

# Los Juegos como Herramienta Didáctica para la Enseñanza de la Programación

Verónica M. Torres, M. Fernanda Villarrubia, Roberto D. Lamas, C. Marcelo Pérez Ibarra  
*Facultad de Ingeniería*  
*Universidad Nacional de Jujuy*  
*San Salvador de Jujuy, 4600*  
*torres.vmt@gmail.com, marvillisi@gmail.com, robertolamas1960@gmail.com,*  
*cmperezi@gmail.com*

## Abstract

*En este trabajo se presenta la experiencia pedagógica realizada por docentes de un curso de fundamentos de programación. Esta experiencia, basada en juegos serios, se llevó a cabo persiguiendo 3 objetivos: 1) afianzar los conocimientos del alumno, 2) motivar al alumno a participar de forma activa en su proceso de aprendizaje, 3) incentivar la creatividad del alumno. Para ello, utilizando el entorno Scratch, se propuso “desarrollar juegos” como actividad final de las 3 primeras unidades temáticas del curso. La participación de casi la totalidad de los alumnos, un gran número de proyectos bien desarrollados y un notable incremento de los accesos al aula virtual de la asignatura demostraron el éxito de la propuesta.*

## 1. Introducción

En las carreras de informática, principalmente las orientadas al desarrollo de software, los alumnos de los primeros años presentan una problemática común: la comprensión de los conceptos asociados a la programación y el desarrollo del pensamiento computacional. Esto, según describen algunos autores, puede estar relacionado a la carencia de habilidades cognitivas (abstracción, capacidad de análisis y capacidad de síntesis) asociadas a la resolución de problemas, aspecto que resulta esencial al momento de formular los pasos de un algoritmo. Así podrían explicarse las dificultades que los alumnos tienen para entender un problema, identificar los datos relevantes y establecer una secuencia lógica de acciones que permitan alcanzar una solución. Sin embargo, éste no es el único factor a considerar en la problemática de la enseñanza de la programación. Las estrategias pedagógicas clásicas, como revelan varias experiencias, no logran despertar el interés y la motivación de los alumnos de los cursos de

programación ya que las actividades propuestas no reflejan para éstos resultados apreciables y/o atractivos.

Existen diferentes propuestas pedagógicas, realizadas por docentes e investigadores, que abordan las dificultades vinculadas a la enseñanza de la programación. Estas propuestas se enfocan en la manera de presentar los contenidos conceptuales y la utilización de recursos que permitan potenciar las habilidades intrínsecas de los alumnos como así también motivarlos para aplicar y afianzar los conocimientos adquiridos.

En este trabajo se describe una experiencia realizada con alumnos de un curso de programación que utilizaron una plataforma de juegos para realizar una actividad de afianzamiento de conocimientos. Además, se reportan los resultados obtenidos de dicha actividad.

El resto del documento se estructura de la siguiente forma: en el apartado 2 se presenta la problemática de la enseñanza de la programación; en el apartado 3 se describen propuestas para el tratamiento de esta problemática; en el apartado 4 se resumen algunas herramientas de programación de juegos; en el apartado 5 se describe la experiencia realizada con alumnos; y finalmente, en el apartado 6 se indican las conclusiones del trabajo.

## 2. Problemática

La programación según lo expresa Joyanes Aguilar [1] consiste en transformar una secuencia de pasos que resuelven un problema (algoritmo) en un programa de computadora que automatice el proceso de resolución. En un sentido más amplio, la construcción de programas implica que el programador sea capaz de analizar el problema a resolver, diseñar los pasos de la solución y finalmente transformar este diseño en un programa de computadora.

Con esta visión, los cursos introductorios a la programación no sólo están destinados a enseñar un

lenguaje sino que apuntan a desarrollar habilidades que permitan a los alumnos entender un problema, proponer soluciones, utilizar un lenguaje de programación y comprobar la corrección de las soluciones planteadas [2]. Sin embargo, esta tarea no resulta sencilla y diversos estudios concluyen en posibles factores que inciden en el aprendizaje de los fundamentos de la programación. Estos factores se asocian tanto a las características del alumno como a las estrategias pedagógicas seguidas por los docentes [3-7]:

- bajo nivel de las habilidades educativas del alumno
- falta de una didáctica adecuada a los nuevos conceptos de la materia
- el tiempo que se destina a la clase
- complejidad de la sintaxis del lenguaje y los conceptos de la programación

Insuasti [5] focaliza la dificultad del aprendizaje de la programación en 4 factores:

1. mecanismos frágiles en la creación de programas de computadora, en particular el uso de la sintaxis de los lenguajes de programación;
2. incapacidad para ver el resultado de los cálculos a la par cuando un programa de computadora se ejecuta;
3. falta de motivación para la programación, la dificultad de comprensión de la lógica compuesta,
4. desconocimiento de las técnicas de diseño.

Un estudio realizado en la Universidad Francisco de Paula Santander [3] intenta identificar las causas de los problemas de aprendizaje de los alumnos que cursan materias de programación. El estudio reveló deficiencias en las habilidades para razonar, tomar decisiones, resolver problemas y afrontar situaciones de la vida cotidiana, académica y profesional.

Las altas tasas de desaprobados y los elevados índices de deserción de los cursos de programación constituyen desde hace años una preocupación para la comunidad académica. Es por ello que con el tiempo se han propuesto variedad de estrategias para tratar esta problemática [2]. Algunas de estas propuestas se describen a continuación.

### **3. Estrategias para la enseñanza de la programación**

A partir de la problemática identificada en el apartado anterior se realizaron diversas propuestas que intentan mitigar las dificultades de aprendizaje de los alumnos de programación. Las soluciones más simples como cambiar de libro, utilizar otro lenguaje de programación, cambiar el orden de los temas, simplificar los enunciados de los

ejercicios, etc., no consiguen mejoras efectivas sin importar cómo se apliquen o combinen [2]. Consecuentemente, docentes e investigadores, intentan formular propuestas pedagógicas sólidas que apunten no sólo a los contenidos curriculares sino también al desarrollo de las habilidades cognitivas del alumno.

En [8] Moroni y Señas proponen incorporar un editor de algoritmos (con funciones de corrección semántica y sintáctica y detección de errores) que permita a los alumnos concentrarse en la lógica del programa independientemente de las características de un lenguaje específico.

En [3] Bohórquez-Chacón y Amaya-Torrado proponen un modelo pedagógico, sustentado en las nuevas tecnologías de la información y la comunicación, que promueve la participación activa del alumno a través de la interacción en un espacio virtual, facilitando el trabajo colaborativo y la construcción del conocimiento. Asimismo, prevé la utilización de herramientas de evaluación y seguimiento de los alumnos.

En [6] Rosanigo y Paur proponen un método de enseñanza basado en problemas. Éste consiste en el planteo de problemas cuya resolución produce el aprendizaje, lo que requiere de la participación activa del alumno. En este enfoque la obtención del resultado no es la finalidad, sino la comprensión del proceso mental que permite alcanzarlo.

En [9] Sarria y Barrios proponen utilizar, como soporte y extensión del dictado presencial, aulas virtuales. Estos espacios virtuales permiten la comunicación tanto sincrónica como asincrónica entre docentes y alumnos.

En [10] Arellano y colaboradores proponen aplicar un nuevo enfoque pedagógico: la clase invertida. Un enfoque de clase invertida implica combinar la clase presencial con estrategias que promuevan la construcción del aprendizaje a partir de actividades de investigación del propio alumno, con el seguimiento y guía del docente. Así se intenta dar soporte a todas las fases del ciclo de aprendizaje, motivando al alumno a ser parte activa del proceso.

El equipo de cátedra, cuya experiencia se describe en el presente trabajo, implementó a lo largo de los últimos 8 años los distintos enfoques aquí presentados sin que se lograran mejoras significativas.

Una propuesta más reciente, la enseñanza de la programación basada en juegos serios, plantea una interesante alternativa de aprendizaje que promueve la comprensión y aplicación de los conceptos básicos de la programación a través de la construcción de juegos.

Los juegos serios son herramientas que tienen como objetivo principal el aprendizaje, planteado desde una perspectiva atractiva y divertida para el alumno. El potencial de los juegos serios para mejorar la eficacia de la enseñanza y el aprendizaje ha sido un tema de debate

los últimos años. Muchas investigaciones afirman que los juegos serios reúnen todas las características necesarias para convertirse en una herramienta educativa capaz de conducir a un aprendizaje efectivo [11-14].

En [15] Willging y colaboradores proponen utilizar robots educativos como recurso de apoyo al aprendizaje e instrumento de motivación de los alumnos. Estos robots pueden controlarse a través de entornos de programación y sensores. La experiencia descrita en este artículo se realiza sobre 2 grupos de alumnos que cursan fundamentos de programación. Un grupo recibe las clases de la forma tradicional mientras que el otro aplica un enfoque basado en juegos (Lightbot, code.org y Scratch)

Insuasti hace referencia a un estudio [5] donde sugiere el uso de ‘Máquinas de Estados Finitos’ para la enseñanza de los fundamentos de programación, y señala que “*se debe ver a la programación practicada como un ejercicio educativo, libre de la preocupación utilitaria es mejor aprendida en un ambiente de juego, diseñada para ilustrar los conceptos seleccionados en la configuración más simple*”. Visto de este modo, es posible introducir los fundamentos de programación para principiantes en un entorno de juego, donde a través de acciones limitadas es posible aprender a controlar rutinas simples tales como condicionales, ciclos e iteraciones. El propósito del uso de las máquinas de estado finito y los juegos de azar, es estimular el aprendizaje de conceptos a través la lúdica y la motivación.

En el siguiente apartado se describen algunas herramientas que permiten implementar un enfoque de juegos serios para el aprendizaje de la programación.

#### 4. Entornos para la programación de juegos

Si bien al momento de programar existen múltiples IDEs (ambientes de desarrollo integrado) que presentan una gran variedad de herramientas para facilitar la tarea del programador, estos ambientes de desarrollo no están diseñados con propósitos educativos. No obstante, en consideración a las dificultades de aprendizaje que presentan los estudiantes de programación y atendiendo a la demanda de mayor motivación por parte del estudiantado, se han desarrollado entornos que permiten “aprender a programar”.

A continuación se describen 8 herramientas destinadas al aprendizaje de los conceptos fundamentales de la programación [16-17]:

- *CodeCombat* es un videojuego (multijugador) en línea. Se trata de una plataforma para que los estudiantes aprendan ciencias de la computación mientras juegan un juego real. Los estudiantes escriben código y ven los cambios en tiempo real. La plataforma está disponible en <https://codecombat.com/>

- *CodinGame* es una plataforma para crear código de forma divertida. El sitio web oficial cuenta con más de 50 juegos que incluyen tutoriales para aprender su funcionamiento; además tiene tres niveles: principiante, medio y avanzado. También proporciona una sección para optimizar el código de algunos juegos y siete juegos multijugador. La plataforma está disponible en <https://www.codingame.com/>
- *Code & Conquer* pone en juego la estrategia. El jugador debe programar las reacciones de su personaje y competir con los personajes de otros jugadores. El juego ofrece tutoriales en forma de retos. Cada desafío introduce nuevos conceptos para mejorar las habilidades de codificación y aprender nuevos algoritmos. La plataforma está disponible en <http://www.codeandconquer.co/>
- En *Checkio*, el jugador se une a un mundo creado por una comunidad de programadores. En ella, los programadores se unen y pueden desarrollar nuevos niveles para que otros los puedan probar. Para ello, es necesario tener nociones básicas de programación porque incluso los primeros ejercicios son algo avanzados. Muchos de los juegos están traducidos al español. La plataforma está disponible en <https://checkio.org/>
- *CodeWars* es un juego para desarrolladores que ya tienen conocimientos de programación, pero quieren mejorar. La idea es competir contra otros y así ganar honor y niveles, pero al mismo tiempo la comunidad servirá para comparar el código creado con el de otros y así aprender diferentes soluciones a los problemas que se presentan. La plataforma está disponible en <http://www.codewars.com/>
- *CodeHunt* es un juego creado por Microsoft. El jugador deberá detectar fragmentos de código que han desaparecido y sustituirlos con su propia solución. Así el jugador aprenderá sobre variables, algoritmos de búsqueda, operadores aritméticos y otras cosas más. La plataforma está disponible en <https://www.codehunt.com/>
- *Alice* es un software educativo libre con el que los estudiantes pueden aprender desde las bases de programación, lógica y matemática computacional básica a la resolución de problemas en un entorno de programación 3D muy interesante a la hora de crear animaciones y juegos. La plataforma está disponible en <http://www.alice.org/index.php>
- *Scratch* es un proyecto del grupo *Lifelong Kindergarten* en el MIT Media Lab., tiene su propio lenguaje y es de uso sencillo. Los estudiantes pueden utilizar Scratch para codificar sus propias historias interactivas, animaciones y

juegos. En el proceso aprenden a pensar creativamente, razonar sistemáticamente y trabajar de manera colaborativa. La plataforma está disponible en <https://scratch.mit.edu/>

Además de estas herramientas pueden mencionarse otras similares tales como *Hackety Hack*, *Light-Bot*, *Logo*, *Simple*, *Stencyl*, *Sodaplay*, *Game Maker* y *Codeacademy*.

En la siguiente sección se describe la experiencia realizada por el equipo de cátedra al incorporar en el dictado 2016 un enfoque de aprendizaje basado en juegos serios utilizando la herramienta Scratch.

## 5. Experiencia

### Descripción de la materia

Programación Estructurada es una asignatura de primer año de la carrera Analista Programador Universitario que se dicta en la Facultad de Ingeniería de la Universidad Nacional de Jujuy, se ubica en el Área de Tecnologías Básicas y tiene un régimen cuatrimestral con una carga horaria total de 90 hs.

La asignatura, objeto de la experiencia, comprende en sus contenidos los conceptos fundamentales de la programación. Ésta, corresponde al primer año de una carrera universitaria orientada a la formación de programadores, con un régimen cuatrimestral y una carga horaria total de 90 hs.

La asignatura tiene por objetivo lograr las siguientes competencias en el alumno:

- Desarrollar en el alumno el razonamiento lógico para el análisis y resolución de problemas mediante algoritmos.
- Desarrollar en el alumno el pensamiento abstracto de modo que pueda enfocarse en aspectos específicos al momento de resolver un problema.
- Diseñar algoritmos utilizando las estructuras de control básicas y tipos de datos simples.
- Modelar problemas sencillos identificando sub-problemas independientes.
- Codificar programas en lenguajes de alto nivel.
- Comprender y aplicar las fases de desarrollo de programas.

Los contenidos de la materia se organizan en 6 unidades temáticas ordenadas de tal forma que el aprendizaje de los conceptos fundamentales de la programación sea un proceso incremental.

Respecto a la modalidad de enseñanza, el equipo de cátedra dicta clases presenciales teóricas, prácticas y de consulta, utilizando un espacio virtual para la comunicación sincrónica y asincrónica con los alumnos. A continuación se describe la modalidad adoptada:

- Clases teóricas: se trata de clases expositivas acerca de los contenidos de la materia que se

ilustran a través del planteamiento de ejemplos y la enunciación de los problemas más comunes que suelen presentarse y cómo resolverlos.

- Clases prácticas: se trata de clases destinadas a la resolución de trabajos prácticos, ya sea en forma individual o grupal, y la explicación y resolución de problemas tipo. Se incentiva a los alumnos a participar en la resolución de los ejercicios, discutiendo las soluciones planteadas. Los ejercicios propuestos en los trabajos prácticos están enfocados en ejercicios básicos de matemática y geometría a fin de simplificar su complejidad.
- Clases de consulta: se trata de clases de apoyo al dictado que se realizan en horarios extra clase. Las consultas pueden realizarse tanto de forma personal como virtual (uso de la plataforma del aula).
- Aula Virtual: la plataforma virtual está destinada a facilitar los recursos pedagógicos de la materia (presentaciones teóricas, trabajos prácticos, videos de la cátedra), como diferentes recursos útiles para contribuir a la comprensión de los conceptos de la materia (publicaciones científicas, videos, tutoriales, etc.). Otro recurso importante es el foro (el aula cuenta con foros de consulta de cada unidad temática destinados a responder inquietudes de los alumnos).

### Descripción de la propuesta

Con el objetivo de afianzar los conceptos fundamentales (algoritmo y programa; fases genéricas de desarrollo de programas; tipos de datos simples, operaciones elementales, variables y constantes; herramientas de modelado y estructuras de control secuenciales, selectivas y repetitivas) desarrollados en las 3 primeras unidades de la materia y teniendo en cuenta que ellos proporcionan las herramientas básicas para comenzar a programar, se consideró oportuno realizar una actividad que motivara a los alumnos a poner en práctica lo aprendido.

La actividad propuesta, basada en juegos serios, se formuló sobre la plataforma Scratch considerando lo siguiente: es un entorno de programación en español que posee una interfaz sencilla basada en elementos gráficos y que no requiere conocimientos específicos acerca de lenguajes de programación; siendo además de uso libre.

El planteo de la actividad consideró 3 propósitos: 1) mitigar las dificultades que presentan los alumnos en la comprensión de los conceptos básicos de programación, 2) aumentar la motivación de los alumnos para realizar las actividades y ejercicios que aplican estos conceptos y 3) incentivar la creatividad en los alumnos

desestructurando las clases propuestas de forma tradicional.

### Planteo de la actividad

Considerando la variedad de proyectos que pueden realizarse en Scratch, la actividad propuesta por la cátedra contempló 5 alternativas: 1) animaciones, 2) historias, 3) animaciones interactivas, 4) problemas extraídos de los trabajos prácticos y 5) juegos.

Se solicitó a los alumnos elegir alguno de los trabajos propuestos y desarrollarlo en Scratch. El desarrollo debía ser individual, salvo en el caso de los videos juegos en el que se permitió trabajar en grupo (3 integrantes como máximo). La presentación de la actividad se realizó en el aula virtual, en el foro habilitado en la sección “Actividades con Scratch”. Aquí los alumnos debían especificar el nombre de los integrantes del equipo (si correspondía) y una breve reseña del trabajo realizado. El trabajo podía adjuntarse como un archivo .sb o .sb2 o bien trabajarse en línea para luego compartirse en la página oficial de Scratch cuyo enlace debía indicarse en el foro de la actividad. El plazo para el desarrollo y publicación de la actividad fue de una semana. La actividad inició el lunes 18 de abril y finalizó el lunes 25 de abril. Durante esa semana las clases prácticas, de las diferentes comisiones, se destinaron a atender consultas específicas acerca del uso de Scratch.

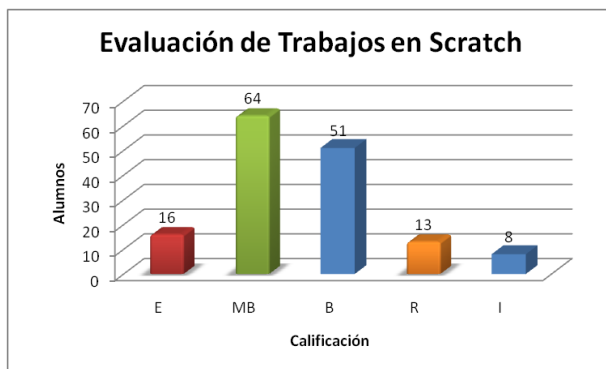
### Discusión de Resultados

Al momento de lanzar la actividad de Scratch la materia registraba 160 alumnos activos. El 95% de este grupo participó efectivamente la actividad, generándose como resultado 105 proyectos (entre presentaciones individuales y grupales). Como ya se explicó, los alumnos podían elegir el tipo de proyecto a presentar, siendo los juegos los únicos trabajos que podían desarrollarse en grupo. La Tabla 1 cuantifica, por tipo, los proyectos presentados. Puede observarse que el 53% de los alumnos optó por desarrollar juegos (81) y que en la mayoría de los casos la presentación se realizó de forma individual (54). Cabe destacar que el desarrollo de juegos fue el tipo de proyecto de mayor complejidad.

**Tabla 1. Proyectos presentados para la actividad de Scratch.**

Tipo de Proyectos	Proyecto	Cantidad
No Interactivas	Animaciones	11
	Historias	2
Interactivas	Animaciones interactivas	3
	Juegos (27 grupales y 54 individuales)	81
	Ejercicios de los trabajos prácticos	8

Los trabajos fueron evaluados y calificados en función de la complejidad y correcta aplicación de los conceptos de programación. En la Figura 1 se muestran las calificaciones obtenidas por los alumnos en la actividad Scratch: 16 alumnos fueron calificados con EXCELENTE (E), 64 con MUY BUENO (MB), 51 con BUENO (B), 13 con REGULAR (R) y 8 con INSUFICIENTE (I). Como puede observarse el 52% de los alumnos fueron calificados como excelentes o muy buenos, el 33% como buenos y el 13% como regulares o insuficientes (proyectos que reflejaban poco o nulo esfuerzo).



**Figura 1. Evaluación de proyectos realizados en Scratch.**

Atendiendo al segundo propósito de la actividad, se evaluó la participación de los alumnos en todos los foros del aula virtual. Como puede observarse en Tabla 2, el foro correspondiente a la actividad de Scratch tiene 5615 accesos (temas, intervenciones, vistas), mientras que los otros foros están por debajo de los 1000 accesos.

**Tabla 2. Acceso de los alumnos a los diferentes foros del aula virtual de la materia.**

Foros	Temas/Intervenciones/Vistas
Introducción a la Programación: Dudas y consultas	10/23/873 (906)
Tipos de Datos: Dudas y consultas	6/21/885 (912)
Actividades con Scratch	108/235/5272 (5615)
Estructuras de Control: Dudas y consultas	7/16/791 (814)
Resolviendo el Trabajo Práctico N° 4 - Codificación en C++	8/13/746 (767)
Modularidad: Dudas y consultas	6/12/654 (672)
Arreglos: Dudas y consultas	6/12/352 (370)
Registros y Archivos: Dudas y consultas	2/2/107 (111)

En la Figura 2 se muestran los niveles de participación de los alumnos en el aula virtual. Puede apreciarse que la

mayor actividad en el aula se registró durante el desarrollo de los proyectos Scratch, superando incluso a las actividades obligatorias como los cuestionarios por unidad temática.

La Figura 3 muestra un fragmento del registro del aula virtual en el que puede observarse los accesos al foro de la actividad Scratch y a los juegos publicados.



Figura 2. Participación de los alumnos en el aula virtual de la materia.

Programación Estructurada - San Salvador: Todos los participantes, Todos los días (Hora local del servidor)

Programación Estructurada - San Salvador ▾ Todos los participantes ▾ Todos los días ▾ Actividades con Scratch ▾ Todas las acciones ▾

Mostrar en página ▾ Conseguir estos registros

Mostrando 5615 registros

Página: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 ...56 (Siguiente)

Fecha	Dirección IP	Nombre completo	Acción	Información
lun 8 de agosto de 2016, 11:37	190.52.34.142	Veronica Torres	forum view forum	Actividades con Scratch
lun 8 de agosto de 2016, 11:35	190.52.34.142	Veronica Torres	forum view forum	Actividades con Scratch
lun 8 de agosto de 2016, 08:32	170.210.241.133	Pablo Gonzalez	forum view discussion	JUEGO CON SCRATCH
lun 8 de agosto de 2016, 08:32	170.210.241.133	Pablo Gonzalez	forum view discussion	TP Scratch
lun 8 de agosto de 2016, 08:32	170.210.241.133	Pablo Gonzalez	forum view discussion	ACTIVIDAD CON SCRATCH
lun 8 de agosto de 2016, 08:32	170.210.241.133	Pablo Gonzalez	forum view forum	Actividades con Scratch
lun 8 de agosto de 2016, 08:32	170.210.241.133	Pablo Gonzalez	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
lun 8 de agosto de 2016, 08:06	170.210.241.133	Pablo Gonzalez	forum view forum	Actividades con Scratch
lun 8 de agosto de 2016, 08:04	170.210.241.133	Pablo Gonzalez	forum view discussion	TP Scratch
lun 8 de agosto de 2016, 08:01	170.210.241.133	Pablo Gonzalez	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
lun 8 de agosto de 2016, 04:41	190.185.204.2	Matias Miguel Serapio	forum view discussion	TP Scratch
lun 8 de agosto de 2016, 04:31	190.185.204.2	Matias Miguel Serapio	forum view forum	Actividades con Scratch
lun 8 de agosto de 2016, 04:31	190.185.204.2	Matias Miguel Serapio	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
dom 7 de agosto de 2016, 23:42	181.177.25.230	Joaquín Daniel Peña Alvarado	forum view discussion	TP Scratch
dom 7 de agosto de 2016, 23:37	181.177.25.230	Joaquín Daniel Peña Alvarado	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
dom 7 de agosto de 2016, 20:13	181.1.0.137	Segovia Gonzalo Fabian	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
dom 7 de agosto de 2016, 18:20	190.137.53.190	Flores Mariela Eida	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
dom 7 de agosto de 2016, 17:30	201.140.225.155	Juan José Saravia	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
dom 7 de agosto de 2016, 14:18	186.190.155.253	Gaston Enzo Toconas	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
dom 7 de agosto de 2016, 14:18	186.190.155.253	Gaston Enzo Toconas	forum view forum	Actividades con Scratch
dom 7 de agosto de 2016, 13:50	179.40.86.1	Emilce Torrez	forum view discussion	ENLACE DEL JUEGO DEL TRABAJO FINAL
dom 7 de agosto de 2016, 13:49	179.40.86.1	Emilce Torrez	forum view forum	Actividades con Scratch

Figura 3. Registro de accesos al foro de la actividad Scratch.

La Tabla 2 y la Figura 2 evidencian un notable cambio de actitud de los alumnos. No sólo ingresaron con mayor frecuencia al aula virtual, también participaron en el foro de la actividad a través de consultas, comentarios y aportes al trabajo de sus pares. Como ilustran las Figuras

4a y 4b, los alumnos “evaluaron” el trabajo de sus compañeros, accediendo a los juegos publicados, este registro permite observar como la coevaluación surgió como estrategia de la didáctica de forma espontánea entre pares (Figura 5) a través de sus enlaces.

 **El gato y El ratón**  
de David Alejandro Largo - viernes, 22 de abril de 2016, 23:07

Hola aquí dejo mi pequeño proyecto. Es sencillo y no se aplica muchas operaciones.  
Objetivo: Comer por lo menos tres porciones de queso, para poder ganar... Evitando ser atrapado por el gato y ser tocado por las barras mas de 1 vez.

<https://scratch.mit.edu/projects/106638763/>

Editar | Borrar | Responder

---

 **Re: El gato y El ratón**  
de María Fernanda Villarrubia - sábado, 23 de abril de 2016, 13:14

Muy bueno!!!

Mostrar mensaje anterior | Editar | Partir | Borrar | Responder

---

 **Re: El gato y El ratón**  
de fabio Choque - domingo, 24 de abril de 2016, 01:38

muy bueno..no lo pude pasar XD!

Mostrar mensaje anterior | Editar | Partir | Borrar | Responder

**Figura 4 a. Comentarios de los alumnos en el foro de la actividad Scratch**

 **Actividad en Scratch**  
de Emanuel Maldonado - viernes, 22 de abril de 2016, 20:09

Buenas Noches, aquí subo mi actividad de scratch, se trata de un juego donde hay que esquivar los rayos de la bruja, es bastante sencillo la verdad , quería implementar un sistema de niveles y cambio en la velocidad a medida que uno pasa de nivel pero ya se hacia mas complicado.

En fin aquí les dejo el link para el que quiera echarle un vistazo: <https://scratch.mit.edu/projects/106626680/>

Editar | Borrar | Responder

---

 **Re: Actividad en Scratch**  
de Matías Fernando Espinoza - sábado, 23 de abril de 2016, 00:21

e.e .. felicidades .. esta genial xD

Mostrar mensaje anterior | Editar | Partir | Borrar | Responder

---

 **Re: Actividad en Scratch**  
de María Fernanda Villarrubia - sábado, 23 de abril de 2016, 13:11

Muy bien!!!

Mostrar mensaje anterior | Editar | Partir | Borrar | Responder

---

 **Re: Actividad en Scratch**  
de fabio Choque - domingo, 24 de abril de 2016, 01:27

Esta muy bueno tu juego!! 😊

Mostrar mensaje anterior | Editar | Partir | Borrar | Responder

**Figura 4 b. Comentarios de los alumnos en el foro de la actividad Scratch**



**Figura 5. Juego, publicado en el sitio de Scratch, desarrollado para la actividad propuesta.**

Resumiendo los principales resultados de la experiencia:

- 95% de los alumnos participó de la actividad
- 52% de los trabajos fueron muy bien calificados
- notable participación de los alumnos en el aula virtual

Evidentemente estos resultados fueron, en comparación a la forma tradicional de dictar la materia, muy positivos ya que reflejan un mayor interés y compromiso de parte de los alumnos.

## 6. Conclusiones

En este trabajo se describe la experiencia llevada a cabo por docentes de un curso de introducción a la programación del primer año de una carrera universitaria orientada a la formación de programadores. Esta experiencia, planteada sobre el entorno de desarrollo de juegos Scratch, se realizó con 3 propósitos: 1) mitigar las dificultades que presentan los alumnos en la comprensión de los conceptos básicos de programación, 2) aumentar la motivación de los alumnos para realizar las actividades y ejercicios que aplican estos conceptos, 3) incentivar la creatividad en los alumnos desestructurando las clases propuestas de forma tradicional.

La actividad propuesta a los alumnos del curso consistió en el desarrollo de animaciones, historias o juegos en Scratch aplicando conceptos tales como contadores, banderas, estructuras de control, etc. Durante la realización de esta actividad se pudo observar:

- participación activa en el foro de la actividad a través de comentarios de los alumnos que probaron los juegos desarrollados por sus pares.
- participación pasiva en el foro de la actividad por parte de los alumnos que sólo accedieron a los juegos desarrollados por sus pares.
- asistencia a clases de consulta: los alumnos se acercaron a los horarios de consulta con animaciones y/o juegos casi terminados y con dudas puntuales.
- la mayoría de los trabajos presentados se realizaron de forma individual aportando ideas originales

Los resultados obtenidos reflejaron un evidente cambio de actitud de los alumnos. Se incrementó notablemente la participación en los horarios de consulta y las intervenciones en el aula virtual, y se logró la producción de trabajos originales que implementaban correctamente los conceptos estudiados.

El equipo de docentes involucrado, así como los alumnos del curso, mostraron gran interés en los recursos y resultados generados por Scratch, destacando la sencillez de la interfaz y la facilidad con la que se pueden obtener resultados visibles.

La experiencia demostró que un enfoque basado en juegos serios, instrumentado adecuadamente, puede impactar positivamente en la forma de enseñar y aprender programación. Además, se pudo observar un mayor compromiso y entusiasmo en la realización de las actividades por parte del estudiantado, sobre todo en las

consultas acerca de la comprensión de conceptos para la resolución de las actividades propuestas, para obtener los resultados esperados.

La situación expuesta permite al equipo docente replantear la propuesta de incorporar Juegos Serios de otro modo, quizás como punto de partida y andamiaje en la implementación de innovadores procesos didácticos a futuro.

## 7. Referencias

- [1] Joyanes Aguilar, L. (2003) Fundamentos de Programación. Algoritmos y estructuras de datos” 3ra Edición. ISBN 9788448136642. Editorial McGraw-Hill. España
- [2] Salcedo, J. A. V. (2009). Proyecto CUPI2 – Una solución integral al problema de enseñar y aprender a programar.
- [3] Bohórquez-Chacón, L. F., & Amaya-Torrado, Y. K. (2016). Diseño de un modelo pedagógico para la enseñanza de fundamentos de programación de computadores basado en el uso de la tecnología como mediación pedagógica. *Respuestas*, 10(1), 30-37.
- [4] Ruiz, J. G., López, M. H., & Brito, J. A. L. (2016). Pensamiento sistémico y desarrollo de competencias, en el aprendizaje de los lenguajes de programación. *Revista electrónica ANFEI Digital*. Año 1, N° 2, Ene-Jun.
- [5] Insuasti, J. (2016) Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Revista educación y desarrollo social*, 10 (2), 234-246. DOI: [org/10/18359/reds.1701](https://doi.org/10.18359/reds.1701)
- [6] Rosanigo, Z. B., & Paur, A. B. (2006). Estrategias para la enseñanza de Algorítmica y Programación. In I Congreso de Tecnología en Educación y Educación en Tecnología.
- [7] Artola, V., Sanz, C. V., Gorga, G., & Pesado, P. (2014). Diseño de un juego basado en Interacción Tangible para la enseñanza de Programación. In XX Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014).
- [8] Moroni, N., & Señas, P. (2005). Estrategias para la enseñanza de la programación. JEITICS 2005 - Primeras Jornadas de Educación en Informática y TICS en Argentina.
- [9] Sarria, W. J. & Barrios Robayo, M. J. (2015). Propuesta para la enseñanza de algoritmia y programación de computadores: Caso de Aula. *Revista Universitaria Experiencia Docente*. Vol 2 Nro 1 Enero-Junio 2015 ISSN 2346-3236. Bogotá Colombia
- [10] Arellano, N. M., Aguirre, J. F., & Rosas, M. V. (2015). Clase invertida: una experiencia en la enseñanza de la programación. In X Congreso sobre Tecnología en Educación & Educación en Tecnología (TE & ET)(Corrientes, 2015).
- [11] Martín, A. C. U., & Aznar, C. T. (2015). Juegos serios como instrumento facilitador del aprendizaje: evidencia empírica. *Biblioteca Digital Universidad de Zulia Opción*, Año 31, No. Especial 3 (2015): 1201 – 1220 ISSN 1012-1587.
- [12] Marcano, Beatriz (2008). Juegos serios y entrenamiento en la sociedad digital. En Sánchez I Peris, Francesc J. (Coord.) *Videojuegos: una herramienta educativa del “homo digitalis”* [monográfico en línea]. *Revista Electrónica Teoría de la Educación: Educación y Cultura en la Sociedad de la Información*. Vol. 9, n° 3. Universidad de Salamanca. ISSN: 1138-9737
- [13] Frittelli, V., Tartabini, M., Teicher, R., Steffolani, F., Serrano, D., Fernández, J., ... & Strub, A. (2013). Desarrollo de Juegos como Estrategia Didáctica en la Enseñanza de la Programación. CONAIISI 2013.
- [14] Muñoz, Carla & Valenzuela, Jorge (2014). Escala de Motivación por el Juego (EMJ): estudio del uso del juego en contextos educativos. *RELIEVE*, 20 (1), art. 4. DOI: 10.7203/relieve.20.1.3878
- [15] Willging, P., Astudillo, G. J., Bast, S. G., Francia, A., Ocelli, M., & Distel, J. M. (2015). Educación con tecnologías: herramientas y tendencias. XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015).
- [16] González, G. (2015). 6 juegos para aprender a programar. Publicado el 18/11/2015. Disponible en <http://blogthinkbig.com/6-juegos-para-aprender-a-programar/>
- [17] Bates, Laura (2016). 10 of the Best Online Programming Tools for Students. Disponible en <https://www.fractuslearning.com/2012/11/14/programming-tools-for-students/ling-measurement-in-objectoriented-systems>", *IEEE Transactions on Software Engineering*, 25, 1, January 1999, pp. 91-121.