



Professional Edition v 0.1.x

---

Modelado de la base de datos  
Revisión: 03/10/17

## Sumario

1 Introducción.....	2
2 Primeros pasos.....	2
3 Restricciones del modelo.....	2
4 Estructuración en paquetes.....	2
5 Definición del modelo y validación del UML.....	3

## 1 Introducción

Este documento explica como modelar una base de datos para su uso con la herramienta Samsara Software Professional Edition.

## 2 Primeros pasos

Para poder empezar con un modelo de base de datos, debemos tener como mínimo un modelo básico inicial, como el explicado en el documento de [primeros pasos](#).

También se puede obtener una plantilla de proyecto en [este enlace](#).

## 3 Restricciones del modelo

Antes de modelar una base de datos, es necesario haber leído los manuales de los módulos de generación de código de BBDD de caracter general, y los de la plataforma de BBDD a utilizar: "[Oracle generator manual](#)" y "[Postgres generator manual](#)".

Para poder definir el modelo de base de datos, debemos utilizar los estereotipos del perfil "<http://www.samsarasoftware.net/database/profile>".

En caso que también se utilicen otras herramientas tales como generación de código ORM, o generación de consultas, también es conveniente conocer las posibles restricciones que definan sobre el modelo de base de datos.

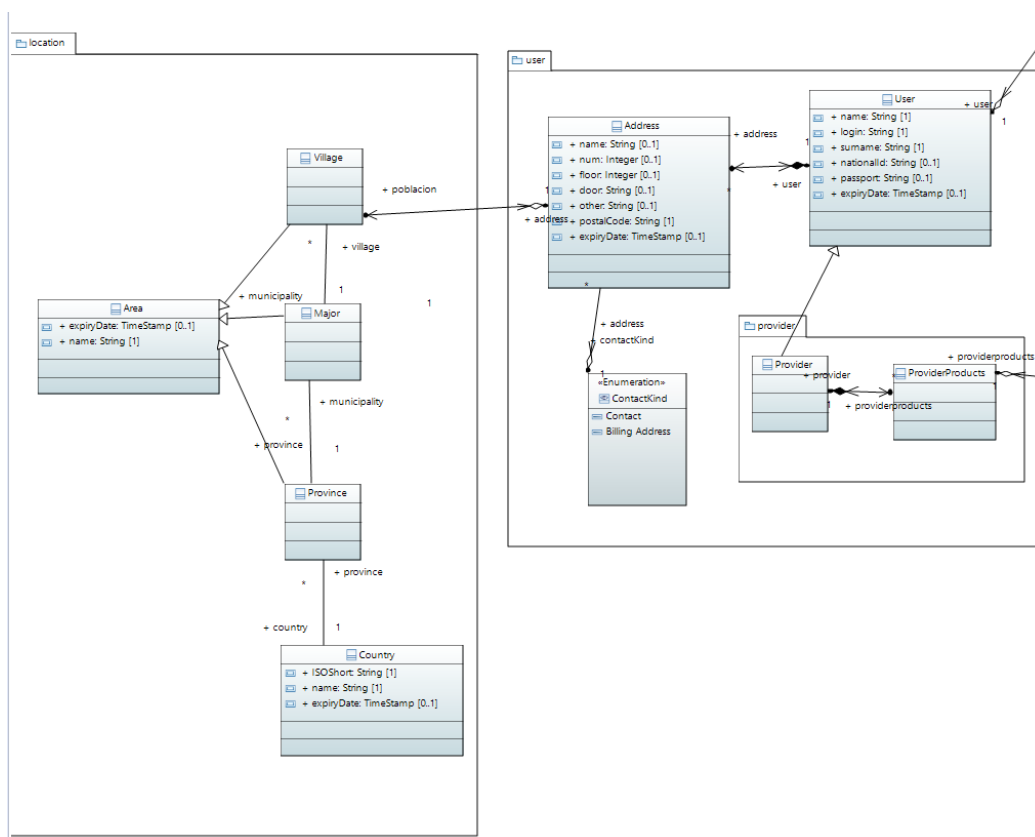
## 4 Estructuración en paquetes

Si vamos a generar código ORM, o queremos estructurar un poco el modelo, lo primero que debemos hacer es definir la estructura de paquetes de los elementos que vamos a generar. Esta estructura puede no corresponderse después en las herramientas utilizadas. Por ejemplo, en Oracle no estructuramos las tablas en paquetes, pero el código JPA, sí que utilizará esta estructura de paquetes.

## 5 Definición del modelo y validación del UML

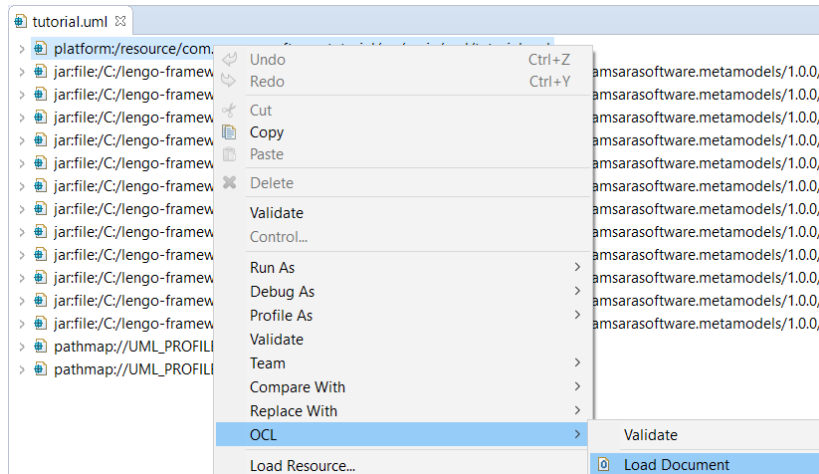
En este punto, empezaremos a definir las clases de cada paquete y sus relaciones entre ellas. Conviene crear un diagrama de clases en el componente anotado con el estereotipo <<Entity>> si nuestro modelo de negocio no es muy complejo, o en cada paquete si el modelo es bastante complejo.

Tomemos como ejemplo el siguiente modelo de base de datos:

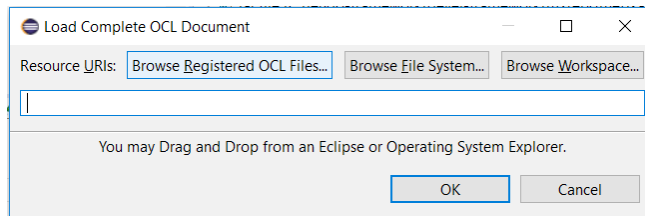


Ahora que hemos creado el modelo, debemos validar que la definición sea correcta. Para ello, abrimos el modelo uml con el "UML Model Editor", y cargamos el documento sql\_constraints.ocl

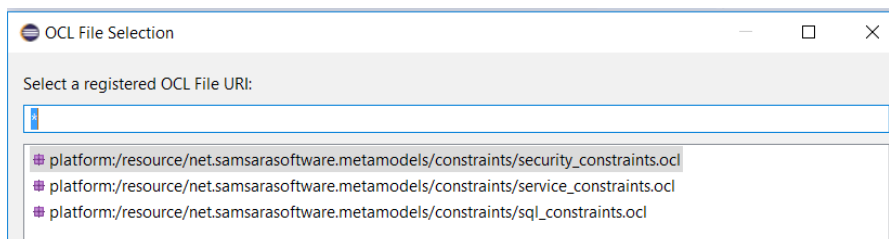
Paso 1: Abrimos el modelo UML con el "UML Model Explorer", y pulsamos botón derecho sobre el modelo. Seleccionamos el menú "OCL" y el submenú "Load Document"



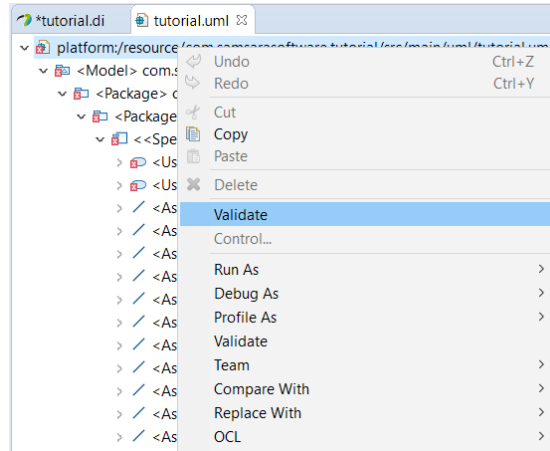
Paso 2: Seleccionamos la opción "Browse Registered OCL Files..."



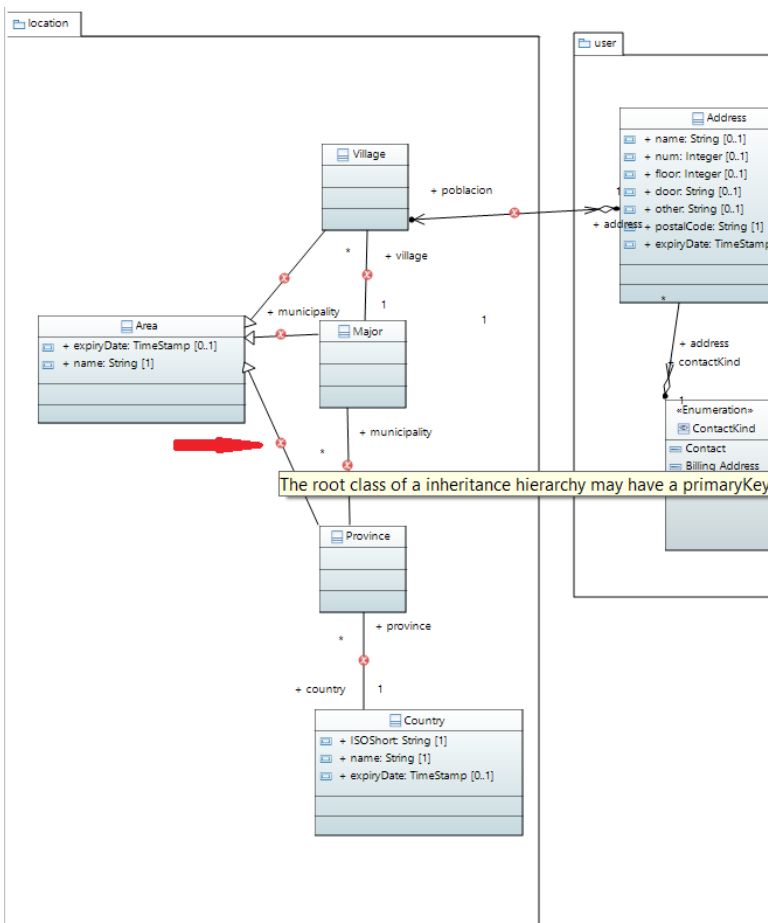
Paso 3: Seleccionamos el documento sql\_constraints.ocl



Paso 4: Validamos el documento UML



Ahora volvamos al editor Papyrus para ver qué errores hemos cometido definiendo el modelo.



Tal y como podemos ver, en las asociaciones han aparecido unas marcas de error. Si colocamos el ratón encima de ellas, nos aparecerá un mensaje descriptivo del error. En el error marcado con la flecha roja, podemos ver que el error indica que la clase Area, que es la clase raíz de la herencia entre Area y Mayor, Village o Major, debe tener un campo ID anotado con el estereotipo <<database::primaryKey>>.

Una vez resueltos los errores, debemos volver a validar el documento hasta haber resuelto todos los errores.

En algunos casos no se nos aparecerá un error, pero sí unas alertas. Por ejemplo, las relaciones 1..N en las que un extremo herede de otro elemento, nos indicará que el extremo de multiplicidad 1 debe ser transformado a multiplicidad 0 para poder ser generado.

