

L.A.Z: Un Lenguaje Específico del Dominio para la Generación Automática de Sitios Web de Instituciones Escolares

Analía Zaldúa¹, Mario Berón¹, Julieta Gatica¹, and Mariano Luzzza¹

¹ Universidad Nacional de San Luis,
Ejército de los Andes 950, San Luis, Argentina
{anzaldua,mberon,mluzza}@unsl.edu.ar
jag81295@gmail.com

Abstract. Antes de la revolución tecnológica actual, los avisos institucionales se debían realizar con la suficiente antelación para que toda la comunidad educativa estuviera debidamente informada, aunque no se disponía de los medios ni del tiempo suficiente para hacerlo. En la actualidad esto cambió debido al avance de las Tecnologías de la Información y de la Comunicación. Se puede observar a nivel global que cada institución tiene presencia en internet a través de un sitio web que la representa y donde realiza la comunicación de los eventos más importantes. La realidad mencionada no sucede con todas las instituciones educativas dado que no cuentan con los recursos necesarios para construir su propio sitio web. Con el fin de solucionar este inconveniente se desarrolló LAZ, un lenguaje específico del dominio que permite especificar un espacio institucional utilizando el vocabulario empleado por los miembros de la institución y cuyo procesador genera automáticamente el entorno web institucional.

1 Introduction

Las instituciones educativas regionales de gestión pública brindan un servicio muy loable a la sociedad como lo es la educación gratuita y de calidad. Para poder llevar adelante dicha tarea, todo el personal realiza mucho esfuerzo para cumplir con los objetivos que propenden a que los estudiantes aprendan significativamente. Entre estas tareas se encuentra la de comunicar información relevante que la institución necesita notificar a su comunidad educativa para estar al tanto de los hechos que acontecen en la misma. Tradicionalmente, este tipo de actividades se lleva a cabo mediante notas en el cuaderno de comunicaciones que los estudiantes hacen firmar por sus tutores, de forma tal, de que conozcan las actividades institucionales. Si bien esta forma de llevar adelante la tarea ha funcionado, muchas noticias e información no llegan a conocerse hasta tanto el interesado se acerque a la institución o bien lea documentos asociados a la misma. A modo de ejemplo se pueden mencionar las noticias de último momento, las orientaciones disponibles en la institución, las fechas de inscripciones que se conocen cerca del inicio de clases, entre otras tantas posibilidades. Con el propósito de resolver este

inconveniente, las instituciones han evolucionado incorporando tecnologías de punta, las cuales han resuelto en gran medida los problemas antes mencionados. En la actualidad muchas instituciones tienen su propio sitio web en donde, cada tanto, van actualizando las noticias que el equipo de gestión desea que la comunidad educativa conozca. Esta aproximación ha resultado un adecuado método de comunicación, sin embargo, aún presenta un inconveniente de difícil solución: *las instituciones que poseen dicho avance tecnológico son, en general, instituciones privadas. Las mismas tienen sustento económico como para pagar a un profesional para que desarrolle el sitio y, si es necesario, lo actualice. Las instituciones públicas adolecen de ésta peculiaridad, con lo cual el diseño de un sitio web es una tarea que puede realizar, de buena voluntad, un docente de la misma, hecho poco probable dado que los mismos están dedicados tiempo completo a la tarea de educar.* Si, por el contrario, un integrante de la comunidad educativa desarrolla el sitio web, la tarea de actualización estaría a cargo de la misma persona dado que, por lo general, será un profesional o idóneo de la informática. Claramente, ninguno de los enfoques mencionados en los párrafos precedentes es viable en las instituciones educativas públicas porque no se disponen de costos ni de idóneos en informática. Para resolver este inconveniente, se desarrolló LAZ (**L**enguaje de **A**nalía **Z**aldúa). LAZ es un lenguaje específico del dominio cuyo propósito es facilitar la creación y actualización de sitios web institucionales, utilizando terminología empleada en las instituciones educativas. Este lenguaje permite especificar la información que estará disponible en el sitio institucional, el cual será generado por P-LAZ (El **P**rocesador de **L**AZ) una herramienta que toma como entrada una especificación LAZ y produce como salida una página web.

El artículo está organizado como se describe a continuación. La sección 2 describe los trabajos relacionados y los proyectos vigentes pertinentes a la temática. La sección 3 presenta el lenguaje específico del dominio LAZ. En la sección 4 se exhibe el procesador de LAZ (P-LAZ) y su arquitectura. La sección 5 presenta un ejemplo de aplicación. Finalmente, en la sección 6 se exponen las conclusiones del artículo y el trabajo futuro.

2 Trabajos Relacionados

Existen numerosas investigaciones referidas al uso, desarrollo e implementación de lenguajes específicos del dominio. En esta sección se describen los principales trabajos relacionados con la temática abordada en el artículo.

L. Cardelli y R. Davies [6] en su investigación presentan un lenguaje de programación para el procesamiento de documentos web llamado WebL. WebL es un lenguaje de scripting orientado a objetos de alto nivel que incorpora dos características novedosas: combinadores de servicios y un álgebra de marcado. Los combinadores de servicios son construcciones de lenguaje que brindan acceso confiable a los servicios web imitando el comportamiento de un internauta cuando se produce una falla al recuperar una página. Bergstra y P. Klint [5] describen cómo se puede usar un lenguaje basado en álgebra de procesos para la descripción

de la arquitectura de coordinación de sistemas de software heterogéneos y distribuidos. Wang, A. W. Appel, J. L. Korn [12] presentan el lenguaje de descripción de sintaxis abstracta Zephyr (ASDL). En su trabajo describen la sintaxis abstracta de las representaciones intermedias del compilador y otras estructuras de datos en forma de árbol. Así como las estructuras léxicas y sintácticas de los lenguajes de programación se describen con expresiones regulares y gramáticas libres de contexto. Giulianelli et. al. en [9] explican una propuesta que permite dentro del enfoque MDA (Model-Driven Architecture) utilizar UML y LEDs en distintos niveles de abstracción y generar mediante transformaciones el código fuente de una determinada aplicación. Esto se debe a que en algunos trabajos académicos surge la disyuntiva de utilizar UML (Unified Modeling Language) ó LED (Lenguaje Específico del Dominio) para modelar un determinado artefacto. Álvarez Herrero [4] en su trabajo sobre las páginas web de los centros educativos y su análisis de la situación en la Comunidad Valenciana refleja que las páginas web se han convertido en un requisito imprescindible de cualquier centro educativo. Más allá de ser plataformas para la información y la comunicación de toda la comunidad educativa, en la actualidad estas páginas web son también escaparates de los centros hacia la sociedad pero que presentan diferentes falencias y la mayoría de los problemas vienen motivados por la falta de una correcta gestión y mantenimiento de las páginas web. Luzzza y su equipo de investigación en [11] presentaron PH-Asistido, una herramienta cuyo principal objetivo consiste en facilitar la enseñanza de la programación a través de la utilización de un dominio atrayente para este propósito: el Proyecto Hoshimi. PH-Asistido es una extensión del Lenguaje del Proyecto Hoshimi con acciones que simplifican la definición e implementación de un algoritmo para el Proyecto Hoshimi. Esto es llevado a cabo a través de un editor visual proactivo que asiste al usuario. La herramienta provee acciones propias del dominio, evitando que el estudiante disperse su atención en obstáculos poco relacionados con el problema a resolver, como la sintaxis del lenguaje. El editor también facilita la parametrización de dichas acciones, acotando las variables al tipo correcto, evitando así otro problema común a la hora de iniciarse en la programación. Gatica y su equipo de investigación [8], presentaron Vinculación 3.0, una herramienta que provee, a las distintas instituciones de nivel medio, terciarias o universitarias, un espacio para compartir su contenido. Esto lo hace a través de una biblioteca digital, una mesa de ayuda y de EDUTECH, un subsistema sencillo que genera páginas web con características básicas de forma automática para las instituciones secundarias. Un área de especial aplicación para los LED es en el ámbito de la enseñanza en programación, ya que posibilitan abstraerse de los problemas particulares de los lenguajes de Propósito General para centrarse en el tema particular que se desea enseñar [10] tal como lo intenta hacer PH-Asistido. En este sentido se pueden mencionar: Scratch [1,3], Alice [2,7], Logo [14,13] son lenguajes específicos del dominio que tienen como finalidad la enseñanza de la programación y son muy utilizados en la academia.

Como es posible percibir, a partir de los trabajos descritos en los párrafos precedentes, los LEDs tienen innumerables aplicaciones. Sin embargo hasta el

momento no fue posible encontrar herramientas que generen espacios institucionales siguiendo el enfoque presentado en este artículo el cual es: *Definir un lenguaje (y su procesador correspondiente) estrechamente relacionado con el que se usa en las instituciones educativas para generar automáticamente sus páginas web. El enfoque además tiene como finalidad facilitar el mantenimiento del espacio web generado.*

3 LAZ: Lenguaje de Analía Zaldúa

LAZ (Lenguaje de Analía Zaldúa) es un lenguaje específico del dominio cuyo propósito es facilitar la creación y actualización de sitios web institucionales. Las palabras claves del mismo, están estrechamente vinculadas con el dominio del problema lo que facilita su aprendizaje y utilización. Esto se debe a que, al utilizar palabras que se emplean en el contexto institucional educativo, los usuarios (directivos, docentes, personal administrativo) se sentirán familiarizados con el lenguaje. LAZ es el punto de partida de diferentes tareas de procesamiento de lenguajes que traducen una especificación en código HTML que especifica el sitio web institucional. Básicamente, una página web institucional debe mostrar el nombre de la institución, información respecto de la misma y del personal. Esto puede ser especificado gramaticalmente de la siguiente manera:

```

institución → < COMIENZO – INSTITUCIÓN >
               quienesSomos informacionInsitucional
               personal
               < FIN – INSTITUCIÓN >

```

donde el no terminal *quienesSomos* es una cadena de caracteres a través del cual la institución da a conocer sus objetivos principales, las bases de su formación, sus valores, entre otras características. El no terminal *informaciónInstitucional* sigue una lógica parecida al no terminal *quienesSomos*; es decir, es una cadena de caracteres compuesta por una o más letras, números y caracteres especiales.

El caso de *personal* es un tanto más complejo dado que, en una institución educativa, se cuenta con diferentes clases de personal. Por una parte, se tiene al equipo de conducción, el cual está formado por un director, un vicedirector y un regente, por otra parte están los docentes y el personal y administrativo. Estas características se especifican de la siguiente manera:

```

personal → directivos docentes administrativos

```

La información asociada a cada miembro de la institución va a depender de lo que las instituciones consideren importante. Atento a este aspecto, se consultó a un grupo de directivos y se concluyó que para el caso de: i) *Directivos*, se registra el nombre, cargo y el correo electrónico o información de contacto; ii) *Profesores*, el nombre, materia, y el correo electrónico de contacto; iii) *Personal administrativo*, el nombre y el correo electrónico de contacto. La especificación gramatical de la información antes mencionada es la siguiente:

```

directivos→ <COMIENZO-DIRECTIVOS> director+ <FIN-DIRECTIVOS>
director→ <COMIENZO-DIRECTOR> nombre cargo contacto <FIN-DIRECTOR>
docentes→ <COMIENZO-DOCENTES> docente+ <FIN-DOCENTES>
docente→ <COMIENZO-DOCENTE> nombre materia contacto <FIN-DOCENTE>
administrativos→ <COMIENZO-ADMINISTRATIVOS>
                administrativo+
                <COMIENZO-ADMINISTRATIVOS>
administrativo→ <COMIENZO-ADMINISTRATIVO>
                nombre contacto
                <FIN-ADMINISTRATIVO>

```

Los no terminales *nombre*, *materia* y *contacto* generan un STRING. A continuación, se muestran todas las producciones que conforman el lenguaje LAZ:

```

institución→ < COMIENZO – INSTITUCIÓN >
                quienesSomos informacionInsitucional
                personal
                < FIN – INSTITUCIÓN >
informaciónInstitucional→ STRING
personal→ directivos docentes administrativos
directivos→ <COMIENZO-DIRECTIVOS> director+ <FIN-DIRECTIVOS>
director→ <COMIENZO-DIRECTOR>
                nombre cargo contacto
                <FIN-DIRECTOR>
docentes→ <COMIENZO-DOCENTES> docente+ <FIN-DOCENTES>
docente→ <COMIENZO-DOCENTE>
                nombre materia contacto
                <FIN-DOCENTE>
administrativos→ <COMIENZO-ADMINISTRATIVOS>
                administrativo+
                <COMIENZO-ADMINISTRATIVOS>
administrativo→ <COMIENZO-ADMINISTRATIVO>
                nombre contacto
                <FIN-ADMINISTRATIVO>
quienesSomos→ STRING
nombre→ STRING
materia→ STRING
contacto→ STRING

```

4 PLAZ: Procesador de LAZ

La figura 1 muestra la arquitectura de P-LAZ. Como se puede observar la arquitectura consta de tres capas: la *Capa de Presentación*, la *Capa de Procesamiento de Lenguajes* y la *Capa de Generación Web*. Esta división se ha realizado con el

propósito de independizar los componentes de forma tal que una modificación en uno de ellos implique pocas o ninguna modificación en el otro.

La *Capa de Presentación* consta de una componente denominada *E-LAZ* (Editor de LAZ) la cual implementa un *Editor Específico del Dominio* que facilita el desarrollo de especificaciones LAZ. El editor posee facilidades para introducir patrones de especificaciones. Esta característica es útil porque reduce la cantidad de errores de sintaxis que el usuario puede introducir a medida que va construyendo la especificación de una institución.

La *Capa de Procesamiento* de Lenguajes está compuesta por los módulos *Lexer*, *Parser* y *Extractor de Información*. El *Lexer* recibe como entrada una especificación LAZ (generada por E-LAZ) y realiza el particionado de la misma en tokens, los cuales son pasados al *Parser* para que se lleve adelante el análisis sintáctico de la especificación. Si la especificación contiene un error sintáctico, el sistema indicará que el árbol de parse no pudo ser construido y le pedirá al usuario que corrija la especificación. En caso contrario, la salida del análisis sintáctico es el árbol de parse correspondiente a la entrada, el cual, a su vez, es la entrada al *Extractor de la Información*. Es importante mencionar que el procesador de lenguaje además de realizar el análisis sintáctico también realiza análisis semántico. Momento en el cual se pueden detectar errores tales como *demasiado directores, institución educativa sin docentes*, entre otros tantos errores. El *Extractor de Información* tiene definidos varios recorridos sobre el *Árbol de Parse*¹ los cuales tienen como finalidad extraer la información requerida por la *Capa de Generación Web*. La información recuperada por el *Extractor de Información* consta de la información institucional plasmada en la especificación la cual fue establecida en base a la experiencia de trabajo en el dominio del problema del equipo de docentes que colaboró en la definición de LAZ. A modo de síntesis se menciona que los recorridos recuperan y almacenan la siguiente información: *Nombre de la Institución*, *Breve Descripción de la Institución*, el *Personal* (directivos, docentes, administrativo), y una *Lista de Noticias*.

La *Capa de Generación Web* consta de varias componentes que colaboran entre sí para generar el código html que define la página web. Tal como fue mencionado en secciones previas, estos módulos hacen uso de la información almacenada en estructuras de datos internas de P-LAZ. Nótese que la generación de código de estas componentes puede ser llevada a cabo utilizando cualquier técnica, considerando que mientras más simple sea más sencillo será comprender y modificar [4].

5 Ejemplo de Aplicación

A modo de prueba se aplicó LAZ y P-LAZ a un colegio estatal de la provincia de San Luis, obviamente se ha resumido la información que se pretende dar a conocer para que el artículo quede autocontenido. La institución seleccionada es el Colegio N° 13 Profesor Roberto Moyano que se encuentra ubicado en la ciudad

¹ Ésta estructura de datos la genera el parser cuando la especificación es correcta

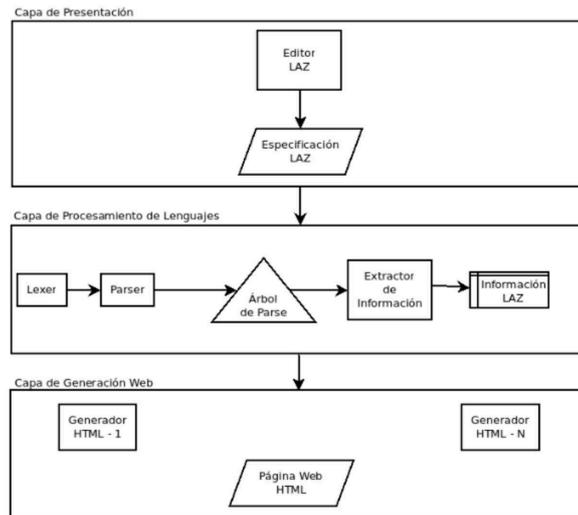


Fig. 1. Arquitectura de P-LAZ

de Juana Koslay en la Provincia de San Luis. El primer paso para comenzar a utilizar la herramienta es recopilar la información que una especificación LAZ necesita de forma obligatoria, a saber: el nombre de la institución, información respecto de quiénes somos, información institucional y personal.

5.1 Especificación LAZ de la Institución

Para comenzar a desarrollar una especificación LAZ el usuario hace uso de E-LAZ, con el cual se genera el patrón de código de una institución sin errores y el usuario solo debe completar con la información de la institución. Para el ingreso de información institucional se debe primero presionar el botón *Nueva Institución* ubicado en la barra de menú y luego completar los campos con la información institucional adquirida (ver figura 2).

Especificación de Personal: Para el ingreso del personal el usuario debe situarse en la sección personal², luego se debe ingresar el tipo de personal que se desea. En este caso se tienen tres tipos de personal: *directivo*, *docente* y *administrativo*. Para ingresar cada uno de esos tipos el usuario simplemente debe posicionarse en el lugar correcto es decir dentro de la sección personal y fuera de un personal específico y luego presionar el botón correspondiente al tipo de personal deseado y completar los datos requeridos. En la figura 2 se muestra la pantalla de P-LAZ con información institucional y un directivo especificado.

Ingreso de Noticias: Para el ingreso de noticias el usuario debe en primer lugar crear un bloque de noticias presionando el botón *Agregar Noticias* y

² Identificado por los terminales < COMIENZO – PERSONAL > y < FIN – PERSONAL >



Fig. 2. Patrón LAZ generado con información

seguidamente colocarse dentro del bloque de noticias creado con anterioridad y agregar una noticia presionando el botón *Agregar Noticia*. En este bloque de noticias se pueden incorporar tantas noticias como el usuario desee (ver figura 3).



Fig. 3. Patrón LAZ generado con información

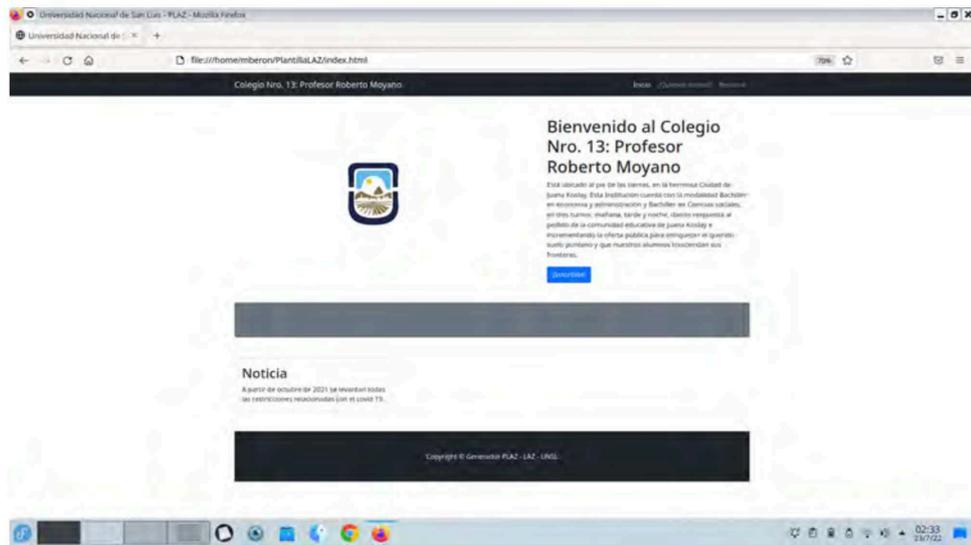


Fig. 4. Resultado del Generador Web

Generación de la Página Web: Una vez construida toda la especificación para la institución se está en condiciones de invocar al generador web. Esta tarea es simple de realizar y se lleva adelante presionando el botón destinado para tal fin (*Generar Página Web*). Luego de realizada esta actividad, si la especificación es correcta, el generador produce como resultado archivos html entre los cuales se encuentra *index.html*, el cual puede ser abierto por un navegador para observar su contenido de la página web especificada por el usuario. La figura 4 muestra el inicio de presentación de la institución.

6 Conclusiones

En este artículo se presentó una investigación que tiene como objetivo: Desarrollar una herramienta que genere espacios web institucionales de manera automática a partir de una especificación escrita en un lenguaje específico del dominio. El espacio web generado es básico e incluye: información de la institución en general, noticias, información académica y contacto con la misma. Para cumplir el objetivo mencionado, se llevaron a cabo las siguientes tareas: i) Se definió LAZ un lenguaje específico del dominio que utiliza terminología propia de las instituciones educativas y que posee construcciones sencillas que facilitan el uso del mismo por parte de los usuarios para especificar los espacios web institucionales; ii) Se implementó P-LAZ un procesador del LAZ el cual además de construir los componentes necesarios para el análisis sintáctico y semántico del lenguaje LAZ provee funcionalidades que facilitan, los miembros de la institución,

la construcción de especificaciones. Además posee una componente que permite generar automáticamente el espacio institucional.

La herramienta presentada en este artículo fortalece el proceso comunicacional entre la comunidad educativa y la institución escolar que es considerado un déficit teniendo en cuenta el contexto actual. El proceso de comunicación mostrará la institución educativa en un portal web de creación y mantenimiento sencillo y económico lo que justifica la importancia de automatizar los procesos.

Como trabajos futuros se pueden mencionar: i) Incrementar la expresividad de LAZ para que las instituciones puedan comunicar más información; ii) Auto-gestión del ambiente de trabajo, en cuanto a la elección de fuentes, formatos, colores del ambiente y que la herramienta sea adaptativa a los requerimientos y elecciones del usuario.

References

1. <http://scratch.mit.edu>.
2. <https://www.alice.org/>.
3. Achal. *Teach Yourself Animation Coding in Scratch 3: Programming for Kids and Beginners*.
4. Juan-Francisco Alvarez-Herrero and Rosabel Roig-Vila. Las páginas web de los centros educativos. análisis de la situación actual en la comunidad valenciana - the websites of schools. analysis of the current situation in the valencian community. *Revista de Comunicación de la SEECI*, pages 129–147, 11 2019.
5. Jan A. Bergstra and Paul Klint. The discrete time toolbus—a software coordination architecture. *Science of Computer programming*, 31(2-3):205–229, 1998.
6. Luca Cardelli and Rowan Davies. Service combinators for web computing. *IEEE Transactions on Software Engineering*, 25(3):309–316, 1999.
7. Wanda P Dann, Don Slater, Laura Paoletti, and Dave Culyba. *Alice 3 to Java: Learning Creative Programming through Storytelling and Gaming*. Pearson, 2017.
8. Julieta Gatica and Camila Olguín. *Vinculación Educativa 3.0: Una herramienta para disminuir las brechas educativas regionales*. Proyecto final integrador de ingeniería en informática, 2021.
9. Daniel Alberto Giulianelli, Claudia Pons, Rocío Andrea Rodríguez, Pablo Martín Vera, and Víctor Fernandez. Integrando uml y dsl en el enfoque mda. In *XVI Congreso Argentino de Ciencias de la Computación*, 2010.
10. David A Ladd and J Christopher Ramming. Application languages in software production. In *USENIX 1994 Very High Level Languages Symposium (USENIX 1994 Very High Level Languages Symposium)*, 1994.
11. Mariano Luzzi, Mario Marcelo Beron, and Pedro Rangel Henriques. Ph-helper-a syntax-directed editor for hoshimi programming language, hl. In *1st Symposium on Languages, Applications and Technologies*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.
12. Daniel C. Wang, Andrew W. Appel, Jeff L. Korn, and Christopher S. Serra. The zephyr abstract syntax description language. In *IN PROCEEDINGS OF THE CONFERENCE ON DOMAIN-SPECIFIC LANGUAGES*, pages 213–227, 1997.
13. Daniel Watt. *Learning with Logo*.
14. Molly Watt and Daniel Watt. *Teaching with Logo: Building blocks for learning*. Addison-Wesley Longman Publishing Co., Inc., 1985.