



Dirección de Educación
a Distancia y Tecnología
Prosecretaría de Grado
SECRETARÍA DE
ASUNTOS ACADÉMICOS



UNIVERSIDAD
NACIONAL
DE LA PLATA

Autores: Profesores Silvia Ana Cappelletti, Claudia Marcela Cappelletti, y Ernesto Guillermo Gottlieb

Institución: Colegio Nacional “Rafael Hernández” - UNLP

Dirección/es de correo/s electrónico/s:

silviacapp@yahoo.com.ar, ccappe1@hotmail.com,
gottliebernestoguillermo@gmail.com

Eje temático: Escuela Media y prácticas con tecnologías digitales.

Tipo de comunicación: Ponencia “Procesando códigos en Processing”

Abstract

El trabajo presenta una experiencia áulica realizada con alumnos de 3° año del Colegio Nacional Rafael Hernández (UNLP) durante la cursada anual 2014 de la materia Informática. Dicha materia pertenece a la Sección Informática del Departamento de Ciencias Exactas.

Ante las dificultades que observamos en nuestros alumnos para resolver problemas a través del razonamiento lógico, teniendo en cuenta también la importancia del uso de las Tics en las aulas, el alto impacto que producen las nuevas tecnologías en los adolescentes y vivenciando diariamente la especial motivación que les provoca el abordaje de situaciones lúdicas, nos surge la necesidad de combinar estos factores para favorecer y consolidar en los alumnos su capacidad lógica, simbólica, su creatividad, su curiosidad. Para ello, debieron desarrollar pequeños programas en computadora que les permitieron pensar y aprender de manera diferente a las que estaban acostumbrados. Se utilizó el lenguaje de programación Processing mediante el cual se realizaron simulaciones generadas en contextos lúdicos, reales o fantásticos.

En esta ponencia, se presentan diferentes instancias de dichas producciones áulicas variando los niveles de dificultad, como así también las distintas reacciones positivas o negativas que los alumnos tuvieron a lo largo del proceso de aprendizaje.

Palabras claves

Programación, Simulación, Razonamiento lógico, Código, Software libre, Situaciones lúdicas, Educación media, Enseñanza, Aprendizaje.

Introducción

Esta ponencia muestra experiencias realizadas con alumnos de 3° año del Colegio Nacional Rafael Hernández de la UNLP, durante la cursada anual 2014 de la materia Informática. Dicha materia pertenece a la Sección Informática del Departamento de Ciencias Exactas.

Desde nuestra asignatura y basándonos en los principios y metodologías que proporciona la Ingeniería de Software¹ como disciplina que se ocupa del desarrollo, operación y mantenimiento de programas informáticos, sumado a la necesidad de aprovechar el alto impacto que provocan las TIC's en los adolescentes, proponemos un trabajo áulico orientado a la resolución de problemas mediante la computadora, promoviendo en los alumnos: la exploración, el tanteo, el ensayo y error sistemático, las modificaciones, el planteo de posibles hipótesis, la construcción de modelos a partir de simulaciones pensadas en diferentes contextos, reales o fantásticos².

Como docentes de la materia Informática a cargo de alumnos adolescentes, pretendemos también favorecer y consolidar en ellos, su capacidad lógica, simbólica, creativa, su curiosidad, como así también su expresión oral y escrita a lo largo del proceso de elaboración de sus producciones escolares informáticas, contribuyendo a formar alumnos con conocimientos significativos, autónomos y capaces de aprender a aprender (Anijovich y Mora, 2009).

James Paul Gee (2004) apunta a la importancia de descubrir los procesos de aprendizajes jugando videojuegos, afirmando que a través de éstos, los jugadores tienen la exigencia de aprender y pensar de manera diferente a la que están

¹https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software

² http://www.eyg-fere.com/ticc/archivos_ticc/anayluis.pdf

acostumbrados. En esta ponencia queremos resaltar el valor educativo de los procesos mencionados cuando planteamos a nuestros alumnos el desarrollo de diversas *situaciones lúdicas mediadas por TIC`s*³. Para llevar adelante dichos procesos, elegimos Processing⁴ que es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java. Su uso es sencillo y permite la elaboración de proyectos multimedia.

Para concluir y promoviendo fundamentalmente la necesidad de resolver problemas a través de razonamientos lógicos, queremos destacar la secuencia de acciones que debieron realizar los alumnos con la guía del docente:

- Identificación del problema a resolver o representar.
- Ambientación del problema en un contexto o escenario.
- Análisis y razonamiento lógico para resolver el problema.
- Codificación y generación del programa.
- Chequeo de errores y retroalimentación.
- Generación de nuevas simulaciones en contextos diferentes, usando la creatividad e innovación.

Desarrollo

En nuestras clases, los alumnos trabajaron en grupos de a dos. Cada grupo contó con una PC, aunque algunos alumnos prefirieron utilizar sus propias netbooks.

Clase a clase, los alumnos agregaron a sus carpetas individuales el material didáctico elaborado por los docentes, que incluyó los códigos correspondientes a las simulaciones planteadas en clases, descargados de un sitio web indicado por el docente.

³ Hasta el años 2012 trabajábamos curricularmente con los alumnos de 3º año utilizando otro software y planteando la resolución de situaciones problemáticas, mayormente, matemáticas.

⁴ <https://www.processing.org/>

PROCESANDO CÓDIGOS EN PROCESSING

Para lograr que los alumnos resolvieran situaciones problemáticas lúdicas, reales o fantásticas, necesitamos inicialmente que ellos incorporaran conocimientos básicos del software a utilizar.

PROGRAMACION BASICA Sin la posibilidad de interacción con el usuario

Desde un comienzo, se presentó a los alumnos el concepto de objeto y a lo largo de las clases, se consideraron como objetos tanto las **figuras geométricas** dibujadas como las **imágenes** que luego se incorporaron en los distintos programas.

➤ Configuración de la pantalla e inserción de objetos

Comenzamos mostrando a los alumnos cómo es la estructura general de un programa usando Processing.

Se definió la *función void setup()*⁵ en donde se estableció el tamaño y el color de fondo de la pantalla, con la posibilidad de redefinir estos datos en otras instancias. Para aplicar colores a los objetos se utilizó la paleta de colores RGB (Red, Green, Blue), que a partir de la combinación de tres números correspondientes al rojo, verde y azul es posible obtener nuevos colores (**ver Figura1**).

Una vez escritos los códigos del programa los alumnos pudieron ver las ejecuciones de los mismos a través del botón “play”.

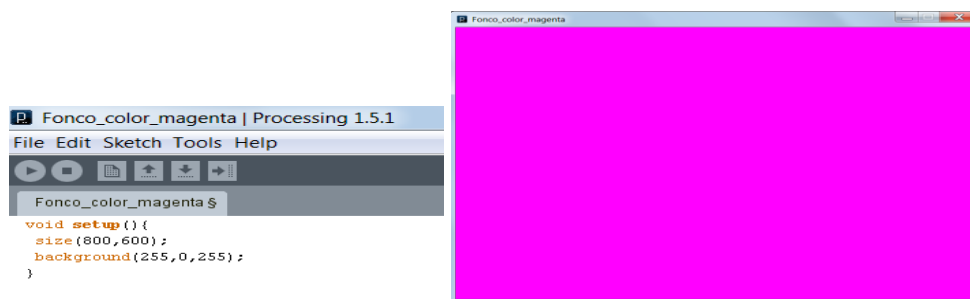


Figura1: Ejemplo de código que establece de color Magenta el fondo de la pantalla.

⁵ En esta función se establecen los valores iniciales del programa.

Luego se mostró a los alumnos cómo dibujar figuras geométricas en la pantalla: rectángulos, cuadrados, elipses, círculos, triángulos y líneas usando código (ver **Figura2**).

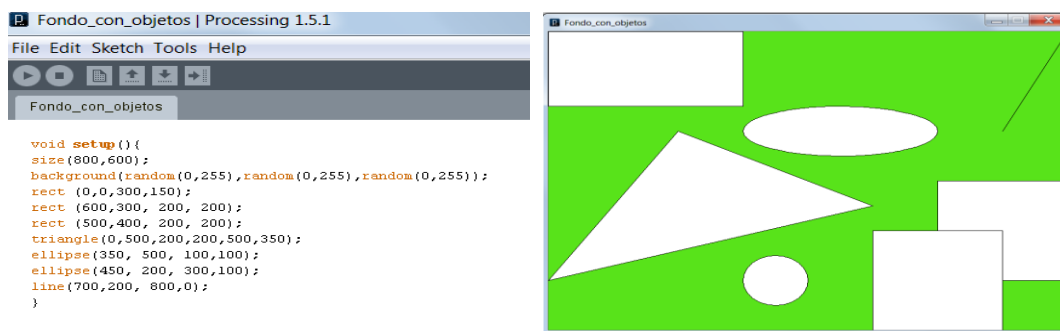


Figura2: Ejemplo de código para dibujar figuras geométricas básicas.

Una vez dibujadas las figuras, se trabajó con el formato de las mismas: se aplicó color al relleno, a los bordes, se incorporaron distintos grados de transparencia en los rellenos de las figuras, grosores en los bordes, etc (**ver Figura3**).

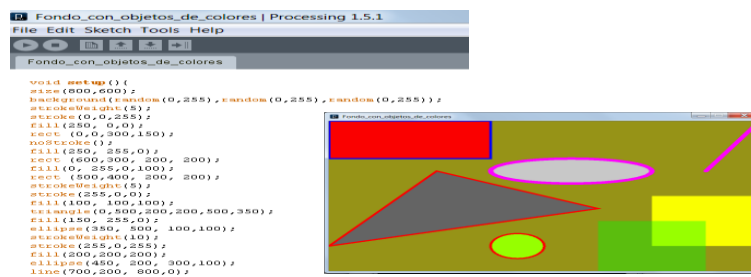


Figura3: Ejemplo de figuras geométricas básicas con formatos personalizados.

En una segunda instancia, se explicó a los alumnos los pasos necesarios para insertar imágenes en el programa, considerando la importancia destacada de la carpeta DATA y la ubicación de la misma al momento de la ejecución del programa (**ver Figura4**).



Figura4: Ejemplo de imágenes incorporadas, una de fondo y otra superpuesta.

➤ **Ubicación y movimiento de objetos en forma aleatoria: una vez que los** alumnos lograron dibujar figuras e insertar imágenes, se trabajó con la ubicación de los objetos en distintos puntos de la pantalla, teniendo en cuenta diferentes tamaños tanto de pantalla como de objetos. Para ello se definió la *función void draw()*⁶.

En simultáneo se consideró el uso de *la función Random* (azar o aleatorio), para variar los formatos de los objetos (colores en rellenos y bordes, grosores en bordes, transparencias) como así también sus ubicaciones, que dan cuenta de un **movimiento aleatorio de los objetos dentro de los límites de la pantalla (ver Figura5)**. También se trabajó con colores aleatorios en el fondo de la pantalla, para que cada vez que se ejecutara el programa, cambiara de color.

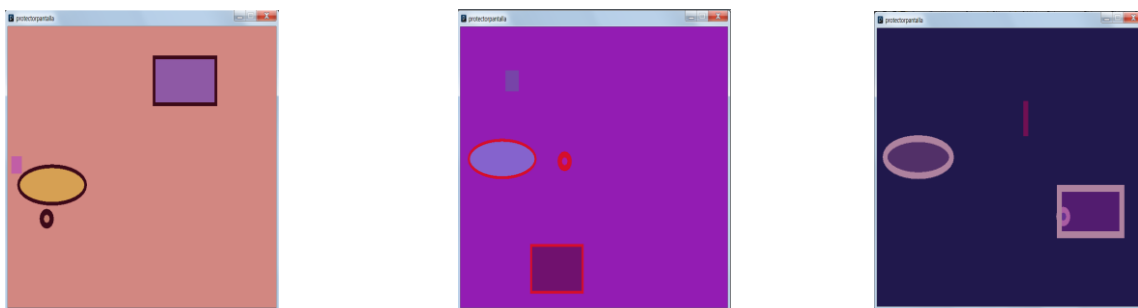


Figura5: Secuencia que muestra el uso de la función *Random* para variar ubicaciones, tamaños y colores de diferentes objetos.

➤ **Uso de la función background:** Resultó importante que los alumnos visualicen en esta instancia, qué ocurre si ocultamos la función *background*. Esta función se utiliza para establecer un color en el fondo de la pantalla, pero tiene un efecto adicional, y es que permite *limpiar la pantalla*, es decir, borra todo lo que había anteriormente. Si la ocultamos, veremos todos los objetos dibujados o insertados hasta el momento, a modo de un historial de objetos (ver **Figura6**). El uso de esta función nos permitió tener una diversidad de resultados con mínimas modificaciones en el código.

⁶ Esta función se activa automáticamente y se ejecutan repetidamente las líneas de código que contiene, hasta que se indique alguna condición de cierre.



Figura6: Secuencia que muestra el historial de objetos dibujados ocultando la función *background*.

- **Movimiento de objetos usando variables:** A continuación se incorporó el concepto de *variable*⁷, para simular el movimiento de los objetos en diferentes direcciones dentro de los límites de la pantalla. Con el movimiento de objetos se logró la simulación de cintas que se cruzan, autos que se mueven, manzanas que caen de un árbol, etc.
- **Uso de la sentencia condicional:** Una vez abordado el movimiento de objetos, surgió la necesidad de analizar la colisión de objetos, el choque de un objeto con el borde de la pantalla, la interrupción del movimiento de un objeto, etc. Para ello se incorporó la sentencia condicional, que es una estructura de control que permite incorporar la capacidad de decisión en el programa.
- **Inclusión de textos:** A través de la incorporación de los mismos pudimos mostrar en pantalla resultados de contadores o variables, escribir mensajes de alerta (**ver Figura7**) o de logros obtenidos, anunciar carteles simples (**ver Figura8**), especificar las instrucciones de un juego, etc. Para ello recurrimos nuevamente a la carpeta DATA en donde se guarda el tipo y tamaño de las letras elegidas.

⁷ Las variables son estructuras que permiten almacenar información temporalmente en la memoria de una computadora, con la posibilidad de “variar” su contenido a lo largo del programa.

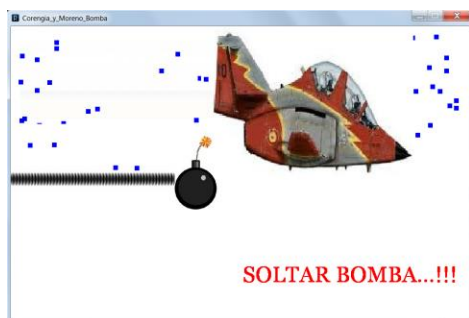


Figura7: Texto como mensaje de alerta.

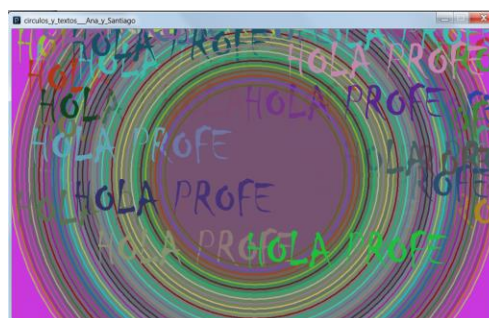


Figura8: Texto como cartel simple.

PROGRAMACIÓN AVANZADA Con la posibilidad de interacción del usuario

➤ **Inclusión de textos:** se utilizaron para mostrar un cartel en pantalla o pedirle al usuario que incorpore información a través del teclado. Por ejemplo, para determinar: el valor inicial de una cuenta regresiva, la velocidad del movimiento de un objeto, cantidad de autos en una carrera, etc.

➤ **Uso del teclado:** se incorporaron funcionalidades a determinadas teclas como el Enter, la barra espaciadora, F1, Control, etc. Por ejemplo, para determinar: el movimiento de objetos a través de las **flechas del teclado (ver Figura9)**, el inicio de un juego clickeando la tecla **Enter**, el disparo de proyectiles usando la tecla **F1**, el frenado de un objeto usando la tecla **S**.



Figura9: Secuencia que simula movimiento a través del uso de las flechas del teclado: derecha, abajo, izquierda.

➤ **Uso del mouse:** logramos con su uso provocar el inicio de nuevos procesos. Trabajamos con dos opciones.

Opción1: Click sobre uno de los botones del mouse. Por ejemplo, para determinar: el inicio de un juego clickeando un cartel que contenga la palabra COMENZAR, el disparo de proyectiles clickeando la imagen de un cañón, la ejecución de un penal en un partido de fútbol (**ver Figura10**).



Figura10: Secuencia que simula la ejecución de un penal haciendo *click* en uno de los botones del mouse.

Opción2: Click sobre uno de los botones del mouse y un posterior arrastre del mouse. Por ejemplo, para determinar: el traslado de un objeto desde un lugar de la pantalla hacia otro, el arrastre de las piezas de un rompecabezas para su armado, etc.

Finalmente, los alumnos elaboraron un Trabajo Final Integrador con los contenidos detallados anteriormente. En dicha instancia simularon una situación lúdica, real o fantástica, en un contexto propuesto por ellos.

Conclusiones

Resultó muy satisfactorio como docentes, observar y guiar el proceso de aprendizaje que lograron los alumnos ante las distintas situaciones problemáticas planteadas. Es interesante además, destacar las variadas actitudes demostradas por los diferentes grupos de alumnos:

- Algunos expresaron su enojo ante la “imposibilidad” de avanzar en el proceso de desarrollo de las simulaciones solicitadas, quienes en general necesitaron de mayor asistencia docente.

- Otros se animaron, preguntaron y propusieron posibles soluciones según sus propias vivencias lúdicas y lograron relacionar contenidos ya abordados en clases anteriores.
- Otros grupos, manifestaron un “desequilibrio” positivo, propio de un desafío, que los llevó a “codificar” antes de “razonar” y ante la frustración de no lograr los resultados esperados, pidieron asistencia docente.
- Algunos grupos, aunque en menor número, mostraron mayor entusiasmo y autonomía al momento de elaborar sus producciones.

Los debates grupales generados a partir de desconciertos, curiosidades, planteos erróneos, nos permitió como docentes, observar el crecimiento de cada grupo respetando individualidades en cuanto a tiempos, capacidades y motivaciones de los alumnos.

Finalmente, a la hora de evaluar, no nos detuvimos simplemente en los resultados obtenidos en la elaboración de los productos finales, sino que **valoramos todo el proceso de aprendizaje logrado por los alumnos.**

Bibliografía

- Anijovich, Rebeca; Mora, Silvia (2009): “Estrategias de enseñanza. Otra mirada al quehacer en el aula”. Buenos Aires, Aique.
- Gee, James Paul (2004): “Lo que nos enseñan los videojuegos sobre el aprendizaje y alfabetismo”. Málaga, Aljibe.
- Marqués, P (1999). Concepciones sobre el aprendizaje. *Documento digital disponible en: <http://www.peremarques.net/aprendiz.htm>.*
- Moroni, N., & Señas, P. (2005). Estrategias para la enseñanza de la programación. In I Jornadas de Educación en Informática y TICs en Argentina. *Documento digital disponible en: <http://sedici.unlp.edu.ar/handle/10915/18901>*
- Material didáctico elaborado por los docentes.