
                             String[]   references,
                             boolean    confirmSubscriptions,
                             String    profile)
```

directory	Directorio destino en el que se depositarán los informes generados
schema	Esquema del mensaje del cual se quieren recoger las suscripciones
report	Informe que se desea aplicar a la suscripciones
type	Tipo del informe PDF, EMF, WMT, PDF/A, FJPEG, EMF, GIF, PS, SWF, HTML4
mergeResults	Indica si se deben fusionar los resultados de las transformaciones
dynamicFilter	Filtro dinámico
maxSubscriptions	Número máximo de suscripciones a recibir
transactionId	Identificador de la transacción en la cual se recogen las suscripciones
reference	Referencias de las suscripciones que se desean procesar
confirmSubscriptions	Indica si se deben confirmar las suscripciones o no
profile	Perfil de ejecución de la petición
>0 Número de suscripciones recogidas <ul style="list-style-type: none"> -1 Error -2 Suscripción no autorizada -3 Informe no encontrada -4 Error realizando el informe -7 Usuario no autorizado el informe -8 Directorio destino no existe 	

EBISubsAndSave

Recoge todos los mensajes que estén en el EBI BROKER en estado suscrito y los copia al directorio que le indique el usuario. Este directorio, se le preguntará al usuario en el momento de la descarga del mensaje

```
public int EBISubsAndSave ( String schema,
                           int maxSubscriptions,
                           String reference)
```

schema	Esquema del mensaje del cual se quieren recoger las suscripciones
maxSubscriptions	Número máximo de suscripciones a recuperar. OPCIONAL
reference	Indica la referencia de los documentos a recoger. Esta puede ir separada por ' '. OPCIONAL
Si > 0 número de suscripciones recibidas	

- 1 Error
- 2 Suscripción no autorizada
- 13 Fallo durante la confirmación de mensajes
- 14 La JVM no soporta operaciones del tipo desktop

EBISubsAndTransform

Recoge todos los mensajes que estén el EBI BROKER, en estado suscrito, y les aplica la transformación que recibe como parámetro.

```
public int EBISubsAndTransform(String directory,
                               String schema,
                               String transformation,
                               String filter,
                               boolean generateLog,
                               int maxSubscriptions,
                               String transformationIniFile,
                               boolean ebiMapFormat,
                               String transactionId,
                               String reference,
                               boolean keepFileName,
                               boolean confirmSubscriptions,
                               String profile,
                               boolean append)
```

directory	Directorio en el que se depositan las suscripciones una vez transformadas
schema	Esquema de las suscripciones a transformar
transformation	Transformación aplicar
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
generateLog	Parámetro obsoleto. Valor por defecto false.
maxSubscriptions	Número máximo de suscripciones a recuperar
transformationIniFile	Parámetro obsoleto. Valor por defecto "".
ebiMapFormat	Este parámetro indica si el resultado del mapeado se quiere generar en formato EBIMAPFormat
transactionId	Identificador de la transacción asociado a la suscripción que se desea recuperar
reference	Referencia del documento a recuperar. Se pueden indicar varias referencias de mensajes, separados por ' '
keepFileName	Conservar el nombre del fichero original
confirmSubscriptions	Indica si se deben confirmar los documentos recogidos o no
profile	Perfil para la ejecución de la transformación
append	Indica si se desea concatenar los ficheros que recoja la suscripción, o sobrescribir su valor. Su valor por defecto es true, es decir, concatena
Si > 0 número de fichero generados en la transformación -1 Error -2 Suscripción no autorizada -3 Transformación no encontrada -4 Error realizando la transformación -7 Usuario no autorizado a transformar -8 Directorio destino no existe	

Durante la ejecución de la transformación, esta tendrá disponible, las siguientes variables, con la información del documento suscrito que esta transformando.

```
@EBIID
@EBIDOMAIN
@EBIAPPLICATION
@EBISCHEMA
@EBIDATE
@EBIREFERENCE
@EBISIZE
```

EBISubsAndTransformAndFTP

Recoge todos los mensajes que estén el repositorio y los envía por FTP, al servidor que se le indique como parámetro

```
public int EBISubsAndTransformAndFTP (    String schema,
                                         String transformation,
                                         String filter,
                                         String host,
                                         int port,
                                         String user,
                                         String assword,
                                         String transferType,
                                         boolean passive,
                                         String destinationPath,
                                         boolean append,
                                         int maxSubscriptions,
                                         String transactionId,
                                         String reference,
                                         String profile,
                                         boolean renameUpload)
```

schema	Esquema de las suscripciones a transformar
transformation	Transformación aplicar
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
host	Máquina en la que se encuentra el servicio de FTP
port	Puerto en el que este escuchando el servicio de FTP
user	Usuario FTP
password	Contraseña FTP
transferType	Tipo de transferencia ASCII o BINARY
passive	Tipo de transferencia, pasiva o activa
destinationPath	Ruta del servidor FTP en la que se depositarán los ficheros
append	Si en el fichero a enviar, ya se encuentra en el servidor FTP indica si se debe sobrescribir o se añade al que ya existe.
maxSubscriptions	Número máximo de suscripciones a recuperar
transactionId	Identificador de la transacción asociado a la suscripción que se desea recuperar
reference	Referencia del documento a recuperar. Se pueden indicar varias referencias de mensajes, separados por ' '
profile	Perfil para la ejecución de la transformación
renameUpload	Indica si durante la subida del fichero al servidor FTP, se debe renombrar para evitar ser procesado por SII
Si > 0 número de suscripciones recibidas y enviadas via FTP -1 Error -2 Suscripción no autorizada -3 Transformación no encontrada -4 Error realizando la transformación -7 Usuario no autorizado a suscribir y transformar -12 Fallo durante la transferencia de fichero vía FTP -13 Fallo durante la confirmación de mensajes	

Durante la ejecución de la transformación, esta tendrá disponible, las siguientes variables, con la información del documento suscrito que esta transformando.

```
@EBIID
@EBIDOMAIN
@EBIAPPLICATION
@EBISCHEMA
@EBIDATE
@EBIREFERENCE
@EBISIZE
```

EBISubsAndTransformAndMail

Esta función recoge la suscripciones del EBI BROKER, les aplica la transformación que se indica como parámetro y envía el resultado por mail como ficheros adjuntos

```
public int EBISubsAndTransformAndMail(
    String schema,
    String transformation,
    String filter,
    int maxSubscriptions,
    String transactionId,
    String[] references,
    String profile,
    String server,
    String fromUser,
    String toUser,
    String subject,
    String text,
    String smtpUser,
    String smtpPassword)
```

schema	Esquema de las suscripciones a transformar
transformation	Transformación aplicar
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
maxSubscriptions	Número máximo de suscripciones a recuperar
transactionId	Identificador de la transacción asociado a la suscripción que se desea recuperar
reference	Array con las referencias de los documento a recuperar.
profile	Perfil para la ejecución de la transformación
server	Servidor de SMTP al cual se envía el mail
fromUser	Usuario que envía el mail
toUser	Usuario destinatario del mail
subject	Asunto del mail
text	Texto del mail
smtpUser	Usuario ESMTP
smtpPassword	Contraseña ESMTP
Si > 0 número de ficheros generados en la transformación y enviados por mail -1 Error -2 Suscripción no autorizada -3 Transformación no encontrada -4 Error realizando la transformación -7 Usuario no autorizado a transformar	

Durante la ejecución de la transformación, esta tendrá disponible, las siguientes variables, con la información del documento suscrito que esta transformando.

```
@EBIID
@EBIDOMAIN
@EBIAPPLICATION
@EBISCHEMA
@EBIDATE
@EBIREFERENCE
@EBISIZE
```

EBISubsAndTransformVFS

Recoge todos los mensajes que estén el repositorio y les aplica la transformación que recibe como parámetro y deposita el resulta en la ruta virtual url

El parámetro url, representa un sistema virtual de ficheros, pudiendo ser este sistema uno de los siguientes, FTP, fichero locales, HTTP/S y SFTP

```

public int EBISubsAndTransformVFS( String url,
                                   String schema,
                                   String transformation,
                                   String filter,
                                   int maxSubscriptions,
                                   String transactionId,
                                   String[] references,
                                   boolean confirmSubscriptions,
                                   String profile,
                                   boolean append,
                                   String[] vfsOptions)

```

url	<p>(1) Ruta virtual donde se depositarán los mensajes resultado de aplicar la transformación. En este caso el nombre de los ficheros se extrae del interfaz</p> <p>(2) Ruta virtual con nombre del fichero a publicar</p> <p>(3) Ruta virtual con máscara con los ficheros a publicar</p> <p>Local Files Proporciona acceso a los ficheros, en sistema local de archivos</p> <pre> URI Format [file://] absolute-path file:///home/someuser/somedir file:///C:/Documents and Settings </pre> <p>HTTP and HTTPS Provee acceso a los fichero a través de un servidor HTTP</p> <pre> URI Format http://[username [: password]@] hostname [: port][absolute-path] https://[username [: password]@] hostname [: port] [absolute-path] http://somehost:8080/downloads/somefile.jar http://myusername@somehost/index.html </pre> <p>FTP Proporciona acceso la los ficheros en un servidor FTP.</p> <pre> URI Format ftp://[username [: password]@] hostname [: port][absolute-path] ftp://myusername:mypassword@somehost/pub/downloads/somefile .tgz </pre> <p>SFTP Proporciona acceso a los ficheros en un servidor SFTP</p> <pre> URI Format sftp://[username [: password]@] hostname [: port][absolute-path] sftp://myusername:mypassword@somehost/pub/downloads/somefil e.tgz </pre>
schema	Esquema de las suscripciones a transformar
transformation	Transformación aplicar
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
maxSubscriptions	Número máximo de suscripciones a recuperar
transactionId	Identificador de la transacción asociado a la suscripción que se desea recuperar
reference	Array con las referencias de los documento a recuperar.
profile	Perfil para la ejecución de la transformación
append	Indica si se debe concatenar el contenido del fichero en el caso que este exista en el destino. No todos los sistema de fichero soportan el modo append
vfsOptions	<p>Opciones de configuración asociadas al sistema de ficheros</p> <pre> setPassiveMode --> true/false setSys --> Para fijar el sistema de ficheros FTP SYST_UNIX UNIX SYST_NT WINDOWS SYST_OS2 OS/2 SYST_OS400 OS/400 SYST_VMS VMS SYST_MVS MVS SYST_NETWORKWARE NETWORKWARE setTimeout --> Timeout en milisegundos setProxyHost --> Servidor Proxy setProxyPort --> Puerto Proxy setProxyType --> Tipo de proxy HTTP vs SOCKS setUsername --> Usuario proxy setPassword --> Contraseña proxy </pre>

	<pre> setDomain --> Dominio proxy setUserDirIsRoot --> Determina si el directorio raiz es el del usuario o el del sistema. Valor por defecto true </pre>
<p>Si > 0 número de fichero generados en la transformación</p> <ul style="list-style-type: none"> -1 Error -2 Suscripción no autorizada -3 Transformación no encontrada -4 Error realizando la transformación -7 Usuario no autorizado a transformar -8 Directorio destino no existe 	

Durante la ejecución de la transformación, esta tendrá disponible, las siguientes variables, con la información del documento suscrito que esta transformando.

```

@EBIID
@EBIDOMAIN
@EBIAPPLICATION
@EBISCHEMA
@EBIDATE
@EBIREFERENCE
@EBISIZE

```

EBISubscriptions

Recoge todos los mensajes que están almacenados en el EBI BROKER en estado suscrito

```

public int EBISubscriptions(String directory,
                           String schema,
                           int maxSubscriptions,
                           boolean confirmSubscriptions,
                           String idMessage,
                           boolean noChangeState,
                           String idTransaction,
                           String reference,
                           boolean keepFileName,
                           boolean checkEmptyFiles,
                           boolean append)

```

directory	Directorio en el que se depositan las suscripciones
schema	Tipo del mensaje del cual se quieren recoger las suscripciones
maxSubscriptions	Número máximo de suscripciones a recuperar
confirmSubscriptions	Indica si se van a confirmar las suscripciones o no
idMessage	Identificador del mensaje a recuperar
noChangeState	Indica si se debe cambiar el estado de la suscripción tras ser recogida
idTransaction	Identificador de la transacción asociado a la suscripción que se desea recuperar
reference	Referencia de la suscripción a recuperar
keepFileName	Indica si se conserva el nombre original con el que fue publicado el mensaje
checkEmptyFiles	Indica si se debe comprobar la integridad de los ficheros descargados. Al menos uno de ellos debe tener tamaño distinto de cero
append	Indica si se debe concatenar el contenido del fichero en el caso que este exista en el destino. No todos los sistema de fichero soportan el modo append
<p>Si > 0 número de suscripciones recibidas</p> <ul style="list-style-type: none"> -1 Error -2 Suscripción no autorizada -3 Directorio destino no existe -7 Usuario no autorizado a suscribir 	

EBISubscriptionsList

Devuelve una lista con todos los mensajes que están en estado suscrito. Esta lista contendrá todos los identificadores de documentos separados por '~'

```
public String EBISubscriptionsList( String schema,
                                   int maxSubscriptions)
```

schema	Tipo del mensaje del cual se quieren recoger la lista de suscripciones
maxSubscriptions	Número máximo de suscripciones a recuperar en la lista
Lista de suscripciones recibidas -1 Error -2 Suscripción no autorizada -7 Usuario no autorizado a suscribir	

EBISubscriptionsPending

Devuelve el número de suscripciones pendientes de recoger, del documento publicado que recibe como parámetro

```
public int EBISubscriptionsPending(String idMessage)
```

idMessage	Identificador del documento duplicado. Se pueden recibir varios identificadores separados por ' '
Si > 0 número de suscripciones pendientes de recoger -1 Error	

EBISubscriptionsVFS

Recoge todos los mensajes que están almacenados en el EBI BROKER, depositándolos en la ruta virtual que se indica en url. El parámetro url, representa un sistema virtual de ficheros, pudiendo ser este sistema uno de los siguientes: FTP, fichero locales, HTTP/S y HTTPS y SFTP

```
public int EBISubscriptionsVFS(      String      url,
                                   String      schema,
                                   int          maxSubscriptions,
                                   boolean      confirmSubscriptions,
                                   String[]     messagesId,
                                   String       transactionId,
                                   String[]     references,
                                   boolean      append,
                                   String[]     vfsOptions )
```

url	<p>(1) Ruta virtual donde se depositarán los mensajes suscritos recuperados del EBI BROKER. En este caso el nombre de los ficheros se extrae del interfaz</p> <p>(2) Ruta virtual con nombre del fichero a publicar</p> <p>(3) Ruta virtual con máscara con los ficheros a publicar</p> <p>Local Files Proporciona acceso a los ficheros, en sistema local de archivos</p> <pre>URI Format [file://] absolute-path file:///home/someuser/somedir file:///C:/Documents and Settings</pre> <p>HTTP and HTTPS Provee acceso a los fichero a través de un servidor HTTP</p> <pre>URI Format http://[username [: password]@] hostname [: port][absolute-path] https://[username [: password]@] hostname [: port] [absolute-path] http://somehost:8080/downloads/somefile.jar http://myusername@somehost/index.html</pre> <p>FTP Proporciona acceso la los ficheros en un servidor FTP.</p> <pre>URI Format ftp://[username [: password]@] hostname [: port][absolute-path] ftp://myusername:mypassword@somehost/pub/downloads/somefile .tgz</pre> <p>SFTP Proporciona acceso a los ficheros en un servidor SFTP</p> <pre>URI Format sftp://[username [: password]@] hostname [: port][absolute-path] sftp://myusername:mypassword@somehost/pub/downloads/somefil e.tgz</pre>
------------	--

schema	Tipo del mensaje del cual se quieren recoger las suscripciones
maxSubscriptions	Número máximo de suscripciones a recuperar
confirmSubscriptions	Indica si se van a confirmar las suscripciones o no
messagesId	Array con los identificadores de los mensajes a recuperar
transactionId	Identificador de la transacción asociado a la suscripción que se desea recuperar
reference	Array con las referencias de los documento a recuperar.
append	Indica si se debe concatenar el contenido del fichero en el caso que este exista en el destino. No todos los sistema de fichero soportan el modo append
vfsOptions	<p>Opciones de configuración asociadas al sistema de ficheros</p> <pre> setPassiveMode --> true/false setSys --> Para fijar el sistema de ficheros FTP SYST_UNIX UNIX SYST_NT WINDOWS SYST_OS2 OS/2 SYST_OS400 OS/400 SYST_VMS VMS SYST_MVS MVS SYST_NETWORKWARE NETWORKWARE setTimeout --> Timeout en milisegundos setProxyHost --> Servidor Proxy setProxyPort --> Puerto Proxy setProxyType --> Tipo de proxy HTTP vs SOCKS setUsername --> Usuario proxy setPassword --> Contraseña proxy setDomain --> Dominio proxy createFolder --> true/false, sino existe el directorio lo crea setUserDirIsRoot --> Determina si el directorio raiz es el del usuario o el del sistema. Valor por defecto true </pre>
<p>Si > 0 número de suscripciones recibidas</p> <ul style="list-style-type: none"> -1 Error -2 Suscripción no autorizada -3 Destino no existe -7 Usuario no autorizado a suscribir 	

EBITransformation

Aplica la transformación que se recibe como parámetro, tomando como entrada el parámetro *dirIn* y dejando el resultado en el directorio *dirOut*.

En el caso que *dirIn*, sea un directorio, se compactan todos lo ficheros y se suben al servidor para ser transformados.

```

public int EBITransformation(
    String transformation,
    String dirIn,
    String dirOut,
    String filter,
    boolean generateLog,
    String appTransSource,
    String transformationIniFile,
    boolean ebiMapFormat,
    String profile)

```

transformation	Transformación aplicar
dirIn	Directorio de entrada de la transformación o fichero que se desean enviar a la transformación
dirOut	Directorio de salida de la transformación
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
generateLog	Parámetro obsoleto. Valor por defecto false
appTransSource	Aplicación que contiene la transformación

transformationIniFile	Parámetro obsoleto. Valor por defecto ""
ebiMapFormat	Este parámetro indica si el resultado del mapeado se quiere generar en formato EBIMAPFormat
profile	Perfil para la ejecución de la transformación
Si > 0 número de fichero generados en la transformación -1 Error -2 Transformación no autorizada -3 Transformación no encontrada -4 Directorio para realizar la transformación no encontrado -7 Usuario no autorizado a transformar	

EBITransformationAndPublish

Esta función enviará al EBI los ficheros que hay en *dirin* en base al *sourceschema*, salvo que el origen sea de base de datos, con lo que el *dirin* se debe dejar vacío. Una vez enviados los fichero al EBI BROKER, se les aplicará la transformación que se indica como parámetro, y el resultado de la misma por un lado se devolverá al cliente, y por otro se publicará en el EBI con el esquema *destinationschema*.

```
public int EBITransformationAndPublish (    String    dirin,
                                           String    dirout,
                                           String    sourceschema,
                                           String    transformation,
                                           String[]  profile,
                                           String    filter,
                                           String    destinationschema,
                                           String    destination,
                                           String    reference,
                                           int        duplicates,
                                           String    mask,
                                           String    maskOut)
```

dirin	(1) Ruta donde se encuentran los ficheros a publicar. En este caso el nombre de los ficheros se extrae del interfaz sourceschema (2) Nombre del fichero a publicar (3) Máscara con los ficheros a publicar (4) Si esta vacío indicará el origen de datos es una base de datos, y la cadena de conexión se extraerá de sourceschema, generado un EBIMapFomar como fichero de publicación
dirout	Directorio donde se depositará el resultado de la transformación
sourceschema	Esquema origen de la publicación
transformation	Transformación que se aplicará al origen de datos
profile	Perfil de ejecución de la transformación. Este perfil será un array de pares [variable1, valor1, variable2, valor2]
filter	Filtro a aplicar a la transformación, o en el caso que el origen sea base de datos, filtro que se aplicará para generar el EBIMapFormat
destinationschema	Esquema que se empleará para publicar el resultado de la transformación.
destination	Destino de la publicación
reference	Referencia de la publicación
duplicates	Indica la acción a tomar con las publicaciones duplicadas 0: dupIgnore. Se generará un nueva publicación aunque esta ya este duplicada 1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador del la publicación que tiene el mismo HASH 2: dupError. Se devuelve un código de error. Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación
mask	Máscara que se aplicará al resultado de la transformación, con el objetivo de publicar solo aquellos ficheros que cumplan la máscara.
maskOut	Máscara que se aplicará al resultado de la transformación, con el objetivo de devolver como resultado de la función solo aquellos ficheros que cumplan la máscara.

	OPCIONAL
<p>Número de documentos generados por la transformación</p> <ul style="list-style-type: none"> -1 = Error -2 = Publicación no autorizada -3 = Destino de la publicación desconocido -4 = Documento incorrecto según la validación sintáctica de su interfaz -5 = Aplicación destino no acepta la suscripción -7 = Usuario no autorizado a publicar -8 = Publicación duplicada -9 = El dominio no permite publicaciones BROADCAST. -10= Esquema no encontrado -14= Error durante la transformación 	

EBITransformationMakeProfile

Esta función servirá para construir un *profile* (perfil) de ejecución para la función de transformación. El resultado de esta función, se pasará como parámetro a la función de transformación. Ejemplo de uso:

```

Paramnames      = Array ("AA","BB")
Paramvalues     = Array ("11", "22")
ContextScript   = '@ID='0725368525088';
resultado=EBITransformationMakeProfile ( Paramnames,Paramvalues,
ContextScript);
WriteLog (resultado);

Resultado:
<?xml version = "1.0"?>
  <Profile>
    <Variables AA = "11" BB = "22"/>
    <ContextScript>@ID:=12314124142</ContextScript>
  </Profile>

```

```

public String EBITransformationMakeProfile (String[] varNames ,
                                           String[] varValues,
                                           String contextScript)

```

varNames	<p>Array con los nombre de las variables</p> <p>Las variables que se pueden pasar en el profile son las siguientes</p> <ul style="list-style-type: none"> Alias User Password Config Domain DataDir Debug EdiwinUser Language Contextscript
varValues	Array con los valores asociados a las variables
contextScript	Script de contexto en el que se ejecutará el la transformación
Cadena de texto, con estructura en XML, que representa el profile, que se debe pasar para ejecutar la transformación	

EBIUpdateAcquireApplication

Esta función actualiza la marca de acceso exclusivo a la aplicación asociada al script

```

public boolean EBIUpdateAcquireApplication (int timeout)

```

timeout	Se actualiza la marca de tiempo los segundos que se indican como parámetro
True o false en función del éxito de la operación	

EBIWaitUntilSubscriptionsGhatered

Espera hasta que todas las suscripciones hayan recogido el mensaje publicado que se pasa como parámetro

```
public boolean EBIWaitUntilSubscriptionsGhatered( String messageIds,
                                                int sampleTime,
                                                int timeOut)
```

messageIds	Identificador del mensaje publicado Se pueden recibir varios identificadores separados por ' '
sampleTime	Periodo de muestro en milisegundos
timeOut	Tiempo máximo de espera en segundos. Si el valor es 0, no se habrá timeout.
True o false en función del éxito de la operación	

LockFiles

Esta función bloqueará la lista de ficheros que recibe como parámetro. También puede recibir el nombre de un esquema, por lo que habrá que recuperar los nombre de los fichero del EBI BROKER. Para ello abrirá los ficheros en modo lectura/escritura en modo exclusivo

```
public boolean LockFiles(String path, String files, int type)
```

path	Ruta donde se encuentran los ficheros
files	Lista de ficheros a bloquear, separados por ';' o esquema a bloquear
type	Tipo de ficheros 0 → Nombres de fichero 1 → Esquema almacenado en el EBI BROKER
True o false en función del éxito de la operación	

UnLockFiles

Esta función desbloqueará toda la lista de ficheros que se encuentre bloqueados

```
public boolean UnLockFiles()
```

True o false en función del éxito de la operación

3.2.3 FICHEROS DE LOG

Los ficheros de LOG que genera el adaptador EDICOM, estarán almacenados dentro del directorio **log** de la aplicación. A su vez, el adaptador EDICOM, diariamente, creará una carpeta con el año mes y día (yyyymmdd) en la que irá copiando los ficheros de LOG pertenecientes a ese día.

Los ficheros de LOG que genera el adaptador EDICOM, son los siguientes:

- (1) **LOG de sesión.** El nombre de fichero del LOG de sesión se llama *ebiAdapterEdiwinSession.log*, y este se ira rotando diariamente. En el LOG de sesión, se registrará toda la actividad del adaptador. En el fichero *log4j.properties*, se puede cambiar el nivel de traza el adaptador. El valor por defecto es INFO, pudiéndose aumentar la traza cambiando por TRACE.

```
log4j.logger.com.edicom=INFO, ... → log4j.logger.com.edicom=TRACE, ...
```

- (2) **LOG de errores.** El nombre de fichero del LOG de errores se llama *ebiAdapterEdiwinSession_error.log*, y también se ira rotando diariamente. En

este LOG, se registrarán todos los errores de ejecución que se produzcan en el adaptador.

- (3) **LOG de ejecuciones de scripts.** El nombre de fichero del LOG de ejecuciones, tendrá el siguiente formato `idscript_nombrescript_añosmesdiahoramínutosegundomilisegunoaleatorio` (yyyymmddhhnnsszzz). La marca de tiempo que se añade al fichero de LOG, corresponde con el momento en el cual se ejecuta el script en el adaptador. Existirá un fichero de LOG de este tipo, por cada script que se ejecute.

Los adaptadores utilizan *log4j*, como tecnología para generar los logs, así que se pueden ampliar los manejadores de logs, acorde con las posibilidades de la tecnología *log4j*

3.3 ADAPTADOR EDIWIN

El Adaptador EDIWIN, es un conector, que hereda todas las propiedades y funcionalidades del adaptador de propósito general EDICOM. Este adaptador esta pensado para conectar al BUS la aplicación EDIWIN. Para facilitar la conexión entre la aplicación EDIWIN y el EBI BROKER, se han diseñado un conjunto de funciones específicas.

3.3.1 CONFIGURACIÓN

La configuración del adaptador se realiza en el fichero de propiedades ***ebiadapter.properties***. Este fichero se encontrará ubicado en la ruta que indique la variable de sistema ***user.dir***, que se indicará en el arranque del adaptador.

PROPIEDAD	DESCRIPCIÓN	VALOR POR DEFECTO
ebiadapter.multInstance	Indica si el adaptador debe conectarse a distintos orígenes de datos	false
ebiadapter.group	Origen de datos asociado al usuario de conexión	
ebiadapter.user	Usuario de conexión	
ebiadapter.password	Contraseña	
ebiadapter.type	Tipo de adaptador	EBIAEDIWIN
ebiadapter.name	Nombre con el que se registrará el adaptador en el EBI BROKER	
ebiadapter.domain	Dominio en el que se registra el adaptador	
ebiadapter.application	Aplicación en la que se registra el adaptador	
ebiadapter.url1	URL de conexión al EBI BROKER	
ebiadapter.language	Idioma del adaptador	es
ebiadapter.country	País del adaptador	ES
ebiadapter.deleteDirLogs	Número de días de permanencia de los logs en el directorio	0
ebiadapter.tr	Intervalo en minutos, de reinicio del adaptador, si no hay	0

	ningún proceso en ejecución. 0 deshabilitado	
ebiadapter.tc	Intervalo en minutos de reinicio del adaptador, independientemente, si hay algo en ejecución o no. 0 deshabilitado.	0
ediwin.ini	Nombre del fichero de configuración que empleará el servidor EDIWIN, para realizar las operaciones que le solicite el adaptador.	EDIWIN.INI
ediwin.url1	URL donde se encuentra ubicado el servidor EDIWIN	
ediwin.url2	Segunda URL donde se puede encontrar un servidor EDIWIN. Solo tiene sentido en entornos distribuidos en los que pueda haber mas de un servidor EDIWIN	
ediwin.url3	Tercera URL donde se puede encontrar un servidor EDIWIN. Solo tiene sentido en entornos distribuidos en los que pueda haber mas de un servidor EDIWIN	
ediwin.timeout	Tiempo máximo (segundos) de la conexión entre el adaptador EDIWIN, y el servidor EDIWIN	3600
ebiadapter.proxyServer	Servidor de PROXY	
ebiadapter.proxyServerPort	Puerto en el que se encuentra el servidor de PROXY	
ebiadapter.proxyUserName	Usuario PROXY	
ebiadapter.proxyPassword	Contraseña del PROXY	
ebiadapter.proxyServerType	Tipo de servidor PROXY	

El adaptador EDIWIN solo funciona en modo servicio, por lo tanto se conectará al al EBI BROKER, y se quedará a la espera de las planificaciones, que se le indiquen

```
java -Xmx512m -Xms128m -Duser.language=es -Duser.dir=... -Duser.log=...
-Djava.io.tmpdir=... -jar ebiaediwin.jar
```

PARÁMETRO	DESCRIPCIÓN	VALOR POR DEFECTO
java	Máquina virtual de Java (versión 1.6 o superior) que ejecuta el adaptador	
-Xmx	Tamaño máximo de la memoria	512m
-Xms	Tamaño mínimo de la memoria	128m
-Duser.language	Idioma en el que funciona el adaptador	es
-Duser.dir	Directorio base del adaptador, donde se encontrarán los ficheros de configuración	\$PWD ó %cd%
-Djava.io.tmpdir	Directorio donde se dejarán los ficheros temporales que utilice el adaptador	\$PWD/tmp ó %cd%\tmp
-Duser.log	Directorio donde se depositarán los ficheros de log	\$PWD/log/ ó %cd%\log\
-jar ebiaediwin.jar	Jar que contiene el adaptador	

3.3.2 VARIABLES DISPONIBLES

Ediwin_GRASP. En instalaciones sobre ASP, indica el grupo ASP en el que se encuentra el usuario con el que se va conectar al servidor EDIWIN.

EdiwinDomain. Se puede indicar de forma explícita el dominio destino en el servidor EDIWIN

3.3.3 FUNCIONES

EBIBulkSubsAndImport

Recoge todos los mensajes que estén en el repositorio del EBI BROKER, en estado suscrito, y los importa en *bulk mode*, es decir, se envían todos los mensajes al servidor EDIWIN, y posteriormente se realiza la confirmación de la entrega.

```
public int EBIBulkSubsAndImport(    String domain,
                                   String user,
                                   String password,
                                   String schema,
                                   int    maxSubscriptions,
                                   String ediwinIni,
                                   String transactionId,
                                   String reference,
                                   String mode,
                                   String process)
```

domain	Dominio EDIWIN en el que se realiza la importación
user	Lista de ficheros a bloquear, separados por ';' o esquema a bloquear
password	Tipo de ficheros 0 → Nombres de fichero 1 → Esquema almacenado en el EBI BROKER
schema	Tipo del mensaje del cual se quieren recoger las suscripciones e importalas
maxSubscriptions	Número máximo de suscripciones a recibir
ediwinIni	Fichero de inicio, con el que se instanciará el servidor para atender la petición. Si este valor es vacío, el sistema determinará el fichero de inicio, en función del grupo/usuario asignado a la operación.
transactionId	Identificador de transacción
reference	Referencia del documento a recuperar. Se pueden indicar varias referencias de mensajes, separados por ' '
mode	Indica el sentido de la importación (IN/OUT). Valor por defecto OUT
process	Proceso que se puede aplicar a la importación (AUTO por defecto)
Si > 0 número de suscripciones importadas en EDIWIN -1 Error -2 Suscripción no autorizada -10 Fallo durante la importación a la estación EDI.	

EBICfdiGetAsyncResult

Recupera de forma asíncrona el resultado de la tarea, que recibe como parámetro en la variable *asynctask*.

```
public int EBICfdiGetAsyncResult(    String cfdiServer,
                                   String wsuser,
                                   String wspassword,
                                   String destinationDir,
                                   String asynctask)
```

cfdiServer	URL del servidor Cfdi
wsuser	Usuario de conexión al servicio
wspassword	Contraseña de conexión al servicio
destinationDir	Directorio destino donde se depositará el resultado, de la operación asíncrona asynctask
asynctask	Identificador de operación de la petición asíncrona de procesado

Si es > 0 , número de ficheros generados por el servidor Cfdi

- 4 Directorio destino no existe
- 5 Error realizando la operación en el servidor Cfdi
- 6 Error en la url de la solicitud

EBICfdiProcessBlock

Esta función manda al servido de CFDI, el bloque de mensajes que se encuentran el directorio *sourceDir*, que se recibe como parámetro.

```
public int EBICfdiProcessBlock(    String    cfdiServer,
                                String    wsuser,
                                String    wpassword,
                                String    sourceDir,
                                String    paramsFile,
                                String    destinationDir,
                                String    taskname,
                                boolean    test,
                                String    idExt)
```

cfdiServer	URL del servidor Cfdi
wsuser	Usuario de conexión al servicio
wpassword	Contraseña de conexión al servicio
sourceDir	Directorio con mascara de los fichero a procesar
paramsFile	Fichero XML, con los parámetro de ejecución
destinationDir	Directorio destino donde se depositará el resultado de la operación
taskname	Nombre la tarea asociada a la petición
test	Identifica si es el entorno de test o de producción
idExt	Identificador externo de la petición
<p>Si es > 0 , número de ficheros generados por el servidor Cfdi</p> <ul style="list-style-type: none"> -3 Fichero de parámetros no existe -4 Directorio destino no existe -5 Error realizando la operación en el servidor Cfdi -6 Error en la url de la solicitud 	

EBICfdiProcessBlockAsync

Esta función manda al servido de CFDI, el bloque de mensajes que se encuentran el directorio *sourceDir*, para que se procesen de forma asíncrona.

```
public String EBICfdiProcessBlockAsync (    String    cfdiServer,
                                           String    wsuser,
                                           String    wpassword,
                                           String    sourceDir,
                                           String    paramsFile,
                                           String    taskname,
                                           boolean    test,
                                           String    idExt)
```

cfdiServer	URL del servidor Cfdi
wsuser	Usuario de conexión al servicio
wpassword	Contraseña de conexión al servicio
sourceDir	Directorio con mascara de los fichero a procesar
paramsFile	Fichero XML, con los parámetro de ejecución
taskname	Nombre la tarea asociada a la petición
test	Identifica si es el entorno de test o de producción
idExt	Identificador externo de la petición
<p>Si es > 0 , número de ficheros generados por el servidor Cfdi</p> <ul style="list-style-type: none"> -3 Fichero de parámetros no existe -5 Error realizando la operación en el servidor Cfdi 	

-6 Error en la url de la solicitud

EBICfdiSignGetCfdiAndNom151

Esta función manda al servido de CFDI, el mensaje que se encuentra en *sourceFile*, para firmarlo y aplicarle la NOM151

```
public int EBICfdiSignGetCfdiAndNom151 (    String    cfdiServer,
                                           String    wsuser,
                                           String    wspassword,
                                           String    sourceFile,
                                           String    destinationDir,
                                           String    pfxFile,
                                           String    password,
                                           String    certUser,
                                           String    certSerial,
                                           String    certPin,
                                           boolean   test,
                                           boolean   nom,
                                           boolean   nomxmlOutput,
                                           String    idExt)
```

cfdiServer	URL del servidor Cfdi
wsuser	Usuario de conexión al servicio
wspassword	Contraseña de conexión al servicio
sourceFile	Fichero origen a procesar
destinationDir	Directorio destino donde se depositarán los resultados
pfxFile	Fichero con el certificado pfx, para realizar la firma
pfxpwd	Contraseña del fichero pfx
serial	
test	
nom	
nomxmlOutput	
idExt	Identificador externo de la petición
Si es > 0 , número de ficheros generados por el servidor Cfdi	
-1 Error general	
-2 Fichero origen no existe.	
-3 Fichero pfx no existe	
-4 Directorio destino no existe	
-5 Error realizando la operación en el servidor Cfdi	
-6 Error en la url de la solicitud	

EBICreateEDIBinari

Dado un directorio de entrada con o sin máscara, para los ficheros que cumplan el criterio de búsqueda, se generará el correspondiente EDIFACT BINARI, con el contenido del fichero encontrado

```
public boolean EBICreateEDIBinari(    String inmask,
                                     String outdir,
                                     String source,
                                     String destination,
                                     String reference,
                                     String text)
```

inmask	Directorio origen con los ficheros de los cuales se quiere generar el BINARIO
outdir	Directorio destino donde se generan los ficheros resultado
source	Origen del mensaje BINARI
destination	Destino del mensaje BINARI
reference	Referencia del mensaje BINARI, en caso de dejarse vacia se generará una de forma aleatoria

text	Texto que se incluirá en el mensaje BINARI
True si la operación tiene éxito false en caso contrario	

EBIEdiwinBulkImportI

Esta función realizará la importación "masiva" de los intercambios que se reciben como parámetro.

```
public int EBIEdiwinBulkImportI (String user,
                                String password,
                                String domain,
                                String[] profile,
                                String file,
                                String ediwinIni,
                                String importType,
                                boolean delete)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
domain	Dominio en el que se va a realizar la importación
profile	Perfil de ejecución de la importación. Este perfil será un array de pares [variable1, valor1, variable2, valor2]
importType	Indica el sentido de la importación en la estación EDIWIN OUT--> salida IN-> entrada
ediwinIni	Fichero de inicio, con el que se instanciará el servidor para atender la petición
intFileName	Fichero con el intercambio que se desea importar en la estación EDIWIN
metaFileName	Fichero con los meta-datos del intercambio que se desea importar en la estación EDIWIN
errFileName	Fichero con los errores asociados al intercambio que se desea importar en la estación EDIWIN
delete	Si la importación tiene éxito se borrará el documento importado
Si >0 número de documentos importados -1 = Error durante la importación -2 = Error de conexión -3 = No existe alguno de los ficheros que se pasan como parámetro -4 = No sea especificado el fichero de intercambio	

EBIEdiwinChangeSituation

Esta operación realiza un cambio de situación de los mensajes que cumplen el filtro de búsqueda

```
public boolean EBIEdiwinChangeSituation ( String user,
                                           String password,
                                           String source,
                                           String destination,
                                           String filter,
                                           boolean groupByInter,
                                           String domain,
                                           String ediwinIni)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
source	Situación origen
destination	Situación destino
filter	Filtro para aplicar el cambio de interlocutor
groupByInter	Indica si se quiere agrupar por interlocutor o no
domain	Dominio de EDIWIN al que se quiere conectar
ediwinIni	Fichero de configuración de acceso a EDIWIN
True si la operación tiene éxito false en caso contrario	

EBIEdiwinConnect

Esta función le indica al servidor EDIWIN, que debe realizar la conexión que se le indica el los parámetros de la llamada.

```
public boolean EBIEdiwinConnect (   String user,
                                   String password,
                                   String protocol,
                                   String partner,
                                   String direction,
                                   String domain,
                                   String ediwinIni)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
protocol	Protocolo por el cual se va establecer la comunicación
partner	Interlocutor que va a realizar la comunicación
direction	Sentido en el que se va realizar la conexión: ENTRADA -> E SALIDA -> S AMBAS -> A VACIAR -> V
domain	Dominio de EDIWIN al que se quiere conectar. OPCIONAL. Valor por defecto la aplicación que este ejecutando el script.
ediwinIni	Fichero de configuración de acceso a EDIWIN. OPCIONAL
True si la operación tiene éxito false en caso contrario	

EBIEdiwinDeleteDocument

Borra en EDIWIN la lista de identificadores de documento, que se recibe como parámetro.

```
public int EBIEdiwinDeleteDocument (   String user,
                                       String password,
                                       String domain,
                                       String[] documentIds,
                                       boolean toTrash )
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
domain	Dominio de EDIWIN al que se quiere conectar.
documentIds	Lista con los identificadores de documentos a borrar
toTrash	Indica si se quiere enviar los documentos a la papelera o no
Si la operación tiene éxito devolverá el número de documentos borrados -2 = Error eliminando documentos -10 = Error conectando a EDIWIN	

EBIEdiwinDeleteVolumeDocument

Borra en el volumen de EDIWIN la lista de identificadores de documentos que recibe como parámetro

```
public int EBIEdiwinDeleteVolumeDocument (String user,
                                           String password,
                                           String domain,
                                           String[] documentIds,
                                           int idVolume)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
domain	Dominio de EDIWIN al que se quiere conectar.

documentIds	Lista con los identificadores de documentos a borrar
idVolume	Identificador del volumen, en el que se encuentran los documentos
Si la operación tiene éxito devolverá el número de documentos borrados -10 = Error conectando a EDIWIN	

EBIEdiwinDeposit

Esta función deposita los documentos que se encuentren en pendiente de enviar según el filtro que se recibe como parámetro

```
public boolean EBIEdiwinDeposit (    String user,
                                   String password,
                                   String filter,
                                   String domain,
                                   String ediwinIni)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
filter	Filtro por el cual se seleccionarán los mensajes que se depositan
domain	Dominio de EDIWIN al que se quiere conectar. OPCIONAL. Valor por defecto la aplicación que este ejecutando el script.
ediwinIni	Fichero de configuración de acceso a EDIWIN. OPCIONAL
True si la operación tiene éxito false en caso contrario	

EBIEdiwinImport

Esta función importará en el servidor EDIWIN, el fichero o directorio que recibe como parámetro.

```
public boolean EBIEdiwinImport( String user,
                                String password,
                                String domain,
                                String file,
                                String ediwinIni,
                                String mode,
                                boolean deleteFile,
                                boolean confirmMessages)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
domain	Dominio de EDIWIN al que se quiere conectar.
file	Fichero o directorio a importar en la estación EDIWIN
ediwinIni	Fichero de configuración de acceso a EDIWIN. OPCIONAL
mode	Indica el sentido de la importación en la estación EDIWIN OUT--> salida IN-> entrada. OPCIONAL. Valor por defecto OUT
deleteFile	Indica si tras una importación con éxito los ficheros deben borrarse o no. OPCIONAL. Valor por defecto false
confirmMessages	Indica si los mensajes que se importen en EDIWIN deben confirmarse en el EBI. Para ello, el nombre de los ficheros que se envíen al EDIWIN, tienen que tener el siguiente formato: xxxxXXXyyyyy_IDMENSAJEEBI.yy La función intentará confirmar el mensaje con el identificador IDMENSAJEEBI
True si la operación tiene éxito false en caso contrario	

EBIEdiwinImportLib

Esta función importa la libreta de direcciones que se le indica como parámetro

```
public boolean EBIEdiwinImportLib ( String user,
                                    String password,
                                    String filename,
                                    String domain,
```

String ediwinIni)	
user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
filename	Fichero con la libreta en formato LI5, a importar
domain	Dominio de EDIWIN al que se quiere conectar. OPCIONAL. Valor por defecto la aplicación que este ejecutando el script.
ediwinIni	Fichero de configuración de acceso a EDIWIN. OPCIONAL
True si la operación tiene éxito false en caso contrario	

EBIEdiwinMant

Esta función lanzará las tareas de mantenimiento del servidor EDIWIN

```
public boolean EBIEdiwinMant (      String user,
                                   String password,
                                   String domain,
                                   String ediwinIni)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
domain	Dominio de EDIWIN al que se quiere conectar. OPCIONAL. Valor por defecto la aplicación que este ejecutando el script.
ediwinIni	Fichero de configuración de acceso a EDIWIN. OPCIONAL
True si la operación tiene éxito false en caso contrario	

EBIEdiwinReceive

Esta función se conectará con el servidor EDIWIN, recuperando la información que se encuentre en systemdir/mask, y depositándola en la ruta local localdir

```
EBIEdiwinReceive ("BATCH", "BATCH", "EDI4", "", "",
                  "\\Edic\\ediwin4data\\buzonesedi\\INTERCAMBIO_EDI\\2_VP_PROD_IN\\*.*",
                  "\\EDICLUSTER\\EDIWIN4_DATA\\DOMINIOS\\EDI4\\ENTRADA\\REC_HTTPS\\", true);
```

```
public boolean EBIEdiwinReceive (String user,
                                 String password,
                                 String domain,
                                 String ediwinIni,
                                 String systemdir,
                                 String mask,
                                 String localdir,
                                 boolean deletesource)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
domain	Dominio de EDIWIN al que se quiere conectar. OPCIONAL. Valor por defecto la aplicación que este ejecutando el script.
ediwinIni	Fichero de configuración de acceso a EDIWIN. OPCIONAL
systemdir	Carpeta de sistema de EDIWIN (ENTRADA ó SALIDA). Accederá al directorio de ENTRADA/SALIDA del dominio
mask	Máscara con los fichero a recuperar
localdir	Ruta local en la que se depositarán los ficheros
deletesource	Indica si se desean borrar los ficheros remotos
True si la operación tiene éxito false en caso contrario	

EBIEdiwinSyncroAddressBook

Esta función permitirá sincronizar la libreta de un dominio EDIWIN desde un dominio maestro.

```
public String EBIEdiwinSyncroAddressBook ( String user,
                                           String password,
                                           String domain,
                                           String masterdomain)
```

user	Usuario del dominio EDIWIN del cual se quiere sincronizar la libreta
password	Contraseña del domino EDIWIN del cual se quiere sincronizar la libreta
domain	Dominio que se quiere sincronizar la libreta
masterdomain	Dominio maestro que se utilizará para sincronizar la libreta
Resultado de la sincronización	

EBIEdiwinTratar

Esta función realiza una llamada al servidor EDIWIN, para ejecutar el comando TRATAR. Si el usuario del script es "BATCH", en la retraducción se consultará si la configuración del interfaz tiene marcado el check de "entrada automática" para lanzar los complementos y pasar a tratado el documento. Con otro usuario se ignorará el check y se hará el cambio de situación cuando se cumpla la condición.

```
public int EBIEdiwinTratar (String user,
                           String password,
                           String filter,
                           String estado,
                           String domain,
                           String ediwinIni,
                           int maxDocuments)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
filter	Filtro que se aplica al comando TRATAR Si no se indica nada en el filtro, la situación de los documentos por defecto es REC (RECIBIDO)
estado	Estado de los documentos a TRATAR. Posibles valores: COR → CORRECTOS (Valor por defecto) ERR → ERRONEOS ALL → TODOS
domain	Dominio de EDIWIN al que se quiere conectar. OPCIONAL. Valor por defecto la aplicación que este ejecutando el script.
ediwinIni	Fichero de configuración de acceso a EDIWIN. OPCIONAL
maxDocuments	Número máximo de documentos a tratar, su valor por defecto es 0, que indica que se quieren tratar todos los documentos. OPCIONAL
0 → El comando TRATAR se ha ejecutado correctamente -1 → Error general de ejecución -2 → El comando TRATAR no se ha ejecutado correctamente	

EBIExtractEDIBinari

Esta función dado un directorio (con o si mascara) intentará extraer los ficheros almacenados en los mensaje del tipo BINARI EDIFACT.

```
public boolean EBIExtractEDIBinari(String inmask, String outdir)
```

inmask	Directorio origen, con o sin mascara, con los ficheros de los cuales se quiere extraer el BINARIO
outdir	Directorio destino donde se depositarán los ficheros extraídos

True si la operación tiene éxito false en caso contrario

EBIExportHeadersReport

Esta función, permite generar un fichero en formato EXCEL o CSV con el contenido de las columnas, de los documentos que cumplan el filtro de búsqueda. Estos documentos pueden estar tanto en la estación como en los volúmenes.

```
public boolean EBIExportHeadersReport(
    String user,
    String password,
    String domain,
    String ediwinIni,
    String filter,
    String columns,
    String fieldMark,
    String valueMark,
    String aliasVolume,
    int idVolume,
    String exportFormat,
    String exportSeparator,
    String destination)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
domain	Dominio EDIWIN, en el cual se quiere generar el listado
ediwinIni	Fichero de inicio, con el que se instanciará el servidor para atender la petición
filter	Filtro para recuperar los documentos
columns	Columnas que se quieren mostrar en el listado
fieldMark	Columna en la cual se quiere aplicar la marca de procesado
valueMark	Valor que se insertará en la columna de marca. Si se quiere dejar vacía, se debe añadir un espacio en blanco
aliasVolume	Identifica el volumen sobre el cual se quiere realizar el listado
idVolume	0 para la estación , otro valor para identificar el volumen
exportFormat	Formato de exportación EXCEL, CSV. Valor por defecto CSV
exportSeparator	Cuando el formato de exportación es CSV, se indicará el separador de columnas, el valor por defecto es ","
destination	Directorio destino donde se generará el listado
True si la operación tiene éxito false en caso contrario	

EBIPublishAS2

Esta función recogerá los mensajes del servidor AS2 y los enviará al EBI, pudiendo aplicar un patrón de búsqueda, bien por el contenido, o bien por el nombre del fichero, para determinar el esquema de publicación en el EBI. En caso de no encontrar ningún patrón, se publicará con el esquema que se reciba en la variable schema.

```
vas2Sever="http://xxx.yyy.zzz/as2server/AS2SERVER/ws";
vas2User="USER";
vas2Password="PASS";
vschema="INTERFAZ_GENERICO_DESADV";
vdestination="EBIACMD";
vwait=false;
vtimeOut=10000;
vreference="";
vduplicate=0;
vkeepfilename=true;
vpatron=
["content",".*ORDER.*","EDI_ORDERS_D_96A_UN_EAN008",".*DESADV.*","INTERFAZ_GENERICO_DESADV_PATRON"];
```

```
vResult=EBIPublishAS2(vas2Sever,vas2User,vas2Password,vschema,vpatron,vdestinati
on,vwait,vtimeOut,vreference,vduplicate,vkeepfilename)
```

Si el envío tiene éxito cambiará de situación el mensaje en el servidor AS2

```
public String EBIPublishAS2 (String      as2Sever,
                             String      as2User,
                             String      as2Password,
                             String      schema,
                             String[]    pattern,
                             String      destination,
                             boolean     wait,
                             int         timeOut,
                             String      reference,
                             int         duplicate,
                             boolean     keepfilename)
```

as2Sever	Dirección IP del servidor AS2
as2User	Usuario AS2 para establecer la conexión web services
as2Password	Contraseña del usuario AS2
schema	Tipo del mensaje de la publicación
pattern	<p>Patrón para determinar el esquema de la publicación. Si no se cumple ningún patrón se aplicará el esquema que se reciba en schema</p> <p>La estructura del patrón será la siguiente</p> <p>[0] content → Localiza el patrón por el contenido filename → Localiza el patrón por el nombre del fichero subject → Localiza el patrón por el asunto del mensaje</p> <p>A continuación por pares</p> <p>[1] Patrón a buscar (basado en las expresiones regulares) [2] Nombre del esquema en el caso de cumplirse el patrón [3] Patrón a buscar (basado en las expresiones regulares) [4] Nombre del esquema en el caso de cumplirse el patrón [5] Patrón a buscar (basado en las expresiones regulares) [6] Nombre del esquema en el caso de cumplirse el patrón</p> <p>Ejemplo: pattern = ["content",".*ORDER.*","ORDERES",".*INV.*","INVRPT"];</p>
destination	Aplicación destino del mensaje
wait	Indica si se debe esperar o no la recolección de todos los mensajes publicados
timeOut	Tiempo máximo de espera en segundos.
reference	Referencia del documento publicado
duplicate	<p>Indica la acción a tomar con las publicaciones duplicadas</p> <p>0: dupIgnore. Se generará una nueva publicación aunque esta ya este duplicada</p> <p>1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH</p> <p>2: dupError. Se devuelve un código de error Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación</p>
keepfilename	Si indica que se conserve el nombre del fichero recibido por el AS2
<p>Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los identificadores de los documentos publicados separados por ' '</p> <p>-1 = Error -2 = Publicación no autorizada -3 = Destino de la publicación desconocido -4 = Documento incorrecto según la validación sintáctica de su interfaz -5 = Aplicación destino no acepta la suscripción -6 = Vence el timeout de espera de recogida de las publicaciones -7 = Usuario no autorizado a publicar -8 = Publicación duplicada</p>	

-10= Esquema no encontrado
 -11= Fallo recuperando el esquema
 -12= Fallo conectando con el servidor AS2

EBIPublishEOS

Esta función recogerá los mensajes del servidor EOS y los enviará el EBI, si el envío tiene éxito cambiará de situación el mensaje en el servidor EOS.

```
public String EBIPublishEOS (
    String eosSever,
    String domainSoap,
    String userSoap,
    String passwordSoap,
    String eosDomain,
    String sslDown,
    String passwordOwn,
    String sfIDown,
    String schema,
    String destination,
    boolean wait,
    int timeOut,
    String reference,
    int duplicate,
    boolean keepFilename)
```

eosSever	Dirección IP del servidor EOS
domainSoap	Dominio SOAP
userSoap	Usuario SOAP
passwordSoap	Contraseña SOAP
eosDomain	Dominio EOS
sslDown	SSID propio
passwordOwn	Contraseña del SSID propio
sfIDown	SFID propio
schema	Tipo del mensaje de la publicación
destination	Aplicación destino del mensaje
wait	Indica si se debe esperar o no la recolección de todos los mensajes publicados
timeOut	Tiempo máximo de espera en segundos.
reference	Referencia del documento publicado
duplicate	Indica la acción a tomar con las publicaciones duplicadas 0: duplgnore. Se generará una nueva publicación aunque esta ya este duplicada 1: dupAccept. No se generará una nueva publicación, ni sus suscripciones correspondientes, y se devolverá el identificador de la publicación que tiene el mismo HASH 2: dupError. Se devuelve un código de error Por publicación duplicada se entiende aquellos documentos, con un mismo HASH dentro de el mismo dominio y aplicación
keepfilename	Si indica que se conserve el nombre del fichero recibido del EOS

Identificador de la publicación en el EBI BROKER. Si se publican varios documentos, se devolverán todos los identificadores de los documentos publicados separados por '|'

-1 = Error
 -2 = Publicación no autorizada
 -3 = Destino de la publicación desconocido
 -4 = Documento incorrecto según la validación sintáctica de su interfaz
 -5 = Aplicación destino no acepta la suscripción
 -6 = Vence el timeout de espera de recogida de las publicaciones
 -7 = Usuario no autorizado a publicar
 -8 = Publicación duplicada
 -10= Esquema no encontrado
 -11= Fallo recuperando el esquema
 -12= Fallo conectando con el servidor EOS

EBIReceiveAS2

Esta función recogerá los mensajes del servidor AS2 y los depositará en el directorio que se le indique como parámetro

```
public String EBIReceiveAS2 (    String directory,
                               String as2Sever,
                               String as2User,
                               String as2Password,
                               boolean originalFileName,
                               boolean confirmMessage)
```

directory	Directorio en el que se depositarán los mensajes
as2Sever	Dirección IP del servidor AS2
as2User	Usuario AS2 para establecer la conexión web services
as2Password	Contraseña del usuario AS2
originalFileName	Indica si se quiere conservar el nombre del fichero original o no
confirmMessage	Se confirman los mensajes en el servidor AS2 una vez descargados y copiados al destino
Identificador/es de los mensajes recuperados de AS2 Si se recogen varios mensajes, se devolverán todos los identificadores de los documentos publicados separados por ' ' -1 = Error -3 = No existe el directorio destino -12= Fallo conectando con el servidor AS2	

EBIRunComplement

Esta función realiza una llamada al servidor EDIWIN, para ejecutar el COMPLEMENTO, que recibe como parámetro

```
public int EBIRunComplement (    String user,
                                String password,
                                String complement,
                                String function,
                                String params,
                                String domain,
                                String ediwinIni)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
complement	Complemento a ejecutar
function	Función de complemento que se quiere ejecutar
params	Parámetros de entrada para la función
domain	Dominio de EDIWIN en el cual se realiza la ejecución del complemento. Este parámetro puede ser opcional, en caso de no venir se asumirá que la aplicación que lo ejecuta, coincidirá con el DOMINIO. OPCIONAL.
ediwinIni	Fichero de inicio, con el que se instanciará el servidor para atender la petición. OPCIONAL.
0 : COMPLEMENTO EJECUTADO -1 : COMPLEMENTO no ejecutando -2 : Error registrando la sesión	

EBIRunIECTE

Esta función realiza una llamada al servidor EDIWIN, para ejecutar la IECTE, que recibe como parámetro.

```
public int EBIRunIECTE (String user,
                        String password,
                        String iecteName,
                        String domain,
                        String ediwinIni)
```


user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
iecteName	Nombre de la IECTE que se desea ejecutar
params	Parámetros de entrada para la función
domain	Dominio de EDIWIN en el cual se realiza la ejecución del complemento. Este parámetro puede ser opcional, en caso de no venir se asumirá que la aplicación que lo ejecuta, coincidirá con el DOMINIO. OPCIONAL.
ediwinIni	Fichero de inicio, con el que se instanciará el servidor para atender la petición. OPCIONAL.
0 : COMPLEMENTO EJECUTADO -1 : COMPLEMENTO no ejecutando -2 : Error registrando la sesión	

EBISendAS2

Esta función enviará al servidor AS2 el fichero que recibe como parámetro

```
public String EBISendAS2 (    String  as2Sever,
                             String  as2User,
                             String  as2Password,
                             String  as2To,
                             String  fileName,
                             String  tanspProt,
                             String  msgOptions,
                             String  mdnOptions,
                             String  subject,
                             String  contentEDI,
                             String  url
                             )
```

as2Sever	Dirección IP del servidor AS2
as2User	Usuario AS2 para establecer la conexión web services
as2Password	Contraseña del usuario AS2
as2To	Dirección AS1/AS2 del destinatario del mensaje
fileName	Fichero con ruta completa que se desea enviar al servidor AS2
tanspProt	Tipo de transporte HTTP, HTTPS, SMTP, FTP
msgOptions	('S','E','C') -> Signed, Ecrypted, Compress.
mdnOptions	('R','S','A=xxxxP=yyyy') -> Request, Signed, AsyncTo=xxx. Parámetros [-Mdn] [-MdnSign] [-MdnA <AsyncProt> <AsyncPort>] * Nota: XXX indica el protocolo de envío del mdn asíncrono [HTTP,HTTPS,SMTP] e YYYY indica el puerto asíncrono de envío.
subject	Asunto añadir al mensaje que se envia al servidor AS2
contentEDI	Permite especificar un content-type específico para el mensaje EDI intercambiado
url	Permite indicar la URL a la que debe enviarse el mensaje
Identificador devuelto por el servidor AS2 -1 = Error -2 = Fallo enviando al servidor AS2	

EBISubsAndImport

Recoge todos los mensajes que estén el EBI BROKER, en estado suscrito, y los importa en la estación EDIWIN.

```
public int EBISubsAndImport (    String user,
                                String password,
                                String schema,
                                int maxSubscriptions,
                                String ediwinIni,
                                String idTransaction,
                                String reference,
                                String mode,
```

String mask)	
user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
schema	Tipo del mensaje del cual se quieren recoger las suscripciones e <u>importarlas</u>
maxSubscriptions	Número máximo de suscripciones a recibir
ediwinIni	Fichero de inicio, con el que se instanciará el servidor para atender la petición. OPCIONAL
idTransaction	Identificador de transacción. OPCIONAL
reference	Referencia del documento a recuperar. Se pueden indicar varias referencias de mensajes, separados por ' '. OPCIONAL
mode	Parámetro opcional para indicar le sentido de la importación (IN/OUT). OPCIONAL
mask	Máscara con los ficheros a enviar a EDIWIN. OPCIONAL
Si > 0 número de suscripciones recibidas -1 Error -2 Suscripción no autorizada -10 Fallo durante la importación a la estación EDI.	

EBISubsAndSendAS2

Esta función recogerá los mensajes suscritos del EBI y los enviará al servidor de AS2

```
public int EBISubsAndSendAS2 (
    String schema,
    int maxSubscriptions,
    String idMessage,
    String idTransaction,
    String reference,
    String as2Sever,
    String as2User,
    String as2Password,
    String as2To,
    String tanspProt,
    String msgOptions,
    String mdnOptions,
    String subjet,
    String contentEDI,
    String url,
    boolean keepFileName,
    boolean confirmSubscription )
```

schema	Tipo del mensaje del cual se quieren recoger las suscripciones
maxSubscriptions	Número máximo de suscripciones a recuperar
idMessage	Identificador del mensaje a recuperar
idTransaction	Transacción a la que pertenece la suscripciones que se están recogiendo
reference	Referencia de la suscripción a recuperar
as2Sever	Dirección IP del servidor AS2
as2User	Usuario AS2 para establecer la conexión web services
as2Password	Contraseña del usuario AS2
as2To	Dirección AS1/AS2 del destinatario del mensaje
tanspProt	Tipo de transporte HTTP, HTTPS, SMTP, FTP
msgOptions	('S','E','C') -> Signed, Ecrypted, Compress.
mdnOptions	('R','S','A=xxxxP=yyyy) -> Request, Signed, AsincTo=xxx. Parámetros [-Mdn] [-MdnSign] [-MdnA <AsyncProt> <AsyncPort>] * Nota: XXX indica el protocolo de envío del mdn asíncrono [HTTP,HTTPS,SMTP] e YYYY indica el puerto asíncrono de envío.
subject	Asunto añadir al mensaje que se envía al servidor AS2

contentEDI	Permite especificar un content-type específico para el mensaje EDI intercambiado
url	Permite indicar la URL a la que debe enviarse el mensaje
keepFileName	Parámetro opcional que nos permite conservar o no el nombre del fichero almacenado en el EBI. Valor por defecto true
confirmSubscription	Se indica si se debe confirmar el mensaje cuando se entregue al servidor AS2 o no. Por defecto el valor es true, con lo cual se confirmará el mensaje una vez se envíe al servidor de AS2, y si se indica como valor false, el mensaje se quedará en pendiente de confirmar, a la espera que se reciba el acuse en el servidor AS2, y se confirme entonces el mensaje
Si > 0 número de suscripciones recibidas y enviadas a AS2 -1 Error -2 Suscripción no autorizada -7 Usuario no autorizado a suscribir -9 Error enviado a As2	

EBISubsAndSendEOS

Esta función recogerá los mensajes suscritos del EBI y los enviará al servidor de EOS

```
public int EBISubsAndSendEOS (String schema,
                             int maxSubscriptions,
                             String idMessage,
                             String idTransaction,
                             String reference,
                             String eosSever,
                             String domainSoap,
                             String userSoap,
                             String passwordSoap,
                             String eosDomain,
                             String ssidDown,
                             String passwordOwn,
                             String ssidExt,
                             String sfIDown,
                             String sfIDExt,
                             String idExtern,
                             String virtualFileName,
                             String format,
                             String recordSize,
                             boolean sign,
                             boolean cipher,
                             boolean compress,
                             boolean signEERP,
                             String cipherSuite,
                             String userData,
                             String virtualFileDesc,
                             boolean keepFileName )
```

schema	Tipo del mensaje del cual se quieren recoger las suscripciones
maxSubscriptions	Número máximo de suscripciones a recuperar
idMessage	Identificador del mensaje a recuperar
idTransaction	Transacción a la que pertenece la suscripciones que se están recogiendo
reference	Referencia de la suscripción a recuperar
eosSever	Dirección IP del servidor EOS
domainSoap	Dominio SOAP
userSoap	Usuario SOAP
passwordSoap	Contraseña SOAP
eosDomain	Dominio EOS
ssidDown	SSID propio
passwordOwn	Contraseña del SSID propio
ssidExt	SSID externo

sfIDown	SFID propio
sfIDExt	SFID externo
idExtern	
virtualFileName	
format	
recordSize	
sign	
cipher	
compress	
signEERP	
cipherSuite	
userData	
virtualFileDesc	
keepfilename	Parámetro opcional que nos permite conservar o no el nombre del fichero almacenado en el EBI. Valor por defecto true
Si >=0 número de documentos recogidos y enviados al EOS -1 Error -2 Suscripción no autorizada -7 Usuario no autorizado a suscribir -9 Error enviado al EOS	

EBISubsAndTransformAndBulkImportD

Esta función realiza una importación en modo BULK al EDIWIN, del resultado de la transformación.

Es necesario que el resultado de la transformación cumpla el esquema XSD EC_Document, ya que este resultado se importará directamente a EDIWIN, sin pasar por los módulos de complementos y el troceador

```
public int EBISubsAndTransformAndBulkImportD (
    String user,
    String password,
    String schema,
    String transformation,
    int maxSubscriptions,
    String transactionId,
    String reference,
    String mask,
    String profile,
    String[] profileEdiwin)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
schema	Tipo del mensaje del cual se quieren recoger las suscripciones
transformation	Transformación aplicar
maxSubscriptions	Número máximo de suscripciones a recibir
transactionId	Identificador de transacción. OPCIONAL
reference	Referencia del documento a recuperar. Se pueden indicar varias referencias de mensajes, separados por ' '. OPCIONAL
mask	Máscara con los ficheros a enviar a EDIWIN. OPCIONAL
profile	Perfil para la ejecución de la transformación
profileEdiwin	Perfil de importación a EDIWIN

Si > 0 número de fichero generados en la transformación

- 1 Error
- 2 Suscripción no autorizada
- 3 Transformación no encontrada
- 4 Error realizando la transformación
- 7 Usuario no autorizado a suscribir y transformar
- 10 Fallo durante la importación a la estación EDI.

EBISubsAndTransformAndImport

Recoge todos los mensajes que estén en el EBI BROKER, en estado suscrito, y les aplica la transformación que recibe como parámetro y los importa a la estación EDIWIN

```
public int EBISubsAndTransformAndImport(String user,
                                         String password,
                                         String schema,
                                         String transformation,
                                         String filter,
                                         boolean generateLog,
                                         int maxSubscriptions,
                                         String ediwinIni,
                                         String transactionId,
                                         String reference,
                                         String mode,
                                         String mask,
                                         String profile)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN
schema	Tipo del mensaje del cual se quieren recoger las suscripciones e <u>importarlas</u>
transformation	Transformación aplicar
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
generateLog	Parámetro obsoleto. Valor por defecto false.
maxSubscriptions	Número máximo de suscripciones a recibir
ediwinIni	Fichero de inicio, con el que se instanciará el servidor para atender la petición. OPCIONAL
transactionId	Identificador de transacción. OPCIONAL
reference	Referencia del documento a recuperar. Se pueden indicar varias referencias de mensajes, separados por ' '. OPCIONAL
mode	Parámetro opcional para indicar le sentido de la importación (IN/OUT). OPCIONAL
mask	Máscara con los ficheros a enviar a EDIWIN. OPCIONAL
profile	Perfil para la ejecución de la transformación
Si > 0 número de fichero generados en la transformación	
<ul style="list-style-type: none"> -1 Error -2 Suscripción no autorizada -3 Transformación no encontrada -4 Error realizando la transformación -7 Usuario no autorizado a suscribir y transformar -10 Fallo durante la importación a la estación EDI. 	

Durante la ejecución de la transformación, esta tendrá disponible, las siguientes variables, con la información del documento suscrito que esta transformando.

```
@EBIID
@EBIDOMAIN
@EBIAPPLICATION
@EBISCHEMA
@EBIDATE
@EBIREFERENCE
@EBISIZE
```

EBISubsAndTransformAndSendAS2

Esta función recogerá y transformará los mensajes suscritos del servidor EBI y los enviará el AS2

```
public int EBISubsAndTransformAndSendAS2(
    String schema,
    String transformation,
    String filter,
    boolean generateLog,
    int maxSubscriptions,
    String idTransaction,
    String reference,
    String profile,
    String as2Sever,
    String as2User,
    String as2Password,
    String as2To,
    String tanspProt,
    String msgOptions,
    String mdnOptions,
    String subje,
    String contentEDI,
    String url,
    boolean confirmSubscription)
```

schema	Tipo del mensaje del cual se quieren recoger las suscripciones
transformation	Transformación aplicar
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
generateLog	Parámetro obsoleto. Valor por defecto false.
maxSubscriptions	Número máximo de suscripciones a recuperar
idMessage	Identificador del mensaje a recuperar
idTransaction	Transacción a la que pertenece la suscripciones que se están recogiendo
reference	Referencia de la suscripción a recuperar. Se pueden indicar varias referencias de mensajes, separados por ' '
profile	Perfil para la ejecución de la transformación
as2Sever	Dirección IP del servidor AS2
as2User	Usuario AS2 para establecer la conexión web services
as2Password	Contraseña del usuario AS2
as2To	Dirección AS1/AS2 del destinatario del mensaje
tanspProt	Tipo de transporte HTTP, HTTPS, SMTP, FTP
msgOptions	('S','E','C') -> Signed, Ecripted, Compress.
mdnOptions	('R','S','A=xxxxP=yyyy') -> Request, Signed, AsyncTo=xxx. Parámetros [-Mdn] [-MdnSign] [-MdnA <AsyncProt> <AsyncPort>] * Nota: XXX indica el protocolo de envío del mdn asíncrono [HTTP,HTTPS,SMTP] e YYYYY indica el puerto asíncrono de envío.
subject	Asunto añadir al mensaje que se envía al servidor AS2
contentEDI	Permite especificar un content-type específico para el mensaje EDI intercambiado
url	Permite indicar la URL a la que debe enviarse el mensaje
confirmSubscription	Se indica si se debe confirmar el mensaje cuando se entregue al servidor AS2 o no. Por defecto el valor es true, con lo cual se confirmará el mensaje una vez se envíe al servidor de AS2, y si se indica como valor false, el mensaje se quedará en pendiente de confirmar, a la espera que se reciba el acuse en el servidor AS2, y se confirme entonces el mensaje
Si > 0 número de suscripciones recibidas y enviadas a AS2 -1 Error -2 Suscripción no autorizada -3 Transformación no encontrada -4 Error realizando la transformación 	

-7 Usuario no autorizado a transformar

-9 Error enviando al servidor AS2

Durante la ejecución de la transformación, esta tendrá disponible, las siguientes variables, con la información del documento suscrito que esta transformando.

@EBIID
@EBIDOMAIN
@EBIAPPLICATION
@EBISCHEMA
@EBIDATE
@EBIREFERENCE
@EBISIZE

EBISubsAndTransformAndSendEOS

Esta función recogerá y transformará los mensajes suscritos del servidor EBI y los enviará el EOS

```
public int EBISubsAndTransformAndSendEOS( String schema,
String transformation,
String filter,
boolean generateLog,
int maxSubscriptions,
String idTransaction,
String reference,
String profile,
String eosSever,
String domainSoap,
String userSoap,
String passwordSoap,
String eosDomain,
String ssIDown,
String passwordOwn,
String ssIDExt,
String sfIDown,
String sfIDExt,
String idExtern,
String virtualFileName,
String format,
String recordSize,
boolean sign,
boolean cipher,
boolean compress,
boolean signEERP,
String cipherSuite,
String userData,
String virtualFileDesc)
```

schema	Tipo del mensaje del cual se quieren recoger las suscripciones
transformation	Transformación aplicar
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
generateLog	Parámetro obsoleto. Valor por defecto false.
maxSubscriptions	Número máximo de suscripciones a recuperar
idTransaction	Transacción a la que pertenece la suscripciones que se están recogiendo
reference	Referencia de la suscripción a recuperar
profile	Perfil para la ejecución de la transformación
eosSever	Dirección IP del servidor EOS
domainSoap	Dominio SOAP
userSoap	Usuario SOAP
passwordSoap	Contraseña SOAP

eosDomain	Dominio EOS
ssidDown	SSID propio
passwordOwn	Contraseña del SSID propio
ssidExt	SSID externo
sfidDown	SFID propio
sfidExt	SFID externo
idExtern	
virtualFileName	
format	
recordSize	
sign	
cipher	
compress	
signEERP	
cipherSuite	
userData	
virtualFileDesc	
keepfilename	Parámetro opcional que nos permite conservar o no el nombre del fichero almacenado en el EBI. Valor por defecto true
Si > 0 número de suscripciones recibidas y enviadas al EOS -1 Error -2 Suscripción no autorizada -3 Transformación no encontrada -4 Error realizando la transformación -7 Usuario no autorizado a transformar -9 Error enviando al servidor EOS	

Durante la ejecución de la transformación, esta tendrá disponible, las siguientes variables, con la información del documento suscrito que esta transformando.

```
@EBIID
@EBIDOMAIN
@EBIAPPLICATION
@EBISCHEMA
@EBIDATE
@EBIREREFERENCE
@EBISIZE
```

EBITransformAndImport

Aplica una transformación sobre los ficheros que se encuentran en el directorio dirIn importando el resultado en el servidor EDIWIN

```
public int EBITransformAndImport ( String user,
                                   String password,
                                   String domain,
                                   String transformation,
                                   String dirIn,
                                   String filter,
                                   boolean generateLog,
                                   String ediwinIni,
                                   String mode,
                                   String mask,
                                   boolean confirmMessages,
                                   String profile)
```

user	Usuario de conexión a EDIWIN
password	Contraseña de conexión a EDIWIN

domain	Dominio sobre el que se van a importar los resultado EDIWIN
transformation	Transformación aplicar
dirln	Directorio de entrada de datos
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
generateLog	Parámetro obsoleto. Valor por defecto false.
ediwinlni	Fichero de inicio, con el que se instanciará el servidor para atender la petición. OPCIONAL
transactionId	Identificador de transacción. OPCIONAL
mode	Parámetro opcional para indicar le sentido de la importación (IN/OUT). OPCIONAL
mask	Máscara con los ficheros a enviar a EDIWIN. OPCIONAL
confirmMessages	Indica si los mensajes que se importen en EDIWIN deben confirmarse en el EBI. . OPCIONAL Para ello, el nombre de los ficheros que se envíen al EDIWIN, tienen que tener el siguiente formato: xxxxXXXyyyyy_IDMENSAJEEBI.yy La función intentará confirma el mensaje con el identificador IDMENSAJEEBI.
profile	Perfil para la ejecución de la transformación. OPCIONAL
Si > 0 número de fichero generados en la transformación -1 Error -2 Suscripción no autorizada -3 Transformación no encontrada -7 Usuario no autorizado a suscribir y transformar -10 Fallo durante la importación a la estación EDI	

getEBIRunComplementResult

Tras la ejecución de la función *EBIRunComplement*, se podrá consultar el valor que ha devuelto esta función a través de esta llamada

```
public String getEBIRunComplementResult()
```

3.3.4 FICHEROS DE LOG

Los ficheros de LOG que genera el adaptador EDIWIN, estarán almacenados dentro del directorio **log** de la aplicación. A su vez, el adaptador EDIWIN, diariamente, creará una carpeta con el año mes y día (yyyymmdd) en la que irá copiando los ficheros de LOG pertenecientes a ese día.

Los ficheros de LOG que genera el adaptador EDIWIN, son los siguientes:

- (2) **LOG de sesión.** El nombre de fichero del LOG de sesión se llama *ebiAdapterEdiwinSession.log*, y este se ira rotando diariamente. En el LOG de sesión, se registrará toda la actividad del adaptador. En el fichero *log4j.properties*, se puede cambiar el nivel de traza el adaptador. El valor por defecto es INFO, pudiéndose aumentar la traza cambiando por TRACE.

```
log4j.logger.com.edicom=INFO, ... → log4j.logger.com.edicom=TRACE, ...
```

- (2) **LOG de errores.** El nombre de fichero del LOG de errores se llama *ebiAdapterEdiwinSession_error.log*, y también se ira rotando diariamente. En este LOG, se registrarán todos los errores de ejecución que se produzcan en el adaptador.

- (3) **LOG de ejecuciones de scripts.** El nombre de fichero del LOG de ejecuciones, tendrá el siguiente formato `idscrip_t_nombrescrip_t_añosmesdiahoraminutosegundomilisegunoaleatorio (yyyymmddhhnnsszzz)`. La marca de tiempo que se añade al fichero de LOG, corresponde con el momento en el cual se ejecuta el script en el adaptador. Existirá un fichero de LOG de este tipo, por cada script que se ejecute.

Los adaptadores utilizan *log4j*, como tecnología para generar los logs, así que se pueden ampliar los manejadores de logs, acorde con las posibilidades de la tecnología *log4j*

3.4 ADAPTADOR SAP

El Adaptador SAP, es un conector, que hereda todas las propiedades y funcionalidades del adaptador de propósito general EDICOM. Este adaptador esta pensado para conectar los sistemas SAP al BUS virtual del EBI BROKER.

Se dispondrá de las siguientes funciones en lenguaje de script, para poder enviar mensajes desde el EBI, al sistema SAP.

- **EBISubsAndImportToSAP.** Recoge todos los mensajes suscritos que se encuentren en el repositorio y los importa en el sistema SAP.
- **EBISubsAndTransformAndImportToSAP.** Recoge todos los mensajes suscritos que se encuentren en el repositorio, les aplica una transformación y los importa en el sistema SAP.
- **EBITransformAndImportToSAP.** Realiza una transformación sobre los ficheros que se le indiquen, y el resultado de la misma lo importa en el sistema SAP.
- **EBIImportToSAP.** Importa en el sistema SAP, los IDOC's que se le indiquen.

El envío de mensajes desde SAP hacia EBI, se realizará a través de un servicio RFC, que pondrá el adaptador de SAP a la escucha, así como el envío de IDOC's desde el adaptador hacia SAP, se realizará a través del RFC,

IDOC_INBOUND_ASYNCHRONOUS

3.4.1 ARQUITECTURA BÁSICA

Se distinguirán dos flujos de transmisión de datos; de **EBI** hacia **SAP** y de **SAP** hacia **EBI**

EBI hacia SAP

La integración de información de EBI hacía SAP se realizará a través del RFC **IDOC_INBOUND_ASYNCHRONOUS**. Este tipo de estructura SAP, será capaz de recibir cualquier tipo de IDOC.

El módulo **IDOC_INBOUND_ASYNCHRONOUS**, esta compuesto por una estructura de control **IDOC_CONTROL_REC_40** y otra de datos **IDOC_DATA_REC_40**.

La inserción de un IDOC en SAP, consistirá en, dado un IDOC, en primer lugar rellenar la estructura IDOC_CONTROL_REC_40 con la información de control, y posteriormente, recorrer el IDOC, e ir insertando, registro a registro, en la estructura de datos IDOC_DATA_REC_40.

La inserción de información en las estructuras de SAP, se realizará a través de llamadas al API de SAP.

Como acuse de entrega del mensaje a SAP, se utilizará el valor de retorno de llamada a la función del API de SAP, que nos permite introducir datos en los módulos de SAP.

Al menos pueden existir 3 escenarios de comunicación entre el adaptador y el sistema SAP:

1. Tanto el adaptador, como el servidor de SAP, residen en la misma red, por lo tanto la comunicación se realiza de forma directa por TCP/IP
2. El adaptador, y el servidor de SAP, residen en redes distintas. En este caso, se establece una VPN para unir el adaptador, a la red en la que esta alojado el servicio de SAP, y una vez unido, la comunicación se realiza vía TCP/IP
3. El adaptador de EBISAP, y el servidor de SAP, residen en redes distintas y se utiliza el SAPROUTE (software perteneciente a SAP), para establecer la comunicación entre ellos. Si la conexión se realiza vía SAPROUTE, es necesario indicar este hecho en las llamadas al API de SAP.

Estructura del RFC IDOC_INBOUND_ASYNCHRONOUS

Tabla IDOC_CONTROL_REC_40

El registro de control de la versión 4 de SAP, tiene la estructura EDI_DC40. Cada IDOC comienza por una estructura de control. Esta estructura contiene información sobre la entrada y salida de IDOC's, así como su procesamiento en el sistema SAP y en los sistemas externos

TABNAM	CHAR 10	Name of table structure
MANDT	CLNT	Client
DOCNUM	CHAR 16	IDoc number
DOCREL	CHAR 4	SAP Release for IDoc
STATUS	CHAR 2	Status of IDoc
DIRECT	CHAR 1	Direction
OUTMOD	CHAR 1	Output mode
EXPRSS	CHAR 1	Overriding in inbound processing
TEST	CHAR 1	Test flag
IDOCTYP	CHAR 30	Name of basic type
CIMTYP	CHAR 30	Extension (defined by customer)
MESTYP	CHAR 30	Message type
MESCOD	CHAR 3	Message code
MESFCT	CHAR 3	Message function
STD	CHAR 1	EDI standard, flag
STDVRS	CHAR 6	EDI standard, version and release
STDMES	CHAR 6	EDI message type
SNDPOR	CHAR 10	Sender port (SAP System, external subsystem)
SNDPRT	CHAR 2	Partner type of sender
SNDPFC	CHAR 2	Partner Function of Sender

SNDPRN	CHAR 10	Partner Number of Sender
SNDSAD	CHAR 21	Sender address (SADR)
SNDLAD	CHAR 70	Logical address of sender
RCVPOR	CHAR 10	Receiver port
RCVPRT	CHAR 2	Partner Type of receiver
RCVPFC	CHAR 2	Partner function of recipient
RCVPRN	CHAR 10	Partner number of recipient
RCVSAD	CHAR 21	Recipient address (SADR)
RCVLAD	CHAR 70	Logical address of recipient
CREDAT	DATS	Created on
CRETIM	TIMS	Time Created
REFINT	CHAR 14	Transmission file (EDI Interchange)
REFGRP	CHAR 14	Message group (EDI Message Group)
REFMES	CHAR 14	Message (EDI Message)
ARCKEY	CHAR 70	Key for external message archive
SERIAL	CHAR 20	Serialization

Tabla IDOC_DATA_REC_40

El registro de datos de la versión 4 de SAP, tiene la estructura EDI_DD40. Cada IDOC contiene un registro de control, seguido de varios registros de datos. Los registros de datos tienen en sus primeros campos la información relativa a la construcción sintáctica del IDOC y asigna los datos al IDOC. El campo SDATA almacenará los datos (hasta 1000 bytes) y los segmentos de servicio se almacenarán en el campo SEGNAM.

SEGNAM	CHAR 30	Segment (external name)
MANDT	CLNT	Client
DOCNUM	CHAR 16	IDoc number
SEGNUM	CHAR 6	Segment Number
PSGNUM	NUMC 6	Number of superior parent segment
HLEVEL	CHAR 2	Hierarchy level of SAP segment
SDATA	LCHR 1000	Application data

SAP hacia EBI

Para establecer la comunicación de SAP hacia EBI, al adaptador pondrá a la escucha un servicio RFC, para que el sistema SAP, pueda enviar los IDOC a EBI, a través de un destino RFC

El adaptador se apoyará en la librerías de SAP **sapidoc3.jar** y **sapjco3.jar** para poner el servicio RFC a la escucha

Para publicar un mensaje en el EBI-BROKER, será necesario conocer la aplicación origen y destino del IDOC, así como el esquema con el que se publicará el IDOC. Para establecer el origen que publica el IDOC, y el destino que lo recibe, se consultarán los valores SDNPOR y RCVPOR del EDI_DC40

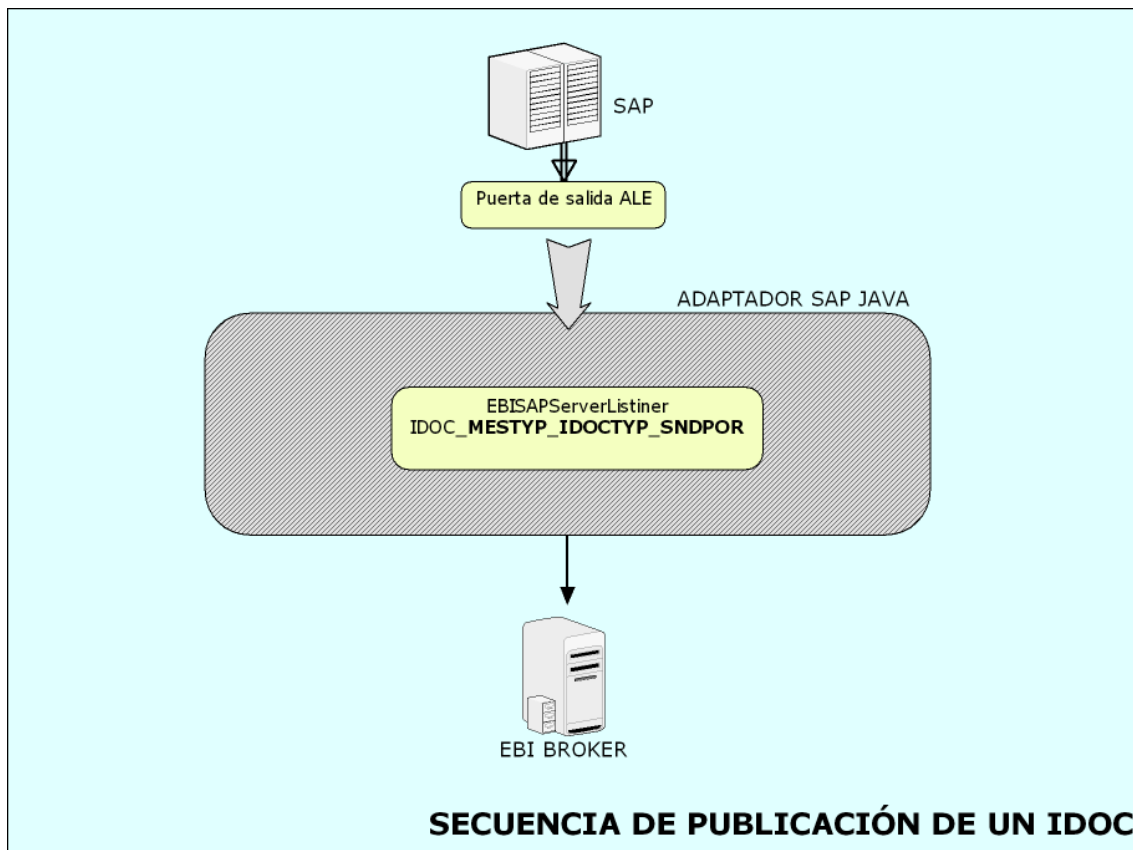
Origen → SDNPOR → Sender port (SAP System, external subsystem)

Destino → RCVPOR → Receiver port

El nombre del esquema con el que se publicará en el IDOC en el EBI, estará formado por los siguientes campos del EDI_DC40:

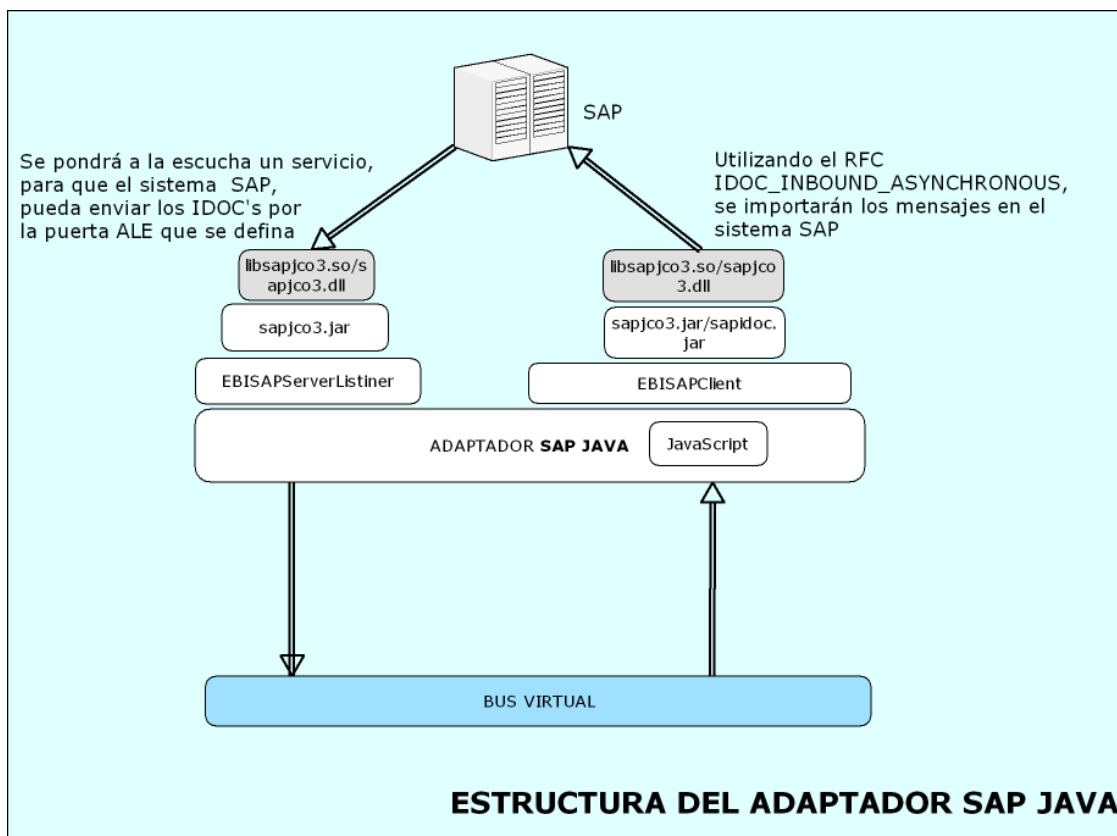
IDOC_MESTYP_IDOCTYP_SNDPOR

La secuencia de publicación de un IDOC, es la siguiente:



3.4.2 ARQUITECTURA DEL ADAPTADOR SAP

Capas en las que está estructurado el adaptador SAP.



3.4.3 INSTALACIÓN

El adaptador SAP en JAVA, a pesar de que Java es independiente de la plataforma de ejecución, las librerías que proporciona SAP para integrar con sus sistema, NO lo son.

Estas librerías de SAP en java, **sapjco3.jar** y **sapidoc.jar**, utilizan a su vez, librerías propias dependientes del sistema operativo sobre el cual se esta ejecutando **sapjco3.dll** y **libsapjco3.so**. Así pues, el grueso del adaptador es el mismo para todas las plataformas, y luego dependiendo del sistema operativo es necesario instalar unas librería u otras.

La distribución del adaptador SAP en JAVA esta estructurada en

ruta_base/ebiasap.jar	Librería que contiene la funcionalidad del adaptador SAP en Java
ruta_base/ebiadapter.properties	Fichero de configuración del adaptador
ruta_base/ebiasap_server.bat	Fichero de arranque del adaptador SAP, para plataformas Windows
ruta_base/ebiasap_server.sh	Fichero de arranque del adaptador SAP, para plataformas Linux
ruta_base/log4j.properties	Fichero en el que se configura el nivel de traza del adaptador
ruta_base/lib	Librerías básicas para el funcionamiento del adaptador
ruta_base/lib/linux/[x86 amd64]	Librerías de SAP, para plataformas Linux de 32 bits (x86) y 64 bits (amd64).
ruta_base/lib/windows[x86 x86_64]	Librerías de SAP para plataformas Windows de 32 bits

	(x86) y 64 bits (x86_64)
--	--------------------------

La librerías de SAP, deberán copiarse al directorio ruta_base/lib, en función de la plataforma, windows o linux, y de la arquitectura, 32 ó 64 bits.

La versión de la máquina virtual de Java, para su funcionamiento, es la 1.6 o superior.

3.4.4 CONFIGURACIÓN

La configuración del adaptador se realiza en el fichero de propiedades **ebiadapter.properties**. Este fichero se encontrará ubicado en la ruta que indique la variable de sistema **user.dir**, que se indicará en el arranque del adaptador.

PROPIEDAD	DESCRIPCIÓN	VALOR POR DEFECTO
ebiadapter.multilInstance	Indica si el adaptador debe conectarse a distintos orígenes de datos	false
ebiadapter.group	Origen de datos asociado al usuario de conexión	
ebiadapter.user	Usuario de conexión	
ebiadapter.password	Contraseña	
ebiadapter.passwordE	Contraseña encriptada	
ebiadapter.type	Tipo de adaptador	EBIAEDIWIN
ebiadapter.name	Nombre con el que se registrará el adaptador en el EBI BROKER	
ebiadapter.domain	Dominio en el que se registra el adaptador	
ebiadapter.application	Aplicación en la que se registra el adaptador	
ebiadapter.url1	URL de conexión al EBI BROKER	
ebiadapter.language	Idioma del adaptador	es
ebiadapter.country	País del adaptador	ES
ebiadapter.deleteDirLogs	Número de días de permanencia de los logs en el directorio	0
ebiadapter.tr	Intervalo en minutos, de reinicio del adaptador, si no hay ningún proceso en ejecución. 0 deshabilitado	0
ebiadapter.tc	Intervalo en minutos de reinicio del adaptador, independientemente, si hay algo en ejecución o no. 0 deshabilitado.	0
ebiadapter.proxyServer	Servidor de PROXY	
ebiadapter.proxyServerPort	Puerto en el que se encuentra el servidor de PROXY	
ebiadapter.proxyUserName	Usuario PROXY	
ebiadapter.proxyPassword	Contraseña del PROXY	
ebiadapter.proxyServerType	Tipo de servidor PROXY	
ebiadaptersapserver.rfcServerEnabled	Habilita/Deshabilita el servicio RFC	

ebiadaptersapserver.system	Identificador del sistema SAP	
ebiadaptersapserver.application Server	Identificador del sistema SAP	
ebiadaptersapserver.sapRouter	SAPRoute para alcanzar el sistema SAP	
ebiadaptersapserver.systemNumber	Número del sistema de SAP	
ebiadaptersapserver.client	Identificador del cliente de SAP	
ebiadaptersapserver.user	Usuario de conexión a SAP	
ebiadaptersapserver.password	Contraseña de conexión a SAP	
ebiadaptersapserver.passwordE	Contraseña encriptada de conexión a SAP	
ebiadaptersapserver.language	Idioma	
ebiadaptersapserver.programId	Identificador del programa	
ebiadaptersapserver.gwService	Gateway service	
ebiadaptersapserver.trace	True o false, para activar o no la traza del servicio RFC	
ebiadaptersapserver.type	Tipo de host remoto A ASHOST (Application Server Host) SYSNR (SAP System Number) Especifica una conexión host-based. B MSHOST (Message Server Host) R3NAME (SAP R3 Name) Especifica una conexión en balanceo de carga, a través de un servidor de mensajes.	A
ebiadaptersapserver.r3name	Identificador del sistema SAP, necesario, cuando se utiliza una conexión balanceada	
ebiadaptersapserver.trimRightIDOC	Si su valor es 1 elimina los espacios en blanco de la derecha del IDOC	
ebiadaptersapserver.SNDPDR	Aplicación origen	
ebiadaptersapserver.RCVPDR	Aplicación origen	
ebiadaptersapserver.IDOCSNDPDR	Emisor que se reflejará en el esquema con el que se publicará el IDOC	
ebiadaptersapserver.tempFiles	Se emplean ficheros temporales en el transito de los documentos de SAP hacia EBI. Esta opción será útil, cuando se transfieran IDOC grandes, con máquina virtuales de Java de 32 bits. 0 → desativado 1 → activado	0
ebiadaptersapserver.duplicates	Control de duplicados en la publicaciones de IDOC desde SAP hacia EBI 0: dupIgnore . Se generará una nueva publicación aunque esta ya este duplicada 1: dupAccept . No se generará una nueva publicación, ni sus suscripciones correspondientes, se devolverá el identificador de la publicación que tiene el mismo HASH 2: dupError . Se devuelve un código de error 3: dupOverride . Se elimina la publicación/suscripción original, y se creará una nueva publicación/suscripción nuevas.	0

El adaptador SAP solo funciona en modo servicio, por lo tanto se conectará al al EBI BROKER, y se quedará a la espera de las planificaciones, que se le indiquen

```
java -Xmx512m -Xms128m -Duser.language=es -Duser.dir=... -Duser.log=...
-Djava.io.tmpdir=... -jar ebiuasap.jar
```


PARÁMETRO	DESCRIPCIÓN	VALOR POR DEFECTO
java	Máquina virtual de Java (versión 1.6 o superior) que ejecuta el adaptador	
-Xmx	Tamaño máximo de la memoria	512m
-Xms	Tamaño mínimo de la memoria	128m
-Duser.language	Idioma en el que funciona el adaptador	es
-Duser.dir	Directorio base del adaptador, donde se encontrarán los ficheros de configuración	\$PWD ó %cd%
-Djava.io.tmpdir	Directorio donde se dejarán los ficheros temporales que utilice el adaptador	\$PWD/tmp ó %cd%\tmp
-Duser.log	Directorio donde se depositarán los ficheros de log	\$PWD/log/ ó %cd%\log\
-jar ebiasap.jar	Jar que contiene el adaptador SAP	

3.4.5 CREACIÓN DEL DESTINO RFC

Para la creación del destino RFC, se utilizará la transacción SM59.

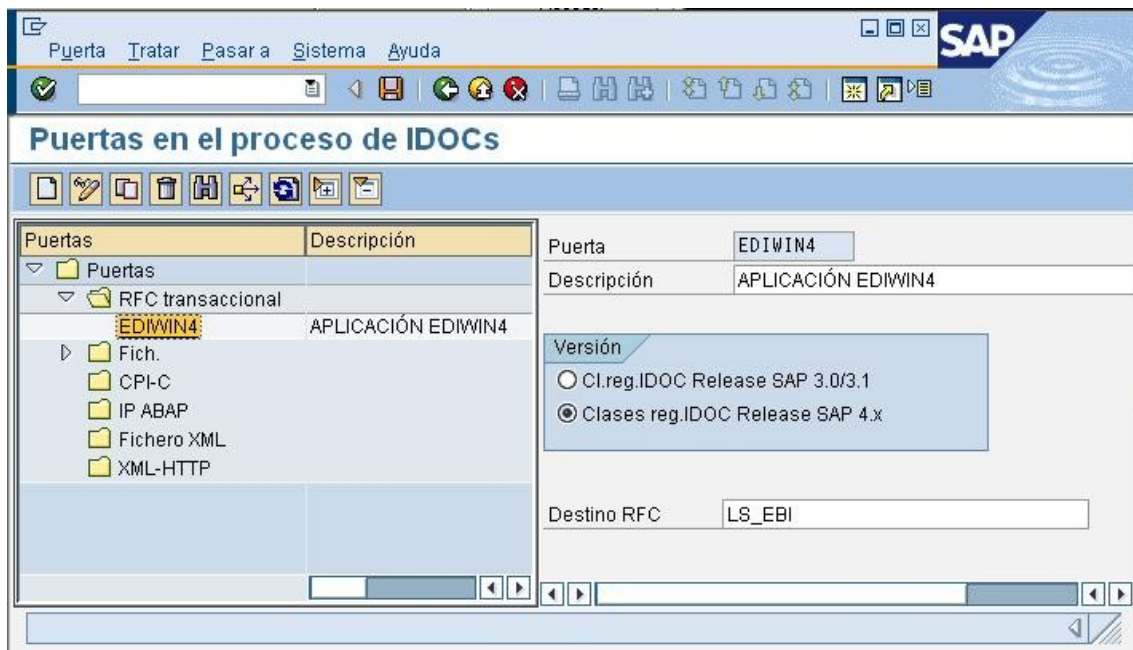


La conexión RFC, la crearemos del tipo TPC/IP, asociándole un identificador de programa (ID programa), que será el que posteriormente se empleará para configurar el conector SAP.

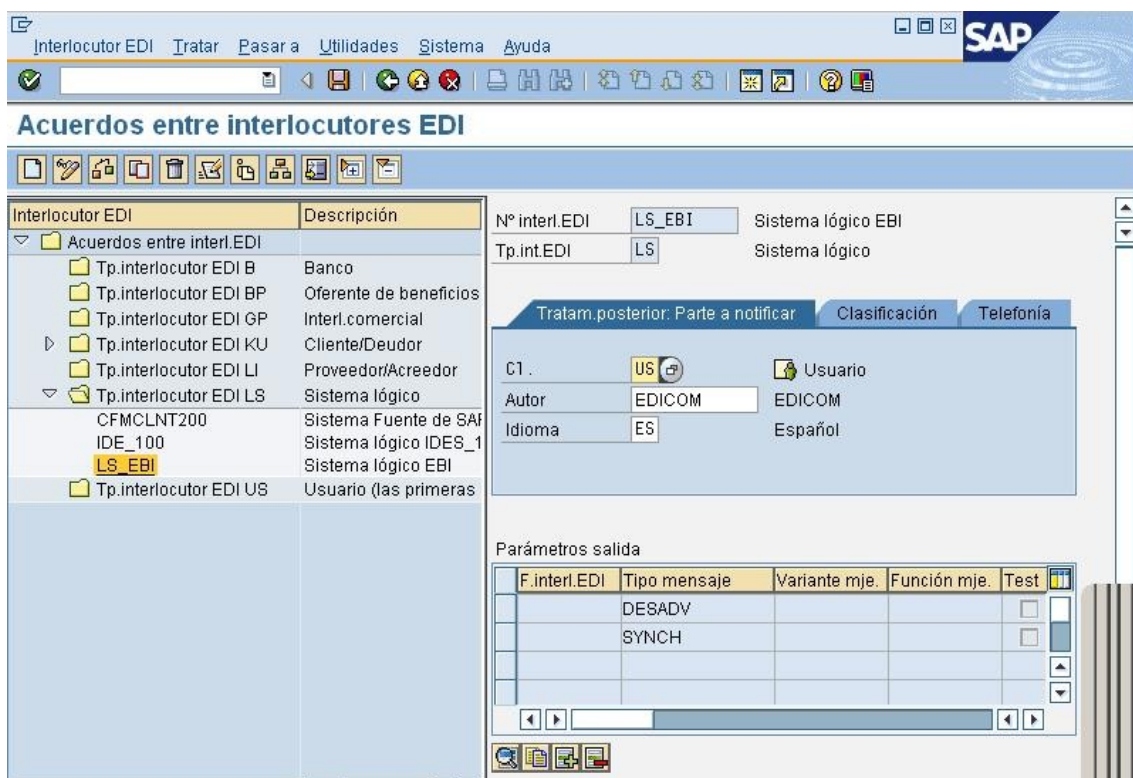
The screenshot shows the 'Destino RFC LS_EBI' configuration window. The 'Destino RFC' field is set to 'LS_EBI'. The 'Tipo conexión' is set to 'T' and 'Conexión TCP/IP'. Under the 'Descripción' tab, 'Descripción 1' is 'Conexión con LS_EBI'. In the 'Forma de activación' section, 'Programa de serv.registrado' is selected. The 'ID programa' field is circled in red and contains the text 'programa.pc'. Below, the 'Opciones gateway' section has empty fields for 'Host gateway' and 'Servicio TCP', and a 'Borrar' button.

3.4.6 CONFIGURACIÓN DE LA PUERTA ALE

Las puertas ALE, se configurarán a través de la transacción WE21. En esta transacción se crearán las puertas, coincidiendo el nombre de la puerta con el RCVPOR.



Una vez, creada la puerta ALE, para asociar esta a un IDOC, se realizará a través de la transacción WE20



Parámetro salida Tratar Pasar a Sistema Ayuda

Acuerdos entre interlocutores EDI: Parámetro de salida

N° interl.EDI: LS_EBI Sistema lógico EBI
 Tp.int.EDI: LS Sistema lógico
 F.int.EDI:
 Tipo mensaje: DESADV Entrega: Aviso de entrega
 Variante de mensaje:
 Función de mensaje:
☐ Test

Opciones salida Control de mensajes Tratam.posterior: Parte a notificar Tel...

Puerta dest.: EDIWIN4 RFC transaccional APLICACIÓN EDIWIN4
 Tam.paq.: 1
☐ Proceso cola
 Modo de salida:
☒ Transf.IDOC inmed. Modo salida: 2
☐ Agrupar IDOCs
 Tipo IDOC:
 Tipo base: DELVR03 Interfase entrega
 Ampliación:

3.4.7 FUNCIONES

EBIImportToSAP

Esta función envía vía RFC'S el fichero que recibe como parámetro, al servidor de SAP

```
public boolean EBIImportToSAP (    String system,
                                String applicationServer,
                                int   systemNumber,
                                String client,
                                String user,
                                String password,
                                String language,
                                String file,
                                String groupName,
                                String sapRouter,
                                String type,
                                String r3name)
```

system	Identificador del sistema SAP
applicationServer	Nombre del equipo que tiene el sistema SAP
systemNumber	Número del sistema de SAP, en el que se va realizar la importación
client	Identificador del cliente de SAP
user	Usuario de conexión de SAP
password	Contraseña de conexión de SAP
language	Idioma 'E'
file	Fichero con ruta completa a importar
groupName	Logon group al que se realizará la conexión. OPCIONAL

sapRouter	Ruta hacia el sistema SAP. OPCIONAL
type	Tipo de la conexión a SAP A, B, E, 2. OPCIONAL
r3name	Si el tipo de conexión es B, se debe indicar el identificador del sistema SAP. OPCIONAL
Si > 0 número de suscripciones importadas en EDIWIN -1 Error -2 Suscripción no autorizada -10 Fallo durante la importación a la estación EDI.	

EBISubsAndImportToSAP

Recoge todos los mensajes que estén el EBI BROKER, en estado suscrito, y los importa en el entorno SAP. Los mensajes una vez recibidos del EBI BROKER, se enviarán vía RFC'S al servidor de SAP.

```
public int EBISubsAndImportToSAP ( String      system,
                                   String applicationServer,
                                   int      systemNumber,
                                   String client,
                                   String user,
                                   String password,
                                   String language,
                                   String schema,
                                   int      maxSubscriptions,
                                   String groupName,
                                   String sapRouter,
                                   String type,
                                   String r3name,
                                   int      maxRetryAttempts)
```

system	Identificador del sistema SAP
applicationServer	Nombre del equipo que tiene el sistema SAP
systemNumber	Número del sistema de SAP, en el que se va realizar la importación
client	Identificador del cliente de SAP
user	Usuario de conexión de SAP
password	Contraseña de conexión de SAP
language	Idioma 'E'
schema	Tipo del mensaje del cual se quieren recoger las suscripciones e importarlas en el sistema SAP
maxSubscriptions	Número máximo de suscripciones a recibir
file	Fichero con ruta completa a importar
groupName	Logon group al que se realizará la conexión. OPCIONAL
sapRouter	Ruta hacia el sistema SAP. OPCIONAL
type	Tipo de la conexión a SAP A, B, E, 2. OPCIONAL
r3name	Si el tipo de conexión es B, se debe indicar el identificador del sistema SAP. OPCIONAL
maxRetryAttempts	Número máximo de reintentos para procesar una suscripción. Si se supera el valor generará un error -14, y se bloqueará la suscripción.
Si > 0 número de suscripciones recibidas e importadas en SAP = -1 Error = -2 Suscripción no autorizada = -10 Fallo durante la importación en el sistema SAP = -11 JCoException exception = -12 IOException exception = -13 JcoRuntimeException = -14 Superado el número máximo de intentos para procesar una suscripción	

EBISubsAndTransformAndImportToSAP

Recoge todos los mensajes que estén el EBI BROKER, en estado suscrito, les aplica la transformación que recibe como parámetro, y los importa en el servidor de SAP

```
public int EBISubsAndTransformAndImportToSAP (
    String system,
    String applicationServer,
    int systemNumber,
    String client,
    String user,
    String password,
    String language,
    String schema,
    String transformation,
    String filter,
    boolean generateLog,
    int maxSubscriptions,
    String transformationIniFile,
    String transactionId,
    String reference,
    String groupName,
    String sapRouter,
    String type,
    String r3name,
    int maxRetryAttempts)
```

system	Identificador del sistema SAP
applicationServer	Nombre del equipo que tiene el sistema SAP
systemNumber	Número del sistema de SAP, en el que se va realizar la importación
client	Identificador del cliente de SAP
user	Usuario de conexión de SAP
password	Contraseña de conexión de SAP
language	Idioma 'E'
schema	Tipo del mensaje del cual se quieren recoger las suscripciones e importarlas en el sistema SAP
transformation	Transformación aplicar
filter	Filtro para el origen FILTRO=REGISTRO[condicion]
generateLog	Parámetro obsoleto. Valor por defecto false.
transformationIniFile	Parámetro obsoleto. Valor por defecto "".
transactionId	Identificador de transacción
reference	Referencia del documento a recuperar
maxSubscriptions	Número máximo de suscripciones a recibir
file	Fichero con ruta completa a importar
groupName	Logon group al que se realizará la conexión. OPCIONAL
sapRouter	Ruta hacia el sistema SAP. OPCIONAL
type	Tipo de la conexión a SAP A, B, E, 2. OPCIONAL
r3name	Si el tipo de conexión es B, se debe indicar el identificador del sistema SAP. OPCIONAL
maxRetryAttempts	Número máximo de reintentos para procesar una suscripción. Si se supera el valor generará un error -14, y se bloqueará la suscripción.
Si > 0 número de suscripciones recibidas e importadas en SAP = -1 Error = -2 Suscripción no autorizada = -10 Fallo durante la importación en el sistema SAP = -11 JCoException exception	

= -12 IOException exception
 = -13 JcoRuntimeException
 = -14 Superado el número máximo de intentos para procesar una suscripción

Durante la ejecución de la transformación, esta tendrá disponible, las siguientes variables, con la información del documento suscrito que esta transformando.

```
@EBIID
@EBIDOMAIN
@EBIAPPLICATION
@EBISCHEMA
@EBIDATE
@EBIREFERENCE
@EBISIZE
```

3.4.8 FICHEROS DE LOG

Los ficheros de LOG que genera el adaptador SAP, estarán almacenados dentro del directorio **log** de la aplicación. A su vez, el adaptador SAP, diariamente, creará una carpeta con el año mes y día (yyyymmdd) en la que irá copiando los ficheros de LOG pertenecientes a ese día.

Los ficheros de LOG que genera el adaptador SAP, son los siguientes:

- (3) **LOG de sesión.** El nombre de fichero del LOG de sesión se llama `ebiAdapterSAPSession.log`, y este se ira rotando diariamente. En el LOG de sesión, se registrará toda la actividad del adaptador. En el fichero *log4j.properties*, se puede cambiar el nivel de traza el adaptador. El valor por defecto es INFO, pudiéndose aumentar la traza cambiando por TRACE.

```
log4j.logger.com.edicom=INFO, ... → log4j.logger.com.edicom=TRACE, ...
```

Si la conexión del servicio RFC, tiene éxito en log de sesión se registrará el echo

```
INFO [main] - Iniciando el servicio SAP
INFO [main] - Propiedades del destino EBI_ABAP_WITHOUT_POOL_SERVER : DEST:
EBI_ABAP_WITHOUT_POOL_SERVER
OWN_HOST:          ED274
PARTNER_HOST:      vm146
SYSTNR:            00
SYSID:             SAD
CLIENT:            100
USER:              EDI_RFC
LANGUAGE:          E
ISO_LANGUAGE:      EN
OWN_CODEPAGE:      4102
OWN_CHARSET:       UTF16
OWN_ENCODING:      utf-16
OWN_BYTES_PER_CHAR: 2
PARTNER_CODEPAGE:  4103
PARTNER_CHARSET:   UTF16
PARTNER_ENCODING:  utf-16
PARTNER_BYTES_PER_CHAR: 2
OWN_REL:           720
PARTNER_REL:       702
PARTNER_TYPE:      3
KERNEL_REL:        720
TRACE:
RFC_ROLE:          C
OWN_TYPE:          E
CPIC_CONVID:       00000000

INFO [main] - Propiedades del servidor EDIWIN : GatewayHost 10.32.24.146,
GatewayService sapgw00, ProgramID EDIWIN
```



```
INFO [main] - El estado del servidor ha cambiado de STOPPED a STARTED en el
servicio con identificador de programa EDIWIN
INFO [JCoDispatcherWorkerThread] - El estado del servidor ha cambiado de
STARTED a ALIVE en el servicio con identificador de programa EDIWIN
```

- (2) **LOG de errores.** El nombre de fichero del LOG de errores se llama *ebiAdapterSAPSession_error.log*, y también se ira rotando diariamente. En este LOG, se registrarán todos los errores de ejecución que se produzcan en el adaptador.
- (3) **LOG de ejecuciones de scripts.** El nombre de fichero del LOG de ejecuciones, tendrá el siguiente formato `idscrip_t_nombrescrip_t_añosmesdiahoramínutosegundomilisegunonoaleatorio (yyyymmddhhnnsszzz)`. La marca de tiempo que se añade al fichero de LOG, corresponde con el momento en el cual se ejecuta el script en el adaptador. Existirá un fichero de LOG de este tipo, por cada script que se ejecute.
- (4) **LOG del servidor RFC.** Si se encuentra activada la traza en el fichero de propiedades, a través de la opción ***ebiadaptersapserver.trace***, en el fichero de LOG ***jco_rfcxxx_yyy.trc*** se registrará, toda la actividad del servidor RFC

Los adaptadores utilizan *log4j*, como tecnología para generar los logs, así que se pueden ampliar los manejadores de logs, acorde con las posibilidades de la tecnología *log4j*

4 ADMINISTRACIÓN EBI.

4.1 INTRODUCCIÓN

El EBI Manager es la interfaz gráfica de usuario de la solución EBI (Edicom Business Integrator), la aplicación de integración de aplicaciones de negocio de EDICOM.

El EBI Pone en contacto distintas aplicaciones a través de un BUS de integración que conecta las distintas aplicaciones y adaptadores que las comunican entre si. El EBI Manager permite al usuario la gestión centralizada de los procesos e intercambios de mensajes que se llevan acabo en el bus, así como la monitorización de dichos procesos con históricos, Logs y alarmas. Por último también la gestión de Usuarios y Objetos (adaptadores).

En "Edicom Suite", la suite de aplicaciones de EDICOM, EBI Manager se ha integrado dentro del panel de control web. Permitiendo al usuario administrador controlar todos los aspectos de la instalación EBI junto al resto de aplicaciones de la Suite.

- Si el usuario tiene experiencia en versiones anteriores de EBI Manager verá que la nueva versión WEB presenta una apariencia y distribución distinta, pero maneja los mismos elementos y conceptos y no presentará ninguna dificultad.
- Si el usuario no tiene experiencia previa con EBI y EBI Manager se recomienda acceder al manual técnico de EBI. Más extenso y detallado.

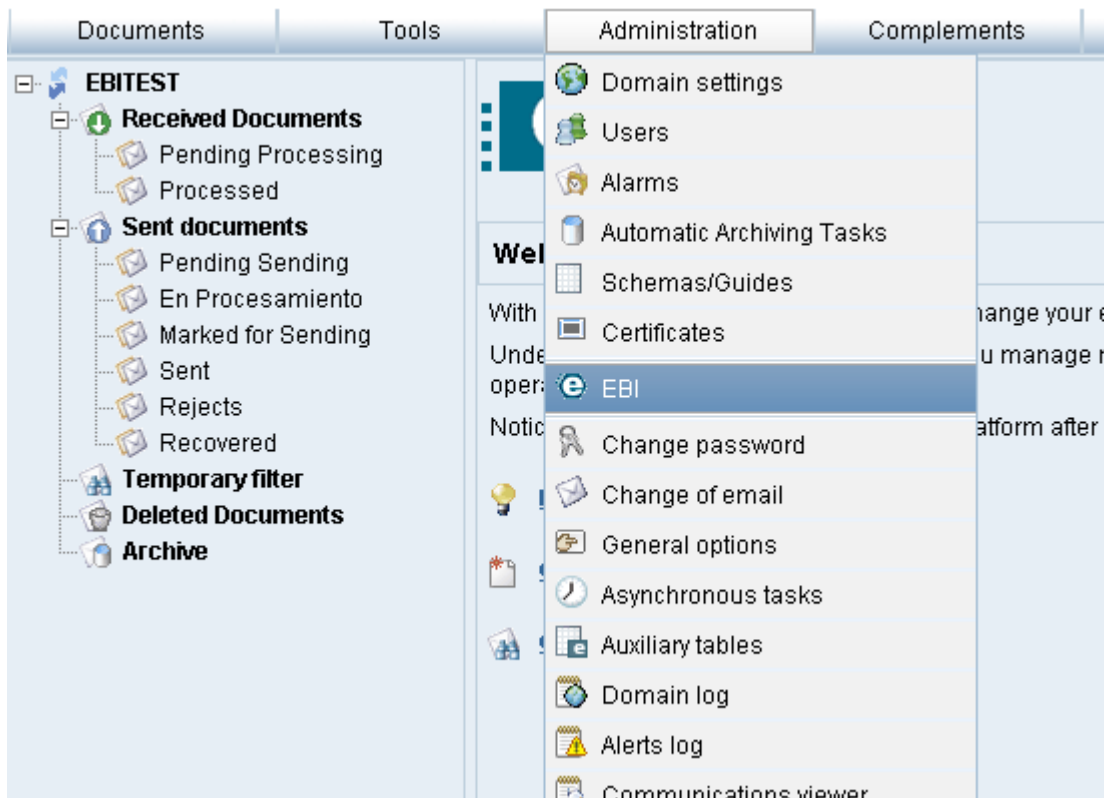
Este manual es un anexo complementario del manual EBI, se centra exclusivamente en la nueva estructura del panel de gestión WEB y la descripción de las funciones que ofrece, pero no sustituye a la lectura del manual de la aplicación EBI el cual sigue siendo la principal referencia de información para las funcionalidades de la aplicación.

Muy Importante: Se recomienda la lectura del manual EBI para nuevos usuarios.

4.2 EBI MANAGER

4.2.1 COMO ACCEDER AL PANEL DE GESTIÓN EBI MANAGER

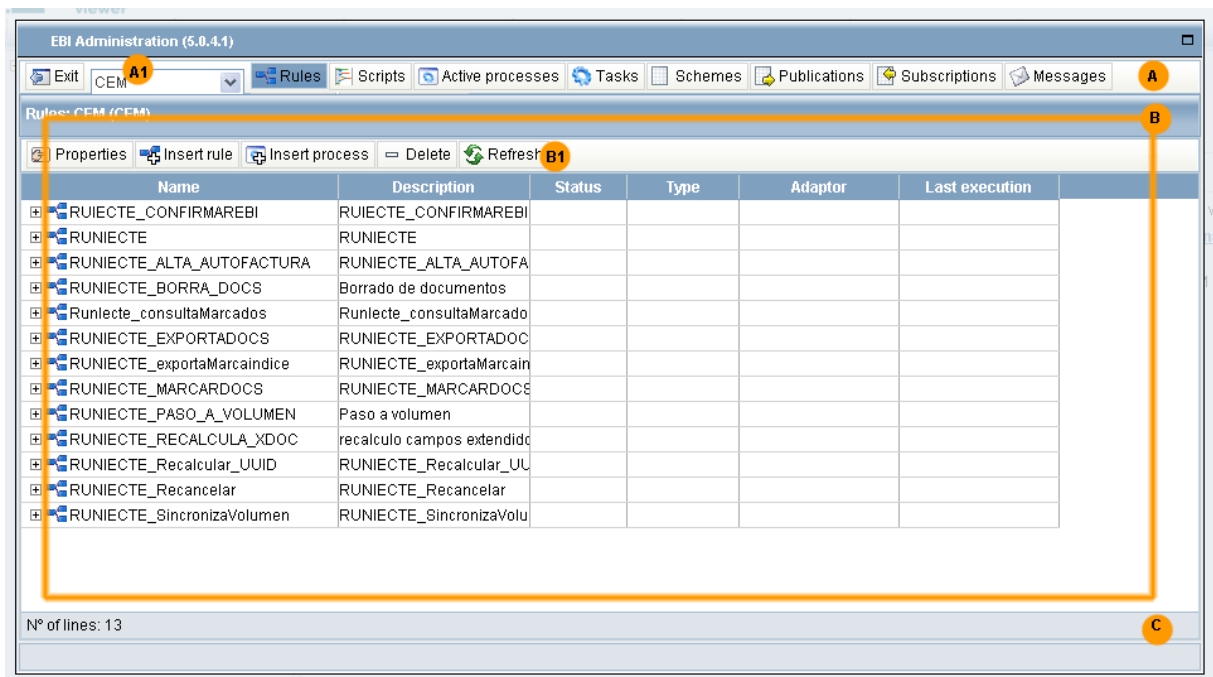
Para acceder al panel de gestión Administración EBI (EBI Manager) primero hay que acceder al panel de gestión WEB de EDIWIN. Tal y como se indica en el manual de Ediwin Suite. El panel de administración EBI es una opción más entre las opciones de Administración del nuevo EDIWIN.



Una vez dentro de la interfaz web. Desde:

- Menú principal **Administración**: opción **EBI**.

Se accede así al diálogo "Administración EBI" distribuido en distintas pestañas y opciones.

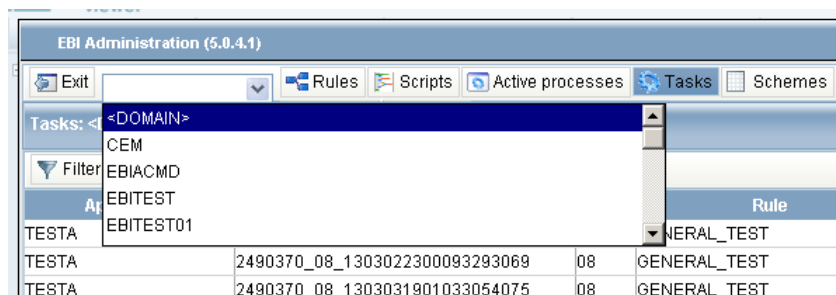


- **A) Menú principal de navegación.** Cada una de estas opciones muestra un gestor distinto. Dentro de este menú se encuentra el seleccionable (**A1**) **Ámbito** que permite seleccionar la aplicación cuya información se va a visualizar. Se trata de una lista con todas las aplicaciones configuradas en el BUS de integración.
- **B) Área principal.** Los distintos diálogos de configuración se muestran aquí, correspondiendo con la pestaña elegida. Cada diálogo o Panel de gestión tiene sus propias opciones situadas en la barra de herramientas (**B1**).
- **C) Barra de estado.** En la barra de estado se muestra información adicional, como el número de elementos visualizado, o también si existen filtros activos en ese momento sobre la vista y hay elementos ocultos.

4.2.2 ÁMBITO; DOMINIO Y APLICACIÓN.

Cada dominio dentro de un servidor representa agrupaciones lógicas de aplicaciones. Solo las aplicaciones de un mismo dominio, se podrán comunicar entre si, utilizando los adaptadores que las conectan al BUS. También aparecerán aplicaciones externas en forma de Alias de aplicación, con los que se pueden gestionar las integraciones entre EBI y otras aplicaciones a través de EBI Manager.

En la lista de aplicaciones se puede elegir que aplicación va a gestionarse y visualizarse en los paneles.














Los distintos paneles de gestión mostrarán la información de cada dominio filtrada por la opción elegida en el ámbito o bien la información conjunta de todas las aplicaciones del dominio (Elegiendo la opción <DOMINIO>). Para cada dominio se podrán crear reglas, enlazar aplicaciones, definir las lógicas de transformación, monitorizar los procesos, ver los mensajes que circulan por el BUS, crear scripts, definir alarmas y, en general, se podrá realizar cualquier operación de EBI a través de los paneles de gestión.

La opción <DOMINIO> permite mostrar todos los elementos de las distintas aplicaciones al mismo tiempo en los paneles (Todos los trabajos, todas las Reglas, etc.). Pero también modifica las pestañas disponibles en el menú principal, ocultando **Publicaciones** y **Suscripciones**, pero mostrando en cambio **Objetos** (Adaptadores) y la gestión de **Usuarios**.

4.3 MENÚ PRINCIPAL DE NAVEGACIÓN



Las opciones del menú principal permiten navegar entre los distintos gestores. Las opciones funcionan como pestañas, modificando el contenido del área principal para mostrar el diálogo correspondiente a cada panel de gestión.

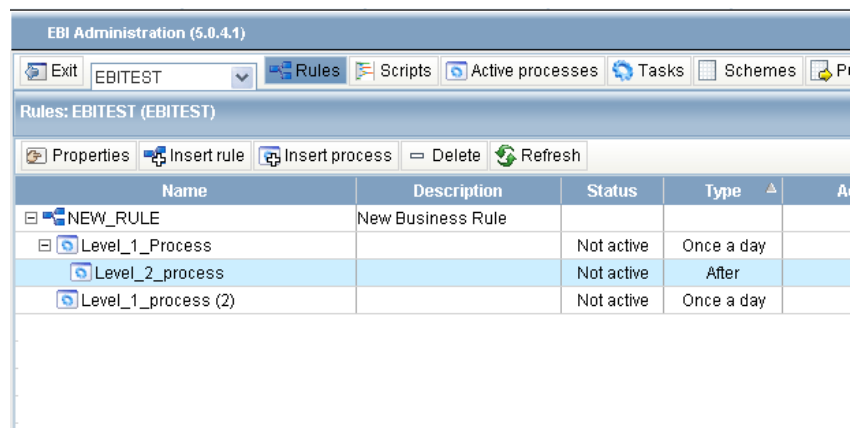
Icono	Descripción
	Reglas. Permite la gestión de reglas de negocio y procesos asociados.
	Scripts. Permite gestionar los scripts configurados.
	Proceso activos. La ventana de procesos activos, dependiendo del ámbito en el cual se este mostrando, <DOMINIO> o aplicación, mostrará todos los procesos que estén en ejecución en ese momento, bien de todo el dominio, bien solo de la aplicación seleccionada.
	Trabajos. Permite la motorización y gestión de los trabajos registrados en el BUS de integración.
	Esquemas. Los esquemas de documento permiten la publicación de los mensajes y también las suscripciones.
	Publicaciones. Permite visualizar y gestionar la lógica de transmisión de mensajes publicados desde el dominio o aplicación.
	Suscripciones. Permite visualizar y gestionar la lógica de transmisión de mensajes suscritos del dominio o aplicación
	Mensajes. Permite visualizar y gestionar los mensajes intercambiados a través del BUS de integración.
	Alarmas. Permite la gestión y creación de notificaciones email a partir de eventos configurables.
	Objetos. Permite la gestión de adaptadores. Disponible solo si el ámbito es <DOMINIO>.
	Usuarios. Permite la gestión de usuarios y sus permisos de accesos. Disponible solo si el ámbito es <DOMINIO>.

4.4 REGLAS

4.4.1 INTRODUCCIÓN

Una regla de negocio es una agrupación lógica de procesos configurados para controlar una determinada integración y el flujo de ejecución de estos procesos, permitiendo definir condiciones de disparo y dependencias entre unos procesos y otros. Los procesos de nivel superior dentro de la regla pueden tener a su vez procesos anidados o dependientes que se ejecutarán cuando finalice la tarea que les precede.

La definición de las reglas de negocio se realiza desde el diálogo de **Reglas**. También aquí se gestiona la definición de los propios procesos. Este diálogo es un listado jerárquico con  **Reglas** como nodos principales y  **Procesos** como elementos anidados a distintos niveles dentro de la Regla. El listado cuenta con la funcionalidad "*Drag and Drop*" para organizar de manera visual los elementos en distintos niveles dentro de la misma regla.



Nota: Los procesos de primer nivel pueden configurarse con condiciones de disparo, los Procesos de nivel 2º o superiores (3º, 4º) Siempre se ejecutan después del proceso precedente. Por eso para ello la condición de disparo se iguala automáticamente a "Después"

Para acceder al gestor de reglas de negocio, desde:

- Menú principal de navegación, pestaña  **Reglas**








Se muestra así el diálogo *Reglas* con las reglas de negocio y sus correspondientes procesos. Las columnas del diálogo *Reglas* muestran información tanto para las Reglas como los Procesos asociados.

- **Nombre.** Indica el nombre de la Regla o Procesos.
- **Descripción.** Indica la descripción dada a la Regla o Proceso.
- **Estado.** Indica el estado del proceso:
 - **Bloqueado.** (Color de fondo Gris) El proceso no se ejecutará aunque se cumplan las condiciones para ello.
 - **Forzado.** (Color de fondo Rojo) El proceso se ejecuta de manera forzada. aunque no se cumplan las condiciones para ello.
 - **No activo.** El proceso está a la espera.
 - **En ejecución.** Esta siendo ejecutado en ese momento.
- **Tipo.** Indica la "condición de disparo" del proceso o su frecuencia de disparo. Si se trata de un proceso de segundo nivel o superior, siempre será "Después de", detrás de la ejecución del proceso precedente.
- **Adaptador.** Indica el tipo de adaptador que controla la ejecución del Proceso.
- **Última ejecución.** Indica la fecha de la última ejecución del Proceso.

Menú.

El diálogo de *Reglas* ofrece las siguientes opciones, distribuidas entre la barra de herramientas y el menú contextual del diálogo (clic derecho del ratón).

Muchas de estas opciones afectan al elemento seleccionado en ese momento en pantalla. Ya sea una regla o un proceso.

Icono	Descripción
	Propiedades. Permite acceder al formulario de configuración de las mismas, según sea el elemento seleccionado en ese momento. <ul style="list-style-type: none"> • Seleccionada una regla. Accede al diálogo <i>Regla</i> con las condiciones de la regla. • Seleccionado un proceso. Accede al diálogo <i>Proceso</i> con las propiedades del proceso.
	Insertar regla. Permite dar de alta una nueva regla. Las reglas son siempre nodos de primer nivel. Muestra el diálogo "Regla" para configurar la nueva regla desde cero.
	Insertar proceso. Permite dar de alta un nuevo proceso. Muestra el diálogo "Proceso" para configurar el nuevo proceso desde cero.
	Duplicar. Permite duplicar el proceso seleccionado (solo se permite duplicar Procesos no reglas). El nuevo proceso se crea dentro de la misma regla y al mismo nivel que el proceso original.
	Borrar. Permite borrar el elemento (Regla o Proceso) seleccionado. Todos los elementos anidados se eliminan también.
	Refrescar. Permite refrescar los estados de los procesos.
	Forzar. Fuerza la ejecución de un proceso seleccionado.
	Bloquear. Bloquea el proceso seleccionado. Si se ejecuta esta opción a nivel de Regla se bloquean todos sus procesos y sub-procesos.
	Desbloquear. Desbloquea el proceso seleccionado. Si se ejecuta esta opción a nivel de Regla se desbloquearán todos sus procesos y sub-procesos que estuviesen bloqueados previamente.
	Mover al primer nivel. Mueve el proceso seleccionado al primer nivel dentro de la regla, moviendo junto a él todos sus procesos anidados.

Nota: Los procesos pueden arrastrarse de una regla a otra o anidarse en otro proceso con la funcionalidad Drag & Drop. Seleccionando un elemento y arrastrándolo dentro de otro.

Propiedades de una regla.

El diálogo *Regla* permite ver y configurar las propiedades de la regla seleccionada o de un nueva regla en el momento de su creación.

- **Nombre.** Indica el nombre de la regla. Este campo no es editable si se accede a través de "Propiedades" de una regla ya creada.
- **Descripción.** Indica la descripción de la Regla.

Una vez terminada la edición, seleccionar **Aceptar** para guardar los cambios.

Propiedades de un Proceso.

El diálogo *Proceso* permite ver y configurar las propiedades del proceso seleccionado o de un nuevo proceso en el momento de su creación.

Algunas de estas opciones solo están disponibles para Procesos de primer nivel. Para los procesos anidados no puede ser configuradas las opciones la condición de disparo, ya que dependen del proceso superior.

- **Nombre.** Indica el nombre del proceso. Este campo es obligatorio.
- **Descripción.** Permite añadir una descripción al Proceso.
- **Adaptador.** Permite seleccionar el Adaptador que ejecutará el proceso.
- **Script.** Permite seleccionar el Script a ejecutar en el proceso. Este campo es obligatorio. El listado seleccionable muestra los Scripts asociados al dominio o aplicación.
- **Notificar error.** Permite marcar la notificación de error al realizar este script. Solo si existe una alarma configurada. Ver [Alarmas](#)
- **Modo Ejecución.** Indica el modo de ejecución de los procesos dentro de la regla.
 - **Encolable.** Si existe algún otro proceso en ejecución dentro de la regla, la ejecución de este proceso se pospondrá hasta que no haya ningún proceso otro proceso en ejecución. (pasará a la cola de procesos)
 - **Concurrente.** El proceso siempre se ejecutará, aunque haya otros procesos en ejecución en ese momento.
 - **Bloqueante.** Si existe algún otro proceso en ejecución dentro de la regla, el proceso no se ejecutará, ni pasará a la cola de procesos.
- **Tiempo máximo de ejecución.** (hh,mm,ss). Indica un límite máximo de ejecución del proceso, Superado el cual el proceso se detiene.

Conjunto de inputs "Condición de disparo".

Importante: Este conjunto de opciones solo está disponible para los procesos de primer nivel, no para los procesos anidados de 2º nivel o superior.

Existen seis condiciones de disparo posibles para la ejecución de un proceso. Cada condición excluye a las demás. A su vez cada condición permite la configuración de distintos parámetros que solamente se mostrarán en pantalla según la opción seleccionada. A continuación se indican las seis condiciones y sus parámetros.

- **Cuando exista el fichero.** Permite lanzar el proceso cuando se detecta un fichero concreto.
 - **Fichero.** Indica el nombre y la ruta donde se esperará encontrar el fichero. La ruta debe ser relativa al adaptador que ejecutará el script.
 - **Borrar.** Indica si se debe borrar el fichero tras la ejecución.

- **Cuando se reciba la suscripción.** El proceso se lanzará cuando se reciba la suscripción seleccionada n numero de veces o cuando finalice el tiempo de espera.
 - **Suscripción.** Permite seleccionar la suscripción de entre las configuradas para la aplicación. Ver [Suscripciones](#).
 - **Suscripciones.** Indica el número de suscripciones mínimo que deben recibirse antes de iniciar el proceso.
 - **Cada.** Indica la referencia temporal (en horas) del tiempo de espera desde que se recibe la suscripción hasta que se ejecuta el proceso.
- **Cuando se confirme la publicación.** Permite lanzar el proceso cuando se confirme la publicación indicada.
 - **Publicación.** Permite seleccionar la publicación de la lista de publicaciones configuradas para la aplicación. Ver [Publicaciones](#).
- **Cuando se rechace la publicación.** Permite lanzar el proceso cuando se confirme que la publicación seleccionada ha sido rechazada .
 - **Publicación.** Permite seleccionar la publicación.
- **Una vez el día.** Permite lanzar el proceso una vez el día indicado.
 - **Fecha.** Indica el día y la hora en que se ejecutará el proceso.
- **Repetidamente.** Permite lanzar el proceso repetidamente a lo largo del día o de la semana con la frecuencia indicada.
 - **Diario.** El proceso se ejecutará diariamente.
 - **Semanal.** El proceso se ejecutará semanalmente los días de la semana elegidos.
 - **Frecuencia diaria.** Este grupo de campos Indica con qué frecuencia a lo largo del día debe ejecutarse el proceso. Válido tanto para la frecuencia diaria como semanal.
 - **Una vez a las...** Indica la hora exacta de una sola ejecución del proceso.
 - **Cada...** Indica la frecuencia entre ejecución y ejecución lo que permite repetir el proceso a lo largo del día repetidamente desde las 00:00 hasta las 23:59 o dentro del plazo indicado por los campos Desde y Hasta.
 - **Desde.** Indica una hora de inicio para la primera ejecución planificada.
 - **Hasta.** Indica la hora final para la ultima ejecución planificada.

- **Intervalos exactos.** Indica la frecuencia entre ejecución y ejecución, sin tener en cuenta la duración de la tarea, con lo que el inicio de la tarea, no acumulará el desfase de la duración de la misma.
- **Desde.** Indica una hora de inicio para la primera ejecución planificada.
- **Hasta.** Indica la hora final para la última ejecución planificada.


Adicionalmente el diálogo *Proceso* muestra campos con información sobre el proceso seleccionado.

- **Última ejecución.** Indica la última fecha y hora de ejecución de este proceso.
- **Ejecuciones.** Indica el número de ejecuciones realizadas de este proceso hasta el momento.

4.5 SCRIPTS


Importante: EBI Manager solo permite la edición de scripts en Java Script. Todos los nuevos scripts se generan por defecto en este lenguaje. En este manual no se explica la sintaxis del lenguaje Java Script.

4.5.1 INTRODUCCIÓN

Los procesos indicados anteriormente consisten principalmente en la ejecución de un determinado Script. Los  Scripts se gestionan independientemente para cada aplicación y las instrucciones programadas en ellos son interpretadas por los adaptador EBI BROKER en el momento indicado por el proceso.

4.5.2 DIÁLOGO SCRIPTS

Para acceder a los Scripts configurados, o programar nuevos scripts, desde

- Menú principal: Opción  **Scripts.**

Se accede así al diálogo *Scripts* con una lista tabular de los scripts configurados y sus opciones de gestión.

Columnas.






En esta lista cada línea corresponde con un script. Cada script tiene las siguientes características reflejadas por las columnas de la tabla.

- **Aplicación.** Indica la aplicación a la que está asociado el script.

- **Nombre.** Indica un nombre para el Script.
- **Descripción.** Indica la descripción del Script.
- **Fecha.** Indica la fecha de la última modificación realizada.
- **Tipo.** Indica el lenguaje utilizado en la sintaxis del script. Actualmente solo se permite la generación de nuevos scripts utilizando la sintaxis Java Script. Pero todavía pueden aparecer scripts en otras sintaxis.

Opciones.

Las opciones de este diálogo se distribuyen entre la barra de herramientas y el menú contextual (botón derecho del ratón). Algunas opciones afectan exclusivamente al elemento seleccionado.

Icono	Descripción
	Editar. Permite acceder al diálogo de edición de "Scripts", para editar el código del script seleccionado.
	Exportar. Permite exportar el Script seleccionado en formato .zip.
	Insertar. Permite insertar un nuevo script en la lista. El nuevo script se asocia automáticamente a la aplicación o domino activo. Desde el diálogo de nuevo script es posible importar un código generado externamente y asociado al nuevo script.
	Duplicar. Permite duplicar el script seleccionado para servir de base a un nuevo Script. Los scripts asociados al valor <DOMINIO> Son plantillas de script, que el usuario puede duplicar como base para sus propios scripts personalizados.
	Borrar. Permite borrar el script seleccionado. (Las plantillas asociadas a <DOMINIO> No pueden borrarse).

4.5.3 INSERTAR SCRIPT.

A través de la opción **Insertar script** se accede al diálogo *Script:(Nombre)*. Este diálogo permite configurar primero las propiedades del nuevo script y después programar el código correspondiente.

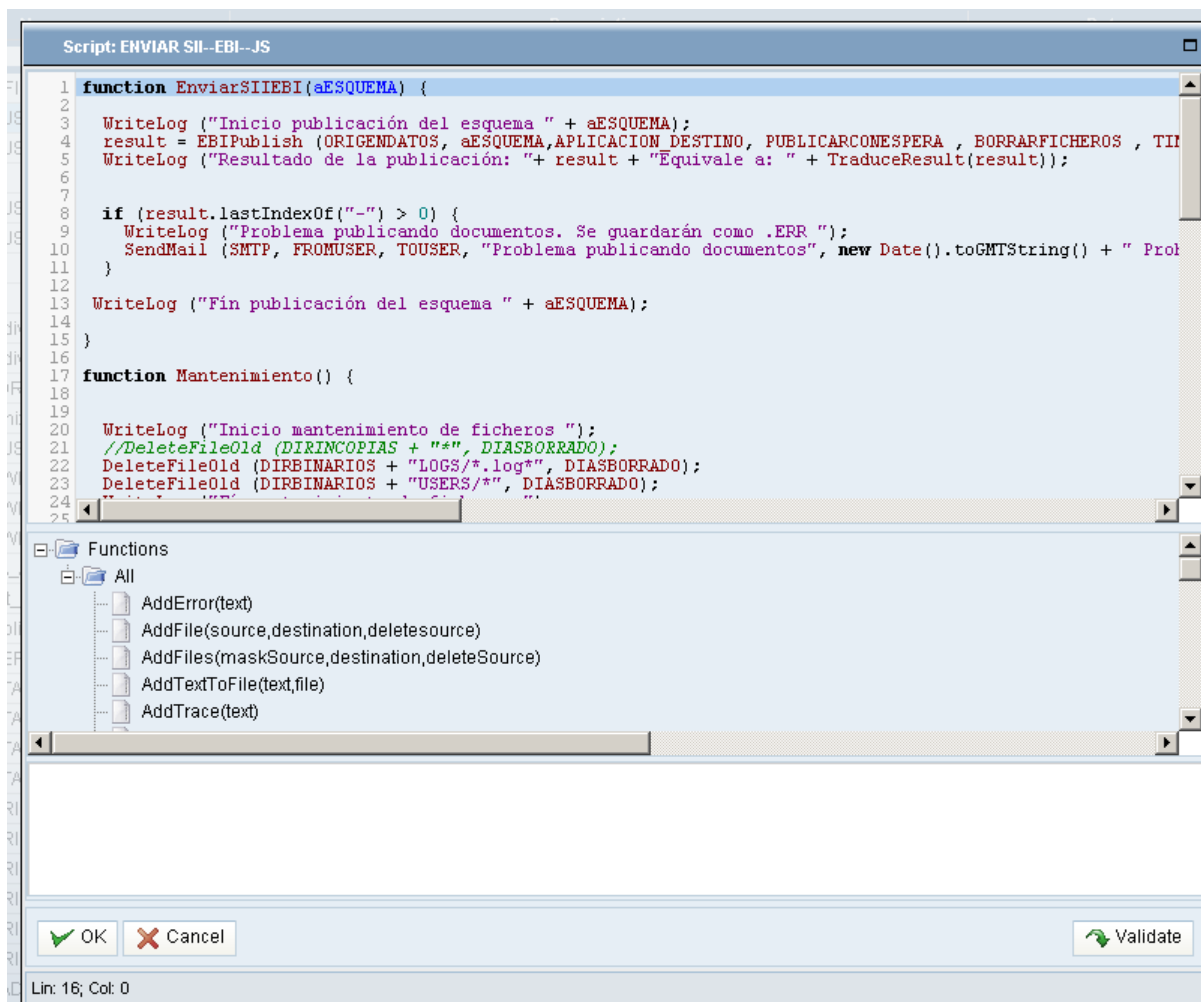
- **Nombre.** Indica el nombre del Script. Este dato es Obligatorio.
- **Descripción.** Indica una descripción para el script. Este dato es Obligatorio
- **Tipo.** Indica la sintaxis del script a generar. Opción restringida en esta versión al lenguaje Java Script.
- **Importar.** Los nuevos scripts se generan con el código en blanco, y en un paso posterior, a través del diálogo de edición, se debe escribir el código correspondiente. La opción **Importar** permite importar directamente fichero externo (en Java Script) con el código ya preparado. Automáticamente se asociará el contenido al nuevo Script.

Editar un script.

Importante: Se recomienda usar siempre Google Chrome o Mozilla Firefox como navegadores para editar scripts.

A través de la opción **Editar** se puede acceder al código del Script seleccionado. La pantalla de edición de scripts consta de dos secciones:

- En la parte superior, el área de edición: la ventana dónde el usuario puede programar el script. Esta ventana resalta la sintaxis Java Script. Si se está visualizando un script indicado como <DOMINIO> en la columna aplicación, no se puede realizar edición alguna. Estos scripts funcionan como plantillas y están pensados para ser duplicados.
- En la parte inferior un listado desplegable de las funciones disponibles en el Adaptador EDICOM preparadas para ser utilizadas en los scripts del usuario. Haciendo clic sobre una función el código se inserta automáticamente en la ventana de código a la altura del cursor. Estas funciones están documentadas en el Manual EBI.



Nota: El resalte de la sintaxis solo funciona bajo Google Chrome o Mozilla Firefox. En Otros navegadores, como Internet Explorer, las funciones de este panel están limitadas.

Validar código.

Una vez programado, el código puede ser validado con la opción **Validar**. La validación detectará errores de sintaxis (Por ejemplo cadenas de texto mal cerradas) pero no errores de programación.

Duplicar Script

La opción duplicar permite generar un nuevo script a partir del script seleccionado en la lista. En el proceso el usuario debe cambiar el Nombre y la Descripción, y tras la duplicación, puede modificar el código original.

Las plantillas de script, las plantillas asociadas al <Dominio> pueden servir como base para nuevos scripts, ya que los scripts duplicados a partir de ellas si son editables. También se puede duplicar cualquier otro script de la lista. Sin embargo al duplicar un script no se puede importar ningún fichero.

4.6 PROCESOS ACTIVOS

4.6.1 INTRODUCCIÓN

El diálogo de *Procesos activos* muestra todos los procesos que estén en ejecución en ese momento asociados a la aplicación seleccionada o todos los procesos en activo si el ámbito seleccionado es <DOMINIO>. Además permite al usuario detener los procesos activos con la opción Detener.

Nota: Es posible que el listado aparezca vacío lo que indicaría que ningún proceso está activo en ese momento dentro del ámbito elegido.

4.6.2 DIÁLOGO PROCESOS ACTIVOS.

Para visualizar los procesos activos, desde:

- Menú principal: Opción  **Procesos Activos**.

Se muestra así el diálogo Procesos Activos, un listado tabular con todos los procesos activos relacionados con la aplicación seleccionada.

4.6.3 COLUMNAS

En este listado cada línea refleja un Proceso activo y cada columna refleja la siguiente información:



- **Identificador**. Indica el identificador del proceso activo.
- **Proceso**. Indica la Regla de negocio en la cual se enmarca el proceso.
- **Script**. Indica el script asociado al Proceso que se está ejecutando.
- **Nombre**. Indica el Nombre dado al proceso dentro de

- **Descripción.** Indica la descripción asociada al proceso activo.
- **Adaptador.** Indica el adaptador encargado del proceso.
- **Última ejecución.** Indica la fecha y hora de la última ejecución completa del proceso activo.
- **Tipo de Proceso.** Indica el tipo de Proceso.
- **Estado.** Indica el estado del proceso activo.
- **Ejecuciones.** Ejecuciones realizadas del mismo proceso.
- **Fecha inicio.** Indica el inicio de la ejecución del proceso.
- **Marca.** Indica la hora de la última actualización de la información visualizada (se actualiza al refrescar). Indicar que es posible que el proceso visible como activo ya no lo esté en el momento. Marca nos muestra cuando se ha tomado la última referencia.

Nota: bajo el ámbito <DOMINO> se visualiza una columna adicional para indicar la Aplicación asociada al proceso.

4.6.4 OPCIONES

Las opciones del diálogo Procesos activos son:

icono	Descripción
	Detener. Permite detener el proceso seleccionado.
	Refrescar. Refresca la información de los procesos activos y actualiza el valor de la columna Marca.

4.7 TRABAJOS


4.7.1 INTRODUCCIÓN

EBI Manager guarda un registro de cada proceso realizado en el BUS de integración. Cada trabajo es el registro de una ejecución independiente de un proceso y sus incidencias.

4.7.2 DIÁLOGO TRABAJOS

El diálogo *Trabajos* permite visualizar este histórico de trabajos y acceder a sus Logs de actividad.

Para acceder al diálogo Trabajos, desde:

- Menú principal: Opción  **Trabajos**.

Se muestra así el diálogo Trabajos con las propiedades de los trabajos realizados en el BUS y permitiendo, con doble clic, acceder a su correspondiente Log.






Columnas

En este listado cada línea refleja un  *Trabajo* y cada columna refleja la siguiente información:


- **Identificador.** identificador único del trabajo formado dinámicamente a partir del identificador del Proceso y la fecha y hora de ejecución.
- **Proceso.** Indica el nombre del proceso ejecutado durante el trabajo.
- **Regla.** Indica la Regla de negocio asociada al proceso ejecutado.
- **Script.** Indica el Script asociado al proceso ejecutado.
- **Fecha Inicio.** Indica la fecha y hora de inicio del trabajo.
- **Fecha Fin.** Indica la fecha y hora de fin del trabajo.
- **Descripción.** Descripción que refleja el resultado del proceso.
- **Código Terminación.** Valor numérico asociado al resultado del proceso:
 - 0 El trabajo ha terminado correctamente
 - -1 Proceso finalizado con errores.
 - -2 Proceso finalizado manualmente.
 - -3 Excepción no controlada durante la ejecución del proceso.
 - -4 Error en la validación del script que ejecuta el proceso.

Opciones



El diálogo Trabajos ofrece las siguientes opciones repartidas entre la barra de herramientas y el menú contextual. Algunas de estas opciones afectan exclusivamente al elemento seleccionado de la lista.

Icono	Descripción
	Filtrar. Permite acceder al diálogo <i>Filtrar</i> .
	Ver Log. Permite acceder al Log del trabajo seleccionado. Acción disponible con doble clic sobre el Trabajo.
	Exportar. Permite exportar el fichero log del trabajo seleccionado.
	Borrar. Permite borrar el trabajo del histórico de trabajos. Esta opción no se puede deshacer.
	Refrescar. Permite refrescar el histórico de trabajos.

4.7.3 FILTRAR

La opción  **Filtrar** permite al usuario definir criterios de filtrado para mostrar únicamente los trabajos que desee. Los criterios de búsqueda pueden referirse tanto a las propiedades visibles en el listado (Columnas) o bien al texto dentro de los logs.

Para acceder al diálogo de filtrado, desde

- Menú contextual. Opción  **Filtrar**.
- Barra de herramientas. Opción  **Filtrar**.

Se accede así al formulario de edición de filtros dividido en dos grupos de opciones. Las opciones cambian según la opción seleccionada.


- **Propiedades.** Muestra las opciones para filtrar por propiedades del trabajo, la información reflejada en el listado. (Es la opción por defecto).
- **Log.** Muestra las opciones para filtrar según al contenido del log.

Si la opción elegida es filtrar por propiedades, los campos mostrados en el formulario son los siguientes.

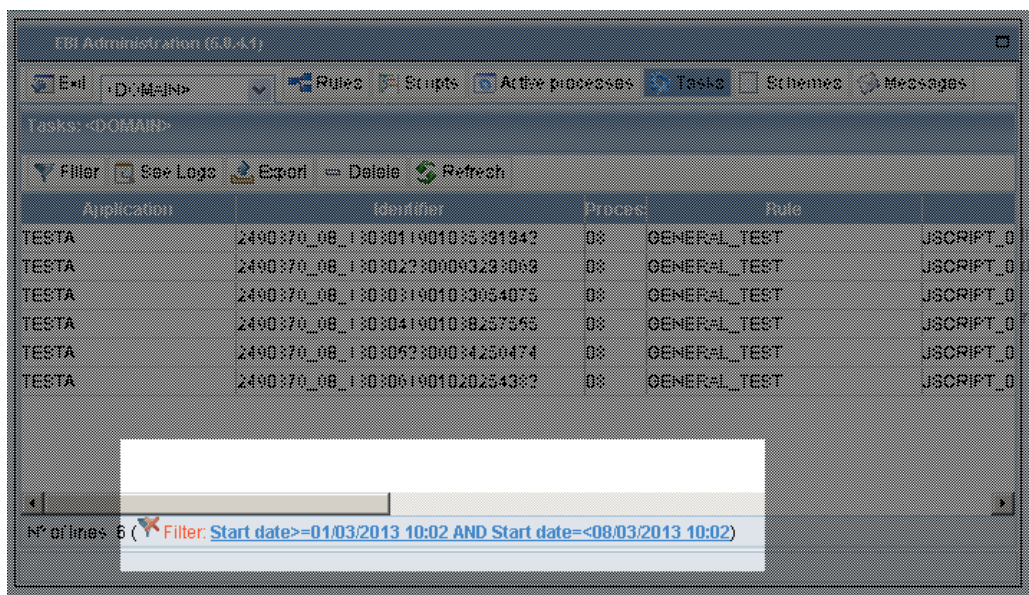
- **Identificador.** Permite filtrar por Identificador.
- **Proceso.** Permite filtrar por el nombre del proceso.
- **Regla.** Permite filtrar por el nombre de la regla.
- **Script.** Permite filtrar por el nombre del script.
- **Descripción.** Permite filtrar por la descripción del trabajo.
- **Fecha Inicio (Desde - Hasta).** Permite filtrar los trabajos por fecha de inicio.
- **Fecha fin (Desde - Hasta).** Permite filtrar los trabajos por fecha de finalización.

Si la opción elegida es filtrar por Log, los campos mostrados en el formulario son solo los siguientes.

- **Log.** Indica cualquier cadena de texto, el filtro mostrará solo aquellos trabajos que presenten la cadena en sus logs.
- **Descripción.** Permite filtrar por la descripción del trabajo.

Una vez configurado el filtro, seleccionar **Aceptar**. El filtro se activará y el listado ocultará los trabajos excluidos. La opción  **Borrar Filtro** permite cancelar el filtro borrando cualquier configuración anterior. La opción **Cancelar** saldrá del formulario sin realizar ningún cambio.

Cuando el filtro está activo, en la barra de estado es posible ver la referencia del **Filtro** activo como indicio de que el usuario no está viendo todos los trabajos. También desde esta referencia en la barra de estado es posible volver a editar los criterios de filtrado o cancelar el filtro.



Nota: Solo se están viendo los trabajos que coinciden con los parámetros del filtro.

4.8 ESQUEMAS

4.8.1 INTRODUCCIÓN

Un esquema de documento es la base para las publicaciones y suscripciones de mensajes.

A la ventana de esquema se puede acceder tanto desde el ámbito <dominio>, como desde el nivel de aplicación. Si se accede a ella desde el ámbito <dominio>, se visualizarán todos los esquemas que existan en ese momento en el dominio (esto incluye a los esquemas de todas las aplicaciones). Pero si se accede a ella desde el nivel de aplicación, se realizará un filtro y solamente se verán los esquemas asociados a dicha aplicación además de aquellos esquemas genéricos disponibles para todas las aplicaciones.

Cuando se hace referencia a un Esquema nos referimos al Nombre, Descripción, Aplicación que lo utiliza y Fecha de creación de este elemento en la aplicación. No necesariamente a una interfaz IN4. Ya que un Esquema puede tener asociado o no un IN4 o la misma interfaz puede estar asociada a varios esquemas distintos en la aplicación o el <dominio>.

Importante: Los esquemas añadidos bajo el ámbito <DOMINIO> se guardarán como genéricos y estarán disponibles para la publicación de cualquier aplicación. Los esquemas añadidos bajo cualquier aplicación solo estarán disponibles para las publicaciones de dicha aplicación.

4.8.2 DIÁLOGO ESQUEMAS

Para acceder al listado de esquemas, desde

- Menú principal: Pestaña  **Esquemas**.

El diálogo *Esquemas* muestra la lista de todas las instancias de esquemas configurados, bien en el dominio o en la aplicación seleccionada. Indicando las referencias más importantes del esquema y permitiendo a su vez activarlo como Disponible para Publicaciones ("Publicable")

Columnas





En este listado cada línea refleja un Esquema y cada columna refleja la siguiente información:

- **Aplicación.** Indica en que aplicación se está utilizando este esquema.
- **Nombre.** Nombre del esquema.
- **Publicable.** Esta columna contiene un check que permite marcar el esquema como disponible para ser utilizado durante una Publicación.
- **Fecha.** Indica la fecha y hora de creación del esquema.
- **Descripción.** Indica una descripción del esquema que complementa al Nombre.

Nota: Al exportar o importar un esquema los ficheros obtenidos aparecen con la extensión IN4 por defecto.



Opciones

El diálogo *Esquemas* ofrece las siguientes opciones repartidas entre la barra de herramientas y el menú contextual. Algunas de estas opciones afectan exclusivamente al elemento seleccionado de la lista.

Icono	Descripción
	Insertar. Permite acceder al Diálogo <i>Esquema</i> para insertar un nuevo Esquema en la lista. El fichero interfaz, de ser necesario, debe Importarse en el momento de la creación.
	Borrar. Permite borrar el esquema seleccionado.
	Exportar. Permite exportar cualquier guía ".IN4" asociada al esquema seleccionado dentro de un fichero comprimido *.zip.
	Refrescar. Permite refrescar la lista de esquemas.

4.8.3 NUEVO ESQUEMA.

El diálogo *Nuevo Esquema* permite dar de alta nuevos esquemas de transformación, nuevas guías de documento, que posteriormente estarán disponibles para la publicación y Suscripción. Para acceder a este diálogo, desde:

- Menú contextual: Opción  **Insertar**
- barra de herramientas: Opción  **Insertar**.

Se muestra así un formulario con los siguientes campos:

- **Nombre.** Permite Indicar el nombre del esquema, El nombre del esquema puede editarse posteriormente. Este valor es Obligatorio.
- **Descripción.** Permite Indicar una descripción complementaria para el esquema.
- **Descripción de los cambios.** Permite Indicar obligatoriamente que cambios han provocado la inserción del nuevo Esquema. Se debe a la gestión de control de cambios de EBIMAP ya que estos esquemas insertados a través de EBI Manager se guardan en el repositorio de datos de EBIMAP.
- **Publicable/Suscribible.** Permite crear el nuevo esquema directamente con la casilla **Publicable** activada, aunque esta opción puede activarse o desactivarse posteriormente. Los esquemas marcados son aquellos que aparecen como disponibles en el gestor de Publicaciones. (Ver [Publicaciones](#))
- **Importar.** Permite importar el fichero interfaz que deberá asociarse al nuevo esquema. Por ejemplo un IN4.

Una vez indicados los datos, seleccionar **Aceptar**, para ver el nuevo esquema en la lista.

4.9 PUBLICACIONES

4.9.1 INTRODUCCIÓN

Una publicación es la emisión de un mensaje desde la aplicación seleccionada al resto de aplicaciones a través del BUS de Integración.

La publicación requiere de un Esquema de mensaje, como base. Lo que indica el tipo de mensaje que va a ser publicado.

Una aplicación solo puede realizar publicaciones de mensajes asociados al <DOMINIO> o la propia aplicación. Y siempre que estén señalados como "publicables" en el gestor de esquemas.

4.9.2 DIÁLOGO PUBLICACIONES

El diálogo publicaciones solo estará disponible bajo el ámbito de una aplicación determinada, El ámbito <DOMINIO> no dispone de esta opción.

Para acceder a este diálogo, seleccionar la aplicación que va a publicar el mensaje, y desde:

- Menú principal: Opción  **Publicaciones**.

Aparece así la lista de publicaciones activas desde la aplicación.

Columnas

En este listado cada línea refleja una publicación y cada columna refleja la siguiente información:

- **Aplicación.** Indica el nombre de la aplicación asociada al esquema base de la publicación, que será la seleccionada o <DOMINO> para los esquemas genéricos.
- **Nombre.** Indica el nombre del esquema base para la publicación.
- **Validar.** Cuando un esquema contiene una estructura se puede realizar una validación sintáctica previa a la publicación, para comprobar si el mensaje es sintácticamente correcto antes de enviarlo al BUS. Para realizar esta validación seleccionar la casilla **Validar**.

Opciones

El diálogo Publicaciones ofrece las siguientes opciones repartidas entre la barra de herramientas y el menú contextual. Algunas de estas opciones afectan exclusivamente al elemento seleccionado de la lista.

Iconos	Descripción
	Insertar. Permite insertar una nueva publicación, eligiendo previamente el esquema que servirá como base. Se mostrará el diálogo <i>Nueva Publicación</i>
	Borrar. Permite eliminar la publicación seleccionada.

4.9.3 NUEVA PUBLICACIÓN.

El diálogo Nueva publicación permite añadir una nueva publicación en la aplicación seleccionada.

Para dar de alta una nueva publicación, desde:

- Barra de herramientas: opción **Insertar**.
- Menú contextual: opción **Insertar**.

Se accede así a una lista de "Esquemas publicables"; una lista con los esquemas disponibles para publicación. Los esquemas disponibles son aquellos marcados como "publicables", asociados a la aplicación o al <DOMINIO> y que todavía no hayan sido elegidos para una publicación.

Tras seleccionar el esquema y aceptar, la nueva publicación se visualizará junto al resto.

La opción **Borrar** simplemente indica que la aplicación dejará de publicar el esquema de mensaje seleccionado. Siendo posible volverlo a añadir más tarde.

4.10 SUSCRIPCIONES

4.10.1 INTRODUCCIÓN

Tras la publicación de un mensaje en el BUS, este será distribuido a todas las aplicaciones que se encuentren suscrito al mismo. Las aplicaciones podrán suscribirse a todos los esquemas publicados por cualquier aplicación que este dentro del dominio.

4.10.2 DIÁLOGO SUSCRIPCIONES

Cada aplicación debe suscribirse independientemente las publicaciones de otras aplicaciones, por lo que esta pestaña no está disponible si el ámbito es <DOMINIO>.

Para acceder al diálogo suscripciones, desde:

- Menú principal: Pestaña  **Suscripciones**.

Se accede así a las suscripciones que recibe la aplicación.




Columnas

En este listado cada línea refleja una publicación y cada columna refleja la siguiente información:

- **Aplicación.** Indica el nombre de la aplicación asociada al esquema.
- **Nombre.** Indica el nombre del esquema.

Opciones

El diálogo Suscripciones ofrece las siguientes opciones repartidas entre la barra de herramientas y el menú contextual.

Icono	Descripción
	Insertar. Permite dar de alta una nueva suscripción en la aplicación seleccionada. Accediendo previamente al diálogo Esquemas para elegir el esquema base de la publicación a suscribirse.
	Borrar. Permite borrar la suscripción. La aplicación ya no recogerá dicho mensaje del bus.
	Refrescar. Permite refrescar la lista de suscripciones.

4.10.3 INSERTAR NUEVA SUSCRIPCIÓN

El gestor de suscripciones permite insertar nuevas suscripciones a mensajes desde la opción Insertar del menú contextual o la barra de herramientas. Con la opción Insertar se accede a la lista de Esquemas disponibles en el dominio,

Para ello se debe seleccionar el esquema a suscribirse de la lista de Esquemas disponibles para suscripción. Solo aparecen los que todavía están pendientes de suscripción, es decir aquellos para los que no existe ya una suscripción activa en esta aplicación.

Una vez seleccionado el esquema, **Aceptar** para ver la nueva suscripción junto al resto. Borrar una suscripción significa simplemente que la aplicación ya no tendrá en cuenta los mensajes de este esquema que circulen por el BUS.

4.11 MENSAJES


4.11.1 INTRODUCCIÓN



A la ventana de mensajes se puede acceder tanto desde el dominio, como desde una aplicación. Si se accede a ella desde el ámbito <DOMINIO>, se visualizarán todos los mensajes que existan en ese momento en el BUS (esto incluye a los mensajes de todas las aplicaciones). Si se accede a ella desde el ámbito de una aplicación, se realizará un filtro y solo se verán los mensajes donde esta aplicación elegida sea Origen o Destino.















4.11.2 DIÁLOGO MENSAJES

Para acceder a este listado de mensajes. desde:

- Menú Principal: Pestaña  **Mensajes**.

Se accede así al diálogo *Mensajes*. Una lista tabulada que muestra tanto mensajes Publicados como Suscritos. Podemos ver el contenido del mensaje haciendo doble clic sobre el o con la opción  **Ver Contenido**.

Cada elemento principal de esta lista corresponde con un mensaje  *Publicado* o  *Suscrito* y sus propiedades más relevantes. A su vez dentro de cada mensaje hay un listado desplegable con sub-elementos que, como si fuera un histórico, reflejan en detalle los distintos estados de validación, Cuando ocurrieron y que aplicación interviene. Ver [Estados de validación de un mensaje](#) para más información.








Messages: <DOMAIN>				
 Filter  Show  Export  Delete  Refresh				
	Type		Identifier	Mes
		Publication notified	1207111103218162557	LISTADO_FALTANTESIND_CRP
	Type	Date		
		Published	11/07/2012 13:03:21	SUPERVISOR-pl
		Publication confirmed	07/08/2012 17:08:06	TESTA-Axis/1.4-1
		Publication notified	07/03/2013 13:10:37	EBITEST-Axis/1.4
		Publication confirmed	1302131008	PRUEBA01_TESTA
		Confirmed	1302131008155885258000	PRUEBA01_TESTA

En cualquier momento se puede acceder al contenido del mensaje, aunque el modo de presentación dependerá del tipo de mensaje, la aplicación involucrada y el sentido de la transmisión (Un Pdf, El código EDI, Los ficheros de integración, etc.)

Columnas

En esta lista cada fila corresponde a un mensaje y sus propiedades se reflejan por las siguientes columnas:

- **Tipo.** Refleja el tipo del mensaje usando un icono como apoyo visual. En los elementos principales refleja el último estado de acuse de la publicación o suscripción. En los elementos anidados refleja los distintos estados atravesados por el mensaje hasta el estado final, como si fuese un histórico.

Icono	Descripción
	Publicado.
	Publicación confirmada.
	Publicación notificada.
	Suscrito
	Mensaje suscrito pendiente de confirmación
	Mensaje confirmado.
	Mensaje bloqueado.

- **Identificador.** Indica el identificador único del mensaje. Formado dinámicamente a partir de la fecha de publicación/suscripción
- **Mensajes.** Indica el esquema asociado a la publicación o suscripción del mensaje.
- **Fecha.** Indica la fecha del último estado del mensaje.
- **Origen.** Indica la aplicación de Origen del mensaje.
- **Destino.** Indica la aplicación de Destino del mensaje.
- **Referencia.** Indica la referencia o número de documento (si existe.) Estos valores de referencia se extraen del propio contenido del documento.
- **Tamaño.** Indica el tamaño en KB del mensaje.
- **Suscripción.** Indica las veces que se ha intentado realizar la suscripción, Lo normal es que aparezca 0 y cuando se procese cambie a 1 y a estado Confirmado. Si hay algún problema durante la suscripción, esta se intentará re-procesar hasta un número máximo de veces preestablecido, incrementando este valor. Cuando este valor supera el número máximo de intentos de suscripción, la suscripción pasa a "Bloqueada"

Estados de validación de un mensaje





- **Publicado.** El estado de "publicado" en un mensaje corresponde con el estado del mensaje tras una publicación por parte de un adaptador conectado al BUS.











- **Publicación confirmada.** El estado Publicación confirmada corresponde a cuando todas las suscripciones asociadas a la publicación pasan a estado confirmado, la publicación cambia de estado publicado a publicación confirmada.
- **Publicación notificada.** El estado Publicación notificada corresponde a cuando la aplicación que realizó la publicación del mensaje, es notificada que dicho mensaje ha sido entregado a todos sus destinatarios/suscriptores. En ese momento la aplicación cambia el estado del mensaje a publicación notificada.
- **Suscrito.** Tras una publicación, las reglas de negocio del Adaptador EBI Broker replicarán el mensaje publicado a todas las aplicaciones suscritas al mismo. Cuando se realiza esta replicación, el estado del nuevo mensaje es Suscrito.
- **(Suscripción) Pendiente de confirmar.** Cuando una suscripción está en proceso, siendo entregada a su aplicación destinataria, esta se marca con el estado temporal "Pendiente de confirmar". Este estado debería ser transitorio, y durar el tiempo que se tarde en lograr entregar el mensaje a la aplicación destinataria.
- **(Suscripción) Confirmado.** Cuando la aplicación destino de la publicación recibe el mensaje al que estaba suscrito, el mensaje cambia de estado y pasa a ser "confirmado".
- **Bloqueado.** Cuando un mensaje se intenta recoger mas de N veces sin éxito, el EBI BROKER lo bloquea, pasando su estado a "bloqueado". El número de veces que puede ser recogido un mensaje, antes de bloquearse, vendrá determinado en la configuración del EBI BROKER.

Opciones

El diálogo *Mensajes* ofrece las siguientes opciones repartidas entre la barra de herramientas y el menú contextual. Algunas de estas opciones afectan exclusivamente al elemento seleccionado de la lista.

Algunas opciones son generales, otras opciones se muestran dependiendo del elemento seleccionado. Por ejemplo la opciones de publicación no estarán visibles en el menú contextual si se selecciona un mensaje Suscrito.

Icono	Opción.
	Filtrar. Permite realizar un filtrado de los mensajes a través de las opciones del diálogo "filtrar".
	Exportar. Exporta el mensaje y los ficheros asociados al disco duro en un fichero comprimido zip.
	Ver. Permite la visualización del contenido de los ficheros asociados al mensaje, Esta Visualización depende siempre de la aplicación y el tipo de mensaje. Pudiendo ser el código fuente, el contenido textual del mensaje o una previsualización PDF a través de un Informe.
	Borrar. Borra el mensaje seleccionado de la lista.



	Refrescar. Refresca el listado de mensajes
	Publicar. Permite cambiar el estado del mensaje a Publicado.
	Publicación confirmada. Permite cambiar el estado del mensaje a Publicación Confirmada.
	Publicación notificada. Permite cambiar el estado del mensaje a Publicación notificada.
	Publicación Rechazada. Permite cambiar el estado del mensaje a Publicación Rechazada.
	Publicación rechazada Notificada. Permite cambiar el estado del mensaje a Publicación Notificada.
	Suscribir. Permite cambiar el estado del mensaje a Suscrito.
	Confirmar. Permite cambiar el estado del mensaje Suscrito a Confirmado.
	Bloquear. Permite cambiar el estado del mensaje Suscrito a Bloqueado.
	Rechazar. Permite cambiar el estado del mensaje Suscrito a Rechazado.

Importante: Cambiar de estado el mensaje manualmente desde el menú contextual no implica necesariamente ningún tratamiento del mensaje. Pero sí puede iniciar cualquier regla asociada a estos estados. El cambio de situación provocará la ejecución de la regla y los procesos asociados.

4.11.3 FILTRAR MENSAJES

El diálogo *Filtrar* contiene un formulario que permite establecer los parámetros del filtro para visualizar en pantalla solo los mensajes relevantes para el usuario.

Para acceder al diálogo filtrar, desde:

- Menú contextual: opción  **Filtrar**
- Barra de herramientas: opción  **Filtrar**

Los campos del formulario de filtro son:

- **Identificador.** Permite filtrar por el contenido del campo identificador.
- **Tipo.** Permite filtrar por el contenido del campo Tipo del mensaje.
- **Mensajes.** Permite filtrar por el contenido del campo Mensajes.
- **Referencia.** Permite filtrar por el contenido del campo Referencia.
- **Origen.** Permite filtrar por Origen.
- **Destino.** Permite filtrar por Destino.
- **Fecha.** Permite filtrar por el contenido de la columna Fecha. Este parámetro esta formado por dos campos de tipo fecha; es posible elegir la fecha *Desde* (completando solo el primer campo) o la fecha *Hasta* (completando solo el segundo campo), pudiendo establecer un intervalo completando ambos.

Una vez configurado los parámetros del filtro, aceptar. Se mostrarán en pantalla solo los documentos coincidentes. En la barra de estado se muestra el filtro activo como

indicio que el usuario no está viendo todos los mensajes. Desde la barra de estado puede cancelarse el filtro o acceder de nuevo al formulario de edición.

Para desactivar el filtro activo seleccionar el botón **Borrar** filtro del formulario o en la barra de estado del listado.

4.12 OBJETOS

4.12.1 INTRODUCCIÓN

El panel de *Objetos* solo está disponible si se ha seleccionado el ámbito <DOMINIO> ya que es una opción genérica que afecta por igual a todas las aplicaciones y dominios.

Se trata de un gestor que permite visualizar los objetos: Adaptadores, Brokers, Managers, que se han conectado en algún momento al dominio y también permite la posibilidad de añadir objetos de forma manual.

4.12.2 DIÁLOGO OBJETOS.

Para acceder al diálogo de Objetos, teniendo seleccionado el ámbito <DOMINO>, desde:

- Menú contextual: Pestaña **Objetos**.

Se accede así a la lista de objetos registrados en el dominio.

Columnas

En este listado cada línea refleja uno objeto y cada columna refleja la siguiente información:

- **Nombre**. Indica el nombre del Objeto.
- **Tipo**. Indica el tipo de Objeto.
- **Validar**. Si esta marcada la opción de validar un adaptador, y está activa la alarma correspondiente, en caso de desconexión del adaptador, se enviará la alarma. Para más información ver [Alarmas](#).

Tipos de Objetos

- **BROKER**. Objeto encargado de controlar y gestionar el acceso al BUS de integración.
- **EBIADAPTER**. Entrada de adaptador genérico EBI.

- **EBIADAPTERCMDJAVA.** Adaptador en modo comando (CMD) multiplataforma, versión JAVA.
- **EBIADAPTEREDICOMJAVA.** Adaptador de propósito general de EDICOM para multiplataforma, versión JAVA.
- **EBIADAPTEREDIWIN.** Adaptador específico para conectar la aplicación EDIWIN al BUS. (Esta versión de adaptador se ve sustituida por la versión JAVA para las nuevas versiones de EDIWIN)
- **EBIADAPTEREDIWINJAVA.** Adaptador específico para conectar la aplicación EDIWIN al BUS. Se trata de la última versión programada en Java.
- **EBIADAPTERIECTEEBI.** Adaptador específico del complemento IECTE_EBI para la estación EDIWIN.
- **EBIADAPTERSAPJAVA.** Adaptador especializado en conexión a sistemas SAP, Programado en Java para proyectos multiplataforma.

4.13 USUARIOS

4.13.1 INTRODUCCIÓN

La gestión de usuarios es una opción avanzada que no debe ser utilizada sin un conocimiento detallado de lo que se está haciendo. Esta sección del manual es únicamente informativa.

Se debe tener en cuenta que los usuarios no corresponden necesariamente con una persona física, todas las aplicaciones y dominios llevan asociados un usuario y su correspondiente password para realizar conexiones y ejecutar procesos BATCH, por lo que, en este caso, realmente se configurarían los permisos de aplicaciones y dominios de cara a acceder y monitorizar otras aplicaciones y dominios del servidor.

Los permisos de usuario y la gestión de usuario solo están disponibles para los usuarios Administradores. Estos tienen acceso completo a todos los permisos y paneles de gestión, incluido el diálogo usuarios.

4.13.2 DIÁLOGO USUARIOS




El diálogo de Usuarios solo está disponible bajo el ámbito de <DOMINIO>. Para acceder al diálogo de usuarios, desde:

- Menú principal: Pestaña  **Usuarios**.

Se muestra así una lista con todos los usuarios dados de alta en el servidor.

Columnas

En esta lista cada fila corresponde a un usuario y sus propiedades se reflejan por las siguientes columnas:

- **Tipo de usuario.** Icono que refleja el estado ( *habilitado* o  *inhabilitado*) y si se trata de un usuario  *Administrador*.
- **Usuario.** Código de usuario. Código que junto con el password permite la conexión de este usuario a las aplicaciones.
- **Nombre.** Nombre descriptivo del usuario.
- **Descripción.** Descripción adicional para el usuario o cargo

Opciones

El diálogo *Usuarios* ofrece las siguientes opciones repartidas entre la barra de herramientas y el menú contextual. Algunas de estas opciones afectan exclusivamente al elemento seleccionado de la lista.

- **Filtrar.** Permite filtrar los usuarios por Nombre, Código o Descripción.
- **Insertar.** Permite insertar un nuevo usuario, configurando a continuación sus propiedades y permisos a través del diálogo *Usuario*
- **Borrar.** Permite borrar el usuario seleccionado, esta acción es irreversible. Los usuarios también pueden inhabilitarse temporalmente sin necesidad de borrado, ver [Diálogo Usuario, Pestaña Datos de Usuario](#).
- **Refrescar.** Permite refrescar la información de este listado.

4.13.3 DIÁLOGO USUARIO.

El alta de un nuevo usuario y la configuración de sus propiedades se realiza desde el diálogo "Usuario" al cual se accede desde:


- Menú contextual: opción **Insertar**.
- Barra de herramientas: opción **Insertar**.
- O bien, si el usuario ya existe previamente, haciendo doble clic sobre él en la lista de usuarios.

Aparece así el diálogo usuario donde definir las propiedades del nuevo usuario o editar las de un usuario ya existente. Las propiedades de un usuario se dividen en cinco pestañas independientes dentro de este diálogo.

Pestaña Datos de usuario.

La pestaña *Datos de usuario* permite configurar las propiedades del usuario.

- **Usuario.** Código único del usuario. No editable una vez generado.
- **Contraseña.** Permite indicar la contraseña del usuario. El campo continuo **Reescriba la contraseña** permite repetir la misma contraseña para confirmar que no hay errores.

- **Nombre.** Permite indicar el nombre asociado al usuario, que puede ser igual o distinto al Código de Usuario.
- **Descripción.** Permite indicar una descripción complementaria del usuario o el cargo que desempeña.
- **Habilitado.** Permite indicar si el usuario está habilitado o inhabilitado. El estado inhabilitado se refleja en el listado de usuarios marcando en rojo la línea correspondiente y cambiando el icono a  Inhabilitado. Del mismo modo puede habilitarse un usuario inhabilitado volviendo a marcar esta casilla.
- **Administrador.** Indica si el usuario es un usuario con privilegios de administrador. Solo un usuario con privilegios de administrador tiene acceso a Insertar nuevos usuarios o configurar sus propiedades.

Pestaña Configuración

Dentro de la pestaña *configuración* se muestra una lista editable a la que pueden añadirse parámetros de configuración aplicables al usuario indicando la opción configurada y el valor correspondiente.



Pestaña Conexiones





Dentro de la pestaña *conexiones* se muestra una lista con todos los adaptadores disponibles en el servidor a los cuales tiene acceso este usuario. Permitiendo habilitar o inhabilitar dichos adaptadores para las planificaciones que realice el usuario.

Pestaña Planificaciones

Dentro de la pestaña *Planificaciones* se muestra una lista con todos los adaptadores disponibles para este usuario. Permitiendo habilitar o inhabilitar dichos adaptadores para las planificaciones que realice el usuario.

Pestaña Permisos




Dentro de la pestaña Permisos se muestra una detallada lista de las  *Aplicaciones* dentro del  *Dominio* (o dominios) a los que este usuario (el que se está configurando) puede tener acceso.

A su vez, dentro de cada uno de estos nodos de primer nivel aparecen los subnodos con los privilegios que, en cada aplicación, tiene acceso el usuario. Los privilegios son de  *Acceso*, de  *Ejecución*, de acceso a  *Scripts* y de acceso a  *Reglas*. La estructura dentro de estos nodos se repite, desplegando una serie de elementos a los que el usuario puede tener o no acceso.

User: EBIADAPTER (EBIADAPTER)

User data	Configuration	Connections	Scheduling	Permits
Name	Access	Create	Delete	
PRODUCCION				
CEM				
Access				
Domains	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Domain	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Applications	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Application	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Rules	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Publications	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Subscriptions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Schemes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Repository	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Scripts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Alarms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Active processes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Statistics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Messages	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Execution				
Scripts				
Scheduling				
EBIACMD				
EBITEST				

OK Cancel

- Permisos de  **Acceso**. Indican a qué diálogos de gestión tendrá acceso el usuario y si puede solo visualizarlos o también crear y eliminar elementos a través de ellos.
- Permisos de  **Ejecución**. Indican a qué funciones tiene acceso el usuario. Las funciones corresponden a las funciones básicas de gestión de Mensajes en EBI. *Suscripción, Publicación, Transformación, Borrado, Suscripción con Transformación y Ejecución de scripts*.
- Permisos sobre  **Scripts**. Indican qué scripts estarán disponibles para los procesos configurados por el usuario. El check "Acceso" permite mostrar u ocultar al usuario los distintos Scripts seleccionados: el usuario no podrá verlos ni usarlos para sus procesos.
- Permisos sobre **Planificaciones**. Indica qué planificaciones estarán disponibles para los procesos configurados por el usuario. La lista corresponde a las reglas configuradas en la estación. El check "Acceso" permite mostrar u ocultar las reglas y procesos asociados: el usuario no podrá verlos ni usarlos para sus procesos. El usuario tampoco podrá monitorizar los resultados de los Procesos asociados a estas reglas en el gestor de trabajos y procesos activos.

Permisos de acceso.

Cada una de estas opciones controlan un panel de gestión distinto de EBI Manager. Cada panel tiene asociado tres checks que reflejan distintos niveles de acceso y control.

- **Acceso:** Marcada, esta casilla permite al usuario acceder a la información correspondiente de esta Aplicación.
- **Crear:** Marcada, esta casilla permite al usuario crear elementos nuevos a través del gestor asociado, dentro de esta Aplicación. (Si el usuario no tiene Acceso, Este check es irrelevante.)
- **Borrar:** Marcada, esta casilla permite al usuario borrar elementos a través del gestor asociado, dentro de esta Aplicación. (Si el usuario no tiene Acceso, Este check es irrelevante.)

A continuación los paneles y elementos a los que el usuario puede tener Acceso.

- **Aplicaciones.** Permite el acceso a las aplicaciones.
- **Aplicación.** Permite el acceso a la aplicación de la cual cuelga el árbol, en el caso de la imagen CEM.
- **Dominios.** Da acceso a los dominios del EBI.
- **Dominio.** Da acceso al dominio del EBI del cual cuelga el árbol, en caso de la imagen se trataría de producción.
- **Reglas.** Permite el acceso al panel de Reglas.
- **Publicaciones.** Permite el acceso al panel Publicaciones.
- **Suscripciones.** Permite el acceso al panel Suscripciones.
- **Esquemas.** Permite el acceso al panel Esquemas.
- **Repositorio.** Permite el acceso al Repositorio.
- **Scripts.** Permite el acceso al panel de Scripts.
- **Alarmas.** Permite el acceso al panel de Alarmas.
- **Procesos activos.** Permite el acceso al panel de Procesos activos.
- **Trabajos.** Permite el acceso al panel de Trabajos.
- **Estadísticas.** Permite el acceso al panel de Estadísticas.
- **Mensajes.** Permite el acceso al panel de Mensajes.

4.14 ALARMAS

4.14.1 INTRODUCCIÓN

Cuando un proceso es ejecutado, correcta o incorrectamente, o cuando se detecta algún tráfico de mensajes en el BUS de integración, EBI puede emitir notificaciones

a una cuenta de correo para ser monitorizado de manera desasistida. (No es necesario que el administrador tenga una sesión iniciada)

El Gestor de Alarmas permite al usuario definir eventos que desencadenarán un aviso email, ya sea por eventos de los procesos de EBI o por errores detectados, por ejemplo al validar publicaciones. Estas alarmas se pueden configurar, activar y desactivar a voluntad.

4.14.2 DIÁLOGO ALARMAS



Para acceder al gestor de alarmas, se debe seleccionar una aplicación, el ámbito <DOMINIO> no tiene disponible este gestor.

- El menú principal: pestaña **Alarmas**.

Se visualiza así el listado de alarmas, donde cada línea corresponde con una alarma configurada




Columnas

En esta lista cada fila corresponde a una alarma y sus propiedades se reflejan por las siguientes columnas:

- **Estado** ( **Activa** o  **Inactiva**). Indica con un icono si la Alarma está habilitada o inhabilitada.
- **Nombre**. Indica el nombre dado a la alarma.
- **Descripción**. Indica la descripción de la alarma.
- **Fecha última modificación**. Indica la fecha de la última modificación en las propiedades de la alarma.
- **Habilitada**. Permite habilitar o inhabilitar la alarma con un check seleccionable.

Opciones



El diálogo *Alarmas* ofrece las siguientes opciones repartidas entre la barra de herramientas y el menú contextual. Algunas de estas opciones afectan exclusivamente al elemento seleccionado de la lista.

	Insertar . Permite añadir una nueva alarma, accediendo al diálogo Configuración
	Duplicar . Permite duplicar la alarma seleccionada.
	Borrar . Permite eliminar una alarma seleccionada.

4.14.3 CONFIGURACIÓN DE ALARMAS

El diálogo *Configuración*, aparece cuando se inserta una nueva alarma o se selecciona una alarma del listado y se accede a sus propiedades (doble clic). Este formulario contiene varios grupos de campos que permiten configurar individualmente cada alarma y los eventos que las desencadenan.

Para añadir una nueva alarma al listado, en la pestaña *Alarmas*, desde:

- La barra de herramientas principal: botón  **Insertar**.
- El menú contextual de alarmas: opción  **Insertar**.

Se muestra así el diálogo de configuración de alarmas donde es necesario cumplimentar la descripción de la alarma, el servidor de correo electrónico, las condiciones de disparo y por último el evento o eventos que inician la alarma.

Datos de la alarma

- **Nombre.** Indica el nombre de la alarma en el listado. Este dato es Obligatorio.
- **Habilitado.** Indica si la alarma se dará de alta habilitada o no inicialmente. Luego puede cambiarse.
- **Descripción.** Indica una descripción complementaria que también se visualizará en el listado.
- **Adaptador.** Indica el adaptador que gestionará el envío de la alarma.

Configuración SMTP.

El grupo *Configuración SMTP* permite la configuración del servidor de correo utilizado por la aplicación para enviar las alarmas así como el destino de dichas notificaciones.

- **Servidor SMTP.** Indica el servidor SMTP para la publicación de los e-mail. Este dato es obligatorio.
- **Puerto.** Indica el puerto de salida para la publicación de los e-mail. Este dato es obligatorio.
- **De.** Indica el remitente del e-mail.
- **Para.** Indica el destinatario del e-mail. Este dato es obligatorio.
- **Asunto.** Indica un asunto fijo para los e-mail de notificación de esta alarma.
- **Usuario y Contraseña.** Permiten indicar el usuario y contraseña en caso de ser requerido por el servidor.

Condiciones de disparo.

El grupo condición de disparo es el nivel de configuración más detallado, permitiendo elegir qué evento o eventos desencadenan la notificación y con qué periodicidad deben notificarse (Una vez al día, Una vez a la semana, Durante un determinado periodo de tiempo) para evitar sobrecargas de notificaciones en la cuenta de correo indicada.

- **Repetidamente.** Indica con que frecuencia debe notificarse la alarma a lo largo de la semana.
 - **Diario.** Cada día.
 - **Semanal.** Semanalmente los días seleccionados.
- **Periodo.** Indica el plazo habilitado para la notificación de alarmas. Las alarmas se notificarán solo si ocurren a partir de la hora inicial (Desde) o hasta la hora final (Hasta) y el intervalo entre ambas. (Por defecto desde las 00:00 hasta las 23:59 si no se indica ningún valor)

Eventos

El resto de esta sección es una lista codificada de posibles eventos a notificar. Una alarma se notificará siempre que se cumpla cualquiera de los eventos indicados.

0000	Información. Si se selecciona este evento, la alarma se disparará siempre que entre un documento duplicado, un cambio de situación y cualquier otro evento que sea de carácter informativo (Prefijo "I")
I101	Usuario no autorizado.
I200	Publicación.
I201	Publicación no autorizada.
I202	Error al validar la publicación.
I203	Destino de la publicación desconocido.
I204	Publicación duplicada.
I210	Suscripción.
I211	Suscripción no autorizada.
I212	Suscripción recibida.
I213	Suscripción bloqueada.
I214	Recogida de suscripciones salientes (minutos).
I214	Suscripción procesando formato EBI.
I220	Transformación.
I221	Transformación no autorizada.
I222	Error durante el proceso.
I230	Ejecución de script.
I231	Error en la ejecución de script.

I232	Excedido el tiempo máximo de ejecución del proceso (Minutos).
I233	Fin de ejecución de proceso inesperado.
I234	Notificación de cambios.
I235	Tiempo de inactividad de la aplicación , enviando mensajes (minutos).
I236	Tiempo de inactividad de la aplicación, recibiendo mensajes (minutos).
E000	Errores críticos en la ejecución del EBI.

Tras la configuración, al **aceptar**, aparecerá una nueva alarma en el listado.