

Modelado y simulación de un sistema mecánico simple: Programar en primer año de ingeniería

Natalia G. Monjelat ^a, Guillermo L. Rodríguez ^b & Patricia S. San Martín ^a

^a Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE:CONICET-UNR), Rosario, Argentina
monjelat@irice-conicet.gov.ar, sanmartin@irice-conicet.gov.ar

^b Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional de Rosario, Rosario, Argentina guille@fceia.unr.edu.ar

Resumen— El trabajo presenta un análisis exploratorio del proceso de construcción de simulaciones de un sistema mecánico utilizando entornos de programación con bloques por parte de alumnos universitarios. Los resultados provienen de un estudio de caso desarrollado en la asignatura de primer año "Introducción a la Ingeniería Mecánica" (Facultad de Ingeniería, UNR). Partiendo del modelado, los participantes diseñaron, programaron y evaluaron sus propias simulaciones utilizando Scratch, que resultó un recurso idóneo para ser utilizado con estudiantes con escasos conocimientos de programación. De esta forma, las simulaciones funcionaron como una herramienta de aprendizaje en una etapa temprana de la carrera. Los resultados reflejan la necesidad de propiciar propuestas educativas que permitan el estudio de los diversos aspectos del modelado, la simulación y la programación en carreras de ingeniería, ya que estas experiencias pueden facilitar el uso de entornos más específicos y potentes, siendo recursos valiosos para enriquecer situaciones curriculares y profesionales.

Palabras Clave— Enseñanza Universitaria; Programación; Modelado y simulación; Ingeniería Mecánica.

Recibido para revisar Abril 21 de 2017, aceptado Mayo 25 de 2017, versión final Agosto 4 de 2017

Modelling and simulation of a simple mechanical system: Programming in first year of engineering

Abstract— This work presents an exploratory analysis of the process of constructing simulations of a mechanical system using programming environments with blocks by university students. We present the results of a case study developed in the subject "Introduction to Mechanical Engineering" (Faculty of Engineering, UNR), from the first year of the course. Based on the modeling, the participants designed, programmed and evaluated their own simulations using Scratch, which proved to be an ideal resource to be used with students with limited programming knowledge. In this way, the simulations functioned as a learning tool in an early stage of the university. The results reflect the need to promote educational proposals that allow the study of the different aspects of modeling, simulation and programming in engineering careers, since these experiences can facilitate the use of increasingly specific and powerful environments, being valuable resources to enrich curricular and professional situations.

Keywords— Higher Education; Programming; Modeling and Simulation; Mechanical Engineering.

1. Introducción

Diferentes recursos tecnológicos se proponen actualmente en la enseñanza de carreras científico-tecnológicas de nivel

superior, dentro de los cuales se destacan las simulaciones por computadora. Unido a éstas, el desarrollo de los modelos físico-matemáticos subyacentes, son beneficiosos para ilustrar conceptos propios de las diversas asignaturas.

Particularmente en las carreras de Ingeniería, se han desarrollado múltiples experiencias que buscan introducir al alumnado en el mundo del modelado y la simulación [1,2], ya que estos saberes resultan un requisito fundamental para su desempeño en el ámbito académico, como así también en su vida laboral.

Por otra parte, crear simulaciones es una tarea compleja que implica la comprensión de conocimientos no solo de ingeniería, sino también de programación informática [3,4]. En este sentido, la mayoría de los alumnos que ingresan a la universidad carecen de los contenidos previos necesarios para abordar estas tareas y deben aprenderlos en un periodo corto de tiempo. Es por ello que un gran número de las experiencias realizadas en primer año apuntan exclusivamente al uso de simulaciones y están vinculadas principalmente a asignaturas centradas en la enseñanza de las ciencias [5-7]. Sin embargo, existen herramientas sencillas que permiten el aprendizaje de nociones básicas de programación y que podrían ser utilizadas para construir simulaciones y trabajar en el espacio universitario con diferentes modelos. En este sentido resulta necesario estudiar las posibilidades que ofrecen las herramientas disponibles para programar simulaciones atendiendo tanto a contenidos específicos como a las características particulares del alumnado.

Considerando estas cuestiones se presentan los resultados de una propuesta pedagógica innovadora realizada en la cátedra "Introducción a la Ingeniería Mecánica" (1º año, carrera Ingeniería Mecánica) de la Facultad de Ciencias Exactas, Ingeniería y Agrimensura de la Universidad Nacional de Rosario (UNR), donde se llevó adelante un proyecto con el objetivo de abordar el análisis de un sistema mecánico a través de la construcción de un modelo simplificado del mismo, construyendo su simulación con herramientas básicas de programación.

Como citar este artículo: Monjelat, N.G., Rodríguez, G.L. and San Martín, P.S., Modelado y simulación de un sistema mecánico simple: Programar en primer año de ingeniería. Educación en Ingeniería, 13(25), pp. 4-11, Febrero, 2018.

2. Construir modelos y simulaciones

Diversas investigaciones destacan el modelado y la simulación entre los recursos que más promueven el aprendizaje activo, crítico y reflexivo, estimulan la creatividad y permiten poner a prueba hipótesis [8,9].

Un modelo se define como la información relativa a un sistema, recabada a los fines de estudiarlo [10]. Esta información está determinada por los propósitos del estudio del mismo, por lo cual esto da lugar a diversos modelos según el enfoque y la profundidad del análisis. Sumado a esto, sabemos que una simulación es el proceso de desarrollar y programar un modelo de un sistema real o no, con el fin de estudiar el comportamiento del mismo en determinadas condiciones.

Por otra parte, Riesco y otros [11] señalan que mientras que hasta hace poco tiempo, el manejo de la informática consistía en el uso de aplicaciones, tanto la evolución del hardware como del software han provocado un cambio en el rol del usuario para pasar de un mero consumidor y operador, a sumar el rol de intervenir estas tecnologías por medio de las opciones que ofrecen para modificarlas y adaptarlas a sus necesidades.

En consonancia, Ingerman [12] define cuatro formas en que los estudiantes interactúan con las simulaciones:

- verlas como una tarea que debe completarse,
- verlas como una representación del fenómeno,
- manipularlas para entender cómo la simulación trabaja ó
- explorar el fenómeno mismo usando la simulación.

Las dos primeras apuntan esencialmente al uso de las simulaciones como recurso didáctico u objeto educativo digital. El tercero y el cuarto, señalan la interacción con la simulación a nivel programación. En este sentido, otros estudios señalan [13,14] que la experiencia de aprendizaje es más valiosa cuando el alumno no sólo usa la simulación, sino que también la diseña, construye, y tras evaluarla, puede modificarla en un proceso que implica la comprensión de los conceptos abarcados. De esta manera, investigaciones previas muestran la importancia de la metodología en espiral a lo largo del trayecto curricular de las carreras, donde los modelos desarrollados y las simulaciones programadas son retomadas con distintos niveles de profundidad [15,16].

Por otra parte, diferentes autores [17-20] nos hablan de las diversas competencias que los estudiantes logran a través de la programación de simulaciones. Estas incluyen: desarrollar el conocimiento de los estudiantes sobre paquetes de software determinados, fortalecer sus habilidades para resolver problemas, comprender los beneficios de las simulaciones como herramientas de aprendizaje, capacidad para transferir conocimientos adquiridos a otras áreas, aproximarse de manera compleja a la realidad, fomentar la interacción entre compañeros, y realizar tareas abiertas.

Considerando estas cuestiones, resulta fundamental proponer experiencias en este sentido realizando estudios de los procesos de construcción de simulaciones en una práctica inicial de primer año para la formación en ingeniería. En nuestro caso se planteó el análisis, construcción del modelo y programación de la simulación de una pelota pequeña rebotando sobre una superficie lisa. Para esto, se presentó el modelo físico matemático de caída libre y tiro vertical (no se tiene en cuenta el rozamiento del aire y por tanto se considera

la pelota como un punto material). Al contener en su descripción sólo los detalles relevantes para su estudio, este modelo es una simplificación del sistema al que hace referencia a través de las ecuaciones matemáticas (1) y (2):

$$vf = v_0 - g T \quad (1)$$

$$Y = v_0 T - \frac{1}{2} g T^2 \quad (2)$$

De esta forma, se propone simular el movimiento de la pelota insertando a través de la programación las ecuaciones antes citadas y agregando a su vez las condiciones de contorno necesarias.

3. Programar en los primeros años de la universidad

Actualmente se está potenciando la enseñanza de la programación y su uso como recurso educativo y como competencia general en distintos lugares del mundo. Diversos autores sostienen que la programación aporta herramientas para dejar de ser meros consumidores de software y comprender cómo funciona, mejorando diferentes habilidades [21,22]. Por esto se propone la enseñanza de la programación no sólo enfocado a la enseñanza de un lenguaje en particular sino al desarrollo del pensamiento computacional incluso como una habilidad básica en nuestra sociedad [23]. Sin embargo, el uso y entendimiento de los lenguajes de programación ha sido históricamente difícil, ya que implica comprender una sintaxis especializada que no admite errores [24]. La mayoría de los lenguajes de programación parecen una escritura extraña para el ojo no entrenado y es por ello que en los cursos introductorios de programación, se ha observado que los estudiantes tienen que llegar a ser especialistas de la sintaxis, antes de poder resolver los problemas [25].

Considerando este escenario, en los últimos años se han desarrollado diferentes propuestas que ofrecen entornos visuales de programación, que permiten a los usuarios centrarse en la semántica, antes que en la sintaxis. Entre las más destacadas se señalan Scratch, Alice, Blockly y Kodu, plataformas en las cuales es necesario arrastrar distintos bloques para generar código [26]. Asimismo, en este marco se destacan iniciativas como "Code.org" que ofrecen actividades de programación con bloques para diferentes usuarios en más de 20 idiomas dentro de la propuesta la "Hora del código", de la que ya han participado más de trescientos millones de personas (<https://ar.code.org/>).

En educación superior también se observa esta tendencia a emplear plataformas de programación con bloques, particularmente en los cursos universitarios iniciales. Aunque existen diferentes herramientas, la revisión de experiencias realizadas en educación superior permite identificar cierta tendencia favorable hacia el uso de Scratch, plataforma desarrollada por el grupo "Lifelong Kindergarten" perteneciente al MIT (Instituto Tecnológico de Massachusetts) [27]. Esta herramienta propone un lenguaje de programación basado en bloques agrupados en categorías que son utilizados para manipular los atributos de los objetos. Aunque fue creada para un público más joven, diferentes estudios señalan su

potencial en el ámbito universitario, particularmente en propuestas de enseñanza de programación para alumnos ingresantes. En este sentido los estudios previos destacan que trabajar con el encastre de bloques para armar los programas, permite que los estudiantes que se inician en la programación puedan concentrarse en la resolución del problema, incorporando la sintaxis en tanto arman los algoritmos, centrándose primero en los problemas lógicos [28-30]. Asimismo, las investigaciones coinciden en señalar esta herramienta como una puerta de entrada a lenguajes más complejos como Java, C o Pascal [31], destacando además el potencial que tiene Scratch para motivar a los estudiantes, permitiendo el acceso a nociones fundamentales de programación [25]. Sin embargo, existen otras herramientas también utilizadas con programadores novatos que permiten un acercamiento a lenguajes más avanzados (por ejemplo Alice, Karel the robot, etc.), pero los estudios en educación superior señalan la flexibilidad que ofrece Scratch para construir múltiples tipos de proyectos, así como su curva de aprendizaje no tan pronunciada como en otros casos [30]. En este sentido, se destaca la existencia de material de ayuda disponible (como wikis y tutoriales) y una extensa comunidad online, donde se cuenta con la posibilidad de observar por dentro los millones de proyectos disponibles en la versión online, que pueden ser tomados como referencia para realizar proyectos nuevos [27].

Por todo lo dicho, propusimos utilizar la plataforma Scratch para construir las simulaciones utilizando bloques, dado que posibilita una oportunidad para el análisis detallado de cómo los estudiantes pueden implementar los modelos ya desarrollados, en un entorno de manera sencilla y completa.

4. Metodología

El presente estudio se propone explorar el proceso de construcción de simulaciones de un sistema mecánico utilizando entornos de programación con bloques por parte de alumnos universitarios ingresantes. Considerando que este recorte en el campo resulta novedoso, se ha desarrollado un estudio de caso exploratorio [32] utilizando técnicas etnográficas [33,34], ya que permite obtener resultados preliminares que contribuyan al conocimiento sobre estas temáticas, que serán empleados para el diseño de futuras experiencias.

4.1 Contexto y participantes

Para los propósitos del estudio se diseñó un taller de seis encuentros dentro del espacio curricular Introducción a la Ingeniería Mecánica, en la carrera Ingeniería Mecánica de la Universidad Nacional de Rosario. Esta asignatura plantea perspectivas pedagógicas activas desde una modalidad de taller físico-virtual [35], abordando la práctica analítica sobre casos de la Ingeniería Mecánica, en el marco interactivo y comunicacional que brindan las Tecnologías de la Información y Comunicación (TIC), como apoyo a los procesos de enseñanza y de aprendizaje [36]. En este sentido, la práctica participante en las diversas actividades propuestas en función de los objetivos planteados, es constitutiva al propio contenido

de la formación que se fundamenta, adoptando una dinámica constructivista dialéctica [37].

En este marco, los desarrollos teóricos y prácticos se trabajan interrelacionando contenidos, promoviendo el aprendizaje colaborativo y la producción cooperativa. De esta forma, la reflexión y discusión de los distintos temas se construye a partir de exploraciones o producciones analíticas realizadas en pequeños grupos. Con este enfoque, la asignatura propone una totalidad de diez trabajos a lo largo del cuatrimestre, individuales y grupales. En este contexto, se propuso realizar la experiencia de construcción de simulaciones en el marco del trabajo final grupal, que busca tener un carácter integrador de las diversas temáticas y ser propositivo en cuanto a las habilidades y actitudes señaladas como necesarias en la labor del ingeniero.

En cuanto al sistema mecánico propuesto para simular, se optó por este ejemplo por tres razones: los conceptos necesarios para la construcción de la simulación son desarrollados en la asignatura Introducción a la Física, que se dicta en paralelo, fortaleciendo la articulación entre actividades curriculares; se contempló la facilidad en la realización experimental de la actividad (bajo costo y facilidad en la obtención de los materiales necesarios) y finalmente, este problema es retomado en tercer año con una complejidad mucho mayor, utilizando otro lenguaje de programación y simulación. De esta forma, se pretende generar una articulación longitudinal de la carrera, que plantea una visión espiralada respecto a la resolución de los problemas de ingeniería, logrando mejoras en las soluciones a partir de la profundidad técnica.

En esta experiencia inicial, se propuso utilizar la herramienta de programación Scratch, ya que una revisión sobre el tema destaca su empleo tanto con estudiantes de secundario como con estudiantes ingresantes, ofreciendo un entorno de bloques que puede adaptarse a las necesidades diversas del alumnado. Los bloques están organizados en diferentes categorías, que a su vez, incluyen otros bloques con múltiples opciones, tal como presenta la Fig. 1.

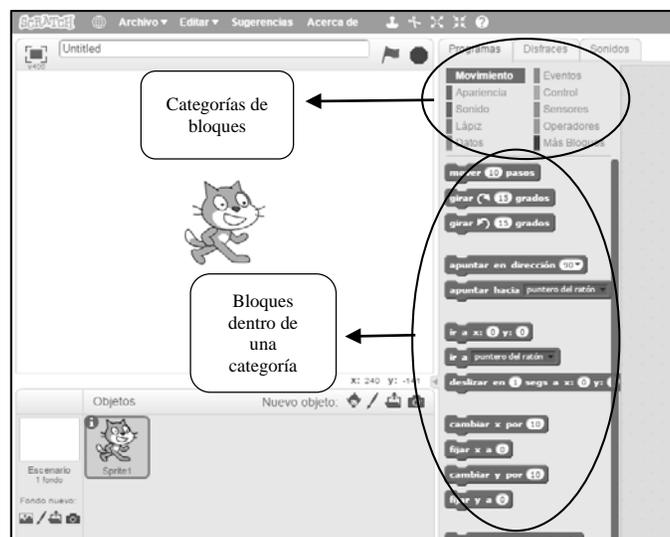


Figura 1. Captura de Pantalla portada de Proyecto Scratch.

Fuente: Los autores

A los fines del estudio, se trabajó con una muestra compuesta por estudiantes ingresantes del espacio curricular descripto. Particularmente, participaron los estudiantes de la comisión de la noche, conformada inicialmente por 19 estudiantes. Sin embargo, debido al desgranamiento propio del primer año participaron de la experiencia de manera completa 15 alumnos, edad media=19.33. El 40% de los estudiantes participantes contaba con formación secundaria técnica, mientras que el 86.7% desconocía Scratch. En cuanto a conocimientos previos de programación, el 53.3% no había programado previamente. Considerando estas cuestiones se propuso una conformación aleatoria de los grupos, constituyéndose seis grupos de entre dos y cuatro participantes. También participaron de la experiencia el docente de la cátedra y una investigadora.

4.2. Recolección y análisis de datos

A lo largo de la experiencia se recogieron diferentes tipos de datos en formatos diversos. Considerando la importancia de recopilar el proceso de construcción de la simulación, se grabaron en audio todos los encuentros presenciales permitiendo contar con datos sobre los problemas encontrados durante la construcción, las posibles soluciones propuestas y las justificaciones esgrimidas por los propios participantes al consultarles sobre las decisiones tomadas durante el proceso al presentar sus simulaciones al resto del grupo en la última clase.

Por otra parte, se recogieron los informes finales elaborados por los grupos así como las simulaciones creadas con la herramienta Scratch (las simulaciones realizadas fueron recopiladas en un estudio creado ad hoc: <https://scratch.mit.edu/studios/1976294/>). Para la elaboración de los informes se solicitó al alumnado seguir un protocolo específico de redacción planteado al comienzo de la actividad por la cátedra, que permite plasmar el proceso constructivo del modelo y de la simulación en su totalidad. Dicho protocolo recoge tanto la definición del problema como una copia del código final elaborado en Scratch, incluyendo a su vez la explicitación de dificultades encontradas a lo largo del trabajo y su resolución por parte del grupo.

Para el análisis de los datos se siguió el enfoque del estudio de caso [38] realizando un análisis de discurso [39] que permitió caracterizar el proceso de construcción de las simulaciones en un entorno de programación como Scratch. Para ello, todas las grabaciones en audio fueron procesadas con la ayuda de OTranscribe (<http://www.otranscribe.com>). Las transcripciones resultantes junto con los informes y las capturas de pantalla de las simulaciones se codificaron con la ayuda del programa Atlas ti (<http://atlasti.com/>). Los datos cuantitativos fueron procesados con la ayuda de SPSS (<http://www-03.ibm.com/software/products/es/spss-stats-base>).

5. Resultados y discusión

Un primer análisis de los datos registrados permitió identificar tres momentos diferenciados pero articulados, en cuanto al proceso de construcción de las simulaciones a lo largo de los seis encuentros que comprendió el taller.

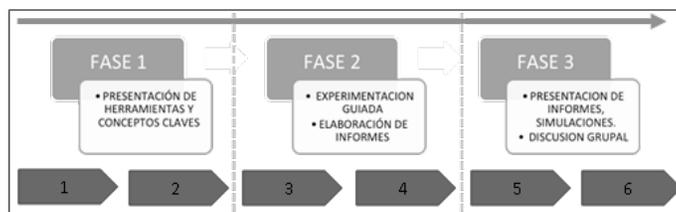


Figura 2. Ordenamiento de Fases de la Actividad.

Fuente: Los autores

Tabla 1
Distribución de bloques utilizados

	Cantidad de grupos	Porcentaje de casos
Eventos	6	100,0%
Variables	5	83,3%
Movimiento	6	100,0%
Apariencia	1	16,7%
Sensores	6	100,0%
Control	6	100,0%
Operadores	5	83,3%

Fuente: Los autores

Como se observa en la Fig. 2, durante los dos encuentros que comprendieron la primera fase, se retomaron conceptos de cinemática ya vertidos en la asignatura que se dicta de manera paralela en el mismo cuatrimestre denominada “Introducción a la Física”.

Por otra parte, se presentaron de manera dialogada las ecuaciones ya citadas, aplicándolas y resolviendo ejemplos. Asimismo se expuso de manera general el entorno Scratch (entorno, tipos de bloques, proyectos online, web, wiki, foros), y cada estudiante creó su cuenta de usuario. También se presentaron los tutoriales que aparecen en la web de la herramienta, se exploraron otros proyectos compartidos por la comunidad y se señaló la posibilidad de re-utilizar código presente en ellos.

Durante los encuentros de la segunda fase se retomaron las nociones básicas de programación (eventos, variables, iteraciones, etc.) y se conformaron los seis grupos definitivos de trabajo para la resolución del problema. A lo largo de los encuentros se discutieron colectivamente las posibles soluciones, los conceptos y ecuaciones utilizadas, generando un espacio para la consulta y resolución de dudas.

Finalmente, durante la tercera fase, los participantes se centraron en la preparación y presentación tanto del informe como de la simulación, que fueron presentados y discutidos en clase durante el último encuentro.

5.1. Análisis de las simulaciones

Los siete grupos de trabajo construyeron las simulaciones utilizando diferentes combinaciones de bloques para la programación del modelo físico matemático. Como se observa en la Tabla 1, el 100% de los grupos utilizó los bloques de eventos, movimiento, sensores y control. Las variables y los operadores representaron un 83,3%, que corresponde a 5 grupos. Por su parte los bloques de apariencia fueron empleados por un solo grupo, representando un 16,7% del total.

Respecto a los bloques utilizados por la mayoría, se observan diferencias en las elecciones dependiendo de la forma

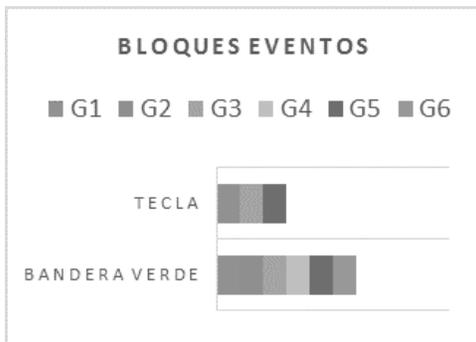


Figura 3. Bloques de Eventos utilizados
Fuente: Los autores

en que estructuraron la programación. En los siguientes gráficos se presenta un análisis pormenorizado de los bloques utilizados por cada grupo, agrupados bajo las categorías predeterminadas de Scratch, previamente presentadas.

Con respecto a la categoría “Eventos”, bloques que permiten iniciar secuencias de programación, se relevaron dos opciones escogidas por los participantes para iniciar sus programas.

Como se observa en la Fig. 3, todos los grupos emplearon el evento “Bandera verde” ya que este bloque es el encargado de iniciar todas las acciones dentro de Scratch. Tres grupos emplearon además un bloque alternativo: “Al presionar X tecla”, con diferentes finalidades. El grupo 1 complejizó su simulación agregando mayores funcionalidades, empleando la tecla espacio para observar la simulación en cámara lenta. El grupo 3 utilizó la tecla espacio para controlar el comienzo de la simulación, empleando la bandera solo para ubicar el objeto de manera inicial. Por su parte el grupo 5 utilizó la tecla espacio pero con la misma finalidad que en la bandera, sin aportar ninguna variación a la programación.

Otras categorías fundamentales para la construcción de la simulación han sido los “Bloques de Control”, que combinados principalmente con los “Bloques de Movimiento”, han permitido controlar la caída de la pelota y su rebote, a partir de condicionales y repeticiones.

Como refleja la Fig. 4, dentro de los bloques de control se observa por un lado el uso de condicionales “Si, sino” y “Si, entonces” o el uso de “Repetir hasta”, como dos opciones diferentes de controlar la secuencia de acciones. Asimismo, se observa una gran diferencia en cuanto a la cantidad de bloques de control empleados, algunos grupos usaron solo uno o dos bloques, mientras que otros emplearon hasta 4 bloques diferentes. Podría inferirse que a mayor cantidad de bloques, más completa la programación, sin embargo esto depende del tipo de bloques que acompañan a los de control.

En este sentido, los grupos que utilizaron los condicionales los combinaron con bloques específicos de movimiento tales como “Posición”, “Ir a X, Y”, “Deslizar a X, Y” y “Fijar X, Y”. Por su parte, los que utilizaron repeticiones operaron con otros bloques de movimiento como “Apuntar en dirección” e “Ir a X, Y”. De esta manera existen bien definidas dos propuestas de bucles para controlar el movimiento de la pelota.

Para reproducir el movimiento de la pelota y sus rebotes, los grupos tuvieron que asignar una posición inicial al objeto pelota, para que desde allí realizara su movimiento de rebote.

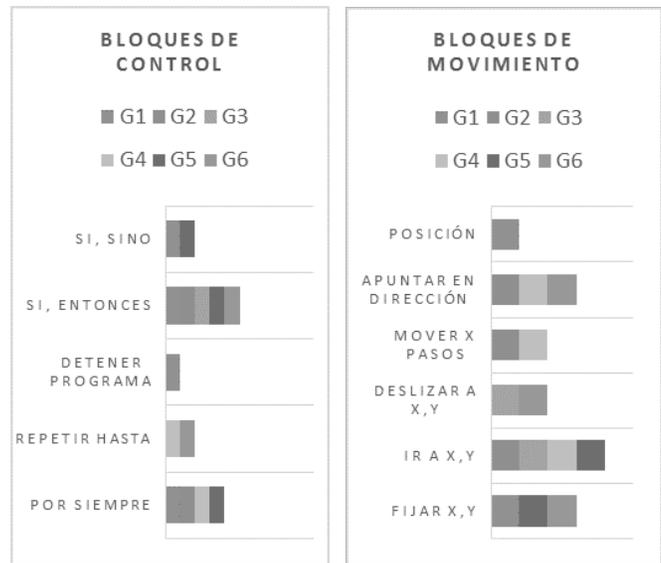


Figura 4. Bloques de Control y Movimiento
Fuente: Los autores

Para esto, el bloque más empleado ha sido “Ir a X, Y” que permite ubicar la posición de un objeto dentro del sistema de coordenadas que ofrece la herramienta. Los grupos que no recurrieron a este bloque utilizaron “Fijar X, Y”, en ambos casos bajo la bandera verde para obtener el mismo resultado.

Por otra parte, en lo que respecta al rebote, los grupos recurrieron a diferentes bloques: “mover una determinada cantidad de pasos”, “deslizar a X, Y” e incluso nuevamente “ir a X, Y”. El grupo 1 utilizó “fijar X, Y” tanto para la posición inicial como para el movimiento, pero en este segundo caso, el bloque se articula con otros de control, variables, operadores e incluso el bloque “posición en X, Y”.

Otro resultado que se desprende del análisis refiere al sistema de coordenadas que presenta Scratch, donde los ejes X-Y establecen el punto (0,0) en el centro de la pantalla. Esto generó dificultades para todos los grupos, forzando a un cambio de signo en las ecuaciones si se quería que la simulación estuviera centrada.

En cuanto al uso de “Bloques de Sensores”, todos los grupos generaron el movimiento a partir de una interacción detectada por un sensor, por lo cual estos bloques también fueron importantes para generar la simulación.

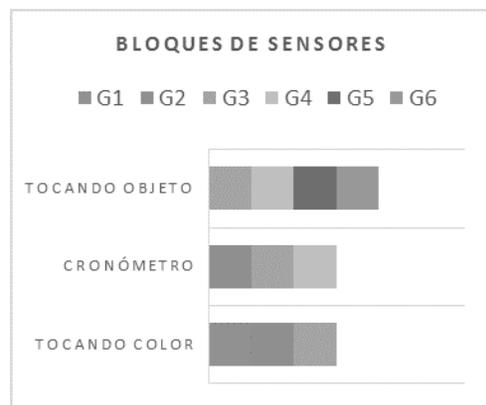


Figura 5. Bloques de Sensores
Fuente: Los autores

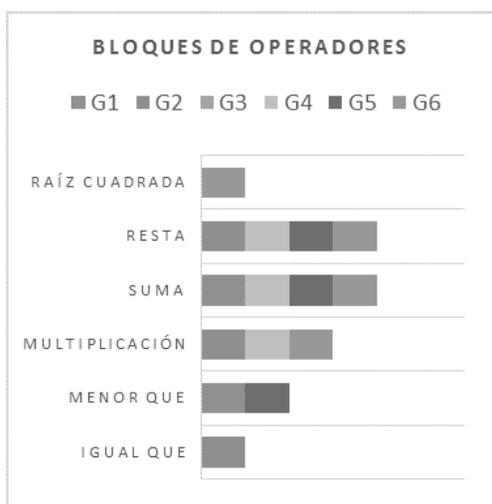


Figura 6. Bloques de Operadores
Fuente: Los autores

Los tipos de bloques empleados en todos los casos se incluyeron dentro de un bloque de control, ya que así lo requiere la programación, denotando esto con una forma particular. Como se observa en la Fig. 5, la mayoría de los grupos optó por crear un segundo objeto, empleando el bloque “Tocando objeto” y luego generar el rebote en base a una interacción entre este objeto y la pelota. Esta propuesta es la más coherente desde lo conceptual en vinculación al fenómeno real. Un solo grupo generó la interacción a partir del color, y otro, aunque creó el objeto, prefirió recurrir también al color. Con respecto a los demás sensores utilizados en los proyectos, se observa que tres grupos controlaron con ellos el tiempo de la caída y los rebotes utilizando el cronómetro.

Otra categoría que no podía faltar si se considera el problema a resolver, es la de “Bloques de Operadores”, ya que permiten introducir ecuaciones matemáticas dentro de la simulación.

En este caso la Fig. 6 presenta un uso diverso de las distintas operaciones, siendo la suma y la resta las más empleadas por los grupos. Aunque las fórmulas fueron las mismas, cada grupo las empleó de manera diferente, utilizando por lo tanto una combinación particular de las operaciones matemáticas disponibles. En este sentido, a la mayoría de los grupos les resultó difícil insertar la fórmula con sus respectivos resultados dentro del programa, ya que pese a comprender la ecuación y como ésta representaba la realidad, no lograban resolver cómo introducirla por medio de los bloques que presenta el programa. En este sentido, algunos estudiantes señalaron esto como una posible limitación en cuanto a la estructura cerrada de los bloques.

El análisis presentado refleja cómo los diferentes programas materializan el comportamiento modelado del sistema real, incluyendo ecuaciones matemáticas y combinaciones particulares de bloques, que muestran diferentes maneras de construir la simulación contando con las mismas herramientas. Las relaciones entre los bloques escogidos por los grupos responden por un lado a la lógica interna de programación y por otro, a las reglas propias del modelo físico matemático. De esta manera, los grupos lograron construir una simulación operativa

y compararla con los resultados reales. Por último cabe señalar que a partir de un marco de trabajo iterativo y colaborativo, aquellos grupos que tuvieron algunas dificultades en las distintas instancias del proceso, tuvieron la posibilidad de optimizar sus prototipos subsanando progresivamente las problemáticas observadas.

6. Conclusiones

Este estudio utilizó el marco del modelado, la simulación y la programación para la enseñanza de la ingeniería en primer año. Los resultados mostraron que la implementación de las simulaciones con bloques fue variada y correcta. La elección de los mismos fue coherente a la propuesta y las diferencias encontradas tuvieron que ver principalmente con decisiones adoptadas por cada grupo. Las dificultades observadas no impidieron la construcción del modelo y su ulterior simulación en Scratch respondiendo a la consigna planteada. En este sentido, la presente experiencia permitió no solo poner a prueba un modelo, sino permitir a los estudiantes el diseño, programación y evaluación de su simulación, apropiándose del contenido conceptual involucrado.

Como sostienen estudios previos ya citados, la posibilidad de construir modelos simplificados es algo fundamental vinculado a la enseñanza, siendo un proceso que puede desarrollarse de manera paulatina y con diferentes niveles de complejidad a medida que el estudiante recibe un mayor andamiaje conceptual. En este caso, los estudiantes emplearon una herramienta sencilla, que les permitió acercarse a nociones básicas de programación. De esta forma, las simulaciones funcionaron como una herramienta de aprendizaje en una etapa temprana de la carrera.

Esta experiencia previa podría facilitar el uso de entornos cada vez más específicos y potentes, siendo un recurso valioso para enriquecer situaciones curriculares. De esta forma, los resultados reflejan la necesidad de propiciar propuestas educativas que permitan el estudio de los diversos aspectos del modelado, la simulación y la programación en carreras de ingeniería.

Referencias

- [1] Fletcher, L.S., Mechanical engineering education for the 21st Century. *International Journal of Engineering Education*, 13(1), pp. 42-51, 1997.
- [2] Fraser, D., Pillay, R., Tjatindi, L. and Case, J., Enhancing the learning of fluid mechanics using computer simulations, *Journal of Engineering Education*, 96(4), pp. 381-388, 2007. DOI: 10.1002/j.2168-9830.2007.tb00946.x
- [3] Coppo, R., Cavallo, A. y Ursua, G., Sistema didáctico para la enseñanza de la programación con metodologías de aprendizaje basado en problemas. En XIII Workshop de Investigadores en Ciencias de la Computación, Rosario, 2011.
- [4] González, A. y Madoz, M.C., Utilización de TIC para el desarrollo de actividades colaborativas para la enseñanza de la programación, En VIII Congreso de Tecnología en Educación y Educación en Tecnología, Red de Universidades con Carreras en Informática, Santiago del Estero, 2013.
- [5] Zacharia, Z.C., Comparing and combining real and virtual experimentation: An effort to enhance students' conceptual understanding of electric circuits, *Journal of Computer Assisted Learning*, 23(2), pp. 120-132, 2007. DOI: 10.1111/j.1365-2729.2006.00215.x

- [6] Castellaro, M. y Ambort, D., Estrategias para la motivación e integración en el primer año universitario: Innovación mediante resolución de problemas y trabajo colaborativo, *Virtualidad, Educación y Ciencia*, 7(12), pp. 85-98, 2016.
- [7] Linder, C., Fraser, D.M. and Pang, M.F., Using a variation approach to enhance physics learning in a college classroom, *The Physics Teacher*, 44, pp. 63-66, 2006. DOI: 10.1119/1.2396777
- [8] Magana, A.J., Brophy, S.P. and Bodner, G.M., Instructors' intended learning outcomes for using computational simulations as learning tools, *Journal of Engineering Education*, 101(2), pp. 220-243, 2012. DOI: 10.1002/j.2168-9830.2012.tb00049.x
- [9] Johri, A. and Olds, B.M., Situated engineering learning: Bridging engineering education research and the learning sciences. *Journal of Engineering Education*, 100(1), pp. 151-185, 2011. DOI: 10.1002/j.2168-9830.2011.tb00007.x
- [10] Gordon, G., *Simulación de sistemas*, México D.F., Ed. Diana, 1991.
- [11] Riesco, M., Fondón, M., Álvarez, D., López, B., Cernuda, A., Juan, A., *Informática: Materia esencial en la educación obligatoria del siglo XXI*, Revista Electrónica ReVisión, 7(3), pp. 53-60, 2014.
- [12] Ingerman, A., Linder, C. and Marshall, D., Learning-focuses in physics simulation learning simulations, In 12th Annual Conference of SAARMSTE, Cape Town, South Africa, 2004.
- [13] Gil Martín M. y García-Barneto, A., Entornos constructivistas de aprendizaje basados en simulaciones informáticas, REEC: Revista Electrónica de Enseñanza de las Ciencias, 5(2), pp. 304-322, 2006.
- [14] Gunstone, R.F., Constructivism and learning research in science education, in *Constructivism in Education: Opinions and Second Opinions on Controversial Issues*, D.C. Phillips Ed., Chicago, IL: University of Chicago Press, 2000, pp. 254-280.
- [15] Brophy, S.P., Magana, A.J. and Strachan, A., Lectures and simulation laboratories to improve learners' conceptual understanding. *Advances in Engineering Education*, 3(3), pp. 1-28, 2013.
- [16] Carbonell, V., Romero, C., Martínez E. and Flórez, M., Interactive simulations as teaching tools for engineering mechanics courses. *European Journal of Physics*, 34(4), pp. 991-1004, 2013. DOI: 10.1088/0143-0807/34/4/991
- [17] Streicher, S.J., West, K., Fraser, D.M., Case, J.M. and Linder, C.J., Learning through simulation: Student Engagement, *Chemical Engineering Education*, 39(4), pp. 288-295, 2005.
- [18] Goodyear, P., A Knowledge-based approach to supporting the use of simulation programs, *Computers and Education*, 16, pp. 99-103, 1990. DOI: 10.1016/0360-1315(91)90049-W
- [19] Parush, A., Hamm, H. and Shtub, A., Learning histories in simulation-based teaching: The effects on self-learning and transfer. *Computers and Education*, 39, pp. 319-332, 2002. DOI: 10.1016/S0360-1315(02)00043-X
- [20] Davies, C.H.J., Student engagement with Simulations: A case study, *Computers and Education*, 39, pp. 271-282, 2002. DOI: 10.1016/S0360-1315(02)00046-5
- [21] Balanskat, A. and Engelhardt, K., Computing our future. *Computer programming and coding. Priorities, school curricula and initiatives across Europe*, Brussels, 2015.
- [22] Lye, S.Y. and Koh, J.H.L., Review on teaching and learning of computational thinking through programming: What is next for K-12?, *Computers in Human Behavior*, 41, pp. 51-61, 2014. DOI: 10.1016/j.chb.2014.09.012
- [23] Yadav, A., Hong, H. and Stephenson, C., Computational thinking for all: Pedagogical approaches to embedding 21st Century problem solving in K-12 Classrooms. *TechTrends*, (May), 2016, pp. 10-13. DOI: 10.1007/s11528-016-0087-7
- [24] Kelleher, C. and Pausch, R., Lowering the barriers to programming. *ACM Computing Surveys*, 37(2), pp. 83-137, 2005. DOI: 10.1145/1089733.1089734
- [25] Malan, D.J. and Leitner, H.H., Scratch for budding computer scientists. In *Thirty-Eighth Sigse Technical Symposium on Computer Science Education*, Covington, 2007. DOI: 10.1145/1227310.1227388
- [26] Kaleliog, F., A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, pp. 200-210, 2015. DOI: 10.1016/j.chb.2015.05.047
- [27] Maloney, J., Resnick, M., Rusk, N., Silverman, B. and Eastmond, E., The scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), pp. 1-15, 2010. DOI: 10.1145/1868358.1868361
- [28] Garcia, D., Harvey, B. and T. Barnes, T., The beauty and joy of computing. *ACM Inroads*, 6(4), pp. 71-79, 2015. DOI: 10.1145/2835184
- [29] Willging, P., Astudillo, G.J. y Bast, S.G., El software de animación como una estrategia innovadora para el aprendizaje de lenguajes de programación, En *XIV Workshop de Investigadores en Ciencias de la Computación*, Misiones, 2012.
- [30] Wolz, U., Leitner, H.H., Malan, D. and Maloney, J., Starting with scratch in CS 1. *ACM SIGCSE Bulletin*, 41(2). SIGSE 2009. New York, Estados Unidos. Marzo 2009.
- [31] Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. and Kafai, Y., *Scratch: Programming for all*. *Communications of the ACM*, 52(11), pp. 60-67. 2009. DOI: 10.1145/1592761.1592779
- [32] Yin, R.K., *Case study research: Design and methods*. SAGE. London. 2003.
- [33] Atkinson, P., Coffey, A., Delamont, S., Lofland, J. and Lofland, C., *Handbook of ethnography*. London: Sage, 2007.
- [34] Walford, G., The nature of educational ethnography, in: *How to do educational ethnography*, G. Walford (Ed.). London: Tufnell Press, 2008, pp. 1-15.
- [35] San Martín, P., Aspectos sociales y tecnológicos del Dispositivo Hipermedial Dinámico desarrollados en diferentes contextos educativos, *Revista de Educación*, 4(5), pp. 81-98, 2013.
- [36] San Martín, P. y Rodríguez, G., Construir un nuevo diseño curricular participando de un Dispositivo Hipermedial Dinámico, *Revista Cognición. Revista Científica de FLEAD*, 22, pp. 1-17, 2009.
- [37] Braccialarghe, D., Introcaso, B. y Rodríguez, G., Hacia la construcción de la modalidad de taller como propuesta de integración entre introducción a la ingeniería y las ciencias básicas, *Revista Argentina de Enseñanza de la Ingeniería*, 4(9), pp. 41-50, 2015.
- [38] Yin, R.K., *Applications of case study research* (3rd ed.). London: Sage, 2011.
- [39] Gee, J.P. and Green, J.L., Discourse analysis, learning and social practice: A methodological study. *Review of Research in Education*, 23, pp. 119-169, 1998. DOI: 10.3102/0091732X023001119

N.G. Monjelat, recibió el título de Lic. en Psicopedagogía en 2008 por la Universidad Nacional de San Martín, Buenos Aires, Argentina, de MSc. en Comunicación y Aprendizaje en la sociedad digital en 2012 y de Dra. en Comunicación, Educación y Sociedad en 2013, estos dos últimos de la Universidad de Alcalá, Madrid, España. De 2010 a 2015 fue investigadora en el Grupo Imágenes, Palabras e Ideas de la Universidad de Alcalá, coordinado por Pilar Lacasa. Asimismo, fue docente en dicha institución y en la Universidad Internacional de la Rioja. Durante su estadía en España, dictó además cursos de programación a docentes de nivel primario y secundario y talleres a niños en estas temáticas. En 2016 se incorporó como becaria postdoctoral en el Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE: CONICET-UNR) donde se radicó posteriormente como Investigadora Asistente, cargo que desempeña actualmente. Sus intereses investigativos incluyen: procesos de enseñanza y aprendizaje en diferentes niveles educativos mediados por tecnologías de la información y la comunicación, investigación cualitativa, pensamiento computacional y programación en contextos educativos, entre otros.
ORCID: 0000-0002-5043-8989

G.L. Rodríguez, recibió el título de Ing. Mecánico en 1998 y de Dr. en Ingeniería en 2011, todos ellos de la Universidad Nacional de Rosario, Rosario, Argentina. También posee el diploma de Profesor en Filosofía, año 2004. Desde 1996 al 2004 trabajó en la industria regional metalmeccánica en las áreas de diseño y mecánica computacional. Comenzó como docente auxiliar en la Universidad Nacional de Rosario en 1997 y es profesor titular dedicación exclusiva desde el año 2011. Desde hace varios años participa activamente en proyectos acreditados de I+D en el área de Nuevas Tecnologías Educativas. Posee publicaciones en diversos congresos tanto nacionales como

internacionales y desarrollos de software. En la actualidad es Investigador asociado del Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE: CONICET-UNR).
ORCID: 0000-0001-7112-5116

P.S. San Martín, es Dra. en Humanidades y Artes por la Universidad Nacional de Rosario, Argentina. Miembro de la Carrera de Investigador Científico y Tecnológico del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina), categoría Independiente. Vicedirectora del Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE:CONICET-UNR). Profesora Titular ordinaria de la Facultad de Humanidades y Artes, Universidad Nacional de Rosario (UNR). Docente- Investigadora categoría I (UNR). Su especialidad de Investigación, Desarrollo e Innovación se centra en las Tecnologías de la Información y Comunicación aplicadas a Educación. Ha dictado cursos de Posgrado en distintas universidades del país. Posee numerosas publicaciones nacionales e internacionales, premios y distinciones por su labor. Ha dirigido Programas, Proyectos y Tesis doctorales, investigadores, becarios doctorales y posdoctorales donde se fue construyendo el marco teórico-metodológico sobre los Dispositivos Hipermediales Dinámicos para investigación, educación y ciudadanía.
ORCID: 0000-0001-7543-045X