

# RITA: una herramienta didáctica-pedagógica innovadora en la escuela secundaria

Experiencias dentro del Proyecto “Articular universidad-escuela con JAVA para fortalecer la Educación-Técnica. Conectar Saberes”

Claudia Queiruga, Laura Fava, Claudia Banchoff Tzancoff, Vanessa Aybar Rosales, Isabel Miyuki Kimura, Matías Brown Bertneche

LINTI (Laboratorio de Investigación en Nuevas Tecnologías en Informática)

Facultad de Informática – Universidad Nacional de La Plata

La Plata, Buenos Aires, Argentina

{claudiaq, lfava, cbanchoff, vaybar }@info.unlp.edu.ar, ikimura@linti.unlp.edu.ar, mbrown@linti.unlp.edu.ar

**Resumen**—Este artículo presenta la herramienta didáctica-pedagógica RITA (Robot Inventor to Teach Algorithms) destinada a alumnos de escuelas secundarias de la provincia de Buenos Aires (Argentina) y los resultados de su aplicación en el proyecto piloto “Articular universidad-escuela con JAVA para fortalecer la Educación-Técnica. Conectar Saberes” de la Facultad de Informática de la UNLP. El propósito de esta herramienta es acercar a los jóvenes estudiantes secundarios a la “programación”, convencidos que el “pensamiento computacional”[4] promueve habilidades consideradas fundamentales para la sociedad del siglo 21. RITA es un juego de programación que estimula la resolución de problemas mediante la construcción de estrategias de supervivencia de robots virtuales; está basado en dos frameworks de código fuente abierto: OpenBlocks[10] y Robocode[11]. Es una herramienta didáctica de carácter lúdico, apropiada y motivadora para adolescentes “nativos digitales”, que permite introducir conceptos de “programación” y del lenguaje JAVA. RITA es un puente entre la educación secundaria y la universitaria.

**Abstract**—This paper presents the RITA (Robot Inventor to Teach Algorithms) didactic-pedagogical tool, meant for students of high schools in the province of Buenos Aires (Argentina), and the results of its application to the pilot project "Articulating University and School with JAVA to Strengthen Technical Education. Connecting Knowledge" presented by the Computer Science School of the UNLP. The purpose of this tool is to bring young high school students closer to "coding", with the strong conviction that "computing thought"[4] promotes skills that are paramount in 21st century society. RITA is a programming game that stimulates problem solving skills by means of the construction of virtual robot survival strategies. The game is based on two open source frameworks: OpenBlocks[10] and Robocode[11]. It is a didactic tool of a fun, appropriate and motivating nature for "digital native" teenagers that allows for the introduction of "coding" and JAVA concepts. RITA is a bridge between high school and university.

**Keywords**—Java, enseñanza de programación, articulación universidad-escuela, juegos

## I. INTRODUCCIÓN

En las últimas décadas, el alto nivel de deserción en los primeros años de las carreras técnicas universitarias de nuestro país, Argentina, es preocupante. Las causas que dificultan la transición y continuidad de los alumnos en la vida académica del sistema universitario son variadas y múltiples. La universidad, la familia, la situación socio-económica y la escuela influyen desde diferentes lugares y perspectivas en la formación y contención de los estudiantes que ingresan a la universidad. Sabemos que la problemática es multicausal, los factores van desde el contexto desfavorable del que provienen muchos adolescentes, la falta de competencias para resolver problemas, éstas requeridas en las carreras vinculadas a la ciencia de la computación, hasta la falta de contención de los ingresantes por parte de las instituciones receptoras. Existen múltiples iniciativas desde el espacio público en Argentina, destinadas a facilitar tanto el ingreso como la permanencia de nuestros jóvenes estudiantes secundarios en nuestra universidad pública. Los programas de becas de la Secretaría de Políticas Universitaria[1], así como los de tutorías abordan cuestiones vinculadas con la ayuda económica y la contención para continuar los estudios. Acciones articuladoras entre la escuela secundaria y la universidad que promuevan habilidades para programar o resolver problemas, no están institucionalizadas[2]. En este marco advertimos la necesidad de articulación entre el nivel de educación secundaria y la universidad. Este artículo describe la incorporación de una herramienta didáctica, llamada RITA (Robot Inventor To Teach Algorithms) en el contexto del proyecto “Articular universidad-escuela con JAVA para fortalecer la Educación-Técnica. Conectar Saberes”[3].

RITA nos permite acercar a estudiantes de escuelas secundarias a la programación desde un enfoque innovador. La programación contribuye al “pensamiento computacional”[4] facilitando el aprendizaje de estrategias de resolución de problemas y de diseño, que es posible trasladar a áreas disciplinares diferentes a las de específicas de Informática[5]. El “pensamiento computacional” promueve el pensamiento

analítico, sistemático, fomenta la creatividad y el trabajo colaborativo, todas ellas habilidades consideradas fundamentales para la sociedad del siglo 21.

El presente trabajo se organiza de la siguiente manera: primero se sintetizan las actividades desarrolladas en la primera etapa del proyecto tendientes a capacitación docente y articulación de contenidos entre los distintos niveles educativos, luego se presentan RITA, la herramienta didáctica usada para la articulación y, las actividades piloto llevadas a cabo con alumnos utilizando esta herramienta. El artículo finaliza con conclusiones y trabajos futuros.

## II. EL PROYECTO ARTICULAR UNIVERSIDAD-ESCUELA CON JAVA PARA FORTALECER LA EDUCACIÓN-TÉCNICA. CONECTAR SABERES

El programa “Conectar Igualdad” es una iniciativa del poder ejecutivo nacional argentino, enfocado en recuperar y valorizar la educación pública con el fin de reducir las brechas digital, educativa y social en el territorio argentino. Consiste en la entrega durante 3 años, de 3 millones de netbooks para estudiantes y docentes de escuelas públicas de educación secundaria de todo el país (durante el año 2012 fueron entregadas más de 2 millones). Adicionalmente el programa desarrolla contenidos digitales didácticos y trabaja en procesos de formación docente. Sin embargo, en lo referente a la enseñanza de “Programación” hay mucha diversidad y los planes de capacitación no llegan a todos los ámbitos educativos de la misma manera.

El programa “Conectar Igualdad” propone entre sus objetivos “formar sujetos capaces de transformar constructivamente su entorno”. Para lograr esta meta, “los medios informáticos deben entrar en la escuela para contribuir con la renovación y mejora de la educación, introduciendo prácticas pedagógicas innovadoras que estimulen la creatividad y la imaginación de los chicos”[6]. Es sabido que los estudiantes de hoy son nativos digitales, haciendo alusión a su fluidez digital o habilidad para manejarse con las tecnologías digitales, pero no muchos son capaces de diseñar, crear e inventar con los nuevos medios. La enseñanza de “programación” cumple un rol central como promotor de estas habilidades.

En este contexto, estamos presentando dentro del proyecto piloto “Articular universidad-escuela con JAVA para fortalecer la Educación-Técnica. Conectar Saberes”, a RITA, una herramienta didáctica y pedagógica específica para la enseñanza de programación para estudiantes de escuelas secundarias. Este proyecto, al que por simplicidad llamamos JET (JAVA en Escuelas Técnicas), es un proyecto de intervención de la Facultad de Informática de la Universidad Nacional de La Plata, iniciado en marzo de 2012, que actualmente trabaja con tres escuelas secundarias técnicas de la provincia de Buenos Aires. Esta intervención internivel trabaja en 3 ejes estructurales: 1) articular contenidos y metodologías de enseñanza entre ambos niveles educativos, 2) actualizar en nuevos paradigmas de programación y contenidos propios del nivel universitario a la comunidad docente de las escuelas intervinientes y, 3) aumentar las expectativas de los estudiantes secundarios desarrollando competencias que

favorezcan su inserción y permanencia en la universidad y que respondan a las necesidades del sector de “Software y Servicios Informáticos” de la región.

Los dos primeros ejes se trabajaron durante el año 2012: el eje referido a la articulación, no sólo abordó aspectos de contenido, sino que trabajó en un sentido más amplio e inclusivo, proponiendo la incorporación de nuevas metodologías de enseñanza; el eje de actualización docente se está llevando a cabo desde marzo de 2012 dándole al proyecto continuidad, fuerza y promoviendo lazos de confianza. Las actividades de actualización docente, están organizadas en encuentros de capacitación técnica en programación orientada a objetos y JAVA. De ellas participan docentes de escuelas secundarias y docentes universitarios. Los docentes de la Facultad de Informática que conforman este proyecto, son los contenidos e instructores de los encuentros de actualización docente. El contenido está organizado en 30 capítulos que incluyen contenido teórico, práctico y talleres integradores y está disponible en la plataforma de gestión de cursos virtuales de la Facultad de Informática (<http://cursos.linti.unlp.edu.ar>) basado en la herramienta Moodle (<http://moodle.org>). Este contenido es el resultado de años de selección de temas, confección, uso y evaluación en diferentes ámbitos educativos[7].

El tercer eje, relacionado con mejorar las aptitudes de los estudiantes, trabajará directamente en el aula para estimular la curiosidad, generar espacios exploratorios, colaborativos y experimentales que favorezcan a la inserción en la vida universitaria y en el mundo laboral. Para alcanzar estos objetivos, desde el proyecto JET, se propone el uso de RITA en las netbooks del programa “Conectar-Igualdad” aportando contenido didáctico en el área de “programación”, estimulando el uso de las netbooks con fines educativos y ampliando los límites del aula. Es sabido que el programa “Conectar-Igualdad” dispone de contenidos didácticos para áreas como matemática, física, lengua, sin embargo las propuestas para un tema novedoso en la escuela como es la “programación”, son escasas. RITA es una propuesta sustentada en juegos motivadores para adolescentes y jóvenes “nativos digitales”. La posibilidad de usar RITA en las netbooks, permite a los alumnos no sólo hacer sus prácticas de programación en los horarios pre-establecidos en la escuela sino también en sus hogares o en dónde lo deseen, ampliando las posibilidades de aprender.

Cabe destacar que el equipo de trabajo de este proyecto es multidisciplinario, lo integran docentes, alumnos y graduados de la Facultad de Informática, especialistas en educación y una docente de la Facultad de Periodismo y Comunicación Social de la UNLP. De esta manera es posible abordar la práctica educativa, tal que como plantea Freire en donde “la complejidad de la práctica educativa es tal, que nos plantean la necesidad de considerar todos los elementos que puedan conducir a un buen proceso educativo, nos impone la necesidad de inventar situaciones creadoras de saberes, sin las cuales la práctica educativa auténtica no podría darse”[8].

Como se mencionó anteriormente, durante el año 2012 se trabajó intensamente con los docentes en su actualización sobre contenidos propios del paradigma orientado a objetos y su

aplicación en JAVA. Asimismo se realizaron jornadas en las que se discutió sobre la didáctica a aplicar en el aula y las herramientas a emplear. De estas jornadas se concluyó que era necesario contar con una herramienta didáctica que permita acercar a los alumnos a la programación desde un enfoque motivador, para lo cual se propuso RITA para el Ciclo Básico de la escuela secundaria. En el caso de las escuelas técnicas, cuyos últimos años están enfocados en las materias de la especialidad, se consensuó en el uso de una herramienta de desarrollo de software de uso profesional, proponiéndose Eclipse[9], un entorno de desarrollo actual de código fuente abierto, ampliamente usado en la comunidad de desarrollo de software.

Luego de la primera etapa de trabajo con los docentes en 2012, en 2013 se comenzó con la bajada de los nuevos contenidos y metodologías al aula de la escuela secundaria. En los cursos del Ciclo Básico de dos de las escuelas intervinientes del proyecto se está utilizando RITA como herramienta didáctica de acercamiento a la "programación". La matrícula total de estos cursos asciende a 92 estudiantes. Asimismo en el Ciclo de Especialidad, se comenzó a trabajar con los nuevos contenidos, con una matrícula de aproximadamente 50 estudiantes.

### III. RITA: ROBOT INVENTOR TO TEACH ALGORITHMS

#### A. ¿Qué es RITA?

RITA (Robot Inventor to Teach Algorithms) es una aplicación JAVA basada en los frameworks de código fuente abierto: *OpenBlocks*[10] y *Robocode*[11]. Mediante la programación en bloques provista por *OpenBlocks*, RITA permite definir estrategias de combate entre robots virtuales situados en el campo de batalla de *Robocode*. RITA es el resultado de la tesina de grado de la Licenciada Vanessa Aybar Rosales, coautora de este artículo (<http://jets.linti.unlp.edu.ar/rita>). Esta tesina consistió en adaptar y extender las funcionalidades brindadas por *Openblocks* para que provea soporte a las clases JAVA provistas por *Robocode*, así como también a las clases JAVA estándares (API JAVA) de modo que el usuario de RITA pueda mediante bloques escribir la estrategia de un robot.

*OpenBlocks* es un framework que permite crear programas mediante la metáfora visual de arrastre de bloques y conectores que deben encastrarse. Los bloques representan abstracciones de las partes de un programa. Esta metodología alienta la creación de programas y promueve el pensamiento sistémico y analítico, sin necesidad de conocer las particularidades de un lenguaje de programación. *OpenBlocks* es distribuido por el Massachusetts Institute of Technology's Scheller Teacher Education Program (MIT-STEP) y surge como tesis de maestría de Roque Ricarose.

Por otro lado, *Robocode* es un juego de programación cuyo objetivo es programar la estrategia de un robot para competir contra otros robots en un campo de batalla. El jugador es el programador del robot y mediante el código que escribe le brindará inteligencia al robot, indicando cómo comportarse y reaccionar frente a eventos ocurridos en el juego. El nombre *Robocode* es una abreviación para "Robot code", en castellano

"codificar un robot". El framework *Robocode* está escrito en JAVA y el comportamiento de los robots también se programa en este lenguaje de programación. Asimismo *Robocode* provee un campo de batalla, donde pequeños robots compiten hasta que sólo uno queda victorioso en la arena. Este framework requiere que el programador tenga conocimientos de metodologías orientada a objetos y JAVA. Cabe aclarar que, si bien el framework *Robocode* propone una batalla, no promueve la violencia porque no se involucran personas, no contiene sangre ni se plantean situaciones de enfrentamiento racial; el objetivo de *Robocode* es promover la competencia de las estrategias implementadas por cada jugador en un sentido positivo

La Fig. 1 muestra esquemáticamente el uso de los frameworks que RITA integra: *Openblock* para programar los robots y *Robocode* para ejecutar en un campo de batalla. Sintéticamente *Openblock* provee el ambiente de programación y *Robocode* el de ejecución.

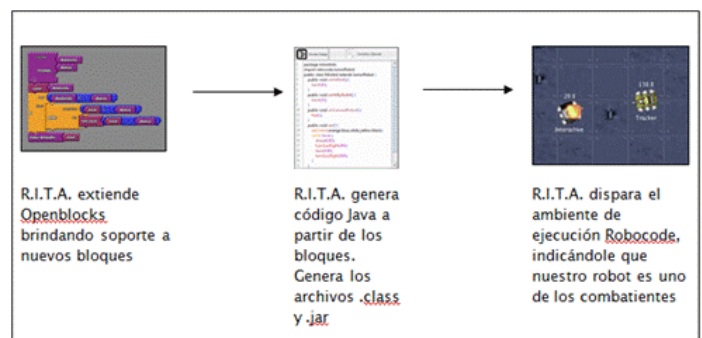


Fig. 1. OpenBlock y Robocode en RITA

RITA además de integrar *OpenBlocks* y *Robocode*, adaptó y extendió las funcionalidades de ambos frameworks.

En el caso de *Openblocks* se incorporaron nuevos bloques que representan la funcionalidad de un tipo particular de robot en *Robocode*, el *JuniorRobot*, y características que ayudan a la programación y aportan *feedbacks* para el jugador:

- Bloques para definir métodos propios del jugador agrupados bajo la categoría "Método".
- Bloques referidos a eventos ocurridos en la batalla, al arranque del robot, a su desplazamiento y acciones en el campo de batalla. Los métodos que implementan las reacciones del robot ante determinados eventos son: `onHitByBullet()`, `onHitWall()`, `onScannedRobot()`, `onHitRobot()`. El método que funciona como punto de entrada para la ejecución de un robot es el `run()`. Los métodos que modelan las acciones y desplazamientos son: `fire()`, `turnGunLeft(i)`, `back(i)`, etc. La Fig. 5 muestra los bloques que permiten programar eventos, mover y arrancar robots.
- Bloques que permiten obtener información actualizada acerca del estado del *JuniorRobot*: información de la energía actual del robot y su posición en el campo de batalla (`energy`, `robotX`, `robotY`, etc.). Estos bloques están disponibles bajo la categoría "Información".

Asimismo RITA implementó funcionalidad para usar bloques que al encastrarse puedan adaptar su forma de manera dinámica (encastre polimórfico) y validaciones en el tipo de dato de retorno de un método y en los nombres de los bloques (de modo que no se permitan bloques con nombre duplicado). Se incorporaron nuevos sonidos y mensajes de información y error, que funcionan como *feedbacks* para el jugador.

Por último, RITA provee funcionalidades no provistas en *OpenBlocks* ni en *Robocode*, entre ellas la traducción de la programación en bloques a código JAVA, permitiendo visualizar el código JAVA generado automáticamente, la posibilidad de especificar el nivel de complejidad de la batalla o seleccionar los robots que competirán en el campo de batalla. Además, RITA incorpora soporte para internacionalización de modo que pueda ser traducido a otros idiomas.

### B. ¿Cómo programar con RITA?

RITA permite crear estrategias de robots simples, conocidos como *JuniorRobot* dentro de las categorías de robots que pueden construirse con *Robocode*. Este tipo de robot es el más sencillo y recomendado en el proyecto *Robocode* para principiantes en programación de robots.

Los robots del tipo *JuniorRobot* permiten poner en práctica los conceptos fundamentales de programación que los alumnos deben aprender al hacer sus primeros pasos en algorítmica. Un *JuniorRobot* mantiene información básica del estado de un robot y además reacciona a eventos que ocurran sobre él o en su entorno.

Al ejecutar RITA, a través de una ventana de diálogo se le solicita al jugador el ingreso del nombre del Robot a construir. La Fig. 2 muestra esta interacción inicial.



Fig. 2. Ventana de ingreso de nombre del robot

La Fig. 3 muestra la pantalla principal de RITA, fundamentalmente formada por dos secciones:

- A la izquierda una sección de *bloques disponibles*.
- A la derecha el *área de trabajo* que contiene la estrategia del Robot. El área de trabajo contiene algunos bloques ya dispuestos y organizados de modo que el jugador cuente con un modelo básico de cómo se estructurará su aplicación usando los bloques.

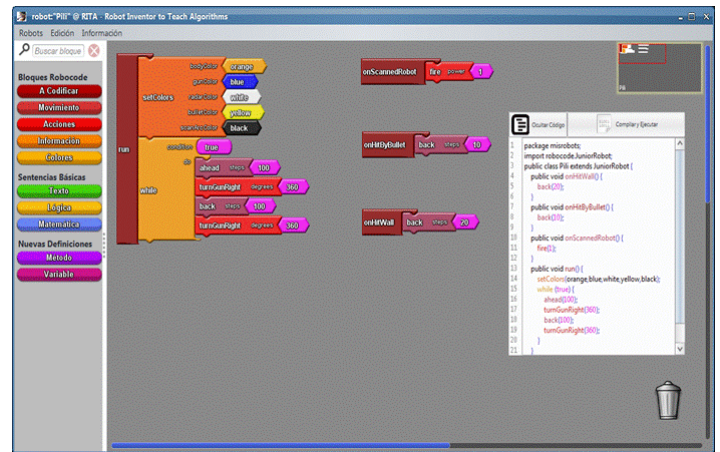


Fig. 3. Pantalla principal de RITA

La sección de *bloques disponibles* mostrada en la Fig. 4, ilustra las tres categorías de *bloques* usados para construir la estrategia del robot:

- *Bloques Robocode*: estos bloques se refieren a métodos e información que brinda la clase *JuniorRobot*. La Fig. 5 muestra los bloques referidos a eventos, movimientos y acciones de un *JuniorRobot*.
- *Sentencias Básicas*: estos bloques se refieren a sentencias que normalmente son encontradas en cualquier lenguaje de programación, como estructuras de control, operaciones lógicas y operaciones matemáticas.
- *Nuevas definiciones*: estos bloques permiten definir métodos para agrupar bloques que podremos invocar desde cualquier otro método del área de trabajo y también nos permiten definir variables propias para guardar valores que nos resulten interesantes para nuestra estrategia. Estos bloques promuevan la modularización de la estrategia.



Fig. 4. Categorías de bloques en RITA



Fig. 5. Bloques de las categorías “A Codificar”, “Movimiento”, “Acciones”

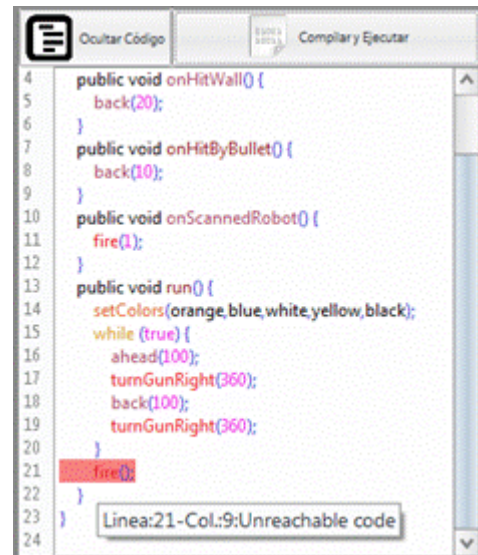


Fig. 6. Panel de código JAVA

La sección de *área de trabajo* contiene la estrategia del Robot programada en bloques y un panel con el código JAVA traducido por RITA automáticamente a partir de esta estrategia. En la Fig. 3 es posible observar este panel de código fuente JAVA, cuya principal función es además de visualizar el código generado, “Compilar y Ejecutar” nuestra estrategia.

Teniendo en cuenta que RITA es concebida como una herramienta que promueve la enseñanza de la programación en jóvenes estudiantes secundarios, la notificación de los errores ocurridos durante la programación fue realizada en forma incremental, a medida que el jugador incorpora bloques en su estrategia, de manera de acompañar y alentar la programación. A pesar de esto, no es posible asegurar con un 100% de certeza que todo lo programado en bloques será correcto. Hay ciertos errores que recién en compilación son evidentes. Al momento de compilar (botón “*Compilar y Ejecutar*”) se detectan los errores de compilación que no pudieron detectarse previamente y se le notifica al jugador mediante un mensaje de error. Asimismo en el panel de código fuente JAVA se indican en color rojo las líneas erróneas. La Fig. 6 muestra el panel con el código JAVA resultante, en el que es posible observar el error de compilación de la línea 21 junto con un mensaje. Asimismo en este panel los nombres de los métodos, estructuras de control, variables, etc mantienen el mismo color del bloque a partir del que se genera el código fuente JAVA.

Como puede observarse la creación de la estrategia de un robot en RITA, no requiere manejar un lenguaje de programación, alcanza con conocer las particularidades de un robot *JuniorRobot*, qué sabe hacer un robot y poner en práctica conceptos básicos de algorítmica. A continuación se describirá la ejecución del robot.

### C. ¿Cómo ejecutar un Robot en RITA para que compita con otros robots?

Para ejecutar un Robot en RITA el jugador debe seleccionar los robots adversarios que competirán con él en el campo de batalla. Esta interacción se realiza mediante una ventana de diálogo, en la que que es posible elegir los robots disponibles. Como muestra la Fig. 7, una vez seleccionados los robots adversarios comienza la competencia en el campo de batalla de *Robocode*. En la parte derecha de pantalla del campo de batalla el jugador puede observar el estado actual de los robots que compiten (energía, vidas, rondas, etc).

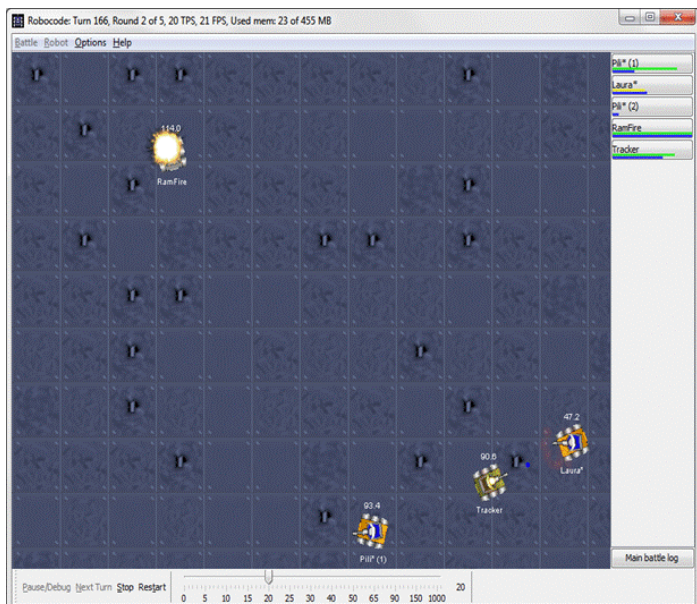


Fig. 7. Campo de batalla de Robocode ejecutado desde RITA

La batalla entre robots, le permite al jugador evaluar cómo se desarrolló su robot en el campo de batalla, evaluar su estrategia y qué tan exitosa resultó. Si el jugador lo desea puede mejorar la estrategia del robot, para ello volverá al editor de bloques de RITA. Aquí queda evidente la existencia de dos entornos de trabajo: uno de edición y otro de ejecución.

#### D. Pruebas de campo y resultados

En el proyecto “Articular universidad-escuela con JAVA para fortalecer la Educación-Técnica. Conectar Saberes” se realizaron cuatro pruebas piloto de RITA con estudiantes: el primer encuentro se desarrolló en septiembre de 2012 en la Facultad de Informática y participaron alumnos de las tres escuelas intervinientes, los tres encuentros restantes se realizaron entre octubre y noviembre de 2012 en cada una de las escuelas, con participación de alumnos de diferentes ciclos (formación básica y especialización). Para todos los encuentros, los docentes de las escuelas seleccionaron perfiles de estudiantes con diferentes características y edades. Los estudiantes participantes de esta experiencia no contaban con conocimientos previos de JAVA, no habían usado RITA ni habían programado con herramientas basadas en bloques.

En el primer encuentro se convocaron sólo a 15 alumnos, con la intención de tener una primera respuesta de los mismos con la herramienta RITA. La experiencia en este primer encuentro fue muy buena, esto nos alentó a ampliar el número de participantes en las restantes pruebas piloto que ascendieron a un total de 116 estudiantes. En todos los encuentros los estudiantes fueron acompañados por docentes e instructores integrantes del proyecto.

La metodología utilizada en los encuentros fue la siguiente:

- Explicación introductoria de Robocode y uso de RITA con un desafío individual guiado. Duración: 50 minutos.

- Desafío grupal, donde cada grupo (2-3 alumnos) creó la estrategia de su robot. Los alumnos no fueron guiados. Los instructores contestaban las preguntas de los alumnos. Duración: 50 minutos.

- Actividad final: puesta en combate de los robots generados por los alumnos.

Para evaluar el impacto de la herramienta en el aula, el grado de aceptación por parte de los alumnos y la posibilidad de su inclusión como herramienta didáctica en el aula, se suministró una encuesta con 6 preguntas a todos los participantes. A continuación se analizan los datos recolectados en las encuestas y se interpretan los resultados.

La Fig. 8 muestra que la explicación proporcionada sobre RITA, permitió que el 93% de los alumnos pudieran construir un robot con una estrategia, utilizando programación con bloques, en el tiempo estipulado.



Fig. 8. Explicación de RITA

La Fig. 9 describe la percepción por parte de los estudiantes del grado de dificultad del uso de RITA para cumplir con el desafío. Durante la experiencia, el 97% de los estudiantes consideró que el uso de RITA no presentó dificultad. Solamente un 3% de los alumnos respondió que les resultó difícil trabajar con RITA.

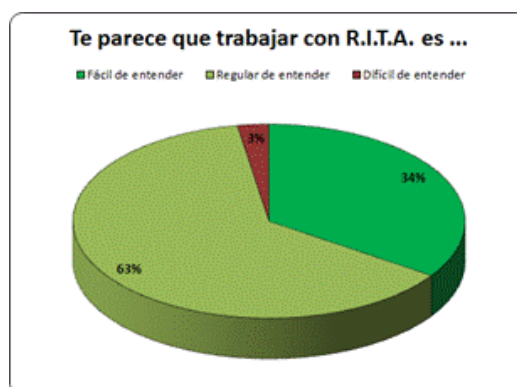


Fig. 9. Grado de dificultad de trabajar con RITA

La Fig. 10 muestra la opinión de los alumnos sobre programar usando un lenguaje de programación tradicional o programar con bloques. A pesar de que la mayoría de los estudiantes tenían experiencia con algún lenguaje de programación que no es JAVA, el 91 % de los estudiantes opinó que les resultó más simple programar con bloques.

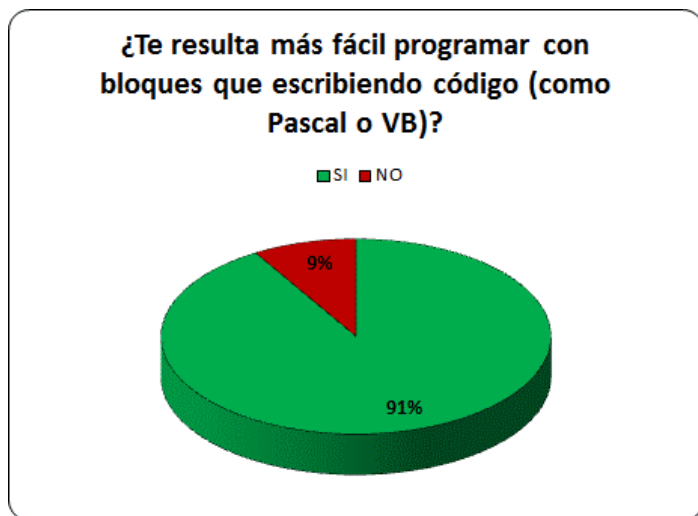


Fig. 10. Grado de aceptación de la programación en bloques

La Fig. 11 muestra la capacidad de combate del robot diseñado para el desafío propuesto. El 73% de los alumnos pudo construir un robot capaz de combatir; el 23% que no lo hizo explicó que con más tiempo de preparación para la estrategia se hubieran animado a participar de la batalla. El restante 4% no respondió la pregunta.

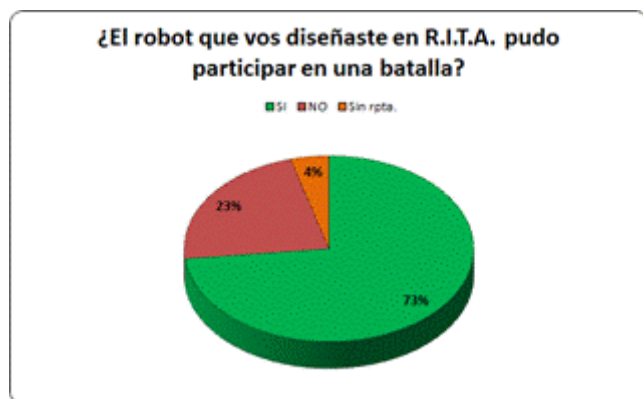


Fig. 11. Capacidad de combate del los robots generados

Ante la pregunta sobre la dificultad para entender el código JAVA generado, la Fig. 12 muestra que el 77% de los alumnos respondieron que les resultó fácil comprender el código JAVA a partir de los bloques, al 20% le resultó difícil y el 3% restante consideró mediana la dificultad.

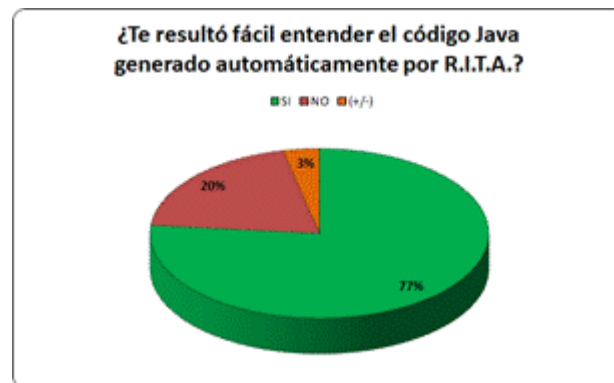


Fig. 12. Grado de comprensión del código JAVA generado

Aunque no estuvo contemplado en la encuesta, durante la prueba piloto los estudiantes indagaron a los instructores y docentes sobre el código JAVA generado por RITA, exponiendo dudas y elaborando sus propias conclusiones acerca del lenguaje JAVA.

Los resultados expuestos en este apartado, dan cuenta de que RITA es una herramienta didáctica que estimula la creación de programas, es apta para estudiantes sin experiencia en programación y es ampliamente aceptada por “nativos digitales”.

## CONCLUSIONES

Existen múltiples iniciativas para fomentar el estudio de carreras universitarias de informática en Argentina, algunas de ellas enfocadas en la generación de juegos, animaciones y simulaciones, todas ellas motivantes para adolescentes y jóvenes “nativos digitales”. El proyecto JET es un aporte al programa nacional “Conectar-Igualdad” proponiendo entre otras cosas a RITA, una herramienta didáctica que estimula el aprendizaje de “programación”. RITA además de acercar a los jóvenes a la “programación”, va más allá, haciendo foco en un paradigma de programación actual y en JAVA, un lenguaje de programación ampliamente aceptado en los distintos ámbitos profesionales. Usar RITA en las netbooks de “Conectar-Igualdad” amplía los límites del aula, promoviendo el aprendizaje fuera de la escuela.

En todas las disciplinas incluir contenidos y actividades que resulten motivadoras para los alumnos resulta la tarea más difícil para el equipo docente. En el área de “programación”, esto puede resultar más complejo si se tiene en cuenta que se trata con conceptos muy abstractos y que la mayoría de los jóvenes tienen ideas preconcebidas sobre las personas que programan (es muy común encontrarse con comentarios como que los programadores son “nerds”, “aburridos”, que se aíslan, etc.). El uso de herramientas didácticas como RITA, permite no sólo enseñar los conceptos de “programación” sino abordarlos desde una perspectiva lúdica y social.

El plan nacional “Conectar-Igualdad”, que incorpora masivamente las TICs en el aula, pone en el centro de la escena el debate y la reflexión acerca de la inclusión y el uso de las tecnologías digitales en los distintos niveles de enseñanza. El proyecto JET además promueve el diseño y la creación con las

tecnologías digitales e incorpora un contenido nuevo, “programación”, en el aula de la educación secundaria.

Entendemos que la intervención planteada en este proyecto de articulación, impactará positivamente debido a que su pilar fundamental es la interacción entre docentes de la escuela y la universidad. Esto permitirá fortalecer a las dos instituciones, promoverá la articulación y facilitará el tránsito a la universidad de los jóvenes estudiantes.

#### REFERENCIAS

- [1] Secretaría de Políticas Universitarias, Ministerio de Educación de la Nación Argentina: <http://portales.educacion.gov.ar/spu/becas-universitarias/>
- [2] Claudia Queiruga, Laura Fava. Articular Universidad Escuela: una experiencia de intervención de la Facultad de Informática de la UNLP. 41 JAIIO, SSI 2013 - 11° Simposio sobre la Sociedad de la Información.
- [3] Sitio web oficial del proyecto Articular universidad-escuela con JAVA para fortalecer la Educación-Técnica. Conectar Saberes: <http://jets.linti.unlp.edu.ar/>
- [4] Jeannette Wing. Computational thinking. Communications of ACM. Vol 49 N° 3, Mar. 2006, 33–35.
- [5] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: Programming for All. Communications of the ACM. Vol. 52 N° 11, November 2009, 60-67.
- [6] Diego Levis. Hablar con el teclado. El habla escrita del chat (y de otros mensajes escritos con computadoras y celulares). Razón y Palabra, N° 53, Instituto Tecnológico de Monterrey, 2006/2007.
- [7] Gabriel Baum, Javier Díaz, Claudia Queiruga, et. all. Actualizando la Enseñanza de Informática en las Escuelas Secundarias Técnicas de la Provincia de Buenos Aires. 38 JAIIO, ISSN 1850-2776, Simposio de la Sociedad de la Información. 2009.
- [8] Paulo Freire. Pedagogía de la autonomía. Ed. Paz e Terra, Sao Paulo, 2004.
- [9] Sitio web oficial de Eclipse: <http://www.eclipse.org/>
- [10] *Open Blocks*, Java library for creating blocks-based programming, download page, accesibe en <http://education.mit.edu/openblocks> [accedido el 9 de abril de 2013]
- [11] *Robocode*, accesible en: <http://robocode.sourceforge.net/> [accedido el 9 de abril de 2013].