

## 17. Manual de usuario

Para que cualquier usuario sea capaz de instalar y utilizar la nueva funcionalidad se presenta una explicación con detalle de los requisitos del sistema que son necesarios para hacer uso de la aplicación, los pasos que hay que realizar para instalar la funcionalidad y por ultimo los *storyboards* o pantallas de la funcionalidad con la explicación de cada una de ellas.

### 17.1.1. Requisitos del sistema

#### 17.1.1.1. Poseidon for UML Standard Edition 5.0

Para poder hacer uso de la funcionalidad es requisitos disponer de Poseidon la versión estándar a partir de la 5.0 para hacer un uso correcto sobre las API de la funcionalidad.

Para obtener la aplicación se puede descargar de la pagina web [11] e instalarlo por ejemplo en la ruta 'c:\Archivos de programa\poseidon'.

#### 17.1.1.2. Apache ANT

Ant es un proyecto de código abierto de la Apache Software Foundation. Apache Ant es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente, durante la fase de compilación y construcción (build). Es similar a Make pero sin las grandes dependencias del sistema operativo.

Esta herramienta, hecha en Java, tiene la ventaja de no depender de las órdenes de la Shell de cada sistema operativo, sino que se basa en archivos de configuración XML y clases Java para la realización de las distintas tareas.

Dentro del fichero XML se incluyen los objetivos a realizar (instrucciones a ejecutar), estas instrucciones se pueden ejecutar de forma independiente o de manera conjunta.

ANT dispone de la acción Makefile usada en Unix de modo que es útil para automatizar el proceso de compilación de nuestra funcionalidad.

Para hacer uso de ANT, necesitaremos descárganos [20] la versión binaria de esta aplicación, en caso contrario también será necesario compilar ANT. Es necesario descargarse la versión 1.7, con la cual se ha trabajado.

Para configurar ANT se define la variable de entorno, a la cual se le denomina ANT\_HOME, indicando la ruta en donde se haya instalado. Para definirla se utiliza la siguiente sintaxis:

```
SET ANT_HOME=directorio_instalacion_ANT
```

Una vez definida la variable de entorno es necesario añadirla a la variable de entorno PATH con la ruta de ANT para poder compilar nuestra funcionalidad, la sintaxis es la siguiente (no hay que dejar espacios en blanco en la definición):

```
SET PATH=%PATH%;%ANT_HOME%\bin
```

### **17.1.1.3. Java**

Para trabajar con Java [21] es necesario disponer de la versión 1.5 del compilador de Java JDK. Una vez instalado, al igual que ANT debemos configurar la variable de entorno JAVA\_HOME. Su sintaxis es la siguiente:

```
SET JAVA_HOME= directorio_instalacion_JAVA
```

### **17.1.1.4. PostgreSQL para crear el esquema físico en la base de datos**

Para poder crear los ficheros en PostgreSQL en la base de datos, se ha utilizado la versión 8.2 de PostgreSQL, se puede descargar el instalador en la pagina web [14].

Una vez instalado PostgreSQL, para conectar a la base de datos desde la aplicación debemos copiar el driver de conexión que hay en la carpeta *jdbc* en el directorio de *PostgreSQL* llamado *postgresql-8.2-505.jdbc3* a la carpeta *lib* del directorio en donde se haya instalado el JRE de Java. La sintaxis es la siguiente:

```
>copy "c:\directorio_PostgreSQL\jdbc\postgresql-8.2-505.jdbc3.jar"  
"c:\directori_Java\jre\lib"
```

## 17.1.2. Instalación del plug-in

### 17.1.2.1. Compilación

Una vez se han realizado los pasos anteriores, tenemos que modificar el fichero `build.xml`, que se encuentra en la carpeta *ant* del directorio de la funcionalidad y modificar la propiedad `poseidon.lib.dir` por la ruta donde esté instalada la librería de la aplicación Poseidon:

```
<property name="poseidon.lib.dir" value="C:\ruta_directorio_Poseidon\lib"/>
```

Para añadir nuestro plug-in a Poseidon se han de copiar las clases que componen el plug-in, compiladas y empaquetadas en un archivo comprimido jar llamado `predefinedconstraint.jar`. Para compilar los ficheros con el código fuente se realizará en el mismo directorio donde se encuentra el fichero `build.xml` y ejecutaremos la siguiente sentencia:

```
>ANT package
```

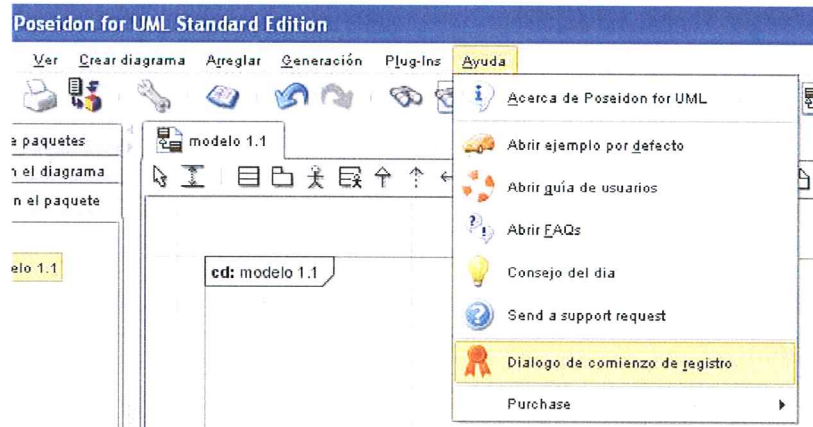
Esta instrucción genera una nueva carpeta llamada *build* dentro del directorio en donde esta la funcionalidad y dentro de esta carpeta, aparece otra carpeta llamada *lib* que es donde se encuentra el archivo `predefinedconstraint.jar` creado.

### 17.1.2.2. Integración en la herramienta Poseidon

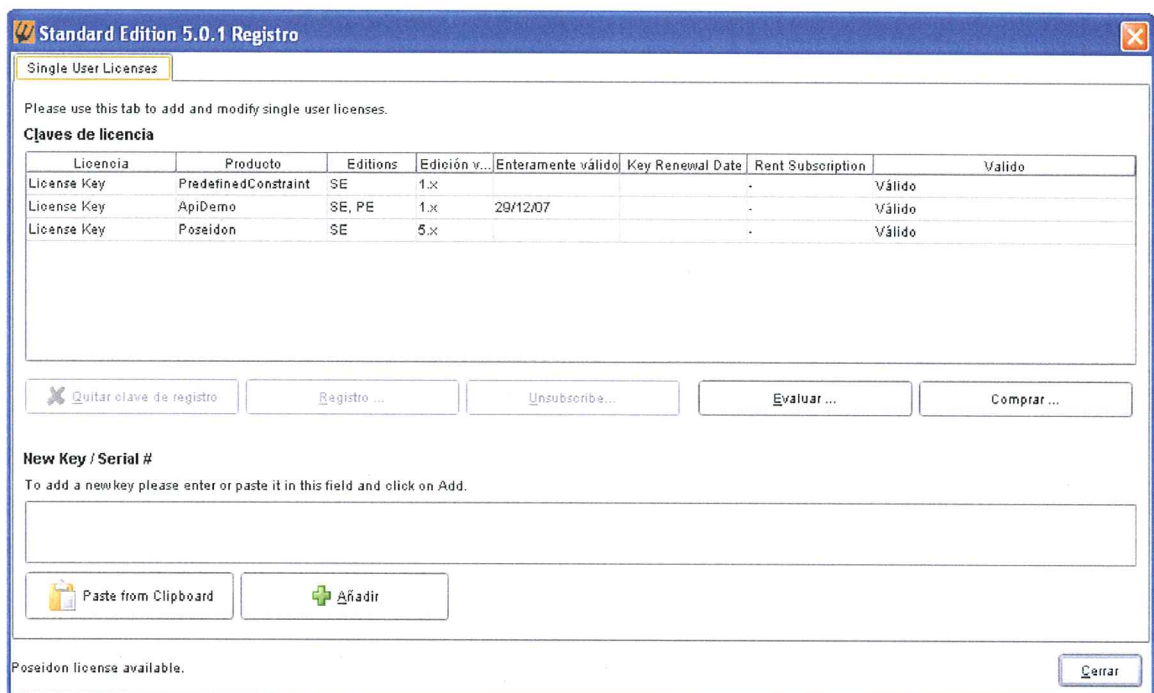
Para poder extender Poseidon mediante un plug-in se necesita una licencia suministrada por Gentleware y añadirla al administrador de licencias. En nuestro caso la licencia obtenida es la siguiente:

```
SxcwIFHQMKWqhmNXqP4nzT9n+H07GNGrSZZA7etqyclKzv8eU10gJUP8++1y  
/hrKU0Tjkw9BzCrT/BxO9ehwHs1V1eh1cAK/6IoF0cGwmdMZ9C5HlstTbUGv  
NFe1hq64zq9DX0iA0NybXqOmm82qIZ4IXMV74YlypWmWZy1yxQ
```

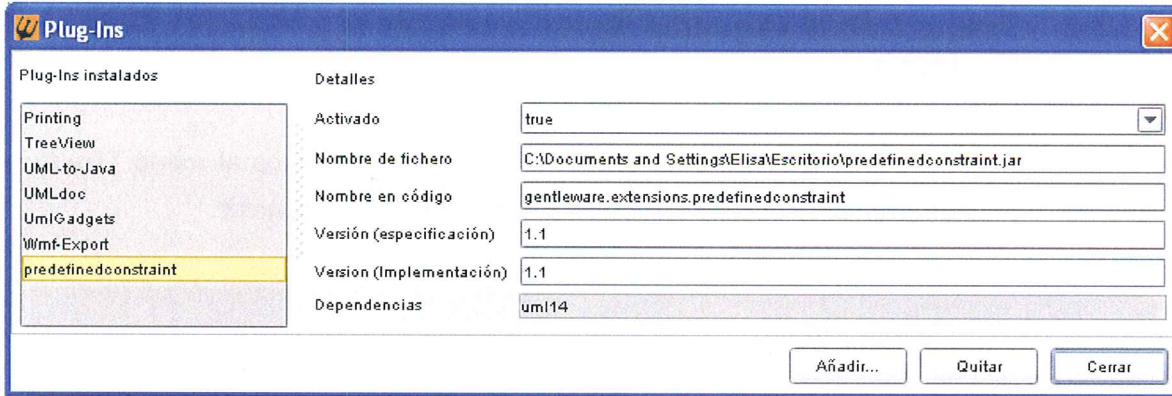
Para añadir la licencia a Poseidon dentro del menú "Ayuda" vamos a la opción "Diálogo de comienzo de registro".



En la ventana que aparece introducimos la licencia anterior y presionamos el botón "Añadir". Después de haber añadido la licencia, aparecerá en la lista de claves de registro una nueva línea con el nombre PredefinedConstraint.



El siguiente paso es añadir el plug-in mediante el fichero jar generado anteriormente yendo al menú "Plug-Ins" y presionando la opción "Plug-Ins Panel". En la ventana que aparece presionar el botón "Añadir" y localizamos el archivo predefinedconstraint.jar generado anteriormente. A continuación, cerramos la ventana y ya podemos trabajar con la nueva funcionalidad.



Para eliminar o quitar el plug-in tendremos que ir al menú "Plug-Ins" y presionar la opción "Plug-Ins Panel", de la lista de plug-ins instalados, seleccionamos el que se desea desinstalar y presionamos el botón "Quitar".

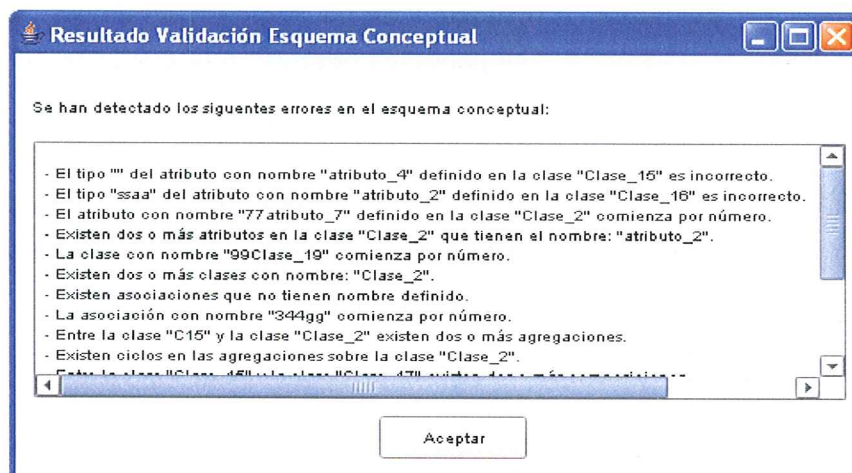
La extensión definida en Poseidon permite traducir esquemas conceptuales a esquemas lógicos con el perfil UML Data Modeling y a esquemas físicos de la base de datos.

### 17.1.3. Ejecución

En este apartado se describen las diferentes funciones que dispone la aplicación.

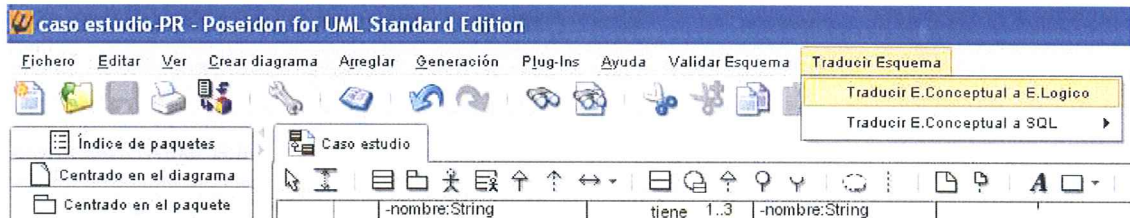
#### 17.1.3.1. Validación del esquema conceptual

Para poder validar un esquema conceptual tendremos que ir al menú "Validar Esquema" y presionar la opción "Validar Esquema Conceptual". Una vez se ha realizado el proceso de validación, se mostrará una ventana con los errores correspondientes si existen:

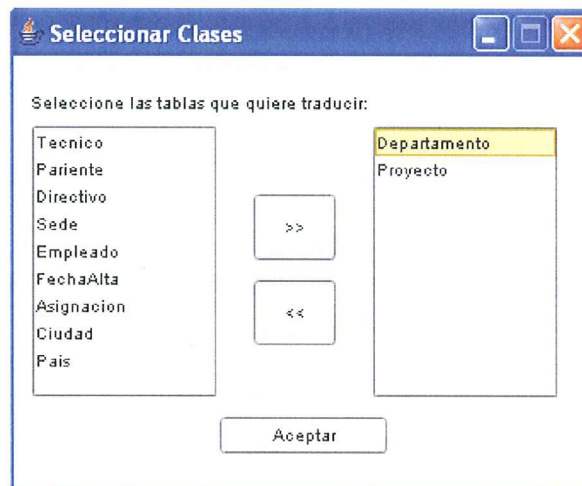


### 17.1.3.2. Traducción al esquema lógico mediante el perfil UML Data Modeling

Para traducir un esquema conceptual al esquema lógico accederemos al menú "Traducir Esquema" y presionaremos la opción "Traducir E. Conceptual a E. Lógico".

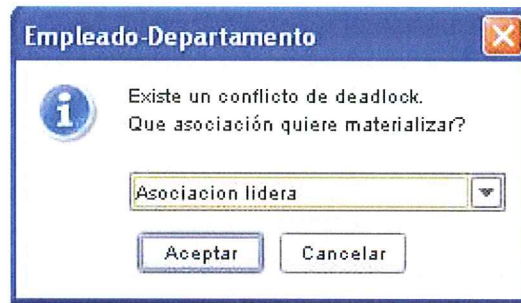


De esta opción surgirá una pantalla para especificar que tablas del esquema conceptual se desean traducir al esquema lógico:

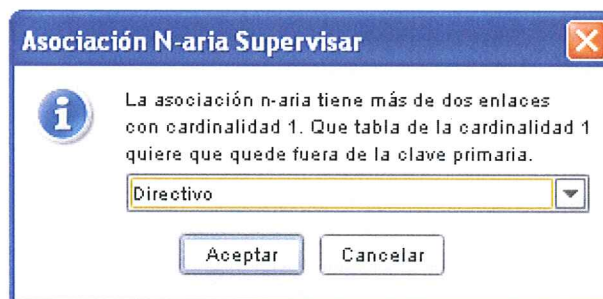


En caso de no haber ninguna clase en el esquema conceptual surgirá un mensaje en el que se expone que se puede traducir el esquema. Si hay clases seleccionadas y existen diferentes formas de traducción al esquema lógico surgirán diferentes pantallas para escoger las opciones de traducción. Las pantallas relativas a la selección de las opciones de traducción son las siguientes:

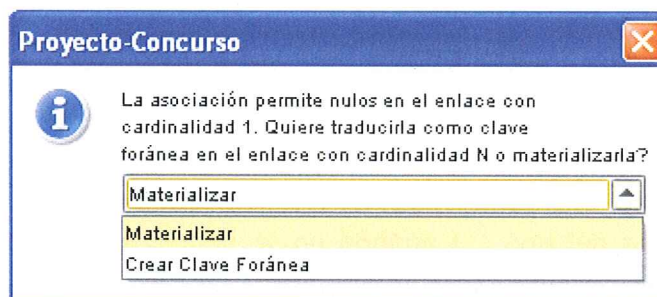
Si existen posibles anomalías al generar el esquema lógico, se muestra una pantalla para escoger la asociación a materializar en caso de conflicto de clave foránea entre dos asociaciones. La pantalla referente al conflicto de deadlock es la siguiente:



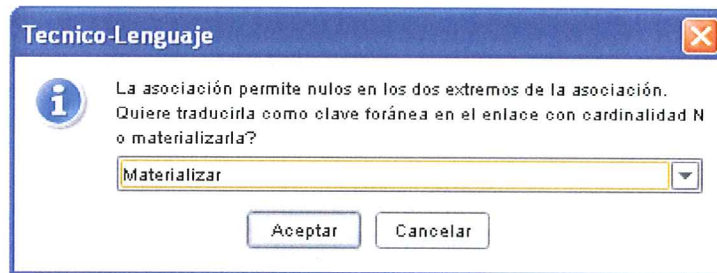
Si existen asociaciones n-arias en donde hay más de un enlace con multiplicidad 1, deberemos escoger que tabla quedará fuera de la clave primaria de la tabla que representa la asociación n-aria:



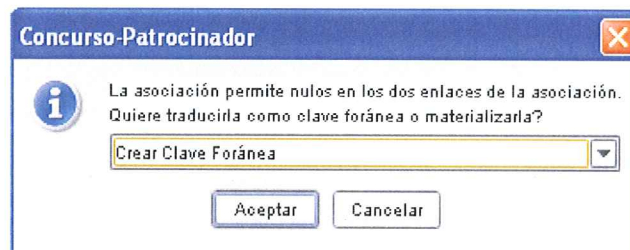
Si existen asociaciones del tipo 1-\* en las que se permite nulos en un enlace de la asociación que tiene multiplicidad 1, entonces deberemos escoger si materializar la asociación o crear la clave foránea en la tabla con enlace con multiplicidad N:



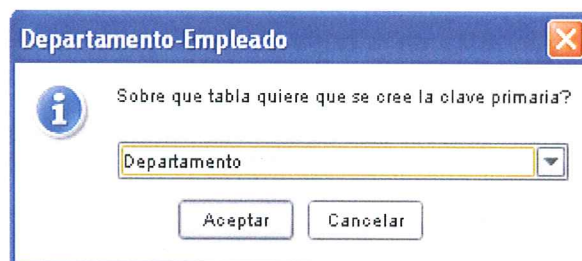
Si existe una asociación del tipo 1-\* en donde los dos extremos de la asociación pueden contener valores nulos, deberemos escoger si se desea materializar o por el contrario traducir la asociación como clave foránea:



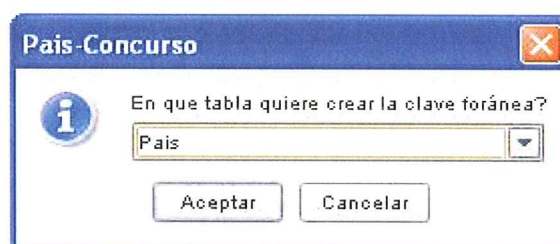
Si existe una asociación del tipo 1-1 que permite nulos en los dos extremos de la asociación, deberemos escoger si se quiere materializar la asociación o crear la clave foránea:



Si existe una asociación del tipo 1-1 que permite nulos en los dos extremos de la asociación y se ha escogido materializarla deberemos elegir de las tablas que participan en la asociación cual será la clave primaria en la asociación materializada:

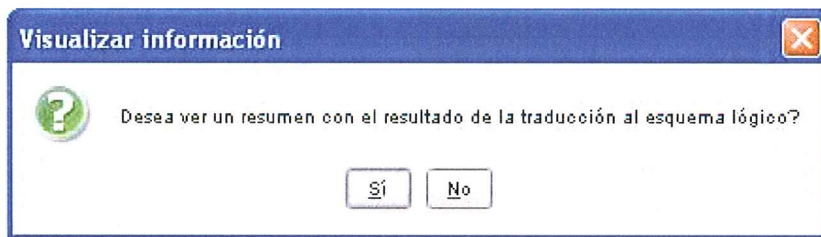


Si existen asociaciones del tipo 1-1 cuando no se materializan, deberemos escoger en que tabla de los extremos de la asociación se quiere crear la clave foránea:

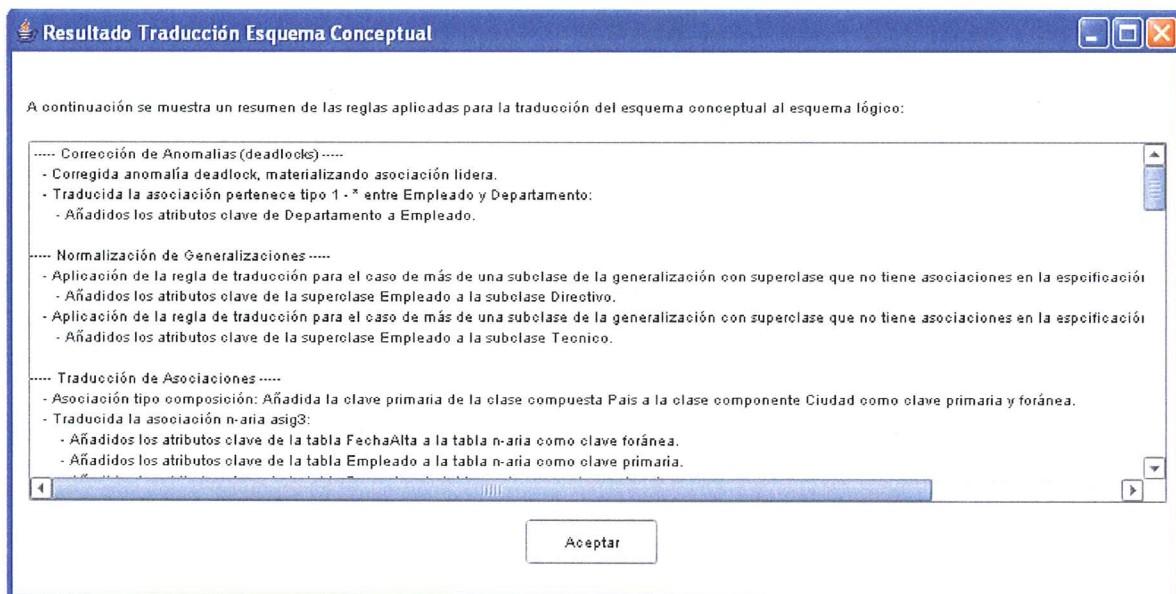


Una vez escogidas las opciones de traducción podremos escoger si queremos ver un resumen con las reglas que se ha aplicado para generar el esquema lógico:



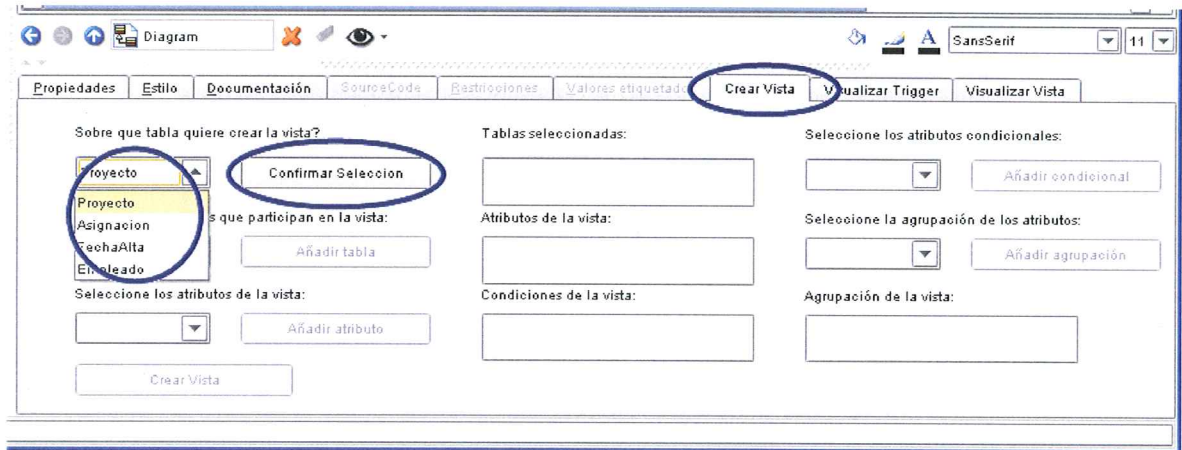


En caso de querer ver la información del esquema lógico se mostrará la siguiente pantalla con la información del esquema lógico:

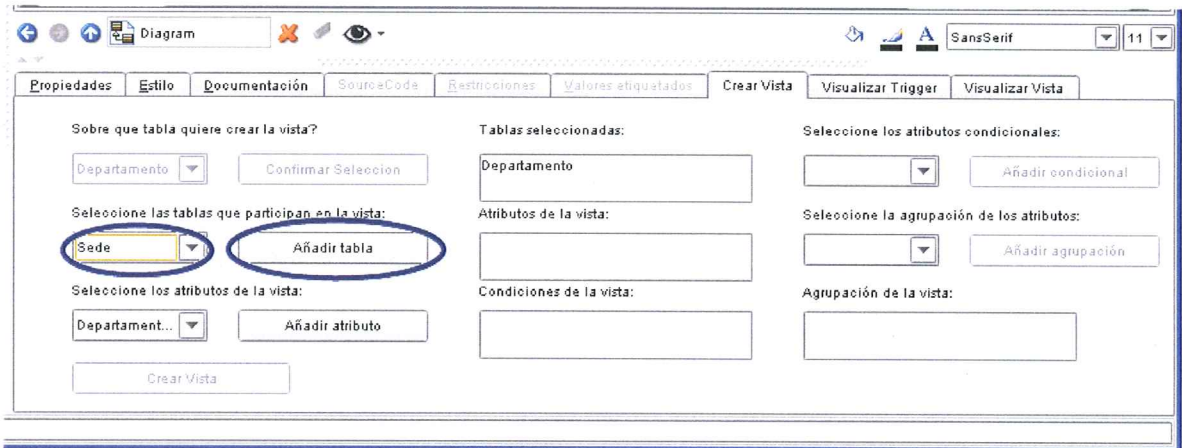


### 17.1.3.3. Creación de vistas

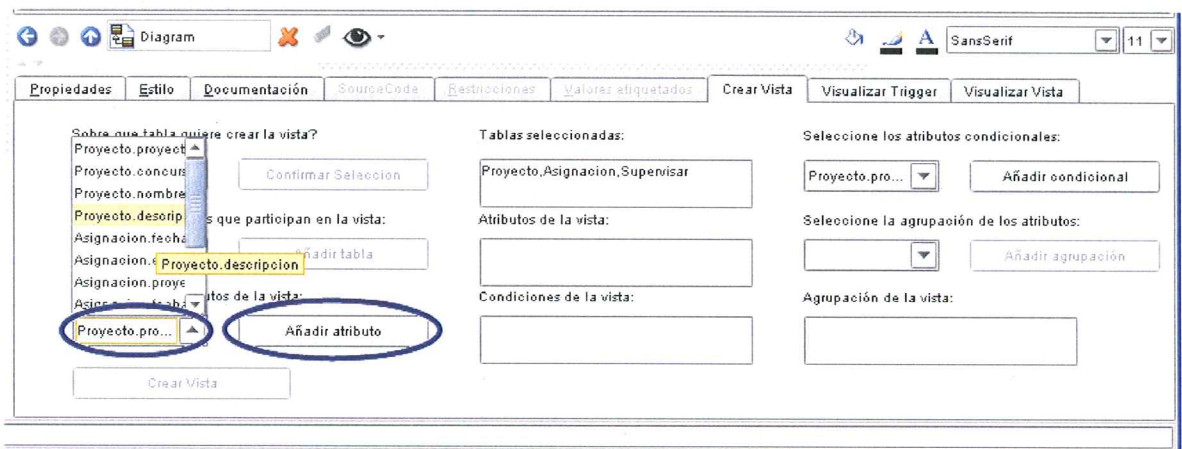
Para crear vistas sobre el esquema lógico generado, deberemos presionar sobre el TabFolder denominado "Crear Vista". Para crear una vista previamente deberemos generar el esquema lógico, una vez generado se activará el desplegable de selección tablas en el que se muestran las tablas del esquema lógico. Posteriormente seleccionaremos la tabla sobre la que se creará la vista y presionaremos el botón "Confirmar Selección", este evento actualizará y activará los demás campos.



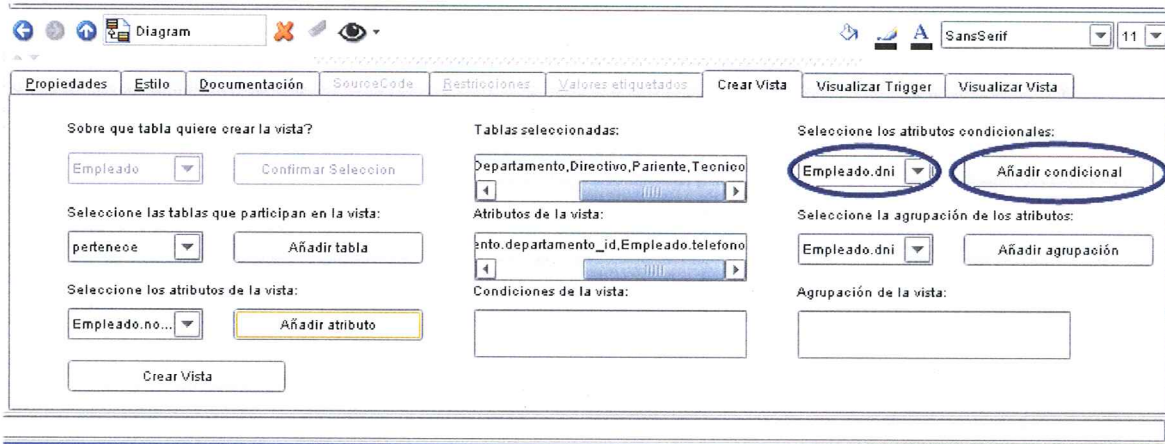
Para escoger tablas que están relacionadas con la tabla principal, deberemos escoger del desplegable de tablas relacionadas, la tabla que nos interesa y presionar el botón "Añadir tabla":



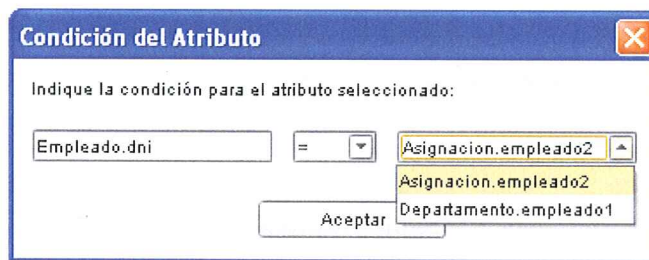
Para escoger los atributos de la vista deberemos elegir el atributo del desplegable en el que se muestran todos los atributos y a continuación presionar el botón "Añadir atributo":



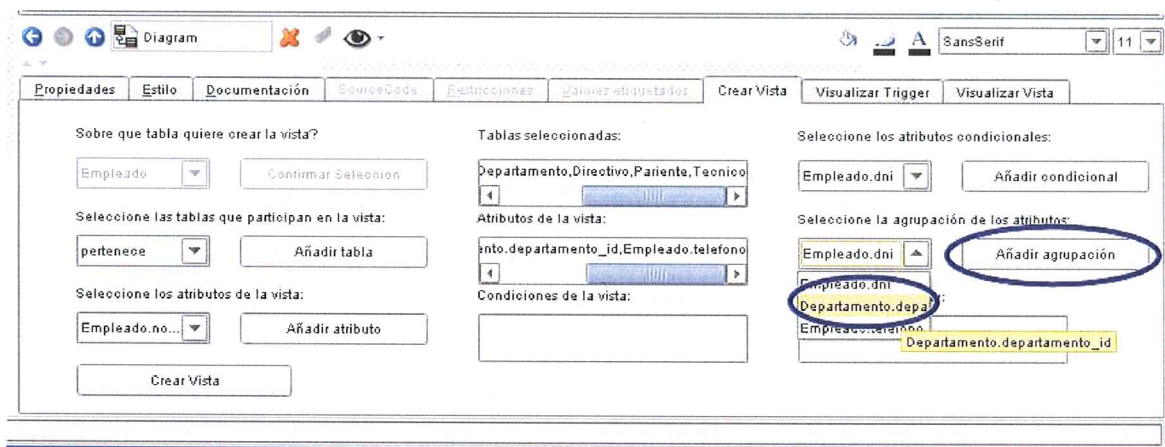
Para añadir condiciones a la vista deberemos seleccionar el atributo clave del desplegable de selección de condiciones y a continuación presionar el botón "Añadir condicional":



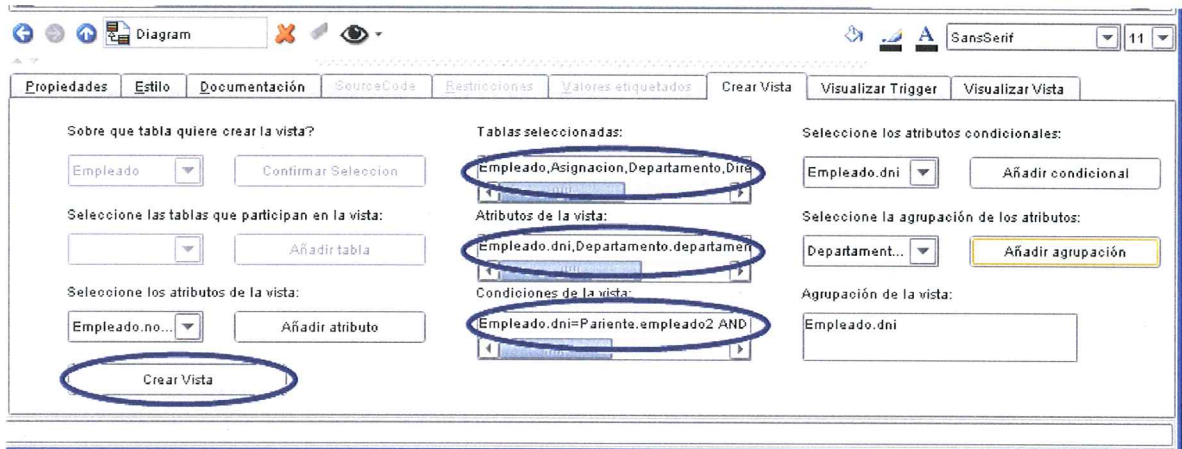
En caso de que exista más de una posible opción de condición sobre la parte derecha de la igualdad se mostrará una pantalla en la que deberemos elegir cual es la condición que se debe cumplir:



Si queremos que la vista esté agrupada por atributos deberemos escoger el atributo en el desplegable que muestra las agrupaciones y a continuación presionar el botón "Añadir agrupación":



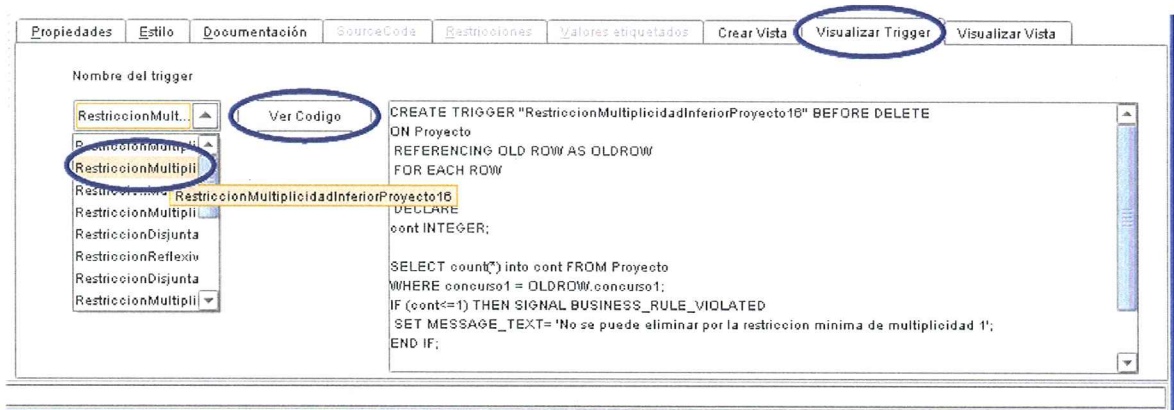
A medida que vayamos añadiendo información para crear la vista las áreas de texto se irán actualizando con la información escogida:



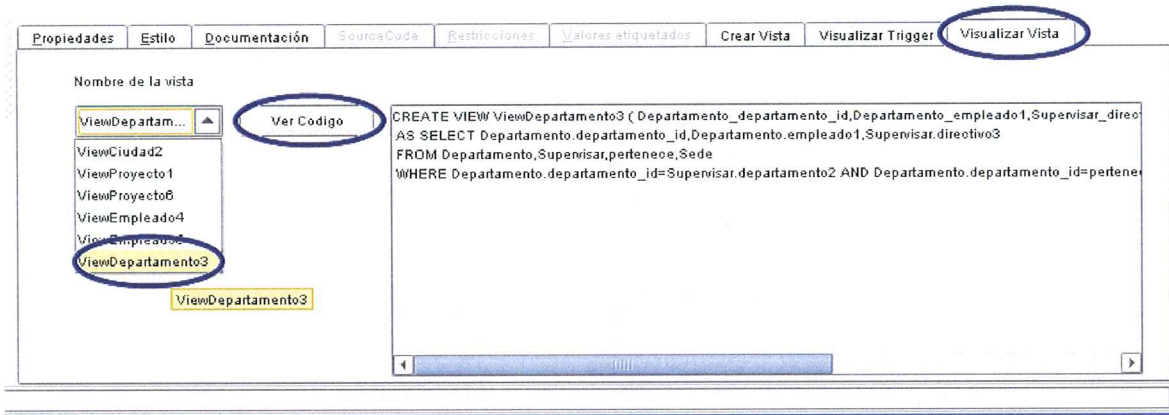
Para crear la vista una vez especificada la información deberemos presionar el botón "Crear Vista" del TabFolder.

#### 17.1.3.4. Visualización de triggers y vistas

Para visualizar el código SQL estándar de los triggers sobre el esquema lógico generado, deberemos presionar sobre el TabFolder denominado "Visualizar Trigger". Sobre el desplegable que aparece, seleccionaremos el nombre del trigger que queremos visualizar y a continuación presionaremos el botón "Ver código":

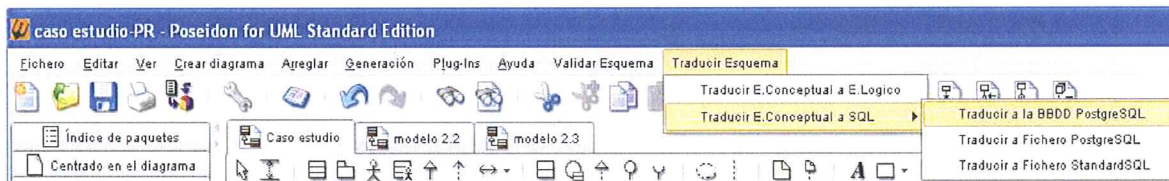


Para visualizar el código SQL estándar de las vistas creadas sobre el esquema lógico, deberemos presionar sobre el TabFolder denominado "Visualizar Vista". Sobre el desplegable que aparece, seleccionaremos el nombre de la vista que queremos visualizar y a continuación presionaremos el botón "Ver código":

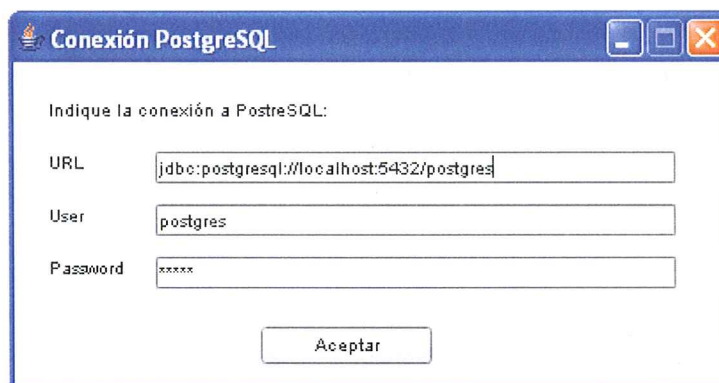


### 17.1.3.5. Traducción a la base de datos de PostgreSQL

Para traducir un esquema conceptual a un esquema físico en PostgreSQL directamente a la base de datos, seleccionaremos el menú "Traducir Esquema" -> "Traducir E. Conceptual a SQL" y el menú "Traducir a la BBDD PostgreSQL":

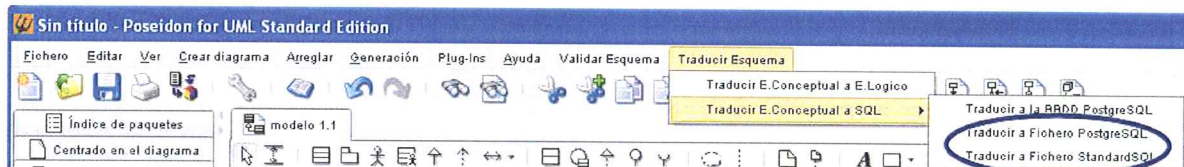


Además, de las pantallas las diferentes opciones de traducción al esquema físico descritas anteriormente, deberemos introducir los datos de conexión a la base de datos de PostgreSQL:



### 17.1.3.6. Traducción a fichero del esquema físico SQL Estándar y PostgreSQL

Para traducir un esquema conceptual a un esquema físico en PostgreSQL a un fichero script SQL, seleccionaremos el menú "Traducir Esquema" -> "Traducir E. Conceptual a SQL" y el menú "Traducir a Fichero PostgreSQL" o "Traducir a Fichero StandardSQL":



Además, de las pantallas las diferentes opciones de traducción al esquema físico descritas anteriormente, deberemos especificar la ruta y el fichero donde se guardaran los scripts SQL para la creación y eliminación de tablas:

