

Enseñando Programación a Niños en Situación Vulnerable

Teaching Programming to Children in Vulnerable Situations

CC.Viviana Harari
Universidad Nacional de La Plata
Facultad de Informática, Laboratorio LINTI
La Plata, Argentina
vharari@info.unlp.edu.ar

Esp. Ivana Harari
Universidad Nacional de La Plata
Facultad de Informática, Laboratorio LINTI
La Plata, Argentina
iharari@info.unlp.edu.ar

Resumen— Este artículo relata la experiencia de un proceso de formación sobre Enseñanza de Programación a niños y adolescentes en situación de vulnerabilidad, quienes acuden a comedores comunitarios de barrios carenciados de las periferias de la ciudad de La Plata. En estas organizaciones se les brinda, además del desayuno y almuerzo, asistencia escolar en diferentes asignaturas. Desde el año 2007, allí se integraron talleres de Computación, organizados por docentes y estudiantes avanzados de la Facultad de Informática de la Universidad Nacional de La Plata UNLP [1], que participan en el proyecto de extensión y voluntariado denominado "Informática Inclusiva". Este proyecto que apunta fundamentalmente a que mediante la enseñanza de Informática se estreche vínculos entre la Universidad y los barrios marginales, fue aceptado tanto en las convocatorias a proyectos de extensión de la universidad como en las de voluntariado del Ministerio de Educación de la Nación Argentina [2]. En este marco, durante el año 2016 se dio inicio a la enseñanza de programación donde se capacitaron a 30 niños de las fundaciones Centro Verde Esperanza y Los Robles [3]. En este artículo, se va a abordar los detalles del proceso educativo, la metodología empleada, los recursos pedagógicos utilizados y los resultados obtenidos. Resultados que, más allá de la falta de recursos, de la disparidad de formación previa de los alumnos y del contexto desfavorable en el que viven, fueron altamente satisfactorios, con un nivel de participación y de interés muy gratificantes. Fue una experiencia muy enriquecedora que vale la pena ser contada.

Abstract—This paper relates the experience of teaching programming to children in vulnerable situations who attend non-profit neighborhood entities in the outskirts of the city of La Plata. In these organizations, children not only receive breakfast and lunch, but also school support. Since 2007, computer basics training courses were included organized by teachers and volunteer students of the Computer Science School of the National University of La Plata, UNLP [1], who belong to the program "Inclusive Computing". This program has been approved over the years, in the successive calls for

Extension Projects and Volunteering Projects awarded by the university and The Ministry of Education of the Nation [2] respectively. During the 2016 school year, programming training was introduced for 30 children belonging to the Centro Verde Esperanza and Los Robles Foundation [3]. The paper will include details of this educational process, the methodology employed, the pedagogical resources used, and the results obtained. Results that in spite of the lack of resources, the difference of children previous training and the unfavorable context where they live, it resulted in great satisfaction, with a high participation and interest level. This was an experience that was very rewarding and it worth telling.

Palabras claves—Alfabetización informática, lenguaje de programación de computadoras, software libre, internet.

Keywords—Computer literacy, computer programming language, open source, internet.

I. INTRODUCCIÓN

La Facultad de Informática de la UNLP, fue creada en el año 1999 y, desde sus inicios vienen desarrollando líneas de acción en las que se afianzan los tres grandes ejes: enseñanza, investigación y extensión. Sobre estos pilares asume un fuerte compromiso con la comunidad a través de significativos aportes científicos y tecnológicos [4].

Desde hace 10 años, la Facultad a través del programa de Informática Inclusiva viene llevado a cabo diferentes proyectos relacionados con el objetivo de acortar la brecha digital en sectores carenciados de la ciudad de La Plata y alrededores [5]. El trabajo que viene realizando es continuo y, no ha sido interrumpido en ningún momento y, por ningún factor.

Actualmente se está trabajando con 10 asociaciones civiles sin fines de lucro situadas en barrios marginales de la ciudad de La Plata y alrededores. Los grupos destinatarios, en su mayoría, son niños de edad escolar primaria, que asisten a escuelas públicas cercanas a sus barrios y, no cuentan con gabinete informático. Esto último habla del perfil de alumnos

con los que se trabaja: alumnos con contacto nulo o escaso de recursos educativos tecnológicos.

A partir de la intervención de la Facultad de Informática en estas asociaciones, permitió que estos niños y niñas pudieran contar con la posibilidad de tener computación, por lo menos, una hora semanal. Esto llevó a que a lo largo de los años pudieran aprender el uso básico de la herramienta "computadora", aplicarla para afianzar conceptos de otras disciplinas, como también utilizar diferentes aplicaciones de ofimática y de acceso a Internet.

En este mismo sentido, se integró la enseñanza de programación, cuya primera experiencia realizada sobre dos asociaciones civiles, se va a explicar en este artículo. Se darán detalles sobre las características de los grupos, la metodología y estrategias utilizadas, las producciones de los alumnos y, su alto nivel de aceptación de la incorporación de esta nueva temática.

II. ASOCIACIONES CIVILES CON LAS QUE SE TRABAJÓ

Centro Educativo y Asistencial Verde Esperanza de la Fundación Tau [6] es una asociación civil sin fines de lucro que tiene como objetivo la realización de acciones de servicio destinada a la asistencia social, médica, educativa y caritativa. Funciona en un barrio marginal de la ciudad de La Plata, desde abril del año 2004. Concurren diariamente cerca de 70 alumnos, en su mayoría provenientes de escuelas públicas barriales, a los que se les brinda apoyo escolar y, se procura satisfacer sus necesidades de útiles y vestimenta. En el Centro funciona un gabinete de computación y un consultorio de Odontología.

El Roble [7], centro de día para niños y adolescentes en situación de vulnerabilidad, funciona desde hace más de 15 años en la ciudad de La Plata. A través de la asociación se implementan diferentes programas y proyectos de asistencia para una gran cantidad de niños y sus familias. Con la primera institución, se viene trabajando hace seis años y con la otra entidad, se trabajó dos. En esta experiencia educativa, participaron 20 alumnos de Centro Verde Esperanza y 10 alumnos de El Roble.

III. PERFIL DE LOS ALUMNOS CON LOS QUE SE TRABAJÓ

Los alumnos, quienes comenzaron con los primeros pasos en programación, asisten diariamente a las asociaciones civiles Centro Verde Esperanza y El Roble. Sus edades varían de los 10 a 13 años. Asisten a escuelas públicas de la región y, en ellas no se les enseña informática. El conocimiento sobre el uso de las tecnologías lo han adquirido en las asociaciones civiles a las cuales asisten diariamente, a través de cursos que se implementan allí, dados por la Facultad de Informática a través del programa de Informática Inclusiva. En general son grupos heterogéneos respecto a edades, capacidades, intereses y comportamientos.

IV. METODOLOGÍA DE ENSEÑANZA APLICADA

El proceso de enseñanza de programación se inició a principios del año 2016, se llevó a cabo durante todo el ciclo lectivo y continúa en vigencia actualmente. Con solo una carga

semanal de una hora y agrupados de a diez niños, se intentó trabajar con una planificación que permitiera, sin perder el eje principal, abordar instancias particulares de aprendizaje, y con actividades que cierren en cada clase. Esto permitió una mejor asimilación de los conceptos en casos donde las clases se vieron interrumpidas por falta de luz, problemas de acceso en días de lluvias o por la discontinuidad de algunos niños que no concurrían por cuestiones de salud.

En las siguientes secciones se describen las etapas involucradas en la planificación, donde se utilizaron diferentes estrategias de enseñanza, como la simulación, la experimentación de ser un robot, el uso de juegos educativos para programar, el uso de lenguajes de programación por encastre, entre otros.

Las actividades estaban organizadas de menor a mayor complejidad, en las cuales se ponía en práctica los conceptos abordados, y se les daba la posibilidad de una revisión, reflexión sobre si el algoritmo hallado era la mejor solución posible y su comparación con otras alternativas.

A. Aprendiendo nuevos conceptos

Para comenzar a hablar sobre la programación e introducir, en los niños y niñas, conceptos tales como: "programar", "programa", "programador" y "lenguaje de programación", se les mostró, a los alumnos, una serie de diferentes videos relacionados con robots y sus diferentes aplicaciones, desde su uso en la industria, hasta su uso en el entretenimiento. Como ejemplo de su aplicación en la industria se mostraron videos de robots construyendo autos, placas de electrodomésticos, entre otros. Como ejemplos del uso de robots en investigación, se mostraron videos de robots enviados al espacio para tomar fotos o recoger material del lugar visitado y, de su uso en entretenimiento, analizaron robots para jugar al fútbol. En la Fig.1 se muestran los alumnos analizando los videos.



Fig. 1. Alumnos trabajando con los videos sobre aplicaciones de robots.

La muestra de los videos produjo la reflexión sobre por qué los robots realizaban las tareas y, quién o quienes le había enseñado a hacerlas. Esas preguntas disparadoras produjeron que se llegara a nombrar y, definir los diferentes conceptos que se tenían como objetivos de la clase.

Se habló de que los robots funcionaban porque tenían "programas" que le decían lo que tenían que hacer, que esos programas lo habían hecho personas entendidas sobre el tema llamadas "programadores" y, que la forma que le habían dado

las órdenes no era con su lenguaje coloquial sino con un lenguaje especial, llamado "lenguaje de programación". Esto fue registrado en el Word donde anotaron las distintas definiciones.

B. Programándonos

Una vez familiarizados con los primeros conceptos relacionados a la programación, se planteó un juego en el que un integrante del grupo actuase de "robot" y, el resto de "programadores". Para eso se les presentó un lenguaje de programación con las sentencias que iba a entender el robot "humano". A continuación se detallan las sentencias que los alumnos debían utilizar para programar al robot.

- **Comenzar:** le indica al robot que se prenda. Con esta acción el robot abrirá los ojos y, estará dispuesto a escuchar otras órdenes.
- **Girar Derecha (GD) :** le indica al robot que gire a derecha 90°.
- **Girar Izquierda (GI):** le indica al robot que gire a izquierda 90°.
- **Avanzar pie izquierdo (API):** le indica al robot que realice un paso con el pie izquierdo
- **Avanzar pie derecho (APD):** le indica al robot que realice una paso con el pie derecho
- **Fin:** Con esta acción el robot cerrará los ojos y, estará ya no escuchará más órdenes.

En función del lenguaje presentado, se plantearon diferentes desafíos, partiendo del hecho de que el robot se encontraría situado en la puerta del aula. Los alumnos, haciendo uso de un editor de textos escribían las órdenes y, luego el alumno o la docente que se habían prestado para hacer el personaje del robot, acataba las órdenes de cada alumno y, se corroboraba quién o quiénes habían sido los mejores programadores. Se consideraba de esa manera a aquel que a través de su programa había logrado el objetivo planteado. A continuación, se puede observar la Fig.2, en la cual se observan los alumnos de Centro Verde Esperanza, simulando ser robots y ejecutando órdenes.



Fig. 2. Alumnos simulando ser robots y ejecutando órdenes.

Una vez realizados varios programas relacionados con diferentes desafíos planteados, se continuó avanzando con la enseñanza de la programación. Se tomó como base los

lineamientos encontrados en el sitio Program.ar [8] y, se le realizaron algunas variaciones, adaptándolas al contexto donde se desarrolló la enseñanza de la programación. Program.ar es un sitio que pertenece a una iniciativa gubernamental que tiene como objetivo llevar a todas las escuelas argentinas el aprendizaje significativo de las Ciencias de la Computación. La iniciativa fue creada en el año 2013 y, es llevada adelante por la Fundación Sadosky del Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación [9].

Se les presentó a los alumnos otro robot (parecido al personaje Wall-e de una película infantil de la compañía Walt Disney), que entendía órdenes para dibujar sobre una hoja cuadrículada. Se les planteó que este nuevo robot comprendía "otras órdenes" diferentes al robot "humano". Esto llevaba a estudiar y aprender otro "lenguaje de programación" para poder programarlo. La presentación del nuevo robot con sus conjunto de sentencias, permitió reflexionar sobre la existencia de diferentes "lenguajes de programación", por lo que los niños comprendieron que no existe un único lenguaje para programar.

Sobre el nuevo robot, se les explicó que siempre iba a encontrarse situado en el extremo inferior izquierdo de la hoja cuadrículada y, que estaba siempre con el lápiz levantado, como se muestra Fig.3.

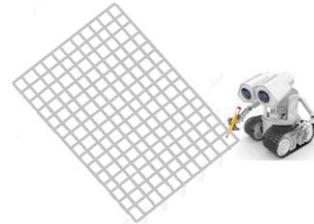


Fig. 3. El robot a programar en la grilla.

Se les comentó que este robot conocía las siguientes sentencias:

- **Comenzar:** le indica al robot que se prenda. Con esta acción el robot está preparado a escuchar otras órdenes.
- **Bajar Lápiz (BL):** le indica al robot que apoye el lápiz sobre la hoja.
- **Subir Lápiz (SL):** le indica al robot que levante el lápiz de la hoja.
- **Rellenar casillero (RC):** le indica al robot que rellene con color el casillero donde tiene apoyado el lápiz.
- **Avanzar casillero arriba (ACAR):** le indica al robot que avance un casillero hacia arriba.
- **Avanzar casillero abajo (ACAB):** le indica al robot que avance un casillero hacia atrás.
- **Avanzar casillero izquierda (ACI):** le indica al robot que avance un casillero a la izquierda.
- **Avanzar casillero arriba (ACD):** le indica al robot que avance un casillero hacia la derecha.
- **Fin:** Con esta acción el robot cerrará los ojos y, estará ya no escuchará más órdenes.

En base a esas sentencias los alumnos realizaron diferentes programas tales como: realizar cuadrados, triángulos y círculos

de diferentes tamaños. Se les daba la consigna y papelitos cuadriculados con lápices, para que pudieran probar sus programas, antes de comunicar que habían finalizado los mismos. Los programas, al igual que en el caso anterior, eran realizados haciendo uso de un editor de textos.

Como forma de incentivar a los alumnos se planteó una competencia. Cada alumno que declaraba haber terminado su programa, se probaba y, si el robotito (representado por la/el docente) lograba realizar la figura estipulada, el alumno obtenía un punto.

Este juego adicional de competencia los incentivó aún más. No solo obtenía dos puntos el que lo había terminado bien y primero, sino obtenían un puntos todos los que habían conseguido realizar el programa en forma correcta.

A continuación, en la Fig.4 se puede observar a los alumnos trabajando sobre la programación del nuevo robot.



Fig. 4. Alumnos programando el robot.

Tanto la experiencia de "jugar" a hacer de "robot humanos" y programarse, como la de programar al robot que sabía dibujar, resultó muy exitosa, porque los alumnos trabajaron muy concentrados en la realización del programa y no querían finalizar con el horario de computación.

C. Programando robots online

Una vez realizados varios desafíos sobre la programación de robots, se les comentó a los alumnos que se iba a comenzar a programar otros robots a través de la computadora y, que en muchos casos íbamos a necesitar contar con conexión a Internet, dado que algunos estaban disponibles en forma online. Este recurso necesario, no siempre se encontraba disponible y se pensaron en diferentes estrategias para continuar con el aprendizaje. Recordar que las clases se dan en los barrios donde se encuentran las asociaciones civiles y que muchas de ellas tienen interrumpido el acceso a Internet o la conexión es muy lenta.

Como primer recurso, se utilizó en su versión online, la aplicación Ligthbot desarrollado por Danny Yaroslavskiu en el año 2013 y, que es un juego que sirve para aprender a programar a personas de todas las edades [10]. Se les comentó que este nuevo robot entendía órdenes de tipo icónico

(imágenes), a diferencia de los anteriores que entendían órdenes de tipo textual. y que, tanto para escribir el programa como para comprobar si funcionaba bien, se iba a utilizar la computadora. Se les presentó a Lightbot, las órdenes que entendía y qué funciones realizaba cada una de ellas. Su ambiente de programación se visualiza en la Fig.5.

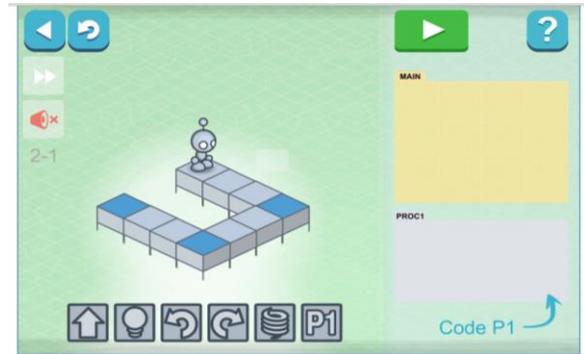


Fig. 5. Ambiente de Ligthbot extraída de <https://lightbo.com/hocflash.html>

Una vez explicada sus componentes sintácticas como semántica, se comenzó con las actividades planteadas para programar allí. Se utilizó solo el nivel "Básico", nivel que permite escribir secuencias y, no introduce otro tipo de estructura de control, ni concepto de procedimiento o función.

El programar en Ligthbot hizo que los alumnos se entusiasmaran y continuaran trabajando muy concentrados y motivados.

Lo destacable de la experiencia del trabajo con Lightbot es que muchos de los alumnos que finalizaron los desafíos planteados en el nivel Básico, comenzaron, por su cuenta, con el siguiente nivel que involucraba uso de "procedimientos".

Lo que se pudo observar es que, como en cada nivel, es acotado el espacio de escritura del programa, se fuerza a que el alumno cree una "sentencia nueva" llamada P1, que involucre otras sentencias.

El objetivo planteado por la herramienta, es que el aprendiz observe si hay algún patrón que se repite en el recorrido que Lightbot debe hacer. En el ejemplo dado el patrón que se repetiría sería:



Los alumnos para poder cumplir con el objetivo del desafío, y por las limitaciones del espacio para colocar sentencias, empezaron a agruparlas formando procedimientos. Esta forma de descubrir el concepto de generar procedimientos permitió comenzar a abordar dicho tema, para que lo encaren de una manera pensada modularmente.

Respecto a estas soluciones halladas por los alumnos en forma de prueba y error o por descubrimiento incentivados por la necesidad que el programa funcione, provocó la intervención del docente quien fue aclarando que, más allá que una solución algorítmica funcione bien, será necesario un diseño de solución bien estructurado y eficiente.

Una vez finalizada la experiencia con Lightbot se continuó con el otro recurso de programación de robots en línea,

llamado Baymax , perteneciente a un proyecto de la empresa Walt Disney, aplicación para aprender a programar a niños, basado en la película "Seis Grandes Héroe" [11]. Su entorno se visualiza en la Fig.6.



Fig. 6. Ambiente de Baymax

Para poder realizar los desafíos planteados en este nuevo proyecto, se les habló de otro robot y de las sentencias (lenguaje de programación), que entendía. Se les dijo que si bien, también entendía sentencias icónicas, no eran las de Lightbot.

Esto sirvió para que se repasasen los conceptos vistos de: "programar", "programador", y "lenguajes de programación", realizando una tarea en donde los alumnos debían completar las definiciones y plantear las similitudes y diferencias entre los lenguajes utilizados tanto para programar a Lightbot como a Byamax. En la Fig.7, se pueden observar imágenes relacionadas con los niños trabajando con esta aplicación.



Fig. 7. Niños utilizando Byamax.

Como se comentó al comienzo del presente apartado, en muchas ocasiones, la conectividad a Internet del lugar donde se dictaban las clases, era muy deficiente. Eso provocaba que no se pudieran cargar las páginas con las aplicaciones correspondientes en todas las máquinas o, en ninguna. Por ese motivo, se tuvo que pensar en una estrategia didáctica que permitiera poder trabajar con todos los alumnos. La estrategia fue la de armar documentos de un editor de textos, con una serie de ejercicios con las actividades planeadas por la aplicación, pedir a los alumnos que realicen las actividades en el documento y, luego la docente, valiéndose de su celular, tabletas o computadora personal que tuviera acceso, prestaba el recurso al alumno para que pueda comprobar la correctitud de

su solución. De esa manera, todos los alumnos trabajaban al mismo tiempo y, de la misma manera.

A continuación, en la Fig.8, se muestra uno de ejercicios planteados en un documento del Word. Los alumnos resolvían el desafío, copiando y pegando los íconos que representan sentencias, escribiendo el programa aunque no haya Internet.

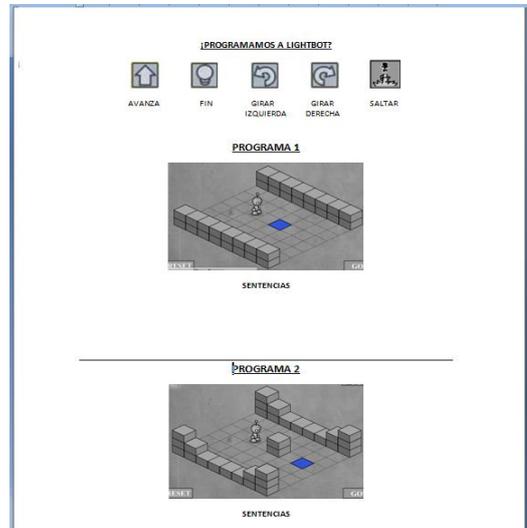


Fig. 8. Trabajo en papel y en Word por falta de Internet.

D. Programando con técnicas de encastre

Con la mirada puesta en el objetivo final, que era que los alumnos finalizaran escribiendo programas utilizando la herramienta Scratch [9], se los introdujo al concepto de "encastre", dado que con esta herramienta, las sentencias son como fichas de encastre y, los alumnos realizar sus programas "encastrando" las órdenes.

Para introducirlos al mecanismo de encastre se les pidió realicen alguno de los desafíos planteados por "Blockly Games [13]. Como está diseñado para finalizar todo el recorrido del aprendizaje de la programación con el uso de lenguajes convencionales basados en texto, se utilizaron solo algunos desafíos. Los desafíos utilizados fueron "Rompecabezas" y "Laberinto". Los mismos permitieron que los alumnos comenzaran con la técnica de encastre básica, como se muestra en la siguiente figura.

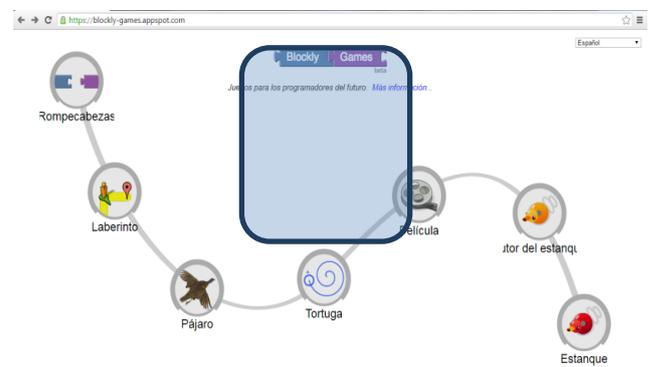


Fig. 9. Trabajo con Blockly Games.

Continuando con la lógica del encastre, se propuso a los alumnos continuar con una programación similar y, se les planteo seguir programando desafíos pero los propuestos por otro programa, denominado "Code.org" [14]. Este sitio propone una serie de cursos para enseñanza de programación tanto a niños como a adolescentes de nivel secundario.

De los desafíos planteados por Code.org, en esta etapa se pensó en utilizar solamente los que planteaban solo proceso "secuenciales", sin introducción de estructuras de control. Pero como ocurrió con la experiencia en Lightbot, los alumnos avanzaron un poco más y, comenzaron con actividades relacionadas con la introducción a la sentencia de repetición.

De la misma manera que el caso de Lightbot, se retomó esas actividades, pero en este caso se le puso más énfasis en la explicación de la estructura de control de repetición, previendo que cuando se enseñe el uso de Scratch, sería una sentencia que utilizarían.

En la Fig.10, se pueden observar ejercicios de Code.org e imágenes de alumnos trabajando con dicha herramienta.

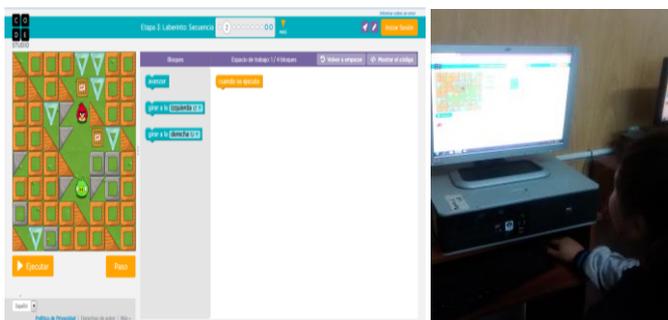


Fig. 10. Programando con Code.org.

Se utilizaron otros recursos que sirven para la enseñanza de la programación como lo son "Pilas Bloques" también propuestos por la iniciativa Program.ar. Pilas Bloques tiene la ventaja que puede descargarse la aplicación, con lo cual permite trabajar en lugares con poca conectividad. De este último, se utilizó solo los desafíos "El alien toca el botón", "El gato en la calle" y "No me canso de saltar" [15]. El último desafío requería el uso de repetición, por lo que se les recomendó que pensarán una solución estructurada. Este desafío planteaba al programador, realizar un programa en la que un gato tenga que saltar 30 veces. El mismo contaba con la orden Saltar, cuya imagen se muestra a continuación:



Se reflexionó sobre las ventajas y desventajas de copiar 30 veces una misma sentencia, o de estructurarla mediante una sentencia de control de repetición. También se les recomendó pensar en la posibilidad que el algoritmo funcione en condiciones más extremas y que implique mínimas modificaciones. Por ello, se les pidió que modifiquen el programa para que el gato saltara 200 veces. A través de incentivar el uso de estructuras de control de repetición cuando la solución al desafío lo amerita, se volvió a ejercicios realizados en otros lenguajes de programación para que los modifiquen. Por ejemplo, se les pidió programar al robot de la

grilla, haciéndolo que camine varios pasos, cuando un paso implicaba las sentencias Avanzo Pie Derecho (APD) y, Avanzo Pie Izquierdo (API) y también, realizar desafíos de code.org sobre repeticiones.

A continuación, se muestra una imagen de una de las actividades que realizaron los alumnos para introducirlos a la sentencia de repetición (Fig.11).

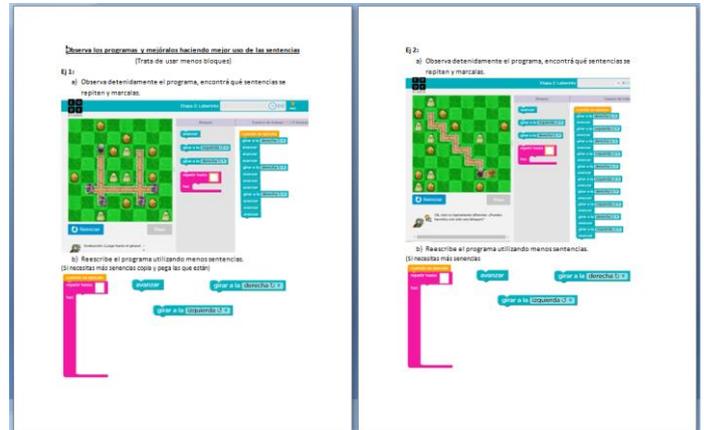


Fig. 11. Actividad sobre procesos repetitivos.

La actividad proponía: observar los programas realizados, analizar qué sentencias veían repetirse, marcarlas y, luego colocarlas en la sentencia "repetir" que aparecía debajo, indicando la cantidad de veces que esas sentencias se repetían.

E. Comenzando a trabajar con Scratch

Con los alumnos ya familiarizados con la técnica de encastre y, con la explicación de la estructura de control de repetición, se presentó a los alumnos la aplicación "Scratch", software con el que se iba a trabajar a partir de esos momentos y, que se utilizaría para desarrollar software creado por ellos.

Se les comentó que la diferencia básica de este software con el resto, es que en los anteriores, se les presentaba desafíos ya definidos que ellos debían cumplir. En este caso, ellos no iban a tener que programar a un robot o personaje, para que cumpla con ciertas consignas ya establecidas por la aplicación, sino que iban a ser libres de elegir los personajes y, programarlos de acuerdo a sus propios objetivos o, a los que los docentes les dieran. Iban a crear sus propios software, con personajes, fondos y acciones que ellos desearan.

Scratch, es una aplicación creada por Grupo Lifelong Kindergarten del MIT Media Lab [12], liderado por Mitchel Resnick. Scratch presenta un lenguaje introductorio para programar, la primer versión salió en el año 2003. Es una herramienta muy potente dado que, por su relativa facilidad de aprendizaje, permite desarrollar programas interesantes, brindando a su vez habilidades, que pueden ser de gran utilidad en el futuro si, el alumno, quiere indagar en lenguajes de programación básicos [16].

A parte de lo expresado en los párrafos anteriores, respecto a las cualidades de la herramienta, se adiciona otro beneficio: la posibilidad de descargarlo e instalarlo en las máquinas locales. Esto permitió no tener que contar con Internet como

recurso y, no depender de su conexión. Como se dijo anteriormente, en los lugares donde se da la capacitaciones, la conexión es muy débil y, a veces nula.

La enseñanza de la herramienta comenzó con la presentación, a los alumnos, de su interfaz del usuario (Fig.11). Se habló sobre sus diferentes secciones: lugar dónde se colocan las sentencias, lugar dónde se ve la ejecución del programa, lugar dónde se pueden elegir los personajes y los fondos, lugar donde se encuentran las sentencias, entre otras cosas.

En la Fig.12, se muestra la interfaz de Scratch.

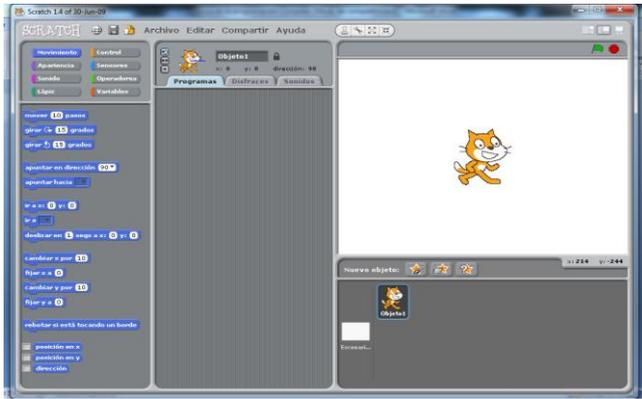


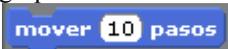
Fig. 12. Ambiente de desarrollo de Scratch.

Se planteó que las sentencias se presentaban agrupadas en el sector superior izquierdo y que las que ellos iban a utilizar, en un principio, eran las siguientes:

Del grupo "Control":

- : Sentencia con la que iban a comenzar todo programa. Con esa se indica que el programa se comienza a ejecutar cuando se cliquee sobre la bandera verde y,
- : Esta sentencia iba a servir para encerrar el conjunto de sentencias a repetir, en caso que lo necesitaran.

Del grupo "Movimiento":

- : Sentencia con la que iban a poder hacer que el personaje avance tantos pasos como le indicasen.

Del grupo "Apariencia":

- : Sentencias con las que iban a poder crear diálogos entre los personajes.

El objetivo de acotar el conjunto de sentencias era para que puedan manejarse con tipos de sentencias parecidas a las que venían utilizando hasta el momento. La sentencia "mover",

equivalía a la sentencia "avanzar" utilizada en el software anterior y, la sentencia repetir también tenía un formato similar al visto. Las últimas se agregaron para que puedan hacer diálogo a los personajes.

La importante, en este primer contacto, era lograr la familiarización con la nueva herramienta.

Lo primero que se les pidió fue que realizaran un programa donde hubiese dos personajes que se encuentran y se saludaran. Los alumnos contaron con la libertad de elegir a los personajes, que podían ser los que les brindaba la aplicación o, podían buscar imágenes a través de la Web. Para ambas alternativas se les explicó la forma de hacerlo. A continuación, en la Fig.13, se pueden observar algunas de las primeras producciones realizadas por los alumnos.

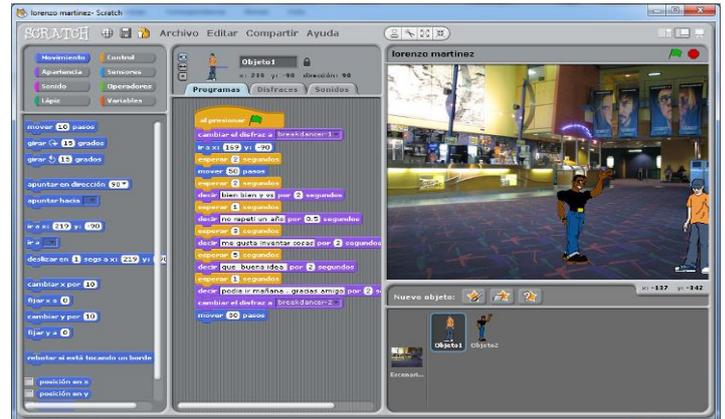


Fig. 13. Producciones de los alumnos con Scratch.

Una vez afianzado con la nueva herramienta se procedió a avanzar en diferentes aspectos, como por ejemplo cómo mejorar el entorno donde se desarrollaba el encuentro de los amigos que habían realizado. En cuyo caso se incorporó el concepto de "Escenario", mostrándoles cómo elegir un escenario para el programa que habían desarrollado.

También, se reflexionó sobre cómo manejar mejor el tiempo de los diálogos de los personajes. Hacer que un personaje hablara recién después que el otro terminara, permitiendo brindar una sensación de diálogo entre ambos. Y, sobre el tema de las ubicaciones como el trabajar posiciones iniciales de los personajes. La ejecución de un programa podía implicar desplazamientos de sus personajes. Esto provocaba que, al iniciar nuevamente el programa, esos personajes debían ser reubicados en forma manual, por los alumnos. Para dar una solución a eso, se les habló de la ubicación de los personajes dentro del escenario. Se los introdujo al sistema de coordenadas soportado por Scratch y, se les marcó, dentro de la interfaz de la aplicación, dónde se encontraba esa información.

Tanto para la sincronización del diálogo como para el manejo posicional de los personajes, se les enseñó las siguientes dos sentencias. Una perteneciente al grupo "Control" y la otra de "Movimiento":

- : Esta sentencia permite indicar la cantidad o fracción de segundos a esperar y, puede

ser aplicada a cualquier acción del personaje. Esto, aplicado en un diálogo permite simular las esperas entre lo que dice un interlocutor y otro.



- Esta sentencia permite colocar al personaje en un lugar determinado del escenario.

Con estas nuevas incorporaciones los alumnos mejoraron su software y siguieron realizando otras simulaciones de encuentro de personajes.

Para potenciar aún más las simulaciones de encuentros de personajes, se incorporó el concepto de "cambio de disfraz". Para eso, se les habló de la posibilidad de que un personaje pudiera tener asociado varios "disfraces" y, que los mismos, pudieran ponérselos, a lo largo de la ejecución de la aplicación. Esos disfraces podían representar estados del personaje, como el estar sentado o el estar parado.

Actualmente, se está trabajando con los alumnos en perfeccionar las aplicaciones realizadas, a través de desafíos más complejos aplicando todo lo visto hasta ahora: sentencia repetir, mover, esperar, cambio de disfraz, entre otras cosas.

V. RESULTADOS OBTENIDOS

Los resultados obtenidos en este proceso de alfabetización en el campo de la programación fueron muy satisfactorios. Se logró trabajar con varias herramientas, recursos educativos y lenguajes de programación, que permitieron introducir a los niños los fundamentos de programación en forma amena, entretenida y motivadora. Recursos recomendados en su mayoría por el Consejo Federal a través del Programa Nacional de Inclusión Digital Educativa y Program.ar [17].

Se puede decir a través del grado de realización de las actividades estipuladas, que:

- un 100% de los alumnos, realizó todas las tareas de programar al robot humano.
- el 85% realizó los desafíos de programar al robot que dibujaba en hoja cuadriculada.
- el 100% realizó las tareas de programar a lightbot, baymax y code.org.
- el 35% avanzó en tareas más complejas por sí solos, descubriendo los conceptos antes de su abordaje.
- el 83% realizó todas las tareas sobre Scratch, del cual un 60% realizó las tareas satisfactoriamente con ayuda adicional.

Se piensa continuar avanzando en la enseñanza de la programación incorporando nuevos conceptos acordes a las edades de los alumnos con los que se trabaja como ser el concepto de variable, y trasladar esta experiencia al resto de los comedores comunitarios y organizaciones civiles sin fines de lucro con las que se trabaja.

VI. CONCLUSIONES

La experiencia de la enseñanza de programación a niños en situación de desventaja, fue muy gratificante. El grado de interés y de participación superó las expectativas. Siendo grupos heterogéneos respecto a la edad, a sus problemáticas, al nivel educativo que poseían, y en un contexto muchas veces desfavorable para un aprendizaje sostenido, se notó en ellos

ansias por aprender. Un interés progresivo que se manifestó en cuestiones como quedarse más tiempo para terminar con el desafío asignado, o adelantarse en la temática para poder lograrlo.

La enseñanza de programación permite adquirir habilidades mentales y cognitivas avanzadas, de pensamiento, raciocinio y creatividad. Posibilita a través del afán de resolver desafíos, aplicar estrategias de razonamiento, como también incursionar en el pensamiento lógico de las posibles soluciones, que pueden ser comprobadas. Y esto se logró más allá de los inconvenientes del propio contexto donde se llevó a cabo la enseñanza, de las micro decisiones y adaptaciones realizadas frente a situaciones imprevistas y ajenas al objeto de aprendizaje. Incursionar en el fascinante mundo de la programación es un camino de ida, donde se comprobó que los niños y jóvenes de barrios marginales de la ciudad de La Plata, están dispuestos a transitar.

REFERENCIAS

- [1] Sitio oficial de la UNLP. www.unlp.edu.ar
- [2] Ministerio de la Secretaría de Políticas Universitarias. portales.educacion.gov.ar/spu/voluntariado-universitario
- [3] Luis Santarsiero (2013). "Comedores comunitarios en La Plata: organización social e intervención alimentaria en espacio barrial". Revista Pilquen. Sección Ciencias Sociales. Vol.16 Nro.1. ISSN 1851-3123. Viedma, Argentina. Junio de 2013.
- [4] Diaz, J.; Banchoff, C.; Harari, V.; Harari, I. y Ambrosi, V. (2012). "Articulating two social based informatics projects". ICIT, International Conference on Informatics Technologies. IEEE proceedings. ISSN 1553-9911. Saratov, Rusia.
- [5] Harari, V. y Harari, I. (2014). Trabajando la inclusión en sectores marginados de la sociedad. Memorias del VI Congreso Nacional de Extensión Universitaria, Rosario, Argentina.
- [6] Sitio oficial de la Fundación Verde Esperanza. <https://www.facebook.com/FundacionJoseTau/>
- [7] Sitio oficial de la Fundación El Roble. <https://www.facebook.com/elroble.asociacioncivil>
- [8] Sitio oficial de Program.ar. <http://program.ar>
- [9] Sitio oficial <http://www.fundacion sadosky.org.ar/>
- [10] Sitio oficial de Lightbot. <http://www.openfl.org/blog/2014/11/07/developer-spotlight-danny-yaroslavski/>
- [11] Sitio oficial de Baymax. <http://juegos.disneylatino.com/grandes-heroes-code-baymax>
- [12] Sitio oficial de Scratch. <https://scratch.mit.edu/>
- [13] <https://blockly-games.appspot.com/>
- [14] Sitio oficial de Code. <https://code.org/>
- [15] Sitio oficial de Pilas Bloques. <http://pilasbloques.program.ar/>
- [16] Marji, Majed (2014). Learn to Program with Scratch (en inglés). San Francisco, California: No Starch Press. pp. xvii, 1-9, 13-15. ISBN 9781593275433.
- [17] Consejo Federal de Educación. http://www.me.gov.ar/consejo/resoluciones/res15/263-15_01.pdf