



Universidad de Murcia

Facultad de Informática

Grado en Ingeniería en Informática

Trabajo Fin de Grado

Integración de un modulo de simulación de robot en el proyecto Descubre

Autor

Alberto López Sánchez

Dirigido por

Juan Antonio Sánchez Laguna

Febrero 2016

*Dedicado a
alguien :-)*

Resumen

Extended Abstract

Índice general

Resumen	V
Extended Abstract	VII
Índice	IX
Lista de figuras	XI
Lista de tablas	XIII
1. Introducción	1
1.1. Situación actual	2
1.2. Usando la electrónica y la robótica para enseñar a programar	3
1.3. Proyecto Descubre la programación	3
1.4. Motivación y enfoque del proyecto	4
2. Estado del arte	9
2.1. El lenguaje Logo y el robot Turtle	9
2.2. Khan Academy	11
2.3. Code.org	11
2.4. Squeak Etoys	12
2.5. Blockly y Scratch	12
3. Análisis de objetivos y metodología	13
3.1. Objetivos	13
3.2. Metodología	13
4. Diseño y resolución del trabajo realizado	15
5. Conclusiones y vías futuras	17
Bibliografía	21

X

ÍNDICE GENERAL

Glosario

23

Acrónimos

25

Índice de figuras

1.1.	Mapa de visualización de eventos de llevados a cabo por el proyecto <i>Hora del código</i> alrededor del mundo. Actualmente 198,473 en todo el mundo, 1,839 en España. Obtenido de [6].	3
1.2.	Sección <i>Crea</i> del proyecto Descubre la programación. Obtenido de [14].	5
1.3.	Sección <i>Explora</i> del proyecto Descubre la programación. Obtenido de [14].	5
1.4.	Sección <i>Aprende</i> del proyecto Descubre la programación. Obtenido de [14].	5
1.5.	Sección <i>Retos</i> del proyecto Descubre la programación. Obtenido de [14].	6
1.6.	Vista del perfil de usuario del proyecto Descubre la programación. Obtenido de [14].	6
1.7.	Sección de estadísticas del usuario del proyecto Descubre la programación. Obtenido de [14].	7
1.8.	Salida del programa escrito en iJava ejecutado por el código 1.3. . . .	7
2.1.	Movimiento de Turtle en forma de cuadrado utilizando únicamente sus instrucciones básicas <code>forward</code> y <code>right</code> . Obtenido de [17]. . . .	10
2.2.	Cuadrado dibujado por el robot Turtle. Obtenido de [17].	11
2.3.	Flor dibujada por el robot Turle. Obtenido de [17].	11

Índice de cuadros

Capítulo 1

Introducción

Como dice M. Lemaire[27], en una sociedad que está incrementando su dependencia a las nuevas tecnologías¹, es imprescindible que las nuevas generaciones desarrollen la habilidad de pensar de manera crítica sobre tecnología.

El *pensamiento computacional*, como se refiere A. Bundy en [19], afecta a investigaciones de casi todas las disciplinas, tanto de ciencias como de humanidades. [...] La informática no solo ha permitido que los investigadores puedan hacer nuevas preguntas, sino también ha permitido aceptar nuevos tipos de respuesta. Por ejemplo, preguntas que requieren el procesamiento de una gran cantidad de datos.

Es necesario comprender la informática y desarrollar el *pensamiento computacional*. Se necesitan desarrollar nuevas tecnologías, nuevo Hardware y Software que automatice tareas largas, complejas y con una alta cantidad de cómputo. Tareas que no siempre los humanos podemos resolver de manera directa. Para conseguir esto, muchas veces es necesario programar.

Pero aprender a programar es mencionado como uno de los 7 grandes retos de la educación informática [30] y diversos estudios [32] muestran que las principales dificultades para un alumno cuando está en el proceso de aprender a programar son (a) como empezar un programa; (b) comprensión de la sintaxis específica del lenguaje de programación; (c) comprensión de la lógica² y (d) problemas a la hora de depurar el código escrito.

Aunque a primera vista aprender a programar pueda parecer una tarea ardua y compleja, tiene sus ventajas. En el estudio realizado por J. Siegmund y otros en [34], podemos ver como los participantes mientras comprendían, analizaban y buscaban errores en pequeños trozos de código, daban claras muestras de estar desarrollando actividad cerebral en regiones del cerebro relacionadas con el procesado del lenguaje,

¹Aquí, con nuevas tecnologías me refiero a productos y proyectos derivados de la Informática y las Tecnologías de la Información y la Comunicaciones (TIC) como puede ser los proyectos que detallamos en la sección 2.

²En este caso me refiero a lógica booleana, o también llamada Álgebra de Boole.

la atención y la memoria de trabajo.

De la misma manera, estudios realizados en niños [20] de entre 6 y 8 años, muestran que estos demostraron mayor capacidad de atención, más autonomía y un mayor placer por el descubrimiento de nuevos conceptos. En la misma línea, un estudio en niños de infantil [23] que utilizaban el Lenguaje Logo demostró que los mismos obtuvieron mejores resultados en pruebas de razonamiento, matemáticas o resolución de problemas. Otro estudio más reciente [28] demuestra que aprender a programar (independientemente del lenguaje) a una corta edad, potencia la creatividad y la habilidad de aprendizaje.

1.1. Situación actual

Actualmente existen muchos proyectos con el fin de introducir la programación como asignatura obligatoria en Educación Primaria y Secundaria. Estos proyectos vienen respaldados por importantes cambios en los planes de estudios de todo el mundo para enseñar programación[22, 29, 13, 26, 18], marcando una clara tendencia social de incluir la programación en los currículos académicos.

Uno de los proyectos más importantes y con más repercusión a nivel global es Code.org[9]. Propone una serie de herramientas y juegos diseñados especialmente para que los niños aprendan programación mientras juegan con sus personajes favoritos de películas y videojuegos. Todo ello acompañado de una infraestructura para que profesores puedan incluir estos proyectos en las aulas, y padres en sus casas. Su proyecto *Hora del código*[6] consigue reunir a niños de todas las edades una hora al día para que la inviertan en programar. Code.org cuenta con decenas de millones de estudiantes de más de 180 países, disponible en más de 30 idiomas. De manera gratuita, cualquiera puede aprender a programar en eventos que se realizan por todo el mundo (figura 1.1). Code.org está apoyado por grandes compañías y personalidades a nivel global como puede ser Microsoft, Google, el Presidente Barack Obama, Mark Zuckerberg, Bill Gates o Walt Disney Company. En la sección 2.3 se abordará más detalladamente las diferentes formas que tiene Code.org de enseñar a programar.

A nivel europeo, la Comisión Europea[21] ha promovido durante el año 2015 la *EU Code Week*[16] como parte de su Estrategia para la Educación y la Formación 2020. Este proyecto consistió en eventos de una semana de duración en la que se enseñaba informática y programación en lugares de toda Europa³.

³En España, se realizaron eventos dentro del marco del proyecto *EU Code Week* en Madrid, Sevilla, Murcia, Asturias, Canarias, Cantabria, Zamora, Cataluña, Ceuta, Badajoz, La Rioja, País Vasco y Valencia.

1.2. USANDO LA ELECTRÓNICA Y LA ROBÓTICA PARA ENSEÑAR A PROGRAMAR³



Figura 1.1: Mapa de visualización de eventos de llevados a cabo por el proyecto *Hora del código* alrededor del mundo. Actualmente 198,473 en todo el mundo, 1,839 en España. Obtenido de [6].

1.2. Usando la electrónica y la robótica para enseñar a programar

Hasta ahora se ha estado hablando de

1.3. Proyecto Descubre la programación

Descubre la programación[14] es un proyecto diseñado en la Facultad de Informática de la Universidad de Murcia y tiene como objetivo ayudar a los alumnos de Secundaria y Bachillerato a que desarrollen sus capacidades descubriendo lo que es la informática y aprendiendo a programar. Así como fomentar la inclusión del aprendizaje de la programación en secundaria y bachillerato.

Para ello, en un mismo sitio web, se integra (a) un conjunto de tutoriales de programación (figura 1.4); (b) una herramienta que permite programar (figura 1.2) y realizar ejercicios o retos propuestos (figura 1.5) y (c) una pequeña red social que permite publicar y compartir con el resto de compañeros los programas realizados (figura 1.3). Adicionalmente se puede consultar las estadísticas de aprendizaje (figura 1.6) y tiempo dedicado (figura 1.7) en la plataforma, tanto por el alumno como por el profesor. De esta manera se permite que los profesores puedan utilizar Descubre en las aulas y realizar un seguimiento de la dedicación y progreso del alumno.

En cuanto a la herramienta para desarrollar programas, el lenguaje utilizado es iJava[33] y ha sido desarrollado por J. A. Sánchez Laguna. iJava es un lenguaje imperativo basado en Java y comparte su sintaxis, aunque se han eliminado todos los componentes del lenguaje Orientado a Objetos. También incorpora un conjunto reducido de funciones de librería clasificadas en los tres grupos siguientes: numéricas,

entrad/salida y gráficas⁴.

A modo de ilustración, podemos ver el código 1.3, un programa que imprime por pantalla la cadena "Hello World.". En el código 1.3 podemos ver un programa un poco más complejo que dibuja círculos de colores en la pantalla según la posición del ratón. En la figura 1.8 se puede ver la salida de este último programa.

```
1 void main() {  
    print("Hello World.");  
3 }
```

Código 1.1: Programa básico en iJava imprimiendo por pantalla la cadena "Hello World.".

```
1 void main() {  
    //repetimos en bucle la funci n 'draw'  
3    animate(draw);  
    }  
5  
void draw() {  
7    //coloreamos el circulo seg n la posici n del raton  
    fill(mouseX, mouseY, 0); //valores RGB  
9    //dibujamos una elipse con radio 50  
    ellipse(mouseX, mouseY, 50,50);  
11 }
```

Código 1.2: Programa en iJava que dibuja un círculo de un color diferente según la posición en la pantalla en la que se encuentra el ratón.

La sintaxis de iJava se parece a muchos de los lenguajes modernos y más utilizados (al ser un subconjunto de la sintaxis de Java), lo cual simplifica el aprendizaje cuando se intenta aprender un nuevo lenguaje. También, gracias a la librería gráfica y matemática, los programas se simplifican mucho en cuanto a complejidad y longitud. De esta manera se consigue aligerar la carga de trabajo que tiene que realizar el alumno para conseguir hacer un programa vistoso, haciendo que la actividad de programar sea más atractiva.

1.4. Motivación y enfoque del proyecto

Las nuevas tecnologías están haciendo que cada vez más gente de todas las edades se interese por la informática, y más concretamente por la programación. Poco a

⁴Si el lector está interesado en aprender más sobre este lenguaje, puede consultar [33], o en [15].

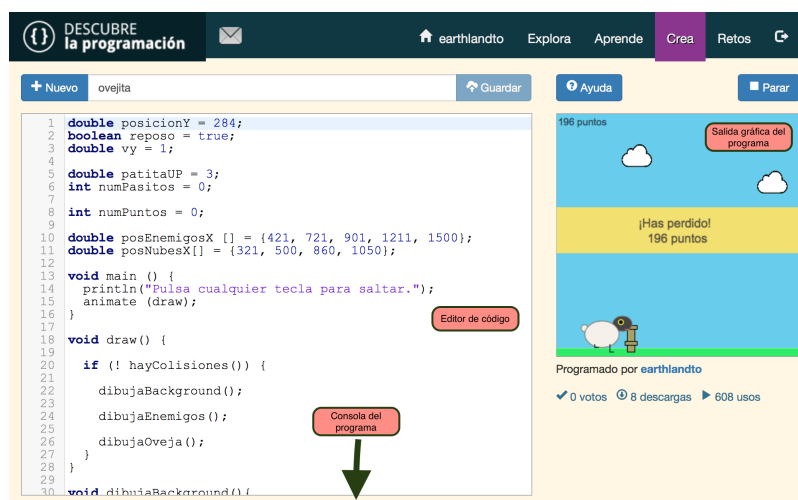


Figura 1.2: Sección *Crea* del proyecto Descubre la programación. Obtenido de [14].

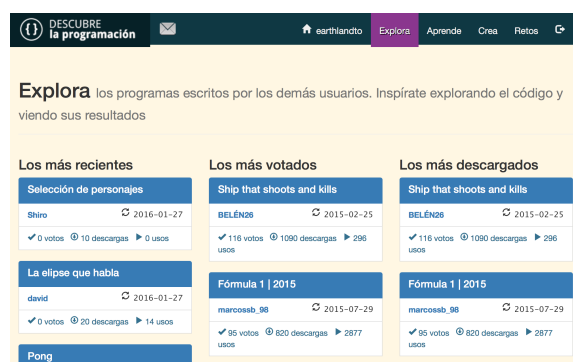


Figura 1.3: Sección *Explora* del proyecto Descubre la programación. Obtenido de [14].

poco la informática deja de ser cosa de un grupo selecto **de gente que entiende su funcionamiento**.

Como ya hemos comentado anteriormente, existe un movimiento que pretende introducir la informática y la programación en las aulas. Se han demostrado los beneficios de enseñar a programar en una edad escolar temprana.

Es en este momento cuando se hace imprescindible tomar decisiones acertadas. Aprender a programar en edades pre-universitaria debe ser algo accesible a cualquier estudiante, independiente de su condición o los antecedentes del mismo. Igualmente, los conceptos de programación deben ser presentados de manera, incremental, empezando por los más simples para luego ampliar a conceptos más complejos, como defiende L. Fernandez y otros en [24].

El proyecto se enfocará desarrollando un simulador de robot de dos ruedas en

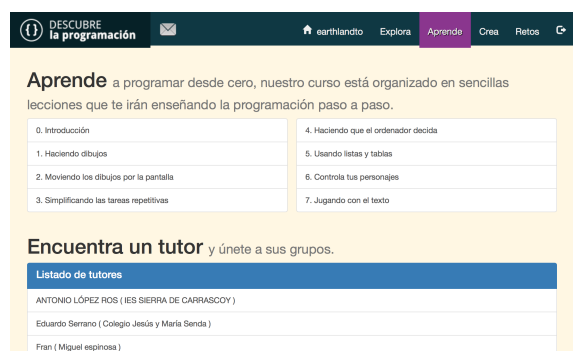


Figura 1.4: Sección *Aprende* del proyecto Descubre la programación. Obtenido de [14].



Figura 1.5: Sección *Retos* del proyecto Descubre la programación. Obtenido de [14].

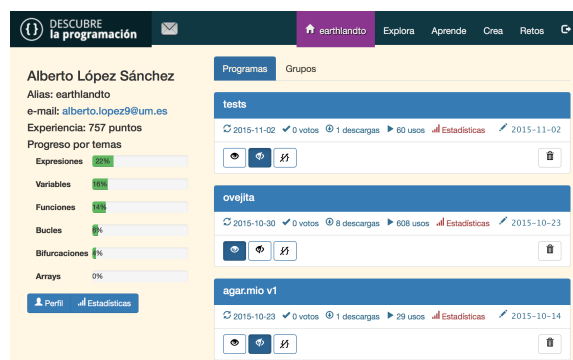


Figura 1.6: Vista del perfil de usuario del proyecto Descubre la programación. Obtenido de [14].

el que los alumnos puedan programar su comportamiento. Se ofrece una librería de funciones para simplificar la interacción con el mismo y conseguir cierta funcionalidad

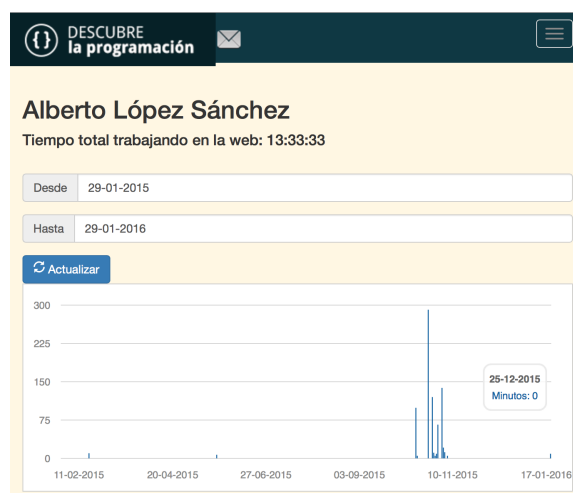


Figura 1.7: Sección de estadísticas del usuario del proyecto Descubre la programación. Obtenido de [14].

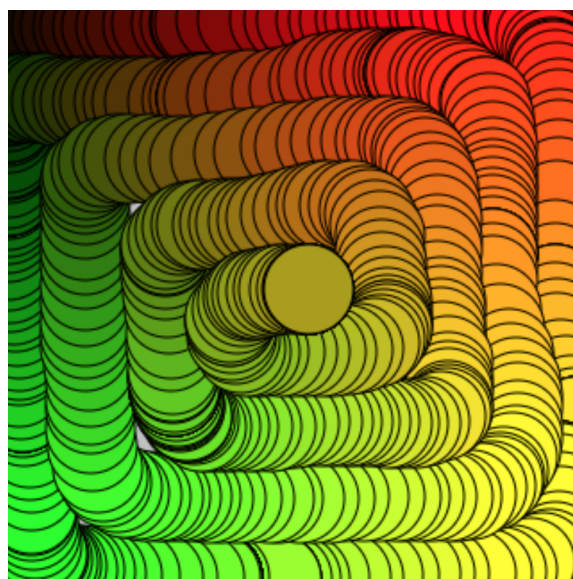


Figura 1.8: Salida del programa escrito en iJava ejecutado por el código 1.3.

extra que simplifique la tarea de comprensión y usabilidad del simulador.

El simulador se integrará en la plataforma Descubre la programación, mencionada en la sección 1.3. Esto conlleva que el estudiante programará el comportamiento del robot en iJava. El robot estará dentro de un circuito con distintos elementos con los que podrá interactuar. Asimismo, el robot dispondrá de una serie de sensores para poder recibir información del mundo que le rodea.

Al incluir el simulador como un módulo de descubre, se pretende reforzar el esfuerzo por parte de sus creadores de hacer llegar la programación al mayor número de estudiantes pre-universitarios posible, proponiendo una alternativa más vistosa, que añade una componente más de entretenimiento a la actividad de programar.

Igualmente, los estudiantes que ya están usando la plataforma Descubre, podrán trabajar los conceptos de programación (bucles, condiciones, variables y funciones, entre otras) en un entorno diferente, renovando así su interés en otra actividad diferente.

Esto permitirá ligar con la siguiente sección en la que se decide ampliar descubrir añadiendo el simulador del robot. La justificación puede ser que de este modo se permite a quien no tiene recursos materiales para comprar el hardware o es menos hábil con los aspectos mecánicos y electrónicos que también programe robots.

En las secciones siguientes se hablará de cuáles son las alternativas ya creadas que trabajan en esta línea. También se verá como se ha desarrollado la idea principal y que resultados se han obtenido. Por último, se analizarán los objetivos conseguidos y que ofrece mi propuesta de diferente con los proyectos ya desarrollados.

Capítulo 2

Estado del arte

A nivel global, actualmente existe una gran cantidad de proyectos con la única intención de enseñar, principalmente a alumnos de Secundaria y Bachillerato, diferentes aspectos de la informática como lo es la programación [8, 9, 3], la robótica [1, 2] e incluso electrónica (con Arduino[12]). Algunos de estos proyectos llevan décadas activos, como lo es el lenguaje Lenguaje Logo[11] y su proyecto Turtle[17]. La mayoría de estos proyectos promueven una enseñanza independiente y autodidacta bajo un entorno on-line y gratuito. De esta manera, el alumno puede aprender a su propio ritmo y desde cualquier parte del mundo.

2.1. El lenguaje Logo y el robot Turtle

El lenguaje Logo, basado en Lisp, fue diseñado como una herramienta para aprendizaje. Todas sus características -interactividad, modularidad, extensibilidad, flexibilidad en los tipos de datos- persiguen esta meta. El lenguaje Logo fue uno de los primeros proyectos en emprender la difícil tarea de enseñar a programar a niños de Primaria.

Como se explica en la página oficial del proyecto Logo[11], durante la década de los 70, en el Massachusetts Institute of Technology (MIT) y diferentes centros de investigación europeos, se llevó a cabo una investigación sobre el uso del Lenguaje Logo en un pequeño grupo de alumnos de Secundaria. [Todo el proceso se ha documentado en diferentes artículos como \[25\] o \[31\].](#)

El proyecto *Turtle* es el proyecto más popular del lenguaje Logo. Nació como una criatura robótica que se movía por el suelo y se podía programar solo con 2 instrucciones básicas: `forward x` y `right y`, para avanzar *x pasos de tortuga* o girar *y* grados hacia la derecha, respectivamente.

Combinando estas dos instrucciones de lo más simples, nuestro robot Turtle puede realizar cualquier movimiento más complejo, como podemos ver en la figura 2.1.

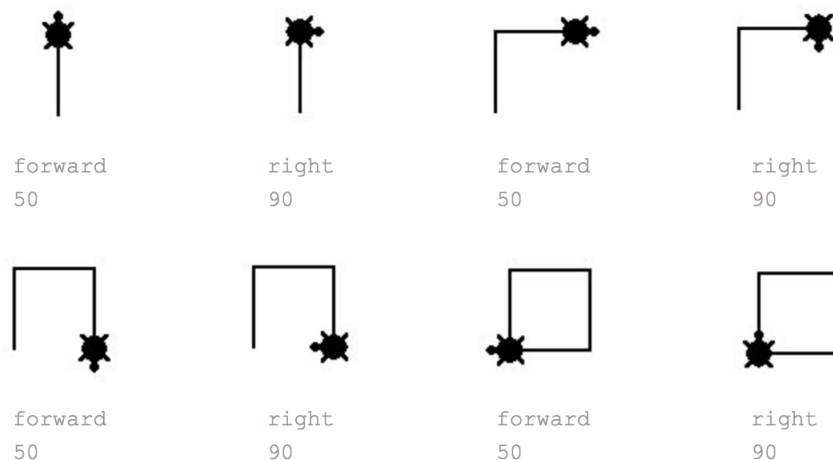


Figura 2.1: Movimiento de Turtle en forma de cuadrado utilizando únicamente sus instrucciones básicas `forward` y `right`. Obtenido de [17].

Para ilustrar brevemente el lenguaje Logo junto con el robot Turtle, vamos a mostrar como se dibujaría una figura que guarda cierto parecido a una flor. En el código 2.1, creamos una función `square` que dibuja un cuadrado de tamaño 50 (*pasos de tortuga*), como podemos ver en la figura 2.2

```
1 to square
  repeat 4 [forward 50 right 90]
3 end
```

Código 2.1: Definición de una función `square` en el Lenguaje Logo consiguiendo que el robot Turtle dibuje un cuadrado. Obtenido de [17].

Y a continuación definimos una función `flower` (código 2.1) que utiliza la función `square` y que dibujará una figura similar a una flor, como se aprecia en la figura 2.3.

```
1 to flower
  repeat 36 [right 10 square]
3 end
```

Código 2.2: Definición de una función `flower` en el Lenguaje Logo consiguiendo que el robot Turtle dibuje una figura con forma floral. Obtenido de [17].

El proyecto Turtle ha sido reproducido y modernizado en proyectos más recientes como Turtle Academy [5], el cual utiliza Scratch[7] para dibujar en la pantalla.

¿Comentarás algo de robomind o codehs que también son descendientes directos de lego? lo bueno es que el primero te hace enfocar la idea de la tortuga hacia un

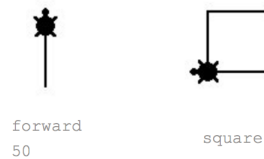


Figura 2.2: Cuadrado dibujado por el robot Turtle. Obtenido de [17].

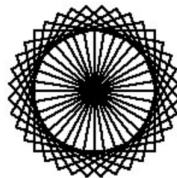


Figura 2.3: Flor dibujada por el robot Turtle. Obtenido de [17].

robot más realista, la segunda que lleva el aprendizaje a entorno web y tecnologías javascript. Todo esto te abre temas para seguir escribiendo el trabajo. Recuerda que sabes más de lo que crees y has hecho más de lo que piensas.

2.2. Khan Academy

Khan Academy[10], con casi 37 millones de usuarios, es uno de los mayores proyectos web para aprender casi de cualquier tema: Matemáticas, estadística, economía o humanidades.

2.3. Code.org

Code.org[9] es una plataforma on-line que se dedica exclusivamente a enseñar a programar y cuenta ya con más de 8 millones de usuarios. Cuenta con una gran comunidad de educadores y apoyos entre los que se puede contar al Presidente B. H. Obama. Lleva a cabo proyectos como *Hour of Code* en el que promueve que los usuarios inviertan una hora diaria programando en diferentes juegos, muchos de ellos utilizando [4].

Parece que estoy haciendo publicidad... shit

2.4. Squeak Etoys

2.5. Blockly y Scratch

Capítulo 3

Análisis de objetivos y metodología

3.1. Objetivos

Este Trabajo Fin de Grado consiste en desarrollar un módulo de simulación de un robot para integrarlo en la plataforma Descubre la programación. Para ello se tendrán que cubrir una serie de subobjetivos que nombraremos a continuación.

- Estudiar el uso y aplicación de diferentes librerías de físicas para generar el robot.
- Comprensión de la plataforma Descubre así como su posterior modificación.
- Creación de un simulador de un robot de dos ruedas y su integración en la plataforma Descubre.
- Modificación del motor de iJava y creación de la API para poder controlar el robot.

3.2. Metodología

Capítulo 4

Diseño y resolución del trabajo realizado

Capítulo 5

Conclusiones y vías futuras

explicar en que se diferencia robode con el resto y porque es mejor o que ventajas tiene explicar como resuelvo los problemas que se encuentra un estudiante cuando programa (intro-3º parrafo)

Bibliografía

- [1] Página oficial de robomind. <http://www.robomind.net>, Consultado el 10 de octubre de 2015.
- [2] Página oficial de moway education. <http://moway-robot.com>, Consultado el 10 octubre de 2015.
- [3] Página oficial de code academy. <https://www.codecademy.com>, Consultado el 12 de diciembre de 2015.
- [4] Página oficial de blockly games. <https://blockly-games.appspot.com>, Consultado el 12 de septiembre de 2015.
- [5] Página oficial del proyecto turtle academy. <https://turtleacademy.com>, Consultado el 25 de enero de 2016.
- [6] Página oficial del proyecto hora del código de code.org. <https://hourofcode.com/>, Consultado el 30 enero de 2016.
- [7] Página oficial del proyecto scratch. <https://scratch.mit.edu>, Consultado el 30 enero de 2016.
- [8] Página oficial de code school. <https://www.codeschool.com>, Consultado el 9 de enero de 2016.
- [9] Página oficial de code.org. <https://code.org>, Consultado el 9 de enero de 2016.
- [10] Página oficial de khan academy. <https://www.khanacademy.org>, Consultado el 9 de enero de 2016.
- [11] Página oficial de la fundación logo. <http://el.media.mit.edu/logo-foundation>, Consultado el 9 de enero de 2016.
- [12] Página oficial del proyecto arduino. <https://www.arduino.cc/>, Consultado el 9 de enero de 2016.

- [13] Página oficial del proyecto código 21 promovido por el departamento de educación de gobierno de navarra. <http://codigo21.educacion.navarra.es>, Consultado el 9 de enero de 2016.
- [14] Página oficial del proyecto descubre la programación. <http://descubre.inf.um.es>, Consultado el 9 de enero de 2016.
- [15] Página oficial del proyecto descubre la programación. guía del lenguaje. <http://descubre.inf.um.es/curso.php>, Consultado el 9 de enero de 2016.
- [16] Página oficial del proyecto eu code week. <http://codeweek.eu>, Consultado el 9 de enero de 2016.
- [17] A logo primer. http://el.media.mit.edu/logo-foundation/what_is_logo/logo_primer.html, Último acceso en enero 2016.
- [18] Noreen S. Ahmed-Ullah. Cps to add computer science as core subject. *Chicago Tribune*, Diciembre 2013.
- [19] Alan Bundy. Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2):67–69, 2007.
- [20] Douglas H Clements. Effects of logo and cai environments on cognition and creativity. *Journal of Educational Psychology*, 78(4):309, 1986.
- [21] Social Committee Commission to the European Parliament, the European Economic and the Committee of the Regions. Draft 2015 joint report of the council and the commission on the implementation of the strategic framework for european cooperation in education and training (et 2020). new priorities for european cooperation in education and training. http://ec.europa.eu/education/documents/et-2020-swd-161-2015_en.pdf, Agosto 2015.
- [22] Sean Coughlan. Computer science part of english baccalaureate. *BBC News*, Febrero 2013.
- [23] Julie Sarama Douglas H. Clements, Michael T. Battista. Logo and geometry. *Journal for Research in Mathematics Education. Monograph*, 10:i–177, 2001.
- [24] L Fernández Muñoz, R Peña, F Nava, and A Velázquez Iturbide. Análisis de las propuestas de la enseñanza de la programación orientada a objetos en los primeros cursos. *Actas de las VIII Jornadas de Enseñanza Universitaria de la Informática (JENUI'02)*, pages 433–440, 2002.

- [25] Wallace Feurzeig et al. Programming-languages as a conceptual framework for teaching mathematics. final report on the first fifteen months of the logo project. 1969.
- [26] Department for Education UK Government. Statutory guidance. national curriculum in england: computing programmes of study. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>, Septiembre 2013.
- [27] Maude Lemaire. Incorporating computer science into an elementary school curriculum. 2014.
- [28] Yuen-Kuang Cliff Liao and George W Bright. Effects of computer programming on cognitive outcomes: A meta-analysis. *Journal of Educational Computing Research*, 7(3):251–268, 1991.
- [29] José Marcos. Los colegios de madrid impartirán clases de programación. *El País*, Septiembre 2014.
- [30] Andrew Mcgettrick, Roger Boyle, Roland Ibbett, John Lloyd, Gillian Lovegrove, and Keith Mander. Grand challenges in computing: Education. a summary. *The Computer Journal*, 48(1):42–48, 2005.
- [31] Roy D Pea and D Midian Kurland. Logo programming and the development of planning skills. technical report no. 16. 1984.
- [32] V Renumol, S Jayaprakash, and D Janakiram. Classification of cognitive difficulties of students to learn computer programming. *Indian Institute of Technology, India*, 2009.
- [33] Juan Antonio Sánchez Laguna. ijava: un nuevo lenguaje para facilitar el paso del paradigma imperativo al orientado a objetos. *Jornadas de Enseñanza Universitaria de la Informática (15es: 2009: Barcelona)*, 2009.
- [34] Janet Siegmund, Christian Kästner, Sven Apel, Chris Parnin, Anja Bethmann, Thomas Leich, Gunter Saake, and André Brechmann. Understanding understanding source code with functional magnetic resonance imaging. In *Proceedings of the 36th International Conference on Software Engineering*, pages 378–389. ACM, 2014.

Glosario de términos

Arduino . 9

Comisión Europea Órgano ejecutivo y legislativo de la Unión Europea. Se encarga de proponer la legislación, la aplicación de las decisiones, la defensa de los tratados de la Unión y del día a día de la Union Europe. 2

Descubre la programación . 3, 5–7, 13

Hardware . 1

iJava . 3, 13

Java . 3

Lenguaje Logo El lenguaje Logo, basado en el lenguaje Lisp, fue diseñado como una herramienta para aprendizaje. Todas sus características -interactividad, modularidad, extensibilidad, flexibilidad en los tipos de datos- persiguen esta meta. 2, 9

Lisp . 9

Software . 1

Turtle El proyecto más popular del Lenguaje Logo ha evolucionado en la Tortuga, originalmente una criatura robótica que se movía por el suelo siguiendo una serie de instrucciones programadas previamente. 9

Acrónimos

API Application Programming Interface. 13

MIT Massachusetts Institute of Technology. 9

TIC Tecnologías de la Información y la Comunicaciones. 1