



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 :-)*

Agradecimientos

Resumen

Abstract

Índice general

Agradecimientos	V
Resumen	VII
Abstract	IX
Índice	XII
Lista de figuras	XIII
Lista de tablas	XV
1. Introducción	1
1.1. Situación actual	2
1.2. Proyecto Descubre la programación	3
1.3. Motivación y enfoque del proyecto	6
2. Estado del arte	9
2.1. El lenguaje Logo y el robot Turtle	9
2.2. Khan Academy	10
2.3. Code.org	11
2.4. Squeak Etoys	11
2.5. Blockly y Scratch	11
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
A. Lenguaje iJava	19

Bibliografía	23
Glosario	25
Acrónimos	27

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 [12].	3
1.2.	Sección <i>Crea</i> del proyecto Descubre la programación. Obtenido de [11].	4
1.3.	Sección <i>Explora</i> del proyecto Descubre la programación. Obtenido de [11].	5
1.4.	Sección <i>Aprende</i> del proyecto Descubre la programación. Obtenido de [11].	5
1.5.	Sección <i>Retos</i> del proyecto Descubre la programación. Obtenido de [11].	5
1.6.	Vista del perfil de usuario del proyecto Descubre la programación. Obtenido de [11].	6
1.7.	Sección de estadísticas del usuario del proyecto Descubre la programación. Obtenido de [11].	6
1.8.	Salida del programa escrito en iJava ejecutado por el código 1.2. . . .	7
2.1.	Movimiento de Turtle en forma de cuadrado utilizando unicamente sus instrucciones básicas <code>forward</code> y <code>right</code>	10
2.2.	Cuadrado dibujado por el robot Turtle.	11
2.3.	Flor dibujada por el robot Turtle.	11

Índice de cuadros

Capítulo 1

Introducción

Como dice Maude Lemaire[24], 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 [17], afecta a investigaciones de casi todas las disciplinas, tanto de ciencias como de humanidades. [...] La informática no solo ha permitido que los investigadores puedan hacerse 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.

Aprender a programar es mencionado como uno de los 7 grandes retos de la educación informática [27] y diversos estudios [29] 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 Janet Siegmund y otros en [31], 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 procesamiento del

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.

lenguaje, la atención y la memoria de trabajo.

De la misma manera, estudios realizados en niños [18] 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 [20] 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 [25] demuestra que aprender a programar (independientemente el lenguaje) potencia la creatividad y la habilidad de aprendizaje en personas de corta edad.

1.1. Situación actual

Actualmente existen muchos proyectos para que se introduzca la programación en las aulas y se aprenda a programar desde edades tempranas, ya sea en el aula con un profesor, en casa con los padres o de manera independiente. Estos proyectos vienen respaldados por importantes cambios en los planes de estudios de todo el mundo para enseñar programación[19, 26, 10, 23, 16], 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[5]. Su proyecto *Hora del código*[12] reúne durante una hora al día a 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). El proyecto tiene como público objetivo principal a niños de primaria y secundaria. 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[32] ha promovido durante el año 2015 la *EU Code Week*[1] como parte de su Estrategia para la Educación y la Formación 2020. Este proyecto consistió en eventos de una semana de duración en las 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.

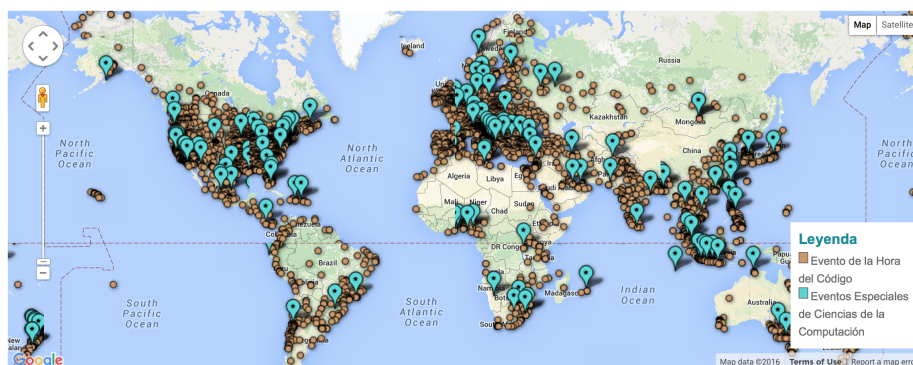


Figura 1.1: Mapa de visualización de eventos de llevados a cabo por el proyecto *Hora del código* alrededor del mundo. Actualmente 198,473 en todo el mundo, 1,839 en España. Obtenido de [12].

1.2. Proyecto Descubre la programación

Descubre la programación[11] es un proyecto desarrollado por 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 (figura 1.3) con el resto de compañeros los programas realizados. Adicionalmente se consultan las estadísticas de aprendizaje (figura 1.6) y tiempo dedicado (figura 1.7) en la plataforma.

En cuanto a la herramienta para desarrollar programas, el lenguaje utilizado es IJava[30] y ha sido desarrollado por J. A. Sánchez Laguna. iJava es un lenguaje imperativo basado en Java y comparte su sintaxis. También incorpora un conjunto reducido de funciones de librería clasificadas en los tres grupos siguientes: numéricas, entrada/salida y gráficas. iJava se estudiará más en detalle en el apéndice A.

En el código 1.2 podemos ver un programa básico escrito en iJava que muestra por pantalla la cadena "Hello World.". En el código 1.2 podemos ver un código 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 del programa.

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

```
3 }
```

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

```
1 void main() {
    noStroke();
3   background(0);
    animate(draw);
5 }

7 void draw() {
    background(0,0,0,0.01);
9   fill(mouseX, mouseY, 0);
    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, lo cual simplifica el aprendizaje de estos nuevos lenguajes. También, gracias a la librería gráfica y matemática, los programas se simplifican mucho, en cuanto a complejidad como en longitud. De esta manera se consigue aligerar la carga de trabajo que tiene que realizar el alumno para conseguir hacer un programa vistoso, enganchándolo más fácilmente a la actividad de programar.

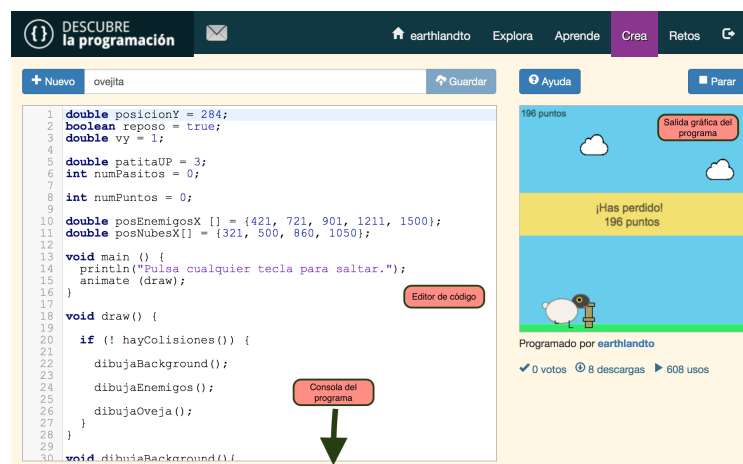


Figura 1.2: Sección *Crea* del proyecto Descubre la programación. Obtenido de [11].

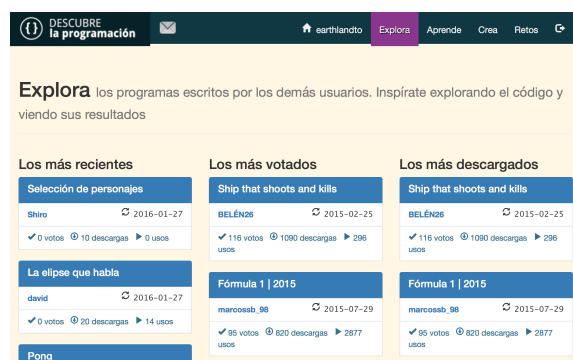


Figura 1.3: Sección *Explora* del proyecto Descubre la programación. Obtenido de [11].

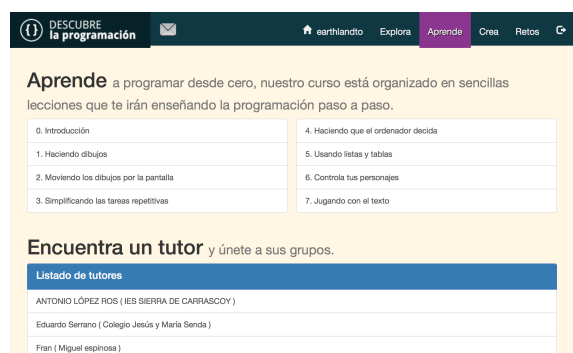


Figura 1.4: Sección *Aprende* del proyecto Descubre la programación. Obtenido de [11].



Figura 1.5: Sección *Retos* del proyecto Descubre la programación. Obtenido de [11].

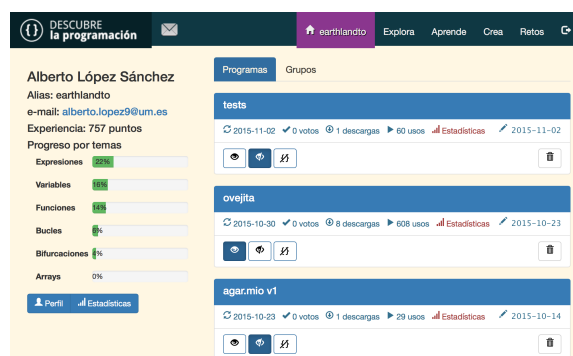


Figura 1.6: Vista del perfil de usuario del proyecto Descubre la programación. Obtenido de [11].



Figura 1.7: Sección de estadísticas del usuario del proyecto Descubre la programación. Obtenido de [11].

1.3. 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 poco la informática deja de ser cosa de un grupo selecto de gente colorque sabe que una aplicación no es *magia*.

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



Figura 1.8: Salida del programa escrito en iJava ejecutado por el código 1.2.

estudiante, independiente de su condición o los antecedentes del mismo. Igualmente, los conceptos de programación deben ser presentados de manera *escalonada y paulatina* [21].

El proyecto se enfocará desarrollando un simulador de robot de dos ruedas en 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 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 1.2. 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 descubrir, 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.

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.

¿Se puede decir que me he explicado el "enfoque del proyecto"?

Capítulo 2

Estado del arte

A nivel global, actualmente existe una gran cantidad de proyectos con la única intención de enseñar, principalmente a algunos de secundaria y bachillerato, diferentes aspectos de la informática como lo es la programación [4, 5, 3], la robótica [14, 8] e incluso chips y electrónica con Arduino[9]. Algunos de estos proyectos llevan décadas activos, como lo es el lenguaje Lenguaje Logo[7] y su proyecto Turtle[2] La mayoría de estos proyectos promueven una enseñanza independiente y autodidacta bajo un entorno on-line. 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[7], 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 [22] o [28].

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 tortuga puede realizar cualquier movimiento más complejo, como podemos ver en la figura

2.1.

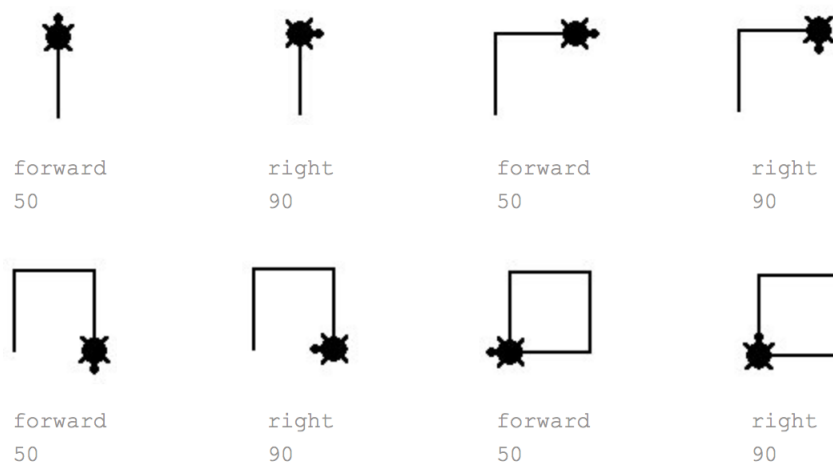


Figura 2.1: Movimiento de Turtle en forma de cuadrado utilizando unicamente sus instrucciones básicas `forward` y `right`.

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 que se muestra a continuación, creamos una función `square` que dibuja un cuadrado de tamaño 50 pasos, como podemos ver en la figura 2.2

```

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

```

Y a continuación definimos una función `flower` que utiliza la función `square` antes descrita y que dibujará una flor, como se puede apreciar en la figura 2.3

```

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

```

El proyecto Turtle ha sido reproducido en proyectos más recientes como Scratch[13], el cual se verá en más profundidad en la sección 2.5.

2.2. Khan Academy

Khan Academy[6], con casi 37 millones de usuarios, es uno de los mayores proyectos web para aprender casi de cualquier tema: Matemáticas, estadística, economía o humanidades.

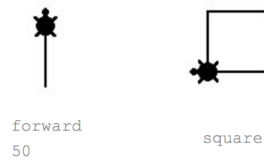


Figura 2.2: Cuadrado dibujado por el robot Turtle.

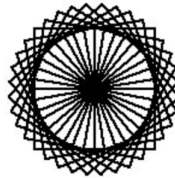


Figura 2.3: Flor dibujada por el robot Turle.

2.3. Code.org

Code.org[5] 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 [15].

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

Apéndice A

Lenguaje iJava

Bibliografía

- [1] Página oficial del proyecto eu code week.
- [2] A logo primer. http://el.media.mit.edu/logo-foundation/what_is_logo/logo_primer.html, Último acceso en enero 2016.
- [3] Página oficial de code academy. <https://www.codecademy.com>, Último acceso en enero 2016.
- [4] Página oficial de code school. <https://www.codeschool.com>, Último acceso en enero 2016.
- [5] Página oficial de code.org. <https://code.org>, Último acceso en enero 2016.
- [6] Página oficial de khan academy. <https://www.khanacademy.org>, Último acceso en enero 2016.
- [7] Página oficial de la fundación logo. <http://el.media.mit.edu/logo-foundation>, Último acceso en enero 2016.
- [8] Página oficial de moway education. <http://moway-robot.com>, Último acceso en enero 2016.
- [9] Página oficial del proyecto arduino. <https://www.arduino.cc/>, Último acceso en enero 2016.
- [10] Página oficial del proyecto código 21 promovido por el departamento de educación de gobierno de navarra. <http://codigo21.educacion.navarra.es>, Último acceso en enero 2016.
- [11] Página oficial del proyecto descubre la programación. <http://descubre.inf.um.es>, Último acceso en enero 2016.
- [12] Página oficial del proyecto hora del código. <https://hourofcode.com/>, Último acceso en enero 2016.

- [13] Página oficial del proyecto scratch. <https://scratch.mit.edu>, Último acceso en enero 2016.
- [14] Página oficial de robomind. <http://www.robomind.net>, Último acceso en enero de 2016.
- [15] Página oficial de blockly games. <https://blockly-games.appspot.com>, Último acceso enero 2016.
- [16] AHMED-ULLAH, N. S. Cps to add computer science as core subject. *Chicago Tribune* (Diciembre 2013).
- [17] BUNDY, A. Computational thinking is pervasive. *Journal of Scientific and Practical Computing* 1, 2 (2007), 67–69.
- [18] CLEMENTS, D. H. Effects of logo and cai environments on cognition and creativity. *Journal of Educational Psychology* 78, 4 (1986), 309.
- [19] COUGHLAN, S. Computer science part of english baccalaureate. *BBC News* (Febrero 2013).
- [20] DOUGLAS H. CLEMENTS, MICHAEL T. BATTISTA, J. S. Logo and geometry. *Journal for Research in Mathematics Education. Monograph 10* (2001), i–177.
- [21] FERNÁNDEZ MUÑOZ, L., PEÑA, R., NAVA, F., AND VELÁZQUEZ ITURBIDE, A. 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)* (2002), 433–440.
- [22] FEURZEIG, W., ET AL. Programming-languages as a conceptual framework for teaching mathematics. final report on the first fifteen months of the logo project.
- [23] FOR EDUCATION UK GOVERNMENT, D. 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.
- [24] LEMAIRE, M. Incorporating computer science into an elementary school curriculum.
- [25] LIAO, Y.-K. C., AND BRIGHT, G. W. Effects of computer programming on cognitive outcomes: A meta-analysis. *Journal of Educational Computing Research* 7, 3 (1991), 251–268.

- [26] MARCOS, J. Los colegios de madrid impartirán clases de programación. *El País* (Septiembre 2014).
- [27] MCGETTRICK, A., BOYLE, R., IBBETT, R., LLOYD, J., LOVEGROVE, G., AND MANDER, K. Grand challenges in computing: Education—a summary. *The Computer Journal* 48, 1 (2005), 42–48.
- [28] PEA, R. D., AND KURLAND, D. M. Logo programming and the development of planning skills. technical report no. 16.
- [29] RENUMOL, V., JAYAPRAKASH, S., AND JANAKIRAM, D. Classification of cognitive difficulties of students to learn computer programming. *Indian Institute of Technology, India* (2009).
- [30] SÁNCHEZ LAGUNA, J. A. 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).
- [31] SIEGMUND, J., KÄSTNER, C., APEL, S., PARNIN, C., BETHMANN, A., LEICH, T., SAAKE, G., AND BRECHMANN, A. Understanding understanding source code with functional magnetic resonance imaging. In *Proceedings of the 36th International Conference on Software Engineering* (2014), ACM, pp. 378–389.
- [32] STAFF, C. New priorities for european cooperation in education and training. Tech. rep., EUROPEAN COMMISSION, Agosto 2015.

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 . 3–6, 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

Scratch . 10

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