



# Edicom Business Integrator Mapping tool

## Guía de Usuario - Diseño de Informes

Título del documento:	EBIMAP
Nombre del fichero:	ES EBIMAP Manual usuario.odt
Versión:	1.0
Estado:	VIGENTE
Fecha:	21/11/2008
Autor:	Francisco Belda

**Revisión, Aprobación**

Revisado por:	Francisco Belda	Fecha: 21/11/2008
Aprobado por:	Francisco Belda	Fecha: 21/11/2008

**Historial de cambios**

Versión	Fecha	Descripción de la acción	Páginas
1.0	21/11/2008	Versión inicial.	Todas
1.0.1	07/11/2011	Recomendación puntual sobre caracteres especiales en las descripciones de reglas de validación.	71
1.0.2	02/12/2011	Se añaden limitaciones listas de equivalencia	61

# Índice de contenido

<b>1 Presentación Ebimap.....</b>	<b>5</b>
1.1 Introducción al Ebimap.....	5
<b>2 Conceptos y Términos.....</b>	<b>7</b>
2.1 Conceptos.....	7
2.2 Términos.....	8
<b>3 El panel de control.....</b>	<b>10</b>
3.1 Introducción.....	10
3.2 Menú Principal.....	11
3.3 Barra de herramientas.....	13
3.4 Barra de acceso a ventanas.....	13
3.5 Área de trabajo.....	13
3.6 Ventana Principal.....	14
<b>4 Diseño de interfaces.....</b>	<b>17</b>
4.1 Introducción.....	17
4.2 Propiedades de la Interfaz.....	18
4.3 Opciones.....	21
4.4 Gestión.....	23
4.5 Ejemplo: Diseño de una Interfaz.....	27
<b>5 Diseño de Mapas (Básico).....</b>	<b>32</b>
5.1 Introducción.....	32
5.2 Propiedades del Mapa.....	34
5.3 Opciones.....	36
5.4 Gestión.....	37
5.5 Pestaña “Información”.....	38
5.6 Diseño de un Mapa de Ejemplo (Sencillo).....	39
<b>6 Diseño de mapas (Intermedio).....</b>	<b>44</b>
6.1 Introducción.....	44
6.2 Creación y asignación de valores.....	44
6.3 Expresiones (creación de scripts).....	47
6.4 Pestaña “Propiedades mapeado”.....	52
6.5 Diseño de un Mapa de Ejemplo. (Intermedio).....	54
<b>7 Diseño de Listas de valores.....</b>	<b>60</b>
7.1 Introducción.....	60
7.2 Gestión.....	60
7.3 Ejemplos: Tipos de listas de valores.....	63
<b>8 Diseño de mapas (Avanzado).....</b>	<b>66</b>
8.1 Introducción.....	66
8.2 Colisión durante la asignación de valores.....	66
8.3 Conversión de valores.....	67
8.4 Pestaña “Propiedades interfaz”.....	67
8.5 Ejemplo: Diseño de mapa (Avanzado).....	69
<b>9 Reglas de validación.....</b>	<b>71</b>
9.1 Introducción.....	71
9.2 Opciones.....	73

9.3Ejemplos: Reglas de validación.....	73
<b>10 Propiedades del Mapa. (Avanzado).....</b>	<b>77</b>
10.1Introducción.....	77
<b>11 Mapas con acceso a base de datos.....</b>	<b>86</b>
11.1Introducción.....	86
11.2Ejemplo: interfaz de acceso a base de datos.....	86
11.3Ejemplo de mapa con acceso a base de datos.....	88
<b>12 Diseño de Mapas Multiples.....</b>	<b>91</b>
12.1Introducción.....	91
12.2Ejemplo: Diseño de Mapa Múltiple.....	91
<b>13 Algoritmo de funcionamiento del Ebimap.....</b>	<b>95</b>
13.1Introducción.....	95

# 1 PRESENTACIÓN EBIMAP

## 1.1 INTRODUCCIÓN AL EBIMAP.

---

### 1.1.1 EBIMAP, HERRAMIENTA DE MAPEADO.

El **ebiMap** es la herramienta de mapeado para Windows de **Edicom**. EbiMap permite mapear datos entre dos modelos de datos totalmente distintos de forma sencilla. Para el caso de la creación de mensajes XML/EDI (EDIFACT,XML,X12) a partir de modelos de datos del sistema informático interno y viceversa.

Sus características especiales permiten introducir al usuario fácilmente al mundo del mapeado de datos para la integración entre sistemas separados de tratamiento de datos:

- **Gráfico:** ebiMap aprovecha al máximo el poder gráfico de Windows, proporcionando acceso visual a los modelos de datos, a los mapeados y los mismos datos, así como métodos sencillos y directos de presentación y manipulación de estos datos.
- **Universal:** es posible utilizar ebiMap como herramienta de mapeado, entre los distintos modelos de datos del sistema informático interno, llegando incluso a la automatización total con la ayuda del modo “Batch” de ejecución, gracias a su comportamiento como mapeador universal de datos.
- **Autónomo:** ebiMap puede trabajar de forma autónoma realizando los mapeados definidos como una aplicación independiente integrada en la operativa del Sistema Informático Interno.
- **Modos avanzados de acceso y generación de datos:** Utilizando tablas de equivalencias, filtrado de Datos, lookups a base de datos vía ADO, generación de datos en cascada, Formulas mediante funciones “Ediwin Script”...
- **Máxima productividad:** por su velocidad y versatilidad, ebiMap también se distribuye de forma totalmente integrada con la estación de usuario **Ediwin**, de Edicom lo cual lo convierte en el mapeador idóneo para realizar EDI.

### 1.1.2 MEJORAS EN EL MAPEADOR.

En esta nueva versión de la herramienta de mapeado ebiMap, el área técnica de Edicom. Ha trabajado para ofrecer una mejor comprensión y generación de los mapeados, ofreciendo las siguientes características:

- Aisla el sistema de gestión interno de los conceptos XML/EDI.
- Esquemas predefinidos para mensajes XML/EDI.

- Esquemas predefinidos para algunas aplicaciones de gestión como “SAP/R3”, “NAVISION Financials” o “Diges AC de DIMONI”, entre otras.
- Perfectamente integrado con el software XML/EDI EDIWIN de Edicom.
- Aisla de posibles cambios en las interfaces del sistema informático interno con el software XML/EDI como consecuencia de cambios en los mensajes estándar.
- Reduce el tiempo de integración por mensaje.
- Permite realizar una validación de contenido de los documentos que envía o recibe. Por ejemplo, puede filtrar aquellas facturas sin número de pedido.
- Incorpora algunas funcionalidades especializadas en XML/EDI: hace transparente la migración de versiones de mensajes y le permite personalizar los datos a enviar a cada interlocutor con el objeto de ahorrar en volumen de información transmitida a la red de comunicaciones mediante la utilización de Filtros.
- Permite definir formatos de archivos propios. Puede usar uno o varios archivos, formato fijo, separadores, XML, ODBC, ADO, etc.

### **1.1.3¿DONDE COMENZAR?**

Este manual permite aprender a manejar ebiMap de manera básica. Al terminar su lectura, se habrá alcanzado un nivel de conocimientos sobre los mapeados con ebiMap suficiente como para crear desde cero interfaces y mapeados, incrementando las posibilidades con el uso y la experiencia.

La estructura de este manual pretende proporcionar al lector la información necesaria para el aprendizaje progresivo y controlado de ebiMap. El manual comienza presentando una visión global de la herramienta y de sus distintas pantallas y opciones para, a continuación, adentrarse en cada uno de los pasos en la creación de un mapeado y en sus respectivas posibilidades.

## 2 CONCEPTOS Y TÉRMINOS.

### 2.1 CONCEPTOS

Este capítulo introduce los conceptos y términos básicos que serán útiles al trabajar con ebiMap.

#### 2.1.1 ¿QUE ES UN MAPEADOR?

Es una herramienta que permite obtener datos desde un “Modelo de Datos origen” a un “Modelo de Datos destino”, permitiendo la manipulación previa de estos datos para adecuarlos a los requerimientos del modelo de datos destino

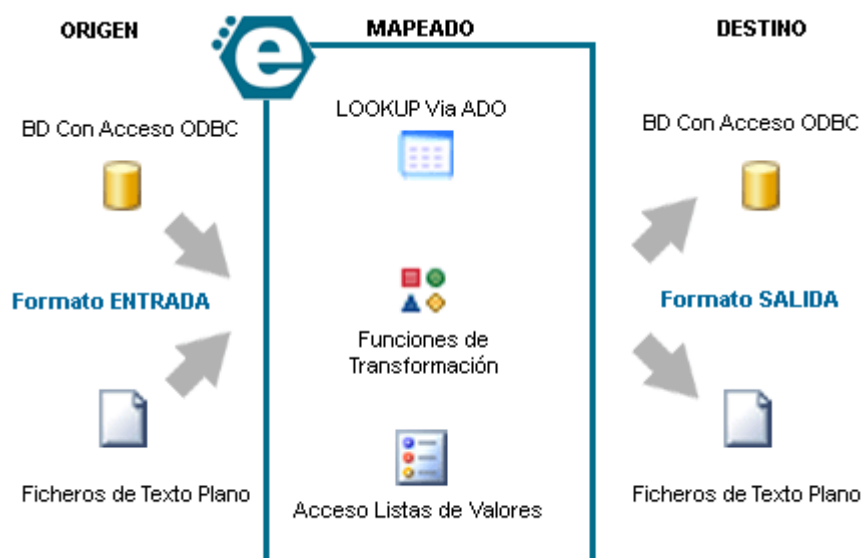
#### 2.1.2 ¿PARA QUE SIRVE UN MAPEADOR?

Un Mapeador funciona como un proceso de traducción de un modelo de datos a otro, y tiene como finalidad proporcionar información en formatos que el sistema de gestión interno del usuario necesite de manera especial. Esto es una gran ventaja porque proporciona flexibilidad de integración con cualquier sistema informático.

#### 2.1.3 ¿CÓMO FUNCIONA?

El ebiMap es un mapeador de datos que permite traspasar los datos almacenados en un determinado modelo de datos “A” a un modelo de datos “B”

El ebiMap tiene la capacidad de acceder a ficheros de texto o bases de datos haciendo uso de drivers ODBC para recuperar los datos de la aplicación origen y almacenarlos en la aplicación destino.



Durante el proceso de mapeado, ebiMap puede realizar conversiones de códigos utilizando listas de equivalencias que pueden ser predefinidas dentro de ebiMap, lookups en tablas auxiliares y calculando datos derivados con la ayuda de las distintas funciones con las que cuenta **ebiMap Script**.

## **2.2 TÉRMINOS**

---

### **2.2.1 INTERFAZ. (MODELO DE DATOS)**

“Interfaz” es el nombre dado a un “Modelo de datos” definido en ebiMap.

La definición de una interfaz en ebiMap consta de una serie de características como la definición jerárquica de los distintos registros que forman el modelo de datos, sus interdependencias, los campos que componen cada uno de estos registros, y la ubicación física de todos los elementos.

Cada registro tiene asociado un “Origen de datos” que puede ser de diferentes tipos: texto delimitado, formato fijo, EDIFACT, XML, ODBC, ADO etc. Incluso es posible crear nuevos tipos de archivos mediante las especificaciones de la Interfaz de programación de Acceso a Datos de Ediwin (IEAD). Las interfaces se diseñan y editan en la ventana de “diseño de interfaces”.

### **2.2.2 MAPA.**

En ebiMap se denomina “Mapa” a la definición del modo en que se realizará el tratamiento de los datos de una interface origen para ser transformados en datos de una interface destino. Los nuevos mapas se diseñan en la ventana de “diseño de Mapas”.

### **2.2.3 MAPEADO MÚLTIPLE (MULTIMAPEADO)**

Se denomina Multimapado el proceso para la generación de varios ficheros de datos distintos de “destino” a partir de la misma interfaz origen. En función de una o varias condiciones, asociadas a diferentes mapas, Los nuevos Multimapados se diseñan en la ventana de “diseño de Mapas Múltiples”.

### **2.2.4 GRUPO DE TRABAJO.**

Dentro de ebiMap se pueden establecer grupos de trabajo en base a las interfaces de trabajo que se utilizan, como por ejemplo, un grupo para interfaces en “EDIFACT”, otro para “X-12” y otro para “XML”.

### **2.2.5 REGISTRO.**



Un “Registro” en ebiMap es cada una de las unidades lógicas que componen un modelo de datos (o “interfaz”). Coincide con el concepto de “Registro” en una base de datos. Un registro consiste en un número de campos secuenciales de acuerdo con las propiedades definidas para cada uno de ellos.

### **2.2.6 CAMPO.**

Un “Campo” en ebiMap es la unidad mínima de información donde se guarda el valor de un solo dato para el cual se definen las propiedades de manera individual.

### **2.2.7 ELEMENTO**

Un “Elemento” en ebiMap, es el componente de cada campo. Los elementos de un campo son entre otras cosas: su Nombre, Tipo (Alfanumérico, Numérico,), Longitud, Formato, caracteres separadores, etcétera.

### **2.2.8 IEAD´S. (MODOS DE ACCESO A DATOS).**

El IEAD (Interfaz Ediwin Acceso a Datos) es el modelo de sintaxis (del código) que tendrá la interface, este modelo de sintaxis es la base de ebiMap para reconocer o implementar estas interfaces.

Estos modelos de sintaxis son:

- IEAD EDIFACT.
- IEAD LOT.
- IEAD TXT.
- IEAD XML.
- IEAD ODBC.

### **2.2.9 LISTA DE VALORES.**

Una “Lista” en ebiMap es, a grandes rasgos, una tabla que está formada por tres columnas de valores (valor1, valor2 y descripción) que pueden utilizarse para realizar equivalencias o conversiones de un valor en relación con el otro valor asociado.

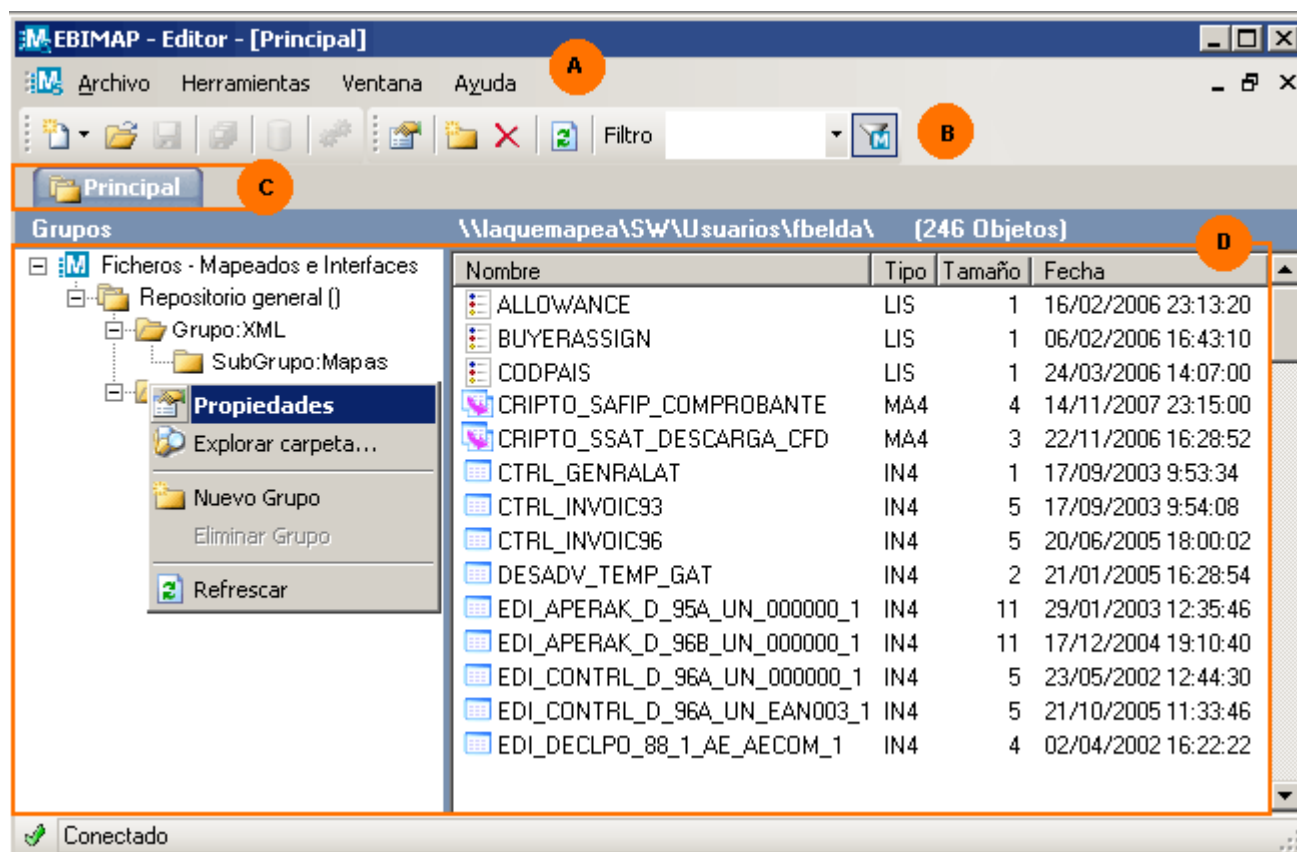
Si un archivo de datos de origen tiene el valor 1 es posible añadir al archivo de destino, el valor 2 correspondiente o sustituir el valor 1 por el valor 2 realizando una conversión. Las Listas de valores se diseñan en la ventana de “*Diseño de listas de valores*”

## 3 EL PANEL DE CONTROL.

### 3.1 INTRODUCCIÓN

Al iniciar ebiMap se abrirá la pantalla principal de ebiMap. Esta pantalla de inicio se conoce como “Panel de control” y por defecto aparecerá la ventana “Principal”.

En el panel de control esta dividido en:



- Una barra de menú principal con menús desplegables, **Archivo, Herramientas, Ventana, Ayuda..(A)**.
- Una barra de herramientas, con botones de acceso rápido a diversas opciones. Estas opciones se ampliarán según la ventana en la que se esté trabajando.(B)
- Una barra de acceso a ventanas, que permite navegar entre las ventanas abiertas gracias a las solapas (o pestañas).(C)
- Un área de trabajo compuesta a su vez por varias secciones que cambiarán según el archivo que estemos editando. En el área de trabajo aparecen las ventanas, la ventana principal, la ventana de diseño de mapas, la ventana de diseño de interfaces, etc. Reflejándose si es necesario en la barra de acceso a ventanas como una nueva solapa. (D)

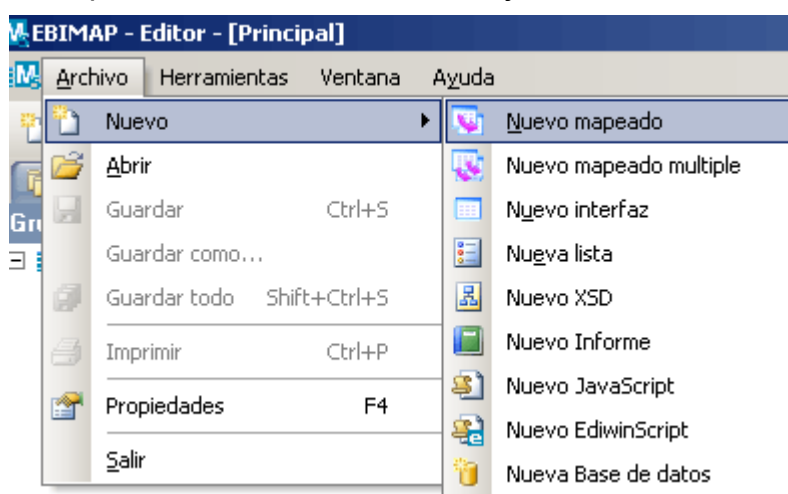
## 3.2 MENÚ PRINCIPAL

La barra de menú principal esta formada por distintos grupos o menus desplegables, según la ventana seleccionada (dependiendo del archivo que se este visualizando en cada momento) este menú principal mostrará menus desplegables adicionales.

### 3.2.1 MENÚ ARCHIVO.

En el menú “**Archivo**” existen diferentes opciones:

- **Nuevo:** la opción “**Nuevo**” permite crear un nuevo objeto. La opción “**Nuevo**” a su vez muestra otro sub-menú donde elegir el tipo de archivo que se desea crear por ejemplo una “interfaz”, un “mapeado”, un “mapeado múltiple” o una “lista de equivalencias”. Al crear un nuevo archivo se abrirá una la ventana de diseño correspondiente en el área de trabajo.



- **Abrir:** la opción “**Abrir**” permite abrir un archivo existente, que puede ser una interfaz, un mapa, un multimapeado o una lista de equivalencias. La opción “**Abrir**” permite importar interfaces, listas de equivalencias y mapas de versiones anteriores además de definiciones sintácticas de los diferentes mensajes de cualquier estándar.
- **Guardar:** permite guardar el archivo activo en el área de trabajo.
- **Guardar como:** permite guardar el archivo activo en el área de trabajo como un archivo texto.
- **Guardar todo:** permite guardar todos los archivos abiertos en el área de trabajo.
- **Imprimir:** permite crear un resumen informativo de las propiedades del objeto activo en el área de trabajo. (Mapeado, interfaz , etc.)
- **Propiedades:** según la ventana activa en ese momento muestra las propiedades del mapa, interfaz o lista que se este diseñando. En el caso de la ventana principal muestra las propiedades del grupo de trabajo

- **Salir:** por último la opción que se utiliza para cerrar ebiMap.

### 3.2.2 MENÚ HERRAMIENTAS

- **Opciones:** permite acceder al panel de configuración general del ebiMap, entre otras cosas permite seleccionar el idioma, la configuración de servidor, y la configuración de la base de datos principal de la aplicación, se recomienda no modificar esta configuración.

### 3.2.3 MENÚ EDICIÓN.

El menú desplegable “**Edición**” muestra las opciones relativas a la edición de archivos. Sólo estará accesible si en la ventana seleccionada se visualiza un archivo.

- **Deshacer** (Ctrl-Z): deshace el último cambio.
- **Rehacer** (Ctrl-Y): rehace el último cambio.
- **Cortar:** corta el texto seleccionado.
- **Pegar:** pega el texto seleccionado.
- **Copiar:** copia el texto seleccionado.
- **Eliminar:** permite eliminar el texto seleccionado.
- **Buscar** (Ctrl-F): permite realizar una búsqueda, el campo para insertar el parametro de búsqueda aparecerá al final de la ventana.

### 3.2.4 MENÚ EJECUCIÓN.

En el menú desplegable “**Ejecución**” se accede a las opciones necesarias para compilar y ejecutar un mapeado o una interfaz. Sólo estará accesible si en la ventana seleccionada se visualiza un archivo.

- **Ejecutar** (F9): ejecutará el mapeado y creará los archivos destino resultantes
- **Compilar**(Ctrl F9): compilará los archivos editados con los nuevos cambios. Paso previo para ser ejecutado.

### 3.2.5 MENÚ VENTANA.

El menú desplegable “**Ventana**” permite configurar la visualización del entorno de trabajo de ebiMap y navegar entre las distintas ventanas abiertas.

- **Cascada:** muestra las diferentes ventanas abiertas de una forma consecutiva una tras otra.

- **Mosaico Vertical:** divide la pantalla en forma vertical en tantas partes como número de ventanas abiertas.
- **Mosaico Horizontal:** divide la pantalla en forma horizontal en tantas partes como número de ventanas abiertas.
- **Minimizar Todo:** minimiza todas las ventanas abiertas en ese instante.
- **Maximizar Todo:** maximiza todas las ventanas abiertas en ese momento.

### 3.2.6 MENÚ AYUDA.


La opción “**Ayuda**” proporciona información sobre el software, muestra la versión del ebiMap y acceso a el documento de ayuda (También accesible con la tecla F1)

## 3.3 BARRA DE HERRAMIENTAS

La barra de herramientas, bajo el menú principal, muestra un acceso rápido a las opciones más utilizadas y a las opciones propias de cada ventana de diseño.

Cada ventana de diseño y la ventana principal tienen una barra de herramientas distinta. Aunque comparten opciones entre ellas, otras opciones son únicas para la ventana en cuestión. Las opciones se comportan según la ventana abierta. Por ejemplo “**propiedades**” en la ventana “*principal*” mostrará las propiedades del grupo de trabajo seleccionado, y en la ventana de “*Diseño de mapas*” mostrará las propiedades del mapa.

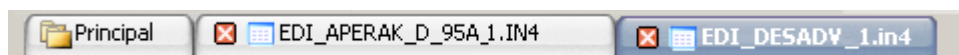
Se puede elegir las barras de herramientas visibles en cada momento con el menú contextual de la barra de herramientas, que aparece con el botón derecho del ratón

*Nota- En la barra de herramientas de la ventana principal la opción  permite mostrar solamente los archivos ebiMap.*

## 3.4 BARRA DE ACCESO A VENTANAS.

La barra de acceso a ventanas permite de una manera rápida el desplazamiento entre las distintas ventanas de trabajo que estén abiertas en el panel de control. Está localizada en la parte superior del área de trabajo, y las ventanas se reflejan en forma de solapas con el nombre del archivo editado.

En la barra de acceso a ventanas se señalará que ventana está activa. Para cerrar una ventana abierta se puede apretar el botón cerrar de la solapa.



*Nota- La ventana “Principal” siempre estará disponible y no se podrá cerrar.*

## 3.5 ÁREA DE TRABAJO.

El área de trabajo puede dividirse en varias ventanas de diseño para trabajar sobre distintos archivos al mismo tiempo. Cada una de estas ventanas está dividida en campos y secciones que dependen del tipo de archivo que se esté editando (por ejemplo un “informe”, una “interfaz” o un “mapa”)

Según la ventana donde se esté trabajando, la configuración del menú principal y la barra de herramientas puede mostrar más o menos opciones.

Ademas existe una ventana “*Principal*” que aparece por defecto a la hora de iniciar el ebiMap.

## 3.6 VENTANA PRINCIPAL

---

### 3.6.1 INTRODUCCIÓN.

La ventana “*Principal*” del ebiMap , la primera que aparece en el panel de control y que no puede cerrarse, permite editar los grupos de trabajo configurados para la aplicación, Un grupo de trabajo es un conjunto de objetos (Interfaces, Listas, Mapas, Informes ) que están agrupados según algún criterio útil para el usuario. Por ejemplo grupos de trabajo según formato de datos (Edifact , X12, XML) o según proyectos.

La ventana esta dividida en dos secciones, una sección esta destinada a mostrar los “*Grupos de trabajo*” en forma de árbol de carpetas y la otra sección a mostrar los “*Objetos*” que contienen estos grupos de trabajo en una lista.

Cuando se seleccione un nodo correspondiente a un grupo de trabajo en la parte de la derecha se visualizarán los objetos contenidos, En esta lista se muestran para cada objeto el nombre, el tipo de objeto, el tamaño y la fecha de creación/modificación.

Las opciones propias de la ventana “*Principal*” están accesibles desde los menús contextuales que aparecen con el botón derecho del ratón y que son diferentes para el área de “*Grupos*” y para el área de “*Objetos*”, que tomará como nombre la dirección donde se almacenan los objetos.

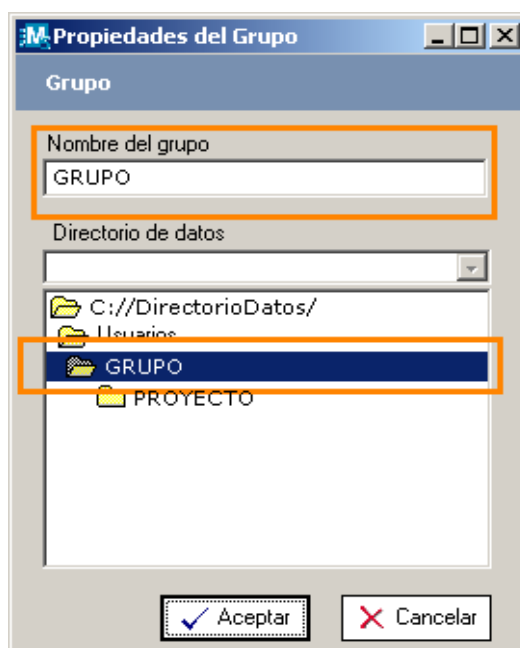
### 3.6.2 ÁREA “GRUPOS DE TRABAJO”

Los grupos de trabajo se pueden definir como una serie de archivos que se utilizan para un proceso de mapeado, estos archivos pueden ser interfaces, mapas o listas de equivalencias. Estos archivos se pueden agrupar para facilitar de gran manera el manejo de lotes de archivos que esta orientados a un determinado proceso de mapeado.

El menú contextual de grupos de trabajo aparece al hacer clic con el botón derecho sobre el área de “*Grupos*” y las opciones disponibles son:

- **Propiedades:** muestra las propiedades del grupos de trabajo, estas son el nombre y la ruta de ubicación donde se encuentran los archivos de ese grupo de trabajo.

- **Explorar Carpeta:** permite abrir el directorio en el navegador del Sistema Operativo.
- **Nuevo Grupo:** permite crear un nuevo grupo de trabajo y acto seguido configurar sus propiedades. Si se crea un nuevo grupo con un nodo seleccionado, este nuevo grupo será un nodo hijo del anterior.
- **Eliminar Grupo:** borra el grupo de trabajo seleccionado.
- **Refrescar:** refresca el listado de grupos de trabajo. Hay que refrescar al crear un nuevo grupo de trabajo.



### 3.6.3 ÁREA “OBJETOS”

En la parte superior del área de objetos, se visualizará la ruta completa del directorio donde están guardados estos objetos. Esta ruta es la que se configura en el panel de propiedades del “grupo de trabajo”

- **Abrir:** abre el archivo seleccionado en una nueva ventana de diseño.
- **Nuevo:** crea un nuevo archivo permitiendo escoger el tipo de archivo entre varias opciones.
- **Renombrar (F2):** permite cambiar el nombre al archivo seleccionado
- **Eliminar (Del):** elimina el objeto seleccionado.
- **Optimizar:** en el caso de un mapeado con interfaz de entrada XML, permite eliminar los elementos no utilizados optimizando el mapa.
- **Ver datos:** permite ver la estructura de los datos contenidos en una interfaz. Sólo funciona con archivos de tipo interfaz (\*.in4)

### 3.6.4 OPCIÓN “FILTRO”.

En la barra de herramientas de la ventana “*Principal*” existe un campo para realizar un filtrado en el listado de objetos, Es posible indicar un valor para que sólo aparezcan los objetos que correspondan con dicho filtro.

Por ejemplo, si se indica “IN4” (y se presiona la tecla “Intro”) sólo aparecerán los archivos interfaz en el listado.



## 4 DISEÑO DE INTERFACES.

### 4.1 INTRODUCCIÓN.

Al crear o editar una nueva interfaz, se muestra la ventana de edición de “Interfaces” dividida en varias secciones principales.

The screenshot displays the 'Interfaces' window with the following components:

- Nombre del interfaz:** A tree view on the left showing a hierarchy of records: UNH (selected), BGM, DTM, FTX, CNT, G01RFF, G02NAD, G03ERC, and UNT. An orange circle labeled **A** highlights the UNH record.
- Campos del registro: UNH:** A table on the right listing the fields of the selected record. An orange circle labeled **B** highlights the first row.
- Propiedades:** A tabbed interface at the bottom. The 'Registros' tab is active, showing the 'Registro UNH' table. An orange circle labeled **C** highlights the 'Búsqueda' tab.
- Registro UNH:** A table showing the properties of the selected record.
- Campo 0062:** A table showing the properties of the selected field (0062).

Nombre	Descripción	Tip	Lon
0062	Número de referencia del	X	14
S009	Identificador de mensaje		0
0065	Identificador del tipo de mens	X	6
0052	Número de versión del tipo ...	X	3
0054	Número de sub-versión del ...	X	3
0051	Agencia controladora	X	2
0057	Código asignado por la aso...	X	6
0068	Referencia de acceso com...	X	35
S010	Estado del traslado		0
0070	Número de secuencia de tr...	N	2
0073	Indicador de primer último ...	A	1

Propiedad	Valor
<b>General</b>	
Nombre	UNH
Descripción	UNH
Acceso a datos	EDIFACT3
Maestro-Detalle	
Filtro	
Etiqueta	UNH
Inicio Etiqueta	1
Estado	Mandatorio
Cardinalidad Mín.	1
Cardinalidad Máx.	1
Admite Valor	false
Es grupo	false
Grupo Virtual	Deshabilitado

Propiedad	Valor
<b>General</b>	
Nombre	0062
Descripción	Número de referencia del me
Tipo de datos	Alfanumérico
Tipo de campo	Dato
Long. Mínima	0
Long. Máxima	14
Decimales	0
Pattern	
Repeticiones	1
Estado	Mandatorio
Valor fijo	
Calificado por	
Lista de valores	

- El árbol de registros en la parte superior izquierda.(**A**)
- El listado de campos contenidos en el registro seleccionado.(**B**)
- El área de propiedades e información en la parte inferior, dividido a su vez en varias pestañas (**C**)

### ***Área árbol de registros de la interfaz.***

Dentro de esta área se muestran los registros contenidos en la interfaz, esta presentación se muestra como una árbol de nodos donde se establece la dependencia entre cada uno de ellos (Padre-Hijo).

### ***El área de Campos del registro***

Dentro de esta área, en la parte izquierda superior, se muestran los campos contenidos en el registro seleccionado. Cada campo tiene una serie de propiedades como el tipo de valor (alfanumérico) la longitud del campo, etc.

### ***El área del Panel de Propiedades***

En la parte inferior de la ventana de diseño de interfaces hay un panel con varias pestañas donde se reflejan distintas informaciones y propiedades de los elementos de la interfaz.

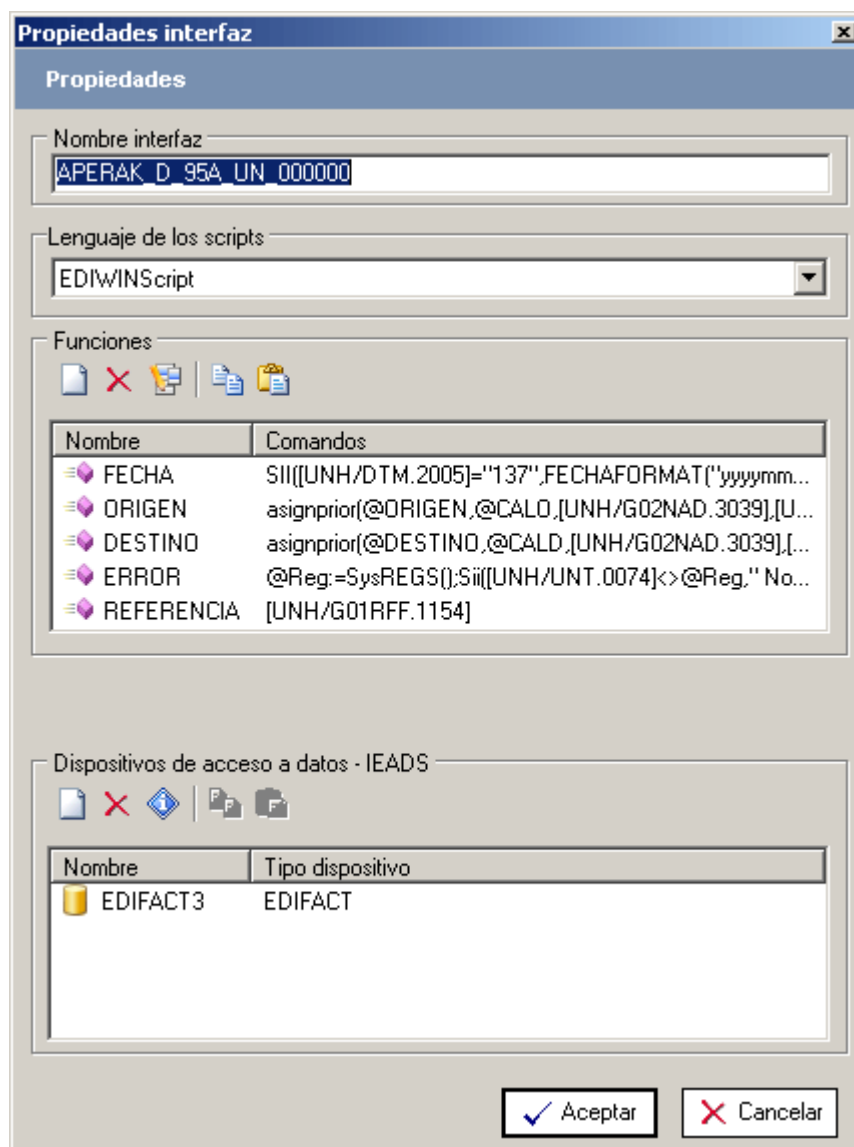
- **Pestaña "Propiedades:** muestra las propiedades del registro y el campo seleccionado en cada momento. A la izquierda las propiedades del registro y a la derecha las propiedades del campo.
- **Pestaña Datos:** esta pestaña permite visualizar los datos del archivo de origen que se intenta abrir con la interfaz, también permite “pegar” el archivo directamente.
- **Pestaña Información:** esta pestaña muestra la información resultante de los distintos eventos que se realizan trabajando con la interfaz (Ejecutar, Guardar, Errores...)
- **Pestaña Registros:** esta pestaña muestra los registros que se han recorrido.
- **Pestaña Búsqueda:** permite realizar una búsqueda entre los campos y registros de la interfaz.

## **4.2 PROPIEDADES DE LA INTERFAZ**

---

Cuando está activa la ventana de diseño de interfaces, la opción “Propiedades” permite acceder al diálogo de propiedades de la interfaz. Para acceder a este diálogo, desde:

- Menú Principal: opción **Propiedades**.
- Barra de Herramientas: opción **Propiedades**.



### ***Nombre de la interfaz.***

El nombre de la interfaz es la ruta completa donde se encuentra este archivo interfaz.

### ***Funciones.***

Una función permite diseñar una serie de instrucciones a ejecutar a la hora de crear una interfaz, por ejemplo un proceso, una comparación, etc.

Dentro de las funciones existe un área de botones que permiten crear una nueva función, eliminar una función, mostrar una función, copiar y pegar una función, etcétera.

- **Nueva función:** al elegir nueva función, ebiMap muestra una ventana donde permite introducir el nombre de la función, y en la parte inferior de esta ventana diseñar la función que se utilizará, esta sección cuenta con un botón “**Validación**” de la función (que indica si la sintaxis de la función es correcta) además de los botones “**Aceptar**” y “**Cancelar**”.
- **Eliminar función:** permite la eliminación de una función seleccionada.
- **Propiedades de función:** muestra la estructura de la función, además permite editar la función.
- **Copia función:** permite copiar una función.
- **Pega función:** permite pegar una función copiada previamente.

Estas dos últimas opciones son muy útiles, ya que permiten al usuario tomar una función ya hecha y sólo retocarla para que funcione de otra forma. Dentro de esta misma sección de función tenemos un área de “*inicialización de variables*” que permite inicializar las variables utilizadas en las funciones diseñadas.

Todas las funciones diseñadas se presentarán en la pantalla con su nombre y los comandos asociados a ellas.

### ***Dispositivos de acceso a datos - IEADs***

Esta es una parte muy importante dentro de las propiedades de la interfaz, ya que es aquí donde se definen los tipos de archivos para los que la interfaz servirá, el tipo de información que podrá contener.

Dentro de esta área, también se dispone de una pequeña barra de menú con las siguientes opciones.

- **Nuevo Origen de datos:** permite indicar un nuevo origen de datos.
- **Borrar un origen de datos:** permite borrar un origen de datos.
- **Propiedades de un origen de datos:** permite editar las propiedades de un origen de datos.
- **Copiar un origen de datos:** permite copiar un origen de datos.
- **Pegar un origen de datos:** permite pegar un origen de datos previamente copiado.

Dentro del botón de “**Nuevo Origen de datos**”, se puede establecer la estructura que contendrá los registros definidos en la interfaz.

Se puede establecer el nombre del origen de datos (que puede ser el mismo nombre del registro o no) y el tipo de dispositivo de datos en que se convertirá. Estos tipos de datos pueden ser:

- **EDIFACT:** indica archivo de origen de datos de tipo EDIFACT.

- **TEXTO:** en ese caso el origen de datos es un archivo plano sin ningún formato establecido.
- **LOT:** el formato “LOT” es un formato propio de ebiMap, este es un formato parecido al “EDIFACT”, sólo que es un poco más legible para el usuario que un Edifact directo, ya que no maneja ningún símbolo separador.
- **XML:** indica archivo de origen de datos de tipo XML.
- **BD:** en el formato BD se establece que la información origen estará contenida en una base de datos. Cuando se indique que el origen es una base de datos, si se pulsa el botón “Parámetros” se visualizará una ventana donde indicar los datos necesarios para contactar con la base de datos.
- **X12:** indica archivo de origen de datos de tipo X12.
- **TRADACOMS:** indica archivo de origen de datos de tipo TRADACOMS.

Esto permite identificar a ebiMap la forma en que los archivos de origen de datos para la interfaz contienen la información. Dentro de esta misma ventana el botón “**Parámetros**” permite visualizar las propiedades de que dispone esa elección del orígenes de datos. Esta ventana muestra las diferentes posibilidades para configurar el origen de datos, estas posibilidades son a grandes rasgos; el tipo de separador de campos, el tipo de datos contenidos (enteros, carácter), representación de números y la ruta completa donde se ubicará el fichero del cual se tomará la información para su lectura.

## 4.3 OPCIONES

### 4.3.1 OPCIONES DEL MENÚ DESPLEGABLE

El área de trabajo de la ventana de “*Edición de interfaces*” tiene un menú desplegable con opciones propias, este menú se despliega a partir de una flecha hacia abajo localizada en la parte superior derecha del Árbol de registros. Las opciones son:

Opción	Descripción
<b>Propiedades</b>	Permite acceder al diálogo de propiedades de la interfaz.
<b>Guardar.</b>	Permite guardar el contenido o modificación de la interfaz, al guardar automáticamente ebiMap ejecuta una compilación
<b>Guardar Como</b>	Permite guardar la interfaz en un formato distinto al actual.
<b>Compilar.</b>	Permite compilar los cambios que se le hayan hecho a la interfaz, y en el caso que exista un error o alguna inconsistencia, pondrá el resultado de la compilación (indicando los errores en rojo) en el panel de propiedades de la interfaz, en la pestaña “Errores”.
<b>Validar.</b>	Permite probar la interfaz con un archivo de datos que contenga exactamente esa misma estructura.
<b>Datos</b>	Permite cargar un archivo de datos en la interfaz. Se accede a la ventana de datos donde se pueden visualizar ordenadamente los campos de la interfaz con datos cargados. Es necesario cargar datos para validar una interfaz.

### 4.3.2 OPCIONES DEL MENÚ CONTEXTUAL ÁRBOL DE REGISTROS.

El área del árbol de registros tiene un menú contextual que aparece con botón derecho del ratón. Las opciones de este menú son:

Opción	Descripción
<b>Nuevo.</b>	Permite agregar un registro al árbol de la interfaz,
<b>Nuevo Hijo</b>	Permite agregar un registro dependiente de un registro ya existente en al árbol de la interfaz,
<b>Importar Registro</b>	Permite importar un registro de una base de datos.
<b>Importar Registro Hijo</b>	Permite importar un registro de una base de datos como registro Hijo.
<b>Cortar</b>	Permite cortar un registro para pegarlo en otro lugar del árbol de registros.
<b>Copiar</b>	Permite copiar un registro para duplicarlo en otro lugar del árbol de registros.
<b>Pegar</b>	Permite pegar un registro copiado o cortado previamente
<b>Borrar</b>	Elimina un registro del árbol de registros.
<b>Expandir</b>	Permite expandir el registro seleccionado o todos los registros del árbol
<b>Contraer</b>	Permite Contraer el registro seleccionado o todos los registros del árbol

### 4.3.3 OPCIONES DEL MENÚ CONTEXTUAL LISTADO DE CAMPOS.

El área del listado de los campos permite acceder a los campos que contiene el registro seleccionado en el árbol de registros, Esta área tiene un menú contextual que aparece con botón derecho del ratón y permite la gestión los campos. Las opciones de este menú son:

Opción	Descripción
<b>Nuevo.</b>	La opción de “nuevo” permite crear un campo dentro de un registro, Este campo nuevo posee ciertas características que pueden ser personalizadas por el usuario. Señalar que las propiedades del campo se modifican en el panel de propiedades de los objetos que componen la interfaz, este panel se tratará con detalle más adelante.
<b>Cortar.</b>	Esta opción permite cortar un campo para después pegarlo en el mismo registro u otro; basta seleccionar el campo y hacer clic en el botón “Cortar”, esta opción es muy práctica ya que permite trasladar un campo ya definido (con todas sus propiedades) de un registro a otro.
<b>Copiar.</b>	La opción permite copiar un campo o un grupo de estos, y pegarlos (con todos sus parámetros) en otro registro dentro de la interfaz, muy práctico cuando existen campos que se repiten varias veces,
<b>Pegar.</b>	Permite pegar un campo cortado o copiado donde se desea.
<b>Borrar.</b>	Elimina un campo o un grupo de campos seleccionados.

## 4.4 GESTIÓN

---

### 4.4.1 CREAR O IMPORTAR UNA INTERFAZ.

Para desarrollar una interfaz nueva hay dos caminos:

#### *Crear nueva interfaz*

Para crear una nueva interfaz seleccionar:

- Menú principal “**Archivo-Nuevo**”: opción **Nueva interfaz**
- Barra de herramientas “**Nuevo**”: botón **Nueva Interfaz**.

#### *Importar una interfaz y editarla a continuación.*

En la barra de menú en la opción de “**Archivo**” seleccionar la opción “**Importar**”, de esta manera aparecerá la ventana de diálogo que permitirá seleccionar el tipo de archivo que se desea importar a ebiMap.

También se mostrará una ventana donde elegir una ruta para depositar el archivo importado. Una vez realizada la importación, se debe abrir el archivo y editarlo para poder trabajar con él.

Para abrir un archivo importado:

- barra de herramientas: botón **Abrir**
- menú principal “**Archivo**”, opción **Abrir**

### 4.4.2 AÑADIR REGISTROS.

Al crear una nueva interfaz el árbol de registros estará vacío. O en el caso de una interfaz importada puede ser necesario modificar el árbol de registros.

La creación de un árbol de registro se debe realizar añadiendo registros. Para ello seleccionar desde:

- Menú contextual de la ventana del árbol: opción **Nuevo Registro**
- Barra de herramientas principal: opción **Añadir** (situándose en el área de registros)

Después de crear un nuevo registro hay que darle un nombre adecuado.

Señalar que la dependencia entre un registro y otro, (relación padre-hijo) se puede definir de modo parecido desde:

- El menú contextual de la ventana del árbol: opción “**Nuevo Hijo-Registro**” (con el registro padre seleccionado)

Esta dependencia será definida mediante la asociación de campos entre los registros.

### ***Propiedades de un registro.***

Dentro de las propiedades de un registro tenemos las siguientes:

- **Nombre** : contiene el nombre del registro.
- **Descripción** : es una descripción breve del registro.
- **Acceso a datos**: en esta parte se define la relación lógica entre la estructura del registro y el origen de datos que fue definido (Edifact, lot, texto,etc).
- **Maestro-Detalle**: permite establecer la relación entre registros dentro de ebiMap
- **Filtro**: permite establecer mediante funciones un filtro de datos.
- **Etiqueta**: es el nombre de la etiqueta que tiene el registro (nombre).
- **Inicio Etiqueta**: posición que indica el inicio de la etiqueta.
- **Estado**: pudiendo tener tres tipos:
- **Condicional**: registro que puede aparecer o no.
- **Mandatario**: registro que debe de forzoso
- **No aplicable**: registro sin uso.
- **Cardinalidad Mínima**: número mínimo de aparición.
- **Cardinalidad Máxima**: número máximo de aparición.
- **Admite Valor**: condición de admisión de valores a registro.
- **Es grupo**: condición de pertenencia a un grupo de registros.

### **4.4.3 AÑADIR CAMPOS AL REGISTRO.**

Una vez creados los registros en el árbol de registros se deben crear los campos asociados a cada registro.

Para ello se debe seleccionar el registro y aparecerá a la derecha en el área de campos, los campos asociados a dicho registro.

La creación de campos se realiza ubicándose en el área de campos (a la derecha ) y seleccionando:

- Menú Contextual: opción **Nuevo**
- Barra de herramientas: botón **Añadir**.



Así se creará un nuevo campo asociado al registro seleccionado en el árbol de registros. Cada vez que se desee agregar un nuevo campo asociado a un registro se deberá seguir el mismo procedimiento. Aunque también se puede cortar , copiar y pegar un campo de un registro a otro. Esto es muy importante a la hora de crear campos relacionados entre registros diferentes. Como se verá a continuación en el enlace entre registros.

### ***Las propiedades de cada campo:***

Estas son las propiedades que puede contener un registro dentro de una interfaz, las propiedades de un campo son:

- **Nombre:** nombre del campo.
- **Descripción:** breve descripción del campo.
- **Tipo de datos:** muestra el tipo de datos que contendrá el campo pudiendo ser alfanumérico, alfabético, numérico, fecha y fecha y hora.
- **Tipo de campo:** la estructura del campo, si es un dato simple, un dato compuesto o un subdato.
- **Longitud mínima:** definición obvia.
- **Longitud máxima:** definición obvia.
- **Decimales:** el número de decimales del campo.
- **Repeticiones:** número de repeticiones del campo.
- **Estado:** pudiendo tener tres tipos:
- **Condicional:** campo que puede aparecer o no.
- **Mandatorio:** campo que debe de forzoso
- **No aplicable:** campo sin uso.
- **Valor fijo:** establece un valor fijo en el campo.
- **Calificado por:** establece una calificación mediante la comparación con otro campo.
- **Lista de valores:** establece valores mediante la aplicación de una lista de valores, introduciendo el nombre de esta lista en la casilla.

### **4.4.4 ENLACE ENTRE REGISTROS (CLAVES AJENAS).**

El enlace entre registro se realiza identificando campos comunes dentro de los registros, y tomando el principio de bases de datos, definiendo este campo como llave (Key) que será común en los registros y que permitirá identificar los datos y su origen común.

Por ejemplo se definen dos registros, uno llamado “cabecera” y uno “líneas”. En estos dos registros se establece una relación mediante claves ajenas (campos comunes).

El campo será llamado “id”, este será la clave principal dentro de la relación entre registros en esta interfaz. Este campo lo compartirán los dos registros

Para relacionar el registro “cabecera” con el registro “líneas” se agrega el campo “id” a ambos y se especifica para “Lineas” la propiedad “**Maestro-Detalle**” con una función como la siguiente:

```
id=Cabecera.id
```

En este caso la propiedad “**Maestro-Detalle**” se especifica en el registro hijo que en este caso es “Lineas” de manera que el campo “id” de líneas sea igual al campo “id” de la cabecera

El área de los campos de la interfaz se encuentra del lado derecho de la ventana de interfaz y contendrá los campos que están incluidos en cada uno de los registros creados dentro del árbol de registros. Para crear un campo basta seleccionar nuevo campo, del menú contextual (aparece con botón derecho del ratón) o desde la barra de herramientas.

#### 4.4.5 COMPILAR LA INTERFAZ.

Una vez establecidas las relaciones y agregados los campos adecuados se procede a compilar, desde:

- Menú Principal “**Ejecución**”: opción **Compilar**.(Ctrl+F9)
- Barra de herramientas: opción **Compilar**.(Ctrl+F9)

Al hacer la compilación aparecerá la información resultante de compilación de la interfaz, El nombre de esta, la fecha. Y todos los posibles errores cometidos durante la compilación. Esta información se lista en la pestaña “*Información*” del panel de propiedades en la parte inferior de la ventana.

#### 4.4.6 VALIDAR LA INTERFAZ.

Esta opción de validación se encuentra en la barra de herramientas y permite comprobar que la estructura diseñada sea congruente, El resultado de la validación se muestra en la pestaña “*Información*” del área de propiedades de la interfaz. En este caso se muestran los datos que lee la interfaz definida.

#### 4.4.7 GUARDAR LA INTERFAZ.

Al hacerlo ebiMap realizará una compilación de la interfaz, El resultado de el proceso de compilación y guardado se muestra en la pestaña “*Información*” del panel de propiedades de la interfaz

Si existiera algún error este se mostraría en color rojo y se preguntaría si desea guardar la interfaz aun con los errores. Esta detección de errores se debe al proceso de compilación que realiza en primera instancia antes de guardar la interfaz.

## 4.5 EJEMPLO: DISEÑO DE UNA INTERFAZ.

En este ejemplo se procede a construir una interfaz sencilla que se utilizará en capítulos posteriores.

Será la recepción de una orden de compra en EDIFACT para transformarla en un fichero de texto plano. En este caso se diseña la interfaz destino (ficheros planos que contendrán la información de la orden de compra).

Lo primero que se debe hacer es estudiar la estructura que contendrá la interfaz, Pensando como se debe estructurar la interfaz de la mejor manera posible.

Se opta por la generación de dos archivos distintos que contendrán separadamente la información de cabeceras de las ordenes de compra (Archivo de Cabeceras) y la información de detalle de las mismas (Archivo de Líneas): Los dos archivos estarán ligados por un campo ID que identificará a que pedido corresponde cada línea.

Cada uno de estos orígenes de datos se reflejará en la interfaz como un registro, el registro CAB y el registro LIN. Señalando a la interfaz donde debe obtener cada tipo de dato para cada uno de los registros. Esto se hace a través de la configuración de el origen de datos (IEAD)de cada uno de estos registros.

El siguiente paso es definir la información que se necesita mapear a la interfaz destino, es decir los campos de cada registro.

Dentro del registro cabecera contendrá la siguiente información:

Nombre campo	Longitud	Tipo	Descripción
NUMPED	15	Alfanumérico	número de pedido
FECPED	15	Alfanumérico	Fecha de pedido
FECEM	15	Alfanumérico	Fecha de embarque
FECAN	15	Alfanumérico	Fecha de cancelación
EANBY	15	Alfanumérico	EAN de comprador
NOMBY	40	Alfanumérico	Nombre de comprador
EANSU	15	Alfanumérico	EAN de proveedor
NOMSU	40	Alfanumérico	Nombre de proveedor
EANST	15	Alfanumérico	EAN de destino
NOMST	40	Alfanumérico	Nombre de destino
TOTAL	15	Alfanumérico	Total de la orden.
ID	5	Alfanumérico	Campo identificador del Cabecera

Dentro del registro de líneas contendrá la siguiente información:

Nombre campo	Longitud	Tipo	Descripción
EANART	20	Alfanumérico	EAN / UPC de artículo
NUMADI	20	Alfanumérico	Número adicional de artículo.
CANT	5	Alfanumérico	Cantidad
PRECIO	15	Alfanumérico	Precio
EMPAQ	15	Alfanumérico	Empaque
DESC	45	Alfanumérico	Descripción
ID	5	Alfanumérico	Campo identificador del Cabecera
IDLIN	5	Alfanumérico	Campo identificador del rLinea

Esta será la estructura de nuestra interfaz destino. Es una interfaz sencilla que procederemos a construir en ebiMap.

En los campos que se han escrito arriba están considerados además de los campos de datos contenidos en los archivos de origen, los campos que servirán para identificar el registro y establecer relaciones entre los registros Cabecera y los registros Linea, estos se llamarán ID para identificar las cabeceras e IDLIN para identificar las líneas. En el caso del registro de líneas, se debe definir tanto el campo ID como el campo IDLIN Estos campos estarán relacionados de la siguiente forma:

Registro	Documento 1	Documento 2
ID	1	2
NUMPED	1123	12233
FECHA PED	19990506	19990506
FECHA EM	19990510	19990511
EAN BY	111111111116	111111111116

#### Archivo LIN

Registro	Línea 1	Línea 2	Línea 3	Línea 4
ID	1	1	2	2
IDLIN	1	2	1	2
EAN	8411111111111	8411111111112	8411111111117	8411111111119

Una vez conocida la forma de relacionarlos, pasaremos a entrar en el Editor de interfaces de ebiMap. Seleccionando:

- Menú principal “**Archivo-Nuevo**”: opción **Nueva interfaz**
- Menú contextual “**Nuevo**”: opción **Nueva Interfaz**

Se mostrará la ventana de creación de nuevo interfaz.

Una vez dentro de la ventana se deben crear los registros de cabecera y de líneas en el árbol de registros de la nueva interfaz (que aparece vacío)

- Se creará el registro [CAB]

- A continuación se creará el Registro [LIN] como Hijo del anterior.

Después de ser creados se procede a rellenar las “Propiedades” de cada registro. Empezando por el registro CAB y después con LIN.

### 4.5.1 CONFIGURACIÓN DE PROPIEDADES DE CAB Y LIN

#### *Propiedades de CAB*

- **Nombre:** CAB.
- **Descripción :** registro de cabeceras.
- **Acceso a datos:** dentro de esta propiedad daremos click en el botón asociado (...) dentro del campo valor para abrir la ventana de “Propiedades de acceso a datos (IEADs)”

De las opción de esta pantalla se seleccionara “Añadir” para añadir un nuevo Origen de datos. En este caso la configuración del nuevo IEAD será

- **Nombre Origen Datos:** el nombre “CAB”
- **Tipo de dispositivo de datos (IEAD):** opción de “Texto Plano”

En este mismo diálogo se accede a la configuración de los parámetros de lectura de este IEAD Texto (a través del botón “Parámetros”) para este ejemplo

#### *Configuración del IEAD Texto:*

- **Nombre Fichero:** CAB.TXT.

Otras opciones de este diálogo que es posible configurar son

- **Numéricos:** Se pueden indicar las siguientes opciones, sobre los caracteres numéricos del IEAD.
  - **Carácter Decimal:** por defecto “,”.
  - **Carácter Miles:** por defecto “.”.
  - **Posición Signo:** indica la posición del signo, a la Izquierda o a la derecha.
  - **Caracteres Más:** permite indicar el tipo de símbolo utilizado para indicar el signo del valor numérico (un “+”, “0”, un espacio, o ninguno)
  - **Delimitar Numéricos:** indica delimitación entre los valores numéricos.
  - **Generar 0 antes decimal:** indica que se generará un cero antes del carácter decimal si no hay ningún valor entero.
  - **Número fijo decimales:** indica que los caracteres decimales serán fijos.

- **Alinear Num. Izda:** indica la alineación de los números a la izquierda.
- **Permitir Num. vacíos:** permite la existencia de números vacíos.
- **Completar con ceros:** los números se completarán con ceros
- **Caracteres Separadores.** Se pueden indicar los caracteres de separación para identificar en el texto plano :
  - **Registros:** el inicio de un nuevo registro.
  - **Campos:** el inicio de un nuevo Campo.
  - **Repetición campo:** el inicio de una repetición del campo.
  - **Delimitador Texto:** el carácter delimitador de texto.
  - **Car. Escape:** el carácter de escape.
  - **Car. Comentario:** el carácter de escape.
- **Opciones:** Se pueden indicar las siguientes opciones de optimización:
  - **Compresión de registro:** se eliminan los últimos caracteres si son blancos.
  - **Compresión de registro:** se eliminan los últimos campos vacíos.
  - **Borrar fichero si está vacío.**
  - **Un fichero por documento** (Opción inhabilitada)
  - **Generar error si restan datos al finalizar el recorrido.**
  - **Forzar compactación de ficheros.**

Con esto finalizaría la configuración del Registro CAB.

### ***Propiedades de LIN.***

- **Nombre:** LIN.
- **Descripción:** registro de líneas.
- **Acceso a datos:** dentro de esta propiedad daremos click en el botón asociado (...) dentro del campo valor para abrir la ventana de “Propiedades de acceso a datos(IEADs)”

De las opción de esta pantalla se seleccionara “Añadir” para añadir un nuevo Origen de datos. En este caso la configuración del nuevo IEAD será

- **Nombre Origen Datos:** el nombre “LIN”
- **Tipo de dispositivo de datos (IEAD):** opción de “Texto Plano”

En este mismo diálogo se accede a la configuración de los parámetros de lectura de el IEAD Texto (a través del botón “**parámetros**”) para este ejemplo

En este caso la única modificación a realizar en los parámetros de lectura del IEAD Texto será el nombre de fichero.

- **Nombre Fichero:** *LIN.TXT*

## 4.5.2 CREAR CAMPOS PARA CAB Y LIN.

Una vez hecha la configuración de las propiedades de cada registro (CAB y LIN) se procede a crear los campos que contendrán cada uno de ellos. Se hace desde el área de campos (a la derecha de la ventana de diseño de la interfaz) con el registro adecuado seleccionado y con la opción “**Nuevo**” del menú contextual de el área de campos.

Una vez creado el campo se procede a rellenar las propiedades de este, las propiedades de estos campos se configuran en el panel de propiedades. (sección inferior de la ventana de diseño de la interfaz) teniendo el campo que se desea editar seleccionado.

De aquí en adelante se repite el mismo procedimiento (tanto para el registro CAB como para LIN ) para construir los campos que anteriormente se han designado con las propiedades indicadas en las tablas del punto anterior.(Nombre, campo, Longitud, Tipo, Descripción)

Ya están especificados los campos que servirán de enlace entre los dos archivos,(ID e IDLIN) estos campos tienen la longitud de 5 y son alfanuméricos.

El siguiente paso será relacionar los archivos de cabeceras y de líneas, esta relación la estableceremos en las propiedades del registro de Líneas en la sección “**Maestro-Detalle**” donde se especifica que `ID=[CAB.ID]` ,

Esta relación significa que el campo ID del registro LIN será igual al campo ID del registro CAB, con el cual relacionamos la cabecera con las líneas del detalle.

Guardar la Interfaz.

El último paso es Compilar y Guardar la interfaz. Pero la guardaremos con el nombre de *PEDIDOS\_ENTRADA\_CLIENTES.IN4*

Para cambiar el nombre desde:

- Menú principal: Opción **Propiedades** (F4)
- Barra de herramientas: botón **Propiedades**

Así se accede al dialogo de propiedades de interfaz donde es posible indicar el nuevo nombre con que se guardará la interfaz en la casilla “**Nombre Interfaz**”

Esta interfaz esta lista para utilizarse en los siguientes ejemplos.

## 5 DISEÑO DE MAPAS (BÁSICO).

### 5.1 INTRODUCCIÓN.

---

En este capítulo se mostrará de forma sencilla como se diseña un mapa.

Un mapa comprende la utilización de dos interfaces, una de origen y otra de destino y una serie de asociaciones entre campos, funciones y comandos para trasladar los datos de un formato a otro. Manteniendo la integridad de los datos en el proceso.

En este capítulo se explicará la base del diseño de mapas.

- Las propiedades de un mapa.
- Asociación de campos entre las interfaces origen y destino.
- Compilar el mapa y ver el resultado
- Como guardar el mapa.
- Como ejecutar el mapa: y ver el resultado en la pestaña de “*Datos*” del panel de propiedades.

Al crear o editar un mapa, se muestra una ventana de diseño de “*Mapas*” con las siguientes áreas

#### 5.1.1 ÁREA DEL ÁRBOL DE REGISTROS DE LA INTERFAZ ORIGEN.

El árbol de registros de la interfaz origen representa de una manera gráfica la composición del origen dentro de un mapa, los registros padre, los registros hijo, los grupos de registros y los campos contenidos en cada uno de ellos.

#### 5.1.2 ÁREA DEL ÁRBOL DE REGISTROS DE LA INTERFAZ DESTINO.

El árbol de registros de la interfaz destino representa a su vez la estructura de registros, registros hijos, grupos de registros y campos a los que se desea mapear el contenido.

#### 5.1.3 PANEL DE PROPIEDADES DE LOS OBJETOS QUE COMPONEN AL MAPA.

En la parte inferior de la ventana de diseño de “*Mapas*” esta el panel de propiedades e información dividido en varias pestañas

##### ***Pestaña propiedades mapeado.***

Dentro de la pestaña “*propiedades mapeado*” se presentan las propiedades relativas al mapeado de cada registro seleccionado.(los campos de origen no tienen propiedades relevantes para el mapeado pero si los campos de destino.)



Esta pestaña esta dividida en dos partes independientes. A la izquierda para los registros de la interfaz de origen, y a la derecha para los registros o campos de la interfaz de destino.

Estas propiedades modelan el comportamiento del mapeado registro a registro y campo a campo.

Propiedades del registro seleccionado de la interfaz de origen

- **Script de inicio:** para definir un posible script de inicio
- **Script de final:** para definir un posible script de final
- **Script inicio Grupo.**
- **Script final Grupo.**
- **Disparar si vacío.** (true o false) Esta opción permite ejecutar un script aunque el registro donde se ha ubicado se encuentra sin información.

Propiedades del registro seleccionado de la interfaz de destino

- **Crear registro si:** para insertar la condición de creación de Registros
- **Script Al crear:** para definir un script que se evalúe antes de crear el registro.
- **Tipo de creación:** “según datos” o “forzado” como se vio en el capítulo anterior.
- **Depende del registro:** para indicar un registro origen del que depende la creación del registro destino.
- **Descripción:** para añadir una descripción al registro destino.

Propiedades del campo seleccionado de la interfaz de destino

- **Valor:** permite asignar un valor de una variable o expresión.
- **Condición:** permite definir una expresión Ediwin Script, directamente o con ayuda del “*Editor de comandos*”.
- **Evaluar:** cuando evaluar la condición, Durante el mapeado, al crear el registro, antes de abandonar el registro
- **Tipo Colisión:** crear, sobrescribir o cascada.
- **Tipo Conversión:** ninguno, normal, inversa o descripción.
- **Lista conversión:** desplegable para elegir una de las listas de conversión presentes en el grupo de trabajo.

***Pestaña propiedades interfaz.***

Dentro de la pestaña “*propiedades interfaz*” se muestran las propiedades relativas a las características de formato de los registros y campos de cada una de las interfaces, Estas propiedades se definen en la ventana de edición de “*Interfaces*” y en la ventana de diseño de “*Mapas*” sólo se visualizan pero no se pueden modificar. Para modificar estas propiedades hay que editar esta interfaz por separado.

***Pestaña Información.***

Dentro de la pestaña “*Información*” se muestra información referente a la compilación o a la validación del mapa, si existiera algún error, es en esta sección donde se proporcionaría la información.

***Pestaña Datos.***

Dentro de la pestaña “*Datos*” se puede visualizar o añadir (pegando el contenido) un archivo de datos para realizar las comprobaciones del mapeado.

***Pestaña Dependencias***

Permite visualizar los objetos que se utilizan en el mapeado. Por ejemplo las dos interfaces, las listas de equivalencia utilizadas, Informes...

***Pestaña Búsqueda***

La pestaña “*Búsqueda*” permite buscar registros y campos en el mapa.

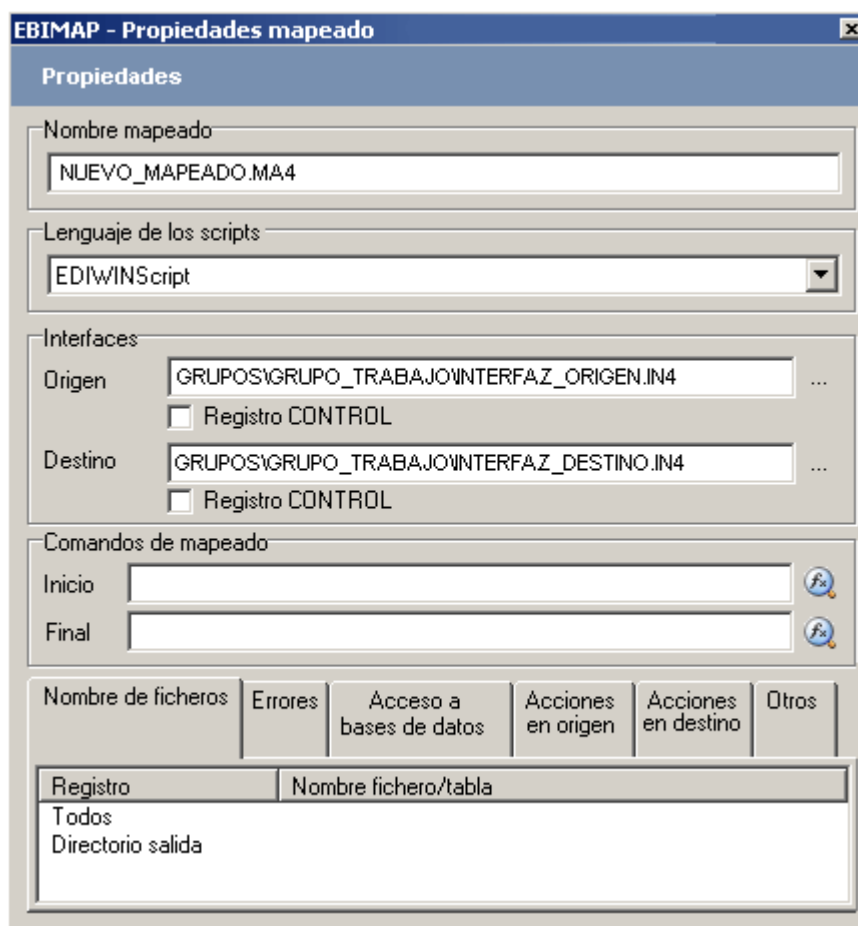
## **5.2 PROPIEDADES DEL MAPA.**

---

Las propiedades del mapa se pueden visualizar y editar desde:

- Barra de herramientas principal: botón **Propiedades**
- Menú principal “**Archivo**”: opción **Propiedades**

Dentro de las propiedades del mapa se definen las interfaces origen y destino y el nombre del mapa, existen también un área para establecer parámetros de inicio y final del mapa (scripts) así como el área de establecimiento de parámetros dividido en varias pestañas.



### 5.2.1 NOMBRE MAPEADO.

El nombre del nuevo Mapa se establece en la ventana de propiedades, en el campo **“Nombre del Mapa”**. Los archivos de Mapas tienen la extensión **“\*.ma4”**

### 5.2.2 INTERFAZ ORIGEN

La interfaz origen que se define dentro de las propiedades del mapa, es la interfaz que contendrá la información que se desea cambiar hacia otro modelo de datos (destino). Cuando se determina el archivo que contiene la interfaz origen, sí se pulsa el botón de Aceptar, inmediatamente se carga su árbol en la parte izquierda de la ventana del mapa.

Ya visualizada la interfaz se puede expandir el árbol de registros y ver en la parte inferior, en el panel de propiedades, las propiedades de esta interfaz origen.

### 5.2.3 INTERFAZ DESTINO.

La interfaz destino que se define dentro de las propiedades del mapa, es la interfaz que recibirá la información contenida en el modelo de datos origen, cuando se determina el archivo que contiene la interfaz destino aparece su árbol en la parte derecha de la ventana del mapa.

Ya visualizada la interfaz de destino, es posible expandir el árbol de registros y ver en la parte inferior, en el panel de propiedades, las propiedades de esta interfaz destino.

En resumen la ventana de diseño de “*Mapas*” estará dividida en dos partes, una para la interfaz de origen(a la izquierda) y otra para la interfaz de destino (a la derecha)

## 5.3 OPCIONES

---

### 5.3.1 MENÚ DESPLEGABLE

Opción	Descripción
<b>Propiedades</b>	Permite acceder a el diálogo de configuración de las propiedades del mapa.
<b>Ejecutar</b>	Permite ejecutar el mapeado. Para ejecutar el mapeado deben cargarse archivos de datos.
<b>Guardar</b>	Permite guardar el mapeado, antes de guardar se produce una compilación.
<b>Guardar como</b>	Permite guardar el mapeado en otro formato distinto a MA4
<b>Imprimir</b>	Permite extraer un resumen de las características del mapeado.
<b>Cerrar</b>	Permite cerrar la ventana de edición de mapas.

### 5.3.2 MENÚ CONTEXTUAL DE LA INTERFAZ DE ORIGEN

Opción	Descripción
<b>Visualizar todo el árbol</b>	Expande todos los nodos del árbol de registros de la interfaz de origen.
<b>Ocultar todo el árbol</b>	Oculto todos los nodos del árbol de registros de la interfaz de origen
<b>Editar Origen</b>	Abre el archivo IN4 Correspondiente a la interfaz de Origen en una nueva ventana de edición
<b>Refrescar</b>	Refresca la estructura del Árbol de registros. Útil si este mismo interfaz está siendo modificado en otra ventana.

### 5.3.3 MENÚ CONTEXTUAL DE LA INTERFAZ DE DESTINO.

Opción	Descripción
<b>Duplica Registro</b>	Permite duplicar el registro seleccionado
<b>Nueva Repetición</b>	Permite añadir una nueva repetición del campo seleccionado
<b>Borrar Repetición</b>	Permite eliminar una repetición del campo seleccionado.
<b>Visualizar todo el árbol</b>	Expande todos los nodos del árbol de registros de la interfaz de destino.
<b>Ocultar todo el árbol</b>	Oculto todos los nodos del árbol de registros de la interfaz de destino
<b>Editar Destino</b>	Abre el archivo IN4 Correspondiente a la interfaz de destino en una nueva ventana de edición
<b>Refrescar</b>	Refresca la estructura del Árbol de registros. Útil si este mismo interfaz está siendo modificado en otra ventana.

## 5.4 GESTIÓN

---

### 5.4.1 CREAR UN NUEVO MAPA.

Lo primero que se debe hacer es crear un nuevo mapa. A través de las opciones:

- Menú Principal **Archivo – Nuevo**: opción **Nuevo Mapeado**.
- Barra de herramientas **Archivo – Nuevo**: opción **Nuevo Mapeado**.

Al haber elegido la opción de nuevo mapa, ebiMap presenta la ventana del nuevo mapa, dividida en dos zonas. Una para la interfaz origen (a la izquierda) y otra para la interfaz destino (derecha). En este caso, al tratarse de un nuevo mapeado, ambas se visualizan vacías.

### 5.4.2 PROPIEDADES DEL NUEVO MAPEADO.

Las propiedades del nuevo mapeado que interesa configurar son su nombre, la interfaz origen y la interfaz de destino.

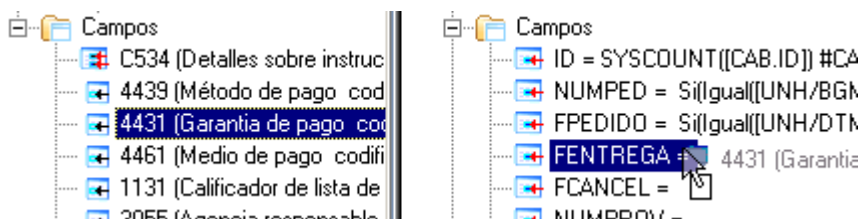
Para acceder a las propiedades del mapeado seleccionar:

- Menú principal: opción **Propiedades**.
- Barra de herramientas: opción **Propiedades**.
- Menú desplegable de la ventana de mapas: opción **Propiedades**.

Se visualizará así el dialogo de propiedades del mapa. Donde definir el nombre, y la interfaz de origen y destino

### 5.4.3 ASOCIACIÓN DE CAMPOS ENTRE LAS INTERFACES ORIGEN Y DESTINO.

La asociación entre los campos de la interfaz origen y la destino se hace de manera muy sencilla, la facilidad de esta herramienta es importante, sólo se necesita tomar el campo origen del árbol de la derecha y “arrastrarlo” al campo destino de la izquierda donde se desea mapear. (Tecnología “Arrastrar y soltar”)



Cuando la interfaz de origen interprete dentro de los datos del archivo de entrada un valor correspondiente a un valor definido

El esquema de la interfaz de origen y de la interfaz de destino pueden ser muy distintos pero el valor siempre se asociará al campo asignado de este modo.

Al asignar un campo origen a un destino a veces es necesario que se cumplan ciertas condiciones de mapeo (comprobar que no es un valor mal formado, o que no corresponde a un valor que se debe evitar mapear. Etc.), esto se consigue aplicando ciertas funciones al mapa.

*Nota- Estos atributos y condiciones dentro de la asociación de campos como el “Tipo colisión” y el apartado “lista de conversión”, se explicarán en capítulos próximos.*

#### 5.4.4 COMPILAR EL MAPA.

Para la compilación sólo es necesario presionar el botón “**Compilar**” dentro de la barra de herramientas o en el menú de propiedades del mapa, después de compilar, en la pestaña información aparecerá la información respectiva a la compilación del mapeado.

#### 5.4.5 GUARDAR EL MAPA.

Para guardar el mapa solo es necesario presionar el botón “**Guardar**” en la barra de herramientas y aparecerá en la pestaña Información el resultado de guardar el mapa.

#### 5.4.6 EJECUTAR EL MAPA:

Para ejecutar el mapa creado sólo es necesario presionar el botón “**Ejecutar**” en la barra de herramientas. En la pestaña de datos aparecerá el resultado de la ejecución del mapa, Es decir, la generación de archivos que esta configurada dentro de la interfaz destino en el mapa.

### 5.5 PESTAÑA “INFORMACIÓN”

#### 5.5.1 INTRODUCCIÓN

En la parte inferior de la ventana de diseño de “*Mapas*” hay una serie de pestañas que muestran distinta información sobre el mapeado, Esta sección es el panel de propiedades.

En el panel de propiedades, entre otras, están la pestaña “*Información*” y la pestaña “*Datos*” estas pestañas sirven para mostrar el resultado de la compilación y el resultado de la ejecución del mapa respectivamente.

Los errores que se muestran en la pantalla de la pestaña “*Información*” están codificados según los siguientes códigos:

### **Errores pestaña “*Información*”.**

<b>Código de error</b>	<b>Definición.</b>
RVA	Reglas de validación. Indica los errores que hacen referencia a las reglas de validación.
REC	Recorrido en origen. Indica los errores que hacen referencia al recorrido de datos del fichero origen.
IEAD.L	Lectura de datos. Indica los errores que hacen referencia a la lectura de datos del fichero origen.
IEAD.E	Escritura de datos. Indica los errores que hacen referencia a la escritura de datos.
EQV	Listas de equivalencia. Indica los errores que hacen referencia al acceso a listas de equivalencia.
LKU	Acceso a BBDD. Indica los errores que hacen referencia al acceso a base de datos.
CNX	Acceso a BBDD. Indica los errores que hacen referencia al establecimiento de conexiones con la base de datos.
AEW	Acceso a Ediwin. Indica los errores que hacen referencia al fallo de funciones que acceden a datos contenidos en la estación Ediwin.

### **Ejemplo:**

```
«EQV»|[D.G02NAD.3036.2] No existe la lista de equivalencia «Lista»
<1382> Script: Equiv([CABECERA.VENDEDOR],"Lista",0)

«RVA»|0;Comprobar Vendedor:

«EQV»|[D.G02NAD.3036.2] No existe la lista de equivalencia «Lista»
<1382> Script: Equiv([CABECERA.VENDEDOR],"Lista",0)

«RVA»|0;Comprobar Vendedor: B15676034

«EQV»|[D.G02NAD.3036.2] No existe la lista de equivalencia «Lista»
<1382>
```

## **5.6 DISEÑO DE UN MAPA DE EJEMPLO (SENCILLO).**

### **5.6.1 INTRODUCCIÓN.**

Para este ejemplo se utilizará como punto de partida la interfaz desarrollada en el Capítulo anterior, Se utilizará para aplicar algunos de los puntos tratados en este capítulo del manual. Dentro de este ejemplo se tratará:

- Configuración propiedades de un mapa.
- Asignación directa de valores entre la interfaz de origen y la de destino.
- Compilar, Guardar y Ejecutar un mapeado.

### 5.6.2 CREAR EL NUEVO MAPA.

Lo primero que se debe hacer es crear un nuevo mapa. A través de las opciones:

- Menú Principal **Archivo – Nuevo**: opción **Nuevo Mapeado**.
- Barra de herramientas **Archivo – Nuevo**: opción **Nuevo Mapeado**.

### 5.6.3 CONFIGURAR LAS PROPIEDADES DEL NUEVO MAPA.

Las propiedades del nuevo mapeado que interesa configurar son su nombre, la interfaz origen y la interfaz de destino.

Para acceder a las propiedades del mapeado seleccionar:

- Menú principal: opción **Propiedades**.
- Barra de herramientas: opción **Propiedades**.
- Menú desplegable de la ventana de mapas: opción **Propiedades**.

Se visualizará así el dialogo de propiedades del mapa. Donde definir el nombre, y la interfaz de origen y destino

- **Nombre mapeado**: en este caso se le dará al nuevo mapeado el nombre de *ASCII\_ORDEN\_COMPRA\_CLIENTES.ma4*.
- **Interfaz Origen**: en este caso *EDI\_ORDERS\_D\_96A\_UN\_EAN008\_1.IN4*

Esta interfaz deberá estar contenida en la carpeta `C:\EDIWIN\REPOSITORIO\Usuario` y contiene la estructura de la interfaz origen del mensaje. Que servirá para obtener los datos del archivo EDIFACT original.

- **Interfaz Destino**: la interface *PEDIDOS\_ENTRADA\_CLIENTES.IN4*

La interfaz creada en un capitulo anterior. Esta interfaz debe estar contenida en la carpeta `C:\EDIWIN\REPOSITORIO\Usuario` y contiene la estructura de la interfaz destino del mensaje. Para generar archivos de formato de texto plano

Una vez realizada esta configuración de propiedades se debe pulsar el botón **“Aceptar”** para que ebiMap muestre las interfaces origen y destino en sus respectivas posiciones dentro de la ventana de diseño de mapeado.

### 5.6.4 ASIGNAR VALORES EN EL MAPA.



Una vez el mapa ya esta definido (es decir tenemos su nombre, su interfaz de origen y destino) se procede a asignar valores de campos de la interfaz destino con respecto a los campos de la interfaz origen.

Empezando por los campos que serán “mapeados directamente”, es decir, solo serán trasladados de una interfaz hacia la otra sin ningún tipo de modificación.

Para este mapa de ejemplo sólo se tomarán algunos campos de la interfaz que se ha diseñado. Aquellos campos que serán mapeados directamente. Es decir, solo serán trasladados de una interfaz hacia la otra sin ningún tipo de modificación. Estos serán:

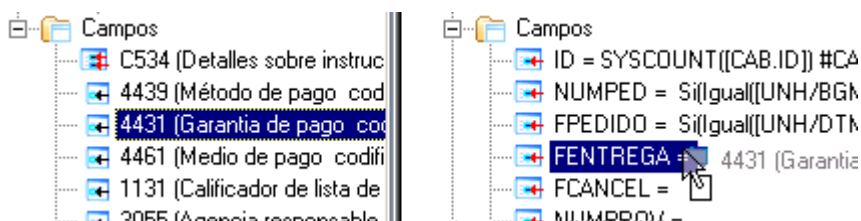
#### ***De registro cabeceras:***

- NUMPED. (número de pedido)
- FECPED. (fecha de pedido)
- EANBY. (código del comprador)
- NOMBY. (Nombre del comprador)

#### ***De registro de líneas***

- EANART. (código del articulo)
- NUMADI.(número adicional del producto)
- CANT (cantidad )
- PRECIO. (el precio)
- EMPAQUE
- DESC. (Descripción del artículo)

Se procede a seleccionar cada uno de estos campos de la interfaz origen y a arrastrarlos a al campo de destino.



Al realizar el arrastre del campo origen al destino las propiedades del registro y del campo destino se habilitarán

Para este ejemplo la correspondencia queda de la siguiente forma:

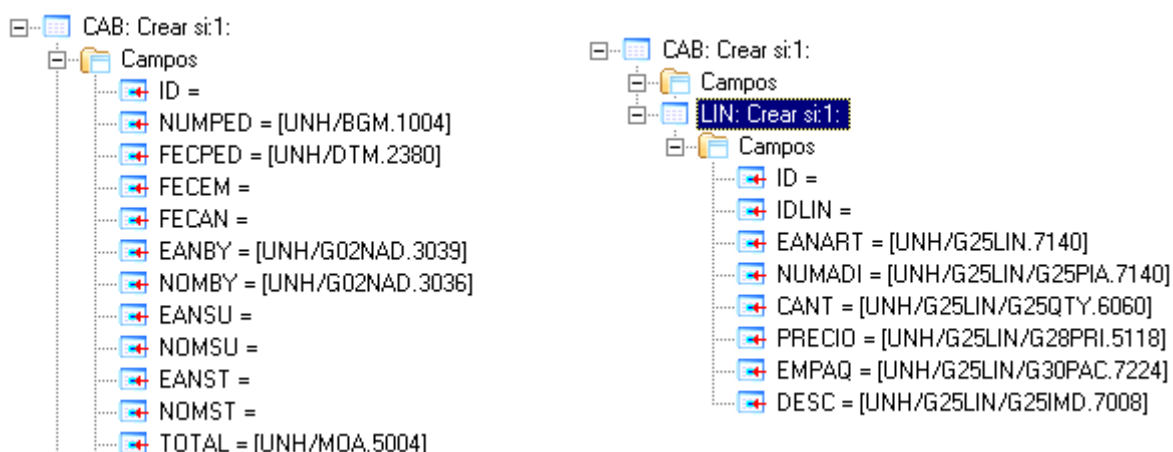
### Asignación Registro Cabeceras

Campos en destino	Campos en origen
[UNH/BGM.1004]	NUMPED
[UNH/DTM.2380]	FECPED
[UNH/G02NAD.3039]	EANBY
[UNH/G02NAD.3036]	NOMBY
[UNH/MOA.5004]	TOTAL

### Asignación Registro Líneas

Interfaz de Origen (EDIFACT)	Interfaz de destino (Texto plano)
[UNH/G25LIN.7140]	EANART
[UNH/G25LIN/G25PIA.7140]	NUMADI
[UNH/G25LIN/G25QTY.6060]	CANT
[UNH/G25LIN/G28PRI.5118]	PRECIO
[UNH/G25LIN/G30PAC.7224]	EMPAQUE
[UNH/G25LIN/G25IMD.7008]	DESC

Una vez que ya se ha hecho la asignación de valores (asignación de campos origen a destino) se procede a compilar el mapa.



### 5.6.5 COMPILAR EL MAPA DE EJEMPLO

Para compilar el mapa de ejemplo tenemos que seleccionar el botón de “Compilar” dentro de la barra de herramientas del mapa.

Al compilar, ebiMap automáticamente mostrará dentro de la pestaña de información toda la información sobre el proceso de compilación del mapa. Señalando cualquier error en color rojo.

### 5.6.6 GUARDAR EL MAPA DE EJEMPLO

El siguiente paso será guardar el mapa. La opción “**Guardar**” esta dentro de la barra de herramientas del mapa. Una ventaja que tiene la opción “**Guardar**” es que antes de hacerlo compila el mapa .mostrando los errores en la pestaña información.

Una vez compilado y guardado el mapa, se puede realizar la ejecución del mismo.

### 5.6.7 EJECUTAR EL MAPA DE EJEMPLO

Para ejecutar el mapa que se ha creado solo es necesario presionar el botón “**Ejecutar**” dentro de la barra de herramientas de la ventana del mapa.

Al ejecutar el mapa aparecen dos respuestas por parte de ebiMap en la zona de propiedades del mapa.

- Dentro de la Pestaña “*Información*” aparecerá el mensaje de compilación y ejecución del mapa.
- Dentro de la Pestaña “*Datos*” aparecerán las estructuras de datos generados en base la ejecución del mapa. Es decir los archivos generados

Como se puede observar, los datos que se han mapeado son los que contiene el mensaje en formato EDIFACT que era el correspondiente a la interfaz origen, estos datos fueron mapeados tanto en el registro cabecera como en el de líneas creando dos archivos resultantes, el *CAB.txt* y el *LIN.txt*.

## 6 DISEÑO DE MAPAS (INTERMEDIO).

### 6.1 INTRODUCCIÓN

---

Dentro de este capítulo se explicará de manera más detallada el uso de funciones que permitirán aumentar la eficiencia de un mapa, utilizando las funciones de “Ediwin Script”. El uso de estas funciones es una gran ventaja ya que permite la conversión o el tratamiento de la información de origen para adecuarla a las necesidades del Sistema Informático Interno de destino.

### 6.2 CREACIÓN Y ASIGNACIÓN DE VALORES.

---

#### 6.2.1 CONDICIONES DE CREACIÓN DE REGISTROS.

La creación de campos en la interfaz de destino está controlada por la propiedad “**Tipo de Creación**” del campo de destino. Puede ser.

- Según datos.
- Forzar por registro.

##### ***Creación de campos según datos.***

Este tipo de creación es muy simple, ya que solo realiza la asignación de un valor de la interfaz origen a la interfaz destino si este existe, es decir, si no hay datos no hace ningún mapeo al campo de la interfaz destino.

##### ***Creación de campos forzado por registro.***

Este tipo de creación fuerza al mapa a realizar el mapeo aunque no exista información dentro del campo de la interfaz origen y por consiguiente a la creación del campo en la interfaz destino.

Así como en la creación de campos existen condiciones de creación, también se tienen condiciones en la creación de registros. Estas condiciones se pueden definir en expresiones “Ediwin Script”.

Los campos para definir estas expresiones están listados dentro de la pestaña “*Propiedades Mapeado*” para el registro o el campo seleccionado de la interfaz de destino.

- **Crear registro si:** para insertar una condición en forma de expresión “Ediwin Script”
- **Script Al crear:** permite definir una expresión que se evalúe antes de crear el registro.
- **Tipo de creación:** permite elegir entre dos opciones “según datos” o “forzado”

- **Depende del registro:** para indicar un registro origen del que depende la creación del registro destino. Si este registro existe el registro de destino se creará.
- **Descripción:** para añadir una descripción al registro destino.

## 6.2.2 MÉTODOS DE ASIGNACIÓN DE VALORES.

Dentro de ebiMap la asignación de valores a campos esta dada por:

- La asignación directa desde un campo de la interface origen a la interface destino sin ningún tratamiento.
- La asignación de un campo de la interface origen a la interface destino aplicando una función a la información que será depositada en el campo destino.
- La asignación de información a un campo en la interface destino calculando el valor a partir de una función o el contenido de una variable dentro del mapa.

Estas expresiones se indican en las propiedades de la pestaña “*Propiedades Mapeado*” de la ventana de edición de “*Mapas*”, y corresponden con el campo de destino entre otras propiedades. Las que interesan para la asignación de valores a los campos son:

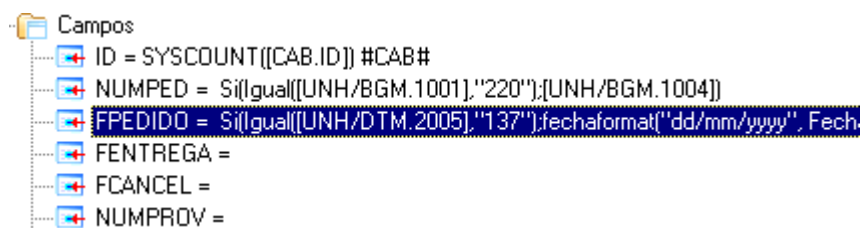
- **Valor:** permite asignar al campo un valor de una variable o expresión.
- **Condición:** permite definir una expresión Ediwin Script, directamente o con ayuda del Editor de comandos para que se genere el campo dentro del registro.
- **Evaluar:** cuando evaluar la condición anterior, “Durante el mapeado”, “al crear el registro” o “antes de abandonar el registro”

### ***Asignación directa***

La asignación directa de un campo origen a uno destino se realiza simplemente arrastrando el campo con el botón derecho del ratón hasta donde se desea mapear. No es necesario modificar ninguna propiedad.

### ***Asignación aplicando una función.***

La asignación de un campo origen a uno destino aplicando una función es necesaria cuando la información origen no está en el formato adecuado o necesita que se le aplique un proceso de transformación para que sea la información adecuada.



Un ejemplo de esto pueden ser las fechas, ya que estas pueden estar en un formato que el sistema de gestión de destino no reconocería, Aplicándole una función de Ediwin Script es posible transformar el formato de dicha fecha en el formato esperado antes de que sea almacenada en la interface destino.

Por ejemplo, al campo destino llamado “FPEDIDO” se le asigna el contenido del valor del campo origen [UNH/DTM.2380], pero antes de esto se quiere convertir la fecha en un formato adecuado además de aplicar una máscara de formato para que tenga la apariencia de dd/mm/yyyy (dos caracteres para el día, dos para el mes y cuatro para el año). Esto se define en el campo de la propiedad “**Valor**”

### ***Asignación utilizando una variable***

La asignación de un valor a un campo de la interfaz destino mediante una variable se utiliza normalmente para calcular información que no está contenida en la interfaz origen pero que es necesario reflejar en la interfaz destino y que tiene un estatus de obligatorio.

Partiendo de una orden de compra en un archivo estándar XML con varias líneas <PURCHASE ORDER LINE> como la siguiente:

```
<PURCHASE ORDER LINE>
<PRODUCT NAME>
<PRODUCT CODE>
<QUANTITY>
<PRICE>
<UNIT OF MEASURE>
</PURCHASE ORDER LINE>
```

Se necesita mapear en la interfaz destino el número de línea que se está mapeando y que permita a el sistema de gestión identificar a que línea del pedido hace referencia.

Debido a que la interfaz origen no especifica el número de línea (no hay ninguna etiqueta que indique este dato dentro de <PURCHASE ORDER LINE>) se debe hacer una función que permita contar cada una de las líneas que se están leyendo desde la interfaz origen.

Una función que se puede aplicar para este caso es la función `sysCount(«param1»)`, Esta función devuelve el número de registros leídos/escritos del tipo línea y si así se define en las propiedades del campo de destino asignara su valor a un posible campo IDLIN que identifique a cada uno de los campos. La función `sysCount(«param1»)` (donde «param1» es el tipo de registro leídos/escrito) se define en el campo de la propiedad **“Valor”**

## **6.3 EXPRESIONES (CREACIÓN DE SCRIPTS).**

---

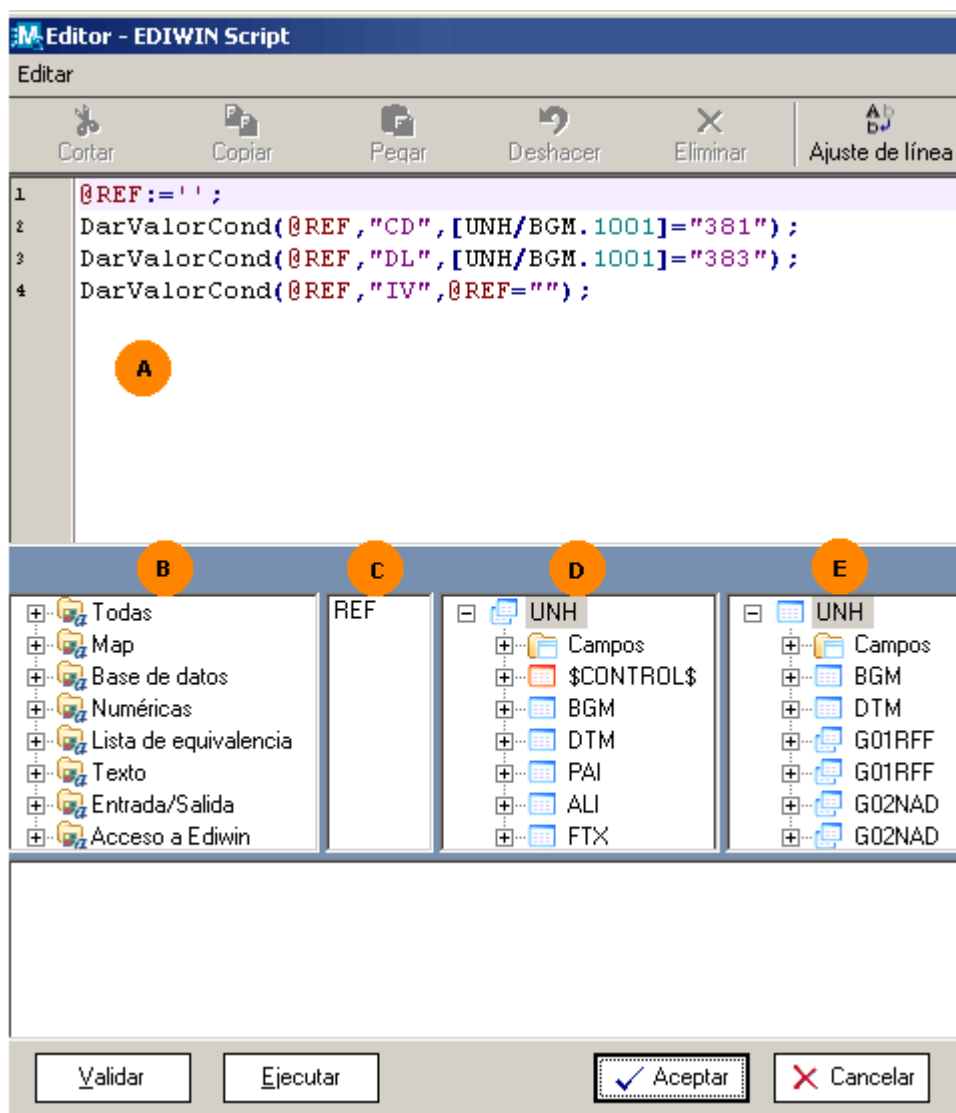
Dentro de ebiMap las funciones permiten hacer un mapa más completo en el aspecto de manipulación de información mapeada en la interfaz origen a la interfaz destino. A continuación se aborda la utilización de funciones y de grupos de funciones dentro de una mapa.

### **6.3.1 ¿QUÉ SON LAS EXPRESIONES?**

Una expresión consiste en una o varias funciones “Ediwin Script” que permiten darle un tratamiento a cierta información contenida o calculada dentro de un mapa.

### **6.3.2 ¿CÓMO SE CONSTRUYEN LAS EXPRESIONES? (USO DE VARIABLES).**

La construcción de estas expresiones se realizan con ayuda del editor de comandos dentro del ebiMap, este editor está dividido en varias secciones que permiten elaborar expresiones muy complejas y muestran todas las funciones y variables disponibles en el mapa.



Este editor de comandos esta compuesto por:

- Área de comandos: es el área que contendrá las expresiones que se desarrollen.(A)
- Área de funciones: en esta área se visualizan las funciones disponibles en Ediwin Script.(B)
- Área de Variables: dentro de esta área se visualizan las variables disponibles en el mapa (C).
- Visor de árbol de interfaz origen: visualiza el árbol origen del mapa y sus componentes (D)
- Visor de árbol de interfaz destino: visualiza el árbol destino del mapa y sus componentes (E)



## ***Declaración de variables de sistema***

EbiMap permite definir variables de sistema, estas variables tienen la característica de estar disponibles cada vez que se lance el proceso de mapeo. Por ejemplo para asignar un valor de un campo de una interfaz a una variable u otra función.

Se puede acceder al área que contendrá la definición de estas variables

- Mediante el botón **“Propiedades”** de la barra de herramientas de la ventana del mapa.
- En el área de título de la interfaz origen existe un menú colgante, para acceder a este pulsar la flecha hacia abajo y seleccionaremos la opción **“Propiedades”**.
- Sobre el título de la interfaz origen dar un clic derecho aparecerá un menú contextual, del cual elegiremos la opción **“Propiedades”**.

Para definir una variable de sistema solo hacer clic con el botón derecho del ratón sobre el área para que ebiMap muestre la ventana del *“Editor de Comandos”*, que permitirá a elaborar las variables a utilizar en el proceso de mapeado.

## ***Usar campos de interfaz como parámetros.***

Se podría decir que la razón principal de las funciones de ebiMap es aplicar cierto tratamiento a determinada información, y como la información está contenida en los campos de la interfaz dentro de las funciones podemos utilizar los campos de la interfaz. Como parámetros.

Un ejemplo de una función utilizando campos de la interfaz podría ser el siguiente:

```
fechaformat("dd/mm/yyyy", FechaIEDI([UNH/DTM.2379], [UNH/DTM.2380]))
```

En este ejemplo existe una función anidada en otra. La acción de esta función es la siguiente:

Del campo [UNH/DTM.2380] aplicarle el formato de fecha contenido en el campo [UNH/DTM.2379] y después aplica la máscara dd/mm/yyyy

Si la fecha es 20010301 (año,mes,día) y el formato de la fecha de referencia para el formateo es 102 (Día,més,año) le dará una apariencia después de la máscara de 01/03/2001.

Como se puede observar también, la lógica de acción en cuanto a las funciones es la misma que en la álgebra general, ya que realizará las funciones tomando la función más interna para después ir desarrollando hasta la última.

## **6.3.3 ¿PARA QUE SIRVEN LAS EXPRESIONES?**

Las expresiones, como ya se ha mencionado anteriormente, sirven para aplicar las funciones de Ediwin Script a campos de la interfaz origen para darle un cierto formato requerido por nuestra interfaz destino.

### 6.3.4 ¿DONDE SE DECLARAN LAS EXPRESIONES?

Las expresiones se declaran en la pestaña “*Propiedades Mapeado*” dentro de los campos de Script. Según el campo donde se defina una expresión esta será evaluada en un momento determinado durante el proceso de mapeado.

### 6.3.5 ¿EN QUE MOMENTO SE UTILIZAN LAS EXPRESIONES?

Hay cuatro posibles situaciones durante la ejecución de un mapeado donde se permite definir una expresión Ediwin Script. Corresponden al flujo de lectura del árbol de registro que realiza ebiMap a la hora de mapear. Y dependiendo del objetivo de la expresión tendrá más sentido definirlo en un momento o en otro de este flujo.

Las expresiones se pueden utilizar:

- Al inicio de un registro (o grupo) de la interfaz origen.(Script de Inicio)
- Al final de un registro (o grupo) de la interfaz origen.(Script de final)
- Al crear un registro en la interfaz destino.
- Al principio y final de la ejecución de un mapa.

Estos scripts se declaran en el panel de propiedades del registro (aparece en la parte inferior de la ventana de edición de mapas)

#### ***Definir expresión al inicio de un registro de la interfaz origen.***

Las expresiones al inicio de un registro se definen en la propiedad “Script Inicio” del registro de origen, dentro de la pestaña “*Propiedades Mapeado*” en la parte inferior de la ventana de edición de mapas.

Una expresión dentro del “**Script de Inicio**” de un registro (o un grupo) se utiliza para evaluar expresiones al inicio de la lectura de estos.

Es decir, las expresiones especificadas dentro del campo “**Script de inicio**” de un registro se evaluarán cada vez que ebiMap recorra y encuentre en un documento de origen el registro.

#### ***Ejemplo:***

Existe una expresión en el campo “**Script de Inicio**” del registro `UNH` de la interfaz origen, la expresión es `@CONT:=0`, es decir, a la variable `CONT` se le ha asignado el valor de 0, lo cual significa que cada vez que ebiMap recorra el documento origen y se encuentre el registro `UNH`, a la variable `@CONT` se asignará el valor 0 (cero).

### ***Definir expresión al final de un registro de la interfaz origen.***

Las expresiones al final de un registro se definen en la propiedad “**Script final**” del registro de origen, dentro de la pestaña “**Propiedades Mapeado**” en la parte inferior de la ventana de edición de mapas.

La utilización de una expresión al final de un registro (o un grupo de estos) se utiliza para evaluar expresiones cuando se ha completado la lectura del registro.

Es decir, las expresiones especificadas dentro del campo “**Script final**” de un registro se evaluarán cada vez que ebiMap recorra todo ese registro hasta el final. Si existe alguna expresión se evaluará antes de pasar al siguiente.

Por ejemplo. Lo más coherente, si la expresión necesita el valor de uno de los campos contenidos en el registro, es esperar hasta el final del recorrido del registro, ya que si se definiera como “**Script de inicio**” ebiMap todavía no habría pasado por el campo, el parametro todavía no tendría valor y la expresión no se ejecutaría correctamente.

### ***Definir expresión en el momento de la creación de un registro de la interfaz destino.***

El ebiMap también permite la ejecución de un script cuando se crea un registro en la interfaz destino. Esta es una forma que ebiMap proporciona para brindar un mejor control de la creación de registros en la interfaz origen de un mapa.

Estas expresiones al crear un registro en la interfaz destino se definen en esta vez para el registro destino, en el campo “**Script Al Crear**”, también en la pestaña de “*Propiedades Mapeo*”

### ***Definir expresión al principio y final de la ejecución de un mapa.***

Otra de las opciones que proporciona ebiMap es ejecutar un script al principio de cualquier proceso dentro del mapa y antes de finalizar el mapa, un ejemplo de esto podría ser la creación de variables globales antes de empezar el mapa, inicializar estas variables, realizar alguna función que permita la creación de un determinado valor, etc.

Estas variables no dependen de un registro, si no de la propia configuración del mapa. Definir una expresión al principio o al final de la ejecución del mapa se debe realizar desde el panel de propiedades del mapa. A través de:

- Menú Principal “**Archivo**”: opción **Propiedades**(F4).
- Barra de herramientas: botón **Propiedades**(F4).
- Menú desplegable de la ventana de diseño de mapas: opción **Propiedades**.

Donde existen dos campos dentro del Área de “*Comandos de Mapeado*”, uno para las expresiones evaluadas al inicio y otra para las expresiones evaluadas al final.

## 6.4 PESTAÑA “PROPIEDADES MAPEADO”

### 6.4.1 INTRODUCCIÓN

En capítulos anteriores se ha explicado la función de las pestañas “*Información*” y “*Datos*”, en este capítulo, y en relación con las funciones y expresiones de Ediwin Script se explicará el uso de la pestaña “*Propiedades Mapeado*” y “*Propiedades Interfaz*”

### 6.4.2 PESTAÑA “PROPIEDADES MAPEADO”

La pestaña “*Propiedades mapeado*” es la primera pestaña que se visualiza, la división de esta sección dentro del panel de propiedades corresponde con las dos secciones en que se divide la ventana en la parte superior, una para la interfaz origen y la otra para las propiedades de la interfaz destino.

Propiedades Mapeado	Propiedades Interfaz	Información	Datos	Dependencias	Búsqueda
Registro UNH		Registro UNH			
Propiedad		Valor			
General		General			
Script de inicio		Crear registro si			
Script de final		Script Al Crear			
Script inicio Grupo		Tipo Creación			
Script final Grupo		Depende del registro			
Disparar si Vacio		UNH			
false		Descripción			

Dentro de estas secciones del panel de propiedades se insertarán los scripts dependiendo de la forma de ejecución que deseemos. Al seleccionar cualquiera de los campos de script de las propiedades del registro (“**Script Al Crear**”, “**Script Inicio**”, “**Script Final**”) se visualiza un botón con tres puntos (...), al presionarlo se accede a la ventana del “*Editor de comandos*” Ediwin Script que facilita la tarea de crear expresiones.

Propiedades registro seleccionado de la interfaz de origen:

- **Script de inicio:** para definir un posible script de inicio
- **Script de final:** para definir un posible script de final
- **Script inicio Grupo.**
- **Script final Grupo.**
- **Disparar si vacío.** (true o false) esta opción permite ejecutar un script aunque el registro donde se ha ubicado se encuentra sin información.

*Nota - Las opciones de script de inicio Grupo y script final Grupo sólo se visualizan cuando existe un grupo de registros hijos contenidos dentro de este, si es solo un registro no se visualizaran.*

Propiedades registro seleccionado de la interfaz de destino:

- **Crear registro si:** para insertar la condición de creación de Registros
- **Script Al crear:** para definir un script que se evalúe antes de crear el registro.
- **Tipo de creación:** “Según datos” o “Forzado” como se vio en el capítulo anterior.
- **Depende del registro:** para indicar un registro origen del que depende la creación del registro destino.
- **Descripción:** para añadir una descripción al registro destino.

## 6.5 DISEÑO DE UN MAPA DE EJEMPLO. (INTERMEDIO)

### 6.5.1 INTRODUCCIÓN

Para este ejemplo se utilizará como punto de partida el mapa desarrollado en el Capítulo anterior, Se utilizará para aplicar algunos de los puntos tratados en este capítulo del manual. Dentro de este ejemplo se tratará:

- Análisis de datos.
- Formateo de fechas.
- Utilización de Variables.

### 6.5.2 ANÁLISIS DE DATOS.

En este ejemplo se va a utilizar el análisis de datos para establecer condicionalmente que valores de la interfaz origen se deben asociar a cada campo de la interfaz destino

En un principio se había establecido en la interfaz origen tres tipos de fechas para ser recibidas:

Nombre campo	Longitud	Tipo	Descripción
FECPED	15	Alfanumérico	Fecha de pedido
FECEM	15	Alfanumérico	Fecha de embarque
FECAN	15	Alfanumérico	Fecha de cancelación

La interfaz de origen , una interfaz EDIFACT, guarda las fechas en un mismo segmento (El segmento DTM) y las diferencia gracias a un valor denominado “Calificador del Segmento” que indica si se trata de una fecha de pedido (calificador 137) , empaque (2) o cancelación. (61)

En EDIFACT las fecha de pedido, embarque y cancelación serían codificadas así.

```
DTM+137:20000223:102'
DTM+2:20000224:102'
DTM+61:20000303:102'
```

En el árbol de registros de la interfaz de origen (EDIFACT) abierta en la ventana de diseño de mapas esto se traduce en una estructura de registros y campos como esta:

Nombre del interfaz	Campos del registro: UNH/DTM			
<div> <div>UNH</div> <div>BGM</div> <div>DTM</div> <div>PAI</div> <div>ALI</div> <div>FTX</div> <div>G01RFF</div> <div>G02NAD</div> <div>G06TAX</div> </div>	Nombre	Descripción	Tipo Datos	Long.
	C507	Fecha hora período	X	0
	2005	Calificador de fecha hora p...	X	3
	2380	Fecha hora período	X	35
	2379	Calificador de formato de fe...	X	3

Se podría decir que cada fecha viene determinada por un Registro con tres campos. Uno para el calificador, otro para el valor en sí (20000223) y un tercero que indica el formato de la fecha (102= ccyymmdd lo que le mapearía la fecha como 23/03/2000)

### **Asignación de la fecha de pedido al registro [CAB/FECPED]**

Ahora se configurara el campo de la interfaz de destino [CAB/FECPED] para que obtenga el valor de la fecha de pedido (y no otra fecha)

1. Se conoce que dentro del registro DTM, el campo 2380 de la interfaz origen contiene la fecha (no se sabe sin embargo que tipo de fecha es)
2. Se mapea este campo origen ([UNH/DTM.2380]) al destino ([CAB/FECPED]).
3. Dentro de la pestaña “propiedades del mapa” en el panel de propiedades, la propiedad condición del campo destino se establece por defecto en “1”. Esto quiere decir que se crearía un campo FECPED siempre (si tiene información el campo [UNH/DTM.2380]) que se encuentre un registro DTM. Se trate o no de una fecha de pedido.
4. Para hacer que se cree sólo (condición) cuando se trate de una fecha de pedido se debe definir una condición. Quitando el “1” e iniciando el editor de comandos de Ediwin Script con el botón adjunto (...)
5. Dentro del editor de comandos, elegir la función Igual(«param1», «param2»). Esta función compara dos cadenas y devuelve un “1” lógico si se cumple la condición de igualdad. Lo que permitirá especificarle a ebiMap que realice la asignación del valor siempre y cuando la condición de igualdad se cumpla.

Los parámetros de esta función para el campo FECPED deben asegurar que la fecha mapeada (campo 2380 del registro DTM ) desde la interfaz origen es la fecha de pedido, es decir que el valor en el campo “Calificador de fecha” (campo 2005) es igual al valor correspondiente a las fechas de pedido (Conocido de antemano es “137”). Por tanto la función de condición quedará así: Igual ([DTM.2005], “137”)

Campo FECPED	
Propiedad	Valor
<b>General</b>	
Valor	[UNH/DTM.2380]
Condición	Igual([DTM.2005], "137")
Evaluar	Durante el mapeado
Tipo Colisión	Crear
Tipo Conversión	Ninguno
Lista Conversión	

Cuando el contenido del campo [DTM.2005] (“calificador de la fecha”) sea igual a la cadena “137” (código EDIFACT para fecha de pedido) realizara la asignación del valor a al campo FECPED donde se ha definido la condición.

Este es un ejemplo de análisis de datos, antes de asignarlos a los campos de la interfaz destino.

*Nota- A continuación se debe proceder del mismo modo para establecer la asignación de modo condicional a FECPEM (empaquetado) y FECAN (cancelación) sabiendo que los códigos EDIFACT para fecha de cancelación y empaque son 61 y 2*

### Asignación del código de interlocutor al registro [CAB/NOMBY]

El caso de los proveedores es el mismo que el de las fechas en EDIFACT. El segmento NAD contiene varios identificadores para los interlocutores, las partes que actúan en el envío o recepción de un documento. Y como en las fechas, existe un calificador para cada uno.

Nombre campo	Longitud	Tipo	Descripción
EANBY	15	Alfanumérico	EAN de comprador
NOMBY	40	Alfanumérico	Nombre de comprador
EANSU	15	Alfanumérico	EAN de proveedor
NOMSU	40	Alfanumérico	Nombre de proveedor
EANST	15	Alfanumérico	EAN de destino
NOMST	40	Alfanumérico	Nombre del destino.

En el ejemplo del mapa, solo interesa mapear los interlocutores que sean:

- PROVEEDOR: contenido en segmento NAD con calificador SU.
- COMPRADOR: contenido en segmento NAD con calificador BY.
- DESTINO: contenido en segmento NAD con calificador ST.

En la interfaz de destino (texto plano) se quiere mapear el campo [CAB/NOMBY] con la información obtenida del nombre del “Comprador”

1. Se conoce que en el campo [UNH/G02NAD.3036] de la interfaz origen vendrá el nombre de la parte .
2. Se conoce que en el campo [UNH/G02NAD.3035] de la interfaz origen vendrá El calificador del código de interlocutor, (si es proveedor, comprador, destino).



3. Se mapea por tanto el campo origen ([UNH/G02NAD.3036]) al campo destino [CAB/NOMBY]
4. Se modifica la propiedad condición y se selecciona como en el caso anterior la función `Igual («param1», «param2»)` con los siguientes valores:

```
Igual ([G02NAD.3035], "BY")
```

Se comprueba que el campo [G02NAD.3035] (calificador de interlocutor) es igual a la cadena "BY" (comprador). Al compilar y ejecutar el mapa. En la pestaña "Datos" se verá el resultado de esta modificación en el mapa.

*Nota- El mismo procedimiento se debe realizar con el interlocutor "Destino" y Proveedor para el campo NOMSU y NOMST.*

### 6.5.3 ASIGNACIÓN DEL PRECIO AL REGISTRO [LIN/PRECIO]

Hasta ahora se han asignado valores condicionales al Registro de cabeceras [CAB]

Dentro de lo que será el detalle del mapa de ejemplo puede darse otro caso, que es el de la moneda. La interfaz origen EDIFACT envía dos tipos de precios, uno que ya incluye descuentos (Calificador AAA) y otro que no los incluye (Calificador AAC), en este ejemplo solo se necesita el precio que no los incluye. (AAC)

Sabiendo

- [UNH/G25LIN/G28PRI.5118] es el valor
- [UNH/G25LIN/G28PRI.5125] es el calificador

la condición que tendría este campo sería:

Se mapearía el campo [UNH/G25LIN/G28PRI.5118] sobre [LIN/PRECIO]

Y se condicionaría su creación a la función:

```
Igual ([G28PRI.5125], "AAC")
```

Una vez hecho compilar y ejecutar el mapa. En la pestaña de "Datos" se verá el resultado de esta modificación en el mapa.

### 6.5.4 FORMATEO DE FECHAS.

Dentro de ebiMap existen una serie de funciones que permiten una gran flexibilidad a la hora de realizar mapeados, como la manipulación de variables, lookups, formato de datos, equivalencias, etc. Un ejemplo de estas funciones que puede realizar ebiMap dentro de un mapeo es el dar formato a las fechas.

En la asignación del valor de las fechas a los campos FECPED, FECEN y FECAM el formato de la fecha era Año, Mes, Día (YYYYMMDD).

Es decir DTM+137:20000223:102 es igual a la fecha 20000223 (YYYYMMDD)

Donde 102 es el calificador del calificador de “formato de la fecha”.

Puede darse el caso que este formato de fecha no es el más apropiado para el archivo destino. Y se necesita cambiar a un formato de fecha distinto (DD/MM/YYYY, por ejemplo).

Existe una función expresamente desarrollada para este caso en “Ediwin Script” que permite realizar este formateo a una fecha EDIFACT a una fecha día / mes / año

```
FechaIEDI(«param1», «param2»)
```

Donde «param2» es una fecha EDI y «param1» es el calificador de formato de la fecha (20000223 y 102, respectivamente en el ejemplo). El resultado de la función es la fecha 23 / 02 /2000.

Esta fecha correctamente formateada es la que se debe mapear dentro del campo CAB/FECPED y no el valor tal y como aparece en la interfaz original. A parte de la condición de creación de registro , en las propiedades del campo FECPED se debe modificar la propiedad valor.

*Nota- Aplicar el mismo procedimiento para el formateo de fechas a los campos FECEN y FECAE en su propiedad “valor”, la función en este caso es la misma para los tres campos. Ya que su propiedad condición la que identifica que valor de origen debe mapearse*

Compilar, guardar y ejecutar el mapa observando que el formateo de la fecha se ha realizado como se esperaba.

## 6.5.5 UTILIZACIÓN DE VARIABLES.

EbiMap proporciona entre otras cosas el uso de variables. Estas variables son muy prácticas ya que facilitan el almacenaje de valores así como datos calculados dentro de un proceso de mapeo, por nombrar sólo algunos de sus usos.

En el mapa de ejemplo se usarán las variables para determinar los códigos de identificación entre el registro de cabecera [CAB] y los detalles [LIN] En este mapa se declararán las variables “@cabeceras” y “@líneas”

### **Declarar Variables**

La variable de “@cabeceras” se debe definir dentro del dialogo de “*propiedades del mapa*” en el área de “*Comandos de mapeado*”.

Para acceder a el dialogo de “*Propiedades del Mapa*”

- Menú principal: opción **Propiedades** (F4)
- Barra de herramientas: botón **Propiedades**
- Menú desplegable de la ventana de edición: opción **Propiedades**

Una vez declarada la variable en los comandos de mapeado, volver a la ventana de diseño de mapa y seleccionar de la interfaz origen el registro padre UNH

En el panel de propiedades, en la pestaña “*Propiedades Mapeado*”, en la propiedad “**Script de inicio**” de el registro `UNH` se debe incrementar `@cabeceras` en uno y `@lineas` inicializarse a cero.

```
@cabeceras: = +1;@lineas: = 0;
```

Una vez hecho esto seleccionar el registro `[LIN]` de la interfaz origen para introducir en el script de inicio el incremento de la variable líneas

```
@lineas: = +1;
```

## Asignar Variables

Ahora se pasará a la asignación de estas variables dentro de los campos que la contendrán dentro de la interfaz destino.

La variable `@cabecera` será asignada al campo `[CAB.id]` del registro de la interfaz destino, además de asignarla se debe modificar la propiedad “**Evaluar**” y seleccionar aquí el valor “Al crear registro” con lo cual se indica a ebiMap que cada vez que se cree el registro se escriba el contenido de la variable. Para el campo `[LIN.id]` del se realiza la misma asignación.

La variable `@líneas` se asigna al campo `[LIN/IDLIN]` de la interfaz destino, además de se debe modificar la propiedad “**Evaluar**” y seleccionar aquí la característica de “Al crear registro” con lo cual se indica a ebiMap que cada vez que se cree el registro se escriba el contenido de la variable. Una vez asignadas las variables se compila el mapa, se guarda y se compila.

De esta forma se puede relacionar tanto las cabeceras como las líneas, sabiendo que detalle pertenece a cada línea.

## 7 DISEÑO DE LISTAS DE VALORES.

### 7.1 INTRODUCCIÓN.

---

#### 7.1.1 ¿QUÉ SON?

Una lista de valores es una lista cuyos elementos están agrupados en pares de valores cuyo significado es equivalente e intercambiable en sistemas o códigos distintos. Por ejemplo el sistema informático de gestión de un emisor y el sistema del receptor de un mensaje.

Si en un envío de productos el artículo “X” tiene el código 001 pero en la empresa proveedora ese artículo tiene un número 999, se puede hacer una lista de conversión para que durante el mapeado, cuando se encuentre un valor 001 en un campo relacionado con la lista de conversión el valor 001 se convierta en el valor 999.

Las listas de equivalencias en ebiMap están compuestas por:

- **Valor1:** valor origen (35 caracteres de longitud por defecto).
- **Valor2:** valor equivalente (35 caracteres de longitud por defecto).
- **Descripción:** breve descripción de la equivalencia (40 caracteres de longitud por defecto).

### 7.2 GESTIÓN

---

#### 7.2.1 CREAR NUEVA LISTA DE EQUIVALENCIA.

En primera instancia, se debe definir la estructura de los campos que contendrá la lista de equivalencias.

- Desde el menú “**Archivo – Nuevo**”: opción **Nueva Lista**
- Desde la barra de herramientas: botón **Nueva lista**

A continuación se mostrará la ventana de diseño de “*Lista de equivalencia*”. Esta pantalla esta estructurada en forma de una tabla donde cada fila tiene (por defecto) tres columnas para cada definir una sola equivalencia.

Donde:

- “Valor1” es el primer valor de la equivalencia.
- “Valor2 es el segundo valor de la equivalencia.
- Descripción es una explicación breve de la equivalencia entre los dos valores.

El área de trabajo tiene un menú contextual con opciones que también están accesibles desde la barra de herramientas, estas opciones son:

Opción	Descripción
<b>Nuevo</b>	permite añadir una nueva fila a lista de equivalencias.
<b>Propiedades</b>	permite observar el nombre y tamaño de los campos que por defecto existen en la lista de equivalencia y los creados por el usuario. Dentro de esta ventana se puede modificar la estructura de una lista de valores.
<b>Eliminar</b>	permite borrar el campo que se encuentre seleccionado.

Para definir la estructura de la lista de equivalencias se debe antes que nada salvar la lista de equivalencia con un nombre

## 7.2.2 INTRODUCIR VALORES.

Para introducir valores en una lista de equivalencias sólo es necesario ubicarse en el casilla donde se desea introducir el valor y empezar a introducir la información deseada.

### ***Limitaciones Listas de Equivalencia***

Las limitaciones de las listas de equivalencia:

- N° máximo de filas: 16384
- N° máximo de campos: 128
- N° máximo de celdas (filas x campos): 65536

## 7.2.3 MODIFICAR ESTRUCTURA DE LISTA DE EQUIVALENCIA.

Para modificar la estructura que se ha definido previamente, Desde:

- Menú principal: opción **Propiedades**.(F4)
- Barra de herramientas: botón **Propiedades**.(F4)
- Menú contextual: opción **Propiedades**.(F4)

Así se accede a la ventana de “estructura de la lista” de equivalencia, Siempre que se accede a la ventana de propiedades aparece una advertencia sobre la posible pérdida de información. Se refiere a la reducción de el número máximo de caracteres o a la eliminación de una columna de valores.

En la ventana de “*Estructura de la lista*” están disponibles las siguientes opciones:

Opciones	Descripción
<b>Nuevo Campo</b>	Permite añadir una nueva columna para campos extra. Para cada nuevo campo se debe definir un nombre para la columna y un número máximo de caracteres.
<b>Propiedades</b>	Permite modificar el nombre o el número de caracteres máximo de una

<b>Campo</b>	columna ya creada. (Se puede perder información si se reduce el número de caracteres máximo de una columna que ya tiene valores en la lista)
<b>Eliminar Campo</b>	Permite eliminar una columna y todos los campos de la lista de valores. El resultado es la pérdida permanente de los valores almacenados en esta columna.

Cualquier modificación o eliminación de un campo que ya contengan información, esta se perderá definitivamente. Por ejemplo si de la lista de valores se elimina la columna “*Descripción*”, esta información ya no podrá ser recuperada.

En el caso de modificar algún campo, también puede haber consecuencias de alteración de información y de su contenido. Por ejemplo si la longitud del campo es 15 caracteres máximo en una lista llena de datos y se cambia la longitud del campo a 10 caracteres, los valores que superen la nueva longitud (entre 10 y 15) serán recortados a un máximo de 10.

## 7.2.4 IMPORTAR DATOS

La importación de datos hacia una lista de valores es auxiliada por un asistente que muestra los diferentes orígenes de esta importación, estos pueden ser:

- Un archivo plano.
- Una base de datos.

El asistente de importación de datos presenta las siguientes pantallas.

### ***Selección del origen de archivo.***

El asistente para importar una lista de equivalencias permite elegir el origen del que se importarán los datos.

- Un fichero de texto plano o una lista de equivalencias de la versión anterior.
- Del portapapeles.
- Una base de datos . Se necesita un alias.

Y la opción “vaciar la lista antes de importar”, que sustituirá la información actualmente almacenada en la lista de equivalencia. Botón “**Siguiente**” para continuar con el asistente.

### ***Visualización de Contenido.***

La visualización de contenido permite ver los datos que se van a importar. Botón “**Siguiente**” para continuar.

### ***Selección de Formato.***

La siguiente ventana pregunta que tipo de estructura tiene el archivo. El tipo de estructura es muy importante ya que indicará a ebiMap como colocar los valores correctamente en la nueva lista.

- **Ancho fijo:** cada campo tiene el ancho definido en la lista de equivalencia.
- **Con separador de campo:** un carácter especial separa cada campo. Este carácter especial se seleccionará a continuación.
- **Lista de equivalencia de la versión anterior:** la estructura ya esta establecida y en este caso solo se comprobará que es correcta. si no tiene la estructura adecuada de las listas de equivalencia de la versión anterior el asistente mostrará un aviso

### ***Elegir alias de base de datos***

En el caso de elegir como origen una base de datos, el asistente para la importación de ebiMap mostrará una ventana donde mediante un Alias que apunte a una base de datos se pueda realizar la importación. Botón **“Siguiente”** para continuar

### ***Indicar consulta.***

A continuación se muestra una ventana donde indicar la consulta (Query) para obtener los datos de la tabla. Pulsando el botón **“Probar”** se puede realizar una prueba para comprobar que se obtienen los datos de la tabla.

Una vez seleccionados los campos que se desea importar a la lista de valores, con el botón **“Siguiente”** se inicia el proceso de importación de los datos obtenidos de esta consulta y se rellenará la información de la lista de valores con los datos obtenidos.

## **7.2.5EXPORTAR DATOS.**

El ebiMap permite exportar los datos de una lista de equivalencia a un archivo.

Para la exportación de una lista de valores sólo hay elegir la opción de **“Exportar”** y ebiMap mostrará la ventana de exportación de lista de valores.

Dentro de esta ventana se puede elegir el archivo donde se desea exportar la lista de valores, así como la distribución de los campos en la exportación, ya sea de ancho fijo o con un separador de campo específico. Una vez indicadas las opciones se procederá a la exportación.

## **7.3 EJEMPLOS: TIPOS DE LISTAS DE VALORES.**

---

### **7.3.1LISTA DE VALORES PARA EQUIVALENCIA (LISTA DE EQUIVALENCIA)**

En este ejemplo se utilizará la información almacenada en una base de datos de Ediwin llamada **“EDIBD”** para importar los datos y crear con ellos una lista de valores. Esta lista de valores permitirá durante el mapeo transformar el código numérico del interlocutor en el nombre **“Literal”**.

- Se sabe que dentro de la Base de datos “*EDIBD*” la relación de interlocutores y su nombre se encuentra en la tabla “INTERLOCUTORES”, y corresponde con los campos “CODINTERLOCUTOR” y “NOMBRE”.
- Se sabe también el nombre de usuario y el contraseña para acceder a la Base de datos.

El procedimiento es el siguiente:

1. Se crea una nueva lista (ebiMap pedirá que se guarde antes de hacer la importación.)
2. Se guardará la lista con el nombre “*Lista\_Interlocutores.lis*” y a continuación se pulsa en el botón “**Aceptar**”.
3. El asistente de importación pedirá que se indique el origen de la importación, en este caso la opción de “**base de datos**”
4. Después pedirá el alias de esta base de datos a la que se accede para obtener los datos para la lista de valores, y el usuario y la contraseña.
5. Una vez indicados los datos de acceso, se selecciona la opción de importar la definición de campos y datos.

Una vez seleccionadas las opciones arriba mencionadas, se procede a construir la consulta en formato SQL a la base de datos. En este ejemplo sería:

```
Select CODINTERLOCUTOR,NOMBRE from INTERLOCUTORES
```

Esta consulta selecciona los campos CODINTERLOCUTORES y NOMBRE de la tabla Interlocutores que es donde están contenidos estos campos.

Es posible observar en la ventana “*Visualización*” que se han traído los datos que cumplen con la consulta y que se desea importar en la lista de valores, Si todo es correcto hacer clic en el botón “**Siguiente**”.

Dentro del asistente se muestra la información de los datos que han resultado de la consulta. Tras clic en el botón “**Siguiente**” se procede a la importación de los datos a nuestra lista de valores. Así se consigue una lista de valores que se puede utilizar en el procesos de mapeo.

### 7.3.2 LISTA DE VALORES PARA CONVERSIÓN (LISTA DE CONVERSIÓN)

En este ejemplo se creará una lista sencilla para convertir los códigos privados de un interlocutor origen para transformarlos en los códigos privados del interlocutor destino. Estos códigos no dependen del formato del mensaje, ya que son los propios interlocutores quienes los definen. Lo único que define el Formato del mensaje es el campo donde se puede recoger dicho dato.

- Se sabe que dentro del documento de origen Edifact, el campo que recoge el código de producto es el [G25PIA.7140]



- Se desea mapear el valor equivalente en el campo [LIN.NUMADI] (número adicional) de la interfaz destino.

Como primer paso se creará una lista pequeña para utilizarla dentro de el mapa, con sólo dos entradas para dos productos “Cereza” y “Manzana”. El proveedor envía los siguientes códigos:

- 519384 que equivale a “CEREZA FRUIT FIL. 18.14KG”.
- 525851 que equivale a “MANZANA FRUIT FIL. 16KG”

Los códigos de estos productos para el proveedor son distintos en su sistema de gestión:

- 001 equivale a CEREZA FRUIT FIL. 18.14KG
- 002 equivale a MANZANA FRUIT FIL 16KG

Por tanto la lista de conversión deseada será:

Valor 1 (origen)	Valor 2 (destino)	Descripción
519384	001	CEREZA FRUIT FIL. 18.14KG)
525851	002	MANZANA FRUIT FIL 16KG)

Dentro de “**Valor1**” se debe especificar los posibles valores que envíe el proveedor y en “**Valor2**” los valores que son equivalentes para el distribuidor y que es necesario escribir en el mapa.

Una vez creada y guardada la lista. Se especifica tanto la carpeta como el nombre del archivo con el que se desea guardar la lista. Por ejemplo *Productos.lis* dentro de la carpeta adecuada para el grupo de trabajo. Al guardar la lista aparecerá dentro de los componentes del grupo de trabajo.

Para que efectivamente la lista de equivalencia se tenga en cuenta a la hora de mapear la interfaz origen. Se debe definir en el mapeado que el campo [LIN.NUMADI] tenga como propiedad de conversión la lista de equivalencias *Productos.lis* para que ebiMap sustituya un valor por otro.

## 8 DISEÑO DE MAPAS (AVANZADO).

### 8.1 INTRODUCCIÓN.

---

En este Capítulo se muestra de forma sencilla como se diseña una mapa avanzado.

El diseño de un mapa avanzado comprende:

- La definición de variables del sistema.
- Evaluación de expresiones en los campos de los registros de la interfaz destino.
- Respuesta a la colisión en la asignación de valores a campos de los registros de la interfaz destino.
- Condiciones avanzadas en la creación de registros. Forzado de registros.

Además al final del capítulo se explicará un ejemplo de creación de un mapa avanzado

### 8.2 COLISIÓN DURANTE LA ASIGNACIÓN DE VALORES.

---

#### 8.2.1 ¿QUÉ ES UNA COLISIÓN?

Una colisión ocurre cuando a un campo de destino, durante un proceso de mapeado, se le quiere asignar un valor, y este campo ya contiene un valor asignado con anterioridad.

EbiMap posee una gestión de colisiones durante un mapa, esto es de gran ayuda ya que permite especificar el comportamiento del mapeado cuando llegue a ocurrir una acción de este tipo.

#### 8.2.2 FORMAS DE RESPONDER ANTE UNA COLISIÓN.

##### ***Creando un nuevo registro.***

En este caso, ante una colisión, se generará un nuevo registro destino de este tipo asignándole a este el valor culpable de la colisión.

##### ***Sobrescribiendo el valor antiguo.***

En este caso, ante una colisión, se sobrescribe el valor del campo actual en el registro que se esta generando.

**Asignando el nuevo valor a otro campo de la interfaz (cascada) según un determinado criterio.**

En este caso, ante una colisión, se genera una repetición del campo en el registro cuyo valor será el dato que produce la colisión. Los criterios se pueden establecer con ayuda del editor de comandos en la opción de “**Condición**” dentro de las propiedades del campo.

## 8.3 CONVERSIÓN DE VALORES.

---

### 8.3.1 LISTA CONVERSIÓN.

Dentro de un mapa en ebiMap se puede hacer uso de listas de conversión de valores (Lista de Valores), estas listas permitirán la transformación de un valor contenido en la interfaz origen a otro equivalente (mediante la lista) en la interfaz destino.

### 8.3.2 TIPO CONVERSIÓN.

El tipo de conversión que se realizará valiéndose de esta lista de conversión puede ser

- **Ninguno:** no se realizará ninguna conversión. Es el valor por defecto obtenido de la interfaz origen .
- **Normal:** realiza una conversión de “**Valor1**” de la lista de valores a “**Valor2**”.
- **Inversa:** realiza una conversión de “**Valor2**” de la lista de valores a “**Valor1**”.
- **Restricción:** sólo permite valores contenidos en la lista de valores.

## 8.4 PESTAÑA “PROPIEDADES INTERFAZ”

---

### 8.4.1 INTRODUCCIÓN

Algunas de las opciones de la Pestaña de “*Propiedades Mapeado*” se han explicado en capítulos anteriores. En este capítulo se continuará explicando el resto de campos de la pestaña que controlan los conceptos que se acaban de explicar como la respuesta ante una colisión de datos, La lista de conversión utilizada y el tipo de conversión la creación de registros forzada o según datos, etc.

La pestaña de “*Propiedades Interfaz*” muestra las propiedades tanto de la interfaz origen como de la interfaz destino, estas propiedades estas visualizadas en “sólo lectura”, si se desea modificar alguna se debe editar directamente la interfaz desde la ventana de diseño de “*Interfaces*”, ya que desde la ventana de diseño de “*Mapas*” no se pueden modificar .

## 8.4.2 PESTAÑA “PROPIEDADES INTERFAZ”

Como se puede observar, al igual que las propiedades del mapa, las propiedades de la interfaz están divididas en dos secciones, una que muestra las propiedades de la interfaz origen y otras de la interfaz destino.

En la ventana de diseño de “*Mapas*” se puede consultar esta información, pero no se puede editar. Para editar estos datos se debe abrir la interfaz y editar el archivo interfaz directamente desde una nueva ventana de diseño de “*Interfaces*”

### *Propiedades registro de la interfaz.*

- **Nombre:** nombre del registro.
- **Descripción:** descripción del registro.
- **Acceso a datos:** tipo de acceso a datos.
- **Maestro-Detalle:** relación maestro-detalle, para indicar claves ajenas.
- **Filtro:** permite indicar un filtro
- **Etiqueta:** la etiqueta no puede ser modificada, en el caso de una IEAD BD
- **Estado:** permite indicar para un registro si es condicional respecto de otro registro o mandatario por definición en la interfaz.
- **Cardinalidad Min:** número mínimo de veces que debe aparecer este registro
- **Cardinalidad Máxima:** número máximo de veces que puede aparecer este registro.
- **Admite valor:** permite indicar que es un registro que no admite valores, por ejemplo un registro que es en realidad un grupo de registros.
- **Es grupo:** permite indicar si se trata de un grupo de registros o no, los grupos de registro no tienen dentro campos, si no más registros.
- **Grupo Virtual:** permite indicar el parametro para grupos virtuales. Sequence, Choice, All, Group, Substitution y Deshabilitado.

### *Propiedades campo de la interfaz.*

- **Nombre:** nombre del campo.
- **Descripción:** descripción del campo.
- **Tipo de datos:** Alfanumérico o Numérico.
- **Tipo de campo:** Dato, Compuesto o Subcampo.
- **Long. Mínima:** longitud mínima de caracteres del campo.

- **Long. Maxima:** longitud máxima de caracteres del campo.
- **Decimales:** si admite decimales o sólo números enteros.
- **Pattern:** restricción al campo gastando una expresión regular.
- **Estado:** permite indicar para un campo si es Condicional respecto de otro registro o Mandatario por definición en la interfaz.
- **Valor fijo:** permite incluir un valor fijo para el campo.
- **Lista de valores:** permite indicar la lista de valores asociada a dicho campo

## 8.5 EJEMPLO: DISEÑO DE MAPA (AVANZADO)

---

Para este ejemplo se utilizará como punto de partida el mapa desarrollado en el capítulo anterior, Se utilizará para aplicar algunos de los puntos tratados en este capítulo del manual. Dentro de este ejemplo se tratará:

- Creación forzada de registros con condicionantes.
- Uso de listas de conversión.

### 8.5.1 CREACIÓN FORZADA DE REGISTROS.

En el mapa la condición de creación de los registros, se ha dejado como “Según datos” (ebiMap lo indica así por defecto), En este caso que se genere el registro en la interfaz destino depende de la condición de que exista algún dato en este registro en la interfaz origen.

En este ejemplo se cambiará esta propiedad de tal forma que se indique a ebiMap que la creación del nuevo registro sea condicionada a la existencia de cierta información. En la interfaz origen. Concretamente , al número del pedido ([UNH/BGM.1004]).

1. Se selecciona el campo [UNH/BGM.1004] de la interfaz origen y se arrastra hasta el registro [CAB] de la interfaz destino.
2. Después de arrastrar el campo, ebiMap mostrará una ventana con la condición de creación del registro [CAB] que en este caso debe ser que el campo 1004 del registro [UNH/BGM] sea diferente a vacío.
3. Después de esto en la propiedad “**TipoCreación**” del registro de destino, seleccionar la opción “Forzar por registro”.

En el caso que un documento no tenga el campo [UNH/BGM.1004], al intentar realizar el mapa este no se generará debido a la no existencia de este campo.

Esta condición se puede aplicar en mapas donde se necesite obligatoriamente el contenido de cierta información, con el fin de controlar y filtrar información incompleta.

## 8.5.2 USO DE LISTAS DE CONVERSIÓN.

Siguiendo con el ejemplo anterior se utilizará la lista de conversión “Productos” para sustituir los códigos de origen de identificación de cada producto por los códigos de destino.

- Se sabe que dentro del documento de origen EDIFACT, el campo que recoge el código de producto es el [G25PIA.7140]
- Se desea mapear el valor equivalente en el campo [LIN.NUMADI] (número adicional) de la interfaz destino.

La lista de conversión solo tendrá 2 entradas (de las muchas que puede tener) , en el caso del ejemplo la lista de conversión era:

Valor 1 (origen)	Valor 2 (destino)	Descripción
519384	001	CEREZA FRUIT FIL. 18.14KG)
525851	002	MANZANA FRUIT FIL 16KG)

Para que efectivamente la lista de equivalencia se tenga en cuenta a la hora de mapear la interfaz origen. Una vez definida hay que modificar las propiedades del campo [LIN.NUMADI] (Campo de destino donde se desea mapear el valor convertido)

Dentro del mapa en el campo [LIN.NUMADI] se debe aplicar la lista de equivalencias “MiLista.lis” para que ebiMap sustituya un valor por otro. El procedimiento será el siguiente:

- Seleccionar el campo destino donde aplicar la lista de equivalencias. (NUMADI) y hacer doble clic en la propiedad “Valor” para que muestre el editor de comandos.
- Dentro del editor de comandos en la sección de “Funciones” elegir la función:

```
Equiv(«param1», «param2», «param3»)
```

Esta función devuelve el valor equivalente a el valor definido en el «param1» utilizando para ello la lista de equivalencia definida en el «param2». El último parametro indica si la conversión es normal o inversa. Es decir si se trata de convertir el valor de la primera columna a la segunda o al revés. Para el ejemplo los parámetros serán:

```
Equiv([UNH/G25LIN/G25PIA.7140], "Productos", 0)
```

Donde [UNH/G25LIN/G25PIA.7140] indica a ebiMap que compruebe el valor de este campo dentro de la lista “Productos.lis”, y que realice una conversión normal. Es decir de la primera columna a la segunda, (Si se indicase el valor ‘1’ la transformación se haría a la inversa. Ahora se debe compilar, guardar y ejecutar de nuevo el mapa para revisar que la lista funciona correctamente.

## 9 REGLAS DE VALIDACIÓN.

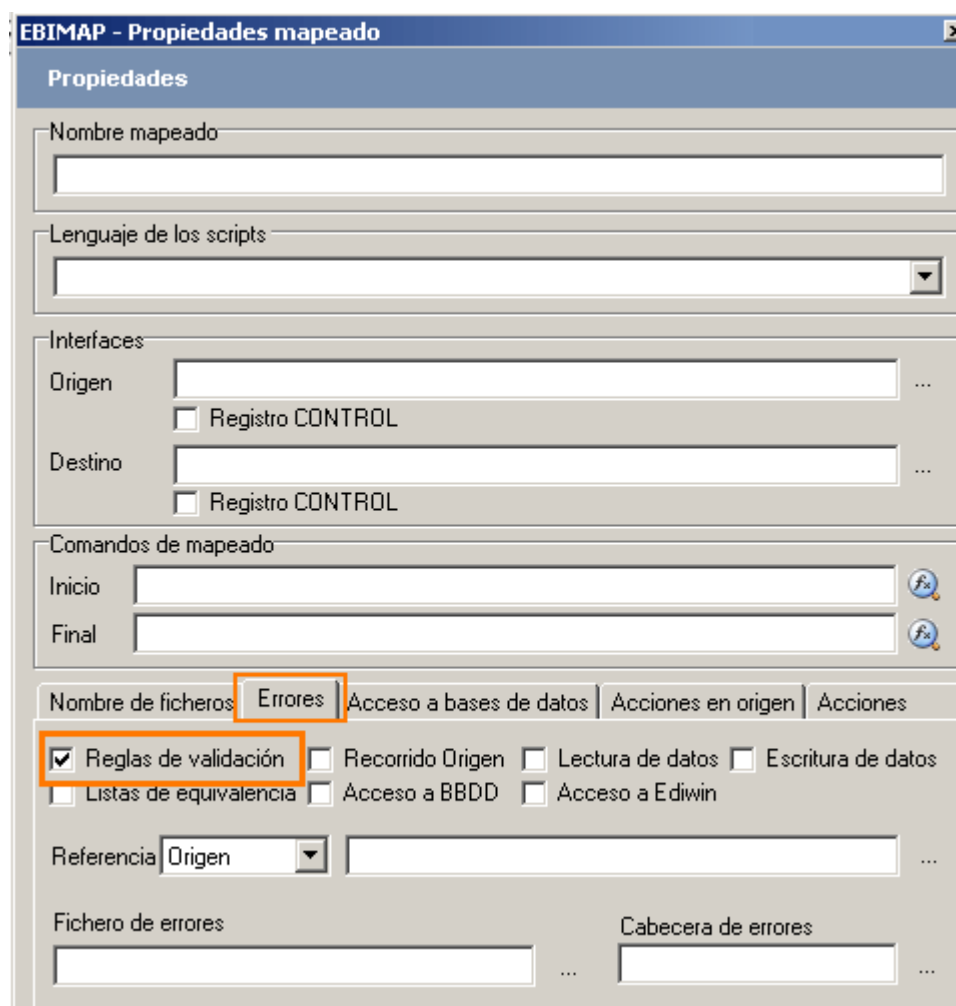
### 9.1 INTRODUCCIÓN

Las reglas de validación permiten la validación de los datos procedentes de la interfaz de datos origen durante la ejecución del mapa al que están asociadas.

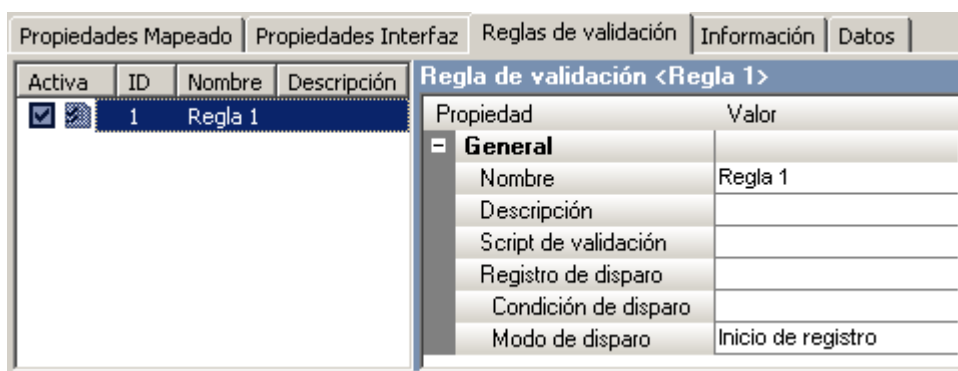
Durante el recorrido de los datos de la interfaz origen, ebiMap lee el registro de datos y si el registro lleva asociado alguna regla de validación, se realiza una comprobación de la condición de disparo de la regla.

Para realizar la configuración de las reglas de validación se debe realizar lo siguiente:

1. Acceder al diálogo de “*Propiedades del mapa.*” Seleccionar la pestaña “*Errores*” y marcar la opción “**Reglas de validación**”



2. Una vez marcado en la pantalla de edición del mapa se mostrará una nueva pestaña para indicar y editar las reglas de validación asociadas a la interfaz de datos origen. Con esta ventana se puede realizar la gestión de las reglas de validación (Creación, Edición, Eliminación, etc) de una forma más sencilla y visual.



### 9.1.1 PROPIEDADES REGLA DE VALIDACIÓN.

Las propiedades de las reglas de validación son:

- **Nombre:** indica el nombre de la regla de validación y permite identificar la regla del resto de reglas.
- **Descripción:** el texto de descripción del error/validación asociado a la regla de validación. Si se lanza una regla de validación durante la ejecución del mapa, el texto indicado en este campo será el que se muestre para notificar el error de validación que ha ocurrido.

**Importante:** se recomienda no utilizar los siguientes caracteres en la descripción o nombre de las reglas de validación, para evitar que Ediwin Viewer, la interfaz web del servidor EDIWIN, no interprete correctamente los errores recibidos.

Barra vertical (|)

Signo de equivalencia o tilde (~)

Punto y coma (;)

Signo de número o almohadilla (#)

- **Script de validación:** indica el script que se evaluará si se lanza la condición de disparo. No siempre es necesario indicar un script de validación.
- **Registro de disparo:** indica el registro al que va asociado la comprobación de la condición de disparo.
- **Condiciones de disparo:** indica la condición que se debe cumplir para el disparo de la regla de validación.
- **Modo de disparo:** indica cuando se ejecutará la regla de validación, existen cuatro opciones:
  - **Inicio de registro:** al inicio del registro de disparo.
  - **Fin de registro:** al final del registro de disparo.



- **Inicio grupo de registros:** al principio del grupo de registros.
- **Fin grupo de registros:** al final del grupo de registros.

## 9.2 OPCIONES

---

### 9.2.1 MENÚ CONTEXTUAL DE REGLAS DE VALIDACIÓN:

Para acceder a las opciones del menú contextual de gestión de las reglas de validación, se debe pulsar el botón derecho del ratón sobre la pestaña de edición de las reglas de validación:

Opción	Descripción.
<b>Nueva regla de validación</b>	Permite crear una nueva regla de validación
<b>Borrar la regla de validación</b>	Permite borrar la regla de validación seleccionada.
<b>Mover regla arriba</b>	Permite desplazar la regla de validación seleccionada una posición hacia arriba.
<b>Mover regla abajo</b>	Permite desplazar la regla de validación seleccionada una posición hacia abajo.
<b>Duplicar</b>	Permite duplicar la regla de validación seleccionada.
<b>Borrar todas las reglas</b>	Permite borrar todas las reglas de validación
<b>Desactivar todas las reglas</b>	Permite desactivar todas las reglas de validación.
<b>Reindexar reglas de validación</b>	Permite indexar (Ordenar las reglas de validación por el código que figura en el campo ID). Esta opción existe porque al mover reglas a diferentes posiciones el campo ID no se actualiza y las reglas se muestran desordenadas por el campo ID.

### 9.2.2 ACTIVAR/DESACTIVAR REGLAS DE VALIDACIÓN

Para activar o desactivar basta con marcar o desmarcar la casilla de la columna “Activa” asociada a las reglas de validación.

## 9.3 EJEMPLOS: REGLAS DE VALIDACIÓN.

---

### 9.3.1 COMPROBACIÓN DE FORMATO DE FECHA VÁLIDA.

A continuación se va a crear una regla de validación que compruebe que las fechas del archivo origen son de un formato válido.

#### *Creación de la regla*

Para crear una nueva regla de validación hay que seleccionar la opción “**nueva regla de validación**” del menú contextual que aparece con doble clic sobre el listado de la pestaña “Reglas de validación”

## Propiedades de la regla

Una vez creada la regla de validación, se indicarán las propiedades de la regla, para el ejemplo son las siguientes:

- **Nombre:** Regla 2\_Validación del formato de fecha.
- **Descripción:** “Formato de fecha incorrecto. Formato indicado: “[UNH/DTM.2379]& “Formato admitido: [101 = AAMMDD, 102 = CCAAMMDD o 203= CCAAMMDDHHMM]
- **Script de validación:** (No se usa en este ejemplo)
- **Registro de disparo:** una de las propiedades más importantes es el registro de disparo al que va asociado la comprobación de la condición de disparo. La forma de indicar el registro de disparo es arrastrar el registro desde la interfaz origen hasta la regla a la que se quiere asociar. Una vez arrastrado, se mostrará automáticamente en la propiedad “**Registro de disparo**”, el registro que se acaba de arrastrar.
- **Condición de disparo:** se usará como condición de disparo la expresión:  
`Y (NO (ESTAEN ([UNH/DTM.2379], "/101/102/203/")) ,DIST ([UNH/DTM.2379], "")) ;`

En este caso se comprueba que el calificador del formato de fecha (“[UNH/DTM.2379]”) “NO”“ESTA EN” la lista que se indica como segundo parámetro (101/102/203) “Y” que sea “distinto” de vacío.(El segundo parametro de la función DIST es “”). El error se disparará en este caso si existe un DTM distinto a vacío que no está en las listas especificadas. Y se creará el error de validación con el texto indicado en el campo descripción.

- **Modo de disparo:** el modo de disparo indica cuando debe ocurrir esta validación. En este caso se indica que la comprobación tendrá lugar al final del registro.

## Propiedades del registro de fecha

Se configuran en este caso las propiedades del mapeado sobre la interfaz origen. La propiedad que interesa configurar para este ejemplo es “**Disparar si vacío**” con el valor igual a “SI”. De esta forma se controla que si el registro que corresponde a la fecha no viene especificado en el fichero origen de datos, se lance la regla para indicar que el formato es erróneo porque el dato fecha no viene en el registro.

## 9.3.2 CONFIGURACIÓN DE DEPENDENCIA DE REGLAS

En este ejemplo se validará, si el punto operacional del comprador es correcto (Longitud igual a 13), y si es correcto, se valida a continuación el número del pedido de la factura (Longitud igual a 10). En el caso de que el punto operacional sea incorrecto no se ejecuta la validación del número de pedido.

*Nota: Se parte de una interfaz origen que corresponde con la guía EDIFACT EDI\_ORDERS\_D\_93*

### **Configuración de la regla de validación del punto operacional del comprador**

- **Nombre:** Regla\_1\_Comprobación punto operacional Comprador
- **Descripción:** “Punto operacional del comprador debe tener longitud = 13.”
- **Script de validación:** @NOVALPED:=1;(Como el punto operacional del comprador es erróneo, actualiza la variable @NOVALPED con el valor igual a 1. Esta variable sirve para saber más tarde si se deben lanzar las reglas de validación dependientes.)
- **Registro de disparo:** [UNH/G06TAX]. El registro de disparo debe ser un registro posterior tanto al registro desde donde se obtiene el punto operacional de comprador como al registro desde donde se obtiene el número de pedido de la factura. De esta forma como el registro desde donde se obtiene el número de pedido es anterior al registro desde donde se obtiene el punto operacional cuando se va a lanzar la regla se conoce si el punto operacional es correcto o incorrecto.
- **Condiciones de disparo:** DIST(@BYOK,"1") (@BYOK, es una variable que se define al configurar las propiedades del registro, ver a continuación)
- **Modo de disparo:** “Fin de registro”

### **Configurar propiedades del registro que hace referencia al punto operacional del comprador.**

- **Script de inicio:** la expresión en el inicio va a ser una expresión compleja. En un primer paso se debe obtener el punto operacional del comprador y después si el código de comprador tiene longitud 13.

```
DARVALORCOND (@BY, [G02NAD.3039], IGUAL ([G02NAD.3035], "BY")) ;
@BYOK = IGUAL (LONG (@BY), 13)
```

- **Disparar si vacío:** “NO”

### **Configuración de la regla de validación del número de pedido**

- **Nombre:** regla\_1\_Comprobación del número del pedido
- **Descripción:** “Número de pedido debe tener longitud = 10 ”
- **Script de validación:** (Ninguno)
- **Registro de disparo:** [UNH/G06TAX]
- **Condiciones de disparo:** Y (Y (DIST (@NUMPED, "" ), DistNum (Long (@NUMPED ), 10)) , NO (@NOVALPED ))

Se deben dar tres casos para la ejecución de la regla:

- Número de pedido @NUMPED distinto de vacío:

- Número de pedido @NUMPED cuya longitud es distinta de 10:
- Variable @NOVALPED igual a 0.

Si el punto operacional del comprador no es válido, como la variable @NOVALPED será igual a “1”, el resultado de la condición de disparo será 0 y no se ejecutará esta regla. De esta forma se consigue que la descripción asociada a esta regla no se indique si el punto operacional del comprador no es válido.

- **Modo de disparo:** “Fin de grupo de registros”

### ***Configurar propiedades del registro que hace referencia al número de pedido***

- **Script de inicio:**

```
DarValorCond(@NUMPED, [G01RFF.1154], Igual ([G01RFF.1153], "ON")) ;
```

Que se traduce como obtener el número de pedido de la factura y almacenarlo en la variable @NUMPED

La regla de validación asociada al Punto operacional del comprador tiene indicada en la propiedad Modo de disparo: “Fin de registro”. En cambio la regla asociada a la validación del Número de pedido, tiene indicada en la propiedad “**modo de disparo**” en “Fin de grupo de registros”.

Con esto se consigue que teniendo ambas reglas el mismo registro de disparo ([UNH/G06TAX]) se lance primero la regla asociada a “Fin de registro” y al final la regla asociada a “Fin de grupo de registros”.

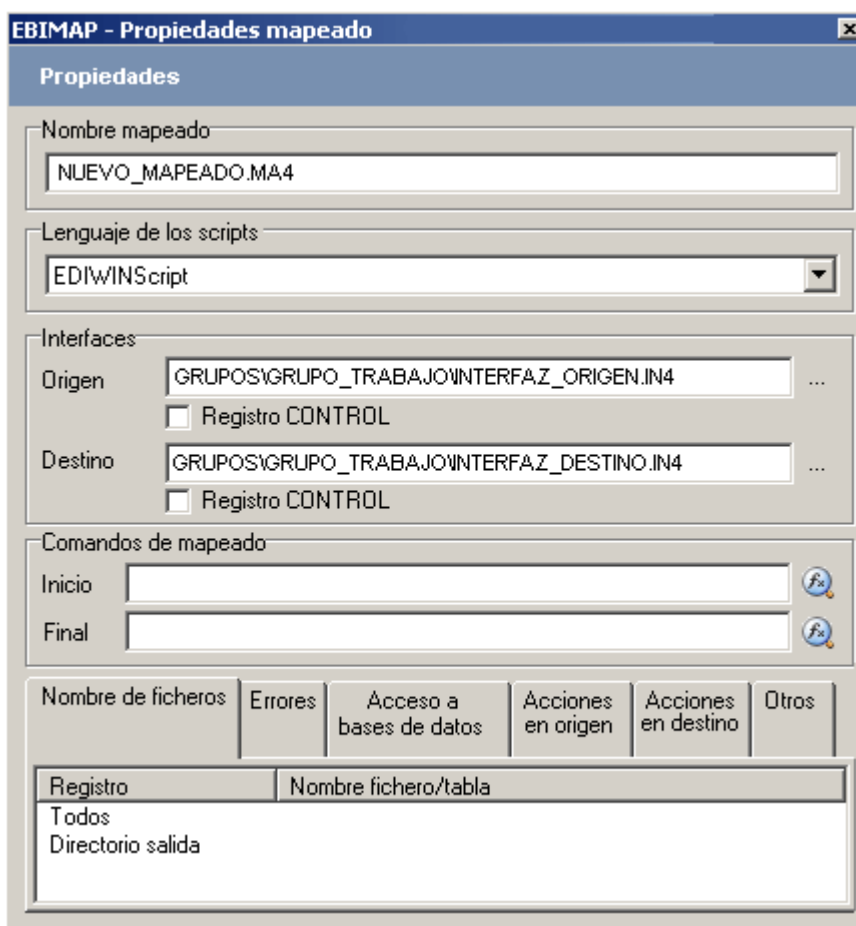
## 10 PROPIEDADES DEL MAPA. (AVANZADO)

### 10.1 INTRODUCCIÓN.

En el panel de propiedades de mapeado se pueden configurar algunas de las opciones avanzadas de mapeado. Para acceder a este diálogo:

- Menú Principal “**Archivo**”: opción **Propiedades**.
- Barra de Herramientas principal: botón **Propiedades**
- Menú desplegable de la ventana de edición: opción **Propiedades**.

Se accede de este modo al diálogo de propiedades.



En otros capítulos anteriores se han hablado de las Áreas de “Nombre Mapeado”, “Interfaces” y “Comandos de mapeado”. En este capítulo se hablará de las pestañas localizadas en la parte inferior, donde se configuran algunas de las opciones avanzadas de mapeado.

#### 10.1.1 PESTAÑA “NOMBRE DE FICHEROS”

Existen dos opciones:

### **Todos.**

Permite especificar mediante una función de script, variable, etc, el nombre del fichero destino generado.

Por ejemplo, en este script donde el nombre del fichero destino generado se crea a partir de la concatenación del punto operacional del vendedor y “Factura.edi”:

```
[CABECERA.VENDEDOR]&"Factura.edi"
```

### **Directorio salida.**

Permite especificar mediante una función de script, variable, etc, el directorio de salida de los ficheros destino generados.

Por ejemplo, en este script donde el directorio destino del fichero generado, depende del tipo de pedido generado. De esta forma los ficheros generados a partir del fichero origen se depositan en directorios destino en función del tipo de pedido.

```
"C:\PEDIDOS\"&[CABECERA.TIPO_PEDIDO]
```

*Nota - Realizando doble clic sobre cada opción, se accede a la pantalla de edición de comandos de Ediwin Script.*

## **10.1.2 PESTAÑA “ERRORES”.**

Permite seleccionar los errores de la ejecución del mapa correspondientes a reglas de validación, lectura de datos, escritura de datos, listas de equivalencia, etc según las opciones de error indicadas. Esta información aparecerá en la pestaña “Información” en la parte inferior de la ventana de edición.

*Nota-Si estas opciones son marcadas, cuando se ejecute el mapeado y se importe a Ediwin si ha habido errores de este tipo, el documento se quedará erróneo y en la opción Contenido del mensaje se visualizarán sólo los tipos de errores que se hubieran marcado durante la compilación, además de los errores sintácticos si los hay, por no cumplir las restricciones de interfaz de datos destino.*

### **Opciones de configuración**

Los errores que se pueden filtrar a un fichero son los siguientes:

- **Reglas de validación:** indica los errores que hacen referencia a las reglas de validación. Código de error en el fichero es «RVA»
- **Recorrido en origen:** indica los errores que hacen referencia al recorrido de datos del fichero origen. Código de error en el fichero es «REC»
- **Lectura de datos:** indica los errores que hacen referencia a la lectura de datos del fichero origen. Código de error en el fichero es «IEAD.L»
- **Escritura de datos:** indica los errores que hacen referencia a la escritura de datos. Código de error en el fichero es «IEAD.E»

- **Listas de equivalencia:** indica los errores que hacen referencia al acceso a listas de equivalencia. Código de error en el fichero es «EQV»
- **Acceso a BBDD:** indica los errores que hacen referencia al acceso a base de datos o el establecimiento de conexiones. Código de error en el fichero puede ser «LKU» (Si se trata de un acceso a datos) o «CNX» (Si se trata de un establecimiento de conexión)
- **Acceso a Ediwin:** indica los errores que hacen referencia al fallo de funciones que acceden a datos contenidos en la estación Ediwin. Código de error en el fichero es «AEW»

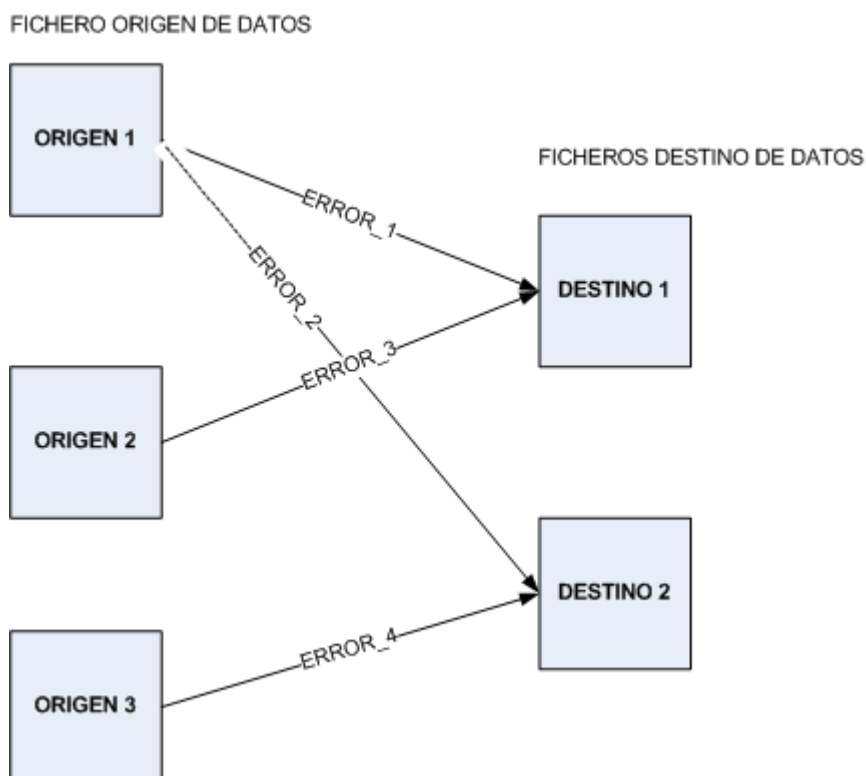
### **Ejemplo:**


```
MAPEADO;ERRORES DE MAPA FACTURA30/09/2004 17:32
DOCUMENTO;DEST;1; Mensaje: 2108014 ; C: \TMP \INVOICD93A.EDI
ERR;1;«IEAD.L»;Registro: CABECERA. Campo: NODO. Sobrepasada la
ERR;2;«EQV»;[D.G02NAD.3036.2] No existe la lista de equivalencia
«Lista».. ERR;3;«RVA»;0;Comprobar Vendedor:...
```

### **Opciones del fichero de error:**

- **Referencia:** puede ser origen o destino.
- **Origen:** indica que realizará la agrupación de errores por documento origen que los provoca.
- **Destino:** indica que realizará la agrupación de errores por documento destino donde se provocan.

Los errores aparecerán agrupados en el resultado de la compilación según el documento (Origen o Destino) al que hacen referencia.



Pulsando el botón  se puede acceder al editor de scripts, para establecer la etiqueta que indicará el documento al que hacen referencia los errores en base a una expresión.

Por ejemplo, se indica que la referencia de agrupamiento de errores sea Destino y en el script se concatene a la cadena “Mensaje” el número de la factura que se indica en el fichero de datos origen que es [CABECERA.NUMFAC]. Para ello se debe indicar:

- **Referencia:** “Destino” (código `DEST` )
- **Script:** " Mensaje: "&[CABECERA.NUMFAC]

Mostraría lo siguiente en el fichero de error:

### Ejemplo:

```
MAPEADO;ERRORES DE MAPA FACTURA 30/09/2004 17:32
DOCUMENTO;DEST;1; Mensaje: 2108014 ; C: \TMP \INVOICD93A.EDI
ERR;1;«IEAD.L»;Registro: CABECERA. Campo: NODO. Sobrepasada ...
ERR;2;«EQV»;[D.G02NAD.3036.2] No existe lista de equivalencia...»
ERR;3;«RVA»;0;Comprobar Vendedor...
DOCUMENTO;DEST;2; Mensaje: 2108013 ;C: \TMP\INVOICD93A.EDI
ERR;1;«EQV»;[D.G02NAD.3036.2] No existe la lista de...
```

*Nota - Como la referencia es Destino, en el fichero de errores se especifica con el código DEST.*



En este ejemplo se indica que la referencia de agrupamiento de errores sea Origen y en el script se concatene a la cadena "Mensaje" el número de la factura que se indica en el fichero de datos origen que es [CABECERA.NUMFAC] y además la fecha del sistema con el formato "dd/mm/yyyy hh:nn" con la cadena de texto " -- Fecha : " delante. Para ello se debe indicar:

- Referencia: "Origen" (código `DEST` )
- Script: "Mensaje:"&[CABECERA.NUMFAC]&"--Fecha:"&Fecha("dd/mm/yyyy hh:nn"&"")

Mostraría lo siguiente en el fichero de error:

### **Ejemplo:**

```
MAPEADO;ERRORES DE MAPA FACTURA30/09/2004 17:54
DOCUMENTO;ORIG;1; Mensaje: 2107939 -- Fecha: 30/09/2004 17:54;
C:\ TMP\INVOICD93A.EDI
ERR;1;«IEAD.L»;Registro: CABECERA. Campo: NODO. Sobrepasada 1
ERR;2;«EQV»; [D.G02NAD.3036.2] No existe la lista equiv. «Lista»
ERR;3;«RVA»;0;Comprobar Vendedor:
DOCUMENTO;ORIG;2; Mensaje: 2108014 -- Fecha: 30/09/2004 17:54 ;
C:\ TMP\INVOICD93A.EDI
ERR;1;«EQV»; [D.G02NAD.3036.2] No existe la lista de equivalencia
«Lista» <1382>
```

*Nota - Como la referencia es Origen, en el fichero de errores se especifica con el código ORIG.  
Importante - En el fichero del log de errores también se indica el directorio temporal y el fichero destino durante la ejecución del mapa.*

### **Ejemplo:**

```
MAPEADO;ERRORES DE MAPA FACTURA30/09/2004 17:54
DOCUMENTO;ORIG;1; Mensaje: 2107939 -- Fecha: 30/09/2004 17:54 ;
C:\ TMP\INVOICD93A.EDI
```

### **Fichero de errores:**

En este campo se indica el fichero en el que se van a escribir los errores durante la ejecución del mapa que se ha configurado. Permite guardar así el resumen de los errores.

Pulsando el botón adjunto al campo (...) se pueden indicar expresiones Ediwin Script, para establecer la ruta y el fichero donde se escribirán los errores de la ejecución del mapa.

Ejemplo: Se puede indicar mediante un script que el nombre del fichero se cree concatenando el número de factura y la fecha del sistema, y de esta forma conocer a que fichero origen hace referencia y fecha/hora del error.

Por ejemplo Script:

```
"C:\TEMP\Log_" & [CABECERA.NUMFAC] & "--" & Fecha ("dd/mm/yyyy hh:nn") & ".log"
```

### Cabecera de errores:

Permite indicar una cabecera o título, donde se inician los errores y logs de cada mapa ejecutado.

Pulsando el botón adjunto al campo (...) se pueden indicar expresiones Ediwin Script, para establecer la cabecera/título de logs o errores de la ejecución del mapa.

*Nota - Si no se indica ningún título o cabecera se mostrará por defecto el nombre del mapa ejecutado concatenado a la fecha del sistema con el formato "mmddyyyy hhmmss"*

Ejemplo: En el script se indica que a la cadena "ERRORES DE MAPA FACTURA" se le concatene la fecha del sistema (función `Fecha()` con el formato indicado en el parámetro) para formar la cabecera.

```
"ERRORES DE MAPA FACTURA"&Fecha ("dd/mm/yyyy hh:nn");
```

Mostraría lo siguiente en el fichero de error:

```
MAPEADO;ERRORES DE MAPA FACTURA 30/09/2004 17:54
DOCUMENTO;ORIG;1; Mensaje: 2107939 -- Fecha: 30/09/2004 17:54 ;
ERR;1;«IEAD.L»;Registro: CABECERA. Campo: NODO. Sobrepasada 1
```

*Importante - No confundir el chequeo de estas opciones, con los errores mostrados en la pestaña información del mapa, cuando se lanza un mapa con el ebiMap interactivo.*

*Nota - La única opción de la pestaña Errores, de propiedades del mapeado que si se desactiva, no se muestra ni en la pestaña Información durante la ejecución de un mapa con el ebiMap Interactivo ni en el fichero de errores especificado, son los errores de Reglas de validación.*

## 10.1.3 PESTAÑA "ACCESO A BASE DE DATOS"

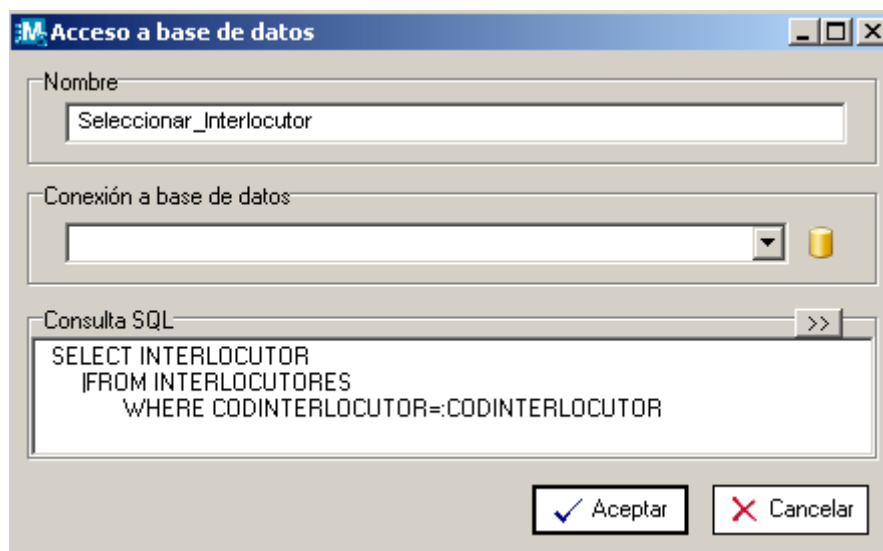
Permite crear, editar y eliminar consultas de acceso a base de datos. Estas consultas una vez creadas son accesibles desde las funciones de scripts correspondientes.

El menú contextual de esta pestaña aparece al hacer clic en el botón derecho del ratón. Las opciones de este menú son.

Opción	Descripción.
<b>Añadir</b>	Permite añadir una nueva consulta
<b>Editar</b>	Permite editar o consultar la configuración de una consulta seleccionada.
<b>Eliminar</b>	Permite eliminar una consulta.

### Ejemplo de creación de una nueva consulta.

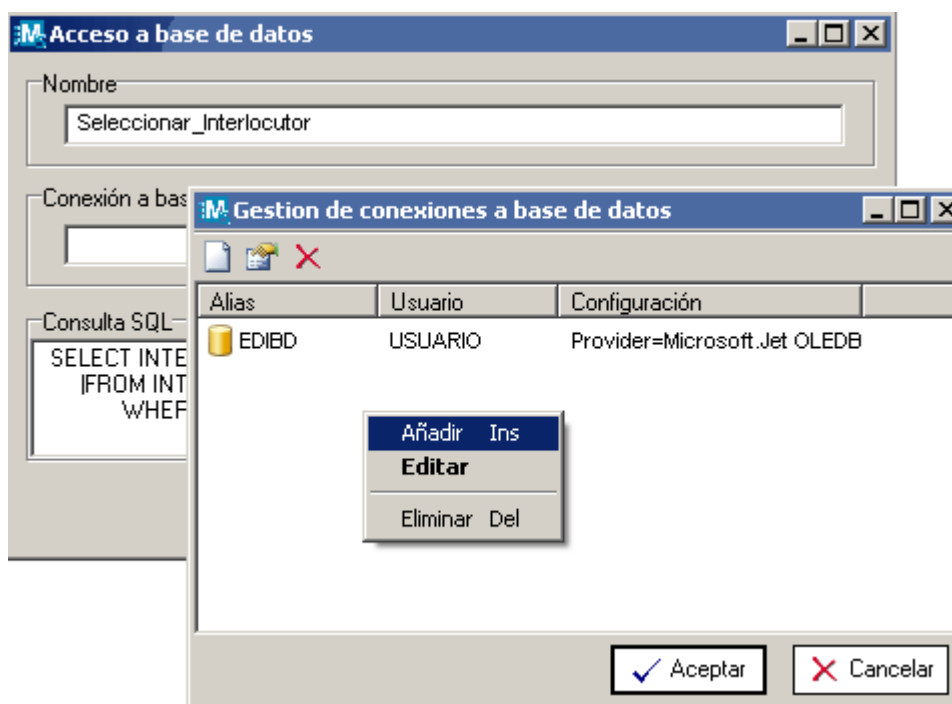
Pulsar el botón derecho del ratón sobre la pestaña "Acceso de base de datos" e indicar la opción "Añadir". Una vez pulsada, se muestra una pantalla como la siguiente donde indicar los datos:



- **Nombre:** nombre de la consulta de la base de datos
- **Conexión a base de datos:** alias de conexión a la base de datos. Un botón adjunto muestra el gestor de conexiones a bases de datos (Alias) donde se puede seleccionar la base de datos o añadir una nueva conexión.
- **Consulta SQL:** permite escribir la consulta SQL realizar. En este ejemplo se ha añadido un parámetro.(CODINTERLOCUTOR)

### ***Gestor de conexiones a bases de datos***

El gestor de conexiones a bases de datos permite seleccionar de una lista las bases de datos configuradas.



Si se quiere añadir una nueva base de datos se debe hacer desde el Menú contextual del gestor. El menú contextual del gestor tiene las siguientes opciones.

Opción	Descripción.
<b>Añadir</b>	Permite añadir una nueva conexión
<b>Editar</b>	Permite editar o consultar las propiedades de la conexión seleccionada.
<b>Eliminar</b>	Permite eliminar una conexión de la lista.

Tras seleccionar la opción “**Añadir**” se accede a la pantalla de edición de los datos de conexión. Estos datos cambian según la base de datos:

- **Nombre de alias:** Alias de la base de datos.
- **Usuario:** Usuario para conectar con la base de datos.
- **Password (Contraseña):** Contraseña para acceder a la base de datos.
- **Tipo de alias:** Tipo de la base de datos. Las opciones soportadas son: Access SQL Server, Oracle, DB2, EXCEL, Ado EBI Adapter.

*Nota: El tipo de alias ADO permite crear cadenas de conexión a cualquier base de datos.*

*Nota: Según el tipo de alias seleccionado se deben indicar una serie de datos como: servidor, nombre de base de datos, servicio, etc.*

### 10.1.4 PESTAÑA “ACCIONES EN ORIGEN”

Esta pestaña permite indicar el tipo de acción a realizar sobre el fichero de datos origen al ejecutarse el mapeado.

Las opciones que se pueden indicar son:

- **Ninguna:** no realiza ninguna acción sobre el fichero origen de datos.
- **Eliminar Datos Origen:** elimina los ficheros origen de datos al finalizar el mapeado. Si el fichero origen de datos es una base de datos realizará un borrado de cada registro de cada una de las tablas de las que haya leído los datos.
- **Renombrar/Marcar:** renombra el fichero de origen de datos al finalizar el mapeado. Con el botón que se encuentra junto al campo se accede a la pantalla de edición de script para indicar funciones de script a realizar sobre el fichero origen de datos.

Un ejemplo de script para renombrar el fichero origen sería el siguiente:

```
@numero&"*.BAK"
```

En este ejemplo, a cada uno de los ficheros origen se le concatenaría como prefijo el número de factura en el nombre y como sufijo la extensión *.BAK*. El \* que figura en el ejemplo sustituye al nombre del fichero origen a renombrar.

### 10.1.5 PESTAÑA “ACCIONES EN DESTINO”

La pestaña “*Acciones en destino*” presenta actualmente una sola opción. “**Permitir acciones en destino**”. aunque el documento de origen este vacío.

### 10.1.6 PESTAÑA “OTROS”

La pestaña “*Otros*” presenta actualmente una sola opción. “**Compilar en modo de ejecución**”, Inhabilitando esta casilla permite evitar el compilar cada vez que se guarda.

## 11 MAPAS CON ACCESO A BASE DE DATOS

### 11.1 INTRODUCCIÓN

---

En este Capítulo se muestra de forma sencilla como se diseña una interfaz de datos con acceso a una base de datos (IEAD de tipo BD) y un mapa con acceso a base de datos a través de la interfaz creada.

Hasta ahora el origen de los datos ha sido , en los ejemplos, un archivo enviado. En este ejemplo el origen de los datos será una base de datos. Del que se deberá mapear ciertos valores almacenados en sus tablas y campos.

### 11.2 EJEMPLO: INTERFAZ DE ACCESO A BASE DE DATOS.

---

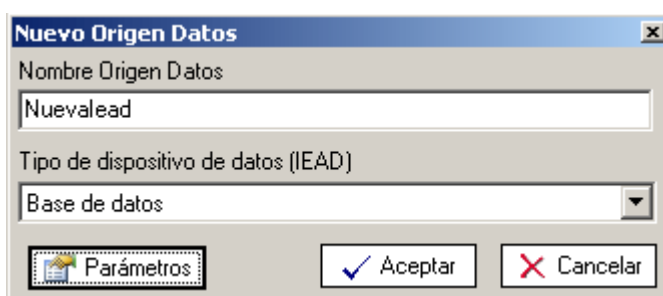
Para crear una interfaz de acceso a base de datos se debe proceder en gran medida como al crear una interfaz de acceso a cualquier otro tipo de IEAD.

Un IEAD es lo que permite acceder a ebiMap a los datos correctamente. Y debe coincidir con el formato en que el origen de datos estructura la información. En este ejemplo el IEAD BD debe ser configurado adecuadamente.

#### 11.2.1 CREAR UNA NUEVA INTERFAZ Y CONFIGURARLA IEAD BD..

El Primer paso es crear la Nueva Interfaz con la opción “**Nueva Intefaz**” del menú principal “**Archivo**”

Una vez en la ventana de edición de “*Interfaces*” acceder al diálogo de “*propiedades*” de la nueva interfaz y añadir en el área “*Dispositivos de acceso a Datos – IEADs*” una nueva IEAD de tipo “Base de datos”.



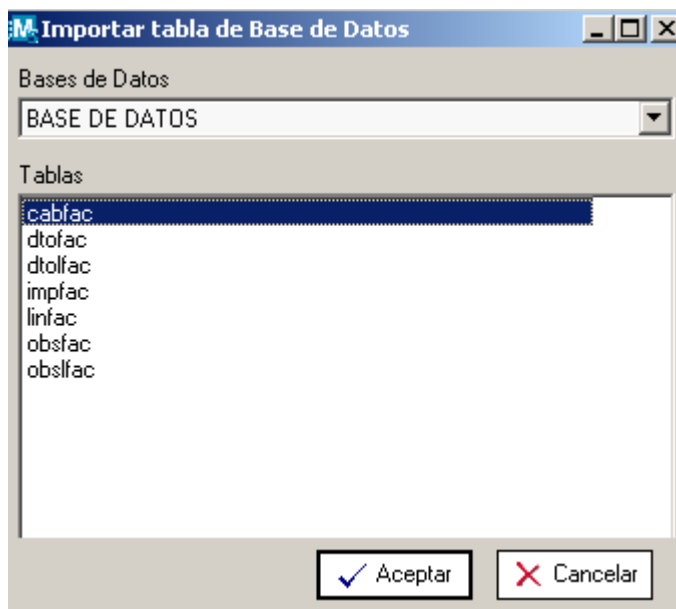
El botón “**Parámetros**” permite indicar los datos de conexión a la base de datos. “Alias”, el tipo de base de datos que puede ser (“Access”, “Oracle”, “Sql Server”, “DB2”), el tipo de conexión (ADO). Y opciones de lectura/escritura.

#### 11.2.2 IMPORTAR REGISTROS

Dentro de nuevo de la ventana de edición de Interfaces , situar el puntero del ratón, sobre el área del árbol de registros y seleccionar la opción para importar registro

- Menú contextual: opción **Importar Registro**.

A continuación se muestra una pantalla donde seleccionar la base de datos y la tabla desde donde se van a importar los campos del registro. En este ejemplo se seleccionará la tabla *CABFAC*.



Una vez seleccionada la tabla *CABFAC*, se mostrará un nuevo registro con los campos correspondientes a dicha tabla ya creados dentro del registro.

Es importante reseñar que el nombre literal de la tabla ("*CABFAC*") deberá estar en la propiedad "**Etiqueta**" del registro. Por defecto cuando se importa el registro, las propiedades "**Nombre**", "**Descripción**" y "**Etiqueta**" del registro se actualizan automáticamente tomándolos de la base de datos. En la interfaz se puede cambiar "**Nombre**" y "**Descripción**", pero no "**Etiqueta**", sino la interfaz no funcionaría correctamente, puesto que la propiedad "**Etiqueta**" es la utilizada internamente por el ebiMap para crear la consulta de acceso a la BBDD

### ***Tablas Relacionadas.***

Si se quiere importar otra tabla que no sea *CABFAC* pero que contiene una clave ajena al registro anterior, se debe seleccionar el registro [CABFAC] y a continuación seleccionar la opción "**Importar registro Hijo**" para importar dicha tabla como un registro hijo de [CABFAC].

Por ejemplo la tabla *LINFAC*. De la base de datos Contiene el campo NUMFAC, que actúa como una clave externa para unir algunas de las entradas de *LINFAC* con determinadas entradas de la tabla *CABFAC*, este campo actúa como clave externa.

El último paso consiste en crear la asociación Maestro-Detalle entre las tablas. Seleccionado el registro [LINFAC] he indicando en la propiedad **“Maestro-Detalle”** la asociación entre los campos.

```
NUMFAC=[CABFAC.NUMFAC ]
```

De este modo el campo [LINFAC.NUMFAC] queda relacionado con el campo [CABFAC.NUMFAC]

## 11.3 EJEMPLO DE MAPA CON ACCESO A BASE DE DATOS

Mapear teniendo como interfaz origen/destino una base de datos no supone una gran diferencia. Pero se debe tener en cuenta como gestionar los registros de las tablas de la base de datos desde donde se obtiene la información. Para que en una próxima ejecución del mapa no se vuelvan a mapear los registros ya mapeados anteriormente.

Para realizar esta gestión de los datos de la IEAD BD se pueden realizar dos tipos de acciones

- Actualizar un campo de control por ejemplo marcando un campo de la tabla como “Leído” o “No Leído”.
- Eliminar los registros leídos de la tabla de la BD, opción que se realiza desde la configuración de propiedades del mapeado. (pestaña “*Acciones en Origen*”)

### 11.3.1 ACTUALIZAR LOS REGISTROS LEÍDOS

**Paso 1** -Para crear un nuevo campo en un registro de la interfaz con la IEAD BD se debe editar esta interfaz previamente al mapeado. Desde la el área de campos. seleccionando el registro y creando un nuevo campo, en este caso “MARCADO”

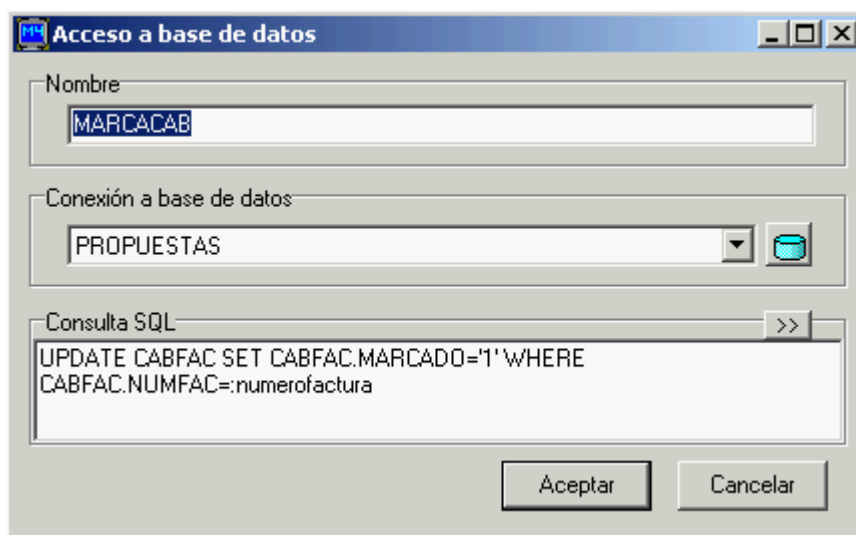
**Paso 2** -De nuevo en la ventana de edición del mapeado, en la propiedad **“Filtro”** de cada registro debe indicar por ejemplo `MARCADO<>“1”` con el fin de leer sólo aquellos registros no leídos. Esto se hace sabiendo que el propio mapa marcará los registros leídos de las tablas de la base de datos indicando `MARCADO:=1.` con una consulta de actualización que se explicará a continuación.

**Paso 3** -Para crear la consulta de actualización, se debe acceder al panel de “*propiedades del mapeado*”, Pestaña “*Acceso a base de datos*” y crear una consulta SQL de actualización

```
Update CABFAC Set CABFAC.MARCADO='1' Where NUMFAC ='númerofactura'
```

**Paso 4** -A esta consulta se le dará el nombre de `MARCACAB` para poder utilizarla en el mapeado. En esta consulta de actualización denominada `MARCACAB`, se actualiza el registro de la tabla `CABFAC` donde el campo `NUMFAC` sea igual al parámetro `númerofactura`.





**Paso 5** -Al parámetro `numerofactura` se le asociará el número de factura que se está leyendo en cada instante del proceso de mapeado. Para ello se usará una expresión en la propiedad de “**Script de Inicio**” del Registro `[CABFAC]`.

En el parámetro “**Script de inicio**” y en “**Script de final**” del registro principal, `[CABFAC]`, se indicarán las siguientes funciones para abrir la consulta de actualización (El Lookup `MARCACAB`) y para ejecutar la consulta y actualizar el campo `MARCADO` de las facturas leídas.

Para el parámetro del registro `[CABFAC]` en el parámetro “**Script de inicio**” se abren las consultas de actualización asociadas a cada tabla de la base de datos del IEAD BD. Con la función Ediwin Script de base de datos `ParamSQL()`:

```
ParamSQL("NombreLookup","Parametro","Valor","Tipo","Cond")
```

Donde:

- `NombreLookup` es el nombre de la consulta definida en las propiedades del mapa que se debe cumplir cuando parametro y valor cumplan la condición definida al final de la función.
- `Parametro` en este caso será “`numerofactura`”
- `Valor` es el campo a comparar, en este caso `[CABFAC.NUMFAC]`,
- `Tipo` es el tipo de datos del campo, N para numérico, X para alfanumérico.
- `Condición` indica si se debe cumplir o no 1 la comparación entre Parametro y Valor.

En este caso las funciones serían:

```
ParamSQL("MARCACAB","numerofactura",[CABFAC.NUMFAC],"X",1);
ParamSQL("MARCAOBS","numerofactura",[CABFAC.NUMFAC],"X",1);
ParamSQL("MARCADTO","numerofactura",[CABFAC.NUMFAC],"X",1);
```

```
ParamSQL("MARCAIMP","numerofactura",[CABFAC.NUMFAC],"X",1);
ParamSQL("MARCALIN","numerofactura",[CABFAC.NUMFAC],"X",1);
ParamSQL("MARCAOBSL","numerofactura",[CABFAC.NUMFAC],"X",1);
ParamSQL("MARCADTOL","numerofactura",[CABFAC.NUMFAC],"X",1);
```

Para el Script de final se ejecutará la consulta de actualización parametrizadas, que se han abierto anteriormente en el script de inicio. Con la función de Ediwin Script para Base de datos **AbrirLookup**("NombreLookup","Condicion") que ejecuta la consulta parametrizada si se cumple la condición "Condicion"), en el caso del ejemplo:

```
AbrirLookup("MARCACAB",1);
```

### 11.3.2 ELIMINAR LOS REGISTROS LEÍDOS.

Para eliminar los registros leídos de cada una de las tablas que se han recorrido basta con marcar la opción "**Eliminar Datos Origen**" dentro de la pestaña "*Acciones en origen*" de la ventana de "*propiedades del mapeado*" del mapa. A la cual se accede con el botón "**Propiedades**" del menú principal o la barra de herramientas.

## 12 DISEÑO DE MAPAS MÚLTIPLES

### 12.1 INTRODUCCIÓN

---

En este Capítulo se mostrará como crear un multimapeado (Mapa múltiple). Sus utilidades y las precauciones que se deben tener en cuenta.

El Mapeado múltiple o Multimapeado permite a partir del mismo fichero de datos origen la posibilidad de obtener varios ficheros de datos destino de tipo distinto (distintos IEAD) dependiendo de alguna condición, por ejemplo en base a un campo del fichero origen de datos.

Por ejemplo: Según el cliente al que se le envía una factura, se debe generar un fichero destino asociado a una determinada guía EDIFACT.

Así, si el cliente “Company A” utiliza la guía de la factura *EDI\_INVOIC\_D\_93* y el cliente “Company B” utiliza la guía de la factura *EDI\_INVOIC\_D\_96*, con la ayuda de un mapa múltiple es posible a partir de la misma interfaz origen de datos generar en función del campo cliente, un fichero destino u otro que se adapte a la guía adecuada.

#### 12.1.1 PRECAUCIONES A LA HORA DE CREAR MAPAS MÚLTIPLES

##### ***Condiciones no excluyentes.***

Es posible también que se quiera crear el fichero de datos destino de ambos mapas porque exista un cliente que quisiera que se le envíe la información en dos guías diferentes. Se podría crear un mapa múltiple que para un cliente “Company C” se cumplieran las dos condiciones con lo que se crearían dos ficheros de datos destino cada uno asociado a su IEAD correspondiente.

Las condiciones para la generación de los ficheros de datos destino no deben ser excluyentes se pueden dar casos, como el que se acaba de indicar en el que interese que se generen dos ficheros de datos destino.

##### ***Multimapeados que acceden a Base de Datos.***

Hay que tener especial cuidado con los mapas que acceden a base de datos, incrementan contadores, crean ficheros, etc si se asocian a mapas múltiples ya que estos mapas serán ejecutados aunque no cumplan la condición de creación del fichero de datos destino.

Todos los mapas indicados en el mapa múltiple serán ejecutados, Aunque al final de la ejecución del multimapeado, el fichero de datos destino se generará a partir del mapa que cumpla la condición especificada.

### 12.2 EJEMPLO: DISEÑO DE MAPA MÚLTIPLE.

---

## 12.2.1 MAPEADOS MÚLTIPLES

Teniendo creados dos mapas (mapa 1 y mapa 2) que tienen la misma interfaz origen de datos (Factura\_D93v7.in4) y dos interfaces de destino distintas:

- **Mapa 1 (de A a B):** *Factura\_D93v7\_S.ma4*
  - **Interfaz Origen (A):** *Factura\_D93v7.in4*
  - **Interfaz Destino (B):** *EDI\_invoic\_d\_93a\_un\_ean007.in4*
- **Mapa 2 (de A a C):** *Factura\_D96v7\_S.ma4*
  - **Interfaz Origen (A):** *Factura\_D93v7.in4*
  - **Interfaz Destino (C) :** *EDI\_invoic\_d\_96a\_un\_ean008.in4*

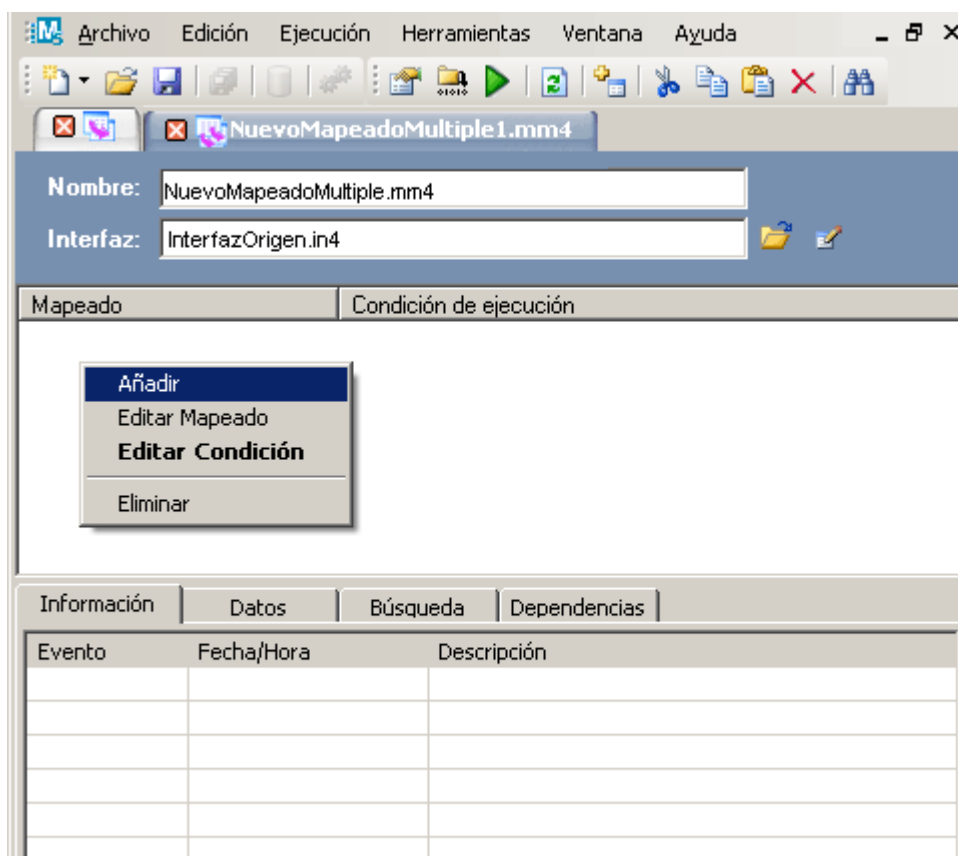
Ahora se va a crear el mapa múltiple para asociar un mapa u otro en función del campo cliente que venga indicado en el fichero origen de datos.

## 12.2.2 CREACIÓN DE UN MAPEADO MÚLTIPLE:

Para crear un mapeado multiple. Primero se debe seleccionar:

- menú principal “**Archivo-Nuevo**” opción “**Nuevo mapeado múltiple**”

Una vez seleccionada la opción se muestra la pantalla de creación de nuevo mapeado múltiple. Distinta a las otras ventanas de edición.



- **Nombre:** nombre del mapa múltiple. Los mapas multiples son archivos de extensión \*.mm4.
- **Interfaz:** permite indicar la interfaz origen de datos. El botón adjunto “**Editar**” permite abrir en una ventana de edición la interfaz seleccionada
- **Mapeado:** en esta lista se pueden añadir mapas y las condiciones que se deben cumplir para que se inicie dicho mapeado en función del campo/campos del fichero de datos origen.
- **Información:** indica la información que se debe mostrar tras la compilación del mapa múltiple.
- En la parte inferior aparecen las pestañas “*Información*”, “*Datos*”, “*Búsqueda*”, “*Dependencias*” que funcionan del mismo modo que en la ventana de edición de “*Mapeados*”

### Área de Mapas

El área de mapas es el lugar donde se añaden las posibilidades de mapeado a partir de la interfaz origen seleccionada. Está área tiene un Menú contextual.

Opción	Descripción.
<b>Añadir</b>	Permite añadir un nuevo mapa. Al hacerlo hay que indicar una condición.
<b>Editar Condición</b>	Permite tanto añadir una condición como modificarla. Accediendo al Editor de Comandos Ediwin Script.
<b>Editar Mapeado</b>	Permite abrir el mapeado seleccionado en una nueva ventana de diseño de “Mapas”
<b>Eliminar</b>	Elimina de la lista el mapeado seleccionado

Una vez indicada la opción “**Añadir**”, se listan los mapas que tienen asociada la interfaz origen de datos y que se encuentren en el mismo directorio. Se debe seleccionar el mapa al cual se quiere asociar una condición de ejecución y pulsar el botón “**Aceptar**”.

A continuación se abrirá la pantalla de edición de scripts, donde se debe indicar la condición de ejecución del mapa que se ha seleccionado. Por último cuando se haya indicado la condición y se haya pulsado el botón “**Aceptar**” se listará un nuevo mapeado asociado a una condición de ejecución.

Una configuración por ejemplo de un mapa múltiple sería:

Si los siete primeros dígitos del campo `[CABECERA.CLIENTE]` no corresponden con el código “8710400” al final de la ejecución del multimapeado se generará el fichero de datos destino que corresponde al mapa *FACTURA\_D93v7\_S.ma4*. Si el campo `[CABECERA.CLIENTE]` corresponde, se generará el fichero de datos destino a partir del mapa *FACTURA\_D96v7\_S.ma4*.

Por tanto se deberían añadir los dos mapas. Y asociar las siguientes condiciones.

Condición 1 asociada al mapa *FACTURA\_D93v7\_S.ma4*

```
Dist(IzqCad([CABECERA.CLIENTE],7),"8710400")
```

Condición 2 asociada al mapa *FACTURA\_D96v7\_S.ma4*

```
Igual(IzqCad([CABECERA.CLIENTE],7),"8710400");
```

## 13 ALGORITMO DE FUNCIONAMIENTO DEL EBIMAP.

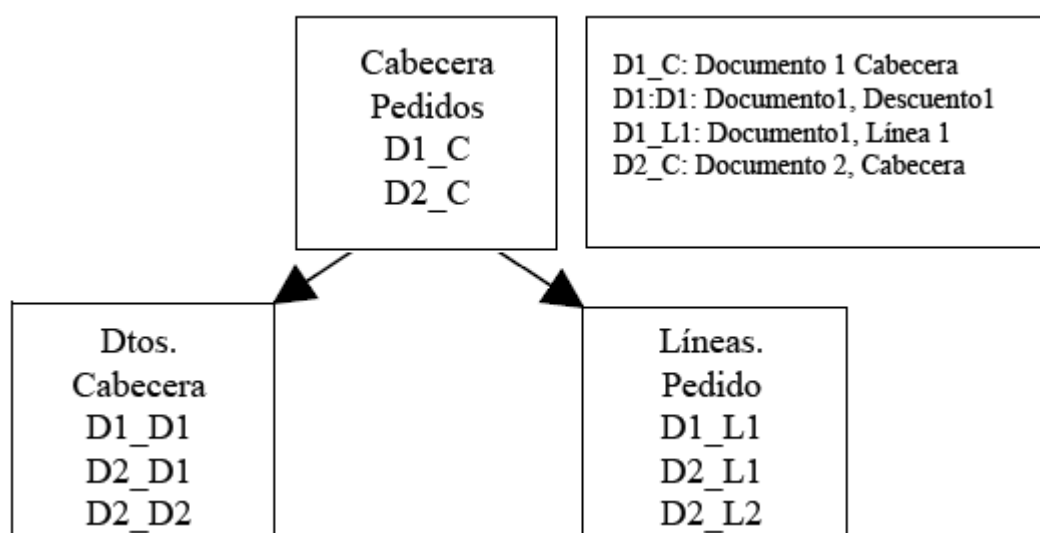
### 13.1 INTRODUCCIÓN.

En este anexo se explica de manera resumida el algoritmo de funcionamiento del ebiMap. Basándose en documentos comerciales. (pedidos, facturas, etc ) documentos que tienen información de cabecera (a quien van dirigidos, el tipo de documento que es, quien lo envía) y otra información correspondiente a la líneas de pedido (una línea por cada artículo que se pide)

Básicamente existen dos aspectos en el funcionamiento de la herramienta de mapeado ebiMap, el primero es comprender cómo se leen los datos de la interfaz origen y el segundo es cómo y cuando se generan los datos de la interfaz destino.

#### 13.1.1 ¿COMO SE LEEN LOS DATOS DE LA INTERFAZ ORIGEN?

La definición de una interfaz se basa en un conjunto de registros tipo, organizados en forma de árbol.



Los registros siempre se leen en orden dentro de cada documento según la estructura del árbol (de arriba abajo y de izquierda a derecha) D1\_C, D1\_D1, D1\_L1, D1\_L2, D2\_C, D2\_D1, D2\_D2, D2\_L1

Por supuesto, para el correcto funcionamiento debe haberse definido la forma en que se identifican los registros. Básicamente existen dos formas:

#### ***Registros de distinto tipo en archivos físicamente distintos.***

Si los registros de distinto tipo están en archivos físicamente distintos (Ej. Un archivo para las cabeceras, uno para los descuentos y otro para las líneas). En este caso los datos en los distintos archivos deben relacionarse mediante claves y claves ajenas de la misma forma en que se hace en una base de datos relacional.

### ***Registros se encuentran en un solo fichero***

Si los registros se encuentran en un solo fichero de forma consecutiva. Por ejemplo.

#### Documento 1

- Datos Cabecera para el Doc 1 (tipo de documento, emisor, receptor, etc...)
- Líneas Documento 1 (una por artículo)
  - Línea #1, (precio, nombre, código, cantidad...)
  - Línea #2, (precio, nombre, código, cantidad...)
  - Línea #3, (precio, nombre, código, cantidad...)

#### Documento 2

- Datos Cabecera para el Doc 2
- Líneas Documento 2
  - Línea #1, (precio, nombre, código, cantidad...)
  - Línea #2, (precio, nombre, código, cantidad...)
  - Línea #3, (precio, nombre, código, cantidad...)

En este caso no es necesario el uso de clave pues la situación un registro Línea o Datos de documento, indica a qué documento (registro cabecera) pertenece. Lo que sí hace falta en este caso es una forma (por ejemplo mediante etiquetas) de identificar el contenido de cada línea del archivo de texto. Por ejemplo, las cabeceras podrían empezar con la etiqueta CAB ,o las líneas mediante LIN. De esta manera se podría indicar a la interfaz un esquema del documento.

Esquema Documento Base. (tanto para el documento 1 como para el 2 , el 3, el 4...)

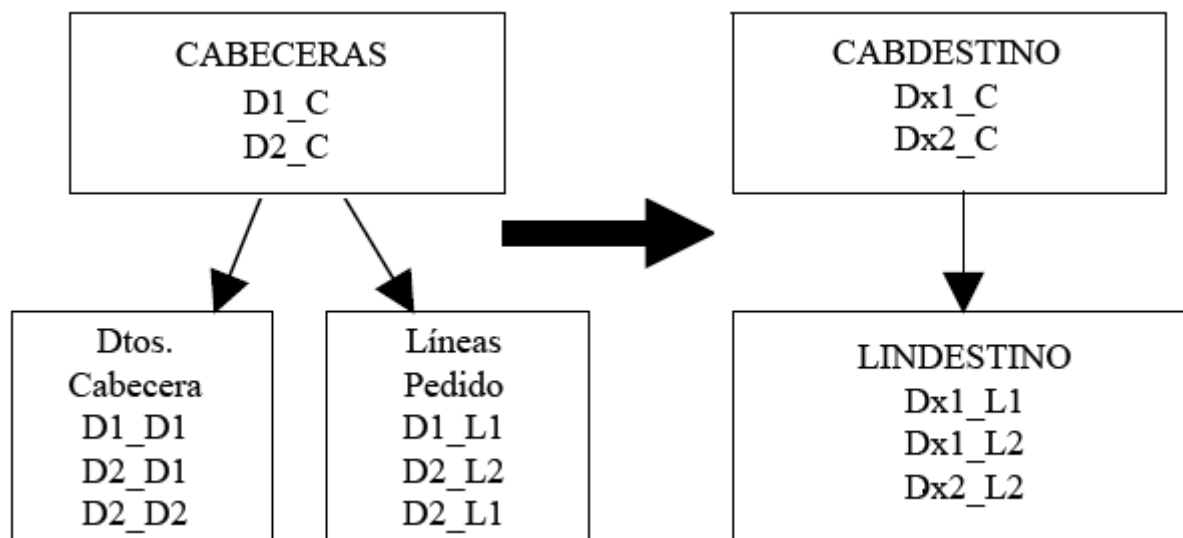
- [CAB]
  - [CAB.Documento]
  - [CAB.Emisor]
  - [CAB.Receptor]
- [LIN] (para la línea #1,#2,#3...)
  - [LIN.Artículo]
  - [LIN.Precio]
  - [LIN.Cantidad]



### 13.1.2 ¿CÓMO SE GENERAN LOS DATOS DE LA INTERFAZ DESTINO?

Una vez definido la interfaz Origen y la interfaz Destino y ha sido verificada su estructura con datos de prueba. Se crea el mapeado que indica a ebiMap como pasar los datos de una interfaz a otra. Para ello debe tenerse en cuenta el proceso interno que realiza el mapeador.

Para los ejemplos que se exponen se supone que se va a realizar un mapeado de la siguiente forma:



Para cada registro que el mapeador lee (en el orden ya conocido) se realizan las siguientes operaciones:

- Se evalúan las expresiones asociadas con el registro origen, ejecutando las funciones y los scripts indicados.
- Se crean aquellos registros de tipo de creación “Forzar Registro” en la interfaz destino. Esta forma de crear registros en la interfaz destino es útil cuando un registro está sólo compuesto por constantes de forma que nunca se va a generar de la forma en que se describe en el siguiente punto.
- Se calcula el valor de aquellos campos de la interfaz destino cuyo valor dependa del registro origen que se ha leído. Es decir, el registro origen leído actúa como disparador de ciertos campos destino. El registro disparador de una fórmula es aquel que se encuentra más abajo y a la derecha en la estructura del árbol.

*Por ejemplo: FechaEDI("101",[CABECERA.FECHA]). En este caso el disparador es el registro de cabecera. Es decir, esta fórmula calculará al leer un registro de cabecera de la interfaz origen.*

De las operaciones que se ejecutan al leer un registro se pueden derivar otras más:

- Cuando se da valor por primera vez a un campo de un registro se dice que el registro “se estrena”. Esta circunstancia se aprovecha para calcular el valor de todas aquellas expresiones que sean “Antes de abandonar el registro”.

- Si se intenta dar valor a un campo de la interfaz destino que ya tiene valor se produce una colisión. Si la gestión de la colisión es del tipo “Al Crear Registro” se generará un nuevo registro. Por ejemplo, si se lee la segunda línea del primer documento se volverá a calcular la fórmula:

```
LINDESTINO=Mult([CABECERA.IVA],[LINEAS.IMPORTE])
```

Sin embargo, como LINDESTINO ya tiene valor, se forzará que se cree un nuevo registro del tipo LINDESTINO. Esto implica que primero se calcularán todos los campos del tipo Antes de abandonar el registro y después el registro actual de tipo LINDESTINO se volcará a disco.