

UAIDesign: Herramienta Colaborativa Multiplataforma para la enseñanza de Diseño de Algoritmos y Programación Estructurada

Walter Salcedo; Christian Müller; Manuel Otero

Universidad Abierta Interamericana

CAETI – Centro de Altos Estudios en Tecnología Informática

Buenos Aires, Argentina.

wsalcedo@fibertel.com.ar; christian.muller197@gmail.com; manuel.otero@line64.com

Profesor Tutor: Ing. Nicolás Battaglia

II Workshop sobre Innovación en Centros Educativos y de Investigación (II WICEI) Innovación en Centros Educativos y de Investigación

Innovación en e-learning, b-learning, m-learning

Resumen

A pesar del paso de los años, los cursos iniciales de materias de programación en la enseñanza universitaria siguen teniendo como herramienta primordial a la programación estructurada como vía de incorporación de las construcciones fundamentales de la programación en los futuros ingenieros. En este trabajo se promueve la integración mental del diseño lógico de una solución a una problemática planteada por sobre el ser experto en un lenguaje de programación, ya que la programación es aplicación de lógica y debe ser independiente de cualquier lenguaje que se utilice. Teniendo en cuenta también que muchos estudiantes se encontraran por primera vez con el planteo de resolver problemas mediante algoritmos los cuales utilizan una determinada simbología de diagramación y que a su vez los mismos, con toda certeza, cuenten con dispositivos móviles, surge entonces la oportunidad de poder apoyar su formación inicial con herramientas colaborativas multiplataforma, las cuales no suplantarán al docente, sino que serán de gran apoyo para el mismo y sus alumnos. Las herramientas colaborativas se suman como un valor agregado para el aprendizaje de la programación estructurada creando el ambiente adecuado tanto desde el aula como desde cualquier lugar no fijo a través de dispositivos móviles.

Palabras clave: Programación Estructurada, Enseñanza y Aprendizaje Colaborativo, Aprendizaje Móvil.

Introducción

Aprender a programar es un proceso que para el alumno que se inicia en el mismo, requerirá adquirir una serie de habilidades específicas.

También se deberán tener en cuenta ciertos factores que se tendrán que minimizar para poder lograr este objetivo, estos factores

siempre están presentes y seguramente algunos de ellos sean imposibles de eliminar de forma total, tales como falta de tiempo para desarrollo de prácticas supervisadas por el docente, el miedo propio de la participación en clase y la desmotivación del alumno por no

poder sobrellevar el ritmo de aprendizaje del curso [1].

La importancia sobre el aprendizaje de diagramas estructurados, dará a entender al alumno los principios de la programación, es decir, sobre cómo resolver situaciones planteadas, generando algoritmos representados en un diagrama.

Este aprendizaje se potencia aún más al utilizar una herramienta multiplataforma como *UAICase* [2], donde el alumno puede diagramar en una plataforma de modelado UML sus propias soluciones a ejercicios planteados por el docente, fomentando el trabajo en equipo y aprovechando la intensa comunicación que nos presta esta herramienta CASE [2] [3].

El paradigma estructurado suele tomar una importancia significativa para todo ingresante de una carrera de ciencias informáticas, ya que suele ser el primer contacto con el arte de la programación, muestra sus conceptos básicos tales como la secuencialidad, la iteración y la decisión, aumentando en forma paralela las capacidades lógicas para resolver situaciones planteadas, demostrándole así al alumno que el algoritmo y posteriormente el código en un lenguaje formal es la expresión de su conocimiento.

La utilización de sistemas de visualización resulta de suma importancia ya que permite al alumno tener una imagen, en el estricto sentido de la palabra, de la forma en que se comporta su algoritmo, detectar errores, corregirlos y así aprender. De hecho, en programación como en otros campos, un error detectado, supone una oportunidad de aprender, por lo que resulta fundamental ofrecer a los alumnos esta posibilidad.

Estamos convencidos que las herramientas de visualización, animación y simulación de algoritmos, facilitan la tarea de aprendizaje. Estos avances en la tecnología crean nuevos contextos de comunicación e interacción entre usuarios, generando nuevos conjuntos de herramientas que reemplazan a los ya existentes que carecen de estas cualidades.

La capacidad de ubicuidad que genera este avance tecnológico, permite disponer de contenidos a lo largo del día, en cualquier momento y en cualquier lugar; al mismo tiempo que genera la necesidad de disponer de nuevas infraestructuras tecnológicas.

Se estimó que sólo hacia finales de 2012 había en el mundo alrededor de seis mil millones de suscripciones de telefonía móvil. El incremento exponencial del número de teléfonos móviles y similares, tanto en países desarrollados como en los países en desarrollo, genera posibilidades equitativas para aumentar el acceso y la calidad en la enseñanza de la programación [4]. El aprendizaje mediante dispositivos móviles, como teléfonos, notebooks o tablets es una modalidad de rápido crecimiento entre las aplicaciones de las TIC en la educación, y por ende un nicho de posibilidades a explorar para personalizar la enseñanza de la programación en aquellos cursos universitarios iniciales de la misma.

UAI Design propone un curso totalmente organizado, presentando todos los requisitos de una herramienta colaborativa, enfocándose en los conocimientos de la programación estructurada y la representación de algoritmos para la resolución de problemas adaptados a la realidad.

Desarrollo

Programación Estructurada.

Estado del Arte y propuesta de apoyo a la enseñanza.

La *programación estructurada* emerge a mediados de los sesenta como evolución al diagrama de flujo existente hasta ese momento para la presentación de algoritmos que dan resolución a problemas planteados en programación.

El mismo se basa en un teorema base llamado el *teorema de la programación estructurada* o de Corrado Bohm y Giuseppe Jacopini (1966) el cual en esencia deja sentado que cualquier problema de programación puede ser resuelto utilizando sólo tres estructuras de control: 1) *Estructura Secuencial o Acción Simple*, 2) *Estructura de Decisión o Acción Condicional*, 3) *Estructura Iterativa o Acción de Repetición*.

A estas estructuras se les llama de control debido a que efectivamente controlan la ejecución de un algoritmo [5].

Sumadas a estas estructuras base aparecen también características propias que introduce esta nueva manera de elaborar resoluciones de problemas mediante algoritmos estructurados como ser:

- *Un único punto de entrada.*
- *Un único punto de salida.*
- *Concepto de módulo.*
- *Concepto de funciones.*
- *Concepto de parámetro.*

Resulta útil también mostrar la definición de ciertos conceptos que a diario nos resultan familiares pero que quizás a veces no nos

detenemos a elaborar y comprender el concepto de los mismos.

En primer lugar, llamamos *algoritmo* a un conjunto finito y ordenado de acciones con las que podemos resolver un determinado problema que se nos plantea, podemos también enmarcar que entendemos por *problema*, el mismo referencia una situación que se nos presenta y que mediante la aplicación de un *algoritmo* nos disponemos a resolver [6].

Para presentar la definición del siguiente concepto de interés, se está de acuerdo que nos importa adquirir destrezas en la elaboración de algoritmos para resolver problemas apoyándonos en el uso de computadoras, una computadora sólo comprenderá lenguajes formales de programación, aquí podemos decir que estamos *codificando* las acciones de un algoritmo cuando escribimos las mismas en un lenguaje formal de programación [6].

A pesar de los años que han pasado desde la enunciación del *teorema de la programación estructurada*, la misma se sigue utilizando masivamente en los cursos iniciales de programación universitaria como medio enseñanza de la programación por su sencillez al permitir dar predominancia al desarrollar la lógica necesaria para dar resolución a problemas en forma algorítmica, independientemente de algún lenguaje forma.

Para apoyar y sumar valor agregado a la enseñanza de la programación estructurada es que se propone una forma de modalidad mixta de enseñanza y aprendizaje basada en herramientas colaborativas multiplataforma, en donde parte se desarrolla en el aula y la parte restante en línea [2], fundamentalmente

promoviendo el uso de dispositivos móviles para el aprendizaje y enseñanza.

Elección de la Programación Estructurada sobre la Programación Orientada a Objetos en cursos iniciales de programación.

Hablar de la enseñanza inicial de la programación mediante métodos de la *programación estructurada* en tiempos actuales en donde la programación orientada a objetos tiene tanta preponderancia, hasta parecía totalmente inadecuado y antiguo.

Incluso ya a inicios de los años 2000, un informe curricular de ACM/IEEE [7] incluía la POO dentro del núcleo de conocimientos básicos que un estudiante de ingeniería de software debía poseer.

Sin embargo, en una etapa inicial de aprendizaje de programación, se debería orientar a los alumnos en dominar de manera efectiva las estructuras básicas de la programación estructurada, *secuencialidad, decisión e iteración*. Si a esto en tal situación inicial se le sumase el hecho de tener que cruzarse con conceptos como objetos, clases, métodos y mensajes, se complicaría la enseñanza inicial [8].

Consideremos a modo de ejemplo un sencillo programa escrito en *Java*, hoy día el lenguaje más utilizado a nivel mundial [9], en el que sólo se muestra el tradicional "Hola Mundo!!!", característico del inicio de aprendizaje de cualquier lenguaje formal [10]:

```
public class HolaMundo
{
    public static void main(String[] args)
    {
        System.out.println("Hola Mundo !!!");
    }
}
```

Sin pretender entrar en mayores detalles, puede observarse que se utilizan conceptos como *clase* y *métodos*, lo cual a nuestro entender sería algo que se podría evitar de incurrir y sería suficiente con utilizar un lenguaje no orientado a objetos como por ejemplo C.

Si bien la comprensión del razonamiento para resolver determinados problemas de programación puede resultar el mismo utilizando OO y procedimientos de programación estructurada, somos partidarios de iniciar en un primer nivel la enseñanza de la programación con un enfoque estructurado haciendo énfasis en sus estructuras básicas y luego sí pasar en un segundo nivel de enseñanza a un enfoque OO y adquirir conceptos tales como *clase, objeto, herencia* y *polimorfismo* entre otros [11].

Aprendizaje Colaborativo.

Nuestra propuesta plantea poder sumar a la enseñanza del diseño de algoritmos un valor agregado propio, ya hoy día, de nuestra vida cotidiana en la enseñanza, hablamos de herramientas de tipo colaborativas y disponibles estas en varios tipos de plataformas, principalmente las móviles.

Las tecnologías de la información y comunicaciones han transformado hace ya varios años nuestras formas de vida, incluso esta nueva era en la que vivimos ha sido bautizada como la sociedad de la información y del conocimiento [12].

El *aprendizaje colaborativo* puede definirse como el compromiso mutuo establecido entre personas de un grupo, las cuales se unen en

un esfuerzo coordinado para dar respuesta a una tarea [13].

El *aprendizaje colaborativo* es caracterizado por la interacción y el aporte de todos los participantes en la construcción del conocimiento y necesario para el logro de metas académicas, ya que facilita la comprensión y la explicación de temas por los propios compañeros, ya sean docentes o alumnos, fortaleciendo a su vez las relaciones sociales entre los mismos [14].

Teniendo en cuenta lo mencionado, consideramos que el uso de herramientas colaborativas multiplataforma como por ejemplo lo es *UAICase* [2], para la enseñanza de la programación a estudiantes iniciales de ingeniería de software, nos permitirá acceder a un mejor resultado en el proceso de asimilación, aprendizaje y resolución de situaciones problemáticas que se expresarán a través de un algoritmo en vista de, en una segunda etapa, poder codificar el mismo en el lenguaje formal procedural que se elija.

Aprendizaje Móvil.

La explosión en la adquisición y utilización de todo tipo de dispositivos móviles con acceso a internet por parte de la población, y con mayor afición por parte de estudiantes universitarios y en particular aquellos que cursan carreras de tecnología, fomentado esto a su vez por la disminución de los costos de accesos a servicios de banda ancha inalámbrica, crean un escenario ideal como nunca antes se nos había presentado para la enseñanza de la programación a través de medios colaborativos multiplataforma.

Basta mencionar para dar más entidad a esta afirmación que hoy en día más del 90% de la población mundial tiene acceso a redes móviles y que existen en el mundo más tablets y teléfonos móviles que gente en el planeta [15], la gran mayoría de ellos con pantallas de excelente resolución y potentes procesadores.

Más que nunca se podrá sacar provecho del concepto de *ubicuidad* para que los alumnos puedan en cualquier momento y lugar fomentar la interacción y colaboración con pares y docentes para el aprendizaje de disciplinas de programación.

Presentado este escenario tan propicio para el desarrollo de la *enseñanza y aprendizaje ubicuo y colaborativo*, planteamos la utilización de *UAICase* [2] para la consecución del objetivo de obtención de habilidades en el desarrollo de diseño de algoritmos que resuelvan problemas de programación en cursos iniciales a nivel universitario.



Figura 1. Pantalla de la Interfaz de acceso a UAICase.

La implementación de dicha herramienta nos propone desafíos en la enseñanza y en la tecnología. Desde el punto de vista pedagógico deberemos tener muy presente que no estamos frente a un simple acceso a

un repositorio de información, tampoco un aula virtual, se trata de un nuevo paradigma en la enseñanza y aprendizaje, una nueva disciplina que nos está mostrando que tendemos a ser más efectivos y eficientes en el trabajo en grupo colaborativo [1].

En cuanto a lo tecnológico se deberá hacer prevalecer las ventajas mencionadas de accesibilidad por sobre, por ejemplo, el reducido tamaño de una pantalla de smartphone comparada con la de una computadora de escritorio o portable.

También la portabilidad deberá ser algo primordial, la aplicación deberá poder correr sobre cualquier plataforma con las mismas bondades y no en una en especial.

El objetivo primordial será poder convertir en algo totalmente natural el hecho de la construcción de algoritmos desde los smartphones de los alumnos, para esto se deberá tener una rápida respuesta de evaluación y colaboración de los docentes.

Conclusiones.

En materias iniciales de programación en la enseñanza universitaria se debería tener como premisa la asimilación por parte de los alumnos de técnicas y métodos que les permitan resolver problemas de una complejidad accesible mediante la aplicación de una resolución lógica previa de los problemas planteados a través de algoritmos por sobre la escritura directa de código de cualquier lenguaje formal.

Teniendo en cuenta que muchos alumnos tienen su primer contacto con la resolución de problemas de programación recién a su

ingreso al nivel universitario, se deberá apuntar a que construya sus primeros algoritmos utilizando las tres estructuras lógicas básicas de secuencia, condicional e iteración; seguido a esto podrá acceder a la utilización de algoritmos básicos de recorrido y búsqueda.

En vista al desafío que enfrenta un alumno inicial de carreras relacionadas a la Ingeniería de Software, resulta de un alto valor agregado poder soportar su enseñanza y aprendizaje de la programación a través de medios colaborativos.

Los medios colaborativos cuentan a su vez con la ventaja de que no sólo pueden ser utilizados en el aula sino también fuera de ella. El otro gran punto a favor de las herramientas colaborativas es que las mismas pueden ser accedidas desde dispositivos móviles, los cuales son ampliamente utilizados, aceptados y forman parte primordial de la vida diaria de la franja etaria de personas en inicio de carreras universitarias.

Esta combinación entre una técnica con más de 40 años de existencia como la programación estructurada y una nueva propuesta fruto del desarrollo de las tecnologías de información y comunicaciones como lo son las herramientas colaborativas multiplataforma en los ambientes académicos, enriquece y motiva el aprendizaje de los conceptos básicos de la programación por parte de los futuros ingenieros en software.

Referencias

- [1] COBOS, Carlos, MENDOZA, Martha y GOMEZ, Luis. Ejercitación Colaborativa y Programación Estructurada. Revista UIS Ingenierías. Facultad de Ingenierías Físico Mecánicas. Universidad Industrial de Santander, Volumen 4, No.2, diciembre de 2005. Págs. 117-128. ISBN: 1657-4583.
- [2] Battaglia, N., Neil, C., De Vincenzi, M., & Martínez, R. (2016, June). UAICase: integración de un entorno académico con una herramienta CASE en una plataforma virtual colaborativa. In XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET).
- [3] Neil C., De Vincenzi M., Battaglia N., Martínez R. (2016). Herramientas Colaborativas Multiplataforma en la Enseñanza de la Ingeniería de Software. In XVIII Workshop de Investigadores en Ciencias de la Computación.
- [4] UNESCO. Policy guidelines for mobile learning (2013). ISBN 978-92-3-001145-1.
- [5] Pérez Pérez I., Monzalvo López C. A. (2010). Análisis del desempeño de los estudiantes en el diseño y construcción de algoritmos secuenciales, selectivos e iterativos. Universidad Autónoma del Estado de Hidalgo. Computación Educativa.
- [6] Sznajdleder, Pablo (2012). Algoritmos a fondo con implementaciones en C y Java. Alfaomega Grupo Editor Argentino 1ra Edición. Módulo 1 1.2, 1.3.2. ISBN 978-987-1609-37-6.
- [7] ACM/IEEE, Computing Curricula 2001, Draft, March, 2000.
- [8] J. García Molina, ¿Es conveniente la orientación a objetos en un primer curso de programación?, VII Jornadas de Enseñanza Universitaria de la Informática (JENU'2001).
- [9] TIOBE Index for August 2017 (2017). Puede ser encontrado en: <https://www.tiobe.com/tiobe-index/>.
- [10] Sznajdleder, Pablo (2013). Java a fondo: estudio del lenguaje y desarrollo de aplicaciones. Alfaomega Grupo Editor Argentino 2da Edición. Capítulo 1 1.2. ISBN 978-987-1609-36-9.
- [11] J. García Molina, et al., Una propuesta para organizar la enseñanza de la Orientación a Objetos, VIII Jornadas de Enseñanza Universitaria de la Informática (JENU'2002), Cáceres, Julio, 2002.
- [12] Figueroa, S. G., & Cordero, R. D. (2012). Informática ubicua: su aplicación en el contexto universitario. In XIV Workshop de Investigadores en ciencias de la Computación.
- [13] Lavigne, G., Ovando, M. P. V., Sandoval, J. O., & Salas, L. M. (1970). Exploración preliminar del aprendizaje colaborativo dentro un entorno virtual/Preliminary study of collaborative learning in a virtual environment.

Revista Actualidades Investigativas en Educación, 12(3).

[14] Brito, Vivina. (marzo, 2004). El foro electrónico: una herramienta tecnológica para facilitar el aprendizaje colaborativo. Edutec. Revista Electrónica de Tecnología Educativa. 17, 1-20. Puede ser encontrado en: <http://dialnet.unirioja.es/servlet/articulo?codigo=926914>.

[15] Ortiz, O, Alcover P, Sánchez F., Pastor, J. A., Herrero R.. Herramienta m-learning para el Aprendizaje de Programación Estructurada en los Primeros Cursos de Ingeniería (2014). VAEP-RITA Vol. 2 Núm. 2. Puede ser encontrado en: http://rita.det.uvigo.es/VAEPRITA/index.php?content=Num_Pub&idiom=Es&visualiza=3&volumen=2&numero=2&articulo=2.