



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

**Construcción de modelos y simuladores con
Squeak-Etoys como recurso de aprendizaje
en la escuela media**

Ricardo Pablo Salvador

Claudia Pons (Directora)

Guillermo Rodríguez (Co-Director)

Tesis presentada para obtener el grado de
Magister en Tecnología Informática Aplicada en Educación

Junio de 2017

1 Agradecimientos

A Laura, por su increíble apoyo y paciencia. A Paula y a Pedro, por incentivar me, aún sin saberlo.

A mis alumnos por inspirarme y motivarme: por ellos este trabajo tiene sentido.

A Doti Vítola y a Gustavo González por sus consejos y su permanente ánimo.

A los profesores Diego Zanarini y Daniela Hadad por su tiempo, asesoramiento y brindar sus cursos para realizar el trabajo de campo.

A Inés y a Mariana por hacer la redacción más clara.

A la directora y al codirector de la tesis, la Dra. Claudia Pons y el Dr. Guillermo Rodríguez, sin quienes esta tesis no se hubiera realizado.

2 Resumen

En esta tesis se estudia la influencia, en el aprendizaje de cinemática, de una actividad didáctica en la que alumnos de secundaria construyen una simulación de un ejercicio típico de “encuentro” utilizando el entorno de programación Squeak-Etoys.

Esta idea tiene lugar a partir de tres hechos. El primero es que el aprendizaje basado en la comprensión, la participación activa de los alumnos y la interacción con sus pares, encuentra en el uso de simulaciones informáticas un recurso condensador de habilidades complejas. El segundo es el surgimiento de entornos de programación que combinan *a)* interfaces de usuario con alto grado de usabilidad, *b)* posibilidades de manipulación de recursos multimedia y *c)* la escritura de código facilitada por medio de “arrastrar y soltar” bloques. Lo anterior llevó a considerar que construir (en lugar de sólo usar) una simulación será tanto o más útil en términos de aprendizaje disciplinar específico y como experiencia áulica en sentido amplio. El tercer hecho es la valoración a nivel mundial de las ciencias de la computación y la programación en el ámbito educativo.

Se realizó un trabajo experimental en el que participaron alumnos de dos comisiones de Física de 4º año de secundaria. En las clases del grupo designado como experimental, los alumnos utilizaron un tutorial interactivo que proporciona las indicaciones y los recursos de programación para construir la simulación de un ejercicio de cinemática.

La experiencia se evaluó mediante un protocolo pre-test / post-test, y los resultados indican que *a)* los alumnos aumentaron las puntuaciones en las evaluaciones de cinemática, *b)* valoran la programación de la simulación como técnica didáctica, *c)* se interesaron en el entorno de programación: en modificar la simulación en modos alternativos, en la posibilidad de utilizarlo en otras asignaturas, y acerca de la programación como disciplina, *d)* se observó un rápido aprendizaje de la interfaz de usuario.

Los resultados animan a diseñar actividades como la descripta, teniendo en cuenta la recepción que tuvo en los alumnos como las inquietudes que ésto manifestaron acerca de la experiencia.

3 Abstract

In this thesis we study the influence, in the learning of kinematics, of a didactic activity in which secondary students construct a simulation through the Squeak-Etoys programming environment.

The starting point comes from three facts. The first is that learning based on comprehension, active participation of students and interaction with peers, finds in the use of computer simulations a resource that concentrate complex skills. The second is the emergence of programming environments that combine *a)* user interfaces with a high degree of usability, *b)* possibilities of manipulation of multimedia resources and *c)* the writing of code facilitated by means of "drag and drop" blocks. This led to consider that constructing (rather than just using) a simulation will be as much or more useful in terms of specific disciplinary learning and as a broad-based aula experience. The third fact is the global acknowledgment of computer science and programming in the educational field.

An experimental work was carried out in which students from two Physics courses of 4th year of middle school participated. In the classes of the experimental group, the students used an interactive tutorial that provides the indications and programming resources to construct the simulation of a kinematics exercise.

The experiment was evaluated using a pre-test / post-test protocol, and the results indicate that *a)* the students increased the scores in the kinematics evaluations, *b)* they value the programming of the simulation as a didactic technique, *c)* the students were interested in the programming environment: in modifying the simulation in alternative modes, in the possibility of using it in other subjects, and in programming as a discipline, *d)* a quick learning of the user interface was observed.

The results encourage the design of activities as described, taking into account the reception that had in the students as the expectations that this manifested about the experience.

4 Sumario

1 Agradecimientos.....	2
2 Resumen.....	3
3 Abstract.....	4
4 Sumario.....	5
5 Introducción.....	10
5.1 Objetivo.....	10
5.2 Motivación.....	10
5.3 Metodología.....	12
5.4 Estructura de la tesis.....	13
6 Computadoras y simulación.....	17
6.1 Sistemas, modelos, simulación y programación como recursos de aprendizaje.....	17
6.1.1 Sistema.....	17
6.2 Modelado de sistema.....	18
6.2.1 Tipos de modelos.....	19
6.3 Simulación.....	19
6.3.1 Tipos de simulaciones.....	20
6.3.2 Componentes de un modelo de simulación.....	20
6.4 Simulaciones en educación.....	21
6.5 La construcción de simulaciones como recurso didáctico.....	24
7 Habilidades intelectuales y habilidades digitales.....	26
8 Aportes del Construcciónismo a la enseñanza mediante computadoras.....	28

9 La importancia de enseñar Ciencias de la Computación.....	32
9.1 Las Ciencias de la Computación.....	32
9.2 El pensamiento computacional.....	34
10 Iniciativas promoviendo las Ciencias de la Computación y la Programación en las escuelas.....	39
10.1 Iniciativas gubernamentales en Argentina.....	39
10.1.1 Experiencias basadas en el modelo de laboratorio.....	39
10.1.2 El Portal Educ.ar.....	40
10.1.3 Las TIC en la Ley de Educación Nacional N.º 26.206.....	40
10.1.4 Experiencias basadas en el modelo 1 a 1 y aulas digitales.....	41
10.1.5 El Programa Conectar Igualdad.....	41
10.1.6 Program.AR.....	42
10.1.7 La “Red de escuelas que programan”.....	43
10.1.8 “La hora del código”.....	43
10.1.9 La importancia de enseñar Computación en las escuelas argentinas: informe de la Fundación Sadosky.....	43
10.2 Experiencia de enseñanza de programación en escuelas medias de La Plata.....	44
10.3 Programación en la educación en otros países.....	45
10.4 Un estudio sobre la influencia de la programación en el aprendizaje de física.....	47
11 Squeak-Etoys: un entorno de programación diseñado para la educación.....	48
11.1 ¿Qué es Squeak?.....	48
11.1.1 Un poco de historia.....	48
11.1.2 La interfaz de usuario de Squeak.....	49

11.1.3 Portabilidad.....	51
11.1.4 Campos de aplicación.....	52
11.2 Etoys.....	52
11.2.1 Alan Kay, las ideas poderosas y el VPRI.....	54
11.2.2 Generalidades.....	55
11.2.3 La interfaz gráfica de Etoys.....	55
11.2.3.1 Barra de herramientas.....	56
11.2.3.2 Provisiones: de aquí obtenemos los objetos de nuestro proyecto... ..	57
11.2.3.3 Objetos y <i>halos</i>	57
11.2.3.4 El visor: categorías, atributos y acciones.....	58
11.2.3.5 Guiones.....	60
11.2.3.6 Clases de morphs u objetos de Etoys.....	61
11.2.4 Repositorios de proyectos.....	63
11.2.5 En resumen.....	64
12 Marco metodológico.....	65
12.1 Etapas del trabajo experimental.....	65
12.1.1 Búsqueda de docentes interesados en realizar la experiencia.....	65
12.1.2 Elección de los grupos experimental y de control.....	65
12.1.3 Elección de los contenidos para aplicar la propuesta didáctica:.....	66
12.1.4 Encuesta sobre características generales del grupo y encuesta sobre Cinemática previa al trabajo experimental.....	66
12.1.5 Realización del trabajo experimental.....	67
12.1.5.1 El tutorial interactivo.....	67
12.1.5.1.1 Formato del tutorial.....	67

12.1.5.1.2 Material impreso.....	70
12.1.5.2 Sumario de diapositivas.....	70
13 Resultados.....	72
13.1 La Escuela.....	72
13.2 Actividad experimental.....	73
13.2.1 Indicaciones previas a la actividad.....	73
13.2.2 Desarrollo en el tiempo.....	74
13.2.3 Encuesta posterior al trabajo experimental.....	76
13.3 Resultados de las encuestas.....	76
13.3.1 Grupo control.....	76
13.3.1.1 Tablas y gráficos.....	77
13.3.1.2 Diagnóstico de características generales del grupo.....	77
13.3.1.3 Grupo control: evaluación de contenidos de Cinemática.....	81
13.3.2 Grupo experimental.....	88
13.3.2.1 Tablas y gráficos.....	88
13.3.2.2 Grupo experimental: diagnóstico de características generales del grupo.....	89
13.3.2.3 Grupo experimental: evaluación de contenidos de Cinemática.....	93
13.3.2.4 Testimonios escritos de alumnos del grupo experimental realizados con posterioridad a la experiencia.....	99
14 Discusión.....	100
15 Conclusiones.....	105
16 Anexos.....	107
16.1 Anexo 1: Encuesta sobre características generales del grupo.....	107

16.2 Anexo 2: Encuesta sobre conocimientos de Cinemática.....	108
16.3 Anexo 3: Encuesta posterior al trabajo experimental.....	109
16.4 Anexo 4: Guía para la Simulación del Ejercicio 14 de Cinemática.....	110
16.5 Anexo 5: Póster JEMU 2016.....	118
16.6 Anexo 6: Taller "Simulaciones en la escuela con Squeak-Etoys", SABTIC 2017.....	119
16.6.1 Diapositivas.....	121
16.7 Otros antecedentes relacionados con el tema de la tesis.....	123
16.8 Anexo 8: Índice de figuras.....	125
16.9 Anexo 9: Índice de tablas.....	128
17 Bibliografía.....	130

5 Introducción

5.1 Objetivo

El objetivo de esta tesis es indagar la utilidad del diseño y desarrollo de modelos y simulaciones informáticas por parte de los alumnos como recurso de aprendizaje en la escuela media, en una disciplina distinta de las relacionadas a la programación misma, utilizando un entorno de programación amigable (Squeak-Etoys).

El aporte que se busca es proponer una metodología de diseño de actividades en las que los alumnos deban modelar y simular contenidos escolares. De este modo, realizarán una actividad que necesariamente implica la puesta en juego de habilidades intelectuales de alto nivel y complejidad, especialmente las vinculadas con el diseño, creación y autoevaluación, así como el trabajo en equipo, la interacción entre pares y el intercambio de experiencias con un recurso motivador y afín a los adolescentes.

5.2 Motivación

Mucho se ha dicho en términos de ventajas y desventajas de los diferentes modos de aprender, optando siempre las declamaciones, las voluntades y las esperanzas por las estrategias activas, participativas, personalizantes y abiertas a la creatividad, frente a las que ponen al que aprende en un rol pasivo y constrictor de la individualidad como fuerza que impulsa el aprendizaje.

Entre los recursos que más promueven el aprendizaje activo, crítico y la metacognición, estimulando la creatividad, y permitiendo poner a prueba hipótesis, por ejemplo, se cuentan las simulaciones (Gil Martín y García Barneto, 2006). La expectativa que motiva esta Tesis es el supuesto de que más valiosa será la experiencia de aprendizaje si es el mismo alumno de escuela media quien no sólo pone a prueba un modelo de simulación, sino que lo diseña, construye, y tras evaluarlo, puede modificarlo en un proceso que implica la metacognición y la apropiación del contenido escolar involucrado.

Actualmente se está revalorizando la enseñanza de las Ciencias de la Computación en el ámbito escolar y la enseñanza de la Programación y su uso como

recurso educativo y como competencia general en distintos lugares del mundo. Un ejemplo de esto son iniciativas tales como el proyecto Raspberry Pi en la Universidad de Cambridge en el Reino Unido (<http://www.raspberrypi.org/about/>), Scratch, un proyecto del Laboratorio de Medios del Massachusetts Institute of Technology (<http://scratch.mit.edu/about/>) y Squeak-Etoys (<http://squeakland.org/about/>).

Code.org es una organización sin fines de lucro apoyada por las empresas de IT más importantes, dedicada a expandir la participación de la educación en Ciencias de la Computación y busca aumentar la participación en esta disciplina de las mujeres y otros grupos sociales que se encuentran actualmente subrepresentados; su visión es que los estudiantes deben tener la oportunidad de aprender Ciencias de la Computación y que la Informática debe ser parte del plan de estudios en la educación formal junto a las demás materias.

En nuestro país, La hora del código es la iniciativa local que reúne el apoyo de Code.org y Program.ar, siendo éste último un proyecto dirigido en conjunto por la Jefatura de Gabinete de Ministros, la Fundación Sadosky, el portal Educ.ar y el Programa Conectar Igualdad. Al igual que Code.org, La hora del código ofrece recursos para aprender a programar de forma amena y divertida como una forma de aproximarse a las Ciencias de la Computación.

Kafai y Quinn (2014), señalan que dada la omnipresencia de computadores en todas sus formas (de escritorio, laptops, tabletas y celulares), se han convertido en una extensión de nosotros mismos, y la programación, en una época asunto de unos pocos eruditos, es hoy reconocida por los educadores como una habilidad crucial, una nueva alfabetización, una respuesta a la necesidad de desarrollar pensamiento computacional en los estudiantes. La premisa es que a través de este tipo de pensamiento los estudiantes pueden resolver problemas, diseñar sistemas para su uso en la vida diaria, y generar progresos e innovación en otras disciplinas (Kafai y otros, 2014).

Según el reporte de la Fundación Sadosky “Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas” (2013), la programación aporta también herramientas para dejar de ser meros consumidores del software, comprender cómo funciona éste y el mundo digital en general, y mejorar las habilidades intelectuales. Por esto se propone la enseñanza de la programación en

las escuelas 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, como lo es la resolución de ecuaciones matemáticas simples (Fundación Sadosky, 2013).

Si bien la Programación en otras épocas presentaba cierto grado de dificultad debida a la escritura de código, actualmente se cuenta con entornos de desarrollo que en ese aspecto facilitan el acceso a legos en la materia, específicamente, maestros y alumnos. Uno de los entornos de desarrollo creados para ser utilizado en la educación es Squeak-Etoys, una implementación de Smalltalk, un entorno conceptual simple y específicamente diseñado para el uso en las aulas (<http://squeakland.org/about/>) en el cual todas las operaciones que usualmente requieren la escritura de código se realizan mediante instrucciones representadas gráficamente por mosaicos que incluyen la sintaxis y la documentación de ayuda, lo que facilita que el alumno y el docente puedan desarrollar actividades interactivas. Por otra parte esta aplicación es gratuita, es multiplataforma, y se integra en navegadores web. Etoys está integrado en el software de las laptops del proyecto One Laptop Per Child (<http://one.laptop.org/about/software>).

5.3 Metodología

La tesis contempla la realización de un trabajo experimental consistente en evaluar el efecto del uso del modelado y la simulación para la construcción de un programa como recurso para el aprendizaje de contenidos escolares mediante un diseño pre-test / post-test en el que participan un grupo control y un grupo experimental, ambos en su totalidad. Para lo esto se cuenta con la colaboración de su profesor de Física, con quien se eligió el contenido específico para el que se diseñó la actividad que realizaron los alumnos del grupo experimental. Previa y posteriormente a esta actividad se realizaron evaluaciones a ambos grupos.

La actividad experimental consistió en tres clases para las que se desarrolló un tutorial interactivo en el cual los alumnos examinaron, modificaron y crearon el código necesario para construir la simulación de un ejercicio habitual de Cinemática.

El impacto en el ámbito de aplicación puede resultar en la extensión de este recurso didáctico a contenidos de diferentes campos.

5.4 Estructura de la tesis

A continuación de esta introducción la tesis se organiza en 12 secciones.

A fin de exponer las posibilidades de utilizar simulaciones en actividades educativas, en el capítulo 6, *Computadoras y simulación* se presentan los conceptos de sistema, modelos y simulación; para finalizar se destacan implicancias de la construcción de simulaciones por parte de los alumnos como recurso de aprendizaje.

En el capítulo 7, *Habilidades intelectuales y habilidades digitales*, se reseña una versión revisada de la taxonomía de objetivos educativos y habilidades intelectuales de Bloom, en la que se incluyen habilidades relacionadas al trabajo con las tecnologías de la información y la comunicación.

El *construcionismo* es la visión de Seymour Papert en relación a la enseñanza de y con computadoras. En el capítulo 8, *Aportes del Construcionismo a la enseñanza mediante computadoras*, se recopilan los postulados de Papert en torno al aprendizaje y la educación, el objetivo de su trabajo, su visión sobre la enseñanza en el futuro y en función de las tecnologías digitales, su consistencia con el constructivismo (así como la nota característica que lo distingue), su visión sobre las formas de utilizar la tecnología y cómo se reflejan en la educación, las características y postulados de lo que él llamó la "filosofía Logo", no restringida al trabajo con ese lenguaje, sino extensible a la educación y aún a otros campos, tal como lo señaló el mismo Papert.

La importancia de enseñar ciencias de la computación, es el título del 9º capítulo. En éste se caracteriza el campo de las ciencias de la computación según el documento "CC-2016 de la Fundación Sadosky", y se resumen opiniones acerca del rol de la programación en el mundo actual. También se analiza el concepto de pensamiento computacional en tanto es un conjunto de competencias propias que surgidas en las tecnologías computacionales son útiles en otros campos. Se analizan los aportes de Brennan y Resnick en una investigación sobre cómo la programación influyen en el desarrollo del pensamiento computacional; también se enumeran conceptos comunes a muchos lenguajes de programación y el pensamiento computacional. Para finalizar se comentan dos trabajos que apoyan el vínculo entre programar y desarrollo de pensamiento computacional.

Desde hace aproximadamente 25 años tiene lugar un sinnúmero de, *Iniciativas promoviendo las Ciencias de la Computación y la Programación en las escuelas*. Tales es el título del capítulo número 10, que reseña iniciativas, tanto gubernamentales como privadas con el objetivo de impulsar las ciencias de la computación y la enseñanza de la programación en las escuelas. Se hace hincapié especialmente en la legislación y las experiencias gubernamentales en Argentina desde los años 90, los modelos basados en modelo de laboratorio, y posteriormente en el modelo 1 a 1, el portal Educ.ar, los programas Conectar Igualdad, Program.AR, la Red de escuelas que programan y La hora del código. Cierran el capítulo los motivos para enseñar programación en las escuelas argentinas que la Fundación Sadosky expone en su reporte "Una propuesta para refundar la enseñanza de la computación en las escuelas argentinas" (2013), una reseña sobre la enseñanza de disciplinas relativas a las ciencias de la computación en distintos países desde 2008, el comentario de dos experiencias: uno sobre la enseñanza de programación en escuelas medias de la ciudad de La Plata, y otro acerca del estudio sobre la influencia de la programación en el aprendizaje de física.

El trabajo de campo que forma parte de esta tesis se basó en un tutorial interactivo desarrollado en Squeak-Etoys. *Squeak-Etoys, un entorno de programación diseñado para la educación*, el título del capítulo 11, en el cual se caracteriza este software de autor a partir del pensamiento de su creador e impulsor, Alan Kay, la historia de Squeak, características principales, campos de aplicación, el surgimiento de Etoys, y un análisis de éste último en cuanto a su interfaz gráfica, los recursos que provee para crear proyectos, el tipo de objetos disponibles, su uso, el acceso a atributos y métodos, construcción de guiones o scripts, y una lista de repositorios de proyectos de distintos lugares del mundo.

El capítulo 12 corresponde al *marco metodológico*. Se describen las etapas y la realización del trabajo experimental: la búsqueda de docentes interesados en realizar la experiencia, la elección de grupos experimental y de control, la elección de los contenidos curriculares en que se trabajaría, las encuestas sobre las características del grupo y la evaluación sobre contenidos de la disciplina, realización del trabajo experimental; se analiza el tutorial interactivo, su desarrollo, su formato, el material impreso elaborado como complemento, y por último se enumeran y comentan las diapositivas del tutorial.

Los *resultados* se detallan en el capítulo 13, sección en la que se hace una descripción de la Escuela donde se realizó el trabajo de campo planteado en la metodología, se detalla la carga horaria y contenidos del espacio curricular Física; se da cuenta de la actividad realizada por el grupo experimental, la preparación, indicaciones previas a la actividad, se transcriben las notas relativas al desarrollo de cada clase; se caracterizan los grupos control y experimental a partir de los resultados de encuestas y evaluaciones expresados en cuadros y gráficas. Finaliza con testimonios redactados por alumnos que participaron de la actividad experimental.

En el capítulo 14 se desarrolla la *discusión*. Se examinan los resultados confrontándolos con el marco teórico, los antecedentes y el objetivo de la tesis, destacando coincidencias y divergencias, se arriesgan eventuales explicaciones y se trazan posibles experiencias futuras. A continuación, se presentan las conclusiones finales de la tesis, en el capítulo 15.

En el capítulo siguiente se encuentran los cuatro *anexos* de la tesis:

- Anexo 1: Encuesta realizada a los alumnos a fin de obtener una caracterización de los grupos, donde se indagan hábitos de uso de tecnología informática, modos de resolver problemas, conocimiento de los conceptos modelo y simulación, y opinión sobre la utilidad de las simulaciones en sus aprendizajes.
- Anexo 2: Evaluación sobre conocimientos de Cinemática, que se realizó previa y posteriormente a la actividad experimental.
- Anexo 3: Encuesta posterior al trabajo experimental donde se vuelve a indagar eventuales modificaciones en los conceptos de modelo y simulación, y acerca de la opinión sobre la utilidad de las simulaciones como recurso de aprendizaje.
- Anexo 4: Guía impresa para la simulación del Ejercicio 14 de Cinemática.
- Anexo 5: Póster presentado en las Jornadas de Escuelas Medias Universitarias (JEMU) realizado los días 5 y 6 de octubre de 2016 en Horco Molle, Tucumán.

- Anexo 6: planificación del taller "Simulaciones en la escuela con Squeak-Etoys", y miniaturas de las diapositivas expuestas en el mismo.
- Anexo 7: ponencias, talleres y otras actividades relativas al tema de la tesis

En el último capítulo se lista la *bibliografía* consultada para la elaboración de esta tesis.

6 Computadoras y simulación

Frente a tecnologías educativas que ponen al alumno en un rol relativamente pasivo, tales como el libro, el audio y el video, las computadoras como máquinas interactivas aportan la posibilidad de un aprendizaje especialmente activo, posibilidad sobre la que contamos con notables aportes de Seymour Papert (destacado discípulo de Piaget, MIT, pionero de la inteligencia artificial, creador del Logo, participante del proyecto OLPC) y Alan Kay (cuyo nombre veremos asociado a Xerox Palo Alto Research Center, ARPANet, Atari, desarrollo de la primera computadora personal gráfica orientada a objetos, SmallTalk, Squeak-SmallTalk, Etoys, Hewlett Packard, Instituto Viewpoints Research).

Jonassen (1996) resume ese uso posible de las computadoras para el aprendizaje crítico en el concepto de computadora como mindtool, y en esa dirección también se crearon los Etoys desarrollados en Squeak, ambiente de desarrollo multimedial, multiplataforma, con licencia de uso libre, y basado en Smalltalk80, paradigma de programación orientada a objetos (POO), tema en el que aún hoy es referencia obligada.

6.1 Sistemas, modelos, simulación y programación como recursos de aprendizaje

El modelado y la implementación de simulaciones de sistemas es una actividad que se utiliza para el estudio de situaciones complejas y a gran escala. Tener en cuenta su enfoque, articulándolo con los entornos de programación (cada vez más potentes y con mayor grado de usabilidad) y de las microcontroladoras de hardware (cada vez más modulares y accesibles), puede ser un recurso valioso para enriquecer situaciones de enseñanza-aprendizaje, incorporando a los modos de representación habituales otras dimensiones: tiempo, movimiento, recursos multimediales, y diseño e interacción con dichas representaciones.

6.1.1 Sistema

Un sistema es un conjunto de objetos reunidos en alguna interacción o interdependencia regular (Gordon, 1991:15).

En un sistema se diferencian objetos según su relevancia o interés en el funcionamiento del mismo. Cuando un objeto tiene interés para el estudio del sistema se denomina entidad; una entidad puede tener diversas propiedades o atributos. A su vez, ocurren procesos (denominados actividades) dentro del sistema que provocan cambios. Las entidades, atributos y actividades que se consideran relevantes en un sistema se escogen en función del propósito que guía el estudio del mismo.

La descripción del conjunto de entidades, atributos y actividades en un momento dado se denomina estado del sistema. Una sucesión de estados de un sistema y los cambios entre sí se utilizan para estudiar su evolución.

Las actividades que ocurren dentro del sistema se denominan endógenas. Hay actividades que producen cambios fuera del mismo, y el ámbito en que ocurren esos cambios es llamado medio ambiente del sistema. Como contrapartida a las actividades endógenas, hay actividades que ocurren en el medio ambiente modificando al sistema y se denominan exógenas.

Analizar una situación con este enfoque implica el conocimiento de sus componentes y sus propiedades, los procesos que relacionan y afectan a esos componentes, al sistema y al ambiente que lo contiene.

6.2 Modelado de sistema

Muchas veces es posible experimentar con el sistema real, sin embargo, cuando no es posible debido a que no ha sido construido, o ante la imposibilidad de experimentar con el sistema o con prototipos, se utiliza un modelo del sistema.

Un modelo se define como la información relativa a un sistema, recabada a los fines de estudiarlo (Gordon, 1991:20). Qué aspectos de toda la información del sistema se reúnen está determinado por los propósitos del estudio del sistema, por lo que diversos propósitos de estudio dan lugar a otros tantos modelos de un mismo sistema.

Al contener en su descripción sólo los detalles relevantes para su estudio, un modelo es una simplificación del sistema al que refiere. Debe ser diseñado para sustituir al sistema de manera que se comporte como éste lo haría ante las pruebas a que sea expuesto. El modelo debe proporcionar un aprendizaje que no sea diferente

del que proporcionaría el sistema directamente; este principio se denomina validez y es fundamental para la utilidad del modelo.

Un modelo se obtiene a partir de dos tareas:

a) determinar su estructura, lo que fija los límites del sistema, identifica entidades, atributos y actividades del mismo.

b) proporcionar los valores que los atributos pueden tener y definen las relaciones involucradas en las actividades.

6.2.1 Tipos de modelos

En Kelton y otros (2008:4) se distinguen dos tipos básicos de modelos:

- Modelos físicos: se trata de réplicas físicas o a escala del sistema; también son llamados modelos icónicos.
- Modelos lógicos o matemáticos: se trata de conjuntos de aproximaciones y suposiciones estructurales y cuantitativas sobre el modo en que funciona el sistema. Por lo general un modelo lógico se implementa en un programa de computadora cuya ejecución representa el desempeño del modelo y permite ponerlo a prueba. El hecho de que se trata de un software hace que generalmente sea más sencillo, económico y rápido experimentar que con el sistema real. Por otra parte, pueden cometerse o inducirse errores que sería imposible replicar en otro tipo de modelos o en la realidad, por ejemplo, en el caso de consecuencias catastróficas o perjudiciales para individuos, la sociedad o el medio ambiente.

Las cotas alcanzadas por la tecnología digital y de software, y la disminución de costos permiten llevar a cabo análisis computacionales de complejos modelos lógicos.

6.3 Simulación

En los casos de modelos simples, las herramientas matemáticas tradicionales como la teoría de colas, ecuaciones diferenciales o la programación lineal son suficientes. Sin embargo también hay modelos cuya complejidad excede al uso de las herramientas mencionadas, y es en estos casos donde la posibilidad de solución está

en usar «simulaciones por computadora, que consisten en métodos para estudiar modelos mediante la evaluación numérica usando software diseñado para imitar al sistema, a menudo en el transcurso del tiempo» (Kelton y otros, 2008:5).

En términos prácticos, una simulación es el proceso de diseñar y construir un modelo computarizado de un sistema real o no, con el fin de estudiar el comportamiento del mismo en determinadas condiciones.

Si bien pueden construirse simulaciones para sistemas sencillos, es en sistemas complejos donde se obtiene mayor beneficio debido a que puede representarse gran complejidad y alto grado de exactitud.

6.3.1 Tipos de simulaciones

Según la influencia del paso del tiempo se clasifican en estáticos o dinámicos.

Según el modo en que cambia el estado del sistema pueden ser continuos o discretos. Sin embargo en el interior del sistema puede haber entidades con atributos de valores continuos o discretos.

Según la naturaleza de sus entradas, los sistemas pueden ser deterministas (cuando las entradas no son aleatorias) o estocásticos (cuando sí existen entradas aleatorias).

6.3.2 Componentes de un modelo de simulación

- Entidades: son objetos dinámicos que son creados, experimentan cambios y pueden permanecer o no en el sistema; por cada tipo de entidad pueden existir muchas copias de la misma, así como de una especie biológica pueden existir muchos individuos.
- Atributos: son las características que permiten individualizar a las entidades al tomar valores específicos, que pueden diferir entre las mismas.
- Variables globales: son información relativa a algún aspecto del sistema, a diferencia de los atributos, propios de las entidades.
- Recursos: los recursos representan cosas tales como personal, equipamiento, espacio de almacenaje.

- Colas: se trata de dispositivos que almacenan entidades cuando éstas no disponen de determinados recursos para continuar un determinado proceso.
- Acumuladores estadísticos: son variables que se actualizan de acuerdo al progreso de la simulación.
- Eventos: se trata de algo que ocurre en un instante de tiempo y que puede cambiar atributos, variables o acumuladores estadísticos.
- Reloj de simulación: se trata de una variable que almacena el valor del tiempo en la simulación.
- Inicio y parada del sistema: cómo se inicia y se detiene la simulación

6.4 Simulaciones en educación

Con el fin de relevar antecedentes de trabajos relacionados con el uso y/o construcción de simulaciones en educación, en febrero de 2016 se realizó una búsqueda en la base de datos Scopus, utilizando las siguientes claves:

(model OR simulation) AND

(school) AND

(software OR informatic OR computer OR programming) AND

("learning strategy" OR "teaching strategy")

Se obtuvo como resultado una lista de 128 publicaciones, cuyos primeros resultados datan de 1993, pero recién desde 2008 y hasta 2016 se registra el 75,69% de las publicaciones.

Respecto del campo temático, el 37,68% de las publicaciones pertenece a las ciencias sociales, el 26,57%, a ciencias de la computación, el 10,63% a ingeniería, y el 31,11% restante, a otras 18 áreas.

El 27,61% de los trabajos encontrados provienen de Estados Unidos, seguido de Taiwan (9,7%), Reino Unido, (7,46%), China (5,22%), Italia (5,22%), Israel (5,22%) y el restante 39,57%, distribuido en otros 25 países.

De los 128 resultados obtenidos se seleccionaron 17 trabajos relativos a experiencias con simulaciones informáticas en contextos de educación formal, significativas para esta tesis por coincidir especialmente en incluir la programación como objeto de estudio, como recurso didáctico o como contenido escolar, mientras que otros aspectos como la duración de las experiencias, la composición de los grupos de alumnos, o la relación con contenidos curriculares fueron divergentes.

De las 17 publicaciones, 12 se refieren a experiencias en que los estudiantes utilizaron simulaciones o las implementaron con recursos distintos de la programación, en tanto que 5 trabajos lo hacen respecto de experiencias en que los alumnos diseñaron e implementaron mediante programación, como estrategia de aprendizaje, una simulación.

En el primer caso las publicaciones responden a alguno de estos grupos:

- experiencias relativas al aprendizaje con videojuegos existentes o diseñados especialmente para la experiencia (*game-based learning*)
- los alumnos usan simulaciones de fenómenos biológicos
- se comparan aprendizajes por medio de a) *simulaciones* como único recursos didáctico, b) *simulaciones* combinadas con actividades de observación directa de la naturaleza, y c) sólo observación directa de la naturaleza
- comparación de aprendizajes resultantes de la construcción de modelos de fenómenos biológicos con recursos convencionales y mediante software de creación de modelos

Dentro del segundo grupo se encuentran trabajos con más aspectos en común con esta tesis, a los que se agregan otros encontrados por otros medios (Queiruga-Fava y Resnick-Brennan):

- En la Facultad de Informática de la Universidad Nacional de La Plata, Queiruga, C. y Fava, L. (2013), dirigieron un proyecto de extensión como parte de acciones de articulación con escuelas técnicas de orientaciones diversas de la Provincia de Buenos Aires, participando alumnos y docentes. Se enseñó programación y JAVA a fin de fortalecer la educación técnica, mejorar la

enseñanza de programación e incorporarla con un enfoque innovador, lúdico y social. La expectativa inicial de enseñar programación orientada a objetos y JAVA se reformuló trasladando estos últimos contenidos a 5º año y utilizando RITA (Robot Inventor to Teach Algorithms) por resultar más accesible a los adolescentes. El proyecto tuvo como resultado el interés de los docentes en continuar con este tipo de actividades, y más del 90% de los estudiantes participantes utilizó RITA sin obstáculos y consideró más fácil programar con este entorno frente a las interfaces tradicionales.

- Ante la observación de dificultades para trabajar con fórmulas físicas, Taub y otros (2015), implementaron un estudio a fin de medir la influencia de programar simulaciones de sistemas relativos a física, utilizando JAVA en un entorno amigable al usuario, con grupos de alumnos de décimo y undécimo grado provenientes de diferentes escuelas y seleccionados en función de su desempeño. Desarrollado semanalmente durante tres años, los resultados del estudio indican que la programación jugó un rol importante en el aprendizaje de física, incluso en diferentes aspectos del mismo.
- En un estudio acerca de la percepción sobre programación, Yohan Hwang, Kongju Mun, Yunbae Park (2016), pensamiento computacional y actitud sobre el aprendizaje de la ciencia en alumnos de secundario con Scratch (programación) y BitBrik (computación física), por medio de actividades de solución de problemas encontraron como resultado una mejora en la percepción de la programación y el pensamiento computacional, el pensamiento autogestionado y el compromiso e interés en la actividad misma.
- Lye, S.Y., Koh, J.H.L. (2014), en una reseña de 27 estudios sobre desarrollo del pensamiento computacional por medio de la programación, observaron la importancia de la programación, que más allá de la codificación, expone a los estudiantes al pensamiento computacional, útil en otros campos distintos del informático. Además, que la disponibilidad de lenguajes de programación gratuitos y fáciles de utilizar, significó un impulso a investigar la introducción del pensamiento computacional en contextos K-12. Por último, proponen que para fomentar este tipo de prácticas debería investigarse más y diseñarse

entornos de aprendizaje basados en el construcciónismo, e integrando procesamiento de la información, andamiaje y reflexión.

- En estudios sobre el diseño de medios interactivos realizado por jóvenes, Resnick y Brennan (2012), elaboraron un marco de referencia para el pensamiento computacional y cómo evaluar logros sobre el mismo. A partir de su interés por cómo la Programación puede ayudar a desarrollar el pensamiento computacional elaboraron una lista de siete conceptos computacionales comunes al mismo y a la mayoría de lenguajes de computación: secuencia, ciclo, evento, paralelismo de secuencias, condicionales, operadores y dato. También observaron estrategias de diseño en torno a las actividades que estudiaban, en relación al pensamiento computacional: ser incremental e iterativo, ensayar y depurar, reusar y remezclar, y en último término, abstraer y modularizar. Los niños y jóvenes participantes del estudio manifestaron experimentar cambios en tres ejes: expresarse usando la tecnología, conectarse e interactuar con otros como modo de enriquecer sus proyectos y formular preguntas y respuestas relativas a la tecnología y a partir de los aprendizajes derivados de programar.

Los trabajos comentados presentan como eje la programación en actividades educativas; como contenido mismo de la enseñanza en un caso, mientras que en el resto, se explora su rol de medio para otros contenidos, ya sea pertenecientes a disciplinas específicas o transversales, como es el caso del pensamiento computacional.

6.5 La construcción de simulaciones como recurso didáctico

Analizar una situación desde el punto de vista sistémico implica no sólo conocer el sistema, sus componentes y su ambiente descriptivamente, sino también sus propiedades y los procesos por medio de los cuales se relacionan y modifican dinámicamente entre sí y con el ambiente; en conjunto favorece, si no hace necesaria, la comprensión de la situación que se analice.

La construcción de un modelo se enfoca en los aspectos relevantes de un sistema para determinado propósito de estudio.

Si bien el campo donde las simulaciones por computadora proveen mayor utilidad es el de estudiar el comportamiento de modelos complejos, considerar la construcción de simulaciones como recurso de aprendizaje implica, para el alumno:

- la necesaria comprensión del sistema con el que se va a simular
- la abstracción de los elementos pertinentes a la modelización
- la utilización de recursos y habilidades relacionados al pensamiento computacional (por ejemplo, diseñar, programar, poner a prueba, evaluar, modificar el diseño) en la implementación de la simulación.
- representar el conocimiento e interactuar con esa representación, involucrando el ensayo/error, la puesta a prueba de hipótesis.
- Interactuar con otros (pares, profesores) en el desarrollo y en la comunicación de la actividad.

Desarrollar una aplicación para tal fin permite evaluar el funcionamiento de la simulación, intervenir en la misma y eventualmente hacer modificaciones para mejorarlala. Lo anterior es importante desde un punto de vista didáctico ya que el alumno (o grupo) realiza una tarea compleja e integral respecto del objeto de estudio, llevando a cabo un conjunto de habilidades cognitivas que abarca desde las más simples a las de mayor complejidad intelectual, y desde las más pasivas a las que le exigen al alumno más compromiso intelectual y puesta en juego de diversos tipos de habilidades.

7 Habilidades intelectuales y habilidades digitales

En los años 50 Benjamín Bloom (Churches, 2009) definió una taxonomía de objetivos educativos y habilidades intelectuales que se convirtieron en una herramienta clave para la planificación de la enseñanza. En esta taxonomía se ordenaban los objetivos o habilidades de pensamiento según un orden creciente de complejidad, comenzando con habilidades de pensamiento de orden inferior y terminando con las habilidades de pensamiento de orden superior. En esta secuencia, dominar una de estas habilidades implica el dominio de las precedentes. En los años 90, Lorin Anderson revisó la Taxonomía de Bloom y publicó en 2001 la Taxonomía Revisada de Bloom (Tabla 1).

Tabla 1: Taxonomía revisada de Bloom

Habilidades de Pensamiento de Orden Inferior	Recordar	Reconocer, listar, describir, identificar, recuperar, denominar, localizar, encontrar
	Entender	Interpretar, resumir, inferir, parafrasear, clasificar, comparar, explicar, ejemplificar
	Aplicar	Implementar, desempeñar, usar, ejecutar
	Analizar	Comparar, organizar, deconstruir, atribuir, delinear, encontrar, estructurar, integrar
Habilidades de Pensamiento de Orden Superior	Evaluuar	Revisar, formular hipótesis, criticar, experimentar, juzgar, probar, detectar, monitorear
	Crear	Diseñar, construir, planear, producir, idear, trazar, elaborar

Churches (2009) agrega a la taxonomía anterior una serie de habilidades propias del trabajo con las NTIC, entre las cuales están las simulaciones y la programación:

- el uso de simulaciones implica experimentar, monitorear, formular hipótesis, habilidades que Bloom (Churches, 2009) relaciona con la de evaluar, que ubica en el segundo lugar de las habilidades intelectuales de orden superior, con su correlato en la complejidad y valor de los aprendizajes que involucra simular.

- a su vez, el uso de Etoys implica en algún grado aprender a programar, lo que es clasificado por Churches también entre las habilidades digitales de orden superior, relacionadas con la habilidad de crear.

Los Etoys brindan recursos para el uso y la construcción de simulaciones por parte de los alumnos poniendo en juego las habilidades intelectuales más complejas constituyéndose en un recurso educativo de inestimable valor.

8 Aportes del Construcccionismo a la enseñanza mediante computadoras

Seymour Papert fue un educador y epistemólogo sudafricano, discípulo directo de Jean Piaget, fundador del laboratorio de Inteligencia Artificial del Massachusetts Institute of Technology (MIT) junto a Marvin Minsky en 1963 y creador del grupo de Epistemología y Aprendizaje en el MIT-MediaLab. Creó el lenguaje Logo, y su enfoque influyó en Squeak-Etoys, Scratch, y en el proyecto One Laptop per Child, entre otros muchos. Ha sido consultor de agencias gubernamentales de países de África, América, Europa y Asia y su trabajo en educación ha sido premiado en numerosas ocasiones.

Centró su trabajo no sólo en la enseñanza, sino en encontrar los medios para ayudar a que los niños aprendan mejor. Creó el concepto de *Construcción*, que define como dar a los niños buenas cosas para hacer de manera que el aprendizaje tenga lugar por el hecho de hacerlas, ya que pensaba que no sólo se debía enseñar a usar las nuevas tecnologías sino a fluir verdaderamente con las mismas, al aprender a diseñar, crear y expresarse a sí mismos con la tecnología; en lugar de sólo interactuar con la tecnología deberían aprender a programar sus propias animaciones, juegos y simulaciones, y en el proceso, aprender habilidades para resolver problemas y estrategias para el diseño de proyectos (Resnick, 2012).

En consistencia con el concepto piagetiano de que el conocimiento no es fruto de la memorización de información sino de la interacción del sujeto con el mundo, las personas y las cosas, sostiene que el aprendizaje se da indirectamente, al utilizar el conocimiento y no sólo memorizándolo. En otras palabras, la forma en que mejor aprendemos es a través de la acción de construir algo externo a nosotros mismos: construir una torre, escribir un cuento, construir un artefacto robótico, programar un videojuego, o realizar una animación, todos son ejemplos de construcción. Este tipo de actividades tienen valor como medio de aprendizaje en varios aspectos: están sujetas a lo que Papert llama “prueba de la realidad”: si no funcionan, son un reto para comprender por qué, y para superar los obstáculos. Pueden compartirse y discutirse con otras personas. Y sirven como objetos de transición para la apropiación personal de estas ideas (Papert, 1999).

Papert destaca dos formas de utilizar la tecnología: como medio de información y como medio de construcción. La primera es la que más percibe la sociedad, como reflejo de los medios de comunicación en sus vidas. Sin embargo, es el modo menos afín a un modo de aprendizaje significativo, a la vez que refuerza un estilo de enseñanza tradicional, en la que el conocimiento es algo que el alumno obtiene de una fuente externa en lugar de ser una construcción en la que articula sus habilidades y conocimientos con situaciones nuevas. Resnick y otros (2010) cuestionan que tal uso de la tecnología (que ejemplifica con el uso de mensajería, interacción con juegos en línea y la navegación en Internet) constituyan una competencia en el campo de las TIC; aún más, lo compara con saber leer, pero no escribir.

También señala estas dos facetas en la educación, dominio en el cual la constructiva está relegada frente a la informativa, y sostiene que las tecnologías digitales podrían corregir ese desequilibrio, y que es en ese sentido en el cual las tecnologías digitales pueden hacer a la educación su mayor aporte.

Con esa convicción respecto del aporte de las tecnologías digitales a la enseñanza en sentido constructivo, uno de los objetivos de su trabajo fue encontrar formas en que la tecnología permitiera a los niños aprender de esa manera y hace 40 años vislumbró que en el futuro los computadores serían omnipresentes y surgirían nuevos modos de aprendizaje en los que los profesores serían una parte activa, inventores de esos nuevos modos.

Como aspectos que interesan en este trabajo, se destacan los siguientes elementos de la propuesta construcciónista, muchos de los cuales están resumidos en las características de la “Filosofía Logo” ¹(Papert, 1999):

- En alusión a la velocidad de los cambios tecnológicos en el mundo actual, Papert afirma que el construcciónismo se enfoca en la adquisición de **habilidades necesarias para participar comprendiendo la construcción de lo que es nuevo**
- la **metacognición**, presente especialmente en toda actividad que requiera construir y hacer funcionar algo

¹ Papert (1999) aclaró que el nombre no implicaba restricción al lenguaje “Logo” sino a un enfoque de enseñanza-aprendizaje,

- La promoción de **la puesta a prueba de ideas nuevas**, la promoción del descubrimiento y la invención
- **el valor del error como una parte del proceso de aprendizaje** porque su comprensión conduce a corregirlo: "Equivocarse es una oportunidad para incrementar la comprensión de lo que se está tratando de hacer" (Papert, 1999).
- **la importancia de los saberes y modelos previos de los alumnos** ya que son los que permiten la asimilación de saberes nuevos, que deben tener sentido para el ahora, y no para un futuro hipotético como una "consignación bancaria".
- **el análisis y la confrontación de diferentes modos de pensar un problema** que se desea resolver, como focos principales, en lugar de llegar a la "respuesta correcta"
- **la programación es un elemento clave de esta cultura**, más allá del lenguaje que se utilice, lo que importa es el espíritu de hacer las cosas.
- **los niños pueden programar a edades muy tempranas y más allá de su edad y estrato social** pueden hacer mucho más de lo que se cree capaces si se les proporcionan herramientas y oportunidades para hacerlo, lo cual significa una cultura intelectual donde se estimulen los proyectos individuales y se facilita el contacto con ideas poderosas
- como se propician situaciones de aprendizaje que pueden ser nuevas aún para los docentes, Papert señala una **analogía con la figura del aprendiz**, de manera que si el objetivo de la enseñanza es que el alumno lleve a cabo un aprendizaje, como si fuera un aprendiz de su maestro, los maestros deberían aprender en presencia de los alumnos y en colaboración con ellos.
- **para los maestros el trabajo es más arduo pero mucho más creativo e interesante**, y Papert confía en que la mayoría elige creativo antes que fácil. También es cierto que para este tipo de trabajo los maestros necesitan tiempo y apoyo para aprender.

- la **capacidad de simulación de los computadores** como recurso educativo
- **esta filosofía** surgió como **expresión de la liberación del aprendizaje** respecto de las restricciones artificiales que le imponían las tecnologías del conocimiento pre-digital.

En resumen, Seymour Papert al elaborar el enfoque construccionalista sentó las bases de la enseñanza con tecnologías digitales, en consistencia con el constructivismo, reconociendo un mundo con cambios cada vez más rápidos, otorgando un lugar de esencial importancia a los intereses y singularidad del alumno en modo de aprender haciendo, para el que los saberes nuevos deben tener sentido actual, valorando la confrontación de modos de pensar diversos y alternativos, dando a la programación una posición clave, asumiendo que los alumnos pueden aprender a programar desde edades tempranas, donde se presentarán situaciones nuevas aún para el docente, que se transforma en un maestro que aprende a la par que sus alumnos, en analogía a la figura maestro-aprendiz y rescatando la capacidad de simulación de las computadoras como recurso educativo.

9 La importancia de enseñar Ciencias de la Computación

9.1 Las Ciencias de la Computación

Según el documento "CC-2016" de la Fundación Sadosky (2013), las Ciencias de la Computación conforman una disciplina definida como

- un cuerpo de conocimiento, ideas y conceptos sustentados por un marco teórico
- un conjunto de técnicas y métodos rigurosos que se aplican a la resolución de problemas y al avance del conocimiento
- una forma de trabajar y razonar que se complementa con teorías y conocimientos prácticos que incluyen técnicas de análisis, modelado y resolución de problemas; este tipo de pensamiento es transversal y aplicable a otros campos, incluso ajenos a la computación.
- un conjunto de conceptos subyacentes que permanecen estables a la vez que la disciplina avanza
- una existencia independiente de las tecnologías; la disciplina tiene más de 50 años, con principios y conceptos fundamentales vigentes que no necesitan de una tecnología específica para ser enseñados.

Entre los contenidos específicos de las Ciencias de la Computación se encuentran:

- la Algorítmica, relativa a la formulación de soluciones efectivas y sistemáticas de problemas
- la Programación, que trata la traducción de las soluciones algorítmicas a lenguajes de computadora
- el Almacenamiento de la información, que comprende las Estructuras de datos y Bases de datos

- Arquitectura de computadoras, acerca de sus componentes y su funcionamiento
- Redes de computadoras, el intercambio de información entre computadoras, que permite el funcionamiento de Internet y las aplicaciones que la utilizan (<http://program.ar/por-que-ciencias-de-la-computacion/>)

Kafai y Quinn (2014), señalan que dada la omnipresencia de computadores en todas sus formas (de escritorio, laptops, tabletas y celulares), se han convertido en una extensión de nosotros mismos, y la programación, en una época asunto de unos pocos eruditos, es hoy reconocida por los educadores como una habilidad crucial, una nueva alfabetización, una respuesta a la necesidad de desarrollar pensamiento computacional en los estudiantes. La premisa es que a través de este tipo de pensamiento los estudiantes pueden resolver problemas, diseñar sistemas para su uso en la vida diaria, y generar progresos e innovación en otras disciplinas (Kafai y otros, 2014).

Riesco y otros (2014) 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.

Citan como ejemplos el uso de un televisor, la configuración de una red hogareña, la configuración de un teléfono celular, la programación de electrodomésticos, la incorporación de conocimiento y el trabajo colaborativo en red, y la definición filtros de correo electrónico.

Todas actividades que requieren que el usuario debe tener la capacidad de entender y realizar operaciones complejas que implican conocer la filosofía de funcionamiento de una máquina programable y que en definitiva es la base del pensamiento computacional.

9.2 El pensamiento computacional

El pensamiento computacional es la actividad intelectual relativa a la formulación de un problema de manera que pueda admitir una solución por medios computacionales, a la vez que la solución pueda ser lleva a cabo por un ser humano o una máquina, o de ambas maneras (Wing, 2010). Este proceso de resolución de problemas involucra las siguientes características:

- formulación de problemas en un modo que habilite el uso de un computador para ayudarnos a resolverlos
- organización y análisis lógico de datos
- representación de datos a través de abstracciones tales como modelos y simulaciones
- automatización de soluciones por medio de una serie de pasos ordenados (pensamiento algorítmico)
- identificación, análisis e implementación de posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos.
- generalización y transferencia de este proceso de resolución del problema a otros problemas.

Estas habilidades son mejor desempeñadas con la concurrencia de las siguientes actitudes esenciales para el pensamiento computacional (CSTA, 2011):

- confianza al enfrentarse a la complejidad
- perseverancia en el trabajo con problemas difíciles
- tolerancia a la ambigüedad
- habilidad para lidiar con problemas de solución abierta
- habilidad para comunicar y trabajar con otros para lograr un objetivo o solución común

Wing (2010) afirma que el pensamiento computacional es la nueva alfabetización del siglo 21, al acercar recursos intelectuales de las ciencias de la computación a las necesidades de cualquiera, con los siguientes beneficios:

- comprender qué aspectos de un problema son susceptibles de ser digitalizados y automatizados
- evaluar la coincidencia entre técnicas y herramientas computacionales y un problema
- comprender las limitaciones de las técnicas y herramientas computacionales
- aplicar estas técnicas y herramientas de nuevas formas
- aplicar estrategias computacionales en cualquier campo

Mientras que para científicos, ingenieros y profesionales en general, se habilitan posibilidades para

- aplicar nuevos métodos (computacionales) a sus problemas
- reformular problemas para que sean susceptibles de aplicar estrategias computacionales
- descubrir nuevas "ciencias" a través del análisis de grandes cantidades de datos.
- hacerse nuevas preguntas antes impensadas o supuestamente sin solución debido a la escala, ahora fácilmente abordables con medios digitales.

Brennan y Resnick (2012), realizaron estudios sobre el diseño de medios interactivos realizado por jóvenes, a partir del cual elaboraron un marco de referencia para el pensamiento computacional y cómo evaluar logros sobre el mismo; interesados en cómo la Programación puede ayudar a desarrollar el pensamiento computacional elaboraron una lista de siete conceptos computacionales comunes a muchos lenguajes de programación, y presentes en el pensamiento computacional:

- secuencia: se refiere a una secuencia de instrucciones individuales que puede ejecutar un computador definiendo una tarea, actividad o comportamiento.

- ciclo: un ciclo es la ejecución de una secuencia múltiples veces.
- evento: un evento es un algo que puede ser detectado o registrado y utilizarlo para desencadenar una determinada actividad o tarea dentro de una secuencia.
- paralelismo: es la posibilidad de que se ejecuten simultáneamente más de una secuencia.
- condicionales: son elementos de un lenguaje de programación que permiten evaluar ciertas condiciones (valores, eventos, atributos de objetos en general) y en función de las mismas ejecutar distintas instrucciones de una secuencia.
- operadores: permiten realizar manipulaciones con los atributos de un objeto.
- dato: el concepto de dato implica poder guardar, recuperar y actualizar valores; un elemento básico en programación son las variables (que almacenan un solo dato, número, cadena de texto, u otro tipo de dato) y las tablas (que pueden almacenar listas de datos en una o más dimensiones).

Además de conceptos, observaron prácticas y estrategias de diseño que les permitieron describir mejor las actividades que estudiaban, considerando que las prácticas computacionales van más allá de qué se aprende, para centrarse en cómo se lo aprende; además, son prácticas aplicables también a otras actividades de diseño. Las estrategias que destacaron en sus observaciones son:

- ser incremental e iterativo: se refiere que diseñar un proyecto no es un proceso lineal sino que observaron que se da en ciclos en que se imagina el proyecto, se desarrolla, se prueba, se adapta, se sigue desarrollando...
- ensayar y depurar
- reusar y remezclar: se refiere a construir sobre lo que otros han hecho, reutilizándolo para proyectos propios
- abstraer y modularizar: alude a la construcción de algo integrando partes más pequeñas.

A partir de entrevistas con los jóvenes y niños que habían participado en la creación de proyectos, Resnick y Brennan encontraron tres ejes acerca de los cambios que los participantes manifestaron experimentar:

- expresarse: si bien estamos rodeados de medios interactivos, nuestra experiencia es como consumidores (por ejemplo: navegar, buscar información, chatear, usar mensajería). Lo cual no es suficiente para desarrollar pensamiento computacional, que necesita que la computación sea más que un objeto de consumo: además debe ser algo que pueda usarse para diseñar y autoexpresarse. Expresarse usando tecnología fue el primer hallazgo.
- conectar: los entrevistados reconocieron el valor de poder conectarse con otros y cómo enriqueció esto sus proyectos, tanto cuando creaban con otros (manifestaron que en forma colectiva fueron capaces de hacer más de lo que hubieran podido hacer individualmente) como cuando creaban para otros, que significaba que otros estaban pendientes de sus proyectos, los apreciaran, utilizaran sus proyectos para reutilizarlos en otros propios o para aprender más sobre Scratch.
- preguntar: se refiere a la capacidad de formular interrogantes y cuestionamientos sobre el mundo tecnológico y también ofrecer respuestas a partir de las herramientas aprendidas programando.

Lye y Koh (2014), tras el análisis de 27 estudios relativos a las tendencias en investigación empírica en desarrollo del pensamiento computacional a través de la programación, sugieren posibles implicancias para la instrucción. Observan la importancia de la programación, que trascendiendo la mera codificación, expone a los estudiantes al pensamiento computacional, aplicable y útil no sólo a los especialistas informáticos. Además, la disponibilidad de lenguajes de programación gratuitos y fáciles de utilizar, ha impulsado a investigar cómo introducir el pensamiento computacional en contextos K-12. A partir de la revisión realizada proponen que para fomentar prácticas y enfoques computacionales debería realizarse más investigación y diseñarse entornos de aprendizaje para la resolución de problemas basados en el constructivismo, con actividades que integren procesamiento de información, andamiaje y reflexión.

Yohan, Kongju y Yunebae (2016), en un estudio sobre la percepción sobre programación y pensamiento computacional y la actitud sobre el aprendizaje de la ciencia en estudiantes de secundaria por medio de actividades de investigación utilizando un lenguaje de programación educativa (Scratch) y componentes de computación física (BitBrik), diseñaron actividades para resolver problemas científicos, entre cuyos resultados se destacaron una mejora en la percepción de la programación y el pensamiento computacional, el pensamiento autogestionado y el compromiso e interés en las actividades.

La conexión entre programación y pensamiento computacional parece corroborarse en las fuentes consultadas; y se coinciden en niveles educativos equivalentes a la educación primaria y media de Argentina.

10 Iniciativas promoviendo las Ciencias de la Computación y la Programación en las escuelas

Actualmente se está revalorizando la enseñanza de las Ciencias de la Computación en el ámbito escolar y la enseñanza de la Programación y su uso como recurso educativo y como competencia general en distintos lugares del mundo. Un ejemplo de esto son iniciativas tales como el proyecto Raspberry Pi en la Universidad de Cambridge en el Reino Unido (<http://www.raspberrypi.org/about/>), Scratch, un proyecto del Laboratorio de Medios del Massachusetts Institute of Technology (<http://scratch.mit.edu/about/>) y Squeak-Etoys (<http://squeakland.org/about/>).

Code.org es una organización sin fines de lucro apoyada por las empresas de IT más importantes, dedicada a expandir la participación de la educación en Ciencias de la Computación y busca aumentar la participación en esta disciplina de las mujeres y otros grupos sociales que se encuentran actualmente subrepresentados; su visión es que los estudiantes deben tener la oportunidad de aprender Ciencias de la Computación y que la Informática debe ser parte del plan de estudios en la educación formal junto a las demás materias.

10.1 Iniciativas gubernamentales en Argentina

10.1.1 Experiencias basadas en el modelo de laboratorio

Son programas que se desarrollaron a partir de los años 90 con el fin de atender a sectores más vulnerables mediante la estrategia de garantizar mayor equidad en el acceso a herramientas digitales.

Este modelo de distribución y localización de equipamiento informático consistió en provisión de computadores, acondicionamiento y construcción de laboratorios de informática, y acciones orientadas a la capacitación docente e incorporación de TIC a la práctica educativa.

Programas

- Programa de Mejoramiento de la Enseñanza

- Media (PRODYMES II) y Plan Social Educativo (PSE), desarrollados en los años 90. Provisión de equipamiento.
- Programa RedEs (1988-1999). Armado de laboratorios de computación.
- Programa de Mejoramiento de la Enseñanza Media II (PROMSE)
- Programa de Apoyo a la Política de Mejoramiento de la Equidad Educativa (PROMEDU)
- Proyecto de Mejoramiento de la Educación Rural (PROMER)
- Programa Integral para la Igualdad Educativa (PIIE)
- Programa para el Fortalecimiento Pedagógico de las Escuelas beneficiarias del PIIE (FOPIIE)

10.1.2 El Portal Educ.ar

Creado en el año 2000 y relanzado en 2003 como portal del Ministerio de Educación de la Nación, tuvo como objetivo ejecutar las políticas del Ministerio en relación a la integración de TIC en el sistema educativo.

Está enfocado en brindar apoyo a docentes, directivos e instituciones por medio de:

- producción de contenidos multimediales
- capacitación en modo presencial y a distancia
- participación en la Campaña Nacional de Alfabetización Nacional

En 2010 Educ.ar tiene 1.800.000 visitas mensuales, produjo más de 50.000 recursos educativos, entregó 1.300.000 CD-ROM a docentes, y 5000 computadoras a 850 escuelas en el marco de un programa de reciclado para el aula.

10.1.3 Las TIC en la Ley de Educación Nacional N.º 26.206

La Ley de Educación Nacional sancionada en 2006, establece en su artículo 4º que el Estado Nacional, las Provincias y la Ciudad Autónoma de Buenos Aires tienen la

responsabilidad principal e indelegable de proveer una educación integral, permanente y de calidad para todos/as los/as habitantes de la Nación, garantizando la igualdad, gratuidad y equidad en el ejercicio de este derecho, con la participación de las organizaciones sociales y las familias.”

Y dispone, en su artículo 11, entre los fines y objetivos de la Política Educativa Nacional,

- garantizar una educación integral que desarrolle todas las dimensiones de la persona y habilite tanto para el desempeño social y laboral como para el acceso a estudios superiores (inciso b),
- desarrollar las competencias necesarias para el manejo de los nuevos lenguajes producidos por las tecnologías de la información y la comunicación (inciso m).

10.1.4 Experiencias basadas en el modelo 1 a 1 y aulas digitales

- a partir de 2008 se inician experiencias piloto del modelo 1 a 1 en escuelas primarias, distribuyéndose aproximadamente 1000 laptops en proyectos localizados en varias provincias y diversos contextos sociales.
- en 2009 el INET (Instituto Nacional de Educación Tecnológica) implementa el Programa Nacional “Una computadora para cada alumno”, aprobado por Resolución CFE N° 82/09 en escuelas técnicas públicas de gestión estatal de todo el país, con el objeto de incorporar las TIC como medio de enseñanza, aprendizaje, herramienta de trabajo y objeto de estudio en la Educación Técnico Profesional.

10.1.5 El Programa Conectar Igualdad

Por el decreto 459/2010, la Presidencia de la Nación Argentina crea este programa con el fin de:

- proporcionar una computadora a alumnas, alumnos y docentes de educación secundaria de escuelas públicas, de educación especial y de Institutos de Formación Docente,

- capacitar a los docentes en el uso de dicha herramienta
- elaborar propuestas educativas con el objeto de favorecer la incorporación de las mismas en los procesos de enseñanza y aprendizaje.

En la Resolución 123/10, la 32° Asamblea del Consejo Federal de Educación del Consejo Federal de Educación, aprueba en su artículo 1° el documento “*Las políticas de inclusión digital educativa. El Programa Conectar Igualdad*”, Anexo I de dicha resolución, que define los lineamientos políticos y técnicos para la incorporación de las TIC en el sistema educativo presentando los criterios que orientan la política educativa en TIC, las líneas estratégicas centrales para una política integral en la temática, y desarrollando la propuesta general para la planificación, articulación y gestión de líneas de acción del Programa, con el objeto de garantizar la igualdad de oportunidades de acceso y uso de las tecnologías.

10.1.6 Program.AR

Program.AR es una iniciativa de la Fundación Sadosky del Ministerio de Ciencia, Tecnología e Innovación Productiva. Surgida en 2012, su objetivo es llevar el aprendizaje significativo de Ciencias de la Computación a las escuelas argentinas, a partir de la importancia que tiene la comprensión del funcionamiento de la tecnología informática como herramienta para la construcción de ciudadanía en el presente (<http://program.ar/quienes-somos/>).

Este proyecto está articulado con más de 15 universidades de toda la Argentina como medio para alcanzar sus objetivos promueve la enseñanza de programación argumentando que

- las actividades escolares y profesionales, los vínculos afectivos y el tiempo de esparcimiento están permeados por dispositivos tecnológicos,
- las Ciencias de la Computación aportan conocimientos necesarios para comprender y decidir sobre ese mundo permeado por la tecnología.
- la programación es parte fundamental de las Ciencias de la Computación y su aprendizaje conlleva habilidades de pensamiento generales, como la

abstracción, la planificación, la descomposición de problemas y el trabajo en equipo. (<http://program.ar/de-que-se-trata/>)

10.1.7 La “Red de escuelas que programan”

La 65° Asamblea del Consejo Federal de Educación de la República Argentina estableció en el artículo 1° de la Resolución 263/15 que "la enseñanza y el aprendizaje de la “Programación” es de importancia estratégica en el Sistema Educativo Nacional durante la escolaridad obligatoria, para fortalecer el desarrollo económico-social de la Nación", aprobando en el artículo 2 la creación de la "Red de escuelas que programan", en el marco de la Iniciativa Program.Ar y el Plan Nacional de Inclusión Digital Educativa (PNIDE) que es responsable de las políticas de inclusión digital del Ministerio de Educación, incluyendo Conectar Igualdad y Primaria Digital (<http://programar.gob.ar/es-oficial-la-programacion-llegara-a-todas-las-escuelas-argentinas>).

10.1.8 “La hora del código”

En nuestro país, *La hora del código* es la iniciativa local que reúne el apoyo de Code.org y Program.ar, siendo éste último un proyecto dirigido en conjunto por la Jefatura de Gabinete de Ministros, la Fundación Sadosky, el portal Educ.ar y el Programa Conectar Igualdad. Al igual que Code.org, La hora del código ofrece recursos para aprender a programar de forma amena y divertida como una forma de aproximarse a las Ciencias de la Computación.

10.1.9 La importancia de enseñar Computación en las escuelas argentinas: informe de la Fundación Sadosky

Según el reporte de la Fundación Sadosky “Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas” (2013), la programación aporta, también, herramientas para dejar de ser meros consumidores del software, comprender cómo funciona éste y el mundo digital en general, y mejorar las habilidades intelectuales. Por esto se propone la enseñanza de la programación en las escuelas no sólo enfocada a la enseñanza de un lenguaje en particular sino al desarrollo del pensamiento computacional, incluso como una habilidad básica en nuestra sociedad, como lo es la resolución de ecuaciones matemáticas simples (Fundación Sadosky, 2013).

10.2 Experiencia de enseñanza de programación en escuelas medias de La Plata

Queiruga y Fava (2013) dirigieron un proyecto de extensión de la Facultad de Informática de la Universidad Nacional de La Plata realizado en el marco de acciones de articulación con escuelas técnicas de la Provincia de Buenos Aires destinado a estudiantes y docentes en el que se enseñó programación y JAVA con el fin de:

- fortalecer la educación técnica (se realizó en escuelas con orientación en programación, electrónica, mecánica y química)
- mejorar la enseñanza de la programación y acercar la programación a los alumnos con un enfoque innovador, lúdico y social
- mejorar la articulación del nivel medio con el nivel superior

Las actividades del proyecto consistieron en instancias de actualización docente en torno a programación orientada a objetos y JAVA y Eclipse (entorno de desarrollo), y generación de estrategias didácticas para la implementación de estos contenidos en el aula. En esta etapa del trabajo se especuló con usar con RITA (Robot Inventor to Teach Algorithms, RITA en el proyecto JETs, [<http://jets.linti.unlp.edu.ar/rita>]) para la iniciación de los estudiantes en programación, debido a su facilidad de uso para los adolescentes. RITA es un entorno de programación en el que se deben diseñar estrategias de combate de robots virtuales; integra OpenBlocks (OpenBlock, accesible en <http://education.mit.edu/openblocks>) y Robocode (Robocode, accesible en: <http://robocode.sourceforge.net/>) y el código se genera arrastrando y soltando bloques en forma similar a Etoys y Scratch.

También se produjeron materiales didácticos de apoyo a la enseñanza de programación en el nivel secundario para docentes y alumnos.

Como resultado del proyecto, se destacan

- interés de los docentes de ambos niveles en continuar con actividades similares en forma regular
- se postergó la enseñanza de JAVA al último trayecto de la escuela secundaria técnica (5º año)

- se utilizó RITA en el tercer año del Ciclo Básico
- las pruebas utilizando RITA se realizaron con estudiantes (masculinos y femeninas) seleccionados de entre 16 y 18 años provenientes de las 3 escuelas participantes del proyecto. Tras las pruebas más del 90% de los estudiantes encuestados no presentó dificultad en el uso de RITA y consideró más fácil programar con dicho entorno que con entornos tradicionales. En algunos casos se llegó a experimentar con circuitos lógicos

A partir de la convergencia de la resolución de problemas, el elemento lúdico (el combate de robots), y la programación, Queiruga enmarca su trabajo con el concepto de "juego serio", que define, citando a Sánchez Gómez (2007), como "objetos y/o herramientas de aprendizaje que poseen en sí mismos objetivos pedagógicos, didácticos, autónomos, autosuficientes y reutilizables, que posibilitan a los jugadores a obtener un conjunto de conocimientos y competencias predominantemente prácticos". El juego serio, afirma, es confluencia de la interactividad y la diversión, valorados por los jóvenes, a la vez que promueve la tolerancia a la frustración, la facilidad para relacionarse con otros, y la alta motivación al logro; todas características que hacen de los juegos serios "un elemento tecnológico ideal para transmitir conocimiento" (Queiruga, 2014:3).

10.3 Programación en la educación en otros países

Riesco y otros (2014), resumen el estado de las ciencias de la computación en algunos países:

- Inglaterra. A partir de 2014 los alumnos ingleses de primaria y secundaria, entre 5 y 16 años reciben educación obligatoria en Informática. En relación a programación y algoritmia, entre los 7 y los 11 años está previsto que diseñen y escriban programas, entre los 11 y los 14, que conviertan pensamiento computacional en algoritmos.
- Estonia En 2012 se lanzó un programa experimental en un grupo de escuelas para introducir la enseñanza de programación en todos los niveles de educación obligatoria, que abarcan habilidades necesarias para la

programación en los primeros años de educación, hasta lenguajes de programación en años superiores.

- Israel tiene en su currículo secundario 5 cursos optativos de informática, que incluyen "Fundamentos de Algoritmia y Programación", "Diseño de Software", "Segundo Paradigma", "Aplicaciones" y "Teoría (autómatas finitos)".
- En Estados Unidos los currículos son regulados por cada uno de los estados, en la mayoría de los cuales no se enseña informática o es optativa. Promovido por figuras como Bill Gates (Microsoft) y Mark Zuckerberg (Facebook) se ha gestado el movimiento Code.org con el objeto de introducir la informática en la educación, proyecto que tiene réplicas en otros países, entre los cuales está Argentina.
- Nueva Zelanda dispone de una materia llamada "Tecnologías Digitales" una de cuyas variantes es "Informática y programación".
- En España, la Ley Orgánica para la Mejora de la Calidad Educativa (LOMCE) aplicada desde 2014, reconoce la importancia de las nuevas tecnologías en la sociedad actual; en la educación primaria, se centra en la búsqueda de información o la utilización de aplicaciones mientras que en secundaria, se hace referencia al uso de aplicaciones específicas para cada asignatura y hay asignaturas optativas como "Tecnologías de la Información y de la Comunicación I y II" que incluyen hardware, redes, seguridad y programación.

Según el documento “Ciencias de la Computación 2016” de la Fundación Sadosky (2013:15):

- En Alemania, desde 2008 se diferencia en los diseños curriculares las Ciencias de la Computación y las Tecnologías de la Información y la Comunicación.
- En la educación secundaria en India se enseñan redes, programación, arquitectura de computadoras, etc., además de manejo de software de oficina.
- Corea del Sur incluye aspectos principalmente de algoritmia y programación, aunque en los nuevos diseños curriculares se hace más énfasis en conceptos fundamentales de ciencias de la computación.

En general, se hace énfasis en contenidos fundamentales de ciencias de la computación más allá del uso de aplicaciones y de tecnologías puntuales; y la programación aparece como un paso posterior.

10.4 Un estudio sobre la influencia de la programación en el aprendizaje de física

A partir de las dificultades que los estudiantes de secundaria tienen al tratar con fórmulas físicas, Taub y otros (2015), llevaron a cabo un estudio en el que se programaron simulaciones de sistemas relativos a física.

Bajo el supuesto de que la experiencia aumentaría la comprensión de fórmulas en física, se realizó la investigación en el marco de un curso de ciencias de la computación para alumnos talentosos de décimo y undécimo grado provenientes de diferentes escuelas; el curso fue desarrollado por profesores ajenos al equipo de investigación. En el curso los estudiantes aprendieron cómo programar modelos de fenómenos físicos, lo que implicó que aprendieran física, matemática, y ciencias de la computación; el software que usaron fue "Easy JAVA Simulations" (un entorno amigable de programación con JAVA) y "Maxima" (un sistema para la manipulación de expresiones numéricas y simbólicas). Mientras que la mayor parte del material fue aprendida en forma independiente por los estudiantes utilizando un libro de texto, los profesores tuvieron un rol de mentores. Se desarrolló durante tres años, en clases semanales de tres horas luego del horario escolar regular.

En consistencia con investigaciones previas, los hallazgos del estudio indican que la programación jugó un rol importante en el aprendizaje de física de los estudiantes, en diferentes aspectos: la representación visual de fenómenos físicos, el conocimiento estructural, el conocimiento procedural y el conocimiento sistémico.

11 Squeak-Etoys: un entorno de programación diseñado para la educación

11.1 ¿Qué es Squeak?

Según sus propios creadores (Ingalls:1997) Squeak es un entorno de desarrollo para construir software educativo, destinado a ser usado y programado por usuarios no técnicos, maestros y niños. Es una implementación actual de Smalltalk, abierta, portable, y traducida a varios idiomas, que funciona en una máquina virtual escrita completamente en el lenguaje Smalltalk. Incluye soporte para procesamiento de color, sonido e imagen independientemente del sistema operativo en que se utilice.

Según Ingalls y otros (1997), con la expectativa de fomentar el desarrollo de Squeak y el intercambio dentro de la comunidad de usuarios, se estableció un contrato de licencia que concede explícitamente el derecho a usarlo en aplicaciones comerciales libres de regalías, con el requisito de que cualquier portación de Squeak o cambios en la biblioteca de clases base debe estar disponible gratuitamente en Internet. Y las nuevas aplicaciones e instalaciones construidas con Squeak no necesitan ser compartidas.

A diferencia de otros software de autor, los proyectos que genera necesitan el intérprete (máquina virtual e imagen) para utilizarse.

Como entorno de autor multimedia permite construir presentaciones, realizar dibujos, grabar e incorporar sonido, texto, imagen y video; todos los cuales se incorporan como objetos modificables y a los que puede asignarse comportamientos (se usarán los términos guión, script, o programa para designar un conjunto de instrucciones de un lenguaje de programación dispuestos para el logro de un objetivo).

11.1.1 Un poco de historia

El entorno de Smalltalk está implementado en sí mismo: compilador, depurador, herramientas de programación y demás están escritas en Smalltalk, código fuente al que el usuario puede acceder, incluyendo las primitivas, y modificar. Una imagen

puede ser ejecutada en distintas plataformas gracias a una amplia base de portabilidad.

Squeak fue desarrollado siguiendo una serie de características:

- un entorno de desarrollo en el cual construir software educativo que pudiera ser usado y programado por no-técnicos y por niños
- un software que funcione en un amplio rango de dispositivos e Internet
- compacto y de bajo consumo de energía
- viable en un entorno de hardware diverso y plataformas cambiantes, aún prescindiendo de los sistemas operativos

Por lo anterior, el sistema ideal sería pequeño, con un kernel portable que pudiera ser rápidamente adaptable a nuevos destinos. Smalltalk reunía las condiciones técnicas pero no cubría las expectativas respecto de control sobre gráficos, sonido, la propia máquina virtual, ni la libertad de portabilidad y distribución del trabajo resultante (y su entorno anfitrión) por medio de Internet. A partir de esas consideraciones se decidió desarrollar un nuevo Smalltalk con esas características y compartirlo con su amplia comunidad (Ingalls y otros, 1997).

11.1.2 La interfaz de usuario de Squeak

La interfaz gráfica de Squeak se llama Morphic. Está escrita en Smalltalk, lo que la hace portable entre diferentes sistemas operativos, y por lo cual tiene la misma apariencia en Linux, MacOS y Windows. Esta interface gráfica tiene dos características: todo lo que se ve en la pantalla son objetos que pueden examinarse y modificarse directamente, por medio del ratón; además, la interface siempre está disponible para responder a las acciones del usuario en el sentido de que la información en la pantalla está continuamente actualizada de acuerdo a los cambios de los objetos que representa (Black y otros, 2007).

Cada objeto visual que compone la interfaz gráfica de Squeak se denomina *morph* y tiene las siguientes posibilidades:

- ser seleccionado y movido

- responder a acciones del usuario
- ejecutar una acción cuando otro morph es soltado encima o cuando éste es soltado sobre otro morph
- ejecutar una acción a intervalos regulares de tiempo
- controlar la posición y tamaño de sus submorphs

(Squeak Swiki, "What are Morphs?", <http://wiki.squeak.org/squeak/1003>, actualizado en agosto de 2016)

Morph, en tanto concepto de un entorno orientado a objetos puros, se define como la clase básica de todo lo que puede ser mostrado en el Mundo (el escritorio de Squeak) y con lo que se puede interactuar.

Ejemplos de morphs son: figuras geométricas, ventanas, cuadros de mensajes (Squeak Swiki, "Morph", <http://wiki.squeak.org/squeak/1820>)

El concepto de morph es importante porque en éste se basa la interfaz de usuario y al ser todos los objetos visibles y modificables en forma interactiva, al vehiculizar todas las acciones del usuario se obtiene un rápido feedback y se requiere poco esfuerzo para la autoría de cualquier tipo de proyecto, aplicación o material en general. Squeak hereda de Smalltalk la posibilidad de modificar un programa aún mientras está funcionando. Esto lo hace tan accesible para la iniciación de programadores noveles como para desarrollos de expertos (Web oficial de Squeak, "Features", <http://squeak.org/features/>).

La figura 1 muestra una captura de pantalla del escritorio de Squeak 2.8, versión que data del año 2000. Fue extraída del sitio "SqueakJS – A Squeak VM in JavaScript" en el que Bert Freudenberg implementó en JavaScript una máquina virtual de Squeak que puede correr imágenes de distintas versiones.

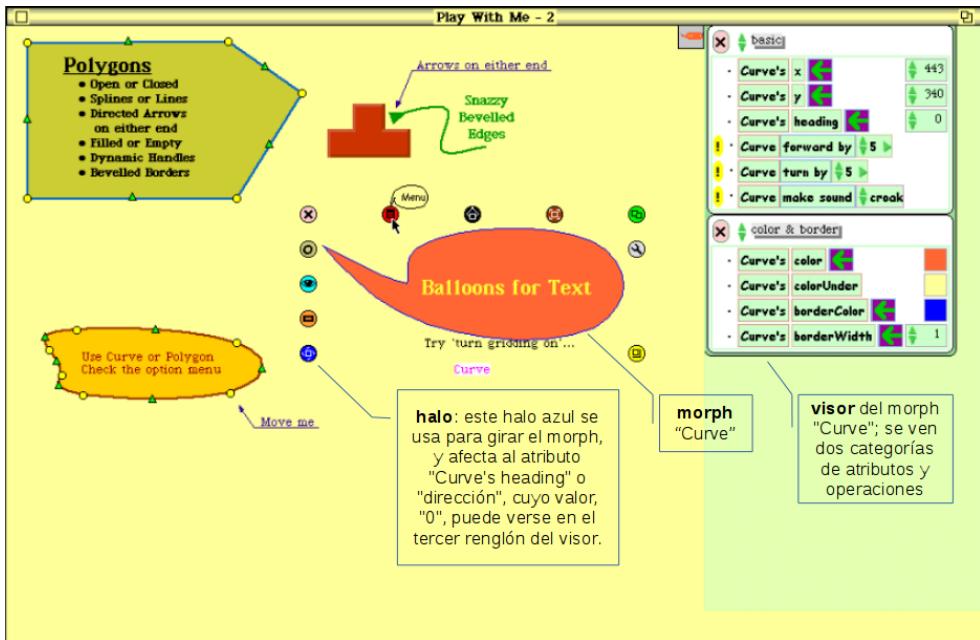


Figura 1: El escritorio de Squeak 2.8 (año 2000)

En el centro de la ventana puede ver el morph "Curve"; está seleccionado y por eso muestra

- su **nombre** ("Curve") debajo y centrado
- una serie de **círculos** de diferente color a su alrededor (llamados "**halos**"), con los que se realizan operaciones como obtener un menú contextual, cambiarlo de lugar, duplicarlo, redimensionarlo, o girarlo
- a la derecha, el **visor**, un panel con los atributos, variables, scripts y operaciones relativas al objeto.

Como se verá, esta característica (morph-halos-visor) se mantiene hasta las versiones actuales.

11.1.3 Portabilidad

La primera versión fue desarrollada en una computadora Macintosh (1996), y rápidamente miembros de la comunidad de usuarios lo portaron a numerosas plataformas, que en su momento fueron Windows 95 y NT, Windows CE, las más comunes variantes de UNIX, y desde agosto de 2016, su versión 5.1 está disponible

para macOS, Windows, Linux x86, Linux ARM, una versión portable llamada "All-in-One" multiplataforma, una versión experimental para procesadores de 64 bits, Raspberry Pi, y OLPC XO.

11.1.4 Campos de aplicación

Muchos proyectos se desarrollaron utilizando Squeak como base, abarcando un amplio abanico de campos, tales como educación, multimedia, gaming, investigación y comercio. Algunos de éstos se encuentran en la sección "Projects" de la página oficial de Squeak, entre los que se destacan parte del software de las laptops del proyecto OLPC (incluido Etoys), Scratch, Croquet, Pharo y Dr. Geo II (<http://squeak.org/projects/>).

11.2 Etoys

Squeak Etoys es un sistema de autoría multimedia especialmente dirigido a aprender ideas poderosas construyéndolas. Se desarrolló a partir de Squeak en Walt Disney Imagineering (la rama de investigación y desarrollo de The Walt Disney Company) y luego fue apoyado por el Viewpoints Research Institute (VPRI) como uno de sus proyectos.

Etoys se inspiró en LOGO, Smalltalk, Hypercard y Starlogo y presenta un estilo unificado, interfaz de usuario, medios de comunicación y entorno de programación para construir cosas con computadoras (Kay, 2007).

Las primeras menciones del término "Etoys" se hallan en el documento "Etoys and SimStories", escrito por Alan Kay y otros integrantes del ImagiLearningGroup de Walt Disney Imagineering en febrero de 1997 (Kay y otros, 1997c). Inspirado en el término de David Vogler "Dtoys" (acrónimo de *interactive digital toys*), "Etoys" se creó para referirse a "*interactive digital toys with an educational twist that are constructed by the children*" (Kay y otros, 1997b:3).

En ese documento se utilizó el término en varios sentidos, todos relacionados:

- para referirse a entornos informáticos que ayudan a aprender ideas construyéndolas y jugando con ellas (éste es el sentido que coincide con la caracterización de Squeak de acuerdo a otro documento de Kay de 1996)

- para referirse a los proyectos (simples simulaciones) construidas por niños acerca de una situación concreta; por ejemplo la animación de una "mariposa" que debe encontrar una "luz" y "volar" en esa dirección.
- también parece utilizarse el término Etoy para referirse a ciertos *morphs*.

En ese momento, y posiblemente a raíz de la primera acepción se usó indistintamente ambos términos, Etoys y Squeak, para referirse a Squeak. Con el tiempo hubo tres hechos que han desligado los términos:

- hubo una multiplicación de aplicaciones basadas en Squeak,
- Squeak evolucionó como entorno de desarrollo y estudio
- Etoys como entorno de autor fue introducido en las laptops de la iniciativa OLPC, y su imagen incorporó objetos (*morphs*) especialmente creados para fines educativos, fue orientado a niños, adolescentes, maestros y legos, y su interface fue simplificada a los mismos fines, aunque conservando ocultas muchas características avanzadas, a la vez que disponibles.
- También hubo otras adaptaciones de Squeak con fines educativos como la del Consejo de Educación de la Junta de Extremadura (España), actualmente abandonado.

En su última versión Etoys contiene algunos componentes licenciados bajo Apache License 2.0 y otras bajo MIT License (<https://opensource.org/licenses/mit-license.php>), por la cual se concede permiso a cualquier persona que obtenga una copia del software y archivos de documentación asociados para negociar el Software sin restricciones, incluyendo, sin limitación, los derechos de uso, copia, modificación, combinación, publicación, distribución, sublicenciar y/o vender copias (licencia de uso de Etoys 5.0).

11.2.1 Alan Kay, las ideas poderosas y el VPRI

Alan Kay es uno de los pioneros de la computación personal, la programación orientada a objetos, y las interfaces gráficas de usuario. Creó Smalltalk en el Xerox PARC, a partir del cual se desarrolló Squeak. PhD en Computer Science, fue elegido miembro de la American Academy of Arts and Sciences, entre otras academias y

asociaciones. Fue miembro de Xerox, Atari, Apple, Disney y HP. Influenciado fuertemente por Papert acerca de las posibilidades de las computadoras como ayuda en el aprendizaje, comprometió al Viewpoints Research Institute (VPRI) en el desarrollo de Squeak-Etoys y en la iniciativa OLPC (Squeakland.org, "Alan Kay", <http://squeakland.org/about/people/bio.jsp?id=1>).

En el prefacio de su libro "*Powerful Ideas in the Classroom - Using Squeak to Enhance Math and Science Learning*" (2003), BJ Allen-Conn y Kim Rose, en referencia al término "idea poderosa", acuñado por Seymour Papert, lo definen como aquella idea que los niños descubren tras confrontar ideas intuitivas en conflicto con el funcionamiento de algún aspecto de la realidad, tras la realización de experimentos y profundizando su estudio con la ayuda de simulaciones que ellos mismos construyen; Papert creía que la exploración de estas ideas mediante experimentación con computadoras y otras actividades sin computadoras podrían servir para confrontar sus intuiciones y reconsiderar su conocimiento intuitivo.

El Viewpoints Research Institute es una organización sin fines de lucro creada en 2001 con sede en Los Ángeles, E.E.U.U., y presidido por Alan Kay, con el objetivo de contribuir a la educación, la computación personal y la investigación en sistemas. Con el diseño de sistemas centrados en el usuario como motivación, el VPRI investiga en cuatro áreas: la enseñanza y aprendizaje de ideas poderosas, su contenido y cómo representarlo, interfaces de usuario como ayuda a aprender y a hacer, y la invención de nuevas tecnologías de computación. El Consejo de Asesores contó entre sus miembros a Vinton Cerf, Nicholas Negroponte, Mitchel Resnick, Jerome Bruner, Douglas Engelbart, Marvin Minsky, Seymour Papert y Neil Postman (Viewpoints Research Institute, web oficial, URL: <http://vpri.org/index.html>).

11.2.2 Generalidades

Integrado en el software de las laptops del proyecto One Laptop Per Child e incluido en los repositorios de las últimas distribuciones de Linux, Etoys es un entorno de trabajo que incluye objetos gráficos (texto, figuras geométricas, imágenes, sonidos,) que pueden ser manipulados, moverse, modificar su aspecto, sensar eventos, ser sensados por otros objetos, enviar mensajes a otros objetos, incluir también código de programa.

Todo esto permite, en definitiva y en coherencia con los objetivos que motivaron su desarrollo, construir modelos de simulación, actividad en la que los alumnos pueden aplicar y poner a prueba sus suposiciones y conocimientos diseñando modelos y construyendo simulaciones que “actúen como puente entre el aprendizaje concreto y el abstracto” (Caton, 2009). Toda esta potencia y una interface gráfica notablemente sencilla, contribuyen a que sea muy fácil aprender y construir, jugar, experimentar y sorprenderse con los Etoys.

Etoys tiene una interfaz de usuario totalmente gráfica, conceptualmente simple y especialmente diseñada para el uso en las aulas (<http://squeakland.org/about/>) en el cual la generación de código se realiza mediante instrucciones representadas gráficamente por mosaicos que incluyen la sintaxis y la documentación de ayuda, lo que facilita que el alumno y el docente puedan desarrollar actividades interactivas. Por otra parte esta aplicación es gratuita, multiplataforma, multilenguaje y en algunos sistemas operativos se integra en navegadores web.

11.2.3 La interfaz gráfica de Etoys

La interfaz gráfica de Etoys, se basa en la interfaz Morphic de Squeak, de la que hereda los elementos principales, especialmente un alto grado de usabilidad (figura 2).

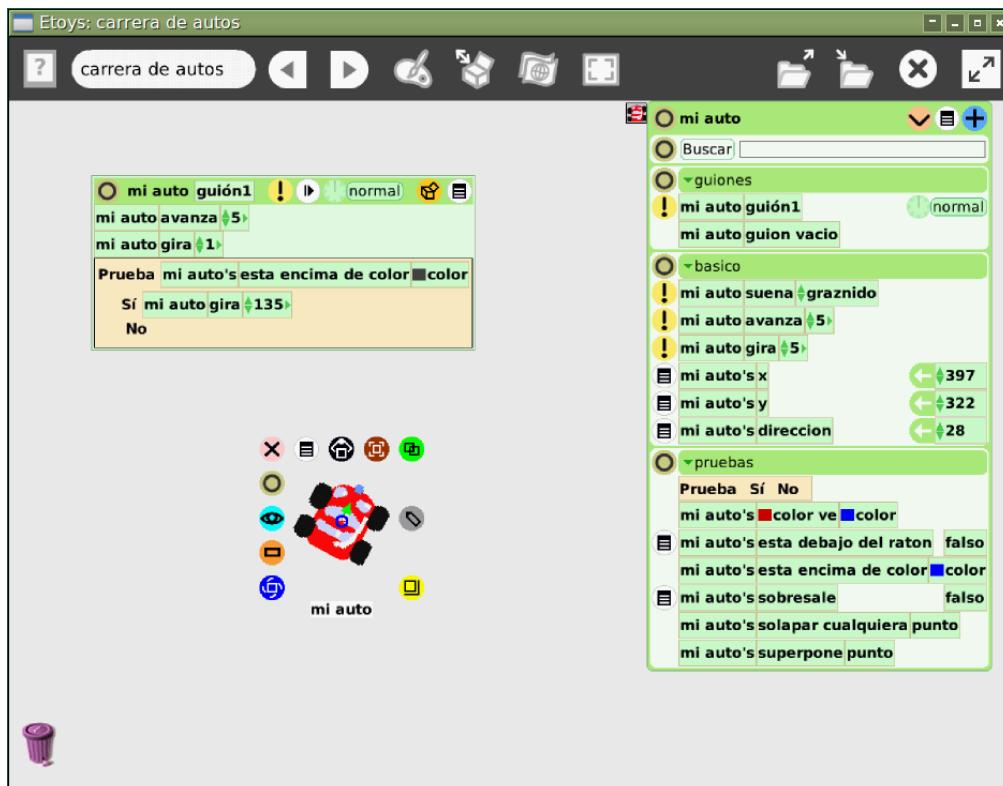


Figura 2: El escritorio de trabajo de Etoys 5.0

11.2.3.1 Barra de herramientas

La barra de herramientas es el punto de partida para las operaciones más comunes: muestra el nombre del proyecto, y botones para dibujar, obtener objetos para el proyecto, cambiar el idioma de los menús e instrucciones, para abrir y guardar proyectos, cambiar a modo de pantalla completa y salir de Etoys (figura 3).

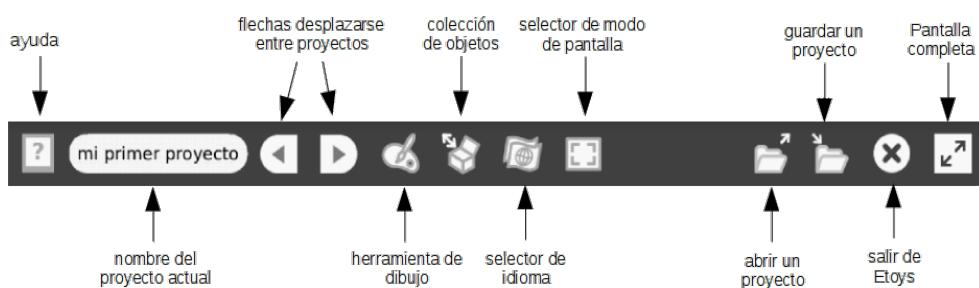


Figura 3: Barra de herramientas

11.2.3.2 Provisiones: de aquí obtenemos los objetos de nuestro proyecto

“Provisiones” (figura 4) es el nombre con el que se denomina a la colección de objetos-morphs que provee Etoys. Al pulsar el botón correspondiente en la barra de herramientas se despliega una lista con algunos objetos de uso habitual, que a su vez contiene el “Catálogo de objetos”, que da acceso a la totalidad de objetos, por medio de categorías, ordenados alfabéticamente, o escribiendo el nombre del objeto:



Figura 4: Panel de "Provisiones" desplegado

11.2.3.3 Objetos y *halos*

Al seleccionar un objeto se muestran su nombre (“auto”, en este caso) y los *halos* (figura 5), una serie de círculos de distintos color y funciones: desplegar un menú, mover el objeto, duplicarlo, cambiar su tamaño, rotarlo, abrir un “visor” del objeto que es un menú que muestra qué scripts o guiones tiene ese objeto, si se han agregado variables al mismo, sus atributos y procedimientos u operaciones que puede realizar el objeto, sus atributos y sus guiones. De acuerdo al tipo de objetos pueden variar los *halos*, en función de sus propiedades: por ejemplo, para los mapas de bits hay un halo para editarlos, y para textos, hay halos para modificar fuente y tamaño.

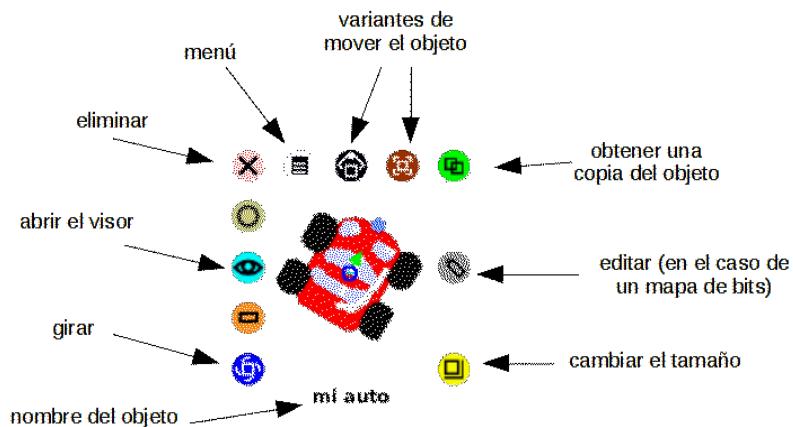


Figura 5: "Halos" que se hacen visibles al seleccionar un objeto

11.2.3.4 El visor: categorías, atributos y acciones

Los visores (figura 6) son la herramienta básica para generar código: proveen los mosaicos de instrucciones y atributos, y control de los guiones. A partir de un visor pueden modificarse los atributos del objeto y ejecutar instrucciones sin generar un guión.

En un visor, los mosaicos se agrupan en categorías, de las cuales algunas son propias de cada objeto ya que contienen atributos y acciones específicas; por ejemplo, el objeto "sector" tiene un mosaico que permite obtener y establecer el valor de su ángulo.

Otras categorías son comunes a todos los objetos, y se listan a continuación (en la mayoría de los casos, un mosaico permite leer y modificar el valor de un atributo, de acuerdo a cómo se utilice):

- Variables: variables que pueden agregarse a conjunto de atributos de un objeto: los tipos de datos incluyen boolean, color, gráfico, número, objeto, punto (coordenadas x,y), sonido y cadena de texto.
- Guiones: lista de guiones o scripts
- Básico: atributos más comunes (coordenadas x-y, hacia dónde apunta el objeto, avanzar, girar y reproducir un sonido)
- Color: lo relativo al color del objeto

- Geometría: atributos geométricos y de ubicación en el plano o en un campo de juegos
- Uso del lápiz: características heredadas de Logo, tales como bajar y subir el “lápiz”, definir tamaño, tipo y color del trazo.
- Pruebas: la estructura “si...entonces” y algunas pruebas relativas a la posición de objetos respecto del “mundo” o de otros objetos; por ejemplo, si un objeto se superpone con otro.
- Movimiento: atributos y órdenes relativos a la ubicación del morph en el espacio y su movimiento.
- Relleno y borde: tipos, colores.
- Manejo de guiones: control de la ejecución de guiones, por ejemplo, iniciar, pausar y detener guiones de un objeto, o también de un grupo de objetos; por ejemplo, si se simulara una población de hormigas, se podría controlar los guiones de todo el grupo con una sola instrucción.
- Sonido: reproducir sonidos en formato WAV, reproducir sonidos de determinada frecuencia
- Observación: características gráficas del entorno del objeto y de orientación respecto de otros
- Arrastrar y soltar: definición de la interactividad del objeto
- Miscelánea: mostrar, ocultar, borrar un objeto y ejecutar órdenes que no están en el visor, entre otras.

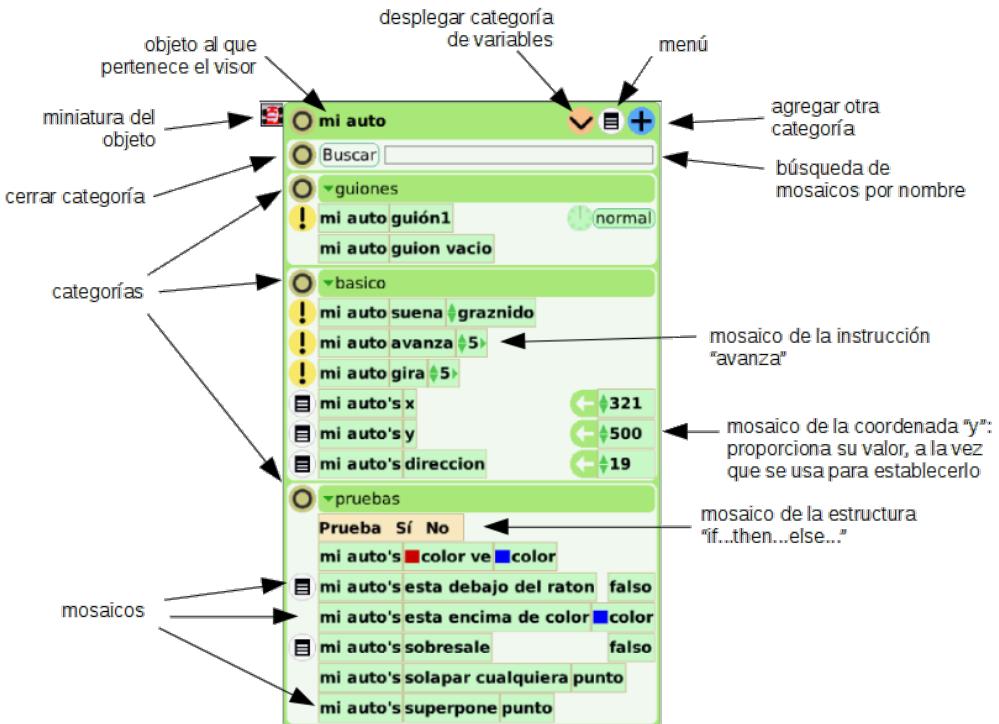


Figura 6: Visor del objeto “auto”

11.2.3.5 Guiones

Los *guiones* o *scripts* (figura 7) se construyen con mosaicos obtenidos de los visores; una vez dentro del guión, cada mosaico que representa una instrucción puede modificarse utilizando otros mosaicos. Las instrucciones están formadas por palabras en el idioma seleccionado, lo que facilita el trabajo de construir guiones, y familiarizarse gradualmente con toda la diversidad de objetos, y categorías de instrucciones y atributos. La secuencia de instrucciones de un script puede ejecutarse una vez o cíclicamente.

El guión siguiente hace avanzar al objeto “mi auto” 5 píxeles y girar 1 grado en sentido horario; si se superpone con algo de color gris, girará 135 grados:

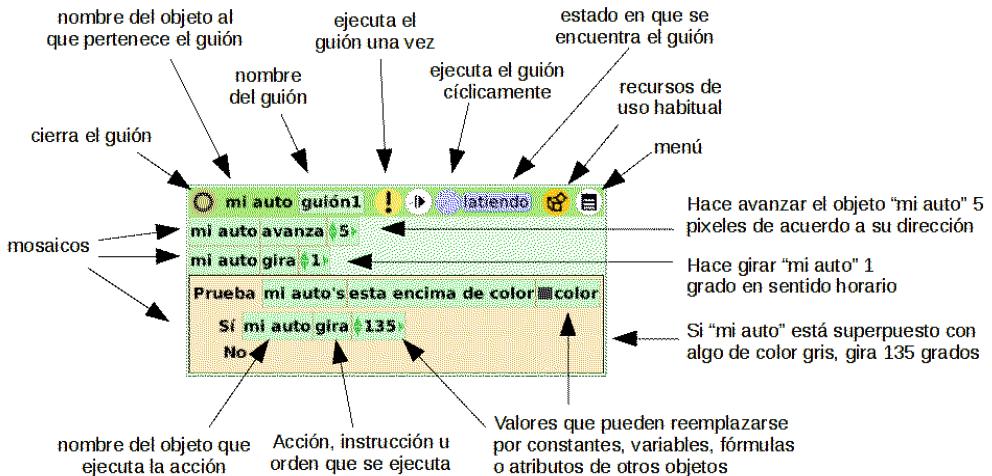


Figura 7: Script o guión, el lugar donde se crea la inteligencia de los morphs

11.2.3.6 Clases de morphs u objetos de Etoys

Etoys provee numerosos morphs u objetos para modelar y simular una idea; la siguiente es una muy incompleta muestra de los objetos incluidos, cuyas características son accesibles y modificables a través de sus visores o de guiones:

- Figuras geométricas (vectoriales): polígonos, círculos, líneas, flechas, curvas, elipses, sector de circunferencia
- Boceto: imagen de mapa de bits que puede editarse; entran en esta categoría las imágenes que se importen.
- Panel de control de ejecución de guiones
- Libro: una secuencia de páginas que pueden navegarse hipertextualmente, y que pueden contener objetos y ejecutar sus guiones, como si se tratara de distintos escritorios de Etoys dentro de un solo objeto.
- Cámara: es un monitor de cámara para hacer capturas de imágenes y video, directamente o controlado por scripts.
- Reloj analógico y reloj digital

- Conector: permite conectar dos objetos mediante una línea y los mantiene unidos aún cuando cambien de posición.
- Teatro de eventos: es un panel para grabar lo que se está realizando dentro de Etoys, útil para hacer tutoriales.
- Explorador de archivos de disco.
- Reproductor de videos MPEG
- Catálogo de objetos: un catálogo organizado de todos los objetos disponibles.
- Teclado de piano
- Campo de juego: un rectángulo que permite contener otros objetos funcionando dentro de sus límites y tratarlos en grupo como un conjunto.
- Preferencias (configuración) de Etoys
- Regla: para medir objetos en píxeles
- Texto: un cuadro para escribir o pegar texto
- Texto deslizable: permite contener grandes cantidades de texto
- Control deslizante: un control que proporciona un valor (configurable) en función de la posición de una perilla
- Biblioteca de sonidos: sonidos predefinidos para usar en proyectos, a los que pueden agregarse más, en formato WAV
- Grabadora de sonidos
- Papelera: una lugar que almacena los objetos que se eliminan
- Línea graduada, vertical y horizontal: pueden usarse como referencia de la posición de objetos respecto de su escala, que es configurable. Este recurso se usó en el trabajo de campo de esta tesis.
- Papel milimetrado y *plano x-y*: variantes del *campo de juegos*, que incluyen escalas graduadas y un fondo cuadriculado.

- una versión de Dr. Geo
- Burbujas de discurso o globos de ayuda: son globos a los que se agregan mensajes que funcionan como mensajes emergentes al pasar el ratón sobre el objeto.
- Calendario: un calendario del que se puede obtener información como buscar una fecha determinada directamente o mediante código
- Reproductor de partituras: reproduce archivos MIDI (Musical Instruments Digital Interface).
- Contenedor: es un espacio donde se pueden almacenar otros objetos, por ejemplo la secuencia de imágenes de una animación, y proporciona información tal como cantidad de objetos, la posición del cursor, que indica, por ejemplo, dónde se incluirían objetos o cuál se eliminaría.
- “Key press”: es un objeto que indica cuándo se ha pulsado una tecla específica.

La anterior es una lista de algunos de los objetos de uso más habitual, y es sólo una fracción del total. Por ejemplo, hay objetos para la gestión de los proyectos y los objetos que contiene, para facilitar la ejecución de guiones, o para la navegación de libros.

11.2.4 Repositorios de proyectos

Otro aspecto fundamental de Etoys que facilita la autoría docente es una colección de ejemplos de proyectos, clases y tutoriales incluidos en la misma aplicación y accesibles desde la pantalla inicial, y el repositorios on-line de proyectos, conformado por trabajos que comparten usuarios de todo el mundo. El hecho de que los proyectos de Etoys mantengan abierto su código hace que puedan reutilizarse, mejorándolos y adaptándolos a la propia necesidad educativa. La siguiente lista incluye algunos repositorios, comenzando por el oficial:

- La “Vitrina”, el acceso al repositorio de proyectos en línea del sitio oficial de Squeak-Etoys (<http://squeakland.org/showcase>)

- Proyectos con Etoys en la web del Plan Ceibal Uruguay (<http://www.ceibal.edu.uy/Busqueda/Paginas/results.aspx?k=etoys#k=etoys#s=11>)
- Biblioteca de proyectos “EtoysIllinois” de la Oficina de Matemáticas, Ciencia y Tecnología de la Universidad de Illinois en Urbana-Champaign, EEUU (<http://etoysillinois.org/library>).
- Software Architecture Group – Prof. Dr. Robert Hirschfeld, Universidad de Postdam, Alemania, <http://www.hpi.uni-potsdam.de/hirschfeld/projects/olpc/>
- También se pueden encontrar numerosas webs particulares de maestros y profesores que publican proyectos, como es el caso del Prof. Rodolfo Converso de la Provincia de Buenos Aires, Argentina (<https://sites.google.com/site/etoyseducativos>).

11.2.5 En resumen

La breve descripción anterior no pretende ser exhaustiva sino que tiene como fin destacar las usabilidad de Etoys y los recursos que proporciona. Su simplicidad es proporcional a la posibilidad de obtener resultados de gran calidad didáctica: al facilitar la tarea de construir el código del proyecto, habilita al usuario a concentrar su tiempo y recursos intelectuales, menos en cuestiones específicas de Programación, y más en el problema que debe resolver al implementar su simulación.

12 Marco metodológico

Se realizó un trabajo experimental consistente en evaluar el efecto del uso del modelado y la simulación en la construcción de un programa como recurso para el aprendizaje, mediante un protocolo pre-test / post-test en el que participaron un grupo control y un grupo experimental. Este último grupo realizó el trabajo experimental en el que se simuló un ejercicio de Física.

12.1 Etapas del trabajo experimental

El trabajo experimental se llevó a cabo en las siguientes etapas:

1. Búsqueda de docentes interesados en realizar la experiencia
2. Elección de los grupos experimental y de control
3. Elección de los contenidos para aplicar la propuesta didáctica
4. Realización del diagnóstico grupal y encuesta previa al trabajo experimental
5. Desarrollo de las clases con el grupo experimental
6. Encuesta posterior al trabajo experimental
7. Análisis de los resultados

12.1.1 Búsqueda de docentes interesados en realizar la experiencia

Se acordó con un profesor de Física la realización del trabajo de campo en dos de sus cursos de 4° año, a cada uno de los cuales se asignó el rol de grupo control y grupo experimental.

12.1.2 Elección de los grupos experimental y de control

Grupo control:

4° año “B” del turno mañana, de 30 alumnos (60% mujeres, 40% varones).

Grupo experimental:

4° año “D” del turno tarde, de 24 alumnos (79% mujeres, 21% varones).

12.1.3 Elección de los contenidos para aplicar la propuesta didáctica:

Se acordó la realización del trabajo de campo en el marco del contenido Movimiento Rectilíneo Uniforme, debido a que ofrece la posibilidad de construir una simulación que devuelva retroalimentación gráfica dinámica en tiempo real, además de juzgarse a priori viable por la cantidad de tiempo disponible. A priori se estimó una duración de entre dos y tres clases de 80 minutos, con una frecuencia semanal, para realizar el trabajo experimental, pero posteriormente, alteraciones en el calendario escolar hicieron que el lapso entre cada clase fuera mucho mayor.

Se eligió un ejercicio de “encuentro” (Cinemática), del cuadernillo de ejercicios del Departamento de Física, Biología y Química de la Escuela. El enunciado del ejercicio es el siguiente:

“Un auto pasa por la localidad “A” a las 8:00 a una velocidad de 40 km/h. A las 10:00 pasa otro auto en su persecución, a 60 km/h. Calcular a qué hora y a qué distancia de la localidad “A” el segundo auto alcanza al primero”

12.1.4 Encuesta sobre características generales del grupo y encuesta sobre Cinemática previa al trabajo experimental

El diagnóstico grupal (Anexo 1) se tomó a los grupos control y experimental antes de que realizaran la evaluación previa al trabajo experimental. Tuvo el objeto de conocer características del grupo en lo que refiere a uso de recursos tecnológicos digitales, forma de abordar problemas, conocimiento sobre modelado y su utilidad en el aprendizaje.

La encuesta sobre contenidos de Cinemática (Anexo 2), previa al trabajo experimental consta de preguntas sobre Movimiento Rectilíneo Uniforme, tema sobre el se realizará la simulación.

12.1.5 Realización del trabajo experimental

El trabajo experimental se realizó en el salón donde habitualmente tiene clase el grupo; se utilizaron netbooks del Plan Conectar Igualdad, en las que está instalado de fábrica Squeak-Etoys. A priori se previó que el trabajo se hiciera en equipos, especialmente para favorecer la cooperación ante una dinámica nueva.

El trabajo se apoyó en un tutorial interactivo diseñado para ofrecer la guía de trabajo y el espacio de trabajo en una misma ventana, de manera que se evitara cambiar de ventana continuamente. Las instrucciones, además, se imprimieron para ofrecer una fuente alternativa, por si se experimentaba alguna dificultad en la lectura, o si se deseaba eliminar la guía del tutorial para contar con más espacio en la pantalla (recordemos que las pantallas de las netbooks tienen sólo 10,1 pulgadas).

Debido a que los alumnos debían intervenir modificando y creando guiones y código, se esperaba que surgieran numerosos imprevistos, por lo que la guía fue redactada con la expectativa de que promoviera el trabajo autónomo y que la intervención del profesor quedara exceptuada de todo lo que pudiera brindarse de forma escrita, para estar dispuesta a la observación del trabajo de los grupos e intervenir en situaciones emergentes.

12.1.5.1 El tutorial interactivo

El tutorial interactivo se desarrolló en Squeak-Etoys, utilizando un recurso llamado “libro” que permite añadir páginas, en forma similar a una presentación multimedia, con la diferencia de que dentro de cada diapositiva también puede contener objetos con sus guiones; además se puede intervenir la presentación mientras se está “leyendo”: es lo que harán los alumnos en el trabajo experimental cuando intenten construir la simulación.

Se segmentó el ejercicio en una serie de problemas menores, en grado creciente de complejidad respecto del uso del entorno de programación a la vez que cada etapa progresaba hacia la resolución del ejercicio mediante la construcción de la simulación.

12.1.5.1.1 Formato del tutorial

En cada diapositiva o página del tutorial se presenta

- información sobre el ejercicio que se intenta resolver
- su articulación con conceptos de programación
- a la vez que se proponen desafíos relativos a su simulación, y pistas para su resolución, que se realiza creando o modificando guiones o scripts de programa dentro de cada diapositiva, sin salir del entorno del “libro”.
- al modificar o crear nuevos guiones al tiempo que se “lee” el tutorial dentro del mismo espacio visual (no hay distinción de modos autor-lector en Etoys), facilita el trabajo de los alumnos al recibir un feedback inmediato de su actividad.
- botones de navegación del tutorial y una barra de progreso que indica la posición dentro de la secuencia.

En el tutorial se usó texto en cuatro colores, para indicar diferentes tipos de mensaje:

- texto en color verde, se usó para redactar operaciones que los alumnos deben realizar con Etoys, como construir y modificar guiones, o configurar parámetros del auto.
- texto en color azul, para destacar reflexiones sobre el problema que se está resolviendo, o sobre los recursos de programación.
- texto en color rojo, para plantear desafíos y avanzar en la actividad.
- texto en color negro, para hacer referencia a conceptos propios de Etoys, como nombres de variables, instrucciones o de guiones.

En la página siguiente (figura 8) se muestra una de las diapositivas, con referencias a sus componentes y su funcionamiento.

Número de diapositiva

Tutorial interactivo

Botón para navegar el tutorial

Pestaña "Ayuda" con algunas indicaciones generales

guía de trabajo: consignas, explicaciones y desafíos de cada diapositiva del tutorial

"Arequito", la localidad "A" del ejercicio

escala graduada, referencia para situar objetos

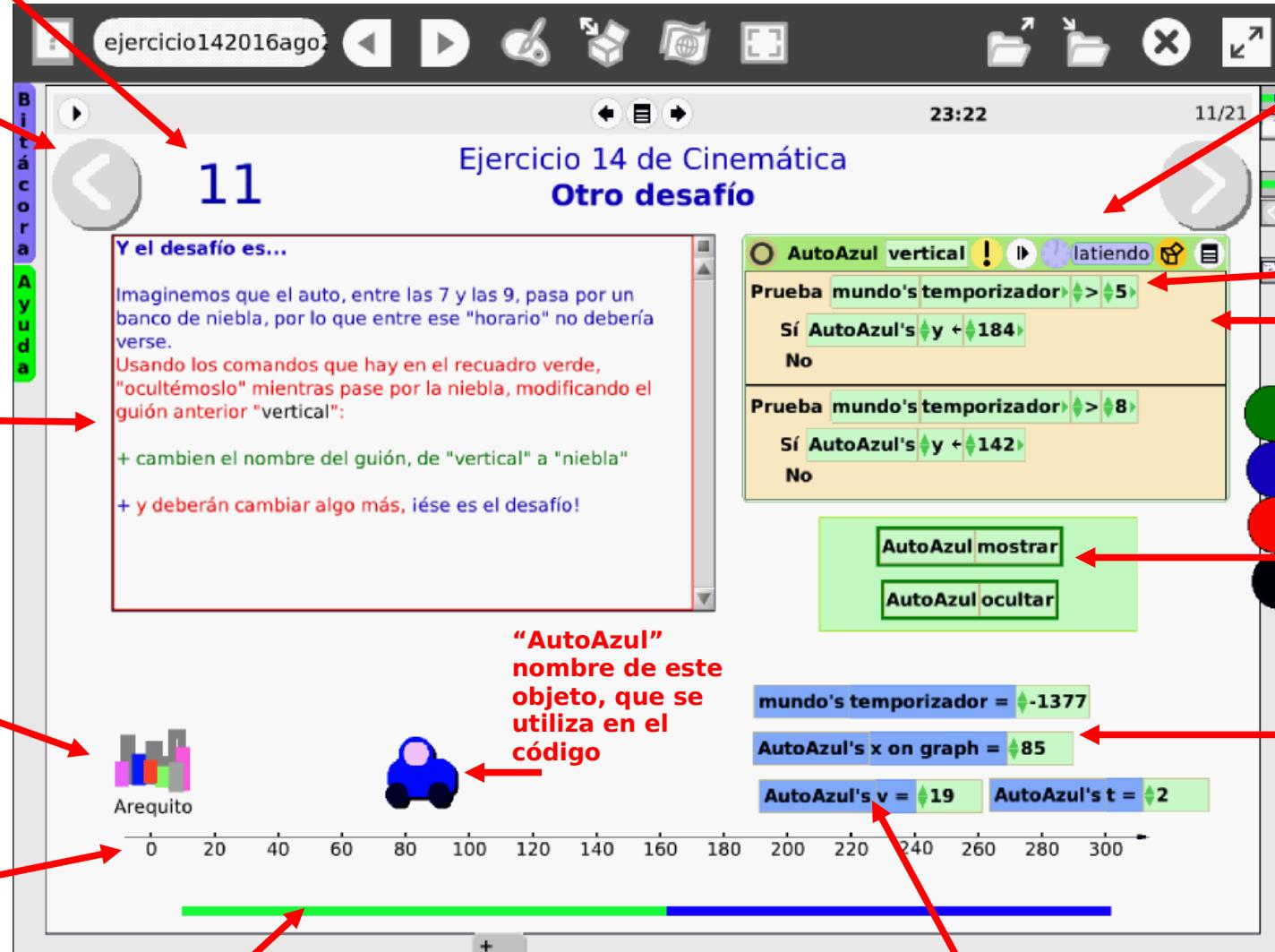


Figura 8: Diapositiva 11 del tutorial interactivo

Guión o script que indica el nombre del objeto, el del guión, controles para ejecutarlo, entre otras cosas.

prueba lógica

bloque que representa la instrucción si...entonces...si no...

mosaicos de instrucciones para "mostrar" u "ocultar" el auto Azul

posición del auto "Azul" en la escala graduada

barra de progreso del tutorial

mosaico de la velocidad del auto "Azul"

12.1.5.1.2 Material impreso

También se incluyó un ejemplar impreso de la guía de trabajo contenida en los cuadros de texto de cada diapositiva, para reducir la superficie de pantalla dedicada a la guía y aumentar la superficie disponible para el trabajo.

12.1.5.2 Sumario de diapositivas

A continuación se comenta en forma sintética cada diapositiva:

- Diapositiva 1: introducción al trabajo experimental: objetivo, en qué consiste la actividad, cómo se hará, y cómo funciona el tutorial
- Diapositiva 2 se analiza el ejercicio que se intentará resolver
- Diapositiva 3 se explica con más detalle el objetivo de esta actividad y qué subproblemas es necesario resolver para su logro
- Diapositiva 4 se muestra la situación de un objeto (el auto azul) sobre una escala graduada y se insta a arrastrar en sentido horizontal el auto usando el mouse y observar el valor de la coordenada horizontal.
- Diapositiva 5 se muestra un ejemplo de guión que establece el valor de la coordenada horizontal y como consecuencia, cambia la posición del auto
- Diapositiva 6 se muestra un ejemplo de asignación del producto de dos constantes a la coordenada x del auto
- Diapositiva 7 se utilizan atributos del objeto (en este caso, las variables "t" y "v", tiempo y velocidad respectivamente) en lugar de las constantes
- Diapositiva 8 se ejecuta el guión cíclicamente (hasta este momento sólo se lo hacía sólo una vez)
- Diapositiva 9 se introduce el uso del reloj del sistema en el guión para controlar el tiempo durante la simulación
- Diapositiva 10 ejemplo de salto condicional
- Diapositiva 11 uso de instrucciones que controlan la visibilidad de un objeto

- Diapositiva 12 construcción de guiones: se proponen un guión e instrucciones fuera del mismo y en forma desordenada y una meta a lograr con esos elementos
- Diapositiva 13 se aplica el caso anterior a mostrar, situar y mover el auto en función de la hora (1 segundo equivale a una hora) ya con datos propios del ejercicio
- Diapositiva 14 construcción de fórmulas utilizando paréntesis
- Diapositiva 15 revisión de lo realizado hasta este punto, confrontación con el ejercicio que quiere resolverse y anticipación del siguiente paso, la automatización de la simulación
- Diapositiva 16 creación de un nuevo guión que inicia la ejecución del guión anterior en forma cíclica
- Diapositiva 17 una vez logrado el comportamiento del auto azul, se enumeran las diferencias entre sus parámetros y los del otro auto que interviene en el ejercicio, señalando que del azul puede crearse una copia, y a ésta, cambiarle sus parámetros, su aspecto (es un auto verde)
- Diapositiva 18 se analiza el comportamiento del nuevo auto y se propone la modificación de sus guiones a fin de adecuarlo al ejercicio.
- Diapositiva 19 en esta diapositiva se propone simular la situación completa que se plantea en el ejercicio: que los dos autos "pasen" a la hora y velocidad correspondientes por el punto de referencia
- Diapositiva 20 se propone obtener las respuestas a las preguntas del ejercicio, a qué hora y a qué distancia del punto de referencia el segundo auto alcanza al primero, en analogía a cómo resolvieron situaciones anteriores, comparando los atributos de cada objeto (cada auto) y obteniendo los valores pertinentes en el momento en que ocurre el encuentro de los dos autos.
- Diapositiva 21 Saludo y agradecimiento por participar en la actividad

13 Resultados

13.1 La Escuela

El trabajo de campo se realizó en la Escuela Superior de Comercio “Libertador General San Martín”, dependiente de la Universidad Nacional de Rosario. La Escuela ofrece la orientación “Comercialización”. Situada en el centro de la ciudad de Rosario, Santa Fe, tiene una población de aproximadamente 1200 alumnos agrupados en comisiones de treinta, que se distribuyen en dos turnos (mañana y tarde), cinco años y cuatro divisiones.

La estructura académica de la Escuela agrupa espacios curriculares afines en Departamentos encabezados por un Jefe y un Adscripto o Coordinador de Área, que coordina cada espacio curricular en particular.

El espacio curricular “Física” está presente de 3° a 5° año, con una carga horaria de 4 horas cátedra semanales y contenidos (tabla 4).

Tabla 4. Unidades didácticas del espacio curricular “Física”

Tabla 2: Unidades didácticas del espacio curricular “Física”

Año	Horas semanales	Contenidos
3°	4	<ol style="list-style-type: none">1. Sonido2. Luz3. Electrostática4. Vectores5. Fenómenos magnéticos
4°	4	<ol style="list-style-type: none">1. Movimiento en 1 dimensión. Movimiento rectilíneo y uniforme. Movimiento rectilíneo uniformemente variado2. Dinámica3. Trabajo y Energía4. Presión
5°	4	<ol style="list-style-type: none">1. Termometría2. Dilatación de gases3. Calorimetría4. Equivalente mecánico del calor

El espacio curricular “Informática”: la Escuela cuenta con este espacio curricular de 1° a 4° año, con una carga horaria de 2 horas cátedra semanal en todos los casos. Los contenidos (tabla 5), no incluyen contenidos de Programación ni Algoritmia.

Tabla 3: El espacio curricular “Informática”

Año	Horas semanales	Contenidos
1°	2	1. Procesamiento de textos I 2. Hardware 3. Historia de la Computación
2°	2	1. Edición de imagen 2. Edición de sonido 3. Edición de videojuego 4. Hipertexto 5. Presentaciones multimedia
3°	2	1. Software 2. Procesamiento de texto II 3. Hoja de cálculo I
4°	2	1. Hoja de cálculo II 2. Sistemas 3. Bases de datos

13.2 Actividad experimental

13.2.1 Indicaciones previas a la actividad

Los siguientes criterios fueron señalados al principio de la primera clase, aunque se recordaron en forma general y particular, durante toda la experiencia, y cada vez que se consideró necesario:

Se señaló que habría cosas no explicadas, a propósito para que intuyan, imaginen y propongan (y lo pongan en práctica), o para que interactúen efectuando preguntas.

Se insistió en que la actividad funciona mejor con una actitud proactiva, por lo que se pidió a los alumnos que prueben, intenten, pongan a prueba este material y sus propias suposiciones, sin esperar a no esperar hacer todo "correcto"; ni esperar a tener una idea brillante para ponerse manos a la obra. ¡No se va a romper nada!

También se destacó que la actividad contiene indicaciones en cada página del tutorial para que cada grupo pueda trabajar a su propio ritmo, y cuando lo necesite, por supuesto, llame al profesor.

13.2.2 Desarrollo en el tiempo

El inicio de la actividad experimental estaba previsto para fines de abril, se esperaba una duración de 2 a 3 módulos de 80 minutos, con una frecuencia semanal, para finalizarla. Por imprevistos que alteraron el calendario escolar, la actividad se desarrolló durante tres clases, el 9 de junio, y el 5 y el 12 de agosto:

- i. 9 de junio, en 2 horas cátedra del profesor del curso (usamos el proyecto "ejercicio14tuto11.047.pr").

Notas de la clase: al comienzo había aún 3 netbooks para desbloquear (a pesar de que todos los días de la semana previa fui al curso para desbloquear las netbooks, y recordarles a los que aún las tuvieran bloqueadas). Para el desarrollo de la actividad se requería cargar el tutorial interactivo (un archivo de 1,1 Mb), por lo que se previó descargarlo de Internet, o desde dos pendrives, opción que se utilizó ya que no se disponía de WiFi y la señal de telefonía móvil no era suficiente para implementar un punto de acceso desde un celular. Se contaba con 10 netbooks, y la tarea de copiar los tutoriales insumió casi 40 minutos de los 80 de la clase, en parte, debido a que algunas máquinas no reconocían los pendrives (o tardaban mucho tiempo en hacerlo) y otras directamente no tenían la configuración de plataformas original y hubo que instalar Etoys.

En los poco más de 40 minutos en que se trabajó en la actividad, los 10 grupos que se formaron trabajaron evidenciando interés, con algunas distracciones pero superando más del 50% de la actividad.

Esta primera actividad incluyó la presentación de la actividad, sus objetivos, la forma de trabajar y el entorno de programación que se usó, sobre el cual el tutorial incluye indicaciones.

- ii. 5 de agosto, en 2 horas cátedra cedidas por una profesora del mismo curso, interesada en la experiencia, se continuó el trabajo utilizando una versión

modificada del proyecto usado en la clases anterior, donde se realizaron correcciones en la redacción quedando con el nombre "ejercicio14-2016ago25.pr" (por error dice "ago"; debería decir "jul" por el mes en que se hizo la modificación).

En esta clase, la mayoría llegó a la diapositiva 16; en ésta se duplica el auto que se construyó durante las instancias anteriores junto con sus características, incluidos sus guiones (llegados a esta etapa del trabajo lograron simular completamente el comportamiento del auto). Como los dos autos que intervienen en el ejercicio responden al mismo modelo de comportamiento, una vez terminado uno de los autos, sólo es necesario duplicarlo y modificar su nombre, los parámetros relativos a velocidad, a tiempo, y el aspecto (esto último, para diferenciarlo visualmente del primer auto). Casi todos, a la hora de modificar el aspecto (uno de los autos del ejercicio es azul y el otro es verde) hicieron un paréntesis para cambiar algo más que el color a su nuevo auto.

- iii. miércoles 12 de agosto en 2 horas libres de la misma profesora que cedió el espacio para el encuentro anterior. La mayoría de los grupos resolvió los desafíos de la penúltima diapositiva, en que simulan la situación planteada en el ejercicio; sin embargo no todos terminaron la actividad de la última diapositiva, donde debían obtener las respuestas del ejercicio sobre hora y distancia del encuentro entre los dos autos. Al llegar a la mitad de la clase, se realizó una evaluación del trabajo de cada grupo y al observar que la mayoría encontraba dificultades en los dos últimos desafíos se resolvió explicarlos a todo el grupo utilizando un proyector, para que posteriormente cada grupo intentara terminar su trabajo.

En las clases experimentales se evidenció motivación y se manifestaron inquietudes acerca de cómo cambiar otros aspectos de la simulación, que en principio no estaba previsto como parte de la actividad:

- cómo modificar el aspecto (colores, formas) de la simulación,
- simular otro tipo de movimiento (realizar cambios de dirección, describir curvas, por ejemplo).

También surgieron consultas sobre la posibilidad de crear simulaciones en otros espacios curriculares, sobre el entorno de programación utilizado y sobre cómo se aprende a programar.

Se observó que el aprendizaje de la interface de usuario, al inicio de la experiencia, insumió una cantidad de tiempo que disminuyó en las siguientes clases.

13.2.3 Encuesta posterior al trabajo experimental

En la encuesta posterior al trabajo experimental se indagan eventuales modificaciones en *a) los conceptos de modelo y simulación y b) la opinión sobre la utilidad de las simulaciones como recurso de aprendizaje*. La encuesta se encuentra en el Anexo 3.

13.3 Resultados de las encuestas

13.3.1 Grupo control

En este grupo, compuesto por un 62% de alumnas y 38% de alumnos, el 70,37% manifestó en el pre-test que había usado software en la escuela para representar situaciones que habitualmente se hacen con lápiz y papel, el 48,15% para resolver problemas y el 18% para construir aplicaciones. El 88,88% afirmó usar tecnología informática para resolver problemas cotidianamente y el 55,56% prefiere resolver problemas en forma grupal.

Frente a la pregunta sobre el concepto de simulación, en el pre-test el 29,63% contestó correctamente, mientras que en el post-test lo hizo el 46,67%. Respecto del concepto de modelo, en el pre-test contestó correctamente el 18,52% y el 40% lo hizo en el post-test. Se observó en el post-test una reducción de la incertidumbre respecto a estos conceptos, de 37,04% a 16,67% y de 40,74% a 6,67% respectivamente.

El 77,78%, en el pre-test, contestó afirmativamente acerca de si el uso y/o construcción de modelos y/o simulaciones podría contribuir a sus aprendizajes escolares, y en el post-test lo hizo el 76,67%.

En la evaluación de contenidos de Cinemática, en el pre-test, el promedio de respuestas correctas fue de 59,26%, mientras que en el post test, fue de 57,06%.

13.3.1.1 Tablas y gráficos

Tabla 4: Cantidad de alumnos en el grupo control

Evaluación PRE-TEST	27
Evaluación POST-TEST	30

Tabla 5: Cantidad de alumnos según sexo

	Femenino	Masculino	No contesta
PRE-TEST	17	10	0
POST-TEST	18	11	1

Tabla 6: Cantidad de alumnos según edad

	15 años	16 años	17 años
PRE-TEST	1	24	2
POST-TEST	0	25	4

13.3.1.2 Diagnóstico de características generales del grupo

Tabla 7: 1. En la Escuela, has usado software para... (marca con una “x”)

Representar situaciones que habitualmente se realizan con lápiz, papel, regla, compás, etc.?	70,37%
Resolver problemas?	48,15%
Usar simulaciones?	11,11%
Construir modelos o simulaciones?	7,41%
Desarrollar aplicaciones?	18,52%

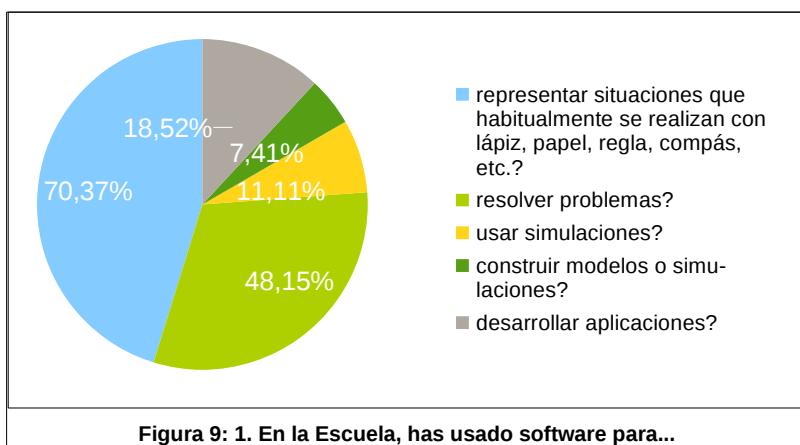


Tabla 8: 2. En la Escuela, has usado software para... (marca con una “x”)

Si	88,89%
No	11,11%

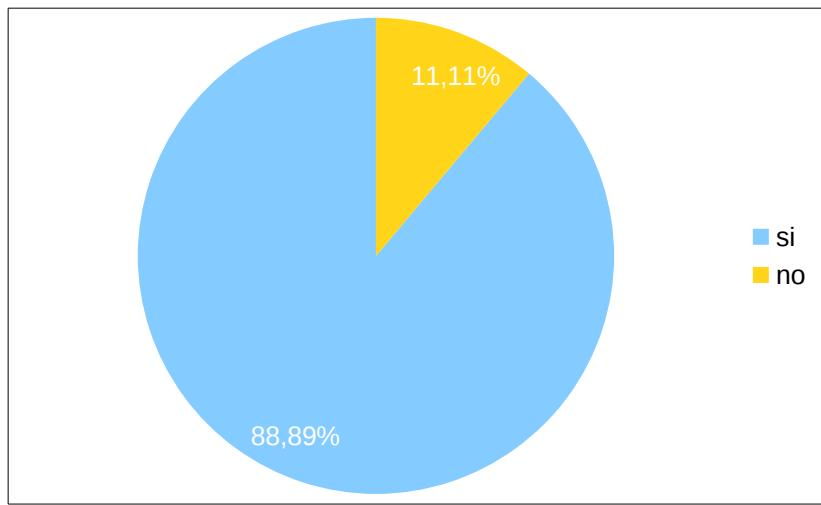


Tabla 9: 3. Ante un problema preferís resolverlo individualmente o junto a otros?

Individualmente	44,44%
Con otros	55,56%

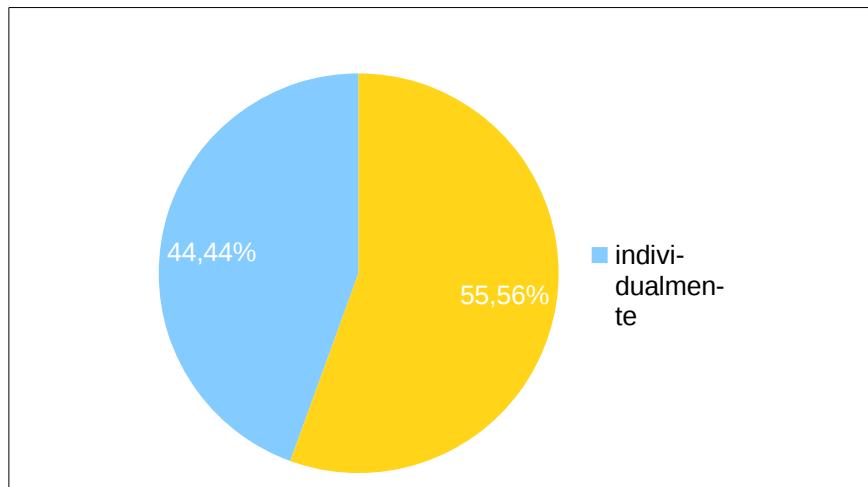


Figura 11: 3. Ante un problema, ¿preferís resolverlo individualmente o junto a otros?

Tabla 10: 4. Qué entendés por...

	Evaluación PRE-TEST			Evaluación POST-TEST		
	Correcto	Incorrecto	No sabe / no contesta	Correcto	Incorrecto	No sabe / no contesta
Modelo	18,52%	40,74%	40,74%	40,00%	53,33%	6,67%
Simulación	29,63%	33,33%	37,04%	46,67%	36,67%	16,67%

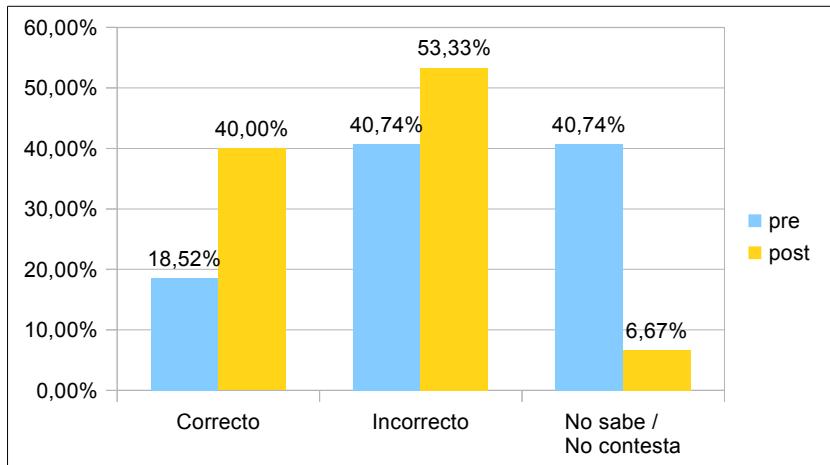


Figura 12: 4.a Noción de "modelo"

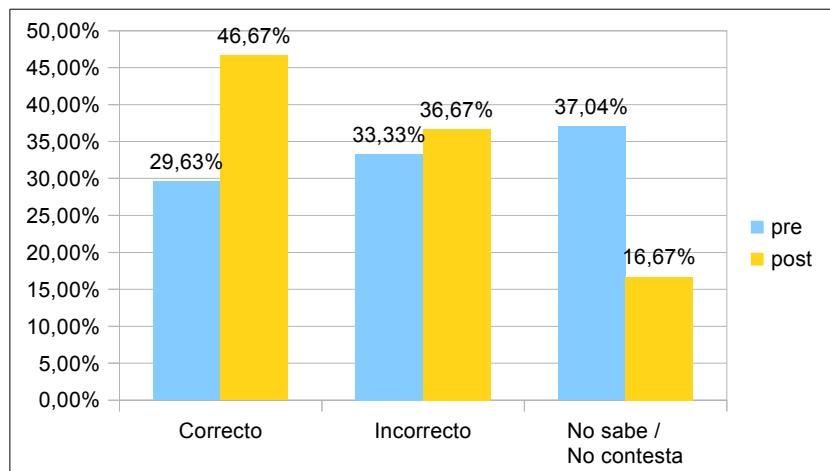


Figura 13: 4.b ¿Qué entendés por "simulación"

Tabla 11: 5. En tu opinión, pensás que el uso y/o construcción de modelos y/o simulaciones puede contribuir a tus aprendizajes escolares?

	Evaluación Pre-test	Evaluación Post-test
Si	77,78%	76,67%
No	0,00%	0,00%
No sé	22,22%	23,33%

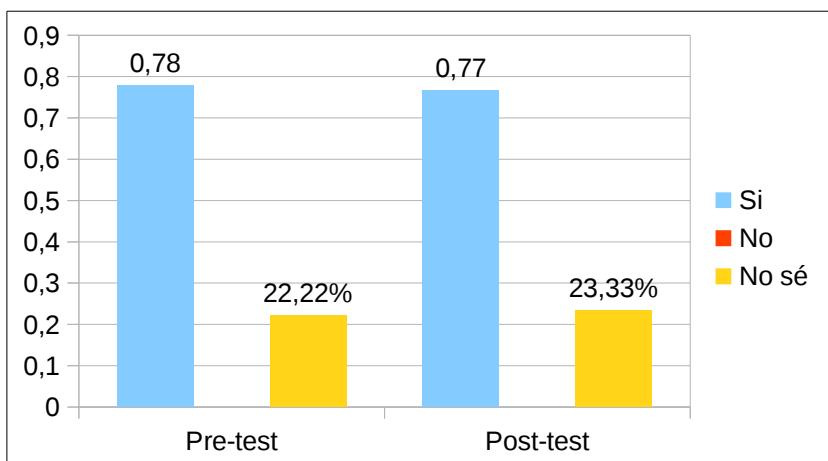


Figura 14: 5. En tu opinión, pensás que el uso y/o construcción de modelos y/o simulaciones puede contribuir a tus aprendizajes escolares?

13.3.1.3 Grupo control: evaluación de contenidos de Cinemática

Tabla 12: 1. Responde las siguientes preguntas: Si un tren recorre diez kilómetros cada diez minutos...

	Evaluación PRE-TEST			Evaluación POST-TEST		
	Correcto	Incorrecto	No sabe / no contesta	Correcto	Incorrecto	No sabe / no contesta
a.- ¿Cuántas kilómetros recorrerá en una hora?	85,19%	14,81%	0,00%	86,67%	10,00%	3,33%
b.- ¿Se puede asegurar que su velocidad es constante?	33,33%	62,96%	3,70%	43,33%	50,00%	6,67%
c.- ¿Se puede asegurar que su aceleración es nula?	33,33%	62,96%	3,70%	60,00%	33,33%	6,67%

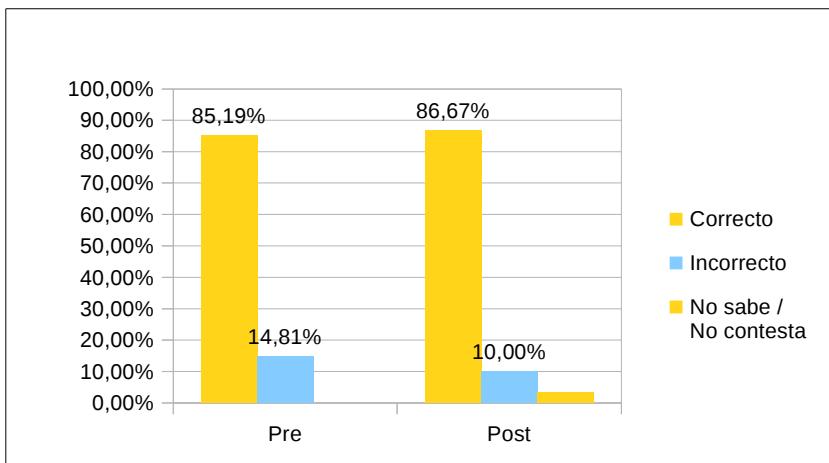


Figura 15: 1.a. Si un tren recorre diez km cada diez minutos, ¿cuántos km recorrerá en una hora?

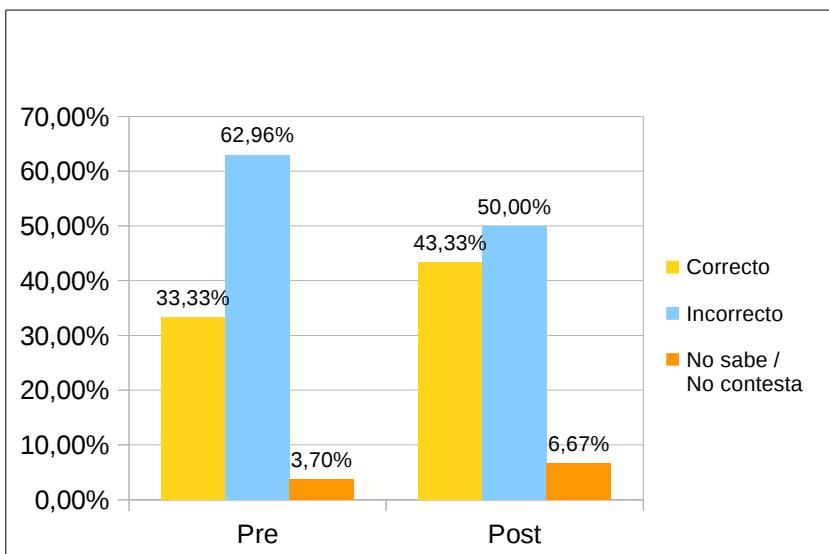


Figura 16: 1.b. Si un tren recorre diez km cada diez minutos, ¿se puede asegurar que su velocidad es constante?

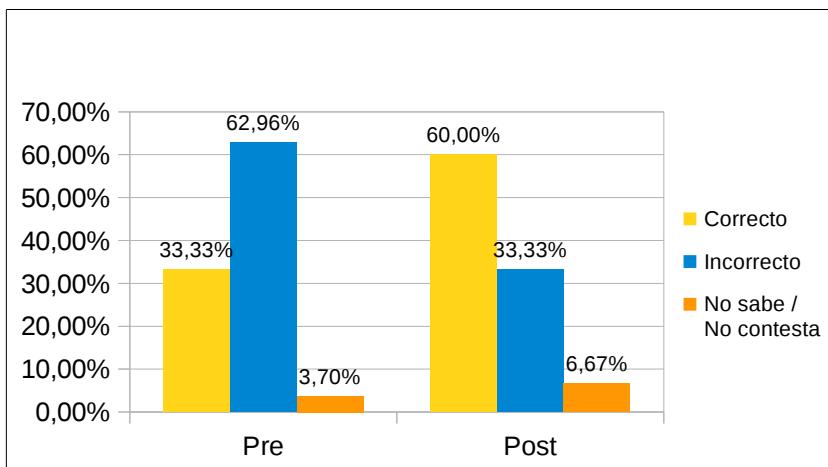


Figura 17: 1.c. Si un tren recorre diez km cada diez minutos, ¿se puede asegurar que su aceleración es nula?

Tabla 13: 2. Responde la siguiente pregunta. ¿Es posible que sea verdadera la siguiente proposición: “Una partícula recorrió una distancia de 100m, su desplazamiento fue nulo”?

Evaluación PRE-TEST			Evaluación POST-TEST		
Correcto	Incorrecto	No sabe / no contesta	Correcto	Incorrecto	No sabe / no contesta
74,07%	18,52%	7,41%	73,33%	13,33%	13,33%

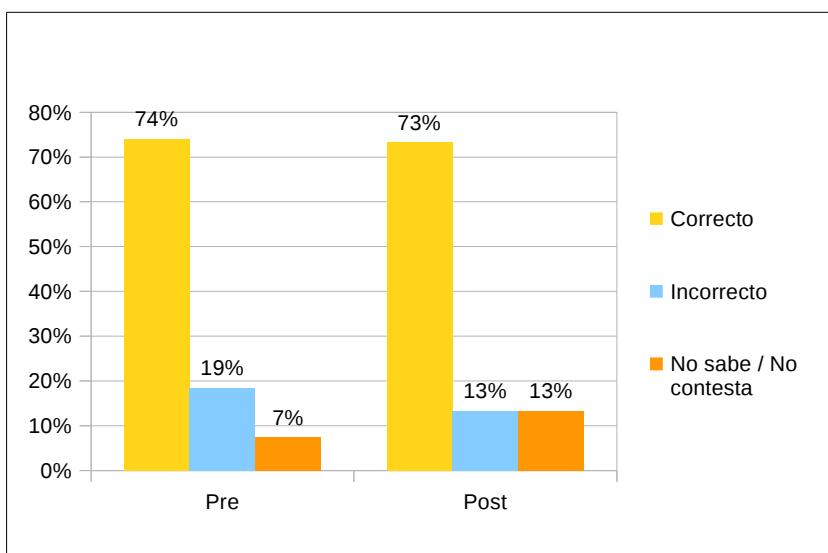


Figura 18: 2 ¿Es posible que sea verdadera la siguiente proposición: “Una partícula recorrió una distancia de 100m, su desplazamiento fue nulo”?

Tabla 14: 3. Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h.

	Evaluación PRE-TEST			Evaluación POST-TEST		
	Correcto	Incorrecto	No sabe / no contesta	Correcto	Incorrecto	No sabe / no contesta
a. ¿Cuánto tiempo tardará cada vehículo en recorrer 100km?	85,19%	3,70%	11,11%	76,67%	3,33%	20,00%
b. ¿Los dos vehículos tienen la misma velocidad?	25,93%	62,96%	11,11%	31,03%	51,72%	17,24%

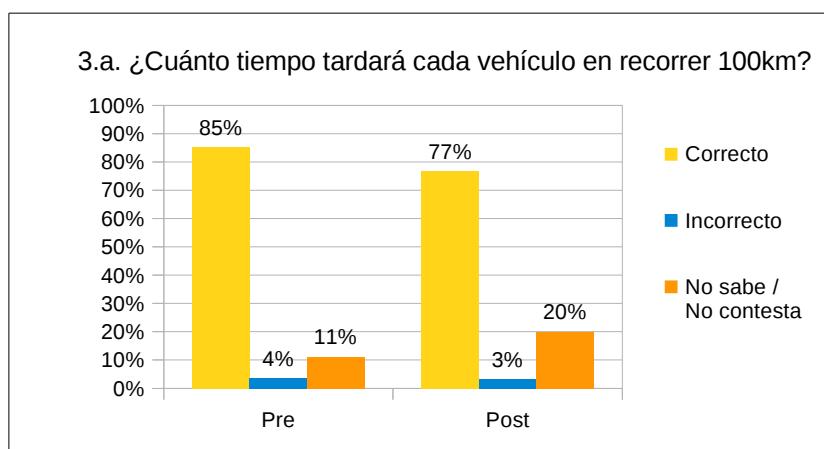


Figura 19: 3.a Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h. ¿Cuánto tiempo tardará cada vehículo en recorrer 100km?

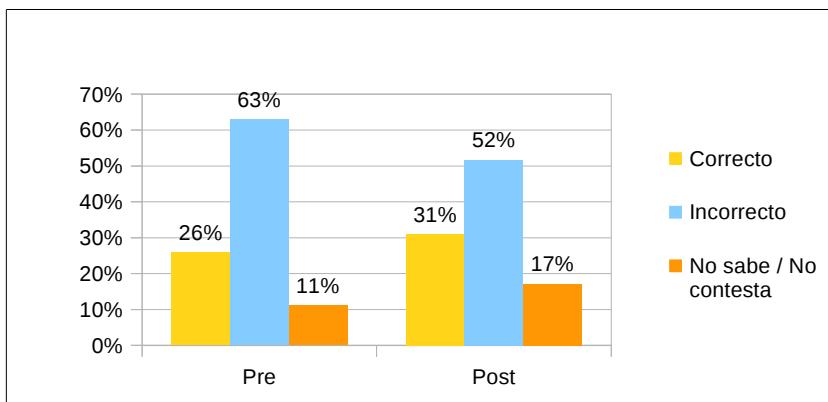


Figura 20: 3.b Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h. ¿Los dos vehículos tienen la misma velocidad?

4. Seleccione la opción correcta: En un Movimiento Rectilíneo Uniforme...

- a.- la velocidad media y la velocidad instantánea son iguales
- b.- el módulo de la velocidad media es mayor que el módulo de la velocidad instantánea
- c.- el módulo de la velocidad instantánea es mayor que el módulo la velocidad media
- d.- ninguna de las opciones anteriores es correcta

Tabla 15: 4. Seleccione la opción correcta: En un Movimiento Rectilíneo Uniforme...

	Correcto	Incorrecto	No sabe / no contesta
Evaluación PRE-TEST	66,67%	25,93%	7,41%
Evaluación POST-TEST	51,72%	41,38%	6,90%

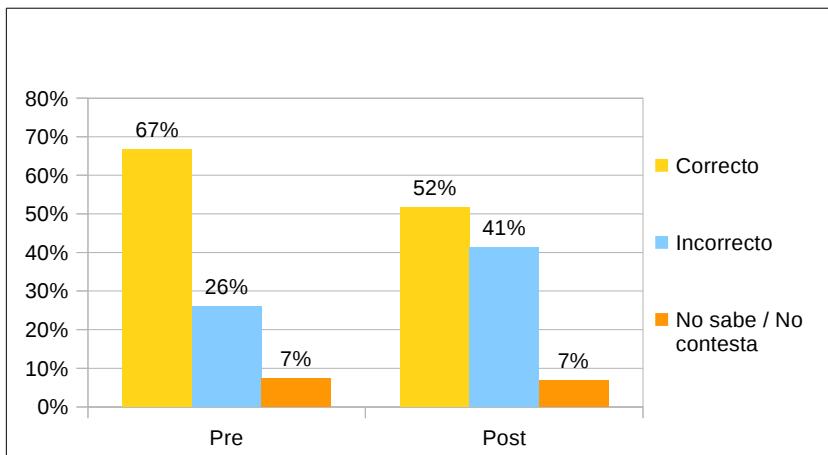


Figura 21: 4. Seleccione la opción correcta: En un Movimiento Rectilíneo Uniforme...

5. Seleccione la opción correcta. La distancia recorrida por una partícula...

- a.- siempre será menor que el módulo del vector desplazamiento
- b.- puede ser menor que el módulo del vector desplazamiento
- c.- siempre será mayor o igual que el módulo del vector desplazamiento
- d.- nunca será igual al módulo del vector desplazamiento

Tabla 16: 5. Seleccione la opción correcta. La distancia recorrida por una partícula...

	Correcto	Incorrecto	No sabe / no contesta
Evaluación PRE-TEST	70,37%	22,22%	7,41%
Evaluación POST-TEST	50,00%	40,00%	10,00%

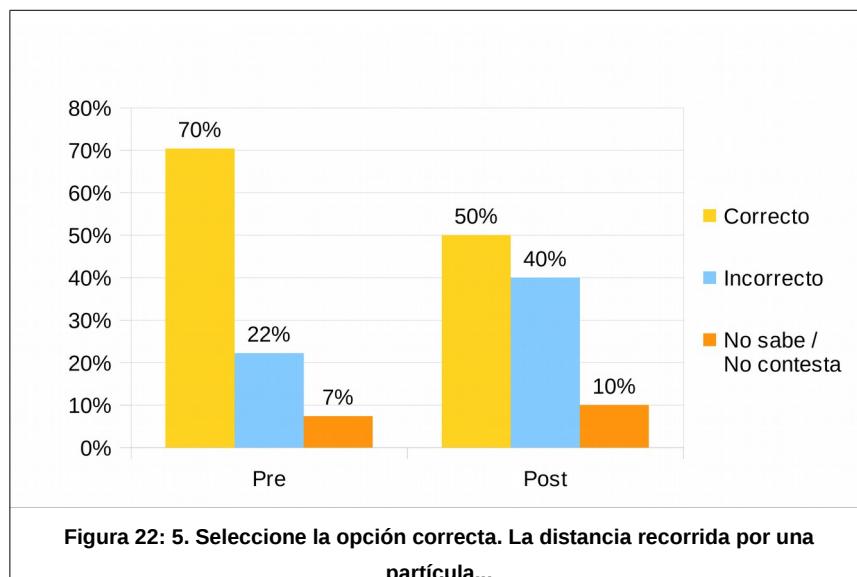


Tabla 17: Resultados generales por evaluación para el grupo control

	Respuestas correctas	Respuestas incorrectas	No sabe / No contesta
Evaluación PRE-TEST	59,26%	34,26%	6,48%
Evaluación POST-TEST	57,06%	37,89%	10,52%
Diferencia entre resultados Pre-Test y Post-Test	-2,20%	-3,63%	4,04%

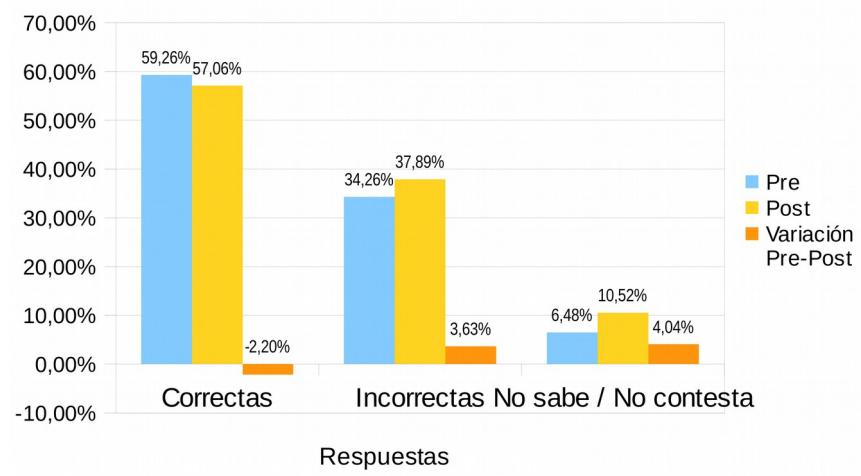


Figura 23: Resultados generales de las evaluaciones para el grupo control

13.3.2 Grupo experimental

El grupo experimental estuvo compuesto por un 79% de alumnas y 21% de alumnos, de los cuales en el pre-test, el 54,17% manifestó haber usado software en la escuela para representar situaciones que habitualmente se hacen con lápiz y papel, y el 37,5% para resolver problemas. El 66,67% afirmó usar tecnología informática para resolver problemas cotidianamente y el 70,83% prefiere resolver problemas en forma individual.

En cuanto a la pregunta sobre el concepto de modelo, el 25% contestó correctamente en el pre-test, mientras que en el post-test lo hizo el 36,36%. Respecto del concepto de simulación, las respuestas correctas en el pre-test fueron el 33,33%, siendo el 59,59% en el post-test. Se observó en el post-test una reducción de la incertidumbre respecto a estos conceptos, de 70,83% a 33,33% y de 58,33% a 8,33% respectivamente.

En la evaluación pre-test, acerca de si el uso y/o construcción de modelos y/o simulaciones podría contribuir a sus aprendizajes escolares, el 54,17 contestó afirmativamente y alcanzando al 72,73% estas respuestas en el post-test .

En la evaluación de contenidos de Cinemática, en el pre-test, el promedio de respuestas correctas fue de 56,25%, mientras que en el post test, fue de 60,42%.

13.3.2.1 Tablas y gráficos

Tabla 18: Cantidad de alumnos en el grupo experimental

Evaluación PRE-TEST	24
Evaluación POST-TEST	22

Tabla 19: Cantidad de alumnos según sexo

	Femenino	Masculino	No contesta
PRE-TEST	19	5	0
POST-TEST	16	5	1

Tabla 20: Cantidad de alumnos según edad

	15	16	17
PRE-TEST	2	14	6
POST-TEST	1	18	5

13.3.2.2 Grupo experimental: diagnóstico de características generales del grupo

Tabla 21: 1. En la Escuela, has usado software para... (marca con una “x”)

Representar situaciones que habitualmente se realizan con lápiz, papel, regla, compás, etc.?	54,17%
Resolver problemas?	37,50%
Usar simulaciones?	8,33%
Construir modelos o simulaciones?	0,00%
Desarrollar aplicaciones?	33,33%

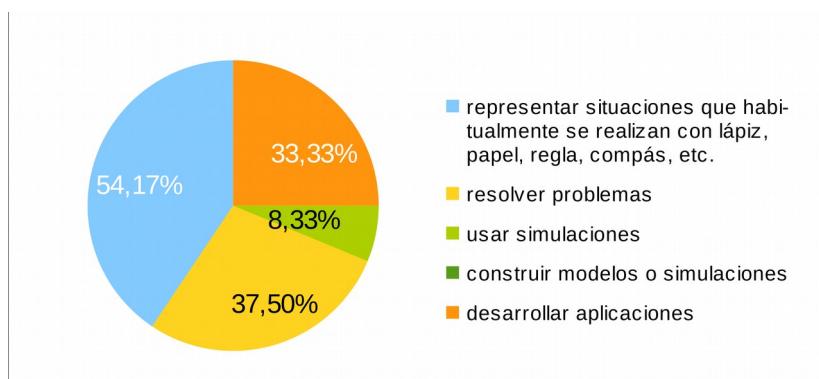


Figura 24: 1. En la Escuela, has usado software para...

Tabla 22: 2. Usás tecnología informática para resolver problemas cotidianamente?

Si	66,67%
No	33,33%

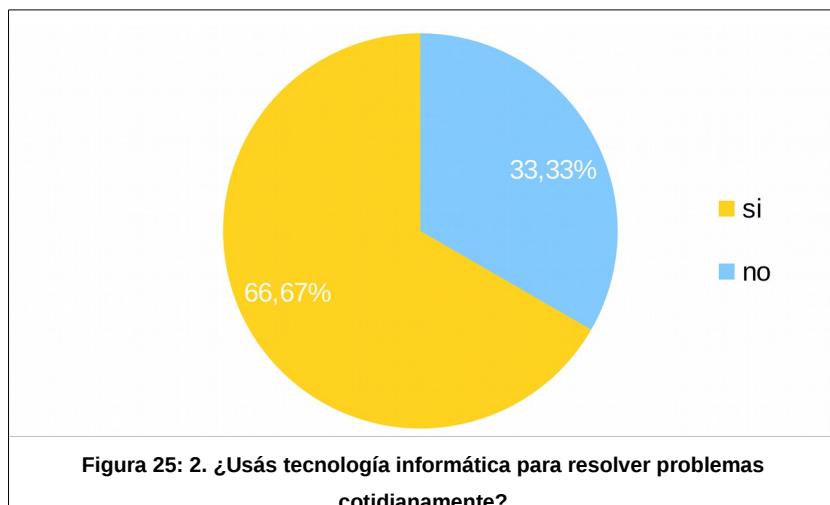


Tabla 23: 3. Ante un problema preferís resolverlo individualmente o junto a otros?

Individualmente	70,83%
Con otros	29,17%

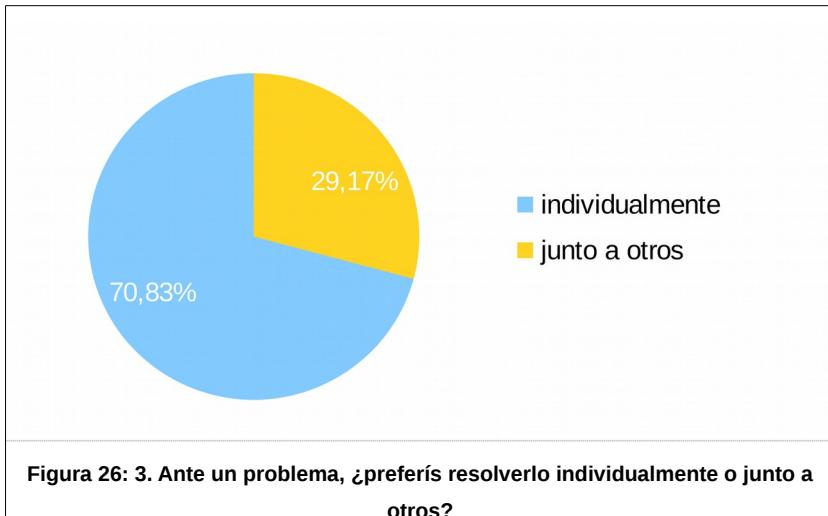
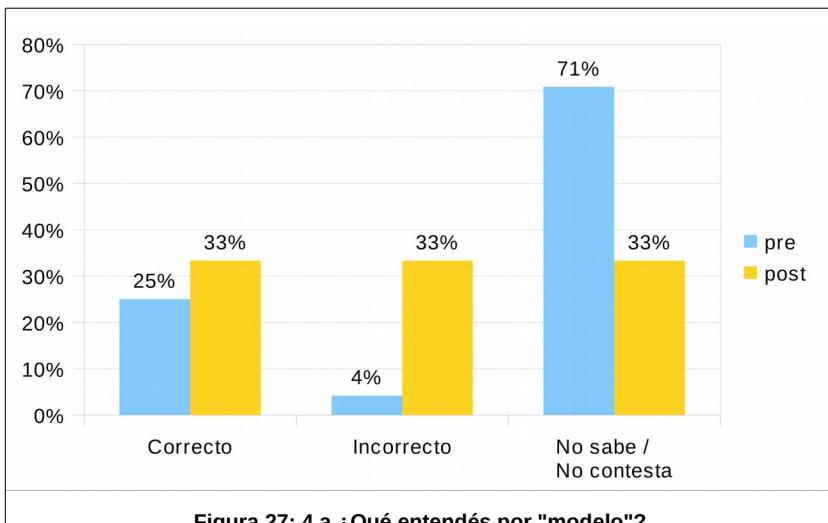


Tabla 24: 4. Qué entendés por...

	Evaluación PRE-TEST			Evaluación POST-TEST		
	Correcto	Incorrecto	No sabe / no contesta	Correcto	Incorrecto	No sabe / no contesta
Modelo	25,00%	4,17%	70,83%	33,33%	33,33%	33,33%
Simulación	33,33%	8,33%	58,33%	83,33%	8,33%	8,33%



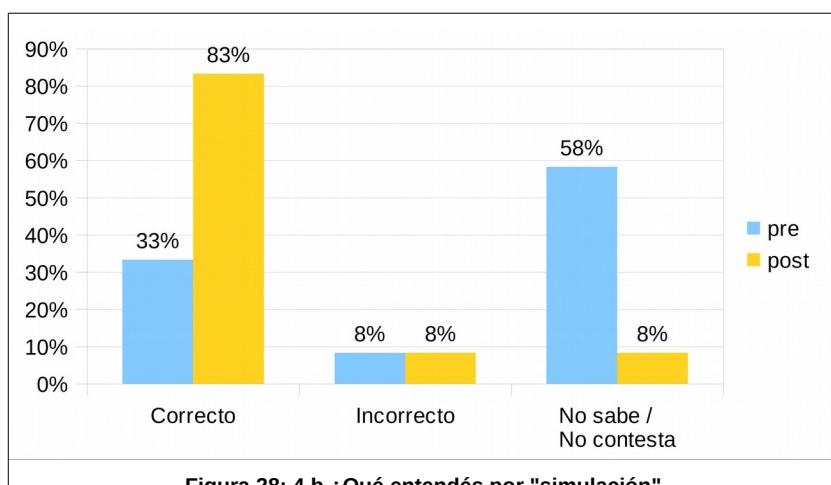


Figura 28: 4.b ¿Qué entendés por "simulación"

Tabla 25: 5. En tu opinión, pensás que el uso y/o construcción de modelos y/o simulaciones puede contribuir a tus aprendizajes escolares?

	Evaluación Pre-test	Evaluación Post-test
Si	54,17%	72,73%
No	0,00%	4,55%
No sé	45,83%	22,73%

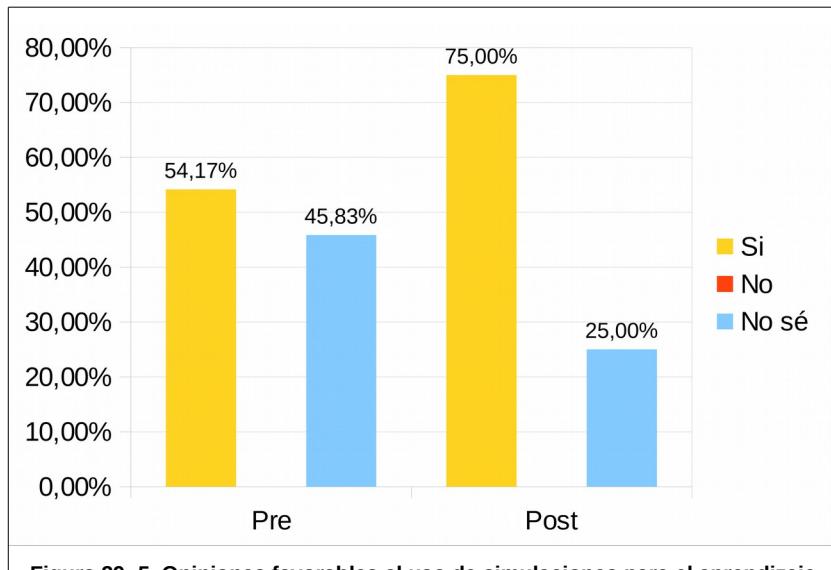


Figura 29: 5. Opiniones favorables al uso de simulaciones para el aprendizaje

13.3.2.3 Grupo experimental: evaluación de contenidos de Cinemática

Tabla 26: 1. Responde las siguientes preguntas: Si un tren recorre diez kilómetros cada diez minutos...

	Evaluación PRE-TEST			Evaluación POST-TEST		
	Correcto	Incorrecto	No sabe / no contesta	Correcto	Incorrecto	No sabe / no contesta
a.- ¿Cuántas kilómetros recorrerá en una hora?	100,00%	0,00%	0,00%	91,67%	8,33%	0,00%
b.- ¿Se puede asegurar que su velocidad es constante?	33,33%	45,83%	20,83%	41,67%	50,00%	8,33%
c.- ¿Se puede asegurar que su aceleración es nula?	37,50%	41,67%	20,83%	50,00%	50,00%	0,00%

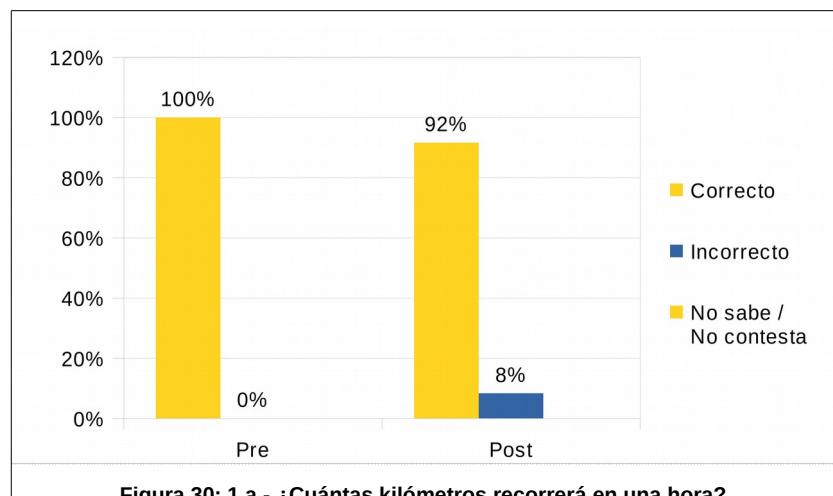


Figura 30: 1.a.- ¿Cuántas kilómetros recorrerá en una hora?

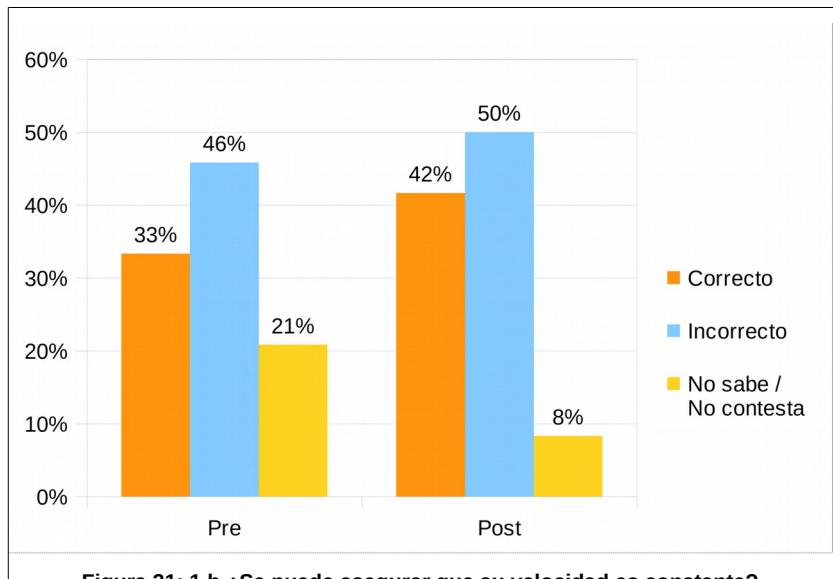


Figura 31: 1.b ¿Se puede asegurar que su velocidad es constante?

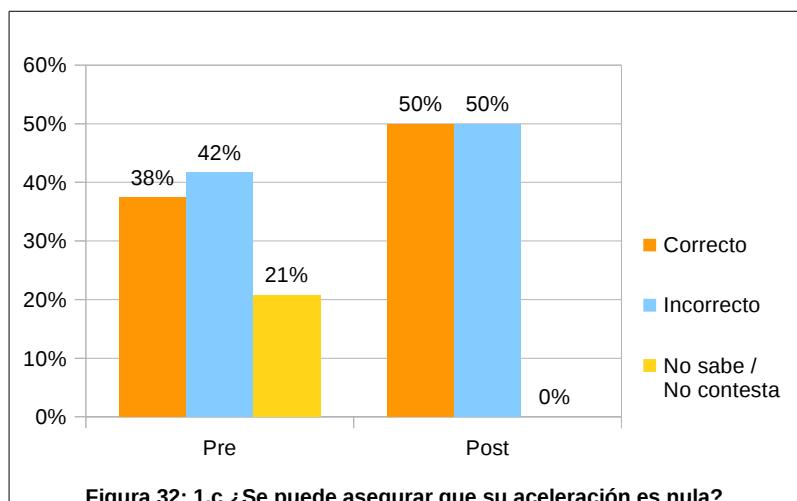


Figura 32: 1.c ¿Se puede asegurar que su aceleración es nula?

Tabla 27: 2. Responde la siguiente pregunta. ¿Es posible que sea verdadera la siguiente proposición: “Una partícula recorrió una distancia de 100m, su desplazamiento fue nulo”?

Evaluación PRE-TEST			Evaluación POST-TEST		
Correcto	Incorrecto	No sabe / no contesta	Correcto	Incorrecto	No sabe / no contesta
75,00%	25,00%	0,00%	83,33%	16,67%	0,00%

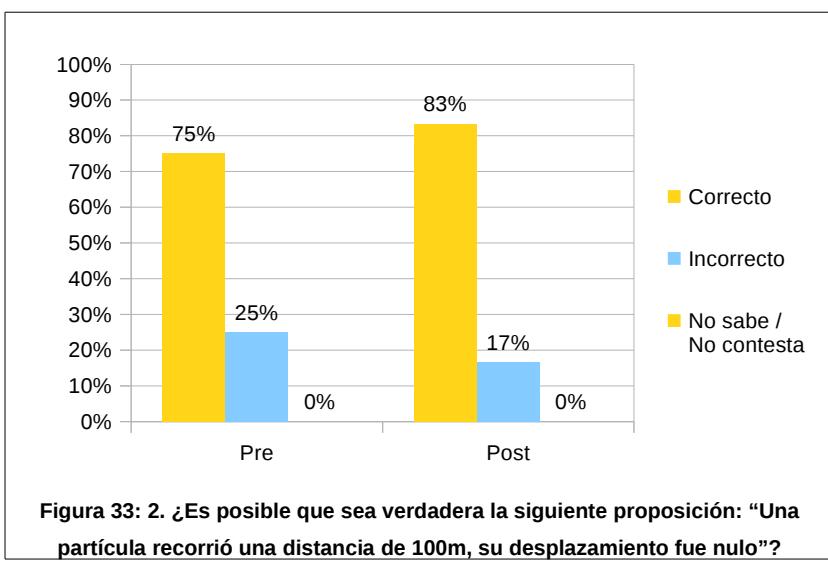
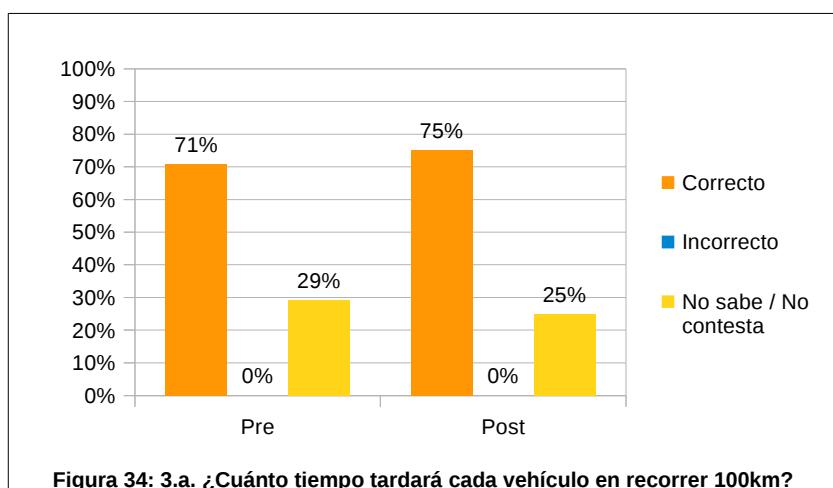
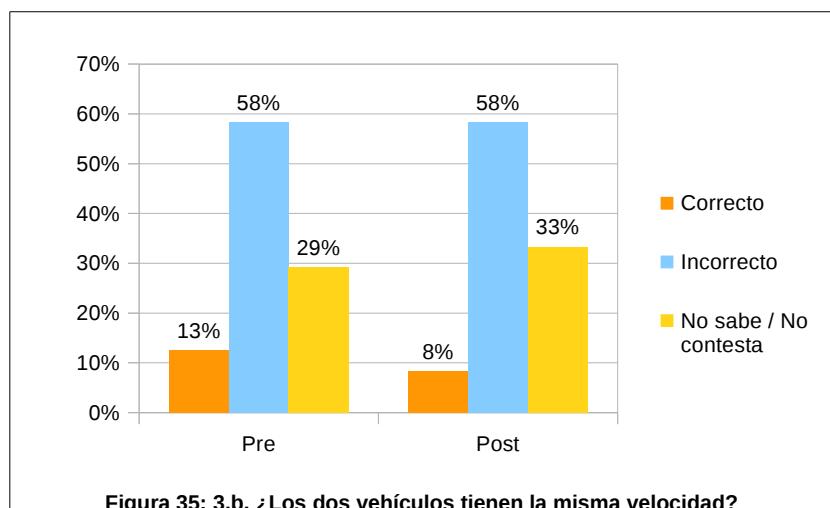


Tabla 28: 3. Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h.

	Evaluación PRE-TEST			Evaluación POST-TEST		
	Correcto	Incorrecto	No sabe / no contesta	Correcto	Incorrecto	No sabe / no contesta
a. ¿Cuánto tiempo tardará cada vehículo en recorrer 100km?	70,83%	0,00%	29,17%	75,00	0,00%	25,00%
b. ¿Los dos vehículos tienen la misma velocidad?	12,50%	58,33%	29,17%	8,33%	58,33%	33,33%



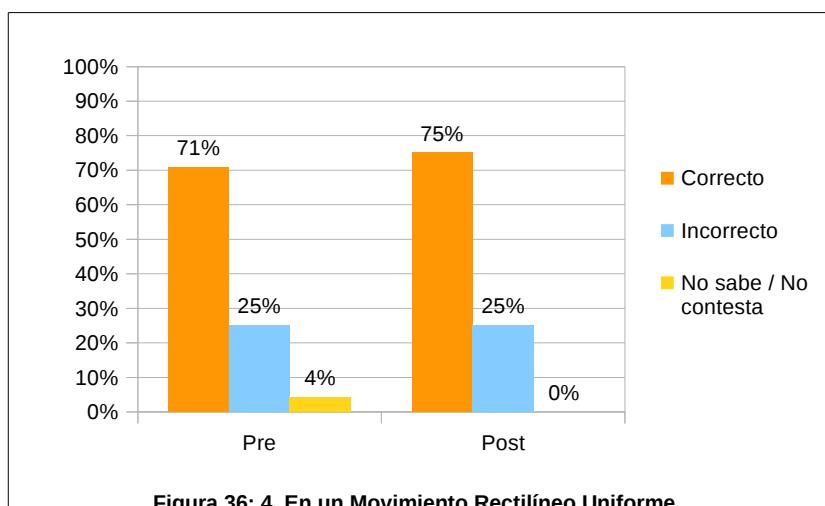


4. Seleccione la opción correcta: En un Movimiento Rectilíneo Uniforme...

- a.- la velocidad media y la velocidad instantánea son iguales
- b.- el módulo de la velocidad media es mayor que el módulo de la velocidad instantánea
- c.- el módulo de la velocidad instantánea es mayor que el módulo la velocidad media
- d.- ninguna de las opciones anteriores es correcta

Tabla 29: 4. Seleccione la opción correcta: En un Movimiento Rectilíneo Uniforme...

	Correcto	Incorrecto	No sabe / no contesta
Evaluación PRE-TEST	70,83%	25,00%	4,17%
Evaluación POST-TEST	75,00%	25,00%	0,00%



5. Seleccione la opción correcta. La distancia recorrida por una partícula...

- a.- siempre será menor que el módulo del vector desplazamiento
- b.- puede ser menor que el módulo del vector desplazamiento
- c.- siempre será mayor o igual que el módulo del vector desplazamiento
- d.- nunca será igual al módulo del vector desplazamiento

Tabla 30: 5. Seleccione la opción correcta. La distancia recorrida por una partícula...

	Correcto	Incorrecto	No sabe / no contesta
Evaluación PRE-TEST	50,00%	41,67%	8,33%
Evaluación POST-TEST	58,33%	41,67%	0,00%

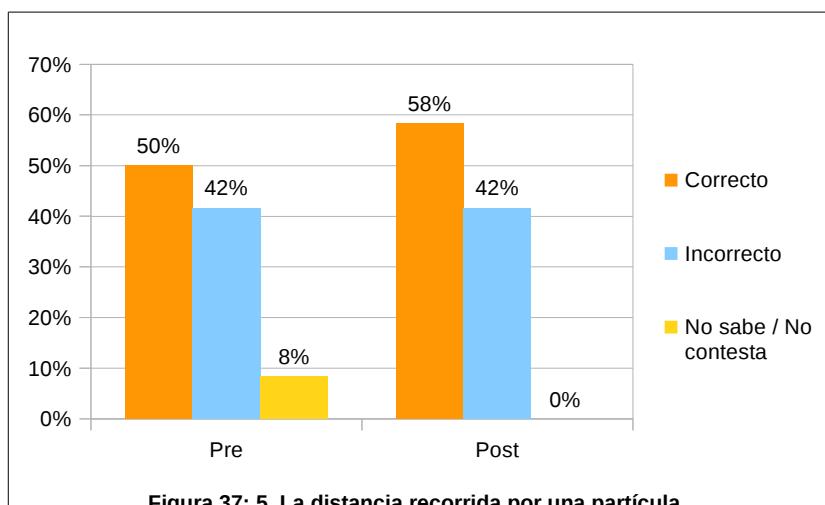


Figura 37: 5. La distancia recorrida por una partícula...

Tabla 31: Resultados generales por evaluación para el grupo experimental

	Respuestas correctas	Respuestas incorrectas	No sabe / No contesta
Evaluación PRE-TEST	56,25%	29,69%	14,06%
Evaluación POST-TEST	60,42%	31,25%	8,33%
Variación Pre-Test / Post-Test	4,17%	1,56%	-5,73%

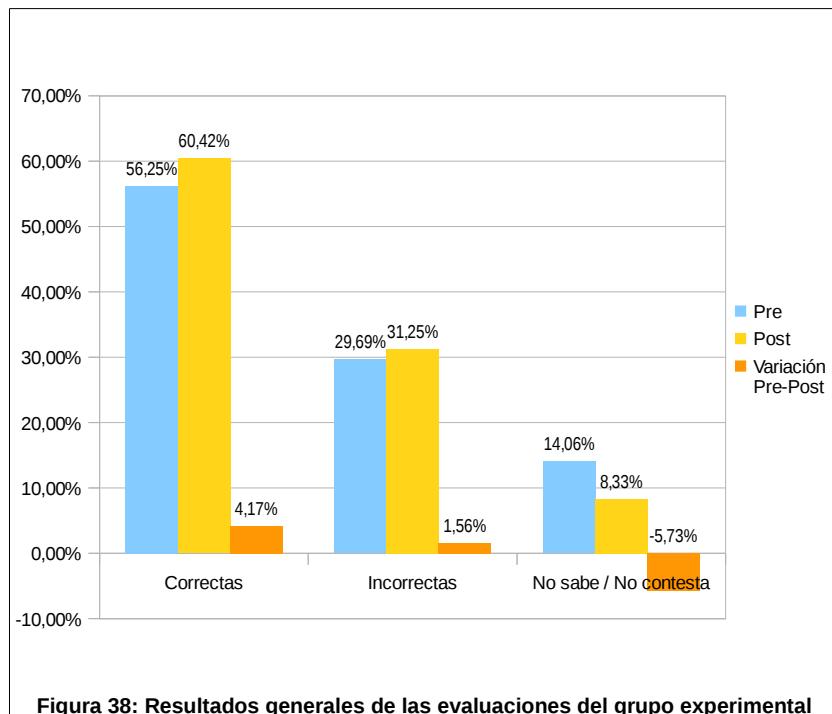


Figura 38: Resultados generales de las evaluaciones del grupo experimental

13.3.2.4 Testimonios escritos de alumnos del grupo experimental realizados con posterioridad a la experiencia

"La Tesis sobre la experiencia del profesor Richard fue dinámica y linda. Fue una llevadera experiencia de combinar informática con física. Para aprender sobre el tema en si fue muy bueno porque poníamos nuestros conocimientos de física y para resolverlo lo hacíamos con técnicas de informática . Fue algo que me ayudo mas a entender física y lo hicimos con la ayuda de la tecnología.

Esta técnica podría ser usada para otras materias y le ayudaría al alumno a comprender y saber emplear el tema".

Gabriela

"La tesis fue una experiencia la cual fue llevadera, ya que con los conocimientos adquiridos en el primer cuatrimestre en fisica el trabajo se hizo mas fácil. Expresamos lo que sabíamos justificando todas las respuestas. Me gusto la experiencia ya que me gusta fisica y la forma en que se pusieron los ejercicios para resolver".

Violeta

"En el trabajo sobre la tesis del profesor Richard el curso 4D TT participo en un trabajo de informatica en el que teniamos que tratar de resolver un problema de fisica, con diferentes formulas y diversas consignas que ayudaran a completar el ejercicio, luego la ultima clase lo pasamos en un proyector".

Guadalupe

"Durante la practica de Richard que realizo en nuestro curso, el nos mostro y nos ayudo a completar problemas en el ambito de la fisica gracias a un programa que desarrollo él, este programa funcionaba aplicando la velocidad y aceleracion entre otros parametros para que un automovil se moviera de distintas maneras.

Este programa nos ayuda a facilitar el cumplimiento y se podria usar en otras materias como matematicas o quimica pero recibiendo modificaciones.

Suerte Richard".

Abraham

14 Discusión

Los resultados del trabajo de campo indican que la construcción de una simulación informática mediante el lenguaje de programación Squeak-Etoys favoreció el aprendizaje de contenidos escolares relativos a Cinemática, en mayor medida que desarrollando las clases con recursos habituales.

La comparación de los resultados de las evaluaciones pre-test y post-test de Cinemática, del grupo control, muestran una disminución de las calificaciones, mientras que los resultados de las evaluaciones del grupo experimental en el post-test son superiores a las del pre-test.

Estos resultados son coincidentes con los mencionados por Taub y otros (2015), quienes hallaron que programar simulaciones de fenómenos físicos fue un factor importante en el aprendizaje de física, tanto en la representación visual de fenómenos físicos, como en el conocimiento estructural, el conocimiento procedural y el conocimiento sistémico.

Esa investigación, sin embargo, tuvo una duración de tres años durante los cuales tuvieron lugar clases semanales y se utilizó JAVA usando un entorno amistoso, mientras que el trabajo experimental de esta tesis consistió sólo en tres clases, y el lenguaje utilizado es totalmente mediado por una interface especialmente diseñada para no expertos donde las órdenes son palabras del lenguaje natural y están traducidas al español, un factor que favoreció el uso de la aplicación y un aprendizaje rápido de la misma.

En el citado trabajo, los alumnos participantes pertenecían a diferentes escuelas y fueron seleccionados por ser alumnos especialmente talentosos, mientras que en el caso de los alumnos participantes en el trabajo de campo de esta tesis, pertenecían en su totalidad a un curso completo, sin mediar ningún tipo de selección, y por tanto, el grupo fue heterogéneo en cuanto a desempeño académico, lo que puede haber afectado las calificaciones, a la vez, que puede considerarse que el recurso propuesto en la actividad es válido por los resultados obtenidos si se considera que otros trabajos parten de una muestra sesgada.

A diferencia de la experiencia de Queiruga (2013) que trabajó con alumnos de escuelas técnicas, y de Taub (un curso de ciencias de la computación), el trabajo experimental de esta tesis se llevó a cabo con grupos de alumnos que no tenían conocimientos de programación o algoritmia; además, la orientación de su carrera (comercial) es ajena al campo específico de la computación, la informática y otras disciplinas técnicas. Esta diferencia respecto de los saberes previos de los alumnos participantes en las experiencias mencionadas, supuso una desventaja para los alumnos del grupo experimental que sin embargo obtuvieron resultados favorables respecto de los contenidos de la asignatura, al igual que de la valoración del recurso didáctico utilizado.

En ambos grupos, control y experimental, se redujo la incertidumbre en la evaluación post-test respecto a las preguntas sobre los conceptos de *modelo* y *simulación*, aunque no se haya correspondido directamente en todos los casos con un aumento de las respuestas correctas; sin embargo esta reducción señala una construcción de significado por parte de los alumnos entre las evaluaciones pre-test y post-test.

En el grupo experimental, las respuestas "no sabe / no contesta" respecto del concepto de "modelo" registraron una disminución de casi 38%, aumentando sobre todo las respuestas incorrectas en el post-test, mientras las respuestas "no sabe / no contesta" sobre "simulación" disminuyeron del 50% al 8% a la vez que aumentaron las correctas también en un 50%. Cabe señalar que el término más utilizado en la actividad experimental, tanto en el tutorial como en la comunicación oral fue "simulación".

Papert (1999) asimila la figura del aprendiz al profesor o maestro que orienta una actividad en clave construccionista, ya que se encontrará con situaciones nuevas e imprevistas aún para el mismo maestro, situaciones en las que deberá realizar aprendizajes y el alumno, participando de la actividad, a su vez aprenderá de cómo lo hace su maestro. Durante las clases experimentales tuvieron lugar situaciones imprevistas, probablemente debido a que el tutorial interactivo permaneció abierto a modificaciones (de igual modo que el propio entorno) por ser su propiedad primaria. Sin embargo esas mismas situaciones emergentes hicieron que la experiencia de cada grupo fuera única al reflejar consecuencias de su propio trabajo y permitir

soluciones al mismo tiempo propias y desafiantes tanto para los alumnos como para el profesor.

También reconoce que su trabajo será más arduo pero mucho más creativo e interesante. Los imprevistos, frecuentes, hicieron el trabajo del profesor, más arduo al responder constantemente a requerimientos de ayuda de los alumnos, y también creativo en tanto cada situación era distinta en cada caso.

El trabajo previo a la actividad con los alumnos, en especial el desarrollo del tutorial interactivo, demandó muchas horas de trabajo en que cada diapositiva era una pieza original de un rompecabezas cuya solución nunca había sido hallada, que sólo poniéndola a prueba iba a revelarse totalmente, y la experiencia satisfizo las expectativas.

El tiempo de trabajo requerido para desarrollar el tutorial, incluyó tareas que se construyeron por primera vez:

- elección de un tema cuyo aprendizaje se enriquezca con la forma de trabajo que se usó en el trabajo de campo
- recrear (el profesor) la simulación y simplificarla desde el punto de vista del código de acuerdo a las expectativas sobre los alumnos
- deconstruir esa simulación para crear la secuencia de diapositivas o páginas del tutorial y su contenido específico; esto implica redactar el guión del tutorial, los objetos gráficos, y los componentes de código
- crear un diseño gráfico común a todas las diapositivas para facilitar la navegación

Sin embargo, el material producto de estas tareas puede ser reutilizado como punto de partida para nuevos tutoriales, utilizando como una plantilla los criterios de diseño señalados en el último punto, y también recuperando no ya el contenido sino la metodología seguida para la realización de las primeras tres tareas. Astudillo (2016,1) pone de manifiesto la importancia de la reutilización como característica propia de los Objetos de Aprendizaje y deseable en los Materiales Educativos Digitales en general. Por otra parte, la idea de reutilizar, reusar o reciclar material es central en la programación y fue ponderada tanto por Papert como por Resnick, a la vez que es

una de las características del pensamiento computacional (en otra escala, ningún sistema operativo se re-escriba totalmente en una nueva versión o distribución, sino que se modifica la anterior).

La presentación del tutorial interactivo a los alumnos incluyó la recomendación de que asumieran la actividad como un juego. La importancia del carácter lúdico de una actividad didáctica es resaltada por Queiruga (2014), en tanto que pone en juego la interacción con los pares, la diversión, la tolerancia a la frustración y la motivación hacia el logro. También, en una revisión de 27 trabajos sobre enseñanza y aprendizaje de pensamiento computacional por medio de la programación, Lye y Koh (2014:59), citando a Jonassen, Kaffai y Resnick, hallaron mejores resultados cuando los problemas que se proponen a los alumnos son de su interés, lo que hace que se comprometan más con la actividad, y sugieren trabajar en torno a problemas tales como diseño de juegos, estrategias de juego y creación de historias digitales. No obstante las diferencias entre el objetivo de esa revisión y el de esta tesis, las sugerencias de Lye y Koh hacen necesario, en la inclusión de construcción de simulaciones, la reformulación de las actividades convencionales a fin de lograr significatividad.

Uno de los rasgos del pensamiento computacional (Wing, 2010) es la posibilidad de generalizar y transferir procedimientos a campos diferentes de Computación y Sistemas, lo que en esta experiencia se ha realizado con éxito por medio de la programación de una simulación, en coincidencia con los hallazgos de resultados positivos en la mayoría de las 27 experiencias analizadas por Lye y Koh (2014:59).

El conocimiento de la interfaz de usuario insumió inicialmente poco tiempo de aprendizaje exclusivo; en pocos minutos, los grupos de trabajo estaban abocados a resolver los primeros desafíos. En el trabajo de Queiruga (2013) se señala un cambio de lenguaje en el proyecto "Enseñar a Programar en la Escuela Secundaria", de JAVA por RITA (un entorno más amigable al usuario, al estilo de Scratch y Squeak-Etoys), por considerar más adecuada la interfaz de usuario para el rango de edad de los alumnos. Esto, en consonancia con el trabajo citado, refuerza la viabilidad de este tipo de interfaces en la introducción de programación como recurso de aprendizaje en contextos legos, como el educativo en general, incluso en la iniciación en carreras específicas de sistemas o computación.

Durante la actividad se observó motivación respecto de la misma y en diálogo con los alumnos se manifestaron interesados respecto de esta forma de trabajar, a la vez que luego del trabajo de campo aumentó un 34% la cantidad de alumnos que señalaron que usar y construir simulaciones podría contribuir a sus aprendizajes. Estos resultados coinciden con los obtenidos por Yohan, Kongju y Yunebae (2016), que mencionan mejoras en la percepción del pensamiento computacional y el compromiso e interés en las actividades por parte de los alumnos.

Además, el grupo de alumnos que participó del trabajo experimental realizó consultas sobre la posibilidad de crear simulaciones relativas a otros contenidos y otros espacios curriculares, lo cual indica motivación e interés en la metodología.

Estas inquietudes manifestadas por los alumnos, al valorar actividades de aprendizaje que ponen el énfasis en el protagonismo del que aprende, son alentadoras y recuerdan a Papert (1999) cuando señala que la educación estaba demasiado volcada sobre su faceta informativa y las tecnologías digitales podrían corregir el desequilibrio y fortalecer la faceta constructiva, a la vez que destacó la capacidad de los computadores para crear simulaciones como recurso educativo.

Además de intentar resolver los desafíos propuestos, algunos alumnos expresaron curiosidad acerca de cómo modificar diversos aspectos de la simulación, más allá de los propuestos, tales como los colores, formas y tamaño de los objetos (los autos), y si sería posible simular otros tipos de movimiento. Las preguntas acerca de realizar estas modificaciones parecen denotar interés en poner su marca personal en la tarea, en expresarse, más allá de lo que las consignas solicitan, en definitiva, apropiación de la actividad. En coincidencia, la expresión fue uno de los primeros hallazgos en la investigación de Brennan y Resnick (2012).

15 Conclusiones

La construcción de una simulación mediante Squeak-Etoys favoreció el aprendizaje de contenidos de Cinemática, evidenciado en el aumento en las calificaciones de las evaluaciones post-test.

Hay coincidencia con otras investigaciones similares con eje en contenidos de Física (Taub y otros:2015).

Los resultados de la actividad experimental fueron satisfactorios aún tratándose de un grupo heterogéneo, a diferencia del caso anterior en que los estudiantes fueron seleccionados.

Hay divergencia respecto de la duración y cantidad de clases de las investigaciones, así como del lenguaje de programación utilizado.

En general se ha recurrido a entornos de trabajo amistosos para el usuario con buenos resultados.

Programar una simulación se mostró como un recurso didáctico válido aún para alumnos sin conocimiento de programación ni algoritmia.

Tras realizar la actividad experimental, los alumnos que comprendían el concepto de simulación aumentó casi tres veces.

La actividad presentó numerosos imprevistos y desafíos para el docente, lo que significó mayor intensidad en la tarea a la vez que promovió una flexibilidad en la asimetría entre los roles docente y alumno, al encarar esos desafíos junto a estos últimos.

El trabajo en torno al tutorial interactivo deja una serie de recursos reutilizables en nuevas actividades, y retroalimentación para mejorarlas, reduciendo el tiempo de desarrollo.

Proponer la experiencia con características lúdicas, desafíos, contenidos significativos para los alumnos y mediatizada por las nuevas tecnología, favoreció el aprendizaje y el interés de los alumnos en los medios utilizados.

En los primeros pasos en programación dan mejores resultados interfaces gráficas amigables similares a RITA, Scratch y Squeak-Etoys, utilizada en el trabajo de campo.

Construir la simulación del ejercicio de Cinemática generó motivación en los alumnos, así como interés y expectativas positivas respecto de usar la misma forma de trabajo en otros aprendizajes.

Los alumnos, además de valorar el tipo de actividad que realizaron, demostraron curiosidad por conocer modos de expresarse y poner su sello personal en la actividad usando las herramientas del entorno de programación utilizado.

16 Anexos

16.1 Anexo 1: Encuesta sobre características generales del grupo

4º Año – Física

Prof. Diego Zanarini / Prof. Ricardo Salvador

Encuesta

Las siguientes preguntas tienen por objetivo conocer características del grupo-curso; responde en forma anónima:

Curso: Edad: Sexo (F/M): Fecha:

1. En la Escuela, has usado software para...

(marca con una “x”)

representar situaciones que habitualmente se realizan con lápiz, papel, regla, compás, etc.?

resolver problemas?

usar simulaciones?

construir modelos o simulaciones?

desarrollar aplicaciones?

2. Usás tecnología informática para resolver problemas cotidianamente?

si no

3. Ante un problema preferís resolverlo individualmente o junto a otros?

individualmente con otros

4. Qué entendés por... (podés usar el reverso de la hoja para contestar)

modelo? simulación?

5. En tu opinión, pensás que el uso y/o construcción de modelos y/o simulaciones puede contribuir a tus aprendizajes escolares?

si

no

no sé

¡Muchas gracias!

16.2 Anexo 2: Encuesta sobre conocimientos de Cinemática

4º Año – Física

Prof. Diego Zanarini / Prof. Ricardo Salvador

Trabajo práctico: Cinemática

1. Responde las siguientes preguntas

Si un tren recorre diez kilómetros cada diez minutos...

- a.- ¿Cuántas kilómetros recorrerá en una hora?
- b.- ¿Se puede asegurar que su velocidad es constante?
- c.- ¿Se puede asegurar que su aceleración es nula?

Justifique todas sus respuestas

2. Responde las siguientes preguntas

¿Es posible que sea verdadera la siguiente proposición: "Una partícula recorrió una distancia de 100m, su desplazamiento fue nulo"?

Justifique

3. Responde las siguientes preguntas

Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h.

- a.- ¿Cuánto tiempo tardará cada vehículo en recorrer 100km?
- b.- ¿Los dos vehículos tienen la misma velocidad?

Justifique todas sus respuestas

4. Seleccione la opción correcta

En un Movimiento Rectilíneo Uniforme...

- a.- la velocidad media y la velocidad instantánea son iguales
- b.- el módulo de la velocidad media es mayor que el módulo de la velocidad instantánea
- c.- el módulo de la velocidad instantánea es mayor que el módulo la velocidad media
- d.- ninguna de las opciones anteriores es correcta

5. Seleccione la opción correcta

La distancia recorrida por una partícula...

- a.- siempre será menor que el módulo del vector desplazamiento
- b.- puede ser menor que el módulo del vector desplazamiento
- c.- siempre será mayor o igual que el módulo del vector desplazamiento
- d.- nunca será igual al módulo del vector desplazamiento

16.3 Anexo 3: Encuesta posterior al trabajo experimental

4º Año – Física

Prof. Diego Zanarini / Prof. Ricardo Salvador

Encuesta

Las siguientes preguntas tienen por objetivo conocer características del grupo-curso; responde en forma anónima:

Curso: Edad: Sexo (F/M):

1. Qué entendés por...

... modelo?

... simulación?

2. En tu opinión, pensás que el uso y/o construcción de modelos y/o simulaciones puede contribuir a tus aprendizajes escolares?

[] si

[] no

[] no sé

¡Muchas gracias!

16.4 Anexo 4: Guía para la Simulación del Ejercicio 14 de Cinemática

Guía para la Simulación del Ejercicio 14 de Cinemática

Sumario

1. Introducción	1
2. Analizando el ejercicio	1
3. Pensando la simulación	2
4. Nuestro auto en la "ruta"	2
5. Nuestro auto andando en la "ruta"	2
6. Hagamos cálculos...	2
7. El tiempo...	3
8. Ciclos, ciclos...	3
9. Y ahora, un poco de movimiento...	3
10. Un poco de inteligencia en nuestro auto	3
11. Otro desafío	4
12. Combinando recursos	4
13. Cambiando el horario	4
14. Hagamos unos paréntesis	4
15. Ya funciona nuestro auto Azul!	5
16. Automatizando la simulación	5
17. Vamos a hacer un "clon"	5
18. Probando el "clon"	6
19. El todo es más que la suma de las partes	6
20. Obteniendo respuestas...	6
21. Hemos llegado al final, o no tanto	6

1. Introducción

¡Hola!

En **esta actividad** intentaremos construir **una simulación del ejercicio 14 de Cinemática**, programándola en Etoys que es una especie de mundo virtual. Es decir que intentaremos mostrar dos autos que a la “hora” y velocidad correspondientes, pasen por un punto hasta que uno alcance al otro y así averiguar a qué hora y a qué distancia del punto de referencia.

Para construir la simulación...

- analizaremos lo que ocurre en el ejercicio y cómo puede simularse en Etoys
- veremos cómo funciona este ambiente de trabajo
- simularemos aspectos parciales a fin de recrear finalmente la simulación completa

En cuadros de texto como éste mismo están las indicaciones para que cada grupo pueda realizar esta actividad a su propio ritmo, a la vez que según lo necesite, cuente, por supuesto, con la asistencia del profesor.

Pasemos a analizar el ejercicio en la siguiente diapositiva...

2. Analizando el ejercicio

Si recordamos la consigna...

Un auto rojo pasa por la localidad de Arequito a las 8:00 a una velocidad de 40 km/h. A las 10:00 pasa otro auto (verde) en su persecución, a 60 km/h. Calcular a qué hora y a qué distancia de la localidad "A" el segundo auto alcanza al primero (he puesto color a los autos y nombre a la ciudad por una cuestión práctica).

Tenemos

- un **lugar** (la localidad de Arequito)
- un **auto rojo** que pasa allí a las 8:00 a 40 km/h.
- un **auto verde** que pasa por allí a las 10:00 en igual dirección, a 60 km/h.

Y el **desafío** es calcular

- **a qué hora** y
- **a qué distancia** de la localidad "A"

...el segundo auto alcanza al primero.

3. Pensando la simulación

El desafío que plantea esta actividad es mostrar, en la pantalla, cada auto pasando por la ciudad de Arequito en el horario indicado, y moviéndose hasta que el verde alcanza al azul, mostrando la hora y los kilómetros recorridos. Para lo cual, podemos anticipar que vamos a...

- utilizar el reloj interno de Etoys, y de acuerdo a la hora que provee
- calcular la posición de cada auto de acuerdo a la hora (por practicidad, 1 segundo equivaldrá a 1 hora),
- dibujar cada auto de acuerdo a ese cálculo, con referencia a una escala cuyo "0" es la ciudad de Arequito.
- comparar si el auto verde alcanzó al azul, caso en el que hallaremos las respuestas que exige el ejercicio 14.

... integrando todas estas operaciones en una estructura que las ejecute automáticamente.

En las siguientes diapositivas experimentaremos con cada aspecto parcial en Etoys para utilizarlo en la simulación. ¡Suerte!

4. Nuestro auto en la "ruta"

Qué tenemos aquí:

- nuestro auto azul
- una escala horizontal en la que 3 pixeles equivalen a 1 kilómetro,

- una etiqueta del atributo "x on graph" (x en el gráfico) del auto cuyo valor indica respecto de qué "kilómetro" de la escala está situado.

Veamos cómo funciona esto:

- Arrastra el auto con el botón derecho del touchpad o del mouse horizontalmente, y observa... el valor de "x on graph".
- Cambia el valor de "x on graph" y observa el auto...

Este es un primer paso; veamos a continuación algo más...

5. Nuestro auto andando en la "ruta"

El rectángulo que vemos a la derecha es un **guión** de programa que tiene una instrucción que ya se imaginan qué hará: situar al auto en el km 125!

Un guión es una serie de instrucciones que genera un comportamiento en un objeto, en este caso, nuestro auto azul.

Ejecuta este guión, haciendo clic-izquierdo en el círculo amarillo que tiene un "!". Observa al auto, observa el valor de "x on graph"...

Cambia el valor de "x on graph" dentro del guión.

... y **prueba** nuevamente el guión.

Pueden empezar a imaginar hacia dónde vamos con esto...

A continuación, vamos a ver una variante.

6. Hagamos cálculos...

Observen que **en este guión hay una modificación**: se establece el valor de "x on graph" mediante un producto. Prueben cambiar los valores y el operador (el "*") por otros.

Sabemos que la distancia que recorre el auto, en M.R.U. equivale al producto del tiempo y la velocidad.

De manera que **si en el guión reemplazáramos los factores** por velocidad y tiempo, obtendríamos una distancia.

Haz... clic-derecho en el auto y luego clic-izquierdo en el círculo celeste: el panel que se despliega contiene casi todas las características de nuestro auto y las instrucciones que puede ejecutar. **Observen** las dos primeras características o "variables" (que he creado a propósito para esta actividad): comienzan con el nombre de nuestro auto "autoazul", y una se llama "t" (por tiempo) y la otra, "v" por velocidad.

Arrastren cada una de ellas desde el panel y suélténlas sobre cada uno de los factores del guión (verán que al pasar sobre el un lugar es posible que ese lugar se recuadre en color verde, y eso significa que puede soltarse el objeto en ese lugar). Arrástrelas fuera del guión también: funcionan como "x on graph", para tenerlas a mano...

Y cambien sus valores y ejecuten el guión...

A estas alturas podemos calcular y ver a qué distancia llega nuestro auto!

Supongamos que podemos cambiar rápidamente el valor de "t" y ejecutar nuestro guión... ¿Qué pasaría?

7. El tiempo...

Tenemos ahora nuestro guión "situar" que ubica al auto en función del producto de las dos variables: "t" y "v". Las tenemos a mano, para poder usarlas más cómodamente...

¡Ah! A propósito de la última pregunta...

Hagan clic-derecho fuera de esta diapositiva, sobre el fondo gris. Se seleccionará el "**mundo**" (miren abajo, en el centro). Usen el círculo o **halo celeste** y en el panel o **visor**, busquen el atributo "**temporizador**" (usando el botón "**buscar**" es lo más práctico).

Observen que a diferencia de otros atributos... no está quieto. Es una especie de segundero. **Cambien** su valor. Y **cambien** también los decimales que muestra (usando el ícono que tiene delante).

Arrastren "**temporizador**" cerca de "x on graph".

¿Qué tal si pudiéramos usarlo para nuestro guión? vamos a la siguiente diapo...

8. Ciclos, ciclos...

Hasta ahora hemos usado el **guión** ejecutándolo sólo una vez, pero ya lo habrán imaginado, como una calesita, también **es posible que se ejecute cíclicamente**, y que es lo mismo que decir **repitiendo** las instrucciones que contiene. Igual que las ruedas de nuestro auto que se mueven cíclicamente.

Hagamos funcionar eso: mientras que con el círculo amarillo con el "!", ejecutamos sólo una vez el guión, para hacerlo continuamente, usaremos... el **reloj**, para **comenzar** y también para **detener** la ejecución continua del guión.

¡Probemos ese reloj! Y mientras tanto, cambiemos los valores de "v" o "t": ¿qué pasa?

¿Y si detienen el guión y siguen cambiando "v" y "t"? Mmm...

Esta ejecución continua es interesante porque si se pudiera utilizar el valor del "**temporizador**" durante la ejecución...

9. Y ahora, un poco de movimiento...

... ejecutar el **guión continuamente es interesante** porque si incluimos el valor del "**temporizador**" que también cambia continuamente... ¡Hagamos una prueba y veamos qué pasa!

- usando el **halo verde**, hagan una copia del temporizador que tienen en la pantalla.
- establezcan la velocidad (AutoAzul's v) entre 5 y 10 km/h.
- Arrastren el **temporizador** sobre la variable "v" (AutoAzul's v) del guión, a fin de reemplazarla...
- Pongamos el **temporizador en 0**, y con un clic-izquierdo en el **reloj**, inicien la ejecución del guión "situar". ¿Qué pasa?

Este es un momento que pueden jugar un poco... con la velocidad y los valores del temporizador

Varíen la velocidad mientras se ejecuta "situar", vean el resultado...

Ojo que puede "irse" el auto... en ese caso, pongan el temporizador a cero.

Hecho esto, vamos por un algo más... en la siguiente diapositiva.

10. Un poco de inteligencia en nuestro auto

... no vendrá mal para nuestra simulación

Nuestro desafío dice que el auto azul pasa por la ciudad de Arequito cuando son las 8:00...

Esto podría traducirse, **en términos de nuestra simulación**, en poner el temporizador en 0 y cuando llegue a 8, situar el auto en el km 0 (las 8:00 son el t0 del auto azul) y seguir mostrándolo en la "x on the graph" que corresponda al temporizador y a la velocidad.

Esto hace necesario un interesante recurso que nos permite probar si se cumplen ciertos tipos de condiciones y en función de la respuesta, realizar algo.

Veamos un ejemplo. Observen el guión "vertical": se ejecuta continuamente y hay dos estructuras similares muy fáciles de comprender y comparar el valor del temporizador con un número, y de acuerdo a si la condición se cumple o no, ejecutan una acción, en este caso, establecer la coordenada y del auto en ciertos valores.

Y un desafío, cambien los valores intentando que el auto "suba" a 480 cuando el temporizador esté entre 100 y 105 y después vuelva a su lugar original.

Suerte... y nos vemos en la próxima diapo, con otro desafío...

11. Otro desafío

Y el desafío es...

Imaginemos que el auto, entre las 7 y las 9, pasa por un banco de niebla, por lo que entre ese "horario" no debería verse.

Usando los comandos que hay en el recuadro verde, "ocultémoslo" mientras pase por la niebla, modificando el guión anterior "vertical":

- cambien el nombre del guión, de "vertical" a "niebla"
- y deberán cambiar algo más, ¡ése es el desafío!

12. Combinando recursos

Recordemos...

- por un lado, vimos cómo simular el movimiento del auto sobre la escala cuyo "cero" es Arequito
- por otra parte, vimos cómo utilizar el valor del temporizador para ejecutar ciertos comandos (cambiar el valor de "y", ocultar/mostrar el auto Azul).

Entonces, siguiendo con los desafíos...

¿Se animan a mostrar el recorrido del auto Azul entre las 0 y las 5, suponiendo que a las 0 pasó por Arequito?

En el cuadro verde (que después pueden mover al "cubo de basura") les dejo algunas cosas que tal vez van a servirles para este cometido.

Un dato: al temporizador pueden "setearlo" en, por ejemplo, "-5"

13. Cambiando el horario

En el ejercicio 14, los autos pasan a distinto horario por la ciudad "A", para nosotros, Arequito.

En analogía con la fórmula que usaron para resolverlo, la nuestra queda así:

$$x = (\text{hora} - 8) * \text{velocidad}$$

que traducido a nuestro guión, quedaría así:

$$\text{Azul's_x-on-graph} = (\text{mundo's_temporizador} - 8) * \text{Azul's_v}$$

Necesitamos los paréntesis; ¿cómo podemos usarlos? Vamos a la próxima diapositiva.

14. Hagamos unos paréntesis

Entonces, ahora debemos usar paréntesis para que la "t0" del auto sea igual a 8, lo que en términos del lenguaje que estamos usando se vería así:

$$\text{"Azul's_x-on-graph} = (\text{mundo's_temporizador} - 8) * \text{Azul's_v"}$$

Para eso, vamos a modificar el guión reemplazando la variable **mundo's_temporizador** por **(mundo's_temporizador - 8)**:

1. con un clic-derecho en la "caja" amarilla, arriba a la derecha del guión "situar", se despliega una lista
 2. elegimos **aleatorio(5)** y lo soltamos sobre "mundo's_temporizador"
 3. con un clic en la palabra "aleatorio" se despliega una **lista** de la que elegiremos "**paréntesis**".
 4. ahora tomamos una **copia del temporizador** y la soltamos sobre el "5" que está entre los paréntesis.
 5. nos situamos sobre la palabra "temporizador" y hacemos clic sobre el **triángulo verde** que aparece a su derecha.
 6. aparecerá un "+", y un clic sobre el mismo despliega una **lista de operadores** (recuerden que estamos necesitando un "-"); elegimos entonces el "-".
 7. sólo resta **cambiar el "1" por el "8"**, la hora a la que pasa Azul por Arequito...
 8. ¡A probar nuestro guión!
- ... Y a pasar a la próxima diapositiva...

15. Ya funciona nuestro auto Azul!

El auto Azul se comporta de acuerdo a la consigna del ejercicio 14: cuando son las 8, "pasa" por Arequito.

Nada más ajustemos la velocidad a 40 (Azul's_v, que ahora está en 12), en el "observador" que hemos dejado debajo de los observadores del "temporizador del mundo" y de "x_on_graph", la posición del auto en la escala graduada en km.

Así planteado el guión que da "comportamiento" al auto Azul, podemos dar otro paso más, automatizar la simulación, cosa que veremos a continuación.

16. Automatizando la simulación

El auto Azul se comporta de acuerdo a la consigna del ejercicio 14: cuando son las 8, pasa por Arequito.

Algo que podemos mejorar es **automatizar** la puesta a cero del reloj y la ejecución del guión "situar", para que todo ocurra con sólo un clic:

(!) Para tener más "lugar" podemos minimizar o colapsar el guión "situar" usando el círculo marrón que se halla arriba a la izquierda.

1. **Poner el reloj en "0".** Busquemos en el visor del mundo la variable "temporizador" y tomándola desde la flecha blanca, suéltenla sobre el fondo de la diapositiva: va a crearse un nuevo guión ("mundo_guion1"), al que podemos bautizar "iniciar". Ahí establecemos el valor del temporizador en "0".
2. **Iniciar el guión "situar".** Abran el visor de Azul (clic_derecho, halo celeste), y con un clic en "basico" u otra categoría, busquen "manejo de guiones". Ahí la primera instrucción es "Azul_comenzar guion_guion_vacio" (es la que necesitamos...). Y como la idea es lanzar el guión "situar", reemplacen "guion vacío" por "situar".
3. También podríamos establecer la velocidad (Azul's_v) dentro del guión, del mismo modo que hicimos con el temporizador.
4. Ahora es más cómodo probar a nuestro auto, usando el círculo amarillo con un "!", ¡probemos!

Nota: el botón "Pánico" nos simplificará la tarea de recuperar al auto en caso de que se "escape".

17. Vamos a hacer un "clon"

El auto Azul se comporta como esperábamos: a las 8, pasa por Arequito a 40 km/h.

El ejercicio 14 se completa con otro auto, verde, que pasa a las 10 por el mismo lugar, en la misma dirección, a 60 km/h.

Las diferencias con el primer auto (Azul) son dos: la hora a la que pasa y la velocidad (ah, y el color).

Podrían estar imaginándolo: si **duplicamos** el auto Azul de nuestra simulación, y cambiamos sus propiedades, podemos simular el comportamiento del segundo auto (el verde) de nuestro ejercicio. ¡Además podríamos simular los dos autos al mismo tiempo!

¡Adelante con eso! **Obtengamos un "clon" de Azul** (clic-derecho, halo verde) y **editemos sus características para que simule al auto Verde** (cambien su nombre por "Verde", su color, y si hay algo más que crean necesario cambiar...)

Y probemos cómo funciona.

18. Probando el "clon"

Al duplicar nuestro auto "Azul" y obtener el auto "Verde", éste "heredó" las características del auto Azul.

Esto es importante porque nos permitió utilizar y modificar sus características, no sólo su imagen, además también de heredar el guión "situar".

Prueben cómo se comporta "Verde", buscando el guión "situar" en su "visor" (clic-derecho en el auto Verde, y clic en el halo celeste, luego buscan la categoría "guiones").

¿Cómo funciona? ¿Hay que hacer alguna modificación? (hay un botón "Pánico" para el auto verde, también)

A continuación, más emoción al volante!

19. El todo es más que la suma de las partes

Todo esto tenemos hasta aquí...

1. creamos el comportamiento del auto Azul
2. lo automatizamos (con el guión "mundo_Iniciar")
3. logramos hacer funcionar también al auto Verde

Y no podía faltar el **desafío**:

Conseguir que los autos "Azul" y "Verde" simulen la situación descripta en el ejercicio 14.

es decir, que cada uno pase por Arequito, a la hora y a la velocidad estipuladas.

Si logran superar este desafío, les queda el último peldaño para llegar a la cima...

20. Obteniendo respuestas...

Habiendo simulado el ejercicio 14, **sólo nos falta responder a las preguntas del enunciado:**

"a qué hora y a qué distancia de A(requito) el segundo (Verde) alcanza al primero (Azul)?"

Si recuerdan lo visto **en las diapositivas anteriores, encontrarán recursos** para encontrar esas respuestas.

Algunas pistas:

¿la simulación nos provee de parámetros que indiquen...

... que Verde alcanzó a Azul?

... a qué distancia lo alcanza?

... a qué hora?

¡A resolverlo!

21. Hemos llegado al final, o no tanto

Espero que hayan llegado a esta instancia habiendo disfrutado la experiencia de trabajar juntos en la simulación propuesta.

Quizás les sirva para imaginar otras utilidades de este modo de representar situaciones.

¡Muchas gracias!

16.5 Anexo 5: Póster JEMU 2016

Póster presentado en las Jornadas de Escuelas Medias Universitarias (JEMU) realizados los días 5 y 6 de octubre de 2016 en Horco Molle, Tucumán.

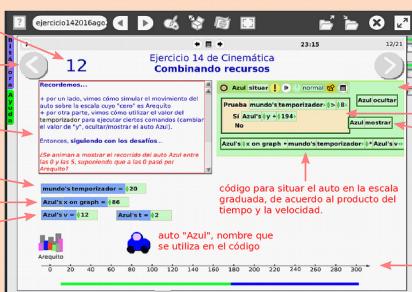


Construcción de una simulación informática sobre un ejercicio de Cinemática con Squeak-Etoys como recurso de aprendizaje

Lic. Ricardo Pablo Salvador, Prof. Diego Zanarini (Esc. Superior de Comercio "Lib. Gral. San Martín" - U.N.R.), Dra. Claudia Pons (Lab. de Investigación y Formación en Informática Avanzada - U.N.L.P.), Dr. Guillermo Rodríguez (Fac. de Ingeniería y Cs. Exactas - U.N.R.)

FUNDAMENTACIÓN	OBJETIVO	METODOLOGÍA
<p>1. La simulación como factor de aprendizaje activo, * metacognición, creatividad, apropiación del contenido escolar</p> <p>2. La Programación como recurso educativo y competencia general</p> <p>3. Entornos de programación gráficos que facilitan la generación de código.</p>	<p>Evaluar el efecto de desarrollar una simulación informática para la resolución de un problema en el aprendizaje de Cinemática.</p>	<p>* dos grupos de alumnos (experimental y control) de 4º de escuela media.</p> <p>* protocolo pre-test / post-test</p> <p>* trabajo en grupos de 2-3 alumnos</p> <p>* construcción de la simulación de un problema de Cinemática en el entorno de programación Squeak-Etoys</p>

Tutorial interactivo



The screenshot shows the Squeak-Etoys environment. The workspace contains code blocks for a 'Azul' object. One block sets a variable 'velocidad' to 120. Another block uses a 'temporizador' to move the car over time. A third block checks if the car has reached a certain position and then hides it. A fourth block shows the car's position on a graduated scale from 0 to 300. The interface also shows a preview window with the number '12' and some explanatory text about the exercise.

MATERIALES

- * tutorial interactivo, desarrollado en Squeak-Etoys que permite a los alumnos desarrollar la simulación en ese mismo lenguaje
- * guía de trabajo que articula: 1) problema a resolver, 2) conceptos programación y 3) conceptos de programación
- * ejemplar impreso de la guía de trabajo
- * netbooks del Programa Conectar Igualdad.

CONCLUSIONES

Desarrollar la simulación de un problema puede ser una actividad motivadora y que comprometa a los alumnos en gran medida, al motivar a los alumnos al proporcionarles competencias de utilidad transversal en una actividad en que convergen creatividad, metacognición y aprendizaje cooperativo.

BIBLIOGRAFÍA

- * Gil Martín, M., García Barreto, A. (2006). "Entornos constructivistas de aprendizaje basados en simulaciones informáticas". REEC: Revista electrónica de enseñanza de las ciencias, ISSN-e 1579-1513, Vol. 5, Nº. 2, 2006. Editor: Universidad de Vigo. España. URL: <http://dialnet.unirioja.es/servlet/articulo?codigo=2041055>
- * Kafai, Y., Burke, Q. (2014). "Connected Code - Why Children Need to Learn Programming". The MIT Press, Cambridge-Massachusetts, Estados Unidos, 2014.
- * Zabala, G., Morán, R., Blanco, S. (2013). "Una propuesta de programación en escuela media mediante el desarrollo de videojuegos con Etoys". Jornadas Argentinas de Informática 2013. Córdoba, Argentina, septiembre de 2013. http://caeti.uai.edu.ar/archivos/350_JAIQ_2013.PDF.

CONTACTO

ricardopablo1@gmail.com

-118/132-

16.6 Anexo 6: Taller "Simulaciones en la escuela con Squeak-Etoys", SABTIC 2017

El taller "Simulaciones en la escuela con Squeak-Etoys" fue realizado en el marco del VIII Simposio de Tecnología de la Información (STIN) / V Seminario Argentina-Brasil de Tecnologías de la Información y la Comunicación (SABTIC), realizado en la Sociedade Educacional Três de Maio (SETREM), los días 25 y 26 de mayo de 2017, en la ciudad de Três de Maio, Rio Grande do Sul, Brasil.

A continuación se incluye la planificación del taller y las diapositivas utilizadas en el mismo.

Planificación

Taller: Simulaciones en la escuela con Squeak-Etoys

Docente

Lic. Ricardo Pablo Salvador

Descripción

Utilizar simulaciones como recurso educativo es cada vez más frecuente; involucra el contenido escolar, favorece la reflexión sobre la propia práctica y motiva a los alumnos por la inclusión de tecnología. Estas ventajas se acentúan si la simulación puede ser construida o su diseño modificado por los alumnos. Squeak-Etoys es un entorno de desarrollo basado en Smalltalk, creado por Alan Kay y apoyado por Jerome Brunner y Seymour Papert con el objeto de ser utilizado en educación; facilita la programación y es adecuado para que el docente y los alumnos construyan y manipulen simulaciones; la construcción de código se hace arrastrando y soltando representaciones gráficas de los elementos del código; corre en Linux, Mac y Windows y está traducido en un sinnúmero de idiomas, entre ellos el portugués.

Objetivos

Que los participantes...

- experimenten utilizar y modificar una simulación
- conozcan Squeak-Etoys
- intervengan y modifiquen una simulación existente
- imaginen posibilidades de construcción de material didáctico interactivo útil a sus actividades áulicas

Contenidos

1. La importancia de las simulaciones como recursos didácticos interactivos
2. Qué es Squeak-Etoys
 1. El entorno de trabajo
 2. Los objetos, visores, atributos, instrucciones y variables
 3. Los guiones o programas: el comportamiento de los objetos

3. Modificar una simulación construida con Etoys
4. Dónde obtener más información y ayuda sobre Squeak-Etoys: material y sitios web recomendados.

Desarrollo

Mostrar a los participantes ejemplos de proyectos: cómo funcionan y cómo pueden ser modificados.

Animar a los participantes a modificar un proyecto.

Imaginar una simulación que pueda vincular con un contenido educativo.

Destinatarios

Maestros y profesores y estudiantes de docencia, desde el nivel preescolar hasta nivel secundario.

Modalidad didáctica

Taller, en grupos de 2-3 participantes.

Duración

2 horas (120 minutos) con un break de 10-15 minutos

Cantidad máxima de alumnos: 30

Recursos

- Proyector
- Los alumnos: un computador (escritorio o portátil) por grupo con Windows o Linux (preferentemente Ubuntu o Debian).
- Software instalado
 - Squeak-Etoys V5.0 instalable o portable (en el caso de Windows)
 - versión de Squeak-Etoys disponible para la distribución (en el caso de Linux), o portable (para arquitecturas de 32 bits).
- Acceso a Internet (no excluyente)

Referencias

- Kay, Alan (2005), "Squeak Etoys, Children & Learning", View Point Research Institute, <http://www.vpri.org/html/writings.php>
- Squeakland – "Hogar de Squeak Etoys" (<http://www.squeakland.org/>)
- Recopilación de tutoriales, ejemplos y libros sobre Etoys (sección "Enlaces útiles del blog <http://tallerdesqueak.blogspot.com.ar>)

Ricardo Pablo Salvador
Licenciado en Educación
Rosario, 12 de marzo de 2017

16.6.1 Diapositivas

El taller tuvo dos secciones: las diapositivas consistieron en la introducción teórica, y a continuación se propuso a los alumnos una serie de “desafíos” que debían resolver modificando el código de las simulaciones (*flappy-sabtic*, *rayomcqueen* y *triángulos*), que se proporcionaron con indicaciones que pueden verse en la diapositiva número 9.

The presentation consists of six slides:

- Slide 1/10:** Title 'Simulaciones en la escuela con Squeak-Etoys' by Ricardo Pablo Salvador. Logos for SETREM and V SABTIC (XVII Fórum de Informática) are shown, along with the date 25 a 26 de maio.
- Slide 2/10:** Title 'Simulaciones en la escuela con Squeak-Etoys' and 'Simulaciones y aprendizaje'. A diagram shows 'aprendizaje' at the top, connected to 'simulación' (in a red box). 'simulación' connects to 'usar' and 'construir'. 'usar' and 'construir' both connect to 'crítico', 'activo', 'creatividad', and 'metacognición'.
- Slide 3/10:** Title 'Construir una simulación'. A diagram shows 'comprender...' (with 'sistema involucrado', 'ambiente', 'componentes', 'propiedades', 'procesos y relaciones') and 'hacer...' (with 'puesta a prueba', 'evaluación', 'modificación'). Both lead to a central box labeled 'comprendiendo del problema!'.
- Slide 4/10:** Title 'Programar en la escuela'. A diagram shows 'revalorización' (with 'ciencias de la computación' and 'enseñanza de la programación') and 'competencia general' (with 'recurso educativo'). Both lead to a central box labeled 'promoción del pensamiento computacional'.
- Slide 5/10:** Title 'Iniciativas en el mundo'. A diagram shows 'Raspberry Pi', 'Scratch', 'Squeak-Etoys', 'Code.org', and 'Program.ar' arranged around a central yellow circle.
- Slide 6/10:** Title 'Programación por no expertos'. It shows a snippet of traditional code in Python, followed by a diagram illustrating the evolution of programming: 'código tradicional' leads to 'Interfaces gráficas' and '+ usabilidad', which leads to 'hoy...' and finally 'creación de código gráficamente'.

Simulaciones en la escuela con Squeak-Etoys

Etoys

7/10

Simulaciones y aprendizaje
Construir una simulación
Programar en la escuela
Iniciativas en el mundo
Programación por no expertos
Etoys
Etoys, cómo se usa?
Desafío Etoys
Conclusión

Simulaciones en la escuela con Squeak-Etoys

Etoys, cómo hacer...

8/10

Simulaciones y aprendizaje
Construir una simulación
Programar en la escuela
Iniciativas en el mundo
Programación por no expertos
Etoys
Etoys, cómo hacer...
Desafío Etoys
Conclusión

Simulaciones en la escuela con Squeak-Etoys

Desafío Etoys

9/10

Simulaciones y aprendizaje
Construir una simulación
Programar en la escuela
Iniciativas en el mundo
Programación por no expertos
Etoys
Etoys, cómo hacer...
Desafío Etoys
Conclusión

Simulaciones en la escuela con Squeak-Etoys

Para seguir aprendiendo...

10/10

Simulaciones y aprendizaje
Construir una simulación
Programar en la escuela
Iniciativas en el mundo
Programación por no expertos
Etoys
Etoys, cómo hacer...
Desafío Etoys
Conclusión

16.7 Otros antecedentes relacionados con el tema de la tesis

- Coordinador del “Taller de Programación con Squeak-Etoys”, para alumnos de la Escuela Superior de Comercio “Libertador Gral. San Martín” (UNR), 24 horas cátedra, septiembre-diciembre de 2015.
- Coordinador de “Taller de Squeak-Etoys” en la I Semana de Informatica do Campus Uruguaiana do IF Farroupilha – III Seminario Argentina-Brasil de Tecnologias da Informacao e da Comunicacao (SABTIC), del 9 al 11 de noviembre de 2015, Uruguaiana, RS, Brasil.
- Coordinador del “Taller de Programación con Squeak-Etoys”, para alumnos de la Escuela Superior de Comercio “Libertador Gral. San Martín” (UNR), 18 horas cátedra, 27 de noviembre de 2014.
- Coordinador del Taller “Primera experiencia con Etoys: transformar mi idea en una aplicación” en Squeakfest 2014, UTN Facultad Regional Córdoba, Córdoba Capital, 3-4 de noviembre de 2014.
- Coordinador del Taller “Squeak-Etoys – Taller introductorio” en Squeakfest 2014, UTN Facultad Regional Córdoba, Córdoba Capital, 3-4 de noviembre de 2014.
- Taller “¿Cómo usar Etoys en el aula?”, Squeakfest 2013 Capítulo Argentina, Universidad Tecnológica Nacional Facultad Regional Rosario, 28 y 29 de octubre de 2013
- Taller "Trabajando Ciencias Sociales con Squeak-Etoys", en las Jornadas "Etoys, Programación, Educación e independencia tecnológica", Escuela N° 146 "Juan Antonio Lavalleja", 7 al 9 de junio de 2013, Atlántida, Uruguay.
- Taller “Desarrollando videojuegos con Etoys”, con la participación del alumno Santiago Ricchi, quien explicó el desarrollo del videojuego realizado en el taller de programación (Escuela Superior de Comercio “Libertador Gral. San Martín” U.N.R.), Squeakfest 2013 Capítulo Argentina, Universidad Tecnológica Nacional Facultad Regional Rosario, 28 y 29 de octubre de 2013

- Coordinador del “Taller de Programación con Squeak-Etoys”, para alumnos de la Escuela Superior de Comercio “Libertador Gral. San Martín” (UNR), 32 horas, 2 de noviembre de 2012
- Ponencia “La programación con Etoys como recurso para el aprendizaje”, presentada en las XI Jornadas de Enseñanza Inicial, Primaria y Media Universitaria, La Plata, 24 al 26 de octubre de 2012.
- Disertante en Squeakfest Argentina 2012, Facultad de Tecnología Informática de la UAI, CAETI, CABA, 21 al 23 de mayo de 2012.
- Ponencia “Posibilidades didácticas de Etoys”, presentada en el 1º Seminario Argentina-Brasil de Tecnologías de la Información y la Comunicación (SABTIC), Universidad Católica Argentina - Campus Rosario, 3 y 4 de noviembre de 2011.

16.8 Anexo 8: Índice de figuras

Figura 1: El escritorio de Squeak 2.8 (año 2000).....	51
Figura 2: El escritorio de trabajo de Etoys 5.0.....	56
Figura 3: Barra de herramientas.....	56
Figura 4: Panel de "Provisiones" desplegado.....	57
Figura 5: "Halos" que se hacen visibles al seleccionar un objeto.....	58
Figura 6: Visor del objeto "auto".....	60
Figura 7: Script o guión, el lugar donde se crea la inteligencia de los morphs.....	61
Figura 8: Diapositiva 11 del tutorial interactivo.....	69
Figura 9: 1. En la Escuela, has usado software para.....	78
Figura 10: 2. ¿Usás tecnología informática para resolver problemas cotidianamente?.....	78
Figura 11: 3. Ante un problema, ¿preferís resolverlo individualmente o junto a otros?.....	79
Figura 12: 4.a Noción de "modelo".....	80
Figura 13: 4.b ¿Qué entendés por "simulación".....	80
Figura 14: 5. En tu opinión, pensás que el uso y/o construcción de modelos y/o simulaciones puede contribuir a tus aprendizajes escolares?.....	81
Figura 15: 1.a. Si un tren recorre diez km cada diez minutos, ¿cuántos km recorrerá en una hora?.....	82
Figura 16: 1.b. Si un tren recorre diez km cada diez minutos, ¿se puede asegurar que su velocidad es constante?.....	82
Figura 17: 1.c. Si un tren recorre diez km cada diez minutos, ¿se puede asegurar que su aceleración es nula?.....	83

Figura 18: 2 ¿Es posible que sea verdadera la siguiente proposición: "Una partícula recorrió una distancia de 100m, su desplazamiento fue nulo"?.....	83
Figura 19: 3.a Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h. ¿Cuánto tiempo tardará cada vehículo en recorrer 100km?.....	84
Figura 20: 3.b Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h. ¿Los dos vehículos tienen la misma velocidad?.....	85
Figura 21: 4. Seleccione la opción correcta: En un Movimiento Rectilíneo Uniforme...	85
Figura 22: 5. Seleccione la opción correcta. La distancia recorrida por una partícula...	86
Figura 23: Resultados generales de las evaluaciones para el grupo control.....	87
Figura 24: 1. En la Escuela, has usado software para.....	89
Figura 25: 2. ¿Usás tecnología informática para resolver problemas cotidianamente?	90
Figura 26: 3. Ante un problema, ¿preferís resolverlo individualmente o junto a otros?	91
Figura 27: 4.a ¿Qué entendés por "modelo"?.....	91
Figura 28: 4.b ¿Qué entendés por "simulación"?.....	92
Figura 29: 5. Opiniones favorables al uso de simulaciones para el aprendizaje.....	92
Figura 30: 1.a.- ¿Cuántas kilómetros recorrerá en una hora?.....	93
Figura 31: 1.b ¿Se puede asegurar que su velocidad es constante?.....	94
Figura 32: 1.c ¿Se puede asegurar que su aceleración es nula?.....	94
Figura 33: 2. ¿Es posible que sea verdadera la siguiente proposición: "Una partícula recorrió una distancia de 100m, su desplazamiento fue nulo"?.....	95

Figura 34: 3.a. ¿Cuánto tiempo tardará cada vehículo en recorrer 100km?.....	95
Figura 35: 3.b. ¿Los dos vehículos tienen la misma velocidad?.....	96
Figura 36: 4. En un Movimiento Rectilíneo Uniforme.....	96
Figura 37: 5. La distancia recorrida por una partícula.....	97
Figura 38: Resultados generales de las evaluaciones del grupo experimental.....	98

16.9 Anexo 9: Índice de tablas

Tabla 1: Taxonomía revisada de Bloom.....	26
Tabla 2: Unidades didácticas del espacio curricular “Física”.....	72
Tabla 3: El espacio curricular “Informática”	73
Tabla 4: Cantidad de alumnos en el grupo control.....	77
Tabla 5: Cantidad de alumnos según sexo.....	77
Tabla 6: Cantidad de alumnos según edad.....	77
Tabla 7: 1. En la Escuela, has usado software para... (marca con una “x”)	77
Tabla 8: 2. En la Escuela, has usado software para... (marca con una “x”)	78
Tabla 9: 3. Ante un problema preferís resolverlo individualmente o junto a otros?.....	79
Tabla 10: 4. Qué entendés por.....	79
Tabla 11: 5. En tu opinión, pensás que el uso y/o construcción de modelos y/o simulaciones puede contribuir a tus aprendizajes escolares?.....	80
Tabla 12: 1. Responde las siguientes preguntas: Si un tren recorre diez kilómetros cada diez minutos.....	81
Tabla 13: 2. Responde la siguiente pregunta. ¿Es posible que sea verdadera la siguiente proposición: “Una partícula recorrió una distancia de 100m, su desplazamiento fue nulo”?.....	83
Tabla 14: 3. Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h.....	84
Tabla 15: 4. Seleccione la opción correcta: En un Movimiento Rectilíneo Uniforme....	85
Tabla 16: 5. Seleccione la opción correcta. La distancia recorrida por una partícula...	86
Tabla 17: Resultados generales por evaluación para el grupo control.....	86
Tabla 18: Cantidad de alumnos en el grupo experimental.....	88

Tabla 19: Cantidad de alumnos según sexo.....	88
Tabla 20: Cantidad de alumnos según edad.....	89
Tabla 21: 1. En la Escuela, has usado software para... (marca con una "x").....	89
Tabla 22: 2. Usás tecnología informática para resolver problemas cotidianamente?....	90
Tabla 23: 3. Ante un problema preferís resolverlo individualmente o junto a otros?....	90
Tabla 24: 4. Qué entendés por.....	91
Tabla 25: 5. En tu opinión, pensás que el uso y/o construcción de modelos y/o simulaciones puede contribuir a tus aprendizajes escolares?.....	92
Tabla 26: 1. Responde las siguientes preguntas: Si un tren recorre diez kilómetros cada diez minutos.....	93
Tabla 27: 2. Responde la siguiente pregunta. ¿Es posible que sea verdadera la siguiente proposición: “Una partícula recorrió una distancia de 100m, su desplazamiento fue nulo”?.....	94
Tabla 28: 3. Los automóviles A y B se mueven por una ruta recta. El automóvil A va hacia el este, su velocímetro indica 100km/h. El automóvil B va hacia el oeste su velocímetro indica 100km/h.....	95
Tabla 29: 4. Seleccione la opción correcta: En un Movimiento Rectilíneo Uniforme....	96
Tabla 30: 5. Seleccione la opción correcta. La distancia recorrida por una partícula...	97
Tabla 31: Resultados generales por evaluación para el grupo experimental.....	98

17 Bibliografía

- Amily Shafila Sharif (2007), "Introducing Squeak Etoys to enhance Science and Mathematics learning", 1st International Malaysian Educational Technology Convention, <http://www.fp.utm.my/epusatsumber/listseminar/20.KonventionTP2007-20/pdf/volume1/69-amily.pdf>
- Astudillo, Javier (2016), "Estrategias de diseño y ensamblaje de Objetos de Aprendizaje", Tesis de Magister en Tecnología Informática Aplicada en Educación, Facultad de Informática, Universidad Nacional de La Plata.
- Black, A., Ducasse, S., Nierstrasz, O., Pollet, D. (2007) "Squeak by example", Square Bracket Associates, Suiza, <http://www.iam.unibe.ch/~scg/SBE/SBE.pdf>
- Churches, Andrew (2009), "Taxonomía de Bloom para la era digital", <http://www.eduteka.org/TaxonomiaBloomDigital.php>, traducido de Churches, Andrew "Bloom's Digital Taxonomy", <http://edorigami.wikispaces.com>
- Computer Science Teachers Association (CSTA) (2011) "Operational definition of computational thinking for K–12 education", <https://www.csteachers.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>
- Freudenberg, Bert "SqueakJS – A Squeak VM in JavaScript", <https://squeak.js.org/run/>, consultado el 27 de diciembre de 2016.
- Fundación Sadosky (2013), "CC-2016 - Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas", pg. 23, <http://www.fundacionsadosky.org.ar/wp-content/uploads/2014/06/cc-2016.pdf>
- Gil Martín, M., García Barneto, A. (2006), "Entornos constructivistas de aprendizaje basados en simulaciones informáticas", REEC: Revista electrónica de enseñanza de las ciencias, ISSN-e 1579-1513, Vol. 5, Nº. 2, <http://dialnet.unirioja.es/servlet/articulo?codigo=2041055>
- Gómez, M. S. (2007) "Buenas Prácticas en la Creación de Serious Games (Objetos de Aprendizaje Reutilizables)" In M. Benito, J. Romo & J. Portillo (eds.), SPDECE, : CEUR-WS.org
- Gordon, Geoffrey (1980), "Simulación de sistemas", Ed. Diana, México D.F., 1° Ed. 1980, 6° reimpresión 1991
- Ingalls, D., Kaehler, T., Maloney, J., Wallace, S., Kay, Alan (1997) "Back to the future: the story of Squeak, a practical Smalltalk written in itself", View Point Research Institute, http://www.vpri.org/pdf/tr1997001_backto.pdf
- Kafai, Y., Burke, Q. (2014), "Connected Code – Why Children Need to Learn Programming", The MIT Press
- Kay, Alan (2005), "Squeak Etoys, Children & Learning", Viewpoints Research Institute, <http://www.vpri.org/html/writings.php>

- Kay, Alan (2007) "Children Learning By Doing – Etoys on the OLPC XO", Viewpoints Research Institute, <http://wiki.laptop.org/images/2/28/OLPCEtoys.pdf>
- Kay, Alan, Rose, K., Ingalls, D., Kaehler, T., Maloney, J. y Wallace, S. (1997), http://www.vpri.org/pdf/hc_etoys_sim_1997.pdf
- Kelton W., Sadowski R., Sturrock D. (2008), "Simulación con Software Arena", 4° edición Ed. McGraw-Hill Inc., España
- Lye, S.Y., Koh, J.H.L. (2014), "Review on teaching and learning of computational thinking through programming: What is next for K-12? ", artículo en "Computers in Human Behavior", Dec2014, Vol. 41, p51-61. 11p. <http://www.sciencedirect.com/science/article/pii/S0747563214004634>
- One Laptop per Child, Wiki del proyecto, URL: http://wiki.laptop.org/go/Etoys#Projects_for_Squeak_on_the OLPC_XO
- Papert, Seymour (1987), "Desafío a la mente – Computadoras y educación", Ediciones Galápago, Buenos Aires.
- Papert, Seymour (1999), "Logo Philosophy and Implementation", Logo Computer Systems Inc. Introduction.
- Queiruga, C., Fava, L. (2013) "Enseñar a Programar en la Escuela Secundaria. Experiencias del proyecto Java en Escuelas Técnicas", "Memorias del 1º Congreso de Extensión de la Asociación de Universidades Grupo Montevideo -AUGM-Extenso 2013", Ed: Universidad de la República, ISBN 978-9974-0-1038. URL: http://www.linti.unlp.edu.ar/uploads/docs/ensenar_a_programar_en_la_escuela_secundaria_experiencias_del_proyecto_java_en_escuelas_tecnicas.pdf
- Queiruga, C., y otros (2014), "El juego como estrategia didáctica para acercar la programación a la escuela secundaria", Laboratorio de Investigación en Nuevas Tecnologías Informáticas, Facultad de Informática, Universidad Nacional de La Plata, trabajo presentado en el XVI Workshop de Investigadores en Ciencias de la Computación 2014. http://jets.linti.unlp.edu.ar/uploads/docs/wicc_2014.pdf
- Resnick, Mitchel (2002), "Rethinking Learning in the Digital Age", The Media Laboratory, Massachusetts Institute of Technology, <http://llk.media.mit.edu/papers/mreswef.pdf>
- Resnick, M., Maloney, J., Monroy-Hernández, A. y otros (2010), "Scratch: Programming for All", Communications of the ACM, Vol. 52 No. 11, Pages 60-67. URL: <http://cacm.acm.org/magazines/2009/11/48421-scratch-programming-for-all/fulltext>
- Resnick, Mitchel (2012), "Reviving Papert's Dream", "Educational-technology – the magazine for managers of change in education", Volume 52, Number 4, JulyAugust 2012, <http://web.media.mit.edu/~mres/papers/educationaltechnology2012.pdf>
- Resnick, M., Brennan, K., (2012), "New frameworks for studying and assessing the development of computational thinking", http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

- Riesco, M., Fondón, M., Álvarez, D., López, B., Cernuda, A., Juan, A. (2014), "Informática: materia esencial en la educación obligatoria del siglo XXI", Revista Electrónica ReVisión, Vol 7, N°3, [http://www.aenui.net/ojs/index.php?journal=revision&page=article&op=viewArticle&path\[\]&path\[\]](http://www.aenui.net/ojs/index.php?journal=revision&page=article&op=viewArticle&path[]&path[])
- Squeak-Etoys, sitio oficial: <http://squeakland.org/about/>
- Stager, Gary (2005), "Papertian Constructionism and the Design of Productive Contexts for Learning ", EuroLogo X, Warsaw – Polonia, 28 al 31 de Agosto de 2005, <http://stager.org/articles/eurologo2005.pdf>
- Taub, Armoni, Bagno, Ben-Ari (2015), "The effect of computer science on physics learning in a computational science environment", Computers & Education Volume 87, September 2015, Pages 10–23, URL: <http://www.sciencedirect.com/science/article/pii/S0360131515000913>
- The MIT License, <https://opensource.org/licenses/mit-license.php>
- Vicario Solórzano, Claudia M. (2009), "Construcción. Referente sociotecnopedagógico para la era digital", Revista Innovación Educativa vol. 9, núm. 47, abril/junio, 2009, pp. 4550, Instituto Politécnico Nacional, México, <http://www.redalyc.org/pdf/1794/179414895005.pdf>
- Wing, Jeannette M. (2010), "Computational Thinking: What and Why?", Scholl of Computer Science, Carnegie Mellon University, <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Yohan Hwang, Kongju Mun, Yunebae Park (2016), "Study of Perception on Programming and Computational Thinking and Attitude toward Science Learning of High School Students through Software Inquiry Activity: Focus on using Scratch and physical computing materials", Journal of the Korean Association for Science Education, URL: http://koreascience.or.kr/article/ArticleFullRecord.jsp?cn=GHOBX_2016_v36n2_325
- Zabala, G., Morán, R. y Blanco, S. (2010), "Physical Etoys: una herramienta libre para el aprendizaje de tecnología con material concreto", en Revista TE&ET edición 2010, Universidad Nacional de La Plata, http://sedici.unlp.edu.ar/bitstream/handle/10915/18405/Documento_completo.pdf?sequence=1
- Zabala, G., Morán, R., Blanco, S. (2013), "Una propuesta de enseñanza de programación en escuela media mediante el desarrollo de videojuegos con Etoys", JAIIO 2013. Córdoba, Argentina. Septiembre de 2013, <http://imgbiblio.vaneduc.edu.ar/fulltext/files/TC113306.pdf>
- Zabala, G., Morán, R., Blanco, S., Teragni, M. (2012), "Physical Etoys: la libertad más allá del mundo digital", 10º Simposio sobre la Sociedad de la Información, SSI 2012, JAIIO – Universidad de La Plata – Buenos Aires, Agosto de 2012, http://41jaiio.sadio.org.ar/sites/default/files/7_SSI_2012.pdf