

# Generación automática de Bases de Datos para Sistemas de Información Sanitaria modelados con un Lenguaje Específico de Dominio

**Arian Calabrese, Martín Lunasco, Juan Cesaretti**  
**Tutores Docentes: Leandro Rocca, Leopoldo Nahuel**

{acalabrese, mlunasco} @alu.frlp.utn.edu.ar  
{jcesaretti, leorocca, lnahuel} @frlp.utn.edu.ar

GIDAS: Grupo de I&D Aplicado a Sistemas informáticos y computacionales  
Universidad Tecnológica Nacional - Facultad Regional La Plata  
gidas@frlp.utn.edu.ar

## Resumen

*El propósito del presente trabajo es contribuir con la creación de una herramienta que permita simplificar y acelerar las primeras fases del proceso de desarrollo de sistemas de información sanitaria. Para ello, se utilizó como base un lenguaje específico del dominio (DSL) denominado SIS\_Static. Este DSL fue definido tomando elementos del estándar de interoperabilidad clínica FHIR (Fast Healthcare Interoperability Resources), desarrollado por la organización internacional HL7 (Health Level Seven). Para implementar la herramienta, se empleó el metaeditor MetaEdit+, que permite la construcción de los modelos necesarios y cuenta con un Editor de Generadores integrado para desarrollar transformaciones de modelo a texto. Estas transformaciones se implementan utilizando el lenguaje MERL (Meta Edit Report Language). En el presente informe se describe el proceso llevado a cabo para desarrollar el generador de scripts en lenguaje SQL que permitirán la creación de las tablas correspondientes a los sistemas modelados. Se tomó como ejemplo la Gestión de Accidentes del Trabajo y se generaron las correspondientes tablas y relaciones, en función del modelo construido.*

## Palabras Clave

Modelado Específico de Dominio (DSM), Lenguaje Específico de Dominio (DSL), Fast Healthcare Interoperability Resources (FHIR), Modelo a Texto (M2T), Bases de Datos, Standard Query Language (SQL).

## Introducción

El Modelado Específico de Dominio [1] (DSM - Domain Specific Language) se focaliza en los conceptos propios del dominio considerado. Con ellos, se define al menos un Lenguaje Específico de Dominio [2] (DSL - Domain Specific Language) y, utilizando estos lenguajes, es posible crear modelos consistentes desde las fases iniciales del análisis del negocio.

Luego se trata de obtener un producto de software, reduciendo al mínimo el trabajo de programación manual de los implementadores. Esto se logra a través de transformaciones automáticas de modelo a texto (M2T). La clave de este enfoque es la determinación de un dominio bien restringido, y la consiguiente especificidad de los DSLs. Sin esto, fracasaría cualquier intento de conseguir el grado de automatización deseado.

Nuestro trabajo surgió de la propuesta de hallar un mecanismo para generar sistemas de información sanitaria de un modo más simple y rápido. Para eso, se decidió restringir el dominio a un estándar de interoperabilidad clínica. Se escogió FHIR [3], que es el último estándar abierto de la organización internacional HL7 [4].

La elección de un estándar proporciona dos grandes ventajas en el proceso de

producción de este tipo de sistemas de información. Por un lado, permite el intercambio de datos entre los sistemas informáticos de los diferentes puntos de atención médica. Lo anterior es algo muy necesario, dado que es altamente probable que los pacientes se atiendan en distintos sitios a lo largo de su vida. Por otro lado, posibilita la utilización de una solución DSM, que provee un gran poder de abstracción para abordar la complejidad siempre creciente de dichos sistemas.

Inicialmente, se definió un DSL y se construyó una herramienta que sirve para editar diagramas estáticos de sistemas de información sanitaria, denominado SIS\_Static [5]. También se programaron transformaciones M2T, para obtener automáticamente, a partir de los diagramas estáticos, el código fuente con la definición de las clases y las operaciones básicas, en distintos lenguajes de programación: Java, Python, PHP y Ruby. Luego se desarrolló otro DSL dinámico [6] (SIS\_Dynamic) con su correspondiente editor, para obtener funcionalidades más específicas, que no pueden ser definidas con SIS\_Static. Finalmente, se presentó un DSL llamado SIS\_Interface [7], junto con la herramienta y las transformaciones para generar automáticamente los archivos HTML y CSS de la interfaz gráfica de usuario.

En el actual trabajo, se presenta una nueva transformación para obtener automáticamente un script en lenguaje SQL que permitirá generar una base de datos, a partir del diagrama estático del sistema modelado.

## **Del Modelado Específico a la Base de Datos Física de un Sistema de Información Sanitaria**

### **a) Herramienta de Transformación**

Para realizar transformaciones de modelo a texto se utilizan generadores, que son programas que toman como entrada un

diagrama y producen como salida un archivo de texto.

MetaEdit+ [8] tiene integrada una herramienta que permite crear, editar y administrar generadores, llamada *Editor de Generadores* [9].

Los generadores se implementan con el Lenguaje de Reportes de MetaEdit+ [10] (o por sus siglas en inglés, MERL), que permite la generación automática de código y documentación, a través de campos de texto, símbolos e identificadores. Estos generadores pueden ser ejecutados, depurados y configurados desde el editor mencionado.

### **b) Implementación de la Transformación**

Un generador es una pieza de software ocupada de aplicar transformaciones, tomando como entrada un modelo, y generando como producto un archivo de texto, en este caso, código SQL ejecutable. De esta manera, el generador definido, permitirá al usuario a través de una simple acción, generar el script SQL necesario, desacoplándose de la implementación específica y centrándose en el diseño del modelo y la lógica de negocio.

Para cumplir con el objetivo planteado, se desarrolló un generador en MERL con la capacidad de producir automáticamente un script en SQL, que implementa la base de datos del sistema modelado. Para ello, es necesario recorrer el modelo, accediendo a todos sus atributos, relaciones y roles de cada objeto en el diagrama. El script generado será capaz de crear las tablas correspondientes para un Sistema de Administración de Bases de Datos Relacional (RDBMS por sus siglas en inglés) específico, en este caso, para MySQL [11].

MERL ofrece diferentes estructuras de control y de datos para interactuar con los modelos de entrada.

El algoritmo desarrollado para concretar la transformación de un modelo expresado

en el DSL SIS\_Static a un script en SQL consta de dos recorridos exhaustivos del diagrama de entrada.

De un primer recorrido, se toma el nombre y la estructura de cada tabla que debe crear el script de salida. Por cada elemento visitado, se evalúa si debe generarse una nueva tabla. Y si corresponde hacerlo, de sus propiedades surgen los nombres y los tipos de cada campo.

En una segunda iteración por todo el diagrama, se establecen los vínculos entre las tablas. Esto se logra procesando las relaciones entre los elementos que componen el diagrama, los roles con los que estos participan en dichas relaciones, y las cardinalidades de estas.

En la Figura 1 se puede observar el editor de generadores y parte del código del generador desarrollado.

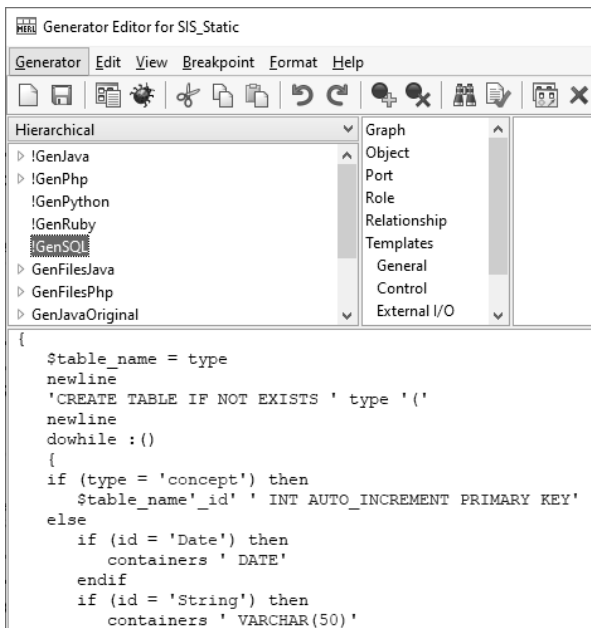


Figura 1 - Pieza del código del generador y el editor.

Por último, para facilitar el uso del generador, se agregó un botón a la barra de herramientas del editor de diagramas estáticos, con un ícono representativo (Figura 2). De esta manera, se proporciona al usuario una forma sencilla de ejecutar la transformación referida para que genere el archivo en formato SQL en un directorio específico.

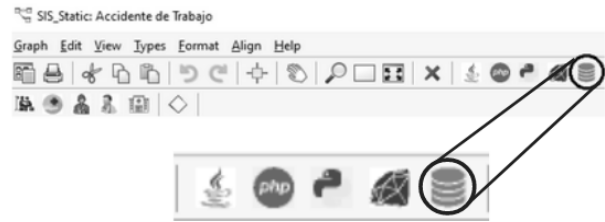


Figura 2 - Barra de herramientas con los generadores implementados.

### c) Ejecución de la Transformación

Con el generador finalizado y el código fuente producido por el mismo (Figura 3), será necesario un entorno de bases de datos que sea compatible con el enfoque elegido para poder ejecutarlo y crear una estructura persistente de datos.

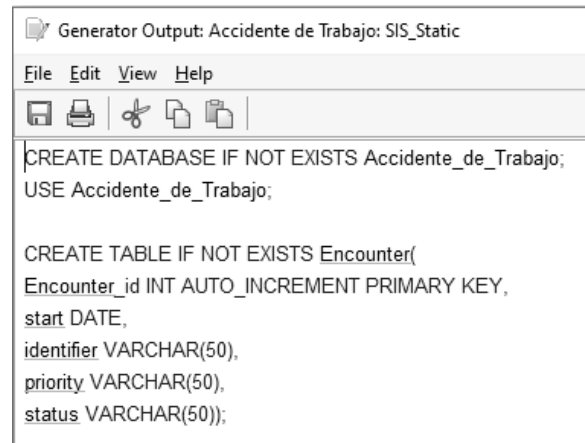


Figura 3 - El producto de la ejecución del generador sobre un diagrama.

Se utilizó una aplicación desarrollada por Oracle Corporation para la generación de la base de datos.

MySQL Workbench [12] es una herramienta visual, que proporciona, entre otras funcionalidades, un entorno para el modelado de datos, un apartado para el desarrollo de código SQL y herramientas de administración integrales para la configuración del servidor, la administración de usuarios, módulos para manejo de migraciones y la generación de copias de seguridad.

Con el archivo en lenguaje SQL y la herramienta seleccionada, se ejecuta el script en este entorno para generar todas las tablas y sus relaciones.

Una vez importado y ejecutado el script, se tendrá la estructura de una base de datos (Figura 4) y el diagrama entidad-relación generado (Figura 5) a partir del modelo desarrollado en MetaEdit+ (Figura 6).

#### d) Estudio de Caso

Se probó el generador aplicándolo a diferentes problemas del dominio de los sistemas de información sanitaria.

De las pruebas realizadas, se seleccionó el caso de la Gestión de Accidentes de Trabajo.

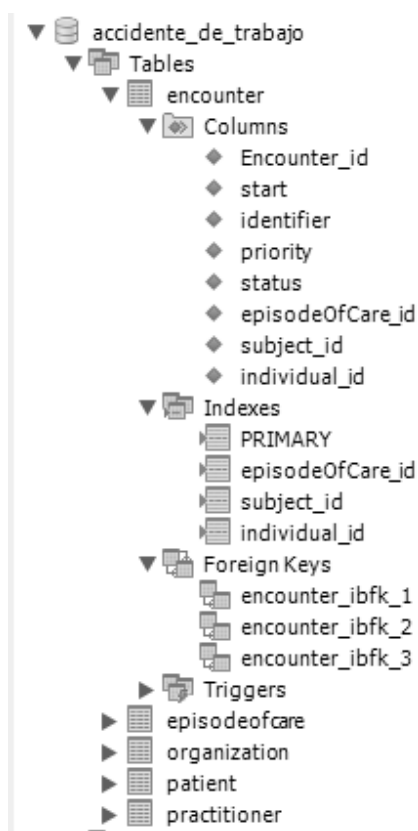


Figura 4 - Esquema importado del script producido por el generador.

Al sufrir un accidente de trabajo, un empleado debe trasladarse al centro médico más cercano, que sea prestador de la ART (Aseguradora de Riesgos del Trabajo) que lo cubre. Allí es atendido por un médico laboral de la ART que completa los informes necesarios para cumplir con el

procedimiento administrativo correspondiente. El trabajador damnificado puede continuar su atención con otro médico responsable, pero quién convalida la denuncia por accidente es el médico laboral de la ART.

Se tomó como entrada para el generador el diagrama de la Figura 6, que fue construido utilizando el DSL SIS\_Static.

Es importante destacar que el mismo generador funcionará para cualquiera de los modelos construidos con dicho DSL.

#### Trabajos Relacionados

En el trabajo *Automatic Code Generation from Database Model Diagram* [13], de Abdullah Al-Muzahid, se presenta el desarrollo de un generador de código automático. Esta herramienta se puede utilizar para generar códigos de software automáticamente en función del diagrama del modelo de base de datos realizado en Microsoft Visio. Los códigos generados que permitirán la operación CRUD estarán escritos en ASP.NET y C# y la base de datos estará en Microsoft SQL Server. Este software también generará automáticamente las tablas y los procedimientos almacenados en función del modelo de datos en la base de datos.

En el trabajo *Tecnología CASE para Modelado Específico de Dominio en Sistemas de Información Sanitaria basado en Estándar de Interoperabilidad Clínica* [14] se presenta la generación automática de código en diferentes lenguajes de programación, a partir de un modelo de un Sistema de Información Sanitaria definido por un DSL. Este trabajo fue utilizado como base para el desarrollo del generador de bases de datos que aquí se describe.

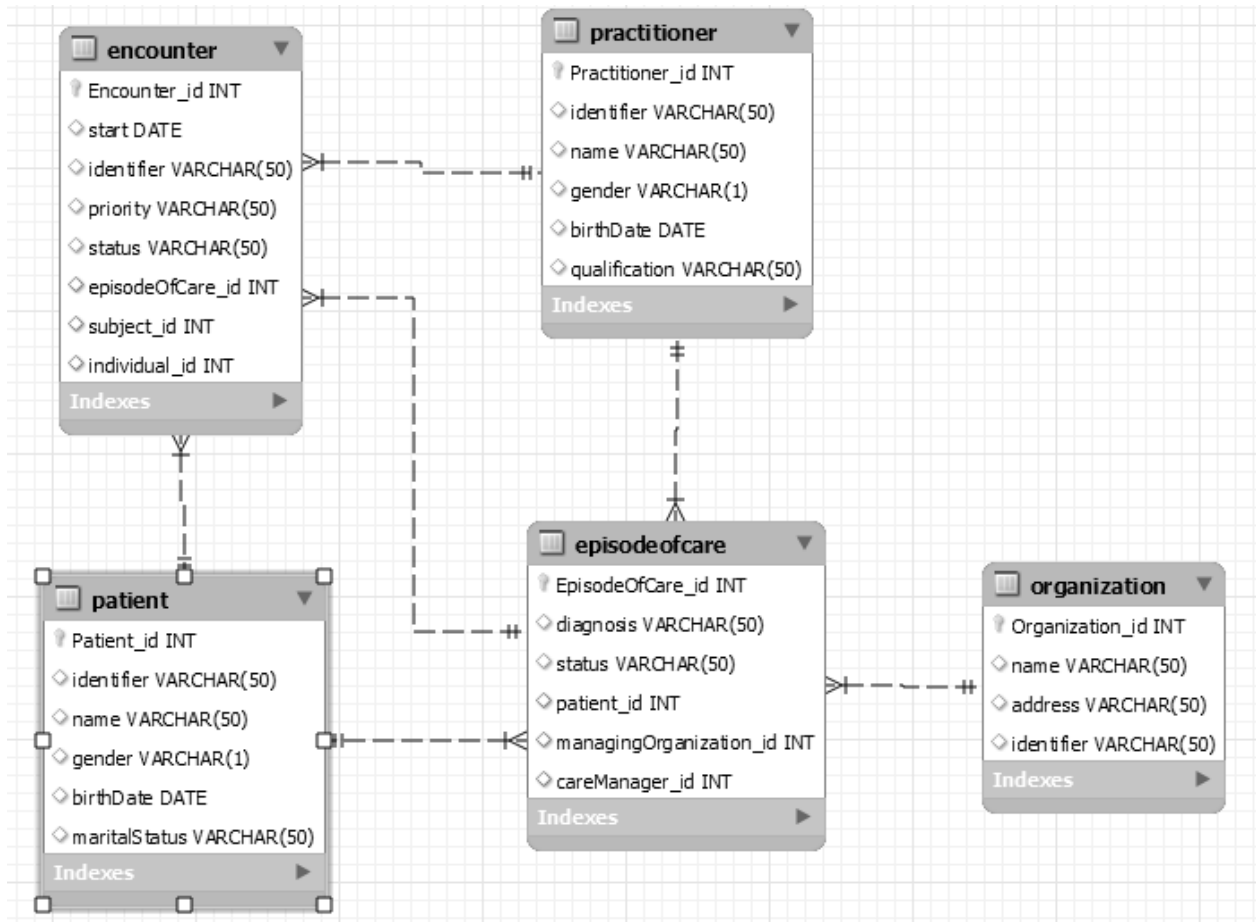


Figura 5 – Diagrama Entidad-Relación generado con MySQL Workbench.

## Conclusión y Trabajos Futuros

Partiendo de un modelo de un Sistema de Información Sanitaria basado en el DSL SIS\_Static se pudo obtener, utilizando el generador desarrollado, el correspondiente script en lenguaje SQL que facilita la generación de las tablas y sus relaciones.

Es importante mencionar que el script generado es compatible con MySQL.

Al ejecutar el script SQL en la herramienta MySQL Workbench se generaron las tablas y las relaciones en

función de los componentes y las relaciones definidos en el modelo construido.

Como líneas de trabajo futuro, se planea avanzar con el desarrollo del generador de scripts SQL para lograr compatibilidad con otros sistemas de gestión de bases de datos, como SQL Server y Postgres.

Con la integración de este nuevo generador a otros módulos desarrollados, se espera completar una herramienta DSM que acelere y simplifique notablemente el proceso de desarrollo de sistemas de información sanitaria, interoperables y de alta calidad.

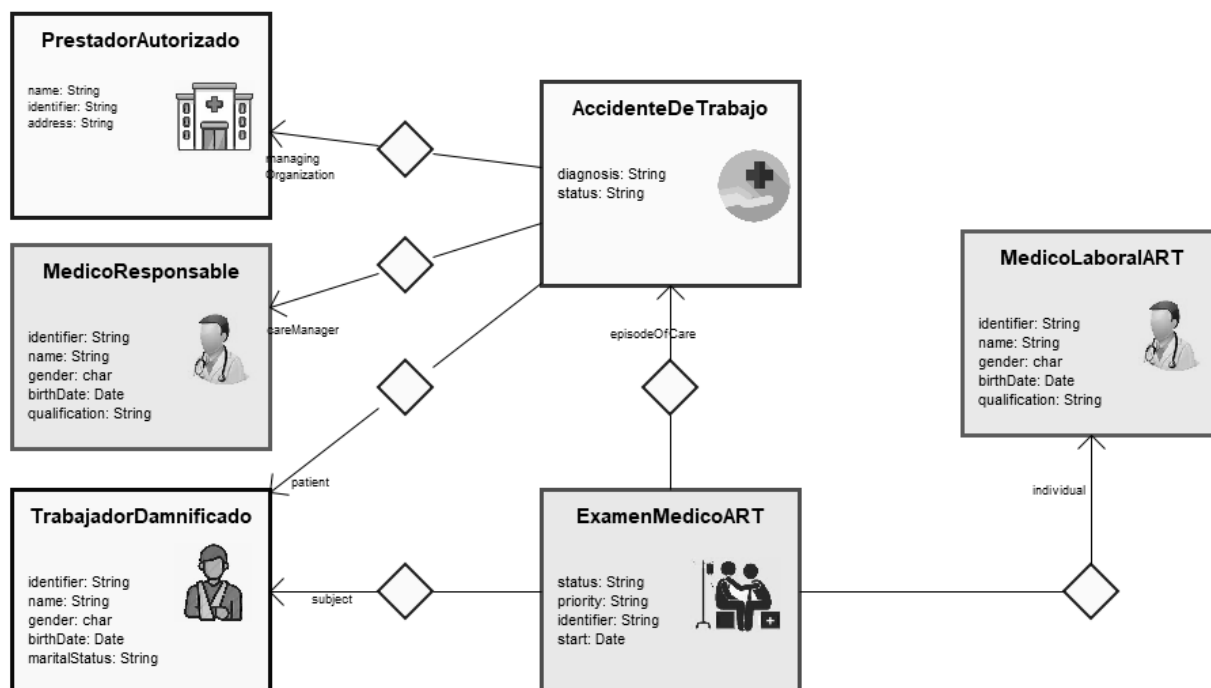


Figura 6 - Diagrama Estático de la Gestión de Accidentes de Trabajo.

## Referencias

- [1] Marjan Mernik, Jan Heering, and Anthony M. Sloane., When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, 2005. doi:10.1145/1118890.1118892
- [2] Kelly, S., Tolvanen, J., (2008). *Domain-Specific Modeling: Enable Full Code Generation*. Hoboken, Estados Unidos: Wiley-IEEE Computer Society
- [3] Resourcelist – FHIR v4.3.0. Recuperado de <https://hl7.org/fhir/resourcelist.html> (accedido el 11/10/2022)
- [4] About Health Level Seven International. Recuperado de <https://www.hl7.org/about/index.cfm?ref=common> (accedido el 11/10/2022)
- [5] Cesaretti J., Paganini L., Rocca L., Caputti M., Zugnoni I. (2019). Herramienta basada en Lenguaje Específico de Dominio para Sistemas elementales de Información Sanitaria. JAIIO 48 – Jornadas Argentinas de Informática. Salta, Argentina.
- [6] Rocca L., Caputti M., Zugnoni I., Paganini L., Cesaretti J., Nahuel L., Giandini R. (2018). Marco de trabajo para el diseño y desarrollo de Herramientas de modelado conceptual basado en DSL utilizando tecnologías GMF. CIITI - Congreso Internacional en Innovación Tecnológica Informática. Buenos Aires, Argentina.
- [7] Cesaretti J., Paganini L., Calabrese A., Lunasco M., Rocca L., Nahuel L. (2021). Construcción de una herramienta de modelado específico para generar sistemas interoperables de información sanitaria. CIACA - Conferencia Ibero Americana Computación Aplicada. Lisboa, Portugal.
- [8] MetaEdit+ Workbench – Build your own modeling tool. Recuperado de <https://www.metacase.com/mwb/> (accedido el 11/10/2022)
- [9] MERL Generator Editor. Recuperado de [https://www.metacase.com/support/55/manuals/mwb/Mw-5\\_3.html](https://www.metacase.com/support/55/manuals/mwb/Mw-5_3.html) (accedido el 11/10/2022)
- [10] MetaEdit+ Reporting Language Generator – MERL Reference. Recuperado de <https://www.metacase.com/support/55/manuals/mwb/Mw.html#Mw-6.html> (accedido el 11/10/2022)
- [11] MySQL Reference Manual. Recuperado de <https://dev.mysql.com/doc/refman/8.0/en/mysqladmin.html> (accedido el 11/10/2022)
- [12] MySQL Workbench - An Unified visual tool for database architects and developers. Recuperado de <https://www.mysql.com/products/workbench/> (accedido el 11/10/2022)
- [13] Abdullah Al-Muzahid - Automatic Code Generation from Database Model Diagram. Recuperado de <https://www.codeproject.com/Articles/1080073/Auto-automatic-Code-Generation-from-Database-Model-Diag> (accedido el 11/10/2022)
- [14] Cesaretti J., Paganini L., Calabrese A., Lunasco M., Rocca L., Nahuel L., Giandini, R. (2021) - Tecnología CASE para Modelado Específico de Dominio en Sistemas de Información Sanitaria basado en Estándar de Interoperabilidad Clínica. CACIC - Congreso Argentino de Ciencias de la Computación. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/130431> (accedido el 11/10/2022)