

# Construcción de una herramienta de soporte para la creación de diagramas de clases destinados a estudiantes con disminución visual.

Ing. Clarisa Liliana Stefanich<sup>1,2</sup>, Nicolas Gabriel del Valle<sup>1,3</sup>, Ing. Laura del Carmen Ligorria,<sup>1,4</sup>  
Ing. Roxana María Manera<sup>1,5</sup>, Ing. Lorena Daniela Peralta<sup>1,6</sup>, Ing. Alejandra Di Gionantonio<sup>1,7</sup>,  
Juan Gabriel Moreno<sup>1,8</sup>

<sup>1</sup>UTN Facultad Regional Córdoba

{<sup>2</sup>clarystefanich, <sup>3</sup>dv.nico13, <sup>4</sup>liuniversidad, <sup>5</sup>roxanamanera, <sup>6</sup>peralta.lorena.d, <sup>7</sup>  
ing.alejandradg}@gmail.com, <sup>8</sup>gabriel\_2008@live.com.ar

## RESUMEN

En los últimos tiempos se ha observado la necesidad de accesibilidad de estudiantes con disminución visual total o parcial a la información concerniente a diagramas visuales llevados a cabo en cátedras de análisis, diseño y programación de software en carreras de ingeniería, tecnicatura y cursos orientados a la informática. Teniendo en cuenta la importancia de los diagramas UML en el cursado de estas asignaturas, se plantea la construcción de una herramienta de software que permita la generación y lectura de diagramas UML por medio de un lenguaje textual formalizado, y que a su vez brinde la posibilidad de interacción por parte del docente y el resto de los estudiantes mediante la manipulación de elementos gráficos (formularios, elementos visuales del diagrama), favoreciendo la construcción de conocimientos colaborativos entre estudiantes y docentes.

**Palabras claves:** UML, visibilidad reducida, diagrama de clase, accesibilidad.

## CONTEXTO

El presente trabajo se realiza en el Área de Investigación del Departamento de Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional - Facultad Regional Córdoba.

## 1. INTRODUCCIÓN

En relación a las herramientas educativas actuales con respecto a la accesibilidad y adaptabilidad en los diferentes niveles de educación, existen esfuerzos enfocados a la

educación preescolar, primaria y secundaria, pero no así en el nivel superior y terciario.

En el ámbito legal y de estándares, se detectan avances en la accesibilidad de la información y del medio físico tanto a nivel nacional como internacional [1][2].

En relación a los antecedentes relevados, se observa la presencia de soluciones propuestas por diferentes universidades y organizaciones, las cuales se describen brevemente a continuación:

En [3] los autores desarrollaron la aplicación “TeDUB”, la cual traduce el contenido de dibujos técnicos (electrónica, arquitectura e ingeniería de software) en descripciones mapeadas jerárquicamente, en formato XML, las cuales son accedidas por dispositivos de salida, voz, efectos de sonido (2D y 3D), Braille y dispositivos hápticos. El usuario puede interactuar con el sistema y puede explorar la jerarquía del dibujo técnico usando el teclado o un joystick. También se puede usar una pantalla braille para navegar.

En [4] los autores crearon el software “Audible Browser”, la cual interpreta archivos XMI 2.1, especialmente de diagramas de clases y estados. Luego el programa representa los nodos del diagrama en grupos de tabs. Un grupo de tabs es un conjunto de ítems comunes que pueden ser leídos usando las teclas de arriba y abajo. Los grupos de tabs para una clase son el nombre, atributos, operaciones y asociaciones, permitiendo diferentes niveles de navegación, desde lo más general a los más específico. Para transmitir la estructura de un diagrama, el software lo presenta utilizando sonidos no verbales en combinación con audio verbal. El audio no verbal se transmite más a la

izquierda o derecha del auricular representando la posición “X” del elemento en el diagrama y el tono representa la posición “Y”.

En [5] los autores generaron “Prisca”, una aplicación que permite transformar los elementos de un diagrama UML a un vista 3D, para luego ser visualizada por medio de una impresora 3D. Los textos son impresos utilizando Braille.

Por otro lado, la página web yUML[6], permite la generación de diversos diagramas UML por medio de un lenguaje textual.

En las herramientas observadas, se visualizan diversas áreas de mejora:

- Disponibilidad offline, permitiendo usar la herramienta sin necesidad de Internet.
- Interacción transparente entre docente y estudiante, manteniendo a cada uno en su contexto de trabajo/conocimiento. Esto implica que la herramienta provee dos vistas simultáneas del mismo diagrama, optimizadas cada una para el docente y el estudiante respectivamente.
- Lenguaje textual compatible con los sintetizadores de voz en español, disponibles en los lectores de pantalla.
- Permitir la creación y edición de diagramas desde cualquiera de las dos vistas: docentes o estudiantes.
- Multiplataforma.
- Creación y edición de diagramas UML de clases, casos de uso, estados y secuencia. Los cuales son los más usados en ámbitos de formación superior.

Estas áreas serán abordadas por el presente trabajo en los siguientes apartados.

## 2. OBJETIVOS

El objetivo de este trabajo de investigación es: “La construcción de una herramienta de software que permita la conversión de un lenguaje formalizado a un diagrama gráfico y su transformación inversa para poder ser interpretados por estudiantes con disminución visual total o parcial por medio de un lector de pantalla”.

## 3. HERRAMIENTAS

Para la creación del módulo de renderizado de la herramienta se utilizó la librería React [7], la cual permite administrar los eventos y el DOM correspondientes a un entorno web. Esto posibilita el empleo de múltiples componentes gráficos probados.

El software está dividido en módulos cohesivos, es decir que los contenidos dentro del mismo están relacionados semánticamente. Cada módulo es flexible en cuanto a la implementación de los algoritmos necesarios para lograr los objetivos del mismo. Esto se logró por medio del uso de la composición de objetos y el patrón estrategia.

Para la construcción de la sintaxis empleada por el estudiante para la generación de diagramas UML, se utilizó como base una gramática libre de contexto, un analizador sintáctico y una definición de la semántica del lenguaje. Dichos conceptos fueron definidos por medio de la librería Ohm [8].

Al emplear React en la construcción de la aplicación, se dispone de la API Canvas, la cual permite la creación y gestión de elementos gráficos. Para facilitar la generación visual de diagramas UML, se empleó la librería Konva [9], la cual es una capa de abstracción sobre Canvas, ocultando la complejidad inherente de dicha API.

HTML y los navegadores web poseen mecanismos maduros de sincronización para con los lectores de pantalla. Con la finalidad de utilizar esta ventaja y para satisfacer el requerimiento de aplicación offline, se empleó la librería Electron [10], la cual incluye funcionalidades que habilitan la ejecución de aplicaciones construidas con tecnología web para ser ejecutadas en un entorno de escritorio. El software queda encapsulado dentro de un navegador minimalista (chromium) y visualizado por medio de un motor de renderizado. La librería anterior satisface el requerimiento de multiplataforma, ya que permite la creación de ejecutables de los sistemas operativos más utilizados (Windows, Linux y MAC OS).

#### 4. ESTADO DE AVANCE

El equipo implementó una herramienta de software que posibilita la conversión bidireccional simultánea entre una definición en lenguaje estructurado y un diagrama gráfico de clases, permitiendo de esta manera la interacción entre un estudiante y un docente, o para los estudiantes entre sí, en un entorno académico.

La interfaz cuenta con dos vistas, una corresponde al estudiante con disminución visual y la otra corresponde al docente o a otro estudiante. Esto fue concebido con miras a separar los ámbitos de interacción de forma tal que el estudiante no necesite interactuar con elementos visuales, y el docente o compañero de grupo no requiere aprender el lenguaje formal de generación de diagramas.

Con respecto a la vista del estudiante con disminución visual, la misma cuenta con dos sub-vistas, una corresponde a una descripción detallada de los elementos del modelo, y la otra corresponde a la definición sólo de las relaciones entre los mismos. Esta última sub-vista permite al estudiante realizar un análisis de grano grueso del diagrama (nombre de entidades y relaciones entre ellos).

El estudiante puede crear, modificar o borrar elementos del diagrama por medio de un lenguaje formal. El mismo está en español y fue construido teniendo como foco la correcta interpretación del mismo por parte de un lector de pantalla con configuración genérica en español.

En referencia a la vista del docente o compañero de grupo, se visualiza el diagrama construido y por medio de formularios e interacción con los elementos visuales del diagrama, se puede crear, editar o borrar los componentes de dicho diagrama.

Ambas vistas mencionadas están sincronizadas entre sí, teniendo como base el vector de elementos creados y sus propiedades.

Con fines a visualizar la interacción entre un docente/compañero de grupo y un estudiante con disminución visual, se expone a continuación el siguiente enunciado:

#### Ejemplo de aplicación

El docente plantea la realización de un diagrama de clases sobre la funcionalidad “Generación de préstamos de libros” correspondiente al dominio “Biblioteca”.

El estudiante crea el diagrama solicitado por medio del lenguaje, como muestra la figura 1:



Figura 1. Lenguaje de generación de diagramas.

El docente visualiza el modelo gráfico elaborado por el estudiante, el cual se observa en la figura 2:

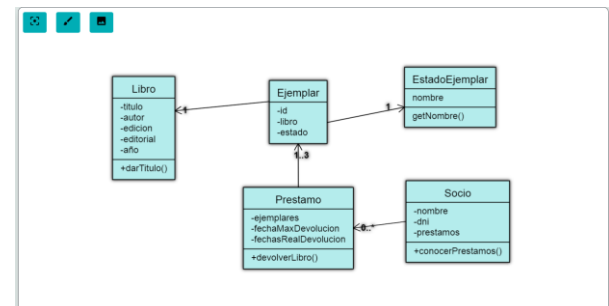


Figura 2. Diagrama de clases generado.

El docente analiza el modelo, y concluye que es necesario de acuerdo al dominio, la inclusión de una nueva clase llamada “Estado Socio”. La misma es generada por medio de la interfaz de la herramienta, sin necesidad de conocer el lenguaje de generación de diagramas utilizado por el estudiante previamente. En las figuras 3 y 4 se observa cómo el docente crea esta nueva clase y su relación correspondiente:

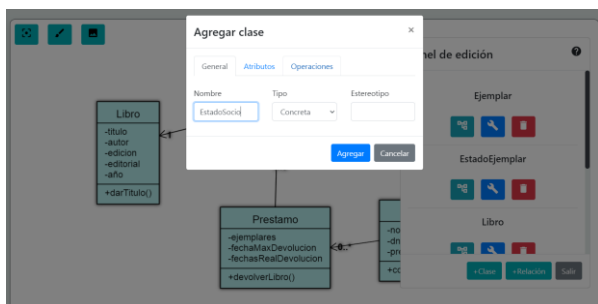


Figura 3. Creación clase “EstadoSocio”.



Figura 4. Creación relación “Socio” a “EstadoSocio”.

El diagrama gráfico se actualiza con las modificaciones realizadas por el docente, reflejadas en la figura 5:

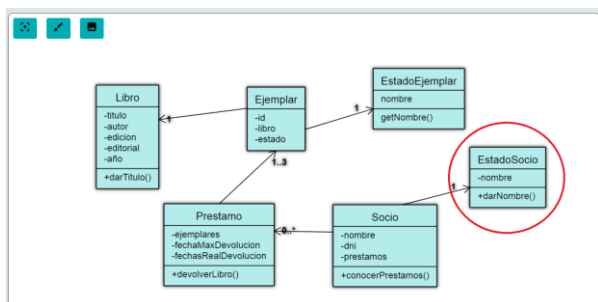


Figura 5. Diagrama con nueva clase.

A su vez, esta modificación se sincroniza con el lenguaje textual percibido por el estudiante, apreciado en la figura 6:

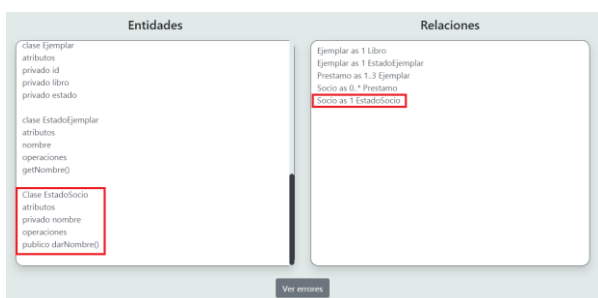


Figura 6. Lenguaje textual actualizado.

## 5. RESULTADOS Y AVANCES

Considerando la interacción del docente, la misma ha sido validada por medio de pruebas ejecutadas por parte de educadores de nivel superior, optimizando la usabilidad de la aplicación.

Hasta el momento el software permite tanto a docentes como a estudiantes la visualización, creación, y la modificación de diagramas de clases.

La investigación proseguirá con la realización de nuevos tipos de diagramas UML, la adaptación de la interfaz para que la misma sea compatible con las guías WCAG [11] y la validación de la herramienta por parte del estudiante.

## 6. FORMACIÓN DE RECURSOS HUMANOS

El equipo de trabajo está conformado por docentes-investigadores pertenecientes a la carrera de grado de Ingeniería en Sistemas de Información.

El grupo está compuesto por una Directora, tres ingenieras investigadoras de apoyo, una ingeniera y estudiantes aspirantes a incorporarse a la carrera de investigador.

Este proyecto enriquecerá la experiencia en la carrera de investigador de los integrantes del mismo.

## 7. BIBLIOGRAFÍA

[1] Ley de educación superior N° 25573. Modificación ley N° 24521. (2002, 11 de Abril). [Online]. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/70000-74999/73892/norma.htm>.

[2] Guenaga, L., Barbier, A. and Eguíluz, A. “La accesibilidad y las tecnologías en la información y la comunicación”. *Trans. Revista de Traductología*, pp. 155-169. 2007.

[3] Födisch, M., Crombie, D., & Ioannidis, G.T. “TEDUB : Providing access to technical drawings for print impaired people”. 2002.

[4] Coburn, S., Owen C. “UML Diagrams for Blind Programmers” en ASEE North Central Section Conference, 2014.

[5] Doherty B., Cheng B. “UML Modelling for Visually-Impaired Persons” en First International Workshop on Human

Factors in Modeling co-located with ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS 2015), Ottawa, Canada, 28 de Septiembre, 2015.

[6] yUML. [Online]. Disponible en: <https://yuml.me/>.

[7] Getting Started - ReactJS. [Online]. Disponible en: <https://reactjs.org/docs/getting-started.html>

[8] GitHub - harc/ohm: A library and language for building parsers, interpreters, compilers, etc. [Online]. Disponible en: <https://github.com/harc/ohm>

[9] Konvajs. Konvajs/Konva: Konva.js is an HTML5 Canvas javascript framework that extends the 2D context by enabling Canvas interactivity for desktop and mobile applications. [Online]. Disponible en: <https://github.com/konvajs/konva>.

[10] Electron. Electron/Electron: Build cross-platform desktop apps with JavaScript, HTML, and CSS. GitHub. [Online]. Disponible en: <https://github.com/electron/electron>

[11] W3C WAI. Web content accessibility guidelines (WCAG) overview. [Online]. Disponible en: <https://www.w3.org/WAI/standards-guidelines/wcag/>.