

FACULTAD DE INFORMÁTICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Título: ProBots3D: Programando Robots en 3D.
Una herramienta libre para enseñar programación
a niños y jóvenes

Tesis de Magister en Tecnología Informática
Aplicada en Educación

Alumna: Claudia M. Banchoff Tzancoff

Director: Francisco Javier Díaz

Co-Directora: Alejandra Zangara

Noviembre 2017

ProBots3D: Programando Robots en 3D. Una herramienta libre para enseñar programación a niños y jóvenes.

Alumna: Claudia Mariana Banchoff Tzancoff

Director: Francisco Javier Díaz

Co-Directora: Alejandra Zangara

Noviembre 2017

e-mail: cbanchoff@info.unlp.edu.ar



Esta obra se distribuye bajo una licencia de Creative Commons [Atribución-CompartirDerivadasIgual 3.0 Unported](#). Es decir, se permite compartir, copiar, distribuir, ejecutar y comunicar públicamente, hacer obras derivadas e incluso usos comerciales de esta obra, siempre que se reconozca expresamente la autoría original y el trabajo derivado se distribuya bajo una licencia idéntica a ésta.

Agradecimientos.

A Leandro y Lucas, que contribuyeron enormemente en el desarrollo final de FI3D.

A Javier y Alejandra, por sus aportes y sugerencias en este trabajo.

A las autoridades, alumnos y docentes del Colegio Estudiantes de La Plata con quienes venimos programando en la escuela desde el ciclo lectivo 2016.

A las autoridades, alumnos y docentes de la Escuela 67 quienes me abrieron las puertas de la escuela y participaron activamente en las distintas actividades propuestas.

A mis compañeros del LINTI, quienes me impulsaron a continuar y finalizar esta tesis.

A Sofía y Fernando que me impulsaron a usar \LaTeX y han trabajado codo a codo en muchas de las actividades mencionadas en este trabajo.

A Vivi, con quien compartí alegrías y sufrimientos durante este desarrollo.

A mi familia: Leo, Luqui, mis viejos, mis sobrinas, mis hermanos y mis hermanas del corazón, que me apoyan incondicionalmente en todo lo que encaro.

Por último, a mis ángeles invisibles por estar siempre ahí para mí.

Índice general

1. Introducción	1
1.1. Fundamentación	1
1.2. Objetivos	2
1.3. Trabajos publicados	3
1.4. Proyectos de I+D y extensión	5
1.5. Tesinas de grado dirigidas	5
1.6. Estructura de este informe	6
1.7. Sobre la tesista	7
2. Programar en la escuela	8
2.1. Introducción	8
2.2. Aprender a programar	11
2.3. Enseñar a programar	12
2.4. Herramientas para enseñar a programar	14
2.4.1. Logo	16
2.4.2. Scratch	17
2.4.2.1. Snap!	20
2.4.2.2. Scratch Jr	22
2.4.2.3. Otras versiones de Scratch	23
2.4.2.4. Comparativa de herramientas basadas en Scratch	24
2.4.3. Alice	25
2.4.4. Lightbot	27
2.4.5. Juegos de Blockly	28
2.4.6. Aplicaciones de Code.org	31
2.4.7. Made with code	32
2.4.8. MIT APPInventor	33

2.4.9.	Gameblox	34
2.4.10.	pilas-engine	35
2.4.11.	Pilas Bloques	38
2.4.12.	CodeCombat y CodinGame	39
2.4.13.	PSeInt	41
2.4.14.	Otras herramientas destacadas	42
2.4.15.	Desarrollos de la Facultad de Informática de la UNLP	42
2.4.15.1.	MiRobot y Zedpy	43
2.4.15.2.	XRemoteBot	44
2.4.15.3.	Dropsy	44
2.4.15.4.	RITA	45
2.4.16.	Tabla comparativa de herramientas	47
2.5.	Iniciativas para enseñar a programar en la escuela	48
2.5.1.	Iniciativas internacionales	50
2.5.1.1.	Code.org y “La Hora del Código”	50
2.5.1.2.	CAS: Computing at School	52
2.5.1.3.	CS Unplugged	52
2.5.1.4.	La Fundación Logo	52
2.5.2.	Iniciativas nacionales	53
2.5.2.1.	Escuelas del Futuro	54
2.5.3.	En la Facultad de Informática de la UNLP	55
2.6.	Resumen del capítulo	56
3.	Experiencias de programación en el aula	57
3.1.	Herramientas utilizadas en experiencias propias	58
3.1.1.	Experiencias con Scratch	58
3.1.2.	Experiencias con Alice	59
3.1.3.	Experiencias con Lightbot	60
3.1.4.	Experiencias con Juegos de Blockly	60
3.1.5.	Experiencias con las herramientas de code.org	60
3.1.6.	Experiencias con Pilas Bloques	61
3.1.7.	Experiencias con pilas-engine	61
3.1.8.	Resumen de las experiencias realizadas	61
3.2.	Enseñando a programar en la escuela primaria	64
3.2.1.	Sistematización propuesta	64

3.2.2.	Secuencias didácticas	69
3.2.2.1.	Secuencia didáctica: algoritmos y programas	69
3.3.	Enseñando a programar en la escuela secundaria	73
3.3.1.	Sistematización propuesta	73
3.3.2.	Secuencias didácticas	75
3.3.2.1.	Secuencia didáctica: introducción al lenguaje de programación Python	76
3.4.	Resumen del capítulo	81
4.	Programar con robots	82
4.1.	Programando con robots y <i>software</i> libre: el proyecto original	82
4.2.	Modalidades de trabajo	87
4.2.1.	Demostraciones	87
4.2.2.	Actividades con docentes	89
4.2.3.	Actividades con estudiantes	90
4.3.	Programando con Robots y Software Libre: la situación actual	92
4.3.1.	El simulador en pilas-engine	92
4.3.2.	Curso a distancia para docentes con XRemoteBot	93
4.3.3.	Niños de primaria: nuevos destinatarios del proyecto	94
4.4.	Resumen del capítulo	95
5.	ProBots3D: los robots en 3D	96
5.1.	FI3D: la Facultad de Informática en 3D	97
5.1.1.	Las premisas para el desarrollo	98
5.1.1.1.	El <i>software</i> libre como base	98
5.1.1.2.	Recursos con licencias libres	99
5.1.2.	Características de la implementación	100
5.1.2.1.	Blender para el modelo y el motor de la aplicación	101
5.1.2.2.	FI3D 1.0: el primer prototipo	101
5.2.	ProBots3D: los robots Múltiple N6 en FI3D	103
5.2.1.	La prueba de concepto	103
5.3.	Resumen del capítulo	107
6.	ProBots3D en el aula	108
6.1.	Incorporando ProBots3D a las actividades en el aula	108
6.2.	ProBots3D en la escuela primaria	109
6.2.1.	Secuencia didáctica: algoritmos y programas - incluye el uso de ProBots3D	110

6.3. ProBots3D en la escuela secundaria	112
6.3.1. Secuencia didáctica: introducción al lenguaje de programación Python - incluye ProBots3D	113
6.4. Resumen del capítulo	119
7. Evaluación de ProBots3D	120
7.1. Metodología utilizada	120
7.2. Evaluación con alumnos de nivel primario	122
7.2.1. Análisis cualitativo de la evaluación	126
7.3. La evaluación con alumnos de nivel secundario	128
7.3.1. Análisis cualitativo de la evaluación	134
7.4. Resumen del capítulo	136
8. Conclusiones y trabajos futuros	137
8.1. Conclusiones	137
8.2. Trabajos futuros	139
A. Bibliografía y referencias	140
A.1. Bibliografía	140
A.2. Referencias web	144
A.2.1. Herramientas utilizadas en la enseñanza de la programación	144
A.2.2. Iniciativas y programas que promueven la enseñanza de las ciencias de la computación	145
A.2.3. Desarrollos en el marco del proyecto “Programando con robots y software libre”	145
A.3. Misceláneos	145
B. Instrumentos de evaluación	146
B.1. Entrevista inicial realizada a alumnos de nivel primario	146
B.2. Entrevista final realizada a alumnos de nivel primario	147
B.3. Encuesta inicial para alumnos de nivel secundario	148
B.4. Encuesta a alumnos de nivel secundario luego de utilizar ProBots3D	149
B.5. Encuesta a alumnos de nivel secundario al finalizar la evaluación	150

Índice de figuras

2.1. Principales características de las herramientas para enseñar a programar.	15
2.2. Implementaciones de Logo actualmente en uso	17
2.3. Ejemplo de un programa en Scratch	18
2.4. Scratch: interfaz de usuario de las dos versiones estables del programa.	18
2.5. Juego de Pacman desarrollado en Scratch	20
2.6. Un programa sencillo desarrollado en Snap!	21
2.7. Juego que muestra cómo funciona un algoritmo para encontrar caminos más cortos.	22
2.8. Scratch Jr en una tableta Android.	23
2.9. S4SL: <i>Scratch for Second Life</i> y S4A: <i>Scratch for Arduino</i>	24
2.10. Distribución de edades de los usuarios de Scratch.	25
2.11. Simulación sencilla en Alice 2.4	26
2.12. Estructura iterativa representada en Scratch, Alice y Python.	27
2.13. Aplicación LightBot en su versión para sistemas Android y web	28
2.14. Juegos de Blockly	29
2.15. Actividades propuestas por Juegos de Blockly	29
2.16. Juegos de Blockly: solución en Javascript y en el lenguaje de bloques.	30
2.17. Algunas de las actividades propuestas por Code.org	31
2.18. Proyectos Yeti y Garden Robot de Made with code.	33
2.19. Vista de diseñador y vista de programación en MIT APPInventor.	34
2.20. Interfaz de usuario de Gameblox en el modo diseño.	35
2.21. Interfaz de usuario de la aplicación provista por pilas-engine.	36
2.22. Recursos provistos por pilas-engine.	37
2.23. Ejemplo de código que utiliza pilas-engine.	37
2.24. Pilas Bloques: sitio oficial con los desafíos propuestos.	38
2.25. “El recolector de estrellas”: desafío en el cual se introducen las estructuras repetitivas.	39
2.26. Captura del nivel más sencillo de CodeCombat.	40

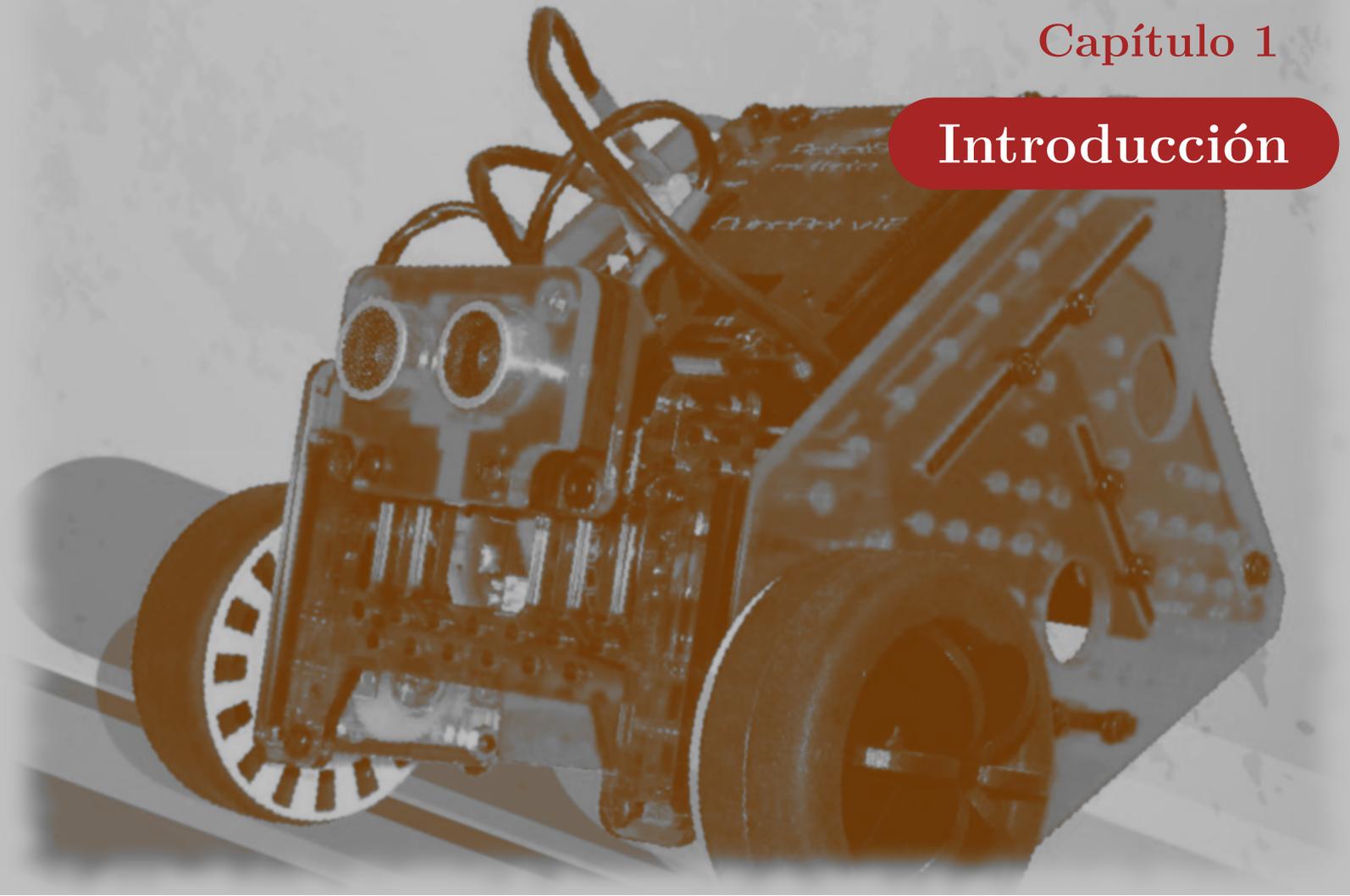
2.27. Desafío “Power of Thor” de CodinGame.	41
2.28. PSeInt	41
2.29. Aplicaciones desarrolladas en la asignatura Seminario de Lenguajes - Opción Python	43
2.30. XRemoteBot: aplicación que permite programar los robots del proyecto “Programando con robots y software libre” en forma remota.	44
2.31. Dropsy: eDucational ROBot Programming SYstem.	45
2.32. Interfaz de usuario de Rita.	46
2.33. Mapa conceptual donde se resumen las iniciativas descriptas.	49
2.34. Eventos registrados en “La hora de código” hasta el mes de enero 2016	51
3.1. Organización de contenidos propuesta para el nivel primario.	69
3.2. Planificación y propuestas para el nivel secundario - correspondiente al 2do. año. . . .	76
3.3. Consola de Python.	78
4.1. Robot <i>scribbler</i> utilizado entre los años 2009 y 2012.	84
4.2. Robot Multiplo N6 utilizado a partir del año 2012.	85
4.3. Algunas actividades realizadas con estudiantes secundarios durante los últimos años. .	91
4.4. El simulador en pilas-engine	93
5.1. FI3D: puerta de acceso al edificio.	100
5.2. FI3D 1.0: vistas de espacios físicos de la facultad y su representación en FI3D.	102
5.3. Captura del ingreso al edificio y del <i>hall</i> de entrada en Probots3D.	104
5.4. Desafío inicial en Probots3D destinado a nivel primario.	104
5.5. Dos versiones de ProBots3D desarrolladas en la prueba de concepto de esta tesis. . . .	105
6.1. Propuesta de uso de ProBots3D en el nivel primario.	109
6.2. Propuesta de uso de ProBots3D en el nivel secundario.	113
6.3. Consola de Python.	117
7.1. Escuela Primaria 67 Enrique Galli de la localidad de City Bell.	122
7.2. Imágenes de los alumnos de la primaria 67 interactuando con ProBots3D en el aula. .	124
7.3. Gráficos sobre el uso de ProBots3D durante la evaluación de los alumnos de primaria.	125
7.4. Respuestas más destacadas en la entrevista final.	126
7.5. Colegio Estudiantes de La Plata de City Bell.	128
7.6. Relevamiento de herramienta conocidas al momento de realizar la evaluación.	129
7.7. Imágenes de los alumnos de la secundaria Estudiantes de La Plata interactuando con ProBots3D en la sala de computación del colegio.	130
7.8. Datos destacados sobre el uso de ProBots3D.	131

7.9. Datos destacados sobre el aporte de ProBots3D para la programación de los robots reales.	132
7.10. Datos destacados luego de programar los robots reales.	133
7.11. Datos destacados sobre el uso de Python.	134
B.1. Encuesta inicial realizada a los alumnos secundarios.	148
B.2. Encuesta realizada a los alumnos secundarios luego de utilizar ProBots3D.	149
B.3. Encuesta final realizada a los alumnos secundarios.	150

Índice de tablas

2.1. Distintas versiones del programa Scratch.	25
2.2. Cursos destinados a estudiantes en la iniciativa code.org.	32
2.3. Tabla comparativa de herramientas que pueden ser usadas con alumnos de nivel secundario.	47
2.4. Tabla comparativa de herramientas que pueden ser usadas con alumnos de nivel primario y secundario.	48
3.1. Resumen de experiencias realizadas en las escuelas primarias.	62
3.2. Resumen de experiencias realizadas con alumnos de nivel secundario.	63
3.3. Resumen de contenidos abordados con cada herramienta.	64
3.4. Organización de contenidos propuesta para el 1er.y 2do año.	66
3.5. Organización de contenidos propuesta para el 3er. y 4to. año.	67
3.6. Organización de contenidos propuesta para el 5to. y 6to. año.	68
3.7. Organización de contenidos propuesta para el primer año de aplicación.	74
3.8. Organización de contenidos propuesta para el segundo año de la sistematización realizada.	75
3.9. Comandos básicos del robot Múltiple N6	79
4.1. Actividades con jóvenes de escuelas secundarias en el marco del proyecto “Programando con robots y software libre”	88
4.2. Actividades con niños de educación inicial y primaria realizados en el marco del proyecto “Programando con robots y software libre”	89
4.3. Módulos de la capacitación llevada a cabo en el marco del proyecto “Entre Robots, juegos y lenguajes: una experiencia de aprendizaje real en las aulas”.	90
5.1. Primitivas para programar en ProBots3D y sus equivalentes en la API de los robots reales.	106
5.2. Primitivas para programar en ProBots3D.	106

6.1. Comandos básicos del robot Múltiplo N6 118



1.1. Fundamentación

Los jóvenes de hoy socializan, se comunican e interactúan mediados por las tecnologías de la información y comunicación (TIC). Dichas tecnologías cumplen un papel fundamental en los actuales procesos de enseñanza-aprendizaje. Por lo general, las TIC se abordan en la escuela como soporte de las restantes áreas del conocimiento, aunque es importante ahondar en el estudio propio de las mismas, en particular, sobre aspectos de la **programación de computadoras**.

Introducir a docentes y estudiantes al mundo de la programación no sólo apunta a un aprendizaje técnico, sino que permite desarrollar una serie de habilidades, como el pensamiento analítico o de solución de problemas, que son muy requeridos en la sociedad actual: “saber programar” amplía las posibilidades del uso de las tecnologías, de las cosas que se pueden crear con las computadoras, y en general de las cosas que se pueden aprender.

Uno de los desafíos que se plantea respecto de la enseñanza de la programación es ver cómo

motivar e incentivar a los alumnos. En busca de este objetivo se han desarrollado diferentes iniciativas y estrategias que promueven la enseñanza de la programación como parte de las Ciencias de la Computación, buscando motivar a los alumnos en este área usando robots, simuladores o a través de competencias.

Existen varias iniciativas, tanto a nivel nacional como internacional, que proponen actividades para introducir las Ciencias de la Computación en las escuelas. Recientemente, el Ministerio de Educación y Deportes de la Nación, lanzó el Plan Nacional Integral de Educación Digital (PLANIED) (Ministerio de educación, 2017) a través del cual se propone integrar la comunidad educativa en la cultura digital. En su sitio web establece que: “Su misión consiste en que todos los estudiantes de la Argentina adquieran las habilidades necesarias para desenvolverse en el mundo actual y en la sociedad del futuro.” Entre sus objetivos se encuentra el de “desarrollar iniciativas orientadas a construir conocimiento sobre la programación y el pensamiento computacional”.

Esta tesis presenta una experiencia en la cual se trabaja la enseñanza de la programación en el aula de la escuela, y una propuesta de uso de una nueva herramienta que complementa las actividades desarrolladas en el marco del proyecto “Programando con robots y software libre”, llevado a cabo en el “Laboratorio de Investigación de Nuevas Tecnologías Informáticas” (LINTI)¹ de la Facultad de Informática de la Universidad Nacional de La Plata (UNLP): **ProBots3D**.

Dado que se utilizan herramientas de *software* libre, tanto ProBots3D como los materiales diseñados en esta tesis podrán ser utilizados y distribuidos libremente por quienes así lo deseen, permitiendo, además, poder controlar y adecuar el uso y acceso al mismo de acuerdo a las características de nuestra región.

1.2. Objetivos

El objetivo de esta tesis es desarrollar y evaluar una propuesta de uso de la herramienta ProBots3D, en el marco de las actividades prácticas del proyecto “Programando con robots y software libre”.

Para cumplir este objetivo se proponen los siguientes objetivos específicos:

- Analizar las estrategias existentes de enseñanza de programación en niños y jóvenes, tanto a nivel nacional como internacional.
- Desarrollar una prueba de concepto para ProBots3D que permitirá manipular los robots del proyecto “Programando con robots y software libre” en un entorno inmersivo basado en la herramienta FI3D, un modelo 3D de la Facultad de Informática de la Universidad Nacional de La Plata.

¹Sitio oficial del LINTI: <http://linti.unlp.edu.ar>. Último acceso: noviembre 2017.

- Elaborar las secuencias didácticas que incorporan ProBots3D con distintos perfiles de alumnos.
- Evaluar la prueba de concepto de ProBots3D basada en una secuencia didáctica elaborada para tal fin.

El proyecto “Programando con Robots y Software Libre” actualmente trabaja con niños y jóvenes de escuelas primarias y secundarias. En el caso de los estudiantes secundarios, los mismos asisten a clases presenciales, realizadas por lo general en el ámbito de la facultad, y realizan sus prácticas utilizando unos pequeños robots que deben ser programados utilizando el lenguaje de programación Python². A través de estas actividades, los jóvenes aprenden los primeros conceptos de programación y los robots brindan una fuerte motivación. Dado el costo de estos recursos, esta tesis incorpora al proyecto una herramienta adicional que permite coordinar actividades con escuelas que no cuenten con todos los recursos y que sirve de “puente” entre el trabajo con lenguajes de programación basados en bloques y Python, el cual es un lenguaje textual.

En el caso de los alumnos de escuelas primarias, no se llega a trabajar con el lenguaje Python, sino que se utilizan los robots como un agente motivador comandando los mismos a través de una interfaz visual. ProBots3D incorpora una nueva herramienta para abordar la programación en este nivel educativo.

1.3. Trabajos publicados

Los resultados parciales de esta tesis dieron lugar a los siguientes trabajos:

- **Programar en la Escuela: nuevos desafíos en las aulas.** Claudia Queiruga, Claudia Banchoff Tzancoff, Sofía Martín, Vanessa Aybar Rosales, Fernando López, Isabel Miyuki Kimura y Noelia Soledad Gómez. Presentado en WICC 2017.
- **FI3D: An Anticipatory Space.** Viviana Harari y Claudia Banchoff. Publicado en International Conference on Engineering Education & Research ICEER 2016. Australia. Noviembre 2016.
- **Programar en la escuela.** Claudia Queiruga, Claudia Banchoff, Sofía Martín, Vanesa Aybar Rosales, Fernando López. Presentado en WICC 2016. Entre Ríos. Abril 2016.
- **La Facultad en 3D: un entorno común para dos propuestas educativas.** Viviana Harari, Claudia Banchoff. Publicado en el Libro de Actas del XXI Congreso Argentino de Ciencias de la Computación (CACIC 2015) ISBN: 978-987-3724-37-4. Junin del 5 al 9 de octubre de 2015.

²Sitio oficial: <https://www.python.org/>. Último acceso: noviembre 2017.

- **Robótica Educativa y Videojuegos en el Aula de la Escuela.** Javier Díaz, Claudia Queiruga, Claudia Banchoff Tzancoff, Laura Fava y Viviana Harari. Actas de la 10a Conferência Ibérica de Sistemas y Tecnologías de Información. Águeda, Portugal Junio de 2015 AISTI | Universidade de Aveiro. ISBN: 978-898-98434- 5 – 5 Vol I. pp 1298-1303.
- **Aplicaciones ludificadas que combinan robots y dispositivos móviles. Experiencias en la Facultad de Informática de la UNLP.** Claudia Queiruga, Claudia Banchoff Tzancoff, Jorge Rosso, Fernando López, Isabel Kimura, Matías Brown Bartneche, Sofia Martín, Pablo Iuliano Aceptado para su publicación en WICC 2014. Ushuaia. Argentina. Mayo 2014.
- **Experiencias de la Facultad de Informática en la Enseñanza de Programación en Escuelas con Software Libre.** Javier Díaz, Claudia Banchoff Tzancoff, Claudia Queiruga, Eliana Sofia Martín. Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación a celebrarse en Buenos Aires, Argentina, del 12 al 14 de noviembre de 2014.

Actualmente se está trabajando en dos proyectos, uno de los cuales se publicará en formato de un manual de Informática para el segundo ciclo de la escuela secundaria y el otro contempla el diseño curricular de una especialización en didáctica de las Ciencias de Computación para docentes de la provincia de Buenos Aires.

- **Manual de Ciencias de la Computación para el segundo ciclo de secundaria.**
 - **Descripción:** el mismo consta de doce capítulos, seis de los cuales dedicados a introducir la temática de programación usando la herramienta Alice. A la fecha de cierre de este informe, el manual se encuentra en proceso de edición, habiéndose entregado los doce capítulos.
 - **Autores:** Claudia Queiruga, Claudia Banchoff, Paula Venosa y Silvina Justianovich.
 - **Financiación:** Fundación Sadosky.
 - **Destinatarios:** docentes de escuela secundaria.
 - **Editorial:** Colihue.
 - **Distribución:** se publicará tanto en formato papel como en formato digital bajo licencia Creative Commons.
- **Diseño curricular de la Especialización en Didáctica de las Ciencias de Computación para nivel terciario.**
 - **Descripción:** la propuesta ha sido entregada en diciembre de 2016. La misma consta de diez módulos los cuales han sido incluidos en la propuesta. Junto a la propuesta se han entregado los programas y las secuencias didácticas correspondientes a ocho de los diez

módulos, los cuales se encuentran en revisión. Desarrollado en conjunto con el ISFD N°95 de La Plata.

- **Autores:** Claudia Queiruga, Claudia Banchoff, Paula Venosa, Glenda Morandi, Soledad Gómez, Néstor Castro, Omar Spandre Silvia Navarro y Jorge Doménico Di Lorenzo.
- **Financiación:** Fundación Sadosky.
- **Destinatarios:** docentes de escuela secundaria.
- **Dictado:** está previsto su dictado en el ciclo lectivo 2018.

1.4. Proyectos de I+D y extensión

Varios de los resultados obtenidos en este trabajo se han desarrollado bajo el marco de diversos proyectos de extensión y/o investigación de los cuales la tesista ha sido directora o co-directora de los mismos. Estos proyectos permitieron obtener una base de experiencias tanto para la elaboración de las propuestas realizadas como de la evaluación final en esta tesis.

- **“Programando con robots y software libre”**, llevado a cabo desde el año 2009 a la fecha en el marco de las líneas de trabajo del LINTI.
- **“Las ciencias y los niños. Una experiencia de programación en la escuela primaria”**. Este proyecto se está llevando a cabo durante el presente año 2017. Fue acreditado y subsidiado en la convocatoria 2016 a proyectos de extensión de la UNLP.
- **“Extensión en vínculo con escuelas”**, llevado a cabo desde 2015 a la fecha. Fue acreditado y subsidiado en las convocatorias 2015 y 2016 a proyectos de extensión de la UNLP.
- **“Explorando el universo con *software libre*”**, realizado en el año 2016 y donde se trabajó en una primera experiencia de programación en escuela primaria. Fue acreditado y subsidiado en la convocatoria 2015 a proyectos de extensión de la UNLP.
- **“Desarrollo de aplicaciones y videojuegos móviles para educación”**, financiado por la Facultad de Informática en el año 2014.

1.5. Tesinas de grado dirigidas

La tesista dirigió las siguientes tesinas de grado en relación a la temática abordada en el presente trabajo:

- **DROPSY: eDucational ROBot Programming SYstem.** Esta tesina fue realizada por los alumnos Matias Javier Fuentes y Diego Walte Fernandez. Fue expuesta en el año 2016 con nota 9 (nueve). Esta tesina fue co-dirigida con la Lic. Claudia Queiruga.
- **XRemoteBot: Un servicio para programar robots en forma remota.** Esta tesina fue realizada por el alumno Fernando López. Fue expuesta en el año 2015 con nota 10 (diez) y co-dirigida con la Lic. Claudia Queiruga.
- **RITA en red: Una extensión de RITA que facilita las competencias en Robocode.** Esta tesina fue realizada por los alumnos Pablo Vilaltella y Santiago M. Rugna y expuesta en el año 2015 con nota 10 (diez). Al igual que la anteriores, esta tesina fue co-dirigida con la Lic. Claudia Queiruga.
- **CarTON: manejando en la ciudad,** realizada por las alumnas Stefanía Ive y María Pia Farinella. Esta tesina fue expuesta en el año 2014 con nota 9 (nueve) y co-dirigida con la CC. Viviana Harari.
- **Aplicaciones complementarias a ROBOCODE que faciliten el aprendizaje de programación en escuelas secundarias..** Esta tesina fue realizada por la alumna Vanesa del Carmen Aybar Rosales y expuesta en el año 2013 con nota 9 (nueve). Fue co-dirigida con la Lic. Claudia Queiruga.

1.6. Estructura de este informe

El presente informe está organizado en ocho (8) capítulos.

El capítulo 1 aborda la fundamentación y objetivos de esta tesis.

El capítulo 2 analiza las distintas iniciativas para enseñar programación a niños y jóvenes, describiendo algunas de las herramientas y estrategias más destacadas.

En el capítulo 3 se describen las experiencias de uso tanto en el aula de la escuela como en otros espacios educativos con distintas herramientas y perfiles de alumnos.

El capítulo 4 describe el proyecto “Programando con robots y software libre” que enmarca las actividades propuestas en esta tesis.

El capítulo 5 presenta la herramienta FI3D desarrollada en el LINTI y una prueba de concepto para ProBots3D desarrollada en el marco de la presente tesis.

En el capítulo 6 se describe una propuesta de uso de ProBots3D inserta en dos secuencias didácticas destinadas a alumnos de nivel primario y secundario.

En el capítulo 7 se describe una evaluación realizada con dos grupos de alumnos de nivel primario y secundario, en donde se pone a prueba las propuestas realizadas en el capítulo 6.

El capítulo 8 presenta las conclusiones y trabajos futuros de esta tesis.

Por último, se incluyen dos apartados donde se detalla las bibliografía y referencias utilizadas, y los protocolos de los instrumentos utilizados en la evaluación de la propuesta desarrollada.

1.7. Sobre la tesista

La tesista desarrolla actividades de investigación, docencia y extensión en el LINTI desde el año 1990 y trabaja en proyectos relacionados a la promoción y uso de *software* libre en educación desde el año 2006. En el sitio del proyecto **Lihuen GNU/Linux**³ pueden encontrarse algunas de las propuestas y actividades realizadas a lo largo de los últimos 10 años.

Desde el año 2008, comenzó a trabajar en proyectos que promueven la enseñanza de la programación en escuelas secundarias, siendo los proyectos más destacados “Programando con robots y software libre”, coordinando las actividades del mismo desde el año 2009 y “Entre Robots, juegos y lenguajes: una experiencia de aprendizaje real en las aulas” financiado por la Fundación YPF llevado a cabo en el año 2012. Los materiales y contenidos pueden consultarse en el sitio **Programando con Robots**⁴.

La tesista integra el equipo de contenidistas que participa de dos convenios entre la Universidad Nacional de La Plata y la Fundación Sadosky los cuales abordan temáticas relacionadas a esta tesis:

- Desarrollo del manual de Ciencias de la Computación para el segundo ciclo de secundaria.
- Diseño curricular de la Especialización en Didáctica de las Ciencias de Computación.

³Sitio oficial: <http://lihuen.info.unlp.edu.ar>. Último acceso: noviembre 2017.

⁴Sitio oficial del proyecto: <http://robots.linti.unlp.edu.ar/>. Último acceso: noviembre 2017.

Programar en la escuela



“Cuando creíamos que teníamos todas las respuestas, de pronto, cambiaron las preguntas.”

– Mario Benedetti

En este capítulo se presentan las principales razones que destacan la importancia de trabajar aspectos de programación con niños y jóvenes. Se describen y analizan las distintas iniciativas, tanto a nivel nacional como internacional, y algunas de las herramientas existentes para trabajar esta temática, haciendo especial hincapié en las más utilizadas en el aula de la escuela.

2.1. Introducción

La discusión sobre la influencia de las tecnologías de la información y comunicación (TIC) en la educación no es nueva. Ellen Wartella y Nancy Jennings en (Wartella, E. & Jennings, N., 2000) realizaron un análisis sobre las discusiones surgidas cada vez que se introdujo un nuevo recurso.

Cada vez que una nueva tecnología se masifica o se introduce en la vida cotidiana comienzan las

preocupaciones respecto del impacto de las mismas en los niños y jóvenes. Esto se planteó con la introducción del cine, la radio, la TV, Internet y, en la actualidad con la inclusión de los celulares, juegos en línea, entre otros.

En general, a lo largo de los años, las discusiones son similares: quienes están a favor de las innovaciones argumentan que las nuevas tecnologías abren nuevos mundos a los niños y jóvenes, mientras que los que están en contra, argumentan que los nuevos medios pueden ser usados para sustituir el aprendizaje de los principios éticos de la vida real, socabando la moralidad de los estudiantes, acercándolos a comportamientos criminales o a una conducta sexual ilícita (Clements, Douglas & Swaminathan, Sudha., 1995). Wartella y Jennings (Wartella et al, 2000), hace más de 15 años, realizaron un análisis sobre el impacto de las nuevas tecnologías en la educación, que podría aplicarse a la situación actual. En este trabajo, entre otras cosas, mencionan cómo la aparición de las primeras películas, a comienzos del siglo XX, introdujo una discusión que podría aplicarse al impacto del acceso a Internet noventa años más tarde: los que estaban a favor mencionaban la posibilidad que supone poder apreciar las maravillas naturales, cómo se mueve una locomotora sobre rieles en vez de ruedas o poder conocer pueblos lejanos, mientras los que se oponían a su uso masivo, mencionaban que las películas expondrían a los niños y jóvenes a escenas de violencia y libertinaje.

Si bien esto ha sucedido en forma constante con la aparición y masificación de nuevos medios, el tiempo y numerosos estudios han demostrado que, con una adecuada guía de los adultos responsables su uso, en todos los aspectos, es ampliamente ventajoso.

En la sociedad de hoy, los niños son expuestos a los nuevos medios a edades muy tempranas. A medida que el interés y la comprensión de los niños crece y el uso de los medios de comunicación se desarrolla, también lo hacen sus capacidades de pensamiento lógico y cognitivas.

Investigaciones sobre el aprendizaje de los niños (Wartella et al, 2000) muestran que extender la interactividad implicada en una experiencia con distintos medios puede afectar el proceso de aprendizaje. La interactividad es un elemento natural en la comunicación presencial (cara a cara), pero también un elemento en la comunicación a través de los medios. Como los nuevos medios posibilitan más interactividad, enriquecen mucho más el proceso de aprendizaje, aunque introducen nuevas preocupaciones: la exposición y la privacidad, entre otras.

La introducción de las computadoras como medio de acceso masivo permite sacar provecho de la interactividad natural provista. Pueden ofrecer oportunidades únicas para aprender a través de la exploración, resolviendo problemas de forma creativa y para una instrucción autoguiada.

La integración de la tecnología en la currícula de las escuelas requiere esfuerzo, tiempo, compromiso y muchas veces un cambio en las creencias. Ya en el año 1995, investigadores en educación (Clements et al, 1995) se preguntaban si la tecnología era introducida por los niños o por los docentes, si los

proyectos debían ser guiados o libres, si se focalizaba en los contenidos o en el conocimiento a adquirir, si las currículas debían ser planificadas o flexibles, con o sin computadora, entre otras cosas. Algunas de estas cuestiones ya fueron resueltas en la actualidad, pero otras siguen tan vigentes como hace más de 20 años. En todo esto el rol del docente es fundamental:

Un buen docente no es necesariamente aquel que conoce la mayoría de los conceptos o información en su campo de estudio, sino el que continuamente puede incorporar nueva información en forma creativa, guiando a sus estudiantes en sus procesos de aprendizajes.

La tecnología puede ser usada meramente para enseñar los mismos viejos materiales de la misma forma. La integración de las actividades con las computadoras pueden incrementar los logros, pero si sólo se incorporan sin una adecuación de las actividades, simplemente se encarecen los recursos sin sentido. Los docentes deben asumir el rol de guías o facilitadores en vez de meros difusores del conocimiento. Manuel Castells en (Fronteiras do Pensamento, 2014) declara que *“el aprendizaje en la mayor parte de las escuelas y universidades es totalmente obsoleto, porque insiste en producir una pedagogía basada en la transmisión de la información”*, pero según Castells, no se necesita de esto, dado que la gran mayoría de los contenidos se encuentran publicados en Internet y lo que se requiere son *“criterios para buscarlos y para combinarlos en los proyectos intelectuales, personales, profesionales. Esa capacidad de empoderamiento intelectual es lo que tiene que transmitir la escuela... empoderar intelectualmente, dar criterios propios.”* Para esto, es necesario un cambio radical dado que los docentes necesitan saber utilizar las TIC de manera didáctica para facilitar un aprendizaje personalizado para sus estudiantes. Pueden utilizar las TIC para diseñar sus materiales que podrían ser desde un simple apunte o un video tutorial en Youtube a la programación de una simulación sencilla.

La educación es la clave para el progreso económico, social y ambiental y los gobiernos a nivel mundial buscan mejorar sus sistemas educativos. El futuro de la educación en el siglo XXI no es simplemente alcanzar a más gente sino mejorar la calidad y diversidad de las oportunidades educacionales. Cómo organizar mejor y dar soporte a los procesos de enseñanza y aprendizaje requiere imaginación, creatividad e innovación (Orr, D., M. Rimini and D. van Damme, 2015).

La educación de calidad es fundamental en una sociedad que requiere de la innovación y la creatividad. Esto no se logra con proyectos aislados sino a partir de un compromiso y trabajo conjunto entre todos los involucrados, tanto docentes, autoridades, investigadores, desarrolladores de materiales educativos y la sociedad en general. Por este motivo, trabajar con recursos educativos abiertos¹

¹Los recursos educativos abiertos son aquellos materiales de enseñanza, aprendizaje o investigación que se encuentran disponibles para su utilización, adaptación y distribución gratuita.

y *software* libre² permite generar experiencias y recursos que pueden retroalimentarse con otras experiencias. Trabajar en un aprendizaje personalizado requiere que los contenidos puedan adaptarse a cada grupo de estudiantes, y este proceso de “adaptar y compartir” requiere de modificaciones de los recursos y, en muchos casos, trabajar sobre el *copyright* de los mismos (Lisette Kalshoven, 2015).

2.2. Aprender a programar

Investigadores pioneros en el tema, como Seymour Papert -uno de los creadores del lenguaje Logo³- y otros posteriores, sostienen que aprender a programar a edades tempranas ayuda a articular y analizar el conocimiento, y a tomar control del proceso de aprendizaje. Aprender lenguajes formales durante la infancia favorece que los niños hablen y piensen de forma más precisa sobre problemas complejos. Por otro lado, aprender a programar estimula la creatividad, las capacidades de atención y de resolución de problemas, siendo todas éstas habilidades requeridas en la sociedad actual (Queiruga C., Banchoff C., Martín S., Aybar Rosales V. & López F., 2016) (Queiruga C., Banchoff Tzancoff C., Martín S., Aybar Rosales V., López F., Miyuki Kimura I. & Gómez S., 2017).

Seymour Papert en (Papert S., 1980) reflexiona sobre cómo aún utilizando los medios más modernos, se sigue poniendo a los niños en una situación pasiva: “*ver y escuchar más que hacer*”. Las computadoras pueden programarse y responder a intercambios con los usuarios, por lo que los mismos tienen mayor control sobre dicho intercambio. Es así como Papert fue uno de los primeros en mencionar las potencialidades que el “saber programar” incorpora en el proceso de aprendizaje de los niños: “*el niño está al mando del software, construyendo su experiencia de aprendizaje.*”

Hace más de 40 años Papert decía:

“Programar una computadora no significa ni más ni menos que comunicarse con ella en un lenguaje que tanto la máquina como el usuario “comprenden”. Y aprender lenguajes es una de las cosas que mejor hacen los niños. Todo niño normal aprende a hablar, ¿por qué no aprendería entonces a hablar con la computadora?”

“En el ambiente Logo la relación se invierte: el niño, incluso en edad preescolar, está al mando; los niños programan a las computadoras y se embarcan en una exploración del modo en que ellos mismos piensan. La experiencia puede ser embriagadora: pensar sobre el pensamiento convierte al niño en epistemólogo, una experiencia no compartida siquiera por la mayoría de los adultos.”

Si bien hay múltiples estudios que corroboran lo propuesto por Papert hace casi medio siglo, en las escuelas de nuestro país son muy pocas las experiencias que pueden encontrarse en donde se enseña

²El *software* libre es aquel *software* que se puede estudiar, modificar y distribuir libremente. En el capítulo 5 se extenderá este concepto.

³Logo es un lenguaje de programación creado en 1967 para enseñar a programar. Muy conocido y difundido.

programación. La mayoría de las escuelas que sí lo hacen son escuelas secundarias técnicas y con las más variadas alternativas. En las escuelas primarias, las experiencias más comunes usan principalmente herramientas como Scratch⁴ y se muestran como experiencias novedosas en congresos y simposios de la especialidad.(Bensabat C., 2014).

Aprender a programar permite a los estudiantes que asuman un papel activo en relación al uso de sus computadoras y de los elementos tecnológicos que los rodean. Los mismos dejan de ser consumidores de información, se los promueve para que puedan dar respuestas imaginativas y divergentes y que pongan en juego sus significados personales.

En resumen, se puede decir que la Programación, como disciplina, promueve el desarrollo de habilidades tales como:

- La **abstracción**, que incluye la apropiación de técnicas como la simplificación de problemas, la definición de soluciones generales aplicables a problemas similares, y la asignación de nombres significativos a las distintas partes de una solución.
- La **operacionalidad**, la cual supone la definición de soluciones en términos de un conjunto de pasos que deben ejecutarse en un orden determinado para alcanzar un objetivo.
- La **creatividad**, inventiva o imaginación constructiva promueve la generación de nuevas ideas o relaciones entre ellas que conducen a soluciones originales.
- El **trabajo colaborativo**, tanto entre estudiantes como entre estudiantes y docentes, el cual fomenta el descubrimiento y la búsqueda de soluciones, coincidencias y diferencias, para mejorar las relaciones y construir aprendizajes significativos de forma colectiva. También ayuda a desarrollar “*el pensamiento crítico, la experiencia de grupo, la definición de la propia identidad, la reflexión y experimentación de valores como la democracia, la participación, la solidaridad, etc*”(Lobato, 1998, citado por Begoña Alfageme González, 2003).
- El **pensamiento computacional**, el cual, según Jeannette Wing “*implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática.*” (Wing, 2006).

2.3. Enseñar a programar

Como se mencionó anteriormente, aprender a programar contribuye a que los niños y jóvenes adquieran nuevas habilidades que ayudan en la comprensión y resolución de problemas, muy valoradas

⁴En la siguiente sección se describen ésta y otras herramientas similares.

en la sociedad actual. Para lograr esto, es fundamental desarrollar metodologías para enseñar a programar que se adecuen a las diferentes realidades e incentiven su incorporación a las distintas currículas educativas.

En la sección anterior se mencionó la aparición del lenguaje Logo hace más de medio siglo pero, sin embargo, recién en los últimos años distintos países comenzaron incorporar esta temática en las escuelas en forma masiva. En nuestro país, como se analizará más adelante en este capítulo, las iniciativas son aisladas y, hasta comienzos del presente año, no existía una política general al respecto.

Al analizar el por qué se demoró la incorporación de esta temática en las aulas, el principal problema que se menciona no fue el lenguaje o la herramienta utilizada (originalmente Logo), sino la forma en que se lo enseñó. En la mayoría de los casos, los estudiantes pueden acceder a utilizar las computadoras de la escuela pocas horas a la semana, y muchas veces sólo las utilizan como herramientas para acceder a información en Internet y generar documentos o presentaciones.

Enseñar a programar implica disponer de más tiempo con las computadoras, de manera tal que los estudiantes puedan resolver las consignas. En un primer momento, a esta dificultad se le sumó el hecho de que la mayoría de los docentes estaban convencidos que los niños y jóvenes deben adquirir habilidades informáticas, pero la programación no estaba incluida en ellas, sólo se trabajaba esta temática con jóvenes o adultos.

Si bien el análisis anterior se basa en experiencias de hace medio siglo, esta situación no ha cambiado demasiado en la actualidad. Las escuelas primarias no cuentan con un espacio curricular especial para incluir esta temática y las escuelas de educación secundaria sólo cuentan con la asignatura “Nuevas Tecnologías de la Información y la Conectividad” (NTICX) ubicada en el 4to. año del ciclo superior (Dirección General de Cultura y Educación de la provincia de Buenos Aires, 2010), con una asignación semanal de dos horas. En algunas escuelas, tanto primarias como secundarias, se cuenta con espacios extracurriculares, pero los contenidos dictados en estas áreas son muy dispares.

En los últimos años, se han creado numerosas iniciativas, que incluyen distintas herramientas y estrategias, mediante las cuales se está intentando incorporar el estudio de las Ciencias de la Computación en general y de la programación en particular en los distintos niveles de la educación. Estas iniciativas tienen mayor o menor impulso en los distintos países y regiones. En nuestro país comenzó con campañas focalizadas en la educación media para fomentar el estudio de carreras técnicas y científicas. Estas iniciativas, aún en la actualidad, se aplican discrecionalmente, de acuerdo a las iniciativas particulares de docentes y directivos. En este último tiempo, esta temática se ha fortalecido notablemente siendo la base para la discusión de un cambio en la currícula escolar en todos sus niveles. La incorporación del estudio de las Ciencias de la Computación, haciendo hincapié en la programación, es fundamental en la formación de niños y jóvenes para que puedan adaptarse y convivir en esta

sociedad cada vez más conectada.

Gibson menciona en (Gibson, 2012) que *“enseñar a nuestros niños y jóvenes a programar para que las máquinas del futuro hagan lo que ellos quieran parece mejor idea que dejar que ocurra justo lo contrario.”*

Estas máquinas del futuro que menciona Gibson no se encuentran tan lejos de nuestra realidad. El mundo de hoy es un mundo programable, donde la habilidad para poder sacar provecho o no de esta realidad marcará una gran diferencia.

Desde hace años se menciona el término “brecha digital”, haciendo referencia a la separación de la sociedad que se da entre aquellos que tienen acceso a las TIC y aquellos que no lo tienen. Esta separación no sólo se basa en la infraestructura y recursos con los que se cuenta, sino también incluye al uso de los mismos. Esto significa que, además de considerar aspectos de accesibilidad a los servicios que ofrece la tecnología, hay que tener en cuenta la posesión de habilidades y conocimientos para hacer un uso adecuado de la misma. (Díaz J., Harari I., Harari V. & Banchoff C. , 2010). Tim Berners-Lee (Berners-Lee, 2013) afirma que hay dos tipos de brechas digitales. La primera y más clara es la descrita anteriormente, es decir aquella que divide a los que cuentan con una computadora y acceso a Internet y los que no. Pero menciona otro tipo de brecha, menos conocida, que es la que separa a aquellos que pueden programar y aquellos cuyas habilidades informáticas se limitan a saber cómo trabajar con aplicaciones estándares, como procesadores de texto y hojas de cálculo. También menciona que el *“código abierto hace que todo vaya más rápido”* dado que aumenta el potencial de compartir código y lograr un desarrollo rápido y de bajo costo.

La revista Smithsonian publicó un reporte especial (Tyre, 2013) donde compara la enseñanza de la programación a la enseñanza de una segunda lengua. En un mundo interconectado, donde todos los objetos son susceptibles de ser programados y adaptados, la nueva brecha digital se refiere a quienes tienen competencia en programación y los que carecen de nociones básicas de este nuevo “segundo lenguaje universal”.

2.4 Herramientas para enseñar a programar

Si bien la propuesta de Papert, con la utilización de Logo, fue una de las primeras iniciativas para enseñar a programar a niños y jóvenes, a lo largo de estos años han surgido numerosas iniciativas que abarcan una gran variedad de herramientas y actividades. En esta sección se describen algunas de las herramientas más utilizadas para la enseñanza de la programación, que fueron seleccionadas luego de un relevamiento exhaustivo de experiencias tanto a nivel nacional como internacional. Estas herramientas tienen un objetivo común: presentar una interfaz simple, cuyos elementos sean fácilmente

identificables por niños y jóvenes, y con las que se puedan generar actividades motivadoras. No todas proponen las mismas estrategias: algunas presentan solamente desafíos que deben resolverse con un conjunto específico de instrucciones; otras presentan todas las instrucciones disponibles para que se pueda desarrollar una aplicación libremente; algunas utilizan un lenguaje visual basado en bloques y otras proponen trabajar con lenguajes textuales. Estas características determinan el tipo de actividad a realizar y limitan el grupo etáreo destinatario.

El foco de esta tesis está puesto en el análisis y uso de herramientas de código abierto, aunque en este relevamiento se han incluido algunas que, si bien son privativas, presentan características que se consideraron destacadas y son de uso libre para fines académicos⁵.

La mayoría de estas herramientas brindan una interfaz que permite construir los programas usando un lenguaje de programación visual basado en bloques. Esta forma, contrapuesta a los lenguajes de programación textuales utilizados en la industria de producción de *software*, permiten abstraer a los niños y jóvenes la sintaxis específica de estos lenguajes textuales, permitiendo enfocarse solamente en los conceptos y no en las cuestiones sintácticas de cada elemento.

La figura 2.1 presenta un diagrama conceptual donde se resumen las principales características analizadas en este tipo de herramientas. En el mismo, se destaca el tipo de lenguaje de programación utilizado, los conceptos que pueden ser abordados con las mismas, los recursos adicionales que incluyen y las posibilidades de uso de cada una.



Figura 2.1. Principales características de las herramientas para enseñar a programar.

En las siguientes secciones se describen las herramientas analizadas, comenzando con las más

⁵En el capítulo 5, sección 5.1.1.1 se aborda la definición de *software* libre y sus ventajas en la adopción de este tipo de *software* en el ámbito educativo.

antiguas y pioneras en el tema, finalizando con dos tablas comparativas que exponen las principales características provistas por cada una de ellas.

Las herramientas analizadas en esta tesis fueron seleccionadas entre las de uso más frecuente en iniciativas para enseñar programación a niños y jóvenes tanto a nivel nacional como internacional.

2.4.1. Logo

Logo es un lenguaje de programación diseñado con fines didácticos por Seymour Papert, Wallace Feurzeig, Cynthia Solomon y Marvin Minsky a mediados de los años '60. Está basado en el paradigma de programación funcional⁶ y fue creado con el objetivo de enseñar conceptos matemáticos a través de la programación. Basado en la teoría constructivista (Pozo J., 2008), el aprendizaje con Logo se centra en el planteo de desafíos que deben ser resueltos en un entorno interactivo donde los niños ponen a prueba sus soluciones. En su libro *Mindstorms* (Papert S., 1993) Papert plantea un ambiente en el cual los niños pueden aprender matemáticas, en particular centrándose en la geometría, de forma natural como aprenden a hablar su lengua materna sin instrucción formal. Para esto utiliza una “Tortuga” proponiendo un ambiente donde la matemática sea más accesible a los jóvenes estudiantes que las formas tradicionales en que la misma se enseña en las escuelas.

Si bien el origen de Logo se sitúa en los años '60, en el año 2014 Boytchev (Boytchev P., 2014) presentó un proyecto denominado “Logo Tree” donde se muestra un árbol genealógico de este lenguaje y todas sus implementaciones. En este informe, cuya última revisión es de octubre de 2014, se presentan trescientas tres (303) implementaciones. La figura 2.2 muestra la proporción de implementaciones aún vigentes de este *software*. El relevamiento clasifica como “Activas” a aquellas que se encuentran en desarrollo y se esperan nuevas versiones de las mismas; “Frizadas” a aquellas implementaciones cuyo desarrollo se encuentra parado pero que aún se encuentran en uso y el resto son aquellas que no tienen más soporte o no se encuentran usuarios que la utilizan. En este relevamiento también se detallan aspectos como plataforma donde se ejecutan y datos de quiénes las han desarrollado. Como puede verse, aún existen más de sesenta (60) implementaciones de Logo aún activas.

⁶Un paradigma de programación es una forma particular de plantear una solución a un problema. Cuando se habla del paradigma funcional se basa en el uso de funciones matemáticas para crear los programas.

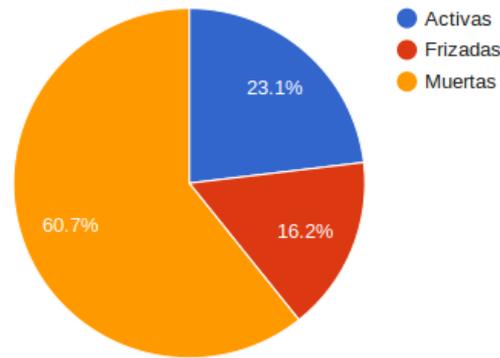


Figura 2.2. Implementaciones de Logo actualmente en uso

Logo brinda una interfaz sencilla y una serie de comandos básicos para mover un personaje centrado en un escenario. Utilizando estos comandos se plantean diversos desafíos basados en la generación de figuras geométricas y/o estrategias para desplazar al personaje. En su versión original, el mismo se trata de una tortuga, el cual, en algunas implementaciones, fue reemplazada por un robot u otros personajes.

Las ideas principales de Logo son las bases de múltiples herramientas que surgieron luego.

2.4.2. Scratch

Scratch⁷ es un proyecto del Grupo *Lifelong Kindergarten* del Laboratorio de Medios del Instituto de Tecnología de Massachusetts (MIT, por sus siglas en inglés: *Massachusetts Institute of Technology*) mediante el cual se pueden trabajar aspectos de programación donde los niños y jóvenes pueden realizar juegos o animaciones interactivas de una manera muy sencilla. Es multiplataforma, es decir que puede ejecutarse en distintos sistemas operativos, y se distribuye bajo licencia GNU *General Public License V2* (GPL2, 1991).

Tanto los objetos como el escenario en Scratch se programan utilizando un lenguaje de programación basado en bloques. La figura 2.3 muestra un programa en Scratch que permite dibujar un cuadrado al presionar la bandera verde de la interfaz de usuario. Cada instrucción está representada por un bloque que debe “encastrarse” con otros para formar los programas.

⁷Sitio Oficial: <https://scratch.mit.edu/>. Último acceso: noviembre 2017.



Figura 2.3. Ejemplo de un programa en Scratch

Scratch tiene dos versiones activas. La versión 1.4, la más antigua, puede descargarse en una computadora para utilizar esta herramienta sin la necesidad de una conexión a Internet. Esta versión viene instalada en las *netbooks* entregadas por los distintos programas gubernamentales de Argentina. La versión 2 tiene un editor en línea accedido desde una navegador web y, aunque también brinda la posibilidad de contar con una versión fuera de línea de este mismo editor, el mismo no está disponible para todas las plataformas⁸. La figura 2.4 muestra una captura de la interfaz de usuario de las dos versiones de Scratch, en la cual se destacan distintas zonas de la interfaz de usuario que se describen a continuación.

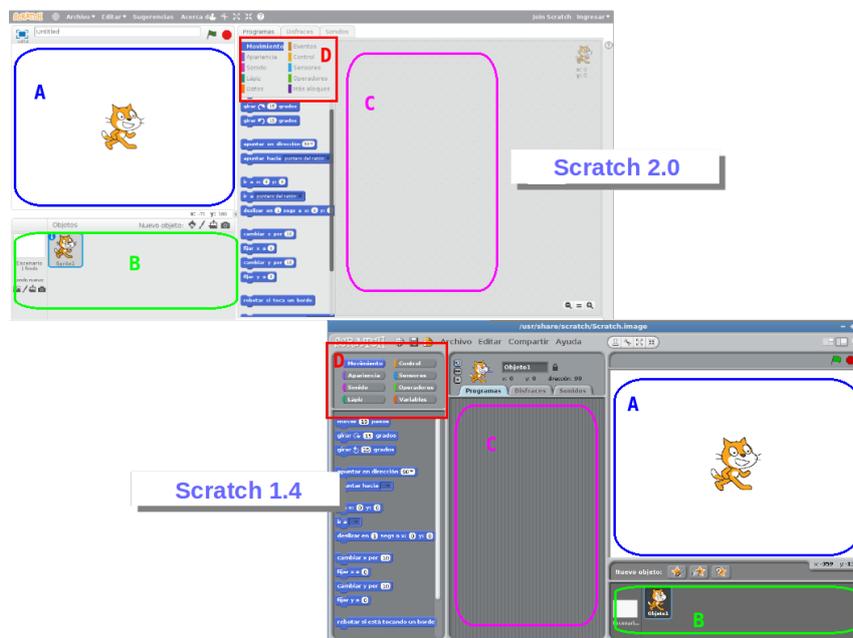


Figura 2.4. Scratch: interfaz de usuario de las dos versiones estables del programa.

⁸Para utilizar el editor de Scratch 2 en forma desconectada es necesario instalar previamente otras herramientas. Más información en <https://scratch.mit.edu/download>. Último acceso: noviembre 2017.

Ambas versiones presentan una interfaz muy similar en la cual se encuentra una zona donde se muestran los personajes y objetos al ejecutar los programas (A); una zona donde se ubica el escenario y los personajes que se incluyen en el programa (B); otra zona donde se ubican los bloques que conforman los programas (C); y una sección donde se encuentran todos los bloques, clasificados en distintas categorías (D). Si bien hay algunas diferencias entre las dos versiones, las categorías provistas son:

- **Movimiento:** aquí se encuentran los bloques que permiten desplazar, girar y ubicar, entre otras acciones, a los objetos incluidos en el programa. Estos bloques sólo están disponibles para los objetos, no para los escenarios.
- **Apariencia:** agrupa bloques que permiten cambiar la apariencia de los objetos: cambiar un disfraz (esto es cambiar la imagen de un objeto), cambiar el fondo de un escenario y mostrar diálogos, entre otras instrucciones.
- **Sonido:** en esta sección se encuentran los bloques que permiten agregar sonidos a los programas.
- **Lápiz:** reúne bloques que permiten dibujar en el escenario de nuestra animación o juego. Incluye bloques tales como “bajar lápiz” el cual al mover el personaje, va dejando un rastro en el escenario de un color y grosor de línea que puede programarse también con bloques de este sector.
- **Control:** agrupa bloques que representan estructuras condicionales y repeticiones y, en la versión 1.4 se encuentran también los bloques relacionados al manejo de eventos. Estos últimos en la versión 2 se encuentran en una categoría separada denominada **Eventos**.
- **Sensores:** corresponde a bloques que permiten obtener información del entorno, como ser el ingreso de datos o si un objeto está tocando a otro, por ejemplo.
- **Operadores:** esta categoría presenta los bloques que permiten construir las expresiones condicionales y relacionales, así como también algunas funciones matemáticas.
- **Variables:** en esta sección se encuentran los bloques que permiten crear y manipular variables y listas.
- **Más bloques:** esta clasificación está presente sólo en la versión 2 y permite crear nuevos bloques, permitiendo trabajar el concepto de abstracción procedural.

Scratch es una herramienta muy utilizada por docentes en muchos lugares del mundo con niños y jóvenes de distintas edades. En Argentina, hay registros de experiencias de uso de Scratch en varias instituciones (Bensabat, C., 2014). Las mismas son muy variadas y de muy diversos contextos, pero no hay una sistematización de su uso. Principalmente se trata de iniciativas individuales llevadas adelante

por docentes interesados en acercar la programación a los niños y jóvenes. Con Scratch se pueden realizar desde animaciones muy sencillas hasta juegos más complejos. La figura 2.5 muestra el popular juego de Pacman desarrollado en Scratch 1.4.

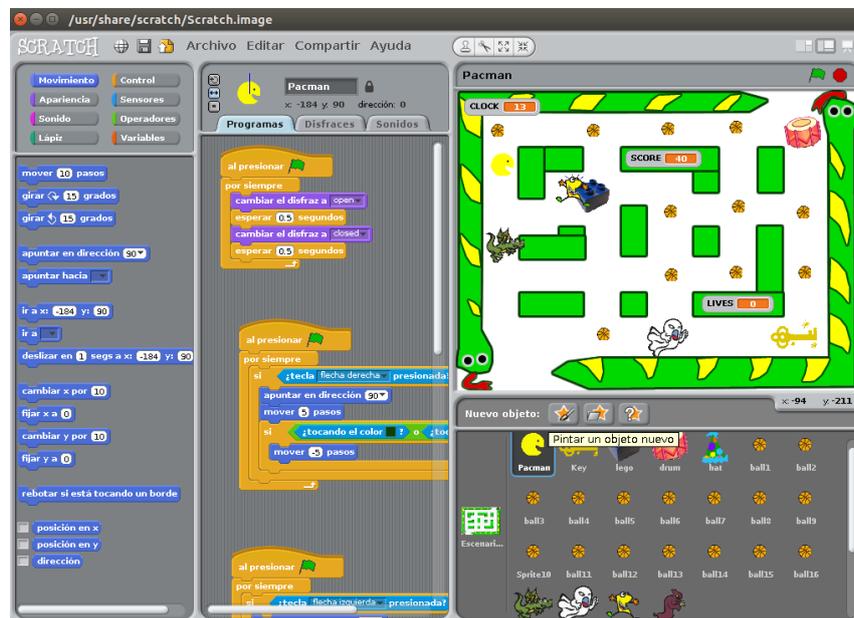


Figura 2.5. Juego de Pacman desarrollado en Scratch

En el sitio oficial de Scratch existe numerosos materiales tanto para docentes como para padres, así como también numerosos libros y publicaciones de distintos autores. Es posible encontrar varios proyectos tales como <https://scratch.mit.edu/projects/2345919/> y <https://scratch.mit.edu/projects/13701368/>.

A lo largo de los años han surgido varias derivaciones de esta herramienta, algunas de las cuales se encuentran aún en desarrollo y otras han sido descontinuadas. En octubre de 2014 se publicó una versión disponible para plataformas IOS y Android, pero las mismas no fueron mantenidas. En el año 2015, se lanzó Scratch Jr., una versión para dispositivos móviles destinada a niños de 5 a 7 años de edad, la cual es descripta en la sección 2.4.2.2.

En las siguientes secciones se describirán algunas herramientas basadas en Scratch.

2.4.2.1 Snap!

Como se mencionó anteriormente Scratch 2 puede utilizarse tanto en modo desconectado como en línea, pero ambas versiones requieren de programas adicionales tales como Adobe Flash Player⁹. Como una alternativa, la Universidad de California en Berkeley presentó Snap!¹⁰, una versión implementada

⁹Sitio de descarga: <https://get.adobe.com/es/flashplayer/>. Último acceso: noviembre 2017.

¹⁰Sitio oficial: <http://snap.berkeley.edu/>. Último acceso: noviembre 2017.

en HTML5¹¹ y Javascript¹².

Snap! es una aplicación web, con una interfaz muy similar a Scratch 1.4 y no requiere de ninguna extensión adicional instalada en el navegador para su funcionamiento. La figura 2.6 muestra una aplicación muy sencilla desarrollada en Snap! donde el personaje es un monigote dibujado en la propia herramienta.

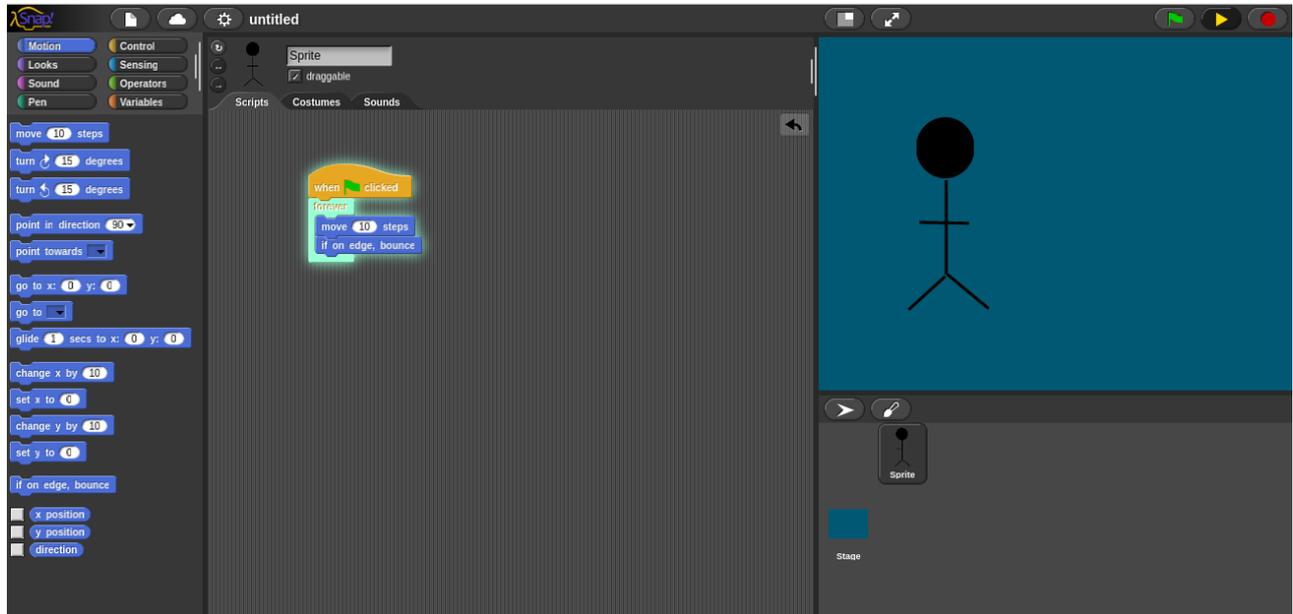


Figura 2.6. Un programa sencillo desarrollado en Snap!

Si bien el sitio oficial y la información presente en el mismo se encuentra en idioma inglés, la disposición de los elementos y las instrucciones están representadas con los mismos bloques que aparecen en Scratch.

En la página del proyecto se encuentran disponibles algunos ejemplos con los que se pueden interactuar y varios materiales para docentes. A modo de ejemplo la figura 2.7 muestra un programa realizado con Snap!, “Path Finder”, que permite armar un escenario y luego mostrar el camino más corto entre dos puntos del mismo.

¹¹HTML es el lenguaje utilizado para el desarrollo de páginas web. La última versión de este lenguaje, HTML 5, fue publicada como recomendación de la W3C en octubre de 2014. Más información en <https://www.w3.org/TR/html5>. Último acceso: noviembre 2017.

¹²Javascript es un lenguaje de programación utilizado principalmente para agregar dinamismo en las páginas web.

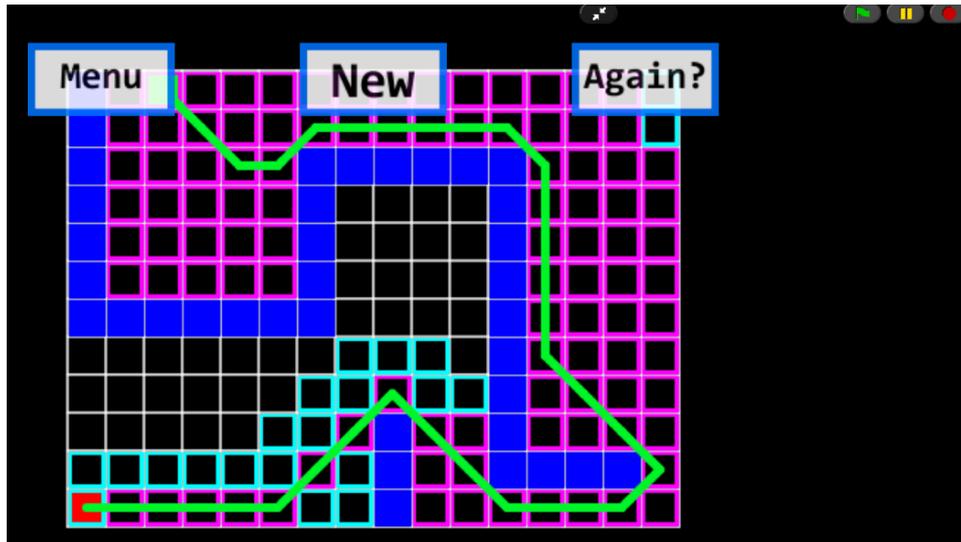


Figura 2.7. Juego que muestra cómo funciona un algoritmo para encontrar caminos más cortos.

También, en el sitio oficial existe una funcionalidad que permite importar los proyectos realizados en Scratch a esta herramienta. En las pruebas realizadas, la importación funcionó adecuadamente, aunque la manipulación de los elementos que componen el juego no siempre pudo efectuarse, sólo se lo pudo ejecutar.

2.4.2.2. Scratch Jr

Pensando en la incorporación de la programación en edades tempranas, en el año 2015 se lanzó Scratch JR¹³. Esta herramienta es una aplicación para dispositivos móviles destinada a niños de 5 a 7 años de edad mediante la cual, al igual que la aplicación original, los niños realizan animaciones interactivas y juegos sencillos. Lo interesante de esta aplicación es que su interfaz fue adaptada a la edad de los destinatarios. La figura 2.8 muestra una captura de esta aplicación ejecutándose en una tableta Android.

¹³Sitio oficial: <http://www.scratchjr.org/>. Último acceso: noviembre 2017.



Figura 2.8. Scratch Jr en una tableta Android.

En el sitio oficial de esta herramienta, además de las opciones de descarga, se encuentran disponibles lecciones destinadas a docentes con guías de actividades¹⁴. Si bien las mismas se encuentran en idioma inglés, son muy sencillas y podrían ser adecuadas a nuestra región.

2.4.2.3. Otras versiones de Scratch

Second Life¹⁵ es un entorno virtual 3D que nació y se popularizó hace varios años y tuvo un auge muy notable en las distintas comunidades. Dada la popularidad que alcanzó este entorno, se desarrolló una versión de Scratch para el mismo. **Scratch for Second Life**, conocido como S4SL¹⁶, permite agregar nuevo comportamiento e interacción a los objetos del entorno. Es una alternativa a la programación tradicional dentro del entorno de Second Life, en el cual se utiliza el lenguaje LSL (“*Linden Scripting Language*”).

Scratch for Arduino, conocido como S4A¹⁷, permite programar la plataforma de *hardware* libre Arduino¹⁸ utilizando la misma interfaz de programación que Scratch. Además de los bloques comunes provistos por Scratch, incorpora algunos nuevos que permiten interactuar con los sensores y otros

¹⁴Guías para docentes: <http://www.scratchjr.org/hoc.html>

¹⁵Sitio oficial: <https://secondlife.com>. Último acceso: noviembre 2017.

¹⁶Sitio oficial: http://web.mit.edu/~eric_r/Public/S4SL/. Último acceso: noviembre 2017.

¹⁷Sitio oficial: http://s4a.cat/index_es.html. Último acceso: noviembre 2017.

¹⁸Arduino es una plataforma electrónica de código abierto. Más información en: <https://www.arduino.cc/>. Último acceso: noviembre 2017.

recursos conectados a una placa Arduino.

Esta versión está enlazada a Snap!, donde se encuentra una versión de esta herramienta adaptada con los bloques de programación para Arduino.

La figura 2.9 muestra una captura de S4A y S4SL¹⁹.

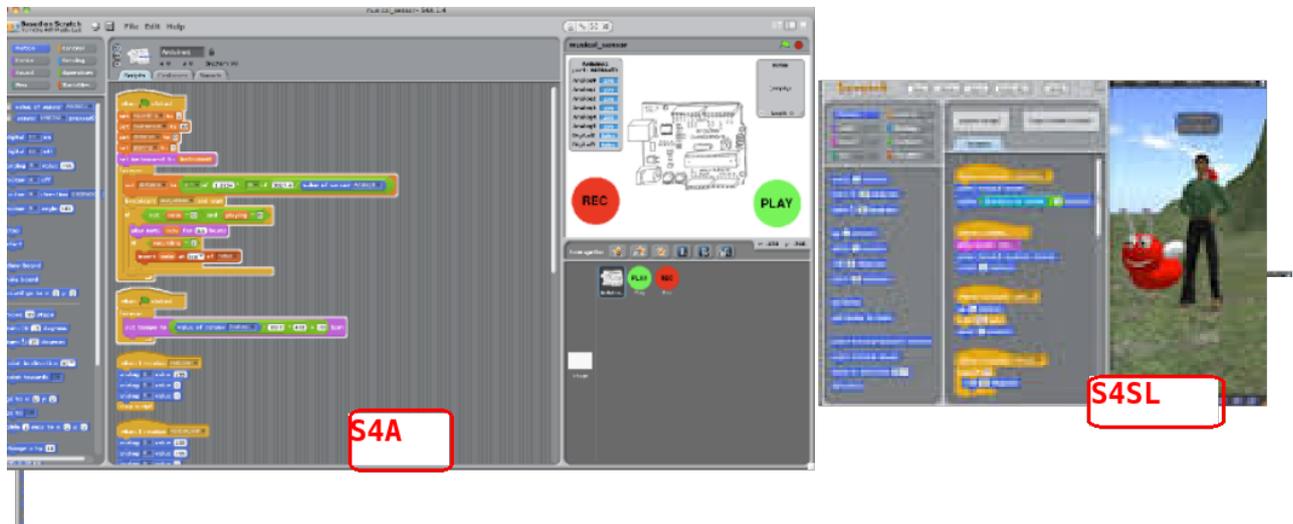


Figura 2.9. S4SL: *Scratch for Second Life* y S4A: *Scratch for Arduino*

2.4.2.4. Comparativa de herramientas basadas en Scratch

En esta sección se resumen las principales características de cada una de las versiones y herramientas derivadas de esta aplicación. Las mismas, limitadas por los distintos entornos de ejecución, proporcionan una muy variada gama de alternativas de uso.

Si bien la interfaz base es la misma, se diferencian entre sí por los bloques específicos que contienen, que permiten el trabajo con cada uno de los distintos entornos.

La tabla 2.1 muestra información resumida de todas estas versiones.

Como puede verse, existen versiones de esta aplicación para casi todos los distintos tipos de plataformas. La mayoría presentan las mismas componentes básicas y agregan las específicas para las plataformas y/o dispositivos para las cuales fueron desarrolladas.

Algunas de estas versiones fueron generadas de modo “experimental” y luego fueron discontinuadas. Otras están en pleno desarrollo y son trabajadas por grupos distintos a los desarrolladores originales.

¹⁹La imagen ilustrativa de S4SL fue adquirida del sitio oficial, dado que esta herramienta fue discontinuada.

Aplicación	Año de lanzamiento	Estado	Dispositivo móvil	PC
Scratch 1.4	2005	Activa		X
Scratch 2	2013	Activa		X
Snap!	2011	Activa	X	X
S4Android	2014	Discontinuada	X	
S4A	2013	Activa		X
S4SL	2009	Discontinuada		X
Scratch Jr	2015	Activa	X	

Tabla 2.1. Distintas versiones del programa Scratch.

En el sitio oficial existe una sección de estadísticas²⁰, donde se muestran distintos aspectos sobre su uso y adopción. Al momento de cerrar este informe, se registraban 26.416.561 de proyectos compartidos. La figura 2.10 muestra el rango de edades de los niños y jóvenes que utilizan Scratch. en la misma s puede ver también que, al momento del cierre de este informe, existían más de dos millones de usuarios registrados.

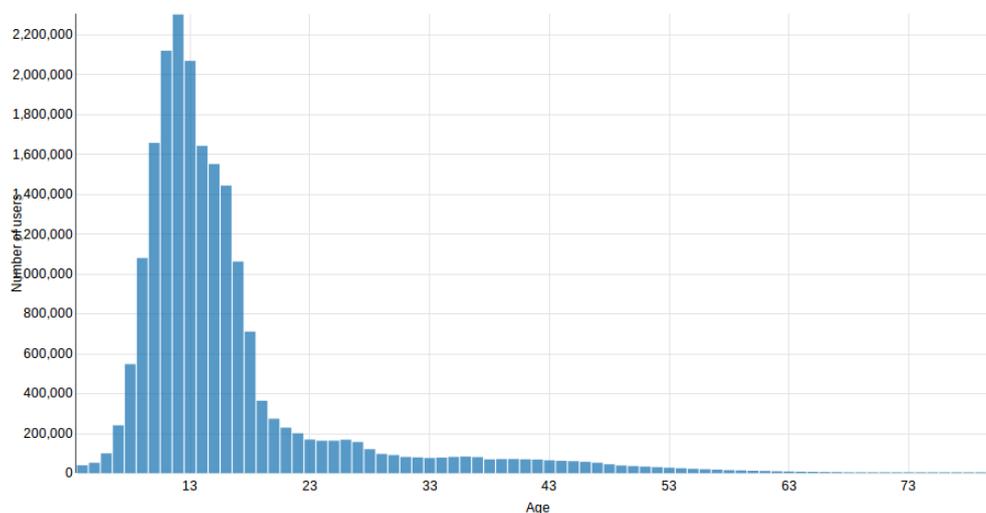


Figura 2.10. Distribución de edades de los usuarios de Scratch.

2.4.3. Alice

Alice²¹ fue diseñada en la Universidad Carnegie Mellon y permite crear animaciones y juegos a través en un entorno 3D utilizando un lenguaje de programación orientado a objetos. Si bien Alice no

²⁰ Estadísticas variassobreScratch: <https://scratch.mit.edu/statistics/>. Último acceso: noviembre 2017.

²¹ Sitio Oficial: <http://www.alice.org>. Último acceso: noviembre 2017.

es una herramienta libre, puede ser distribuida en forma gratuita con fines educativos.

Tanto el sitio oficial de Alice, como los materiales de apoyo se encuentran mayormente en idioma inglés. Existe una extensión que puede descargarse y aplicarse al *software* de manera tal de tener la interfaz de usuario en español, pero cabe aclarar que no todos los elementos se encuentran totalmente traducidos y algunas de las bibliotecas de objetos, no se acceden correctamente luego de la instalación de esta extensión.

Existe una versión libre de Alice en idioma español, denominado **Rebeca a través el espejo**²² la cual sí se distribuye bajo licencia GNU GPL v3 (GPL3, 2007). Esta herramienta, desarrollada por la Escuela Técnica Superior de Ingeniería Informática de la Universidad Rey Juan Carlos, sólo está disponible para sistemas Windows y Mac. Su código fuente está disponible en Google Code²³, aunque datan de enero de 2011.

Alice presenta dos versiones: Alice 2 y Alice 3, siendo esta última la más nueva y promovida. La figura 2.11 muestra una captura de un programa que realiza una animación sencilla en la versión 2.4. Cabe aclarar que Alice 3 no puede ejecutarse correctamente en computadoras con bajos recursos, por lo que las experiencias realizadas se basan en Alice 2.

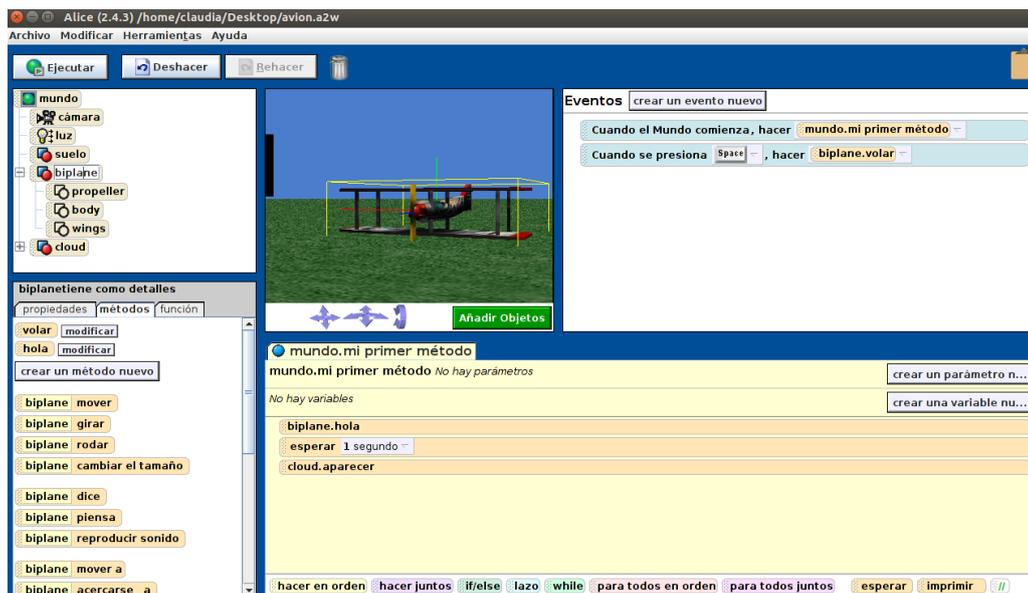


Figura 2.11. Simulación sencilla en Alice 2.4

Alice está pensada para jóvenes de 15 a 18 años. Los escenarios, denominados “mundos” son entornos 3D provistos por la herramienta, en los cuales se pueden agregar distintos objetos. Al igual que Scratch, Alice provee varios escenarios y una gran variedad de objetos listos para ser usados. Pero, existe una gran diferencia entre ambas herramientas: Alice utiliza el paradigma de programación

²²Sitio oficial <http://www.gmr.v.es/rebeca-es/index.html>. Último acceso: noviembre 2017.

²³Sitio de descarga de Rebeca: <https://code.google.com/archive/p/rebeca/downloads>. Último acceso: noviembre 2017.

orientado a objetos²⁴ mientras que Scratch se basa en el paradigma procedural. Cuando se incorpora un nuevo objeto al mundo de Alice éste tiene incorporado un conjunto de propiedades y las acciones que puede realizar a través de métodos y funciones asociadas al mismo.

Alice utiliza bloques textuales para las instrucciones, acercando, de alguna manera, a la notación usada por los lenguajes de programación textuales. La figura 2.12 muestra la instrucción iterativa correspondiente a la sentencia **for** de un lenguaje de programación textual expresada en los bloques de Scratch y Alice y en el lenguaje Python. Como puede notarse, la transición de Alice hacia un lenguaje textual parece bastante adecuada. Los programas escritos en Alice se aproximan más a los escritos en lenguajes de programación textuales, por lo que podría utilizarse como una buena herramienta para realizar la transición de los lenguajes de bloques como Scratch, pensados para estudiantes más jóvenes a un lenguaje de programación como Python, que requiere manejar aspectos sintácticos y más estrictos en la escritura de los programas.

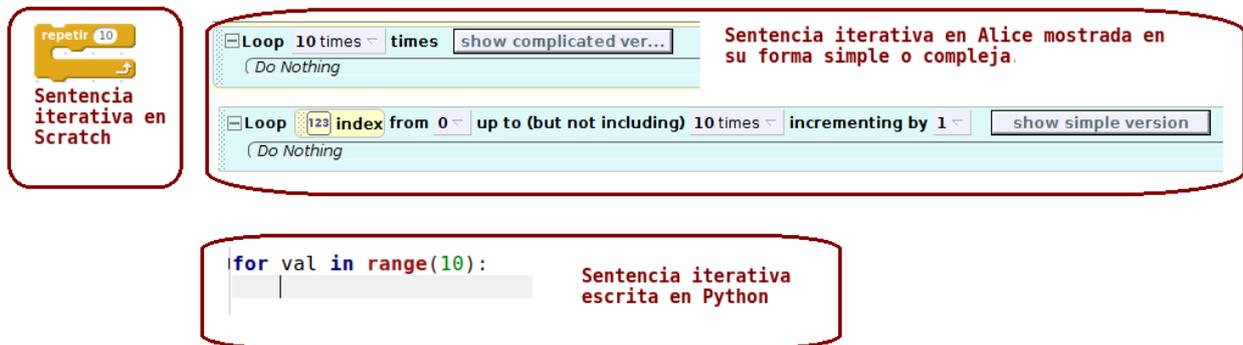


Figura 2.12. Estructura iterativa representada en Scratch, Alice y Python.

2.4.4. Lightbot

Lightbot²⁵ es una herramienta en la cual se proponen distintos desafíos que deben resolverse con un conjunto específico de instrucciones.

Es posible utilizar Lightbot a través de un navegador web, desde su sitio oficial, o desde una aplicación para dispositivos móviles. La figura 2.13 muestra la captura de dos de los niveles propuestos en esta aplicación en su versión móvil y web. Como puede verse ambas versiones presentan una interfaz de usuarios muy similar.

²⁴Este paradigma de programación propone pensar en las soluciones a los problemas identificando los objetos que lo componen y estableciendo las relaciones e interacciones entre los mismos.

²⁵Sitio oficial: <https://lightbot.com/>. Último acceso: noviembre 2017.

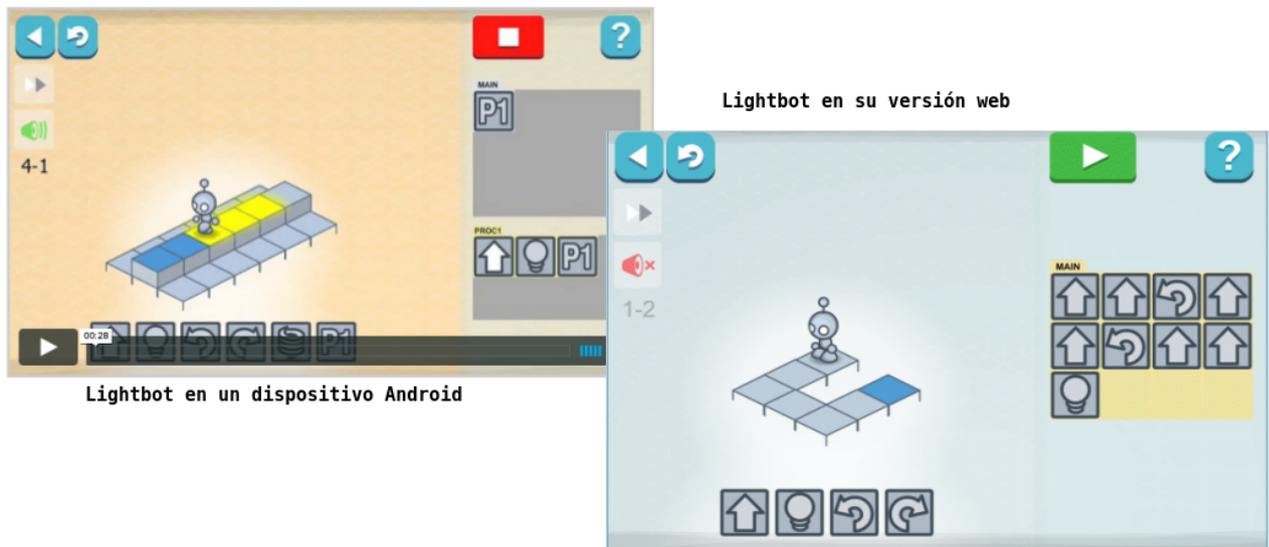


Figura 2.13. Aplicación LightBot en su versión para sistemas Android y web

El objetivo de esta aplicación es construir los programas que permitan al personaje desplazarse por el escenario desde un punto inicial hasta el destino final, respetando las consignas dadas: por ejemplo, encender (colorear) algunos de los casilleros por los que pasa.

La aplicación es muy sencilla y las instrucciones se muestran en una paleta icónica, la cual incrementa la cantidad de instrucciones a medida que los niveles se hacen más complejos. En el primer nivel sólo es posible contar con movimientos de avanzar y girar e iluminar los cuadros, pero luego se agregan más instrucciones tales como saltar o repetir y se trabaja el concepto de abstracción con la formulación de procedimientos.

Esta aplicación no es libre, aunque se puede descargar en forma gratuita en las distintas plataformas. Además, es posible adquirir (comprar) actividades nuevas. En la página oficial se ofrecen actividades para niños de 4 a 8 años y para 9 años en adelante.

2.4.5. Juegos de Blockly

Juegos de Bloclly²⁶ es una herramienta que comprende una serie de juegos educativos para enseñar a programar a niños y jóvenes. Los mismos se encuentran ubicados en una secuencia de acuerdo a su nivel de complejidad, como se muestra en la figura 2.14.

²⁶Sitio oficial: <https://blockly-games.appspot.com>. Último acceso: noviembre 2017.

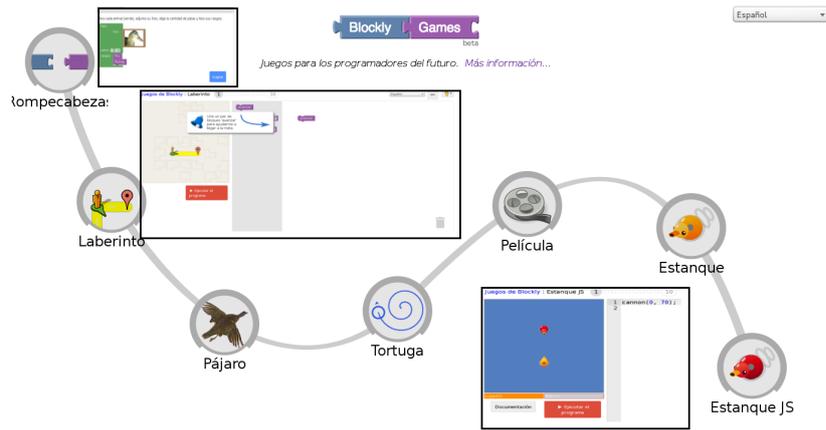


Figura 2.14. Juegos de Blockly

La secuencia de juegos comienza con una actividad sencilla donde, a través del encastre piezas de un *puzzle*, se completan las descripciones de una serie de animales. Luego se incrementa la complejidad introduciendo los distintos constructores a través de diversos juegos hasta culminar con sentencias escritas en el lenguaje Javascript.

La figura 2.15 muestra todos los juegos propuestos que se describen a continuación:

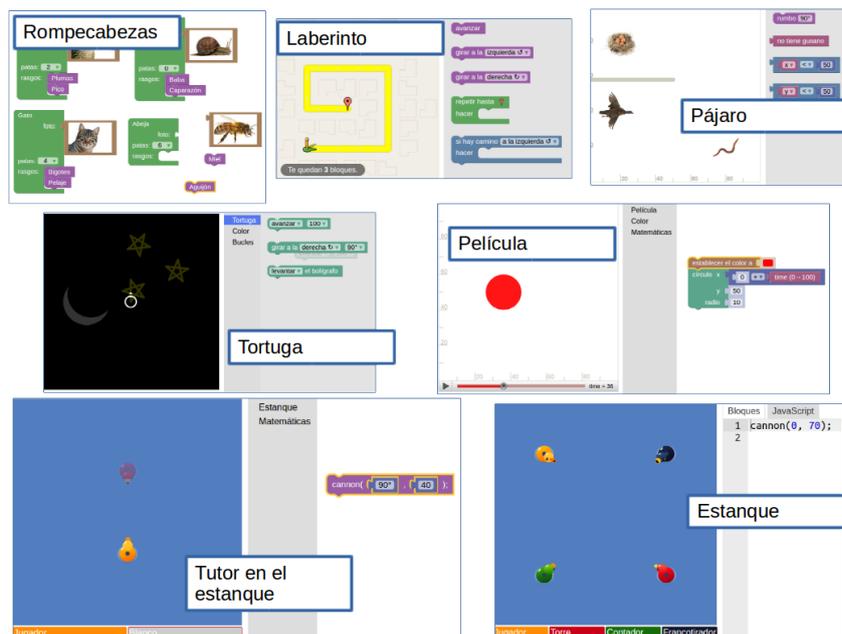


Figura 2.15. Actividades propuestas por Juegos de Blockly

- **Rompecabezas:** presenta una actividad introductoria donde se pueden experimentar con la forma de interacción de las piezas. El objetivo es completar un rompecabezas de animales, asociando cada característica con el animal correspondiente.
- **Laberinto:** moviendo un personaje a través de una grilla se introducen los primeros conceptos de secuencias y estructuras de control repetitivas y condicionales.

- **Pájaro:** en esta actividad se trabaja con mucho más detalle las instrucciones condicionales. Los último niveles son bastantes complejos.
- **Tortuga:** trabajando en forma similar a la tortuga de Logo, esta actividad está orientada a trabajar en profundidad las estructuras iterativas.
- **Película:** en esta actividad se comienza a trabajar con fórmulas matemáticas y expresiones lógicas sencillas.
- **Tutor del estanque:** si bien en las actividades previas, al resolver con éxito un desafío se muestra la solución en el lenguaje Javascript, en esta actividad se hace una aproximación más concreta a dicho lenguaje.
- **Estanque:** esta actividad, similar a la anterior, permite que se programe tanto con el lenguaje de bloques como en Javascript directamente, a través de una solapa que abre un editor de texto muy simple.

Juegos de Blockly muestra, desde las primeras actividades, el código Javascript correspondiente a la solución de cada desafío. Por ejemplo, en el caso de uno de los desafíos de la actividad del Laberinto, la figura 2.16 muestra la solución en el lenguaje de bloques y la correspondiente solución en el lenguaje Javascript.

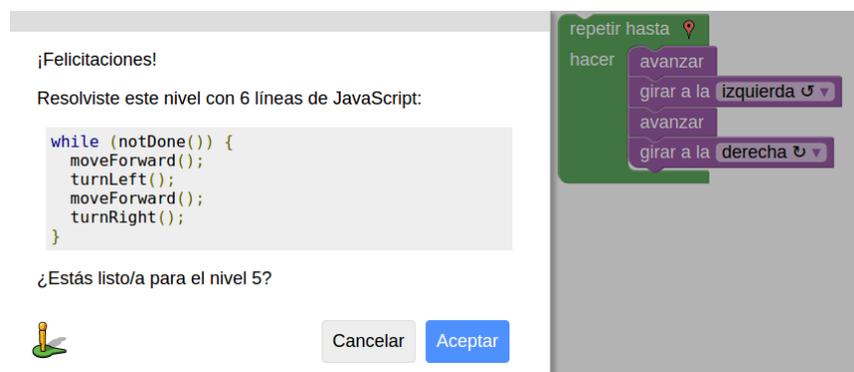


Figura 2.16. Juegos de Blockly: solución en Javascript y en el lenguaje de bloques.

Esta característica permite un acercamiento también a un lenguaje de programación textual, pero complica a la hora de introducir los primeros conceptos, opacando una de las características más importantes de los lenguajes de bloques: la abstracción de la sintaxis de los lenguajes textuales. Al visualizar código en Javascript, aparecen palabras en idioma inglés²⁷ y otros símbolos que pueden

²⁷El análisis que se realiza de todas las herramientas se basa en experiencias o posibilidades de aplicación en Argentina. Por lo tanto, si algunas herramientas, o los mismos lenguajes de programación, introducen términos en idioma inglés, esto presenta una complicación a la hora de trabajar con los niños y jóvenes.

confundir a los niños y jóvenes en un primer momento. Las últimas actividades están orientadas a la programación en este lenguaje y los desafíos son más complejos.

La implementación de Juegos de Blockly utiliza la librería libre Blockly²⁸, y puede utilizarse en forma desconectada descargando su código en su página en GitHub²⁹, posibilitando la utilización de esta herramienta en contextos donde no se dispone de una conexión a Internet.

2.4.6. Aplicaciones de Code.org

Code.org³⁰ es una organización pública sin fines de lucro que ofrece materiales tanto para docentes como alumnos y la comunidad en general que quieran introducirse en el mundo de la programación. Entre los recursos ofrecidos se encuentran cuatro cursos destinados a estudiantes que utilizan varias herramientas basadas, al igual que Juegos de Blockly, en la librería Blockly. Entre estas herramientas se encuentran laberintos, desafíos que proponen dibujar figuras geométricas, animaciones y juegos. Cada uno de estos desafíos se presentan en diversos grados de dificultad, y, en forma incremental, van introduciendo los distintos conceptos y estructuras involucrados en la programación de computadoras.

La figura 2.17 muestra algunas de las actividades propuestas en los distintos cursos destinados a alumnos.

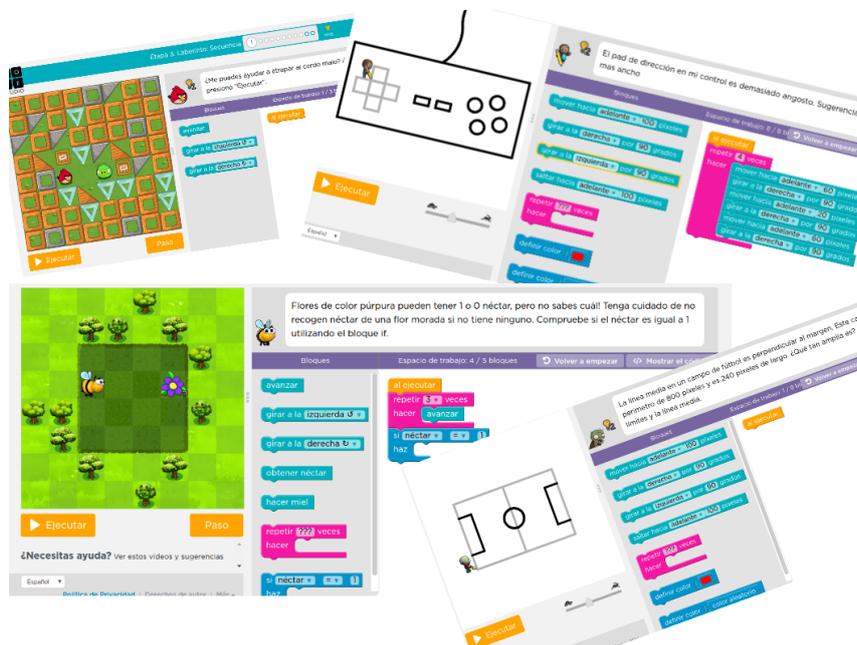


Figura 2.17. Algunas de las actividades propuestas por Code.org

Al igual que los Juegos de Blockly, los desafíos propuestos por esta iniciativa permiten ver el código

²⁸Sitio oficial: <https://developers.google.com/blockly/>. Último acceso: noviembre 2017.

²⁹Sitio de descarga: <https://github.com/google/blockly-games>. Último acceso: noviembre 2017.

³⁰Esta iniciativa se describe en detalles en la sección 2.5.1.1.

Javascript correspondiente a las soluciones, pero no se muestra en forma automática al finalizar un desafío, como ocurre con la herramienta anterior, sino que es posible verlo a través de una solapa independiente en la interfaz de usuario.

Los cursos propuestos para los alumnos presentan no sólo las actividades a desarrollar, sino secuencias didácticas destinadas a los docentes para su uso en el aula. Cabe aclarar que no todas estas guías se encuentran disponibles en idioma español. La tabla 2.2 presenta una síntesis de los cursos disponibles.

Curso	Destinatarios	Descripción
Curso 1	Destinado a niños pequeños que se inician en la lecto-escritura, con edades entre 4 y 6 años.	Propone 18 actividades tanto desconectadas como conectadas usando Internet.
Curso 2	Destinado a alumnos que pueden leer, con edades de 6 años en adelante.	Propone 19 actividades tanto desconectadas como conectadas usando Internet.
Curso 3	Destinado a alumnos entre 8 y 18 años. Es una continuación del curso 2.	Propone 21 actividades tanto desconectadas como conectadas usando Internet.
Curso 4	Destinado a alumnos entre 10 y 18 años. Es una continuación de los cursos 2 y 3.	Propone 22 actividades tanto desconectadas como conectadas usando Internet.

Tabla 2.2. Cursos destinados a estudiantes en la iniciativa code.org.

Algo destacado de esta iniciativa es la posibilidad de realizar el seguimiento de estos cursos. Algunas de las actividades también proveen una autoevaluación.

Si bien su implementación se basa en la librería libre Blockly, las actividades propuestas en los cursos no son libres y la gran mayoría requiere de una conexión a Internet para su uso.

2.4.7. Made with code

Made with code³¹ es un sitio web que presenta varios proyectos a través de los cuales se puede resolver consignas programando con bloques. La figura 2.18 muestra dos ellos: Yeti y Garden Robot.

³¹Sitio oficial del proyecto Made with code: <https://www.madewithcode.com/>. Último acceso: noviembre 2017.

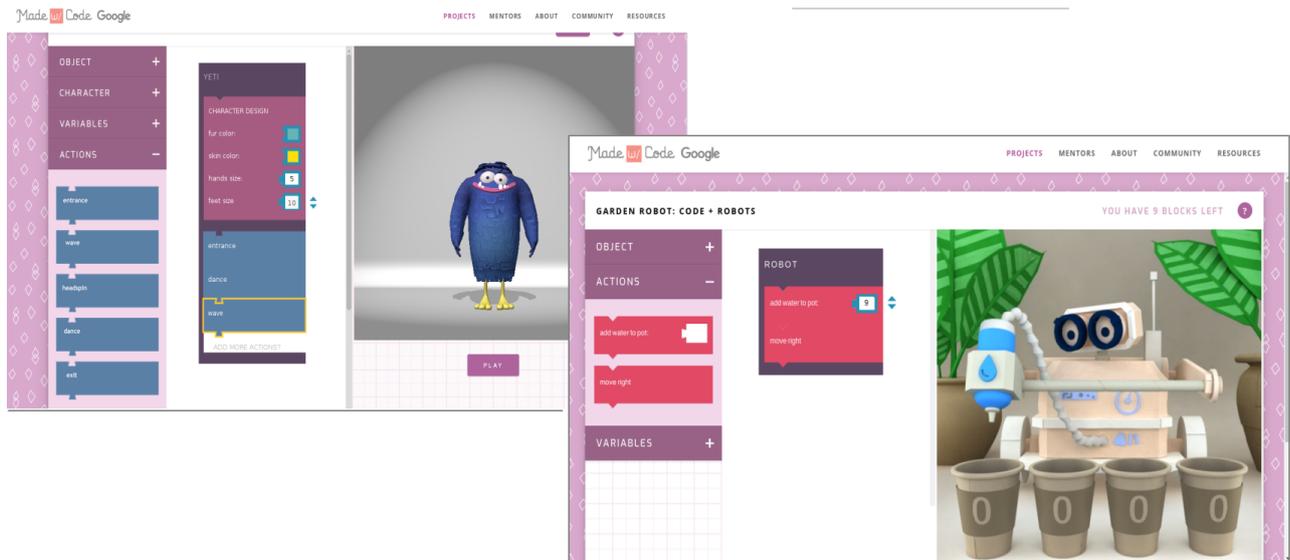


Figura 2.18. Proyectos Yeti y Garden Robot de Made with code.

Los proyectos que incluye son muy interesantes y motivadores, pero tanto su interfaz de usuario como la documentación se encuentra totalmente en idioma inglés. Esto complica su adopción en las aulas de nuestro país.

2.4.8. MIT APPInventor

La herramienta MIT APP Inventor³² propone diseñar y programar aplicaciones para dispositivos móviles Android.

Presenta dos modos de trabajo: diseñador y programación. La figura 2.19 muestra su interfaz de usuario en el modo diseñador y en el modo de programación.

En el modo diseñador, distintos elementos pueden “arrastrarse” a un espacio que simula la pantalla del dispositivo móvil. Luego, en el modo “programación”, estos elementos pueden programarse utilizando un lenguaje basado en bloques similar a los que presentan las herramientas descritas anteriormente.

Las aplicaciones desarrolladas con MIT APP Inventor pueden probarse, depurarse e instalarse directamente en un celular o tableta Android. De una manera muy simple y de esta manera se facilita y promueve la experimentación del desarrollo de aplicaciones móviles.

Además de las instrucciones básicas provistas por las otras herramientas, MIT APP Inventor cuenta con instrucciones que permiten acceder a las funciones propias del dispositivo móvil: tomar fotos con la cámara incorporada en el dispositivo, obtener datos de los sensores del celular como el acelerómetro, el GPS, compartir recursos a través de redes sociales, etc..

³²Sitio oficial: <http://appinventor.mit.edu/>. Último acceso: noviembre 2017.

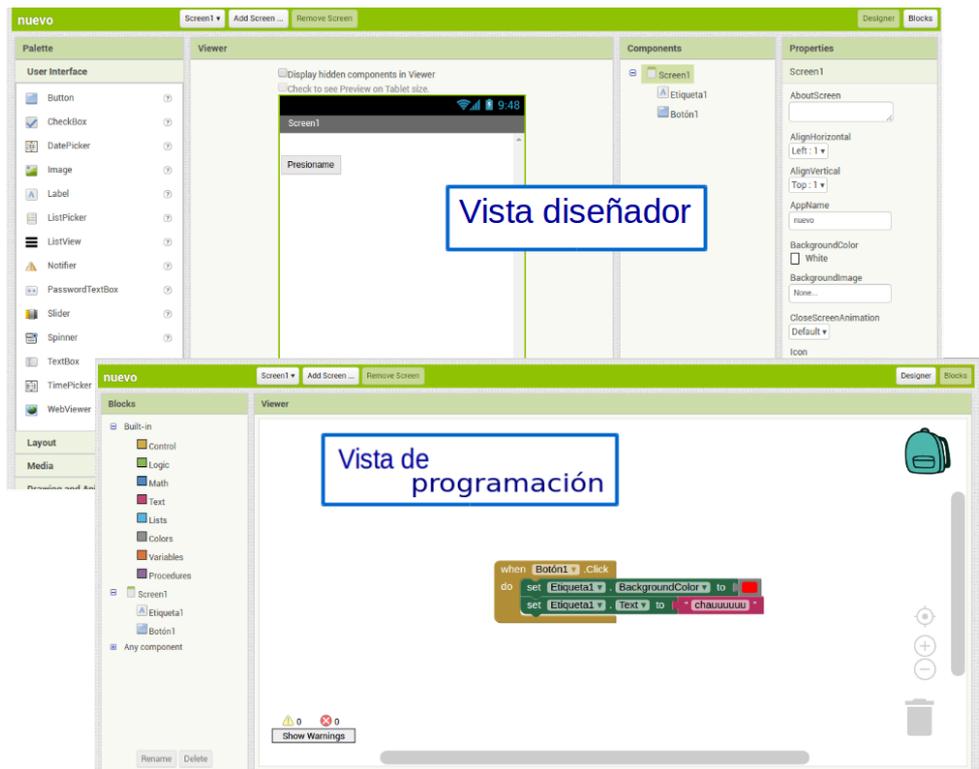


Figura 2.19. Vista de diseñador y vista de programación en MIT AppInventor.

Al igual que en el caso de Scratch, en el sitio oficial de esta herramienta se encuentran disponibles numerosos recursos, entre ellos guías para docentes, aunque cabe aclarar que la gran mayoría de los mismos se encuentran en idioma inglés.

Cabe destacar que para trabajar con MIT AppInventor es necesario contar con una cuenta de Google y conexión a Internet.

2.4.9. Gameblox

Gameblox³³ es una herramienta bastante similar a MIT AppInventor, pero cuyo objetivo es desarrollar juegos para dispositivos móviles iOS y Android.

Gameblox fue desarrollada en el marco del “MIT Scheller Teacher Education Program” y, al igual que Scratch y MIT AppInventor cuenta con un editor en línea a través del cual se diseñan y programan las aplicaciones³⁴.

Si bien se encuentra en idioma inglés es una herramienta muy amigable y con un modo de trabajo muy similar a MIT AppInventor. El editor de aplicaciones presenta también dos modos: diseño y programación.

³³Sitio oficial: <https://gameblox.org/>. Último acceso: noviembre 2017.

³⁴Editor de Gameblox: <https://gameblox.org/editor/>. Último acceso: noviembre 2017.

La figura 2.20 muestra su interfaz de usuario en modo de diseño.

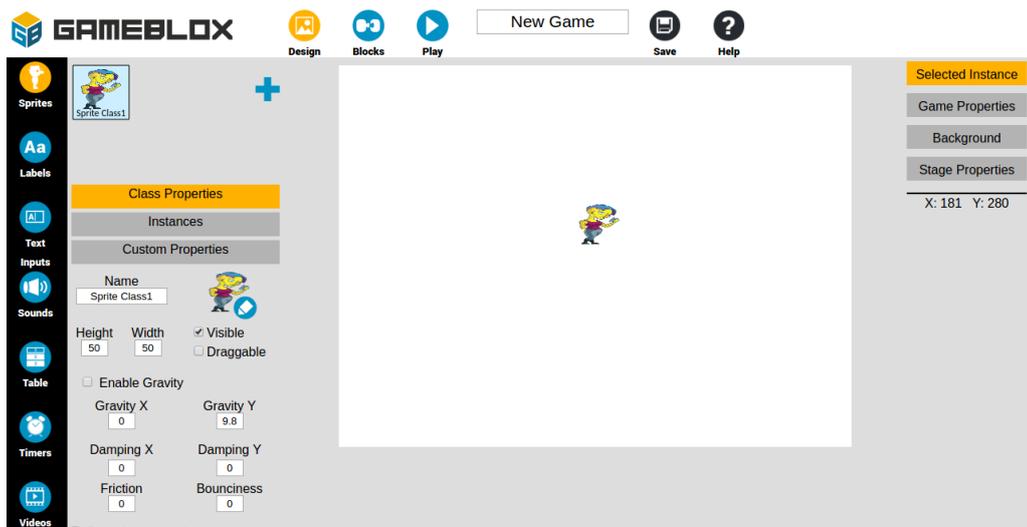


Figura 2.20. Interfaz de usuario de Gameblox en el modo diseño.

2.4.10. pilas-engine

El *framework* pilas-engine³⁵ es un *framework* desarrollado en Argentina por Hugo Ruscitti, pensado para el desarrollo de videojuegos sencillos.

A diferencia de las herramientas mencionadas hasta el momento, la programación en pilas-engine no se realiza utilizando bloques sino que se codifica directamente en el lenguaje de programación Python.

pilas-engine presenta una interfaz totalmente en español y los nombres de los recursos disponibles también se encuentran en nuestro idioma. Esto facilita su uso por parte de alumnos y docentes con pocos conocimientos de programación al trabajar con un lenguaje textual.

pilas-engine puede utilizarse tanto como una librería a ser incluida en los programas Python como cualquier otra librería de este lenguaje, aunque algo importante de destacar es que incluye una aplicación en la cual se cuenta con ejemplos, la interfaz de programación de aplicaciones (API, por sus siglas en inglés *Application Programming Interface*) y el manual completo junto con un intérprete de Python a través del cual es posible realizar las primeras prácticas con el entorno.

La figura 2.21 muestra la interfaz de usuario de la aplicación provista.

³⁵Sitio oficial: <http://pilas-engine.com.ar/>. Último acceso: noviembre 2017.



Figura 2.21. Interfaz de usuario de la aplicación provista por pilas-engine.

Como puede verse, esta aplicación provee:

- **Intérprete (A):** se trata de un intérprete de comandos en el cual se pueden ingresar las distintas instrucciones y visualizar en forma inmediata el resultado de las mismas.
- **Editor (B):** utilizando este editor sencillo es posible escribir los programas y luego ejecutarlos y/o guardarlos. En este caso el resultado también puede ser visualizado en el mismo contexto de pilas-engine.
- **Ejemplos (C):** una de las características más importantes de este *framework* es que provee numerosos ejemplos de distinta complejidad los cuales no sólo muestran su funcionamiento sino también su código fuente. Esto permite explorar la forma en la cual dichos ejemplos han sido programados.
- **Manual y Documentación sobre la API:** si bien ambos recursos se pueden consultar en el sitio web oficial de la herramienta, al descargarla en una computadora se encuentran disponibles en forma desconectada.
- **Aspectos básicos de configuración (D):** permite configurar algunos aspectos tales como la habilitar los *joystick* y/o el sistema de audio.

La figura 2.22 muestra algunos de los recursos provistos por este *framework* descriptos anteriormente.

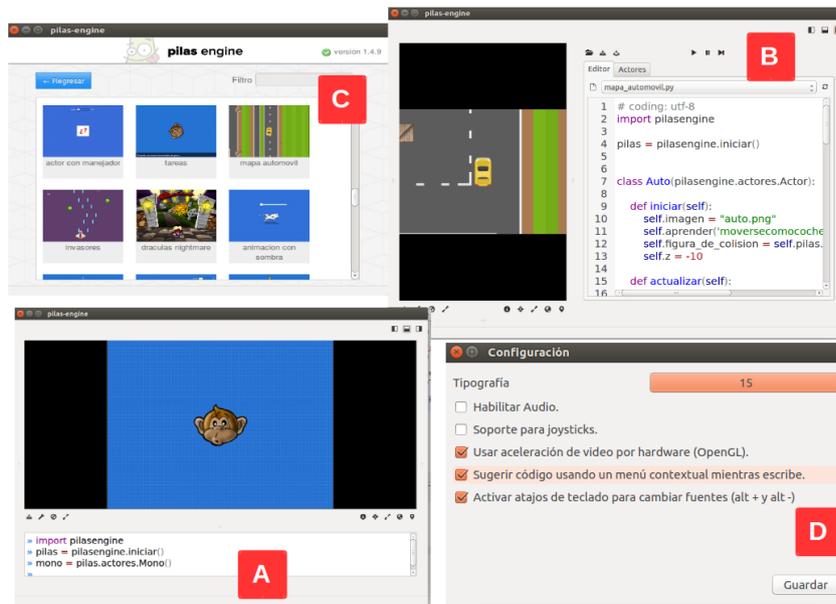


Figura 2.22. Recursos provistos por pilas-engine.

pilas-engine provee una biblioteca de recursos disponibles que pueden utilizarse libremente al tratarse de una herramienta publicada bajo una licencia libre. Entre estos recursos pueden mencionarse actores, habilidades y elementos de la interfaz de usuario. Los actores representan a los objetos que se incluyen en el videojuego, por ejemplo una nave, una pelota o un mono, los cuales ya cuentan con habilidades tales como disparar en el caso de la nave, rebotar, en el caso de la pelota o gritar en el caso del mono.

La figura 2.23 muestra una captura de un código escrito en Python en el editor provisto por pilas-engine, en el cual se crea una nave y una bomba. Como puede verse los nombres de los recursos se encuentran en español y el editor tiene una facilidad de autocompletado que asiste en la edición.

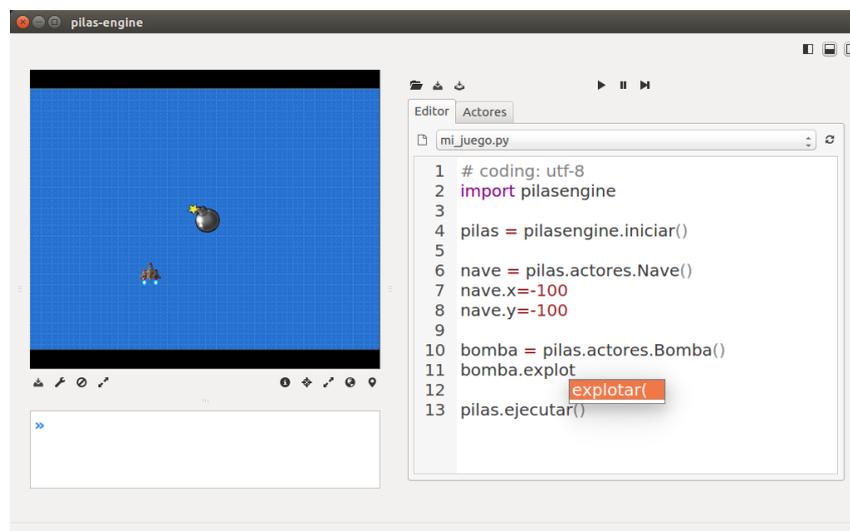


Figura 2.23. Ejemplo de código que utiliza pilas-engine.

2.4.11. Pilas Bloques

Pilas Bloques³⁶ es una herramienta desarrollado por la Fundación Sadosky, que propone varios desafíos agrupados didácticamente.

Es posible utilizar Pilas Bloques desde su sitio oficial o descargar una versión trabajar en forma desconectada³⁷.

La figura 2.24 muestra una captura del sitio oficial con algunos de los desafíos propuestos.

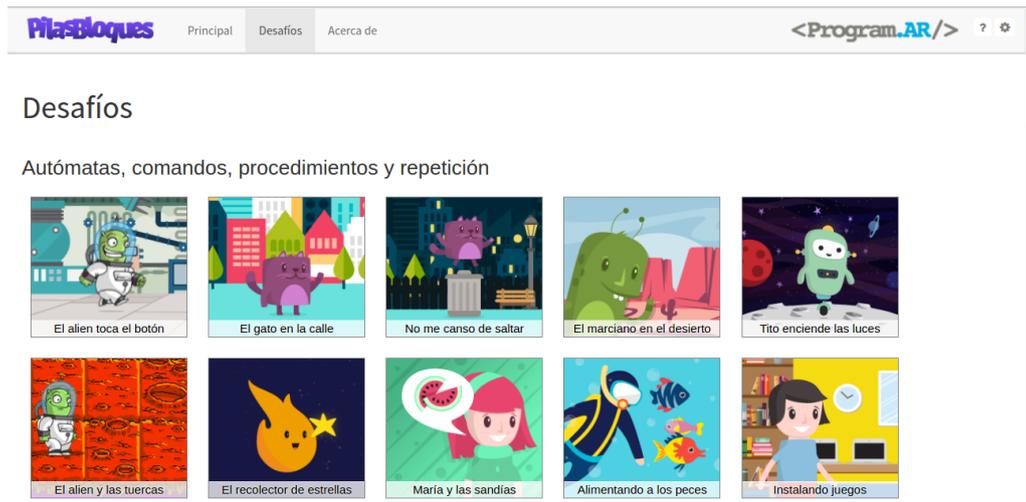


Figura 2.24. Pilas Bloques: sitio oficial con los desafíos propuestos.

Cada desafío está acompañado de una guía donde se menciona el objetivo, y las sugerencias para su resolución. La figura 2.25 muestra una captura del desafío “El recolector de estrellas”. Como puede verse, la actividad cuenta con una breve descripción y una sección “¿Sabías que?” en la cual se presentan guías y sugerencias para la resolución.

A partir de las últimas versiones de esta herramienta es posible guardar y recuperar las soluciones a los desafíos. Estas soluciones se almacenan en un dispositivo de almacenamiento local.

Los desafíos de Pilas Bloques se encuentran clasificados en distintas categorías que, en forma incremental van introduciendo los distintos conceptos de programación. Las categorías establecidas al momento de cerrar este informe son:

- **Autómatas, comandos, procedimientos y repetición:** esta serie de doce (12) desafíos proponen actividades que incorporan las nociones de secuencia, repeticiones y procedimientos como una forma de crear nuevas instrucciones. Esto ultimo permite trabajar aspectos de abstracción procedural.

³⁶Sitio oficial: <http://pilasbloques.program.ar/>. Último acceso: noviembre 2017.

³⁷Al momento de cerrar este informe se encuentran disponibles versiones para los sistemas Microsoft Windoes, Huayra Linux y Mac OS

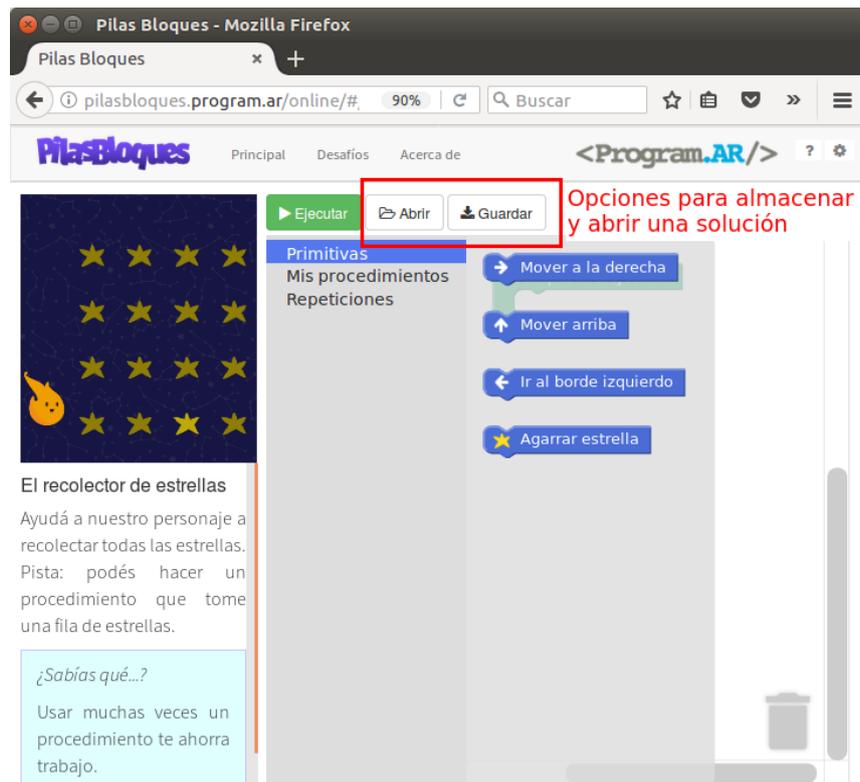


Figura 2.25. “El recolector de estrellas”: desafío en el cual se introducen las estructuras repetitivas.

- **Alternativa condicional:** esta sección propone seis (6) desafíos en los que introducen las estructuras condicionales simples y compuestas.
- **Repetición condicional:** en siete (7) desafíos se trabaja la estructura de repetición condicional. Esta estructura se presenta como “Repetir hasta...”.
- **Sensores numéricos:** en esta sección, que consta de dos (2) desafíos se introduce el concepto de variable.
- **Parametrización de soluciones:** esta sección propone dieciséis (16) actividades en donde se trabajan procedimientos con parámetros.

Pilas Bloques es una herramienta libre y es posible descargar su código fuente desde Github³⁸.

2.4.12. CodeCombat y CodinGame

Estas dos herramientas permiten aprender distintos lenguajes de programación textuales a través de juegos. En ambos casos no se brindan tutoriales o cursos, sino que se espera que las características del lenguaje se vayan aprendiendo a medida que se avanza en el juego.

³⁸<https://github.com/Program-AR/pilas-bloques>. Último acceso: noviembre 2017.

CodeCombat³⁹ es un proyecto de código abierto en el cual, a través de un juego de desafíos, se aprende lenguajes de programación textuales tales como Python, Javascript o Lua⁴⁰. El usuario elige cuál es el lenguaje de programación por defecto que quiere utilizar de esta lista reducida.

Esta herramienta brinda la posibilidad de registrarse como alumno, docente o un individuo sin ningún rol en particular. La opción para docentes, permite armar clases en base a este juego, crear un aula virtual invitando a sus alumnos a participar y visualizar el progreso de los mismos.

La figura 2.26 muestra una captura del desarrollo del desafío inicial en CodeCombat.



Figura 2.26. Captura del nivel más sencillo de CodeCombat.

Si bien la página principal se encuentra en idioma español, los desafíos no están totalmente traducidos, al igual que los cursos que provee. Tratándose de una iniciativa de código abierto esto puede adecuarse para adaptar esta plataforma a iniciativas locales.

De la misma forma que CodeCombat, **CodinGame**⁴¹ permite aprender a programar en un lenguaje de programación textual a través de un juego de desafíos. A través de la resolución de estos desafíos los usuarios pueden plantear soluciones en más veinte lenguajes de programación diferentes. Por cada uno de ellos CodinGame brinda un código predeterminado para trabajar, el cual debe ser modificado para que funcione correctamente.

La figura 2.27 muestra el punto inicial para comenzar con el desafío “*Power of Thor*”. Se presenta una breve descripción y un pequeño tutorial y luego los desafíos según los niveles de complejidad.

Algo de destacar de ambas iniciativas es la forma en la cual se intenta capturar la atención de los usuarios: resolver desafíos a través de juegos. La diferencia más destacada de ambas es que CodeCombat es una iniciativa de código abierto mientras que CondinGame no lo es.

³⁹Sitio oficial: <https://codecombat.com/>. Último acceso: noviembre 2017.

⁴⁰Lua es un lenguaje de programación multiplataforma creado en el año 1993 en la Pontificia Universidad Católica de Río de Janeiro.

⁴¹Sitio oficial: <https://www.codingame.com>. Último acceso: noviembre 2017.

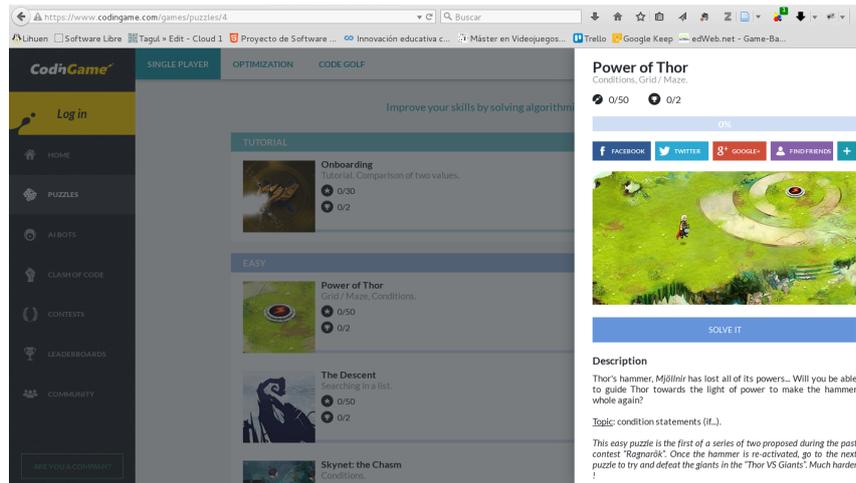


Figura 2.27. Desafío “Power of Thor” de CodinGame.

2.4.13. PSeInt

PSeInt⁴² es una herramienta utilizada principalmente por docentes de escuelas secundarias que permite escribir programas en un pseudocódigo en español y luego ejecutarlos. La figura 2.28 muestra un ejemplo del mismo.

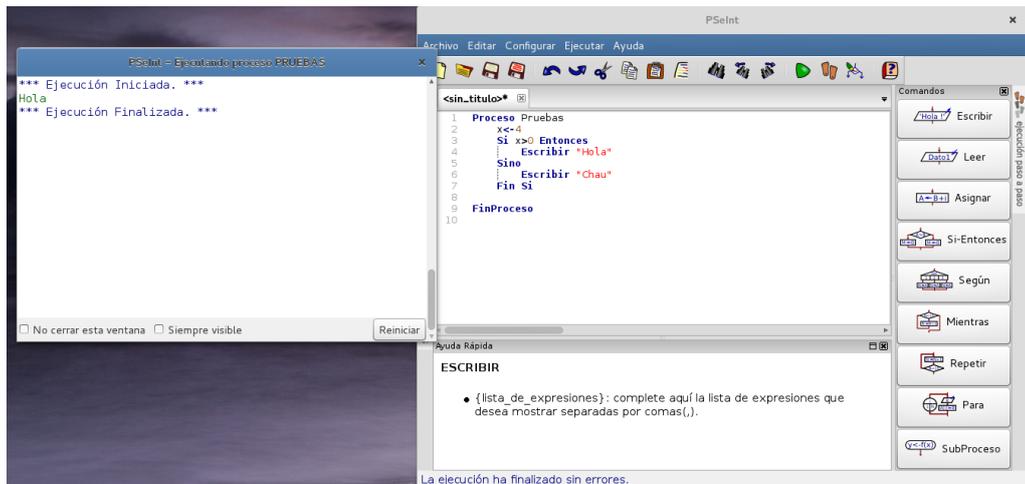


Figura 2.28. PSeInt

La característica más destacada de esta herramienta es que está totalmente en español, algo muy interesante para nuestro país. Aunque esto podría también ser visto como una desventaja, dado que si se quiere escalar a un lenguaje de programación real, los mismos proveen sus palabras reservadas en inglés. Si bien esto no es una gran impedimento, es una las críticas y cuestionamientos más mencionados en todas las experiencias realizadas con estudiantes y docentes a lo largo de los años. Las últimas versiones incluyen un módulo que permite exportar el código desarrollado a lenguajes de programación reales tales como C, Python o Pascal.

⁴²Sitio de descarga: <http://pseint.sourceforge.net/>. Último acceso: noviembre 2017.

2.4.14. Otras herramientas destacadas

En los últimos tiempos han surgido muchos productos destinados a trabajar aspectos de programación con niños y jóvenes. Muchos de ellos requieren de un *hardware* específico o de otros dispositivos tangibles para su utilización. A continuación se mencionan algunos de ellos:

- **Kano**⁴³: es un *kit* basado en Raspberry PI⁴⁴ para construir y programar su propia computadora. Está pensado para niños pequeños y provee un sistema operativo propio (Kano OS) y herramientas basadas en codificación en bloque similar a la interfaz provista por las herramientas anteriormente descritas.
- **LEGO®**: es uno de los pioneros en proponer opciones de robótica educativa. LEGO® ofrece varios robots que pueden construirse y luego programarse. Al momento de cerrar este informe el modelo ofrecido es el Mindstorm EV3⁴⁵ el cual puede programarse usando una aplicación para varias plataformas. Algo de destacar de esta oferta es que los productos ofrecidos, si bien son sólidos, muy atractivos y poseen mucha documentación y recursos de soporte son privativos y muy costosos.
- **Google Bloks**⁴⁶: es un proyecto de investigación destinado a niños que promueve el pensamiento computacional a través de la “programación tangible”, es decir que utiliza elementos que se pueden ver y tocar para armar los programas.
- **Bee-Bot**⁴⁷: se trata de una abeja-robot destinada a niños pequeños la cual puede programarse para que se desplace por el suelo. Es un producto que se adquiere acompañado de materiales pensados para docentes y padres que desean incorporar aspectos de programación y resolución de problemas desde edades muy tempranas. Si bien se trata de un recurso totalmente cerrado y privativo es interesante destacar.

2.4.15. Desarrollos de la Facultad de Informática de la UNLP

Esta sección presenta algunas herramientas desarrolladas en la Facultad de Informática de la UNLP en las cuales esta tesista estuvo involucrada en su desarrollo. Algunas son el resultado de trabajos finales de carrera y otras implementadas en la cátedra “Seminario de Lenguajes- opción Lenguaje Python”, del cual esta tesista es parte del plantel docente.

⁴³Sitio oficial de Kano: <http://us.kano.me/>. Último acceso: noviembre 2017.

⁴⁴Sitio oficial: <https://www.raspberrypi.org/>. Último acceso: noviembre 2017.

⁴⁵<https://www.lego.com/es-ar/mindstorms/products/mindstorms-ev3-31313>. Último acceso: noviembre 2017.

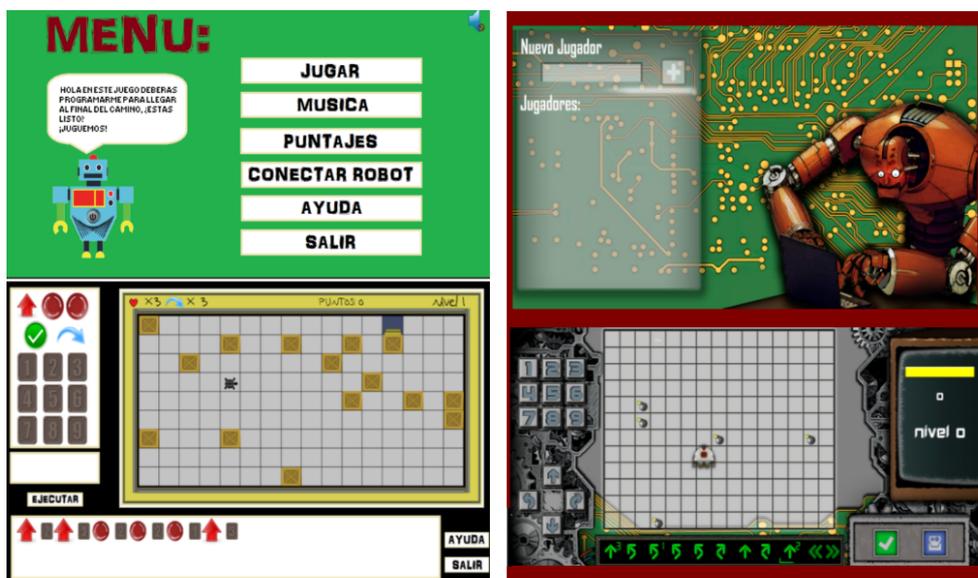
⁴⁶<https://projectbloks.withgoogle.com/>. Último acceso: noviembre 2017.

⁴⁷Sitio oficial: <https://www.bee-bot.us/>. Último acceso: noviembre 2017.

2.4.15.1. MiRobot y Zedpy

Dado que la aplicación Lightbot no es libre, aunque es una herramienta didáctica muy buena para el trabajo con niños pequeños, en la asignatura “Seminario de Lenguajes - opción Lenguaje Python”, durante el ciclo lectivo 2015, se propuso el desarrollo de una versión reducida de la misma. Los trabajos se desarrollaron usando pilas-engine para los sistemas operativos Microsoft Windows y Linux.

De los trabajos presentados por los alumnos, se destacan los realizados por los alumnos Sebastián Muiña y Cristian Steib al cual denominaron Zedpy y el trabajo de los alumnos Agustín De Luca y Francisco Segarra, denominado MiRobot, mostrados en la figura 2.29



MiRobot: aplicación desarrollada por los alumnos Agustín De Luca y Francisco Segarra en la asignatura Seminario de Lenguajes - Opción Python

Zedpy: aplicación desarrollada por los alumnos Sebastián Muiña y Cristian Steib en la asignatura Seminario de Lenguajes - Opción Python

Figura 2.29. Aplicaciones desarrolladas en la asignatura Seminario de Lenguajes - Opción Python

Ambos trabajos presentan un escenario en el cual un pequeño robot debe moverse para llegar a un objetivo. En el caso de Zedpy, utiliza una API (“*Application Programming Interface*”) desarrollada en la facultad que permite la conexión con los robots Múltiple N6 utilizados en el proyecto “Programando con robots y software libre”⁴⁸. Esto permite que, activando la opción correspondiente, el robot real reproduzca los movimientos hechos por el robot virtual. La aplicación MiRobot utiliza un personaje similar a la tortuga de Logo y no se conecta con el robot real.

Ambos trabajos fueron presentados en el III Congreso Internacional de Videojuegos y Educación realizado en la Universidad Nacional de Tres de Febrero en agosto de 2015 (Banchoff C., Harari V.

⁴⁸Este proyecto, al igual que los desarrollos realizados en el marco del mismo se describen en detalle en el capítulo 4

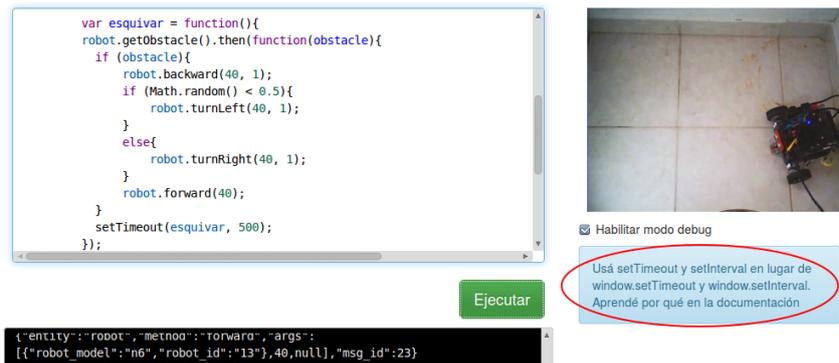


Figura 2.30. XRemoteBot: aplicación que permite programar los robots del proyecto “Programando con robots y software libre” en forma remota.

& Martin S., 2015). Zedpy también fue presentado en el evento “Programando Robots con Software Libre” en el Planetario Ciudad de La Plata en octubre de 2015 al cual asistieron docentes y alumnos de escuelas secundarias de la región de La Plata y alrededores.

2.4.15.2 XRemoteBot

XRemoteBot⁴⁹ es una herramienta de código abierto que desarrolló el Lic. Fernando López en el marco de su trabajo final de carrera (López F., 2015) en el año 2015. Esta herramienta permite trabajar con los robots del proyecto “Programando con robots y software libre” en forma remota a través de Internet. Con XRemoteBot se pueden escribir programas en Python, Ruby⁵⁰ y Javascript, y visualizar la ejecución de los mismos a través de una interfaz web por medio de *streaming* de video.

La figura 2.30 muestra la interfaz de usuario de la aplicación web en la cual se usa Javascript como lenguaje de programación.

Esta herramienta se pensó, principalmente, para el trabajo con aquellas instituciones que no cuentan con los robots y sabiendo que un simulador no es suficiente como elemento de motivación.

2.4.15.3 Dropsy

Dropsy⁵¹ es una herramienta de código abierto creada en el año 2016 por los alumnos Matías Fuentes y Diego Fernández en el marco de su tesina de licenciatura “*Dropsy: eDucational RObot Programming SYstem*”. Esta tesina, defendida en diciembre de 2016, se centró en el desarrollo de una herramienta basada en Blockly para ser usada en dispositivos móviles con sistemas Android. Los programas realizados en Dropsy permiten manipular los robots utilizados en el proyecto “Programando

⁴⁹Sitio de descarga: <https://github.com/fernandolopez?tab=repositories>. Último acceso: noviembre 2017.

⁵⁰Ruby es un lenguaje de código abierto publicado en el año 1995. Más información en <https://www.ruby-lang.org/es/>. Último acceso: noviembre 2017.

⁵¹Sitio de descargar: <https://github.com/dropsy-unlp>. Último acceso: noviembre 2017.

con robots y software libre” en forma remota. Surge como un complemento a XRemoteBot agregando una interfaz de programación mediante bloques para dispositivos móviles.

La figura 2.31 muestra una captura de su interfaz de usuario.

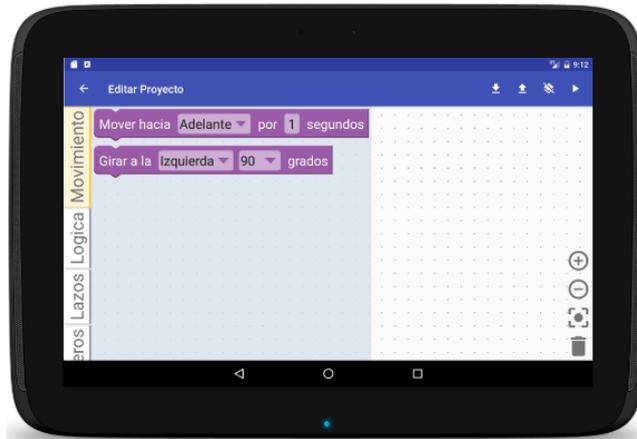


Figura 2.31. Drosy: eEducational RObot Programming SYstem.

2.4.15.4 RITA

RITA⁵², “*Robot Inventor to Teach Algorithms*”, nace como una herramienta didáctica desarrollada por Vanesa Aybar Rosales, en el marco de su tesina de grado en la Facultad de Informática en el año 2010. A través de una interfaz basada en bloques es posible programar las estrategias de unos robots sencillos, que luego se enfrentan en un campo de batalla. RITA integra dos *frameworks* de código fuente abierto: OpenBlocks⁵³ y Robocode⁵⁴. Con OpenBlock se diseñan y escriben los algoritmos que corresponden a las estrategias de batallas y Robocode permite la ejecución de esos algoritmos y se puede observar el comportamiento programado en un formato gráfico.

En sus orígenes, RITA fue pensada para brindar una primera aproximación al lenguaje de programación Java, pero luego de las experiencias en las escuelas, sólo se utiliza la interfaz basada en bloques.

La figura 2.32 muestra a captura de us interfaz de usuario.

⁵²Sitio de descarga: <https://github.com/vaybar/RITA>. Último acceso: noviembre 2017.

⁵³Sitio oficial: <http://education.mit.edu/openblocks>

⁵⁴Sitio oficial: <http://robocode.sourceforge.net>

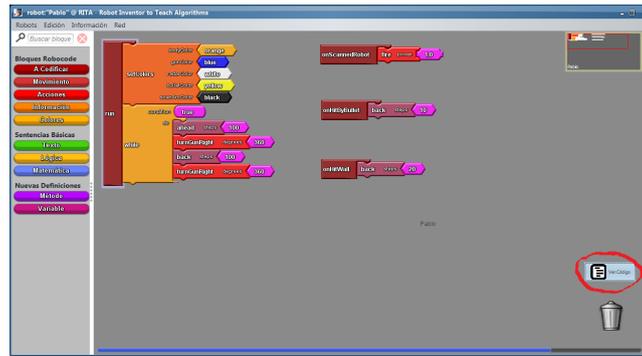


Figura 2.32. Interfaz de usuario de Rita.

En el año 2015 se aprobó una segunda tesina de grado la cual plantea una extensión de RITA. La herramienta generada, denominada RITA en Red, agrega como funcionalidad adicional a la herramienta estándar la posibilidad del trabajo en red, donde los alumnos pueden diseñar y compartir sus robots entre ellos. RITA en red fue desarrollada por los alumnos Pablo Villatella y Santiago M. Rugna.

2.4.16. Tabla comparativa de herramientas

Dada la gran cantidad de herramientas existentes, se elaboraron, a modo de resumen, dos tablas comparativas en las cuales se presenta un panorama general de las características más destacadas de las herramientas descriptas en este capítulo.

Cabe destacar que en este resumen sólo se presentan aquellas herramientas que se utilizan o podrían ser utilizadas en el aula de una escuela argentina, ya sea por el idioma o por los recursos involucrados, teniendo en cuenta las prestaciones provistas por cada una.

La tabla 2.3 agrupa aquellas herramientas pensadas sólo para estudiantes de nivel secundario.

Nombre de la aplicación / Criterio de comparación	MIT APPIn- ventor	Alice	RITA	Pilas Engine	Code- Combat	Xremo- teBot/- Dropsy
Aplicación móvil	NO	NO	NO	NO	NO	SI
Uso desconectado	NO	SI	SI	SI	NO	NO
Uso en línea	SI	NO	NO	NO	Si	Si
Lenguaje asado en bloques	SI	SI	SI	NO	NO	Alguna versión
Lenguaje textual	NO	NO	NO	SI	SI	Alguna versión
Incluye guías para docentes	SI	SI	SI	SI	SI	SI
Incluye sólo desafíos a resolver	NO	NO	NO	NO	NO	NO
Es <i>software</i> libre	NO	NO	SI	SI	SI	SI
Requiere <i>hardware</i> específico	NO	NO	NO	NO	NO	SI
Interfaz en español	SI	Parcial- mente	Parcial- mente	SI	SI	SI
Documentación en español	SI	Muy poco	SI	SI	SI	SI

Tabla 2.3. Tabla comparativa de herramientas que pueden ser usadas con alumnos de nivel secundario.

La tabla 2.4 agrupa aquellas herramientas que pueden ser destinadas a alumnos de educación primaria tanto como secundaria.

Nombre de la aplicación / Criterio de comparación	Scratch	Lightbot	Code.org	Juegos de Blockly	Pilas Bloques
Aplicación móvil	NO	Alguna versión	NO	NO	NO
Uso desconectado	Alguna versión	Alguna versión	Algunos desafíos	SI	SI
Uso en línea	SI	SI	SI	SI	Si
Lenguaje de programación basado en bloques	SI	SI	SI	SI	SI
Lenguaje de programación textual	NO	NO	NO	Algunos desafíos	NO
Incluye guías para docentes	SI	NO	SI	NO	SI
Posee versión para niños pequeños	Alguna versión	NO	El curso 1	NO	NO
Incluye sólo desafíos a resolver	NO	SI	SI	SI	SI
Es <i>software</i> libre	Alguna versión	NO	NO	SI	SI
Requiere <i>hardware</i> específico	NO	NO	NO	NO	NO
Interfaz en idioma español	SI	SI	Algunos desafíos	SI	SI
Documentación en idioma español	SI	NO	Algunos desafíos	NO	SI

Tabla 2.4. Tabla comparativa de herramientas que pueden ser usadas con alumnos de nivel primario y secundario.

2.5. Iniciativas para enseñar a programar en la escuela

Como se mencionó anteriormente, desde hace algunos años existen varias iniciativas que fomentan la inserción de las ciencias de computación o la informática, como disciplina, en las escuelas. El objetivo principal de estas iniciativas es promover que niños y jóvenes conozcan los fundamentos de las ciencias

de la computación y, en particular, se inicien en el mundo de la programación.

Para motivar y lograr un mayor interés en esta temática, considerada tradicionalmente como “aburrida” o “para *nerds*”, se comenzó una campaña de “desmitificación” de las ciencias de la computación promoviendo actividades lúdico-sociales que muestran no sólo que programar puede ser interesante y divertido, sino que se puede integrar a múltiples áreas disciplinares.

Las propuestas de incluir el estudio de las ciencias de la computación, y de la programación como parte de ésta, en la escuela permite enseñar en qué se basa la tecnología que se utiliza en las actividades cotidianas, de manera que los estudiantes puedan tomar control de dicha tecnología y dejen de ser simples usuarios de las mismas. La programación tiene un papel fundamental en esto, pero no es lo único. Saber cómo funciona Internet, cómo las computadoras ejecutan los programas y cómo se puede utilizar distintas aplicaciones en forma segura también es parte fundamental de esta formación.

En esta sección, se presentan las iniciativas más destacadas, tanto a nivel internacional como nacional describiendo las características generales de cada una de ellas. En el plano nacional, se presentan con especial foco aquellas iniciativas que se llevan a cabo en la Facultad de Informática de la Universidad Nacional de La Plata en las que esta tesista está involucrada.

La figura 2.33 muestra un mapa conceptual que resume las distintas iniciativas que se presentarán en esta sección y las características más destacadas de cada una, destacando el proyecto “Programando con robots y software libre” que da marco a esta tesis.

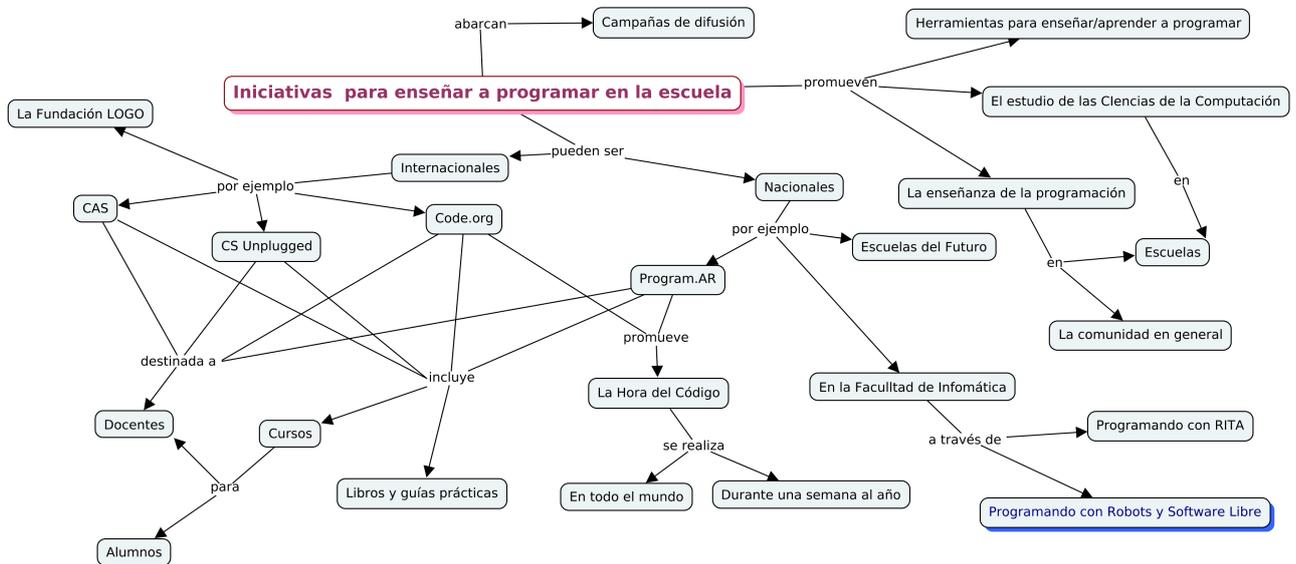


Figura 2.33. Mapa conceptual donde se resumen las iniciativas descriptas.

2.5.1. Iniciativas internacionales

Países tales como Israel, Nueva Zelanda, Japón, Finlandia y Reino Unido cuentan con iniciativas que promueven la incorporación de la enseñanza de las ciencias de computación en las escuelas, en particular sobre contenidos de programación. En el año 2013, el reporte generado por los grupos “*Informatics Europe & ACM Europe Working Group*” (Informatics Europe and ACM Europe, 2013) dio lugar a varios análisis del estado del arte en diferentes países de Europa. Walter Gander en (Gander, 2015) recalca que lo primero que se deba hacer es identificar correctamente los términos usados. En particular, menciona que las ciencias de la computación en las escuelas involucran dos conceptos: la alfabetización digital y la informática en sí misma. Donde la alfabetización digital abarca obtener la fluidez en el uso de las herramientas informáticas y de Internet; y la informática cubre la ciencia detrás de la tecnología de la información. Esta ciencia, caracterizada por sus propios conceptos, métodos, contenidos, etc., ha tomado un papel similar al de las matemáticas, en un campo interdisciplinar subyacente tanto a nivel científico como en el área de la ingeniería y el progreso económico.

Por lo tanto, la incorporación de las ciencias de la computación en las currículas escolares (en especial, aspectos de programación) es una iniciativa mundial. En la sociedad actual, donde habilidades como la creatividad e ingenio para resolver problemas son muy requeridas, estas iniciativas han cobrado mayor notoriedad y difusión.

En las siguientes secciones se describen algunas de las iniciativas más destacadas, comenzando por la que sin lugar a dudas es la más conocida: “Code.org”.

2.5.1.1. Code.org y “La Hora del Código”

Como se mencionó anteriormente, Code.org ⁵⁵ es una organización pública sin fines de lucro apoyada por diversos grupos y fundaciones a nivel internacional cuyo objetivo principal es promover en estudiantes de todas las edades el aprendizaje de las ciencias de computación y en especial, la programación, a través de actividades de acceso libre y gratuitas.

Los cursos en línea brindados por Code.org se encuentran en la plataforma **Code Studio** ⁵⁶, y fueron descritos en este informe en la sección 2.5.1.1.

Una de las actividades principales que promueve Code.org es “**La Hora del Código**”⁵⁷ una campaña en la que participan más de 180 países. En su sitio web, es posible visualizar un mapa, reproducido en la figura 2.34, donde aparecen los eventos registrados. Como puede verse, hay numerosos

⁵⁵En secciones anteriores ya se ha mencionado esta iniciativa, describiendo los recursos ofrecidos.

⁵⁶Sitio oficial: <https://studio.code.org>. Último acceso: noviembre 2017.

⁵⁷Sitio oficial: <https://hourofcode.com/es>. Último acceso: noviembre 2017.

eventos en muchos países y algo interesante de destacar es que, si bien el foco son los niños y jóvenes, no hay restricciones de edades para los participantes.



Figura 2.34. Eventos registrados en “La hora de código” hasta el mes de enero 2016

En realidad, es posible organizar “La Hora del Código” en cualquier momento del año, pero el objetivo fundamental de esta campaña es que se realicen en una semana en particular, para que el impacto sea mayor. En el caso del año 2015, la misma se realizó entre el 7 y el 13 de diciembre, en celebración de la “Semana Educativa de las Ciencias de la Computación”. Aunque la fecha de realización no es tan apropiada para la agenda escolar argentina, debido a que en dichas fechas las clases están culminando en todos los niveles, se respetó la fecha internacional. Durante los ciclos lectivos 2016 y 2017, esta actividad, en Latinoamérica, se realizó en la primer semana de octubre, siendo ésta una fecha mucho más adecuada a las actividades escolares de la región.

No existe una hora específica para realizar este evento, se puede celebrar “La hora de código” en cualquier momento de esa semana, o, en realidad cuando sea posible.

En cada país se busca difundir esta actividad a través de personalidades destacadas y conocidas, principalmente por los niños y jóvenes. En el caso de Argentina algunos de los encargados de la difusión fueron el empresario y conductor Mario Pergolini, el periodista y matemático Adrián Paenza y el basquetbolista Emanuel Ginóbili, quienes participaron de un video promocional⁵⁸, destacando la importancia de esta temática a nivel general e intentando eliminar la imagen de que la programación es sólo para “nerds”.

⁵⁸Video promocional realizado por la iniciativa Program.AR: <https://www.youtube.com/watch?v=HrBh2165KjE>. Último acceso: noviembre 2017.

2.5.1.2. CAS: Computing at School

CAS⁵⁹ es una comunidad en la cual participan docentes de distintos países, siendo la gran mayoría del Reino Unido. En su sitio oficial se pueden encontrar tanto foros donde se discuten y anuncian eventos de interés, como materiales organizados para los distintos niveles educativos. Entre los recursos disponibles hay guías de actividades, evaluaciones, referencias, libros y tutoriales sobre distintas herramientas y tecnologías, entre otros. Si bien casi la totalidad de los recursos disponibles se encuentran en idioma inglés, la mayoría de los mismos son revisados y publicados bajo licencias Creative Commons⁶⁰ de manera tal que los mismos pueden utilizarse libremente.

Cabe aclarar que para registrarse en esta comunidad es necesario validar los antecedentes, debiendo, en el momento de la registración, indicar una referencia para comprobar la procedencia del usuario.

2.5.1.3. CS Unplugged

Computing Science Unplugged⁶¹ es una colección de actividades para enseñar los conceptos básicos de las ciencias de la computación sin una computadora, es decir, con actividades desconectadas.

Todas las actividades están publicadas bajo una licencia Creative Commons, por lo que se pueden copiar, compartir y modificar libremente. Las actividades fueron recopiladas en un libro el cual está traducido a más de veinte idiomas, siendo el español uno de ellos.

En el sitio oficial de esta iniciativa, además del libro se pueden acceder a videos e información de interés para la comunidad educativa.

2.5.1.4. La Fundación Logo

La Fundación Logo⁶² es una organización sin fines de lucro dedicada a apoyar a educadores, padres y estudiantes en su compromiso con la computación creativa. Apoyada por numerosas instituciones, en su sitio oficial se encuentran numerosos recursos y grupos de discusión donde se promueven eventos y materiales destinados a la enseñanza de la programación como una forma de fomentar la creatividad y el pensamiento computacional. Los recursos se basan en experiencias y herramientas similares a Logo, algunas de las cuales fueron nombradas en la sección 2.4 de este informe.

Esta organización funciona desde hace más de 20 años y, a modo de ejemplo, en la actualidad se encuentra promoviendo 4 eventos de verano, realizados en instituciones norteamericanas durante los meses de junio y julio del corriente año y el “Scratch Day” a realizarse el 2 de diciembre de 2017.

⁵⁹Sitio oficial: <https://www.computingatschool.org.uk/>. Último acceso: noviembre 2017.

⁶⁰Estas licencias se describen en la sección 5.1.1.2.

⁶¹Sitio oficial: <http://csunplugged.org/>. Último acceso: noviembre 2017.

⁶²Sitio oficial: <http://el.media.mit.edu/logo-foundation/index.html>. Último acceso: noviembre 2017.

2.5.2. Iniciativas nacionales

En la República Argentina, desde hace varios años, se fomentan desde el estado nacional diferentes políticas que estimulan el estudio de carreras de Ingeniería e Informática. Dichas iniciativas están relacionadas a la gran demanda de profesionales informáticos en la actualidad (OPSSI, 2015).

Existen iniciativas destinadas a alumnos que deben ingresar o ya se encuentran en la universidad, tales como programas de becas que impulsan el ingreso y egreso de jóvenes en carreras de informática e ingeniería, las que contribuyen a la formación de recursos en el área. Una de las iniciativas más destacadas es **Program.AR**⁶³, que trabaja para que las ciencias de la computación, y en particular la programación, estén presente en todas las escuelas del país. Apoyada por la Fundación Sadosky⁶⁴ y con la participación de numerosas universidades del país propone construir consensos colectivos a través de foros de discusión de carácter federal, democráticos e inclusivos.

En el marco de esta iniciativa, se encuentran en desarrollo dos proyectos de alcance nacional:

- **Cuatro manuales de ciencias de la computación** para el primer y segundo ciclo de primaria y para el primer y segundo ciclo de secundaria. El manual correspondiente al segundo ciclo de secundaria fue desarrollado por el equipo de la UNLP, del cual esta tesista forma parte.
- **Diseño curricular de la “Especialización en didáctica de las ciencias de computación”** destinado a docentes de escuelas secundarias y que se espera comience a dictarse en el ciclo lectivo 2018.

Los programas nacionales tales como “Conectar Igualdad”⁶⁵, “Primaria Digital”⁶⁶, los programas de mejoras de la escuela secundaria técnica y otros similares a nivel provincial, han permitido equipar las escuelas primarias y secundarias con laboratorios de computadoras e infraestructura de conectividad, además de haber permitido contar con el modelo uno a uno (una computadora por alumno), con lo cual muchos establecimientos ya no tienen la brecha de la falta de recursos para abordar temas relacionados a las TIC.

En el año 2012, el estado nacional lanzó el programa “Dale Aceptar”⁶⁷ que proponía la apertura de concursos para alumnos de escuelas secundarias a los que se los desafiaba a construir videojuegos, animaciones y a que resuelvan coartadas utilizando diferentes herramientas digitales. La inscripción era abierta y no requería conocimientos previos de programación. Los premios eran muy atractivos para los jóvenes que participan como ser consolas de videojuegos. Si bien esta iniciativa tuvo bastante

⁶³Sitio oficial <http://program.ar/>. Último acceso: noviembre 2017.

⁶⁴Fundación Sadosky: <http://www.fundacionsadosky.org.ar/>. Último acceso: noviembre 2017.

⁶⁵Sitio oficial: <http://planied.educ.ar/programas/conectar-igualdad/>. Último acceso: noviembre 2017.

⁶⁶Sitio oficial: <http://planied.educ.ar/programas/primaria-digital/>. Último acceso: noviembre 2017.

⁶⁷Sitio oficial: <http://www.daleacceptar.gob.ar/>. Último acceso: noviembre 2017.

éxito en sus cuatro ediciones previas, no promovía actividades en la escuela secundaria como parte de la formación integral de los alumnos, era más una iniciativa particular.

A partir del lanzamiento de Program.AR en el año 2014, se trabajan en lineamientos y propuestas para llevar adelante en las escuelas, como ser la promoción de “La Hora del Código” en la Argentina.

En marzo de 2017, se publicó la resolución 1536-E/2017 del Ministerio de Educación y Deportes de la Nación (Ministro de Educación y Deportes, 2017) en la cual se crea el “Plan Nacional Integral de Educación Digital” (PLANIED) en el ámbito de la Secretaría de Innovación y Calidad Educativa, cuya misión principal es la de *“integrar la comunidad educativa en la cultura digital, favoreciendo la innovación pedagógica, la calidad educativa y la inclusión socioeducativa.”* Entre los objetivos específicos de dicho plan se encuentra el de *“desarrollar iniciativas orientadas a la construcción de conocimiento sobre la programación y la robótica.”*

PLANIED, engloba a las iniciativas “Conectar Igualdad” y “Primaria Digital”, los cuales fueron actualizados en el mes de agosto del corriente año, lanzando en conjunto el proyecto “Escuelas del Futuro”⁶⁸ el cual promueve distintas acciones tanto a nivel primario como secundario.

Escuelas del Futuro

Como se mencionó en la sección anterior, Escuelas del Futuro se enmarca en el “Plan Nacional Integral de Educación Digital” (PLANIED) el cual combina la entrega de tecnología educativa con propuestas pedagógicas para su utilización. Incluye iniciativas tanto para el nivel primario como secundario. En ambos casos este proyecto busca *“propiciar la alfabetización digital de los alumnos, a través de la integración de áreas de conocimiento emergentes, como la programación y la robótica, y facilitar recursos digitales y propuestas pedagógicas, que favorezcan el aprendizaje de campos tradicionales del saber, como las ciencias naturales, la matemática, las prácticas del lenguaje, y las lenguas extranjeras”*(Escuelas del Futuro, 2017).

En este marco se cuentan las siguientes iniciativas:

- Para el nivel secundario: **ExperimentTIC**⁶⁹, **DronLab**⁷⁰, Escuela RED⁷¹, y propuestas para aprender inglés, entre otras.
- Para el nivel primario se destacan: Miniprogramadores⁷² y Matijuegos⁷³, entre otras.

Si bien se trata de un proyecto muy nuevo, las iniciativas y propuestas son muy alentadoras.

⁶⁸Sitio oficial: <http://educacion.gob.ar/escuelas-del-futuro>. Último acceso: noviembre 2017.

⁶⁹<http://educacion.gob.ar/escuelas-del-futuro/seccion/322/experimentic>

⁷⁰<http://educacion.gob.ar/escuelas-del-futuro/seccion/325/dronlab>

⁷¹<http://educacion.gob.ar/escuelas-del-futuro/seccion/329/escuela-red>

⁷²<http://educacion.gob.ar/escuelas-del-futuro/seccion/367/miniprogramadores>

⁷³<http://educacion.gob.ar/escuelas-del-futuro/seccion/306/matijuegos>

2.5.3. En la Facultad de Informática de la UNLP

En esta sección se incluyen algunas de las iniciativas en las cuales esta tesista está involucrada. Las mismas son llevadas a cabo en el ámbito de la Facultad de Informática. Desde el año 2008, se viene trabajando en promover la enseñanza de la programación impulsando dos propuestas didáctico-pedagógicas para enseñar a programar en las escuelas secundarias: “Programando con robots y software libre” y “Programando con RITA”. Ambas propuestas acercan a los docentes y estudiantes de las escuelas a la programación de computadoras implementando estrategias innovadoras y herramientas didácticas diferentes. En ambas iniciativas se priorizó trabajar con herramientas y lenguajes de programación libres, utilizados ampliamente en comunidades de programadores. Esta decisión está vinculada a que la enseñanza de la programación no quede limitada al uso de las herramientas propuestas, sino que sea un habilitador para la incorporación de conceptos y herramientas de programación más avanzadas.

Una de las ventajas de trabajar con *software* libre es la apropiación de los recursos, es decir la posibilidad de generar una versión personalizada de los mismos para cada comunidad. Esto se logra porque, al disponer del código fuente es posible adecuarlo y adaptarlo a las necesidades concretas de cada grupo de usuarios. Cuando se enseña a programar, disponer del código fuente es fundamental para que el estudiante pueda estudiar cómo fue hecho, analizar cómo puede adaptarlo y luego distribuir las adaptaciones que se realicen para que otros usuarios las tomen. Todas estas cuestiones están fuertemente vinculadas con la creatividad e intervienen habilidades relacionadas a la resolución de problemas y al pensamiento crítico. La filosofía del *software* libre promueve la creatividad y el trabajo colaborativo.

Los proyectos “Programando con robots y software libre” y “Programando con RITA” no sólo utilizan licencias libres de *software* sino que a la vez promueven el uso de contenidos libres. Todo el material y recursos generados en estas propuestas, se publican y distribuyen con licencias Creative Commons.

“Programando con Rita” se inicia en el año 2012, a través de la utilización de la herramienta didáctica RITA descrita en la sección 2.4.15.4. El uso de RITA la cual favorece el aprendizaje de los conceptos básicos de programación y constituye un punto de partida interesante para comenzar a entender y familiarizarse con el lenguaje de programación Java (Queiruga C., Fava L., Banchoff Tzancoff C., Aybar Rosales V., Miyuki Kimura I. & Brown Bertneche M, 2013). El trabajo con Java es más aplicable a escuelas secundarias técnicas con orientación informática debido a que cuentan con espacios curriculares en los que es posible definir un proceso de aprendizaje en torno a la programación. (Díaz J., Banchoff Tzancoff C., Queiruga C. & Martin S., 2014)

El proyecto “Programando con robots y software libre” se describe en detalle en el capítulo 4, debido a que el uso de ProBots3D se enmarca en dicho proyecto.

A partir del año 2016 se comenzó a trabajar en la enseñanza de programación con niños y docentes de escuelas primarias. En el capítulo 3 se detallan las actividades realizadas y planificadas. En estas últimas se prevé la inclusión de ProBots3D.

2.6. Resumen del capítulo

Incluir la enseñanza de programación ya no es una preocupación de las cámaras que agrupan a las empresas de desarrollo de *software* que buscan, principalmente, la formación de recursos para cubrir la demanda de empleo en el área.

En el mundo actual, donde todos los objetos son susceptibles de ser programados y “adaptados”, la nueva brecha digital se refiere a quienes tienen competencia en programación y los que carecen de las mismas. Debido a esto, día a día se incorporan más iniciativas que abordan este tema.

En este capítulo se presentaron distintas iniciativas a nivel nacional e internacional cuyo objetivo común es encontrar formas interesantes y motivadoras para enseñar a programar, ya sea en el ámbito de la escuela o a nivel general. A lo largo del capítulo, se describieron y analizaron algunas de las herramientas más populares utilizadas para este fin, con especial interés en aquellas implementadas en el marco de actividades donde esta tesista ha participado.

Experiencias de programación en el aula



“Cuando un niño aprende a programar, el proceso de aprendizaje es transformado. Se vuelve más activo y auto-dirigido...El nuevo conocimiento es una fuente de poder y se experimenta como tal desde el momento en que empieza a formarse en la mente del niño.”

– Seymour Papert

Desde hace varios años, la tesista viene trabajando en distintos contextos sobre la incorporación de la programación en las escuelas. Algunas de estas experiencias se desarrollaron en el marco de actividades del proyecto “Programando con robots y software libre”, centrando las mismas en los talleres realizados para alumnos del nivel secundario principalmente; y otras experiencias fueron realizadas en el marco de su actividad profesional independiente del proyecto antes mencionado.

Durante el ciclo lectivo 2016, se realizó una experiencia con 120 estudiantes de nivel secundario¹, y 30 de nivel primario², a través de la cual se pudo sistematizar tanto el uso de las distintas herramientas

¹Esta actividad fue realizada a nivel particular, donde la tesista se desempeñó como docente de la asignatura extra-programática “Informática” en una escuela secundaria.

²En este caso las actividades se realizaron en el marco de un proyecto de extensión de la Facultad de Informática.

como de las actividades más adecuadas para las distintas edades.

En el caso de los estudiantes de nivel secundario, la bajada áulica de los temas fue realizada directamente por la tesista, mientras que en el caso de los alumnos de primaria, la misma fue realizada por el maestro Emilio Pérez en el marco del proyecto de extensión “Explorando el universo con *software* libre”³.

Las experiencias realizadas dieron lugar a una propuesta de sistematización y organización de los contenidos relacionados a la enseñanza de la programación para ambos niveles. Dicha propuesta fue implementada durante el ciclo lectivo 2017 en varios contextos. En particular, a nivel primario, se trabajó en todos los años de nivel primario de la Escuela Estudiantes de La Plata, abarcando a unos 360 alumnos; y parte de la propuesta en cuatro escuelas primarias de la región, abarcado a unos 200 alumnos aproximadamente. En todos los casos las actividades fueron llevadas adelante por los docentes a cargo de los cursos. En cuanto al nivel secundario, la propuesta fue implementada por la tesista en la Escuela Estudiantes de La Plata, estando a cargo de los 70 alumnos de 2do. año y coordinando las actividades de 68 alumnos de 3er. año.

3.1 Herramientas utilizadas en experiencias propias

De todas las herramientas presentadas en el capítulo 2 se seleccionaron algunas con las cuales se realizaron experiencias en el aula de la escuela durante los ciclos lectivos 2016 y 2017. Los contextos y recursos de cada institución donde se realizaron las distintas experiencias motivaron la elección de las herramientas elegidas.

A continuación se describen las experiencias realizadas con las distintas herramientas en los últimos dos años. Estas experiencias dieron lugar a una propuesta de organización de contenidos y a la elaboración de un conjunto de secuencias didácticas. Estas secuencias didácticas sirven de base para la propuesta de uso de ProBots3D.

3.1.1. Experiencias con Scratch

Esta herramienta fue utilizada por la tesista en una experiencia en la escuela secundaria con 120 niños y jóvenes de 2do. y 3er. año, durante el ciclo lectivo 2016, y con 70 durante el ciclo lectivo 2017. En el nivel primario, durante el ciclo lectivo 2016, se trabajó con 30 alumnos de 6to. año, pero durante el ciclo lectivo 2017, la cantidad de alumnos se incrementó notablemente, siendo en la actualidad más de 500 alumnos. Esto se debió a que durante el ciclo lectivo 2017, se trabajó la temática en dos

³Esta experiencia dio lugar al proyecto de extensión “Las ciencias y los niños. Una experiencia de programación en la escuela primaria”, acreditado y subsidiado en la convocatoria 2016 de la UNLP, el cual se lleva a cabo durante el ciclo lectivo 2017 en 5 escuelas primarias de la Plata, City Bell y Berisso.

contextos: por un lado en el marco del proyecto de extensión “Las ciencias y los niños. Una experiencia de programación en el aula”, del cual participan cuatro escuelas de La Plata, City Bell y Berisso. Por otro lado, también durante este ciclo lectivo, la tesista pasó a coordinar las actividades de Informática de la Escuela Estudiantes de La Plata, de la localidad de City Bell. En el plan propuesto se coordinó el trabajo con todos los niños de nivel primario. La organización de los contenidos a abordar en dicha escuela se basan en la propuesta presentada en la sección 3.2.

De acuerdo a los recursos de *hardware* y conectividad de cada institución, se utiliza Scratch, en sus versiones 1.4 y 2. La diferencia fundamental entre ambas versiones, respecto de los contenidos a abordar, radica en que con Scratch 1.4 no es posible introducir aspectos de abstracción, debido a que no es posible definir nuevos bloques que representen procedimientos. Pero esto no fue una limitante, debido a que esta versión se utilizó con los niños más pequeños y esta temática no fue abordada.

Si bien no fue la primera herramienta de programación con la que se interactuó, los estudiantes no tuvieron complicaciones a la hora de desarrollar animaciones y juegos sencillos.

Se trabajó en la exploración de la herramienta, la creación de fondos y disfraces personalizados para luego realizar animaciones e historietas animadas. Con los estudiantes mayores, se desarrollaron juegos sencillos destinados a niños de preescolar en los cuales se pueden reconocer letras, colores y números.

3.1.2. Experiencias con Alice

Como se mencionó en la introducción de esta tesis, la tesista integra el equipo de la Universidad Nacional de La Plata que se desarrolló el manual de ciencias de la computación destinado al segundo ciclo de educación secundaria. A solicitud de la Fundación Sadosky, la herramienta a utilizar para la ejercitación fue Alice 2.4. Por este motivo, se han realizado numerosas actividades y secuencias didácticas.

En una evaluación personal, Alice no parece ser la herramienta más adecuada para introducir los primeros conceptos de programación en nuestra región debido, principalmente a que la versión en español no está totalmente traducida y, requiere trabajar con el paradigma de programación orientada a objetos, por lo que no se ha incluido su uso en las actividades del proyecto.

Por este motivo, si bien se desarrollaron numerosas actividades, las mismas no fueron puestas en práctica en las experiencias realizadas en las distintas escuelas.

3.1.3. Experiencias con Lightbot

Esta herramienta fue utilizada en las experiencias realizadas tanto en el ciclo lectivo 2016 como 2017. Fue trabajada con 190 estudiantes de 14 a 16 años de nivel secundario. Se la utilizó, junto con otras herramientas, en una primera aproximación a la programación antes de pasar a trabajar con Scratch.

Se trabajaron los aspectos básicos de programación y el concepto de abstracción, dado que es posible escribir procedimientos para resolver determinadas tareas.

3.1.4. Experiencias con Juegos de Blockly

Durante los ciclos lectivos 2016 y 2017, se trabajó esta herramienta con 190 estudiantes de nivel secundario. Las primeras actividades propuestas por la herramienta son muy sencillas y motivadoras para los estudiantes, pero las que se encuentran al final del recorrido son bastantes complicadas.

En general, se utilizaron las primeras tres actividades propuestas: Laberinto, Pájaro y Tortuga. La actividad del Laberinto también fue trabajada con los alumnos de nivel primario.

En el nivel primario se trabajó con la actividad de Rompecabezas y los primeros niveles de la actividad Laberinto.

3.1.5. Experiencias con las herramientas de code.org

Varias de las herramientas propuestas por code.org fueron usadas en actividades introductorias y de evaluación.

En la semana de 3 de octubre de 2016 se celebró la Hora del Código (ver 2.5.1.1) en Latinoamérica utilizándose varias de las actividades propuestas de esta iniciativa. En particular, se mencionan los juegos de Minecraft⁴, o Star Wars⁵. Lo mismo sucedió con la edición 2017 de este evento donde se trabajaron estos desafíos con los alumnos que aún no los habían realizado.

Los recursos propuestos por code.org fueron trabajados, además de los niveles primario y secundario, en el nivel preescolar, con la sala de cinco años.

- **Nivel preescolar:** se trabajaron los primeros desafíos del curso 1⁶, el cual está pensado para niños más pequeños. En este caso se trabajó con aproximadamente 60 niños de última sala de un Jardín de Infantes.

⁴Sitio Oficial: <https://code.org/minecraft>. Último acceso: noviembre 2017.

⁵Sitio Oficial: <https://code.org/starwars>. Último acceso: noviembre 2017.

⁶Curso 1: <https://studio.code.org/s/course1>. Último acceso: noviembre 2017.

- **Nivel primario:** con estos niños se trabajaron las actividades más sencillas del curso ²⁷ intercalando actividades de Juegos de Blockly.
- **Nivel secundario:** con estos niños y jóvenes se trabajaron varias de las herramientas propuestas, incluyendo las actividades realizadas en la Hora del Código.

En ninguno de los casos se ejecutaron en forma sistemática las actividades propuestas en los cursos, sino que se diseñaron nuevas propuestas de uso en base a la combinación y uso de las distintas herramientas, dependiendo de las edades y recursos de cada grupo escolar y escuelas.

3.1.6. Experiencias con Pilas Bloques

Se utilizó Pilas Bloques en ambos grupos de estudiantes: primario y secundario. En todos los casos, los niños pudieron resolver los desafíos con mayor o menor dificultad.

Si bien la herramienta fue cambiando durante los últimos dos años, esta tesista la utilizó en algunas de las actividades iniciales para introducir los conceptos básicos de secuencia y abstracción procedural en el caso de la escuela secundaria. En particular, los desafíos tales como “Instalando juegos” y “La gran aventura del mar encantado”, fueron utilizados para introducir la noción de procedimientos.

3.1.7. Experiencias con pilas-engine

Si bien esta herramienta se trabaja desde hace tres años en el Seminario de Lenguajes, opción Lenguaje Python, con los alumnos de 2do. año de las carreras de la Facultad de Informática, también se realizó una experiencia con 50 estudiantes de nivel secundario durante el ciclo lectivo 2017, en donde se introdujo la herramienta y se desarrolló una aplicación sencilla en dos talleres de 6 horas en total. Una experiencia fue realizada en la Facultad de Informática y otra experiencia se llevó a cabo en la EET 2 de Berisso durante el mes de julio de 2017.

3.1.8. Resumen de las experiencias realizadas

A modo de resumen de las experiencias realizadas se elaboraron dos tablas que detallan los aspectos más relevantes de cada experiencia. La tabla 3.1 muestra los detalles más destacados de las experiencias realizadas en colegios primarios.

²⁷Curso 2: <https://studio.code.org/s/course2/stage/3/puzzle/1>. Último acceso: noviembre 2017.

	Escuela EdeLP	Escuela 67	Escuela 18	Escuela 9	Escuela Anexa
Edades de alumnos	6 a 12 años	8 a 10 años	11 y 12 años	11 y 12 años	11 y 12 años
Cantidad de alumnos	350	60	60	60	60
Año	2017	2017	2017	2017	2016-2017
Tipo de herramientas usadas	Herramientas con lenguajes basados en bloque e icónicos				
Rol de la tesista en la experiencia	Coordinación	Coordinación y capacitación a las docentes	Coordinación y capacitación a las docentes	Supervisión de la actividad	Supervisión de la actividad
Resultados generales	Se consolidó la noción de programa. Los alumnos más grandes realizaron animaciones con Scratch.	Se consolidó la noción de programa. Los alumnos más grandes realizaron animaciones con Scratch.	Los alumnos realizaron animaciones sencillas sin profundizar en los conceptos subyacentes.	Desafíos de Pilas Bloques y animaciones con Scratch.	Se consolidó la noción de programa y se realizaron animaciones con Scratch.
Comentarios	Se instaló el tema en la currícula.	Se trabajó con las netbooks del programa Primaria Digital usando Linux.	Realizaron una feria donde los niños y niñas mostraron sus trabajos a sus familias.	Se trabajó con las netbooks del programa Primaria Digital usando Linux.	Se prevé su incorporación a la currícula para el año 2018 a toda la escuela.

Tabla 3.1. Resumen de experiencias realizadas en las escuelas primarias.

Del mismo modo que en el caso de las instituciones de nivel primario, la tabla 3.2 muestra las experiencias realizadas con alumnos de escuelas secundarias. En este caso se describe en forma separada el caso de dos escuelas donde se trabajó recientemente y en un apartado general se resume la información de los talleres llevados a cabo en la Facultad de Informática con alumnos de varios establecimientos.

	Escuela EdeLP	Escuela EET2 Berisso	Talleres varios
Edades de alumnos	13 a 15 años	16 a 17 años	15 a 18 años
Cantidad de alumnos	190 alumnos	20	250
Año	2016 - 2017	2017	2012 - 2017
Tipo de herramientas usadas	Basadas en Bloques	Python y pilas-engine	RITA, Python y pilas-engine.
Rol de la tesista en la experiencia	Coordinación y dictado de clases	Coordinación y dictado de clases	Coordinación y dictado de los talleres
Resultados generales	Se desarrollaron animaciones y juegos sencillos.	Se desarrollaron juegos sencillos	Se desarrollaron juegos sencillos y actividades con los robots.
Comentarios	Se instaló el tema en la currícula.	Se realizó en las instalaciones de la escuela.	Algunos talleres se realizaron en el ámbito de la Facultad de Informática y otros en los mismos colegios.

Tabla 3.2. Resumen de experiencias realizadas con alumnos de nivel secundario.

Como se mencionó en las secciones anteriores, algunas temáticas fueron trabajadas con varias herramientas. La tabla 3.3 muestra, en forma resumida, los contenidos abordados con cada una durante los ciclos lectivos 2016 y 2017. Salvo pilas-engine que fue utilizada sólo con estudiantes secundarios, el resto fueron trabajadas tanto en nivel primario como secundario.

Herramienta	Contenido trabajado
Scratch	Secuencias de instrucciones. Estructuras condicionales y repetitiva. Abstracción procedural
Lightbot	Secuencias de instrucciones y abstracción procedural
Juegos de Blocly	Secuencias de instrucciones. Estructuras condicionales y repetitivas
Herramientas de code.org	Secuencias de instrucciones. Estructuras condicionales y repetitivas
Pilas Bloques	Secuencias de instrucciones. Estructuras condicionales y repetitiva. Abstracción procedural
pilas-engine	Secuencias de instrucciones. Estructuras condicionales y repetitiva. Abstracción procedural. Introducción al lenguaje Python

Tabla 3.3. Resumen de contenidos abordados con cada herramienta.

3.2. Enseñando a programar en la escuela primaria

En función de las herramientas analizadas y las experiencias realizadas se diseñó una propuesta de sistematización que incluye una organización de contenidos y las secuencias didácticas para introducir cada tema en cada nivel educativo. Dicha propuesta está siendo aplicada en varias escuelas de la región.

En los capítulos 6 y 7 se retomará esta propuesta con la inclusión de ProBots3D y la evaluación de su uso.

3.2.1. Sistematización propuesta

Las experiencias exitosas realizadas durante el ciclo lectivo 2016 permitieron planificar actividades que incluye la enseñanza de programación en los distintos años en más instituciones. Algunas de las actividades se realizan en el marco del proyecto de extensión “Las ciencias y los niños. Programar en la escuela primaria”, del cual esta tesista es la co-directora y otras en el marco de su actividad profesional particular como docente y coordinadora del área de Informática en la Escuela Estudiantes de La Plata.

Las escuelas en las que se está trabajando en la actualidad con esta temática son:

- **Escuela Graduada Joaquín V. González, Anexa de la UNLP:** las actividades en la escuela se llevan a cabo bajo la coordinación de la CC. Viviana Harari. Desde el mes de mayo de 2017, se está trabajando con 60 niños y niñas de 6to. año.

- **Escuela 9 de Berisso:** las actividades en esta escuela están coordinadas por la APU Sofía Martín, quien es directora del proyecto de extensión e integrante del LINTI. En esta escuela se trabaja con los niños de 6to. año.
- **Escuela 18 de Gonnet:** en esta escuela se está trabajando con los niños y niñas de 5to. y 6to. año. Las actividades en la escuela están siendo coordinadas por esta tesista quien proporciona el apoyo necesario a las maestras encargadas de las divisiones afectadas.
- **Escuela 67 de City Bell:** las actividades en esta escuela son llevadas adelante por la docente Giselle Molina, quien ocupa el cargo de EMATP (“Encargado de Medios de Apoyo Técnico y Pedagógico”) y coordinadas por esta tesista. Se trabaja desde el mes de agosto con los niños y niñas de 3er. y 4to. año de la escuela.
- **Escuela Estudiantes de La Plata:** en esta escuela se trabaja con todos los años del colegio primario. Se coordinaron actividades con las docentes a cargo del área “Computación” y las mismas se llevan a cabo en el segundo semestre de 2017.

En el caso de la escuela Estudiantes de La Plata la planificación se presentó como parte del proyecto institucional de Informática que abarca a todo el nivel primario y el primer ciclo de secundaria (en este caso sólo 2do. y 3er. año). Debido a que este año es el primer año en el cual se comienza a trabajar la temática, se han realizado tres planificaciones dividiendo a los niños en tres franjas: 1er. y 2do. año; 3er. y 4to. año; y 5to. y 6to. año.

En las restantes escuelas, se trabajó esta propuesta en forma parcial. En algunos casos, se coordinó una actividad adicional que incluye el uso de los robots del proyecto “Programando con robots y software libre”. Es en esta actividad donde se incluye el uso de ProBots3D.

La tabla 3.4 muestra la organización de contenidos propuesta para 1er. y 2do. año de la escuela.

Tema	Herramienta/Actividad
Algoritmos y programas	
Concepto de algoritmo	Se trabaja en la formulación y seguimiento de algoritmos sencillos y en la resolución de laberintos simples, identificando las instrucciones y secuencias necesarias para su resolución. Esta actividad no requiere del uso de computadoras.
Programación con bloques	Se trabaja con Juegos de Blockly, el desafío del Rompecabezas y, en caso de contar con conexión a Internet, también con las actividades iniciales del curso 1 de code.org. Se plantea la metodología de trabajo y el paradigma de encastrados dados por los lenguajes de bloques.
Programas e instrucciones	
Programas e instrucciones	Se trabaja en la etapa 4 del curso 1 de code.org (niveles 1 al 15.), el cual permite reforzar la metodología de trabajo y practicar puntos cardinales. Requiere acceso a Internet.
Observar y corregir un programa	Actividad de <i>debugging</i> del curso 1 de code.org. Se presenta un programa y se analiza qué hace el mismo y si es correcto. Requiere acceso a Internet.
Programas simples	Se realizan los primeros programas usando Pilas Bloques. Desafíos “Alien toca botón” y “El gato en la calle”. Es posible trabajar en línea como desconectado.
Programas simples	Estas actividades requieren de conexión a Internet y permiten ejercitar los conceptos vistos. Se trabaja con las etapas 7, 8 y del curso 1 y la etapa 4 del curso 2 de code.org.
Uso de Scratch	
Presentación de Scratch	Uso del editor de disfraces. Presentación de la herramienta. Scratch 2 o, de no contar con acceso a Internet, se puede utilizar Scratch 1.4.

Tabla 3.4. Organización de contenidos propuesta para el 1er. y 2do año.

La tabla 3.5 muestra la organización de contenidos propuesta para 3er. y 4to. año de la escuela. Cabe aclarar que en esta propuesta se tiene en cuenta que es la primera vez que se encara el tema, por lo que algunas actividades se repiten en la organización presentada para los primeros años.

Tema	Herramienta/Actividad
Algoritmos y programas	
Concepto de algoritmo	Actividad desconectada. Se trabaja en la formulación y seguimiento de algoritmos sencillos y en la resolución de laberintos simples, identificando las instrucciones y secuencias necesarias para su resolución.
Programación con bloques	Se trabaja con Juegos de Blockly, el desafío del Rompecabezas y, en caso de contar con conexión a Internet, también con las actividades iniciales del curso 1 de code.org. Se plantea la metodología de trabajo y el paradigma de encastrados dados por los lenguajes de bloques.
Programas e instrucciones	Se propone el trabajo con la etapa 3 del curso 2 de code.org. el cual permite reforzar la metodología de trabajo. Requiere acceso a Internet.
Programas simples	Se realizan los primeros programas usando Pilas Bloques. Desafíos “Alien toca botón” “El gato en la calle”. Es posible trabajar en línea como desconectado.
Programas más complejos	
Programas simples	Se propone trabajar con el desafío de Pilas Bloques “Instalando Juegos”, niveles 1 al 9 de la etapa 4 del curso 2 de code.org, los niveles 1 y 2 del laberinto de Juegos de Blockly. Ejercitación para fijar el concepto.
Estructuras repetitivas	Se propone trabajar con los niveles 3 a 5 del Laberinto de Juegos de Blockly, Desafío de Pilas Bloques “No me canso de saltar” e “Instalando Juegos”. Niveles 4 al 12 de la etapa 3 del curso 3 de code.org. Se propone realizar las actividades primero sin indicación alguna y luego hacer notar que hay secuencias de instrucciones que se repiten, por lo que se puede utilizar la estructura de control repetitiva.
Uso de Scratch	
Explorando la herramienta	Uso del editor de disfraces. Presentación de Scratch 2 o, de no contar con acceso a Internet, se puede utilizar Scratch 1.4. Se propone explorar la herramienta: las galerías de personajes y fondos.
Programas básicos	Explorar y experimentar con los bloques movimientos y apariencia. Realizar programar simples que puedan mover personajes y cambiar sus disfraces.
Programas básicos	Explicar el uso del bloque esperar y de los bloques de diálogos (decir y pensar). Proponer el armado de una historietita animada sencilla.

Tabla 3.5. Organización de contenidos propuesta para el 3er. y 4to. año.

La tabla 3.6 muestra la organización de contenidos propuesta para 5to. y 6to. año de la escuela. Como en el caso anterior, cabe aclarar que en esta propuesta se tiene en cuenta que es la primera vez

que se encara el tema, por lo que algunas actividades se repiten en la organización presentada para los años anteriores.

Tema	Herramienta/Actividad
Algoritmos y programas	
Concepto de algoritmo	Actividad desconectada. Se trabaja en la formulación y seguimiento de algoritmos sencillos y en la resolución de laberintos simples, identificando las instrucciones y secuencias necesarias para su resolución.
Programas e instrucciones	Se trabaja con las etapas 3 y 4 del curso 2 de code.org. Esta actividad permite trabajar las diferencias entre algoritmos y programas y la introducción del concepto de autómeta.
Programas simples	Se realizan los primeros programas usando Pilas Bloques. Desafíos “Alien toca botón” y “El gato en la calle”. Es posible trabajar en línea como desconectado.
Programas más complejos	
Estructuras repetitivas	Se propone trabajar con desafío de Pilas Bloques “No me canso de saltar”, “Instalando Juegos” y “La gran aventura del mar encantado”. También se propone trabajar la etapa 2 del curso 3 de code.org. Se propone realizar las actividades primero sin indicación alguna y luego hacer notar que hay secuencias de instrucciones que se repiten, por lo que se puede utilizar la estructura de control repetitiva.
Estructuras repetitivas	Para reforzar el concepto, se propone el trabajo con los desafíos de Pilas Bloques “Alimentando a los peces” y “El alien y las tuercas”.
Uso de Scratch	
Explorando la herramienta	Uso del editor de disfraces. Presentación de Scratch 2 o, de no contar con acceso a Internet, se puede utilizar Scratch 1.4. Se propone explorar la herramienta: las galerías de personajes y fondos.
Programas básicos	Explorar y experimentar con los bloques movimientos y apariencia. Realizar programar simples que puedan mover personajes y cambiar sus disfraces.
Programas básicos	Explicar el uso del bloque esperar y de los bloques de diálogos (decir y pensar). Proponer el armado de una historieta animada sencilla.
Programas básicos	Explicar los bloques de eventos de teclas con los cuales armar una animación interactiva sencilla.

Tabla 3.6. Organización de contenidos propuesta para el 5to. y 6to. año.

3.3.2. Secuencias didácticas

Además de la planificación realizada para el año 2017, se han desarrollado las secuencias didácticas para abordar los distintos temas en casi todos los años escolares con los cuales se trabaja.

Para la elaboración de las secuencias didácticas se ha tomado como referencia la guía propuesta por Díaz-Barriga (Díaz Barriga, 2013), describiendo para cada secuencia sus objetivos y las actividades que incluye. Para esta primera propuesta no se han incluido evaluaciones en las mismas.

Para cada actividad se indica: objetivos, modalidad de trabajo, edades de los destinatarios, materiales requeridos y luego el desarrollo, en el cual se tiene en cuenta una actividad introductoria o inicial, el desarrollo propiamente dicho y un cierre.

En esta sección sólo se presentan las secuencias didácticas que luego se verán afectadas por la propuesta de uso de ProBots3D. Cabe aclarar que las mismas están destinadas a los docentes a cargo de los cursos, considerando que los mismos pueden no tener conocimientos previos sobre programación.

La figura 3.1 muestra, en un cuadro resumido, la organización de contenidos propuesta para el nivel primario destacando el lugar donde se inserta la secuencia didáctica donde luego se incluirá la herramienta ProBots3D.



Figura 3.1. Organización de contenidos propuesta para el nivel primario.

A continuación se detalla la secuencia didáctica denominada “Algoritmos y programas” en la cual se introducen dichos conceptos. Se describen sus objetivos, destinatarios, recursos necesarios y el desarrollo de las actividades que la componen.

3.3.2.1. Secuencia didáctica: algoritmos y programas

El eje central de esta secuencia didáctica es introducir las características de un programa de computadoras y la noción de algoritmo. En la organización de contenidos propuesta corresponde al tema “Programas e instrucciones”.

Objetivos de la secuencia: que los estudiantes logren:

- Comprender las diferencias entre un algoritmo expresado en lenguaje natural y un programa de computadora.
- Comprender los conceptos de instrucción, programa y secuencia.

Esta secuencia didáctica incluye un única actividad denominada “Mi primer programa” detallada a continuación.

Actividad 1: mi primer programa

Objetivos de la actividad: que los alumnos puedan:

- Identificar las instrucciones que conforman un programa.
- Comprender que un programa de computadora debe escribirse en un lenguaje específico y no ambiguo.

Modalidad de trabajo: un primer momento a trabajar con todo el grupo y un segundo momento a trabajar en grupos reducidos.

Edades de los destinatarios: al tratarse de una actividad inicial, la misma puede realizarse con niños y niñas de 6 a 12 años. Se proponen dos herramientas distintas en el caso de primer y segundo ciclo en el momento 2 de la actividad.

Materiales: una computadora por grupo con acceso a Internet⁸.

Desarrollo:

Esta actividad se plantea en dos momentos: en el primero se retoman los conceptos de algoritmo e instrucciones vistos anteriormente con una consigna sencilla. Los alumnos deberán definir distintas secuencias de instrucciones y uno de ellos será el autómata que las ejecute. En el segundo momento se trabajará con una aplicación que presenta distintos desafíos de complejidad incremental en varios niveles.

⁸Para este trabajo, en el ciclo lectivo 2017, se desarrolló una aplicación que puede suplir esta actividad y se trabaja en forma desconectada en aquellas instituciones que no cuentan con acceso a Internet. La misma se encuentra en etapa de testeo.

Momento 1: me convierto en robot

Para comenzar la actividad, elegir a uno de los alumnos e indicarles que, a partir de ese momento, se convierte en un robot y el resto de sus compañeros deberán guiarlo para que resuelva distintas consignas. Por ejemplo, se pueden proponer consignas tales como: “dirigirse de un punto a otro en el aula”, “escribir alguna frase en el pizarrón”, “ir a buscar a la directora” o “guiarlo a su banco para que se siente”. Posicionar al alumno en un punto inicial para tomar de referencia y luego de elegir la consigna, indicar que comiencen a elaborar las instrucciones. Escribir en el pizarrón las instrucciones que vayan surgiendo, de acuerdo al nivel de lecto-escritura de los niños, se puede utilizar el lenguaje castellano o gráficos que representan las distintas acciones (como flechas para avanzar, girar, etc.) A modo de ejemplo, si la consigna fuera: “guiar al estudiante a su banco para que se siente”, pueden surgir instrucciones tales como:

Avanzar 4 pasos.
 Girar a la derecha.
 Avanzar 4 pasos.
 Girar a la izquierda.
 Sentarse.

Una vez que todos estén de acuerdo con lo realizado, trabajar las siguientes ideas:

- Nuestro robot “sabe” qué significa “avanzar”, “girar” y “sentarse”. ¿Qué pasaría si en lugar de ordenar que avance, le dijéramos que “camine”? ¿Lo haría? Se espera que los alumnos respondan que “sí”, dado que pueden asociar que: “cuando camina, avanza”.
- Preguntar: ¿qué pasaría si cuando llega a su silla, la misma no se encuentra y la orden “sentarse”? ¿Lo haría? En este momento destacar que, “un robot no piensa”, sólo ejecuta las instrucciones, por lo tanto, si no le ordenamos que se fije si está la silla y, en caso que esté se siente, va a intentar sentarse con o sin silla.
- Otra pregunta a realizar puede ser: ¿todos pensaron en las mismas instrucciones? Es muy probable que algunos hayan propuesto otro camino hacia su silla, y, por ende, haya surgido otra secuencia de instrucciones. De ser así, indicarles que hay muchas maneras de resolver un problema, quizás algunas sean mejores que otras, pero no son únicas.

Al finalizar la puesta en común, indicarles que un robot que ejecuta fielmente las instrucciones sin pensar en alternativas ni analizar el contexto por sí mismo se llama autómatas. En este momento,

trabajar el concepto de **programa**. Explicarles que un programa es una secuencia de instrucciones, como las que ellos definieron, pero que debe estar escrito en un lenguaje más específico, para que pueda ser entendido por un autómatas. Un autómatas no va a pensar en que las instrucciones “avanzar” y “caminar” corresponden a la misma acción. Si no conoce la instrucción “caminar” no sabrá qué hacer cuando se la ordenen. Tampoco pensará que si no hay silla no debe sentarse. Si la orden es “sentarse”, lo hará aunque no haya silla. Se puede volver a las instrucciones escritas en el pizarrón y marcar una por una para identificarlas y señalar que el **conjunto de instrucciones** es el **programa**.

Momento 2: programando en la computadora

En este momento dividir a la clase en grupos reducidos y situarlos con una computadora por grupo. Indicarles que deben abrir el navegador e ingresar a la actividad del sitio: <https://studio.code.org/s/course2/stage/3/puzzle/1>. En el caso de trabajar con los niños más pequeños, ayudarles a abrir el navegador y explicar las consignas. En este caso se propone utilizar la actividad de secuencia del curso 1 <https://studio.code.org/s/course1/stage/4/puzzle/1>

Proponer a los alumnos que realicen todos los niveles de la actividad.

Cierre

A modo de cierre retomar el concepto de algoritmo trabajado en clases anteriores y contraponerlo con la noción de programa. Se pueden proponer preguntas tales como: ¿en qué se parecen y en qué se diferencian? Es importante que luego de esta actividad surja que:

- Ambos son secuencias de instrucciones que permiten resolver un problema,
- Ambos están compuestos por instrucciones que no deben ser ambiguas. Se puede ilustrar esto con un ejemplo tal como: si la instrucción es girar, se debe indicar hacia qué lado y cuánto; o si la instrucción es salir por la puerta, y hay varias puertas, se debería explicitar cuál.
- Un programa está escrito para que sea ejecutado por un autómatas, por lo tanto debe estar escrito en un lenguaje específico y se debe respetar el orden en forma estricta. Por ejemplo, si la orde es “sentarse”, el autómatas debe hacerlo independientemente que haya o no silla.
- En la actividad realizada en la computadora, el lenguaje utilizado es mucho más específico: se usan bloques que representan las instrucciones. Estos bloques no pueden estar desconectados: para que formen el programa deben “encastrarse”. En este caso, el autómatas es la computadora, que ejecuta las instrucciones al presionar el botón correspondiente.
- Otro aspecto a destacar es que, aunque se nos ocurran más instrucciones, sólo se disponen de las que están presentes en el programa, con lo cual son las únicas instrucciones disponibles para resolver el problema.

3.3. Enseñando a programar en la escuela secundaria

Al igual que en el caso de la escuela primaria, en esta sección se presenta una sistematización con las actividades trabajadas con estudiantes de escuela secundaria. En el capítulo 6 se plantea una propuesta para la incorporación de ProBots3D y en el capítulo 7 su evaluación.

Es importante aclarar que esta planificación está pensada para escuelas de nivel secundario no técnicas, en las cuales los únicos espacios curriculares donde puede introducirse esta temática son las materias extracurriculares, si es que las tienen, o la materia NTICX (Nuevas Tecnologías de la Información y la Conectividad) correspondiente al 4to. año. Las escuelas técnicas, dependiendo de sus orientaciones presentan más espacios para esta temática, y, según las experiencias realizadas por esta tesista a lo largo de los años, los utilizan con una gran variedad de propuestas.

3.3.1. Sistematización propuesta

Teniendo en cuenta las experiencias realizadas se propuso una organización de contenidos para abordar la temática de programación en la escuela. La misma está dividida en dos años de ejecución, pensando en dos horas semanales de trabajo en el aula.

La tabla 3.7 muestra la organización de contenidos propuesta para el primer año.

La tabla 3.8 muestra la propuesta para el segundo año de la experiencia. En este año, se hace una experiencia con el uso de los robots del “Programando con robots y software libre” y es en esta organización de contenidos donde se propone incorporar el uso de ProBots3D.

Tema	Herramienta/Actividad
Algoritmos y programas	
Concepto de algoritmo	Actividades desconectadas donde se trabaja en la formulación y seguimiento de algoritmos sencillos, identificando las instrucciones y secuencias necesarias para su resolución.
Programas e instrucciones	Se introduce el concepto de “autómata”. Desafíos de Pilas Bloques: “Alien toca botón”, “El gato en la calle” e “Instalando Juegos”. Del curso 2 de code.org, las etapas 3 y 4.
Estructuras repetitivas	Desafíos de Pilas Bloques provistos en la categoría “Autómatas, comandos, procedimientos y repetición”. Los primeros 5 niveles del desafío del Laberinto de “Juegos de Blockly” y la etapa 3 del curso 3 de code.org. Uso de la estructura repetitiva para reconocer patrones y simplificar el programa.
Abstracción procedural	Desafíos de Pilas Bloques provistos en la categoría “Autómatas, comandos, procedimientos y repetición”, ahora utilizando la opción “Mis procedimientos”. Nivel 2 de la herramienta Lightbot.
Uso de Scratch	
Exploración	Se explora la herramienta y se trabaja con los bloques de movimientos, apariencia y control para realizar una historieta animada.
Abstracción y variables	Se plantean problemas que requieren la creación de nuevos bloques para resolverlos. Concepto de variable.
Eventos	Se plantea un juego sencillo donde se introducen los eventos del teclado y la interacción entre personajes.
Estructuras condicionales	
Estructuras condicionales	Sección “Alternativa condicional” de Pilas Bloques. Del curso 2 de code.org, la etapa 13. El desafío “Pájaro” de Juegos de Blockly. Uso de los sensores de Scratch. Se introducen las estructuras condicionales simples y compuestas.
Aspectos de lógica proposicional	Con actividades desconectadas se introducen aspectos básicos de la lógica proposicional y su relación con la programación. Se trabajan los operadores lógicos de negación, conjunción y disyunción y sus tablas de verdad.
Repeticiones condicionales	Sección “Repetición condicional” de Pilas Bloques. Etapas 7 y 8 del curso 3 de code.org. Desarrollo de juegos sencillos con Scratch.
Batallas de robots con RITA	Se presentan actividades que permiten explorar la herramienta y luego generar una competencia con los robots.

Tabla 3.7. Organización de contenidos propuesta para el primer año de aplicación.

Tema	Herramienta/Actividad
Algoritmos y programas	
Concepto de algoritmo y programa	Actividades desconectadas donde se repasa el concepto de algoritmo e instrucciones. Usando el desafíos de Pilas Bloques: “Alien toca botón” para destacar las características de un programa de computadora.
Repaso de conceptos básicos	Usando Scratch realizar un juego sencillo. Se repasan los conceptos de estructuras repetitivas, condicionales, abstracción y uso de variables.
Programando una APP	
MIT APP Inventor	Exploración de la herramienta y desarrollo de una <i>app</i> simple.
Lenguajes textuales: Python	
Programando robots con Python	Se presenta el proyecto “Programando con robots y software libre” y el lenguaje de programación Python. Diferencias con lenguajes basados en bloques. Distintas formas de ejecución.
El <i>framework</i> pilas-engine	Se usa pilas-engine. Se presenta el <i>framework</i> : actores y habilidades.
Funciones y variables	Se presentan los elementos básicos de la interfaz de usuario en pilas-engine. Manejo de sonidos y funciones de dibujo.
Estructuras de control	Se trabaja con las estructuras de repetición y condicionales a través del uso de las funciones de dibujo de pilas-engine.
Eventos y colisiones	Se presentan los conceptos y se plantea el desarrollo de un juego sencillo.

Tabla 3.8. Organización de contenidos propuesta para el segundo año de la sistematización realizada.

3.3.2. Secuencias didácticas

En esta sección se presenta una de las secuencias didácticas de la organización de contenidos propuesta para el segundo año de la experiencia. La misma se centra en la programación con los robots en el lenguaje Python. Esta secuencia es la que articulará luego la propuesta de inclusión de ProBots3D y está pensada para alumnos que ya hayan realizado una primer experiencia con los lenguajes de bloques, en el caso de esta planificación, los alumnos que hayan trabajado las actividades del primer año detalladas previamente.

La figura 3.2 muestra un cuadro resumido de la planificación propuesta para el nivel secundario

visualizando el lugar donde se inserta la secuencia didáctica donde luego se incluirá la herramienta ProBots3D.

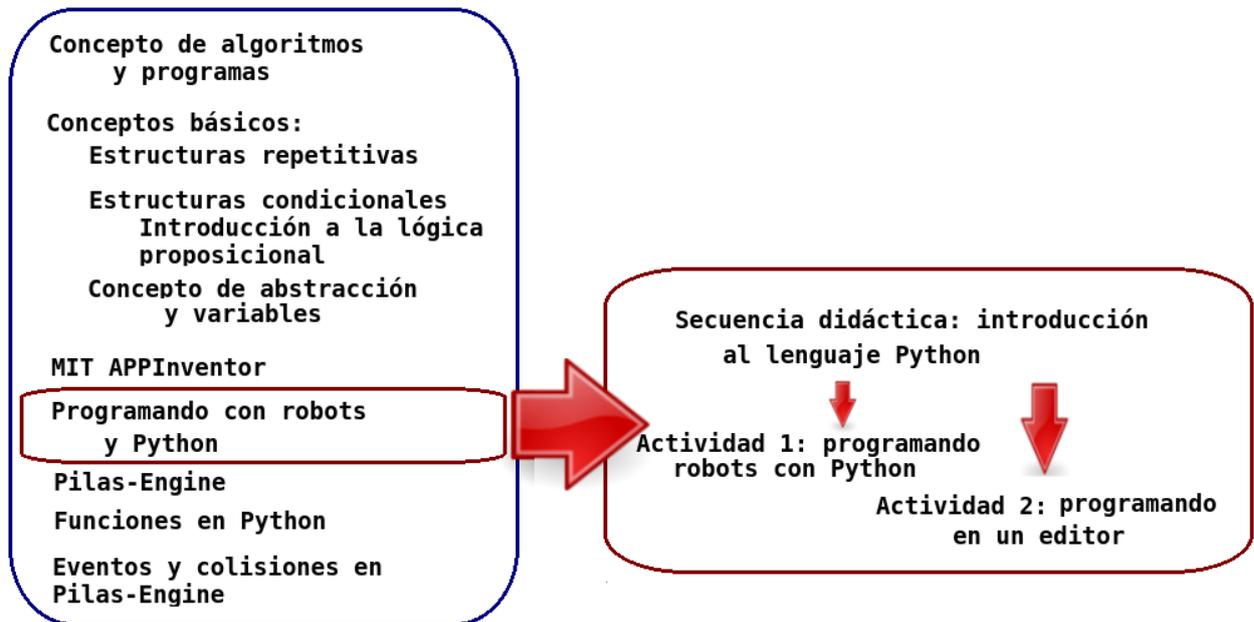


Figura 3.2. Planificación y propuestas para el nivel secundario - correspondiente al 2do. año.

A continuación se detalla la secuencia didáctica denominada “Introducción al lenguaje de programación Python”, la cual propone dos actividades. La primera de ellas es la que articulará con la inclusión de ProBots2D.

3.3.2.1. Secuencia didáctica: introducción al lenguaje de programación Python

El eje central de esta secuencia didáctica es la presentación de un lenguaje de programación textual contraponiéndolo con los lenguajes basados en bloques, previamente utilizados. Para ello, se proponen dos actividades en las que los alumnos podrán trabajar algunos de los conceptos abordados en lenguajes basados en bloques, en un lenguaje textual. La primera actividad, que se describe en detalle en este informe dado que es la que se utilizará luego para introducir ProBots3D, permite presentar los robots y los mecanismos básicos para manipularlos a través del lenguaje Python. La segunda actividad muestra cómo trabajar con un entorno de desarrollo integrado (IDE, por sus siglas en inglés *Integrated Development Environment*). Como esta actividad no tiene relevancia para esta tesis, sólo se presentan los objetivos y modalidad de trabajo.

Objetivos de la secuencia: que los estudiantes logren:

- Comprender las diferencias entre los lenguajes de programación basados en bloques y los lenguajes textuales.
- Aproximarse a la sintaxis básica y forma de ejecución del lenguaje Python.
- Conocer las primitivas básicas para la manipulación de los robots Múltiplo N6.

Actividad 1: programando robots con Python

Objetivos de la actividad: que los estudiantes puedan:

- Conocer las características básicas de los robots Múltiplo N6.
- Programar los robots a través de una consola de Python.

Modalidad de trabajo: actividad en grupos reducidos.

Materiales: un robot por grupo y una computadora.

Desarrollo

Esta actividad se formula en dos momentos: un primer momento donde se presentan los dispositivos con los cuales se trabajará, es decir los robots Múltiplo N6 y los comandos necesarios para su manipulación. En un segundo momento se presenta el lenguaje Python como el lenguaje a utilizar para programar los robots.

Momento 1: introduciendo los robots y sus comandos

Al comenzar la actividad presentar a los alumnos un robot y mencionar sus características principales:

- Es un robot educativo donde tanto el *hardware* como el *software* utilizado son libres, es decir que tanto las especificaciones de la electrónica como la de los programas necesarios para su utilización están accesibles y pueden utilizarse libremente.
- Posee tres ruedas que le permiten desplazarse: dos delanteras con tracción y una rueda trasera que le permite girar.
- Se mueve gracias a dos motores de corriente continua.

- Posee dos sensores de línea y un sensor ultrasónico que permite detectar obstáculos.
- El módulo XBee es el que permite la conexión con la computadora. Se conecta a través de un conector USB, y permite la comunicación con uno o más robots al mismo tiempo (utilizando una misma computadora).

Explicar que para poder trabajar con estos robots es necesario que en la computadora se instalen las librerías necesarias⁹.

Explicar a los alumnos que de ahora en más se trabajará con el lenguaje de programación Python. En este momento se puede explicar que Python es un lenguaje interpretado y que para tener una primera aproximación se trabajará con una consola de comandos. La figura 6.3 muestra una imagen de una consola típica en un sistema Linux. Al abrir esta consola es posible comenzar a escribir las instrucciones que Python interpretará y ejecutará. Puede hacerse una primera aproximación mostrando en la consola de Python sentencias simples tales como:

```
print ("Hola")

x = x + 1

print (x)
```

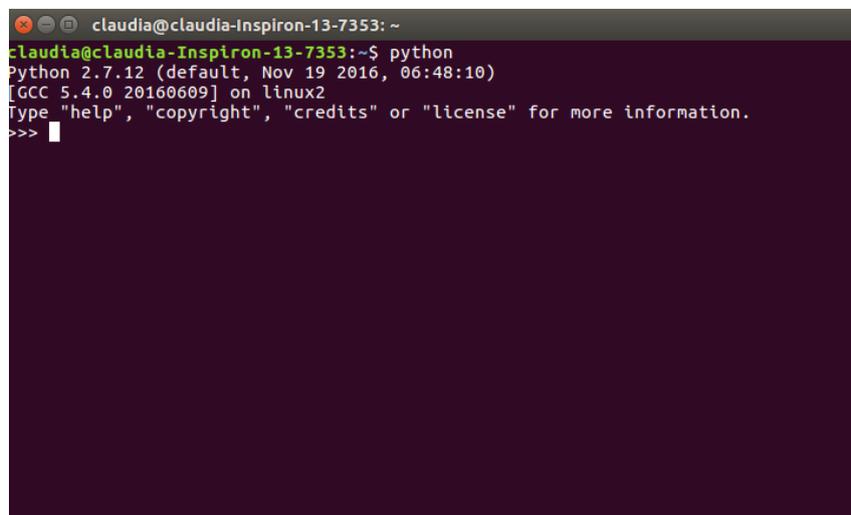


Figura 3.3. Consola de Python.

Como primera actividad con este nuevo entorno se testeará que se tiene instalado todo lo necesario para comenzar a trabajar con los robots. Para ello, indicar a los alumnos que abran la consola de Python y tipeen la siguiente sentencia:

⁹Tanto las librerías necesarias como la forma de instalación depende de los sistemas con que se cuente en cada caso.

```
from duinobot import *
```

Explicarles que con esta sentencia se importará o se agregarán al entorno de Python todas las funciones necesarias para poder trabajar con los robots.

A continuación, indicarles que ahora deben introducir las instrucciones necesarias para poder conectar con el robot. Para ello, pedirles que tipeen las siguientes instrucciones:

```
[1] modulo_comunicacion = Board("/dev/ttyUSB0")
[2] modulo_comunicacion.report()
[3][1, 2, 3]
```

Antes de comenzar a analizar estas instrucciones recordar el concepto de variable. Preguntar: *¿se acuerdan para qué se usan las variables?* . En este ejemplo, *modulo_comunicacion* es una variable con la cual se va a referenciar al módulo de comunicaciones. La línea [1] establece esta asignación y la línea [2] permite saber qué robots están disponibles. Cada robot posee un número identificador y, en el ejemplo, se muestran disponibles los robots con identificador 1, 2 y 3. Para manipular un robot, se lo debe referenciar también a través de una variable. Proponer que tipeen la siguiente sentencia:

```
robot1 = Robot(modulo_comunicacion, 1)
```

robot1 es el nombre de la variable con la que se referencia al robot cuyo identificador es 1 utilizando el módulo de comunicaciones referenciado con la variable *modulo_comunicacion*. A partir de este momento es posible comenzar a darle órdenes al robot.

Momento 2: manipulando los robots a través de la consola de Python

Para comenzar a programar los robots primero se deben conocer cuáles son las instrucciones que entienden los mismos.

La tabla 3.9 muestra los comandos básicos con los que se trabajará en esta primera actividad.

Instrucción	Descripción
<i>forward(velocidad, tiempo)</i>	Avanza a una velocidad y tiempo dado.
<i>backward(velocidad, tiempo)</i>	Retrocede a una velocidad y tiempo dado.
<i>turnLeft(velocidad, tiempo)</i>	Gira a la izquierda a una velocidad y tiempo dado.
<i>turnRight(velocidad, tiempo)</i>	Gira a la derecha a una velocidad y tiempo dado.

Tabla 3.9. Comandos básicos del robot Múltiple N6

Proponer que experimenten con estas instrucciones y comprobar los movimientos del robot. A modo de ejemplo se puede presentar:

```
[1]modulo_comunicacion = Board("/dev/ttyUSB0")
[2]modulo_comunicacion.report()
[3]robot1 = Robot(modulo_comunicacion, 1)
[4]robot1.forward(50,3)
[5]robot1.turnLeft(80,1)
```

Las líneas [1] a [3] son las mismas que ya trabajaron previamente. La línea [4] mueve el robot hacia adelante a una velocidad de 50¹⁰ durante 3 segundos. La línea [5] gira al robot hacia la izquierda a una velocidad de 80 durante 1 segundo.

Cierre

A modo de cierre realizar un repaso de lo visto en la actividad, haciendo hincapié en que el robot Múltipl N6 se programa en Python y que este lenguaje tiene algunas reglas sintácticas estrictas que se irán viendo a medida que avancen las actividades. En particular, en esta actividad se vio la sentencia *import* que permite importar las instrucciones necesarias para manipular el robot y realizar la conexión con el mismo. Luego también se vio como se definen variables en Python: las mismas se expresan como una secuencia de caracteres que deben comenzar con una letra y contener cualquier combinación de letras, números y el símbolo “_”. Una característica importante a tener en cuenta es que Python distingue entre letras mayúsculas y minúsculas. Por lo tanto si se define una variable con el nombre *robot1*, y luego la usamos como *ROBOT1*, Python pensará que se trata de dos variables distintas.

Actividad 2: programando en un editor

Objetivos de la actividad: que los estudiantes puedan:

- Conocer las características de un editor de programas sencillo.
- Reforzar las primitivas para manipular los robots a través de una consola de Python.
- Introducir el concepto de función en Python.

¹⁰La velocidad máxima que el robot alcanza se representa con el valor 100, por lo tanto en este ejemplo el robot irá a una velocidad media. Es importante aclarar que la velocidad real depende también de la carga de las baterías.

Modalidad de trabajo: actividad en grupos reducidos.

Materiales: un robot por grupo y una computadora.

Desarrollo

En esta actividad se presenta un IDE (por sus siglas en inglés de “*Integrated Development Editor*”) y la primera noción de función en Python.

No se detalla la descripción completa de esta actividad dado que no es de relevancia para esta tesis.

3.4. Resumen del capítulo

En este capítulo se presentaron las experiencias realizadas en el aula de la escuela con algunas de las herramientas analizadas en el capítulo 2. Estas experiencias, llevadas a cabo tanto en el nivel primario como secundario, dieron lugar a una propuesta de organización de contenidos para los distintos niveles educativos. En esta organización se describieron las herramientas utilizadas y los conceptos trabajados con cada grupo destinatario.

Al finalizar el capítulo se presentaron las secuencias didácticas que serán utilizadas como propuesta de uso de ProBots3D.

Programar con robots



“La imaginación es más importante que el conocimiento.”

– Albert Einstein

En este capítulo se describe el proyecto “Programando con robots y software libre” a través del cual se trabaja con niños y jóvenes en la enseñanza de la programación utilizando pequeños robots como elementos motivadores. En una primera instancia se describe el proyecto original y las actividades realizadas en el marco del mismo. Luego, se detallan algunas actividades más recientes que incluyen a niños más pequeños, jornadas específicas y algunas herramientas desarrolladas en el marco de este proyecto.

4.1. Programando con robots y *software* libre: el proyecto original

Como se mencionó al comienzo de este informe, “Programando con robots y software libre” es un proyecto de la Facultad de Informática de la Universidad Nacional de La Plata llevado a cabo por

docentes e investigadores del LINTI y en cual también participan alumnos de la facultad.

El objetivo principal de este proyecto es trabajar sobre los conceptos básicos de programación en la escuela, utilizando unos pequeños robots para la realización de las actividades prácticas.

Como se mencionó en el capítulo 2, en Argentina la enseñanza de la programación no está incluida en la currícula escolar oficial de las escuelas. En las escuelas secundarias por lo general, sólo se trabaja con programación en escuelas de orientación técnica, en especial aquellas con especialidad en informática o electrónica; y en las restantes escuelas secundarias por iniciativa particular de algún docente interesado. En todos los casos, las metodologías y herramientas son muy heterogéneas dependiendo, fundamentalmente, del perfil y capacidades de los docentes a cargo. (Díaz J., Banchoff C, Martin S. & López F., 2012)(Díaz et al, 2014). Con respecto a las escuelas primarias, el trabajo en programación es casi inexistente. Sólo algunas iniciativas aisladas, las cuales utilizan principalmente herramientas como Scratch.

En sus inicios, el proyecto “Programando con robots y software libre” tenía como un objetivo secundario que los alumnos trabajen con un lenguaje de programación utilizado en la industria de producción de *software*, en este caso Python, utilizando los robots como elementos motivadores. Esto fue trabajado con más de 300 alumnos de nivel secundario con resultados muy positivos (Díaz et al, 2012).

Teniendo en cuenta que se trata de un proyecto llevado a cabo en una universidad pública, que tiene principalmente impacto directo en la comunidad relacionada a la escuela secundaria, se trabajan tres aspectos laterales no menos importantes asociados al proyecto:

- Incentivar el interés de los niños y jóvenes en el uso de la tecnología con fines educativos.
- Difundir y trabajar con *software* y recursos libres.
- Acercar las propuestas de la Facultad de Informática a la escuela secundaria.

El proyecto “Programando con robots y software libre” inicia sus actividades en el año 2008, basado en una iniciativa desarrollada en el *Georgia Tech Institute* denominada IPRE (*Institute for Personal Robots in Education*)¹. A partir de la visita de un docente del IPRE en ese año, se formó un equipo de trabajo del cual esta tesista es la coordinadora. En sus comienzos, se trabajó con los robots denominados *Scribblers* (“garabateadores” en inglés) que eran los utilizados por el proyecto de Georgia Tech. La propuesta del IPRE tenía dos puntos destacados:

- Los robots utilizados eran muy económicos (U\$S 100 en el año 2008).

¹Sitio oficial del *Institute for Personal Robots in Education* (IPRE): <http://roboteducation.org/>. Último acceso: noviembre 2017.

- Utilizaban el lenguaje Python como lenguaje de programación.

Entre los años 2009 y 2010, se adquirieron siete robots *Scribblers*, importados de Estados Unidos, y se realizó una experiencia en dos escuelas, entregándoles dos robots a cada una de ellas en forma de comodato². La figura 4.1 muestra una imagen de estos robots, los cuales fueron utilizados desde los inicios del proyecto hasta el año 2012.



Figura 4.1. Robot *scribbler* utilizado entre los años 2009 y 2012.

Los robots *Scribblers* son inalámbricos: utilizan la tecnología *bluetooth* para la comunicación entre la computadora y el robot. Poseen sensores que permiten reconocer obstáculos, detectar contrastes de colores y tomar fotografías utilizando una cámara web incorporada. Estos robots, en su momento, tenían otra gran ventaja: su bajo costo. Cuando se comenzó a trabajar en el proyecto, los mismos tenían un costo de U\$S100, y tanto el *software* a utilizar como los materiales didácticos, si bien en idioma inglés, estaban disponibles para su uso.

Los *Scribblers* pueden programarse utilizando el lenguaje de programación Python por medio de una API (“*Application Programming Interface*”) denominada *Myro*³ desarrollada en el *Georgia Tech Institute*. Al inicio del proyecto se adecuaron los materiales disponibles, traduciendo el libro de referencia⁴ al idioma español y adaptando las guías prácticas y las actividades propuestas para los alumnos y docentes de nuestra región. Si bien los robots *Scribblers* eran robustos y de bajo costo, no fue

²Cabe aclarar que los robots adquiridos eran patrimonio de la facultad y, por lo tanto, por cuestiones administrativas no podían ser donados a una institución externa.

³Sitio de referencia: <http://www.roboteducation.org/resources.html>. Último acceso: noviembre 2017.

⁴Manual de Referencia de Myro: http://wiki.roboteducation.org/Myro_Reference_Manual. Último acceso: noviembre 2017.

posible su adquisición por parte de las instituciones públicas de nuestro país, debido a complicaciones en los procesos administrativos para la importación de los mismos. Esta restricción hizo que el proyecto no pueda escalar respecto a la cantidad de instituciones involucradas, utilizando en las experiencias los robots adquiridos por el equipo de trabajo.

En el año 2012, a partir de una iniciativa conjunta con la Fundación YPF de Argentina y con el apoyo de la Dirección General de Escuelas Técnicas de la Provincia de Buenos Aires, se presentó la oportunidad de trabajar en el proyecto “Entre robots, juegos y lenguajes: una experiencia de aprendizaje real en las aulas”, para el cual fue necesario la adquisición de doscientos robots a distribuir en diez escuelas secundarias técnicas de la provincia de Buenos Aires. Por tal motivo, se comenzó a trabajar conjuntamente con la empresa argentina RobotGroup para adecuar uno de los modelos de robots que ellos vendían a las necesidades específicas del proyecto, en particular, la posibilidad de trabajar con los robots en forma inalámbrica y con el lenguaje Python. Se adquirieron los doscientos robots y se distribuyeron los mismos en las diez escuelas previstas, siete de las cuales ya se encontraban trabajando con la Fundación YPF y las tres restantes fueron propuestas por la Dirección General de Escuelas Técnicas de la Provincia de Buenos Aires.

Los robots utilizados corresponden al modelo *Multiplo N6* y la figura 4.2 muestra una imagen de los mismos. Para la interacción con estos robots se utilizó una API desarrollada en la Facultad de Informática de la UNLP en conjunto con la empresa RobotGroup (Lanfranco E., Bogado J., Vilaseca D. & da Silva Gillig J., 2012), permitiendo la reutilización de los materiales utilizados hasta el momento con los robots *Scribblers*.

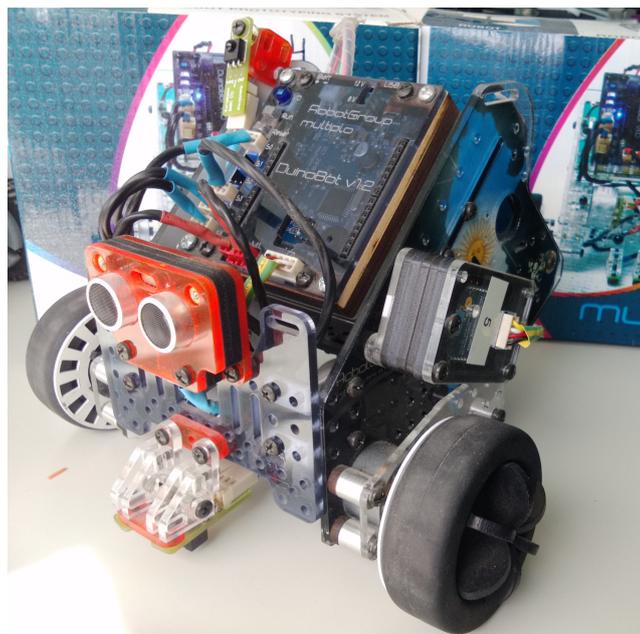


Figura 4.2. Robot Multiplo N6 utilizado a partir del año 2012.

Una de las características más destacadas de los robots *Multiplo N6* es que se basan en *hardware* libre, es decir que la especificación para su construcción está disponible para que pueda ser reproducido por cualquier entidad que así lo requiera. Al igual que los robots *Scribblers*, también éstos son inalámbricos, aunque utilizan otro protocolo de comunicación para la interacción con la computadora. En particular, los primeros modelos adquiridos utilizan un módulo denominado XBee⁵. Este módulo es una solución integrada que brinda un medio inalámbrico para la interconexión y comunicación entre dispositivos. Utiliza el protocolo IEEE 802.15.4⁶ para crear redes punto a multipunto o punto a punto. En el caso de los robots, un módulo se conecta a la computadora y, a través del mismo, es posible conectarse con uno o más robots, lo que hace más interesante la programación, dado que es posible desde un mismo programa comandar varios robots para que interactúen entre sí.

Los robots *Multiplo N6* tienen menos funciones que los *Scribblers*. Pueden desplazarse hacia adelante y atrás, girar a derecha e izquierda y cuentan con sensores que permiten detectar obstáculos y sensores seguidores de líneas. Pero no disponen de una cámara para tomar fotografías ni detectar contrastes de colores como lo hacen los *Scribblers*.

Al igual que los *Scribblers*, estos nuevos robots también pueden programarse en lenguaje Python y la API desarrollada se implementó de manera tal que permite la reutilización de los materiales didácticos existentes, conservando los nombres de los comandos que manipulan los robots. Esta API se encuentra disponible en GitHub⁷.

Como se mencionó en párrafos anteriores, tanto los robots *Scribblers* como los *Multiplo N6* se pueden programar usando el lenguaje de programación Python. Al ser éste un lenguaje de alto nivel de abstracción e interpretado, es sencillo observar los resultados de la ejecución de un programa en forma directa sobre el robot. Con pocas líneas de código es muy simple realizar pruebas que permiten mover el robot.

Como plataforma de trabajo se utilizó el sistema operativo libre GNU/Linux, como una forma más de difundir su uso. Esto introdujo un nuevo desafío debido a que la mayoría de los alumnos y docentes desconocían su uso. Por lo tanto, se trabajó en una capacitación en este sistema operativo en segundo plano. A partir de la implementación del programa Conectar-Igualdad se debió adaptar esta capacitación a las distintas versiones de GNU/Linux instaladas en las *netbooks* entregadas por el programa. Las últimas *netbooks* se entregaron con la distribución Huayra Linux⁸, lo que permitió estandarizar las guías y actividades.

⁵Puede encontrarse documentación sobre XBee en <https://www.digi.com/support/supporttype?type=documentation>. Último acceso: noviembre 2017.

⁶El estándar 802.15.4 define los niveles de red básicos para dar servicio a un tipo específico de red inalámbrica de área personal

⁷Sitio en GitHub: <https://github.com/Robots-Linti/duinobot>. Último acceso: noviembre 2017.

⁸Sitio oficial: <http://huayra.conectarigualdad.gob.ar/>. Último acceso: octubre 2017.

4.2. Modalidades de trabajo

A lo largo de los años, las actividades del proyecto fueron varias y cada una cuenta con modalidades de trabajo distintas. Las mismas abarcan:

- **Demostraciones:** se realizan en exhibiciones o eventos donde se quiere difundir el proyecto y/o la enseñanza de la programación en las escuelas.
- **Actividades con docentes:** principalmente se trata de cursos o talleres básicos para los docentes que poseen los robots en la escuela y/o desean incorporar su uso a la currícula de sus clases.
- **Actividades con alumnos:** talleres que se realizan en el ámbito de la Facultad de Informática destinados solamente a estudiantes de nivel secundario.

4.2.1. Demostraciones

En estos casos se trata de actividades de muy corta duración donde, dependiendo de las edades de los estudiantes participantes, se proponen algunas ejercitaciones que incluyen, en todos los casos, la manipulación de los robots por parte de los asistentes. Es decir, que no se trata solamente de realizar una simple demostración de uso, sino que se incluye alguna actividad interactiva donde los participantes programan los robots.

En los inicios del proyecto, el mismo fue presentado en varios congresos realizados en el país, entre los que se destacan:

- Dos “demos educativas” realizadas en el marco de TE&ET en los años 2009 y 2011. La primera realizada en la ciudad de La Plata, en la Facultad de Informática, y la segunda en la ciudad de Salta.
- Un taller realizado con docentes de todo el país en el marco de las Jornadas de Software Libre realizadas en las 41 JAIIO en agosto de 2012 en la ciudad de La Plata.
- Una competencia de robots en el marco de PyCon 2012, llevada a cabo en la ciudad de Quilmes, donde se otorgó como premio un robot Multiplo N6 al ganador de la competencia.

En el mes de abril de 2012, se desarrolló la exposición TEC La Plata, realizada en la República de los Niños de la ciudad de Gonnet y el proyecto estuvo presente durante los 11 días que duró la

exposición, atrayendo la atención de numeroso público general y de las escuelas que concurrieron a dicha muestra.

Para los niños más pequeños, en el año 2016 se desarrolló una aplicación móvil, a la que se denominó “*DuinoBotSocks*”⁹ que permite la manipulación del robot a través de una tableta o teléfono inteligente con sistemas Arduino, y que muestra los comandos necesarios para realizar cada movimiento, sin requerir el ingreso de los mismos.

Para los estudiantes de nivel secundario, además de utilizar *DuinoBotSocks*, se propone directamente el uso de la consola de comandos de Python.

En el marco del proyecto también se realizaron numerosos eventos donde los mismos estudiantes que participaron de los mismos expusieron sus trabajos. La tabla 4.1 muestra algunas imágenes de los eventos relacionados a estudiantes de nivel secundario y la tabla 4.2 los relacionados a niños de educación inicial y primaria.

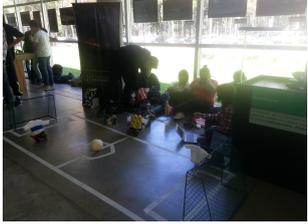
Evento	Descripción	Imagen
Participación en Ferias de Ciencias. Año 2009 a la fecha.	Alumnos de escuelas secundarias presentan sus proyectos en Ferias de Ciencias realizadas en las respectivas escuelas.	
Primera Jornada Programando con Robots y <i>Software Libre</i> . Año 2014	Se realizó en el Planetario Ciudad de La Plata en el mes de octubre. A este evento concurrieron más de 80 jóvenes de 5 escuelas de La Plata, Berisso y Quilmes.	
Segunda Jornada Programando con Robots y <i>Software Libre</i> . Año 2015	Se realizó en el Planetario Ciudad de La Plata en el mes de octubre. A este evento concurrieron más de 70 jóvenes de 5 escuelas de La Plata, Berisso y Ensenada.	

Tabla 4.1. Actividades con jóvenes de escuelas secundarias en el marco del proyecto “Programando con robots y software libre”

⁹Disponible en <https://github.com/Robots-Linti/DuinoBotSocks>. <http://huayra.conectarigualdad.gob.ar>

Evento	Descripción	Imagen
Jornada de cierre del Proyecto “Informática Inclusiva”. Año 2014	En la jornada de cierre de este proyecto se realizó una actividad con niños y niñas entre 6 y 12 años.	
Jornadas con niños de educación primaria. Año 2016	Se realizaron varias experiencias en escuelas primarias de la región. La imagen corresponde a una experiencia realizada en la escuela 67 de City Bell en noviembre.	
Jornada con niños de educación inicial. Año 2016	Como cierre un proyecto con el Jardín Estudiantes de La Plata se realizó una primera experiencia en la enseñanza de programación con la sala de 5 años.	

Tabla 4.2. Actividades con niños de educación inicial y primaria realizados en el marco del proyecto “Programando con robots y software libre”

4.2.2. Actividades con docentes

Si bien en todas las actividades se propone que los docentes acompañen a sus alumnos, en el año 2012 se preparó y dictó una capacitación a más de 150 docentes de las escuelas participantes del proyecto “Entre Robots, juegos y lenguajes: una experiencia de aprendizaje real en las aulas”, ubicadas en las ciudades de La Plata, Berisso, Ensenada, Quilmes, Florencio Varela, Berazategui y Avellaneda. En esta capacitación se utilizaron los veinte robots que cada escuela recibió en el marco de dicho proyecto y las actividades se realizaron en las instalaciones de cada una de las escuelas. Dicha capacitación se basó en las experiencias previas y se organizó en cuatro módulos.

Las primeras actividades estaban orientadas a conocer el entorno de trabajo (GNU/Linux) y a manipular los robots a través de la consola de comandos del lenguaje Python. Mediante esta aplicación se pueden introducir los comandos para manipular el robot y, a través de la conexión inalámbrica, estas órdenes se transmiten al mismo y el resultado puede apreciarse en forma inmediata. A medida que se avanza en la experiencia, se van introduciendo aspectos sintácticos y semánticos de Python y el uso de entornos de programación que favorecen el desarrollo de los programas. En este caso, se utilizó el entorno Geany¹⁰, un entorno de desarrollo sencillo, multiplataforma y multilenguaje.

El programa de esta capacitación puede apreciarse en la tabla 4.3. Este programa fue adaptado a lo largo de los años y es la base de algunos de los talleres realizados en la actualidad.

¹⁰Sitio oficial: <https://www.geany.org/>. <http://huayra.conectarigualdad.gob.ar>

Módulo	Temas
Módulo 1	Descripción del proyecto. Conceptos básicos: introducción al <i>software</i> libre y las licencias Creative Commons. El sistema GNU/Linux. Los robots: características básicas. Introducción a Python.
Módulo 2	Operadores y expresiones. Variables y tipos de datos básicos. Uso y definición de funciones. Las estructuras de control repetitivas.
Módulo 3	Aspectos básicos de lógica. Sensores del robot. Estructuras de control condicionales.
Módulo 4	Tipos de datos. Trabajando con cadenas de caracteres. Manipulación de archivos.

Tabla 4.3. Módulos de la capacitación llevada a cabo en el marco del proyecto “Entre Robots, juegos y lenguajes: una experiencia de aprendizaje real en las aulas”.

Es de destacar también que para acompañar esta experiencia se elaboró un libro de referencia denominado “Manual de programación con robots para la escuela.” publicado en http://robots.linti.unlp.edu.ar/uploads/docs/manual_programando_con_robots.pdf bajo licencia Creative Commons.

4.2.3. Actividades con estudiantes

Desde el año 2009 se realizan en la Facultad de Informática talleres destinados a alumnos de nivel secundario. Durante los primeros seis años, los mismos estaban destinados a estudiantes que estaban en los últimos años de la escuela secundaria y que tenían intenciones de seguir alguna carrera relacionada a Informática e Ingeniería. Estas actividades, conocidas como “pasantías académicas” (Spinosa et al, 2012), permitían a los alumnos no sólo experimentar con las primeras nociones de la programación, sino que también eran un medio para que conocieran el entorno de estudio en la facultad y se relacionasen con estudiantes universitarios.

Desde el año 2015, estas actividades se enmarcan en el proyecto “Extensión en vínculo con la escuela secundaria” (Banchoff Tzancoff C., Queiruga C., Martin S., Miyuki Kimura I., Brown Barnetche M. & Gómez S. ,2015) coordinando las mismas con profesores de distintas escuelas de la región, quienes concurren con sus alumnos a la facultad. En estos casos abarcan a todos los estudiantes de un cierto curso. Los profesores que concurren luego continúan las temáticas vistas en actividades en la escuela.

En todos los casos, se propone que las actividades que los estudiantes realizan con los robots estén

orientadas a actividades sociales y lúdicas, promoviendo la creatividad en el desarrollo de las mismas.

La primera motivación está dada por querer mover al robot, en donde se plantean carreras, batallas, bailes etc., pero pasada esta primera instancia, se trabajan actividades más complejas que requieren conocimientos más avanzados del lenguaje y se proponen actividades interdisciplinarias, convocando a docentes de otras áreas. Entre las experiencias realizadas podemos destacar la realización de obras de teatro y coreografías con participación de alumnos y robots. Con los docentes de área de plástica y lengua, los estudiantes trabajan en la redacción de los guiones y la caracterización de los personajes de las obras de teatro. Con estas producciones las escuelas realizan muestras en jardines de infantes de la zona y en exposiciones escolares.

En lo posible se trata que los estudiantes concurren con las *netbooks* del programa Conectar-Igualdad, si es que las tienen.

La figura 4.3 muestra algunas de estas experiencias realizadas.



Figura 4.3. Algunas actividades realizadas con estudiantes secundarios durante los últimos años.

Dependiendo de la procedencia de las escuelas, estas actividades se desarrollan en cuatro encuentros (puede ser dos en el caso que la escuela se encuentre alejada). Los contenidos de los encuentros se basan en la propuesta de capacitación descripta previamente en la tabla 4.3.

- **Encuentro 1:** el objetivo de este primer encuentro es que los estudiantes pueden conectarse con el robot y programar los movimientos básicos.

- **Encuentro 2:** en este segundo encuentro se retoman los ejercicios realizados previamente y se introduce un entorno de programación. A través de dicho entorno se pueden guardar los programas para luego ejecutarlos. Se trabaja en el desarrollo de programas simples, incorporando la idea de abstracción funcional, y se incorporan en forma intuitiva el uso de los sensores de los robots.
- **Encuentro 3:** a partir del uso de los sensores, se introducen las estructuras de control condicionales. En este encuentro se trabajan sobre las mismas complejizando los ejercicios realizados previamente.
- **Encuentro 4:** se propone el desarrollo de un proyecto que involucre el uso de los robots en forma lúdica. Ejemplos de estos proyectos son: obras de teatro, carreras de robots, etc.

4.3. Programando con Robots y Software Libre: la situación actual

El proyecto continúa en la actualidad. Se trabaja con los robots *Multiplo N6*, aunque algunos de ellos se han modificado de forma tal de utilizar conexión WIFI¹¹ en vez de XBee. Este cambio se debió, fundamentalmente al costo de adquisición de dicho módulo respecto al módulo de conexión WIFI. Aunque este cambio es totalmente transparente al momento de interacción con los robots, dado que se ha respetado la interfaz de comandos e interconexión. Se han integrado algunas aplicaciones más para el trabajo con los robots, las que serán descriptas en secciones más adelante de este capítulo.

Pero el cambio más destacado respecto del proyecto es la inclusión de nuevos destinatarios. Recientemente se incorporó el trabajo con niños más pequeños, de nivel primario, con lo cual se buscaron alternativas al uso del lenguaje de Python, debido a que, con estos niños no se trabaja con lenguajes textuales. En la sección 2.4 se analizaron varias alternativas que se utilizan y, en capítulo siguiente, se describirá ProBots3D y la propuesta de su inclusión en este nuevo contexto.

4.3.1. El simulador en pilas-engine

Como se mencionó en la sección 2.4.10, pilas-engine es un *framework* que provee una biblioteca especialmente programados para el desarrollo de videojuegos sencillos. Entre estos recursos cuenta con un conjunto de “actores”.

El equipo de trabajo del proyecto agregó a la biblioteca de actores provistas por el *framework* un actor nuevo denominado **Robot**. De esta manera se implementó un simulador sencillo¹² de los robots

¹¹WIFI o Wireless Fidelity, se refiere a las redes de datos inalámbricas, las cuales usan ondas de radio para proveer la conexión de red.

¹²En <https://goo.gl/jVY6N6> puede encontrarse mayor información, como así también un video que muestra su uso: <https://www.youtube.com/watch?v=9akPYNCTjzo>. Último acceso: noviembre 2017.

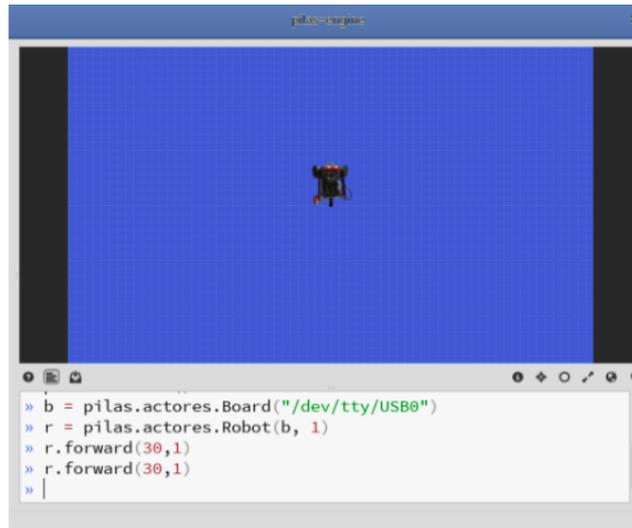


Figura 4.4. El simulador en pilas-engine

Múltiplo N6. La figura 4.4 muestra una captura del intérprete utilizando el simulador. El nuevo actor incorporado muestra una imagen del robot Múltiplo N6 y las instrucciones necesarias para que sea programado en el entorno de pilas-engine. Estas instrucciones son las mismas que utiliza el robot real, por lo que es posible intercalar ambas experiencias. Si bien la utilización de un simulador no es tan atrayente como los robots reales, esta herramienta permitió el trabajo con aquellas instituciones que no cuentan con robots físicos, ya sea que no los pueden adquirir o no cuentan con el número suficiente de los mismos para el trabajo con sus alumnos.

El simulador en pilas-engine fue presentado en la IV Conferencia Internacional de Software Libre, CISL 2013, (Díaz J., Banchoff C., Martín S., Gómez del Mónaco A. & Lanfranco E., 2013) y utilizado en el trabajo con jóvenes de 2do. y 3er. año de escuela secundaria durante el ciclo lectivo 2014. En una primera jornada con los estudiantes se mostraron los robots reales realizando una demostración con los mismos en la cual se presentaron las instrucciones necesarias para su manipulación básica. Luego, el docente a cargo de los cursos continuó su trabajo con el simulador durante las clases siguientes, para finalizar las actividades en una jornada donde los alumnos pueden probar sus programas realizados en el simulador, con los robots reales.

El simulador es muy sencillo y no suple al robot real como elemento motivacional, pero es un buen recurso para continuar, en una segunda etapa de trabajo, incorporando el uso de pilas-engine para el desarrollo de juegos sencillos.

4.3.2. Curso a distancia para docentes con XRemoteBot

Como se mencionó en la sección 2.4, XRemoteBot es una herramienta que permite trabajar con los robots en forma remota a través de Internet. Esta herramienta se pensó, principalmente para el

trabajo con aquellas instituciones que no cuentan con los robots y sabiendo que un simulador no es suficiente como elemento de motivación.

Se desarrolló un curso virtual con docentes del EPET N° 1 de Posadas, provincia de Misiones. Para el desarrollo del curso se utilizó el entorno virtual de enseñanza y de aprendizaje (EVEA) Moodle¹³ y participaron del mismo 7 docentes.

En el EVEA se publicaron las clases y actividades prácticas coordinando días específicos donde se conectó la cámara de video y se dejaron los robots encendidos para su uso.

Esta actividad se llevó a cabo en los meses de abril y mayo de 2016 y permitió trabajar en una primer experiencia de uso de XRemoteBot.

Durante el mes de octubre de 2017 se puso en contacto una docente del Instituto Hindenburg de Eldorado, Misiones para planificar una actividad para el ciclo lectivo 2018. El pasado 11 de noviembre, esta docente y dos de sus estudiantes realizaron unas pruebas básicas donde programaron un robot a través de esta aplicación visualizando el mismo por el *streaming* de video. Esta prueba resultó muy satisfactoria, pudiendo no sólo entender los comandos básicos de los robots sino visualizar, en un tiempo adecuado, la ejecución de los mismos. Por tal motivo se está programando una actividad más abarcativa con alumnos y docentes de varios cursos para el ciclo lectivo 2018.

4.3.3. Niños de primaria: nuevos destinatarios del proyecto

A fines del año 2015 se comenzó a trabajar con niños y niñas de escuelas primarias. Esto obligó a repensar la interfaz de programación con los robots. Si bien Python es un lenguaje sencillo, al tratarse de un lenguaje textual, posee aspectos de sintaxis que sacan de foco el objeto de enseñanza, especialmente en niños y niñas más pequeños. Por este motivo, se comenzó a trabajar con herramientas que utilizan lenguajes visuales basados en bloques o icónicos.

Con estos alumnos se pensó en una metodología totalmente distinta. Antes de introducir el trabajo con los robots, se trabajan los aspectos básicos de programación y la introducción a los lenguajes de bloques con los desafíos de Pilas Bloques o de Code.org (descriptos en la sección 2.4 de este informe). Luego, se introducen los robots reales como elementos motivadores y, en esta etapa es donde se incorpora ProBots3D. Tanto las características de esta herramienta como la metodología desarrollada se describe detalladamente en los siguientes capítulos.

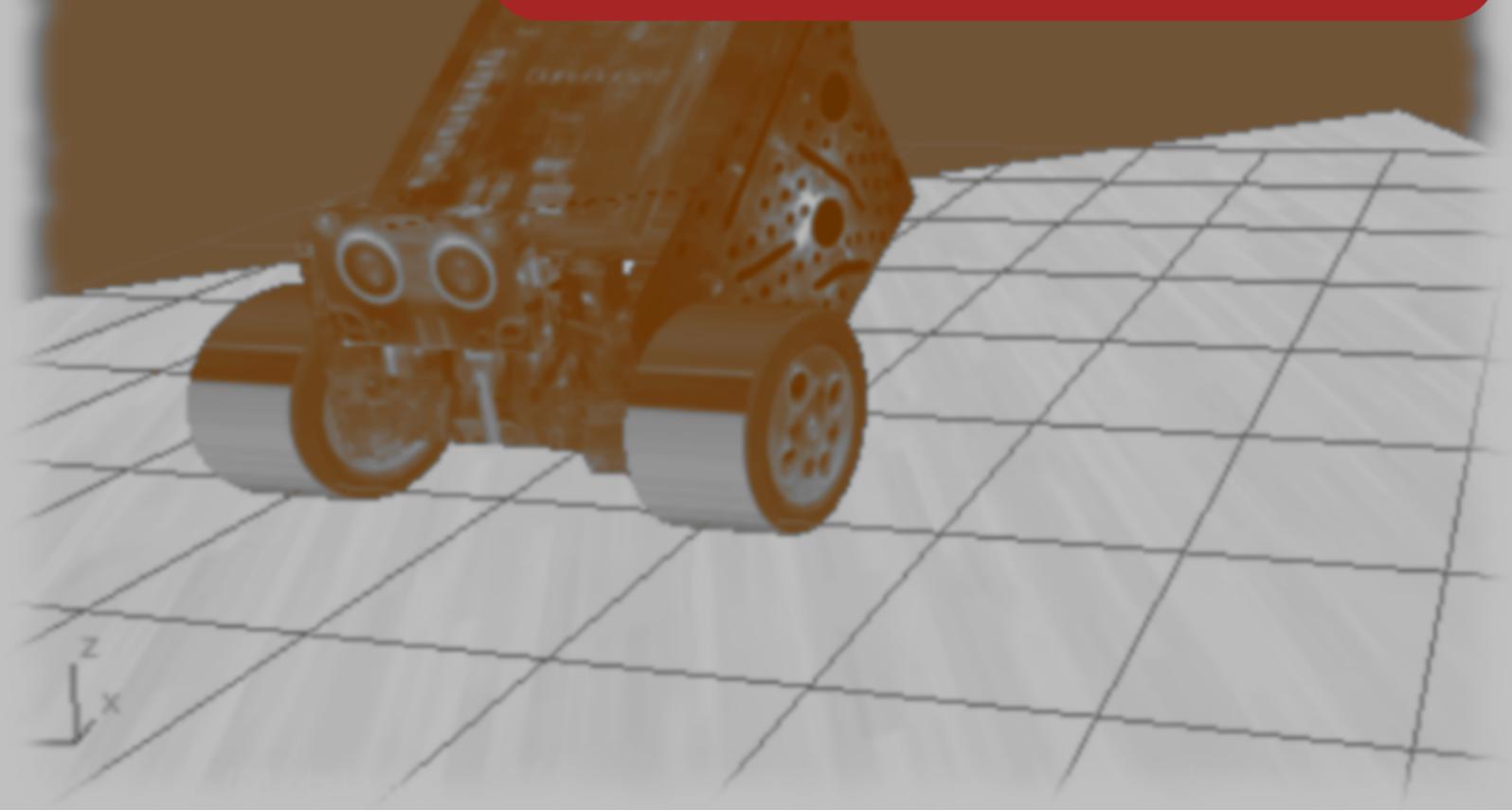
¹³El curso se dictó en el <https://cursos.linti.unlp.edu.ar/>

4.4. Resumen del capítulo

El trabajo con robótica educativa no es algo nuevo y existen muchas iniciativas que proponen su uso. Algunas de ellas utilizando prototipos propios y otras que utilizan robots como LEGO® que, si bien son muy completos y muy atractivos, son bastantes costosos y se basan en tecnologías privativas. Estas dos características complican su adopción en aquellas instituciones que cuentan con escasos recursos, por lo que son usadas en general en iniciativas privadas.

En este capítulo se describió en detalle el proyecto “Programando con robots y software libre”, el cual da marco al trabajo realizado en esta tesis. El mismo se inició para abarcar esta temática en base al uso de recursos libres. Si bien el proyecto fue variando a lo largo de los años, aún en la actualidad despierta interés tanto en los estudiantes, de todos los niveles educativos, como en los mismos docentes. El hecho de incorporar niños más pequeños como destinatarios del proyecto, motivó la incorporación de nuevas herramientas, tales como ProBots3D.

ProBots3D: los robots en 3D



“La mayor manera de predecir el futuro es inventándolo.”

– Alan Kay

En los capítulos anteriores se describió el marco teórico y contextual de esta tesis detallando las actividades e iniciativas donde se aplicará la propuesta elaborada.

En este capítulo se presenta la herramienta ProBots3D¹, desarrollada en el marco de esta tesis. La misma fue diseñada como un complemento para las actividades prácticas que se realizan en las escuelas. Se pensó en un elemento motivador tanto para alumnos como docentes, que permita incorporar aspectos de programación en el aula de la escuela y, al mismo tiempo, mostrar el espacio físico de la Facultad de Informática. Independientemente del nivel educativo con el que se trabaja, siempre se trata de acercar a los niños, jóvenes y sus docentes a la facultad a través de jornadas, talleres y/o simplemente paseos que permitan mostrar no sólo las particularidades de la disciplina, sino también las posibilidades que brinda la universidad. El entorno de ProBots3D permite realizar una primera

¹En el CD que acompaña a este informe se encuentran las dos versiones desarrolladas de este software.

aproximación a este espacio.

5.1. FI3D: la Facultad de Informática en 3D

ProBots3D fue concebida como una integración a FI3D, un modelo de la Facultad de Informática de la Universidad Nacional de La Plata en 3D, que permite realizar un recorrido virtual por las instalaciones de la facultad.

La Facultad de Informática de la Universidad Nacional de La Plata es una facultad muy joven, a la cual ingresan en promedio de alrededor de 800 alumnos por año². Las carreras que se dictan en la misma son muy demandadas, tanto en la región como en otras regiones del país. Muchos alumnos, principalmente los que provienen de regiones más lejanas, desconocen la zona en donde está situada y las instalaciones propias que la misma ofrece.

Para saber dónde está ubicada la facultad geográficamente o cómo son los alrededores, basta con consultar cualquier mapa en Internet, como puede ser Google Map³ o OpenStreetMap⁴. Poder ver el lugar antes de concurrir al mismo genera menos stress (Harari V. & Banchoff C. ,2016) al momento de concurrir al lugar físico real. Para poder complementar y extender la información que otorga un mapa geográfico y, para ser utilizado por varios de los proyectos de extensión que se realizan con alumnos secundarios, potenciales alumnos o visitantes de la facultad, se construyó un modelo 3D representativo de la institución, al cual se lo denominó **FI3D**.

FI3D no fue concebido con un fin específico más que sólo mostrar una versión virtual de la facultad, donde cualquier persona pudiera recorrer y visitar las instalaciones. A partir de este desarrollo se planteó la posibilidad de integrar algunas de las actividades realizadas en la facultad a este modelo. de manera tal de potenciar las mismas con una herramienta motivadora para los niños y jóvenes que la utilicen.

FI3D es un desarrollo de *software* libre, donde tanto los modelos involucrados como las herramientas de desarrollo y el producto final son libres⁵. Usando FI3D como base, se desarrolló una herramienta que permite simular el trabajo con los robots del proyecto “Programando con robots y software libre” y trabajar los primeros conceptos de programación con niños y jóvenes.

En las siguientes secciones se describen algunas de los conceptos relacionados al *software* libre, justificando el por qué de su adopción, las características propias de FI3D y, por último, la herramienta ProBots3D, desarrollada en el contexto de esta tesis.

²Anuario Estadístico 2016: indicadores comparados: <http://www.unlp.edu.ar/indicadores>. Último acceso: noviembre 2017.

³La Facultad de Informática en Google Maps: <https://goo.gl/AQT4nD>

⁴La Facultad de Informática en OpenStreetMap:<https://goo.gl/8QeYAb>

⁵Se puede acceder al código fuente en <https://github.com/clauBanchoff/FI3D>. Último acceso: noviembre 2017.

5.1.1. Las premisas para el desarrollo

Como se mencionó en secciones anteriores, esta tesis propone una contribución al proyecto “Programando con robots y software libre” y está enmarcada en el proyecto de investigación del LINTI “Internet del Futuro: ciudades digitales inclusivas, innovadoras y sustentables, IoT, ciberseguridad, espacios de aprendizaje del futuro”, donde una de las premisas de la gran mayoría de los desarrollos y actividades llevadas a cabo tienen como eje conductor el uso de *software* y recursos libres.

5.1.1.1. El *software* libre como base

El movimiento conocido como *software* libre es mucho más que un estilo de distribución o de adquisición de *software*: propone un estilo de trabajo basado en la cooperación, en la seguridad y en la no-concentración monopólica del poder sobre el *software*. Esta forma de pensar da lugar a una corriente más filosófica o conceptual, que se interesa más por las concepciones acerca del uso que se hace del *software*, y permite contribuir al mejoramiento del conocimiento global, integrando a quienes quieran participar de la construcción del mismo. **El *software* libre es una manera de pensar la tecnología, su construcción, uso y distribución.**

El *software* libre tiene sus orígenes en la década del '70. Más precisamente, en el año 1971, cuando el desarrollador de *software* Richard Stallman, que por entonces trabajaba en el MIT (“Massachusetts Institute of Technology”), comenzó a relacionarse con sus colegas “compartiendo” sus desarrollos de manera tal de facilitar el uso de sus sistemas en otras organizaciones. En el año 1980, se decide actualizar los equipos reemplazando las computadoras que se utilizaban hasta el momento. En las nuevas máquinas, el *software* utilizado hasta entonces no era compatible y los desarrolladores, que hasta ese momento compartían libremente sus sistemas, fueron obligados a firmar un acuerdo de confidencialidad, donde no podían divulgar sus códigos sin orden expresa de la organización donde trabajaban. Es así que Stallman, un día que necesita modificar un *software* para una de sus impresoras, se encuentra con que, debido a estos acuerdos de no divulgación, le niegan el acceso a su propio código fuente. Este hecho, aparentemente sencillo, resulta el puntapié inicial para crear el proyecto que dio origen al *software* libre. Este proyecto, denominado GNU (“GNU’s Not Unix”)⁶, tuvo como principal objetivo crear un sistema operativo totalmente libre. El desarrollo del núcleo o “corazón” del sistema se demoró en un comienzo completándose finalmente con el trabajo de Linus Torvalds, dando origen a lo que hoy se conoce como GNU/Linux, un sistema operativo que consta de múltiples aplicaciones.

Hoy en día, existen cientos de distribuciones de GNU/Linux, cada una orientada a distintas áreas. Ésta es una de las consecuencias directas de la filosofía de desarrollo del *software* libre, la cual se

⁶Sitio oficial: <https://www.gnu.org/home.es.html>. Último acceso: noviembre 2017.

basa en la idea de “compartir y colaborar”, en el trabajo en comunidad y en el libre acceso a los códigos fuentes de los programas, para que todos aquellos que lo deseen puedan enriquecer el trabajo y orientarlo hacia sus propias necesidades o características.

Una definición más precisa de este concepto la podemos encontrar en el sitio web del proyecto GNU, donde se menciona que los usuarios de un *software* libre poseen cuatro libertades esenciales:

- La libertad de ejecutar el programa, para cualquier propósito (**libertad 0**).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que uno quiera (**libertad 1**). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias del *software*. (**libertad 2**).
- La libertad de distribuir copias de las versiones modificadas a terceros (**libertad 3**), posibilitando así a toda la comunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

En resumen, se puede decir que:

El software libre es una cuestión de la libertad que los usuarios tienen para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

En todos los casos, es necesario contar con el código fuente del *software* para poder modificarlo y estudiarlo. No se habla de un *software* que no tiene licencia de uso, sino que se distribuye con “licencias libres”, es decir, con licencias que le dan a los usuarios todas las libertades que enuncia su definición. Con la expansión de esta corriente, se desarrollaron muchos formatos de licencias libres distintas, con más o menos grados de libertad. En (GNU, 2016) puede observarse los distintos tipos de licencias para *software* y su grado de compatibilidad con la definición del *software* libre: GNU GPL (GNU GPL, 2016). En este sitio tienen registrado 125 tipos de licencias de las cuales 93 son libres y 53 compatibles con la licencia GPL.

Sin entrar en detalles sobre los aspectos de las licencias, se buscaron herramientas de desarrollo y formas de publicar los resultados y recursos del proyecto que permitan respetar los principios básicos del *software* libre. FI3D está disponible bajo la licencia GNU GPL v3 (GPL3, 2008).

5.1.1.2. Recursos con licencias libres

Todos los recursos asociados que no son productos de *software* también se encuentran publicados con licencias libres. Particularmente se ha adoptado una licencia *Creative Commons*⁷ que permite

⁷Sitio oficial: <http://www.creativecommons.org.ar/licencias>. Último acceso: noviembre 2017.

distribuir tanto los modelos y la documentación como las guías didácticas de manera tal que las mismas modificarse y compartir libremente. Para ello se eligió la licencia “Reconocimiento – Compartir Igual” de este conjunto de licencias.

Uno de los objetivos de las licencias Creative Commons es el de promover la publicación y divulgación de la cultura en general, por lo que ofrecen distintas variedades para animar a los autores a compartir sus obras con las condiciones que ellos prefieran. Cabe aclarar que, en la gama de licencias ofrecidas, existen algunas que no encuadran en la definición de “libres”, como por ejemplo las versiones “No comercial” y “Sin obra derivada” dado que las mismas restringen parte de su uso.

5.1.2. Características de la implementación

Como se mencionó anteriormente, FI3D se pensó para sensibilizar a los nuevos ingresantes a la facultad y sus entornos familiares, respecto de los espacios físicos dentro de la misma: las oficinas administrativas, las aulas, patios, biblioteca entre otras áreas.

FI3D brinda un escenario en 3D basado en la estructura del edificio de la facultad. Básicamente, permite realizar un recorrido virtual por las distintas aulas, oficinas y jardines utilizando un avatar. El usuario, a través de este avatar, puede recorrer las instalaciones del edificio, trasladándose a través del mismo⁸.

La figura 5.1 muestra una captura de la imagen de ingreso a la aplicación, en donde puede visualizarse un avatar, a través del cual se podrá realizar el recorrido, y la puerta de ingreso a la facultad.

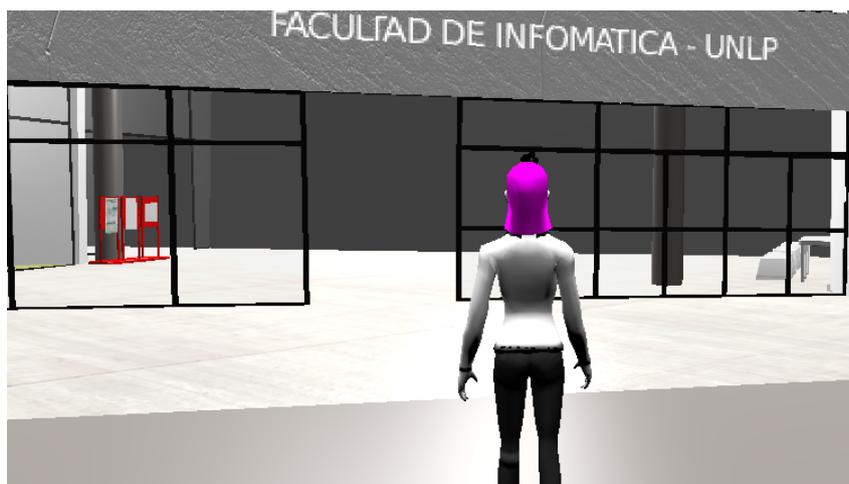


Figura 5.1. FI3D: puerta de acceso al edificio.

⁸Algo importante de destacar es que, a través de este recorrido, también se puede acceder a una vista virtual de la terraza del edificio donde se encuentran instalados unos paneles solares: 72 módulos foto-voltaicos que inyectan energía a la red de la facultad y contribuyen al abastecimiento de energía del edificio. Si bien este escenario no tienen ningún efecto específico sobre el desarrollo, provee un marco ideal para el planteo de otras nuevas iniciativas futuras basadas en FI3D, como podría ser la explicación y actividades de energías alternativas, como es la energía solar.

Si bien en diciembre de 2015 se inauguró una nueva etapa del edificio, con los consiguientes espacios sumados, los mismos aún no están incluidos en este prototipo.

5.1.2.1. Blender para el modelo y el motor de la aplicación

Como se mencionó anteriormente, uno de los objetivos planteados para este desarrollo fue la utilización de *software* libre. Esto dio lugar a otro trabajo de investigación referido a estas herramientas y las posibilidades reales de uso. Se analizaron varias herramientas que cumplen con la premisa y se decidió por el uso de Blender⁹ para el desarrollo del primer prototipo.

Blender es una herramienta ampliamente utilizada desde hace muchos años. Permite la creación de modelos 3D y 2D, animaciones, simulaciones, “renderizado”, composición y seguimiento de movimiento y creación de videojuegos. Si bien Blender es una herramienta con la cual se pueden crear aplicaciones sin escribir líneas de código en un lenguaje de programación, provee una API que contiene las funciones y módulos que pueden ser utilizados por otro *software*. Esta API permite no sólo la codificación de los *scripts* que componen la aplicación, sino que permite personalizar la herramienta y ampliar su funcionalidad. La API provee dos módulos importantes para su uso: BGE (“*Blender Game Engine*”), que contiene la funcionalidad para acceder a los elementos que componen la aplicación que se intenta desarrollar y, BPY (“*Blender Python*”), que permite acceder a los datos, clases y funciones del propio Blender, permitiendo crear nuevos elementos y modificando el entorno mismo provisto por Blender.

Si bien en un primer momento sólo se pensó en utilizar esta herramienta para el modelado de los recursos necesarios, luego se decidió continuar con el desarrollo del *software* manteniendo el mismo entorno.

El modelado de los recursos estuvo a cargo de Máximo Bibiloni, estudiante avanzado de la carrera Licenciatura en Diseño Multimedial, quien colaboró muy estrechamente en el desarrollo de los modelos originales de este trabajo.

5.1.2.2. FI3D 1.0: el primer prototipo

FI3D es una aplicación de escritorio disponible para los sistemas Microsoft Windows y GNU/Linux.

El modelo correspondiente a la versión inicial de este *software* usada en esta tesis abarca:

- La puerta de acceso principal del edificio.
- El hall de entrada, la biblioteca y los pasillos correspondientes a la planta baja donde aparecen las puertas de las aulas.

⁹Sitio oficial: <https://www.blender.org/>. Último acceso: noviembre 2017.

- Las escaleras que permiten acceder al primer y segundo piso.
- El acceso y pasillos del primer piso, donde aparecen las puertas de las aulas.
- La sala de PC del primer piso y el acceso a la misma.
- El hall del segundo piso y los pasillos laterales.
- La entrada de acceso al LINTI.
- El patio interno detrás de la biblioteca y los espacios abiertos alrededor del buffet.
- La cartelería, mesas, bancos y macetas presentes en la planta baja.

Las aulas y dependencias que no se encuentran aún modeladas, aparecen con sus respectivas puertas cerradas y no es posible abrirlas.

Además de la estructura del edificio y algunos de los elementos presentes como las mesas y los carteles, también se modeló un avatar que el usuario utiliza para moverse en las instalaciones.

La funcionalidad provista por este primer prototipo permite desplazar el avatar por todo el modelo con la interacción del usuario a través del uso del teclado, usando las teclas de flechas. El *software* brinda una pequeña ayuda con los atajos de teclado utilizados para su uso a través de la tecla “F1”.

La figura 5.2 muestra algunas de las capturas de la aplicación FI3D desarrollada. En la misma pueden observarse dos espacio físicos reales de facultad y su correspondiente representación en FI3D.



Figura 5.2. FI3D 1.0: vistas de espacios físicos de la facultad y su representación en FI3D.

5.2. ProBots3D: los robots Múltiplo N6 en FI3D

Como se mencionó en la sección 4.3, en el marco del proyecto “Programando con robots y software libre” se desarrollaron varias herramientas para complementar el uso de los robots. Entre ellas, la aplicación DuinoBotSocks, la cual está siendo utilizada actualmente para introducir las actividades iniciales con los robots tanto en niños pequeños como en los jóvenes adolescentes. Pero esta herramienta no muestra una simulación de los robots, sino que requiere contar con los robots reales y conectarse a los mismos a través de una conexión WIFI. Probando esta aplicación en las distintas actividades surge la idea de desarrollar una nueva aplicación que, siguiendo con la misma idea, permita trabajar con un simulador de los robots y poder encarar actividades sin la necesidad de contar con estos recursos. Si bien ya se cuenta con un simulador desarrollado en pilas-engine, el mismo es un entorno 2D en el que, además de las primitivas del robot, también se encuentra todos los restantes recursos del *framework* utilizado.

Al comienzo del desarrollo de ProBots3D se pensó sólo en una herramienta que diera apoyo a las actividades de nivel secundario. Por tal motivo, pareció totalmente adecuado basar el desarrollo en FI3D, de manera tal que además de trabajar los aspectos de programación básicos relacionados al proyecto, se comenzara a familiarizar a los estudiantes con el lugar físico donde se sitúa la facultad promoviendo a ProBots3D como un “espacio anticipatorio” (Harari et al, 2016).

Dado que en el transcurso del desarrollo de ProBots3D se comenzó a trabajar con niños de escolaridad primaria, igualmente se dejó FI3D como escenario de base debido a que esta tesista considera muy importante que los niños, jóvenes y sus grupos cercanos se familiaricen con los espacios de la universidad desde edades tempranas (Harari y Harari, 2015).

5.2.1. La prueba de concepto

Antes de definir un prototipo funcional de ProBots3D se desarrolló una prueba de concepto del mismo que permitió evaluar las características a incluir en una versión más completa. Una prueba de concepto es un desarrollo resumido de manera tal de poder evaluar las potencialidades de una idea o propuesta antes de desarrollar siquiera un primer prototipo (Pentakalos, 2008). En esta prueba de concepto, se abordan las primeras actividades relacionadas a conocer el robots y las primitivas básicas. En el capítulo 6 se describirán en detalle cada una de ellas y su integración con el resto de las actividades.

Como se mencionó en la sección anterior, esta prueba de concepto se basa en la herramienta FI3D, al cual se agregaron varios modelos de los robots Múltiplo N6 que se encuentran dispersos por distintos

puntos del edificio. Los mismos tienen asociados desafíos o consignas que los alumnos deben resolver. Cada una de ellas aborda distintos aspectos de la programación con los robots.

Al igual que en FI3D, se ingresa al edificio virtual manipulando un avatar que, al hacer contacto con los robots, despliega los desafíos de programación.

La figura 5.3 muestra dos capturas de ProBots3D: una correspondiente al ingreso al edificio y otra con una imagen del hall de entrada donde puede visualizarse uno de los robots ubicado cerca del ascensor.



Figura 5.3. Captura del ingreso al edificio y del *hall* de entrada en Probots3D.

Cada desafío presenta una interfaz similar a la mostrada en la figura 5.4 en la cual aparece el robot a programar, una pizarra donde se muestra el programa y una sección con las instrucciones básicas en forma icónica. De acuerdo al desafío puede aparecer uno o más obstáculos u otros elementos los cuales permiten definir la consigna específica. En el caso que se muestra en esta figura, se trata del primer desafío de la versión para escuela primaria, el cual solicita que se programe al robot de manera tal de poder acercarse al cubo rojo, pero con la restricción de usar sólo dos instrucciones.



Figura 5.4. Desafío inicial en Probots3D destinado a nivel primario.

Pensando en los estudiantes con que trabaja el proyecto, se desarrollaron dos versiones de ProBots3D: una para poder trabajar con niños de nivel primario y otra para jóvenes de escuelas secundarias.

Ambas versiones se muestran en la figura 5.5.



Figura 5.5. Dos versiones de ProBots3D desarrolladas en la prueba de concepto de esta tesis.

En la versión destinada a alumnos de nivel primario, se presentan desafíos muy simples y las instrucciones y textos presentes se encuentran en idioma español y letras mayúsculas imprenta, de manera tal de poder trabajar tanto con los niños de primero como de segundo ciclo¹⁰.

En la versión destinada a alumnos de nivel secundario, la pizarra muestra las instrucciones en el lenguaje Python y una de las consignas además permite experimentar libremente con el robot pudiendo así explorar las instrucciones provistas. También se visualizan las instrucciones necesarias en Python para realizar la conexión con el robot, aunque esto no es necesario en esta modalidad virtual, permite fijar dichas instrucciones para el momento de trabajar con los robots reales.

En ambas versiones los desafíos están pensados para trabajar los aspectos básicos de un programa de computadora, con lo cual los desafíos proponen desplazar al robot por distintos lugares del escenario utilizando las instrucciones primitivas. En esta prueba se incluyeron las instrucciones mostradas en la tabla 5.1. Cabe aclarar que la API para programar los robots reales incluye primitivas parametrizadas, pudiendo indicar la velocidad y el tiempo para cada instrucción de movimiento. Sin embargo, con el fin de no complicar en esta instancia con la noción de parámetros, en ProBots3D las instrucciones de

¹⁰En las distintas experiencias de trabajo con escuelas primarias, una crítica y pedido constante de los docentes es que para trabajar con los niños más pequeños es indispensable que los textos se encuentren en letra imprenta mayúscula. Dado que para los mayores esto es indistinto, en esta versión se dejaron todos los textos en este tipo de letra.

movimiento desplazan al robot virtual una distancia fija a una misma velocidad, aunque en la versión para el nivel secundario estos parámetros se visualizan con un valor fijo.

	Instrucción	Primitiva en Python	Descripción
	AVANZAR	forward(velocidad, tiempo)	Esta instrucción permite que el robot avance hacia adelante.
	RETROCEDER	backward(velocidad, tiempo)	Esta instrucción es la inversa a la anterior. Permite movilizar al robots hacia atrás una distancia fija a una misma velocidad..
	DERECHA	turnRight(velocidad, tiempo)	Esta instrucción, como su nombre lo indica, permite que el robot gire a la derecha a una velocidad constante y un ángulo fijo ¹¹ .
	IZQUIERDA	turnLeft(velocidad, tiempo)	Esta instrucción permite que el robot gire a la izquierda, con las mismas características de la instrucción anterior.

Tabla 5.1. Primitivas para programar en ProBots3D y sus equivalentes en la API de los robots reales.

Además de elegir las instrucciones para programar al robot, es posible ejecutar y reiniciar el código. Ambas acciones descriptas en la tabla 5.2.

	Descripción
	Esta acción permite que el robot comience a ejecutar las instrucciones que componen el programa desplazándolo por el escenario.
Clic medio del <i>mouse</i>	Al hacer clic con el botón del medio del <i>mouse</i> , se reinicia el escenario del desafío: el robot se mueve a la posición inicial y se reinicia en la pizarra el programa, eliminando todas las instrucciones previas.

Tabla 5.2. Primitivas para programar en ProBots3D.

El robot se manipula únicamente a través de su programación. Es decir que no existe forma de moverlo sin escribir un programa que lo haga. Para esto, se deben seleccionar con el *mouse* los íconos correspondientes a las instrucciones adecuadas, las cuales se van agregando en la pizarra ubicada en el entorno formando el programa. Como se mencionó anteriormente, en idioma castellano e imprenta mayúscula para la versión de nivel primario y en lenguaje Python para la versión para el nivel secundario.

La consigna a realizar con el robot aparece al pie de la aplicación. Estas consignas indican la acción que debe realizar el robot. Por ejemplo, moverse hacia algún lugar específico de su entorno teniendo en cuenta ciertas restricciones como ser, utilizar un determinado número de movimientos o pasar por un determinado objeto.

Los desafíos propuestos tiene una complejidad incremental. La prueba de concepto desarrolla tres consignas simples que permitirán evaluar la manipulación de la herramienta y su vinculación con las restantes actividades del proyecto.

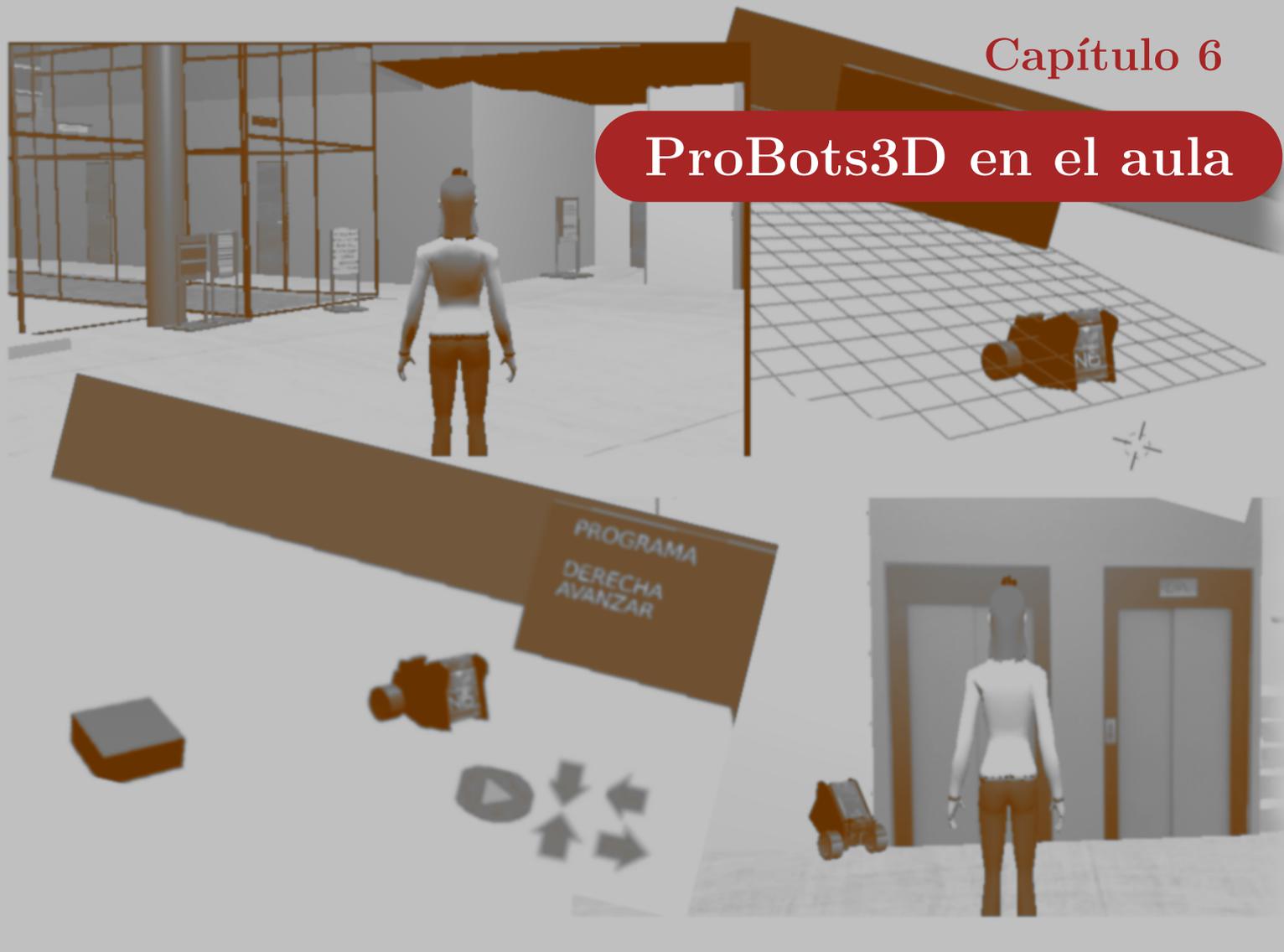
Al finalizar cada desafío se notifica si fue realizado en forma correcta o incorrecta. Dada la simplicidad de estos primeros desafíos, no se presentan ayudas en los mismos, pero para lo desafíos que requieran mayor nivel de programación se prevé la incorporación de ayudas y guías.

5.3. Resumen del capítulo

En este capítulo se describió FI3D, desarrollada por el equipo de trabajo del LINTI, pensada y realizada con el objetivo de poder ofrecer a los futuros ingresantes de la Facultad de Informática de la UNLP un instrumento que les permita sensibilizarse con el lugar donde iniciarán sus estudios universitarios. Debido a su licenciamiento libre pudo ser utilizada como base para crear la herramienta en la que se basa esta tesis de maestría, a la que se denominó: ProBots3D. El uso de *software* libre no sólo permite reutilizarlo en el desarrollo de nuevas aplicaciones, sino que también permite escalarlos y modificarlos para que las mismas puedan ser adaptadas y ajustadas a otras necesidades. Teniendo en cuenta esto y, pensando que el escenario virtual de las instalaciones de la facultad podría servir como base para propuestas de trabajos de alumnos, se eligió desarrollar el trabajo con Blender. Si bien la herramienta utilizada presentó algunas limitaciones, se piensa continuar con ella en la etapa de desarrollo y luego, en una segunda etapa, migrar a otra herramienta de *software* libre. Aunque existen otras herramientas que son privativas, muy populares y que podrían ser más eficientes a la hora de poner en producción el *software*, el mismo no podría ser reutilizado para otros contextos.

ProBots3D, desarrollada en el marco de esta tesis, es una herramienta que no sólo permite trabajar sobre los primeros aspectos de programación, sino que además enmarca estas actividades en un entorno real, la Facultad de Informática, con una réplica de los robots reales.

Se generaron dos versiones de ProBots3D teniendo en cuenta que será usado por niños de escuela primaria y por jóvenes de nivel secundario adecuando las salidas textuales y las consignas en cada versión.



“Todo el mundo debería aprender a programar, porque te enseña a pensar.”

– Steve Jobs

En este capítulo se describe la propuesta de uso de ProBots3D, teniendo en cuenta los dos niveles educativos con los que se trabajará: primario y secundario. Esta propuesta de uso se basa en la prueba de concepto desarrollada, que incluye las funcionalidades necesarias.

Se presentan las secuencias didácticas y las actividades incluidas en las mismas.

6.1. Incorporando ProBots3D a las actividades en el aula

En el capítulo 3 se presentó una propuesta de organización de contenidos para los niveles primario y secundario, y una secuencia didáctica en cada nivel educativo.

En esta sección se presentan las modificaciones realizadas a dichas secuencias con la inclusión de ProBots3D. Cabe aclarar que la organización general de los contenidos para ambos ciclos educativos

no ha sufrido modificaciones.

6.2. ProBots3D en la escuela primaria

La secuencia didáctica en la cual se propone incorporar ProBots3D en el nivel primario es la denominada “**Algoritmos y programas**”, presentada en la sección 3.2.2.1 de este informe. El objetivo de esta secuencia didáctica es introducir la definición y las características básicas de un **programa de computadoras**. Esta secuencia, compuesta originalmente por una única actividad denominada “Mi primer programa”, agrega una nueva actividad, denominada “Programando robots”, en la cual se propone, a través del uso del *software* ProBots3D, reforzar lo visto hasta el momento e introducir el concepto de **lenguaje de programación**. Como la actividad 1 de la secuencia didáctica se mantiene sin modificaciones, en esta sección sólo se mencionan sus objetivos, modalidad de trabajo y el cierre, de manera tal de poder visualizar más fácilmente la vinculación con la actividad 2, la cual completa la nueva versión de esta secuencia didáctica.

La figura 6.1 muestra la organización de contenidos propuesta para el nivel primario¹, destacando la temática donde se trabajará con la secuencia didáctica “Algoritmos y programas”, que se modifica con la inclusión de la herramienta ProBots3D.

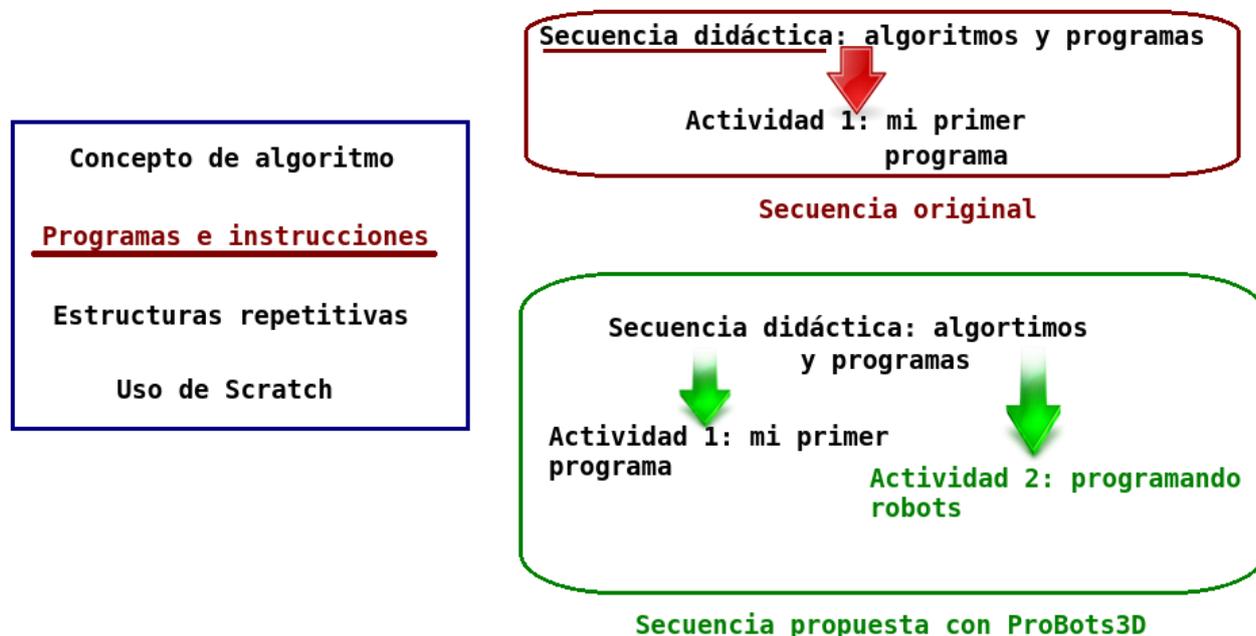


Figura 6.1. Propuesta de uso de ProBots3D en el nivel primario.

A continuación se describe la secuencia didáctica completa.

¹Cabe aclarar que para simplificar este gráfico se presenta la propuesta más completa, correspondiente a los años 5to. y 6to. Los años anteriores no incluyen el tema “Estructuras repetitivas”

6.2.1. Secuencia didáctica: algoritmos y programas - incluye el uso de ProBots3D

El eje central de esta secuencia didáctica es introducir las características de un **programa de computadoras**.

Objetivos de la secuencia: que los alumnos logren:

- Comprender las diferencias entre un algoritmo expresado en lenguaje natural y un programa de computadora.
- Comprender los conceptos de instrucción, programa y secuencia.
- Aproximarse al concepto de lenguaje de programación.

Actividad 1: mi primer programa

Objetivos de la actividad: que los alumnos puedan:

- Identificar las instrucciones que conforman un programa.
- Comprender que un programa de computadora debe escribirse en un lenguaje específico y no ambiguo.

Modalidad de trabajo: un primer momento a trabajar con todo el grupo y un segundo momento a trabajar en grupos reducidos.

Edades de los destinatarios: al tratarse de una actividad inicial, la misma puede realizarse con niños y niñas de 6 a 12 años. Se proponen dos herramientas distintas en el caso de primer y segundo ciclo en el momento 2 de la actividad.

Materiales: una computadora por grupo con acceso a Internet².

Desarrollo:

El desarrollo completo de esta actividad puede verse en la sección 3.2.2.1.

²Para este trabajo, en el ciclo lectivo 2017, se desarrolló una aplicación que puede suplir esta actividad y se trabaja en forma desconectada en aquellas instituciones que no cuentan con acceso a Internet. La misma se encuentra en etapa de prueba.

Cierre

A modo de cierre retomar el concepto de algoritmo trabajado en clases anteriores y contraponerlo con el de programa. Se pueden proponer preguntas tales como: ¿en qué se parecen y en qué se diferencian? Es importante que luego de esta actividad surja que:

- Ambos son secuencias de instrucciones que permiten resolver un problema,
- Ambos están compuestos por instrucciones que no deben ser ambiguas. Se puede ilustrar esto con un ejemplo: si la instrucción es girar, se debe indicar hacia qué lado y cuántos grados o durante cuánto tiempo; o si la instrucción es salir por la puerta, y hay varias puertas, se debería explicitar cuál.
- Un programa está escrito para que sea ejecutado por un autómatas, por lo tanto debe estar escrito en un lenguaje específico y se debe respetar el orden en forma estricta. Por ejemplo, si el orden es “sentarse”, el autómatas debe hacerlo independientemente que haya o no silla.
- En la actividad realizada en la computadora, el lenguaje utilizado es mucho más específico: se usan bloques que representan las instrucciones. Estos bloques no pueden estar desconectados: para que formen el programa deben “encastrarse”. En este caso, el autómatas es la computadora, que ejecuta las instrucciones al presionar el botón correspondiente.
- Otro aspecto a destacar es que, aunque se nos ocurran más instrucciones, sólo se disponen de las que están presentes en el programa, con lo cual son las únicas instrucciones disponibles para resolver el problema.

Actividad 2: programo mi robot

Objetivos de la actividad: que los alumnos puedan:

- Reforzar el concepto de programa como una secuencia de instrucciones.
- Comprender el concepto de lenguaje de programación.

Modalidad de trabajo: grupos reducidos.

Edades de los destinatarios: al tratarse de una actividad inicial, la misma puede realizarse con niños y niñas de 6 a 12 años.

Materiales: una computadora por grupo con acceso al *software* ProBots3D.

Desarrollo:

Al comenzar la actividad retomar con los alumnos el concepto de programa. Recordar que un programa es una secuencia de instrucciones que un autómata debe ejecutar para resolver una tarea. También se puede retomar el concepto de autómata y las diferencias entre una persona y un autómata, al momento de seguir un algoritmo. Pedir a los alumnos que mencionen un ejemplo de autómata. Se espera que relacionen el concepto con una computadora o un robot.

Pedir a los alumnos que ejecuten el *software* ProBots3D explicando previamente que este programa les permite ingresar y recorrer el edificio de la Facultad de Informática de la UNLP, y que, en algunos lugares hay escondidos unos robots que deben encontrar para programarlos. Explicarles que los robots que ven en la aplicación son una réplica de los robots reales con los que se trabaja en un proyecto en la Facultad³.

Indicar a los alumnos que deben encontrar al menos uno de los robots y resolver el desafío propuesto.

Cierre

A modo de cierre dialogar con los alumnos de manera tal de reforzar las siguientes ideas:

- Existen distintas formas de escribir nuestros programas. Para eso se usan los **lenguajes de programación**. En las actividades anteriores usaron bloques que se encastran y en ProBots3D las instrucciones se eligen de una paleta de instrucciones icónicas que representan las instrucciones textuales que se ven en la pizarra.
- No hay una única forma de resolver un problema. Seguramente en alguno de los desafíos surgieron más de una forma de resolverlo. Por lo tanto es importante hacer notar que distintos programas pueden resolver el mismo problema.

6.3. ProBots3D en la escuela secundaria

La secuencia didáctica en la cual se propone incorporar ProBots3D es la denominada “**Introducción al lenguaje de programación Python**”, desarrollada en la sección 3.3.2 de este informe. Los objetivos esta secuencia didáctica son introducir la sintaxis básica del lenguaje Python y conocer las primitivas básicas para la manipulación de los robots Múltiplo N6.

A esta secuencia se le agrega una actividad inicial en la cual se propone el uso de ProBots3D. Como se mencionó anteriormente, en la versión de ProBots3D para el nivel secundario, aparecen las instrucciones primitivas del robot Múltiplo N6, por lo tanto, se utiliza esta herramienta como

³En aquellos casos donde sea posible, se llevará a los alumnos a la Facultad de Informática para poder realizar una experiencia con los robots reales o se organizará una jornada en la escuela donde se trabajará con los mismos.

una primera aproximación a la API del robot. La figura 6.2 muestra la organización de contenidos propuesta para el nivel secundario y la secuencia didáctica donde se incluye el uso de la herramienta ProBots3D.

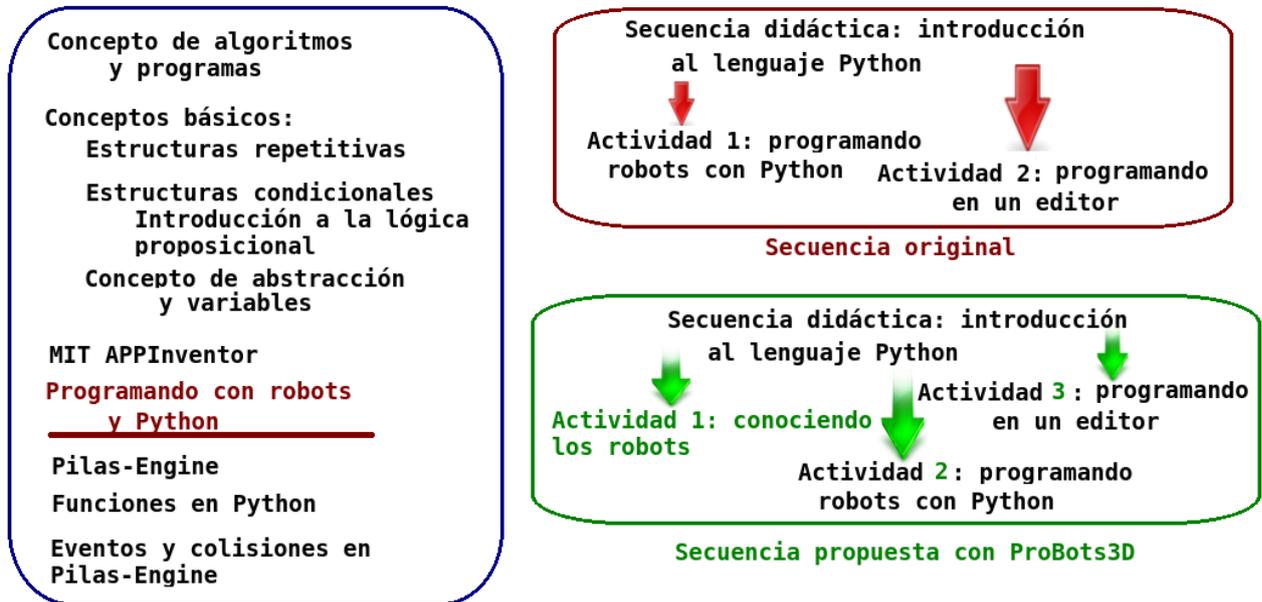


Figura 6.2. Propuesta de uso de ProBots3D en el nivel secundario.

De esta manera esta secuencia didáctica queda conformada por tres (3) actividades. Como la actividad 3 no fue modificada, en esta sección sólo se mencionan sus objetivos y modalidad de trabajo. A continuación se describe la secuencia didáctica completa.

6.3.1. Secuencia didáctica: introducción al lenguaje de programación Python - incluye ProBots3D

El eje central de esta secuencia didáctica es la presentación de un lenguaje de programación textual contraponiendo con los lenguajes basados en bloques, previamente utilizados. Para ello, se proponen tres actividades en las que los alumnos podrán trabajar algunos de los conceptos trabajados en lenguajes basados en bloques, en un lenguaje textual. La primera actividad introduce las primitivas del robot Múltiple N6 a través de la herramienta ProBots3D. La segunda actividad permite presentar los robots reales y los mecanismos básicos para manipularlos a través del lenguaje Python. La tercera actividad muestra cómo trabajar con un entorno de desarrollo integrado (IDE).

Objetivos de la secuencia: que los alumnos logren:

- Comprender las diferencias entre los lenguajes de programación visuales y los lenguajes textuales.
- Aproximarse a la sintaxis básica y forma de ejecución del lenguaje Python.
- Conocer las primitivas básicas para la manipulación de los robots Múltiplo N6.

Actividad 1: conociendo a los robots

Objetivos de la actividad: que los alumnos puedan:

- Conocer las instrucciones básicas de los robots Múltiplo N6.
- Conocer el entorno de la Facultad de Informática.

Modalidad de trabajo: actividad en grupos reducidos.

Materiales: una computadora por grupo con acceso al *software* ProBots3D.

Desarrollo:

Comenzar la actividad explicando a los alumnos que van a trabajar con una herramienta que les permite ingresar y recorrer el edificio de la Facultad de Informática de la UNLP en forma virtual y que, en ciertos lugares de las instalaciones, se encuentran unos robots. Explicarles que los robots que ven en la aplicación son una réplica de los robots reales con los que se trabaja en el proyecto “Programando con robots y software libre” que seguramente podrán manipular en algunas de las clases siguientes⁴.

La consigna a realizar por los estudiantes es encontrar los robots y resolver los desafíos que se presentan. Indicarles que también es importante que presten atención al entorno que se presenta dado que el mismo es un modelo de las instalaciones de la facultad, lo cual puede resultarles interesante a aquellos que deseen concurrir luego a la facultad.

Dar unos minutos para recorrer el entorno y luego que hayan resuelto alguno de los desafíos hacer una puesta en común para destacar las siguientes ideas:

- En la pizarra de ProBots3D aparecen las instrucciones escritas en el lenguaje de programación Python, tal como deberían escribirlas si trabajarían con los robots reales y Python.

⁴Esta secuencia didáctica está pensada para trabajarse en los talleres realizados en la facultad, con los robots propios del proyecto y en aquellas escuelas que dispongan de los mismos.

- La primera instrucción que aparece permite incorporar al entorno de trabajo de Python las instrucciones necesarias para poder trabajar con los robots.

```
from duinobot import *
```

- Luego aparecen las instrucciones que permiten conectar la computadora con el robot. Explicarles que si bien estas instrucciones no son necesarias en ProBots3D, las mismas si deben utilizarse cuando trabajen con los robots reales.

```
modulo_comunicacion = Board("/dev/ttyUSB0")
robot1 = Robot(modulo_comunicacion, 1)
```

- En este código también aparecen variables. Preguntar: *¿se acuerdan para qué se usan las variables?* . Recordar el concepto con los alumnos, remarcando que *modulo_comunicacion* es la variable con la cual se va a referenciar al módulo de comunicaciones y *robot1* hace referencia al robot que vamos a programar.
- Al mover al robot aparecen las instrucciones equivalentes en el lenguaje Python. Las mismas son:

```
robot1.forward(30,1)
robot1.turnLeft(30,1)
robot1.backward(30,1)
robot1.turnRight(30,1)
```

Cierre

A modo de cierre realizar un repaso de las instrucciones mostradas por ProBots3D. Destacar que el robot Múltiple N6 se programa en Python y que este lenguaje tiene algunas reglas sintácticas estrictas que se verán más adelante, pero las instrucciones mostradas son las mismas que deberán utilizar luego con los robots reales.

Actividad 2: programando robots con Python

Objetivos de la actividad: que los estudiantes puedan:

- Conocer las características básicas de los robots Múltiple N6.
- Programar los robots a través de una consola de Python.

Modalidad de trabajo: actividad en grupos reducidos.

Materiales: un robot por grupo y una computadora.

Desarrollo:

Esta actividad se formula en dos momentos: un primer momento donde se presentan los dispositivos con los cuales se trabajará, es decir los robots Múltiplo N6 y los comandos necesarios para su manipulación. En un segundo momento se presenta en lenguaje Python como el lenguaje a utilizar para programar los robots.

Momento 1: introduciendo los robots y sus comandos

Al comenzar la actividad presentar a los alumnos un robot y preguntar si los reconocen. Se espera que puedan asociarlos con los robots vistos en ProBots3D. Luego, explicar las características principales:

- El Múltiplo N6 es un robot educativo donde tanto el *hardware* como el *software* utilizado son libres, es decir que tanto las especificaciones de la electrónica como la de los programas para su utilización están accesibles y pueden utilizarse libremente.
- Posee 3 ruedas que le permiten desplazarse: dos delanteras con tracción y una rueda trasera que le permite girar.
- Se mueve gracias a dos motores de corriente continua.
- Posee dos sensores de línea y un sensor ultrasónico que permite detectar obstáculos.
- El módulo XBee es el que permite la conexión con la computadora. Se conecta a través de un conector USB, y permite la comunicación con uno o más robots al mismo tiempo (utilizando una misma computadora).

Explicar a los alumnos que para poder trabajar con estos robots es necesario que en la computadora se instalen las librerías necesarias⁵.

Explicar que a partir de ahora se trabajará con el lenguaje de programación Python. En este momento se puede explicar que Python es un lenguaje interpretado y que para tener una primera aproximación se trabajará con una consola de comandos. La figura 6.3 muestra una imagen de una consola típica en un sistema Linux. Al abrir esta consola es posible comenzar a escribir las instrucciones que Python interpretará y ejecutará.

Antes de comenzar a programar a los robots, se trabajará con este nuevo entorno y la primera actividad a realizar es testear que la instalación fue correcta. Antes de comenzar recordar con los

⁵Tanto las librerías necesarias como la forma de instalación depende de los sistemas con que se cuente en cada caso.

```

claudia@claudia-Inspiron-13-7353: ~
claudia@claudia-Inspiron-13-7353:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>

```

Figura 6.3. Consola de Python.

estudiantes la actividad previa, y las instrucciones mostradas en la pizarra de ProBots3D. Indicarles que ahora deben ingresar esas mismas instrucciones en la consola de Python. Por lo tanto, deben tipear la siguiente sentencia:

```
»> from duinobot import *
```

Recordarles que con esta sentencia se importará o se agregarán al entorno de Python todas las funciones necesarias para poder trabajar con los robots. A continuación, indicarles que deben introducir las instrucciones necesarias para poder conectar con el robot. Para ello, pedirles que tipeen las siguientes instrucciones:

```
[1]»> modulo_comunicacion = Board("/dev/ttyUSB0")
```

```
[2]»> modulo_comunicacion.report()
```

```
[3][1, 2, 3]
```

Antes de comenzar a analizar estas instrucciones recordar el concepto de variable, destacando que *modulo_comunicacion* es la variable con la cual se va a referenciar al módulo de comunicaciones. Las dos instrucciones hasta ahora utilizadas, ya aparecieron previamente en ProBots3D y fueron explicadas, pero la línea [2] muestra una instrucción que hasta ahora no se vio. Explicarles que esta orden permite saber qué robots están disponibles. Cada robot posee un número identificador y, en el ejemplo, se muestran disponibles los robots con identificador 1, 2 y 3. Para manipular un robot, se lo debe referenciar también a través de una variable. Proponer que tipeen la siguiente sentencia:

```
»> robot1 = Robot(modulo_comunicacion, 1)
```

robot1 es el nombre de la variable con la que se referencia al robot cuyo identificador es 1 utilizando el módulo de comunicaciones referenciado con la variable *modulo_comunicacion*. A partir de este momento es posible comenzar a darle órdenes al robot.

Momento 2: manipulando los robots a través de la consola de Python

Para comenzar a programar los robots primero se deben conocer cuáles son las instrucciones que entienden los mismos. La tabla 6.1 muestra los comandos básicos con los que se manipula a robot y que aparecieron en la actividad previa con ProBots3D.

Instrucción	Descripción
<i>forward(velocidad, tiempo)</i>	Avanza a una velocidad y tiempo dado.
<i>backward(velocidad, tiempo)</i>	Retrocede a una velocidad y tiempo dado.
<i>turnLeft(velocidad, tiempo)</i>	Gira a la izquierda a una velocidad y tiempo dado.
<i>turnRight(velocidad, tiempo)</i>	Gira a la derecha a una velocidad y tiempo dado.

Tabla 6.1. Comandos básicos del robot Múltiplo N6

Proponer experimentar con estas instrucciones y comprobar los movimientos del robot. A modo de ejemplo se puede presentar:

```
[1] modulo_comunicacion = Board("/dev/ttyUSB0")
[2] modulo_comunicacion.report()
[3] robot1 = Robot(modulo_comunicacion, 1)
[4] robot1.forward(50,3)
[5] robot1.turnLeft(80,1)
```

Las líneas [1] a [3] son las mismas que ya trabajaron previamente. La línea [4] mueve el robot hacia adelante a una velocidad de 50⁶ durante 3 segundos. La línea [5] gira al robot hacia la izquierda a una velocidad de 80 durante 1 segundo.

⁶La velocidad máxima que el robot alcanza se representa con el valor 100, por lo tanto en este ejemplo el robot irá a una velocidad media. Es importante aclarar que la velocidad real depende también de la carga de las baterías.

Cierre

A modo de cierre realizar un repaso de lo visto en la actividad. Recalcar que ProBots3D les permitió conocer las instrucciones primitivas que deben utilizar luego en Python permitiendo un primer acercamiento a este lenguaje sin tener que escribir los textos exactos, dado que Python tiene algunas reglas sintácticas estrictas que se irán viendo a medida que avancen las actividades.

En particular, en esta actividad se vio la sentencia *import* que permite importar las instrucciones necesarias para manipular el robot y realizar la conexión con el mismo.

Luego también se vio como se definen variables en Python: las mismas se expresan como una secuencia de caracteres que deben comenzar con una letra y contener cualquier combinación de letras, números y el símbolo “_”. Una característica importante a tener en cuenta es que Python distingue entre letras mayúsculas y minúsculas. Por lo tanto si se define una variable con el nombre *robot1*, y luego la usamos como *ROBOT1*, Python pensará que se trata de dos variables distintas.

Actividad 3: programando en un editor

Objetivos de la actividad: que los estudiantes puedan:

- Conocer las características de un editor de programas sencillo.
- Reforzar las primitivas para manipular los robots a través de una consola de Python.
- Introducir el concepto de función en Python.

Modalidad de trabajo: actividad en grupos reducidos.

Materiales: un robot por grupo y una computadora.

Desarrollo:

En esta actividad se presenta un IDE (por sus siglas en inglés de *Integrated Development Editor*) y la primera noción de función en Python. No se detalla la descripción completa de esta actividad dado que no es de relevancia para esta tesis.

6.4. Resumen del capítulo

En este capítulo se presentó la propuesta de incorporación de ProBots3D en actividades áulicas.

Se reformularon las secuencias didácticas presentadas en el capítulo 3 replanteando sus objetivos y actividades al incorporar el uso de esta nueva herramienta.

Evaluación de ProBots3D



“Me lo contaron y lo olvidé; lo vi y lo entendí; lo hice y lo aprendí.”

– Confucio

En este capítulo se presenta la evaluación de la prueba de concepto de ProBots3D, realizada con alumnos de nivel primario y secundario. La misma se encuentra inserta en la aplicación de las secuencias didácticas definidas en el capítulo 6 que proponen la inclusión de ProBots3D como parte de las actividades.

Se describe la metodología utilizada, los perfiles de alumnos con los que se trabajó y los resultados obtenidos con cada grupo.

7.1. Metodología utilizada

La evaluación sobre el uso de ProBots3D fue realizada con dos grupos de alumnos correspondientes a nivel primario y secundario. En ambos casos, los alumnos ya habían trabajado los primeros conceptos

de programación. Es decir, que deberían identificar el concepto de programa de computadora como una secuencia de instrucciones que deben respetarse o ejecutarse en forma estricta para llevar a cabo una tarea.

Es importante destacar que la evaluación de ProBots3D se llevó a cabo en el contexto de dos secuencias didácticas distintas de acuerdo al grupo de alumnos con los cuales se trabajó, con diferentes aproximaciones a sus potenciales usos educativos.

En ambos grupos de alumnos, previo a la actividad, se relevó información de interés que permitió tener un panorama más preciso respecto a los conocimientos previos de cada alumno. En el caso de los alumnos de nivel secundario, este relevamiento se realizó a través de una encuesta que fue completada directamente por los mismos estudiantes, mientras que en el caso de los alumnos de nivel primario, el relevamiento se realizó a través de una breve entrevista realizada por la tesista a los alumnos.

Antes de comenzar la evaluación, en ambos grupos, se dio una breve explicación sobre el uso de ProBots3D y las consignas en cada caso. Durante la evaluación, la tesista realizó una observación directa de la actividad, registrando la interacción de los alumnos y auxiliando a los mismos ante las inquietudes planteadas.

Se llevó un registro de cada grupo en el cual se observaron los siguientes aspectos:

- **Actitud frente al *software*.** En particular, se registró si el alumno se mostró motivado o no en el uso de Probots3D.
- **Acceso a la consigna.** En este caso, se tomó nota si el alumno leyó las indicaciones y utilizó las ayudas o simplemente requirió de la ayuda del docente a cargo.
- **Cantidad de intentos necesarios para resolver la consigna planteada.**
- **Cantidad de ayuda requerida.** Se registró cuántas veces se convocó a la docente a cargo a lo largo de la experiencia.
- **Actitud frente a la resolución de la consigna.** En particular, se observó la reacción del alumno, tanto si resolvió la consigna en forma exitosa como si no lo hizo.

Al finalizar la actividad se interrogó a los alumnos participantes de manera tal de poder evaluar el impacto del uso de ProBots3D. Dado que el objetivo de la evaluación en los dos grupos destinatarios es distinto, a continuación se describe, en forma detallada, la metodología utilizada en cada caso.

7.2. Evaluación con alumnos de nivel primario

En la sección 6.2 se describió la organización de contenidos propuesta para enseñar programación en primaria y la secuencia didáctica en la cual se incluye el uso de ProBots3D. La prueba de concepto implementada abarca las funcionalidades básicas requeridas en dicha secuencia.

El objetivo planteado en esta evaluación es analizar si el uso de ProBots3D permite profundizar los conceptos básicos aprendidos por los alumnos hasta momento, poniendo especial énfasis en mostrar que existen muchas formas de escribir los programas de computadoras, es decir, muchos lenguajes en los que se puede comunicar a la computadora que se desea que haga.

Los destinatarios de esta evaluación fueron 20 niños y niñas, entre 9 y 10 años, que cursan el 4to. año de la Escuela 67 Enrique Galli, de la localidad de City Bell, provincia de Buenos Aires. La figura 7.1 muestra la ubicación geográfica de la misma y una imagen de puerta de acceso¹.



Figura 7.1. Escuela Primaria 67 Enrique Galli de la localidad de City Bell.

Esta escuela brinda una jornada extendida, por lo que los alumnos permanecen en la escuela durante ocho horas diarias. La gran mayoría de los niños y niñas provienen de familias humildes, quienes trabajan en la zona de quintas de la región y donde la única experiencia en el uso de computadoras la obtienen en la escuela.

La institución cuenta con un carrito de *netbooks* entregado por el programa Primaria Digital hace

¹Cabe aclarar que esta imagen fue obtenida desde el *software* Google Maps. En el momento de realizar la evaluación, la escuela se encontraba en obra, con lo cual la apariencia del acceso no es la misma.

tres años, con el cual trabajan en clase. Pero, dado que son las únicas computadoras disponibles, la experiencia se hizo con una *notebook* llevada por la tesista².

Los alumnos del curso donde se realizó la evaluación, trabajaron previamente las actividades propuestas en la secuencia didáctica descrita en la sección 3.2.2.1, por lo que la evaluación se enfocó en la actividad denominada “Programo mi robot”, siguiendo los lineamientos definidos en la misma y cuyos objetivos son:

Objetivos de la actividad: que los alumnos puedan:

- Reforzar el concepto de programa como una secuencia de instrucciones.
- Comprender el concepto de lenguaje de programación.

Para la evaluación con el *software*, los alumnos se agruparon en parejas, las cuales trabajaron en la *notebook* instalada con ProBots3D previamente. Las parejas se armaron de acuerdo al orden en el cual se encontraban ubicados en el aula.

Dado que se disponía sólo de una *notebook* con el *software* instalado se realizó una explicación general a toda la clase explicando las consignas y pautas generales y luego se trabajó individualmente con cada pareja de alumnos.

Como se mencionó anteriormente, antes de comenzar la actividad con cada pareja, se realizó una breve entrevista para recabar información sobre los conocimientos adquiridos hasta el momento. Las preguntas realizadas se encuentran disponibles en el Anexo B de este informe.

De esta entrevista se puede destacar que:

- El 70 % de los alumnos no tenían claro el concepto de programa de computadora. Conocían el concepto en forma intuitiva pero no lo asociaban al término “programa de computadora”.
- Sólo el 20 % de los alumnos reconocían que un programa está formado por instrucciones.
- El 100 % respondió que no se puede indicar de cualquier forma las instrucciones a la computadora, sino que deben hacerse de una forma especial, aunque ninguno de ellos conocía el término “lenguaje de programación”.
- El 100 % de los alumnos habían comenzado a utilizar Scratch la semana anterior a la evaluación.

Antes de comenzar la actividad con cada pareja, se les explicó la forma de interacción básica con el *software* y la consigna a realizar. Se les indicó que debían buscar los robots escondidos en el

²Cabe aclarar que ProBots3D no funciona adecuadamente en las *notebooks* de la escuela.

edificio y luego “programarlos” según la consigna. Dado que la gran mayoría no tenía claro el concepto de programa de computadora, en ese momento se volvió a trabajar dicho concepto, destacando el significado “programar” los robots mencionado en la consigna del *software*.

La figura 7.2 muestra algunas fotos de los niños y niñas trabajando con ProBots3D en el aula.



Figura 7.2. Imágenes de los alumnos de la primaria 67 interactuando con ProBots3D en el aula.

Como la experiencia se realizó en el aula de clase, se les dio un tiempo a cada pareja de manera tal que pudieran trabajar todos los alumnos. Cada pareja de alumnos trabajaron aproximadamente entre 10 minutos. La dispuso de un total de dos horas para toda la evaluación.

Durante la evaluación, la tesista realizó un registro de la actividad en una planilla pre-armada con los aspectos mencionados en la sección anterior, en la cual también se incluyeron observaciones destacadas en la experiencia que no fueron previamente pensadas. De esta observación se puede mencionar que:

- El 50 % de los niños y niñas no requirieron ayuda adicional mientras usaban ProBots3D.
- El promedio de intentos para resolver la consigna fue de 3 intentos.
- Dos de las parejas resolvieron la consigna programando al robot para ir en reversa, en vez de hacia adelante, como era lo esperado.
- Sólo una de las parejas se tomó un tiempo adicional para recorrer el edificio, el resto fueron directamente hacia donde estaban los dos robots más visibles y trabajaron con las consignas de los mismos.

- El 100 % de los alumnos se mostraron muy motivados ante el *software* solicitando repetir la experiencia más de una vez.

La figura 7.3 muestra los datos más destacados respecto del uso de ProBots3D y la consigna dada relevados en la observación de la experiencia.

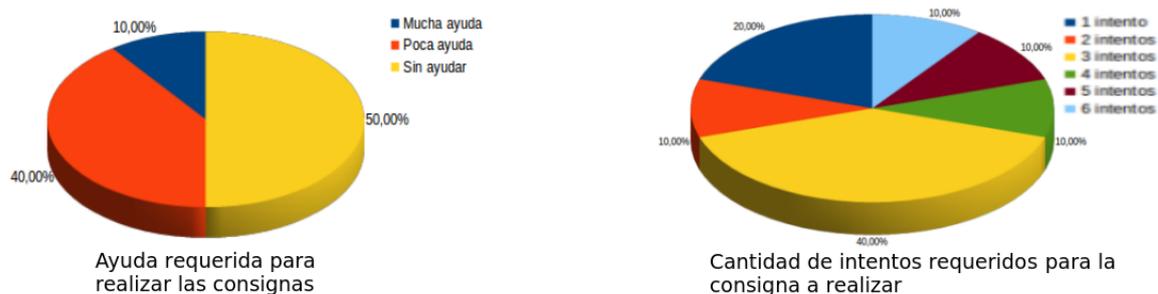


Figura 7.3. Gráficos sobre el uso de ProBots3D durante la evaluación de los alumnos de primaria.

Al finalizar la actividad, se interrogó verbalmente a los alumnos de manera tal de obtener su impresión sobre el *software* y evaluar el impacto de uso del mismo en función de los objetivos planteados para la actividad.

Las preguntas realizadas al finalizar esta actividad se encuentran disponibles en el Anexo B del presente informe.

Del análisis realizado de las entrevistas se destacan las siguientes observaciones:

- El 50 % de los niños y niñas identificaron correctamente el programa que resuelve la consigna. Del 50 % restante, sólo el 10 % no pudo asociar el concepto de programa con lo realizado en la consigna, el 40 % restante indicó correctamente las instrucciones utilizadas, pero no las asoció con el término “programa”.
- Respecto a las instrucciones relacionadas con los movimientos específicos del robot, el 100 % las identificó correctamente: el 80 % mencionó la instrucción textual mostrada en una sección de la interfaz de usuario de ProBots3D (ADELANTAR, DERECHA, etc.) y el 20 % restante señaló el ícono que la representaba en el panel de instrucciones.
- El 100 % de los alumnos respondió correctamente a la pregunta sobre cómo se ejecuta el programa, señalando el ícono provisto para tal fin.
- El 100 % de los alumnos solicitaron incluir más desafíos de programación en futuras versiones del *software*.

La figura 7.4 muestra los datos más destacados respecto del uso de ProBots3D y la consigna dada.

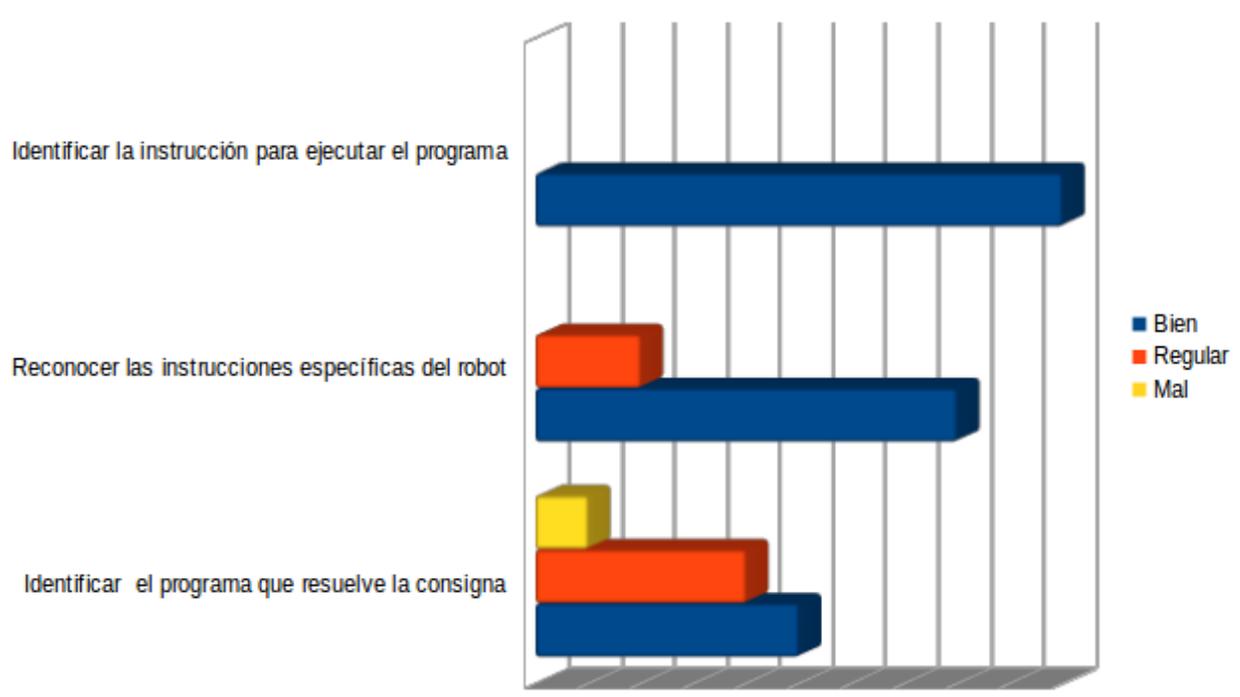


Figura 7.4. Respuestas más destacadas en la entrevista final.

7.2.1. Análisis cualitativo de la evaluación

Al finalizar la evaluación se realizó un análisis de toda la experiencia teniendo en cuenta los instrumentos mencionados anteriormente:

- El cumplimiento de la consigna incluida en ProBots3D.
- La observación directa realizada a los alumnos que llevaban adelante las actividades.
- La información recabada a través de las entrevistas.

Como se mencionó anteriormente, la experiencia se realizó en el aula de la escuela, estando presente, aparte de la tesista, la docente a cargo del curso, la profesora Giselle Molina. A medida que los alumnos se acercaban al escritorio donde se encontraba la *notebook* con ProBots3D, en parejas, para trabajar en la actividad “Programo Mi Robot”, el resto de la clase continuaban trabajando con la docente con la herramienta Scratch³.

Todos los alumnos pudieron trabajar con la consigna dada, resolviendo la misma sin mayores dificultades. Algunos de ellos pidieron repetir la experiencia, aunque en estos casos los datos de uso

³La escuela 67 participa en el proyecto de extensión “Las ciencias y los niños. Una experiencia de programación en la escuela primaria”. En este ciclo lectivo, los alumnos de 3ero. y 4to. año están siguiendo la propuesta de contenidos descrita en la sección 3.2 del capítulo 3.

no fueron relevados ni tenidos en cuenta en el análisis. Los datos mencionados en la sección anterior, corresponden a la primera interacción de los alumnos con el *software*.

En forma general, todos los alumnos se mostraron muy motivados con la herramienta, queriendo conocer los robots reales, los cuales serán llevados a la escuela en la jornada de cierre del proyecto de extensión “Las ciencias y los niños”, en el cual la escuela participa. También les interesó el hecho que el escenario virtual donde se desplazan en el *software* corresponde a un lugar real, preguntando sobre los lugares y objetos presentes en el mismo.

La observación de la experiencia mostró que los alumnos buscaron y encontraron los robots a programar en el entorno dado sin ninguna dificultad. Todos ellos requirieron de una prueba previa al desarrollo de la consigna para resolver la misma, experimentando con las instrucciones provistas para la manipulación del robot. Luego de esta breve interacción, se dispusieron a resolver la consigna dada sin mayores problemas.

Al finalizar la actividad se les volvió a interrogar respecto del significado de los términos *programa* e *instrucciones* mostrando haber reforzado dichos conceptos, que en la entrevista inicial se pudo ver no estaban tan afianzados. Respecto al concepto de *lenguaje de programación*, que era desconocido hasta el momento para los niños, se presentó una primera definición comparando el lenguaje utilizado en ProBots3D con el lenguaje de bloques utilizado hasta el momento.

Esto último permitió establecer el cumplimiento de los objetivos de la actividad. Cabe recordar que el mismo era analizar si el uso de ProBots3D permite reforzar los conceptos básicos aprendidos hasta momento e introducir el concepto de lenguaje de programación.

Respecto del rol docente frente a la actividad, la tesista acompañó a los alumnos durante la ejercitación, permitiendo que exploraran el *software* de acuerdo a las consignas presentes en el mismo. Se intervino solamente para reforzar los conceptos al finalizar la actividad. Se instó a que leyeran los textos con las consignas y pensarán las soluciones antes de plasmarlas en la herramienta. Esto último fue bastante difícil de lograr dado el impulso de los niños a trabajar en modo “prueba y error”.

Cabe destacar que la profesora Molina, docente a cargo de la clase, no participó de la evaluación, sino que ella continuó con el desarrollo de la clase con el resto de los alumnos. Al finalizar la actividad, se le presentó ProBots3D, quien quedó muy motivada también.

En líneas generales, la evaluación fue muy satisfactoria. Los niños y la docente a cargo se mostraron muy motivados y el uso de ProBots3D permitió reforzar los conceptos básicos vistos al momento. También despertó la inquietud en el 100 % de los niños de querer programar los robots reales y conocer el lugar que sirve de escenario al *software*. Aprovechando esta inquietud, se les dió información sobre la Facultad de Informática y la Universidad. Si bien son niños pequeños, esta tesista cree que es muy

importante inculcar desde edades tempranas la posibilidad de estudiar, y que en la región cuentan con muchas opciones para tal fin, siendo la Universidad de La Plata una de ellas.

7.3. La evaluación con alumnos de nivel secundario

En la sección 6.3 se describió las actividades enmarcadas en la secuencia didáctica que propone el uso de ProBots3D con alumnos de nivel secundario. Al igual que en el caso de la evaluación con alumnos de nivel primario, la prueba de concepto realizada incluye todas las funcionalidades requeridas en dicha secuencia.

Los destinatarios de esta evaluación fueron 19 alumnos que cursan el 2do. año de la escuela Estudiantes de La Plata, de la localidad de City Bell, provincia de Buenos Aires. La figura 7.5 muestra la ubicación geográfica de la misma y una imagen de puerta de acceso de la escuela.



Figura 7.5. Colegio Estudiantes de La Plata de City Bell.

La mayoría de los alumnos que concurren a esta escuela provienen de familias de clase media que cuentan con acceso a múltiples recursos informáticos fuera del colegio.

Antes de realizar esta evaluación, los alumnos trabajaron previamente las actividades propuestas en la sección 3.3.2, por lo que esta evaluación se enfocó en las actividades denominadas “Conociendo a los robots” y “Programando robots con Python”, siguiendo los lineamientos definidos en las mismas.

El objetivo de la evaluación, en este caso, es analizar si el uso de ProBots3D facilita a los alumnos la interacción posterior con los robots reales. Dado que en la pizarra mostrada en este *software* se visualizan las instrucciones que requieren los robots reales escritas en el lenguaje

Python, se espera que esto permita a los alumnos familiarizarse con dicho lenguaje y recordar las instrucciones básicas para manipular los robots reales.

La evaluación estuvo a cargo de la tesista, quien se desempeña como docente del curso en la escuela y las actividades se llevaron a cabo en una de las salas de computación de la misma.

Antes de comenzar la actividad se entregó a los alumnos una breve encuesta a través de la cual se pudo recabar información sobre sus conocimientos previos en relación a la temática abordada. En esta encuesta se incluyeron también un par de preguntas relacionadas al entorno donde se muestran los robots, es decir, el edificio de la Facultad de Informática, de manera de comprobar luego la veracidad del modelo.

De esta encuesta puede destacarse que:

- El 80% de los alumnos respondieron correctamente sobre la definición de programa.
- El 70% de los alumnos mostraron conocer el concepto de variable, aunque no todos luego lo asociaron al código mostrado en el lenguaje Python en ProBots3D⁴.
- Sólo el 10% de los alumnos reconocieron uno de los lenguajes de programación textuales presentados en la encuesta.

En la encuesta se los consultó respecto a las herramientas usadas hasta el momento. La figura 7.6 muestra las respuestas de los alumnos.

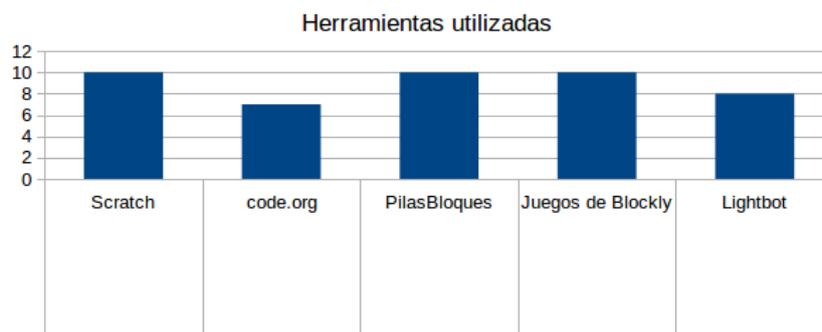


Figura 7.6. Relevamiento de herramienta conocidas al momento de realizar la evaluación.

Al igual que en el caso de los alumnos de la escuela primaria, se agruparon en parejas, las cuales trabajaron con una *notebook* instalada con el *software* ProBots3D previamente⁵.

⁴Cabe aclarar que hasta este momento los alumnos utilizaron lenguajes basados en bloques y la forma de utilizar y visualizar las variables es distinta.

⁵Cabe aclarar que no fue posible instalar el *software* en las computadoras de la escuela debido de no disponer de los permisos necesarios de administración sobre las computadoras.

La figura 7.7 muestra algunas fotos de los alumnos trabajando con ProBots3D en el aula.



Figura 7.7. Imágenes de los alumnos de la secundaria Estudiantes de La Plata interactuando con ProBots3D en la sala de computación del colegio.

Antes de comenzar la evaluación se explicó a cada pareja la forma de interacción básica con el *software* y la consigna a realizar.

Al igual que en el caso de los alumnos de nivel primario, durante la experiencia, la tesista realizó una observación directa de la actividad, registrando la interacción de los alumnos y auxiliando a los mismos ante las inquietudes planteadas.

De esta observación se destaca que:

- El 60 % de los alumnos no requirieron ayuda adicional mientras usaban ProBots3D.
- El promedio de intentos para resolver la consigna fue de 2 intentos.
- Tres parejas se tomaron un tiempo adicional para recorrer el edificio, el resto fueron directamente hacia donde estaban los dos robots más visibles y trabajaron con esas consignas. Una de ellas, conocía el edificio de la facultad debido a que su abuelo fue docente de la misma.
- Al igual que en el caso de los niños de la escuela primaria el 100 % de los alumnos se mostraron muy motivados ante el *software*.

La figura 7.8 muestra los datos más destacados respecto del uso de ProBots3D y la consigna dada.

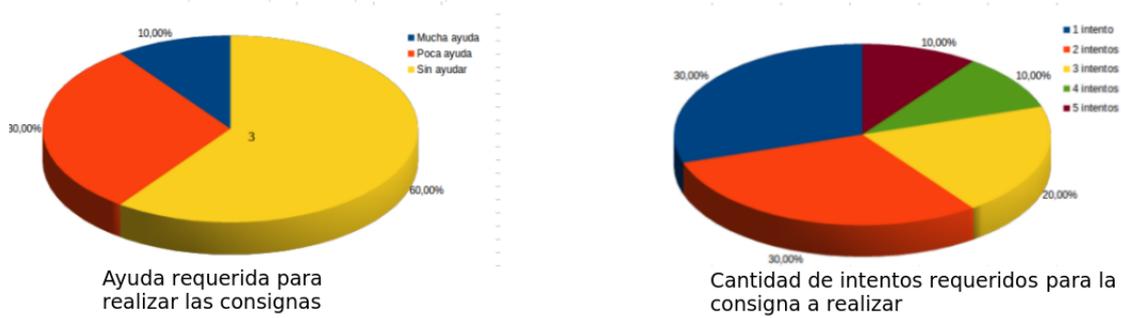


Figura 7.8. Datos destacados sobre el uso de ProBots3D.

Al finalizar la actividad “Conociendo a los robots”, se les entregó a los alumnos una segunda encuesta destinada a evaluar el *software* y analizar el cumplimiento de los objetivos de la actividad:

Objetivos de la actividad: que los alumnos puedan:

- Conocer las instrucciones básicas de los robots Múltiplo N6.
- Conocer el entorno de la Facultad de Informática.

De esta encuesta se destacan los siguientes datos:

- El 100 % de los alumnos identificaron correctamente las instrucciones para manipular el robot en ProBots3D.
- El 60 % de los alumnos reconocieron las variables presentes en el código.
- El 30 % reconocieron el edificio de la Facultad de Informática, indicando que ya había estado en el lugar previamente.
- El 70 % indicó que pensaba que programar los robots reales les sería más sencillo experimentando previamente con ProBots3D. Cabe aclarar que esta pregunta fue realizada previo al uso de los robots reales.

La figura 7.9 muestra algunos de estos datos.

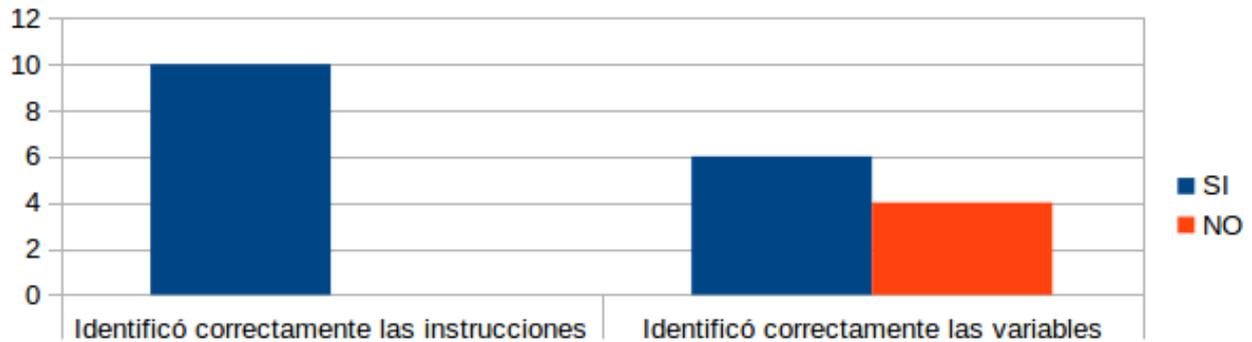


Figura 7.9. Datos destacados sobre el aporte de ProBots3D para la programación de los robots reales.

A continuación, se llevó a cabo la actividad “Programando robots con Python”, en donde los alumnos tuvieron el primer contacto con los robots reales y un lenguaje textual. Durante esta actividad, también se realizó una observación de la misma, llevando un registro de la experiencia.

En este caso se puede mencionar que:

- El 100 % de los alumnos se mostraron muy motivados para programar los robots, interrogando a la docente sobre sus características principales, funciones incluidas, forma de conexión, entre otras cosas.
- La gran mayoría tuvieron inconvenientes, en un primer momento, con el uso del lenguaje textual. Principalmente surgieron aspectos de la sintaxis propia de Python:
 - Mostraron dificultades con la característica *case sensitive* del lenguaje, es decir que trata a letras mayúsculas y minúsculas como distintos caracteres y por ende, indica un error al no introducir exactamente el texto que se espera. Por ejemplo, fue un error común ingresar “*turnleft*” y no “*turnLeft*”.
 - Errores de tipeo.
 - Disposición de algunos caracteres especiales en el teclado de la *notebook*.
- La gran mayoría de los alumnos se mostraron muy cuidadosos con el uso del robot. Al inicio de la actividad se explicó los cuidados necesarios para su correcto uso y en todo momento se tuvo especial recaudo con los mismos.

Luego de finalizar la actividad, siguiendo los lineamientos planteados en la misma, se les entregó a los alumnos una tercera encuesta para analizar el grado de cumplimiento de los objetivos planteados en la actividad y de la evaluación en general, es decir si ProBots3D facilitó o no la primera interacción con

los robots reales, que deben ser programados en el lenguaje de programación Python. Cabe recordar que los objetivos planteados para esta última actividad eran:

Objetivos de la actividad: que los estudiantes puedan:

- Conocer las características básicas de los robots Múltiplo N6.
- Programar los robots a través de una consola de Python.

La figura 7.10 muestra algunos de los datos más destacados relevados en la encuesta.

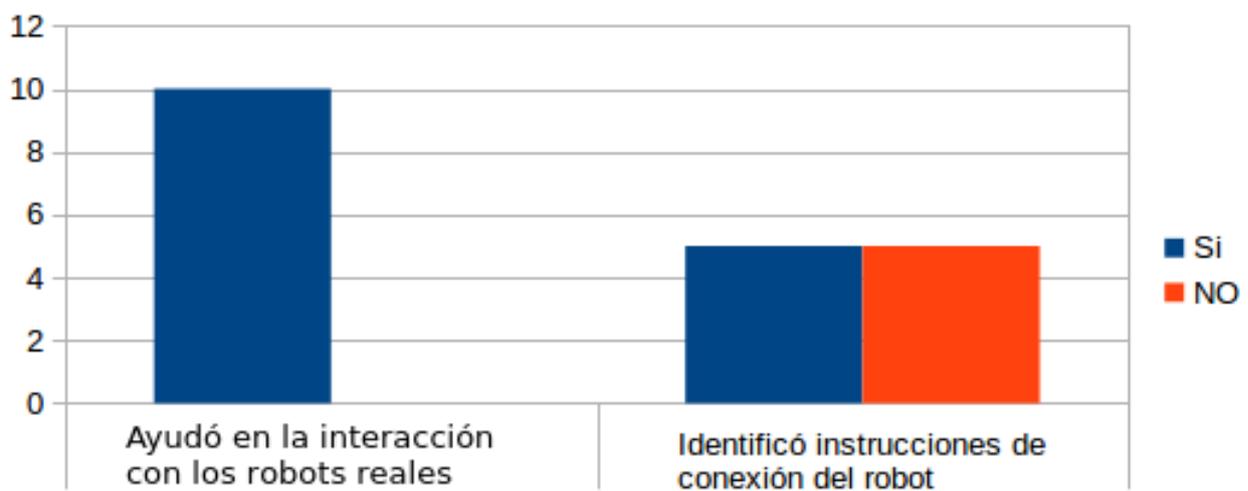


Figura 7.10. Datos destacados luego de programar los robots reales.

Como puede verse, el 100 % de los alumnos reconoció que haber utilizado ProBots3D previamente les ayudó a interactuar luego con los robots reales. Una de las alumnas mencionó que disponer de ProBots3D es muy útil en caso que la docente no esté disponible, dado que podría experimentar con las instrucciones y luego trasladar las mismas al entorno real.

Si bien el 100 % de los alumnos identificó correctamente las instrucciones para manipular los robots y las pudo trabajar en el lenguaje Python sin problemas, la mayoría mencionó que les incomodaba recordar y tipear las instrucciones y que produzca un error en caso de escribir las órdenes incorrectamente. La figura 7.11 muestra algunos de los aspectos destacados respecto al uso del lenguaje Python.

Algo de destacar es que ninguno de los alumnos mencionaron el hecho de que las instrucciones estuvieran en el idioma inglés.

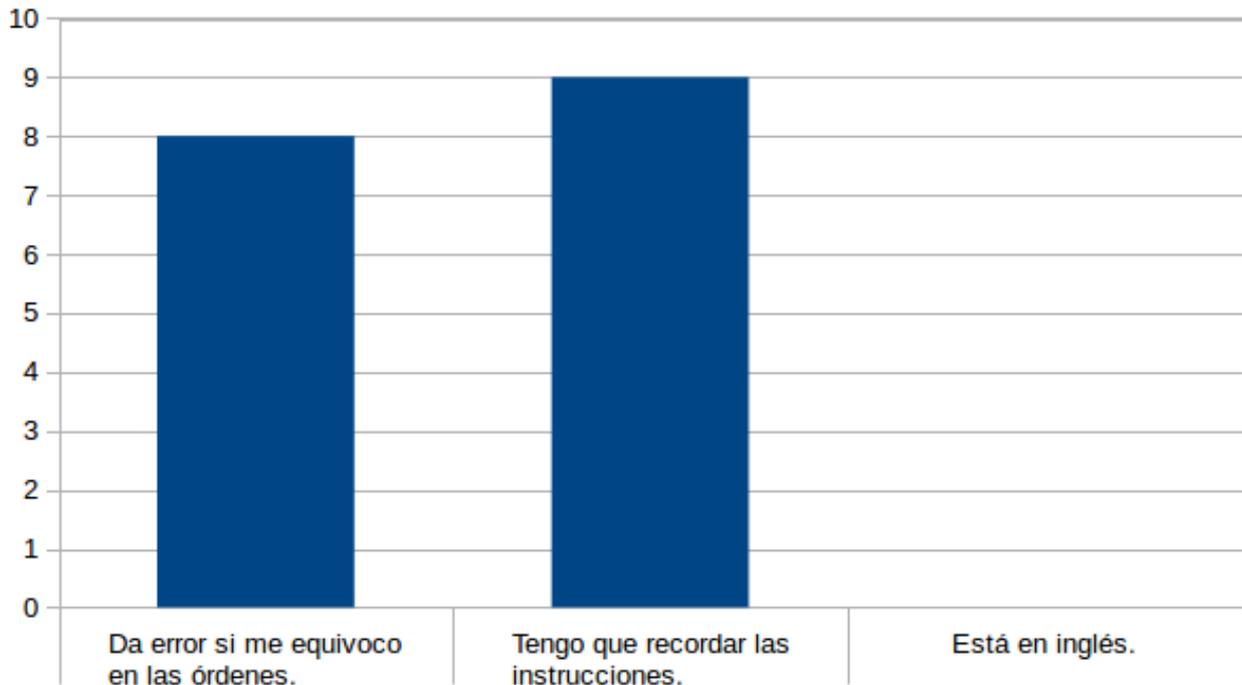


Figura 7.11. Datos destacados sobre el uso de Python.

Respecto a las instrucciones necesarias para conectar el robot con la computadora, sólo el 50% pudo identificar las mismas, aún cuando se encontraban presentes en ProBots3D en todo momento al interactuar con los robots en 3D.

Las tres encuestas realizadas se encuentran disponibles en el Anexo B de este informe.

7.3.1. Análisis cualitativo de la evaluación

Al finalizar la evaluación se realizó un análisis de toda la experiencia teniendo en cuenta los instrumentos mencionados anteriormente:

- El cumplimiento de la consigna incluida en ProBots3D.
- La observación directa realizada a los alumnos que llevaban adelante las actividades.
- La información recabada a través de las encuestas.

Como se mencionó anteriormente, la experiencia se realizó en una de las salas de computación de la escuela, y estuvo a cargo de la tesista quien se desempeña como docente del curso en la asignatura extra programática “Informática”.

Todos los alumnos pudieron trabajar con la consigna dada, resolviendo la misma sin mayores dificultades. Al igual que en el caso de los alumnos de nivel primario, varios de los alumnos pidieron

repetir la experiencia, y, como en el caso de los niños de primaria, los datos de uso de estos nuevos intentos no fueron relevados ni tenidos en cuenta en el análisis. Los únicos datos tenidos en cuenta corresponden a la primera interacción con el *software* y los robots.

En forma general, todos los alumnos se mostraron motivados con la herramienta, mencionando que les resultó de ayuda para luego manipular los robots reales.

La observación de la experiencia mostró que los alumnos buscaron y encontraron los robots a programar en el entorno dado, y, a diferencia de los niños de nivel primario, sólo una de las parejas de alumnos requirió ayuda adicional sobre el uso del *software* y el resto necesitó una asistencia mínima o nula para la realización de las consignas propuestas al momento de programar el robot.

En la experiencia con los robots reales, se notó alguna dificultad al momento de introducir las instrucciones que permiten comunicar al robot con la computadora. Como se mencionó anteriormente, aunque dichas instrucciones estaban en forma permanente visualizadas en la interfaz de ProBots3D, los alumnos no prestaron suficiente atención a las mismas, requiriendo luego volver a visualizarlas.

Respecto a las instrucciones para manipular el robot, ninguno de los estudiantes tuvo inconvenientes en recordarlas y utilizarlas. Sólo se presentaron cuestiones referidas a errores de tipeo, o de utilización de mayúsculas y minúsculas correctamente.

De esta manera, los objetivos de la actividad se lograron sin mayores inconvenientes.

Al finalizar la actividad los alumnos se mostraron interesados en conocer más funcionalidades de los robots. En este caso se les comentó el uso de los sensores y los mismos los asociaron con el uso de las instrucciones condicionales vistas en otras herramientas en clase.

Respecto del rol docente frente a la actividad, la tesista estableció las guías y las consignas correspondientes y se instó a los alumnos a explorar la herramienta de manera tal de poder resolver las mismas. Se intervino solamente para reforzar algunos conceptos.

En líneas generales, la evaluación fue muy satisfactoria. Los alumnos se mostraron muy motivados y el uso de ProBots3D permitió no sólo reforzar algunos conceptos básicos vistos hasta el momento sino que también les facilitó la interacción con los robots reales, donde deben utilizar un lenguaje de programación textual por primera vez.

Respecto al lugar que sirve de escenario al *software*, varios de los alumnos manifestaron conocerlo. Al indicar que se trata de la Facultad de Informática y que en el segundo piso se encuentra el laboratorio de investigación donde esta tesista desarrolla sus actividades, los mismos utilizaron la herramienta para “conocerlo”.

Al igual que en el caso de los niños y niñas de nivel primario, se les dió información sobre la Facultad de Informática y la Universidad, trabajando el hecho de inculcar desde edades tempranas la

posibilidad de estudiar, y las opciones disponibles en la ciudad de La Plata.

7.4. Resumen del capítulo

En este capítulo se describió la evaluación llevada a cabo con alumnos de nivel primario y secundario. En ambos casos, dicha evaluación fue muy positiva, logrando los objetivos planteados.

En el caso de los niños de nivel primario, ProBots3D se utilizó para reforzar los conceptos básicos vistos hasta el momento y permitió introducir el concepto de lenguaje de programación, mostrando otras alternativas a las conocidas por los alumnos.

En el caso de la evaluación en el nivel secundario, el uso de ProBots3D facilitó la introducción de los robots reales y el lenguaje de programación Python. Experimentar previamente con ProBots3D permitió a los alumnos conocer los aspectos básicos necesarios para interactuar con los mismos y presentarles un lenguaje textual, en este caso, Python.

En todos los casos se comprobó el poder motivador de los robots reales tanto en niños, jóvenes como en docentes. Tener la posibilidad de programar estos robots y poder visualizar cómo los mismos se mueven al ejecutar las instrucciones tiene un efecto muy particular generando iniciativas para continuar con actividades similares.

Conclusiones y trabajos futuros

8.1. Conclusiones

La sociedad del siglo XXI requiere de ciudadanos creativos, que puedan afrontar y gobernar la tecnología cada día más presente en la vida cotidiana.

Nuestros jóvenes requieren adquirir habilidades que no los limiten a ser simples usuarios de aplicaciones sino que puedan afrontar nuevos desafíos aplicando un pensamiento creativo, procedimental y que puedan adquirir pautas de razonamiento lógico.

Por este motivo, enseñar a programar en todos los niveles educativos es una necesidad que comienza a abordarse desde distintas iniciativas gubernamentales y privadas. Programas como “Escuelas del futuro” son una muestra de ello.

Esta tesis planteó como objetivo desarrollar y evaluar una propuesta de uso de la herramienta ProBots3D, en el marco de las actividades prácticas del proyecto “Programando con robots y software libre” .

Se realizó un análisis de las iniciativas existentes para la enseñanza de la programación, tanto a nivel nacional como internacional, y un relevamiento de las herramientas más utilizadas en las mismas.

ProBots3D surge como una herramienta de apoyo al proyecto antes mencionado. Dado que actualmente se trabaja tanto con alumnos de nivel secundario como primario, se desarrollaron dos versiones distintas de esta aplicación. En el caso de los alumnos de nivel primario, se utilizó ProBots3D como una forma de reforzar conceptos básicos, y en el caso de los alumnos secundarios, ProBots3D sirvió como “puente” para el trabajo con un lenguaje de programación textual.

Para ambos niveles educativos se diseñaron propuestas didácticas que incluyen el uso de ProBots3D y una organización de contenidos alrededor de las mismas.

La prueba de concepto se utilizó en una evaluación realizada en dos escuelas de la ciudad de City Bell, con dos perfiles de alumnos muy distintos y con un acceso a recursos fuera del ámbito de la escuela muy diferente. Aún con estas diferencias sustanciales, la temática a abordar implicó trabajar con distintos recursos, con sistemas operativos Microsoft Windows o GNU/Linux y distintas herramientas de soporte que, por lo general, no son utilizadas en otros contextos fuera de la escuela.

En el nivel secundario, se planteó la siguiente pregunta motivadora: ¿la utilización de ProBots3D ayudó a una mejor aproximación al uso de los robots reales? Luego de la evaluación realizada, se puede decir que esto fue así.

En el nivel primario, ProBots3D se utilizó como una herramienta más, que simplemente permitió mostrar y presentar un elemento motivador para continuar trabajando la temática.

En ambos casos, el uso de ProBots3D permitió presentar también la Facultad de Informática y la Universidad de La Plata como instituciones cercanas y a disposición de todos los que quieran acceder a ella para continuar sus estudios. Es muy importante, inculcar estas ideas desde edades tempranas de manera de naturalizarlas.

Como herramienta didáctica, ProBots3D demostró ser un recurso con amplias potencialidades educativas tanto en el nivel primario como secundario. En ambos casos, permitió profundizar los conceptos básicos de la programación como ser: la noción de programa, secuencia, instrucción, lenguaje de programación y ejecución de un programa.

En el caso de los estudiantes secundarios, además demostró ser un recurso valioso en la transición al uso de un lenguaje textual. A través del uso de ProBots3D los alumnos pudieron explorar las construcciones necesarias para abordar la programación de los robots en Python acercando la sintaxis de dicho lenguaje previo a su uso.

8.2. Trabajos futuros

Habiendo concluido esta primera instancia, se proponen los siguientes trabajos futuros relacionados:

- Implementar las dos versiones de ProBots3D de manera tal de incluir más desafíos en cada una y así poder encarar otros aspectos de la enseñanza de la programación, como por ejemplo las estructuras de control y abstracción procedural.
- Exportar los modelos y programas realizados a otro motor de juegos de manera tal de lograr un procesamiento más eficiente y de contar con otras plataformas de ejecución como web y móvil.
- Generar, a partir de ProBots3D, un juego de desafíos que permita ludificar (o “gamificar”) la aplicación, incorporando niveles de dificultad, puntos obtenidos, entre otras cosas al realizar las distintas consignas de programación.
- Extender FI3D de manera tal de incorporar los nuevos espacios y recursos presentes en la actualidad.
- Incorporar nuevos avatares que permitan que cada estudiante elija y personalice el mismo según sus gustos y preferencias.
- Evaluar la posibilidad de integración de ProBots3D al TIVU Virtual, espacio anticipatorio educativo basado en un juego serio, realizado por la CC Viviana Harari.
- Establecer un procedimiento de actualización del estado del arte tanto en lo referido a las herramientas existentes como en las propuestas de enseñanza de las ciencias de la computación en general como de la programación en particular. Esto permitirá adecuar las propuestas de organización de contenidos y las secuencias didácticas utilizadas.

Bibliografía y referencias



A.1. Bibliografía

Alfageme González B. (2003). Modelo colaborativo de enseñanza-aprendizaje en situaciones no presenciales: un caso de estudio. Universidad de Murcia. Tesis doctoral. Disponible en <http://www.tdx.cat/bitstream/handle/10803/10768/Alfageme2de3.pdf>. Último acceso: noviembre 2017.

Banchoff C., Harari V. & Martin S. (2015). Desarrollo de videojuegos educativos para los primeros años de escuela. Una forma de integrar docencia, extensión e investigación en la Facultad de Informática. Memorias del III Congreso Internacional de Videojuegos y Educación 2015. Disponible en: <https://drive.google.com/open?id=0Bx9hKg4aa0xAbkd5d2htTIhhbVFMRmpQLW40TIU1bG9IU0Vv>. Último acceso: noviembre 2017.

Banchoff Tzancoff C., Queiruga C., Martin S., Miyuki Kimura I., Brown Barnetche M. & Gómez S. (2015). Extensión en vínculo con escuelas. Memorias del XIII Congreso Latinoamericano de Extensión Universitaria. Cuba.

Bensabat, C. (2014). El Pensamiento Computacional en el Modelo 1 a 1. Memorias del Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación (Buenos Aires, Argentina, 12, 13 y 14 de noviembre de 2014) ISBN - 10 84-7666-210-6. ISBN - 13 978-84-7666-210-6. Sección: Los videojuegos en la educación. Disponible en <http://www.oei.es/historico/congreso2014/memoriactei/986.pdf>. Último acceso: noviembre 2017.

Berners-Lee Tim (2013). Saber programación es la nueva brecha digital, según Berners-Lee. PC World Professional España, Disponible en: <https://goo.gl/p8wZmm>. Último acceso: noviembre 2017.

Boychev P., (2014). LOGO TREE PROJECT. Disponible en <http://www.elica.net/download/papers/logotreeproject.pdf>. Último acceso: noviembre 2017.

Clements, Douglas & Swaminathan, Sudha. (1995). Technology and School Change New Lamps for Old?. Childhood Education. 71. . 10.1080/00094056.1995.10522619.

Díaz-Barriga A., (2013). Guía para la elaboración de una secuencia didáctica. UNAM Comunidad del conocimiento, pp. 2016-2031. Disponible en: <https://goo.gl/nGWgAR>

Díaz J., Harari I., Harari V. & Banchoff C. (2010). Impulsando Prácticas Sociales en Informática: Alfabetizando en TICs a Diferentes Sectores Sociales. Anales de “Universidad 2010” 7mo Congreso Internacional de Educación Superior - III Taller Internacional “La Virtualización en la Educación Superior”. La Habana, Cuba. ISBN: 978-959-16-1164-2.

Díaz J., Banchoff C, Martin S. & López F. (2012). Aprendiendo a programar con juegos y robots. Memorias de TE&ET 2012. ISBN: 978-987-28186-0-9. Disponible en: <http://teyet2012.et.unnoba.edu.ar/wp-content/uploads/2012/08/4698.pdf>. Último acceso: noviembre 2017.

Díaz J., Banchoff C., Martin S., Gómez del Mónaco A. & Lanfranco E. (2013). Robots a pilas. Presentado en el CISL 2013. Conferencia Internacional de Software Libre en Argentina. CABA. Argentina.

Díaz J., Banchoff Tzancoff C., Queiruga C. & Martin S. (2014). Experiencias de la Facultad de Informática en la Enseñanza de Programación en Escuelas con Software Libre. Memorias del Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación (Buenos Aires, Argentina, 12, 13 y 14 de noviembre de 2014). ISBN - 10 84-7666-210-6. ISBN - 13 978-84-7666-210-6. Sección: Los videojuegos en la educación. Disponible en <http://www.oei.es/historico/congreso2014/memoriactei/1426.pdf>. Último acceso: noviembre 2017.

Dirección General de Cultura y Educación de la provincia de Buenos Aires (2010). Diseño Curricular para la Educación Secundaria Nuevas Tecnologías de la Información y la Conectividad. Disponible en <https://goo.gl/3YIQZO>. Último acceso: noviembre 2017.

Fronteiras do Pensamento, (2014, 7 abr). Manuel Castells - A obsolescência da educação [Archivo de video]. Disponible en: <https://www.youtube.com/watch?v=eb0cNrE3I5g>. Último acceso: noviembre 2017.

Gander, W. (2015). Informatics–New Basic Subject. Bulletin of EATCS, 2(116).

Gibson J. Paul (2012). Teaching graph algorithms to children of all ages. Proceeding ITiCSE '12 Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education. ISBN: 978-1-4503-1246-2. 34-39.

GPL2 (1991). GNU General Public License, version 2. Disponible en: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html> Último acceso: noviembre 2017.

GPL3 (2007). GNU General Public License, version 3. Disponible en: <https://www.gnu.org/licenses/gpl-3.0.en.html>. Último acceso: noviembre 2017.

Harari I & Harari V. (2015). La Universidad como alternativa para los sectores marginados: un trabajo continuo. Memorias del XIII Congreso Latinoamericano de Extensión Universitaria. Cuba.

Harari V. & Banchoff C. (2016). FI3D: An Anticipatory Space. International Conference on Engineering Education & Research ICEER 2016. ISBN: 978-0-646-95724-1, 154-160.

Informatics Europe & ACM Europe (2013). Informatics education: Europe can-not afford to miss the boat. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education, April 2013. Disponible en <http://www.informatics-europe.org/images/documents/informatics-education-europe-report.pdf>. Último acceso: noviembre 2017.

Escuelas del Futuro, (2017). <http://educacion.gob.ar/escuelas-del-futuro>. Último acceso: noviembre 2017.

Lanfranco E., Bogado J., Vilaseca D. & da Silva Gillig J. (2012). Modificaciones realizadas al Robot Múltiplo N6 para permitir programación interactiva. Anales de las 41 JAIIO - ISSN 1850-2776

GNU (2016). Lista de licencias con comentarios. Disponible en <http://www.gnu.org/licenses/license-list.html>. Último acceso: noviembre 2017.

GNU GPL (2016). GNU GPL, (2016). GNU General Public License. Disponible en <http://www.gnu.org/licenses/gpl.html>. Último acceso: noviembre 2017.

Lisette Kalshoven, (2015). Copyright & education: innovation made difficult. Disponible en <https://goo.gl/U3GWRb>. Último acceso: noviembre 2017.

López F. (2015). XRemoteBot: un servicio para programar robots en forma remota. Disponible en <http://sedici.unlp.edu.ar/handle/10915/51032>. Último acceso: noviembre 2017.

(Ministerio de educación, 2017) . Plan Nacional Integral de Educación Digital. Disponible en: <http://planied.educ.ar/wp-content/uploads/2016/04/Res.1536-E2017.pdf>. Último acceso: noviembre 2017.

Ministro de Educación (2015) <http://www.me.gov.ar/consejo/resoluciones/res15/244-15.pdf>. Último acceso: noviembre 2017.

OPSSI (2015). Observatorio Permanente de la Industria del Software y Servicios Informáticos de la República, Iniciativa de CESSI (OPSSI). Reporte semestral sobre el Sector de Software y Servicios Informáticos de la República Argentina. Disponible en: <http://goo.gl/LCZvRi>. Último acceso: noviembre 2017.

Orr, D., M. Rimini & D. van Damme (2015). Open Educational Resources: A Catalyst for Innovation, OECD Publishing, Paris. <http://dx.doi.org/10.1787/9789264247543-en>. Último acceso: noviembre 2017.

Papert S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*, Basic Books Co., New York, 1980

Papert S. (1993). *Mindstorms: Children, Computers, And Powerful Ideas*. Basic Books; Edición: New Ed (14 de julio de 1993) ISBN: 978-0465046744.

Pentakalos O. (2008) *Proof-of-Concept Design*. Disponible en <https://goo.gl/LHnTdQ>. Último acceso: noviembre 2017.

Pozo J.I. (2008). *Aprendices y Maestros. La psicología cognitiva del aprendizaje*. Capítulo 3. Alianza Editorial. ISBN: 978-84-206-8349-2.

Queiruga C., Fava L., Banchoff Tzancoff C., Aybar Rosales V., Miyuki Kimura I. & Brown Bertneche M (2013). RITA: una herramienta didáctica-pedagógica innovadora en la escuela secundaria. Experiencias dentro del Proyecto “Articular universidad-escuela con JAVA para fortalecer la Educación-Técnica. Conectar Saberes”. Memorias del Simposio de Informática y Sociedad de CLEI2013. ISBN: 9781479913381

Queiruga C., Banchoff C., Martín S., Aybar Rosales V. & López F. (2016). *Programar en la escuela*. WICC 2016. Red de Universidades con Carreras en Informática (RedUNCI). ISBN: 978-950-698-377-2, 403-407. Disponible en <http://sedici.unlp.edu.ar/handle/10915/52766>. Último acceso: noviembre 2017.

Queiruga C., Banchoff Tzancoff C., Martín S., Aybar Rosales V., López F., Miyuki Kimura I. & Gómez S. (2017). *Programar en la Escuela: nuevos desafíos en las aulas*. WICC 2017. Red de Universidades con Carreras en Informática (RedUNCI) ISBN: 978-987-42-5143-5, 732-736. Disponible en <http://sedici.unlp.edu.ar/handle/10915/61343>. Último acceso: noviembre 2017.

Spinosa A., Ferreyra A., Demarco L., Erbetta C., Matínes A., Kosubsky L., Banchoff C. & Rentarí Y. (2012). Experiencias universitarias en el Liceo “V́ctor Mercante”: un espacio alternativo de formaci3n acad3mica. XI Jornadas de Enseñanza Inicial, Primaria y Media Preuniversitaria (JEMU). La Plata, Bs. As. Octubre 2012.

Tyre P. (2013). Educating Americans for the 21st Century. Is Coding the New Second Language? Disponible en: <https://www.smithsonianmag.com/innovation/is-coding-the-new-second-language-81708064/?c=y%3Fno-ist>. Último acceso: noviembre 2017.

Wing J. (2006). Computational thinking. Communications of ACM. Vol 49, N° 3.Mar. 2006, 33–35

Wartella, E., & Jennings, N. (2000). Children and Computers: New Technology. Old Concerns. The Future of Children, 10(2), 31-43. doi:10.2307/1602688

A.2. Referencias web

A.2.1. Herramientas utilizadas en la enseñanza de la programaci3n

Alice: <http://www.alice.org>

CodeCombat: <https://codecombat.com/>

CodinGame: <https://www.codingame.com>

Gameblox: <https://gameblox.org/>

Juegos de Blocly: <https://blockly-games.appspot.com>

Lighbot: <https://lightbot.com/>

Made with code: <https://www.madewithcode.com/>

MIT APP Inventor: <http://appinventor.mit.edu/>

Pilas Bloques: <http://pilasbloques.program.ar/>

pilas-engine: <http://pilas-engine.com.ar/>

PSeInt: <http://pseint.sourceforge.net/>

RITA: <https://github.com/vaybar/RITA>

Scratch for Arduino (S4A): http://s4a.cat/index_es.html

Scratch Jr.: <http://www.scratchjr.org/>

Scratch: <https://scratch.mit.edu/>

Snap!: <http://snap.berkeley.edu/>

A.2.2. Iniciativas y programas que promueven la enseñanza de las ciencias de la computación

Code.org: <https://code.org/>

Computing at School: <https://www.computingatschool.org.uk/>

Computing Science Unplugged: <http://csunplugged.org/>

Escuelas del Futuro: <http://educacion.gob.ar/escuelas-del-futuro>

La Fundación Logo: <http://el.media.mit.edu/logo-foundation/index.html>

La Hora del Código: <https://hourofcode.com/es>

Program.AR: <http://program.ar/>

A.2.3. Desarrollos en el marco del proyecto “Programando con robots y software libre”

API robots: <https://github.com/Robots-Linti/duinobot>

Dropsy: <https://github.com/dropsy-unlp>

DuinoBotSocks: <https://github.com/Robots-Linti/DuinoBotSocks>

XRemoteBot: <https://github.com/fernandolopez?tab=repositories>

A.3. Misceláneos

FI3D: <https://github.com/clauBanchoff/FI3D>

Proyecto Lihuen GNU/Linux: http://lihuen.linti.unlp.edu.ar/index.php/P%C3%A1gina_principal

Proyecto “Programando con robots y software libre” : <http://robots.linti.unlp.edu.ar/>

Instrumentos de evaluación



En el presente Anexo se incluyen los protocolos de los instrumentos utilizados en la evaluación de ProBots3D. Se incluyen las preguntas formuladas en las entrevistas a los alumnos de nivel primario y las encuestas realizadas con los estudiantes de nivel secundario durante el proceso de evaluación de la propuesta presentada en esta tesis.

B.1. Entrevista inicial realizada a alumnos de nivel primario

La entrevista que se realizó antes de comenzar la actividad con los alumnos de nivel primario participantes incluyó las siguientes preguntas:

Preguntas relacionadas con los conceptos teóricos a trabajar:

- ¿Sabes qué es un programa de computadoras?
- ¿Cómo está formado un programa?

- ¿Puedo indicar de cualquier forma las instrucciones a la computadora?
- ¿Qué es el lenguaje de programación?

Preguntas relacionadas con las herramientas conocidas:

- ¿Usaron Pilas Bloques?
- ¿Qué otra herramienta usaron?

B.2. Entrevista final realizada a alumnos de nivel primario

La entrevista que se realizó antes de finalizar la actividad con los alumnos de nivel primario participantes incluyó las siguientes preguntas:

Preguntas relacionadas con los conceptos teóricos a trabajar:

- ¿Cuál fue el programa que resolvió la consigna?
- ¿Con qué instrucción el robot se mueve hacia adelante?
- ¿Cómo se ejecuta el programa?
- ¿El lenguaje de programación que usamos es el mismo que el que usa Pilas Bloques o algunas de las herramientas que usaron antes?

Preguntas relacionadas con la experiencia con ProBots3D:

- ¿Les resultó fácil encontrar los robots?
- ¿Con cuál robot jugaron?
- ¿Reconocen el lugar donde se está trabajando?
- ¿Conocían las instalaciones de la facultad antes de comenzar a interactuar con ProBots3D?
- ¿Les gustaría jugar más desafíos?
- ¿Les gustaría programar los robots reales?

B.3. Encuesta inicial para alumnos de nivel secundario

La presente encuesta fue completada en formato papel por los alumnos participantes.

Usando ProBots3D

Encuesta inicial

1

Respondé las siguientes preguntas antes de trabajar con ProBots3D.

Algunos conceptos básicos

- ¿Qué es un programa de computadoras?
.....
- ¿Para qué usamos las variables?
.....
- ¿A que denominamos lenguaje de programación?
.....
- ¿Cuáles de los siguientes lenguajes de programación conocés?
..... Python C Pascal VisualBasic Ninguno

Herramientas de programación conocidas

- ¿Qué tipo de programas hiciste?
.... Animaciones Juegos sencillos Sólo resolví desafíos
- Marcá cuáles de las siguientes herramientas usaste:
.... PilasBloques Code.org Scratch
..... Alice Juegos de Blockly LightBot

Un poco más de información ...

- ¿Conocés la Facultad de Informática?
..... SI NO
- ¿Te interesa alguna de las carreras que se dictan en la facultad?
..... SI NO

Figura B.1. Encuesta inicial realizada a los alumnos secundarios.

B.4. Encuesta a alumnos de nivel secundario luego de utilizar ProBots3D

La presente encuesta fue completada en formato papel por los alumnos participantes.

2

Usando ProBots3D

Respondé luego de utilizar ProBots3D

Sobre el programa

1. ¿Pudiste encontrar fácilmente los robots?
..... SI NO
2. ¿Con cuál jugaste? (¿en qué lugar se encontraba?)
.....
3. ¿Cuántos robots encontraste?
.... 0 1 2 34 Más de 4
4. Marcá cuál de las siguientes instrucciones permite mover el robot hacia adelante:
..... mover forward avanzar adelantarmove-forward
5. Marcá cuál de las siguientes instrucciones permite girar el robot hacia la izquierda:
..... girarIzquierda turnLeft turnRight leftgirar
6. ¿Observaste en el programa mostrado en la pizarra algunas sentencias que no ingresaste?
.....SINO
7. ¿Qué son r y b en el programa mostrado?
.....
8. ¿Para qué sirven?
.....

El mundo de ProBots3D

1. ¿Reconocés el lugar que muestra ProBots3D?
..... SI NO
2. ¿Cuál es?
.....
3. ¿Conocés el lugar real?
..... SI NO
4. ¿Pensás que ahora que ya usaste ProBots3D, será más fácil programar los robots reales?
..... SI NO

Figura B.2. Encuesta realizada a los alumnos secundarios luego de utilizar ProBots3D.

B.5. Encuesta a alumnos de nivel secundario al finalizar la evaluación

La presente encuesta fue completada en formato papel por los alumnos participantes.

3

Programando los robots

Encuesta final

Respondé las siguientes preguntas luego de trabajar con los robots.

Sobre el uso de ProBots3D

1. ¿Te ayudó utilizar ProBots3D para programar a los robots?
..... SI NO
2. ¿Qué te costó más de trabajar con el lenguaje Python?
..... da error si me equivoco en las órdenes.
..... tengo que recordar las instrucciones.
..... está en inglés.
..... no me costó porque me acordaba de usar ProBots3D.
..... no me costó, aunque no me acordaba de los programas de ProBots3D.
3. ¿En este caso, hay instrucciones que no ingresaste?
..... SI NO
4. ¿Cómo referencías al robot para luego indicarle que se mueva?
.....
5. ¿Cuántas variables utilizaste?
.....

Sugerencias para mejorar

1. ¿Qué te gustaría agregar o cambiar a ProBots3D?
.....
.....
.....
.....

Figura B.3. Encuesta final realizada a los alumnos secundarios.