

Plataforma para la Programación Tangible

Celeste Ramos

Jorge Rodríguez

Laura Cecchi

{celeste.ramos, j.rodri, lcecchi }@fi.uncoma.edu.ar

Grupo de Investigación en Lenguajes e Inteligencia Artificial
Departamento de Teoría de la Computación - Facultad de Informática
UNIVERSIDAD NACIONAL DEL COMAHUE

Resumen

En las últimas décadas la enseñanza de las Ciencias de la Computación en el ámbito de la educación ha ido evolucionando. En este contexto, se fue incorporado a la currícula la utilización de distintas herramientas que permitan a los estudiantes comenzar a programar.

Para desarrollar un programa, en la mayoría de los lenguajes existentes, se requiere de ciertos conocimientos y habilidades previas, como por ejemplo saber contar, leer y escribir.

La Programación Tangible permite la utilización de piezas físicas como componentes de programación. Cada pieza tiene características y funcionalidades específicas que al secuenciarlas es posible generar un programa.

En este trabajo se presenta una Línea de Investigación que propone definir y diseñar un lenguaje de Programación Tangible. Asimismo, se diseñará e implementará un entorno Web que permita, a los estudiantes no alfabetizados, programar en dicho lenguaje.

Palabras Clave: PROGRAMACIÓN TANGIBLE - ENSEÑANZA DE LA PROGRAMACIÓN - EDUCACIÓN EN CIENCIAS DE LA COMPUTACIÓN - PLATAFORMA DE PROGRAMACIÓN TANGIBLE - ENTORNO WEB PARA LA PROGRAMACIÓN TANGIBLE.

Contexto

Esta línea de investigación se desarrolla, por un lado, en el contexto de los temas de in-

terés que promueve el Grupo de Investigación en Lenguajes e Inteligencia Artificial (GILIA), de la Facultad de Informática. En particular, se enmarca en el ámbito de dos proyectos de investigación miembros del GILIA, ambos financiados por la Universidad Nacional del Comahue y con una duración de cuatro años a partir de enero del 2017: *Agentes Inteligentes. Modelos Formales y Aplicaciones para la Educación (04/F015)* y *Agentes Inteligentes y Web Semántica (04/F014)*.

Por otro lado, el trabajo está en el contexto del Convenio Marco de Colaboración firmado durante 2016 entre la Facultad de Informática y el Ministerio de Educación de la Provincia del Neuquén. Particularmente, se trabaja con el Consejo Provincial de Educación de la Provincia de Neuquén.

1. Introducción

Hoy en día es muy común ver a niños desde los 3 años utilizar celulares, tablets o computadoras para acceder a distintas aplicaciones con fines lúdicos o formativos.

En la última década en muchas escuelas la educación en informática tradicional se ha ido modificando. Progresivamente, la formación en ofimática está perdiendo relevancia dando lugar a la incorporación de tópicos que ayudan a desarrollar el Pensamiento Computacional [5, 6, 10].

El Pensamiento Computacional es una forma de trasponer prácticas y conceptos fundamentales de las Ciencias de la Computación para la resolución de problemas no necesariamente computacionales [12, 11]. En este contexto se están utilizando distintas herramientas que permiten a los estudiantes comenzar a programar alrededor de los 8 años.

Una de las formas más amigables para desarrollar el Pensamiento Computacional, en niños desde los 3 años de edad, es la Programación por medio de Interfaces de Usuario Tangibles (TUI) o simplemente Programación Tangible. Se fundamenta en que los niños aprenden de una manera activa: aprenden a través de experiencias con personas, objetos y demás cosas que puedan percibir de su mundo.

Las TUI [4] son una forma de interacción hombre máquina, novedosa y poco explorada, donde la interfaz se amplía al integrar la computación con objetos físicos y entornos que están vinculados a representaciones digitales. En definitiva, permiten que el usuario pueda manipular de forma física la información digital [8].

En la Programación Tangible los estudiantes construyen código ensamblando, encadenando y/o conectando objetos físicos, que representan instrucciones precisas, que luego se traducen a un lenguaje de programación generando un programa [9].

Actualmente, existen múltiples herramientas de Programación Tangible [7] que permiten a los estudiantes aprender a programar desde los primeros años de edad.

Si bien, estas herramientas están adaptadas a las distintas edades, muchas no son fáciles de combinar debido a la exclusividad de su diseño, ya que sus licencias de software y/o hardware suelen ser propietarias. Por otro lado, su alto costo las hace poco accesibles para la gran mayoría de los niños o para las escuelas a las que éstos concurren.

En esta Línea de Investigación se propone definir y diseñar un Lenguaje de Programación Tangible, que sea comprensible para estudiantes no alfabetizados. A diferencia de las herramientas que se estuvieron analizando, para su definición se priorizará la utilización de elementos de bajo costo, accesibles y de uso cotidiano.

Además, se desarrollará una herramienta Web con licencia Open Source, que permita a los estudiantes programar en el lenguaje definido. Dicha herramienta será accesible desde cualquier navegador, por lo que sólo se necesitará tener un dispositivo con browser y conexión a internet, lo que permitirá su fácil acceso sin necesidad de instalarla. Por otra parte, para su utilización no se requerirá de dispositivos de hardware específicos o exclusivos.

El trabajo presentado está estructurado como sigue. En la siguiente sección se introducen los proyectos de investigación que dan contexto al trabajo y la línea en desarrollo. En la sección 3 se detallan los avances en este trabajo y los trabajos futuros. Finalmente, se comentan aspectos en relación a la formación de recursos humanos.

2. Línea de investigación y desarrollo

Esta línea de investigación, desarrollo y transferencia propone definir y diseñar un lenguaje de Programación Tangible. Asimismo, tiene como objetivo diseñar e implementar un Entorno Web de Programación Tangible que permita a estudiantes no alfabetizados programar en dicho lenguaje.

Este trabajo se desarrolla en el marco de las actividades de los proyectos de investigación *Agentes Inteligentes. Modelos Formales y Aplicaciones para la Educación* y *Agentes Inteligentes y Web Semántica*. Por un lado el proyecto *Agentes Inteligentes. Modelos Formales y Aplicaciones para la Educación*, busca desarrollar modelos teóricos que contribuyan a la producción de un marco conceptual de referencia, a fin de asistir la inclusión de la computación en la educación. En este sentido se plantea como objetivo el diseño e implementación de herramientas que los soporten. Por otra parte, el diseño de lenguajes caracterizados por gramáticas formales y el desarrollo de entornos que cumplan los principios de la Web Semántica son objetivos específicos del proyecto de investigación *Agentes Inteligentes y Web Semántica*.

Así, la definición y caracterización formal de un lenguaje de Programación Tangible, como el desarrollo de un entorno Web de programación, que contribuye a hacer posible la construcción de saberes del área de conocimientos algoritmos y programación, a sujetos no alfabetizados, es de interés para ambos proyectos, colaborando en su diseño e implementación. Los resultados de este trabajo permitirán ampliar la población estudiantil facilitando el desarrollo de los procesos de enseñanza y de aprendizaje de la programación.

En este sentido se plantean los siguientes ejes de trabajo para el desarrollo de la plataforma:

Lenguaje de Programación Tangible

Los Lenguajes de Programación Tangible (LPT) asumen que piezas físicas pueden constituirse en elementos sintácticos de un lenguaje (variables, instrucciones, operadores, entre otros). La colección de piezas físicas proporciona al programador, posiblemente un sujeto no alfabetizado, la posibilidad de secuenciar u ordenar elementos físicos para construir un algoritmo [7, 9].

Como punto de partida de este trabajo se define un LPT simple, que no involucra el uso de variables, ni de las estructuras de control alternativa y repetitiva, y que se compone de un conjunto reducido de primitivas. Se espera que a través de este lenguaje se pueda programar un comportamiento sencillo de un agente o bot, en un mundo representado en forma de cuadrícula en un entorno Web. Las instrucciones para modelar el comportamiento del agente incluirán, entre otras, avanzar o girar a derecha.

En esta plataforma, las piezas físicas con las que se trabajará serán trozos de papel, cartulina u otros recursos de uso habitual en la escuela en los que se pueda distinguir símbolos impresos con un fondo blanco.

Los estudiantes construyen programas en el LPT colocando una secuencia de tarjetas que representan las instrucciones del lenguaje.

Entorno Web

En el momento en que el programa físico está elaborado, se digitaliza usando una cámara fo-

tográfica convencional. La imagen capturada se procesa para producir un programa digital que luego se ejecuta. Finalmente, el resultado de la ejecución se mapea directamente sobre el mundo representado en el entorno web.

El desarrollo del entorno Web está dividido en tres etapas:

1. Digitalización del Programa Tangible (PT): es el proceso que permite la lectura del PT por medio de una cámara web o de un dispositivo móvil.
2. Procesamiento del Programa: es el proceso por el cual la versión digital del PT, esto es, una imagen, es representado en un lenguaje susceptible a ser ejecutado sobre el mundo.
3. Ejecución y Visualización de los Resultados del PT: realiza la ejecución del programa mostrando el comportamiento del agente dentro del mundo, representado en forma de grilla.

Mediante este desarrollo, se espera que la herramienta propuesta contribuya a simplificar y enriquecer los procesos de enseñanza y de aprendizaje de la programación, alcanzando una población no alfabetizada. Se busca evitar que los estudiantes interactúen con la sintaxis de un lenguaje de programación textual. Por otra parte, al ser una herramienta Web se eliminan los aspectos relacionados a la instalación y configuración del sistema, sin demandar recursos costosos ni específicos.

3. Resultados obtenidos y esperados

En una primer etapa se definió y diseñó un LPT simple, prestando especial atención a la utilización de recursos de bajo costo y de una sintaxis que resulte accesible y familiar a niños no alfabetizados.

Dicha sintaxis está compuesta de símbolos que son intuitivos para los alumnos y que no requieren saber leer ni escribir para poder ser

interpretados. Cada símbolo es un gráfico de color negro que se ubica en el centro de un cuadro con fondo blanco, cuyo límite se establece con un borde de color negro, como muestra la Figura 1. La simplicidad de la representación de los símbolos permite que puedan ser impresos, dibujados o contruidos sobre piezas físicas (papel, cartulina, cartón, etc).

Inicialmente, se definieron como símbolos flechas, con cuatro direcciones distintas, donde su semántica indica las acciones posibles a ejecutar por el bot, sobre un mundo representado en forma de cuadrícula. En este sentido las cuatro acciones iniciales definidas para el bot son: avanzar, retroceder, girar a la izquierda y girar a la derecha.

Un programa tangible (PT) desarrollado en este lenguaje se interpreta de izquierda a derecha con sus instrucciones, piezas físicas, ubicadas en forma secuencial. En la Figura 1 se puede observar un PT representado por diez piezas (o instrucciones). Cuando ese PT sea ejecutado en el entorno Web, los movimientos del bot serán mapeados en el mundo como avanzar tres casilleros, girar a la derecha, avanzar dos casilleros, luego girar a la izquierda, retroceder un casillero, girar a izquierda y finalmente, avanzar otro casillero.

Una vez definido el LPT, se diseñó la arquitectura del entorno Web basada en el Modelo Vista Controlador (MVC), que permitió clasificar sus componentes y definir las relaciones entre ellos. Los componentes de esta arquitectura son:

- *Modelo*: contiene una representación de los lenguajes utilizados:
 - Lenguaje de Programación Tangible (LPT): está compuesto de símbolos impresos en piezas físicas, cuya semántica se corresponden con las acciones para el bot a manipular.
 - Lenguaje Intermedio (LI): es un lenguaje que, como su nombre lo indica hace de intermediario entre el LPT y el mundo. El objetivo de utilizar este tipo de lenguaje es desacoplar el LPT

de la forma en que se representa el mundo (interfaz de usuario).

Para este desarrollo se optó por representar el LI con el formato de texto JSON[3] (JavaScript Object Notation), ya que es una notación simple, muy utilizada para el intercambio de datos.

- *Vista (o interfaz de usuario)*: tiene dos funciones. La primera consiste en obtener la imagen del programa definido por el usuario y enviarla al controlador. La segunda, consiste en mostrar la ejecución de las acciones del bot en el mundo.
- *Controlador*: contiene tres funcionalidades
 - Recibe la imagen de la vista para analizar su contenido y, en base a éste, genera el PT.
 - Una vez generado el PT, lo traduce al LI.
 - Finalmente, interpreta el LI para poder ejecutar sus instrucciones en la interfaz de usuario.

Considerando las etapas desarrollo descritas en la sección 2, se comenzó con la etapa de digitalización del PT, que tiene como objetivo la implementación de una aplicación que permita leer las tarjetas (o instrucciones) que lo componen. Esto se logró utilizando una cámara convencional y herramientas de Realidad Aumentada implementadas con A-Frame y AR.js [1, 2]. A-Frame es un framework Web para el desarrollo de ambientes de Realidad Virtual y AR.js es una biblioteca JavaScript para Realidad Aumentada, que posibilita el seguimiento de marcadores.

Actualmente, se encuentra en desarrollo la segunda etapa, Procesamiento del Programa, donde se interpreta el contenido del PT para luego traducirlo a un programa en LI. Asimismo, se están evaluando las tecnologías que se utilizarán en la interfaz de usuario, para mostrar el mundo y los movimientos que el agente realice en base a la acciones obtenidas del LI.



Figura 1: Ejemplo de un Programa en LPT

Por último, se desea que el trabajo realizado pueda ser tomado como base para futuros trabajos de investigación, es decir que sea susceptible a ser extendido o modificado para implementar nuevas funcionalidades, sin la necesidad de realizar cambios complejos sobre la arquitectura general del sistema.

4. Formación de Recursos Humanos

Uno de los autores de este trabajo se encuentra desarrollando su tesis de grado de la Licenciatura en Ciencias de la Computación, en la temática de esta línea de investigación.

Por otra parte, otro de los autores de este trabajo cursa la Maestría en Enseñanza en Escenarios Digitales que desarrollan de manera conjunta las Universidades Nacionales de Cuyo, Comahue, Patagonia Austral, Patagonia San Juan Bosco, San Luis, Chilecito y La Pampa.

Referencias

- [1] A-FRAME Homepage <https://aframe.io/>. Accedido por última vez Marzo 2020.
- [2] AR-JS Homepage <https://github.com/AR-js-org/AR.js>. Accedido por última vez Marzo 2020.
- [3] JSON Homepage <https://www.json.org/json-es.html>. Accedido por última vez Marzo 2020.
- [4] A. N. Antle and A. Wise. Getting down to details: Using learning theory to inform tangibles research and design for children. *Interacting with Computers*, 25(1):1–20, 2013.
- [5] C. P. de Educación de la Provincia de Neuquén. Diseño Curricular Jurisdiccional de los tres primeros años de la Escuela Secundaria Neuquina. Resolución N°1463/18, 2018.
- [6] S. Furber. *Shut down or restart? The way forward for computing in UK schools*. The Royal Society Education Section, 2012.
- [7] S. Papavlasopoulou, M. N. Giannakos, and L. Jaccheri. Reviewing the affordances of tangible programming languages: Implications for design and practice. In *2017 IEEE Global Engineering Education Conference (EDUCON)*, pages 1811–1816. IEEE, 2017.
- [8] J. M. Ruzafa. *Una arquitectura para aplicaciones educativas basadas en mundos virtuales e interfaces tangibles*. PhD thesis, Universidad Autónoma de Madrid, 2018.
- [9] T. Sapounidis, I. Stamelos, and S. Demetriadis. Tangible user interfaces for programming and education: A new field for innovation and entrepreneurship’, innovation and entrepreneurship in education (advances in digital education and lifelong learning, volume 2), 2016.
- [10] R. Society. After the reboot: Computing education in UK schools. *Policy Report*, 2017.
- [11] J. Wing. Research notebook: Computational thinking—what and why. *The Link Magazine*, 6, 2011.
- [12] J. M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.