



Universidad de Buenos Aires
Facultad de Ingeniería
75.52 - Taller de Programación 2
Documentación Técnica

Ayudante: Calvo
2do Cuatrimestre 2012
Grupo Still in Beta

<i>Nombre</i>	<i>Padrón</i>
<i>Fernando Romera Ferrio</i>	<i>88406</i>
<i>Hector Aquino Filho</i>	<i>88064</i>
<i>Guido Ygounet</i>	<i>88246</i>
<i>Gonzalo Almendro</i>	<i>80698</i>

Indice

Introducción	3
Reglas de conversión aplicadas	4
Decisiones de diseño	5
Modelo de datos	6
Persistencia	13
Interfaz	14
Herramientas y librerías utilizadas	15

Documentación Técnica

Introducción

Para realizar el trabajo en cuestión se extendió la funcionalidad del trabajo realizado el cuatrimestre anterior. La nueva funcionalidad comprende la posibilidad de transformar un DER o diagrama de entidad y relación en un diagrama de tablas o lógico.

Reglas de conversión aplicadas

A fin de lograr una correcta conversión entre ambos diagramas se aplicaron las siguientes reglas válidas:

1. Todas las entidades que no forman jerarquías se convierten a una tabla con el mismo nombre.
2. Solo se convierten los atributos de tipo caracterización.
3. Si el atributo es monovalente y compuesto se crea un atributo por cada componente, el nombre del atributo se arma anteponiendo el nombre del atributo padre al del atributo componente, por ejemplo Domicilio - Calle , Domicilio - Número.
4. Si el atributo es polivalente se crea una nueva tabla para almacenar los valores del atributo. Si el atributo es compuesto la clave primaria queda formada por la clave de la entidad más un atributo id , la tabla contiene además un atributo por cada componente. Si el atributo no es compuesto la clave primaria queda formada por la clave de la entidad más el valor del atributo.
5. La conversión no admite atributos identificadores polivalentes.
6. Si el identificador de la entidad está formado sólo por atributos, estos definen la clave primaria de la tabla.
7. Si el identificador está formado sólo por otras entidades, la clave primaria de la tabla se construye con las claves de las tablas asociadas con las entidades identificadoras.
8. Si el identificador es una mezcla de atributos y entidades, la clave primaria se forma con esos atributos más las claves de las tablas asociadas con las entidades identificadoras.
9. Si una relación es binaria y tiene una cardinalidad mínima y máxima igual a uno para una de las entidades, entonces la relación se colapsa en la tabla asociada a esa entidad creando un clave foránea formada por la clave primaria de la tabla asociada a la otra entidad, y agregando los atributos de la relación como atributos de la tabla.
10. Para las demás relaciones se crea una tabla con el nombre de la relación. Los atributos de esta tabla son las claves primarias asociadas a las entidades relacionadas más los atributos de la relación. La clave primaria queda formada por las claves de las tablas asociadas a las entidades con cardinalidad mayor a uno.
11. Las jerarquías se convierten de tres formas distintas: colapsando en padre , colapsando en hijos , o sin colapsar. La decisión del tipo de conversión depende del peso de cada tipo. El peso de colapsar en hijos se calcula sumando los atributos monovalentes de todos los hijos y el doble de la cantidad de relaciones de cada hijo. El peso de colapsar en padres es infinito si el tipo de jerarquía no es total exclusiva, en caso contrario se calcula como la suma de los atributos del padre más cuatro veces la cantidad de relaciones del mismo. Finalmente el peso de no colapsar es fijo e igual a 10. El menor peso decide el tipo de conversión.

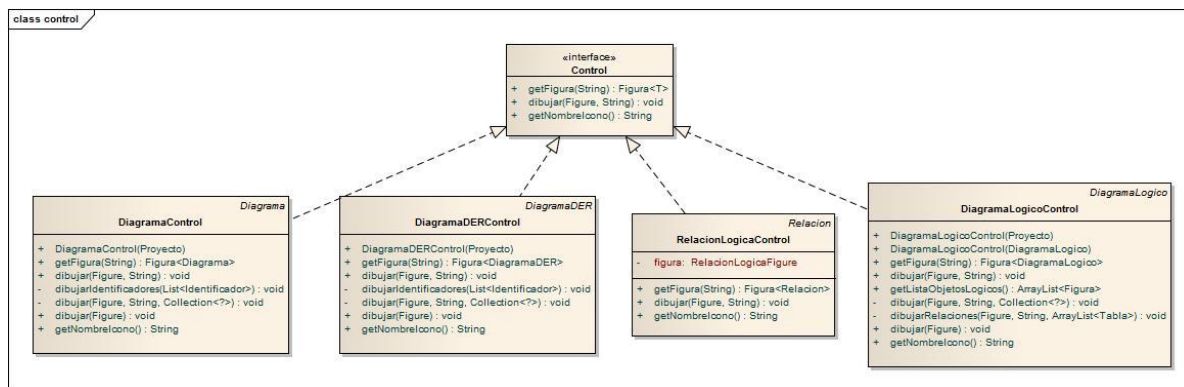
Como aclaración, cabe destacar que utilizando la validación del diagrama existente (ya desarrollada en el trabajo original) y por tratarse de la aplicación de reglas válidas, se asume que el diagrama lógico obtenido es válido; es decir, no resulta necesario validar la correctitud del mismo.

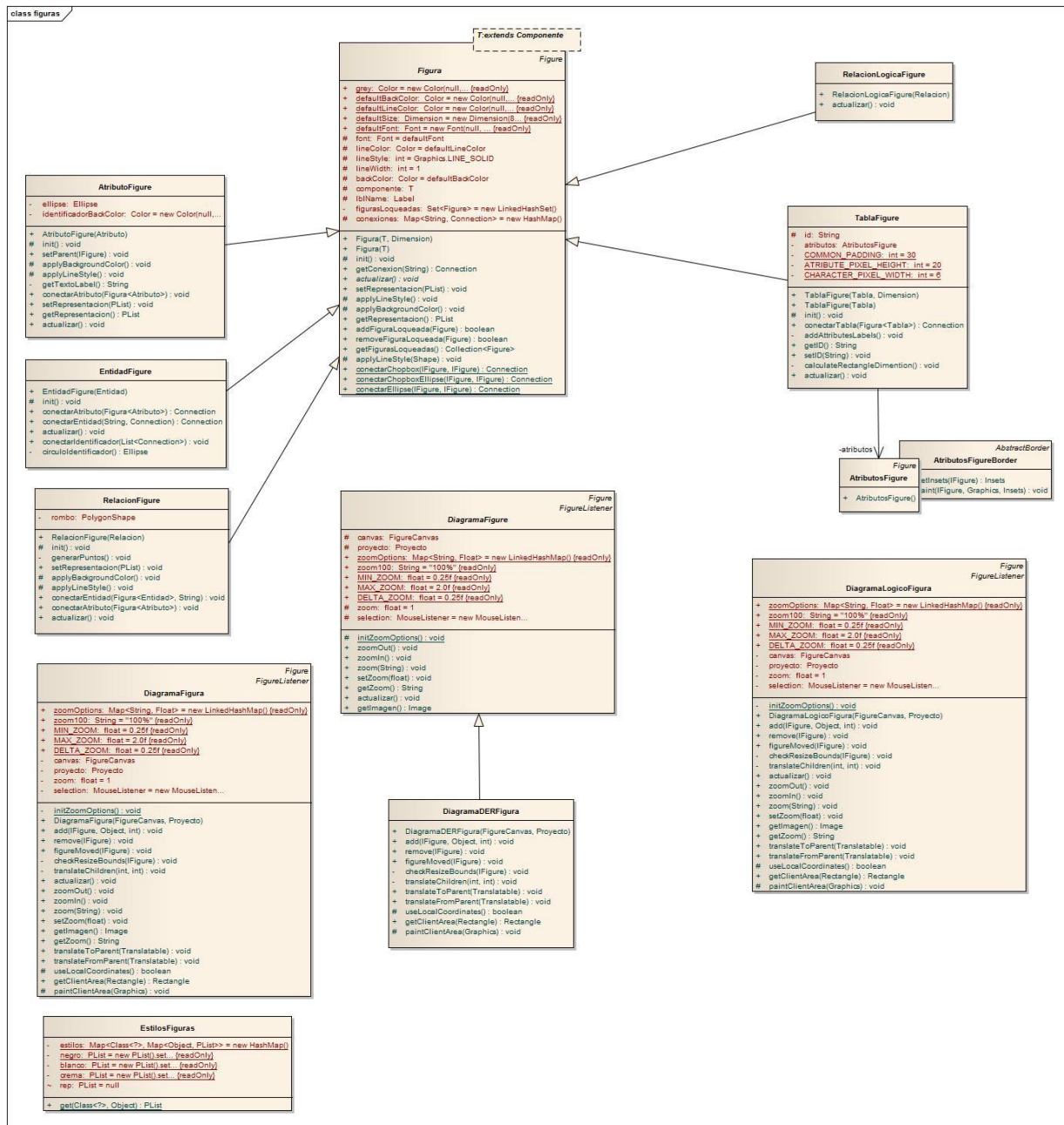
Decisiones de diseño

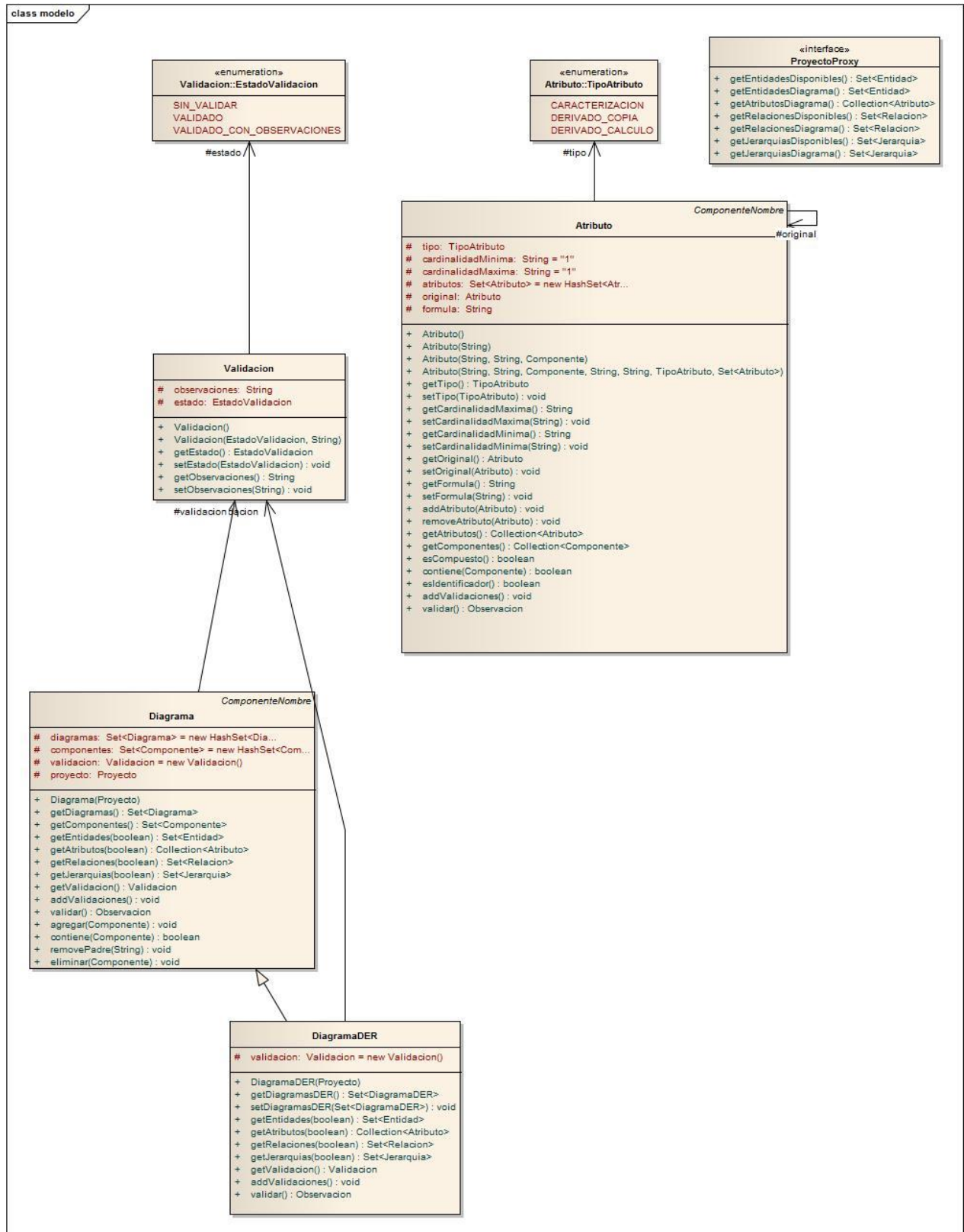
1. Los cambios que deseen realizarse sobre el diagrama siempre deben realizarse en el DER y luego volver a realizar la conversión a diagrama lógico. Se decidió de este modo dado que el DER es el diagrama que mejor representa la realidad del modelo desarrollado. A su vez, el mismo posee la funcionalidad que permite validar la correctitud del mismo. Es por esto, que se supone que los cambios deben realizarse siempre a este nivel, luego debe validarse que el mismo sea correcto y a continuación volver a convertirlo en diagrama lógico (el cual será correcto dado que se utilizan reglas válidas sobre un diagrama validado previamente).

Modelo de datos

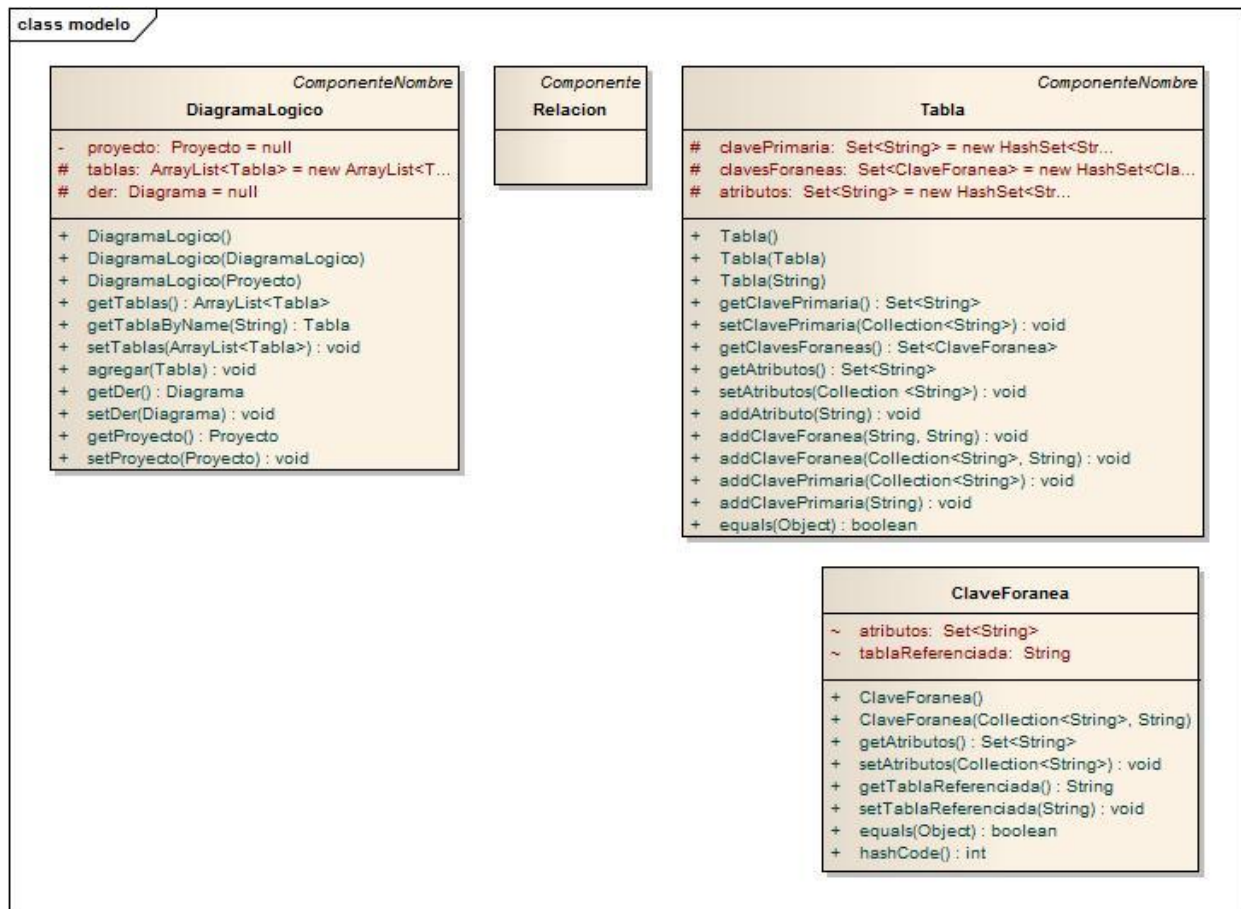
Para el desarrollo de la solución planteada se extendió algunos paquetes ya existentes de la aplicación Mereditor, con el fin de complementar y extender la funcionalidad presentada por las clases de los mismos. Esto permitió agregar a la aplicación existente la funcionalidad que permite convertir a diagrama lógico un diagrama de entidad relación ya existente. A fin de mostrar la solución planteada, se adjuntan a continuación una simplificación de los diagramas de clases de los paquetes de la aplicación Mereditor original que fueron extendidos o modificados.

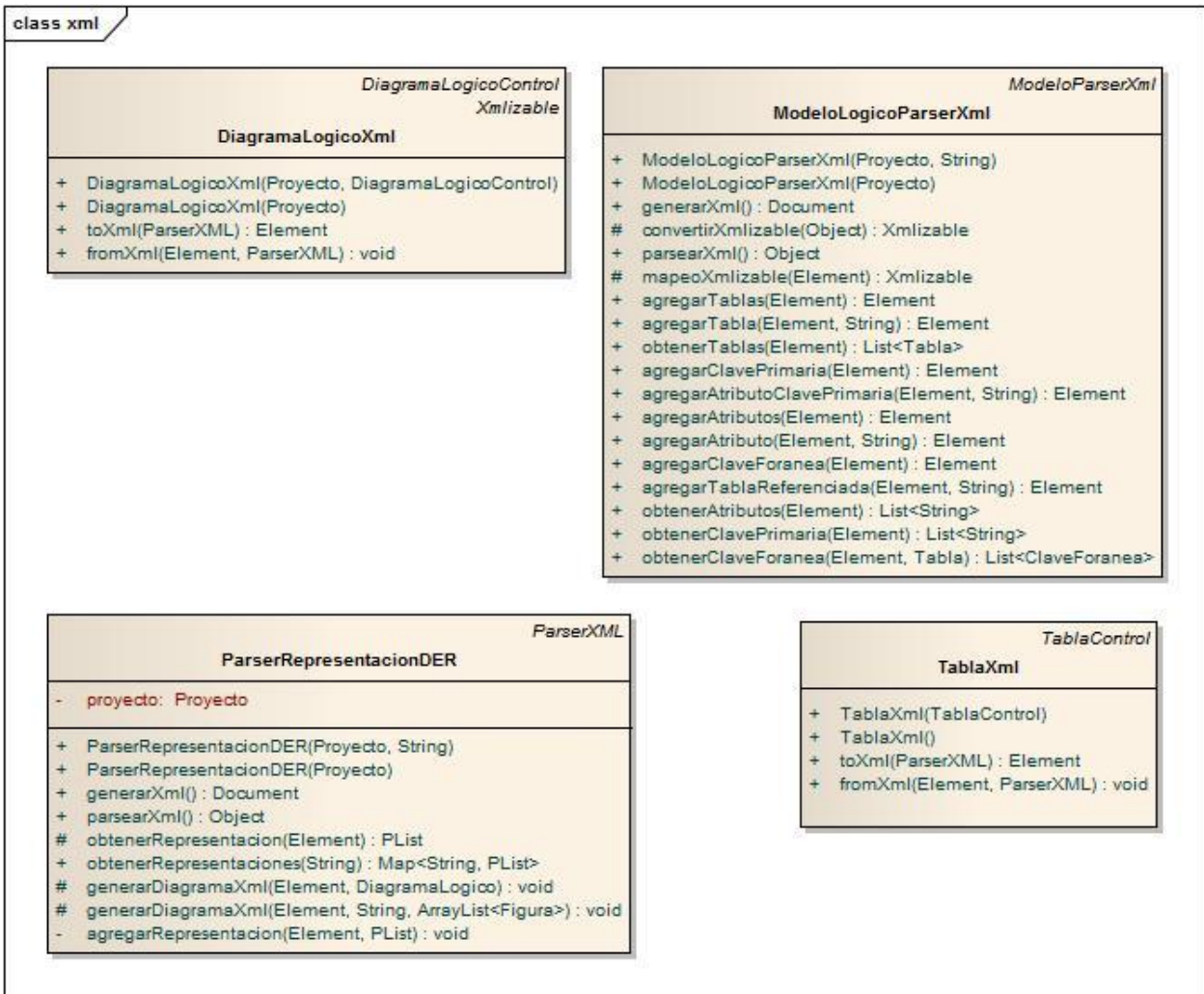













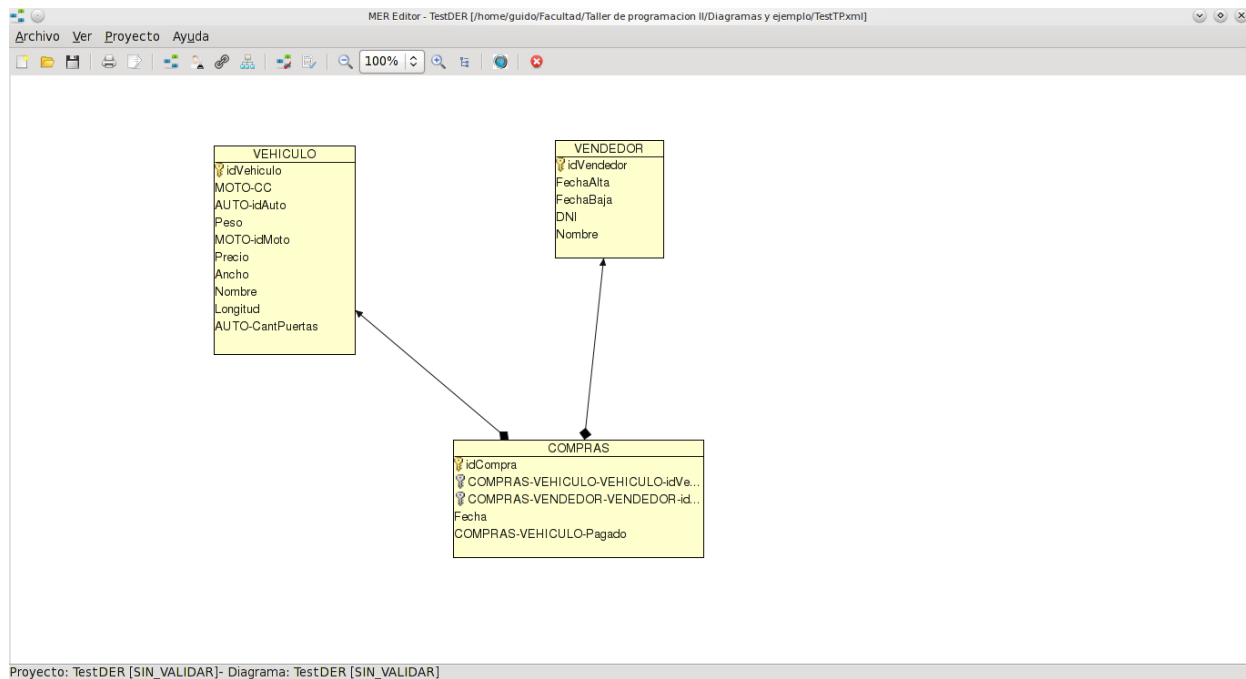
Persistencia

Para la persistencia de los nuevos diagramas lógicos se procedió a la creación de dos nuevos archivos xml. Estos archivos se almacenan en el archivo general del proyecto, de forma tal de mantener la consistencia con la persistencia ya desarrollada en el trabajo que se extendió, utilizando los nombres siguientes:

1. *Nombre-log-rep.xml*: el mismo almacena la forma en la que debe representarse el diagrama lógico en cuestión.
2. *Nombre-logicos-comp.xml*: en el mismo se almacena los componentes que conforman el diagrama lógico en cuestión.

Interfaz

En cuanto a la interfaz, se extendió la interfaz existente agregando a la barra de tareas de la misma un botón  permite realizar la conversión a diagrama lógico del diagrama que se encuentra abierto al momento de presionar dicho botón. A su vez, se agregó en el árbol de la izquierda (Explorador de objetos) el nuevo diagrama lógico.



Herramientas y librerías utilizadas

Por tratarse de una extensión en funcionalidad, se utilizaron las mismas herramientas y lenguaje que en el trabajo original; estas se detallan a continuación:

1. Java - versión 1.7. Se programó indistintamente sobre los sistemas operativos Linux y Windows
2. Eclipse. Como IDE para el desarrollo del trabajo práctico.
3. Se optó por utilizar un repositorio de código GIT hosteado en sourceforge.net; esto permitió interactuar de forma fácil entre los miembros del grupo y que el tutor pudiera ver los avances del mismo en todo momento.
4. Para la interfaz gráfica se continuó utilizando SWT + Draw2d, el framework GEF y JFace.
5. El parseo de los XML se realizó utilizando las funcionalidades que provee DOM.