

Anexos

Apéndice A

Manual de usuario

A.1. Introducción

El presente documento tiene como propósito describir de manera clara las funcionalidades de la aplicación *Herramienta para el aprendizaje del álgebra relacional y la optimización de consultas*. Contiene información detallada para su correcto manejo, de manera que cualquier persona que lo lea sea capaz de utilizar la herramienta y aprovecharla lo máximo posible.

Visión general del manual

A continuación se resumen las distintas partes de que consta el manual de usuario:

- Perspectiva general de la aplicación: incluye una descripción general del programa y los requerimientos necesarios para hacerlo funcionar.
- Gua rápida de instalación: describe los pasos básicos a seguir para una correcta instalación y puesta en marcha de la aplicación.
- Tutorial completo: detalla todas las funcionalidades que ofrece la aplicación.

A.2. Perspectiva general de la aplicación

Descripción

La Herramienta para el aprendizaje del álgebra relacional es una aplicación gráfica que permite definir consultas en álgebra relacional y árboles de expresiones en álgebra relacional, ejecutarlos sobre relaciones de ejemplo, y optimizarlos.

La herramienta tiene principalmente una finalidad educativa, e incluye en consecuencia opciones de ejecución y optimización paso a paso, detección de errores, y sugerencias.

Requerimientos

Los requerimientos de la aplicación son los siguientes:

- Un PC con sistema operativo Windows, Linux o MacOS.
- Última versión de la maquina virtual de java (JVM) instalada.

A.3. Guia rápida de instalación

Instalación

A continuación se detallan los pasos a seguir para la instalación de la Herramienta de aprendizaje del álgebra relacional y la optimización de consultas:

1. Introduzca el CD de la aplicación en el lector correspondiente.
2. Copie el contenido del CD al directorio donde desee instalar la aplicación.
3. Asegúrese de que todo el contenido de CD ha sido copiado correctamente.

Puesta en marcha

Una vez instalada la aplicación busque el ejecutable *hadAR.jar* en la ubicación donde instaló la aplicación y ejecútelo.

A.4. Tutorial completo

La ventana principal de la aplicación contiene distintos elementos (ver Figura A.1)

1. Menú principal de la aplicación.
2. Barra de herramientas general.
3. Ventana de relaciones.
4. Ventana de relación seleccionada.
5. Ventana de expresiones en álgebra relacional.

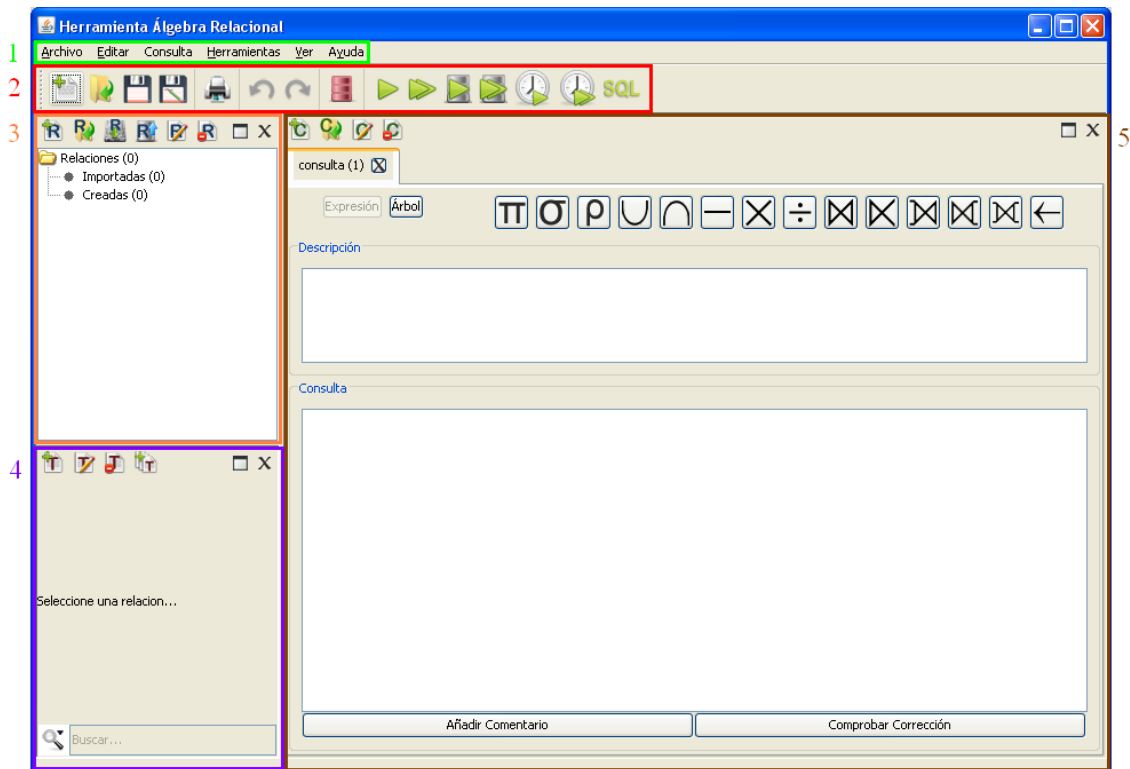


Figura A.1: Ventana principal de la aplicación

A.4.1. Menú principal

Archivo

El submenú *Archivo* incluye las siguientes funciones (ver Figura A.2):

- *Nuevo proyecto*: crea un nuevo proyecto vacío. Se entiende por proyecto un nuevo conjunto de relaciones, tuplas y consultas. Si hay cambios en el actual pregunta al usuario antes de crear el nuevo. Esta opción es accesible también desde la barra de herramientas general.
- *Nueva relación*: abre la ventana de creación de una nueva relación. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relaciones.
- *Nueva consulta*: crea una nueva pestaña para una nueva consulta en la ventana de expresiones en álgebra relacional. Esta opción es accesible también desde la barra de herramientas propia de la ventana de expresiones en álgebra relacional.

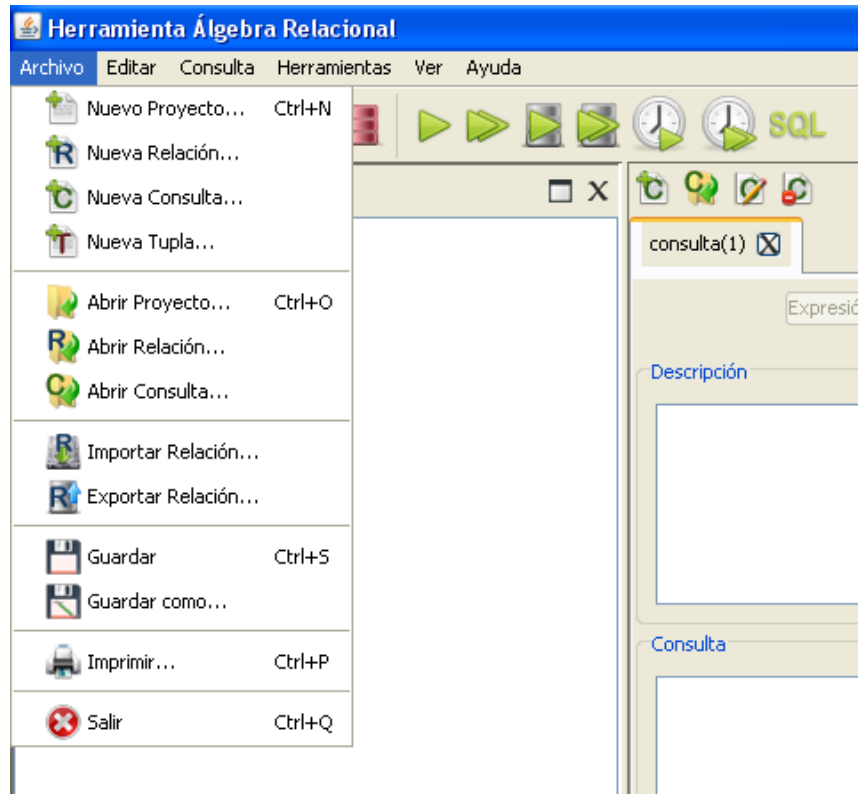


Figura A.2: Opciones del submenú Archivo

- *Nueva tupla*: abre la ventana de creación de una nueva tupla para la relación actualmente seleccionada. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relación seleccionada.
- *Abrir proyecto*: abre la ventana para la selección de un proyecto previamente guardado y lo establece como proyecto actual. Esta opción es accesible también desde la barra de herramientas general.
- *Abrir relación*: abre la ventana para la selección de un archivo XML que contenga una o mas relaciones. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relaciones.
- *Abrir consulta*: abre la ventana para la selección de un archivo XML que contenga una consulta en álgebra relacional. Esta opción es accesible también desde la barra de herramientas propia de la ventana de expresiones en álgebra relacional.
- *Importar relación*: extrae las relaciones de la base de datos externa. Para ello previamente se ha tenido que conectar a dicha base de datos. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relaciones.

- *Exportar relación*: introduce la relación seleccionada en la base de datos externa. Para ello previamente se ha tenido que conectar a dicha base de datos. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relaciones.
- *Guardar*: guarda los cambios del proyecto actual. Esta opción es accesible también desde la barra de herramientas general.
- *Guardar como*: guardar el proyecto actual en un fichero .ral Esta opción es accesible también desde la barra de herramientas general.
- *Imprimir*: abre la ventana de selección de impresión permitiendo imprimir una consulta, árbol o relación del proyecto actual. Esta opción es accesible también desde la barra de herramientas general.
- *Salir*: cierra la aplicación. Si hay cambios no guardados en el diagrama actual pregunta al usuario si desea guardar antes.

Editar

El submenú *Editar* incluye las siguientes funciones (ver Figura A.3):

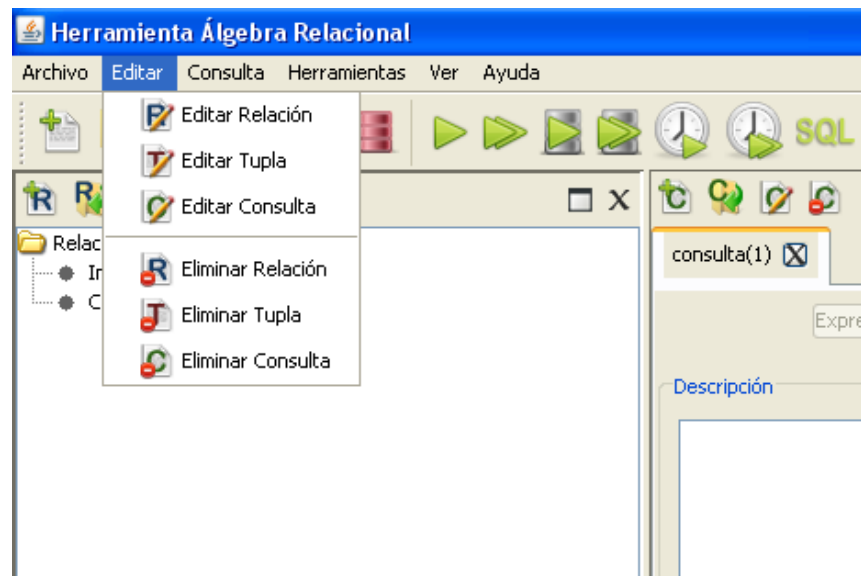


Figura A.3: Opciones del submenú Editar

- *Editar relación*: abre la ventana para la edición de la relación actualmente seleccionada. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relaciones.

- *Editar tupla*: abre la ventana para la edición de la tupla actualmente seleccionada. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relación actual.
- *Editar consulta*: abre la ventana para la edición de la consulta actualmente seleccionada. Esta opción es accesible también desde la barra de herramientas propia de la ventana de expresiones en álgebra relacional.
- *Eliminar relación*: elimina la relación actualmente seleccionada. Antes de ello pide al usuario confirmación. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relaciones.
- *Eliminar tupla*: elimina la tupla actualmente seleccionada. Antes de ello pide al usuario confirmación. Esta opción es accesible también desde la barra de herramientas propia de la ventana de relación actual.
- *Eliminar consulta*: elimina la consulta actualmente seleccionada. Antes de ello pide al usuario confirmación. Esta opción es accesible también desde la barra de herramientas propia de la ventana de expresiones en álgebra relacional.

Consulta

El submenú *Consulta* incluye las siguientes funciones (ver Figura A.4):

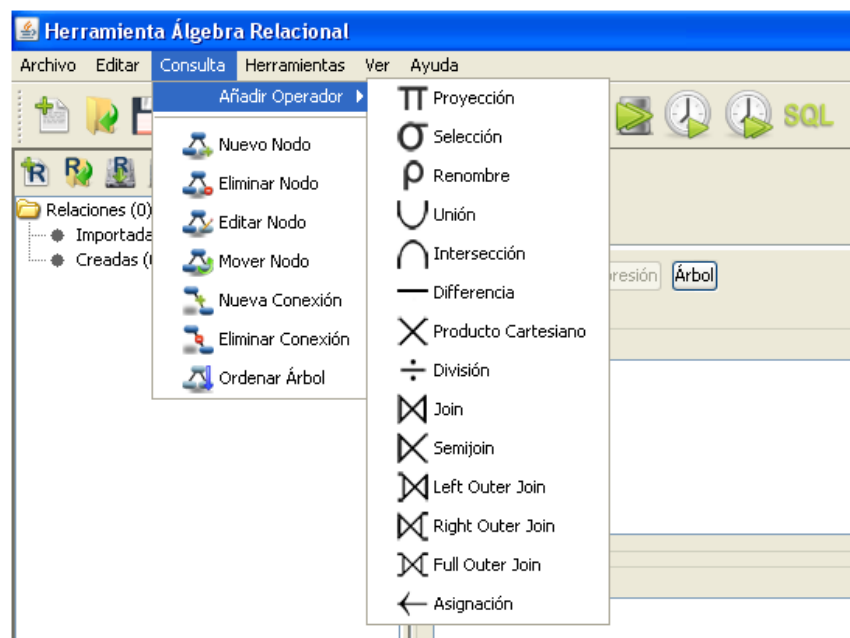


Figura A.4: Opciones del submenú Consulta

- *Añadir operador*: permite la inclusión de un nuevo operador en la consulta en álgebra relacional o en el árbol de álgebra relacional. Estos operadores se explican en la sección dedicada a la barra de herramientas de la ventana de expresiones en álgebra relacional
- *Nuevo nodo*: permite la inclusión de un nuevo nodo en el árbol actual. Esta opción es accesible también desde la barra de herramientas del panel de diseño de árboles.
- *Eliminar nodo*: elimina los nodos seleccionados del árbol actual. Antes de realizar la operación pide confirmación al usuario. Esta opción es accesible también desde la barra de herramientas del panel de diseño de árboles.
- *Editar nodo*: edita el nodo seleccionado en el árbol actual. Esta opción es accesible también desde la barra de herramientas del panel de diseño de árboles.
- *Mover nodo*: permite mover el nodo seleccionado en el panel del árbol actual. Esta opción es accesible también desde la barra de herramientas del panel de diseño de árboles.
- *Nueva conexión*: permite incluir una conexión entre dos nodos del árbol actual. Esta opción es accesible también desde la barra de herramientas del panel de diseño de árboles.
- *Eliminar conexión*: permite seleccionar los nodos cuya conexión queremos eliminar. Esta opción es accesible también desde la barra de herramientas del panel de diseño de árboles.
- *Ordenar árbol*: ordena el árbol actual automáticamente para su mejor lectura adaptándose al espacio disponible. Esta opción es accesible también desde la barra de herramientas del panel de diseño de árboles.

Herramientas

El submenú *Herramientas* incluye las siguientes funciones (ver Figura A.5):

- *Ejecutar consulta*: ejecuta la consulta actual en la propia aplicación y genera una ventana con los resultados. Esta opción es accesible también desde la barra de herramientas general.
- *Ejecutar consulta paso a paso*: ejecuta la consulta actual en la propia aplicación generando una ventana con los resultados paso a paso permitiendo al usuario avanzar por ellos. Esta opción es accesible también desde la barra de herramientas general.

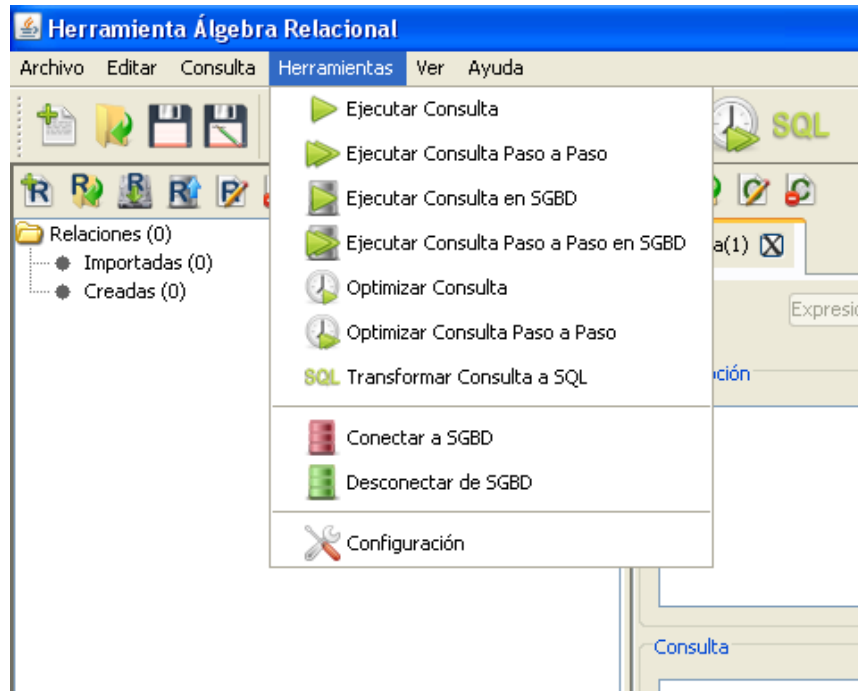


Figura A.5: Opciones del submenú Herramientas

- *Ejecutar consulta en SGBD*: ejecuta la consulta actual en el sistema gestor de bases de datos externo y genera una ventana con los resultados. Esta opción es accesible también desde la barra de herramientas general.
- *Ejecutar consulta paso a paso en SGBD*: ejecuta la consulta actual en el sistema gestor de bases de datos externo generando una ventana con los resultados paso a paso permitiendo al usuario avanzar por ellos. Esta opción es accesible también desde la barra de herramientas general.
- *Optimizar consulta*: optimiza la consulta automáticamente y sustituye su consulta por una expresión optimizada. Esta opción es accesible también desde la barra de herramientas general.
- *Optimizar consulta paso a paso*: optimiza la consulta actual mostrándole en una nueva ventana los pasos seguidos con explicaciones. Esta opción es accesible también desde la barra de herramientas general.
- *Transformar consulta a SQL*: transforma la consulta actual a SQL y muestra el resultado en una ventana nueva. Esta opción es accesible también desde la barra de herramientas general.
- *Conectar/Desconectar a SGBD*: abre la ventana para la conexión al SGBD y en caso de estar ya conectado cierra la conexión. Esta opción es accesible también desde la barra de herramientas general.

- *Configuración*: abre la ventana de configuración permitiendo la modificación de ciertos parámetros de la aplicación.

Ver

El submenú *Ver* incluye las siguientes funciones (ver Figura A.6):

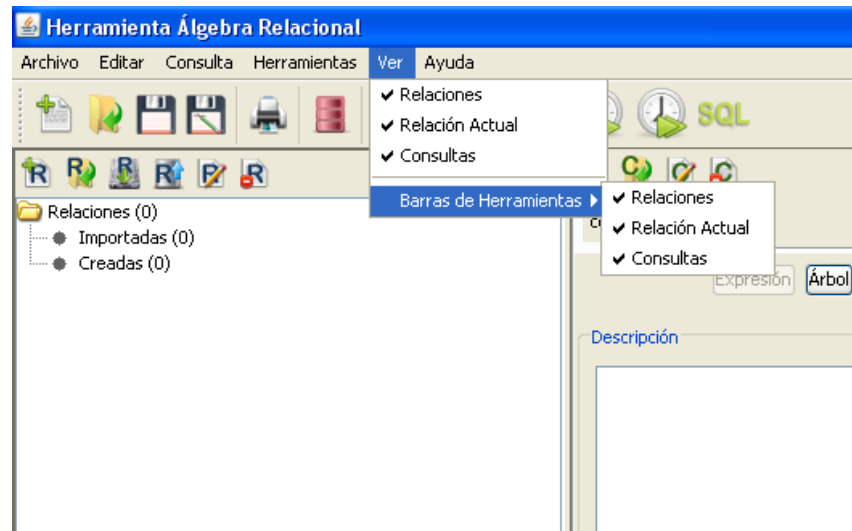


Figura A.6: Opciones del submenú Ver

- *Relaciones*: en caso de estar seleccionado (denotado con el *tick* de la izquierda) muestra la ventana de relaciones. En caso de deseccionarse cierra dicha ventana.
- *Relación Actual*: en caso de estar seleccionado (denotado con el *tick* de la izquierda) muestra la ventana de relación seleccionada. En caso de deseccionarse cierra dicha ventana.
- *Consultas*: en caso de estar seleccionado (denotado con el *tick* de la izquierda) muestra la ventana de expresiones en álgebra relacional. En caso de deseccionarse cierra dicha ventana.
- *Barra de herramientas*: permite cerrar o abrir las barras de herramientas de las distintas ventanas:
 - *Relaciones*: cierra/abre la barra de herramientas de la ventana de relaciones.
 - *Relación actual*: cierra/abre la barra de herramientas de la ventana de relación seleccionada.
 - *Consultas*: cierra/abre la barra de herramientas de la ventana de expresiones en álgebra relacional.

Ayuda

El submenú *Ayuda* incluye las siguientes funciones (ver Figura A.7):

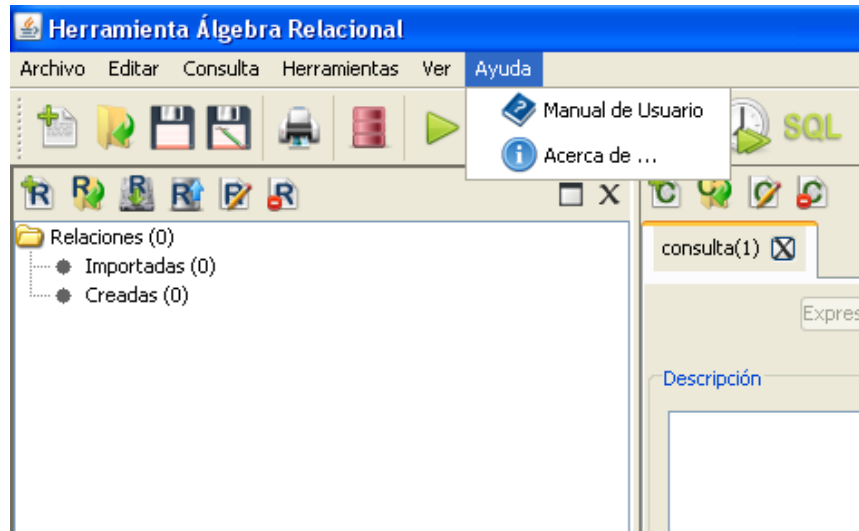


Figura A.7: Opciones del submenú Ayuda

- *Manual de usuario*: abre el PDF de este mismo manual.
- *Acerca de*: muestra información sobre la aplicación.

A.4.2. Barra de herramientas general

Contiene los siguientes botones de izquierda a derecha (ver Figura A.8):



Figura A.8: Opciones de la barra de herramientas general

-  - *Nuevo proyecto*: crea un nuevo proyecto vacío. Si hay cambios en el actual pregunta al usuario antes de crear el nuevo. Esta opción es también accesible desde el submenú *Archivo*.
-  - *Abrir proyecto*: abre la ventana para la selección de un proyecto previamente guardado y lo establece como proyecto actual. Esta opción es también accesible desde el submenú *Archivo*.

-  - *Guardar*: guarda los cambios del proyecto actual. Esta opción es también accesible desde el submenú *Archivo*.
-  - *Guardar como*: guardar el proyecto actual en un fichero .ral. Esta opción es también accesible desde el submenú *Archivo*.
-  - *Imprimir*: abre la ventana de selección de impresión permitiendo imprimir una consulta, árbol o relación del proyecto actual. Esta opción es también accesible desde el submenú *Archivo*.
-  - *Conectar/Desconectar a SGBD*: abre la ventana para la conexión al SGBD y en caso de estar ya conectado cierra la conexión. Esta opción es también accesible desde el submenú *Herramientas*.
-  - *Ejecutar consulta*: ejecuta la consulta actual en la propia aplicación y genera una ventana con los resultados. Esta opción es también accesible desde el submenú *Herramientas*.
-  - *Ejecutar consulta paso a paso*: ejecuta la consulta actual en la propia aplicación generando una ventana con los resultados paso a paso permitiendo al usuario avanzar por ellos. Esta opción es también accesible desde el submenú *Herramientas*.
-  - *Ejecutar consulta en SGBD*: ejecuta la consulta actual en el sistema gestor de bases de datos externo y genera una ventana con los resultados. Esta opción es también accesible desde el submenú *Herramientas*.
-  - *Ejecutar consulta paso a paso en SGBD*: ejecuta la consulta actual en el sistema gestor de bases de datos externo generando una ventana con los resultados paso a paso permitiendo al usuario avanzar por ellos. Esta opción es también accesible desde el submenú *Herramientas*.
-  - *Optimizar consulta*: optimiza la consulta automáticamente y sustituye su consulta por una expresión optimizada. Esta opción es también accesible desde el submenú *Herramientas*.
-  - *Optimizar consulta paso a paso*: optimiza la consulta actual mostrándole en una nueva ventana los pasos seguidos con explicaciones. Esta opción es también accesible desde el submenú *Herramientas*.
-  - *Transformar consulta a SQL*: transforma la consulta actual a SQL y muestra el resultado en una ventana nueva. Esta opción es también accesible desde el submenú *Herramientas*.

A.4.3. Ventana de relaciones

La ventana de relaciones es la parte de la interfaz encargada de tratar con las relaciones disponibles para el proyecto actual. En ella podrá acceder a las relaciones que haya cargado/importado desde la base de datos externa o desde ficheros de relaciones externos. También podrá acceder a las relaciones que haya generado, así como crear nuevas relaciones. Esta ventana se divide en las siguientes partes (ver Figura A.9):

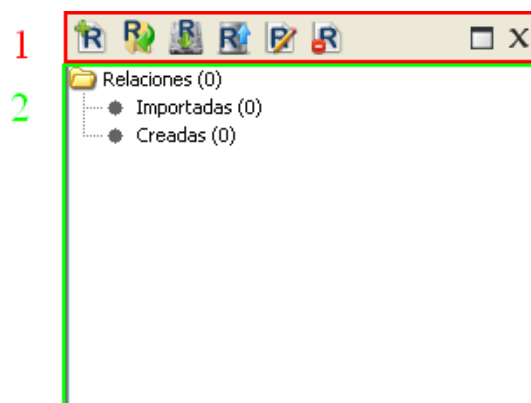


Figura A.9: Ventana de relaciones

1. Barra de herramientas de la ventana de relaciones.
2. Árbol de relaciones actuales.

Barra de herramientas de la ventana de relaciones

Contiene los siguientes botones de izquierda a derecha (ver Figura A.10):



Figura A.10: Barra de herramientas de la ventana de relaciones

- - *Nueva relación*: abre la ventana de creación de una nueva relación. Esta opción es accesible también desde el submenú *Archivo*.
- - *Abrir relación*: abre la ventana para la selección de un archivo XML que contenga una o mas relaciones. Esta opción es accesible también desde el submenú *Archivo*.

-  - *Importar relación*: extrae las relaciones de la base de datos externa. Para ello previamente se ha tenido que conectar a dicha base de datos. Esta opción es accesible también desde el submenú *Archivo*
-  - *Exportar relación*: introduce la relación seleccionada en la base de datos externa. Para ello previamente se ha tenido que conectar a dicha base de datos. Esta opción es accesible también desde el submenú *Archivo*
-  - *Editar relación*: abre la ventana para la edición de la relación actualmente seleccionada. Esta opción es accesible también desde el submenú *Editar*.
-  - *Eliminar relación*: elimina la relación actualmente seleccionada. Antes de ello pide al usuario confirmación. Esta opción es accesible también desde el submenú *Editar*.
-  /  - *Maximizar/minimizar*: maximiza o minimiza la ventana de relaciones, de tal forma que ocupe toda la ventana de la aplicación o sólo una parte.
-  - *Cerrar*: cierra la ventana de relaciones y para volver a ser abierta debe utilizarse el submenú *Ver*.

Árbol de relaciones actuales

Muestra las relaciones actuales del proyecto divididas en secciones (ver Figura A.11). Además del nombre de la relación se informa entre paréntesis del número actual de tuplas que contiene cada relación, así como del número de relaciones que compone cada una de las secciones generales:

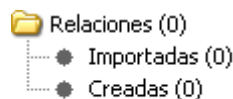


Figura A.11: Árbol de relaciones actuales

- *Importadas*: en esta sección aparecen todas las relaciones que han sido extraídas o bien de la base de datos externa o bien de ficheros de relaciones *.rel*.
- *Creadas*: en esta sección aparecen todas las relaciones creadas por el usuario o que han sido creadas durante la ejecución de una consulta (si utiliza el operador de asignación se crearán las relaciones correspondientes y se les asignarán los datos apropiados).

Creación de una nueva relación

A continuación se explica el proceso de creación de una nueva relación mostrando las ventanas que intervienen. El primer paso será hacer click en el icono de nueva relación tras lo cual se mostrará una ventana como la que aparece en la Figura A.12. En dicha ventana debe introducir el nombre que desea para la relación en el cuadro de texto etiquetado como *Nombre de la relación* (no puede haber dos relaciones con el mismo nombre así que si el nombre ya esta siendo utilizado se le avisará con una ventana de error). Ahora, para introducir los diferentes atributos no tiene mas que apretar el botón *Nuevo atributo* tras lo cual se añadirá un nuevo atributo a la lista de atributos en la parte central de la ventana. En el primer cuadro de texto que aparecerá debe introducir el nombre del atributo, en el siguiente espacio seleccione desde la lista desplegable el dominio del atributo, y en el siguiente desplegable seleccione si desea añadir alguna propiedad a ese atributo (*primary key*, *not null*...). Cuando haya completado la información referente a la relación que está creando, pulse el botón de *Aceptar* para confirmar la creación de la relación y que pase al árbol de relaciones en el menú *Creadas*.

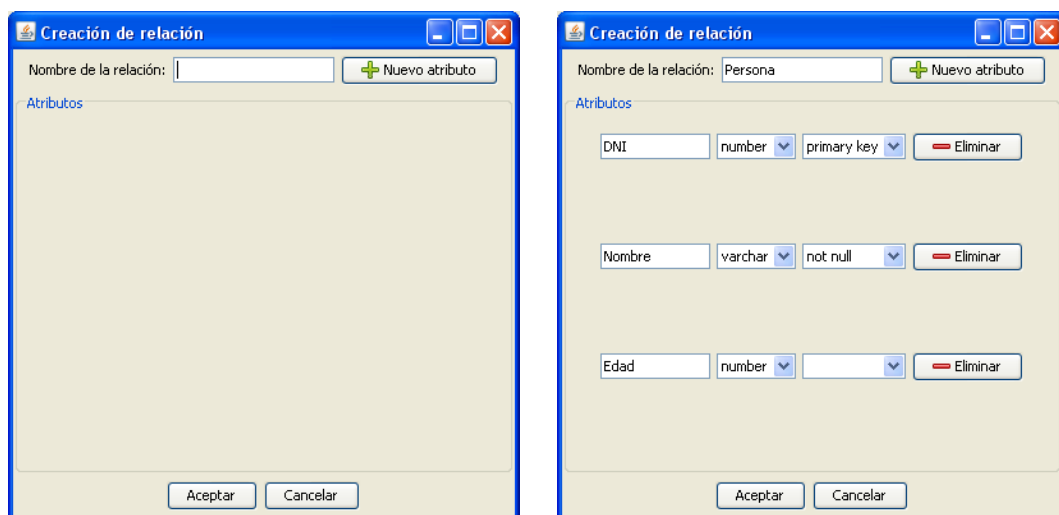


Figura A.12: Creación de una nueva relación

Abrir fichero de relaciones

Tiene la posibilidad, como se ha comentado previamente, de abrir un fichero externo con relaciones y tuplas. Para abrir un fichero haga click en el botón *Abrir relación* tras lo cual se abrirá una ventana que le permitirá seleccionar el fichero que desea abrir (ver Figura A.13). Debe seleccionar entonces un archivo de relaciones (*.rel*); dichos archivos son generados por la aplicación cuando guarde su proyecto. Una vez abierto el fichero se procederá a su lectura generando las relaciones encon-

tradas. Si el archivo no presenta el formato adecuado se mostrará una ventana de error (esto podría ocurrir si modifica el archivo al margen de la aplicación).

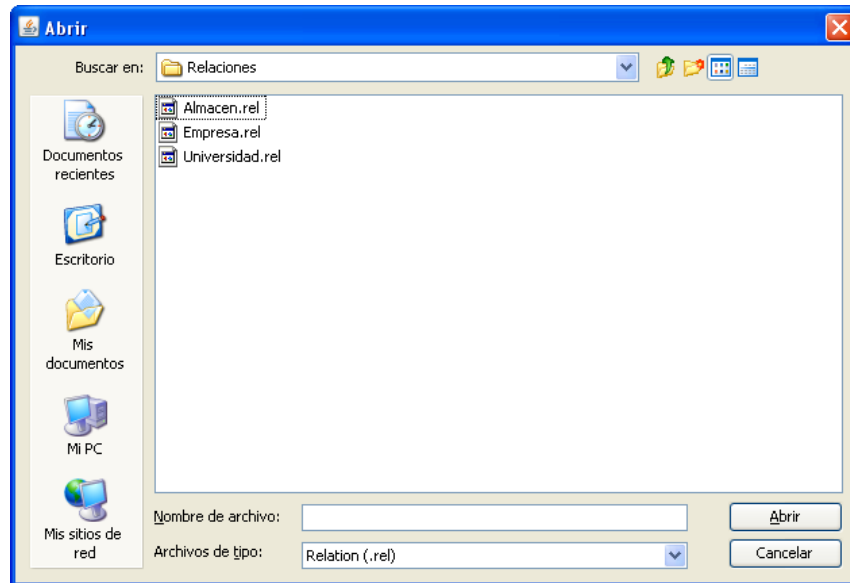


Figura A.13: Selección de fichero para su apertura

Importar relaciones desde una base de datos externa

Si desea importar las relaciones desde una base de datos externa, haga click en el botón *Importar relación*. Es necesario estar conectado a la base de datos para proceder a la importación. Si no está conectado en ese momento se mostrará una ventana avisándole y será necesario que se conecte. Si se encuentra conectado, a continuación se mostrarán todas las relaciones que se encuentran en su base de datos, como se puede ver en el ejemplo de la Figura A.14. Seleccione la relación que desea importar haciendo click en ella (si desea importar más de una relación mantenga pulsada la tecla *ctrl* mientras hace click en las relaciones deseadas). Además del nombre de la relación se puede ver también el número de tuplas de la misma. Dichas tuplas serán importadas junto a la relación.

Exportar relación a base de datos externa

Puede exportar relaciones creadas en la aplicación o abiertas/importadas a su base de datos externa. Para ello es necesario estar conectado a la base de datos. Seleccione en primer lugar la relación a exportar y haga click en el botón *Exportar relación*. A continuación se exportará la relación a su base de datos. Si ocurriera algún problema ajeno a la aplicación en el momento de la exportación (pérdida de conexión a la base de datos, problemas de cuota en la base de datos, problemas

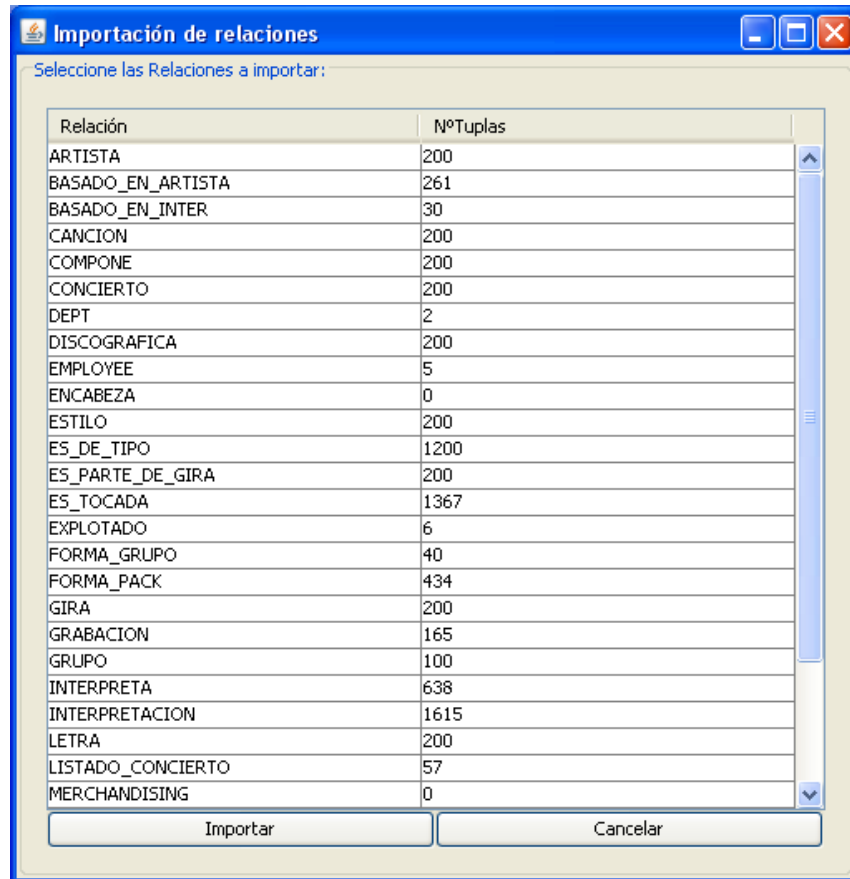


Figura A.14: Ventana de selección de relaciones a importar

de permisos...) será informado a través de una ventana de error en la que podrá ver el código *ORACLE* del error así como el mensaje asociado a dicho código. Si la exportación se realiza correctamente el icono de la relación cambiará para confirmarlo, como se puede ver en el ejemplo de la Figura A.15.

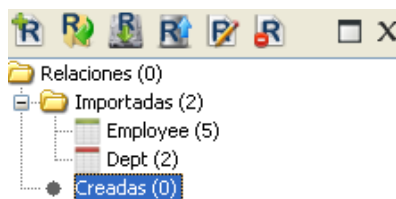


Figura A.15: Ejemplo de relación exportada

Editar una relación

Para editar el nombre de una relación, selecciónela dentro del árbol de relaciones del proyecto y pulse el botón *Editar relación*. Una ventana se le mostrará con el nombre actual de la relación, escriba dentro el nuevo nombre y acepte. Si elige un nombre que ya está siendo utilizado se mostrará un mensaje de error.

Eliminar una relación

Si desea eliminar una relación del proyecto actual, seleccione dicha relación del árbol jerárquico y a continuación pulse el botón *Eliminar relación*. Como en otras operaciones destructivas se le pedirá confirmación a través de una ventana cuando realice este proceso.

Si desea eliminar todas las relaciones creadas (o todas las relaciones importadas) haga click en el submenú del árbol de relaciones actuales y pulse el botón de *Eliminar relación*.

Debe tener alguna relación seleccionada antes de pulsar el botón de *Eliminar relación* o de lo contrario no ocurrirá nada.

A.4.4. Ventana de relación seleccionada

Esta parte de la interfaz se encarga de interactuar con las tuplas que contiene una relación. Se muestran en esta ventana, además de las tuplas, los nombres de los atributos. Esta información se ofrece en forma de tabla cebrada para su mejor legibilidad. El modelo relacional establece que las tuplas de una relación no contienen ningún orden, así que se muestran en orden de creación. Si desea cambiar la ordenación (a ordenación alfabética o numérica) en función del valor de un atributo, haga click en el nombre de dicho atributo. A continuación se explican las diferentes partes que componen esta ventana y sus funcionalidades (ver Figura A.16):

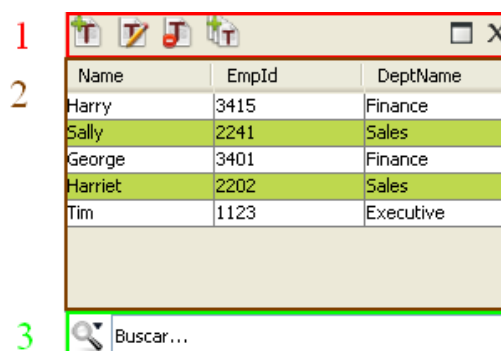


Figura A.16: Ventana de relación seleccionada

1. Barra de herramientas de la ventana de relación seleccionada.
2. Tuplas de la relación.
3. Barra de búsqueda.

Barra de herramientas de la ventana de relación seleccionada

Contiene los siguientes botones de izquierda a derecha (ver Figura A.17):



Figura A.17: Barra de herramientas de la ventana de relación seleccionada

- - *Nueva tupla*: abre la ventana de creación de una nueva tupla para la relación actualmente seleccionada. Esta opción es accesible también el submenú *Archivo*.
- - *Editar tupla*: abre la ventana para la edición de la tupla actualmente seleccionada. Esta opción es accesible también desde el submenú *Editar*.
- - *Eliminar tupla*: elimina la tupla actualmente seleccionada. Antes de ello pide al usuario confirmación. Esta opción es accesible también desde el submenú *Editar*.
- - *Generar tuplas automáticamente*: abre la ventana de generación de tuplas automáticamente permitiendo al usuario seleccionar la cantidad de tuplas así como la base de generación (este proceso se explica más adelante).
- - *Maximizar/minimizar*: maximiza o minimiza la ventana de relaciones, de tal forma que ocupe toda la ventana de la aplicación o sólo una parte.
- - *Cerrar*: cierra la ventana de relaciones y para volver a ser abierta debe utilizarse el submenú *Ver*.

Tuplas de la relación

Muestra las tuplas de la relación actual así como sus atributos. En la Figura A.18 se puede ver un ejemplo del contenido de esta sección para una relación de ejemplo. Haciendo click en el nombre del atributo se ordena la tabla alfabéticamente en función del contenido de ese campo.

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales
Tim	1123	Executive

Figura A.18: Ejemplo de tuplas de una relación

Barra de búsqueda

Le permite buscar en las tuplas de una relación seleccionando el atributo y la cadena a buscar. Esta búsqueda se realiza nada mas teclear la cadena patrón, no es necesario que pulse enter o apriete ningún botón, la búsqueda se realiza conforme vaya escribiendo. Además, este componente contiene los siguientes botones de izquierda a derecha (ver Figura A.19):



Figura A.19: Barra de búsqueda

- - *Selección de atributo*: muestra un desplegable con los atributos de la relación para realizar la búsqueda sobre uno de ellos.
- *Campo de búsqueda*: permite la escritura de una cadena que será buscada en las tuplas de la relación actual a modo de filtro.

Crear una nueva tupla

Para crear una nueva tupla seleccione en primer lugar una relación desde el árbol de relaciones actuales y a continuación haga click en el botón *Crear tupla*. Una vez hecho esto aparecerá un cuadro de dialogo donde se le pedirá que introduzca los valores para los respectivos atributos separados por comas. Si introduce menos o más valores de los necesarios será avisado por medio de un mensaje de error.

Editar una tupla

Si desea editar una tupla seleccione en primer lugar la tupla elegida y pulse el botón *Editar tupla*, tras lo cual aparecerá una ventana con un cuadro de texto en el que se mostrarán los valores actuales de los diferentes atributos. Edite estos valores

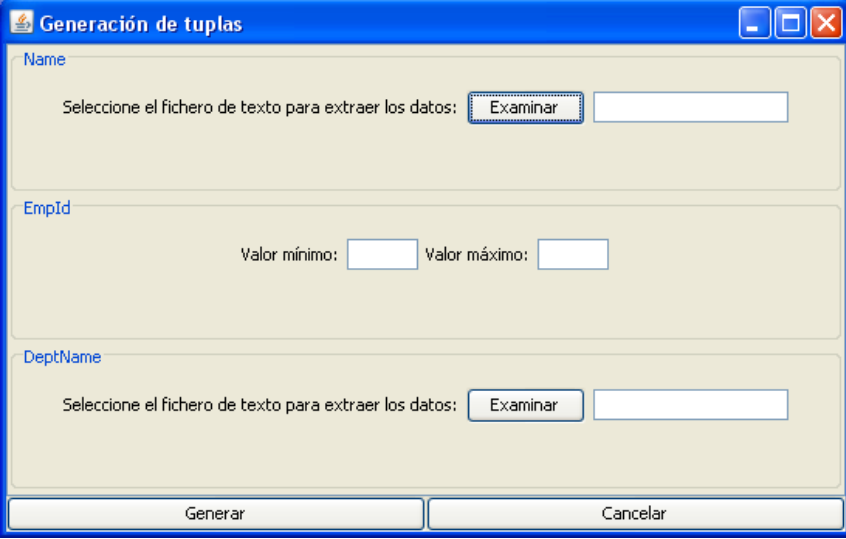
y a continuación haga click en *Aceptar*, si introduce menos o mas valores de los necesarios será avisado por medio de un mensaje de error. También es posible editar un valor de una tupla haciendo click en la casilla que contiene dicho valor en la tabla y modificándolo allí mismo.

Eliminar una tupla

Al igual que con las relaciones, el proceso para eliminar una tupla es, en primer lugar, seleccionar la tupla deseada, después pulsar el botón *Eliminar tupla* y por último se requerirá confirmación por ser una operación destructiva, acepte y la tupla será eliminada de la relación. Si elimina una tupla se actualizará la tabla de las tuplas de la relación seleccionada así como la etiqueta de la relación en el árbol, decrementando su número de tuplas.

Generar automáticamente tuplas

Está incluida en la aplicación la opción de generar automáticamente tuplas para las relaciones del proyecto actual. Esta opción le permitirá mediante un proceso automático o guiado, generar el número de tuplas que desee. Para generar las tuplas seleccione en primer lugar la relación, a continuación haga click en el botón *Generar tuplas*. Ahora se mostrará una ventana en la que puede introducir rangos de valores para atributos numéricos o ficheros externos para atributos de tipo cadena (puede ver un ejemplo en la Figura A.20). Si deja vacía alguna de las opciones la aplicación se encargará de crear un patrón para esos atributos. Cuando pulse el botón *Aceptar* se le pedirá introducir el número de tuplas que quiere que se generen.



The image shows a Windows-style dialog box titled "Generación de tuplas". It has a blue title bar with standard window controls. The dialog is divided into three main sections, each with a blue header label: "Name", "EmpId", and "DeptName". The "Name" section contains the text "Seleccione el fichero de texto para extraer los datos:", an "Examinar" button, and an empty text input field. The "EmpId" section contains "Valor mínimo:" and "Valor máximo:" labels, each followed by an empty text input field. The "DeptName" section contains the same text label "Seleccione el fichero de texto para extraer los datos:", an "Examinar" button, and an empty text input field. At the bottom of the dialog are two buttons: "Generar" and "Cancelar".

Figura A.20: Generación automática de tuplas

A.4.5. Ventana de expresiones en álgebra relacional

La parte más importante de la aplicación es la de definición y modificación de consultas en álgebra relacional y árboles de consultas en álgebra relacional. En esta ventana es donde se realiza ese trabajo. Aquí podrá abrir sus consultas desde ficheros externos, escribir nuevas consultas introduciendo los operadores del álgebra relacional a través de la barra de operadores, modificar los árboles asociados a las consultas, etc. La estructura de esta ventana varía un poco si estamos en el panel de creación de expresiones o en el de creación de árboles; se puede ver en la Figura A.21 la ventana de creación de expresiones y en la Figura A.22 la de creación de árboles. A continuación se explican sus componentes:

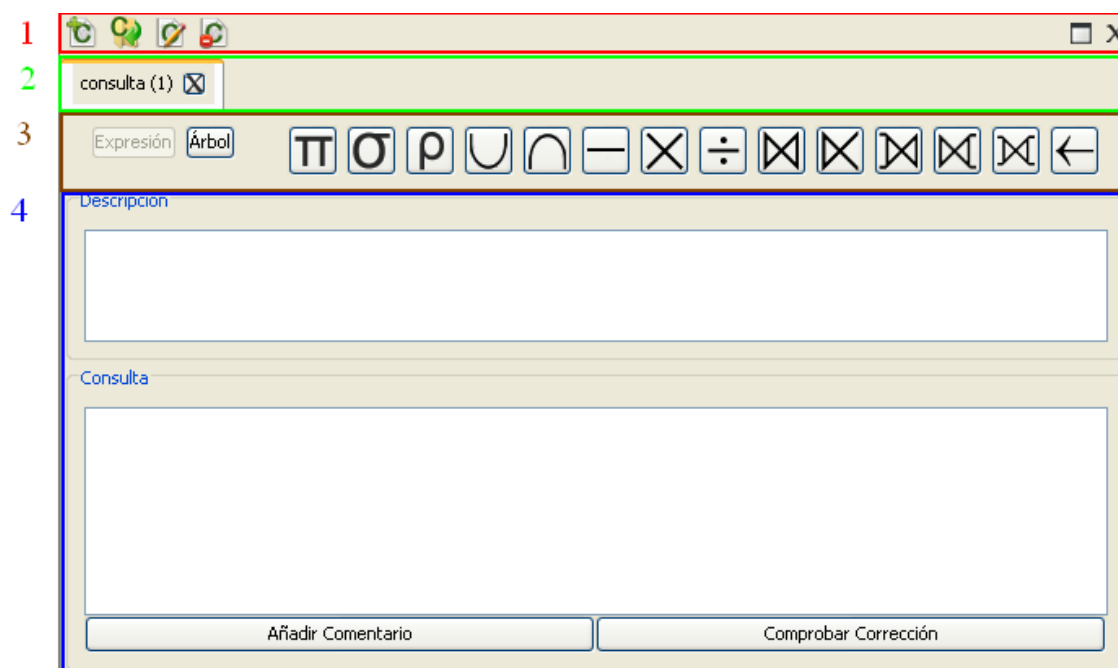


Figura A.21: Ventana de expresiones en álgebra relacional

1. Barra de herramientas de la ventana de expresiones en álgebra relacional.
2. Pestañas con las diferentes consultas.
3. Barra de operadores.
4. Panel de expresión en álgebra relacional / árbol de expresión en álgebra relacional.
5. Barra de opciones de árbol (sólo disponible en modo de edición de árboles).

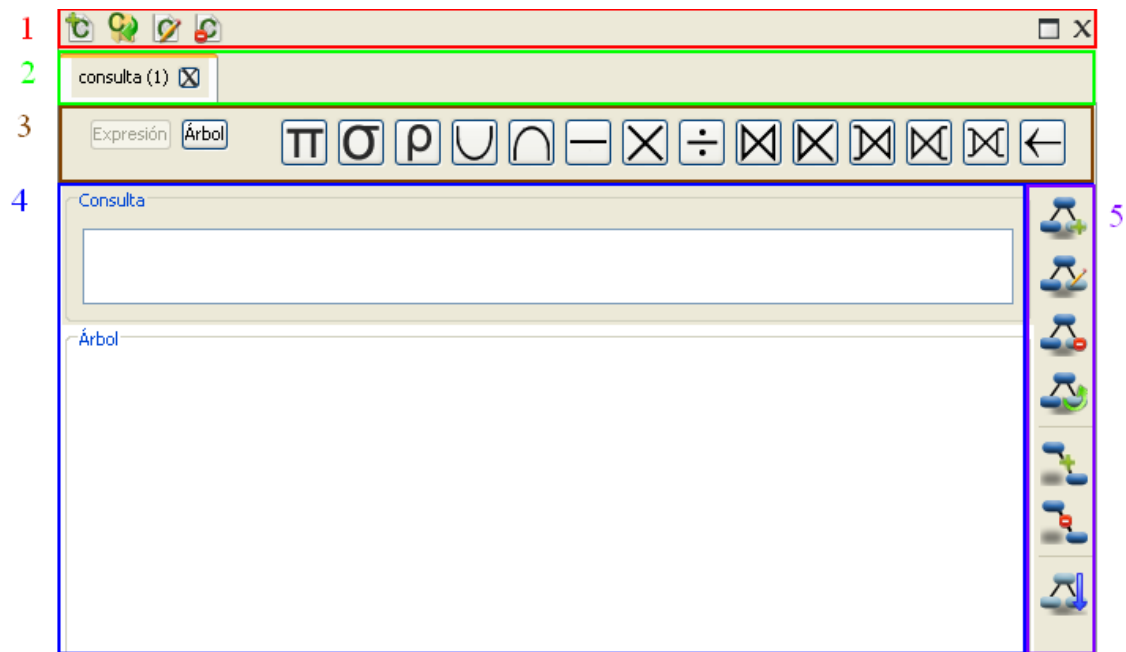


Figura A.22: Ventana de árbol de expresión en álgebra relacional

Barra de herramientas de la ventana de expresiones en álgebra relacional

Contiene los siguientes botones de izquierda a derecha (ver Figura A.23):



Figura A.23: Barra de herramientas de la ventana de expresiones en álgebra relacional

- - *Nueva consulta*: crea una nueva pestaña donde podrá definir una nueva consulta. Esta opción es accesible también desde el submenú *Archivo*.
- - *Abrir consulta*: abre la ventana para la selección de un archivo XML que contenga una o más consultas. Esta opción es accesible también desde el submenú *Archivo*.
- - *Editar consulta*: abre la ventana para la edición de la consulta actualmente seleccionada. Esta opción es accesible también desde el submenú *Editar*.
- - *Eliminar consulta*: elimina la consulta actualmente seleccionada. Antes de ello pide al usuario confirmación. Esta opción es accesible también desde el submenú *Editar*.
- - *Maximizar/minimizar*: maximiza o minimiza la ventana de relaciones, de tal forma que ocupe toda la ventana de la aplicación o sólo una parte.

- **x- Cerrar:** cierra la ventana de relaciones y para volver a ser abierta debe utilizarse el submenú *Ver*.

Pestañas con las diferentes consultas

En estas pestañas encontrará las consultas disponibles en el proyecto. Puede navegar libremente entre ellas y están identificadas por el nombre de la consulta. Además, puede cerrar una consulta haciendo click en el botón de cerrar (indicado con una *X*) que aparece al lado del nombre de dicha consulta. En la Figura A.24 se puede ver un ejemplo con varias pestañas.

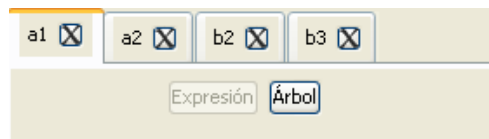


Figura A.24: Ejemplo de varias consultas abiertas

Barra de operadores

Debido a que el álgebra relacional utiliza para denotar sus operadores símbolos especiales, se ofrece la posibilidad de añadir un nuevo operador a su consulta actual de una forma gráfica a través de la barra de operadores. Dicha barra le permite también alternar entre la visión del panel de expresiones y el de árboles. Este cambio entre el modo de edición de expresiones en modo textual y el modo de edición de árboles se realiza de forma dinámica. A continuación se explican sus componentes (ver Figura A.25).

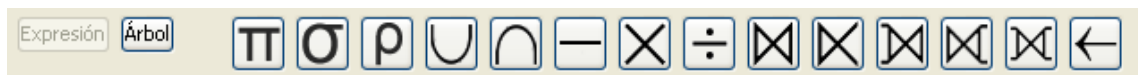


Figura A.25: Barra de operadores

- *Expresión:* permite cambiar al panel de expresiones en álgebra relacional. Aparece deshabilitado cuando ya estamos en ese panel.
- *Árbol:* permite cambiar al panel de árbol de expresión en álgebra relacional. Aparece deshabilitado cuando ya estamos en él.
- **Π**- *Proyección:* añade el operador proyección a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- **σ**- *Selección:* añade el operador selección a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).

- ρ - *Renombre*: añade el operador renombre a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\cup$ - *Unión*: añade el operador unión a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\cap$ - *Intersección*: añade el operador intersección a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $-$ - *Diferencia*: añade el operador diferencia a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\times$ - *Producto cartesiano*: añade el operador producto cartesiano a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\div$ - *División*: añade el operador división a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\bowtie$ - *Join*: añade el operador join a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\ltimes$ - *Semijoin*: añade el operador semijoin a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\ltimes\!\!\!\bowtie$ - *Left outer join*: añade el operador left outer join a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\bowtie\!\!\!\ltimes$ - *Right outer join*: añade el operador right outer join a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\bowtie\!\!\!\bowtie$ - *Full outer join*: añade el operador full outer join a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).
- $\leftarrow$ - *Asignación*: añade el operador asignación a la consulta actual. Ya sea como nodo (en el panel de árbol) o textual (en el panel de expresión).

Agregar un nuevo operador

Para agregar un nuevo operador, en función del panel en el que se encuentre, siga los siguientes pasos:

- Panel de definición de *expresiones*: haga click en el botón correspondiente al operador seleccionado y este se incluirá a su expresión en álgebra relacional en la posición del texto en la que se encuentre el cursor.

- **Panel de definición de árboles:** dispone de dos formas de introducir un nuevo operador en el árbol actual, la primera de ellas es hacer click en el botón correspondiente al operador, esperar a que el borde de dicho botón cambie a color verde (indicando que va a añadir un nuevo operador) y mover el ratón hasta la posición en la que desea añadir el operador, a continuación haga click en esa posición para confirmarla. Si desea añadir varios operadores del mismo tipo, es posible utilizar la segunda forma para mayor comodidad. Esta forma consiste en hacer doble click en el botón correspondiente al operador seleccionado, con lo que el borde del botón cambiará a color rojo (indicando que va a introducir varios operadores del mismo tipo) y a continuación hacer click en la posición donde queramos introducir el nodo, si queremos introducir otro más simplemente haga click en la siguiente posición donde se añadirá el operador (puede ver un ejemplo en la Figura A.26).

Si desea deseleccionar un nodo puede hacer click en el mismo o hacer click en otro.

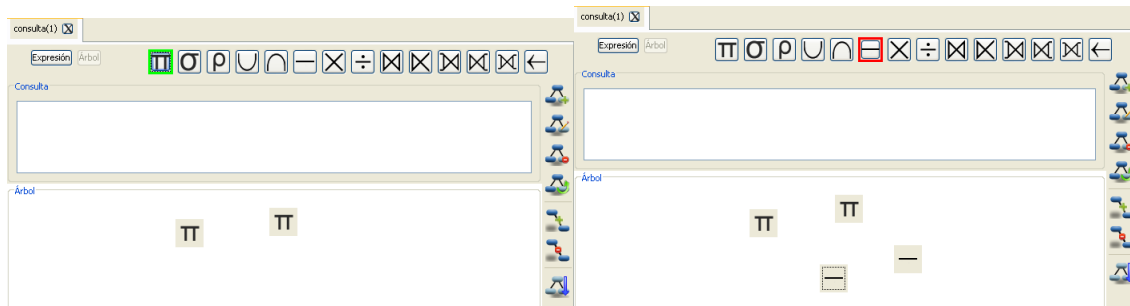


Figura A.26: Agregando un nuevo operador al árbol

Panel de expresiones en álgebra relacional

Aquí podrá definir y manipular sus consultas en álgebra relacional, como se puede ver en el ejemplo de la Figura A.27, consta de dos partes diferenciadas: la primera de ellas es el campo donde puede escribir la descripción de la consulta, y la segunda es donde escribirá la consulta propiamente dicha.

Introducir una consulta consiste en ir agregando operadores del álgebra relacional y texto. Dispone en la parte inferior de dos botones:

- **Añadir comentario:** utilizando este botón podrá agregar comentarios textuales a sus consultas. Estos comentarios resultan de gran utilidad cuando se definen consultas complejas utilizando pasos intermedios. Para agregar un nuevo comentario sitúese en la línea en la que desea insertarlo y pulse el botón. De esta forma se mostrará una ventana en la que se le permitirá introducir el texto que desee y el comentario quedará añadido al final de la línea seleccionada. Si

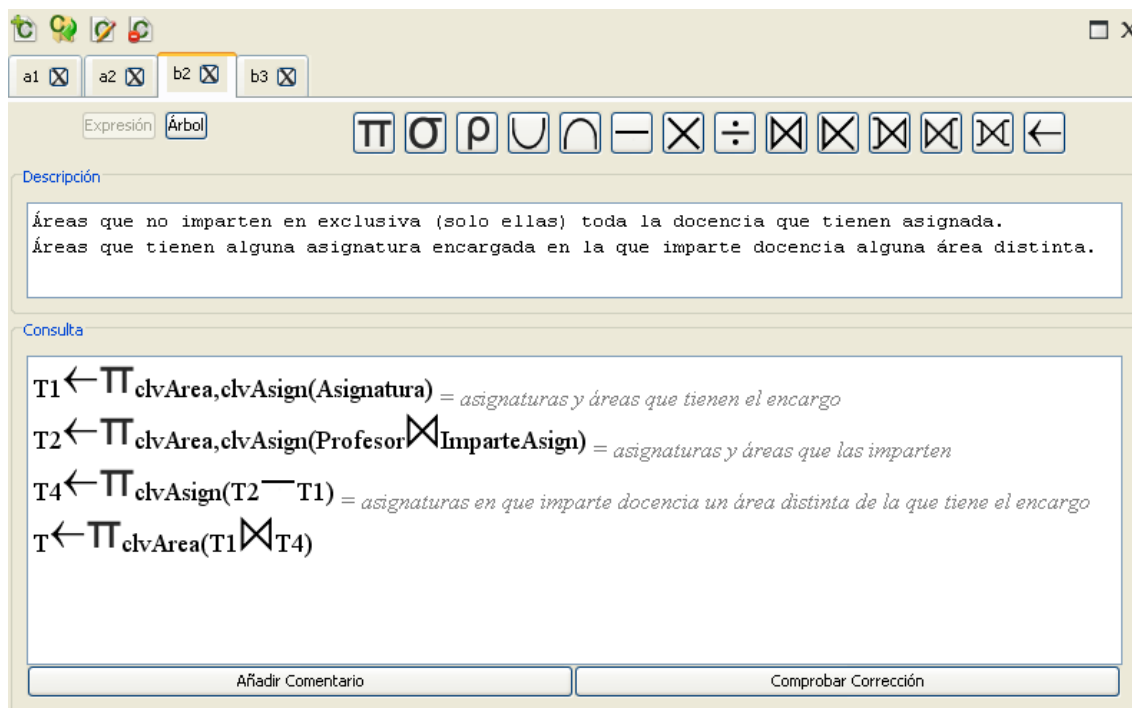


Figura A.27: Ejemplo de expresión en álgebra relacional

desea editar un comentario ya introducido haga click sobre él y se mostrará una ventana de edición. Para borrar un comentario simplemente haga click sobre él y borre todo el texto que contiene.

- *Comprobar corrección:* este botón le permitirá comprobar en todo momento la corrección sintáctica de la consulta que ha introducido. Cuando pulse el botón se procederá a analizar la consulta y si hay algún problema se le mostrará una ventana avisándole de él. Recuerde que las consultas introducidas deben respetar las reglas sintácticas del álgebra relacional explicadas en la asignatura de *Ficheros y Bases de Datos*.

Panel de árboles de expresiones en álgebra relacional

Desde este panel se le permitirá la definición y manipulación de árboles asociados a expresiones en álgebra relacional. En el ejemplo de la Figura A.28 puede observar el aspecto de dicho panel y su división en secciones.

Esta parte de la interfaz esta dividida en tres bloques. El primero de ellos es el campo de texto donde puede ver la expresión en álgebra relacional cuyo árbol está definiendo. Este campo es informativo y no es posible su edición; si desea editar la consulta en modo textual cambie al panel de expresiones en álgebra relacional pulsando el botón *Expresión* de la barra de operadores.

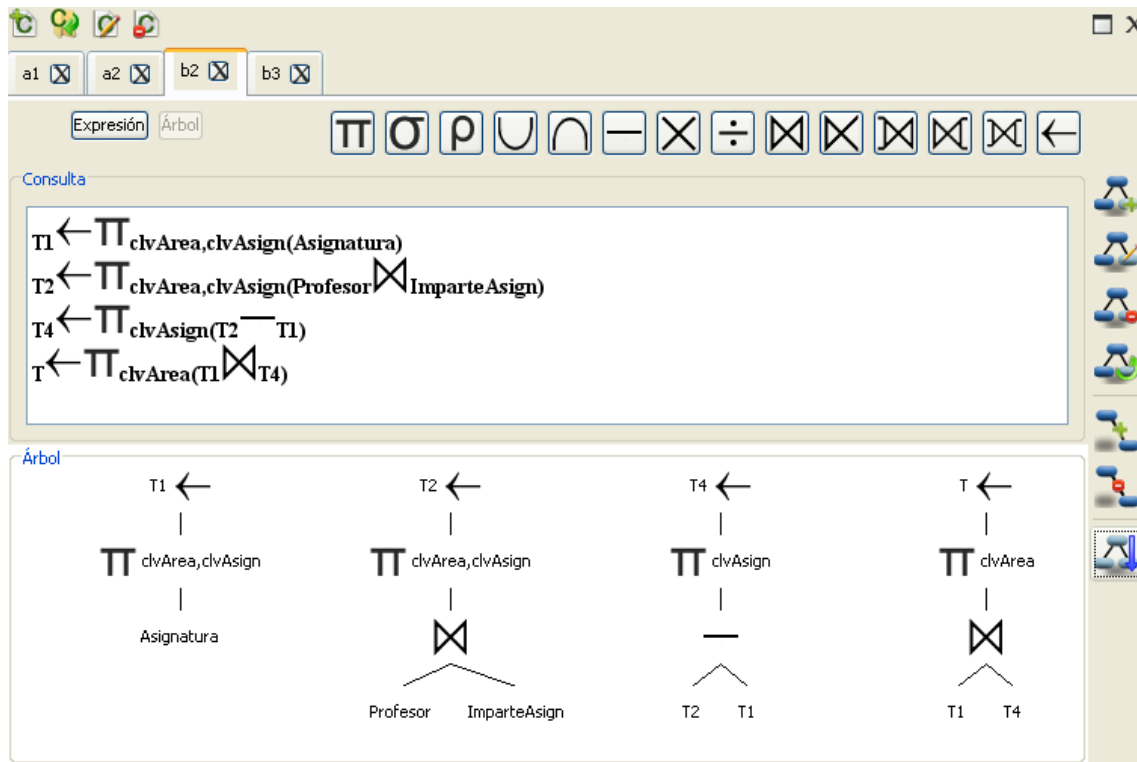


Figura A.28: Ejemplo de árbol

El siguiente bloque es el espacio de definición de los árboles. En él podrá ver el árbol que está definiendo y añadirle nuevos nodos. Este espacio le permite trabajar como si de una hoja de papel se tratase, de tal forma que puede introducir los nodos, moverlos, editarlos, etc.

A la izquierda del panel puede ver el último bloque de esta sección. Se trata de una nueva barra de herramientas con las operaciones que se pueden realizar sobre el árbol (en la Figura A.29 se puede ver en detalle esta barra). Consta de los siguientes elementos:

- *Añadir nuevo nodo*: le permite añadir un nuevo nodo (una nueva relación) al árbol actual. Se le pedirá un nombre para la relación y esta aparecerá en la posición inicial del panel esperando a que la mueva.
- *Editar nodo*: permite editar el contenido del nodo seleccionado. Nótese que solamente son editables los nodos del tipo:
 - *Relación*
 - *Proyección*
 - *Selección*
 - *Renombre*



Figura A.29: Barra de edición de árbol

- *Asignación*
 - *Eliminar nodo*: seleccione el nodo o nodos que desea eliminar y haga click en este botón para eliminarlos. Se consultará antes de realizar la operación destructiva.
 - *Mover nodo*: seleccione el nodo que desea mover y haga click en este icono, a continuación desplácese por el panel en buscar del lugar apropiado y haga click en esa posición para confirmarla como nueva posición del nodo.
 - *Crear conexión*: esta opción le permite conectar dos nodos del árbol. Para ello haga click en este botón y a continuación seleccione dos nodos. El primero de ellos será el nodo padre y el segundo el nodo hijo.
 - *Eliminar conexión*: para eliminar la conexión existente entre dos nodos haga click en este botón y después seleccione los nodos conectados.
 - *Ordenar árbol*: esta opción le permite ordenar el árbol automáticamente de forma que quede centrado y nivelado. Esta opción es muy útil cuando se trate de consultas complejas ya que le ayudará a maximizar el espacio disponible (en la Figura A.30 se puede ver un ejemplo de la ordenación de tal forma que el árbol de la parte superior se ordena y se muestra el resultado en el de abajo).

A.4.6. Ejecución de consultas

Como se ha explicado anteriormente hay dos formas de ejecutar las consultas, utilizando una base de datos externa (para lo cual necesita disponer de conexión a la misma) o ejecutarlas directamente en la aplicación. Vamos a explicar cada una de ellas un poco más en detalle.

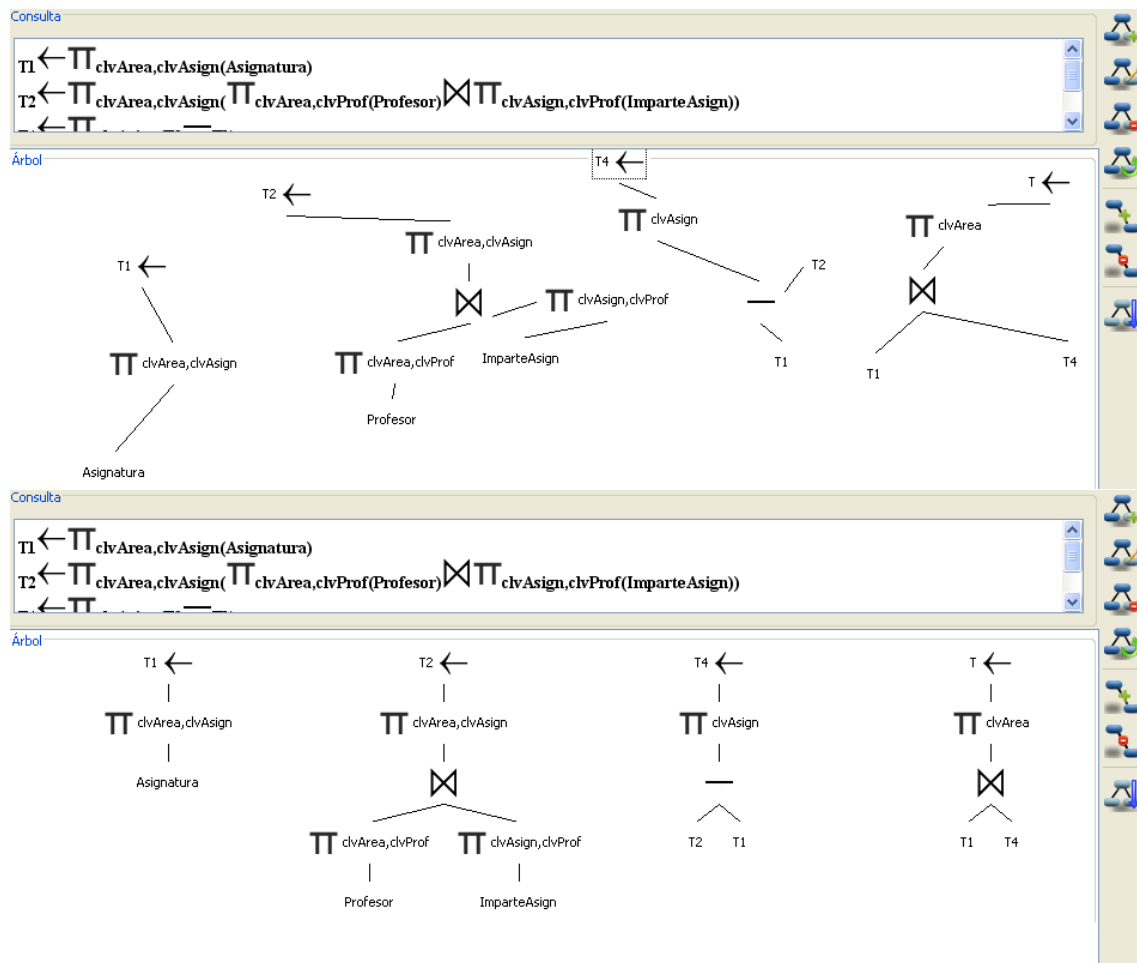


Figura A.30: Ordenación de un árbol automáticamente

Ejecución en la aplicación

Una vez definida su consulta (ya sea mediante la edición en modo expresión o en modo árbol) puede probarla sobre las relaciones de ejemplo que haya introducido. La ejecución puede ser automática o paso a paso. Si selecciona la ejecución automática (por medio de los botones explicados en las secciones anteriores), la consulta se evaluará y aparecerá una ventana donde podrá comprobar los resultados. En la Figura A.31 puede ver un ejemplo de la ejecución de una consulta sobre las relaciones previamente introducidas.

En la ventana que aparece con los resultados, se le mostrará en primer lugar la consulta ejecutada y después una tabla representando la relación resultado. Al igual que en el panel de la relación seleccionada, la tabla que se muestra aquí puede ser ordenada en función de sus atributos y también puede buscar tuplas a través del campo de búsqueda inferior.

Si ejecuta su consulta paso a paso, la ventana que se mostrará será similar a la

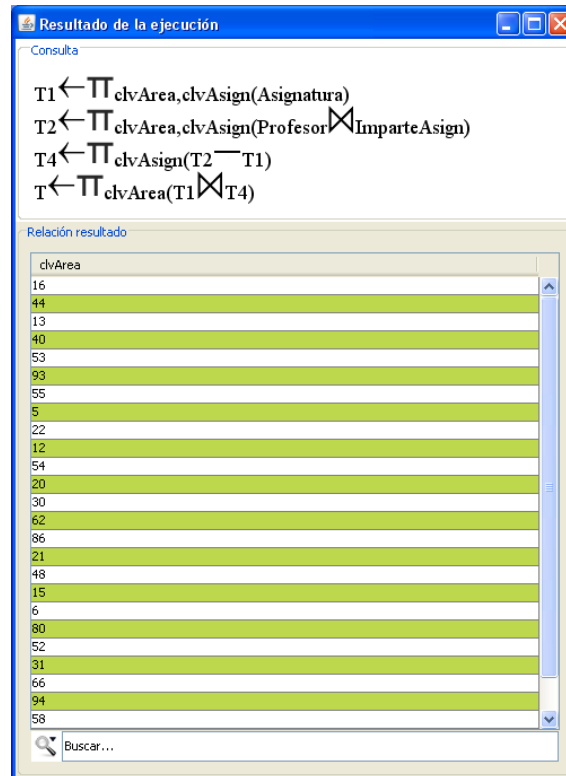


Figura A.31: Ejecución de una consulta en la aplicación

anterior, solo que en esta dispondrá de los resultados parciales que la consulta haya ido generando en cada uno de sus pasos (en la Figura A.32 se puede ver un ejemplo).

Como puede observar en el ejemplo, cada uno de los pasos se muestra en una pestaña en la que aparece la parte de la consulta ejecutada en ese paso (nótese que la operación que se está realizando en ese instante aparece marcada en otro color), la tabla con los resultados de esa consulta parcial, y abajo dos botones que le permitirán moverse de una manera rápida entre cada uno de los pasos.

Tanto en la ejecución automática como en la ejecución paso a paso, la creación de las ventanas depende de la correcta ejecución de la consulta. Antes de ser ejecutada se comprobará que su consulta sea sintácticamente correcta, y si durante la ejecución se encontrara algún problema se le avisará por medio de una ventana de error para que lo corrija.

Debe tener en cuenta que el tiempo de ejecución de las consultas depende de la cantidad de tuplas así como de los operadores que haya introducido, así, si la consulta tiene productos cartesianos o join de relaciones con muchas tuplas, la gran cantidad de operaciones que deberá realizar la aplicación puede ralentizar el tiempo de ejecución.



Figura A.32: Ejecución de una consulta paso a paso en la aplicación

Ejecución en la base de datos externa

Como ya se ha dicho, es requisito imprescindible para la ejecución de una consulta en la base de datos externa que disponga de conexión en el momento de la ejecución con dicha base de datos. La aplicación se encargará de convertir la consulta que usted haya introducido en álgebra relacional a SQL para poder lanzarla sobre la base de datos. Recuerde que las relaciones sobre las que la consulta actúa deben estar presentes en la base de datos y que dispone de la opción de exportar relaciones. Cuando lance una consulta a la base de datos, si ocurriera algún problema será informado mediante el código de error del mismo y el mensaje asociado. Tenga en cuenta que si la base de datos externa es accedida concurrentemente por varias

personas puede ser que no obtenga los mismos resultados ejecutando una consulta en la base de datos externa o en la aplicación.

Una vez que la base de datos ejecute la consulta, el resultado de la misma será mostrado de forma análoga a los resultados mostrados en la ejecución sobre la aplicación. Debe tener en cuenta que dependiendo de la conexión a la base de datos externa este proceso puede resultar mas costoso que la ejecución en la propia aplicación.

A.5. Optimización de consultas

Las consultas que usted haya introducido pueden ser optimizadas gracias a una funcionalidad de la aplicación. Recuerde que la optimización no tiene porque producir el árbol mas óptimo puesto que para consultas complejas las posibilidades crecen en gran medida y puesto que se tienen que realizar estimaciones en cuanto al número de tuplas en cada paso (esto sólo se sabrá en tiempo de ejecución). Con estas ideas en mente, usted puede optimizar sus consultas haciendo click en el botón de *Optimización* o a través de la opción análoga del menú *Herramientas*. Al igual que con la ejecución dispone de dos modos de optimización, automática o paso a paso. La optimización automática aplica al árbol de su consulta actual una serie de reglas de optimización de tal forma que su árbol se verá modificado para mejorar su tiempo de ejecución. Si selecciona la opción de optimizar paso a paso. Se abrirá una nueva ventana, similar a la de ejecución paso a paso, en la que podrá ver cada una de las acciones que se han desarrollado. En la Figura A.33 puede ver un ejemplo de optimización paso a paso.

En la ventana de optimización paso a paso se encuentran las siguientes secciones:

- *Explicación*: aquí tendrá una breve explicación de la regla de optimización que se ha utilizado durante ese paso.
- *Árbol anterior*: aquí podrá ver una imagen del estado del árbol antes de aplicar la regla de optimización.
- *Árbol actual*: esta imagen muestra el resultado del árbol una vez aplicada la regla de optimización.

Además dispone de botones en la parte inferior para moverse cómodamente por los diferentes pasos, aunque también puede moverse seleccionando alguna de las pestañas que se muestran en la parte superior.

Como ya se ha explicado, para la realización de la optimización se tienen en cuenta estimaciones del número de tuplas que cada operación produce. Con objetivo didáctico se ha añadido la opción de seleccionar para cada una de las operaciones si deseamos que se haga la estimación pesimista (teniendo en cuenta que la operación

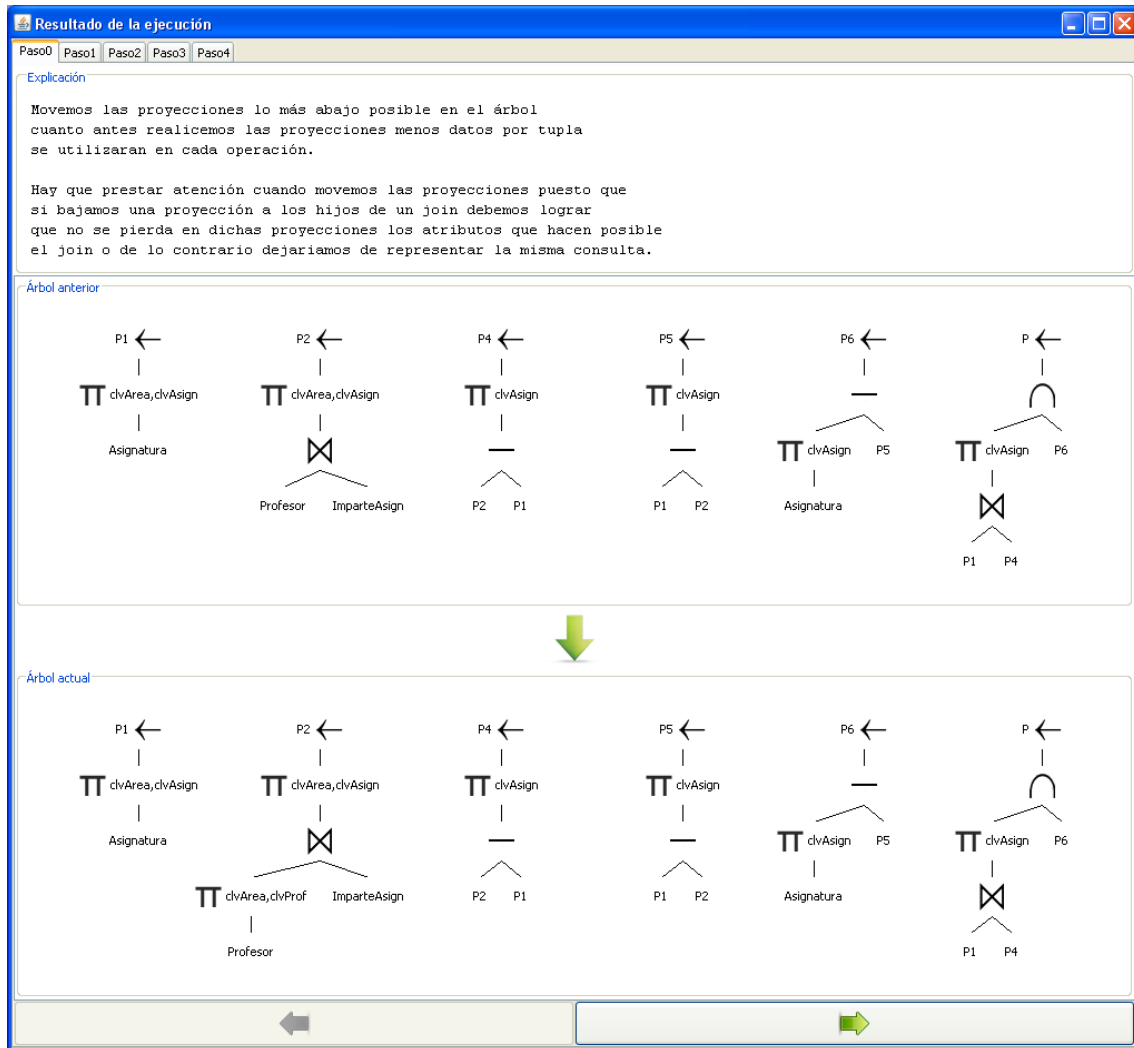


Figura A.33: Optimización paso a paso de una consulta

produce el máximo número de tuplas posible) o podemos introducir el porcentaje de selección, en cuyo caso sólo un tanto por cierto de las tuplas máximas serán tomadas como estimación. Esta opción permite al usuario comprobar los cambios que se pueden producir en la optimización dependiendo de número de tuplas que contenga cada relación o de la selectividad de cada operación. Para seleccionar el comportamiento para cada operador utilice la vista de árbol y haga click con el botón derecho del ratón sobre la relación seleccionada, un menú se desplegará en cuyas opciones dispondrá de lo comentado previamente (ver Figura A.34).

Esta opción solo esta disponible en el caso de nodos que sean operadores. Si selecciona un nodo que sea una relación, la opción anterior será sustituida por la posibilidad de suponer la cantidad de tuplas que contiene esa relación (ver Figura A.35).

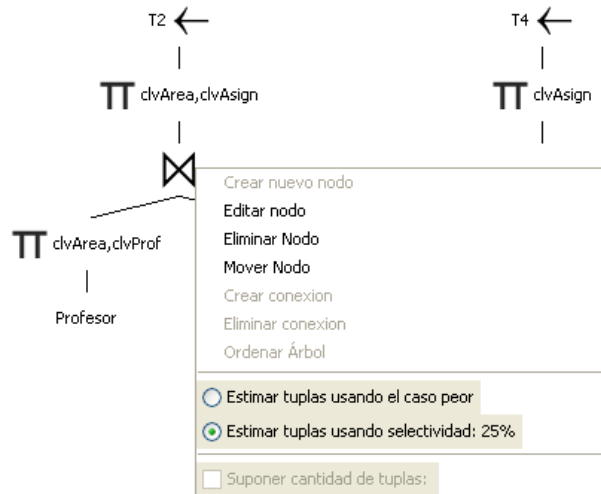


Figura A.34: Opciones para la estimación del número de tuplas



Figura A.35: Opciones para la suposición del número de tuplas

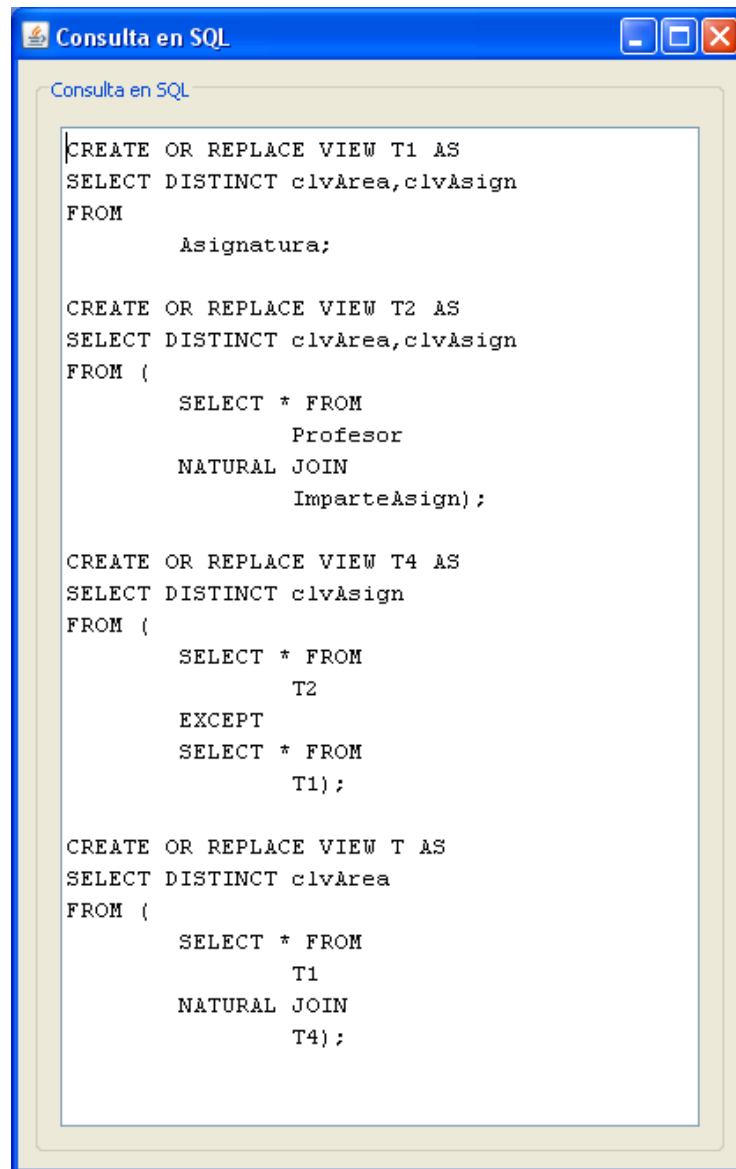
A.6. Traducción a SQL

Otra funcionalidad de la aplicación es la traducción automática de consultas en álgebra relacional al SQL. Si desea traducir una consulta que haya introducido no tiene más que hacer click en el botón de la barra general o en el menú correspondiente. La traducción se realizará a continuación y cuando esta termine se le mostrará en una nueva ventana su consulta en SQL en un cuadro de texto del que podrá copiarla si desea. Utilizando como base la consulta de la Figura A.36, en el ejemplo de la Figura A.37 se puede observar el resultado de su traducción a SQL.

Si ha creado asignaciones en la consulta que ha definido debido a que estaba realizando una consulta compleja mediante pasos intermedios (como ocurre en el ejemplo), se creará una vista por cada asignación encontrada. De esta forma se facilita la comprensión de código SQL y le permitirá ver fácilmente la relación existente

$T1 \leftarrow \Pi_{\text{clvArea}, \text{clvAsign}}(\text{Asignatura}) = \text{asignaturas y áreas que tienen el encargo}$
 $T2 \leftarrow \Pi_{\text{clvArea}, \text{clvAsign}}(\text{Profesor} \bowtie \text{ImparteAsign}) = \text{asignaturas y áreas que las imparten}$
 $T4 \leftarrow \Pi_{\text{clvAsign}}(T2 - T1) = \text{asignaturas en que imparte docencia un área distinta de la que tiene el encargo}$
 $T \leftarrow \Pi_{\text{clvArea}}(T1 \bowtie T4)$

Figura A.36: Consulta en álgebra relacional para su traducción a SQL



```

CREATE OR REPLACE VIEW T1 AS
SELECT DISTINCT clvArea, clvAsign
FROM
    Asignatura;

CREATE OR REPLACE VIEW T2 AS
SELECT DISTINCT clvArea, clvAsign
FROM (
    SELECT * FROM
        Profesor
    NATURAL JOIN
        ImparteAsign);

CREATE OR REPLACE VIEW T4 AS
SELECT DISTINCT clvAsign
FROM (
    SELECT * FROM
        T2
    EXCEPT
    SELECT * FROM
        T1);

CREATE OR REPLACE VIEW T AS
SELECT DISTINCT clvArea
FROM (
    SELECT * FROM
        T1
    NATURAL JOIN
        T4);

```

Figura A.37: Traducción a SQL de una consulta

entre el álgebra relacional y SQL.

A.7. Configuración de la aplicación

La aplicación permite la configuración de ciertas opciones. Para acceder a la ventana de configuración basta con hacer click en *Configuración* en el menú *Herramientas*. Esta ventana esta compuesta por tres pestañas, la primera de ellas (ver Figura A.38) es la encargada de mostrarle opciones generales. A continuación se explican sus partes:

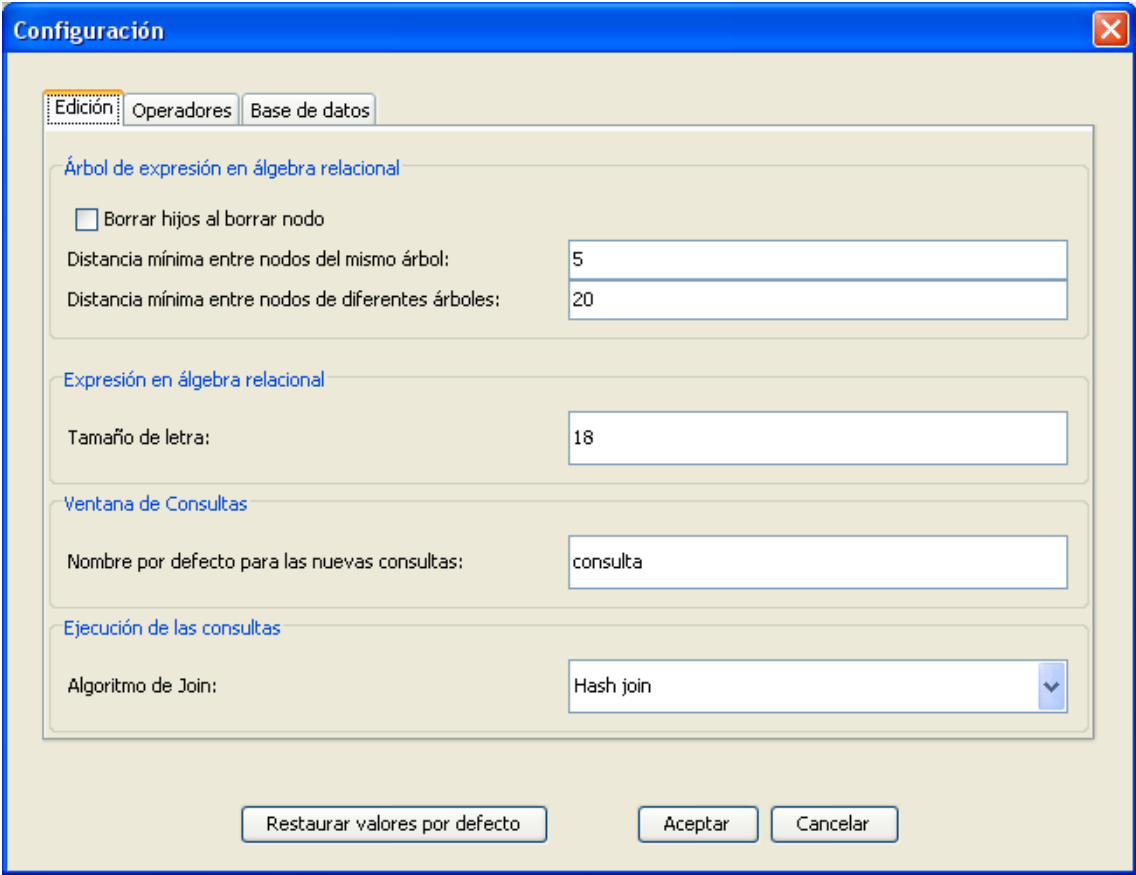


Figura A.38: Configuración pestaña *Edición*

- *Borrar hijos al borrar nodo*: como su propio nombre indica, si selecciona esta opción, cada vez que borre un nodo de un árbol se borrarán todos sus hijos. Si no selecciona la opción solamente se borrará el nodo actual y sus hijos pasarán a ser hijos del padre del nodo eliminado. Por defecto esta opción está desactivada.
- *Distancia mínima entre nodos del mismo árbol*: cuando haga click en el botón de *Ordenar árbol* se organizará automáticamente el árbol en función al espacio

disponible en pantalla para mostrarse de la mejor forma. El valor que establezca en este campo determinará la distancia mínima a la que se encontrarán los nodos de un mismo árbol. El valor por defecto de esta opción es 5 px.

- *Distancia mínima entre nodos de diferentes arboles*: al igual que la opción anterior, el valor de este campo determinará la distancia mínima a la que se situarán los nodos de distintos árboles pero de la misma profundidad. El valor por defecto de esta opción es 20 px.
- *Tamaño de letra*: aquí puede establecer el tamaño de la fuente utilizada en la ventana de expresiones en álgebra relacional, tenga en cuenta que los símbolos de los operadores son imágenes de un tamaño prefijado y por lo tanto solo tendrá control sobre la parte escrita por usted. El tamaño por defecto de la letra es de 18 pt.
- *Nombre por defecto para las nuevas consultas*: cada vez que crea una nueva consulta se le da un nombre por defecto que puede editar cuando le plazca, este nombre puede ser cambiado desde esta opción. El nombre por defecto es *consulta(x)* (donde x es el número de consultas realizadas en ese proyecto).
- *Algoritmo de Join*: a la hora de ejecutar una consulta con operaciones de Join, esta opción determinará el algoritmo que se utilizará para su realización, tenga en cuenta que el tiempo de ejecución dependerá (además de las tuplas que tengan las relaciones) del algoritmo de Join elegido. Por defecto aparece seleccionado el algoritmo de *Hash join*.

La siguiente pestaña tiene que ver con los operadores del álgebra relacional y puede ser consultada en la Figura A.39. Esta pestaña le permite editar la apariencia que van a tener los operadores en el árbol, puede elegir el color para cada uno de los operadores y esto se aplicará tanto a los nuevos operadores que introduzca como a los que ya había introducido.

Por último, se encuentra la pestaña de *Base de datos*, en ella podrá introducir varios parámetros utilizados en la conexión a la base de datos externa, entre ellos el usuario y password que desea utilizar (para no tener que estar metiéndolo todo el rato) y la URL de la base de datos (por si esta fuera modificada). En la Figura A.40 se puede ver esta pestaña.

Recuerde que para aplicar los cambios seleccionados debe hacer click en *Aceptar* y que si quiere volver a los valores por defecto puede hacer click en el botón *Restaurar valores por defecto*.

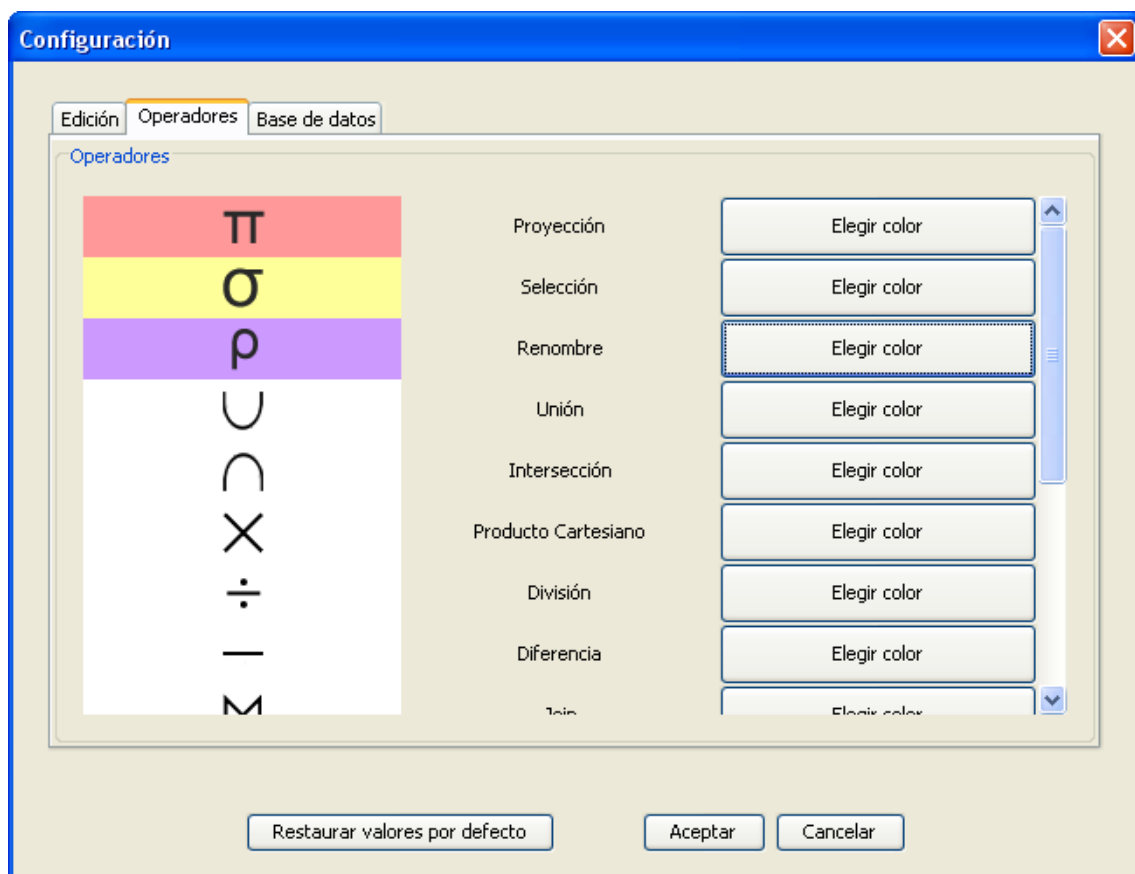


Figura A.39: Configuración pestaña *Operadores*

The image shows a Java Swing window titled "Configuración" with a standard Mac OS X title bar (blue with a red close button). The window has three tabs: "Edición", "Operadores", and "Base de datos", with the last one being the active tab. Inside the window, there is a section titled "Parámetros de conexión" containing three text input fields. The first field is labeled "Usuario:" and contains the text "user". The second field is labeled "Password:" and contains the text "pass". The third field is labeled "URL:" and contains the text "jdbc:oracle:thin:@hendrix-oracle.cps.unizar.es:1521:vicio". At the bottom of the window, there are three buttons: "Restaurar valores por defecto", "Aceptar", and "Cancelar".

Label	Value
Usuario:	user
Password:	pass
URL:	jdbc:oracle:thin:@hendrix-oracle.cps.unizar.es:1521:vicio

Figura A.40: Configuración pestaña *Base de datos*

Apéndice B

Conceptos del álgebra relacional

En este anexo se presenta un resumen del estudio realizado para la comprensión del tema a tratar en este proyecto. Se hablará acerca del modelo relacional, los diferentes operadores del álgebra relacional, así como de la optimización de consultas.

B.1. El modelo relacional

Para ser capaces de comprender con exactitud el álgebra relacional es necesario realizar un repaso acerca de modelo relacional.

Antes de definir una relación necesitamos definir en primer lugar otros dos términos, dominio y atributo. Cada atributo de una relación se caracteriza por un nombre y por un dominio. El dominio indica qué valores pueden ser asumidos por una columna de la relación. A menudo un dominio se define a través de la declaración de un tipo para el atributo (por ejemplo diciendo que es una cadena de diez caracteres), pero también es posible definir dominios más complejos y precisos.

En el modelo relacional podemos pensar en las relaciones como tablas. Más formalmente, la relación es un conjunto de tuplas, y cada tupla tiene exactamente el mismo conjunto de atributos. Además podemos decir que se cumplen las siguientes propiedades para toda relación:

1. No hay tuplas duplicadas.
2. El orden de las tuplas es insignificante.
3. El orden de los atributos también es insignificante.
4. Todas las entradas en las relaciones son valores atómicos.

B.2. El álgebra relacional

El álgebra relacional es un lenguaje procedural de consulta y está formado por cinco operaciones fundamentales. Estas operaciones son la proyección, selección, producto cartesiano, unión y diferencia. Toda consulta puede ser generada con esas cinco operaciones, pero se incluyen además una serie de operaciones adicionales que simplifican la definición de consultas usuales. Estas operaciones adicionales incluyen la intersección, el natural join, outer joins y la división. El operador de renombre algunas veces es incluido como operación auxiliar aunque podría considerarse una operación fundamental.

Todos los operadores del álgebra relacional utilizan como operandos relaciones. Además, estos operadores producen como resultado una nueva relación. Los operadores pueden unirse para expresar consultas más complejas.

Para las operaciones del álgebra relacional, una relación es vista como un conjunto de tuplas. Estas operaciones del álgebra relacional son pues las operaciones utilizadas en la teoría de conjuntos con algunos operadores adicionales.

B.2.1. Operaciones Fundamentales

A continuación se explicarán las cinco operaciones fundamentales del álgebra relacional.

Proyección

La proyección es una operación unaria que se denota por la letra griega pi (π). De una forma intuitiva podemos definir la proyección sobre una relación como la misma relación omitiendo algunas columnas. Definiendo la operación de una forma más formal, la operación de proyección copia la relación que utiliza como argumento dejando algunas columnas atrás. Las columnas que queremos conservar se pasan a la operación como una lista de atributos. Así pues, la sintaxis de la proyección se define de la siguiente forma:

$$\pi_{a_1, \dots, a_n}(R) = \{ t[a_1, \dots, a_n] : t \in R \}$$

donde $t[a_1, \dots, a_n]$ es la restricción de la tupla t al conjunto $\{a_1, \dots, a_n\}$. Así pues

$$t[a_1, \dots, a_n] = \{ (a', v) \mid (a', v) \in t, a' \in a_1, \dots, a_n \}$$

Nótese que si la proyección produce dos filas idénticas, la fila duplicada debe ser eliminada dado que la relación es un conjunto y no está permitido que contenga tuplas idénticas.

Persona			$\pi_{\text{Edad, Peso}}(\text{Persona})$	
Nombre	Edad	Peso	Edad	Peso
Harry	34	80	34	80
Sally	28	64	28	64
George	29	70	29	70
Helena	54	54	54	54
Peter	34	80		

Tabla B.1: Ejemplo de Proyección

Atendiendo al ejemplo de la Tabla B.1, supongamos que el predicado de *Persona* es “*Nombre es Edad años y Peso kilos*”. Entonces la proyección representaría el predicado, “*Existe Nombre tal que Nombre es Edad años y Peso kilos*”.

Nótese que tanto *Harry* como *Peter* tienen la misma edad y peso, pero como el resultado es una relación, y por lo tanto un conjunto, esa combinación parece una sola vez en el resultado.

Selección

La selección es una operación unaria que selecciona las tuplas que satisfacen un predicado dado. Al igual que la proyección selecciona solamente un subconjunto de atributos, la selección selecciona un subconjunto de tuplas. Esta operación se denota por la letra minúscula griega sigma (σ). La relación pasada como argumento se expresa entre paréntesis y la condición de selección se anota al lado del operador. Por lo tanto, quedará escrita como $\sigma_{a \vartheta b}(R)$ ó $\sigma_{a \vartheta v}(R)$ donde:

- a y b son nombres de atributos
- ϑ es una operación binaria del conjunto $\{=, \neq, <, \leq, >, \geq\}$
- v es un valor constante
- R es una relación

Más formalmente la semántica de la selección se define como:

$$\begin{aligned}\sigma_{a \vartheta b}(R) &= \{ t: t \in R, t(a) \vartheta t(b) \} \\ \sigma_{a \vartheta v}(R) &= \{ t: t \in R, t(a) \vartheta v \}\end{aligned}$$

En la Tabla B.2 se pueden observar dos ejemplos de selecciones. La primera selección nos da como resultado una relación en la que solamente aparecen las tuplas de *Persona* que cumplen que su valor de edad es menor o igual que 34.

En la segunda, solamente aparecen las tuplas que cumplen la condición de que su edad es igual a su peso.

Persona			$\sigma_{\text{Edad} \geq 34}(\text{Persona})$		
Nombre	Edad	Peso	Nombre	Edad	Peso
Harry	34	80	Harry	34	80
Sally	28	64	Helena	54	54
George	29	70	Peter	34	80
Helena	54	54			
Peter	34	80			

$\sigma_{\text{Edad} = \text{Peso}}(\text{Persona})$		
Nombre	Edad	Peso
Helena	54	54

Tabla B.2: Ejemplo de Selección

Producto Cartesiano

Hasta ahora no hemos sido capaces de combinar la información de varias relaciones. El producto cartesiano es la primera operación binaria capaz de esto y se denota con una cruz ($\times$). El producto cartesiano de dos relaciones, R y S , se escribe como $R \times S$. O más formalmente:

$$R \times S = \{(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m) \mid (a_1, a_2, \dots, a_n) \in R \wedge (b_1, b_2, \dots, b_m) \in S\}$$

El producto cartesiano genera como resultado una relación cuyas tuplas son formadas combinando cada posible par de tuplas: siendo uno de ellos de la relación R y el otro de la relación S . Si hay n_1 tuplas en R y n_2 tuplas en S , entonces hay $n_1 n_2$ tuplas en la relación resultante de aplicarles el producto cartesiano.

Si ambas relaciones que sirven como operandos contienen un atributo con el mismo nombre, la relación final contendría dos atributos con el mismo nombre por lo que se provocaría un conflicto. Para estos casos la operación de renombrar puede ayudar a evitar tales situaciones.

Como se puede observar en el ejemplo de la Tabla B.3, el producto cartesiano de las relaciones *Persona* y *Coche* da como resultado una relación cuyas tuplas son formadas por las combinaciones de las tuplas de *Persona* con las tuplas de *Coche*.

Unión

La operación binaria de unión se denota, al igual que en teoría de conjuntos, por $\cup$. La unión se utiliza para unir el contenido de sus argumentos, sin embargo, la unión del modelo relacional no es tan general como la unión del modelo matemático ya que la del modelo relacional solamente une relaciones compatibles.

Persona			Coche	
Nombre	Edad	Peso	Matricula	Kilometraje
Harry	34	80	12505ABC	25000
Sally	28	64	58462DDK	15000
George	29	70		

Persona $\times$ Coche				
Nombre	Edad	Peso	Matricula	Kilometraje
Harry	34	80	12505ABC	25000
Sally	28	64	12505ABC	25000
George	29	70	12505ABC	25000
Harry	34	80	58462DDK	15000
Sally	28	64	58462DDK	15000
George	29	70	58462DDK	15000

Tabla B.3: Ejemplo de Producto Cartesiano

$$R \cup S = \{ t \mid t \in R \vee t \in S \}$$

Debemos asegurar que la unión se aplica a dos relaciones compatibles para ello. Así pues, para que una operación de unión como la anterior sea correcta, se requiere que se cumplan dos condiciones:

1. Las relaciones R y S deben tener la misma aridad. Es decir, deben tener el mismo número de atributos.
2. El dominio del atributo i de R y el dominio del atributo i de S deben ser iguales o compatibles.

Empleado			Cliente			Empleado $\cup$ Cliente		
Nombre	Edad	Peso	Nombre	Edad	Peso	Nombre	Edad	Peso
Harry	34	80	Peter	34	80	Harry	34	80
Sally	28	64	Helena	54	54	Sally	28	64
George	29	70	George	29	70	George	29	70
						Peter	34	80
						Helena	54	54

Tabla B.4: Ejemplo de Unión

En el ejemplo de la Tabla B.4 se observa la unión de dos relaciones (*Empleado* y *Cliente*) compatibles para la unión. La relación resultante contiene la unión de ambas sin contener tuplas repetidas.

Diferencia

La última de las operaciones fundamentales del álgebra relacional es la diferencia. Esta operación se denota con el símbolo $-$. Para poder aplicar este operador, las dos relaciones tienen que ser compatibles. El resultado de la expresión $R - S$, es una relación obtenida al incluir todas las tuplas de R que no aparecen en S . Formalmente:

$$R - S = R \setminus S = \{ t \mid t \in R \wedge t \notin S \}$$

Por supuesto, la relación resultante no contiene tuplas duplicadas.

Empleado			Cliente		
Nombre	Edad	Peso	Nombre	Edad	Peso
Harry	34	80	Peter	34	80
Sally	28	64	Helena	54	54
George	29	70	George	29	70

Empleado - Cliente		
Nombre	Edad	Peso
Harry	34	80
Sally	28	64

Tabla B.5: Ejemplo de Diferencia

Supongamos que quisiéramos obtener las personas que son empleados de nuestra empresa pero que nunca han sido clientes como ocurre en el ejemplo de la Tabla B.5. La diferencia de las relaciones *Empleado* y *Cliente* nos da como resultado una relación que contiene las tuplas de *Empleado* que no aparecen en *Cliente*.

Renombre

Como se ha mencionado anteriormente, la operación de renombre se omite algunas veces de la lista de operaciones fundamentales o, en ocasiones, se clasifica como una operación auxiliar. La operación de renombre, denotada por la letra minúscula griega rho (ρ), es una operación unaria. El resultado de aplicar el operador de renombre a una relación es una relación idéntica a la original excepto por el hecho de que los atributos tienen nuevos nombres. Esta operación es muy útil para evitar conflictos en la operación de producto cartesiano, como se ha comentado previamente.

B.2.2. Construcción de expresiones

Los cinco operadores fundamentales dan al álgebra relacional el poder para formular consultas complejas. El álgebra relacional es definida por las siguientes reglas:

1. Si R es una relación, entonces es una expresión algebraica.
2. Si $E1$ y $E2$ son expresiones algebraicas, entonces también lo son $E1 \cup E2$, $E1 - E2$ y $E1 \times E2$. Para poder aplicar la unión y la diferencia, $E1$ y $E2$ deben ser compatibles.
3. Si E es una expresión algebraica, entonces también lo son $\pi_L(E)$, $\sigma_P(E)$, $\rho_S(E)$, siendo L una lista de nombres de atributos contenidos en el esquema de E , P un predicado con los atributos de E y S una lista de nuevos nombres.
4. Si E es una expresión, también lo es (E) .

B.2.3. Operadores adicionales

Hemos introducido las cinco operaciones fundamentales del álgebra relacional. Con estos cinco operadores disponemos del potencial necesario para expresar cualquier consulta en álgebra relacional. Sin embargo, algunas consultas muy utilizadas se convierten en expresiones muy complejas cuando nos vemos restringidos a utilizar estas operaciones. Es por esto que surgen un conjunto de operaciones adicionales que simplifican las consultas.

Es necesario notar que los operadores adicionales no incrementan el potencial del álgebra relacional. Cualquier consulta que pueda ser expresada utilizando los operadores adicionales puede ser expresada también solamente con operadores fundamentales. La única razón para su introducción es pues la simplificación de consultas comunes. A continuación se explican algunos de ellos.

Intersección

La primera operación que añadimos al álgebra relacional es la intersección, la cual es denotada por $\cap$ y se trata de una operación binaria. La intersección no se considera una operación fundamental porque puede ser expresada fácilmente utilizando dos operadores de diferencia. Así pues, se requiere que las relaciones que sirven de operandos sean compatibles para la unión. Entonces podemos reescribir la intersección de la siguiente forma:

$$R \cap S = R - (R - S)$$

Formalmente podemos definir la intersección como:

$$R \cap S = \{ t \mid t \in R \wedge t \in S \}$$

Después de aplicar el operador de intersección obtenemos una relación que contiene solamente aquellas tuplas de R que aparecen también como tuplas en S . No necesitamos eliminar filas duplicadas porque la relación resultante no puede contener alguna (dado a que ninguno de los operandos contiene alguna).

Empleado			Cliente		
Nombre	Edad	Peso	Nombre	Edad	Peso
Harry	34	80	Peter	34	80
Sally	28	64	Helena	54	54
George	29	70	George	29	70

Empleado $\cap$ Cliente		
Nombre	Edad	Peso
George	29	70

Tabla B.6: Ejemplo de Intersección

La intersección de las relaciones Empleado y Cliente (ejemplo de la Tabla B.6) da como resultado una nueva relación que contiene aquellas tuplas de Empleado que aparecen en Cliente.

División

La división es una operación binaria denotada por $\div$. El resultado de aplicar la operación de división $R \div S$ consiste en las tuplas de R , en las cuales sólo se seleccionan los atributos únicos de R (que aparecen en la cabecera de R pero no en la de S), tales que existen todas sus combinaciones con las tuplas de S en R .

Más formalmente la semántica de la división se define como:

$$R \div S = \{ t[a_1, \dots, a_n] : t \in R \wedge \forall s \in S ((t[a_1, \dots, a_n] \cup s) \in R) \}$$

donde $\{a_1, \dots, a_n\}$ es el conjunto de atributos únicos de R y $t[a_1, \dots, a_n]$ es la restricción t de este conjunto. Normalmente se requiere que los atributos de la cabecera de S sean un subconjunto de los de R porque de otra forma el resultado de la operación será siempre vacío.

La simulación de la división con los operadores fundamentales se realiza de la siguiente forma. Asumimos que $a_1, \dots, a_n$ son los atributos únicos de R y que $b_1, \dots, b_n$ son los atributos de S . Como primer paso obtenemos la proyección de R con sus atributos únicos y construimos todas las combinaciones de esto con las tuplas de S :

$$T := \pi_{a_1, \dots, a_n}(R) \times S$$

Ahora restamos R a esa relación:

$$U := T - R$$

Nótese que en U tenemos las combinaciones posibles que podrían estar R , pero no están. Así que si ahora obtenemos la proyección de los atributos únicos de R entonces obtendremos las tuplas de R para las cuales no están presentes todas las combinaciones con las tuplas de S en R :

$$V := \pi_{a_1, \dots, a_n}(U)$$

El último paso a realizar es restar a la proyección de los atributos únicos de R la relación V :

$$W := \pi_{a_1, \dots, a_n}(R) - V$$

Completed			
Student	Task	DBProject	Completed $\div$ DBProject
Fred	Database1	Nombre	Nombre
Fred	Database2	Database1	Fred
Fred	Compiler1	Database2	Sara
Eugene	Database1		
Eugene	Compiler1		
Sara	Database1		
Sara	Database2		

Tabla B.7: Ejemplo de División

Si DBProject contiene todas las tareas del proyecto de una base de datos (ejemplo de la Tabla B.7) entonces el resultado de la división del ejemplo contiene todos los estudiantes que han completando todas esas tareas.

Natural Join

Otra operación adicional que vamos a describir es el *natural join* (o reunión natural). Es una operación binaria denotada por $\bowtie$. El natural join es una operación muy utilizada que nos permite combinar relaciones que tienen algún atributo común sin necesidad de especificar nada. Se utiliza, por ejemplo, para la obtención de la información referente a varias relaciones a través de su clave primaria.

Considerando dos relaciones R y S donde $(A_1, \dots, A_n, B_1, \dots, B_n)$ son los atributos de R y $(B_1, \dots, B_n, C_1, \dots, C_n)$ los atributos de S , podemos definir el natural join de una manera formal como sigue:

Persona			Dept	
Nombre	EmpId	NombreDept	NombreDept	Jefe
Harry	3415	Finanzas	Finanzas	George
Sally	2241	Ventas	Ventas	Harriet
George	3401	Finanzas	Producción	Charles
Harriet	2202	Ventas		

Persona $\bowtie$ Dept			
Nombre	EmpId	NombreDept	Jefe
Harry	3415	Finanzas	George
Sally	2241	Ventas	Harriet
George	3401	Finanzas	George
Harriet	2202	Ventas	Harriet

Tabla B.8: Ejemplo de Natural Join

$$R \bowtie S = \{ r \cup s_{[C1, \dots, Cn]} \mid r \in R \wedge s \in S \wedge r_{[B1, \dots, Bn]} = s_{[B1, \dots, Bn]} \}$$

En el ejemplo de la Tabla B.8 se puede comprobar como la información de que el departamento de *Producción* esta dirigido por *Charles* se pierde en la relación resultado.

Outer Joins

La última operación adicional que vamos a introducir es el *outer join*. Un outer join es similar al natural join explicado anteriormente. La única diferencia es que el outer join mantiene la información que habría sido perdida reemplazando la información faltante por nulls.

Hay tres tipos de outer joins, left, right y full. El *left outer join* se denota por $^+ \bowtie$. Siendo $r_1, r_2, \dots, r_n$ los atributos de la relación R y siendo $\{(\omega, \dots, \omega)\}$ la relación unitaria de los atributos que son únicos en la relación S (aquellos que no son atributos de R), entonces el left outer join puede ser descrito en términos del natural join (y utilizando operadores fundamentales) como sigue:

$$(R \bowtie S) \cup ((R - \pi_{r1, r2, \dots, rn}(R \bowtie S)) \times \{(\omega, \dots, \omega)\})$$

En la Tabla B.9 se puede ver un ejemplo de left outer join donde se puede observar cómo aparece toda la información de la relación *Persona* y se completan la relación resultante con *nulls* (ω).

Persona			Dept	
Nombre	EmpId	NombreDept	NombreDept	Jefe
Harry	3415	Finanzas	Ventas	Harriet
Sally	2241	Ventas	Producción	Charles
George	3401	Finanzas		
Harriet	2202	Ventas		
Tim	1123	Ejecutivo		

Persona ⁺ ⋈ Dept			
Nombre	EmpId	NombreDept	Jefe
Harry	3415	Finanzas	ω
Sally	2241	Ventas	Harriet
George	3401	Finanzas	ω
Harriet	2202	Ventas	Harriet
Tim	1123	Ejecutivo	ω

Tabla B.9: Ejemplo de Left Outer Join

En cuanto al *right outer join*, su estructura es muy similar al left outer join. Se denota por $\bowtie^+$. Siendo $s_1, s_2, \dots, s_n$ los atributos de la relación S y siendo $\{(\omega, \dots, \omega)\}$ la relación unitaria de los atributos que son únicos en la relación R (aquellos que no son atributos de S). Entonces, igual que con el left outer join, el right outer join puede ser simulado usando la siguiente expresión:

$$(R \bowtie S) \cup (\{(\omega, \dots, \omega)\} \times (S - \pi_{s_1, s_2, \dots, s_n}(R \bowtie S)))$$

En la Tabla B.10 se puede ver un ejemplo de left outer join donde se puede observar cómo aparece toda la información de la relación *Dept* y se completan la relación resultante con *nulls* (ω).

Y por último el *full outer join*, denotado por $\bowtie^+$ y cuyo resultado aplicado a dos relaciones R y S es el conjunto de todas las combinaciones de tuplas de R y S que son iguales en sus atributos comunes, además de las tuplas de S que no tienen pareja en los atributos de R y las tuplas de R que no tienen pareja en los atributos de S .

En la Tabla B.12 se puede ver un ejemplo de left outer join donde se puede observar cómo aparece toda la información de las relaciones *Persona* y *Dept* completando la relación resultante con *nulls* (ω).

Persona				
Nombre	EmpId	NombreDept	Dept	
Harry	3415	Finanzas	NombreDept	Jefe
Sally	2241	Ventas	Ventas	Harriet
George	3401	Finanzas	Producción	Charles
Harriet	2202	Ventas		
Tim	1123	Ejecutivo		

Persona $\bowtie^+$ Dept			
Nombre	EmpId	NombreDept	Jefe
Sally	2241	Ventas	Harriet
Harriet	2202	Ventas	Harriet
ω	ω	Producción	Charles

Tabla B.10: Ejemplo de Right Outer Join

Persona				
Nombre	EmpId	NombreDept	Dept	
Harry	3415	Finanzas	NombreDept	Jefe
Sally	2241	Ventas	Ventas	Harriet
George	3401	Finanzas	Producción	Charles
Harriet	2202	Ventas		
Tim	1123	Ejecutivo		

Persona $^+ \bowtie^+ \text{ Dept}$			
Nombre	EmpId	NombreDept	Jefe
Harry	3415	Finanzas	ω
Sally	2241	Ventas	Harriet
George	3401	Finanzas	ω
Harriet	2202	Ventas	Harriet
Tim	1123	Ejecutivo	ω
ω	ω	Producción	Charles

Tabla B.11: Ejemplo de Full Outer Join

Semijoin

El semijoin es similar al natural join y se expresa mediante el símbolo $\triangleright <$. El resultado de la operación de semijoin $R \bowtie S$ es el conjunto solamente de aquellas

tuplas de R para las cuales hay una tupla en S que es igual en cuanto a sus atributos comunes. Por ejemplo, considerando las tablas *Persona* y *Dept*, se puede ver en la Tabla B.12 el resultado de aplicar dicha operación.

Persona			Dept	
Nombre	EmpId	NombreDept	NombreDept	Jefe
Harry	3415	Finanzas	Ventas	Harriet
Sally	2241	Ventas	Producción	Charles
George	3401	Finanzas		
Harriet	2202	Producción		

Persona $\triangleright <$ Dept		
Nombre	EmpId	NombreDept
Sally	2241	Ventas
Harriet	2202	Producción

Tabla B.12: Ejemplo de Semijoin

El semijoin puede ser simulado utilizando un natural join. Si $a_1, a_2, \dots, a_n$ son los nombres de los atributos de R , entonces:

$$R \triangleright < S = \pi_{a_1, a_2, \dots, a_n}(R \bowtie S)$$

Además, como podemos simular el natural join por medio de operadores fundamentales, el semijoin también puede ser representado por medio de dichos operadores.

B.3. Optimización de consultas

Las consultas pueden ser representadas por un árbol, donde:

- Los nodos internos son operadores.
- Las hojas son relaciones.
- Los subárboles son subexpresiones.

El objetivo principal es transformar los árboles de expresiones en árboles de expresiones equivalentes, donde el tamaño medio de las relaciones fruto de las subexpresiones del árbol son menores de lo que eran antes de la optimización. A continuación se presentan un conjunto de reglas, que pueden ser utilizadas en dichas transformaciones.

B.3.1. Selección

Las reglas acerca de los operadores de selección juegan el papel más importante en la optimización de consultas. La selección es una operación que puede decrementar de forma muy efectiva el número de filas de sus operandos, así que si conseguimos desplazar las selecciones en el árbol de la expresión hacia las hojas, las relaciones internas (generadas por las subexpresiones) verán reducido su tamaño.

Propiedades básicas de la selección

La selección es idempotente (aplicaciones múltiples de la misma selección no tienen efecto adicional más allá de la primera), y conmutativa (el orden en que las selecciones son aplicadas no tiene efecto en el resultado).

$$\begin{aligned}\sigma_A(R) &= \sigma_A \sigma_A(R) \\ \sigma_A \sigma_B(R) &= \sigma_B \sigma_A(R)\end{aligned}$$

Dividiendo selecciones con condiciones complejas

Una selección cuya condición es un conjunto de condiciones simples es equivalente a una secuencia de selecciones con esas mismas condiciones simples, y también, selecciones cuya condición es una disyunción son equivalentes a la unión de selecciones. Estas propiedades pueden ser utilizadas para unir selecciones de tal forma que se tengan que evaluar menos selecciones, o para dividir las de tal forma que éstas puedan ser desplazadas por el árbol de forma separada.

$$\begin{aligned}\sigma_{A \wedge B}(R) &= \sigma_A(\sigma_B(R)) = \sigma_B(\sigma_A(R)) \\ \sigma_{A \vee B}(R) &= \sigma_B(R) \cup \sigma_A(R)\end{aligned}$$

Selección y producto cartesiano

El producto cartesiano es la operación más costosa de evaluar. Si las relaciones de entrada tienen n y m filas, el resultado contendrá nm filas. Por lo tanto es muy importante hacer lo posible por disminuir el tamaño de ambos operadores antes de aplicar el producto cartesiano. Esto puede ser realizado de forma eficiente, si el producto cartesiano es seguido por un operador de selección, por ejemplo $\sigma_A(R \times P)$. Teniendo en cuenta la definición de la operación de join, esto ocurrirá muy a menudo. Si el producto cartesiano no es seguido por un operador de selección, podemos tratar de bajar alguno de niveles superiores utilizando las reglas del operador de selección.

En el ejemplo de abajo dividimos la condición A en las condiciones B , C y D utilizando las reglas de división de condiciones de selección complejas, así que $A = B \wedge C \wedge D$ y B contiene sólo atributos de R , C contiene sólo atributos de P y

D contiene la parte de A que contiene atributos de R y de P . Nótese que B , C o D podrían ser vacíos. Así pues:

$$\sigma_A(R \times P) = \sigma_{B \wedge C \wedge D}(R \times P) = \sigma_D(\sigma_B(R) \times \sigma_C(P))$$

Selección y operadores de conjuntos

La selección es distributiva en cuanto a los operadores de resta, intersección y unión. Las tres reglas siguientes son utilizadas para bajar las selecciones por debajo de las operaciones antes comentadas en el árbol de la expresión. Nótese que en los operadores de resta e intersección solamente puede aplicarse la operación de selección a uno de los operadores después de la transformación. Esto puede tener sentido en algunos casos, donde uno de los operandos es pequeño, y la sobrecarga de evaluar la operación de selección es mayor que los beneficios de utilizar una relación más pequeña como operando.

$$\begin{aligned}\sigma_A(R - P) &= \sigma_A(R) - \sigma_A(P) = \sigma_A(R) - P \\ \sigma_A(R \cup P) &= \sigma_A(R) \cup \sigma_A(P) \\ \sigma_A(R \cap P) &= \sigma_A(R) \cap \sigma_A(P) = \sigma_A(R) \cap P = R \cap \sigma_A(P)\end{aligned}$$

Selección y proyección

La selección es asociativa con la proyección si y solo si los campos referenciados en la condición de selección son un subconjunto de los campos de la proyección. Realizar la selección antes que la proyección puede ser útil si el operando es un producto cartesiano o un join. En otros casos, si la condición de selección es relativamente costosa de computar, mover la selección puede reducir el número de tuplas que deben ser comprobadas.

$$\pi_{a1, a2, \dots, an}(\sigma_A(R)) = \sigma_A(\pi_{a1, a2, \dots, an}(R)) \text{ siendo los campos de } A \subseteq \{a1, a2, \dots, an\}$$

B.3.2. Proyección

Propiedades básicas de la proyección

La proyección es idempotente, así que una serie de proyecciones es equivalente a la proyección más externa.

$$\pi_{a1, a2, \dots, an}(\pi_{b1, b2, \dots, bn}(R)) = \pi_{a1, a2, \dots, an}(R) \text{ donde } \{a1, a2, \dots, an\} \subseteq \{b1, b2, \dots, bn\}$$

Proyección y operadores de conjuntos

La proyección es distributiva sobre la unión.

$$\pi_{a_1, a_2, \dots, a_n}(R \cup P) = \pi_{a_1, a_2, \dots, a_n}(R) \cup \pi_{a_1, a_2, \dots, a_n}(P)$$

La proyección no es distributiva sobre la intersección y la diferencia. Como contraejemplos:

$$\begin{aligned}\pi_A(\{ \langle A = a, B = b \rangle \} \cap \{ \langle A = a, B = b' \rangle \}) &= \emptyset \\ \pi_A(\{ \langle A = a, B = b \rangle \}) \cap \pi_A(\{ \langle A = a, B = b' \rangle \}) &= \{ \langle A = a \rangle \}\end{aligned}$$

y

$$\begin{aligned}\pi_A(\{ \langle A = a, B = b \rangle \} - \{ \langle A = a, B = b' \rangle \}) &= \{ \langle A = a \rangle \} \\ \pi_A(\{ \langle A = a, B = b \rangle \}) - \pi_A(\{ \langle A = a, B = b' \rangle \}) &= \emptyset\end{aligned}$$

siendo b distinto de b'

B.3.3. Renombre

Propiedades básicas del renombre

Una serie sucesiva de renombres de una variable pueden ser convertidos en un sólo renombre. Las operaciones de renombre que no tienen variables en común pueden ser reordenadas de forma arbitraria unas de otras, lo cual puede ser utilizada para juntar los renombres.

$$\begin{aligned}\rho_{a/b}(\rho_{b/c}(R)) &= \rho_{a/c}(R) \\ \rho_{a/b}(\rho_{c/d}(R)) &= \rho_{c/d}(\rho_{a/b}(R))\end{aligned}$$

Renombre y operadores de conjuntos

El renombre es distributivo sobre las operaciones de resta, unión e intersección.

$$\begin{aligned}\rho_{a/b}(R - P) &= \rho_{a/b}(R) - \rho_{a/b}(P) \\ \rho_{a/b}(R \cup P) &= \rho_{a/b}(R) \cup \rho_{a/b}(P) \\ \rho_{a/b}(R \cap P) &= \rho_{a/b}(R) \cap \rho_{a/b}(P)\end{aligned}$$

B.3.4. Otras operaciones

Respecto a otras operaciones podemos citar las siguientes reglas:

- Join y producto cartesiano son asociativas.
- Join es conmutativa.
- Conmutatividad de las operaciones de conjuntos: la unión y la intersección son conmutativas pero la diferencia no.
- Asociatividad de las operaciones de conjuntos: la unión y la intersección son asociativas pero la diferencia no.

Apéndice C

Análisis y diseño de la aplicación

En este anexo se presenta una descripción del análisis y diseño que se llevó a cabo antes de realizar la implementación con el objetivo último de comprender la naturaleza del problema y facilitar la comprensión del sistema. A lo largo del anexo se mostrará el documento de especificación de requisitos, diagramas de clases, diagramas de casos de uso y diagramas de trazas de eventos, así como la descripción de los distintos módulos y algunos prototipos de la ventana principal. Los diagramas siguen una notación similar al lenguaje de modelado UML. No obstante, no se pretende ser completamente riguroso, por lo que en ocasiones se ha optado por describir textualmente algunas partes y facilitar al lector la comprensión de los mismos.

C.1. Especificación de Requisitos Software

Esta sección presenta el documento de Especificación de Requisitos Software (ERS) de la herramienta para el aprendizaje del álgebra relacional y optimización de consultas según la última versión del estándar IEEE 830 [18]. Según IEEE, un buen Documento de Requisitos, pese a no ser obligatorio que siga estrictamente la organización y el formato dados en el estándar 830, sí deberá incluir, de una forma o de otra, toda la información presentada en dicho estándar.

C.1.1. Introducción

Propósito

El propósito es definir cuáles son los requerimientos que tiene la herramienta para el aprendizaje del álgebra relacional y la optimización de consultas.

La aplicación se desarrollará como proyecto final de carrera y facilitará a los alumnos el aprendizaje del álgebra relacional y sus manipulaciones para el procesamiento de consultas.

Ámbito

El software debe permitir la definición y manipulación de expresiones en álgebra relacional. Así como la definición y manipulación de árboles de expresiones del álgebra relacional.

Una vez definidas las expresiones del álgebra relacional se permitirá la interpretación de las mismas. Además se permitirá que dicho proceso se realice de forma automática o paso a paso.

También se permitirá la traducción entre álgebra relacional y SQL con objeto de que dichas expresiones puedan ser ejecutadas sobre bases de datos reales y para que el usuario entienda la relación entre ambos lenguajes.

Por último se permitirá la generación de planes para la ejecución de las consultas con aplicación de distintas técnicas de optimización (tanto en modo manual/interactivo como en automático) a partir de datos estadísticos.

No es el objetivo de la aplicación el manejo y explicación de relaciones por lo que se permitirá la inclusión de algunas a modo de ejemplo o la obtención de las mismas de bases de datos existentes pero no se explicará con detalle este proceso puesto que para ello se creó otra herramienta como PFC.

Se espera como objetivo fundamental que los usuarios de la aplicación tengan en ella una ayuda para la comprensión y el aprendizaje del álgebra relacional.

Definiciones, Acrónimos y Abreviaturas

- Álgebra relacional: lenguaje formal con una serie de operadores que trabajan sobre una o varias relaciones para obtener otra relación resultado, sin que cambien las relaciones originales.
- Árbol de expresión en álgebra relacional: estructura de datos en forma de árbol en la que cada hoja son operandos y los nodos internos son operadores aplicados a los hijos.
- Relación: es una estructura de datos que consiste en una cabecera y un conjunto sin orden de tuplas que comparten el mismo tipo.
- Tupla: Una tupla se define como una función finita que asocia unívocamente los nombres de los atributos de una relación con los valores de una instancia de la misma. En términos simplistas, es una fila de una tabla relacional.
- Atributo: en el contexto del álgebra relacional, un atributo (columna) es un conjunto de tipos de datos, uno por cada fila de la relación.
- SQL: Lenguaje de consulta estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas.

- Consulta: método para acceder a los datos en una base de datos.
- Plan de ejecución de consulta: conjunto de pasos que tiene que realizar el motor de una base de datos para ejecutar una consulta.

C.1.2. Descripción general

Perspectiva del producto

El sistema a desarrollar no estará relacionado con otros productos, aunque posteriormente a su desarrollo será considerada su integración con una aplicación anterior destinada a la docencia de bases de datos.

Funciones del producto

Las funciones que debe realizar el producto a grandes rasgos son:

- Definición y manipulación de relaciones de ejemplo
 - Definición de relaciones de ejemplo
 - Obtención de relaciones de bases de datos existentes
- Definición y manipulación de expresiones en álgebra relacional
 - Definición de expresiones textuales
 - Definición de árboles sintácticos
 - Interpretación de consultas
 - Traducción álgebra relacional a SQL
- Optimización de consultas
 - Generación de planes de consultas optimizadas
 - Optimización guiada de consultas

Características de los usuarios

Los usuarios finales de esta aplicación serán alumnos de asignaturas de Bases de Datos y profesores de las mismas que deseen utilizarla como apoyo a las explicaciones. Deberán tener un cierto conocimiento y algo de experiencia en el manejo de programas con interfaces de ventanas (algo que se da por hecho entre estos usuarios) y en cuanto al conocimiento de la materia serán necesarios alguno conceptos básicos relacionados con las bases de datos y el álgebra relacional (sobre todo de terminología).

Restricciones

La aplicación será realizada con un lenguaje de programación orientado a objetos como es Java. No se necesitará un hardware potente ya que las funcionalidades de la aplicación no lo requieren aunque cuanto mayor sea la potencia menor será el tiempo de ejecución de las consultas. Se permitirá al usuario obtener datos de ciertas bases de datos siempre y cuando se tenga conexión con las mismas.

Suposiciones y dependencias

Al utilizar como lenguaje de programación Java, y gracias a que se trata de un lenguaje multiplataforma, el cambio de sistema operativo no supondría un cambio en estos requisitos.

Requisitos futuros

Se puede considerar en un futuro añadir funcionalidades extra como podrían ser la interpretación de consultas en lenguaje natural.

También se podría considerar la integración de la herramienta en proyectos anteriores dedicados a la docencia de bases de datos.

C.1.3. Requerimientos específicos

Requerimientos funcionales

1. Definición de expresiones de álgebra relacional. De forma “textual” o a través de ayudas gráficas.
2. Definición de arboles sintácticos de álgebra relacional.
3. Obtención de arboles sintácticos de expresiones de álgebra relacional. Mostrados de forma gráfica y permitiendo su posterior manipulación.
4. Guardado/carga de expresiones de álgebra relacional.
5. Interpretación (automática o paso a paso) de expresiones de álgebra relacional sobre relaciones de ejemplo.
6. Guardado/carga de relaciones de ejemplo.
7. Conexión a bases de datos externas para la obtención de relaciones de ejemplo.
8. Introducción de datos en las relaciones.

9. Traducción de álgebra relacional a SQL.
10. Optimización (automática o paso a paso) de consultas introducidas por el usuario.
11. Imprimir árboles, relaciones y consultas.
12. Comprobación de la corrección de consultas y árboles.

Requerimientos de interfaces externos

La interfaz de usuario constará de ventanas adaptadas a la configuración de la resolución de pantalla aunque se recomendará el uso de resoluciones de 1280x1024 o superiores para la correcta visualización del sistema, y en la medida de lo posible se permitirá la configuración de la disposición de los diferentes elementos.

La aplicación está pensada para funcionar en cualquier sistema operativo que disponga de Java.

La obtención de relaciones de bases de datos externas se verá delimitada por el estado de la conexión en el momento del uso así como la ejecución de las consultas en el sistema gestor de bases de datos.

Requerimientos de rendimiento

Al tratarse de una aplicación que no permite la ejecución de múltiples usuarios al mismo tiempo, no se esperan problemas derivados de la carga que tenga que soportar el sistema. Cuanto mayores sean las relaciones de ejemplo con las que se trabajen mayor será el tiempo empleado en la realización de las consultas y explicaciones. Se ofrece además al usuario la posibilidad de generar tuplas automáticamente y seleccionar el algoritmo de join deseado. En función de estos aspectos, el tiempo de proceso puede aumentar, penalizando el rendimiento de la ejecución de las consultas.

Restricciones de diseño

En un principio no se encuentran limitaciones que restrinjan las decisiones relativas al diseño dado a que la herramienta está orientada para usuarios sin necesidades de interfaces especiales.

Atributos del sistema

Se espera una alta fiabilidad y corrección del sistema puesto que va a ser utilizado por estudiantes para el aprendizaje del álgebra relacional. Se espera también que la

mantenibilidad futura, de ser necesaria, sea lo más fácil posible puesto que no se realizaría por el desarrollador de la herramienta.

En cuanto a la seguridad no es un requisito fundamental del sistema, ya que todos los usuarios podrán realizar todas las funciones disponibles, por lo que no se utilizarán mecanismos de seguridad como pudieran ser logins (exceptuando la conexión al sistema gestor de bases de datos externo donde, será necesario que el usuario introduzca sus datos de acceso).

C.2. Diagrama de clases

El diagrama de clases del sistema se puede ver en la Figura C.1.

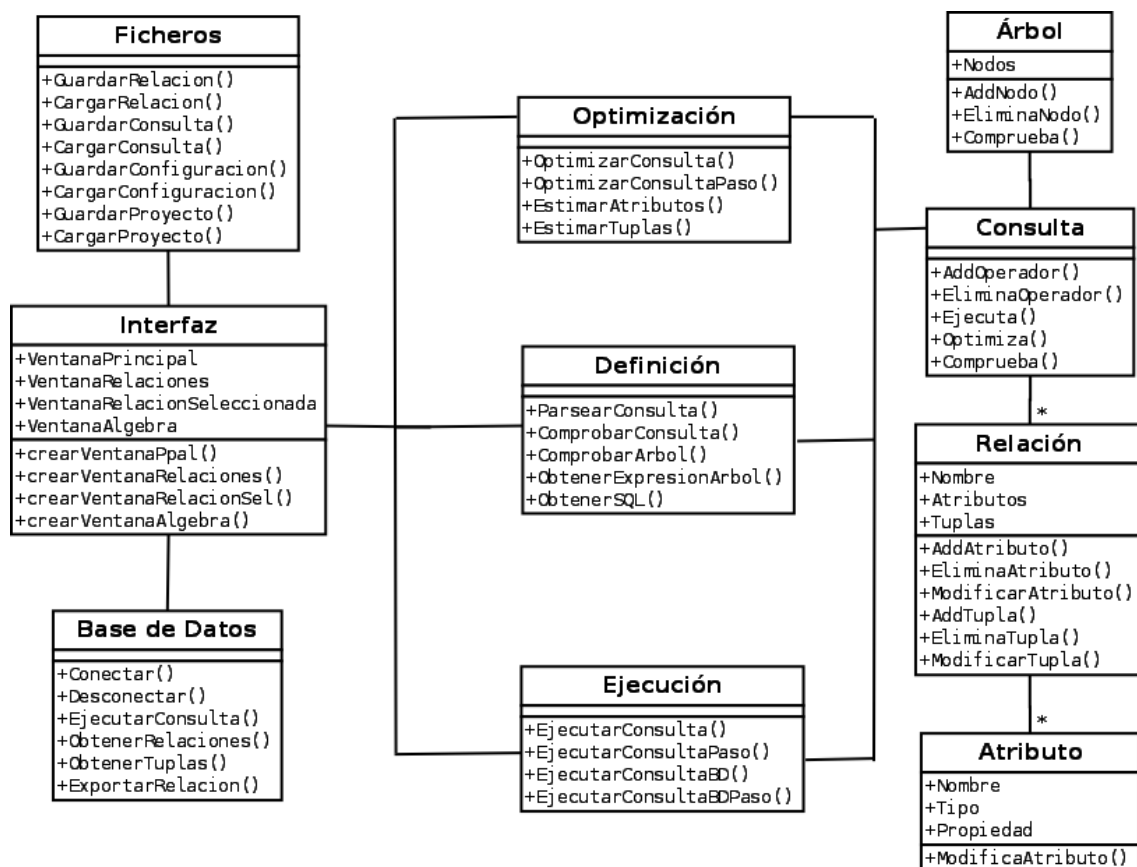


Figura C.1: Diagrama de clases en alto nivel de la aplicación

C.3. Casos de uso y trazas de eventos

En esta sección vamos a analizar los posibles usos que podrán darse a la herramienta. Para ilustrarlo vamos a emplear algunos diagramas de casos de uso y de trazas de eventos. En primer lugar pueden verse posibles usos generales de la herramienta en el diagrama de casos de uso de la Figura C.2.

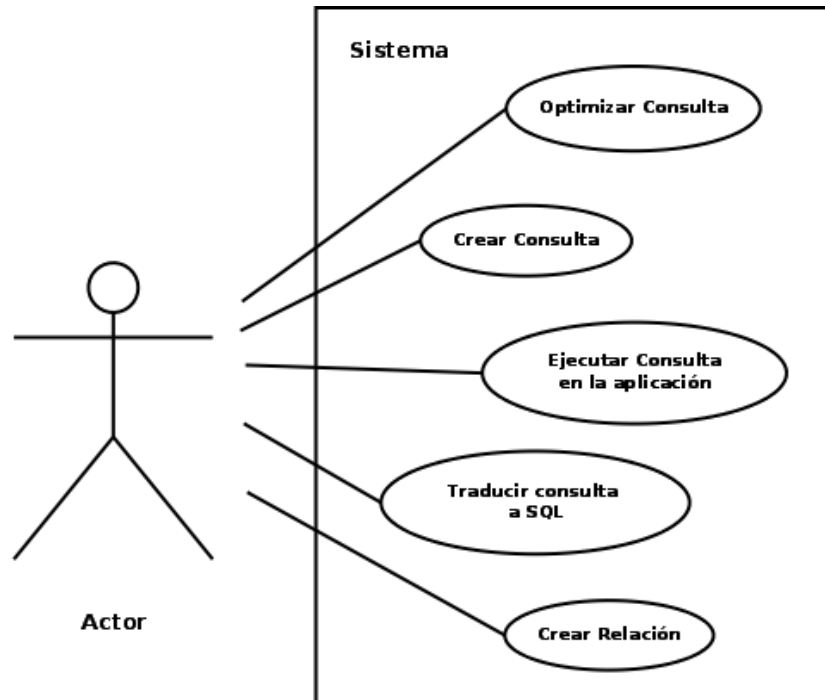


Figura C.2: Diagrama de casos de uso de la herramienta

Veamos ahora un ejemplo de posible escenario de uso para algunos de estos casos de uso. En la Figura C.3 podemos ver una posible traza de eventos para el caso de uso de creación de una relación. En este ejemplo en concreto, el usuario crea una relación con un atributo y la guarda cuando ha terminado en un fichero XML. En el diagrama se pueden ver las comunicaciones que se producen entre el usuario, el módulo de interfaz, y el módulo de manejo de ficheros.

En la Figura C.4 se puede ver un segundo escenario para el caso de uso de ejecutar una consulta. En este ejemplo concreto el usuario obtiene las relaciones desde un fichero XML así como la consulta, y posteriormente realiza la ejecución obteniendo una ventana con los resultados de la misma. En este caso se producen comunicaciones entre el usuario, el módulo de interfaz, el módulo de manejo de ficheros y el módulo de ejecución.

C.4. Módulos del sistema

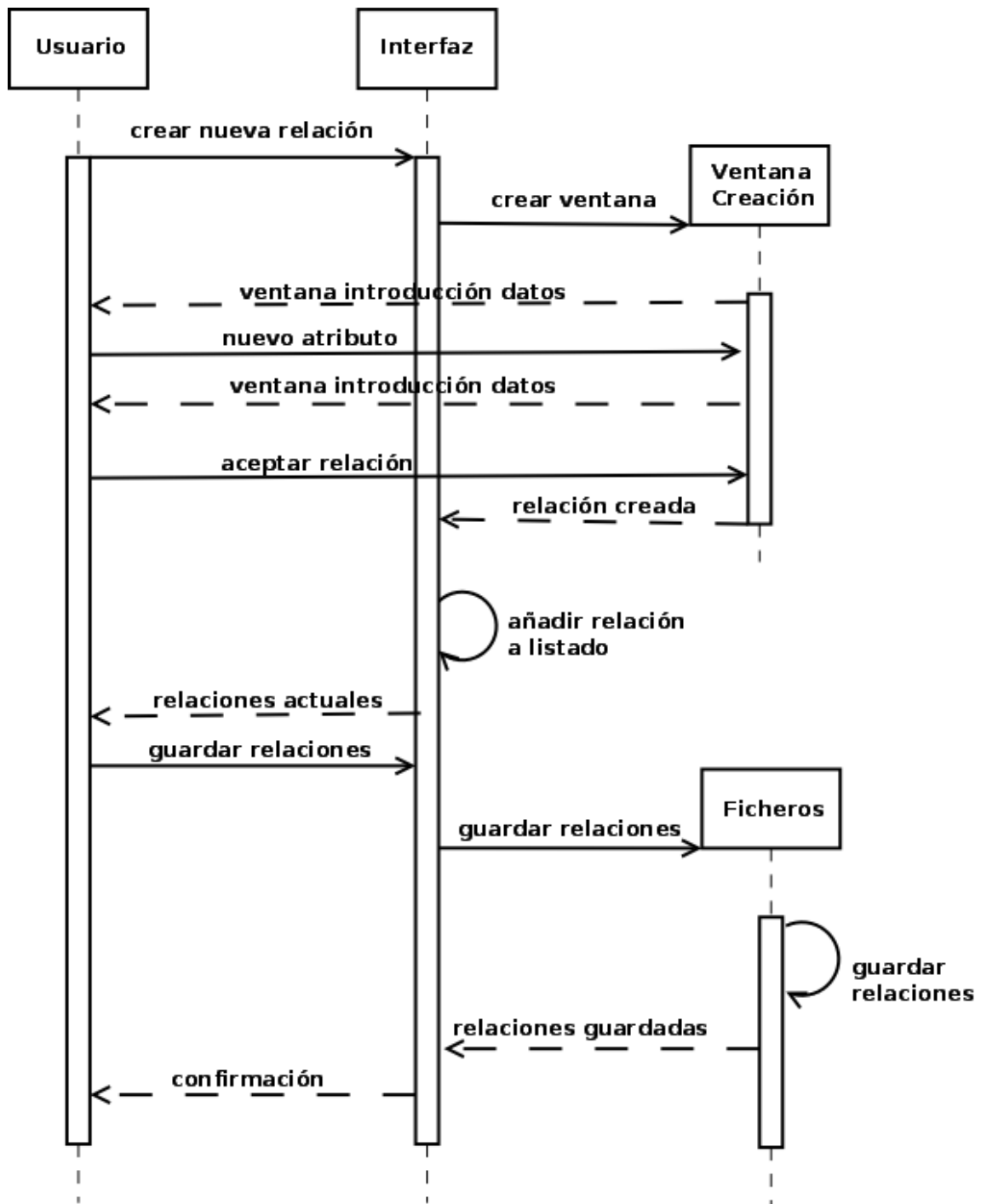


Figura C.3: Ejemplo de traza de eventos para la creación de una relación

A continuación se describen en detalle cada uno de los módulos del sistema.

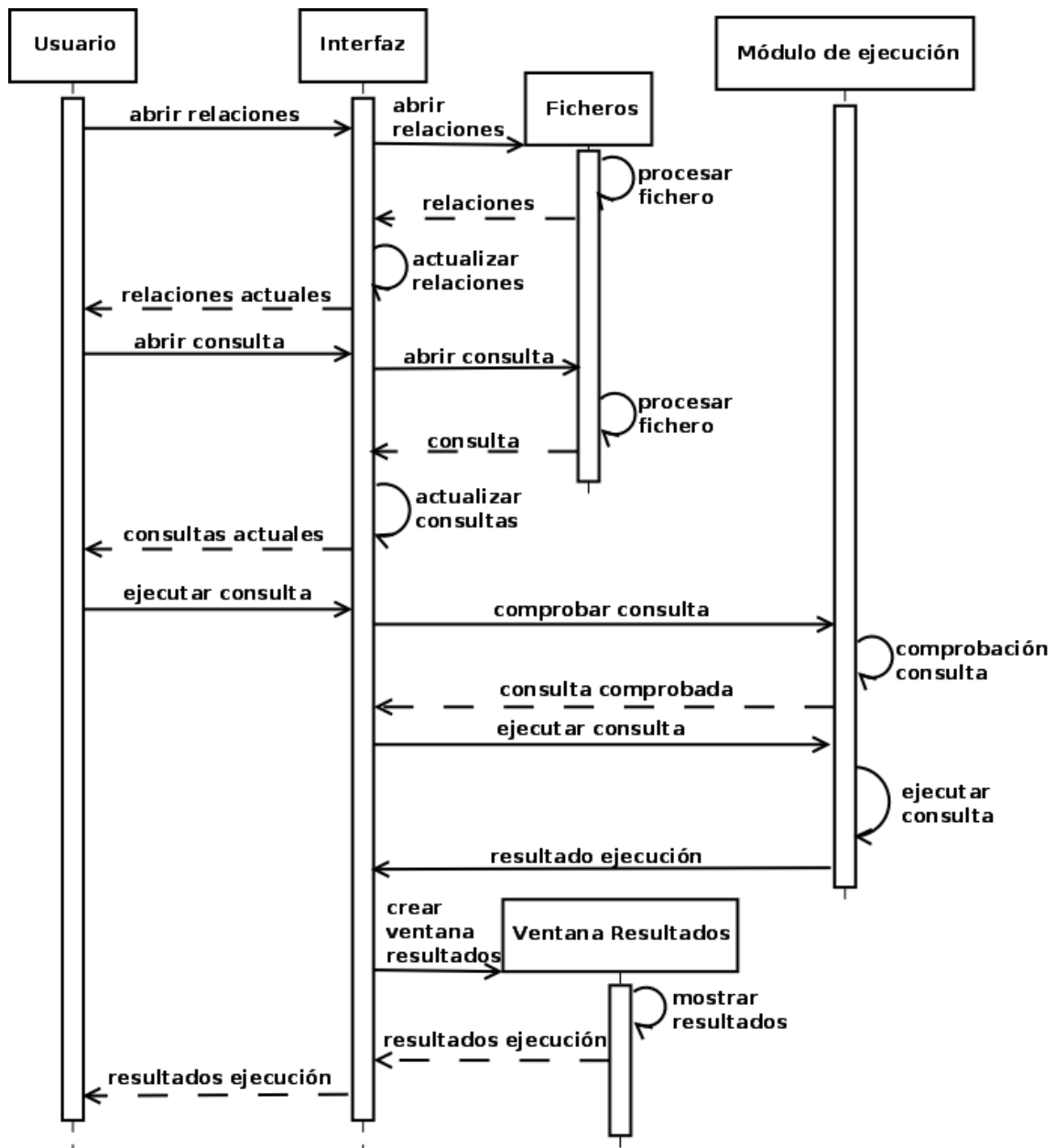


Figura C.4: Ejemplo de traza de eventos para la ejecución de una consulta

C.4.1. Módulo de interfaz gráfica

Este módulo contiene todas las clases necesarias para el manejo de la interfaz de la aplicación.

Ventana principal

La ventana principal esta formada por el menú, la barra de herramientas general y los paneles de relaciones, relación seleccionada y creación de consultas. En la Figura C.5 se puede ver el diagrama de clases de la ventana principal (las clases en color verde son proporcionadas por las bibliotecas de Java).

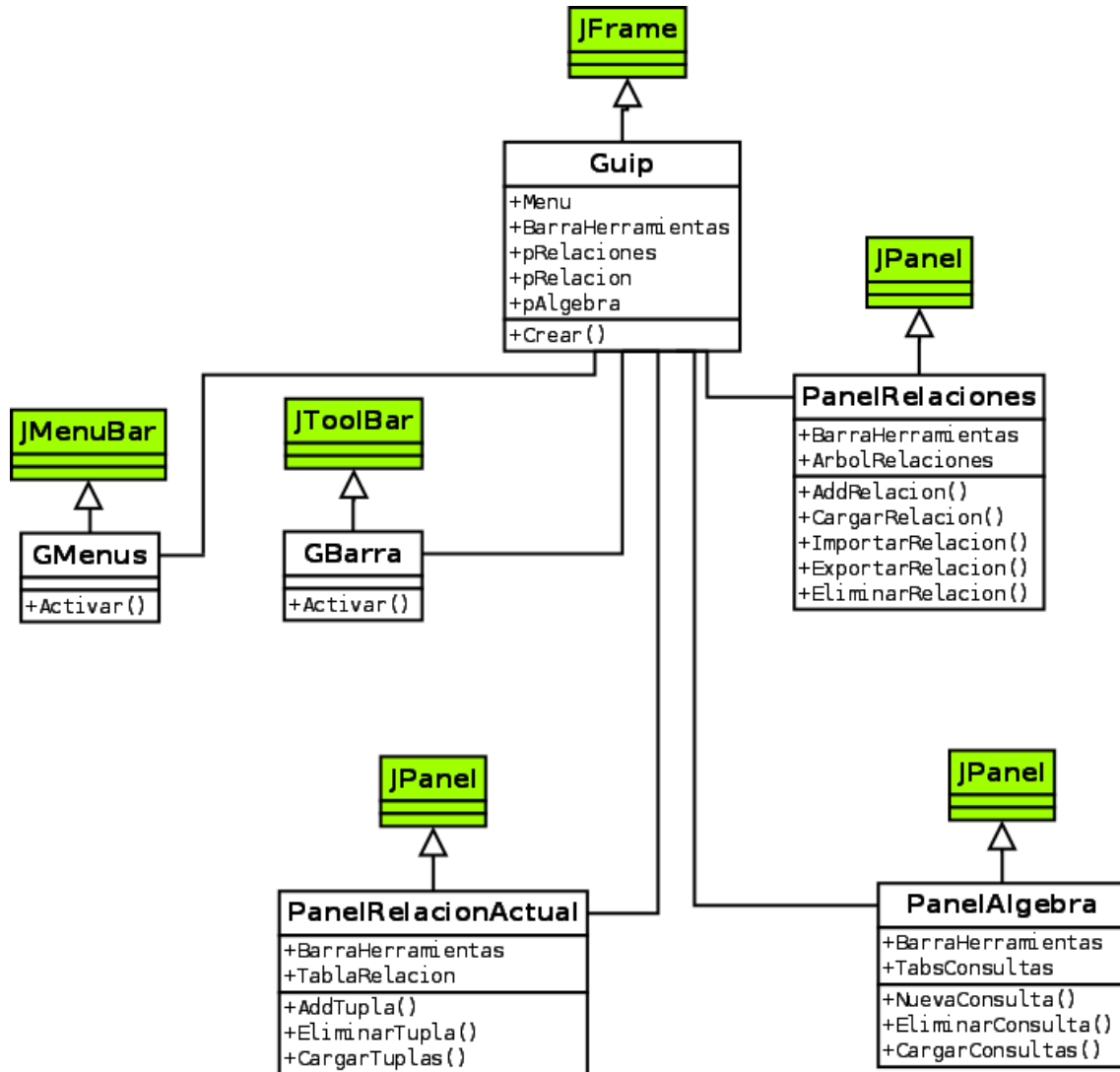


Figura C.5: Diagrama de clases de la ventana principal

Panel de relaciones

El panel de relaciones que se encarga de mostrar las relaciones del proyecto actual (ya sean importadas o creada) está compuesto de una barra de herramientas y un

árbol jerárquico donde se representan las relaciones con su número de tuplas. En la Figura C.6 se puede ver el diagrama de clases de este panel.

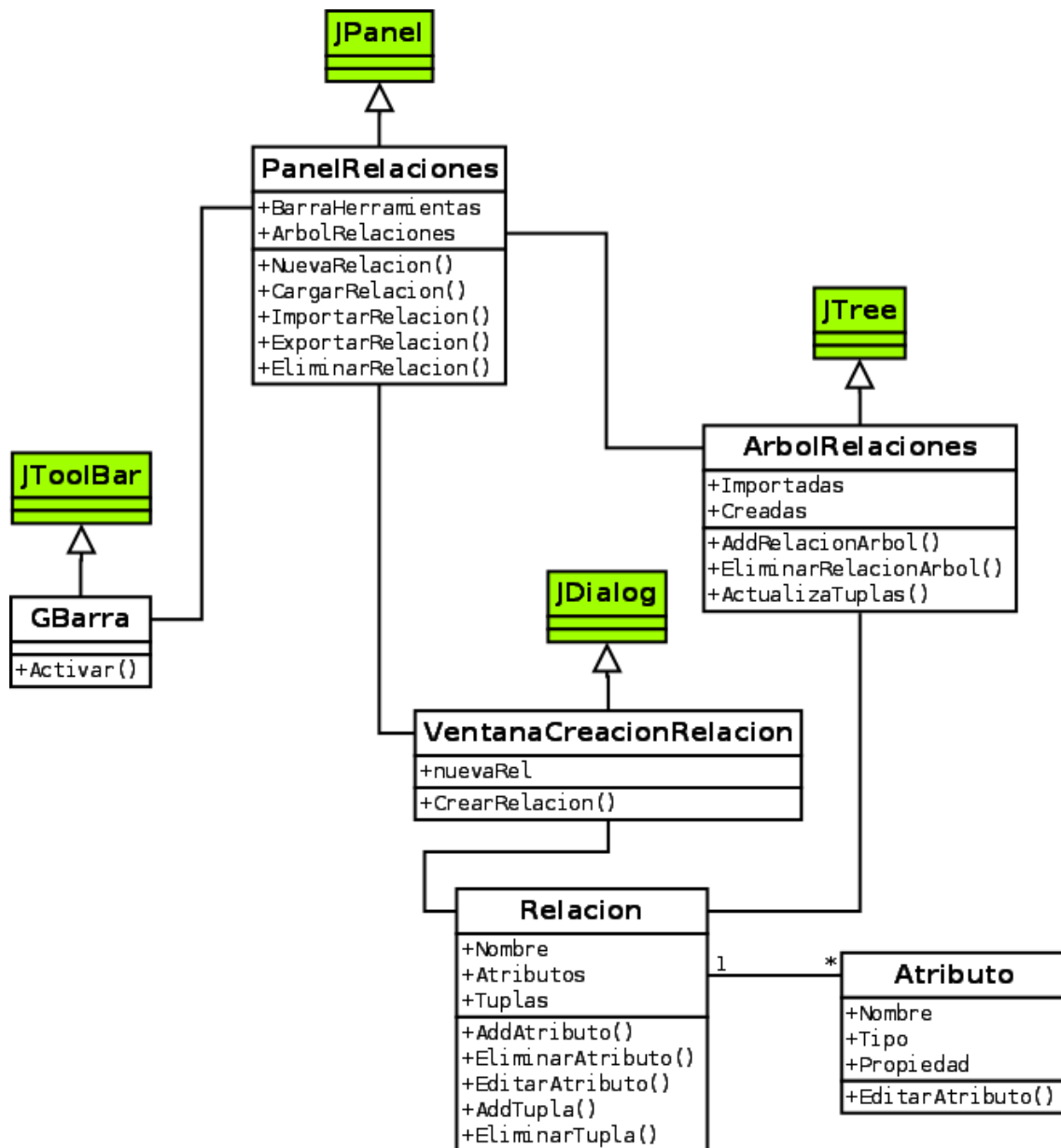


Figura C.6: Diagrama de clases del panel de relaciones

Panel de relación seleccionada

Este panel muestra la información de la relación que ha sido seleccionada. En él se encuentra una barra de herramientas con las opciones propias del panel, una tabla en la que se muestran las tuplas de la relación actual, y una barra de búsqueda

para restringir los resultados mostrados en la tabla. En la Figura C.7 se puede ver el diagrama de clases de este panel.

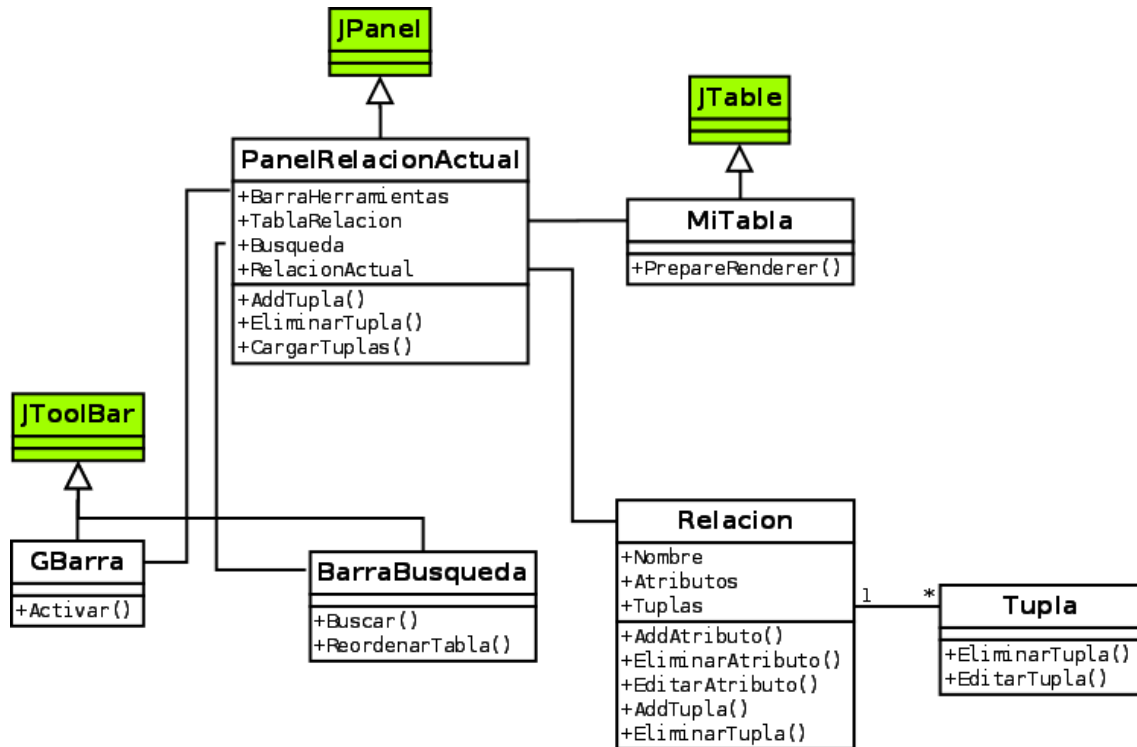


Figura C.7: Diagrama de clases del panel de relación seleccionada

Panel de expresiones en álgebra relacional

Este es el último componente de la ventana principal. En él se muestra una barra de herramientas con las opciones de este panel, una pestaña por consulta abierta en el proyecto actual, y en cada una de dichas pestañas una barra de herramientas con los operadores del álgebra relacional y el panel para la creación de consultas en modo textual o el panel para la creación de árboles en función de la opción elegida por el usuario. Si se está creando la consulta en modo textual, se nos mostrará un área de texto donde podremos introducir la descripción de la consulta y el área de texto donde deberemos introducir dicha consulta. Si nos encontráramos en la opción de creación de árboles se nos mostraría un área de texto no editable con la consulta asociada al árbol, una zona de dibujo donde crear el árbol y una barra de herramientas con las opciones propias de la creación de árboles. En la Figura C.8 se puede ver el diagrama de clases de este panel.

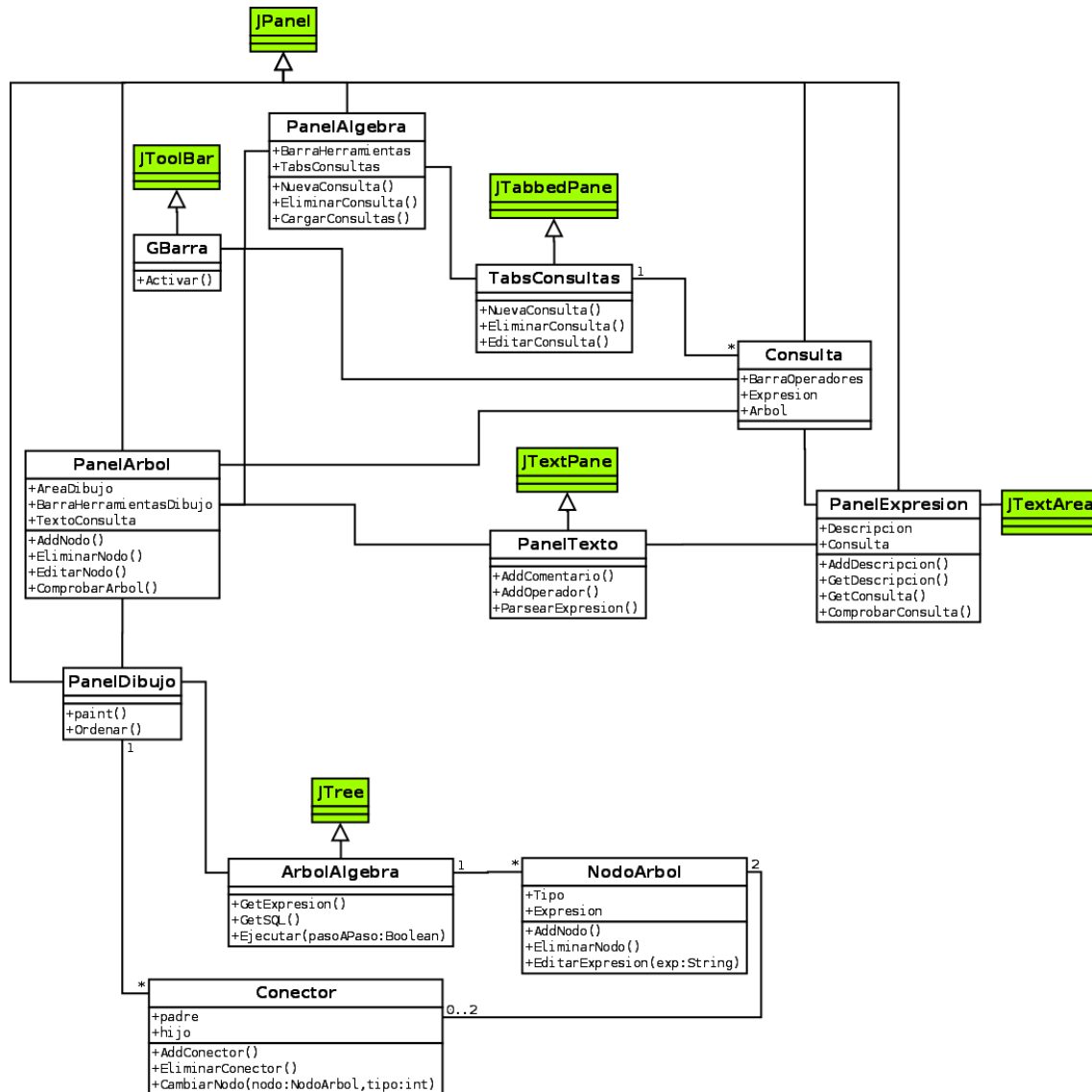


Figura C.8: Diagrama de clases del panel de expresiones

C.4.2. Módulo de álgebra relacional

El módulo de álgebra relacional contiene todas las funciones que permiten construir, ejecutar y optimizar consultas. Atendiendo a sus funcionalidades, este módulo ha sido dividido a su vez en los módulos de optimización, ejecución y definición.

Módulo de optimización

El módulo de optimización se encarga de optimizar las consultas introducidas por el usuario. Para realizar el proceso de optimización se sirve, entre otras, de las siguientes funciones:

- *OptimizarConsulta*: realiza la optimización completa de una consulta en base a las reglas de optimización de expresiones en álgebra relacional y a las estadísticas de las relaciones.
- *OptimizarConsultaPaso*: realiza la optimización de una consulta paso a paso, lo que implica que cada vez que se realiza una acción sobre la consulta, su resultado se muestre sobre una ventana que contiene la información de todos los pasos del proceso completo.
- *EstimarAtributos*: en función de los operadores de la consulta, estima qué atributos contendrá la relación resultante de aplicar esa operación para hacer posible la aplicación de las reglas de optimización en consultas no triviales.
- *EstimarTuplas*: dada una operación de la consulta a optimizar estima el número de tuplas que contendrá la relación resultante. Esto es necesario para poder aplicar la optimización basada en estadísticas de las relaciones. Así pues, se intercambian operaciones teniendo en cuenta las tuplas resultantes de aplicar las operaciones en distinto orden.

Módulo de ejecución

Este módulo implementa la ejecución de las consultas introducidas por el usuario. Como se ha comentado previamente, la ejecución puede ser llevada a cabo en la propia aplicación o en la base de datos externa, y puede realizarse de manera automática o paso a paso. Para llevar a cabo estas acciones, las funciones mas importantes del módulo son:

- *EjecutarConsulta*: ejecuta la consulta actual sobre la aplicación. Para ello se han definido métodos para cada uno de los operadores del álgebra relacional utilizados en la herramienta y en el caso de join se han implementado distintos algoritmos en función del método de join elegido.
- *EjecutarConsultaPaso*: ejecuta la consulta actual sobre la aplicación paso a paso de tal forma que el resultado de cada uno de estos pasos se muestra en una misma ventana permitiendo al usuario seguir el proceso completo.
- *EjecutarConsultaBD*: ejecuta la consulta en la base de datos externa obteniendo los resultados y mostrándolos en la aplicación.
- *EjecutarConsultaBDPaso*: ejecuta la consulta sobre la base de datos externa paso a paso y muestra el resultado de cada uno de ellos.

Módulo de definición de consultas.

La definición de las consultas, ya sea en forma textual o a través de un árbol de expresión en álgebra relacional, es el trabajo desarrollado en este módulo. Para llevar a cabo estas tareas se vale de distintas funciones, de entre las cuales las más importantes son:

- *ParsearConsulta*: utilizando como parámetro una consulta en modo textual efectúa el análisis de la misma, permitiendo de esta forma obtener el árbol asociado. En la implementación de esta función se utilizan conceptos del análisis léxico y sintáctico con el objeto de pasar de una cadena de texto a una estructura de tipo árbol.
- *ComprobarConsulta*: comprueba que la consulta introducida sea sintácticamente correcta, de tal forma que si se encuentran errores se avisa al usuario para que pueda corregirlos.
- *ComprobarArbol*: comprueba que el árbol introducido sea correcto, de tal forma que todas las hojas sean relaciones y que todas las operaciones tengan el número de parámetros necesarios.
- *ObtenerExpresionArbol*: recorre el árbol actual, obteniendo la consulta en álgebra relacional en modo textual asociada a dicho árbol.
- *ObtenerSQL*: partiendo del árbol de una expresión en álgebra relacional, traduce la consulta a SQL.

C.4.3. Módulo de manejo de ficheros

Este módulo se encarga de todas las operaciones que requieren la utilización de ficheros externos, y son las siguientes:

- *GuardarRelacion*: guarda las relaciones y tuplas actuales en un fichero XML (.rel).
- *CargarRelacion*: carga las relaciones y las tuplas existentes en un fichero XML (.rel).
- *GuardarConsulta*: guarda la consulta seleccionada en un fichero XML (.que).
- *CargarConsulta*: carga la consulta existente en un fichero XML (.que).
- *GuardarConfiguracion*: guarda la configuración actual del usuario en un fichero XML (.cfg).

- *CargarConfiguracion*: carga la configuración del usuario desde un fichero XML (.cfg).
- *GuardarProyecto*: guarda el proyecto actual, incluyendo todas las relaciones, tuplas y consultas, en un fichero XML (.ral).
- *CargarProyecto*: carga un proyecto, incluyendo todas las relaciones, tuplas y consultas, existentes en un fichero XML (.ral).

C.4.4. Módulo de manejo de base de datos externa

Este módulo es el encargado de gestionar las comunicaciones con la base de datos externa. Sus funciones principales son:

- *Conectar/desconectar*: realiza la conexión/desconexión de la base de datos externa informando de los posibles errores encontrados.
- *EjecutarConsulta*: ejecuta la consulta introducida por el usuario sobre la base de datos externa y devuelve sus resultados, además de los posibles errores en la ejecución.
- *ObtenerRelaciones*: obtiene las relaciones (tablas) que el usuario tiene definidas en la base de datos externa.
- *ObtenerTuplas*: obtiene las tuplas de las relaciones de la base de datos que el usuario quiera importar.
- *ExportarRelacion*: exporta la relación seleccionada y sus tuplas a la base de datos externa, gestionando los posibles errores al interactuar con el SGBD.

C.4.5. Comunicación entre módulos

En la Figura C.9 podemos ver un diagrama que muestra la comunicación entre los módulos diseñados. Comentando dicho diagrama podemos decir que el módulo de interfaz gráfica servirá como conexión entre los diferentes módulos del sistema. Se ha optado por esta decisión puesto que un control central de la comunicación nos permitirá gestionar de una forma más adecuada los recursos y los posibles errores. De esta forma, el módulo de álgebra relacional no accederá directamente al sistema gestor de bases de datos ni a los diferentes ficheros externos, sino que será necesaria la comunicación con el módulo de interfaz gráfica para la extracción de la información. Es decir, la obtención de las relaciones (y sus tuplas), ya sea desde la base de datos o desde un fichero externo, será realizada por el módulo de interfaz gráfica (puesto que el usuario confirmará todas estas acciones a través de componentes visuales), y a continuación serán transferidos dichos datos al módulo de álgebra relacional correspondiente.

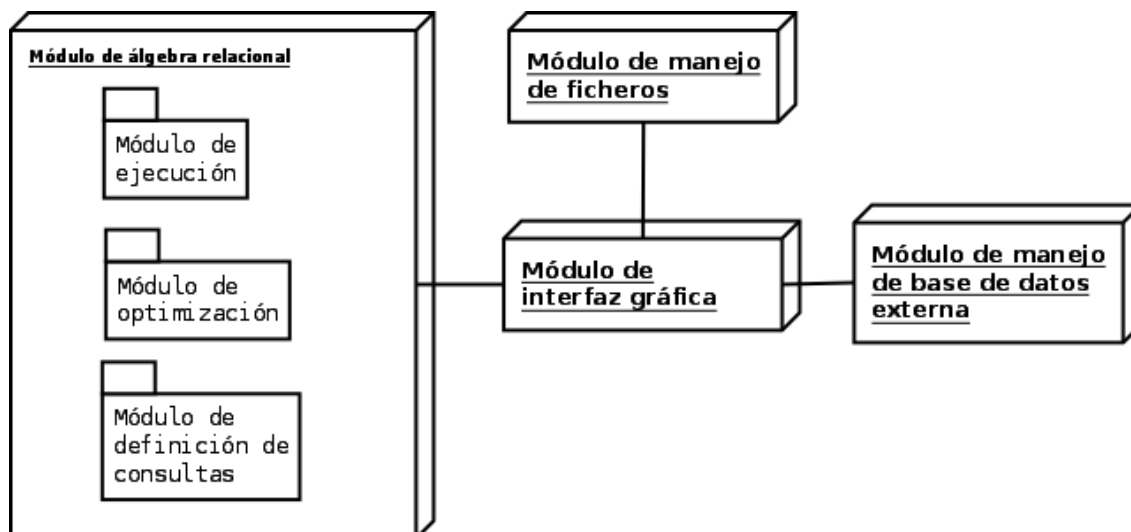


Figura C.9: Comunicación entre los módulos del sistema

C.5. Prototipado de la ventana principal

Una vez elaborado el análisis y el diseño de los diferentes componentes de la aplicación se empezó a pensar en diseñar la interfaz gráfica. De este proceso de diseño salieron varios prototipos para la ventana principal que fueron estudiados en diferentes reuniones junto al director del proyecto, que actuaba en este caso como cliente. El primero de estos prototipos (Figura C.10) trato de aunar todos los componentes de la aplicación en un solo panel, de tal forma que la interfaz no resultara muy complicada. Enseguida se vieron problemas, puesto que este tipo de diseño hacía necesaria la navegación entre las pestañas para poder obtener toda la información y esto podría resultar molesto para el usuario. Pero de este diseño se obtuvo una idea interesante y que sería profundizada mas adelante, la necesidad de que cada componente de la aplicación tuviera su propia barra de herramientas con las opciones específicas de ese componente.

El segundo prototipo (ver Figura C.11) trató de evitar el problema que había surgido con el primero. En este diseño se planteó la posibilidad de que toda la información estuviera visible en el mismo momento de tal forma que evitábamos al usuario tener que navegar en exceso entre las opciones. Aunque este diseño parecía acogerse a la idea que teníamos para la aplicación surgió un nuevo problema. Al tratarse de ventanas independientes unas de otras, el usuario necesitaría tiempo para ajustarlas en función a su resolución de pantalla y cada cambio producido en una ventana requeriría de un reajuste, lo cual nos pareció que conllevaría una pérdida de tiempo que podría evitarse.

Así pues, se creo otro prototipo más (Figura C.12) que utilizaba las ideas extraídas de los dos anteriores para crear con ellas una interfaz mixta que permitiría al usuario tener toda la información en pantalla sin necesidad de muchas navega-

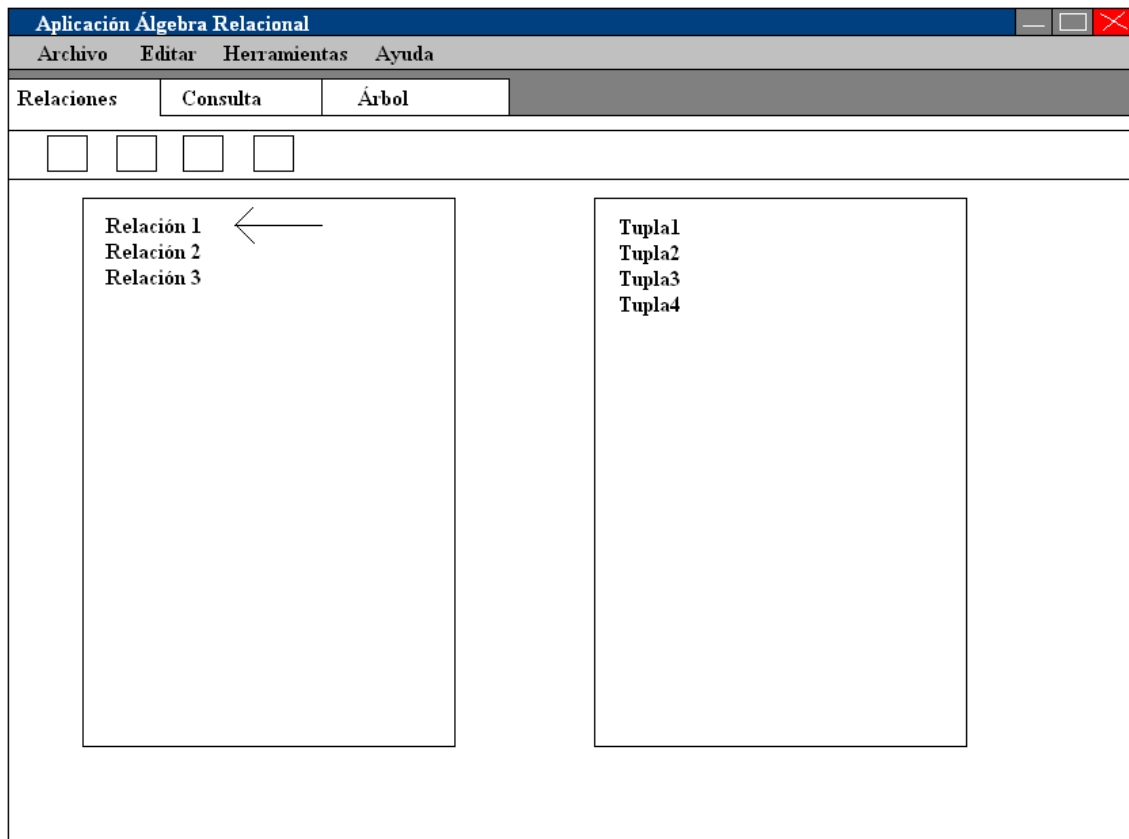


Figura C.10: Primer prototipo de la ventana principal

ciones y cuyas partes se reestructurarían automáticamente permitiendo al usuario elegir el tamaño de una parte en concreto reorganizando el resto. Se ha incluido también la Figura C.13 que muestra el aspecto real de la aplicación para que se pueda comprobar cómo el prototipo ha sido adaptado de una forma satisfactoria.

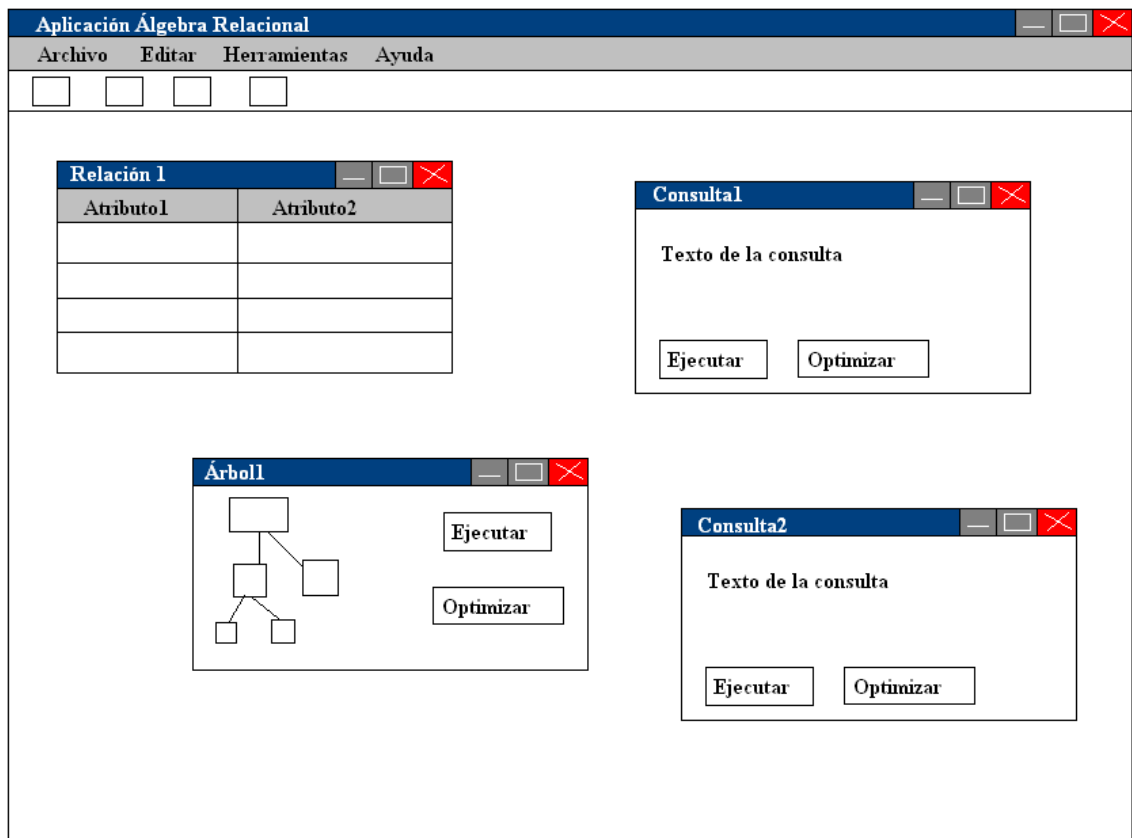


Figura C.11: Segundo prototipo de la ventana principal

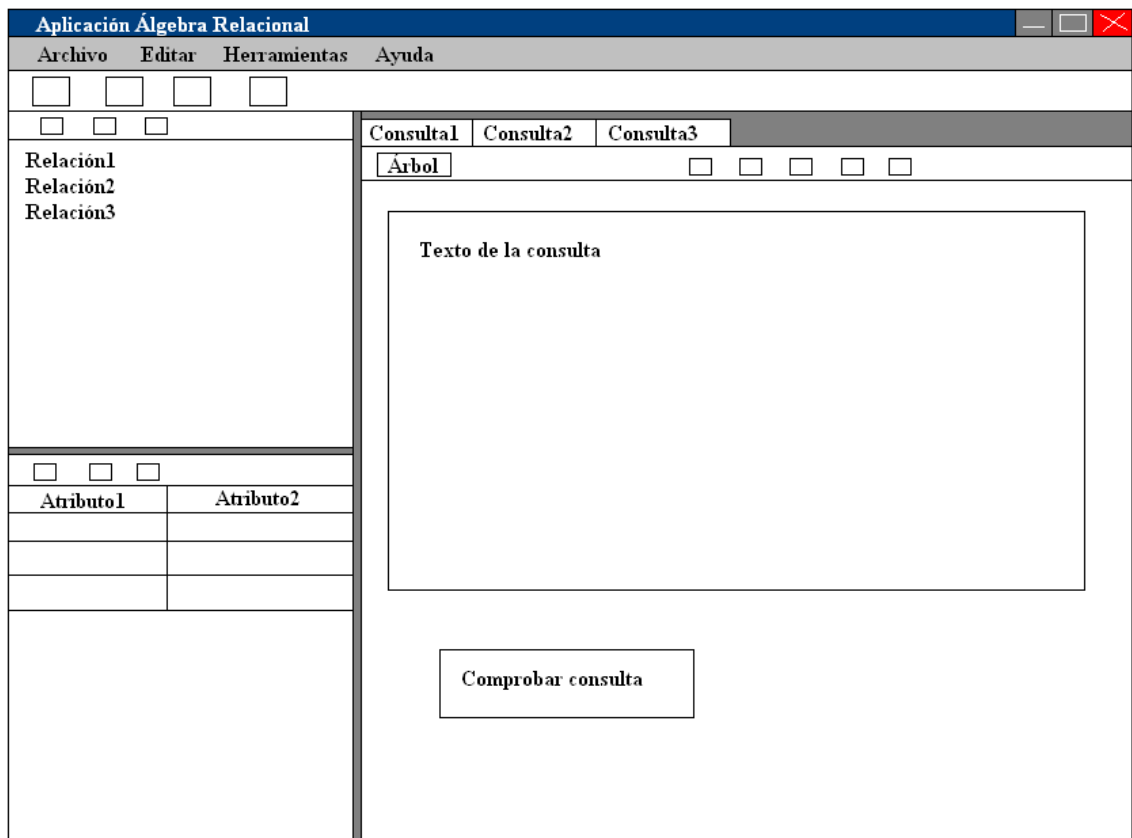


Figura C.12: Prototipo final de la ventana principal

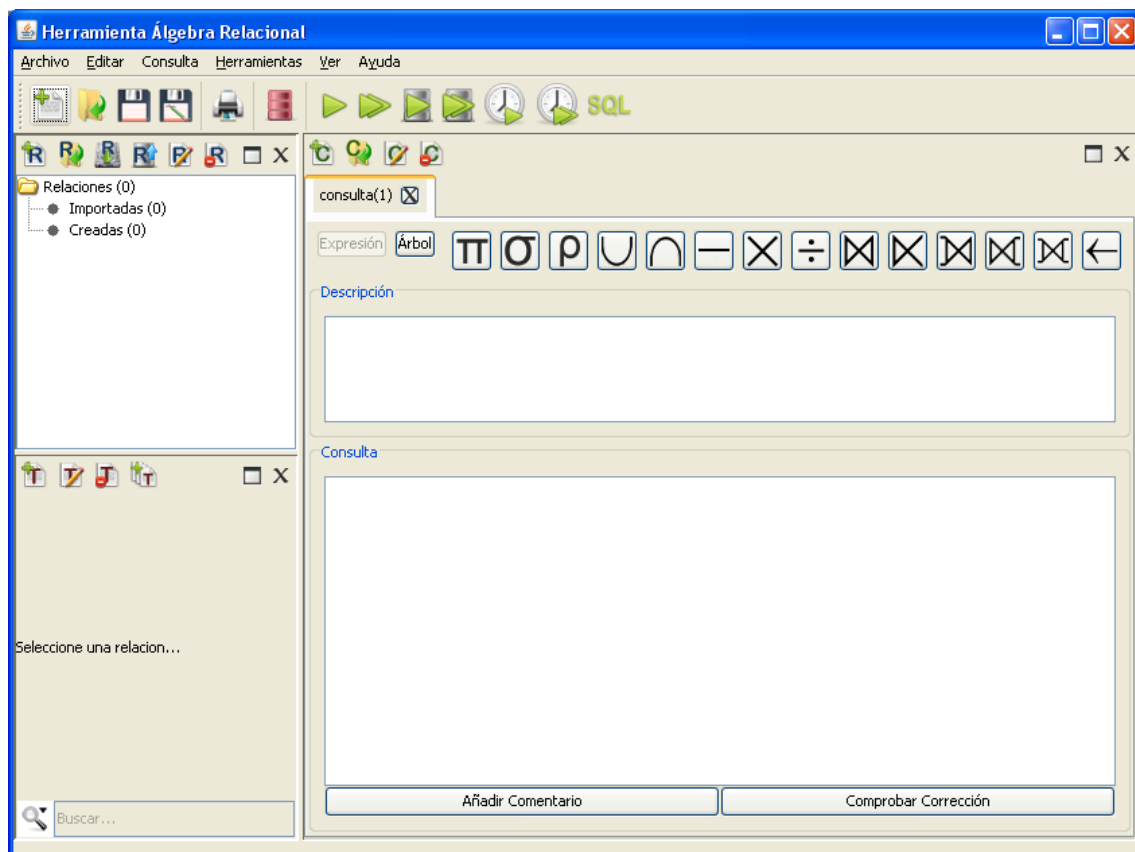


Figura C.13: Aspecto final de la ventana principal

Apéndice D

Ejemplos de uso

En este anexo se van a presentar distintos ejemplos de uso de la herramienta, detallando el funcionamiento de las principales operaciones que realiza, como son la introducción de relaciones, la definición de expresiones en álgebra relacional, la definición de árboles de expresiones en álgebra relacional, la ejecución de consultas, la optimización de consultas y su traducción a SQL.

D.1. Introducción de relaciones

Nuestro objetivo va a ser la creación de una nueva relación que va a representar a una persona. Queremos almacenar de dicha persona la siguiente información:

- DNI
- Nombre
- Edad
- Peso

El primer paso que seguimos es pulsar el botón de *Nueva relación* que aparece en la barra de herramientas del panel de relaciones. La ventana de introducción de datos nos aparecerá en pantalla (Figura D.2), procedemos a introducir el nombre y hacemos click en el botón de *Nuevo atributo* cuatro veces indicando que la relación va a tener cuatro atributos, tras lo cual la ventana de creación pasará a verse como en la Figura D.2.

Una vez que aceptemos, la nueva relación será creada y se podrá ver en el árbol jerárquico de relaciones actuales bajo la sección de *Creadas* (en la Figura D.3 puede ver el resultado final).

Creación de relación

Nombre de la relación:

Nuevo atributo

Atributos

Aceptar Cancelar

Figura D.1: Creación de una relación

Creación de relación

Nombre de la relación:

Nuevo atributo

Atributos

DNI	number	primary key	Eliminar
Nombre	varchar		Eliminar
Edad	number		Eliminar
Peso	number		Eliminar

Aceptar Cancelar

Figura D.2: Datos introducidos para la relación

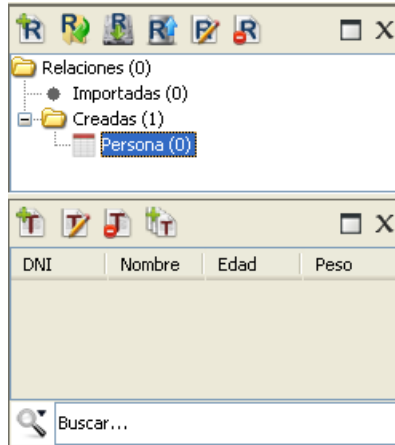


Figura D.3: Nueva relación creada

D.2. Introducción de tuplas para una relación

A continuación y partiendo de la relación creada en la sección anterior, vamos a introducirle unas cuantas tuplas. Primero introduciremos una tupla de forma manual, para ello hacemos click en el botón *Nueva tupla* de la ventana de relación seleccionada y se nos mostrará una ventana en la que escribiremos los valores de los atributos separados por comas (ver Figura D.4).

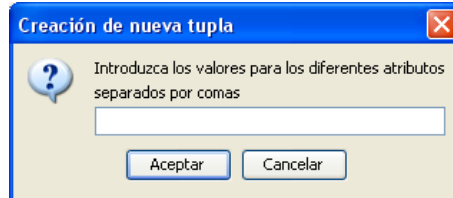


Figura D.4: Ventana inserción tupla

Queremos guardar la información de *Pedro*, con DNI *77777777* que tiene *25* años y que pesa *90* kilos, así que procedemos a transcribir esta información a la ventana quedando como resultado lo que aparece en la Figura D.5 (nótese que hemos introducido dos veces el peso de manera intencionada para ver el error que se produciría).

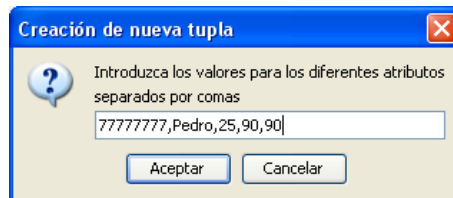


Figura D.5: Ventana inserción tupla con datos insertados

Se nos muestra a continuación un error (Figura D.6) por haber introducido un dato más de lo que se esperaba, así que se nos vuelve a mostrar la ventana con los datos que habíamos metido para que solucionemos el problema.

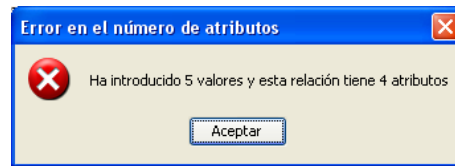


Figura D.6: Ventana de error en la introducción de datos

Una vez arreglado esto, la nueva tupla se crea y se hace visible en el panel de relación actual (ver Figura D.7).

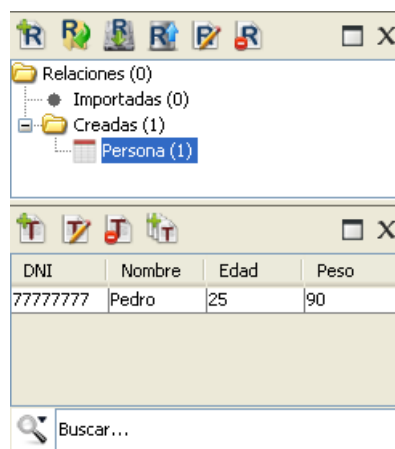


Figura D.7: Nueva tupla creada

A continuación introduciremos 40 tuplas de forma automática, para ello tenemos que hacer click en el botón de *Generar tuplas* de la barra de herramientas de la ventana de relación seleccionada. Después se nos mostrará la ventana que posibilita el guiado para la generación automática de tuplas (ver Figura D.8). En dicha ventana podemos introducir ciertos parámetros para que la generación automática cree tuplas parecidas a la que hemos metido manualmente.

Así pues, vamos a decirle que queremos que los nombres se extraigan de un fichero externo en el que hemos metido unos cuantos nombres separados por comas (ver Figura D.9), le diremos también que la edad va a estar comprendida entre los 20 y los 50 años, y que el peso va a tener como limite inferior los 50 kilos y como máximo los 100. No vamos a introducir ningún limite para el DNI puesto que al ser la clave primaria de la relación dejaremos que la aplicación se encargue de generar valores adecuados para este campo.

El resultado una vez introducidos estos parámetros se puede ver en la Figura D.10.

Generación de tuplas

DNI

Valor mínimo: Valor máximo:

Nombre

Seleccione el fichero de texto para extraer los datos:

Edad

Valor mínimo: Valor máximo:

Peso

Valor mínimo: Valor máximo:

Figura D.8: Ventana para generación guiada de tuplas

nombres - Bloc de notas

Archivo Edición Formato Ver Ayuda

Pepe, Juan, María, Pedro, Roberto, Sergio, Angel, Alberto, Marta

Figura D.9: Fichero de nombres

Generación de tuplas

DNI

Valor mínimo: Valor máximo:

Nombre

Seleccione el fichero de texto para extraer los datos: C:\a relacional\nombres.txt

Edad

Valor mínimo: Valor máximo:

Peso

Valor mínimo: Valor máximo:

Figura D.10: Ventana de generación de tuplas con datos

Una vez que presionemos el botón de *Generar* se nos requiere el número de tuplas que queremos que cree, así que introducimos 40 tuplas (ver Figura D.11).

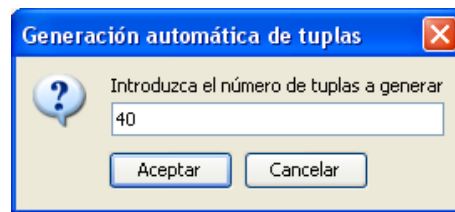


Figura D.11: Número de tuplas a generar

Tras esto la aplicación generará automáticamente las tuplas según el patrón que le hemos marcado y aparecerán en la ventana de relación actual, como se puede ver en la Figura D.12.

DNI	Nombre	Edad	Peso
77777777	Pedro	25	90
77777779	Maria	47	93
77777780	Sergio	41	77
77777781	Marta	46	57
77777782	Pepe	43	95
77777783	Sergio	46	71
77777784	Marta	27	79
77777785	Pepe	24	95
77777786	Maria	35	80
77777787	Roberto	20	77
77777788	Sergio	21	77
77777789	Juan	45	53
77777790	Angel	31	92
77777791	Marta	28	74
77777792	Alberto	20	81
77777793	Roberto	25	53

Figura D.12: Tuplas generadas de forma satisfactoria

No estamos seguros de si habrá introducido alguna tupla que tenga para el valor de peso *58* (ya que se genera un valor aleatorio entre los límites propuestos) y para nosotros es importante que este valor no aparezca nunca (en este ejemplo concreto de uso), así que vamos a hacer uso de la opción de búsqueda integrada en la ventana de relación actual. Hacemos click en primer lugar en el botón con forma de lupa para seleccionar el atributo *Peso*, que es el que vamos a consultar, e introducimos

en el cuadro de texto 58, la búsqueda se va realizando en tiempo real así que cuando metemos el 5 se nos muestran todas aquellas tuplas en las que su valor para el *Peso* empieza por 5. Y el resultado de la búsqueda se puede ver en la Figura D.13.

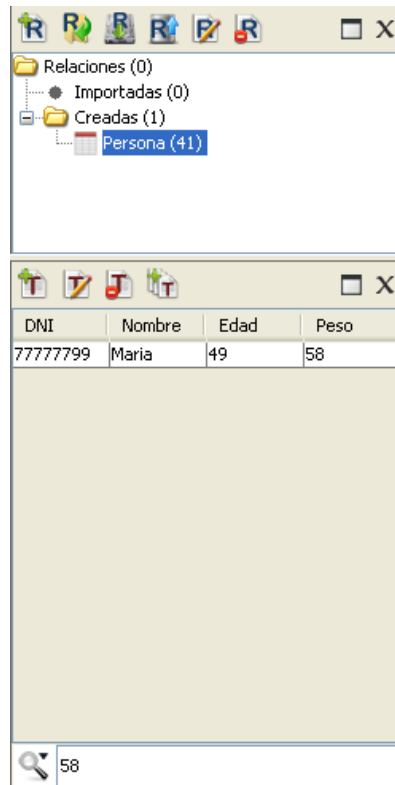


Figura D.13: Búsqueda de un valor entre las tuplas

Ahora vamos a borrar la tupla cuyo valor de *Peso* es 58. Para ello, en primer lugar, simplemente hacemos click en la tupla para que quede seleccionada, y posteriormente hacemos click en el botón *Eliminar tupla* de la barra de herramientas de la ventana de relación seleccionada. Se nos pide confirmación, al tratarse de una operación destructiva, como se puede ver en la Figura D.14.

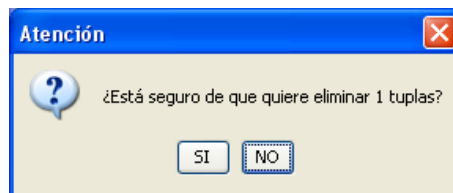


Figura D.14: Confirmación para la eliminación de tuplas

Confirmamos la operación pulsando el botón *SI* y la tupla queda eliminada de las tuplas de la relación *Persona*.

D.3. Creación de una consulta en álgebra relacional

Vamos a proceder ahora a crear una consulta en álgebra relacional. La consulta que vamos a generar utiliza relaciones pertenecientes a la *Base de datos de la Universidad* introducida en la asignatura *Ficheros y Bases de Datos*. En la Figura D.15 se puede ver el esquema de dicha base de datos para que se comprenda mejor el proceso que vamos a ir siguiendo.

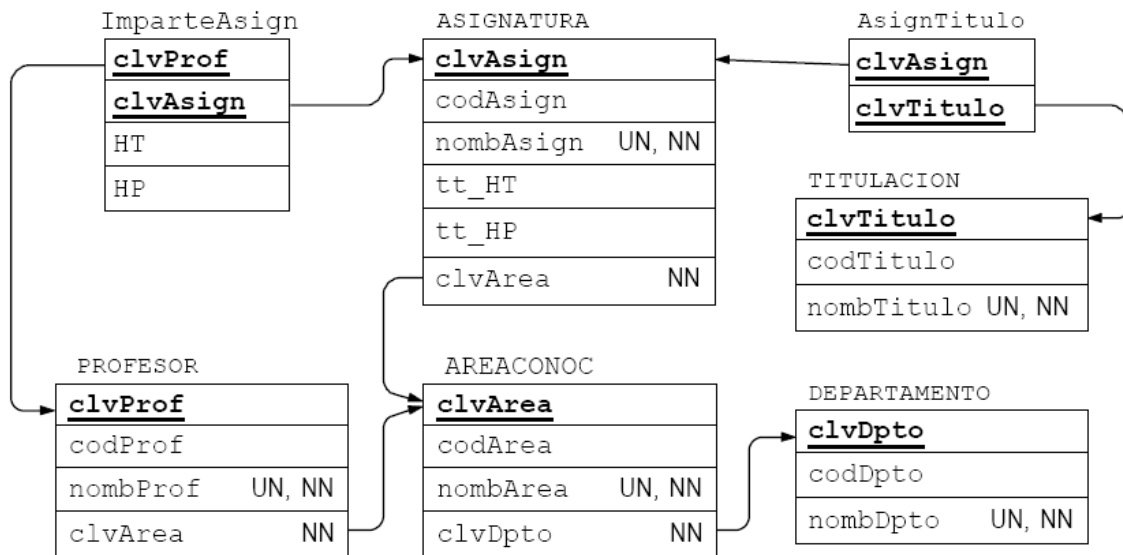


Figura D.15: Base de Datos de la Universidad (*Ficheros y Bases de Datos*)

En primer lugar vamos a introducir, utilizando la misma base que en los ejemplos explicados anteriormente, las relaciones a la aplicación y vamos a crear una serie de tuplas de forma automática. Dado a que el proceso ya ha sido explicado mostramos solamente el resultado final (ver Figura D.16).

Una vez tenemos las relaciones con sus tuplas vamos a empezar a crear la consulta. Para este ejemplo ha sido elegida una consulta aparecida en el examen de la asignatura de la convocatoria de septiembre del año 2002 y responde al título:

“Listado de los pares($clvDpto, clvArea$) correspondientes a los departamentos que imparten docencia en al menos todas las asignaturas encargadas a dicha área. No se incluirán las áreas que no tienen encargada ninguna asignatura.”

Vamos a utilizar la creación de consultas en modo árbol, ya que nos va a permitir utilizar una forma más gráfica de representación.

La primera idea que nos surge al contemplar esta consulta es que vamos a necesitar realizar diferentes pasos intermedios ayudándonos del operador de asignación.

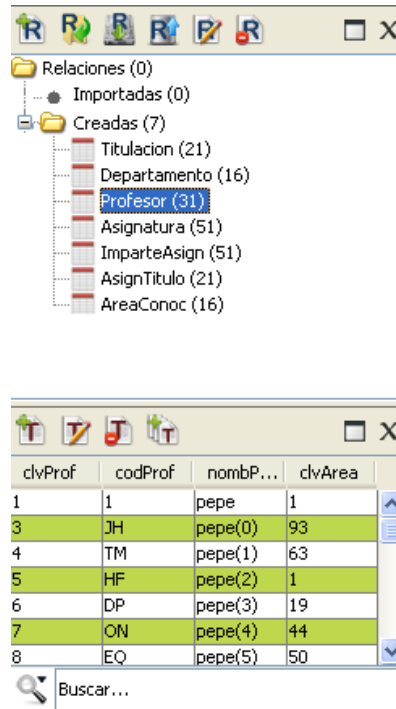


Figura D.16: Generación de relaciones de la BD de la universidad

Así pues, en primer lugar vamos a necesitar obtener los departamentos con sus asignaturas impartidas y con el área de dichas asignaturas. Descompondremos esta parte para realizarla de forma más sencilla. Primero calcularemos las asignaturas junto con el área que tiene el encargo, para ello haremos una proyección de los atributos *clvAsign* y *clvArea* de la relación *Asignatura*, como se puede ver en la Figura D.17.

Por otro lado también vamos a necesitar los departamentos con las asignaturas que imparten, así que vamos a asignar la parte que hemos escrito antes a una relación *R1* para poder utilizarla posteriormente, y vamos a crear este nuevo paso. Para este paso vamos a necesitar los atributos *clvDpto* y *clvAsign*, y además vamos a tener que realizar dos operaciones de join de tal forma que reunamos la información de las relaciones *AreaConoc*, *Profesor*, *ImparteAsign*. Teniendo en cuenta que necesitamos que las relaciones a reunir cuenten con uno o mas atributos con el mismo nombre, decidimos efectuar los joins como se puede ver en la Figura D.19.

Llegados a este momento, vamos a realizar la ejecución paso a paso de lo que llevamos de la consulta para asegurarnos de que vamos por buen camino. Hacemos click en el botón de ejecución paso a paso y comprobamos la consulta. En la Figura D.19 se puede ver parte de la ventana de resultados que se nos ofrece y donde podemos ver el resultado de aplicar el segundo join. De momento la consulta parece correcta, atendiendo a los resultados generados, así que proseguimos con la siguiente parte.

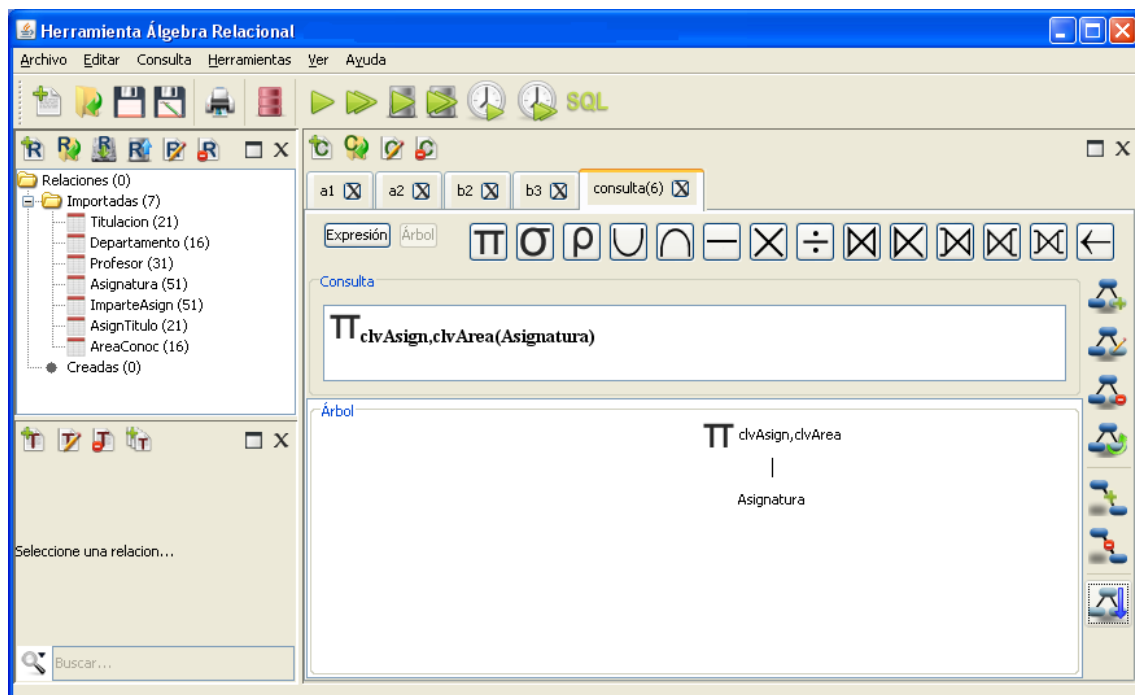


Figura D.17: Ejemplo de creación de consulta paso 1

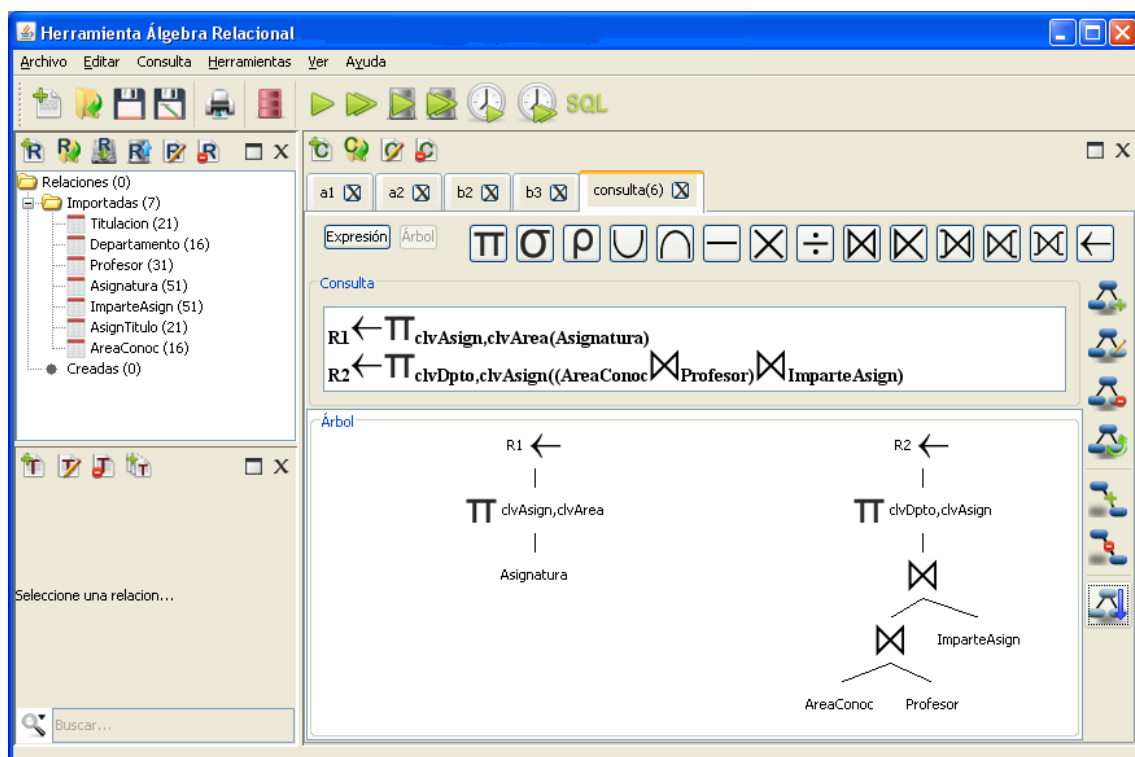


Figura D.18: Ejemplo de creación de consulta paso 2

Resultado de la ejecución

Paso0 Paso1 Paso2 Paso3 Paso4 Paso5

Consulta

(AreaConoc X Profesor) X ImparteAsign

Relación resultado

clvArea	codA...	nomb...	clvDpto	clvProf	codProf	nomb...	clvAs...	HT	HP
1	1	RyF	1	1	1	Pedro	1	10	15
1	1	DD	1	5	HF	Juan	5	93	38
13	EF	AyR	80	12	RU	Maria	12	40	21
5	KE	ISB	69	20	TG	Sergio	20	19	32

Buscar...

Figura D.19: Ejemplo de creación de consulta paso 3

Ahora estamos en condiciones de poder obtener los departamentos con sus asignaturas impartidas y con el área de dichas asignaturas. Para ello y valiéndonos de los resultados parciales que hemos ido obteniendo, no tenemos mas que realizar el join de dichos resultados (que habían sido almacenados en sendas relaciones $R1$ y $R2$ gracias al operador de asignación). Además, vamos a hacer la proyección de los atributos que nos interesan para no arrastrar información innecesaria a lo largo de la consulta. En la Figura D.20 se puede ver el aspecto del árbol después de haber aplicado lo explicado.

Ahora sería necesario obtener todas las posibles imparticiones de las asignaturas por los departamentos, junto con el área a que están asignadas. Para ello utilizaremos el operador de producto cartesiano y dos de las relaciones que hemos creado en los pasos anteriores. Como se puede ver en la Figura D.21 hemos añadido esta operación al árbol y vamos a comprobar que de momento vayan bien las cosas, así que procedemos a ejecutar paso a paso la consulta. Nada mas hacer click en el botón de ejecución paso a paso se nos muestra un mensaje de error que nos informa que la relación $R1$ no puede ser creada, puesto que ya existe (ver Figura D.22). Esta relación ha sido creada cuando hemos ejecutado la consulta paso a paso anteriormente. Así que para poder continuar borramos las relaciones $R1$ y $R2$ que se nos han creado en la ejecución anterior. Una vez solucionado esto, ejecutamos paso a paso la consulta y en la Figura D.23 se puede ver el resultado.

Hemos comprobado que todo funciona como debe y recapitulando tenemos asignado a la relación $R3$ los departamentos, asignaturas impartidas y área de las asignaturas; y a la relación $R4$ todas las posibles imparticiones de las asignaturas por los departamentos, junto con el área a que están asignadas. Así que si procedemos a realizar la resta de ambas relaciones nos quedaremos con los departamentos con

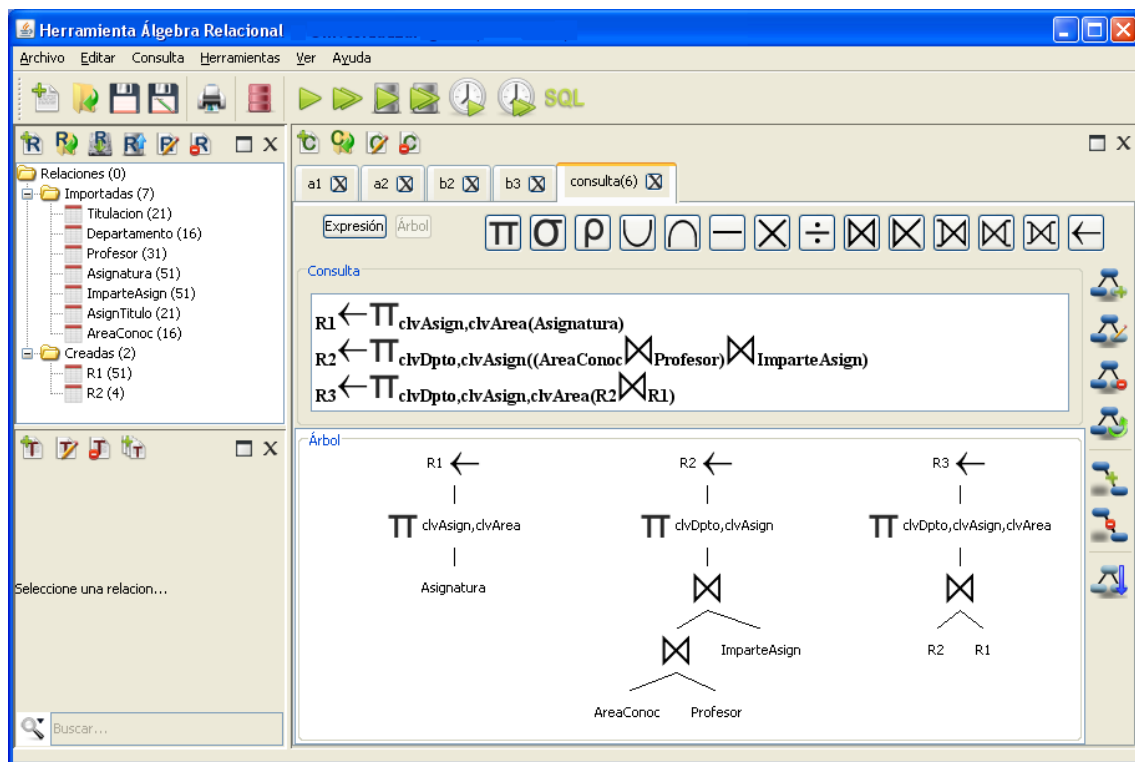


Figura D.20: Ejemplo de creación de consulta paso 4

asignaturas no impartidas y el área a que están asignadas. Una vez realicemos dicha resta obtendremos los departamentos que *no* cumplen lo que nos pedían en el enunciado de la consulta y todos los departamentos junto a las asignaturas y sus áreas, podemos restar estas dos relaciones y obtendremos lo que andábamos buscando, el listado de los pares (*clvDpto, clvArea*) correspondientes a los departamentos que imparten docencia en al menos todas las asignaturas encargadas a dicha área. Puede haber más soluciones para la consulta propuesta, pero en la Figura D.24 se puede ver el aspecto final del árbol de la solución desarrollada en este ejemplo. Y en la Figura D.25 se puede ver el resultado de ejecutar esta consulta sobre las relaciones de ejemplo.

D.4. Ejecución de una consulta en la base de datos externa

Partiendo de la consulta creada en el ejemplo anterior, vamos a ejecutarla sobre la base de datos externa. Lo primero de todo es conectarnos a dicha base de datos externa ya que no lo estábamos (el botón de conexión aparece en rojo {} mostrando que no estamos conectados). Así pues hacemos click en el botón de conexión y a continuación escribimos nuestros datos en la ventana que aparece (ver Figura D.26).

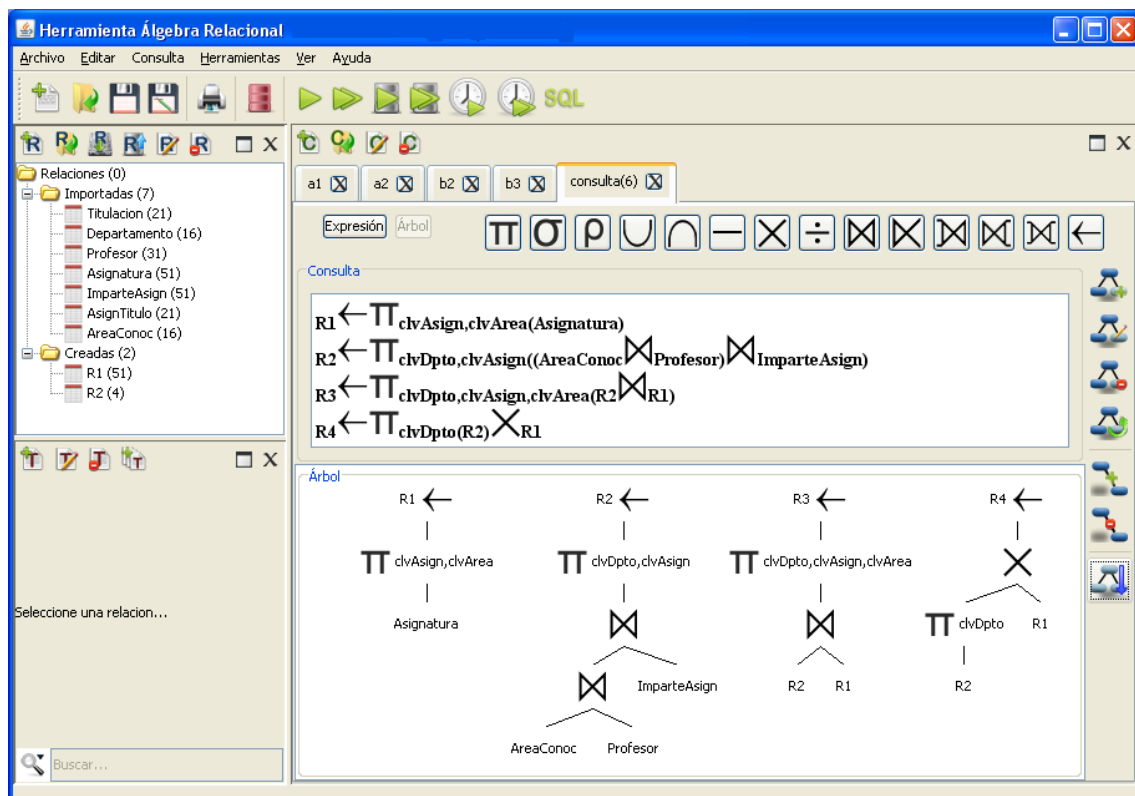


Figura D.21: Ejemplo de creación de consulta paso 5

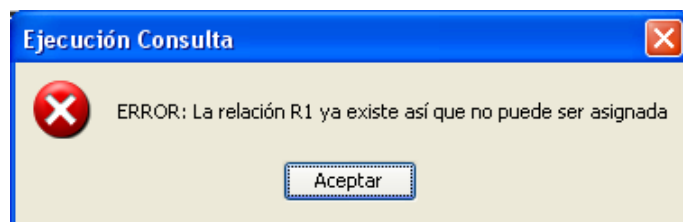


Figura D.22: Ejemplo de creación de consulta paso 6

El siguiente paso es exportar todas las relaciones que vayan a ser utilizadas en la consulta y que no se encuentren ya en la base de datos. Para cada una de las relaciones hacemos click en el botón de *Exportar relación*, de tal forma que el árbol jerárquico de relaciones pasa a tener el aspecto que se puede ver en la Figura D.27.

Ahora estamos en condiciones de realizar la ejecución en la base de datos externa. Para ello hacemos click en *Ejecutar en Base de Datos* y la consulta se ejecuta y se nos muestra el resultado (puede verse la traducción a SQL de la consulta en la Figura D.28).

Resultado de la ejecución

Paso0 Paso1 Paso2 Paso3 Paso4 Paso5 Paso6 Paso7 Paso8 Paso9 Paso10 Paso11

Consulta

$\Pi_{clvDpto(R2)} \bowtie R1$

Relación resultado

clvDpto	clvAssign	clvArea
1	1	1
1	3	16
1	4	44
1	5	13
1	6	40
1	7	53
1	8	93
1	9	55
1	10	5
1	11	22
1	12	12
1	13	55
1	14	54
1	15	20
1	16	30
1	17	62
1	18	86
1	19	21
1	20	48
1	21	15
1	22	6
1	23	22
1	24	80
1	25	52
1	26	31

Buscar...

Figura D.23: Ejemplo de creación de consulta paso 7

D.5. Optimización de una consulta

A continuación vamos a mostrar el proceso para la optimización de una consulta. Para ello hemos introducido una consulta sobre la *Base de Datos de la Universidad* (ver Figura D.15) en la aplicación mediante al proceso antes explicado (ver Figura D.29). También hemos cargado una serie de relaciones y tuplas que nos harán falta para la consulta desde un fichero externo (ver Figura D.30).

Hacemos click en el botón de optimizar paso a paso para que se nos muestre la ventana del resultado de este proceso. Vamos a ir explicando cada uno de los pasos. En el primero de ellos obtenemos la información que se puede ver en la Figura D.32. En el campo de explicación se nos avisa de que se ha aplicado una optimización basada en estadísticas y se han intercambiado varias relaciones. En concreto, se ha comprobado que tal y como estaba definido el árbol, en primer lugar se iba a realizar

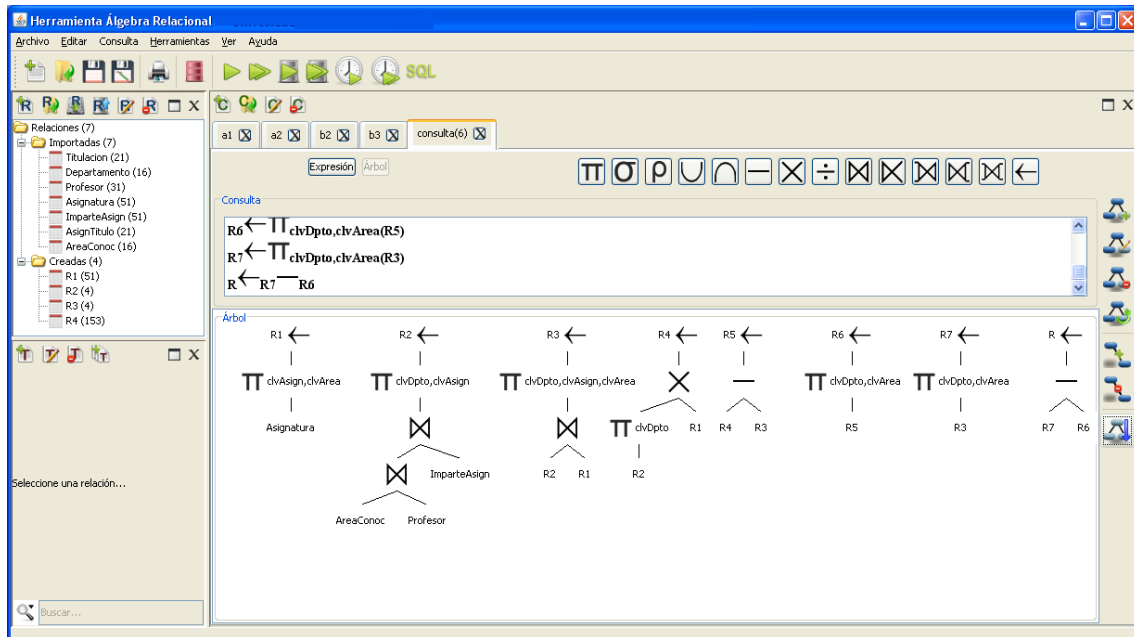


Figura D.24: Ejemplo de creación de consulta paso 8

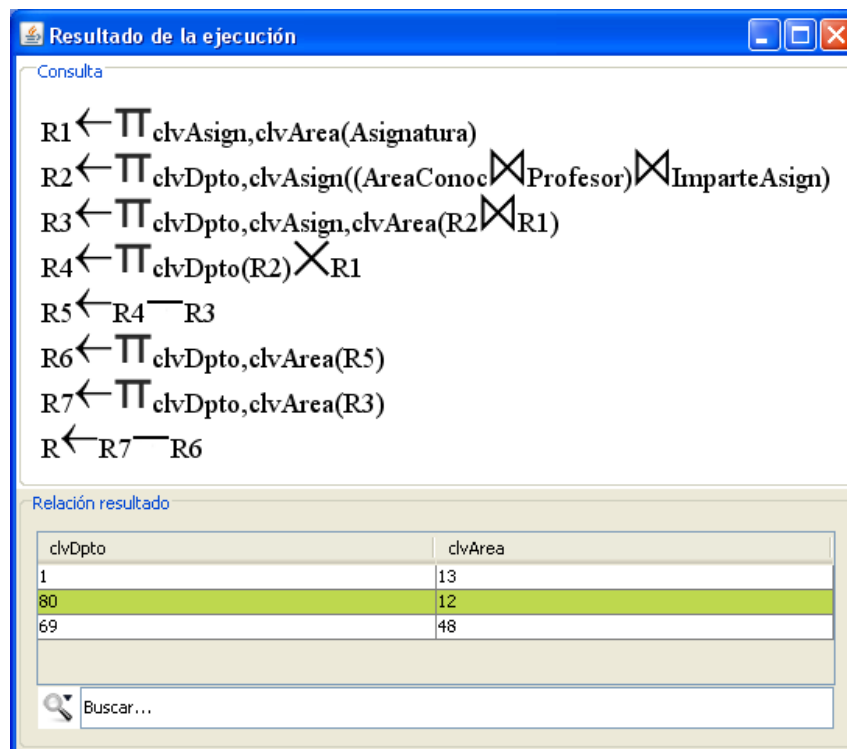


Figura D.25: Ejemplo de creación de consulta paso 9

el join de las relaciones *AreaConoc* y *Profesor* con 116 y 131 tuplas respectivamente,



Figura D.26: Conexión a la base de datos externa

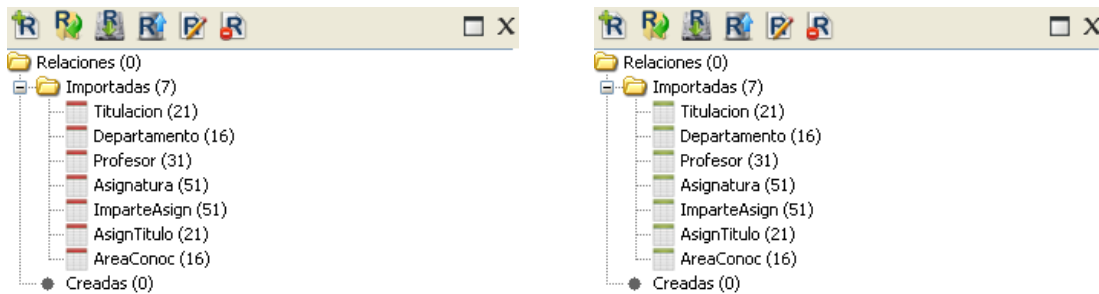
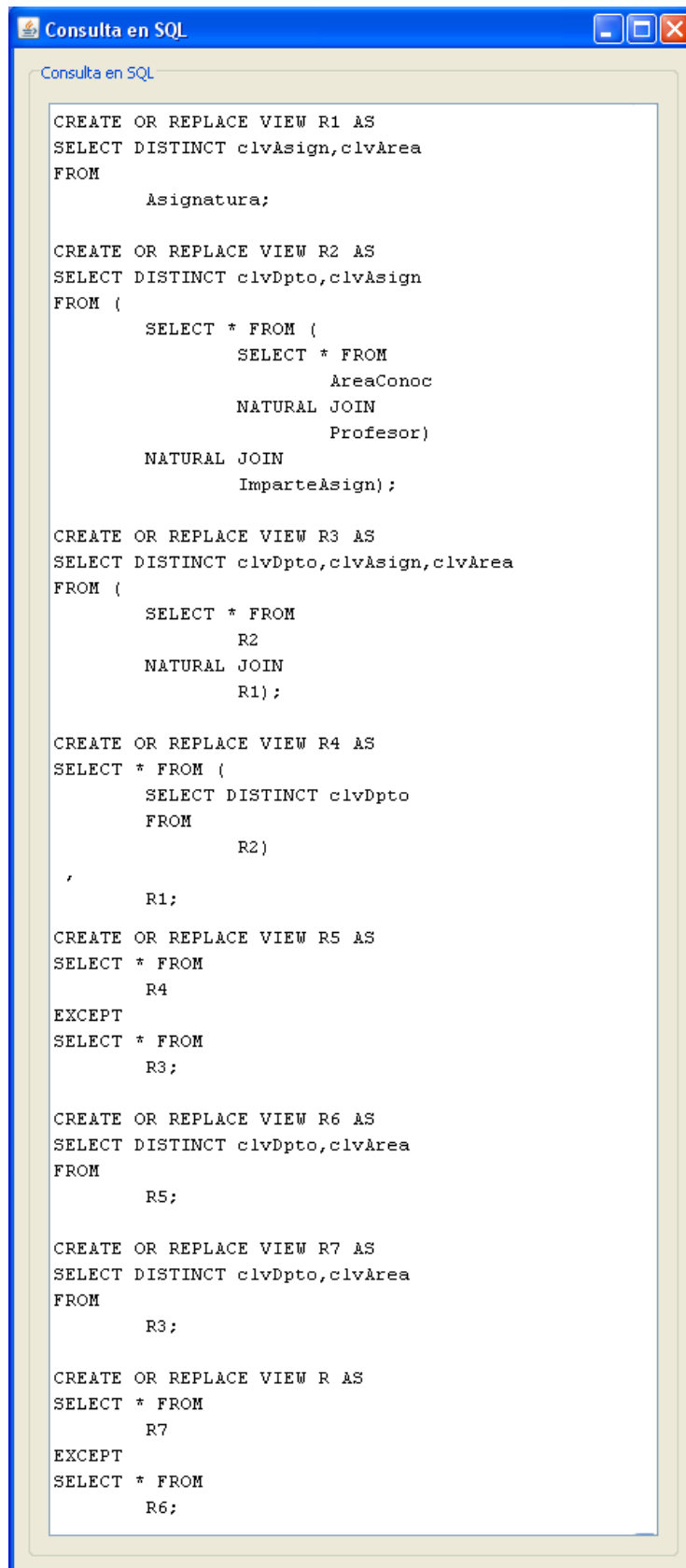


Figura D.27: Antes y después de la exportación

y el resultado de ese join se iba a reunir con la relación *ImparteAsign* (que tiene 51 tuplas). Suponiendo que el primer join seleccionara el máximo número de tuplas que puede, de esta primera operación resultaría una relación con 116×131 tuplas que posteriormente tendría que ser utilizada para otra operación de join con una relación de 51 tuplas. Si en primer lugar realizamos el join que menos tuplas pueda generar, el de *AreaConoc* y *ImparteAsign*, tendremos una relación con menos tuplas para el siguiente join.

En los siguientes pasos se procede a tratar de desplazar las proyecciones lo más abajo posible (ver Figura D.32). Llevando a cabo esta medida, conseguiremos que la cantidad de información por tupla sea menor, ya que se conservarán menos atributos. Como dice la explicación, hay que tener cuidado cuando bajamos proyecciones debajo de joins, puesto que hay que mantener los atributos comunes a las relaciones que intervienen en el join, dado que si no la condición de join cambiaría y no estaríamos representando la misma consulta. En este caso hemos desplazado la proyección de *clvDpto* y *clvAsign*, consiguiendo de esta forma que los joins que se ejecutan más abajo generen menos cantidad de atributos por tuplas y por consiguiente menos cantidad de datos inservibles.

El mismo proceso se realiza para cada una de las asignaciones y finalmente se puede comprobar en la Figura D.33 el árbol de la consulta que teníamos antes de las optimizaciones frente al árbol de la consulta que hemos obtenido como resultado de las mismas.



```
CREATE OR REPLACE VIEW R1 AS
SELECT DISTINCT clvAsign,clvArea
FROM
    Asignatura;

CREATE OR REPLACE VIEW R2 AS
SELECT DISTINCT clvDpto,clvAsign
FROM (
    SELECT * FROM (
        SELECT * FROM
            AreaConoc
        NATURAL JOIN
            Profesor)
    NATURAL JOIN
        ImparteAsign);

CREATE OR REPLACE VIEW R3 AS
SELECT DISTINCT clvDpto,clvAsign,clvArea
FROM (
    SELECT * FROM
        R2
    NATURAL JOIN
        R1);

CREATE OR REPLACE VIEW R4 AS
SELECT * FROM (
    SELECT DISTINCT clvDpto
    FROM
        R2)
,
    R1;

CREATE OR REPLACE VIEW R5 AS
SELECT * FROM
    R4
EXCEPT
SELECT * FROM
    R3;

CREATE OR REPLACE VIEW R6 AS
SELECT DISTINCT clvDpto,clvArea
FROM
    R5;

CREATE OR REPLACE VIEW R7 AS
SELECT DISTINCT clvDpto,clvArea
FROM
    R3;

CREATE OR REPLACE VIEW R AS
SELECT * FROM
    R7
EXCEPT
SELECT * FROM
    R6;
```

Figura D.28: Traducción a SQL de la consulta a ejecutar

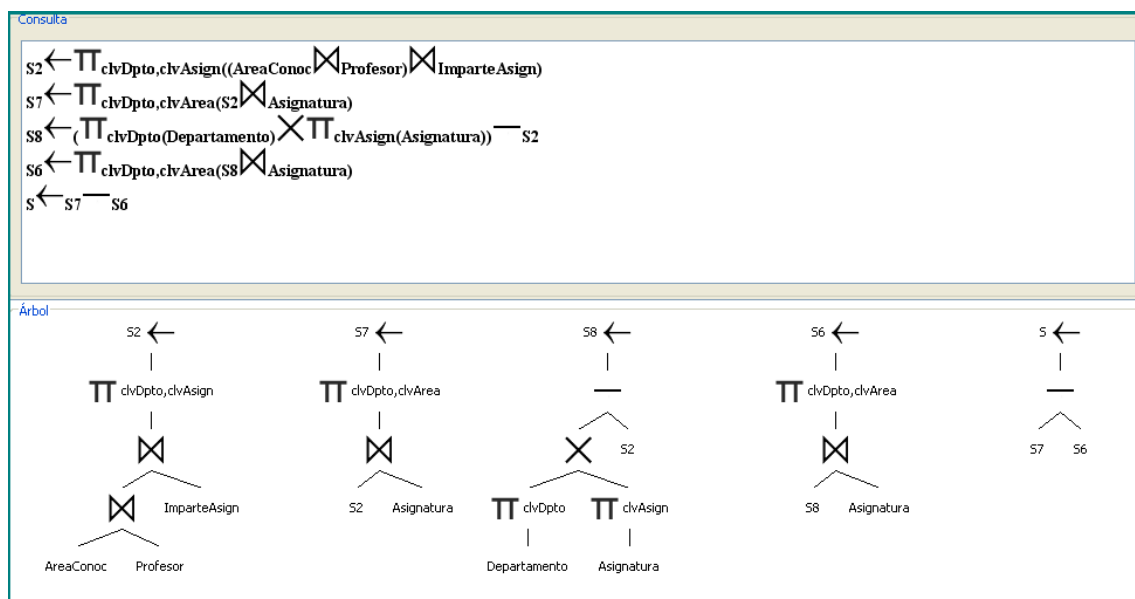


Figura D.29: Consulta introducida para su optimización

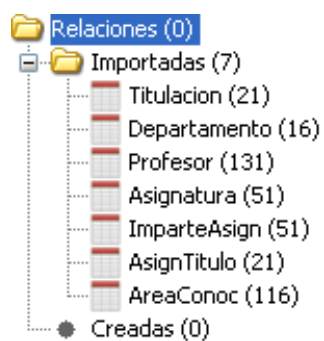
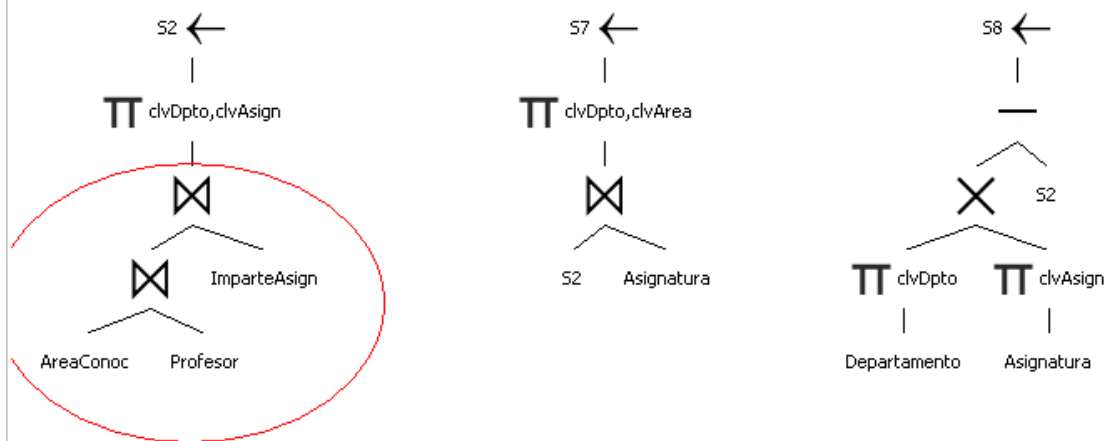


Figura D.30: Relaciones y tuplas creadas para la optimización

Explicación

Utilizando las estadísticas del tamaño de las tablas estimamos el coste del árbol actual y lo modificamos para que el join que menos tuplas crea se ejecute en primer lugar

Árbol anterior



Árbol actual

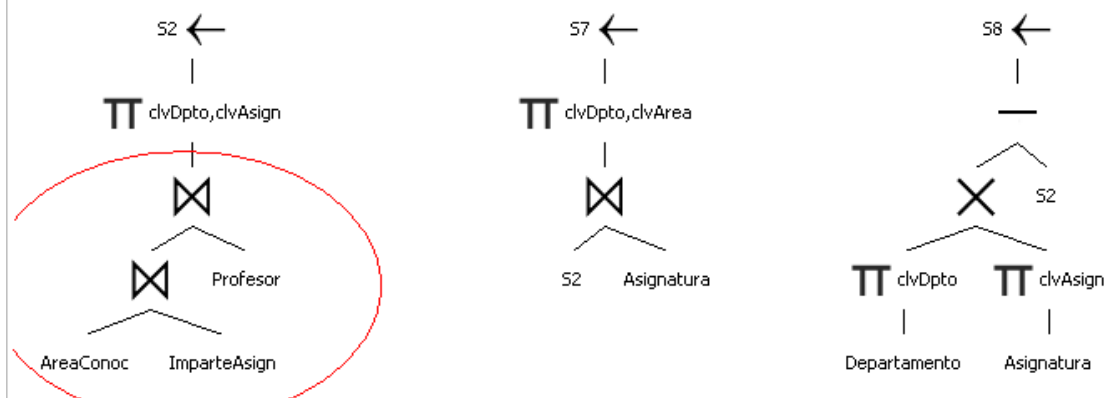


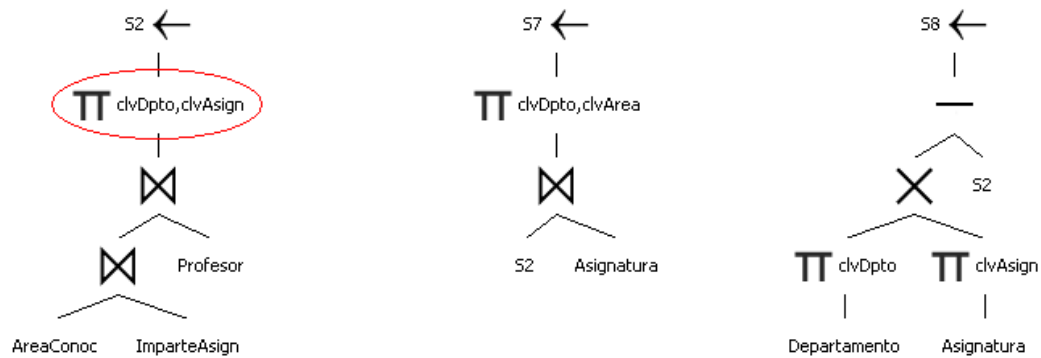
Figura D.31: Optimización de la consulta paso 1

Explicación

Movemos las proyecciones lo más abajo posible en el árbol cuanto antes realicemos las proyecciones menos datos por tupla se utilizaran en cada operación.

Hay que prestar atención cuando movemos las proyecciones puesto que si bajamos una proyección a los hijos de un join debemos lograr que no se pierda en dichas proyecciones los atributos que hacen posible el join o de lo contrario dejaríamos de representar la misma consulta.

Árbol anterior



Árbol actual

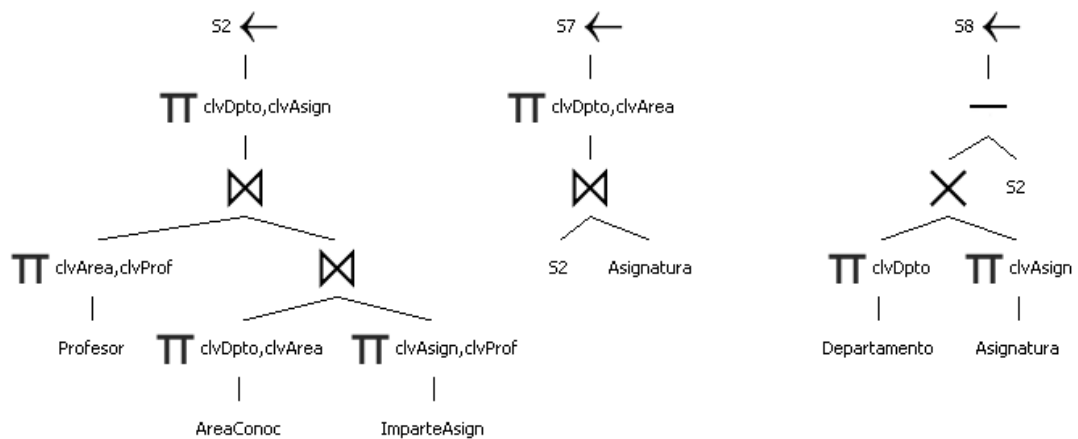


Figura D.32: Optimización de la consulta paso 2

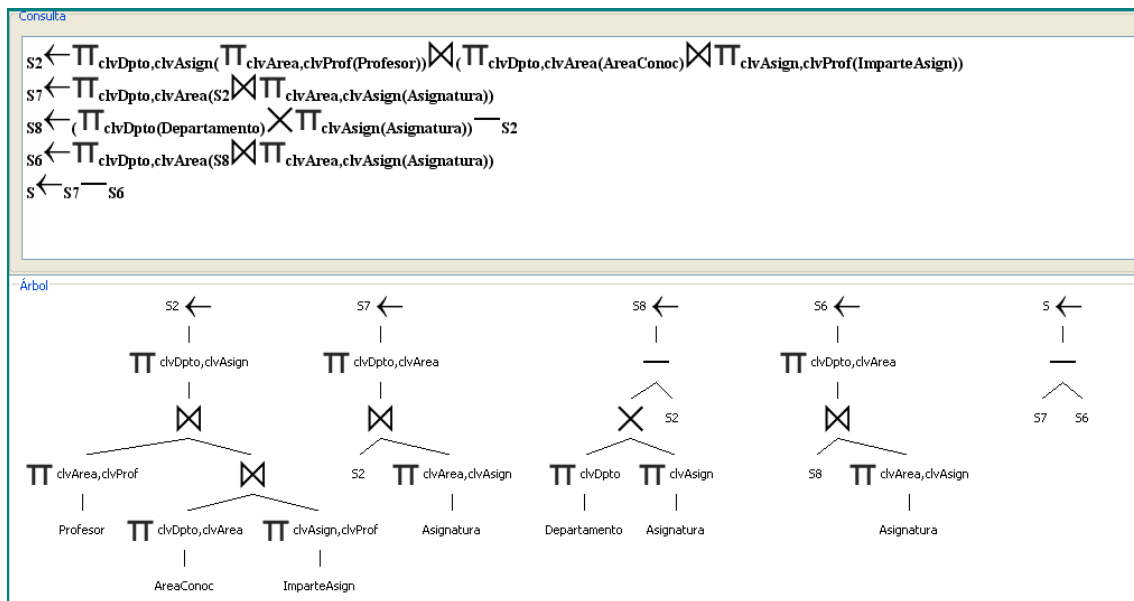
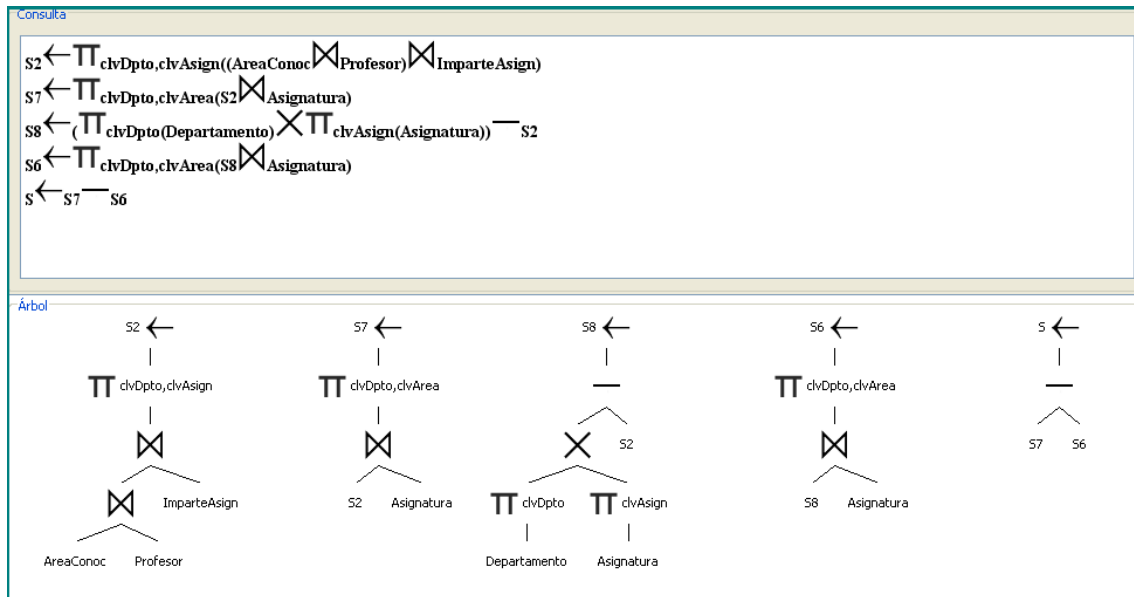


Figura D.33: Consulta y su optimización

Bibliografía

- [1] Santiago Velilla - Apuntes de Ficheros y Bases de Datos.
- [2] Edgar F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377-387, 1970.
- [3] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. McGraw-Hill, 2003.
- [4] Oracle - <http://www.oracle.com/>. Última consulta 04/06/2010. 3.2.4
- [5] Java - <http://java.sun.com>. Última consulta 04/06/2010. 2.4
- [6] Sun Microsystems - <http://es.sun.com/>. Última consulta 04/06/2010. 2.4
- [7] NetBeans - <http://netbeans.org/>. Última consulta 04/06/2010. 2.5
- [8] Gantt Project - <http://www.ganttproject.biz/>. Última consulta 04/06/2010. 2.5
- [9] Dia - http://dia-installer.de/index_en.html. Última consulta 04/06/2010. 2.5
- [10] Adobe Photoshop CS4 - <http://www.adobe.com/products/photoshop/>. Última consulta 04/06/2010. 2.5
- [11] L^AT_EX - <http://www.latex-project.org/>. Última consulta 04/06/2010. 2.5
- [12] MiK_TE_X - <http://miktex.org/>. Última consulta 04/06/2010. 2.5
- [13] L_YX - <http://www.lyx.org/>. Última consulta 04/06/2010. 2.5
- [14] Relational - <http://galileo.dmi.unict.it/wiki/relational/doku.php>. Última consulta 04/06/2010. 3.2.1
- [15] LEAP - <http://leap.sourceforge.net/>. Última consulta 04/06/2010. 3.2.2
- [16] WinRDBI- <http://winrdbi.asu.edu/>. Última consulta 04/06/2010. 3.2.3

- [17] RelationalQuery- <https://relationalquery.dev.java.net/>. Última consulta 04/06/2010. 3.2.4
- [18] ANSI/IEEE Std. 830-1984 - Guía del IEEE para la Especificación de Requerimientos Software. 4.1, C.1
- [19] W3C XML - <http://www.w3.org/XML/>. Última consulta 04/06/2010.
- [20] Query Processing and Optimization - <http://cnx.org/content/m28213/latest/>. Última consulta 04/06/2010.
- [21] Query Optimization- <http://www-db.in.tum.de/teaching/ss10/qo/>. Última consulta 04/06/2010.
- [22] S. Sumathi and S. Esakkirajan. *Fundamentals of Relational Database Management Systems*. Springer, 2007.
- [23] C. J. Date. *SQL and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media, Inc., 2009.
- [24] C. J. Date. *Logic and Databases: The Roots of Relational Theory*. Trafford Publishing, 2007
- [25] Enrique Rivero Cornelio, Luis Martínez Fuentes, Israel Alonso Martínez. *Bases de datos relacionales : fundamentos y diseño lógico*. Madrid: Universidad Pontificia Comillas, 2005.