

Estrategias pedagógicas para la enseñanza de la programación¹

Lucas Spigariol

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Departamento de Ingeniería en Sistemas de Información, Medrano 951 (C1179AAQ), Ciudad Autónoma de Buenos Aires, Argentina

lspigariol@gmail.com

Recibido el 18 de diciembre de 2015, aprobado el 2 de febrero de 2016

Resumen

En el ambiente profesional de Sistemas de Información la Programación Orientada a Objetos (POO) constituye un paradigma de creciente preponderancia como modo de construir soluciones informáticas. Asimismo, la universidad tiene un rol fundamental en formar profesionales actualizados tecnológicamente para fortalecer el proyecto productivo nacional. En este trabajo se aborda una experiencia singular que se llevó adelante en la materia Paradigmas de Programación de la carrera de Ingeniería en Sistemas de Información de la UTN.BA. Sus docentes desarrollaron un *software* educativo, denominado *Ozono*, que dentro del ambiente de desarrollo del lenguaje Smalltalk, se utiliza como estrategia pedagógica para enseñar a programar en *Objetos*. Por otra parte, con los aportes de las pedagogías críticas –en especial de Paulo Freire y del construccionismo de Seymour Papert, creador de *Logo*– se analiza la utilización de la herramienta con el fin de facilitar el proceso de aprendizaje de los estudiantes, su inscripción en la propuesta curricular de la asignatura y la perspectiva del contexto profesional, prestando especial atención a la práctica docente y a la experiencia de los mismos estudiantes.

PALABRAS CLAVE: ENSEÑANZA – SOFTWARE EDUCATIVO – PROGRAMACIÓN ORIENTADA A OBJETOS – OZONO – PEDAGOGÍA

Abstract

Assuming that Object-oriented Programming constitutes a paradigm with growing dominance as a way to build solutions in the professional environment of Information Systems and that the university has a fundamental role in education of updated technologically professionals to strengthen the national productive project, is boarded a singular experience that is realized in the signature "Programming Paradigms" in career of Information Systems Engineering (UTN.BA), where teachers have developed an educational software, called Ozono, that within the development environment of Smalltalk language, is used as a pedagogical strategy for teaching to program in Objects. With the contributions of critical pedagogies –especially Paulo Freire– and constructionism of Seymour Papert –the creator of Logo–, is analyzed the tool usage to facilitate the learning process of students, its inclusion in the proposed curriculum of the subject and the perspective of professional context, with special attention to teaching practice and experience of the students themselves. Also, guidelines for the use and maintenance of such software arise.

KEYWORDS: TEACHING, EDUCATIONAL SOFTWARE - OBJECT-ORIENTED PROGRAMMING - OZONO - PEDAGOGY

¹ El presente trabajo forma parte de la Tesis de Maestría en Docencia Universitaria: *Estrategias pedagógicas para la enseñanza de la programación. Un estudio de caso en la carrera de Ingeniería en sistemas de la UTN FRBA*, dirigida por

Dr. Oscar Bruno.

Introducción

"El estudio no se mide por el número de páginas leídas en una noche, ni por la cantidad de libros leídos en un semestre.

Estudiar no es un acto de consumir ideas, sino de crearlas y recrearlas"
Paulo Freire

Cuando un estudiante quiere aprender a programar se enfrenta a la computadora y tiene no solo que entender sino también expresarse en ese lenguaje formal con toda la complejidad de su sintaxis y lo abstracto de su simbología. Debe además, manipular entornos de desarrollo con múltiples opciones, interpretar y depurar errores, realizar seguimientos, validar resultados y todo lo que implica lograr que un programa funcione y lo haga correctamente. Es, sin lugar a dudas, una experiencia fuerte y desafiante para algunos y frustrante para otros. Así, en las primeras materias del área de la programación, el estudiante se encuentra atravesado por diferentes tensiones: la polisemia de su lenguaje natural cargado de sobreentendidos y ambigüedades, frente a la rigurosidad del lenguaje de programación. Se trata de la incuestionable mecánica computacional que no entiende la lógica de sentido común del modo de razonamiento humano. El acalorado entusiasmo del alumno –o desazón, cuando las cosas no salen– contrasta con la frialdad de la computadora. Si bien conoce la descripción del problema a resolver no sabe cómo expresar la implementación de su solución mediante un programa concreto. La multiplicidad de formas en que las personas interpretamos la información supera ampliamente lo acotado de los tipos de datos con que los lenguajes se manejan.

El problema central que aborda este trabajo es la dificultad que representa para un estudiante de los primeros años de la carrera de Sistemas de Información aprender a programar utilizando una herramienta profesional de desarrollo de *software*. El uso de un lenguaje de programación de alto nivel para desarrollar soluciones informáticas con su propio ambiente de trabajo introduce una dimensión práctica y una aproximación profesional ineludible para una carrera de ingeniería en la actualidad. Al mismo tiempo, permite ir aprendiendo a construir soluciones,

probándolas y validando los resultados con la máquina. Pero por otra parte, estas herramientas tienen una multiplicidad de opciones y requerimientos específicos que están pensados para agilizar la tarea de quien ya sabe programar, pero que, para quien está dando sus primeros pasos, representa una dificultad adicional a la complejidad propia de la programación. El uso de las herramientas profesionales de desarrollo generalmente requiere del manejo de una amplia cantidad de conceptos y conocimientos incluso para programar algo relativamente sencillo y que funcione.

En el ambiente educativo, frente a esta situación hay dos caminos típicos. Por un lado, se pueden retrasar las instancias de práctica concreta con los lenguajes hasta tener una base conceptual más sólida, lo cual tiene el riesgo de demorar demasiado la práctica. Por otra parte, se pueden explicar muchos conceptos para acceder más rápidamente a una práctica concreta, pero eso tiene el riesgo de apabullar a los estudiantes con numerosos conceptos sin el tiempo necesario para interiorizarlos. En ambos casos la disociación entre teoría y práctica que se genera es contraproducente y el desafío consiste en encontrar mediaciones pedagógicas que acerquen al estudiante a las herramientas de desarrollo profesional.

Utilizar determinadas herramientas y no otras es más que una elección sobre la mejor forma de transmisión de un contenido. La elección del docente acerca de qué recursos didácticos aplicar y la manera en que los utiliza no es arbitraria, ni neutral ni inocua, sino que va de la mano de la importancia, orientación y sentido que le otorga a los contenidos prescriptos y a la percepción que tiene del campo profesional. De allí que contar con un panorama claro del ejercicio de la profesión, permitirá seleccionar los recursos y estrategias didácticas adecuadas.

Una certeza que impulsa este trabajo es el conocimiento de ciertas experiencias innovadoras que se dan en la UTN.BA y que ameritan una mirada atenta para indagar más allá de lo superficial. En particular, la experiencia que se lleva adelante en la materia Paradigmas de Programación de la carrera de Ingeniería en Sistemas de Información. En dicha cátedra un grupo de docentes desarrolló un *software* específico, de-

nominado *Ozono*, que se utiliza como soporte para enseñar a programar dentro del paradigma de la Programación Orientada a Objetos (POO), con el lenguaje Smalltalk, como recurso que facilita los primeros pasos en el proceso de aprendizaje. Se analiza el uso que se hace de *Ozono* en tanto herramienta, su inscripción en la propuesta curricular de la asignatura y el enfoque que se le imprime al dictado de la materia, atendiendo tanto a la mirada de los estudiantes como de los docentes, y sin perder de vista el contexto profesional actual.

El marco pedagógico desde el cual se aborda el trabajo tiene como principal aporte la pedagogía crítica, con Paulo Freire como autor de referencia, y la perspectiva de Seymour Papert, con su aplicación de las teorías pedagógicas constructivistas al ámbito del desarrollo de sistemas.

***Ozono*: un software para aprender a programar en objetos**

El *software* educativo denominado *Ozono* fue desarrollado por un equipo de docentes univer-

sitarios frente a ciertas dificultades que notaban en el proceso de aprendizaje de sus estudiantes.²

El objetivo central de *Ozono* consiste en realizar ejercicios prácticos utilizando la computadora desde el inicio de la cursada sin que el estudiante deba conocer todos los conceptos del paradigma que se requieren para usar el lenguaje normalmente. Se basa en la posibilidad de crear objetos y enviar mensajes interactivamente sin definir clases, permite ir aplicando progresivamente los conceptos de delegación, polimorfismo y colecciones, entre otros, a la vez que cuenta con una interfaz gráfica que posibilita visualizar los objetos y sus referencias en el ambiente.

La Figura 1 muestra la ventana central de la herramienta, denominada Object Browser, donde se desarrollan los ejercicios. El ejemplo muestra objetos creados con sus correspondientes atributos y métodos. En la sección izquierda de la ventana se ven las opciones principales disponibles (referencias, *workspace*, tests, papelerera y diagrama) y la más importante es la que está desplegada y en la que se ven las referencias a



Fig. 1. Object Browser. Captura de pantalla de la ejecución de *Ozono*

²El proyecto fue desarrollado por docentes de la materia Paradigmas de Programación de la UTN.BA en un equipo impulsado por los ingenieros Nicolás Passerini y Carlos Lombardi, con la participación especial de varios colaboradores más. La primera versión de Ozono utilizada en un curso de paradigmas fue en el año 2006. Entonces se llamaba Object Browser y funcionaba sobre Dolphin Smalltalk 6.0. En 2009, cuando la nueva versión de Dolphin Smalltalk eliminó la posibilidad de contar con licencias educativas, se decidió la migración a Pharo, con el objetivo de seguir contando con una herramienta de código abierto. Finalmente en 2011, se incluyeron otras herramientas, llegando a una configuración similar a la actual. Esta versión se denominó Learning Object Oriented Programing "Aprendiendo a Programar en la Orientación a Objetos" (LOOP) y fue presentada en la conferencia del European Smalltalk User Group (ESUG) en el año 2011, obteniendo el tercer lugar en los 8° Innovation Technology Awards. Durante ese tiempo, cada vez más docentes se fueron involucrando en el proyecto, y su uso se extendió a la Facultad Regional Delta, UTN y a las Universidades Nacionales de Quilmes y de San Martín, en materias en las que también se enseña POO.

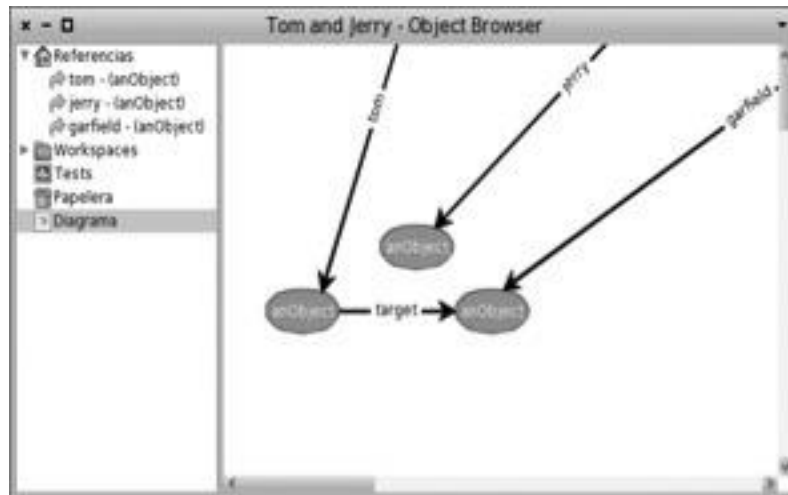


Fig. 2. Diagrama de objetos. Captura de pantalla de la ejecución de *Ozono*

los objetos creados (dos en el ejemplo).

Ozono permite crear un objeto mediante un comando interactivo en el que basta con ingresar un identificador con el que se lo va a conocer en el ambiente. Para llegar a definir un objeto de los que se ve en la figura fue necesario con el menú contextual *clickear* "nuevo objeto" y luego escribir *tom*. Dicho en otras palabras, no se requiere definir la clase, ni explicitar la forma en que se la instancia, ni ubicarla en la jerarquía, ni saber de antemano qué atributos o métodos necesita, lo que sí es necesario, en caso de utilizar directamente Smalltalk u otros lenguajes profesionales de objetos.

Luego, interactivamente se le pueden ir agregando los atributos y desarrollando los métodos, de manera que el objeto pueda responder a los mensajes que se le envíen. La sección inferior de la ventana es la que se utiliza para escribir el código de cada método. En el ejemplo, se ve el método "isGoingToGetFat" del objeto *tom*, que cuenta con una única línea en las que se envía el mensaje *weight* al objeto referenciado como *target* que es un atributo de *tom* y luego una comparación con el número 100, que también es un objeto.

Otra característica importante es que incluye un entorno gráfico de sencillo manejo, en el que se representa un diagrama de objetos, del cual se muestra un ejemplo en la Figura 2.

En el diagrama, los objetos son representados con óvalos, y las relaciones entre ellos, con flechas. Los objetos se pueden desplazar con el

mouse y las flechas se acomodan automáticamente. De esta manera se exhibe el estado interno de los objetos en el momento. A medida que se evalúan los mensajes del *workspace* o se corren los tests que provocan que el estado interno de los objetos se modifique, se actualiza el diagrama redireccionando las flechas de ser necesario.

Para crear objetos polimórficos, al igual que en Smalltalk, con *Ozono* solo se requiere que entiendan el mismo mensaje. No se necesita aprender ningún otro concepto o construcción específica del lenguaje. Lo que no permite hacer, y de alguna manera determina el alcance del *software*, es la definición de clases y, en consecuencia, todo el manejo de herencia que trae aparejado.

Los creadores de *Ozono* señalan que respecto del desaprovechamiento del Paradigma de Objetos se perciben en la industria del *software* algunas inercias y problemas:

- Muchos desarrollos no sacan provecho de las tecnologías basadas en objetos, sino que sus abstracciones son precarias, el polimorfismo inexistente y persiste una clara separación entre estado y comportamiento.
- En muchos lados se continua programando de forma procedural y se utiliza a las clases como módulos o estructuras, alternativamente. Incluso en equipos con mejor dominio del paradigma aún es frecuente ver una estricta orientación a clases. Esto significa que muchas veces los pro-

gramadores recurren a la subclasificación como única manera de tener comportamiento diferenciado, cuando bien podrían encontrar nociones más generales y simples y obtener distintas combinaciones de comportamiento empleando composición entre objetos.

Ozono como recurso pedagógico

Para desarrollar la herramienta y orientar el dictado de la materia los creadores del *Ozono* hicieron su propia lectura del panorama actual del ejercicio de la profesión. Durante la presentación del *software* a la comunidad científica, afirmaban: "Es muy frecuente que Proyectos de IT que utilizan herramientas donde se pueden aplicar las ideas de objetos, el aprovechamiento de las potencialidades que brinda el trabajo con objetos resulta escaso" (Lombardi, 2007).

Antes de implementar *Ozono*, los docentes veían que el uso del lenguaje representaba para los estudiantes una dificultad ya que era necesario manejar una gran cantidad de conceptos y atender a numerosas cuestiones propias de una herramienta profesional antes de poder desarrollar algo relativamente sencillo y que funcione en la computadora.

La propuesta educativa universitaria desde una lectura crítica del contexto profesional

Freire (1994), propone como criterio fundamental de la tarea docente lo que denomina enseñar a "leer el mundo". Su propuesta consiste en tomar la distancia necesaria para ver lo que sucede en el mundo, entendiéndolo como el contexto en el cual se logra una comprensión más exacta del objeto de estudio. En este sentido, propone revisar que los contenidos sean saberes socialmente relevantes, de carácter histórico y en relación con conocimientos académicos, abandonar interpretaciones mecanicistas o idealistas de la realidad, y asumir que las relaciones entre la conciencia y el mundo son dialécticas. A su vez, plantea la necesidad de mantener la intencionalidad de formar un sujeto capaz de insertarse activamente y transformar la realidad.

Desde esta perspectiva, resulta pertinente una lectura del contexto en el que ha de desempeñarse el profesional de sistemas para poder realizar una mirada crítica de la propuesta educativa universitaria. Lejos de asumir una concepción mercantilista de la formación universi-

taria que la reduzca a satisfacer las demandas presentes de la actividad privada. Es importante para orientar las capacidades que se pretenden desarrollar en los estudiantes contar con una mirada amplia y crítica de la actualidad profesional, del proyecto de desarrollo productivo del país y sobre todo de las tendencias que se vislumbran a futuro. A su vez, es fundamental tener presente que la universidad es un actor central que influye en cómo se va configurando el ambiente profesional de la actividad. Por lo tanto, debe analizarse qué y cómo se enseña en una disciplina en auge y con gran dinamismo como lo es el desarrollo de sistemas. En el mismo sentido, revalorizar, difundir, fomentar y también justificar pedagógicamente el desarrollo de herramientas tecnológicas específicas tendientes a facilitar, potenciar y orientar la tarea de programar.

El uso de tecnología educativa en el contexto de una propuesta pedagógica

Basándose en la concepción del modelo de enseñanza de Piaget, Seymour Papert hace referencia a que la incorporación de la computación en la enseñanza, por sí misma no constituye un cambio en la educación si no se cuestiona la naturaleza del proceso de aprendizaje. Él propone un cambio en el equilibrio entre la transferencia de conocimiento a los estudiantes –ya sea mediante libros, docentes o *software*– y la producción de conocimiento por parte de ellos.

Realmente habrá ido más allá del primer impacto si las computadoras desempeñan un papel en servir como mediadoras de un cambio en los criterios que gobiernan los tipos de conocimiento que se valoran en la educación (Papert, 2002, p. 17).

Papert, creador del *Logo*, lenguaje pionero de la enseñanza de la programación, destaca la importancia de que los maestros aprendan a medida que enseñan. En referencia a la forma de dar clases que denomina "instruccionista", sostiene que cuando la enseñanza consistía en pararse delante de una clase y hablar, el maestro tenía que saberlo todo previamente. En cambio, desde el modelo "construccionista" la enseñanza propone un trabajo con el estudiante a partir de proyectos creativos que impliquen la construcción de algo –como sucede con el desarrollo de *software*–. De esta manera, el recurso informático no se limita a cómo conseguir nuevos materiales o a repetir los mismos esquemas de instrucción pero con una computadora en lugar de un pizarrón, sino que

propicia una nueva dinámica de aprendizaje, una redefinición de los roles y una relación más fluida e interactiva entre profesor y estudiante.

Recuperando el aporte de la tecnología en educación Burbules (2001) expresa que

... la tecnología no es solo la cosa, sino la cosa y las pautas de uso con las que se aplica, lo más nuevo, tal vez, no sea la tecnología, la cosa en sí, sino todos los otros cambios que la acompañan.

Por lo tanto, es menester hacer el análisis de las herramientas tecnológicas en su contexto de utilización. A su vez, es desde una propuesta pedagógica con criterios y objetivos claros que tiene sentido utilizar una herramienta concreta en determinado momento del proceso educativo y con una intencionalidad definida.

Lejos de pensar que se trata de atributos intrínsecos de la herramienta que por sí solos garantizan buenos resultados, hay que ver quién, cómo y para qué se la utiliza. Y es oportuno denominarla "herramienta" recuperando el valor artesanal que tiene su utilización. Apelando a su sentido originario como continuidad de la mano del obrero, del campesino o del artista, *Ozono* es una herramienta en tanto que depende de cómo, cuándo y para qué la utilice, en este caso el trabajador de la educación, que tiene a su cargo la conducción del proceso de enseñanza.

Por qué y para qué utilizar una herramienta como Ozono.

En qué consiste la intencionalidad pedagógica de los docentes que deciden utilizar esta herramienta, cuáles son los conceptos y habilidades que ella refuerza y pondera.

Desde el punto de vista de los contenidos, la característica principal que todos los docentes valoran es que sin necesidad de explicar muchos conceptos, les permite a los estudiantes crear, ver y entender qué es un objeto, –el concepto fundante del paradigma– e inmediatamente enviarle *mensajes*. Así observar el sentido, el "para qué" de dicho objeto y lograr una aplicación que funcione y resuelva problemas. Con estas dos ideas ya es posible avanzar en prácticas concretas, no solo con ejercicios en "lápiz y papel", sino codificando soluciones que se ejecutan y se pueden probar utilizando la computadora.

Se destaca la necesidad de subrayar la identidad del paradigma de objetos y su diferenciación de la programación procedural. Para los estudiantes con conocimientos previos en el *paradigma imperativo* el problema típico es seguir programando de manera procedural en lenguajes de objetos, lo que refuerza la propuesta de *Ozono* de presentar desde el primer momento un modelo de trabajo a partir de objetos y mensajes, en vez de instrucciones y que omite la presencia de clases que puedan asociarse equívocamente a estructuras de datos propias de la programación estructurada.

Son reiteradas e insistentes las apreciaciones de los docentes acerca de poner en un segundo plano el concepto de clase, como planteaban los creadores de la herramienta. En esta tensión se entienden mejor las apreciaciones sobre la centralidad del concepto de objeto. Algo que no se puede hacer utilizando *Ozono*, pero lejos de ser visto como una limitación es la frontera que le da sentido, es definir clases. No se trata de una decisión arbitraria ya que hay un convencimiento de que "el objeto y el mensaje son los conceptos centrales en vez de las clases." (Griggio, 2011).

Consultados en relación con los elementos teóricos cuyo aprendizaje facilita la utilización de *Ozono*, los estudiantes destacaron en primer lugar su utilidad para comprender los conceptos de *objeto* y *mensaje*, que constituyen las ideas fundantes del paradigma, sin las cuales no tiene sentido hablar de los demás conceptos.

Respecto de las funcionalidades básicas se observó que la forma interactiva de "creación de objetos" y la ventana de definición de "atributos y métodos" de dichos objetos cuenta con preferencias positivas de los estudiantes: buena o muy buenas por encima del 85%. (Creación de objetos: Muy bueno 55% y Bueno 30%; Atributos y métodos: Muy Bueno 49% y Bueno 41%). Ambas son las opciones que se utilizan necesariamente para la realización de cualquier ejercicio.

A continuación del concepto de *objeto-mensaje*, el más mencionado por ellos es *polimorfismo*. Luego siguen los conceptos de *encapsulamiento* y *delegación* –que guardan estrecha relación entre sí dentro de la lógica de funcionamiento del paradigma– para cerrar con el concepto de referencias.

Por su parte, los docentes destacan la importancia del *polimorfismo* como concepto fundamental para las buenas prácticas profesionales y desde allí hacen uso de *Ozono*. Uno de ellos afirma que

extender el uso de *Ozono* permite permanecer en los conceptos más puros de objetos, avanzado en conceptos y herramientas para los cuales no son necesarias las clases, principalmente el polimorfismo, que se ve mejor al alejarlo del concepto de tipo o clase y más centrado en el comportamiento de los objetos.

La necesaria y conflictiva articulación entre teoría y práctica

La dicotomía entre teoría y práctica es uno de los problemas más recurrentes de la educación universitaria. Paulo Freire recomienda respecto de los sistemas educacionales, evitar todo tipo de propuesta "que menospreciase la teoría, negándole toda importancia y enfatizando exclusivamente la práctica como la única valedera, o bien negase la práctica atendiendo exclusivamente a la teoría" (Freire, 1994, p.34).

Retomando lo que Freire llamó *educación bancaria*, muchas veces en la actualidad universitaria se tiene una concepción de enseñanza que se reduce a transferir el concepto vacío de contenido y no atender a la comprensión precisa de este contenido por parte del estudiante. Es evidente que para la enseñanza de una disciplina instrumental como la programación es conveniente desarrollar una dinámica que combine lo teórico con lo práctico, donde la práctica permita comprender mejor el sentido de los conceptos teóricos y a la vez que dichos conceptos permitan plasmar soluciones prácticas más adecuadas.

A su vez, la teoría constructorista de Papert entiende el espacio educativo como una instancia de creación en la que teoría y práctica confluyen de manera tal que es imposible –o no tiene sentido– diferenciarlas.

Las diversas experiencias relevadas de creación de herramientas de *software* educativo para aprender a programar giran también en torno a esta preocupación. Por último, en un acercamiento al ambiente institucional cabe citar lo que la misma UTN afirma al respecto, cuando propone "una nueva relación donde la práctica deja de ser la mera aplicación de la teoría para

convertirse en fuente del conocimiento teórico". (SAyP 2007, p. 5)

Como indicador de la articulación entre teoría y práctica, se puede observar el tiempo que un estudiante necesita para realizar una práctica concreta, lo que está influenciado en buena medida por la necesidad de conocimientos previos. Más allá de lo que aquel conoce de materias anteriores o lo que haya aprendido por su trabajo o fruto de su curiosidad personal, el foco está en ver cuánto de lo que forma parte del contenido teórico de la materia es necesario explicar antes de poder realizar una práctica concreta. Es cierto que se puede considerar práctica a la elaboración de ejercicios en papel, de diagramas y pseudo-códigos, a los que incluso se les puede dar un formato de programa real, pero tratándose de una materia de programación dentro de una carrera de ingeniería la práctica a la que se apunta es la que se realiza sobre la máquina, donde el programa se ejecuta y permite conocer los resultados o efectos producidos.

Los lenguajes de programación profesionales que se encuadran dentro de lo que plantea el paradigma de la POO tienen como concepto central, precisamente, el concepto de *objetos*. En el caso del lenguaje Smalltalk, pese a tener integrado un entorno de desarrollo propio, visualmente agradable y con diversas herramientas y opciones para facilitar el desarrollo, no es para nada trivial su uso. Si se quiere lograr un programa sencillo, que por ejemplo tenga solo un *objeto* y haga una tarea básica, para que funcione se debe definir primero la *clase* de la cual ese *objeto* va a ser instancia, y a dicha -hay que ubicarla mediante la *herencia* dentro de una jerarquía de *clases* que viene definida por el lenguaje. Además, a la porción de código que realiza la tarea que se requiere no hay que escribirla sobre el objeto mismo sino, previendo que pueda haber una generalización en la que haya miles de objetos similares, se la debe definir en la clase, con un *método*, del cual hay que especificar que se trata de un *método de instancia*. Por supuesto, el código que se escriba debe cumplir con los requisitos de sintaxis propios del lenguaje. Finalmente, hay varias formas de probar el programa; la más sencilla requiere que se abra una sección especial del ambiente de desarrollo –el *workspace*– y codificar allí la secuencia de *instanciación*, el envío del mensaje y prever la forma de observar el resultado.

Los primeros en utilizar *Ozono*, en su doble rol de docentes y desarrolladores, afirman que el principal sentido de haber creado el *software* fue lograr que "los alumnos puedan experimentar rápidamente la aplicación de los conceptos de objetos, mensajes y relaciones, haciéndolos andar en una computadora y pudiendo aprender de los resultados que obtienen" (Lombardi, 2007).

Esa intuición inicial es confirmada por los docentes actuales, quienes utilizan *Ozono* desde la primera clase para ir probando con sencillez los programas que desarrollan, sin necesidad de muchos conceptos previos.

De las numerosas opiniones de los estudiantes una de ellas expresa con claridad: "Es una herramienta muy didáctica para aquella persona que no sabe programar y una buena introducción para encarar el paradigma de objetos." Otro estudiante afirma: "Destaco la facilidad con la que incluye los conceptos básicos del paradigma y los lleva a la práctica para los que acabamos de iniciarnos en el paradigma. Es una buena manera de empezar a entender los objetos y ver cómo se relacionan." Sin rodeos ni eufemismos, un estudiante resume una sensación compartida con muchos otros: "Una cosa es entender la explicación del profesor pero otra cosa es ver cómo funciona en la práctica". Más del 60% de los estudiantes valora la posibilidad que ofrece el *software* al permitir practicar con la computadora desde el primer momento.

Si se retoma el ejemplo anterior, para lograr hacer el mismo programa con un objeto que realiza una tarea y que funcione en *Ozono*, ciertamente hay que tener alguna idea del concepto de *objeto* y de *mensaje*, que es precisamente el punto que se quiere practicar, pero no es necesario conocer de *clases*, *herencia*, ni otras definiciones del lenguaje, como tampoco preocuparse por la visualización de resultados. Y otro aspecto a considerar, más allá del manejo técnico, es que el alumno se concentra en un objeto en particular y la tarea que debe hacer, sin desviar la atención hacia una posible generalización, ciertamente más abstracta.

Uno de los docentes lo explica comparando su experiencia actual con lo que pasaba años anteriores: "El alumno antes del uso de *Ozono*

encontraba una barrera inicial conceptual muy difícil de franquear: la creación de objetos mediante clases. Esto se resolvía retrasando el uso del lenguaje de programación y su prueba en la computadora hasta más avanzado el curso, al introducir el concepto de clase".

La gradualidad y complejidad del proceso de enseñanza-aprendizaje

El aprendizaje es un proceso complejo que requiere articulación y gradualidad. Lo abultado del elenco de contenidos que se prescriben en los programas de las asignaturas, las relaciones entre ellos y lo potencialmente inabarcable que se esconde detrás de cada término en el ámbito profesional actual, hace necesaria una cuidadosa estrategia de selección, secuenciación y articulación de su desarrollo. De esta manera, se trata de no apabullar al estudiante de entrada, de no plantear un ritmo demasiado intenso o una velocidad excesiva, como por otra parte tampoco subestimar su capacidad.

En este sentido, es valioso rescatar los aportes de Lev Vigotsky (1993) sobre lenguajes y mediación, y poder articular el uso de herramientas que permitan el aprendizaje gradual de la programación con su concepto de "Zona de desarrollo próximo", entendida como aquellos conceptos o habilidades que el estudiante no conoce, pero que están a su alcance de ser aprendidos, con las mediaciones adecuadas. Otro elemento a destacar, es la categoría de "andamiaje" que se inscribe en la teoría sociohistórica y consiste en una estructura de sostenimiento de las prácticas de aprendizaje. Expresado por Jerome Bruner (1988), en continuidad con los trabajos mencionados de Vigotsky, permite que el estudiante participe en actividades y tareas que vayan más allá de sus conocimientos individuales, pero en las que podrá lograr un mejor desempeño gracias al soporte de "andamiaje" del educador o compañero con más experiencia. El objetivo que se persigue dentro de la propuesta docente es que *Ozono* cumpla la función de "andamiaje", que sea una mediación adecuada y provisoria para que el estudiante aprenda.

Precisamente, una de las características de *Ozono* es que en un primer momento se ocultan los aspectos más difíciles de entender para el estudiante y se los revela progresivamente. En otras palabras, analizando técnicamente la implementación de la herramienta, lo que

hace es que ciertas tareas que en el entorno habitual de Smalltalk se deben hacer mediante código, se automaticen con un click, y de esta manera queden ocultos al usuario. Así, el estudiante necesita comprender un conjunto acotado de conceptos para poder construir su solución informática. Una vez afianzados estos conceptos, podrá dar un paso más y lo que antes resolvía automáticamente la herramienta, ahora lo aprenderá a hacer mediante el código que es como se trabaja profesionalmente.

Un docente actual recuerda su experiencia como alumno de la materia usando *Ozono*, y destaca la importancia de la gradualidad de la enseñanza y de empezar pronto a trabajar en la máquina: "Fue muy positivo, porque favoreció el aprendizaje gradual del paradigma. Me facilitó mucho el proceso de aprender a pensar en objetos y en los mensajes que entienden desde el primer día".

Relación con conocimientos previos

En primer lugar, entre quienes no tenían conocimientos previos de POO, alrededor del 60% de los estudiantes, la mitad pondera que *Ozono* fue de mucha ayuda para entender los conceptos del paradigma. Con una opinión más moderada, el 25% comenta que le ayudó a comprender mejor algunos conceptos y un 20% que le ayudó a entender el lenguaje en particular. El restante 5%, por su parte, afirma que la herramienta en cuestión le ayudó poco o nada.

Por otra parte, alrededor del 40% de los estudiantes manifestó tener conocimientos previos de POO. Entre ellos, solo el 10% afirma que el uso de *Ozono* le ayudó poco o nada para su aprendizaje. Para algunos "olvidarse" de las clases y pensar en objetos representa una dificultad más que una mediación didáctica y se muestran disconformes con su uso. La mayoría, más del 60%, valora que fue de ayuda para comprender mejor algunos conceptos del paradigma y para entender y utilizar el lenguaje de programación en particular.

Hay quienes tienen una mirada solidaria, asumiendo que si bien a ellos no les ayudó mucho, entienden que fue de utilidad para sus compañeros. Los casos más interesantes son los de cerca de un 30% de los estudiantes quienes reconocen que creían saber del Pa-

radigma de Objetos, pero que al avanzar con el uso del *Ozono* se dieron cuenta no era así, que tenían una idea equivocada de lo que es programar en objetos.

Si bien en ambos grupos se reconoce ampliamente la utilidad de *Ozono* como ayuda para el aprendizaje, se constata una mayor valoración positiva entre quienes no tenían conocimientos previos en programación en objetos por sobre los que sí los tenían. A su vez, es significativo señalar cómo en algunos estudiantes (quienes creían conocer el paradigma y se dieron cuenta de lo contrario) se pone en evidencia que el aprendizaje, lejos de ser sencillamente acumulativo, provoca una reestructuración de los conocimientos, "desaprendiendo" ciertas cosas y reorganizando los saberes previos. Es el proceso que desde el constructivismo se denomina de acomodación y asimilación, que conduce a un nuevo equilibrio cognitivo donde lo nuevo se integra con lo anterior. Vale aclarar que no se debe confundir la gradualidad del aprendizaje con una linealidad unidireccional, ya que precisamente se trata de reorganizar y no de acumular.

El pasaje de Ozono a Smalltalk

Dentro del proceso gradual de incorporación de complejidad, llega un momento en que *Ozono* ya no resulta útil por la necesidad de abordar temas que no contempla, en particular el concepto de *clase*. Los docentes asumen que una vez transcurrido cierto tiempo del desarrollo de la materia, habiendo entendido, madurado y puesto en práctica una serie de conceptos sin la necesidad de recurrir a las *clases*, el estudiante ya está en condiciones de dejar de lado *Ozono* y pasar a usar el entorno del Smalltalk tal cual es –en particular la implementación de Pharo– y en él empezar a definir *clases* para luego continuar con otros conceptos del paradigma en esa línea, como la *herencia*, que es una relación entre *clases*, o la redefinición de *métodos*, que solo tiene sentido en un contexto de *herencia*.

Uno de los docentes lo expresa de esta manera: "Cuando me aseguro que saben el concepto más importante que es modelar usando polimorfismo, me doy cuenta que es el momento para dejar de usar *Ozono* y pasar al ambiente de Smalltalk. No hay que dejar pasar mucho tiempo porque si no la adopción del concepto de *clase* es más traumática".

Retomando la idea de considerar a *Ozono* como un ejemplo del concepto metafórico de "andamiaje" de Bruner, en tanto medio de brindar apoyo para que el estudiante pueda adquirir un nuevo conocimiento, se reconoce también su carácter provisorio, tal como lo señalan los pedagogos constructivistas. Cuando el estudiante comprende la dinámica del envío de mensajes a los objetos, las referencias y el *polimorfismo* ya no tiene sentido seguir utilizando dicho *software*, por lo que se lo deja de lado y se pasa el lenguaje de programación profesional.

Ozono está integrado en el entorno de Smalltalk y guarda cierta analogía visual y funcional, pero la experiencia de los estudiantes y la mirada de los docentes da cuenta de un abanico de posiciones que muestra que aún le falta suavizar didácticamente ese salto.

Quienes ven la continuidad entre ambas herramientas, por ejemplo, destacan: "Está bueno que se pueda trabajar a nivel objeto porque así pude comprender mejor cómo funciona el paradigma y cómo utilizar el lenguaje. Cuando pasamos al System Browser de Smalltalk todos los conocimientos que aprendí con *Ozono* fueron aplicables."

En contrapartida, a muchos estudiantes les resultó complejo hacer el cambio: "Me parece que *Ozono* no está muy integrado a Pharo. De una semana para la otra se deja de usar porque no soporta clases y es como volver a aprender a usar otro sistema."

Los docentes son conscientes también que la transición del uso de *Ozono* al entorno clásico del Smalltalk aún presenta ciertas complejidades y si bien intentan mostrar la continuidad notan que resulta un cambio brusco para algunos de sus alumnos.

La importancia de la motivación y del protagonismo del estudiante

El aprendizaje no es una acumulación de saberes sino un proceso de construcción y los errores no deben ser vistos como un fracaso sino como una oportunidad de aprendizaje. Como afirma Gardner en relación a la importancia de la motivación, los estudiantes "serán mucho más hábiles en aquellas ocupaciones que tienen que ver con sus intereses y esfuerzos, y que son valoradas por sus adultos y sus iguales en el entor-

no" (Gardner 2008, p. 115). La importancia del modo de acercamiento, la predisposición y la actitud del estudiante frente a su propio objeto de estudio son factores que influyen en su proceso de aprendizaje. A su vez, no se trata exclusivamente de una cuestión del alumno en sí sino que está condicionado por el contexto en el cual se desarrolla la situación. Por lo tanto, resulta oportuno el análisis acerca de en qué manera el uso de una herramienta pedagógica como *Ozono* tiene un efecto sobre dichos factores.

Entre los estudiantes, las tres cuartas parte reconoce un impacto favorable, detallando que el uso de *Ozono* despertó su interés por la programación orientada a objetos (55%) y en otros casos fue un elemento motivante (19%). En contrapartida, un 5% remarcó que le provocó aburrimiento. Completando el panorama, una franja intermedia de alrededor del 20% afirma que le resultó indiferente.

Una característica fundamental que *Ozono* aporta a la motivación del estudiante es la funcionalidad que realiza el "diagrama de objetos" a partir de los objetos creados y del código escrito. Lo interesante, es que el diagrama se va modificando automáticamente conforme a las pruebas que se realizan o los nuevos objetos que se crean.

A partir del trato directo con los estudiantes en los trabajos de laboratorio y de su anterior experiencia como alumnos, una de las docentes ayudantes explica cómo el *software* permite fijar las ideas a partir de su representación gráfica:

Para armar modelos de objetos, a los estudiantes les cuesta entender al principio algo abstracto. Pensar que hay un objeto en algún lado, saber que hay un entorno que no sabemos bien cómo es, hablar de referencias pero no plasmarlas, los deja con una idea de "programar en papel" poco aplicable. Al incorporar una herramienta como *Ozono*, todo esto se torna pragmático y empírico: ¡Puedo ver lo que estoy haciendo! Veo cómo la referencia es una "flecha" que apunta ahora a otro objeto, que es un redondelito de color, y cómo cambia el sistema con cada mensaje que envío.

El diagrama de objetos es el aspecto en el que más se detienen los estudiantes a opinar y plantear sugerencias, lo que habla de la importancia que adquiere para los alumnos y marca una pis-

ta clara para futuras mejoras en el *software* que encare el cuerpo docente.

La potencialidad de crear herramientas tecnológico-pedagógicas en la universidad

En un país que necesita de un sólido desarrollo tecnológico propio y en el marco de la universidad pública, se reconoce el rol activo de la UTN como innovadora en materia científico-tecnológica y como ámbito donde se investiga y se desarrolla tecnología. Con la mirada puesta sobre la importancia creciente del uso de *software* como recursos pedagógico, se identifica a la carrera de Ingeniería en Sistemas de Información como el lugar donde la necesidad y la capacidad confluyen y es posible desarrollar el *software* que se utilice con fines educativos en la misma carrera. Que los docentes de programación puedan utilizar como recurso pedagógico un *software* creado por ellos mismos es una singularidad que permite un sinnúmero de posibilidades.

Frente a la serie de problemas o posibles mejoras, trabajar con los estudiantes con una herramienta propia es una experiencia con un potencial increíble. Por un lado es útil porque permite ir corrigiendo dichos errores, pero principalmente les permite a los docentes seleccionar los conceptos de programación que consideran oportuno incorporar u ocultar en la herramienta, para ir orientando paulatinamente el proceso de aprendizaje, proporcionando un terreno propicio para la introducción gradual de los conceptos acorde a la planificación que ellos mismos hacen de la materia.

Al respecto, uno de los desarrolladores de Ozono recuerda:

... Hacer Loop –como lo llamábamos entonces– fue un buen ejercicio. Hacer una herramienta con el objetivo de mejorar nuestra llegada como docentes es muy valioso no solo para el estudiante que va a usarla, sino que lo fue para mí como docente. Entender y discutir lo que queremos enseñar es básico para el crecimiento del docente y de la materia dada, y desarrollar una herramienta puso sobre la mesa cuestiones a discutir que a veces hasta hubieran parecido axiomáticas.

Aludiendo a las reflexiones de Paulo Freire so-

bre el aprendizaje continuo del docente, para los integrantes del equipo docente, el hecho de participar de un proceso de uso y también de definición, creación y modificación del mismo *software* que luego utilizan en el aula, constituyó una oportunidad muy valiosa de formación.

Por su parte, los estudiantes (62%) reconocen el esfuerzo que realiza la facultad en materia de investigación y desarrollo de herramientas informáticas pedagógicas como Ozono. Un 30% sostiene que les generó curiosidad ver cómo estaba desarrollado internamente el *software* y les gustaría poder hacer cosas similares.

Impacto y resultados

En la actualidad, entre los docentes –incluidos los desarrolladores del Ozono– hay una amplia coincidencia en evaluar positivamente la herramienta. Un indicador valioso que observan es la comprensión de los conceptos de la materia y su puesta en práctica en situaciones concretas. En este sentido, los docentes coinciden en señalar que la respuesta que se ve en los estudiantes permite aumentar el nivel de complejidad de trabajos prácticos y exámenes, sin que ello signifique la fractura del proceso de aprendizaje, en términos generales. Esta tendencia es consistente con las modificaciones en las planificaciones de los últimos años respecto de los anteriores, en las que se incluyen temas que antes no aparecían.

Los docentes expresan en función del alcance de los ejercicios de carácter práctico:

El impacto es netamente positivo, aumentando el nivel de compromiso inicial. Los alumnos adquieren notablemente un conocimiento más preciso de conocimientos y más rápidamente, por lo que es factible ampliar el temario y enseñar más conceptos o bien los mismos pero con mayor profundidad.

Respecto de la evaluación, otro docente agrega que “el tipo de problemas que surge de la corrección de los exámenes es diferente, ya que hay ciertos errores que anteriormente eran más frecuentes y ahora lo son menos”.

En cuanto a la percepción de los estudiantes: su apreciación global del valor pedagógico de la herramienta es en promedio, cercana a 8 (7,75), en una escala de 1 a 10. Más del 60% pondera la

posibilidad de practicar con la computadora desde el primer momento y un porcentaje similar destaca la importancia de poder ver gráficamente cómo son y cómo se relacionan los objetos. Luego, colectando alrededor de la cuarta parte de las adhesiones, los estudiantes afirman que trabajar con *Ozono* sobre el lenguaje Smalltalk les resultó mucho más sencillo que otros lenguajes (23%) y ver que los programas funcionaban rápidamente fue gratificante (27%). En el otro extremo, ante la pregunta directa, algunos estudiantes (5%) acuerdan en que *Ozono* les pareció un poco básico, incluso infantil, remarcando que esperan otra cosa de la universidad.

Conclusiones

En síntesis, lo que brinda una mayor utilidad a la herramienta es el momento, el enfoque y la forma de uso que se hace de ella, acorde a una ponderación de la importancia relativa de los temas de la asignatura desde la percepción del campo profesional, a las decisiones acerca del orden y gradualidad en que se presentan los temas y a la importancia que se le da a la resolución de problemas reales utilizando un lenguaje concreto como articulación entre teoría y práctica.

El detenimiento en los aspectos destacables de la experiencia desde el punto de vista pedagógico y de sistemas no oculta las dificultades y

problemas detectados. Tanto en la experiencia directa de los estudiantes como de los docentes, como lo que se puede ver desde la mirada analítica, con algunos matices, son coincidentes los problemas en el uso de *Ozono*, en particular la dificultad de tener que cambiar de herramienta en la mitad de la materia, para enfrentarse a Smalltalk.

Los docentes reconocen el desafío que supone la transición de *Ozono* a Smalltalk, y por eso están trabajando intensamente en una nueva versión de la herramienta superadora de la actual, denominada Wollok. Este *software*, mantiene el mismo enfoque pedagógico, pero, en lugar de ser un *software* educativo que se monta sobre Smalltalk y se limita a los primeros conceptos de la programación orientada a objetos, consiste en un lenguaje de programación en sí mismo y con su propio ambiente de desarrollo. De este modo, se pueden abordar todos los conceptos del paradigma.

Agradecimientos

A quienes con ímpetu investigador crearon *Ozono*, a quienes como estudiantes juegan con él para aprender a programar, a quienes como docentes lo usan para enseñar y seguir aprendiendo y a quienes asuman el desafío de seguir utilizándolo y recreándolo.

Referencias

- BURBULES, N. (2001). Educación: riesgos y promesas de las nuevas tecnologías de la información. Buenos Aires, Granica.
- FREIRE, P. (1994). Carta a quien pretende enseñar. Buenos Aires, Siglo Veintiuno.
- GARDNER, H. (2008). La mente no escolarizada. (3ª ed.) Buenos Aires, Paidós.
- GRIGGIO, C.; LEIVA, G.; POLITO, G.; DECUZZI, G. y PASSERINI, N. (2011). A programming environment supporting a prototype-based introduction to OOP
- LOMBARDI, C.; PASSERINI, N. y CESARIO, L. (2007). "Instancias y clases en la introducción a la programación orientada a objetos". Smalltalks 2007 – Primera Conferencia Argentina de Smalltalk, 2007.
- PAPERT, S. y HAREL, I. (2002). Situar el Construcciónismo. Centro Latinoamericano para la Competitividad y el Desarrollo Sostenible, Alajuela. Traducido de PAPERT, SEYMOUR. P. y HAREL, I. (1991). Constructionism Ablex Publishing Corporation.
- SAyP (Secretaría Académica y de Planeamiento) (2007) "El currículum" en Didáctica en la Universidad. Facultad Regional Buenos Aires. Universidad Tecnológica Nacional.
- VIGOTSKY, L. (1993). Pensamiento y lenguaje. En Obras Escogidas. Tomo 2. Madrid, Visor.