

# CONSTRUCCIÓN DE UNA HERRAMIENTA DE MODELADO ESPECÍFICO PARA GENERAR SISTEMAS INTEROPERABLES DE INFORMACIÓN SANITARIA

Juan Cesaretti, Lucas Paganini, Arián Calabrese, Martín Lunasco,  
Leandro Rocca y Leopoldo Nahuel  
GIDAS, Grupo de I&D Aplicado a Sistemas informáticos y computacionales,  
UTN - FRLP, La Plata, Argentina  
{jcesaretti, lpaganini, acalabrese, mlunasco,  
leorocca, lnahuel}@frlp.utn.edu.ar

## RESUMEN

Los sistemas de información sanitaria requieren dos capacidades difíciles de llevar a la práctica: adaptabilidad e interoperabilidad. Para lograr la primera, se adoptó el enfoque del Modelado Específico de Dominio (DSM, *Domain-Specific Modeling*). Se desarrolló un Lenguaje Específico de Dominio (DSL, *Domain-Specific Language*) estático con su respectivo editor. Este sirve para realizar diagramas estructurales con un alto nivel de abstracción, y a partir de ellos, generar automáticamente código ejecutable en diferentes lenguajes de programación. Además, los elementos y relaciones de este DSL fueron tomados de FHIR (*Fast Healthcare Interoperability Resources*), un estándar abierto de interoperabilidad clínica, para dotar a los sistemas de esa calidad. En este artículo presentamos un nuevo DSL, con su editor, que permite modelar interfaces gráficas de usuario, vinculando estas vistas con los diagramas estructurales. Así pueden obtenerse los archivos HTML y CSS correspondientes, mediante una transformación de modelo a texto. La expresividad del nuevo DSL fue probada en diferentes casos de estudio. Esto representa un avance significativo en la construcción de una herramienta DSM que posibilite la generación automática de productos de software, desde especificaciones gráficas de alto nivel, en el ámbito de los servicios sanitarios. Se trata de una propuesta innovadora, que simplifica y acelera el proceso de desarrollo de dichos sistemas.

## PALABRAS CLAVE

Estándar de interoperabilidad clínica, fast healthcare interoperability resources (FHIR), lenguaje específico de dominio (DSL), modelado específico de dominio (DSM).

## 1. INTRODUCCIÓN

Una de las dificultades que presentan los sistemas de información sanitaria es su complejidad creciente. Estos necesitan adaptarse a las constantes innovaciones tecnológicas (Organización Panamericana de la Salud, 2021). Además, en los países industrializados, la esperanza de vida va en aumento. Y las personas mayores padecen distintas enfermedades crónicas, debido al sedentarismo, la obesidad y otros problemas causados por su estilo de vida. Se estima que el 17% de los pacientes en EE.UU. tienen más de seis afecciones crónicas. Así, una misma persona consulta a distintos especialistas, y se plantea la necesidad de integrar toda la información registrada por ellos. Y esta información de atención médica debe ser accesible desde distintas organizaciones y puntos geográficos, en virtud de la movilidad de los pacientes (Braunstein, 2018).

Una forma de abordar esta problemática es explotando el poder de abstracción de la MDE (Ingeniería de Software Dirigida por Modelos), que considera a los modelos como los elementos centrales del desarrollo del software (García Molina et al., 2013). Y resulta muy conveniente combinar este enfoque con el uso de un estándar de interoperabilidad sanitaria (Organización Panamericana de la Salud, 2016).

Se han publicado diversos trabajos que tomaron esta iniciativa, con el propósito de extender o especializar un lenguaje de propósito general como UML (*Unified Modeling Language*), o lograr transformaciones de modelos entre el estándar HL7 (*Health Level Seven*) y UML (Pfaff et al., 2019; Olivero et al., 2020).

El propósito de este proyecto, a diferencia de estas experiencias previas, es construir una herramienta para generar sistemas de información sanitaria, de manera automática. Con tal fin, proponemos encarar el desarrollo en el contexto de la MDE, pero desde otra perspectiva: la del DSM (Modelado Específico del Dominio). Este enfoque trabaja con lenguajes propios, restringidos a cada dominio o ámbito de interés. Son los llamados Lenguajes Específicos de Dominio, también conocidos por su acrónimo: DSL (Domain-Specific Language). Esta especialización permite una mayor automatización, que no podría lograrse usando un lenguaje de modelado de propósito general (Kelly y Tolvanen, 2008).

En una primera etapa, definimos un DSL para modelar los aspectos estructurales de los sistemas, al que denominamos *SIS\_Static*. Sentamos sus bases en un estándar abierto de interoperabilidad clínica: FHIR, de la organización internacional HL7 (Benson, 2016). Del mismo, se seleccionó un subconjunto relevante de recursos que fueron tomados como bloques de construcción del lenguaje. Y se implementó un editor gráfico para poder crear, visualizar y editar modelos usando el lenguaje *SIS\_Static*. Para eso se utilizó la herramienta MetaEdit+, que posee un editor generador integrado en el que se programaron transformaciones de modelo a texto, para obtener código ejecutable en distintos lenguajes: Java, Python, Php y Ruby.

En este artículo presentamos un nuevo DSL, complementario del anterior, que sirve para modelar interfaces gráficas de usuario (GUI), y que constituye un avance significativo en el proceso de desarrollo de la solución DSM que estamos construyendo.

## 2. DESARROLLO: DSL SIS\_INTERFACE

Se creó un DSL para modelar la GUI de un sistema, luego de especificar su estructura con el DSL *SIS\_Static*. El nuevo lenguaje fue denominado *SIS\_Interface*. A continuación se describirán sus características.

### 2.1 Lenguaje SIS\_Interface

La sintaxis abstracta del DSL *SIS\_Interface* se definió con un metamodelo (figura 1). Para realizarlo se utilizó GOPPRR (*Graph-Object-Property-Port-Role-Relationship*), que es un metalenguaje específico para describir lenguajes de modelado (Kelly y Tolvanen, 2008, pp. 411-414).

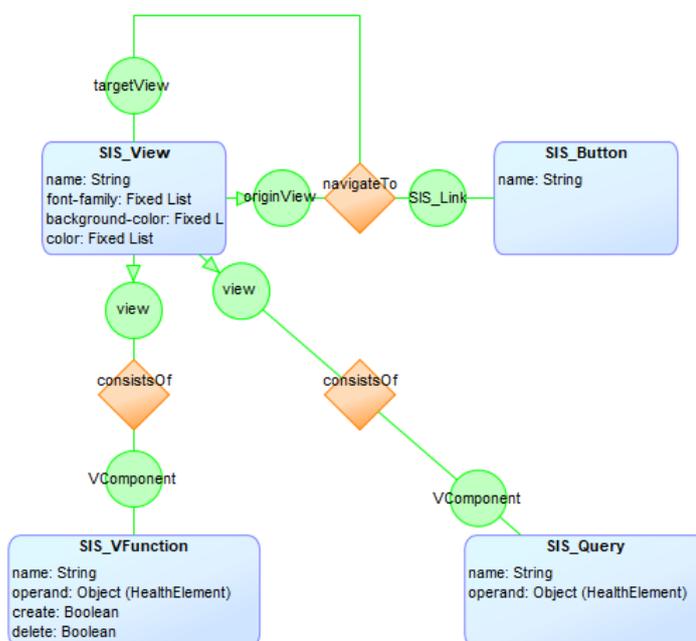


Figura 1. Metamodelo del SIS\_Interface

### 2.1.1 Elementos

*SIS\_View*: Representa una ventana o pantalla que contiene los componentes necesarios para que el usuario visualice o ingrese los datos asociados a una unidad funcional del sistema. Las propiedades de las *SIS\_View* son: título, tipo de fuente, color de frente y de fondo.

*SIS\_Vfunction*: Determina a qué operaciones se da sustento, en la ventana contenedora (creación, modificación o eliminación). Además, establece sobre qué entidad se aplican dichas operaciones. Esto se materializa conectando la propiedad “operand” de la *SIS\_Vfunction* con un objeto existente en el modelo estático correspondiente.

*SIS\_Query*: Este elemento incorpora a la ventana contenedora una consulta sobre los datos.

*SIS\_Button*: Es la representación de un botón que, al ser presionado, abre otra ventana.

### 2.1.2 Relaciones

*consistsOf*: Vincula una ventana con un componente (*SIS\_Vfunction* o *SIS\_Query*).

*navigateTo*: Establece la posibilidad de navegar desde una ventana hacia otra, por la acción de un botón, que también participa en esta relación ternaria.

## 2.2 Editor para el modelado de GUI

Se creó un editor gráfico en MetaEdit+. La Figura 2 muestra el listado de elementos y relaciones válidos, junto a sus símbolos. En la Figura 3 puede verse la ventana del editor. Para agregar un elemento, se arrastra desde la paleta superior hasta el espacio de trabajo. Los vínculos se establecen seleccionando una relación en dicha paleta, y luego los elementos de origen y destino, en el gráfico. Solamente si la relación es válida, se incorpora al gráfico.

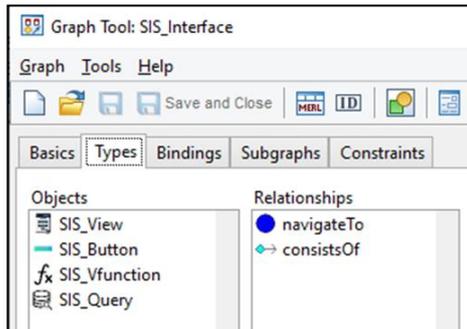


Figura 2. Bloques de construcción del DSL

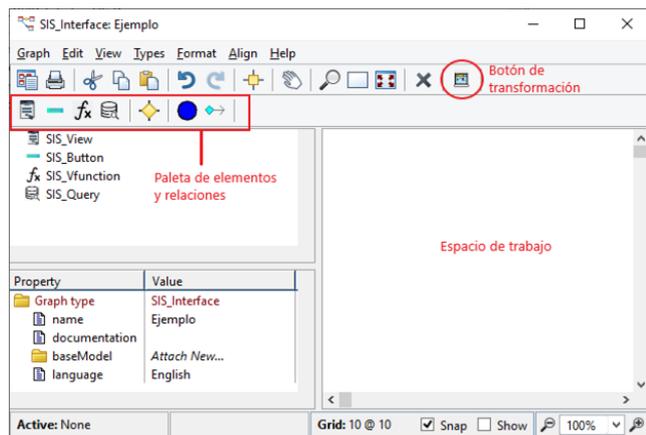


Figura3. Editor SIS\_Interface

## 2.3 Generación automática de GUI

Se implementó una transformación de modelo a texto (M2T), utilizando el editor-generador integrado en MetaEdit+, para obtener los archivos HTML y CSS, a partir del diagrama construido con el editor *SIS\_Interface*. Dicha transformación es un programa escrito en MERL (*MetaEdit+ Reporting Language*), que navega por todos los elementos del gráfico de entrada, obtiene de él todos los datos de forma y estilo necesarios, y produce como salida el código para definir la GUI. De este modo, puede generarse la GUI tan solo presionando el botón de transformación, ubicado en la barra de herramientas superior.

En la Figura 4 se muestra un diagrama de ejemplo, realizado con el editor del DSL *SIS\_Interface*. Y la Figura 5 es la captura de una de las ventanas generadas automáticamente a partir de dicho diagrama.

Los diagramas y archivos de código fuente generados automáticamente, para algunos casos de estudio, se encuentran disponibles en un repositorio: <https://tinyurl.com/38xvfrxd>

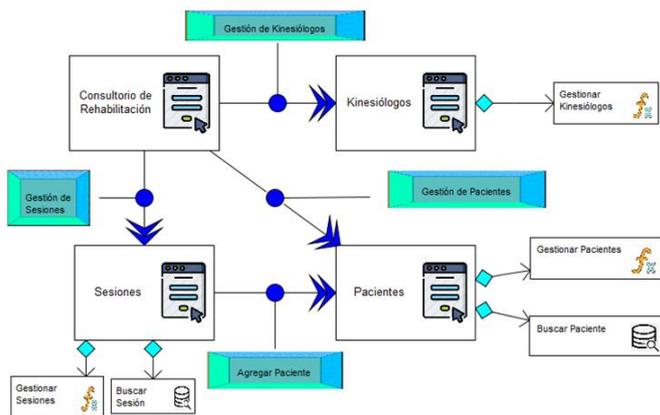


Figura 4. Diagrama de GUI realizado con SIS\_Interface

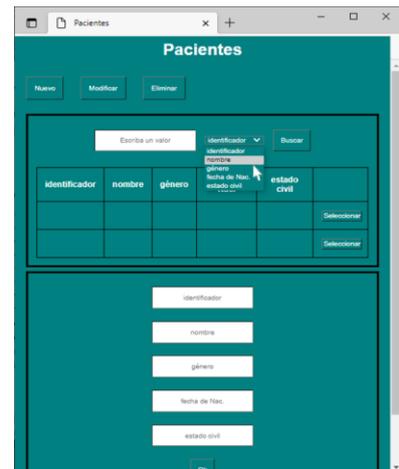


Figura 5. Ventana generada

### 3. CONCLUSIÓN

Se probó la expresividad del nuevo editor, modelando algunos casos concretos del mundo real. Además, con transformaciones M2T, se obtuvieron satisfactoriamente los archivos HTML y CSS de cada GUI modelada.

El DSL SIS\_Interface, junto con el DSL SIS\_Static, ofrece un mecanismo muy útil para aligerar el trabajo de los desarrolladores de software. Cada editor verifica automáticamente los modelos construidos y detecta tempranamente posibles errores, y los evita.

Si bien el subconjunto de recursos tomados de FHIR es limitado, más adelante puede extenderse fácilmente, para tener una mayor cobertura del dominio.

Como trabajo futuro, se planea refinar un DSL dinámico, que ya está en construcción, y servirá para capturar funcionalidades específicas, y los eventos que activan el comportamiento de los sistemas. Asimismo, se implementará una transformación, para obtener el script en SQL que genere la base de datos. Con la integración de todo esto, 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.

### REFERENCIAS

- Benson, T. y Grieve, G., 2016. *Principles of health interoperability: SNOMED CT, HL7 and FHIR*. Springer-Verlag, Londres, Inglaterra.
- Braunstein M., 2018. Health Care in the Age of Interoperability: The Potential and Challenges. *IEEE pulse*, Vol. 9, No. 5, pp. 34–36.
- García Molina, J. et al, 2013. *Desarrollo de software dirigido por modelos: Conceptos, métodos y herramientas*. Ra-Ma, Madrid, España.
- Kelly, S. y Tolvanen, J., 2008. *Domain-Specific Modeling: Enable Full Code Generation*. Wiley-IEEE Computer Society, Hoboken
- Olivero, M. et al., 2020. Facilitating the design of HL7 domain models through a model-driven solution. *BMC Medical Informatics and Decision Making*, Vol 20, p. 96.
- Organización Panamericana de la Salud, 2016, Revisión de estándares de interoperabilidad para la eSalud en Latinoamérica y el Caribe, Washington D.C., EE. UU.
- Organización Panamericana de la Salud, 2021, De la evolución de los sistemas de información para la salud a la transformación digital del sector de la salud. *Informe de la conferencia sobre IS4H*. Washington D.C, EE. UU.
- Pfaff, E. et al., 2019. Fast Healthcare Interoperability Resources (FHIR) as a Meta Model to Integrate Common Data Models: Development of a Tool and Quantitative Validation Study. *JMIR medical informatics*, Vol. 7, pp. 229-241.