

Compilador en línea para apoyar el aprendizaje del lenguaje de programación Java

Online compiler to support the learning of the Java programming language

Carlos R. Jaimez-González

Universidad Autónoma Metropolitana, México

cjaimez@correo.cua.uam.mx

Wulfrano A. Luna-Ramírez

Universidad Autónoma Metropolitana, México

wluna@correo.cua.uam.mx

Resumen

En este artículo se presenta un compilador en línea para apoyar el aprendizaje de los fundamentos del lenguaje de programación Java, el cual fue diseñado para complementar el proceso de enseñanza-aprendizaje de las materias de programación orientada a objetos de las licenciaturas que se imparten en la Universidad Autónoma Metropolitana, Unidad Cuajimalpa.

Se discuten una serie de herramientas que son similares al compilador en línea que se presenta en este artículo, así como también se muestra una tabla comparativa con sus características más relevantes. A partir del análisis comparativo se concluye que ninguna de las herramientas examinadas está diseñada para apoyar en el aprendizaje del lenguaje de programación Java, característica que destaca en el compilador que se presenta en este texto.

El compilador en línea permite a los alumnos introducir y editar código fuente en el lenguaje de programación Java, compilarlo, ejecutarlo y visualizar la salida del programa o la lista de errores que ocurrieron en tiempo de compilación o ejecución; todo ello desde la misma interfaz web del compilador en línea. Los temas, ejemplos y explicaciones de apoyo al aprendizaje que se incluyen en el compilador en línea se agrupan en dos apartados: los que cubren los fundamentos del lenguaje de programación Java y los que explican los principios del paradigma orientado a objetos.

Palabras Clave: Apoyo al Aprendizaje, Compilador en Línea, Lenguaje de Programación Java, Paradigma Orientado a Objetos.

Abstract

This paper presents an online compiler to support the learning of the fundamentals of Java programming language, which was designed to complement the teaching-learning process of undergraduate object oriented programming courses given at the Metropolitan Autonomous University, Campus Cuajimalpa.

A series of tools similar to the online compiler presented in this paper are discussed, as well as a comparative table is shown with their most relevant features. After having carried out the comparative analysis it is concluded that none of the examined tools is designed to support the learning of the Java programming language, which is a feature highlighted in the online compiler presented in this text.

The online compiler allows students to enter and edit source code in the Java programming language, compile it, execute it and visualize the program output or the list of errors that occurred in compilation or execution time; all of them from the online compiler web interface. The topics, examples and explanations to support the learning, which are included in the online compiler are grouped in two sections: those that cover the fundamentals of the Java programming language and those that explain the principles of the object oriented paradigm.

Keywords: Java Programming Language, Learning Support, Object Oriented Paradigm, Online Compiler.

Fecha Recepción: Diciembre 2016

Fecha Aceptación: Julio 2017

1. Introducción

En este artículo se presenta un compilador en línea de apoyo al aprendizaje de los fundamentos del lenguaje de programación Java, el cual fue diseñado para complementar el proceso de enseñanza-aprendizaje de las materias de programación orientada a objetos de las licenciaturas que se imparten en la Universidad Autónoma Metropolitana, Unidad Cuajimalpa (UAM-C). En particular, este compilador en línea fue diseñado para la licenciatura en Tecnologías y sistemas de información (LTSI) (UAM-C, 2017a), en la cual se imparte el programa de estudios con el nombre de Programación orientada a objetos (UAM-C, 2017b).

En esta Unidad de Enseñanza-Aprendizaje (UEA), como se le denominan a las materias en esta universidad, el lenguaje de programación Java es el más utilizado para su impartición, motivo por el cual se decidió utilizarlo para el diseño del compilador en línea que se presenta en este artículo, aunque cabe la posibilidad de extender su funcionalidad a algún otro lenguaje de programación orientado a objetos, tal como C++.

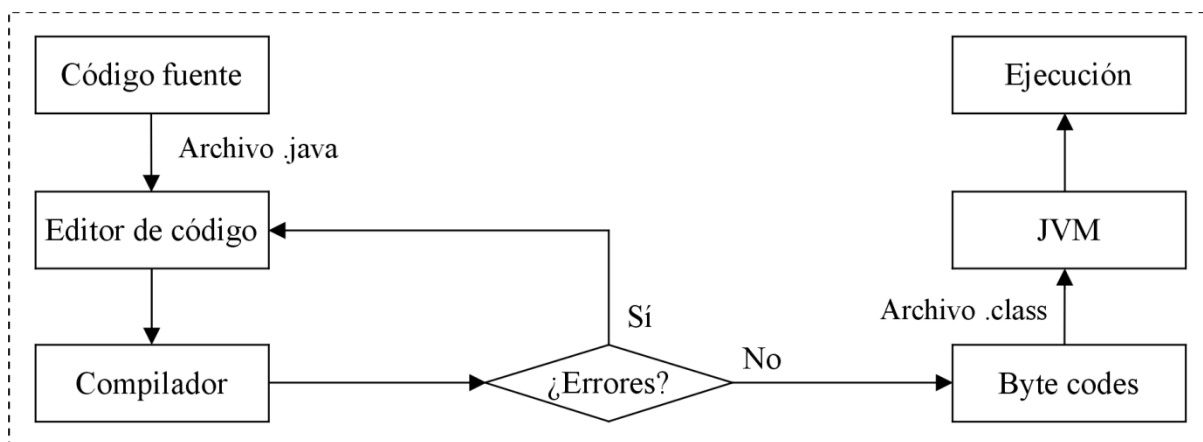
Aunque existen algunos proyectos que se enfocan en apoyar el aprendizaje de la programación en general, algunos de ellos son muy simples en su funcionalidad (DJGPP, 2013), algunos otros son muy costosos (CodePress, 2007; Compilr, 2017), y otros no permiten la posibilidad de trabajar con el lenguaje de programación Java (Codepad, 2017). Existen también algunas herramientas que permiten trabajar con varios archivos fuente al mismo tiempo (Compilr, 2017; DJGPP, 2013), así como compilar y ejecutar los programas desde un ambiente web (CodeMirror, 2017).

En el caso del lenguaje de programación Java, es necesario instalar su ambiente de desarrollo (*Java Development Kit*, JDK por sus siglas en inglés) para que un usuario sea capaz de compilar y ejecutar programas en una computadora. La compilación y ejecución de programas puede realizarse de dos formas: 1) a través de una línea de comandos; y 2) mediante la utilización de un ambiente de desarrollo integrado (*Integrated Development Environment*, IDE por sus siglas en inglés).

En la Figura 1 se ilustra el proceso de edición, compilación y ejecución de un programa escrito en el lenguaje de programación Java. Para el proceso completo se consideran los siguientes pasos: a) la creación de los archivos de código fuente con extensión .java, mediante un editor en el que se escribe el programa; b) el envío del programa para ser compilado; c) la

compilación del programa; d) la generación del archivo con extensión .class resultante de la compilación (en caso de no existir errores de sintaxis); e) la ejecución del programa mediante la máquina virtual de Java (*Java Virtual Machine*, JVM por sus siglas en inglés); y, finalmente, f) la visualización de la salida del programa en la consola de la línea de comandos o del IDE.

Figura 1. Proceso de edición, compilación y ejecución de un programa en el lenguaje Java.



Fuente: Elaboración propia.

El compilador en línea que se presenta en este texto contiene un editor de código en línea para el lenguaje de programación Java, pero con la posibilidad de extenderse a otros lenguajes de programación orientados a objetos. Dentro de las características principales del compilador en línea están: la compilación y ejecución de programas; la visualización de la salida del programa o la lista de errores ocurridos y su lugar de ocurrencia; un ambiente web desde donde se puede interactuar y realizar todo este proceso. De esta forma, es posible practicar la programación orientada a objetos editando, compilando y ejecutando programas desde un mismo lugar, sin la necesidad de navegar a otros sitios, ni tampoco la instalación de herramientas adicionales en la computadora desde donde se utiliza el compilador en línea, ya que lo único necesario para su uso es un navegador web y acceso a Internet.

El resto del artículo se encuentra organizado de la siguiente manera. En la siguiente sección, se describen herramientas existentes que son relevantes al compilador en línea que se presenta en este artículo. Posteriormente, se presenta el compilador en línea propuesto; su arquitectura; el proceso de edición de código fuente, compilación de código fuente y ejecución de

programas; las tecnologías y herramientas de apoyo que se han utilizado para su desarrollo; la ayuda que se implementará en el compilador en línea, así como los temas y ejemplos que serán incluidos. Finalmente, se proporcionan conclusiones y trabajo futuro.

2. Herramientas existentes

Esta sección describe algunas herramientas existentes que están relacionadas con el compilador en línea que se presenta en este artículo, tales como ambientes de desarrollo integrado en línea, compiladores en línea, intérpretes en línea, servicios de línea de comandos, depuradores en línea, servicios de compilación remota, editores de código en línea, entre otros.

Ideone

Es un compilador y depurador disponible en línea (Ideone, 2017), el cual permite compilar y ejecutar código fuente en 40 lenguajes de programación, entre los cuales destacan Java y C++. Ideone contiene un editor de código con resaltado de sintaxis, es multilenguaje, permite la compilación remota, es posible la entrada de datos en tiempo de ejecución, y permite la visualización de la salida del programa. Todas las características y opciones mencionadas son configurables a través del programa, ya sea para determinar el tiempo de ejecución del programa, para especificar si recibe parámetros de entrada, entre otras. Adicionalmente, Ideone proporciona un servicio web para hacer uso de su funcionalidad a través de un programa, es decir, sin utilizar la interfaz proporcionada de manera predeterminada, lo cual permite a desarrolladores experimentados la creación de su propia interfaz con la funcionalidad requerida.

Al ejecutar un programa en Ideone, el código fuente se envía al servidor para su ejecución, lo cual genera una página con tres pestañas: la primera de ellas muestra el código fuente del programa con el resaltado de sintaxis; la segunda pestaña muestra los parámetros de entrada del programa, así como la salida de éste en caso de que no haya errores, ya que si los hubiera, estos serán también visualizados en la pestaña; finalmente, la tercera pestaña muestra algunas notas y comentarios para compartir el código fuente, lo cual es una característica importante de Ideone.

Codepad

Es un compilador e intérprete de código (Codepad, 2017), el cual funciona para diez diferentes lenguajes de programación, entre ellos Python, Ruby y C++; sin embargo, no incluye al lenguaje de programación Java. Codepad contiene un editor de código fuente multilenguaje; permite la compilación remota; y, en cuanto a la ejecución, la salida del programa puede ser visualizada directamente en el compilador. Esta herramienta es sencilla en su interfaz gráfica, ya que únicamente presenta algunos elementos: el editor de texto, la configuración del lenguaje a utilizar y la posibilidad de compilar o ejecutar el programa a través de botones.

Codepad tiene algunas desventajas: no proporciona resaltado de sintaxis del código fuente; ejecuta solamente programas sencillos, ya que no permite trabajar con más de un archivo a la vez; el área del editor es muy limitada en espacio; su funcionalidad está orientada a la depuración de código fuente, por lo que resalta las líneas de código en las que hubo errores al momento de compilar o ejecutar.

Compilr

Es una aplicación web (Compilr, 2017), en la cual es posible crear proyectos en diferentes lenguajes de programación. La versión gratuita de esta aplicación web permite almacenar tres proyectos en la nube, pero mediante una mensualidad es posible incrementar el espacio de almacenamiento y el número de proyectos que se pueden crear. Compilr proporciona una interfaz para navegación de archivos; contiene un editor de código que se caracteriza por el resaltado de sintaxis; tiene un compilador remoto que permite la compilación y carga de bibliotecas remotamente; es posible ejecutar programas con parámetros de entrada; la visualización de la salida se muestra directamente en la aplicación y permite la ejecución de *applets* (Oracle Corporation, 2017). El diseño de la interfaz de Compilr es muy similar a la de algunos IDE comerciales, tales como Eclipse (Eclipse IDE, 2017) y Netbeans (NetBeans IDE, 2017), ya que la distribución de los elementos de pantalla es muy parecida en el menú, editor de texto, explorador de archivos, barras de acceso, entre otros.

En cuanto a la compilación y ejecución de programas, éstas se realizan a través de pestañas, donde en una de ellas se muestran los errores que existen en el código fuente, con una descripción y número de línea. Otra de las pestañas muestra la consola, donde se visualiza la

salida del programa y es posible el paso de parámetros de entrada al inicio y en tiempo de ejecución. La ejecución de *applets* no funciona correctamente, ya que después de haberlo probado con varios ejemplos, siempre finaliza con un error de carga a través de una ventana emergente. Finalmente, cabe señalar que esta aplicación web es capaz de ejecutar programas que no necesariamente tengan interfaz gráfica, sino que la salida sea a través de su consola, además de que es posible compilar y ejecutar programas que contengan más de un archivo de código.

DJGPP Public Access Cross-Compiler

Es un servicio de escritorio (DJGPP, 2013), el cual permite al usuario crear pequeños programas en DOS, aunque no tiene acceso a un compilador. El lenguaje de programación permitido por este servicio es C, el cual además incluye la biblioteca Math. La interfaz gráfica de este servicio solamente contiene un área de texto para introducir el código fuente y dos opciones: una para mostrar alertas de compilación exitosa y otra para optimizar la salida del programa al realizar la compilación.

La compilación se lleva a cabo mediante un botón de Compile, con el cual se genera un archivo binario con extensión .exe, el cual debe descargarse a la computadora para poder ejecutarlo, ya que la ejecución no se lleva a cabo en el programa directamente. El servicio está limitado a la compilación de un solo archivo de código fuente, el cual debe tener un tamaño menor a 1 000 bytes; y cuya ejecución no exceda más de 200 000 bytes de memoria.

JXXX Compiler Service

Es un servicio de compilación remota (JXXX Compiler Service, 2014), el cual permite compilar archivos de código fuente escritos en el lenguaje de programación Java. Esta herramienta permite compilar cualquier tipo de archivo de código fuente Java, incluyendo *applets*, los cuales además pueden ser ejecutados directamente sin necesidad de incrustarlos en una página web para su visualización. Con este servicio es posible la carga de bibliotecas remotamente, la selección de diferentes versiones del JDK, así como la facilidad de compilar hasta cinco archivos fuente al mismo tiempo.

Los usuarios de JXXX Compiler Service principalmente lo utilizan para la compilación y ejecución de *applets*, ya que para estos proporciona una serie de opciones que no se tienen en otras herramientas, tales como las siguientes: la posibilidad de especificar un archivo HTML para visualizar *applets*; la activación de cinco opciones de compilación, entre las que destacan la de optimización y la de generación de tablas de depuración; la posibilidad de seleccionar entre cinco versiones diferentes de JDK; la opción de mostrar errores de compilación en pantalla; la descarga de archivos de código fuente ya compilados, de manera individual o en conjunto.

Online Compiler

Es un sitio web para editar y compilar código fuente (Online Compiler, 2017), el cual se utiliza principalmente para detectar errores de sintaxis o de compilación; cuenta con un editor de texto para el código fuente; es multilinguaje, ya que permite trabajar con varios lenguajes de programación (Basic, Pascal, Java, Fortran, C y C++); permite elegir el sistema operativo (Windows o Linux) para el que se realizará la compilación; la interfaz cuenta con un botón para compilar el código fuente y para redireccionar a una página donde es posible descargar el archivo compilado o ejecutable, dependiendo del sistema operativo elegido. La principal desventaja de esta herramienta es que no muestra la salida resultante de los programas, sin embargo, es posible habilitar que muestre los errores en tiempo de compilación.

CodePress

Es un editor de código fuente en línea (CodePress, 2007), el cual fue desarrollado en el lenguaje JavaScript; permite el resaltado de sintaxis conforme se escribe el texto en el navegador web; cuenta con la función para autocompletar texto; es posible tener múltiples instancias de la herramienta; permite la escritura de código fuente en varios lenguajes de programación. A diferencia de las herramientas presentadas anteriormente, en las cuales se tienen funciones de compilación y ejecución de programas, CodePress es únicamente un editor de código fuente, cuyo principal objetivo es que pueda ser incrustado en una aplicación web, a través de un archivo JavaScript para proporcionar las funcionalidades mencionadas.

EditArea

Es un editor de código fuente (EditArea, 2010), el cual fue desarrollado en el lenguaje JavaScript; para la edición de código fuente utiliza un área de texto; tiene resaltado de sintaxis; genera código fuente bien formateado e indentado; realiza búsqueda y reemplazo con expresiones regulares; además de que permite tener múltiples instancias de la herramienta al mismo tiempo.

CodeMirror

Es un editor de código fuente (CodeMirror, 2017), el cual también fue escrito en JavaScript y puede ser utilizado en un navegador web, además de tener la posibilidad de incrustarlo en una página web; tiene resaltado de sintaxis; generación de código fuente bien formateado; búsqueda y reemplazo; autoindentación; editor de tamaño automático, el cual se ajusta al tamaño de las líneas de código para una mejor visualización de los archivos fuente.

Tabla comparativa

La Tabla 1 muestra una comparación de once características de las herramientas existentes descritas en las secciones previas. Las herramientas que se consideraron para esta comparación son las siguientes: H1) Compilr, H2) CodePad, H3) DJGPP, H4) Ideone, H5) JXXX Compiler Service, H6) OnlineCompiler. Las características que se observaron se listan a continuación: 1) Editor de texto, 2) Resaltado de sintaxis, 3) Multilenguaje, 4) Soporte para más de un archivo fuente, 5) Carga de bibliotecas, 6) Compilación remota, 7) Datos de entrada, 8) Visualización de salida del programa, 9) Visualización de *applets*, 10) Gratuito, 11) Apoyo para el aprendizaje. Se muestra “Sí” en caso de que la herramienta tenga la característica, o “No” en caso de que la herramienta no la tenga.

Tabla 1. Tabla comparativa de las herramientas analizadas.

Características	H1	H2	H3	H4	H5	H6
Editor de texto	Sí	Sí	Sí	Sí	No	Sí
Resaltado de sintaxis	Sí	Sí	No	Sí	No	No
Multilinguaje	Sí	Sí	No	Sí	No	Sí
Soporte para más de un archivo fuente	Sí	No	No	No	Sí	No
Carga de bibliotecas	Sí	No	No	No	Sí	No
Compilación remota	Sí	Sí	Sí	Sí	Sí	Sí
Datos de entrada	Sí	No	No	Sí	No	No
Visualización de salida del programa	Sí	Sí	No	Sí	Sí	Sí
Visualización de <i>applets</i>	Sí	No	No	No	Sí	No
Gratuito	No	Sí	Sí	Sí	Sí	Sí
Apoyo para el aprendizaje	No	No	No	No	No	No

Fuente: Elaboración propia con información de las herramientas analizadas.

El compilador en línea propuesto en este artículo pretende tener todas las características mostradas en la Tabla 1. Muchas de las herramientas analizadas son muy completas, como puede observarse en la tabla comparativa y en las descripciones proporcionadas previamente. Sin embargo, ninguna de ellas satisface la característica de apoyo al aprendizaje, la cual debe de contemplar el editor de código como pieza fundamental, ya que, a través de éste, es que los usuarios escriben código fuente, razón por la cual la implementación del editor contempla funciones para la programación, tales como el resaltado de sintaxis. Adicionalmente, otra parte fundamental para el compilador en línea es propiamente el compilador de código fuente, el cual permite convertir el código fuente en código ejecutable, con lo cual es posible ejecutar un programa y visualizar su salida. Finalmente, la ayuda al alumno como apoyo al aprendizaje es otra de las características esenciales en el compilador en línea, la cual comprende la visualización de errores de compilación y su mapeo a las líneas del código correspondientes en el programa.

El análisis realizado proporciona un panorama de las herramientas existentes y los problemas que cada una resuelve de manera parcial, con lo cual puede observarse que no hay una herramienta o ambiente que conjunte las características deseables. La siguiente sección presenta

el compilador en línea para apoyar el aprendizaje de los fundamentos del lenguaje de programación Java.

3. Compilador en línea propuesto

En esta sección se presenta el compilador en línea propuesto, su arquitectura, las tecnologías y herramientas de apoyo que se están utilizando para su implementación (edición de código, compilación de código y ejecución de programas), así como la sección de ayuda que será incorporada como parte del compilador en línea.

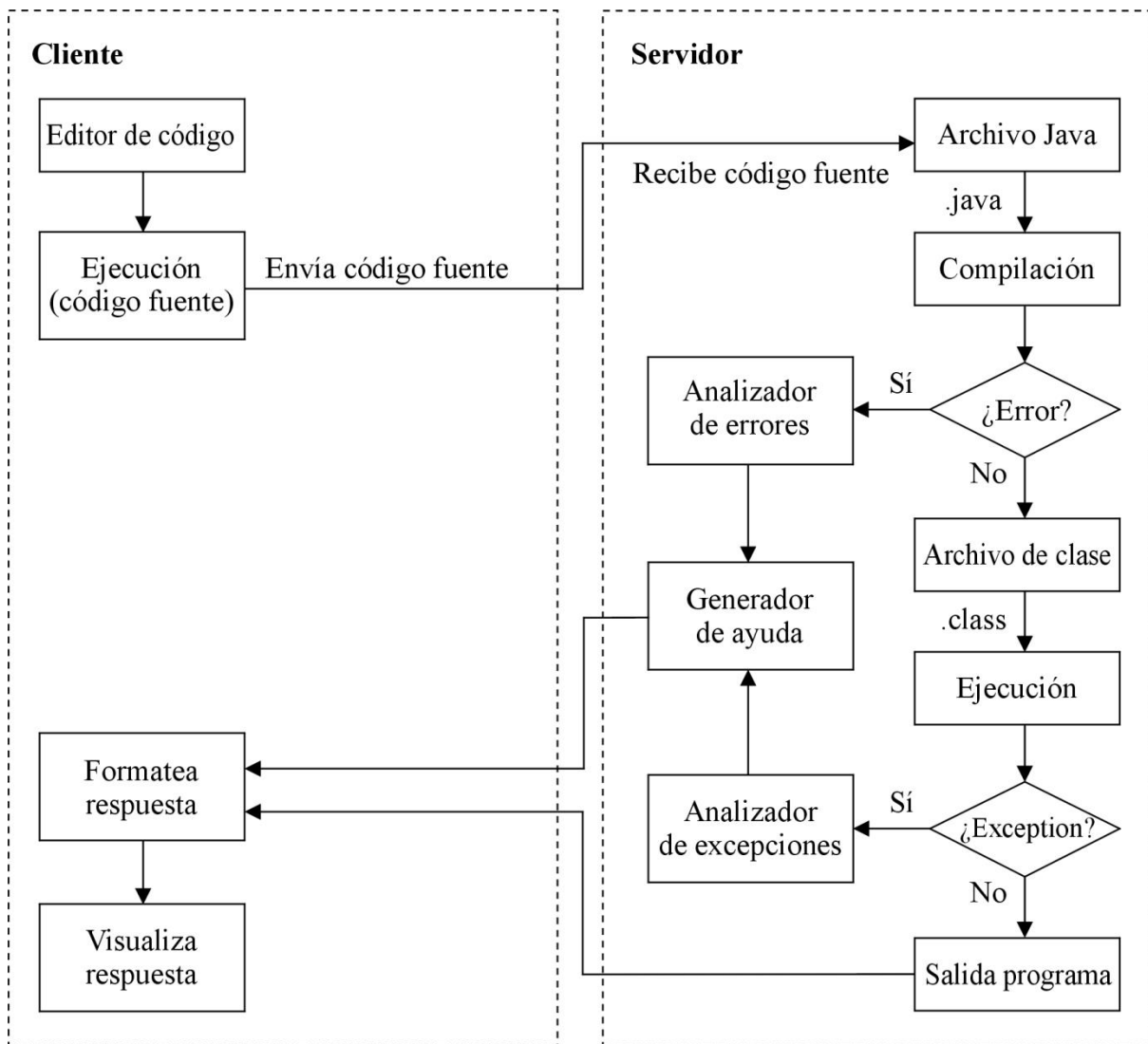
Arquitectura del compilador en línea

El compilador en línea utiliza una arquitectura cliente-servidor, dado que es una herramienta que hace uso del paradigma web. Del lado del servidor, se tiene instalado el JDK para permitir la compilación del código fuente, la ejecución de programas, la captura de las salidas de programas, así como la recepción de errores o excepciones. Del lado del cliente, se tiene el editor de código fuente, donde el usuario introduce el código fuente del programa y mediante un botón se envía al servidor para su compilación y ejecución. El proceso completo entre cliente y servidor se ilustra en la Figura 2.

Tecnologías y herramientas de apoyo al compilador en línea

Para la implementación del compilador en línea se han utilizado diversas tecnologías y herramientas de apoyo, entre las que destacan las siguientes: Apache Tomcat como servidor web, contenedor de *servlets* y páginas de servidor Java (*Java Server Pages*, JSP por sus siglas en inglés); archivos JSP, los cuales se utilizan para la generación de contenido dinámico y consultas a fuentes de datos; Java como lenguaje de programación del lado del servidor; la interfaz *JavaCompiler* del JDK, la cual es utilizada para la funcionalidad de compilación de archivos de código fuente; HTML para la estructura y contenido de la interfaz del compilador en línea; hojas de estilo CSS para la presentación del contenido del compilador en línea; y JavaScript como el lenguaje para realizar páginas web dinámicas e interactivas, además de proporcionar comunicación asíncrona con el servidor.

Figura 2. Arquitectura cliente-servidor: edición, compilación y ejecución de programas en Java.



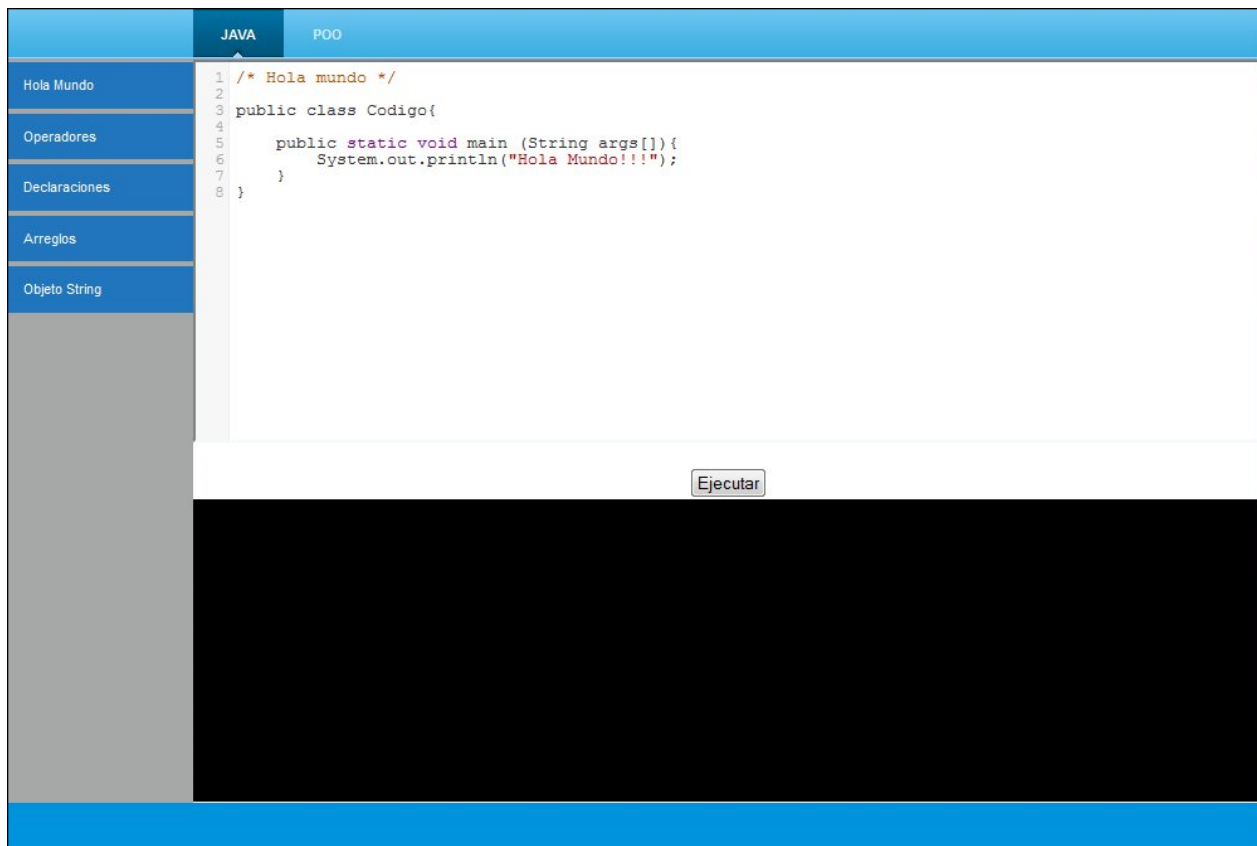
Fuente: Elaboración propia.

Edición en el compilador en línea

Para implementar la funcionalidad de edición de código fuente en el compilador en línea se eligió la herramienta CodeMirror por ser el editor más completo; esto después de haber analizado tres editores de código fuente (CodeMirror, 2017; CodePress, 2007; EditArea, 2010), como se mencionó en una sección previa. Por el momento, la funcionalidad implementada en el compilador en línea es el resaltado de sintaxis, el autoindentado y la tabulación, pero está planeado incorporar en una siguiente versión el autocompletado y la previsualización en HTML.

La Figura 3 muestra la versión preliminar de la interfaz del compilador en línea, en la cual pueden observarse tres paneles o secciones: en el panel del lado izquierdo se muestra un menú, en el cual se incluyen algunos ejemplos del temario que se presenta en una sección posterior de este artículo; la sección del lado derecho de la interfaz está dividida en dos, el panel superior contiene el editor de código fuente, mientras que el inferior contiene la consola de compilación y ejecución de programas, donde también se muestra la salida del programa. La Figura 3 muestra una captura de pantalla del editor de código fuente del compilador en línea con una clase de ejemplo, la cual puede ser compilada y ejecutada al presionar el botón Ejecutar.

Figura 3. Interfaz del compilador en línea con ejemplos, editor de código y consola.



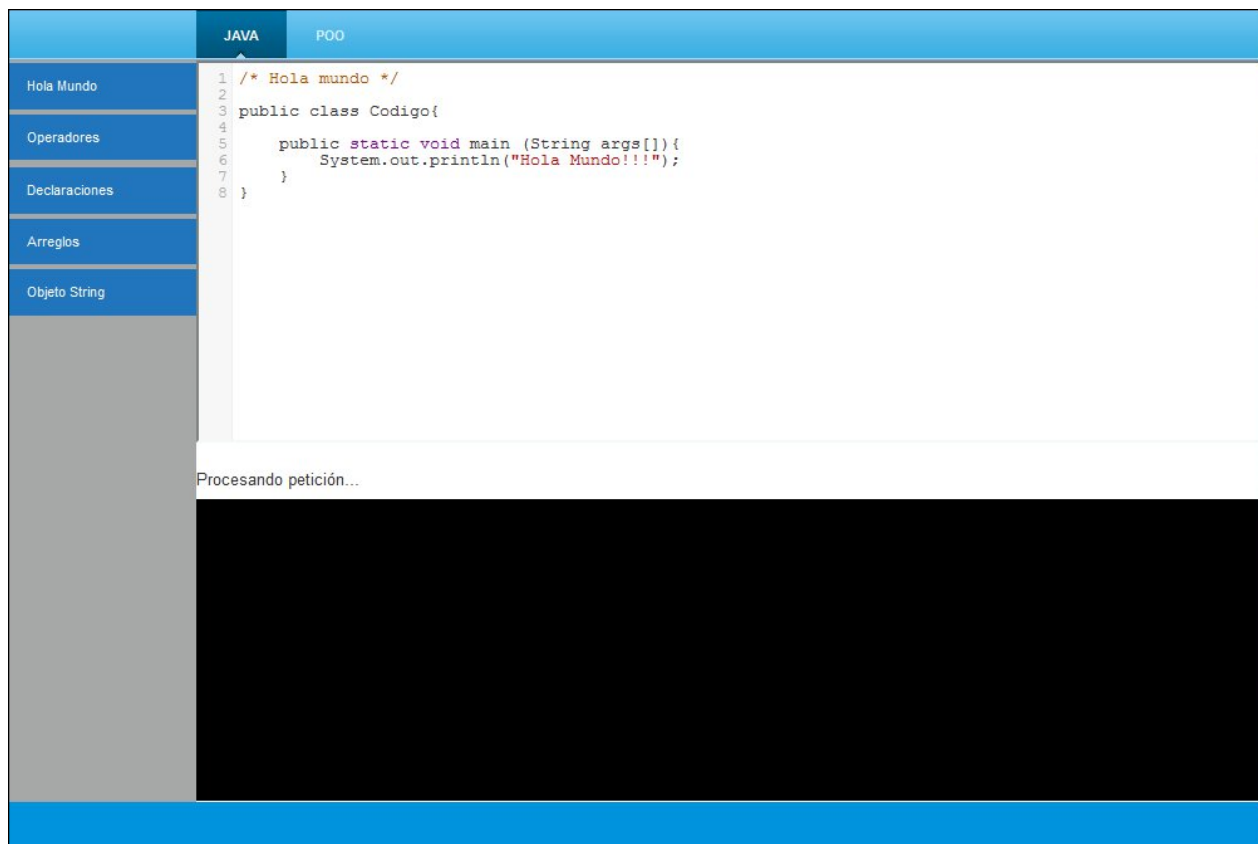
Fuente: Elaboración propia.

Compilación y ejecución en el compilador en línea

Para implementar la compilación de código fuente y la ejecución de programas fue necesario hacer uso de la interfaz JavaCompiler del JDK del lado del servidor, la cual proporciona métodos para realizar la compilación del código escrito en el editor web, así como métodos para invocar programas. Adicionalmente, la interfaz JavaCompiler es capaz de recolectar todos los errores generados durante la compilación del código fuente y la ejecución de programas.

En la Figura 4 se muestra una captura de pantalla del compilador en línea, justamente después de que el usuario ha presionado el botón Ejecutar, con lo cual se manda una petición al servidor para que éste compile el código fuente y lo ejecute. Mientras se compila y ejecuta el código fuente en el servidor, el compilador en línea muestra el mensaje "Procesando petición...", como se observa en la Figura 4.

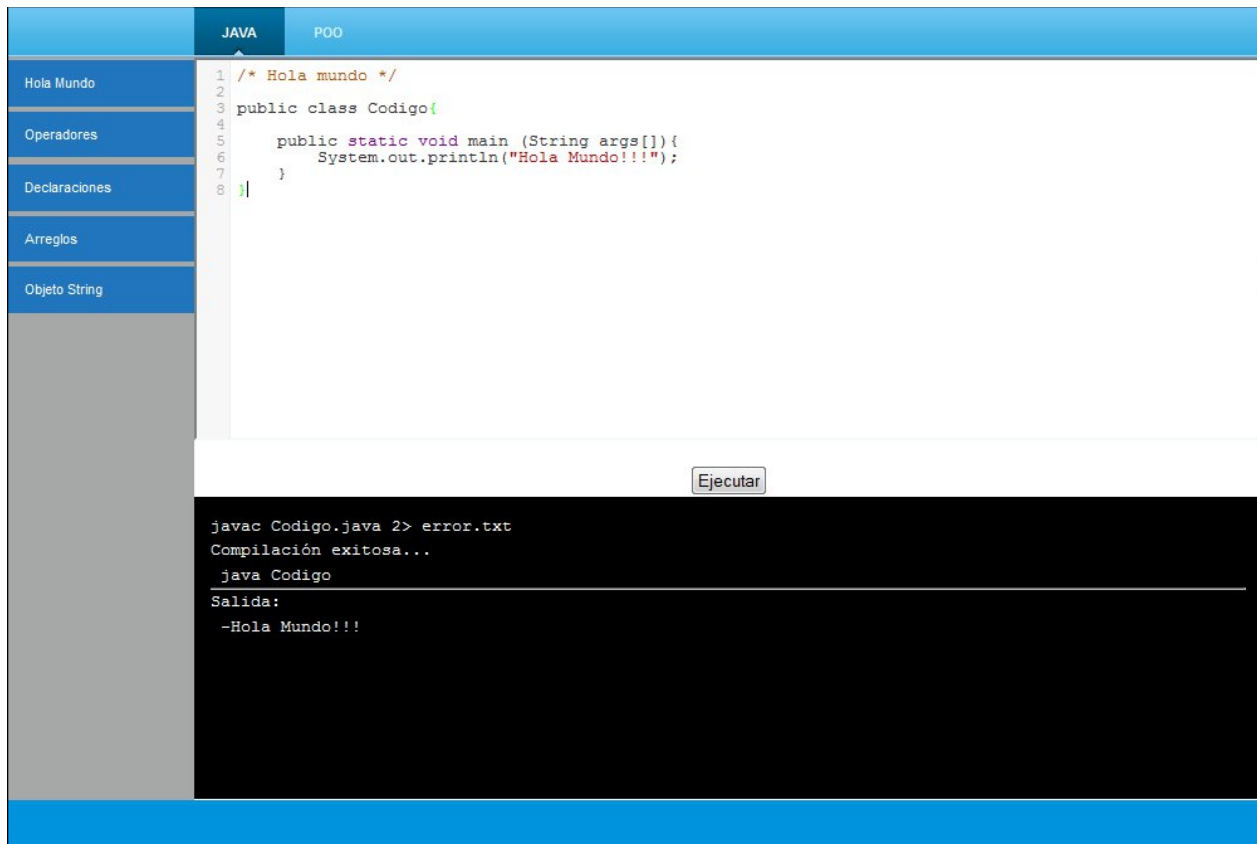
Figura 4. Interfaz del compilador en línea al presionar el botón *Ejecutar*.



Fuente: Elaboración propia.

Dado que el código fuente que se introdujo en el editor de código no contiene errores, el servidor es capaz de compilarlo y ejecutarlo, generando una respuesta de parte del servidor con la salida del programa, la cual es enviada al navegador web y se muestra en la consola del compilador en línea, como se ilustra en la captura de pantalla de la Figura 5.

Figura 5. Interfaz del compilador en línea mostrando la salida del programa.



The screenshot shows a web-based Java IDE. On the left is a navigation menu with items: 'Hola Mundo', 'Operadores', 'Declaraciones', 'Arreglos', and 'Objeto String'. The main editor area contains the following Java code:

```

1  /* Hola mundo */
2
3  public classCodigo{
4
5      public static void main (String args[]){
6          System.out.println("Hola Mundo!!!");
7      }
8  }

```

Below the code editor is an 'Ejecutar' button. The output console at the bottom shows the following text:

```

javacCodigo.java 2> error.txt
Compilación exitosa...
javaCodigo
Salida:
-Hola Mundo!!!

```

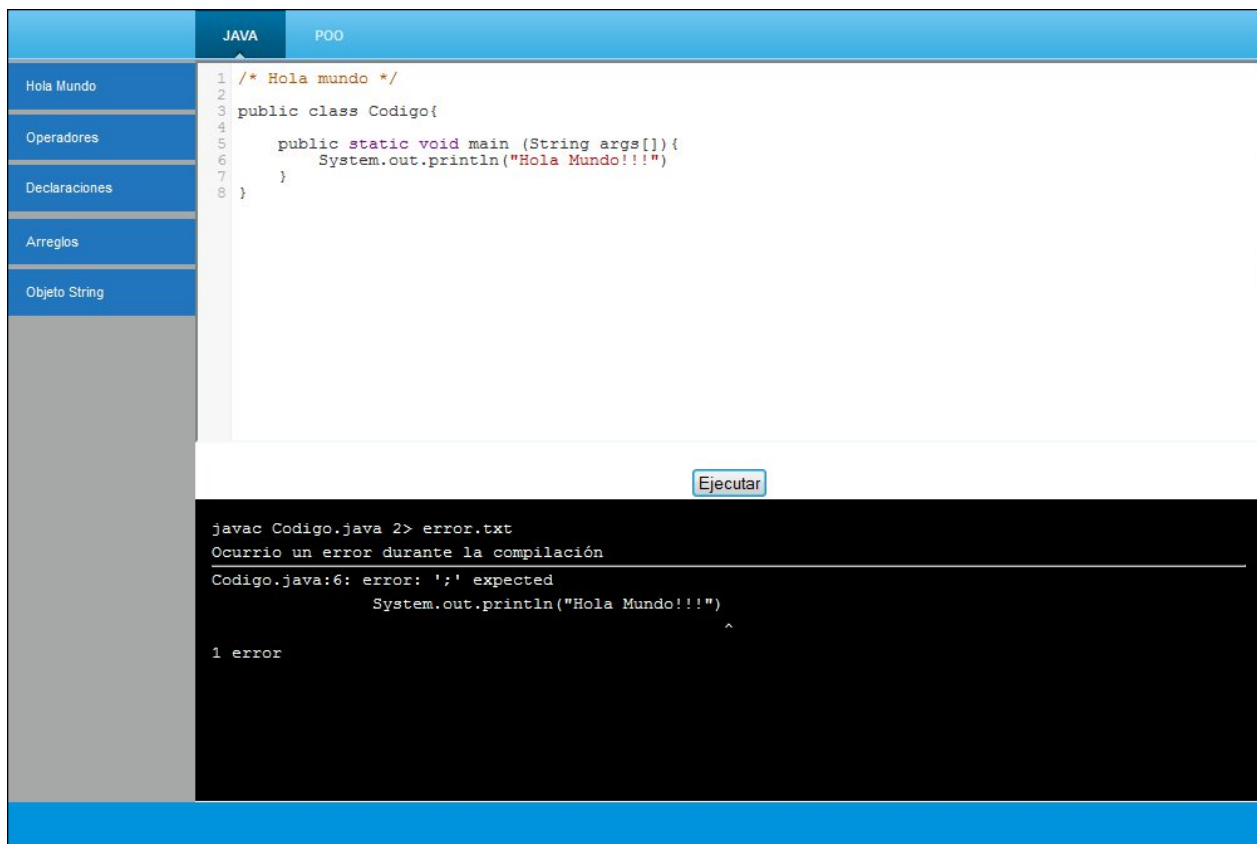
Fuente: Elaboración propia.

En caso de que el código fuente tuviera errores al enviarlo al servidor, éste no sería capaz de compilarlo ni ejecutarlo, por lo que generaría una lista con los errores que encontró, la cual sería enviada de regreso al navegador web para que éste a su vez la muestre en la consola del compilador en línea.

La Figura 6 muestra una captura de pantalla del compilador en línea, en donde la clase que se encuentra en el editor de código contiene un error de sintaxis en la línea 6, falta un ";" al final

de la sentencia. Al enviar este código fuente al servidor mediante el botón Ejecutar, éste intenta compilarlo, pero dado que hay un error se dispara un mensaje, el cual es enviado al navegador web y desplegado en la consola que se encuentra en el panel inferior del compilador en línea, como se muestra en la captura de pantalla de la Figura 6.

Figura 6. Interfaz del compilador en línea con error en el código fuente.



Fuente: Elaboración propia.

Ayuda en el compilador en línea

El compilador en línea proporcionará información acerca de los errores y excepciones ocurridos en tiempo de compilación del código fuente o en tiempo de ejecución de los programas. La funcionalidad para proveer la ayuda al alumno que utiliza el compilador en línea aún no está implementada; sin embargo, se tiene planeado que será proporcionada a través de un menú, en el cual habrá acceso a una serie de ejemplos con explicaciones breves, los cuales servirán de apoyo

a los alumnos para familiarizarse con el lenguaje de programación Java y el paradigma orientado a objetos. Los ejemplos que serán incorporados en la ayuda serán los correspondientes a los temas mencionados en una siguiente sección de este artículo, los cuales están inspirados en bibliografía sobre la enseñanza de Java y del paradigma orientado a objetos.

El área en la que se encuentra el editor de código será la que visualizará el material de ayuda más importante, ya que sus características son ideales para tal efecto: 1) edición de código con resaltado de sintaxis; 2) compilación y ejecución en línea a través del botón Ejecutar; 3) visualización de errores de compilación o ejecución en la interfaz del compilador en línea, con la posibilidad de consultar un ejemplo relacionado con el tipo de error; 4) relación de errores y excepciones directamente en el código fuente, mediante el resaltado de la línea donde ocurrió el error; y 5) visualización de *applets*.

Manejo de usuarios en el compilador en línea

El manejo de usuarios en el compilador en línea permitirá a los usuarios registrados el almacenamiento de código fuente y los programas generados dentro del compilador en línea, con el objetivo de acceder a ellos en ocasiones posteriores. También será posible guardar diferentes versiones de un mismo código fuente o programa compilado. Dado que el manejo de usuarios no es fundamental para el funcionamiento del compilador en línea, esta característica será implementada en una versión posterior.

Temario para el compilador en línea

Se ha realizado una revisión de material orientado al aprendizaje del paradigma orientado a objetos con el lenguaje de programación Java, debido a que el compilador en línea propuesto está ideado como una herramienta de apoyo al aprendizaje. El objetivo de la revisión ayudó a determinar los temas y contenidos que son fundamentales, con lo cual es posible establecer el tipo de ejemplos que deben incluirse en el compilador en línea, así como la información que será mostrada junto con los ejemplos correspondientes. La consulta se basó principalmente en tres libros de texto (Arnold, Gosling y Holmes, 2005; Eckel, 2006; Flanagan, 2005), en cuyos primeros capítulos se abordan los mismos temas en un orden muy similar. La revisión de los índices temáticos determinó los contenidos y definiciones que serán incluidos en el compilador

en línea, así como el tipo de ejemplos sencillos que proporcionen práctica en la programación y su capacidad de integración con las explicaciones breves que se planean incorporar. El temario que se cubrirá en el compilador en línea se dividió en dos secciones: por un lado, los fundamentos del lenguaje de programación Java y, por otro, los principios del paradigma orientado a objetos.

Temas para los fundamentos del lenguaje de programación Java:

- Operadores aritméticos, de concatenación, de comparación y booleanos.
- Declaraciones de variables, constantes e inicialización.
- Estructuras de control de flujo de programa condicionales (*if/else*), de iteración (*for*, *while* y *do while*), de selección (*switch/case*) y otros (*break* y *continue*).
- Arreglos de una o varias dimensiones, de diferentes tipos de datos.
- Clase *String* (atributos y métodos).
- Excepciones.

Temas para los principios del paradigma orientado a objetos:

- Clases (atributos, métodos y modificadores).
- Objetos (constructores e inicialización).
- Herencia.
- Encapsulamiento.
- Polimorfismo.

En la primera versión del compilador en línea solamente se procesan programas que no requieren el uso de archivos de entrada/salida y aquellos que no tengan interacción con el alumno durante su ejecución. Todos los programas servirán de ejemplo para uno o más de los temas indicados anteriormente. Estas restricciones se aplicarán también para aquellos programas que los alumnos deseen ejecutar en el compilador en línea, que no sean propiamente los programas que estarán de ejemplo.

Seguridad en el compilador en línea

En cuanto a la seguridad del servidor web donde reside el compilador en línea, se incluirán medidas para garantizarla, tales como restricciones en el tiempo de ejecución de los programas, uso de memoria limitado para compilación y ejecución de programas. Todo esto es debido a que

el objetivo del compilador en línea es proporcionar al alumno programas sencillos para apoyar su aprendizaje del lenguaje de programación Java, y no para ser un servidor en el que puedan ejecutarse programas que demanden muchos recursos.

4. Conclusiones y trabajo futuro

En este artículo se presentó un compilador en línea para apoyar el aprendizaje de los fundamentos del lenguaje de programación Java, el cual está compuesto de un editor de código en línea, inicialmente para el lenguaje de programación Java, pero puede extenderse a otros lenguajes de programación. El compilador en línea se encuentra en desarrollo, pero ya se cuenta con una versión preliminar, la cual permite al alumno introducir código fuente en el lenguaje de programación Java, compilarlo, ejecutarlo y visualizar la salida del programa o la lista de errores que ocurrieron en tiempo de compilación o ejecución desde la misma interfaz web del compilador en línea.

Se realizó un análisis de herramientas que son similares al compilador en línea que se presentó en este artículo, proporcionando una descripción de cada una de ellas y mostrando una tabla comparativa de sus características más relevantes. A partir del análisis comparativo pudo concluirse que las herramientas analizadas son muy completas, sin embargo, ninguna de ellas está diseñada para apoyar el aprendizaje del lenguaje de programación Java, característica que destaca en el compilador en línea que se está desarrollando. Los temas, ejemplos y explicaciones que serán incluidos en el compilador en línea se dividieron en dos secciones: los que cubrirán los fundamentos del lenguaje de programación Java y los que explicarán los principios del paradigma orientado a objetos.

Como trabajo futuro se continuará con el desarrollo del compilador en línea, el cual incluye la incorporación de los temas y ejemplos propuestos de acuerdo con el temario presentado. También se desarrollará la funcionalidad de ayuda a los alumnos, la cual mostrará el lugar exacto del código fuente donde ocurrió un error, así como las sugerencias de ejemplos correspondientes para reforzar sus conocimientos de un tema en particular. Una vez concluido el compilador en línea, será necesario realizar pruebas de funcionalidad con alumnos de la materia de programación orientada a objetos, con el objetivo de determinar la efectividad del compilador en línea en el proceso de enseñanza-aprendizaje. Finalmente, se creará un instrumento de evaluación

para valorar diversos aspectos, tales como el diseño de interfaz, la utilidad y las características didácticas del compilador en línea.

Bibliografía

- Arnold, K., Gosling, J., Holmes, D. (2005). *The Java Programming Language* (4^a Ed.). Michigan: Addison-Wesley Professional.
- CodeMirror (versión 5.30.0) [Software] (2017). Berlín, Alemania. Recuperado de <http://codemirror.net/>.
- Codepad [Software] (2017). San Francisco, CA: Sauce Labs. Recuperado de <http://codepad.org/>.
- CodePress (versión 0.9.6) [Software] (2007). Recuperado de <http://codepress.sourceforge.net/>.
- Compilr [Software] (2017). Carpinteria, CA: Lynda.com. Recuperado de <https://index.co/company/Compilr/>.
- DJGPP (versión 2.0) [Software] (2013). Deerfield, NH: Delorie Software. Recuperado de <http://www.delorie.com/djgpp/compile/>.
- Eckel, B. (2006). *Thinking in Java* (4^a Ed.). Massachusetts: Pearson Education, Inc.
- Eclipse IDE [Software] (2017). Ontario, Canada: The Eclipse Foundation. Recuperado de <https://www.eclipse.org/ide/>.
- EditArea (versión 0.8.2) [Software] (2010). Recuperado de <http://www.cdolivet.com/editarea/>.
- Flanagan, D. (2005). *Java in a Nutshell: A Desktop Quick Reference* (5^a Ed.). Sebastopol, CA: O'Reilly Media.
- Ideone (versión sun-jdk-1.8.0_51) [Software] (2017). Gdynia, Poland: Sphere Research Labs. Recuperado de <http://ideone.com/>.
- JXXX Compiler Service [Software] (2014). Suiza: Innovation. Recuperado de http://www.innovation.ch/java/java_compile.html.
- NetBeans IDE (versión 8.2) [Software] (2017). Redwood Shores, CA: Oracle Corporation. Recuperado de <https://netbeans.org/>.
- Online Compiler [Software] (2017). Recuperado de <http://www.onlinecompiler.net/>.

Oracle Corporation (2017). *The Java Tutorials. Lesson: Java Applets*. Redwood Shores, CA: Oracle Corporation. Recuperado de <http://www.oracle.com/technetwork/java/applets-137637.html>.

Universidad Autónoma Metropolitana, Unidad Cuajimalpa (UAM-C) (2017a). Plan de Estudios de la Licenciatura en Tecnologías y Sistemas de Información. México: UAM Cuajimalpa. Recuperado de <http://hermes.cua.uam.mx/archivos/PlandeEstudioTSI.pdf>.

Universidad Autónoma Metropolitana, Unidad Cuajimalpa (UAM-C) (2017b). Programa de la Unidad de Enseñanza-Aprendizaje Programación Orientada a Objetos, de la Licenciatura en Tecnologías y Sistemas de Información. México: UAM Cuajimalpa. Recuperado de <http://dccd.cua.uam.mx/archivos/PDFprogramas/tecnologias/460006.pdf>.