

Sistema de Marcado de Estructuras de Código Fuente para Programadores con Discapacidad Visual - Mejoras de Interfaz e Integración con Herramientas de Desarrollo

Source Code Structures Markup System for Visually Impaired Programmers - Interface Improvements and Integration with Development Tools

Tomás Assenza, Valentín Fontana

Dpto. Ing. en Sistemas de Información, Facultad Regional Santa Fe,

Universidad Tecnológica Nacional

tassenza@frsf.utn.edu.ar – valefontana15@gmail.com

Resumen

Al realizar tareas de programación se aprovechan distintas utilidades que permiten al desarrollador posicionarse en el código y mejorar su edición para obtener un programa funcional como respuesta a un problema. Estas herramientas están provistas en los entornos integrados de desarrollo (IDEs), que conforman soluciones completas para generar software. Los programadores con discapacidad visual se enfrentan con la problemática de que la mayor parte de la información que brindan estos entornos proviene de pistas visuales a las cuales no pueden acceder. En este trabajo se extiende un sistema de marcado estructural de código fuente (SIMAE) para asistir a estos desarrolladores en la navegación y posicionamiento del código, introduciendo marcas a modo de comentarios que les permiten saber dónde inicia y finaliza cada estructura de código. Se presentan mejoras en las interfaces de usuario y la integración de la herramienta en distintos IDEs de amplio uso en la industria.

Palabras clave

Programación, Discapacidad visual, Código fuente, IDE

Abstract

When carrying out programming tasks, different utilities are used that allow the developer to position himself in the code and improve its edition to obtain a functional program in response to a problem. These tools are provided by integrated development environments (IDEs), which make up complete solutions to generate software. Visually impaired programmers face the problem that most of the information provided by these environments comes from visual cues that they cannot access. In this work, a source code structural markup system (SIMAE) is extended to assist these developers in the navigation and positioning on the code, introducing tags as comments that allow them to know where each code structure begins and ends. Improvements are presented in the user interfaces and the integration of the tool in different IDEs widely used in industry.

Keywords

Programming, Visual disabilities, Source code, IDE

Introducción

Existe una significativa demanda de desarrolladores de software, lo que motiva a cada vez más personas a dedicarse a esta industria y a estudiar carreras relacionadas con Informática o Ciencias de la Computación. El crecimiento de esta industria y la complejidad del proceso de desarrollo de software dieron lugar a la creación de herramientas para asistir al programador de modo que pueda realizar su trabajo correctamente produciendo software fácil de comprender y mantener por otras personas.

Los entornos integrados de desarrollo (IDEs) cuentan con utilidades para realizar con mayor facilidad las tareas de creación y edición de código. Como ejemplo de estas ayudas podemos mencionar pistas tales como el resaltado de código o la indentación, donde se utilizan espacios como forma de diferenciar contextos de código. El problema que actualmente existe es que muchas características brindadas por los IDEs no están disponibles para desarrolladores con discapacidad visual, por tener fuertes componentes visuales en su funcionamiento.

La persona ciega utiliza herramientas específicas para trabajar con una computadora, entre las cuales se destacan los lectores de pantalla, que permiten identificar e interpretar lo que se muestra en la misma presentándole la información mediante sintetizadores de texto a voz, íconos sonoros, o una salida Braille. A la hora de desarrollar código, los lectores de pantalla interactúan con los editores de texto de los IDEs leyendo una línea por vez. Por lo tanto, el programador ciego debe recorrer todo el código secuencialmente para llegar a tener un contexto que le permita interpretar correctamente un programa y tener una percepción global del mismo.

Existen algunos trabajos en la literatura sobre esta temática. Baker et al. [1] describen la existencia de barreras claramente identificables para el aprendizaje en la educación superior referida a Ciencias de Computación para las personas ciegas. Además identifica a Visual Studio, Eclipse, Notepad y Notepad++ como los editores o IDEs más usados dentro de la muestra que realizaron, indicando las dificultades que se encuentran los programadores ciegos en el uso y aprendizaje de los mismos. Por otro lado, Falase et al. [2] presentan una herramienta que permite a desarrolladores con discapacidad visual diferenciar los distintos niveles de indentación mediante dispositivos sensoriales.

En el contexto de las tecnologías de asistencia para programadores ciegos, en este trabajo se describen y presentan avances obtenidos en el desarrollo de un Sistema de Marcado Estructural de Código (SIMAE). Esta herramienta se vale de la adición de comentarios para agregar marcas a cada una de las estructuras de código, las cuales son obtenidas y reproducidas por los lectores de pantalla permitiendo al programador ubicarse en el comienzo y fin de las estructuras.

El presente trabajo está enmarcado en el Proyecto de I+D “Desarrollo y Despliegue de un Sistema de Marcado Estructural de Código Fuente para Programadores con Discapacidad Visual” (código SIECAFE0008484TC). El software cuenta con un prototipo desarrollado y presentado previamente por Assenza et al. [3,4]. En esta contribución se amplía la producción previa con nuevas funcionalidades en las interfaces de usuario, se describe la integración de la herramienta en algunos IDEs y se comenta sobre las devoluciones de los primeros usuarios (testers).

La herramienta se encuentra publicada en un sitio Web [5] y disponible para su descarga libre y gratuita. En dicho sitio también se encuentran las instrucciones de uso y documentación para integrar la herramienta en los distintos IDEs, para facilitar su uso en el desarrollo cotidiano de software que pueda realizar el usuario.

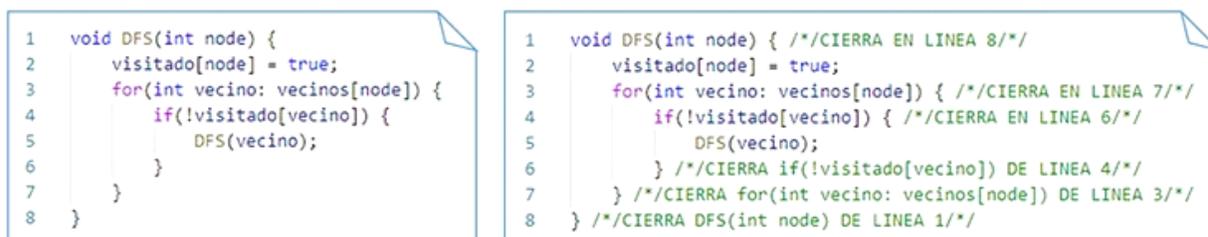
Desarrollo

SIMAE fue desarrollado en el lenguaje de programación Java, utilizando el compilador de gramáticas ANTLR [6] para generar los componentes de análisis léxico y sintáctico que permiten identificar correctamente las estructuras gramaticales de los lenguajes de programación de interés.

Actualmente, la utilización de SIMAE está disponible mediante una interfaz por línea de comandos (CLI – command line interface) y una interfaz gráfica (GUI – graphical user interface). La herramienta soporta los lenguajes C++, Java y Python. La elección inicial de los mismos se basó en su elevado uso dentro de la industria del software.

Además, en la actualidad se está trabajando en el soporte de los lenguajes C# y Rust, y se planea agregar nuevos lenguajes de forma continua. Paralelamente, se trabaja en el rediseño de los módulos de código para mejorar la mantenibilidad, lo que permitirá la introducción de nuevas funcionalidades de manera más sencilla. También, en la mejora de la internacionalización, incorporando nuevos idiomas para la traducción del marcado y de los mensajes de retroalimentación y, por último, en la automatización del proceso de instalación del sistema.

En la Fig. 1 se puede observar un ejemplo del funcionamiento de SIMAE. Una vez que se realiza el proceso de marcado, se modifica el código fuente original incorporando marcas, que son comentarios insertados en las líneas de inicio y fin de las estructuras presentes en el código del archivo de entrada, junto con los números de línea en los que inician y finalizan las mismas. La herramienta permite a su vez el desmarcado, es decir la eliminación de las marcas introducidas previamente.



```
1 void DFS(int node) {
2     visitado[node] = true;
3     for(int vecino: vecinos[node]) {
4         if(!visitado[vecino]) {
5             DFS(vecino);
6         }
7     }
8 }
```

```
1 void DFS(int node) { /*CIERRA EN LINEA 8*/
2     visitado[node] = true;
3     for(int vecino: vecinos[node]) { /*CIERRA EN LINEA 7*/
4         if(!visitado[vecino]) { /*CIERRA EN LINEA 6*/
5             DFS(vecino);
6         } /*CIERRA if(!visitado[vecino]) DE LINEA 4*/
7     } /*CIERRA for(int vecino: vecinos[node]) DE LINEA 3*/
8 } /*CIERRA DFS(int node) DE LINEA 1*/
```

Figura 1 – Marcado de código por SIMAE

Utilización por interfaz gráfica de usuario (GUI)

Una de las opciones disponibles para utilizar SIMAE es mediante una interfaz gráfica de usuario, adaptada mediante herramientas de accesibilidad. En la Fig. 2 se muestra la interfaz generada, la cual permite seleccionar el lenguaje de programación y trabajar con una lista de archivos mediante la adición o eliminación de elementos, para luego proceder al procesamiento de los mismos. Entre las facilidades de accesibilidad se admite la utilización de atajos de teclado que facilitan el accionamiento de los botones. Los atajos de teclado se describen en la Tabla 1.



Figura 2 – Interfaz Gráfica de Usuario

Atajo de teclado	Boton accionado
Alt + A	Agregar Archivos...
Alt + C	Agregar Carpetas...
Alt + Q	Quita el archivo seleccionado de la lista
Alt + T	Quita todos los archivos de la lista
Alt + S	Activa la opción "Solo quitar marcas"
Alt + P	Procesa los archivos de la lista

Tabla 1 – Lista de atajos de teclado en GUI.

Utilización mediante interfaz por línea de comandos (CLI)

En la Tabla 2, presentada a continuación, se resume el listado revisado de parámetros y opciones de uso de la aplicación mediante la CLI. Al utilizar esta interfaz es necesario especificar el archivo de entrada. Además, como parámetro opcional, se puede indicar el archivo de salida mediante la opción --out. En el caso que este parámetro no se incluya, se sobrescribe directamente el archivo de entrada. Se permite además especificar el lenguaje de programación mediante la opción --lang, o el mismo es detectado automáticamente en caso de no indicar la opción a partir de la extensión del archivo de entrada. También se permite especificar el idioma en el que se realizará el marcado, el cual se usará además en la retroalimentación al usuario, mediante la opción --locale. Esta opción actualmente permite seleccionar los idiomas español e inglés, y en caso de no indicarse utiliza por defecto aquel del sistema operativo.

Opcion	Nombre completo	Parametro	Descripción
-o	--out	<outputFile>	Archivo de salida (por defecto es <inputFile>)
-p	--lang	<language>	Lenguaje de programación del archivo (java8, c++, python3)
-l	--locale	<locale>	Idioma utilizado para el marcado y mensajes
-u	--untag		Quita las marcas de SIMAE de <inputFile>
-s	--sound		Reproduce un sonido para indicar el resultado del proceso (sólo CLI).
-g	--gui		Muestra la interfaz gráfica
-h	--help		Muestra la descripción de uso
-v	--version		Versión de SIMAE

Tabla 2 – Lista de parámetros para CLI.

Integración con IDEs

Para mejorar la experiencia de usuario se trabajó en la integración de esta herramienta en distintos entornos integrados de desarrollo mediante tareas automatizadas (macros), facilitando el uso mediante una serie de atajos de teclado. SIMAE soporta actualmente la integración con Visual Studio Code, Eclipse, Notepad++ y Zinjal, permitiendo realizar el marcado y desmarcado directamente desde los mismos mediante la ejecución de la aplicación en segundo plano.

Para decidir qué IDEs serían los primeros en ser implementados se tuvo en cuenta los comentarios de los usuarios testers en base a su experiencia de programación en los lenguajes soportados.

Retroalimentación de usuarios

Una vez desarrollado un primer prototipo funcional de SIMAE se comenzó el contacto con usuarios testers para obtener sus devoluciones, permitiendo así evaluar y confirmar la utilidad de la herramienta y considerar la implementación de nuevas funcionalidades. Para esto, en la página web de SIMAE [5] (Fig. 3), se incluyó un formulario de Google donde, además de permitir que los usuarios puedan anotarse para participar como testers (evaluadores), cuenta con la posibilidad de registrarse para participar de un grupo de WhatsApp.

Asimismo, se mantuvieron reuniones con los primeros usuarios para asistirlos en la instalación de la herramienta y conocer mejor su contexto a la hora de trabajar en programación.

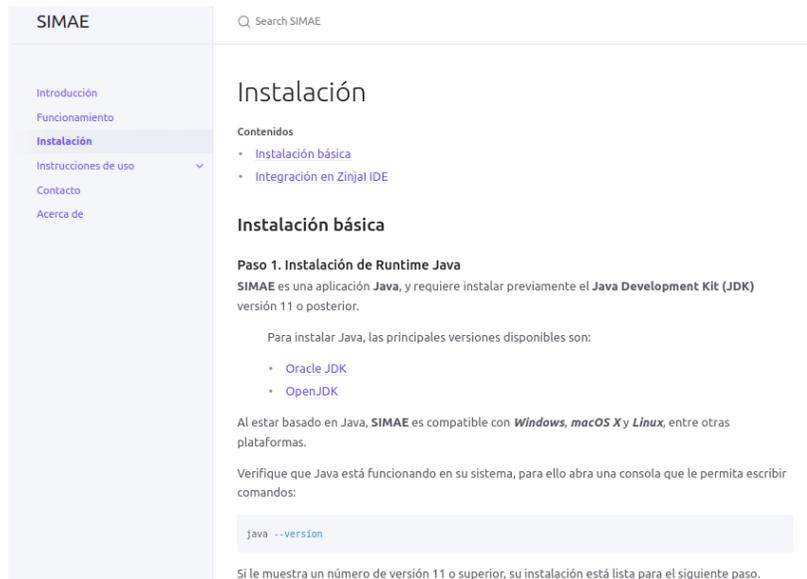


Figura 3 – Página web de SIMAE

Conclusiones y trabajos futuros

La herramienta se encuentra disponible para su descarga desde la página web, así como su código fuente el cual puede accederse desde un repositorio público en Github [7], existiendo por lo tanto la posibilidad de recibir sugerencias o aportes a la herramienta por la comunidad.

La retroalimentación de los usuarios mostró que una herramienta de marcado estructural como la propuesta es de mucha utilidad para programadores con discapacidad visual, para facilitarles la contextualización dentro de un código fuente dado y agilizarles el proceso de codificación. Por otro lado, se enfatizó la necesidad de tener una adaptación completa de la herramienta en los distintos IDEs, para poder usarla con mayor facilidad.

Como líneas de trabajo futuro, se prevé finalizar la implementación del soporte para los lenguajes C# y Rust, automatizar la integración en los IDE descritos en el trabajo (para evitar la necesidad de configurarlos manualmente), y generar un instalador para facilitar la adopción de la herramienta. A mediano/largo plazo se espera incrementar el soporte de la herramienta con los distintos IDEs disponibles en el mercado, incluyendo la posibilidad de re-diseñar el software en forma de plug-in dentro de los entornos más utilizados.

Referencias

- [1] Baker, C., Bennett, C., Ladner, R.E. (2019). Educational Experiences of Blind Programmers. Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, 759–765.
- [2] Falase, O., Siu, A., Follmer, S. (2019). Tactile Code Skimmer: A Tool to Help Blind Programmers Feel the Structure of Code. The 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19). Association for Computing Machinery, New York, NY, USA, 536–538.

[3] Assenza, T.; Ballardini, C.; Marchetti, P.; Golobisky, M.F. (2020). Herramienta de Soporte para Programadores con Discapacidad Visual mediante el Mercado de Código Fuente. IEEE Congreso Bional de Argentina (ARGENCON 2020), 1-4.

[4] Assenza, T.; Ballardini, C.; Marchetti, P.; Golobisky, M.F. (2020). Extensión e Integración de un Sistema de Mercado de Estructuras de Código para Programadores con Discapacidad Visual. Encuentro Argentino y Latinoamericano de Ingeniería - CADI CLADI CAEDI 2021.

[5] Página web de SIMAE. <https://simae.info>

[6] Página web de ANTLR. <https://www.antlr.org/>

[7] Repositorio de SIMAE en Github. <https://github.com/tiflo-sf/simae>

Agradecimientos

Agradecemos a la Universidad Tecnológica Nacional - Secretaría de Ciencia, Tecnología y Posgrado, por el financiamiento del proyecto PID “Desarrollo y Despliegue de un Sistema de Mercado Estructural de Código Fuente para Programadores con Discapacidad Visual” - código SIECAFE0008484TC, dentro del cual se enmarca este trabajo, y al equipo del proyecto incluyendo los docentes Dr. Pablo Marchetti (director), Dra. Ma. Fernanda Golobisky (co-directora), Ing. César Ballardini e Ing. Daniel Ambort, por su constante ayuda y conducción en la realización de nuestras actividades como becarios I+D.

El equipo completo agradece al Concejo Municipal de la Ciudad de Santa Fe por reconocer este proyecto con una declaración de interés, así como a los distintos medios de comunicación que recientemente dieron difusión a la herramienta (Aire de Santa Fe, Cable y Diario, Telefó Santa Fe, Santa Fe Canal, Radio 10 de Buenos Aires, Diario La Nación, entre otros), contribuyendo a que la misma llegue a sus usuarios finales.