

# **Un Lenguaje Específico de Dominio para la Generación Automática de Código en Sistemas de Monitoreo Inalámbrico de Silos Bolsa**

**Claudia Pons<sup>1</sup>, Carlos Luna<sup>2,3</sup>, Daniel Calegari<sup>3</sup> y Nora Szasz<sup>2</sup>**

<sup>1</sup> CONICET - Universidad Abierta Interamericana, Ciudad de Buenos Aires, Argentina  
claudia.pons@vaneduc.edu.ar

<sup>2</sup> Facultad de Ingeniería, Universidad ORT Uruguay, Montevideo, Uruguay  
{luna,szasz}@ort.edu.uy

<sup>3</sup> Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay  
{cluna, dcalegar}@fing.edu.uy

## **RESUMEN**

El silo bolsa es una técnica de almacenamiento de bajo costo que consiste en mantener los granos en bolsas herméticas de polietileno, las cuales son almacenadas en el mismo campo donde se realiza la cosecha. Es probable que varios millones de toneladas de granos sean almacenados en silos bolsa en los próximos años, tanto en países de Latinoamérica como en otras partes del mundo. Las investigaciones confirman que aunque los silos bolsa tengan algunas limitaciones, ellos ofrecen a los productores rurales una solución relativamente barata y segura para el almacenamiento de granos. En este artículo presentamos un sistema computacional inalámbrico de bajo costo que permite monitorear y adaptar las condiciones internas del grano almacenado en silos bolsa, con el fin de mejorar su estado de conservación. Este sistema está dirigido a pequeños y medianos productores rurales y fue desarrollado utilizando técnicas modernas de ingeniería de software.

**Palabras claves:** tecnología agropecuaria, sistema de monitoreo, silo bolsa, modelado de software, comunicación inalámbrica.

## **ABSTRACT**

Silobag is an ad-hoc low-cost storage technique that consists in keeping the grains into hermetic polyethylene bags which are stored in the same field of crop (on-farm). It is likely that several million tons of grain will be stored in harvest bags within the next years, in Latin American countries and also in other parts of the world. Research confirms that although silobags have some limitations, they offer to growers a relatively cheap and reliable grain storage solution. In this paper we present a low-cost wireless software system for monitoring and adapting the internal conditions of the grain stored into silobags in order to improve their preservation state. This system is addressed to small and medium growers and it was developed using modern techniques of software engineering.

**Keywords:** agro technology, monitoring system, silobag, software modelling, wireless communication.

## **1. INTRODUCCIÓN**

El silo bolsa (Bartosik et al., 2002) surgió en la década del 90, permitiendo al productor rural almacenar los granos en su propio campo, reduciendo de ese modo la incertidumbre y los riesgos de no poder contar con un lugar de acopio adecuado antes de la comercialización. El silo bolsa es una bolsa plástica blanca de tres capas con filtro de rayos ultravioleta. El tamaño más utilizado es de 60-75 metros de largo, por 2.74 metros de altura. Cada

bolsa puede almacenar varias toneladas de grano. Este sistema genera un ambiente hermético en el interior de la bolsa en el que aumenta la concentración de dióxido de carbono (CO<sub>2</sub>) y disminuye la concentración de oxígeno (O<sub>2</sub>). Esta atmósfera propicia el control de la actividad de insectos y reduce el desarrollo de hongos. Además, constituye una barrera física, por lo que si el grano se embolsa libre de infestación, se puede mantener en estas condiciones durante todo el período de almacenamiento sin necesidad de aplicar pesticidas.

La temperatura regula la velocidad de cualquier proceso biológico; si es alta, el proceso en cuestión se acelera, y si es baja, se retarda. Su efecto cobra importancia cuanto mayor es la humedad del grano almacenado. La medición de temperatura y humedad resulta muy importante, porque permite determinar si existe actividad biológica dentro de la bolsa y tomar las medidas correspondientes antes de que se deteriore el grano.

Los autores de este trabajo participan de un proyecto de investigación en curso (Proyecto LACCIR, 2010), el cual tiene como finalidad diseñar tecnologías para el seguimiento y la adaptación automática de las condiciones internas de los granos almacenados en silos bolsa, con el fin de optimizar su estado de conservación. Dado que los resultados de este proyecto están especialmente dirigidos a apoyar a pequeños y medianos agricultores, un requisito obligatorio de nuestro trabajo es que la solución desarrollada sea de bajo costo. En este contexto, se ha estudiado el efecto de la evolución de los niveles de humedad sobre la calidad de los granos. A partir de este estudio se ha definido un modelo matemático para la predicción del comportamiento de la humedad y la temperatura, y el análisis de sus efectos en las condiciones internas (Pons et al., 2009). Asimismo, se han diseñado sensores inalámbricos de bajo costo para medir los niveles de humedad, temperatura, acidez y dióxido de carbono (Fueyo y Epeloa, 2010), a los efectos de monitorear dichas condiciones internas. La siguiente etapa del proyecto consiste en incluir un mecanismo automático de adaptación de las condiciones internas de los silos bolsa.

En este artículo describimos algunos de los resultados del proyecto, en particular presentamos una infraestructura para la construcción de sistemas de monitoreo, que permiten registrar mediciones de diferentes sensores, monitorear la evolución de estas mediciones y que permitirán luego activar, una vez diseñados, dispositivos específicos para adaptar las condiciones internas de los silos bolsa. Toda la información se obtiene a través de sistemas de software de control, consistentes de microcontroladores con escasos recursos energéticos y de hardware. La interacción del usuario con este sistema es a través de una interface muy simple y amigable.

Para lograr estos objetivos se construyó una solución integral siguiendo los lineamientos de los paradigmas denominados Ingeniería Dirigida por Modelos (Stahl and Völter, 2006) y Modelado Específico de Dominio (Kelly and Tolvanen, 2008). Estos paradigmas están basados esencialmente en la definición de modelos específicos para la representación de los diferentes aspectos de un sistema a construir y en la generación automática de los sistemas de monitoreo a partir de dichos modelos.

Este trabajo representa una alternativa a la tecnología que presentamos previamente en (Calegari et al., 2010) y constituye a la vez un avance sobre el trabajo referido, al contemplar aquí el sistema inalámbrico que complementa al software.

El resto del artículo está estructurado de la siguiente manera. En la Sección 2 se describen las características principales de los paradigmas que contextualizan la solución al problema. En la sección 3 se presentan el proceso de construcción y uso de la infraestructura específica de dominio. En la sección 4 se describe el sistema inalámbrico que complementa al software. Finalmente, en la Sección 5 se incluyen trabajos relacionados y en la Sección 6 se exhiben las conclusiones y los lineamientos para el trabajo futuro.

## **2. MODELADO ESPECÍFICO DE DOMINIO**

La Ingeniería Dirigida por modelos (MDE, por sus siglas en inglés) así como el Modelado Específico de Dominio (DSM, por sus siglas en inglés) son metodologías de desarrollo de software en las cuales se busca incrementar el grado de abstracción en la representación de las aplicaciones, más allá de los lenguajes de programación convencionales, utilizando conceptos y reglas tomados directamente del dominio del problema de software a resolver.

La metodología propone comenzar el desarrollo con una definición formal de los conceptos y las reglas que caracterizarán a las representaciones de alto nivel (modelos), las cuales serán el artefacto principal de

especificación del software. La obtención de representaciones de un menor nivel de abstracción, como por ejemplo código fuente, es lograda a través de un proceso automático de derivación, el cual toma como entrada los modelos formales confeccionados de acuerdo a los conceptos y reglas definidos. Esta derivación es posible gracias a la cota que se pone en la cantidad de aspectos considerados del problema de software a tratar, la cual conforma el “dominio específico” de los modelos y promueve un incremento en la semántica de los conceptos y reglas utilizadas en las descripciones. Tal incremento permite la generación de código completa, a partir de los modelos.

En contraste con otros procesos de desarrollo de software, la metodología requiere, por parte de unos pocos desarrolladores con experiencia en el dominio particular, la creación de tres elementos principales que serán la herramienta fundamental del equipo que llevará a cabo la construcción del software:

- un lenguaje de dominio específico formal a través del cual se definen las soluciones,
- un generador que transforme los modelos a código ejecutable, y
- un framework de dominio que sirva como base al código generado en el punto anterior, para que la transformación sea más sencilla y reusable.

### 3. CONSTRUCCIÓN DE LA SOLUCIÓN ESPECÍFICA DE DOMINIO

En esta sección explicaremos de qué forma hemos implementado los 3 elementos mencionados arriba utilizando la herramienta Microsoft DSL Tools (Cook et al., 2007). Por razones de espacio las descripciones son parciales. Todos los detalles pueden consultarse en (Martínez y Lavie, 2010).

#### 3.1 EL LENGUAJE GRÁFICO DE DOMINIO ESPECÍFICO

La definición del DSL consta de 2 partes:

**Modelo de dominio:** es un modelo de los conceptos descriptos por el lenguaje. El modelo de dominio define la sintaxis abstracta del DSL.

**Notación:** los conceptos del dominio son mapeados a figuras con el fin de poder representarlos en diagramas. Define la sintaxis concreta del DSL.

La información del dominio se clasificó en dos categorías, información estática e información dinámica.

La Información estática define tipos de objetos. Por ejemplo, los tipos de sensores (marca, modelo, variable que mide, rangos de medición, descripción general), tipos de variables a monitorear (temperatura, humedad, CO<sub>2</sub>), tipos de contenidos (al modelar el tipo de contenido se debe poder especificar las características del mismo: tiempo de almacenamiento máximo recomendado, valores máximos y mínimos para las variables de mediciones, tiempo máximo de almacenamiento recomendado).

La información dinámica representa a los objetos del sistema. Por ejemplo:

- *SiloBolsa*: este concepto modela el silo bolsa y sus características. Entre la información más importante se identificó: fabricante, modelo, material, espesor, tamaño, largo y ancho, descripción general, posición geográfica, red de sensores, contenido que almacena (tipo de contenido, fecha de almacenamiento, nivel inicial de temperatura, humedad, CO<sub>2</sub>, O<sub>2</sub>, porcentajes de granos dañados en el momento que se almacenó, porcentaje de materia extraña cuando se almacenó), etc.
- *Sensor*: modela la red de sensores que tiene cada silo bolsa. Para esto se debe poder representar cada sensor del silo bolsa y la información asociada a él (posición del sensor en el silo bolsa, tipo de sensor, identificación del sensor, frecuencia de monitoreo, acciones a ejecutar en caso de que una medición supere un umbral).
- *Registro de Medición*: Se debe poder modelar un registro de medición, el cual representa el resultado del sensor luego de tomar una medición. La información para este registro es: sensor que generó el registro, variable que mide, fecha y hora de la medición, valor de la medición.

Para implementar el DSL es necesario definir su sintaxis abstracta a través de un meta modelo, es decir crear un diagrama del dominio usando las DSL Tools. Las siguientes figuras muestran la implementación de algunos de estos conceptos. En la figura 1 los silos bolsa y su información se representaron con la clase SiloBolsa. Una SiloBolsa tiene una lista de sensores, la cual representa la red de sensores asociada a la misma. Un Sensor puede estar asociado a un solo silo bolsa y solo existe si éste existe, por lo tanto la relación “silobolsa tiene sensores” (SiloBolsaHasSensors) es de tipo embebida. La figura 2 ilustra la relación entre el silo bolsa y su contenido.

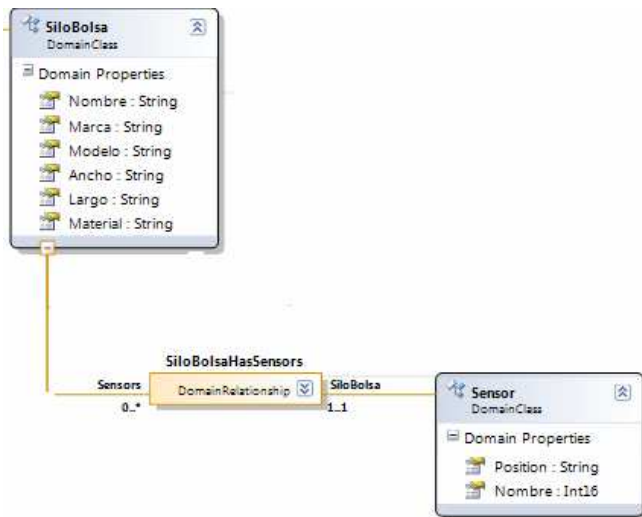


Figura 1: Silo bolsa y Sensor

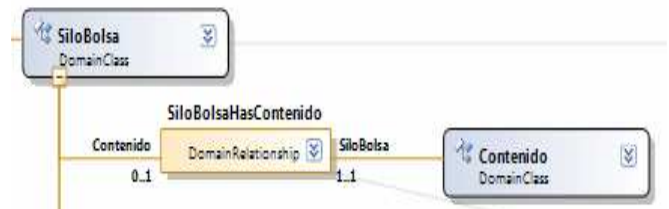


Figura 2: Silo bolsa y contenido

Como siguiente paso, utilizando el MS DSL Designer hemos asociado una sintaxis concreta a nuestro DSL. Dicha sintaxis utiliza gráficos e intenta ser familiar y amigable para los expertos en el dominio de los silos bolsa. La figura 3 muestra el aspecto gráfico de todos los elementos del DSL en el ToolBox del sistema. La definición completa puede leerse en (Martínez y Lavie, 2010). Luego, la figura 3 muestra un pequeño ejemplo de esta notación. Este modelo representa un silo bolsa con contenido de tipo Soja. Este contenido es monitoreado por un sensor de marca Siemens el cual permite tomar los valores de CO2.

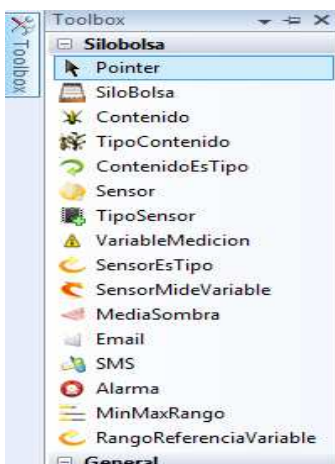


Figura 3: Toolbox

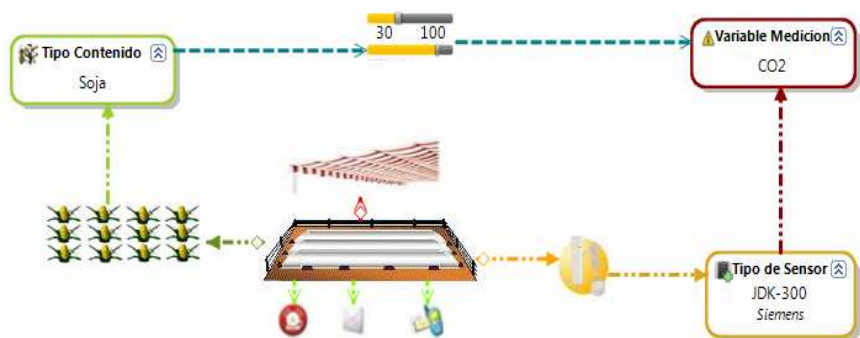


Figura 4: Un sistema especificado con el DSL

## 3.2 DEFINICIÓN DEL GENERADOR DE CÓDIGO

En esta sección se define cuál es el código a generar a partir del modelo creado usando el DSL y cómo es realizada la generación. A partir del modelo diseñado usando el DSL, se generan tres artefactos:

- *Clases C#*: se generan las clases necesarias para representar el modelo.
- *Grafo de objetos*: se generan las instancias de todas las clases generadas anteriormente y se relacionan estos objetos entre sí para representar el modelo creado con el DSL.
- *Loader del Modelo*: se necesita definir un objeto que ofrezca una API que permita poder trabajar con el grafo de objetos antes mencionado. Esta API será particularmente usada por el Framework de Dominio.

Para la generación de estos artefactos hemos definido en archivos *templates* los algoritmos que recorran el modelo y generan los artefactos. A continuación se describe con un ejemplo cuáles son los artefactos generados.

### 3.2.1 GENERACIÓN DE CLASES

En primer lugar se crean las clases C# para representar cada concepto de nuestro modelo. A continuación se da un ejemplo parcial de la clase del silo bolsa generada:

```
/// </summary>
/// Double-derived base class for SiloBolsa
/// </summary>
public abstract partial class SiloBolsaBase {
    public virtual string Nombre { get; set; }
    public virtual string Marca { get; set; }
    public virtual int Ancho { get; set; }
    ...
    public virtual Contenido Contenido { get; set; }
    public virtual System.Collections.Generic.IList<Sensor> SensorList { get; set; }
    public virtual System.Collections.Generic.IList<RegistroMedicion>
        RegistroMedicionList { get; set; }
}
```

Esta clase SiloBolsa contiene las variables de instancias necesarias para representar las propiedades de los silos bolsa. De manera similar se genera el código del resto de las clases (Contenido, Sensor, entre otros).

### 3.2.2 GENERACIÓN DEL GRAFO DE OBJETOS

Una vez creadas todas las clases, el segundo paso es crear todas las instancias de estas clases para representar la información. Por ejemplo, a continuación se muestra el código para la instanciación de un silo bolsa y la asignación de valores a sus variables:

```
SiloBolsa aSiloBolsa = null;
aSiloBolsa = new SiloBolsa();
aSiloBolsa.Nombre = "SiloBolsa1";
aSiloBolsa.Marca = "plastar";
aSiloBolsa.Ancho = 100;
```

Por simplicidad, en este ejemplo sólo se asignan valores a variables de instancias con tipos simples. Sin embargo, en la generación real de código para instanciación de las clases se inicializan todas las relaciones para generar un grafo de objetos consistente con el modelo.

### 3.2.3 GENERACIÓN DEL *LOADER*

El tercer elemento que se genera es un objeto que contiene el grafo de objetos y que brinda una API para acceder a este grafo. El código es el siguiente:

```
public class SilobolsasSystem
{
    private static SilobolsasSystem aSilobolsasSystem;
    public IList<SiloBolsa> Silos { get; set; }
    public IList<VariableMedicion> VariableMedicionList { get; set; }
    public IList<TipoContenido> TipoContenidoList { get; set; }
    public IList<TipoSensor> TipoSensorList { get; set; }
    public static SilobolsasSystem GetSilobolsasSystem() {
        aSilobolsasSystem = new SilobolsasSystem() {
            VariableMedicionList = new List<VariableMedicion>(),
            TipoContenidoList = new List<TipoContenido>(),
            TipoSensorList = new List<TipoSensor>(),
            Silos = new List<SiloBolsa>()
        };
    }
    //...
```

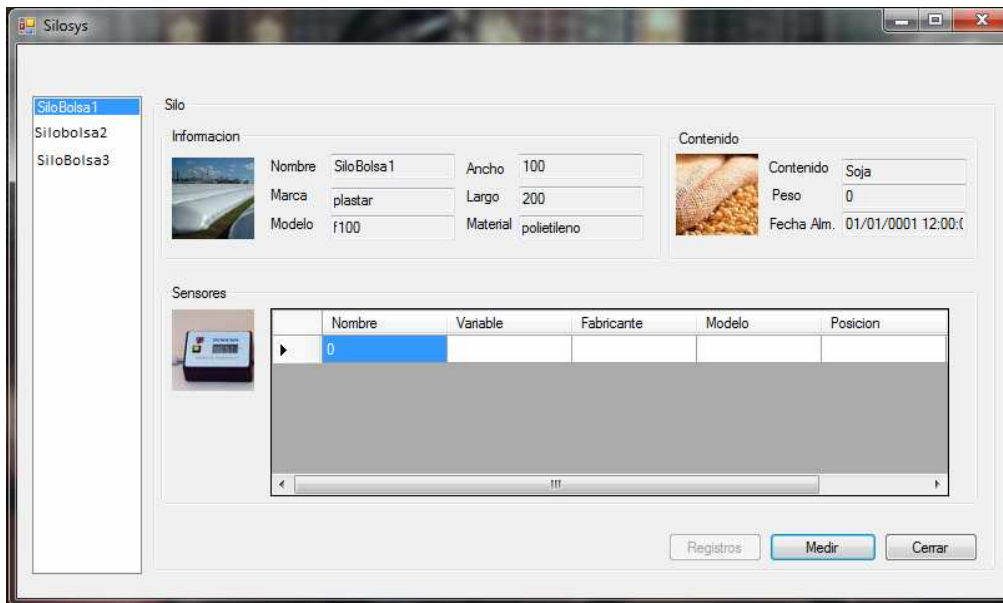
La clase SilobolsasSystem tiene las variables de instancias en las cuales almacenará todas las instancias creadas. Además tiene el método getSiloBolsasSystem(), que funciona de *builder*. Este método crea una instancia del SilobolsasSystem y le carga el grafo de objetos. Esta instancia de SilobolsasSystem será usada luego por el Framework de Dominio para acceder al grafo de objetos.

### 3.3 DEFINICIÓN DEL FRAMEWORK DE DOMINIO

Se definió un framework simple con el fin de mostrar el procedimiento para integrar los artefactos generados a partir del DSL con un framework del dominio. Dicho framework provee las siguientes funcionalidades:

- Listado en pantalla de todas los silos bolsa del sistema.
- Detalle de las propiedades de cada silo bolsa.
- Listado en pantalla de los sensores de un silo bolsa.
- Listado de todas las mediciones hechas por los sensores de un silo bolsa.
- Activación de los sensores para que registren las mediciones.
- Exportación a Excel.

La figura 5 muestra la pantalla principal de la aplicación. A la izquierda, en la pantalla, se listan todos los silos bolsa. Cuando un silo bolsa es seleccionado, su información es desplegada. El botón *Medir* hace que los sensores del silo bolsa seleccionado tomen mediciones y generen registros. Si un silo bolsa posee registros de medición, presionando el botón *Registros* se despliega una nueva pantalla mostrando la información de los registros. Esta nueva pantalla tiene un botón *Exportar*, a través del cual se permite generar un archivo Excel con la información de los registros.



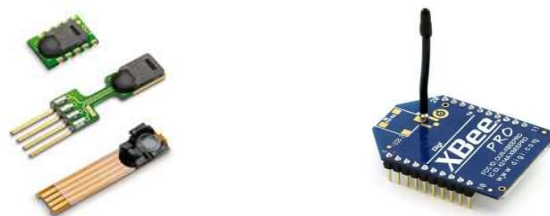
**Figura 5: Pantalla del framework del dominio**

#### **4. DISPOSITIVOS INALÁMBRICOS DE MEDICIÓN**

En esta sección describimos brevemente el hardware diseñado, ver figura 6, para el sensado inalámbrico de los datos que alimentan al sistema de software descrito en las secciones anteriores.

Los sensores se disponen a lo largo del silo bolsa realizando perforaciones en puntos específicos del mismo, donde el número de éstos dependerá de las dimensiones del silo –según lo establece el Instituto Nacional de Tecnología Agropecuaria (INTA) de Argentina, por ejemplo. El módulo de RF de los transmisores enviará los datos al receptor que procesará la información obtenida y permitirá visualizarla. El protocolo utilizado en la comunicación inalámbrica fue zig-bee (ZigBee, 2007) del estándar IEEE 802.15.4 debido a su reducida tasa de transferencia de datos, escalabilidad de red y bajo consumo de potencia. Esta última característica permite asegurar una alta durabilidad de las baterías. De los módulos existentes en el mercado que funcionan con este protocolo se selecciono los X-Bee, ya que no requieren de ningún componente externo y además resultan ser los más económicos.

Para la selección de los sensores se consideraron los siguientes factores: que midan varias variables en forma simultánea para evitar consumos excesivos por circuitería adicional, que posean una salida digital para facilitar su comunicación con la etapa de RF a través de un Uc, que la alimentación sea la misma que la del módulo X-Bee y, por último, que el error de medición sea aceptable. Los sensores que más se adaptaron a dichos requerimientos son los SHTXX de la marca Sensirion.



**Figura 6: Encapsulados de los sensores y módulo Xbee PRO**

Los dispositivos tienen un bajo consumo de energía. Esto permite con pilas AA tener una autonomía de 2 años, correspondiente al tiempo máximo de ensile. El receptor o nodo origen establece una comunicación uno a uno (peer-to-peer) con cada nodo esclavo para recibir la información de la medición. Esta disposición permite un ahorro de energía, ya que solamente se transmitirá si el usuario pide información.

La solución propuesta para introducir el sensor dentro del silo bolsa se observa en la figura 7. Consiste en una varilla de acero inoxidable de 1.30 metros de longitud y 2 cm de diámetro. En uno de sus extremos se ubica el sensor que se protege mediante una pieza roscada con perforaciones. Estas perforaciones permiten que el grano entre en contacto con el sensor sin que éste se dañe, facilitando una buena lectura de temperatura y humedad intergranaria.



**Figura 7: Varilla de medición**

El sensor está montado sobre una placa de circuito impreso y recubierto para su protección con una resina epoxi. El extremo opuesto posee una rosca que se acopla a una manopla, utilizada para facilitar la introducción de la varilla dentro del silo bolsa.

La conexión del sensor con el transmisor se realiza con un cable de 4 conductores mallado para evitar interferencias externas. El transmisor está montado sobre un gabinete plástico tipo Rocket para exteriores y se dispone a un lado de la varilla, pero en el exterior del silo bolsa.

## **5. TRABAJOS RELACIONADOS**

En (Barbieri, 2009) se define un lenguaje gráfico para modelar sistemas en el dominio de la agricultura de precisión. Este lenguaje es una extensión del estándar UML y fue implementado mediante el mecanismo de perfiles. El objetivo es lograr derivación automática de código a partir de los modelos. Si bien es un trabajo relacionado, no incluye el concepto de silo bolsa.

Por otra parte, existen varias propuestas de ambientes de desarrollo dirigido por modelos para aplicaciones de software que incluyen redes inalámbricas de sensores. Estos ambientes consiguen elevar el nivel de abstracción del lenguaje evitando la programación completa de aplicaciones mediante lenguajes tradicionales, y en su lugar se usan modelos que describen la aplicación en función de conceptos y terminología propios de las redes de sensores. Algunos de estos ambientes son presentados en (Naumowicz et al., 2009), (Moñino Martínez, 2008) y (Terfloth and Schiller, 2008). Estos lenguajes permiten configurar redes complejas, lo cual aún no ha sido abordado en nuestro proyecto. Como trabajo futuro nos proponemos incluir estos conceptos en nuestro ambiente de desarrollo, adaptados apropiadamente para los silos bolsa.

Finalmente, este trabajo representa una alternativa a la tecnología que presentamos previamente en (Calegari et al., 2010), así como constituye un avance sobre el trabajo referido. En particular, el trabajo anterior hizo énfasis en la generación de un software para la gestión integral de los silos bolsas, en tanto el actual se centra en la configuración de los silos bolsas y particularmente en la incorporación de un sistema inalámbrico de monitoreo. Esta visión complementaria hace pensar en la construcción de una herramienta que fusione ambas.

## **6. CONCLUSIONES Y TRABAJO FUTURO**

En este artículo presentamos una infraestructura para la construcción de sistemas inalámbricos de monitoreo de silos bolsa. En particular, hemos definido un DSL que permite especificar la configuración específica del sistema de monitoreo. Luego el código ejecutable se deriva automáticamente.



Dada la independencia entre los modelos y el código, la metodología DSM soporta naturalmente la evolución de la arquitectura subyacente (plataforma de ejecución y framework de dominio) sobre la cual se implementa el sistema, reutilizando en su totalidad o en gran parte los modelos generados. De este modo, el software especificado es atemporal en términos tecnológicos y las representaciones de alto nivel que lo definen siguen siendo válidas mientras el dominio del problema siga siendo el mismo. La proximidad modelo-dominio también tiene como ventaja el hecho de que los modelos funcionen como artefactos tanto de captura de requerimientos como de especificación de software. En consecuencia, los clientes (productores rurales) pueden participar con mayor actividad en el desarrollo.

Debido a que el código fuente es producido por los generadores automáticamente, dicho código se supone libre de errores y eficiente en el uso de recursos de ejecución y memoria, por lo que no necesita ser modificado u optimizado. Adicionalmente, todo el código generado seguirá un único estilo de programación y diseño. La faceta desfavorable de esta propuesta consiste en que es necesario invertir trabajo extra en el desarrollo de los componentes de la arquitectura (meta-modelos, editores, generadores, framework de dominio, etc.). Este esfuerzo inicial puede ser amortizado considerablemente debido al incremento en la productividad en el desarrollo alcanzado y aumenta proporcionalmente a la cantidad de “repeticiones” que se tenga en los desarrollos de software llevados a cabo dentro del mismo dominio.

El DSL que implementamos es lo suficientemente extensible como para incorporar en un futuro diferentes tipos de sensores, actuadores y medidas a ser extraídas. Además, el sistema final ha sido diseñado para incorporar predicción del comportamiento de las condiciones internas de los silos bolsa, generación de alarmas ante cierto comportamiento riesgoso y ejecución de actuadores específicos a partir de las alarmas configuradas.

El objetivo a mediano plazo es contar con un sistema inteligente de monitoreo y adaptación automática de las condiciones internas de los granos almacenados en silos bolsa, de bajo costo. En la búsqueda de alcanzar este objetivo, está planificada la incorporación de algún tipo actuador y la inclusión del modelo matemático para la predicción del comportamiento de la humedad y la temperatura, desarrollado para este proyecto.

Consideramos que dada la importancia y difusión que tienen actualmente los silos bolsa en países de la región, éste constituye un aporte relevante que, a través de la incorporación de tecnología, permitirá mejorar las condiciones de almacenamiento de granos en silos bolsa, detectar condiciones riesgosas de acopio, actuar para mejorar factores críticos y como consecuencia, extender el tiempo de conservación de la producción, sin degradar la calidad de la misma.

Este sistema presenta una mejora con respecto a la forma con la que se toman los datos en la actualidad, facilitando y agilizando la obtención de mediciones. Al colocar sensores fijos dentro de los silos que transmiten de forma inalámbrica se elimina la necesidad de tener personal dedicado exclusivamente a dicha tarea, lo que representa una fuerte ventaja económica. Es posible obtener mediciones con una frecuencia mayor que la actual sin la necesidad de realizar perforaciones innecesarias que alteren las condiciones del silo bolsa.

Respecto a la escalabilidad del sistema, una de las ventajas que posee este diseño frente a las opciones actuales es que permite realizar mediciones en un campo con más de un silo bolsa. Teóricamente es posible establecer 65536 sensores; estos pueden funcionar en forma conjunta enviando datos cuando son requeridos al dispositivo receptor. Además, es posible enviar los datos de temperatura y humedad a Internet por medio de un servidor dedicado a esta tarea; esto representa una de las más interesantes mejoras a realizar en un futuro cercano. En la actualidad Internet es un medio masivo de comunicación y se presenta cada vez en una mayor cantidad de dispositivos; la posibilidad de disponer de esta información en la web le permitirá al productor realizar un seguimiento continuo del estado de los granos almacenados.

## **AGRADECIMIENTOS**

Este trabajo fue parcialmente financiado por Microsoft® a través del programa LACCIR RFP 2008 Research Founding Initiative.

## REFERENCIAS

- Barbieri, S. (2009). "Perfil UML para Agricultura de Precisão". *Memorias IX Workshop Brasileiro de Inteligência Competitiva e Gestão do Conhecimento*, Belém, Brazil.
- Bartosik, R., Rodríguez, J., Malinarich, H., Maier, D. (2002). "Silobag: evaluation of a new technique for temporary storage of wheat in the field". *Proceedings of 8th International Working Conference on Stored Product Protection*, pp 1018-1023, CAB International.
- Calegari, D., Luna, C., Canapé, M., Sierra, F., Szasz, N., Pons, C. (2010). "Ingeniería Dirigida por Modelos Aplicada al Control Automático del Almacenamiento en Silos Bolsa". *Memorias del II Congreso de AgroInformática (CAI 2010)*, pp. 623-636, Buenos Aires, Argentina.
- Cook, S., Jones, G., Kent, S., Wills, A. (2007). *Domain-Specific Development with Visual Studio DSL Tools*, Addison-Wesley Professional.
- Fueyo, A., Epeloa, J. (2010). "Medidor Inalámbrico de Temperatura y Humedad Inter-granaria para Silo Bolsa". Tesis de Ingeniería Electrónica, Universidad Nacional de la Plata, Argentina.
- Kelly, S., Tolvanen, J. (2008). "Domain Specific Modeling: Enabling Full Code Generation". Wiley-IEEE Computer Society.
- Martínez, M., Lavie, S. (2010). "Aplicando MDD al desarrollo de sistemas agropecuarios: modelado de sistemas de control de calidad de granos almacenados en silos bolsa". Tesis de Licenciatura en Informática, Universidad Nacional de la Plata, Argentina.
- Moñino Martínez, J. (2008). "Aplicación del Modelado Específico de Dominio a las Redes de Sensores Inalámbricos". Tesis de Ingeniería. Escuela Técnica Superior de Ingeniería de Telecomunicación, Universidad Politécnica de Cartagena, España.
- Naumowicz, T., Schröter, B., Schiller, J. (2009). "Prototyping a Software Factory for Wireless Sensor Networks". *7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009)*; Poster. Berkeley, USA.
- Pons, J., Marinissen, J., Pons, C., Oriente, S. (2009). "Características Higroscópicas de Forrajes Almacenados en Bolsas Plásticas". *Memorias del 1er Congreso Argentino de AgroInformática*, Mar del Plata, Argentina.
- Proyecto LACCIR RFP 2008 "Low Cost Computer-Based System for Quality Evaluation and preservation of Grains Stored in Polymere Bags". (2009). Latin American and Caribbean Collaborative ICT Research (LACCIR), <http://www.lifia.info.unlp.edu.ar/laccir/>, 04/14/2011 (último acceso).
- Stahl, T., Völter, M. (2006). *Model-Driven Software Development*, John Wiley & Sons, Ltd.
- Terfloth, K., Schiller, J. (2008). "Ruling Networks with RDL: A Domain-Specific Language to Task Wireless Sensor Networks". *Rule Representation, Interchange and Reasoning on the Web, International Symposium, RuleML*, Orlando, USA. LNCS 5321 Springer 2008, pp.127-134.
- ZigBee Alliance (2007). <http://www.zigbee.org>, 03/14/2011 (último acceso).

## Autorización y Renuncia

*Los autores autorizan a LACCEI para publicar el escrito en las memorias de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito*