



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño de actividades educativas en Scratch para la dinamización del Museo de Informática

TRABAJO FIN DE GRADO
Grado en Ingeniería Informática

Autora: Cynthia España Sanjuan

Tutor: Xavier Molero Prieto

15 de abril de 2015

Resumen

El presente trabajo se centra en la elaboración de actividades didácticas sobre programación orientadas a los jóvenes que visiten el Museo de Informática situado en la Escola Tècnica Superior d'Enginyeria Informàtica dentro de la Universitat Politècnica de València. Estas actividades están diseñadas para la plataforma Scratch, dado que la programación resulta una habilidad que en la actualidad y según está planteado en la educación española, solo llega a desarrollarse en profundidad dentro de algunas titulaciones universitarias.

Por una parte se analiza la importancia del pensamiento computacional y la programación en el siglo XXI y por otra el estado actual de su integración desde los niveles de la Educación Primaria, pasando por educación secundaria y bachillerato. Además, se incluye una explicación extensa de la plataforma Scratch, sus orígenes, funciones y herramientas. Finalmente se proponen una serie de actividades adaptadas tanto a la educación primaria como secundaria y bachillerato y se muestran los resultados estadísticos derivados de incluir este tipo de talleres como actividades dentro del Museo de Informática.

Palabras clave: educación, programación, Scratch, pensamiento computacional, Museo de Informática

Resum

El present treball es centra en l'elaboració d'activitats didàctiques sobre programació orientades als joves que visiten el Museu d'Informàtica situat a l'Escola Tècnica Superior d'Enginyeria Informàtica dins la Universitat Politècnica de València. Aquestes activitats estan dissenyades per a la plataforma Scratch, ja que la programació és una habilitat que en l'actualitat i segons està plantejat en la educació espanyola, només arriba a desenvolupar-se en profunditat dins d'algunes titulacions universitàries.

D'una banda s'analitza la importància del pensament computacional i la programació al segle XXI i per una altra l'estat actual de la seva integració des dels nivells de l'educació primària, passant per educació secundària i batxillerat. A més s'inclou una explicació extensa de la plataforma Scratch, els seus orígens, funcions i eines. Finalment es proposen una sèrie d'activitats adaptades tant a l'educació primària com secundària i batxillerat i es mostren els resultats estadístics derivats d'incloure aquest tipus de tallers com a activitats dins del Museu d'Informàtica.

Paraules clau: educació, programació, Scratch, pensament computacional, Museu d'Informàtica

Abstract

This work focuses on the development of educational programming activities oriented to young people who visit the Museum of Informatics located in the Escuela Tècnica Superior d'Enginyeria Informàtica inside the Universitat Politècnica de València. These activities are designed to be used by the platform Scratch since programming is a skill that today, and as it is raised in Spanish Education, only reaches deep developing into some university degrees.

On one hand the importance of computational thinking and programming in the XXI century is analyzed, and on the other, the current state of integration from the levels of Primary Education, Secondary Education and Baccalaureate is detailed. Besides an extensive explanation of the Scratch platform, its origins, functions and tools will be included. Finally, a series of activities tailored to both the primary and secondary school and higher education are proposed, and derived statistical results of including such activities within the Computing Museum are displayed.

Keywords: education, programming, Scratch, computational thinking, Museum of Informatics

Índice general

| | |
|---|-----------|
| 1. Motivación y objetivos | 1 |
| 1.1. Motivación | 1 |
| 1.2. Objetivos | 2 |
| 1.3. Estructura de la memoria | 2 |
| 2. La tecnología informática en la actualidad | 5 |
| 2.1. La programación, una habilidad necesaria | 5 |
| 2.2. Enseñemos a los niños a programar | 7 |
| 2.3. Todo el mundo debería saber programar | 9 |
| 2.4. Herramientas de programación educativa | 10 |
| 2.4.1. XLogo | 10 |
| 2.4.2. Blockly | 11 |
| 2.4.3. Alice | 13 |
| 2.4.4. Code Monster from Crunchzilla | 13 |
| 2.4.5. Kodu Game Lab | 15 |
| 2.4.6. Otras herramientas | 16 |
| 3. El entorno de programación Scratch | 19 |
| 3.1. ¿Por qué elegir Scratch? | 19 |
| 3.2. Proyecto Scratch | 20 |
| 3.2.1. Finalidad del proyecto Scratch | 24 |
| 3.2.2. Usuarios de Scratch | 27 |
| 3.2.3. Scratch y el pensamiento computacional | 28 |
| 3.2.4. <i>Scratch Day</i> | 28 |
| 3.2.5. Visión futura | 29 |

| | | |
|-----------|---|-----------|
| 3.2.6. | MIT, lugar de origen | 31 |
| 3.3. | Entorno de programación en Scratch | 34 |
| 3.3.1. | Elementos básicos | 35 |
| 3.3.2. | Interfaz de Scratch | 35 |
| 3.3.3. | Multimedia | 37 |
| 3.3.4. | Menús y barras de herramientas | 38 |
| 3.3.5. | Manipulación de datos en Scratch | 40 |
| 3.4. | Scratch en la literatura | 43 |
| 3.4.1. | <i>Learn to program with Scratch</i> | 48 |
| 3.4.2. | <i>Scratch Programming</i> | 50 |
| 4. | La interacción con el mundo físico | 55 |
| 4.1. | La tarjeta PicoBoard | 55 |
| 4.2. | Integración de la PicoBoard con Scratch | 56 |
| 4.3. | Otros dispositivos | 57 |
| 4.3.1. | Raspberry Pi | 57 |
| 4.3.2. | Arduino | 58 |
| 4.3.3. | LEGO WeDo | 58 |
| 4.3.4. | MaKey MaKey | 60 |
| 4.3.5. | Kinect | 60 |
| 5. | La informática en la educación | 63 |
| 5.1. | España: la tecnología y la educación | 63 |
| 5.2. | Informática en la educación española | 64 |
| 5.2.1. | Educación primaria | 65 |
| 5.2.2. | Educación secundaria | 65 |
| 5.2.3. | Bachillerato | 68 |
| 6. | Actividades pedagógicas | 71 |
| 6.1. | Actividades para educación primaria | 71 |
| 6.1.1. | El deseo de Currutago | 71 |
| 6.1.2. | Una mascota especial | 73 |
| 6.1.3. | ¡Asústame! | 76 |
| 6.1.4. | La batalla de baile | 79 |

| | |
|--|------------|
| 6.2. Actividades para educación secundaria | 80 |
| 6.2.1. ¡Adivina el número! | 80 |
| 6.2.2. Explota la pelota | 82 |
| 6.2.3. ¡Atrápame si puedes! | 85 |
| 6.2.4. Scratch Invaders | 87 |
| 6.3. Actividades para bachillerato | 91 |
| 6.3.1. Flappy Scratch | 91 |
| 6.3.2. Scratch Test | 94 |
| 6.3.3. Pong | 99 |
| 6.3.4. El adivinador de frutas | 102 |
| 7. Integración en el Museo de Informática | 107 |
| 7.1. Oferta de la actividad | 107 |
| 7.2. Organización de la actividad | 107 |
| 7.3. Resultados obtenidos | 112 |
| 7.3.1. Organización del taller | 112 |
| 7.3.2. Desarrollo de la actividad | 113 |
| 7.3.3. Valoración global | 114 |
| 7.3.4. Tu opinión nos interesa | 116 |
| 7.4. Conclusiones | 116 |
| 8. Conclusiones y trabajo futuro | 117 |
| 8.1. Consideraciones finales | 117 |
| 8.2. Trabajo futuro | 118 |

Índice de figuras

| | |
|---|----|
| 2.1. Uso del ordenador según edad en España | 6 |
| 2.2. Mitch Resnick en la conferencia TED, 2012 | 7 |
| 2.3. Mitch Resnick en la charla TED, 2012 | 8 |
| 2.4. <i>Hour of Code</i> de code.org | 9 |
| 2.5. Gráfico comparativo | 10 |
| 2.6. Interfaz de XLogo | 11 |
| 2.7. Ejemplo de ejecución en XLogo | 12 |
| 2.8. Entorno gráfico web de Blockly | 12 |
| 2.9. Exportación de proyectos a diferentes lenguajes de programación . | 12 |
| 2.10. Interfaz de Alice | 13 |
| 2.11. Interfaz de Story Telling Alice | 14 |
| 2.12. Pantalla de Code Monster from Crunchzilla | 14 |
| 2.13. Pantalla de Kodu Game Lab | 15 |
| 2.14. Especificación de condiciones en Kodu Game Lab | 16 |
| 2.15. Pantallas de LMC | 17 |
| 2.16. Pantallas de Greenfoot | 17 |
| 2.17. Pantallas de Hot Soup Processor | 18 |
| 3.1. Logo y lema de Scratch: « <i>Imagina, crea, comparte</i> » | 20 |
| 3.2. Página web de Scratch | 21 |
| 3.3. Entorno de Scratch 2.0 en su versión <i>offline</i> | 22 |
| 3.4. Tendencias mensuales de actividad en Scratch | 23 |
| 3.5. Ejemplo de bloque repeat en forma de letra C | 24 |
| 3.6. Compartición de proyectos en Scratch en entorno web | 26 |
| 3.7. Compartición de proyectos en Scratch en entorno escritorio | 26 |

| | |
|--|----|
| 3.8. Número de usuarios de la aplicación por edades de Scratch | 27 |
| 3.9. Localización de eventos del <i>Scratch Day 2014</i> | 29 |
| 3.10. Eventos del <i>Scratch Day 2014</i> en España | 30 |
| 3.11. Interfaz del <i>Android Scratch Player</i> | 30 |
| 3.12. Interfaz del <i>ScratchJr</i> | 31 |
| 3.13. Mitch Resnick, líder de <i>The Lifelong Kindergarten</i> | 32 |
| 3.14. Miembros del <i>Computer Clubhouse</i> en México | 33 |
| 3.15. Interfaz web de <i>Build in Progress</i> | 33 |
| 3.16. Interfaz de <i>DressCode</i> | 34 |
| 3.17. <i>Makey Makey</i> usando pastilina a modo de <i>gamepad</i> | 35 |
| 3.18. Diferentes disfraces que puede adoptar un objeto | 36 |
| 3.19. Interfaz de Scratch dividida en sus diferentes elementos | 36 |
| 3.20. Editor de imágenes de Scratch | 38 |
| 3.21. Herramienta de sonido de Scratch | 39 |
| 3.22. Menú general | 39 |
| 3.23. Barra de herramientas general | 39 |
| 3.24. Herramientas de objeto | 40 |
| 3.25. Herramientas de fondo | 40 |
| 3.26. Herramientas de edición de imagen | 41 |
| 3.27. Herramientas de edición de sonido | 41 |
| 3.28. Instrucciones de manejo de variables | 42 |
| 3.29. Instrucciones de manejo de listas | 43 |
| 3.30. Portada de <i>Programación Scratch para niños</i> | 44 |
| 3.31. Portada de <i>Scratch para niños... y no tan niños</i> | 45 |
| 3.32. Portada de <i>Super Scratch Programming Adventure!</i> | 46 |
| 3.33. Portada de <i>Scratch 2.0 Game Development Hotshot</i> | 47 |
| 3.34. Portada de <i>Learn to program with Scratch</i> | 48 |
| 3.35. Portada de <i>Scratch Programming</i> | 51 |
| 4.1. La tarjeta PicoBoard y sus elementos | 56 |
| 4.2. Instrucciones de Scratch para la PicoBoard | 57 |
| 4.3. Raspberry Pi | 58 |
| 4.4. Interfaz de S4A | 59 |

| | |
|---|----|
| 4.5. LEGO WeDo usando Scratch | 59 |
| 4.6. Componentes del kit MaKey MaKey | 60 |
| 4.7. MaKey MaKey usando un plátano | 61 |
| 4.8. Stephen Howell presentando Kinect2Scratch, 2012 | 61 |
| 5.1. Portada de <i>Informática 4º E.S.O.</i> | 66 |
| 5.2. Portada de <i>Tecnología de la Información y la Comunicación</i> | 68 |
| 6.1. El deseo de Currutago en ejecución | 72 |
| 6.2. Ubicación de la variable «respuesta» | 74 |
| 6.3. Instrucciones condicionales utilizadas | 75 |
| 6.4. Una mascota especial en ejecución | 75 |
| 6.5. Instrucciones utilizadas en el objeto murciélago | 76 |
| 6.6. Ejemplo de bucle simple | 77 |
| 6.7. ¡Asústame! en ejecución | 77 |
| 6.8. Uso de instrucciones para la PicoBoard (en color gris) | 78 |
| 6.9. La batalla de baile en ejecución | 79 |
| 6.10. Implementación del fondo del proyecto | 80 |
| 6.11. Código de implementación del Jugador 1 | 81 |
| 6.12. ¡Adivina el número! en ejecución | 81 |
| 6.13. Código del objeto mago | 83 |
| 6.14. Instrucción de aleatoriedad en Scratch | 83 |
| 6.15. Explota la pelota en ejecución | 84 |
| 6.16. Programa del objeto pelota | 85 |
| 6.17. Programa del contador en el fondo | 86 |
| 6.18. ¡Atrápame si puedes! en ejecución | 86 |
| 6.19. Programación del objeto pez | 87 |
| 6.20. Condición de parada en la variable de tiempo | 88 |
| 6.21. Scratch Invaders en ejecución | 88 |
| 6.22. Implementación del objeto misil | 89 |
| 6.23. Implementación del objeto nave | 90 |
| 6.24. Flappy Scratch en ejecución | 92 |
| 6.25. Disfraces del objeto obstáculo | 92 |
| 6.26. Código del objeto obstáculo | 93 |

| | |
|--|-----|
| 6.27. Código del objeto gato | 95 |
| 6.28. Scratch Test en ejecución | 96 |
| 6.29. Listas en Scratch | 97 |
| 6.30. Implementación con listas y variables | 98 |
| 6.31. Envío de mensaje en Scratch | 99 |
| 6.32. Recepción de mensaje en Scratch | 100 |
| 6.33. Pong en ejecución | 100 |
| 6.34. Código del objeto pelota | 101 |
| 6.35. Código incluido en el escenario | 102 |
| 6.36. Foto de la fruta | 103 |
| 6.37. Mensaje con el nombre de la fruta | 103 |
| 6.38. Código para adivinar el limón | 104 |
| 6.39. Código del objeto limón | 104 |
| 6.40. Programa en ejecución | 105 |
| | |
| 7.1. Información del taller en la web del museo | 108 |
| 7.2. Taller de Scratch | 109 |
| 7.3. Boletín del Taller de Scratch | 110 |
| 7.4. Encuesta sobre el Taller de Scratch del Museo de Informática | 111 |
| 7.5. Grado de satisfacción global con la organización del taller | 113 |
| 7.6. Grado de satisfacción detallado con la organización del taller | 113 |
| 7.7. Grado de satisfacción global con el desarrollo de la actividad | 114 |
| 7.8. Grado de satisfacción detallado con el desarrollo de la actividad | 115 |
| 7.9. Grado de satisfacción general | 115 |
| 7.10. Grado de satisfacción general detallado | 116 |

CAPÍTULO 1

Motivación y objetivos

En este primer capítulo expondremos las razones que han llevado a cabo a la elaboración del presente trabajo así como los objetivos a conseguir.

1.1 Motivación

El Museo de Informática¹ de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València fue inaugurado el 11 de diciembre del año 2001 y ofrece un fondo museográfico que permite el recorrido histórico a través de las últimas décadas de la informática.

Cabe destacar el reconocimiento de su actividad por parte de la Conselleria d'Educació, Cultura i Esport de la Generalitat Valenciana, que el 13 de mayo de 2013 lo incluyó como museo oficial de la Comunitat Valenciana² así como su incorporación dentro del Consejo Internacional de Museos (ICOM) de la UNESCO en el año 2015.

Dentro de las actividades culturales que ofrece tales como la visita a su exposición permanente, exposiciones temporales o conferencias, incluye un apartado dedicado a la educación. En este ámbito se encuentra la realización de visitas guiadas para alumnos de educación primaria y secundaria, bachillerato y ciclos formativos así como talleres de retroprogramación para acercar a los visitantes tanto la historia de la informática como despertar en ellos el interés por la programación. Dentro de esta visión, el Museo de Informática quiere ofrecer un apartado más de actividades, orientadas al mundo de la programación y la educación por las cuales puedan acercar a sus visitantes, muchos nacidos en la era de las nuevas tecnologías, la opción no solo de entender estas tecnologías sino de usarlas como herramientas para crear y expresarse.

¹Página accesible en <http://museo.inf.upv.es>.

²Resolución publicada en el DOCV, el 28 de mayo de 2013.

1.2 Objetivos

El trabajo tiene como objetivo principal diseñar un conjunto de actividades didácticas basadas en el entorno de programación Scratch orientadas a estudiantes de educación primaria, secundaria y bachillerato, que les permita establecer un contacto lúdico con la programación, una de las destrezas y habilidades necesarias para desenvolverse en el mundo tecnológico del siglo XXI.

En nuestra opinión, los requisitos esenciales para diseñar este conjunto de actividades deben tener en cuenta:

1. Los recursos y el tiempo necesarios para llevarlos a cabo.
2. El nivel de los alumnos que participan en las actividades.
3. La incorporación de la interacción del ordenador mediante el mundo físico con el uso de tarjetas con sensores como pueden ser de luz, sonido o resistencia eléctrica.

Además del diseño de estas actividades como objetivo principal, se debe tener también en cuenta como objetivo secundario la dinamización y promoción de dichas actividades utilizando como plataforma principal el Taller de Scratch que imparte el Museo de Informática.

Este trabajo incluye el estudio y análisis de todas las posibles opciones y herramientas disponibles para implementar dichas actividades y exponer las razones por las cuales se decide a utilizar como medio el entorno de programación Scratch.

1.3 Estructura de la memoria

El presente trabajo se estructura en ocho capítulos. A continuación se expondrá un breve resumen de cada uno de ellos.

- **Capítulo 1.** Se expone la motivación por la cual se ha llevado a cabo este trabajo, así como los objetivos esenciales a abarcar.
- **Capítulo 2.** Se analiza la importancia de la programación como herramienta fundamental del siglo XXI. Además, se enumeran algunas herramientas educativas para programar, que podemos encontrar hoy en día de forma gratuita.
- **Capítulo 3.** En este capítulo se detalla en profundidad la plataforma Scratch, su origen, funcionalidad y herramientas, así como su bibliografía existente.

- **Capítulo 4.** Esta unidad se centra en la interacción de Scratch con el mundo físicos y los dispositivos externos compatibles con el programa, haciendo espacial énfasis en la tarjeta PicoBoard.
- **Capítulo 5.** Este capítulo pasa a exponer la situación actual española con respecto a la incorporación de la programación en su sistema educativo. Se analizan con detalle los ciclos de educación primaria, secundaria, y bachillerato, para poder adecuar las actividades propuestas a cada uno de los niveles.
- **Capítulo 6.** En esta unidad se recogen las actividades pedagógicas propuestas. Para cada uno de los ciclos de enseñanza, se proponen cuatro actividades adaptadas al nivel de dificultad correspondiente. En cada una de estas actividades se recogerán conceptos teóricos sobre programación que el alumno logrará aprender mientras las lleva a cabo.
- **Capítulo 7.** Este capítulo analiza por un lado, la motivación del Museo de Informática, su funcionamiento y actividades. Por otro, recoge datos estadísticos sobre el Taller de Scratch, actividad en la cual se plantea difundir los proyectos propuestos en el presente trabajo.
- **Capítulo 8.** El último capítulo recoge las principales conclusiones obtenidas. Analiza si se han cumplido los objetivos planteados, así como la utilidad futura que puede tener el presente proyecto, enumerando posibles mejoras y trabajos derivados del mismo.

CAPÍTULO 2

La tecnología informática en la actualidad

En este capítulo pasaremos a exponer el estado actual de la tecnología en nuestra sociedad, la importancia de la informática y la demanda de trabajo en este sector. Además, se expondrán algunas de las herramientas actuales de programación educativas que existen, analizando su funcionamiento y características.

2.1 La programación, una habilidad necesaria

Según el Instituto Nacional de Estadística (INE), en una nota de prensa presentada el 25 de octubre de 2013¹, el porcentaje de jóvenes (entre 10 y 15 años) que usan un ordenador en nuestro país se sitúa en el 95.2 %. También afirma que el uso del ordenador es una práctica mayoritaria en edades anteriores a los 10 años tal y como muestra la gráfica de la Figura 2.1.

Por otro lado, según la Conferencia de Directores y Decanos de Ingeniería Informática (CODDII), la tecnología informática debería ser cursada por todos los estudiantes de secundaria y bachillerato tal y como señalaron en una declaración conjunta con la Asociación de Enseñantes Universitarios de la Informática (AE-NUI) el 23 de junio de 2014².

Esta declaración conjunta afirma que la Comisión Europea estima que hacia el año 2020 se podrían necesitar cerca de 900.000 profesionales en el sector de las TIC.

¹Documento disponible en la web del INE (<http://www.ine.es>) bajo el nombre de *Encuesta sobre Equipamiento y Uso de Tecnologías de Información y Comunicación en los Hogares (TIC-H)*. Año 2013.

²Documento disponible en la web de la CODDII (<http://www.coddii.org>) bajo el nombre *Por la inclusión de asignaturas específicas de ciencia y tecnología informática en los estudios básicos de la enseñanza secundaria y bachillerato*. Año 2014.



Figura 2.1: Uso del ordenador según edad en España

Por lo tanto se encuentra la paradoja de que hay una generación que ha nacido inmersa en una era digital que se siente poco atraída por el estudio de la informática, la cual debería, por otra parte, considerarse una competencia básica e introducirse en edades tempranas como la física o las matemáticas.

Así pues, ambas instituciones concluyen que la ciencia informática debería además, aplicarse en cualquier área de conocimiento de secundaria y bachillerato y promover habilidades como la programación mediante talleres y actividades extraescolares.

Finalmente, otro estudio de la CODDII del año 2013³ asegura que la empleabilidad en ese año de la Ingeniería Informática superaba en España el 91 % siendo el porcentaje de estudiantes que trabajan mientras acaban la carrera de un 55 %.

En conclusión, la temprana edad en la que empieza a utilizarse el ordenador, el elevado porcentaje de jóvenes que lo usan, así como la demanda profesional tanto en el área puramente informática como en el resto de áreas profesionales de conocimientos informáticos, requieren una metodología educativa que enseñe desde jóvenes no solo a saber manejar o «leer» las tecnologías informáticas sino también a crear o «escribir» para adaptarlas a sus necesidades, y para ello la habilidad de programar debería ser fundamental.

La importancia de la programación ya no solo en los profesionales de la informática sino en todos los sectores y personas, es el argumento principal de dos reportajes que pasarán a analizarse a continuación.

³Documento accesible en <http://coddii.org/wp-content/uploads/2013/04/coddinforme-empleabilidad-2013.pdf> bajo el nombre *Empleabilidad 2013*.

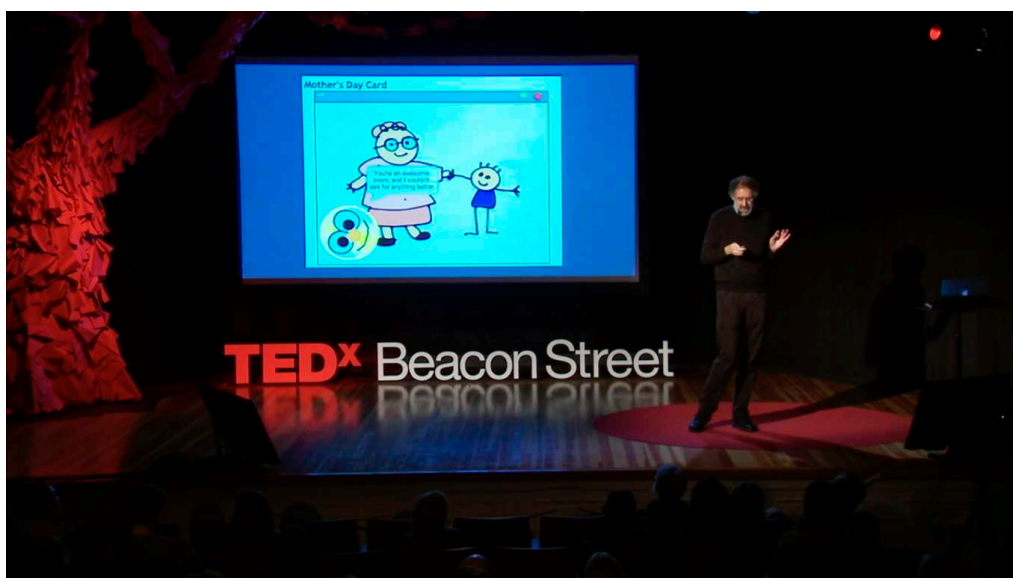


Figura 2.2: Mitch Resnick en la conferencia TED, 2012

2.2 Enseñemos a los niños a programar

Mitch Resnick, profesor y cabeza del *Lifelong Kindergarten Group*, grupo que desarrolla Scratch, defiende la importancia de que todo el mundo sepa programar⁴. Ejemplo de ello lo podemos encontrar en una charla TED titulada titulada Enseñemos a los niños a programar: («*Let's teach kids to code*») en la que habló de la importancia de enseñar a los niños a programar tal como muestra la Figura 2.2.

Los jóvenes de hoy en día son llamados nativos digitales [11] porque han nacido inmersos en una sociedad llena de tecnologías que, aparentemente, saben controlar. Pero este término no resulta del todo correcto ya que queda ampliamente demostrado que los jóvenes son expertos en escribir mensajes, jugar a videojuegos o navegar por Internet pero tienen prácticamente un conocimiento nulo en escribir código o crear programas a su gusto. Así pues, puede decirse que saben «leer» pero no «escribir» con las nuevas tecnologías. Éste es un problema al que se le empieza a intentar buscar soluciones ocasionando debates sobre educación y que ya en Internet muchas páginas web intentan solucionar dedicándose exclusivamente a enseñar a los jóvenes de manera *online* a programar.

La programación no es una habilidad cerrada que solo sirve para el propósito de crear código para un ordenador por un programador o informático, sino que sirve para desarrollar otras habilidades y aprender otras nuevas al igual que leer y escribir no sirve solo para formar a escritores sino que ayuda a abrir fronteras

⁴Material accesible desde la web de TED <http://www.ted.com> *Let's teach kids to code* por Mitch Resnick, 2012.



Figura 2.3: Uso de la cámara de vídeo con Scratch

y facilita la adquisición de muchos más conocimientos en el resto de ámbitos del aprendizaje.

Así mismo, mientras aprendemos a programar podemos programar aprendiendo ya que conceptos como el de «variable» acaban siendo imprescindibles en la programación, por lo que al darles un uso práctico quedan más firmemente aprendidos que si se enseñaran de forma teórica desde una clase sin adaptarlo a un contexto personal.

El papel de Scratch como herramienta de aprendizaje de la programación es muy importante ya que no solo permite crear tu propio programa, sino compartirlo y ver proyectos compartidos pudiendo así ampliarlos al gusto. Además, usa características como sensores de vídeo o audio que permite que se pueda aprender divirtiéndose (véase la Figura 2.3).

Finalmente, mediante la programación pueden inculcarse habilidades generales como el diseño, por el cual a partir de una idea compleja y abstracta se logra descomponer en partes más pequeñas y simples hasta que se llega a un proyecto concreto o el trabajo en grupo y la colaboración, así como la habilidad de solucionar problemas y la persistencia ante algo que no funcione bien. Habilidades que son útiles no solo en la informática sino en cualquier ámbito profesional.

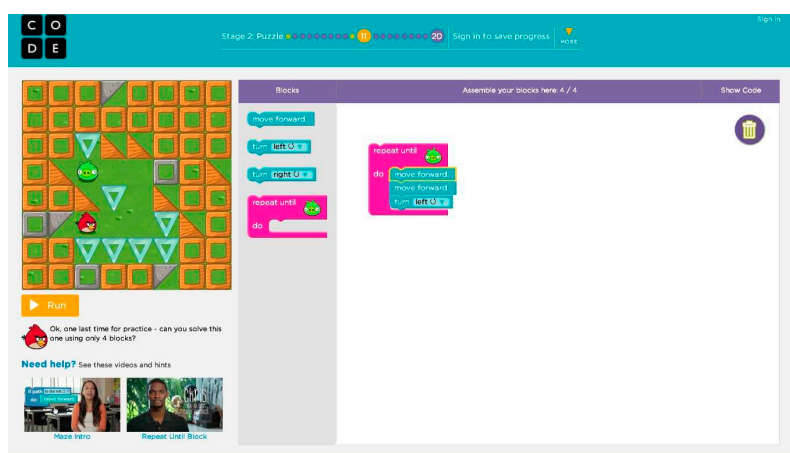


Figura 2.4: Hour of Code de code.org

2.3 Todo el mundo debería saber programar

Code.org⁵ es una organización sin ánimo de lucro cuya finalidad consiste en la de despertar en los jóvenes estudiantes el interés por programar y enseñarles de forma sencilla cómo hacerlo. Su método se llama Hora del Código (*Hour of Code*) y enseña al alumno de forma gradual, mediante su página web y el uso de Scratch, a aprender a programar en intervalos de una hora y un total de veinte lecciones. En la Figura 2.4 se ilustra una de las lecciones del *Hour of Code*.

Esta organización la idea de que todo el mundo debería saber programar, una habilidad que los niños de hoy en día desconocen y que resulta esencial para cualquier ámbito de la actualidad. La habilidad de programar actualmente es necesaria en cualquier tipo de industria y no solo la meramente informática, ya que los ordenadores y los programas o aplicaciones se encuentran por todas partes. La programación es algo que puede aprenderse y que no debería resultar diferente de aprender a tocar un instrumento o practicar un deporte.

Según uno de los pequeños cortos que tienen disponibles en su página web titulado *Code Stars*⁶ en Estados Unidos en los próximos diez años habrán alrededor de un millón cuatrocientos mil trabajos en informática en contraste a los solo 400.000 graduados cualificados para esos trabajos, por lo que se creará una carencia de un millón de puestos (véase la Figura 2.5).

Empezar a programar parte de la idea de algo propio que se quiera crear y la determinación de llevarlo a cabo, forma parte de la cultura de crear cosas y es una forma de expresar creatividad y trabajar en equipo. Además, integrar las asignaturas de ciencia con la programación tiene un efecto beneficioso para el estudiante, ya que tal como se menciona en el documental, las notas de una clase en particular donde se usó, subieron en un 30 %.

⁵Accesible desde su sitio web <http://www.code.org>.

⁶Material accesible desde la web de Code.org <http://www.code.org> *Code Stars*, 2014

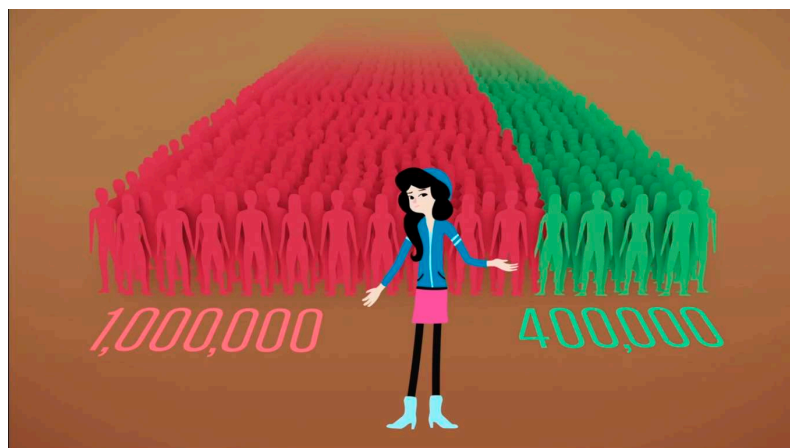


Figura 2.5: Demanda y oferta de puestos de trabajo en informática en EE.UU. según el documental *Code Stars*

Finalmente, es esencial empezar a desarrollar la habilidad de programar desde pequeño, ya que no es necesario aprender una gran cantidad de información antes de ser capaz de empezar a programar ni tampoco ser un genio, de la misma forma que no se necesita serlo para leer.

2.4 Herramientas de programación educativa

Además del entorno de desarrollo Scratch, el cual se tratará en profundidad en el siguiente capítulo, existen otras herramientas y lenguajes de programación diseñados con la finalidad de conseguir que los estudiantes empiecen desde cero en un entorno de programación entretenido y vayan progresando de forma fácil para concluir en entornos más profesionales. En los siguientes apartados se tratarán algunos ejemplos de estas herramientas.

2.4.1. XLogo

Se trata de un intérprete del lenguaje LOGO de código abierto⁷. El lenguaje LOGO nació como derivación del lenguaje LISP (*List Processing*), diseñado en los años 70 del siglo XX por Seymour Papert, matemático y psicólogo. Papert creó la teoría de aprendizaje del constructivismo por la cual se afirma que la construcción de artefactos (como el hecho de escribir un programa) ayudan al aprendizaje y además a construir las propias estructuras de conocimiento del estudiante. Mediante programación en LOGO, ha descrito guías de aprendizaje para niños por las cuales al final, serían capaces de aprender matemáticas de una forma natural. Según Seymour Papert, una aula abarrotada de tecnología es beneficiosa ya que

⁷Accesible desde su página web <http://xlogo.tuxfamily.org/>.

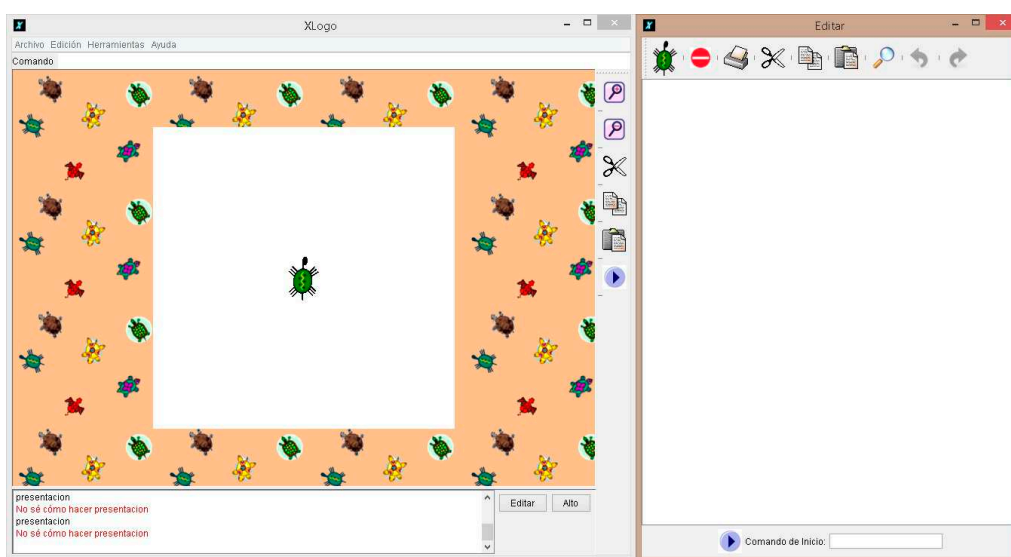


Figura 2.6: Interfaz de XLogo

al contrario de lo que se puede pensar, no aísla al estudiante sino que le ayuda a interactuar con sus compañeros y profesores [9].

Así pues, el lenguaje LOGO está orientado a introducir en programación a los niños, ya que se concentra en la creación de imágenes y formas, haciéndolo un lenguaje ideal para principiantes y especialmente fácil y atractivo para ellos.

El usuario mueve un objeto llamado «tortuga» dentro de la pantalla mediante instrucciones simples como «avanza» o «retrocede» haciendo que con cada movimiento se dibuje una línea, consiguiendo así crear los gráficos. LOGO también puede manipular listas y palabras (véanse las Figuras 2.6 y 2.7).

XLogo dispone de una versión de escritorio así como una versión vía navegador y requiere tener instalado Java para poder ejecutarse.

2.4.2. Blockly

Se trata de una herramienta de código abierto orientada a estudiantes y desarrollada por Google que ofrece a su vez un apartado para desarrolladores que quieran crear proyectos externos con ella. La herramienta es accesible vía web mediante un navegador⁸. En la Figura 2.8, puede observarse la ejecución de Blockly por la cual podemos hacer que un personaje se mueva a través de un laberinto mediante la combinación de bloques que determinan su comportamiento.

Blockly nos permite crear secillos scripts mediante el arrastre de bloques y luego exportar el resultado a lenguajes como JavaScript, Python, XML y Dart. En la Figura 2.9 podemos observar como, al presionar el icono señalado mediante un círculo, genera el código en el lenguaje que deseemos (en este caso JavaScript).

⁸Accesible en <https://blockly-games.appspot.com/>.

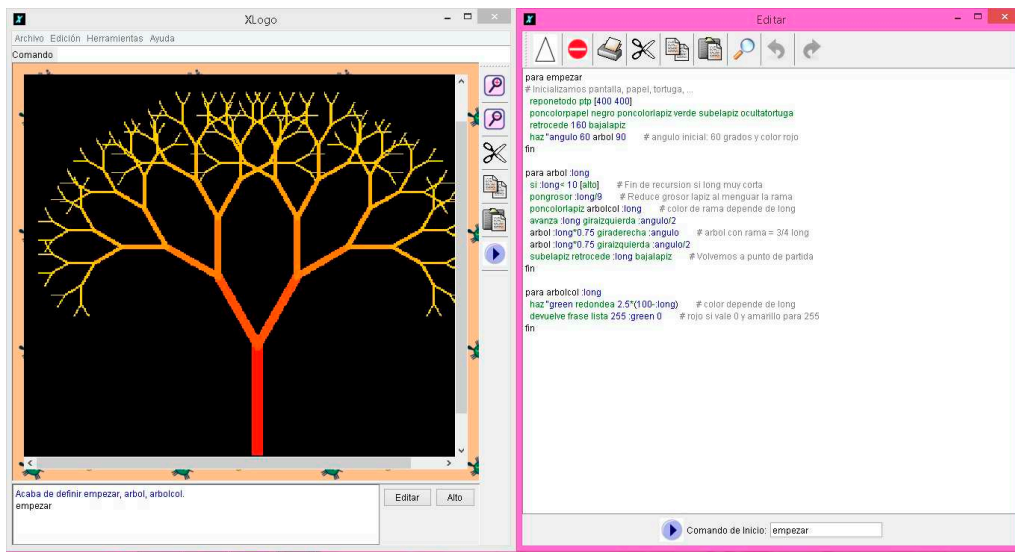


Figura 2.7: Ejemplo de ejecución en XLogo

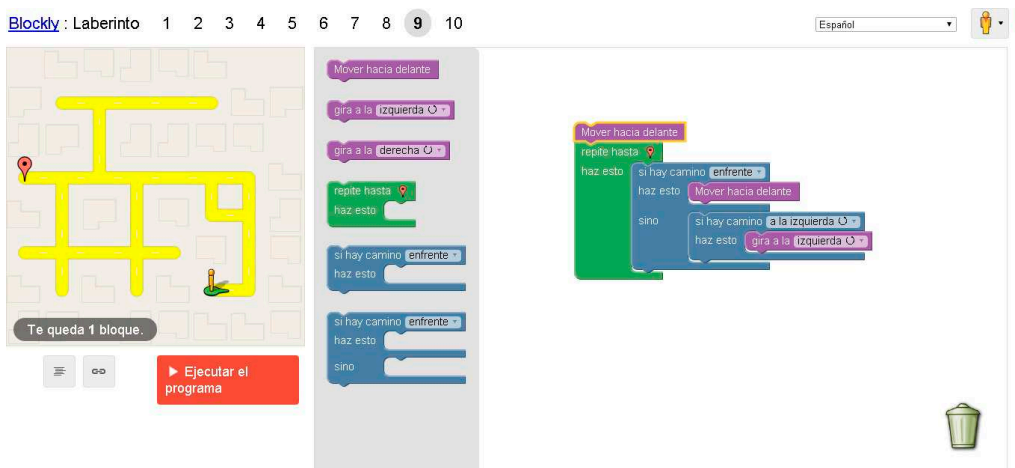


Figura 2.8: Entorno gráfico web de Blockly

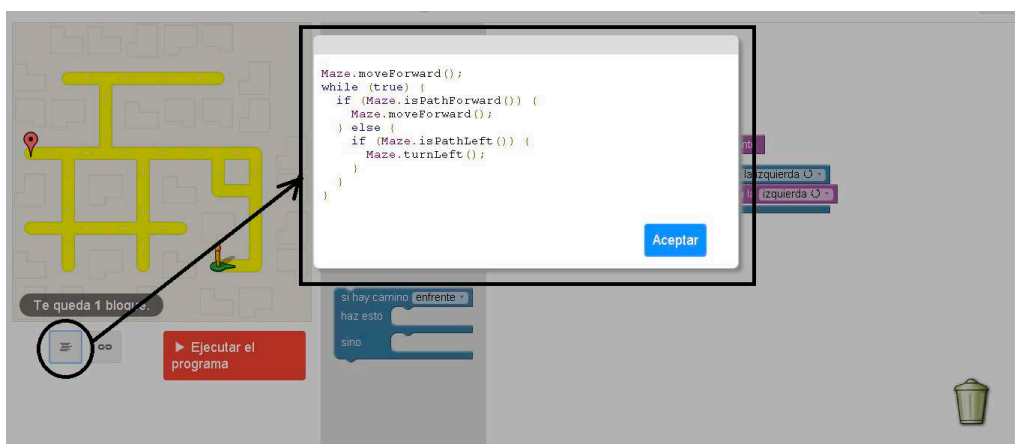


Figura 2.9: Exportación de proyectos a diferentes lenguajes de programación

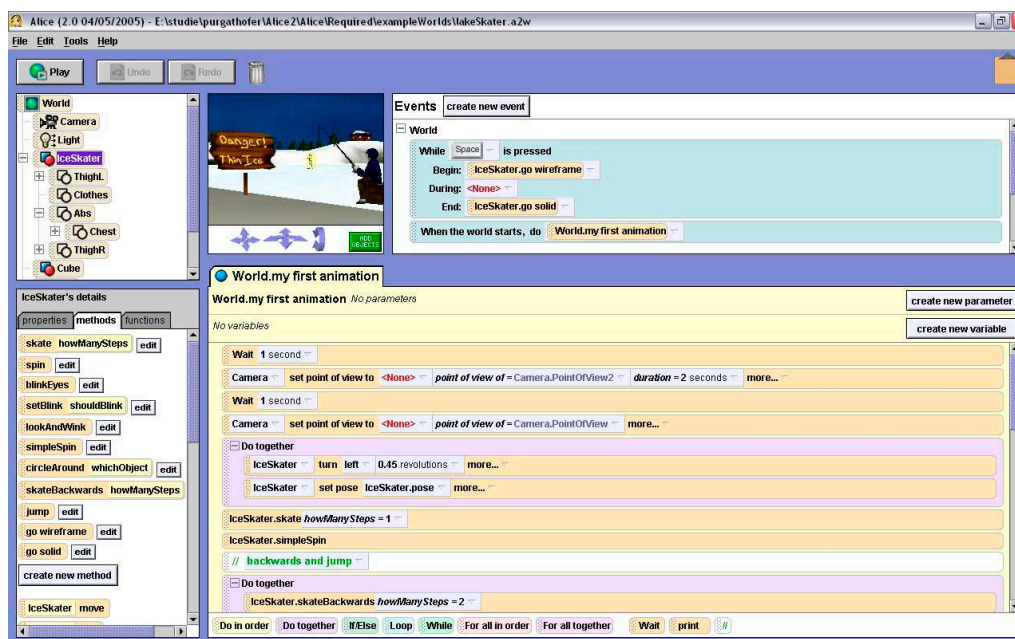


Figura 2.10: Interfaz de Alice

2.4.3. Alice

Alice es una herramienta de escritorio orientada a la programación 3D que permite crear animaciones, juegos interactivos o vídeos (véase la Figura 2.10). Se trata de un proyecto desarrollado conjuntamente entre investigadores de diversas universidades americanas y se diseñó con el objetivo de introducir a los estudiantes de secundaria y bachillerato en la programación orientada a objetos.

Al igual que Blockly o Scratch, el funcionamiento de Alice se basa en el arrastre de bloques a una zona de eventos para especificar el comportamiento del objeto que se verá en la ejecución final.

Existe una versión de Alice diseñada específicamente para niños llamada Story Telling Alice, en la cual se pone especial énfasis en el aspecto de poner al alcance del usuario todas las herramientas posibles y necesarias para desarrollar programas que cuenten una historia o cuento, haciéndola más divertida para los niños (véase la Figura 2.11).

2.4.4. Code Monster from Crunchzilla

Code Monster from Crunchzilla es un tutorial interactivo, *online* y gratuito en forma de juego desarrollado por Greg Linder, informático que ha trabajado en empresas como Microsoft y Amazon. En esta última, por ejemplo, diseñó su actual motor de recomendaciones. Está destinado a jóvenes entre 9 y 14 años que, mediante un conjunto de 59 lecciones, aprenden de forma gradual la programación mediante JavaScript.

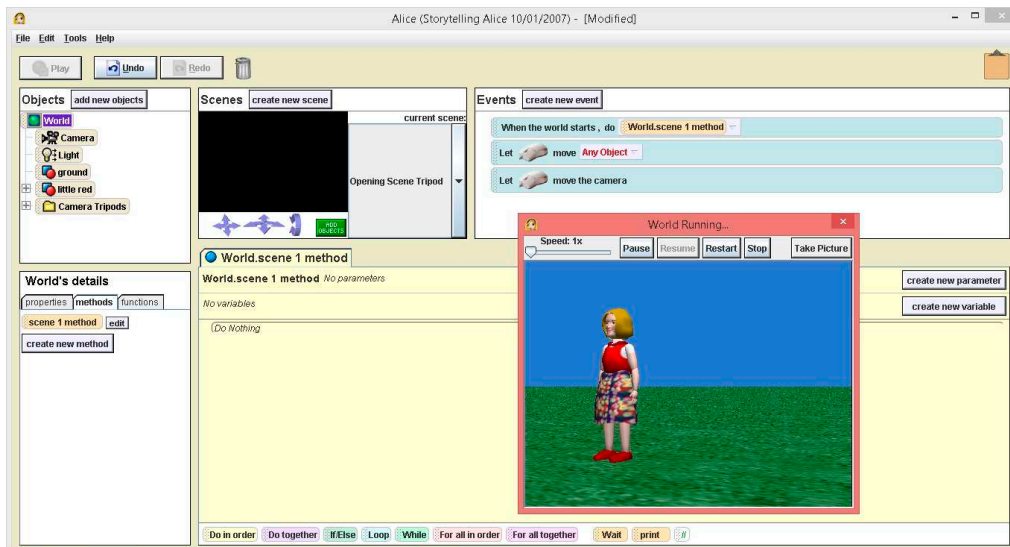


Figura 2.11: Interfaz de Story Telling Alice

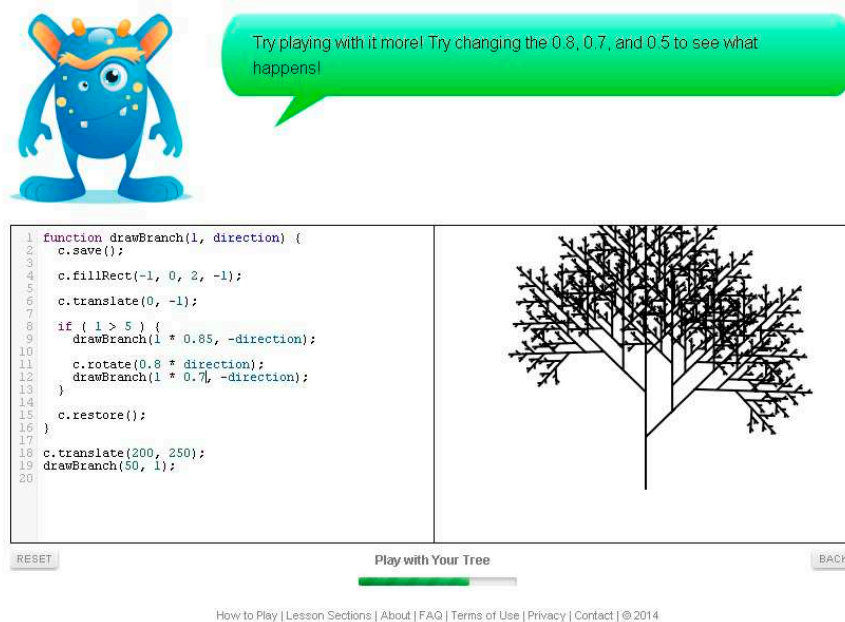


Figura 2.12: Pantalla de Code Monster from Crunchzilla



Figura 2.13: Pantalla de Kodu Game Lab

La interfaz se divide en dos paneles tal como muestra la Figura 2.12. En el panel de la izquierda se observa el código programado por el cual se va guiando al alumno a través de diálogos con el Code Monster y en el de la derecha aparecen inmediatamente los resultados del código introducido. La aplicación abarca desde conocimientos básicos como parámetros o variables hasta aspectos más complejos como funciones o animación.

2.4.5. Kodu Game Lab

Kodu Game Lab es una herramienta de programación visual para niños desarrollada por Microsoft en el año 2009 y enfocada a la creación de videojuegos en 3D. Puede utilizarse tanto en su versión de PC (de forma gratuita) como en la videoconsola XBOX 360. En la Figura 2.13 se muestra la interfaz del programa en su versión de escritorio con los menús de que dispone.

La herramienta permite interacción rápida e intuitiva ya que únicamente usa elementos visuales para la programación, sin recurrir a código escrito. Dentro, los programas se dividen en páginas que, a su vez, se componen de reglas y acaban dividiéndose en condiciones o acciones permitiendo crear mundos, donde diferentes objetos dentro de ellos interactúan según las órdenes que se les programan. En la Figura 2.14 se puede observar la interfaz de especificación de condiciones de Kodu Game Lab.

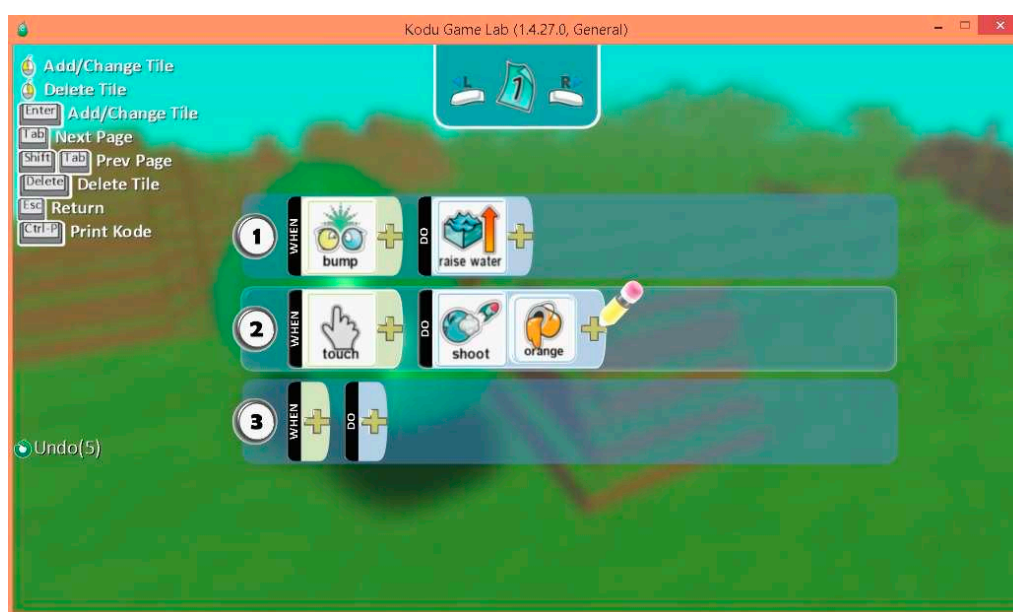


Figura 2.14: Especificación de condiciones en Kodu Game Lab

2.4.6. Otras herramientas

Existen una gran variedad de herramientas orientadas al aprendizaje de la programación por los más jóvenes, aparte de las mencionadas, que abarcan desde la implementación de interfaces para adaptar de forma sencilla lenguajes de programación tales como BASIC o ensamblador, como la creación de lenguajes nuevos con el propósito de familiarizar a los usuarios con un entorno de programación. Algunos ejemplos de estas herramientas pasan a enumerarse a continuación.

Existen una gran variedad de herramientas orientadas al aprendizaje de la programación por los más jóvenes, aparte de las mencionadas, que abarcan desde la implementación de interfaces para adaptar de forma sencilla lenguajes de programación tales como BASIC o ensamblador, como la creación de lenguajes nuevos con el propósito de familiarizar a los usuarios con un entorno de programación. Algunos ejemplos concretos de estas herramientas pasan a enumerarse a continuación.

- LMC (*Little Man Computer*) simula un modelo simple de la arquitectura de John von Neumann que puede ser programada en ensamblador o código máquina (véase la Figura 2.15). La aplicación se encuentra accesible desde la página web⁹.
- Greenfoot es un entorno de desarrollo en Java gratuito de propósito educativo que permite el desarrollo fácil de aplicaciones en 2D tal y como muestra

⁹Página web alojada en <http://www.gcsecomputing.org.uk/lmc/lmc.html>.

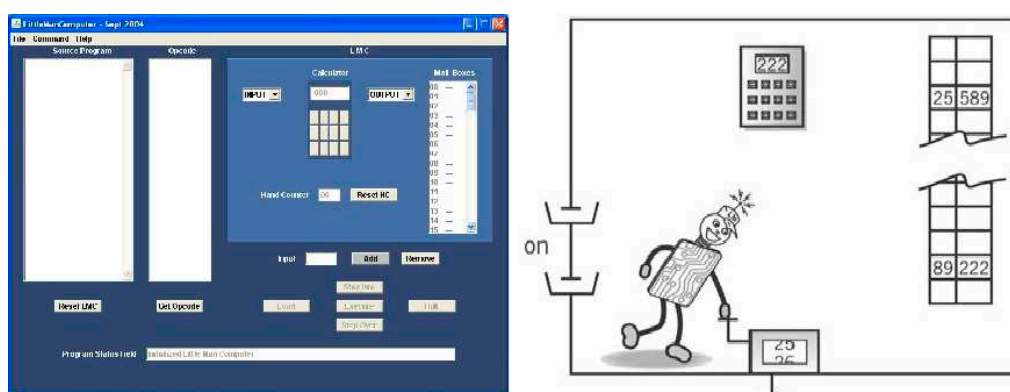


Figura 2.15: Pantallas de LMC

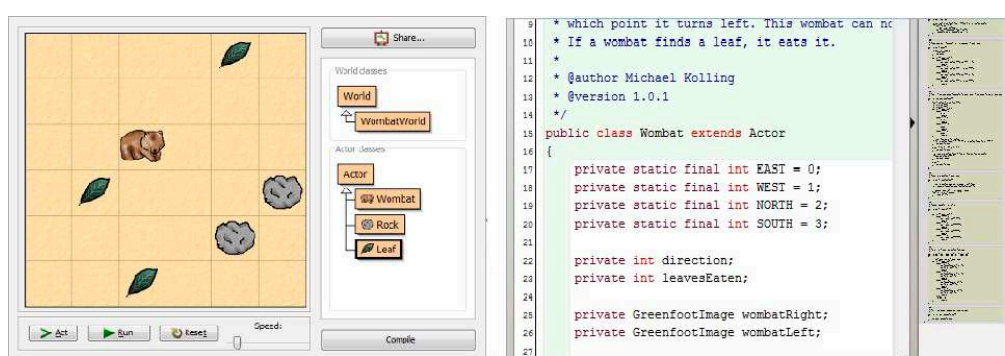


Figura 2.16: Pantallas de Greenfoot

la Figura 2.16. La aplicación está disponible para Mac, Windows y Ubuntu desde su página web¹⁰.

- Hot Soup Processor es un programa desarrollado en Japón para enseñar programación en los institutos. Partió como derivación del lenguaje BASIC pero ha ido evolucionando con los años a un lenguaje de programación propio. La Figura 2.17 muestra su última versión estable correspondiente a la 3.3 y que puede encontrarse en su página web¹¹.

En resumen, todas estas aplicaciones educativas que tienen el propósito de enseñar la programación y hacer que el estudiante se familiarice con su entorno, tienen en común, ya sea en su versión web o mediante la descarga de software, que son totalmente gratuitas para el usuario y además suelen venir acompañadas por manuales y ejemplos descargables disponibles en la web de la aplicación. Además inciden en el uso de sonidos, gráficos, vídeos etc. que ayudan a mantener el interés en los alumnos de menor edad. En el próximo capítulo veremos que Scratch no es una excepción.

¹⁰ Accesible en <http://www.greenfoot.org/>.

¹¹ Accesible en <http://hsp.tv/>.

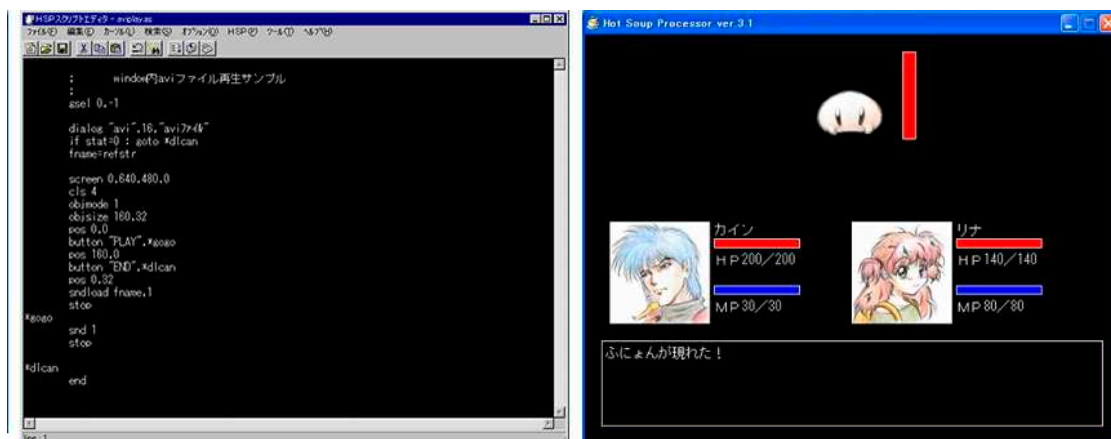


Figura 2.17: Pantallas de Hot Soup Processor

CAPÍTULO 3

El entorno de programación Scratch

En este capítulo se describe con detalle el entorno de programación Scratch, primero situando su origen y conociendo tanto su finalidad como su visión de futuro y, posteriormente, explicando su funcionamiento.

3.1 ¿Por qué elegir Scratch?

Para el presente trabajo, se ha decidido la utilización de Scratch para la creación de las actividades educativas para enseñar a programar por diversas razones:

- **Plataforma muy desarrollada.** Tanto la web como la aplicación están constantemente actualizadas y se encuentran en más de 40 idiomas.
- **El componente social.** El sitio web de Scratch permite darse de alta mediante una cuenta de usuario para así poder compartir los proyectos creados, comentar, añadir a favoritos y puntuar otros proyectos. De ahí su lema «*Imagina, programa, comparte*» tal como se muestra en la Figura 3.1. Aunque los usuarios no registrados no pueden disfrutar de este componente social, pueden programar en la plataforma sin necesidad de registrarse. Actualmente, la página web de Scratch alberga más de ocho millones de proyectos¹.
- **Trabajo colaborativo.** Todos los trabajos alojados en la web de Scratch incorporan la visualización de su código, permitiendo así, su ampliación o modificación por cualquier otro usuario.

¹8.648.906 según la página web de Scratch (<http://scratch.mit.edu>) consultada en marzo de 2015.



Figura 3.1: Logo y lema de Scratch: «Imagina, crea, comparte»

- **Facilidad de uso.** Scratch ahorra la escritura de código que puede inducir a errores, reemplazándolo por el arrastre de bloques que se conectan entre sí como un puzzle, facilitando su uso y comprensión.
- **Variedad de bibliografía.** Scratch posee una amplia bibliografía tanto física como digital de su versión 1.4 y de la actual 2.0.

3.2 Proyecto Scratch

Scratch es una plataforma de programación gratuita y una comunidad *online* orientada a los estudiantes para aprender a programar mediante la creación de historias interactivas, juegos y animaciones. Scratch ayuda a los jóvenes a aprender y a pensar creativamente, razonar sistemáticamente y trabajar colaborativamente, habilidades esenciales para la vida en el siglo XXI. Es accesible desde su página web `scratch.mit.edu` mostrada en la Figura 3.2.

El proyecto Scratch nació en el año 2003 de la mano del grupo de investigación *Lifelong Kindergarten* del MIT Media Lab, liderado por Mitch Resnick. Este grupo cree que es muy importante para todos los niños que crezcan aprendiendo a diseñar, expresarse y a usar su creatividad, ya que Mitch y su grupo afirman que hoy en día no suelen tener esa oportunidad [15].

El término «scratch» proviene del nombre dado a una técnica de hip hop llamada «scratching» por la cual los disc-jockeys experimentan con la música a base de reproducir los discos de vinilo hacia delante y hacia atrás con sus manos, mezclando clips de música de forma creativa. La programación en Scratch intenta ser algo similar, mezclando gráficos, animaciones, fotos, música y sonido de una forma altamente interactiva [14].

La herramienta de trabajo Scratch se trata de un proyecto de desarrollo cerrado y de código abierto. Esto significa que el equipo de desarrollo no persigue la contribución de la comunidad de usuarios, sino que se reserva la elaboración de la versión estándar. Por otro lado, es de código abierto porque el equipo de desarrollo espera en un futuro liberar el código fuente para que pueda experimentarse con extensiones y modificaciones del programa [12].

The image shows the Scratch website homepage. At the top, there is a navigation bar with the Scratch logo, links for 'Crear', 'Explorar', 'Comentar', 'Ayuda', a search bar, and 'Únete a Scratch' and 'Ingresar' buttons. Below the navigation bar, the main heading reads 'Crea historias, juegos y animaciones' and 'Comparte con gente de todo el mundo'. There are three cartoon characters: an orange cat with 'PRUÉBALO', a blue cat with 'VER EJEMPLOS', and a yellow character with 'UNIRSE A SCRATCH' and 'es gratis'. To the right, there is a preview of a Scratch script: 'when clicked' followed by a 'repeat 10' loop containing 'move 10 steps', 'change color by 25', 'play drum for 0.2 beats', and 'say Welcome to Scratch for 2 secs'. Below this, it says 'Una comunidad de aprendizaje creativo con 8.689.471 proyectos compartidos' and provides links for 'ACERCA DE SCRATCH', 'PARA DOCENTES', and 'PARA PADRES'. The 'Proyectos Destacados' section features five project thumbnails: 'How To Make A Quiz', 'Pi Collector', 'Pixel Town 2 v3.3', 'Monday Mews', and 'Typewriter text on...'. The 'Estudios Destacados' section features four study thumbnails: 'Scratch Writing Group', 'Super π Day!!!', 'Vector Art', and 'Just Like Real Life'.

Figura 3.2: Página web de Scratch

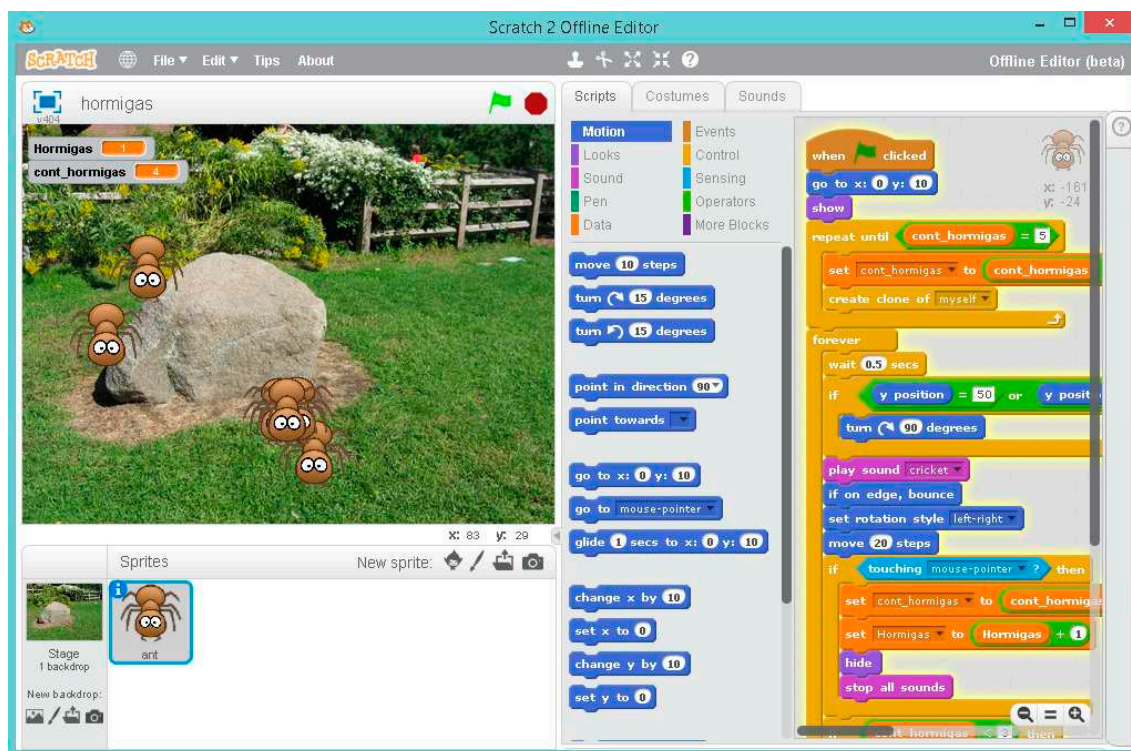


Figura 3.3: Entorno de Scratch 2.0 en su versión *offline*

Los precedentes de Scratch son el micromundo de LOGO, los *e-toys* de Squeak y LogoBlocks. El lenguaje utilizado por Scratch se basa, por un lado, en LOGO, en especial sus primitivas, presentando un entorno donde sus múltiples objetos pueden evolucionar e interactuar, y por otro lado, en Squeak (lenguaje en el que está desarrollado) de donde toma el modo de trabajo de arrastre de bloques hacia el área de programación en lugar de tener que escribir el código. Estos bloques, encajan entre ellos si componen una sintaxis correcta, ayudando al usuario a centrarse únicamente en la lógica de su algoritmo de programación sin tener que preocuparse por las reglas del lenguaje o la corrección lingüística [12].

Desde 2013 se encuentra operativa la segunda versión estable de Scratch, llamada Scratch 2.0, cuya interfaz se muestra en la Figura 3.3. La primera versión del programa (Scratch 1.0) fue desarrollada en el año 2006 como plataforma únicamente *online*, creando su versión descargable en enero de 2007. La última versión anterior a la actual fue la 1.4. Scratch, en su versión de escritorio, está disponible para Windows, Mac y Linux, se usa en más de 150 países y ha sido traducido a 40 idiomas. En el caso de España, está disponible en todas sus lenguas oficiales: castellano, catalán, euskera y gallego.

En marzo del año 2007 la página web de Scratch fue completamente rediseñada añadiendo el componente social, por lo que los usuarios empezaron a ser capaces de compartir sus proyectos y acceder a los de otros usuarios, también llamados *Scratchers*.

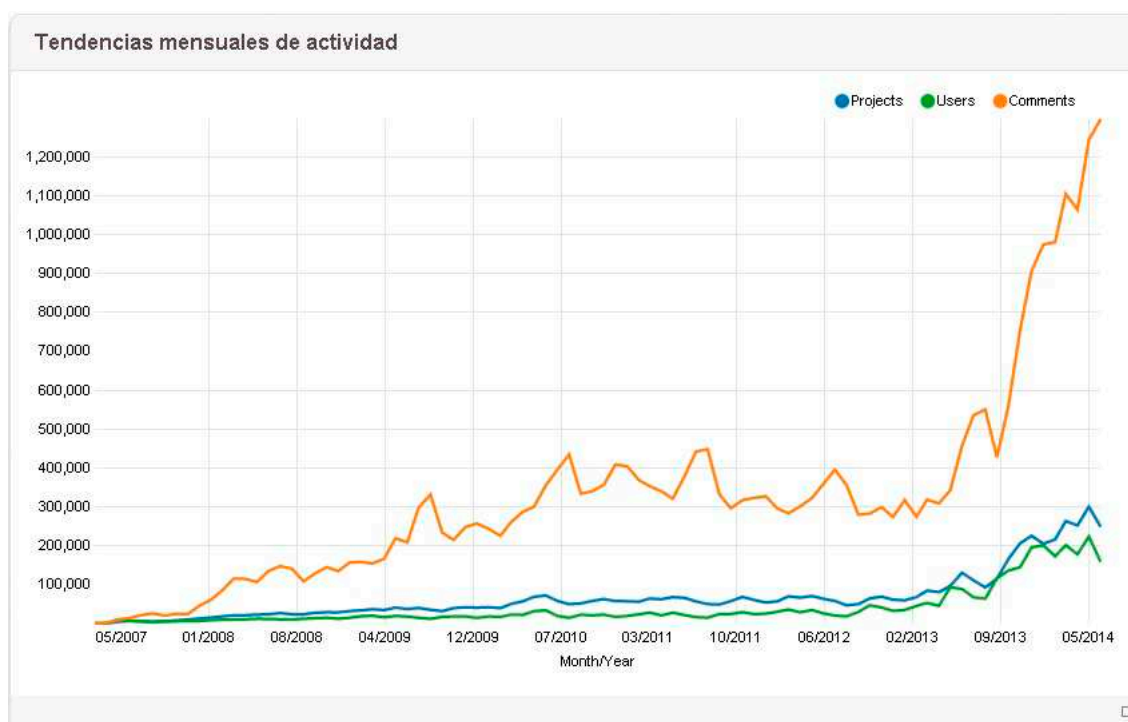


Figura 3.4: Tendencias mensuales de actividad en Scratch: usuarios, proyectos y comentarios en la página web

La comunidad en línea de Scratch posee dos partes diferenciadas, una dedicada a la comunidad de usuarios o *Scratchers* y accesible desde la página web de Scratch (<http://scratch.mit.edu>) y otra específica para educadores y docentes llamada ScratchEd (<http://scratched.media.mit.edu>) donde se resuelven cuestiones sobre Scratch, su aplicación a la docencia así como la posibilidad de comunicarse con otros profesores y de compartir recursos.

Actualmente² la página web de Scratch dispone de un total de 8.648.906 proyectos compartidos con un total de 5.915.725 usuarios registrados, cifras que aumentan con bastante frecuencia. La actividad en este último año del proyecto Scratch ha crecido de forma notable según las estadísticas ofrecidas por la web. Un ejemplo de ello es el número de proyectos nuevos compartidos por mes, siendo esta cifra de 97.242 proyectos en mayo de 2013 frente a los 300.368 de mayo de 2014, solo un año después, tal como se muestra en la gráfica de la Figura 3.4.

El proyecto Scratch se ha financiado por instituciones como la National Science Foundation, Fundación Intel, Microsoft, Fundación MacArthur, Fundación LEGO, Fundación Code-to-Learn, Google, Dell, Fastly, Inversoft y el consorcio de investigación del MIT Media Lab. También acepta donaciones de usuarios o personas que quieran ayudar al desarrollo de Scratch.

²Datos de la página web a fecha de marzo del año 2015.



Figura 3.5: Ejemplo de bloque repeat en forma de letra C

3.2.1. Finalidad del proyecto Scratch

Scratch tiene la finalidad educativa de enseñar a programar, habilidad importante en la sociedad actual. Al aprender a programar usando Scratch se desarrollan habilidades importantes para resolver problemas, diseñar proyectos y comunicar ideas.

El principal objetivo del proyecto Scratch es desarrollar una plataforma de acercamiento a la programación que resultase atractiva para la gente que no se haya planteado verse como programadores. Asimismo, busca ser sencillo para que cualquier persona pueda crear sus propios juegos, historias o animaciones y además pueda compartirlos con el resto del mundo.

Mediante la sencillez de Scratch y su entorno lúdico en 2D se intenta inculcar en los jóvenes y no tan jóvenes, una mayor comprensión de las nuevas tecnologías así como mejorar la valoración educativa que despiertan los algoritmos de programación. Las características principales de Scratch se resumen en:

- **Gran interacción.** El proceso de programación con Scratch se basa en la agrupación de bloques según su función. Así pues el *Scratcher* solo debe arrastrarlos al área de trabajo. Estos bloques pueden conectarse entre sí si la sintaxis es correcta, quedando desconectados aquellos en los que no lo sea. Las formas y colores de los bloques son diferentes según su función y requisitos. Así, por ejemplo, las estructuras de control tales como for o repeat toman la forma de la letra C para sugerir que deben introducirse más bloques dentro de ellas tal y como muestra la Figura 3.5.

Scratch está diseñado para ser altamente interactivo. Al hacer clic sobre una pila de bloques que se hayan programado, podemos ver inmediatamente el resultado que produce pudiendo incluso realizar cambios en la programación mientras está ejecutándose, haciendo fácil de esta forma la experimentación con nuevas ideas.

- **Más significativo.** La gente aprende mejor cuando siente que está involucrado en lo que está aprendiendo, esto es, si aprende mientras trabaja en

proyectos personales que le resultan significativos. Scratch está orientado principalmente a dos conceptos de diseño que asegura esa particularización:

1. *Diversidad*: al dar soporte para la creación de una gran gama de proyectos como animaciones, juegos, simulaciones o historias, se asegura que el rango de gente que pueda usarlo para hacer sus proyectos personales sea muy amplio.
2. *Personalización*: Scratch permite la fácil importación de audio o vídeo ya sea en forma de fotos, imágenes, grabaciones de sonido o música.

Estas prioridades han influenciado el aspecto de Scratch, como por ejemplo el hecho de que se muestren los gráficos en 2D en lugar de en 3D, ya que la importación de imágenes resulta más sencilla en un entorno 2D.

De esta forma se asegura que, mientras los *Scratchers* trabajan en sus propios proyectos que tienen un gran significado para ellos, se encuentran más predispuestos a aprender conceptos científicos o matemáticos que tengan que ver con el proyecto en el que están inmersos que si aprendieran esos mismos conceptos de la forma habitual, mediante una clase teórica.

- **Más social.** La aplicación de Scratch siempre ha ido desarrollándose paralelamente con el desarrollo de la página web. Uno de los aspectos importantes para Scratch es estar unido a una comunidad donde los usuarios puedan colaborar, criticar y apoyarse unos a otros, así como participar, modificar y expandir el trabajo de otros.

El concepto de compartir se incluye en la interfaz de Scratch mediante el menú «Compartir» en su versión web tal y como muestra la Figura 3.6, así como el icono situado arriba de la pantalla en su versión de escritorio como puede observarse en la Figura 3.7.

Al presionarlo, el proyecto se integra en a la página web de Scratch donde, una vez subido, cualquiera podrá ejecutarlo desde su navegador o bajarlo para la versión de escritorio del programa permitiendo, además, realimentación tal como comentarios o votos. Además el código del proyecto es completamente visible para así poder ser modificado o expandido por cualquier otro usuario. En los 27 meses siguientes al lanzamiento de Scratch, más de 500.000 proyectos habían sido compartidos en la página web [14]. Todos los proyectos compartidos en la web se encuentran bajo licencia *Creative Commons*³.

Alrededor de un 15% de los proyectos de la web de Scratch en noviembre del año 2009 eran ampliaciones o modificaciones de otros que ya habían sido subidos. Para evitar conflictos con los usuarios de los proyectos originales, la web añade en cada uno de los proyectos derivados de ellos, un

³Licencia de derechos de propiedad intelectual, se puede encontrar más información desde su página web (<http://es.creativecommons.org/>).

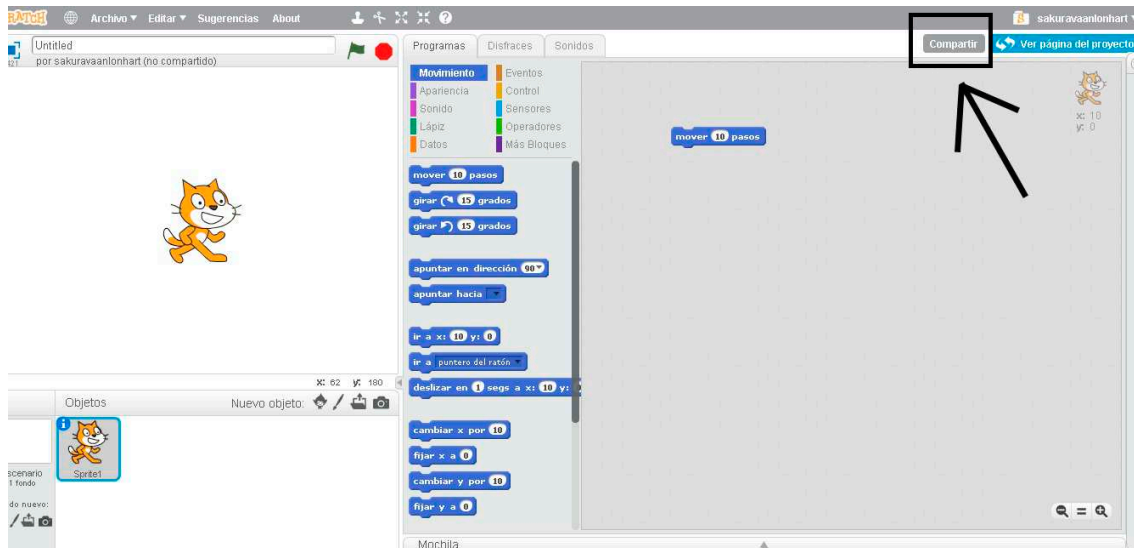


Figura 3.6: Compartición de proyectos en Scratch en entorno web

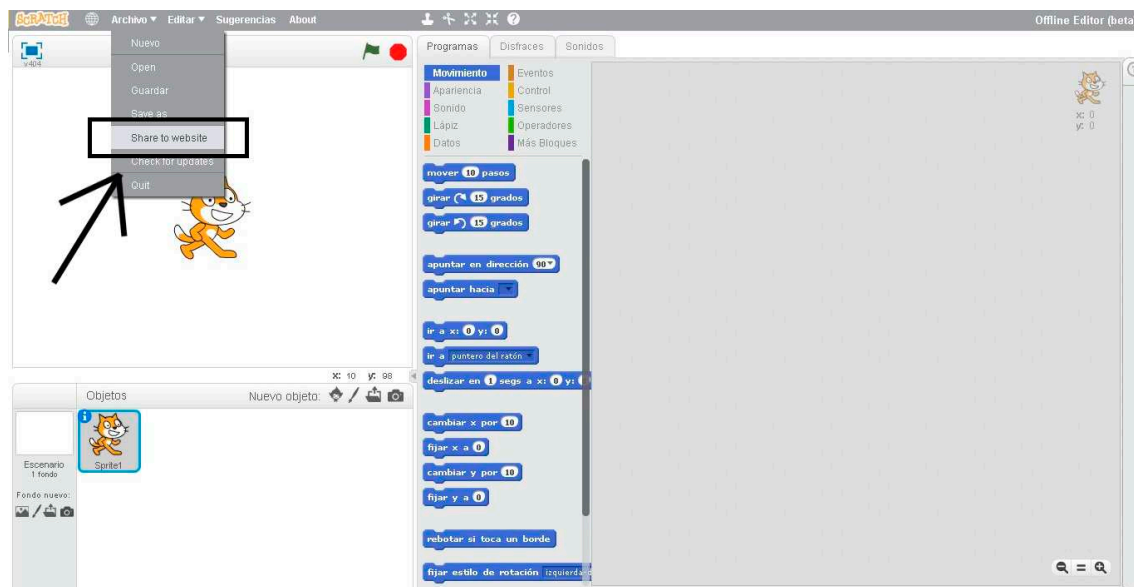


Figura 3.7: Compartición de proyectos en Scratch en entorno escritorio

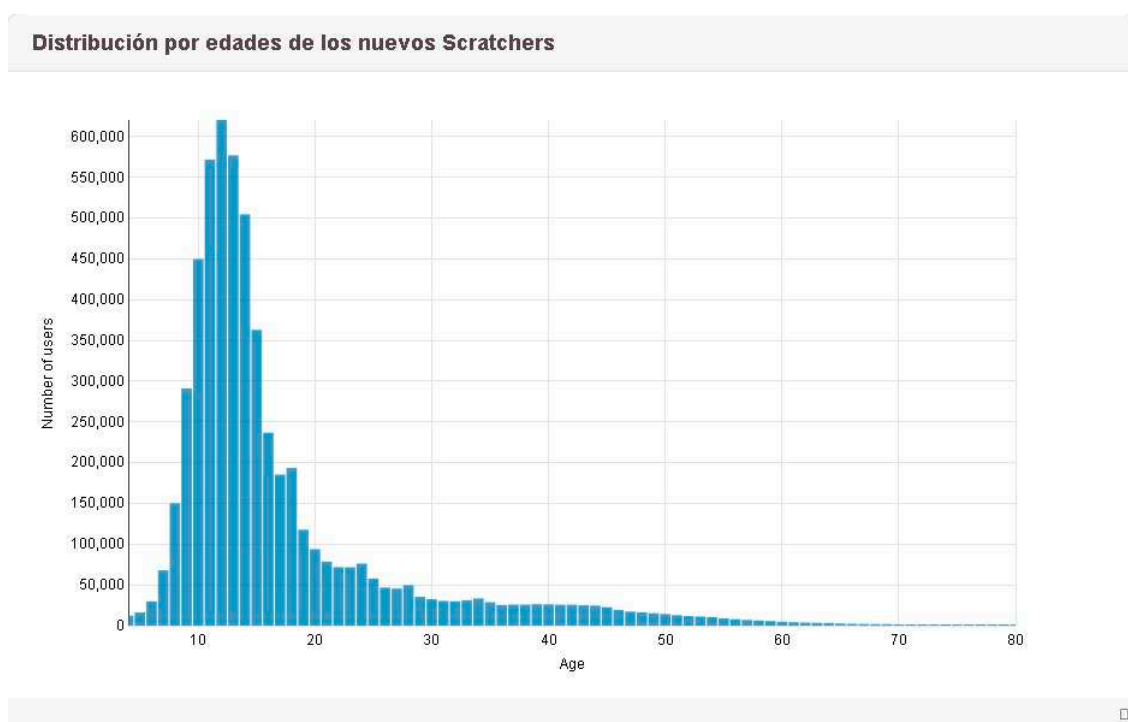


Figura 3.8: Número de usuarios de la aplicación por edades de Scratch

enlace hacia el proyecto original y de éste, enlaces a todos los proyectos que han hecho uso de él.

Otra importancia del componente social es el trabajo colaborativo que se da en la plataforma por la que muchos usuarios han creado «compañías» para trabajar juntos y aprender unos de otros. Para aumentar este trabajo colaborativo, se está poniendo especial interés en traducir Scratch a al mayor número de idiomas posibles para que así pueda extenderse de manera internacional. Un grupo voluntario de traductores se encarga de estas traducciones estando ya disponible en 40 idiomas. Los *Scratchers* de todo el mundo pueden ver proyectos de cualquier otra parte del planeta y leerlos en su propio idioma.

3.2.2. Usuarios de Scratch

Scratch está específicamente enfocado para jóvenes de edades entre 8 y 16 años aunque puede ser usado por personas de todas las edades. En la Figura 3.8 podemos observar la distribución de edades de los nuevos *Scratchers* que se dieron de alta en el mes de marzo de 2015 donde se observa que el mayor número de usuarios se encuentran en el rango de edad de 8 a 18 años.

Scratch no requiere ningún conocimiento específico en programación o en lenguajes de programación para utilizarlo, por eso es tan atractivo para su uso en las escuelas. La plataforma resulta apta para el uso en todos los niveles educativos

(desde la educación primaria hasta la universidad) y en todas las disciplinas (matemáticas, lengua, física, idiomas...).

3.2.3. Scratch y el pensamiento computacional

El concepto de *computational thinking* o pensamiento computacional se define como «los complejos procesos involucrados en formular problemas y las soluciones representadas de forma que pueden ser eficientemente llevadas a cabo por un agente de tratamiento de de información» [19] o, lo que es lo mismo, una habilidad para resolver problemas complicados de manera algorítmica que, además, ayuda a mejorar la eficiencia de los procesos. Este concepto, usado por primera vez por Seymour Papert en 1996, no solo es esencial para los ingenieros informáticos sino para cualquier persona en general ya que es aplicable a todas las disciplinas existentes [8].

Scratch ayuda al aprendizaje mediante el pensamiento computacional ya que está enfocado a actividades basadas en el diseño ofreciendo un gran abanico de herramientas para ayudar a resolver dichas actividades. Esto hace que los *Scratchers* se transformen en *computational thinkers* [13], personas que no solo son consumidoras de multimedia sino que son capaces de usarla para crear, diseñar y expresarse.

3.2.4. Scratch Day

El *Scratch Day* es un día dedicado a llevar a cabo una red global de eventos para fomentar en los niños la programación con Scratch. Se celebra desde el año 2009 y suele tener lugar en mayo. El *Scratch Day* del año 2015 tendrá lugar el 16 dicho mes. En el *Scratch Day* que se celebró el día 14 del año 2014, se organizaron 260 eventos en 56 países de todo el mundo tales como Argentina, Nigeria, España o Vietnam⁴ tal y como se muestra en el mapa de la Figura 3.9 extraído de la página web.

La finalidad de los eventos organizados este día es conseguir que los usuarios se reúnan y colaboren entre ellos para crear proyectos y compartir sus experiencias e ideas. Para aquellos que no tengan ningún evento cerca o no lo puedan atender, pueden utilizar el foro especial dedicado al *Scratch Day* para ver las imágenes subidas por otros usuarios y poder mantener su propio *Scratch Day* con la comunidad de *Scratchers*.

⁴Datos según la página web dedicada al *Scratch Day* (<http://day.scratch.mit.edu/>), año 2015



Figura 3.9: Localización de eventos del *Scratch Day* 2014

El *Scratch Day* 2014 en España

El último *Scratch Day* celebrado se llevó acabo de manera global el 14 de mayo del pasado año 2014. En España se organizaron un total de 16 eventos en ciudades como Sevilla, Valencia, Madrid, Palma de Mallorca o Barcelona, tal y como muestra el mapa de eventos de la página web de la Figura 3.10

3.2.5. Visión futura

En la actualidad, *The Lifelong Kindergarden*, desarrollador del grupo, sigue trabajando en actualizaciones de su última versión de Scratch (Scratch 2.0) ya sea por medio de traducciones del programa y de la página web para hacerlo un proyecto multilingüe como mediante la creación de *shariables* es decir, variables que podrán ser compartidas vía web [12].

Este grupo también está centrado en expandir su proyecto hacia plataformas móviles como pueden ser tabletas o teléfonos inteligentes. En concreto existe ya una versión Alpha de Scratch para smartphones Android desarrollada en J2ME y llamada *Android Scratch Player* tal y como muestra la Figura 3.11. Sin embargo aún presenta algunos problemas de implementación tales como la incompatibilidad de muchos de los bloques usados en Scratch o el cierre inesperado de la aplicación al rotar la pantalla del dispositivo.

Dentro del ámbito de las tabletas, se ha desarrollado una plataforma llamada ScratchJr para iPads que se encuentra accesible para descarga desde la web <http://www.scratchjr.org/>. Esta variante de Scratch está centrada en niños pequeños de 5 a 7 años y muestra una interfaz mucho más simple que la de Scratch para



Figura 3.10: Eventos del *Scratch Day 2014* en España

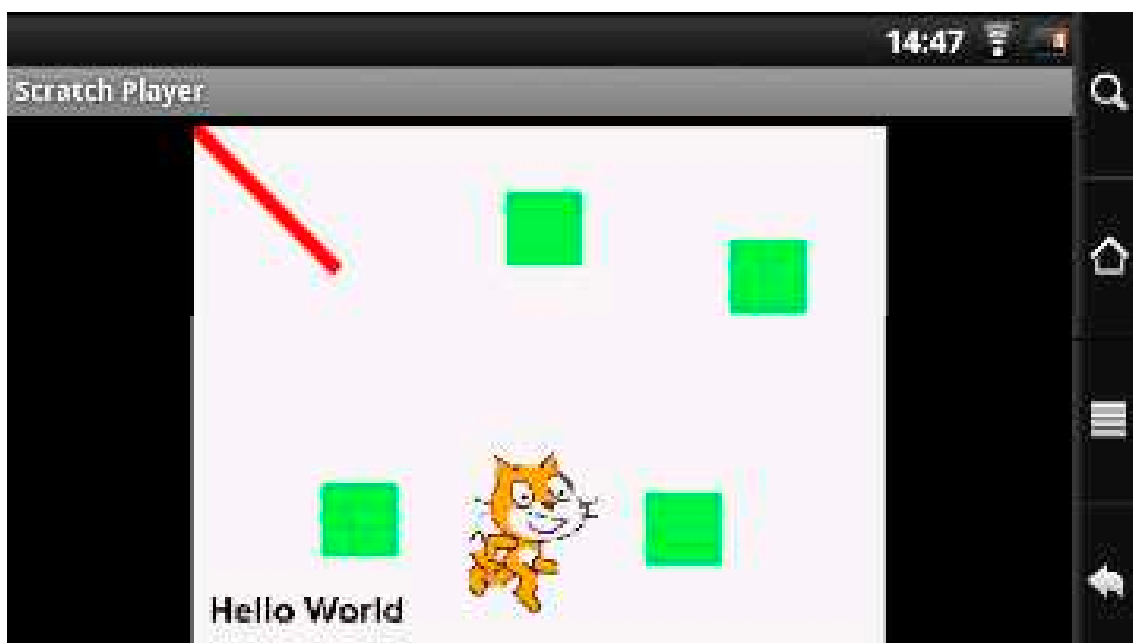


Figura 3.11: Interfaz del *Android Scratch Player*



Figura 3.12: Interfaz del *Scratch Jr*

permitir a los más pequeños que puedan ya aprender a programar y a expresarse mediante su creatividad (véase la Figura 3.12).

3.2.6. MIT, lugar de origen

El MIT *Massachusetts Institute of Technology* es una universidad privada situada en Cambridge, Estados Unidos formada por un total de 32 departamentos académicos con un fuerte énfasis en la investigación, la ingeniería y la educación tecnológica. Cuenta con 76 premios Nobel entre su personal haciéndola reconocida a nivel mundial y altamente selectiva en la admisión de nuevos miembros.

The Lifelong Kindergarten

Como hemos mencionado, Scratch está desarrollado por *The LifeLong Kindergarten*, un grupo de investigación liderado por Mitch Resnick (véase la Figura 3.13) situado en el MIT Media Lab, departamento de investigación centrado en la creatividad a la hora de abordar sus proyectos.



Figura 3.13: Mitch Resnick, líder de *The Lifelong Kindergarten*

The Lifelong Kindergarten encuentra que habilidades como la creatividad o el diseño son primordiales a la hora de desarrollar cualquier actividad, pero creen que el mundo actual no permite a los más jóvenes desarrollar estas habilidades ya que en las escuelas les enseñan patrones y hechos, sin darles demasiada oportunidad de diseñar, crear o expresarse por ellos mismos. Así pues, centran todos sus proyectos en hacer que los niños y jóvenes actuales se conviertan en pensadores creativos [13] como lo son los niños en su período de guardería (*kindergarten*) en la que no tienen barreras a la hora de desarrollar su creatividad.

Su proyecto más conocido es Scratch, el cual es financiado por instituciones como Microsoft, Nokia o la Fundación Nacional de Ciencia de Estados Unidos, aunque han estado o están involucrados en otros proyectos como los que se exponen a continuación.

- **Computer Clubhouse.** Este término hace referencia a un grupo de actividades extraescolares relacionadas con la programación que se desarrollan en ámbitos de exclusión social. Los miembros trabajan en proyectos (animaciones, historias interactivas, vídeos musicales...) basados en sus intereses mediante el uso de nuevas tecnologías y con el apoyo de mentores expertos tal y como se muestra en la Figura 3.14.
- **Build in Progress.** Se trata de una plataforma de compartición *online* que ofrece las herramientas necesarias para que la gente pueda subir a la red la documentación y el diseño de los proyectos que tienen en desarrollo. A su vez permite que otros usuarios puedan ver los proyectos de los demás para que sirvan de ayuda en los suyos propios e incluso permite partir de proyectos existentes de otros usuarios para modificarlos o ampliarlos.

Build in Progress se encuentra vía *online* como puede observarse en la Figura 3.15 a través de la web <http://buildinprogress.media.mit.edu/> y mediante una aplicación para dispositivos Android.



Figura 3.14: Miembros del *Computer Clubhouse* en México

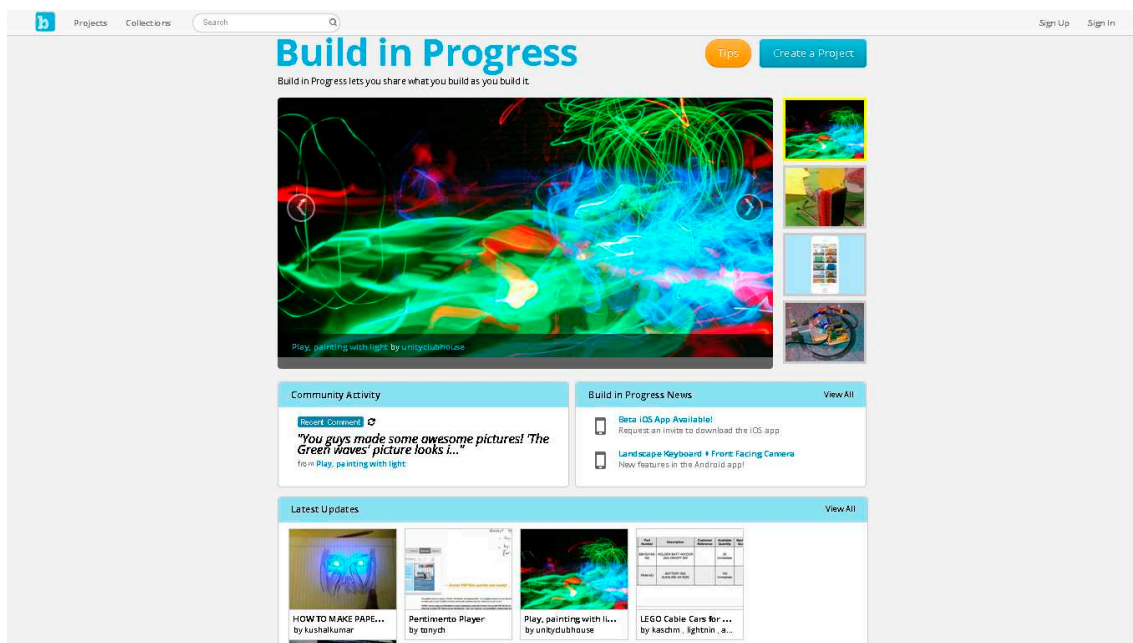


Figura 3.15: Interfaz web de *Build in Progress*

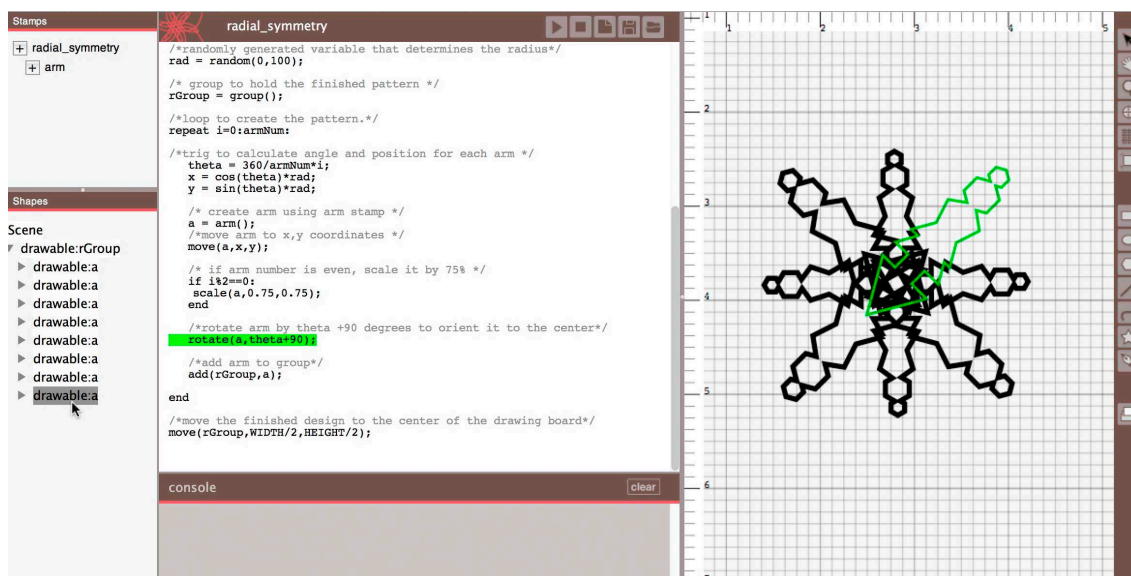


Figura 3.16: Interfaz de *DressCode*

- **DressCode.** Herramienta de dibujo y fabricación de diseños gráficos mediante la combinación de un editor de dibujo con uno de programación. Cualquier figura creada mediante el editor de dibujo se traducirá en código en el editor de programación tal como muestra la Figura 3.16, permitiendo al usuario modificar el dibujo de una forma más precisa mediante código. Los diseños realizados mediante el programa son compatibles con máquinas de impresión digital, lo que facilita la creación de objetos físicos con los diseños implementados.
- **Makey Makey.** Consiste en un dispositivo que puede convertir cualquier objeto de nuestra vida diaria en una interfaz para el ordenador. Se conecta vía USB y no requiere ningún tipo de programa específico ni habilidad en programación. Se compone de una placa central con los diferentes controles que puede emular y unas pinzas para conectar dicha placa con el objeto que queramos. Puede usarse desde plastilina (véase la Figura 3.17) hasta cubos de agua o cereales que actuarán como el ratón del ordenador o las teclas del teclado.

3.3 Entorno de programación en Scratch

En el siguiente apartado se dispondrá a hacer una explicación de los componentes y el funcionamiento de Scratch, analizando sus elementos y controles.

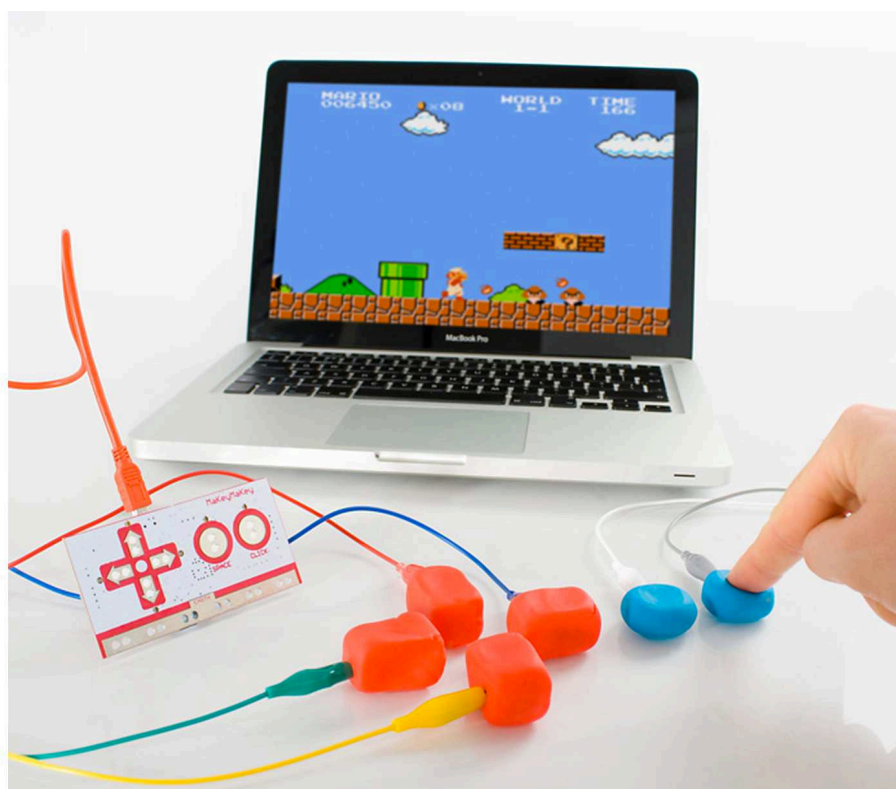


Figura 3.17: Makey Makey usando pastilina a modo de *gamepad*

3.3.1. Elementos básicos

Los proyectos en Scratch se componen de *sprites* u objetos. Éstos pueden ser desde personajes del proyecto (animales, cosas, personas...) hasta el propio fondo del mismo, llamado escenario. Cada uno de estos objetos puede tener uno o más disfraces, es decir, diferentes aspectos que puede tomar un mismo objeto (por ejemplo hacer que un personaje ande) tal y como muestra la Figura 3.18.

Las instrucciones que se le pueden asociar a cada objeto se agrupan en bloques según su funcionalidad. Cada una de estas instrucciones encajan unas con otras formando programas o *scripts*. Al hacer doble clic sobre uno de estos programas, se ejecutan las instrucciones en orden de arriba a abajo.

3.3.2. Interfaz de Scratch

La interfaz de Scratch se puede dividir en diversos elementos que pasan a exponerse a continuación.

- **Escenario.** El escenario es donde se ve el resultado de la programación de los objetos que se mueven e interactúan entre ellos y con el fondo o *background*, tal como muestra la Figura 3.19 en la división marcada con un 1. El

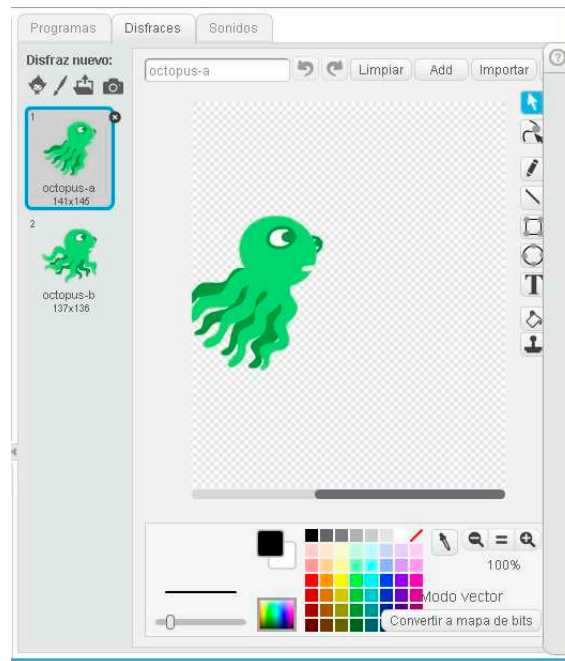


Figura 3.18: Diferentes disfraces que puede adoptar un objeto

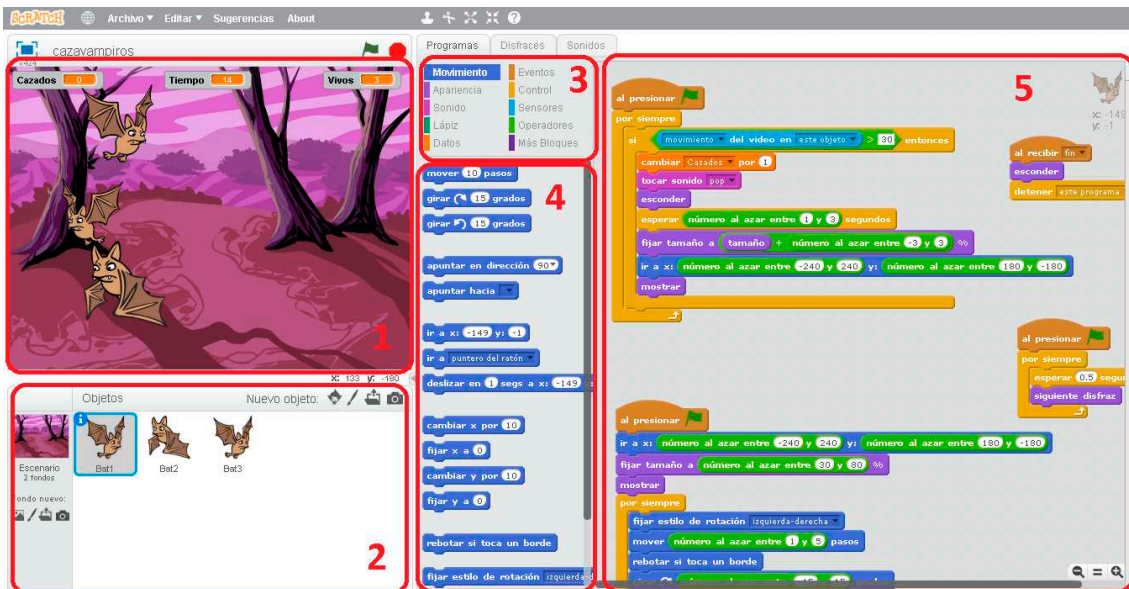


Figura 3.19: Interfaz de Scratch dividida en sus diferentes elementos

escenario tiene 480 píxeles de ancho y 360 de alto y se divide en un plano cartesiano xy . El centro del escenario corresponde a las coordenadas $x=0$; $y=0$.

- **Gestión de objetos.** Dentro de la gestión de objetos (véase la Figura 3.19, división marcada con un 2) encontramos, por un lado, la lista de objetos donde muestra las miniaturas de todos los objetos incluidos en el proyecto, y por otro, las herramientas que Scratch dispone para ellos.
- **Paleta de bloques.** En la paleta de bloques se encuentran distribuidas todas las instrucciones en apartados según su función (véase la Figura 3.18, división marcada con un 3). Al pulsar una agrupación, se nos mostrarán las instrucciones que contiene (véase la Figura 3.18, división marcada con un 4).
- **Área de programa.** El área de programa es donde se sitúan todas las instrucciones que queremos que realice cada uno de los objetos como se muestra en la Figura 3.19, división marcada con un 5.

Al arrastrar un bloque desde la paleta de bloques al área de programa, una iluminación blanca indicará las posibles ubicaciones correctas para dicho bloque dentro del programa. Cada bloque se enganchará con otro si la sintaxis del programa es correcta. Pueden insertarse bloques en medio de una pila ya formada de instrucciones. Si dos bloques son incompatibles no podrán conectarse.

Para mover una pila de instrucciones dentro del área de programa, bastará con hacer clic en la primera instrucción que la forme. Una pila de instrucciones puede duplicarse e incluso moverse de objeto, arrastrándola hacia el objeto deseado dentro del área de gestión de objetos.

3.3.3. Multimedia

Scratch permite la edición de contenido multimedia de imágenes o sonidos tales como objetos o disfraces o música que quiera incorporarse a los proyectos o cualquier otro efecto sonoro.

- **Disfraces.** Para poder acceder a la edición de disfraces, basta con hacer clic en la pestaña de disfraces situada arriba del área de programa. Scratch dispone de una galería de objetos en la que algunos de ellos tienen más de un disfraz. Asimismo ofrece la oportunidad de cargar nuestros propios disfraces desde el disco duro, y a su vez, mediante el editor de imagen que tiene el programa incorporado, podemos modificar imágenes que ya se encuentren en la galería de Scratch o crear nuestros propios nuevos disfraces tal como muestra la Figura 3.20.

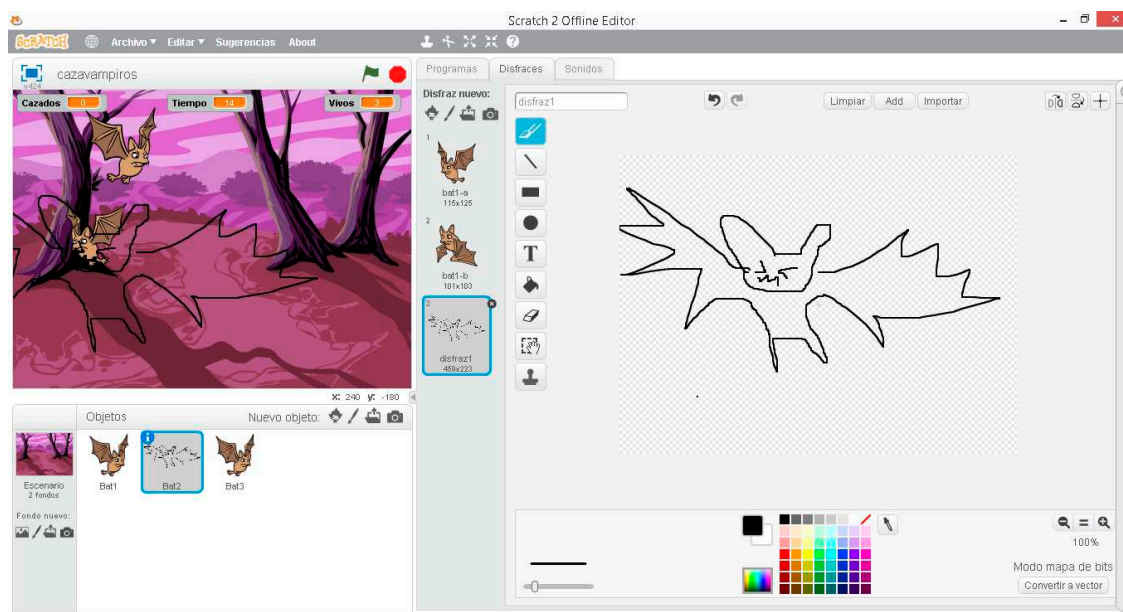


Figura 3.20: Editor de imágenes de Scratch

Es posible convertir un disfraz en un nuevo objeto o exportar una copia del disfraz como un archivo separado. Scratch reconoce muchos formatos de imagen como JPG, BMP, PNG o GIF (incluyendo GIF animados).

- **Sonidos.** Los proyectos en Scratch admiten sonidos y música. Para acceder a las herramientas de sonido basta con seleccionar, desde el área de gestión de objetos, el objeto en el cual queremos incorporar audio para después hacer clic en la pestaña Sonido que se encuentra encima del área de programación y que nos mostrará la interfaz para sonidos que aparece en la Figura 3.21.

Al igual que con los disfraces, Scratch permite incorporar los sonidos de tres formas diferentes. O cargándolos desde la galería de sonidos predefinida del programa, o grabando nuevos sonidos o bien importándolos desde una ubicación del disco duro. Cada uno de los sonidos deberá incorporarse mediante las instrucciones correspondientes al programa, y para cada objeto deberemos cargar los archivos de audio que necesitemos.

3.3.4. Menús y barras de herramientas

Scratch posee varios menús y barras de herramientas que pueden ser generales o específicos según la función que estemos realizando, tal como se enumeran a continuación.

- **Menús y barras de herramientas generales**
 - **Menú general.** El menú general podemos encontrarlo situado arriba a la izquierda del programa. Este menú sirve tanto para la gestión de

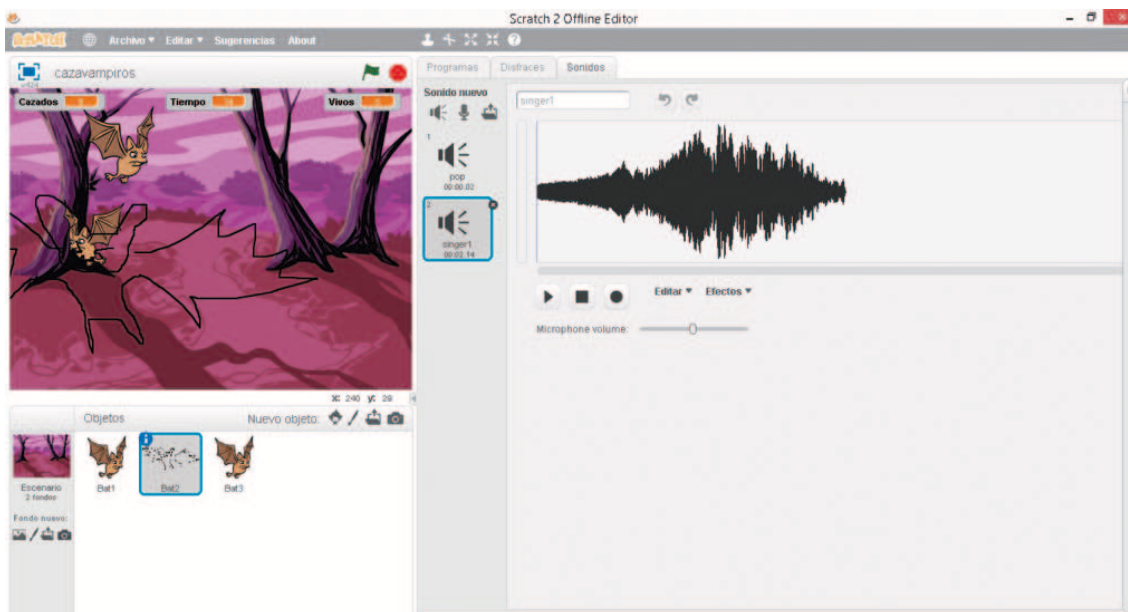


Figura 3.21: Herramienta de sonido de Scratch



Figura 3.22: Menú general

los archivos de Scratch, de vía de acceso a la web donde se aloja o para la configuración del idioma del mismo tal como se muestra en la Figura 3.22.

- **Barra de herramientas general.** Se encuentra a la derecha del menú general. Esta barra nos permite agrandar o empequeñecer objetos, borrarlos o clonarlos así como acceder a la ayuda del programa tal como se muestra en la Figura 3.23.
- **Herramientas de área de gestión de objetos**
 - **Herramientas de objeto.** La Figura 3.24 muestra las herramientas de objeto que permiten cargar objetos desde diferentes ubicaciones ya sea desde la galería que Scratch lleva de forma predefinida, desde nuestro ordenador o bien dibujándolo el mismo usuario desde el editor de imagen.



Figura 3.23: Barra de herramientas general



Figura 3.24: Herramientas de objeto



Figura 3.25: Herramientas de fondo

- **Herramientas de fondo.** Al igual que las herramientas de objeto, las de fondo permiten elegir el origen de los fondos que se van a utilizar como escenarios en el proyecto, desde predefinidos hasta creados por el usuario (véase la Figura 3.25).
- **Herramientas de edición de imágenes.** Scratch posee un editor de imágenes parecido a programas como Paint de Windows o Gimp de Linux y comparte con éstos muchas de sus herramientas. El editor de Scratch puede ser usado tanto para crear nuevos objetos o fondos como para editar nuevos disfraces. Además, permite editar imágenes vectoriales (no pierden resolución ni se pixelan al aumentarlas), o imágenes de mapa de bits (imágenes cuya resolución depende del número de píxeles que contengan). En la Figura 3.26 podemos ver el aspecto y las diferentes herramientas que incluye el editor de Scratch.
- **Herramientas de edición de sonido.** Al igual que con las imágenes, Scratch permite personalizar los sonidos utilizados en el programa, editando los existentes o grabando de nuevos tal como muestra la Figura 3.27.

3.3.5. Manipulación de datos en Scratch

Scratch permite hacer operaciones con datos ya sean números (enteros o reales) y caracteres. Además permite la creación de listas de elementos.

- **Variables.** Las variables son las unidades empleadas en programación para almacenar un dato. Este concepto resulta muy útil a la hora de tomar decisiones a partir de hechos o para poder crear interacción con el usuario. Scratch dispone de la opción de creación de variables así como instrucciones para manejarlas dentro del bloque «Datos» según se muestra en la Figura 3.28.

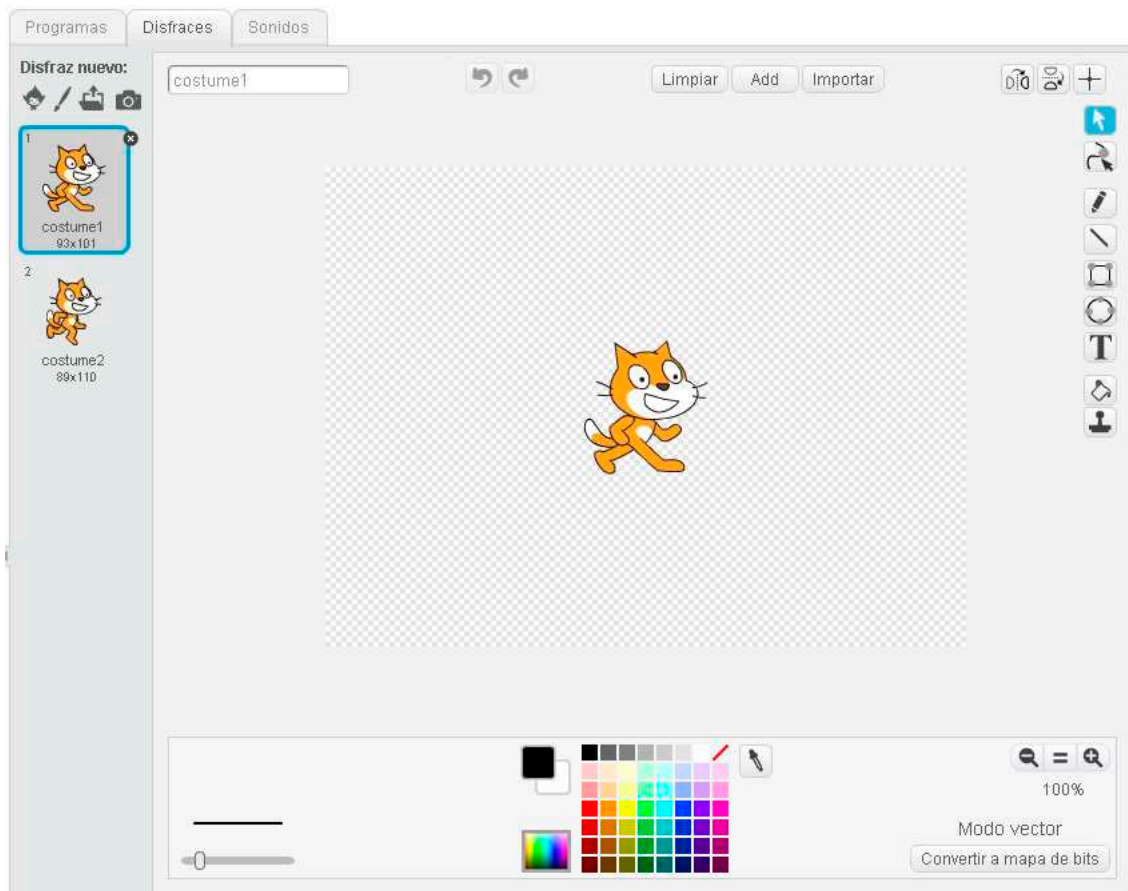


Figura 3.26: Herramientas de edición de imagen

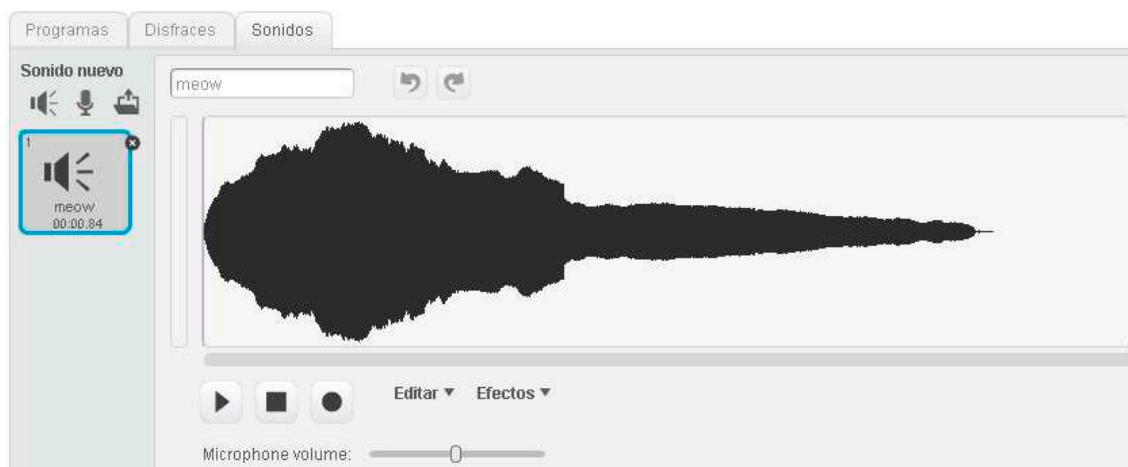


Figura 3.27: Herramientas de edición de sonido



Figura 3.28: Instrucciones de manejo de variables



Figura 3.29: Instrucciones de manejo de listas

- **Listas.** Cuando en un programa se necesita administrar muchas variables u ordenarlas de una forma específica siguiendo un algoritmo determinado, es esencial utilizar las listas. Scratch también incluye la opción de crear listas desde el bloque de «Datos» así como instrucciones para tratarlas, tal como muestra la Figura 3.29.

3.4 Scratch en la literatura

Se puede encontrar una amplia bibliografía sobre Scratch en su página web⁵ tanto para el desarrollo de proyectos mediante esta plataforma en su versión ac-

⁵Página accesible en http://wiki.scratch.mit.edu/wiki/Scratch_Books, año 2015.



Figura 3.30: Portada de *Programación Scratch para niños*

tual (la 2.0) como en la anterior (la 1.4), estando disponibles además en diversos idiomas.

Dentro de nuestras lenguas oficiales, solo es posible encontrar material bibliográfico en castellano, aunque aún no existe una gran variedad de títulos. Aún así es posible encontrar ejemplos como *Programación Scratch para niños* [17] o *Scratch para niños... y no tan niños* [3], ambos disponibles únicamente en versión Kindle. En el primero (véase la Figura 3.30), se enseñan conceptos de programación mediante unidades teóricas y ejemplos mientras que en el segundo (cuya portada se muestra en la Figura 3.31), se plantean proyectos en cada uno de sus capítulos, que el lector deberá resolver aprendiendo de esta forma, los diferentes conceptos de programación que expone el libro.

La mayoría del material bibliográfico de Scratch está enfocado a la gente más joven que desconoce la programación, intentando lograr que se inicien en la ma-



Figura 3.31: Portada de *Scratch para niños... y no tan niños*



Figura 3.32: Portada de *Super Scratch Programming Adventure!*

teria mediante el diseño de proyectos relacionados con el arte, la ciencia o juegos, aunque también existen libros avanzados. Algún ejemplo podría ser el título *Super Scratch Programming Adventure!* [18] (véase la Figura 3.32), o el libro *Scratch 2.0 Game Development Hotshot* [1] cuya portada se muestra en la Figura 3.33. Ambos libros están enfocados a un sector más adelantado en la programación y ofrecen ideas y ejemplos para la creación de videojuegos mediante Scratch.

Además, Scratch posee detalladas guías de referencia, tanto de su versión 2.0 como de la anterior 1.4. disponibles en diversos idiomas. Dentro de las guías que podemos encontrar en nuestros idiomas oficiales, podríamos destacar la versión en catalán para Scratch 2.0, que puede encontrarse *online* dentro de la página web del proyecto ScratchCatalà⁶, y la guía de referencia en castellano [6] que está disponible tanto para la versión 2.0 como la 1.4.

En los siguientes analizaremos con profundidad dos de los libros que podemos encontrar en la bibliografía y, que debido a su nivel de complejidad, son recomendables para usuarios que quieran iniciarse en la programación.

⁶Página web accesible en <http://scratchcatala.com>

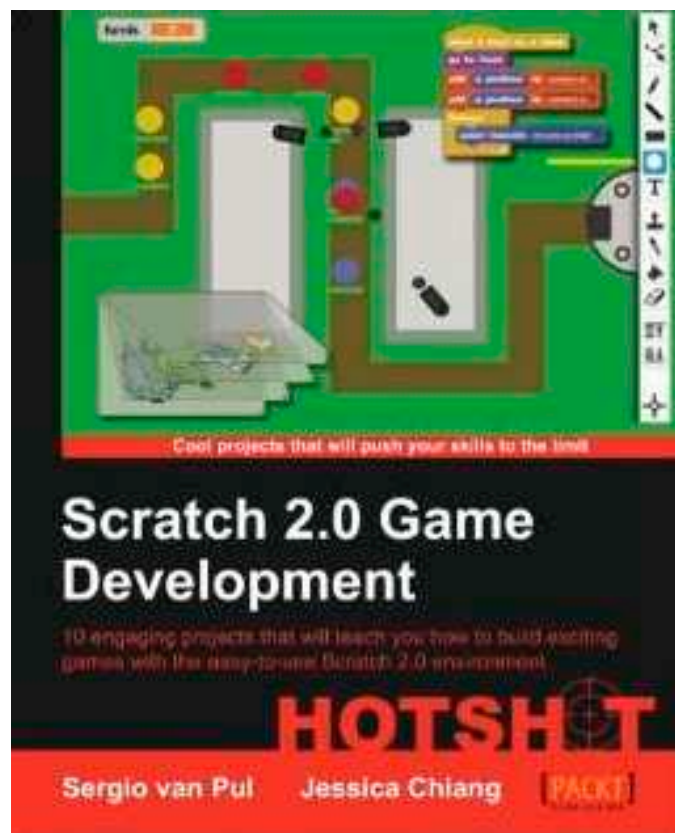


Figura 3.33: Portada de *Scratch 2.0 Game Development Hotshot*

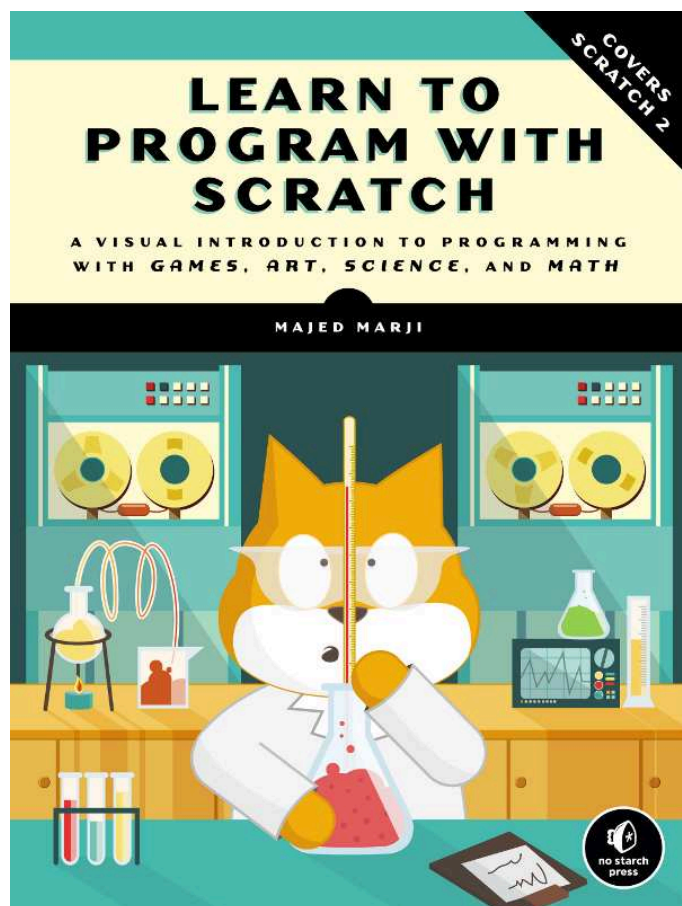


Figura 3.34: Portada de *Learn to program with Scratch: a visual introduction to programming with games, art, science, and math*

3.4.1. *Learn to program with Scratch*

Este libro escrito por Majed Marji que tiene por título completo *Learn to program with Scratch: a visual introduction to programming with games, art, science, and math* [10] está orientado tanto para aquellos que quieran aprender Scratch como aquellos que quieran enseñarlo. Por una parte puede ser utilizado por estudiantes de educación primaria o secundaria, así como universitarios que no posean base de programación, o para autoestudio, ya que el libro asume que no existe ningún conocimiento en esta materia antes de su lectura. Por otra parte, puede ser útil para profesores que necesiten ejemplos sobre algún concepto en particular sobre la programación o como apoyo dentro de la asignatura que impartan.

Todos los ejemplos descritos en este título están implementados exclusivamente para la versión 2.0 de Scratch tal y como indica su portada (véase Figura 3.34).

La estructura que sigue el libro es por capítulos. Cada uno de ellos se centra en un concepto concreto sobre programación y lo aborda mediante ejercicios prác-

ticos y actividades. Todas las unidades temáticas se estructuran en un resumen inicial sobre el tema a tratar y varios ejemplos en forma de proyectos que el lector debe construir. Destaca la ayuda teórica que el libro ofrece, ya sea mediante el texto que refuerza con imágenes, las notas o los párrafos destacados que usa para darle más pistas al lector sobre la solución de la actividad planteada. Además, los programas propuestos en cada una de las unidades didácticas así como recursos adicionales para reforzarlas pueden encontrarse de manera *online* desde la web del libro⁷.

Este título en concreto se divide en diez capítulos. Nueve son unidades temáticas con respecto a la programación y sus procedimientos mientras que el último se centra en la importancia de compartir y el componente social de Scratch. Los capítulos que podemos encontrar pasan a enumerarse a continuación.

1. *Getting started*. Esta es la unidad inicial del libro. En ella se explora el entorno de trabajo de Scratch, explicando su estructura y órdenes básicas para acabar proponiendo al lector la creación de su primer juego mediante el uso de esta plataforma.
2. *Motion and drawing*. En este capítulo el libro se centra en profundizar algo más en las herramientas de programación de Scratch, en este caso las que se utilizan para la creación de disfraces y las órdenes de lápiz. Mediante este tipo de herramientas el usuario puede incorporar animación a sus proyectos o hacer que Scratch dibuje por él. Durante toda la unidad se proponen ejemplos para ilustrar estos conceptos.
3. *Looks and sound*. Partiendo de la base vista en el capítulo anterior, en este sigue tratando el concepto de animación en Scratch y añade información sobre las herramientas de sonido que el programa incluye, cómo utilizarlas y cuáles son las instrucciones que las implementan. Durante todo el capítulo se exponen varios ejemplos para que el lector trabaje con lo que va aprendiendo.
4. *Procedures*. En esta unidad se explica el concepto de «divide y vencerás» en programación, es decir, eliminar la idea de crear un solo código que funcione como un gran bloque y, en su lugar, crear diferentes procesos que hagan pequeñas tareas más sencillas por separado y que en conjunto logren alcanzar la solución deseada. Para ello y mediante ejemplos, el libro introduce conceptos disponibles en Scratch como son el uso de mensajes, la herramienta de creación de bloques o la programación estructurada.
5. *Variables*. En este capítulo se nos introduce el concepto de variable. Mediante los ejemplos que le propone al lector, explica qué es una variable así como las instrucciones de Scratch que las manejan, el tipo de datos que

⁷Recursos disponibles para descarga en la web <http://nonstarch.com/learnscratch>, año 2015.

soporta el programa y cómo recibir información externa del usuario para poder crear programas interactivos.

6. *Making decisions*. Para escribir programas que comparen valores, evalúen expresiones lógicas o tomen decisiones basadas en resultados, Scratch posee estructuras condicionales tales como *si/sino* que ayudan a la resolución de problemas básicos. Mediante esta unidad se explican estas estructuras, se enseña a construir expresiones lógicas que se evaluarán en esas condiciones así como a manejar el flujo de control en este tipo de instrucciones.
7. *Repetition: a deeper exploration of loops*. En esta unidad se cubre de forma detallada el concepto de repetición y bucle dentro de la programación. Para ello, mediante el uso de actividades planteadas al lector, explica las estructuras que Scratch posee con relación a las estructuras de repetición, diferencia los bucles según sean por contador o por eventos, e introduce el concepto de recursión.
8. *String processing*. Scratch posee una amplia gama de herramientas e instrucciones para almacenar cadenas de texto y manejarlas. En este capítulo se explica cómo se almacena este tipo de estructuras, qué instrucciones se pueden usar para procesarlas así como técnicas de proceso de cadenas de textos o programas en los que puede resultar útiles usarlas.
9. *Lists*. Además de las variables con las que Scratch permite al usuario almacenar un dato, también permite el uso de listas por el cual es posible almacenar varios valores. En este capítulo, el libro explica el concepto de listas y su uso, cómo deben crearse, manejarse y acceder a ellas, en qué tipo de programas pueden ser útiles e incluye un apartado dedicado a algoritmos de búsqueda y clasificación básicos como el algoritmo de búsqueda lineal o el de ordenación de burbuja (*Bubble sort*).
10. *Sharing and collaboration*. En este último capítulo el libro remarca el componente social de Scratch por el cual cualquier usuario puede subir a internet sus proyectos, reutilizar los que ya hayan en la web hecho por otros usuarios y compartir su experiencia y conocimiento con el resto de personas.

3.4.2. Scratch Programming

Este título escrito por Sean McManus[16], está orientado para enseñar a los jóvenes entre primaria, secundaria, bachiller y primeros cursos universitarios las bases de la programación partiendo de cero y mediante el uso de la plataforma Scratch al igual que el título explicado anteriormente. Aunque a diferencia del anterior, en este todos los proyectos están adecuados tanto para la versión 2.0 del programa como la 1.4 (véase la Figura 3.35) indicando, si las hay, las diferencias entre ambas versiones a la hora de crear los diferentes programas que se utilizan.

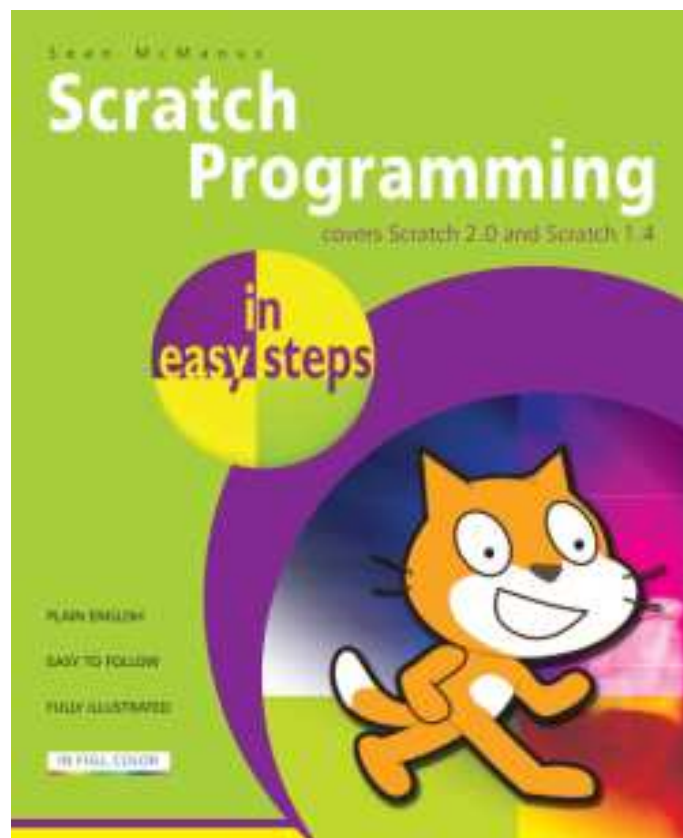


Figura 3.35: Portada de *Scratch Programming*

El libro se organiza en once capítulos. A diferencia del libro anterior, en el cual cada unidad se centraba en un concepto concreto de la programación, en éste cada capítulo se centra en la creación de un juego, introduciendo en cada uno los conceptos teóricos que se necesiten. Aún así el libro recomienda que se siga el orden de los capítulos sin alteraciones puesto que la dificultad de los proyectos se va incrementando y los conceptos teóricos parten de lo visto anteriormente. Cada unidad se estructura en los pasos a seguir para crear el proyecto elegido, agrupándolos según el concepto teórico que traten. A su vez incluyen texto con imágenes explicatorias, consejos y información adicional o a tener en cuenta. Los contenidos específicos de cada unidad se explicarán a continuación.

1. ***Introducing Scratch***. Este primer capítulo sirve como introducción a la plataforma Scratch, pasando a describir sus versiones, cómo crear una cuenta en su plataforma web así como ofreciendo una guía de instalación para el ordenador de bajo coste Raspberry Pi. La última parte del capítulo la dedica a explicar el entorno de trabajo, los diferentes bloques que se pueden encontrar y los comandos básicos como crear un nuevo proyecto, guardar un proyecto activo o compartir un proyecto finalizado.
2. ***Drawing with Scratch***. Esta unidad aún no plantea la creación de un proyecto sino que explica algunos conceptos relativos a la animación mediante Scratch, las instrucciones de dibujo que posee o el manejo de las coordenadas dentro del entorno.
3. ***Spiral Rider***. Este juego muestra un laberinto en pantalla y a dos personajes en cada uno de los extremos. La finalidad es que el personaje manejado por el usuario llegue hasta el otro sin tocar ni una vez ninguna de las paredes del laberinto. Para este proyecto se incluye la creación de variables, la programación de los comandos de movimiento o el uso de animaciones y objetos entre otros.
4. ***Super Dodgeball***. Este capítulo nos ayuda a crear un juego que va lanzando pelotas de forma aleatoria por el escenario. El jugador tiene que ir esquivándolas para que no le vaya descontando vidas y que aparezca la pantalla de *Game Over*. También aparecerán de vez en cuando helados por la pantalla para aumentar la vida del jugador si llega a tocarlos. Para poder realizar este proyecto el libro explica el concepto de variables, su creación y manejo, el uso de coordenadas para el movimiento del personaje principal, las funciones aleatorias así como las herramientas de edición de imágenes de Scratch.
5. ***Cosmic Chorus***. En este proyecto el usuario controlará mediante el teclado un coro de marcianos. Las técnicas de Scratch que se emplean y se detallan en el capítulo son la de edición y creación de audio (con explicaciones diferentes para la versión 2.0 y la 1.4), la animación de un objeto con múltiples

disfraces, la clonación de objetos, la incorporación de una pantalla de inicio o el paso de mensajes.

6. *Quiz Break*. En esta unidad se nos propone la creación de un juego de tipo test que le pregunte al jugador sencillas preguntas matemáticas que tendrá que responder dentro de un tiempo límite. Si se responden todas las preguntas correctamente, el juego nos recompensa con un baile efectuado por el personaje principal. El libro introduce de esta forma la interacción entre usuario y programa, la forma en la que comparar lo introducido por el jugador con la respuesta correcta, la unión de sentencias de texto con números o variables, la creación de una variable que lleve la cuenta del tiempo o el uso de instrucciones operacionales para crear cálculos.
7. *Hangman*. El siguiente proyecto que se encuentra el lector es el de la creación del juego del ahorcado. En este juego el usuario debe adivinar una palabra misteriosa que se le muestra por pantalla mediante la introducción de letras. Por cada letra incorrecta que introduzca, se irá dibujando la figura de un ahorcado. Si el ahorcado se dibuja totalmente, el jugador pierde el juego. Para este proyecto, el libro enseña al lector el concepto de lista y las instrucciones que pueden manejarlas en Scratch, el paso de mensajes para crear programas que sean más sencillos o a añadir efectos gráficos en los objetos mediante las instrucciones que nos ofrece el entorno de trabajo.
8. *Space Swarm*. En este capítulo se propone la creación de un juego en el que un personaje principal, manejado por el jugador, va disparando misiles hacia los enemigos que aparezcan y que se acerquen a él. El jugador perderá si es tocado por un enemigo tres veces. Para este proyecto, el libro explica técnicas de imagen que incluye Scratch como los efectos para hacer desaparecer un objeto, el uso de banderas o *flags* para intercambiar información entre varios objetos, la creación de una tabla de puntuaciones más altas que guarde los resultados de cada partida o la incorporación de música en bucle dentro del juego.
9. *Scratch hardware projects*. Esta unidad se dedica a exponer diferentes proyectos que pueden ser implementados mediante periféricos que permiten interacción con Scratch tales como una cámara digital *webcam*, la tarjeta Pi-coBoard o la Raspberry Pi.
10. *Seven shorties*. En este capítulo se ofrecen siete proyectos cortos para que el lector, con todo lo aprendido, pueda construirlos sin necesidad de explicaciones tan detalladas como en los anteriores.
11. *Making and sharing projects*. El último capítulo del libro, al igual que el analizado anteriormente, está dedicado a la componente social y de compartición que incluye Scratch, dando una pequeña introducción sobre los pasos generales a la hora de hacer programas así como los *bugs* o fallos más frecuentes que pueden darse, para terminar hablando de la importancia de

compartir lo creado y de recibir realimentación de la comunidad que pueda ayudar a profundizar más en aspectos aprendidos o en crear proyectos más complejos.

CAPÍTULO 4

La interacción con el mundo físico

Scratch se expande más allá del entorno de programación convencional permitiendo al usuario utilizar periféricos externos que le dan la libertad de poder añadir la interacción con el mundo físico en sus proyectos. En las siguientes secciones se nombra y detalla algunos de ellos y se muestran ejemplos de su uso con Scratch.

4.1 La tarjeta PicoBoard

La tarjeta PicoBoard, mostrada en la Figura 4.1, es una pieza hardware que permite que los proyectos de Scratch interactúen con el mundo real. Actualmente funciona en Scratch 1.4 tanto en su versión *offline* como *online* mientras que en Scratch 2 solo funciona en su versión *online*.

La versión anterior a la tarjeta PicoBoard se llamaba ScratchBoard e incluía los mismos componentes. En la actualidad, sigue siendo compatible con el programa Scratch y puede encontrarse por un precio de 30 €.

La forma en que la PicoBoard proporciona una forma de crear proyectos de Scratch que sientan y respondan a los estímulos del mundo real es mediante elementos como los que pasan a exponerse a continuación.

- **Sensor de sonido.** El sensor de sonido mide la intensidad de los sonidos que lo alcanzan. Mediante un rango de números del 0 al 100, este sensor nos mostrará en Scratch el dato recogido. Cuanto más alto sea el sonido, más alto será el valor.
- **Sensor de luz.** El sensor de luz mide la intensidad de luz que lo alcanza mostrando su valor en Scratch. Al igual que el sensor de sonido mostrará un número en el rango de 0 a 100. Cuanto más luz detecte, más alto será este valor.

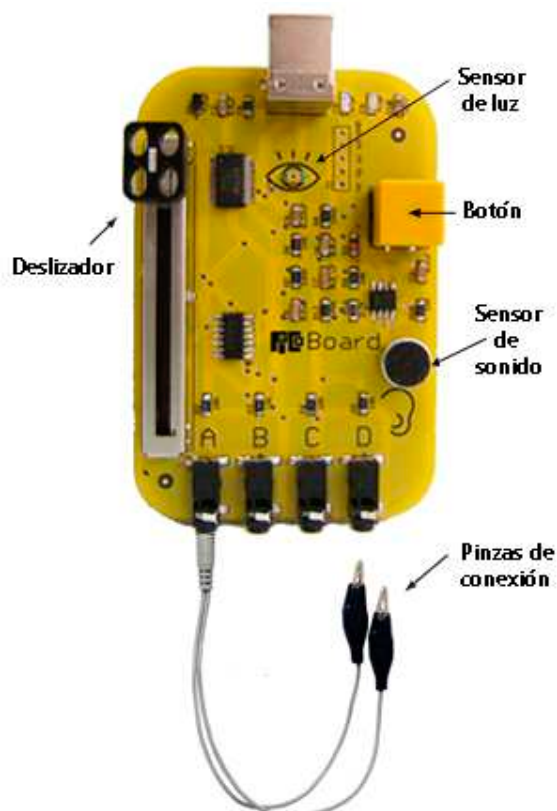


Figura 4.1: La tarjeta PicoBoard y sus elementos

- **Deslizador.** El deslizador muestra un valor en el rango de 0 a 100 que cambia según lo deslicemos de un extremo a otro.
- **Botón.** El botón devuelve el valor Verdadero o Falso dependiendo de si está presionado o no.
- **Pinzas de conexión.** Con estas pinzas el usuario puede construir todo tipo de sensores personalizados. Funcionan indicando un valor del 0 a 100 según la cantidad de resistencia eléctrica detecten en su extremo de metal.

4.2 Integración de la PicoBoard con Scratch

La tarjeta PicoBoard posee unas instrucciones específicas en Scratch para poder utilizarla en los proyectos creados en el programa. A su vez estas instrucciones incluyen un menú desplegable para elegir el sensor que se desee utilizar. En la Figura 4.2 se aprecia los sensores asociados al deslizador, el botón, la luz, el sonido y las cuatro resistencias.



Figura 4.2: Instrucciones de Scratch para la PicoBoard

4.3 Otros dispositivos

Además de la PicoBoard, existen otros dispositivos compatibles con Scratch como los que se pasan a nombrar a continuación.

4.3.1. Raspberry Pi

Scratch puede ser instalado y usado desde la computadora Raspberry Pi¹ [16]. Podemos comprar una Raspberry Pi por 32 €. Los proyectos creados con Scratch pueden hacer uso de los pines que esta placa posee en su esquina superior izquierda llamados pines de entrada/salida de propósito general (GPIO pins). En la Figura 4.3 podemos observar su ubicación mediante el cuadro resaltado en rojo.

Mediante estos pines y la instalación en el dispositivo de un programa especial para poder interactuar con Scratch, podrían crearse proyectos como por ejemplo un programa que encendiera o apagara leds conectados a la placa.

¹Página web accesible en <http://www.raspberrypi.org/>.

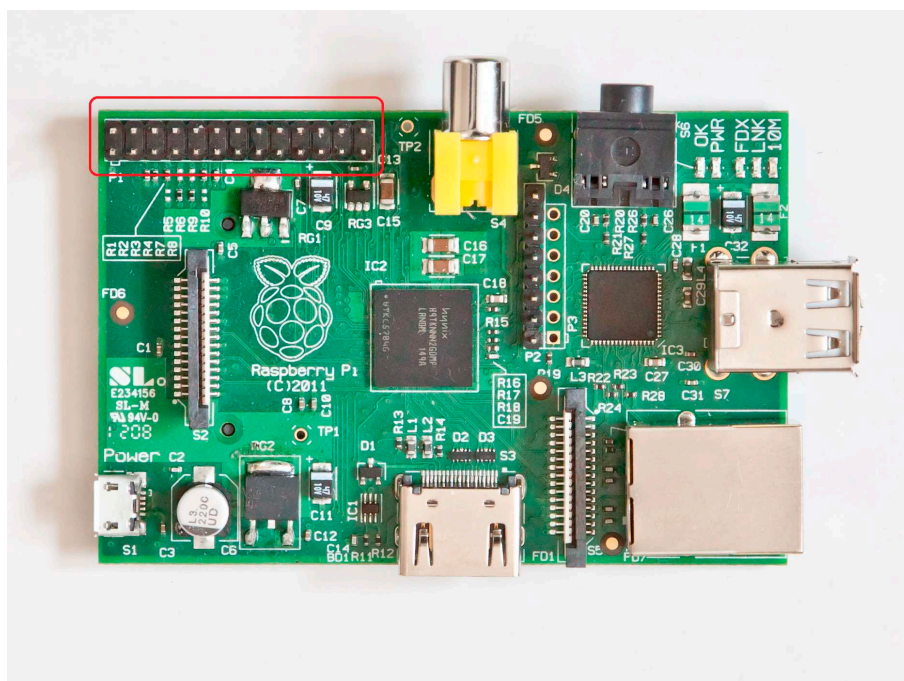


Figura 4.3: Raspberry Pi

4.3.2. Arduino

Existe un proyecto llamado Scratch for Arduino (S4A)² desarrollado en Barcelona por el grupo de Investigación EDUTECH que mediante la modificación de Scratch a través de una extensión instalada tanto en la placa Arduino como en el programa, proporciona bloques nuevos para tratar con los sensores y actuadores de la placa.

En la interfaz de S4A (Figura 4.4), la tarjeta Arduino aparece representada como un *sprite* u objeto más por lo que es posible añadir varias tarjetas al proyecto.

4.3.3. LEGO WeDo

El LEGO WeDo³ es un kit de construcción con herramientas robóticas destinado a niños con edades entre 7 y 11 años. Permite a los usuarios construir sus propios robots interactivos y programarlos mediante instrucciones de Scratch tal como muestra la Figura 4.5.

El kit incluye, además de piezas LEGO para la construcción de los robots, un motor, sensores de luz, de distancia, de inclinación y el conector *hub* para conectar el robot LEGO WeDo al ordenador.

²Proyecto accesible en <http://s4a.cat/>.

³Proyecto accesible desde su página web ubicada en <http://education.lego.com/en-us/lego-education-product-database/wedo/9580-lego-education-wedo-construction-set>.

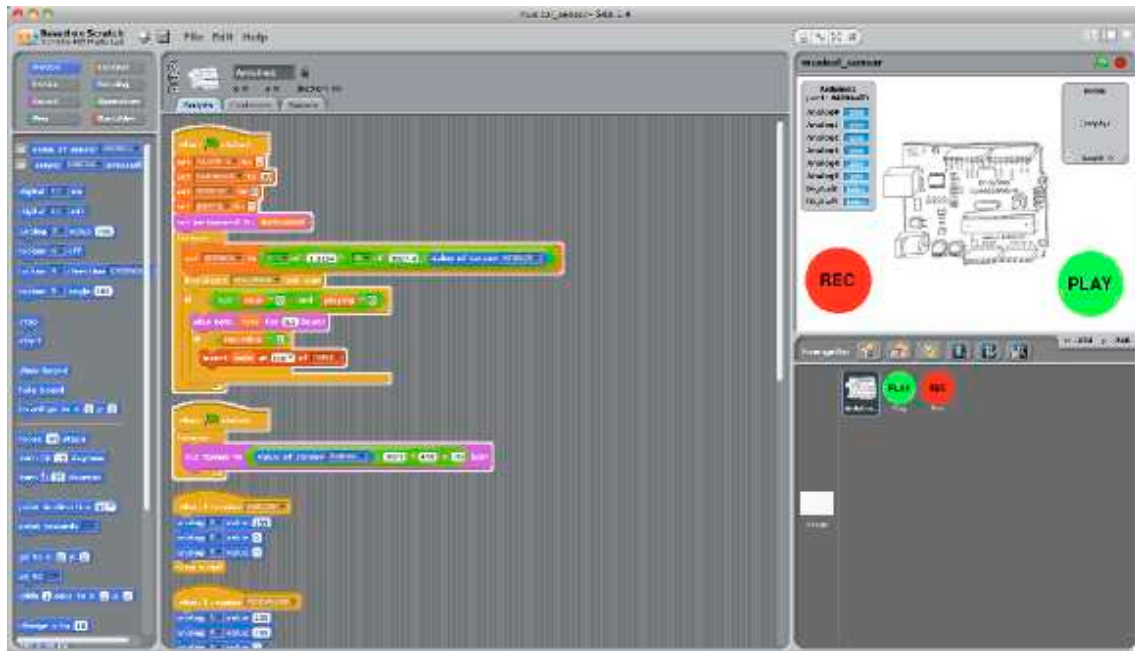


Figura 4.4: Interfaz de S4A



Figura 4.5: LEGO WeDo usando Scratch

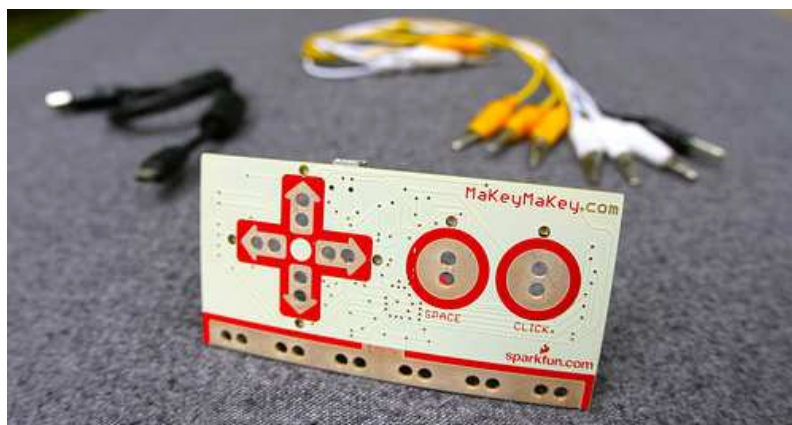


Figura 4.6: Componentes del kit MaKey MaKey

LEGO WeDo es compatible con la version tanto offline como *online* de Scratch 1.4 mientras que en Scratch 2.0 solo es compatible en su versión *online* mediante la incorporación de la correspondiente extensión al programa.

4.3.4. MaKey MaKey

Como se dijo en la sección 3.2.6 el MaKey MaKey es un kit compuesto por una placa hardware, un conjunto de pinzas de cocodrilo y un conector USB que permite convertir cualquier objeto en un gamepad (véase la Figura 4.6).

MaKey MaKey⁴ funciona conectando la placa hardware a cualquier objeto cotidiano mediante los cables de pinza de cocodrilo, como se indica en la Figura 4.7. Al tocar el objeto conectado, la conexión con MaKey MaKey hace que éste envíe al ordenador un mensaje que el ordenador interpreta como si fuera recibido desde un teclado o un ratón, que le indica que se ha presionado una tecla o un botón. Es por esta razón que además de con Scratch, MaKey MaKey es compatible con cualquier programa, ya que todos son capaces de leer de un teclado o de un ratón. El precio de MaKey MaKey ronda los 45 €.

4.3.5. Kinect

Kinect es un conjunto de sensores (cámaras, micrófonos...) desarrollado por Microsoft para ser utilizado en su videoconsola XBOX y que puede adquirirse por unos 150 €. Kinect permite al usuario interactuar con la consola sin necesidad de contacto físico con un mando u otro controlador. El proyecto Kinect2Scratch⁵ desarrollado por Stephen Howell (véase Figura 4.8) permite conectar un Kinect a Scratch y utilizar la información de todos los sensores que posee para usarlo en los proyectos creados con el programa.

⁴Proyecto accesible en su web <http://makeymakey.com/>.

⁵Accesible desde su página web <http://scratch.saorog.com/>.

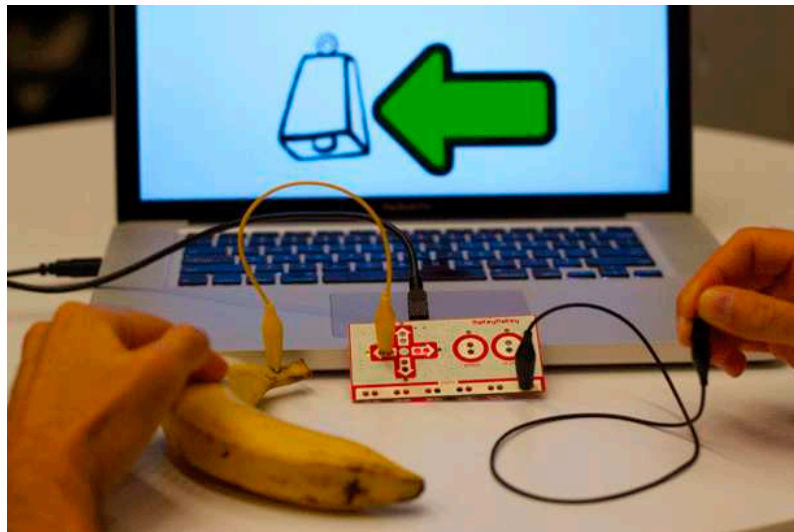


Figura 4.7: MaKey MaKey usando un plátano

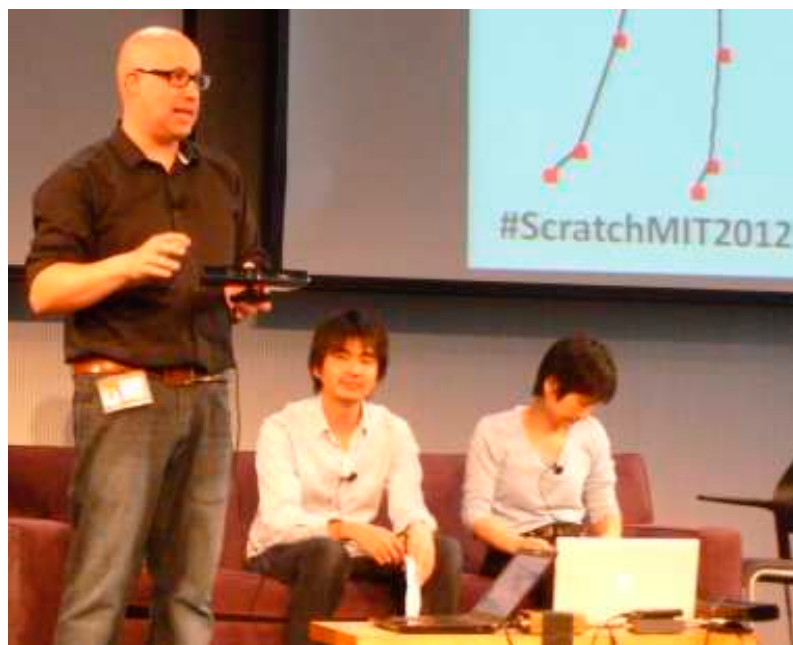


Figura 4.8: Stephen Howell presentando Kinect2Scratch, 2012

CAPÍTULO 5

La informática en la educación

Los cambios y efectos de la digitalización suponen una importante revolución a nivel mundial y una consciencia extendida sobre la misma, a diferencia de lo que pasó durante el surgimiento del cine y su industria a principios del siglo XX [7].

Así pues, la presencia de la informática y las nuevas tecnologías en el sistema educativo se ha desarrollado de forma notable en los últimos años. Estos cambios educativos se deben en gran parte a una fuerte presión social y económica, por lo que el hecho de que el motor del cambio sea de origen externo a la propia educación ha motivado inicialmente en muchos casos, que hayan sido pocos los planes pensados para su implantación a largo plazo y en gran escala [2].

De esta forma, aunque los ámbitos educativos implantan las nuevas tecnologías dentro de sus planes, existe una inadecuación de los marcos institucionales actuales que no son capaces de dar rápida respuesta a los desafíos y avances de las nuevas tecnologías.

5.1 España: la tecnología y la educación

Las nuevas tecnologías representan un aspecto esencial de nuestra cultura que sobre todo influencia a los más jóvenes que experimentan con ellas. El frenetismo evolutivo de dichas tecnologías puede entrar en conflicto muchas veces con la educación, una institución de evolución más pausada y lenta.

Este conflicto plantea algunas barreras, tal como indica un documento publicado por la Conferencia de Decanos y Directores de Escuelas de Ingeniería Informática (CODDII), junto con la Asociación de Enseñantes Universitarios de

la Informática (AENUI)¹. Las soluciones actuales, en su mayoría, se resumen en concentrar todo el apartado tecnológico dentro de una asignatura específica como puede ser la informática o tecnología, encapsulando el aprendizaje de éstas sin asociarlo al resto de materias impartidas. Además, dicha asignatura suele ser complementaria, por lo que no todos los alumnos la llegan a cursar. Como resultado, la interacción de estas nuevas tecnologías con el resto de ámbitos y sectores culturales, profesionales o educativos, acaba siendo, en gran parte, un proceso de autoaprendizaje por parte del alumno en su fase como estudiante de primaria, secundaria o bachillerato, que llevará a cabo en el momento que se le plantee el problema en su vida diaria, sin más apoyo o directrices que las que pueda buscar por sí mismo. En resumen, la informática debe introducirse como una ciencia fundamental para que los estudiantes estén preparados para los empleos del siglo XXI y la sociedad de la información.

Asimismo, según este documento conjunto de la CODDII y la AENUI, es fundamental en estos estudiantes el desarrollo del concepto de pensamiento computacional (*computational thinking*), un proceso de resolución de problemas por el cual se formulan de forma que permitan el uso de un ordenador, necesitando para su resolución la organización y el análisis lógico de la información que debe representarse a través de abstracciones como modelos o simulaciones. Finalmente se llegará a un conjunto de soluciones mediante el uso del pensamiento algorítmico y de entre ellas, se identificará y elegirá aquella que resulte más efectiva y eficiente en pasos y recursos para finalmente generalizar y transferir este proceso de resolución de problemas para lograr resolver una gran variedad de familias de problemas [19].

5.2 Informática en la educación española

Después de la realización de un estudio por el cual hemos consultado a varios centros su programa educativo con referencia a la informática, se puede determinar de manera global que todos los centros, como pilar fundamental de conexión con las nuevas tecnologías, disponen de conexión a Internet que tanto alumnos como profesores pueden utilizar.

También se incluyen asignaturas de informática con diversos contenidos según el curso estudiado por los alumnos y que se especificarán más detalladamente en las siguientes secciones.

¹Documento publicado en la página web de la CODDII <http://www.coddii.org> en 2014 titulado: *Enmiendas y propuestas CODDII al Proyecto de Real Decreto por el que se establece el currículo básico de la educación secundaria obligatoria y del bachillerato*.

5.2.1. Educación primaria

Dentro de la educación primaria no existe ninguna asignatura específica sobre informática, tal como recoge el Ministerio de Educación, Cultura y Deporte dentro del Real Decreto 126/2014, de 28 de febrero, por el que se establece el currículo básico de la educación primaria².

Este documento enuncia entre otros, los planes de asignaturas para este bloque educativo entre las cuales no se incluye ninguna materia informática ni que tenga relación específica con ella.

5.2.2. Educación secundaria

En la educación secundaria, según el Ministerio de Educación, Cultura y Deporte de España, Informática es una asignatura a elegir junto con ocho más dentro del 4º curso de la E.S.O³. A su vez, el Real Decreto 1631/2006, de 29 de diciembre, por el que se establecen las enseñanzas mínimas correspondientes a la educación secundaria obligatoria publicada en el BOE núm. 5, en viernes 5 de enero de 2007⁴ establece que los contenidos de la materia se deben estructurar en cuatro grandes bloques: uno de seguridad informática, un segundo de tratamiento de imagen, vídeo y sonido, un tercero sobre publicación y difusión de contenido en la Web y un cuarto bloque dedicado a Internet y las redes sociales.

Cabe destacar que entre los objetivos a alcanzar por la materia podemos encontrarlos tan variados como conocer y valorar las posibilidades que ofrece la tecnología o adoptar conductas de seguridad en Internet. Pero entre todos ellos, no se encuentra ninguno relativo a la programación o a la importancia de aprender a programar, así como tampoco se menciona en los bloques que deben formar la materia o en los criterios de evaluación para la superación de la misma.

Para concretar el análisis de las materias dadas en informática, se ha utilizado como guía el libro *Informática 4º E.S.O* [4] cuya portada puede verse en la Figura 5.1 y que es usado por estudiantes de 4º de la E.S.O. que deciden cursar la asignatura.

El libro se estructura en los cuatro bloques que determina el Ministerio de Educación, Cultura y Deporte de España dividiendo los contenidos en temas que incluyen una parte teórica, una práctica, una de ampliación de las materias dadas

²Datos extraídos del Boletín Oficial del Estado número 52 con fecha del 1 de marzo de 2014 que puede accederse desde <http://www.boe.es/boe/dias/2014/03/01/pdfs/B0E-A-2014-2222.pdf>

³Datos extraídos de la página web del Ministerio de Educación, Cultura y Deporte <http://www.mecd.gob.es/dms-static/c312bb0b-10d0-48a6-b4d0-4465f1248abb/organizacion-de-la-eso-pdf.pdf>

⁴Disponible a marzo de 2015 en http://www.boe.es/diario_boe/txt.php?id=B0E-A-2007-238

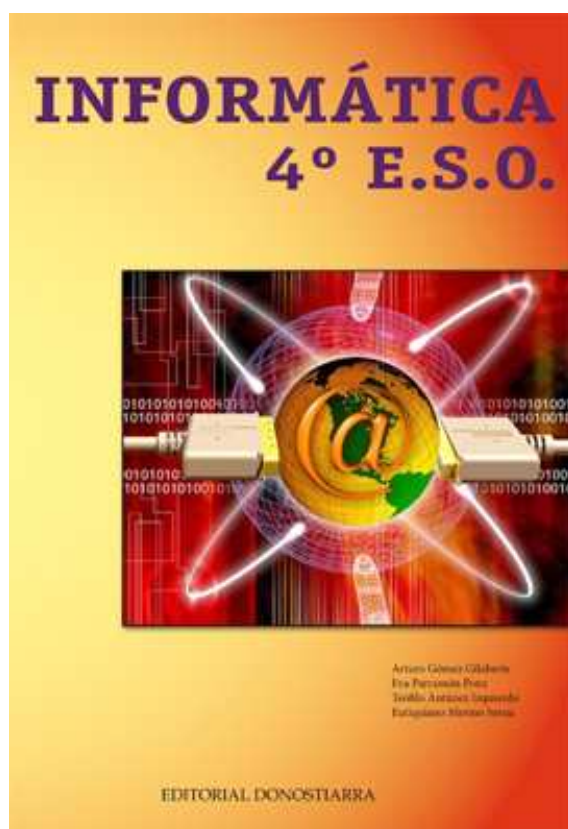


Figura 5.1: Portada de *Informática 4º E.S.O.*

en el tema y una final de recordatorio con los conceptos más importantes de cada unidad. Los doce temas que lo componen se enumeran a continuación.

1. **El ordenador y los sistemas operativos.** Se tratan conceptos como el hardware y el software de un ordenador, su estructura física y lógica y los sistemas operativos Windows y Linux.
2. **Redes locales.** Se explica el concepto de red, el tamaño que poseen, las redes iguales y las cliente-servidor, Ethernet, los dispositivos de una red LAN, el protocolo y servicios TCP/IP, los grupos de trabajo y las redes de dominio.
3. **Conexiones inalámbricas y dispositivos móviles.** Se estudian los dispositivos móviles, el concepto de computación ubícua, la diferencia entre ambas y la definición de conexión sin cables así como el tipo de conexiones que existen.
4. **Internet.** Se estudia el concepto, evolución histórica y fundamento técnico de Internet, los servicios que ofrece y el mundo electrónico.
5. **Herramientas colaborativas. Redes sociales.** Se explica el concepto de Web 2.0, blogs, wikis y redes sociales así como varios ejemplos de herramientas colaborativas.
6. **Seguridad en Internet.** En este tema se centran en la seguridad tanto en la máquina como en las personas explicando conceptos como los certificados digitales o la propiedad y distribución software en redes P2P.
7. **Imagen digital.** Se trata la edición de imágenes de mapa de bits.
8. **Imagen vectorial.** Se aborda la creación y edición de imágenes vectoriales.
9. **Edición de audio.** El sonido digital así como su edición son los conceptos centrales de este tema.
10. **Edición de vídeo.** Al igual que en el tema de sonido, se centra en la edición y creación de contenidos de vídeo.
11. **Presentaciones.** Este tema se centra en explicar el programa Powerpoint y su uso para crear presentaciones.
12. **Diseño de páginas web.** En este último tema se explica el concepto de página web así como una breve introducción al lenguaje HTML.

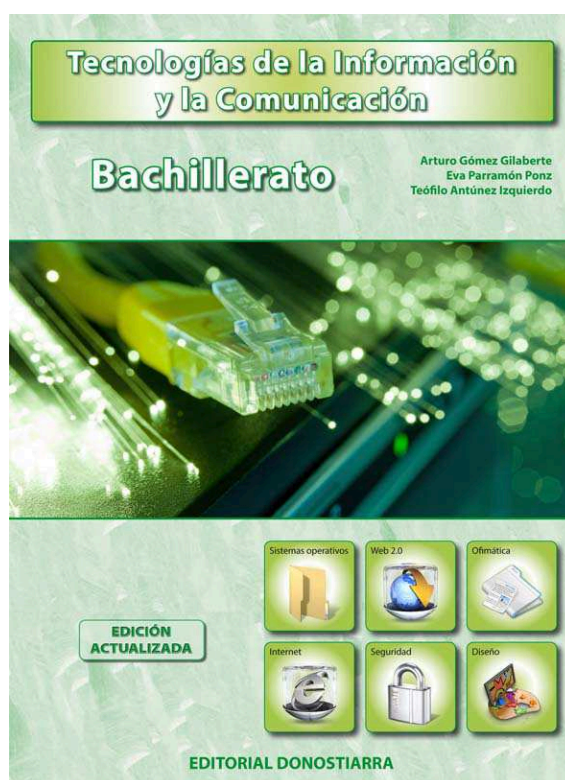


Figura 5.2: Portada de *Tecnología de la Información y la Comunicación*

5.2.3. Bachillerato

En bachillerato, según el Ministerio de Educación, Cultura y Deporte de España, la asignatura que recoge contenidos informáticos es llamada Tecnologías de la Información y la Comunicación, a elegir junto con ocho más tanto en primer curso como en segundo⁵. En el boletín, el cual incluye el Real Decreto 1105/2014 de 26 de diciembre por el que se establece currículo básico de la educación secundaria obligatoria y bachillerato, los contenidos de esta asignatura en primer curso de bachillerato se deben estructurar en cinco bloques: uno sobre la sociedad de la información y el ordenador, otro sobre la arquitectura de ordenadores, un tercero sobre software para sistemas informáticos, un cuarto para redes de ordenadores y un último bloque dedicado a la programación. Según el mismo, en el segundo curso los bloques se componen de uno inicial de programación, un segundo sobre publicación y difusión de contenidos y un último sobre seguridad.

Para concretar el análisis de las materias dadas, se ha utilizado como guía el libro *Tecnologías de la Información y la Comunicación* [4] cuya portada puede verse en la Figura 5.1 y que es usado por estudiantes de bachillerato que deciden cursar la asignatura.

⁵Información extraída de la página web del BOE accesible en <http://www.boe.es/boe/dias/2015/01/03/pdfs/B0E-A-2015-37.pdf>, marzo 2015.

El libro se estructura en catorce bloques que se dividen, a su vez, en un aparte teórica, una práctica y una final de ampliación. Los bloques recogidos en el libro pasarán a enumerarse a continuación.

1. **La sociedad de la información y el ordenador. Hardware y software.** Se tratan conceptos como la globalización tecnológica, las tecnologías de la información así como la definición de arquitectura de ordenadores o programas y aplicaciones.
2. **Sistemas operativos.** Este bloque trata en profundidad los sistemas operativos, sus funciones, clasificación y evolución así como el sistema operativo de Microsoft, Windows.
3. **Redes locales.** Se analizan conceptos como las redes informáticas, el modelo OSI o las especificaciones IEEE 802 entre otros.
4. **Internet. Redes sociales y trabajo colaborativo.** Se describe Internet y se aprende a crear grupos de contacto, hilos en foros así como a utilizar un metabuscador.
5. **Seguridad.** Se centra en la seguridad tanto en la máquina como en las personas y analiza diferentes tipos de encriptación.
6. **Edición de imágenes. Imagen digital: GIMP. Dibujo vectorial: AutoCAD, Inkscape.** Este bloque trata el uso de diversos programas para la creación y edición de imágenes vectoriales.
7. **Edición de audio y vídeo.** Este apartado explica diversos métodos de edición tanto de sonido como de vídeo.
8. **Procesadores de texto.** En este bloque se explican las acciones y conceptos fundamentales de los procesadores de texto así como el uso avanzado del entorno de trabajo Microsoft Word 2010.
9. **Presentaciones.** Mediante el programa Powerpoint se le enseña al estudiante a crear presentaciones tanto para CD-ROM como portátiles.
10. **Hojas de cálculo.** Este bloque trata en profundidad la edición y creación de hojas de cálculo, tanto con el programa Calc como el Excel.
11. **Gestores de bases de datos.** Esta unidad profundiza en el concepto de base de datos y su estructura así como en el programa de manejo de bases de datos de LibreOffice.
12. **Diseño de páginas web.** Se estudia el lenguaje HTML y se aplican conceptos de diseño de páginas web.

13. **Programación.** Este bloque es el único que se dedica al tema de la programación en todo el libro. Se explican características como la historia y evolución de los lenguajes de programación así como conceptos como los tipos de datos o los operadores. Finalmente, se dedica un pequeño apartado introductorio a la programación en el lenguaje C.
14. **Simulación.** Este último bloque introduce al alumno en programas de simulación y le propone actividades tales como simulaciones electrónicas, matemáticas o físicas.

En conclusión, los alumnos de bachillerato, al acabar este ciclo, tendrán una ligera idea sobre programación si han cursado alguna asignatura optativa relacionada con la informática, pudiendo darse el caso de alumnos que empiezan la universidad sin tener conocimientos previos sobre programación o lenguajes de programación.

CAPÍTULO 6

Actividades pedagógicas

En el presente capítulo se proponen actividades adecuadas a cada nivel según lo descrito en el anterior. Dichas actividades entretendrán al alumno consiguiendo que aprenda conceptos de programación de una forma dinámica y divertida.

6.1 Actividades para educación primaria

Las actividades orientadas a la educación primaria deben ser las más simples de todas ya que en esta etapa los alumnos no conocen ningún aspecto de la programación ni han interactuado nunca con ningún lenguaje. Las actividades deben estar orientadas a lo básico y a hacerles aprender los conceptos más sencillos mientras se divierten programando.

6.1.1. El deseo de Currutago

Este sencillo programa nos muestra a un mago por pantalla que nos pregunta qué deseo queremos realizar y nos da a elegir entre tres opciones tal y como muestra la Figura 6.1.

Al responderle según las teclas que nos indica, el mago hará realidad el deseo. En este sencillo proyecto se estudian los siguientes conceptos:

- **Funcionamiento básico de un programa.** Mediante este proyecto en el que aún no se incluyen bucles, el alumno puede observar de forma sencilla cómo funciona un programa, indicando el orden en que se ejecuta cada instrucción, empezando por la primera que se incluya y terminando por la última. Además, al no tratarse de un ejemplo complicado puede llegar a visualizar el concepto de *computational thinking* o pensamiento computacional por el cual, teniendo un problema inicial (un mago que debe preguntarnos qué deseamos y luego cumplir nuestro deseo), conseguimos dividirlo en



Figura 6.1: El deseo de Currutago en ejecución

una serie de problemas más pequeños o pasos secuenciales que el ordenador es capaz de entender y ejecutar.

- **La interacción con el usuario y la entrada de datos.** El alumno escribe la pregunta para que el usuario la responda y aprende que ese dato lo puede encontrar en la variable «respuesta» dentro del bloque «Sensores» para poder analizarlo (véase la Figura 6.2). Mediante este ejercicio, comprende el concepto de variable en programación por el cual sabe que si desea manejar datos, antes debe recibirlos de algún sitio y guardarlos para poder tratarlos.
- **Uso de condicionales.** El programa necesita instrucciones condicionales para llevarse a cabo (véase la Figura 6.3) ya que tiene que elegir entre una orden u otra dependiendo de la respuesta obtenida. Las instrucciones condicionales son la parte principal del proyecto e indican de forma clara para qué son útiles, cuál es su estructura, cómo funcionan y ante qué situaciones deben usarse. A su vez, se muestra que las condicionales pueden concatenarse, usándose varias dentro de un mismo programa.
- **Uso de efectos y diálogos.** El proyecto necesita que se le añadan efectos gráficos como el cambio de color o el efecto correspondiente para que el personaje desaparezca. Además el alumno deberá incorporar diálogos y preguntas, por lo que mediante este proyecto aprende a distinguir entre las diferentes posibilidades de las instrucciones relacionadas con la apariencia del objeto, instrucciones que distinguen diálogos simples y diálogos que esperan respuesta y entre los diferentes bloques que componen Scratch.

6.1.2. Una mascota especial

Este proyecto muestra a un murciélago por pantalla, que mediante el uso de una cámara de vídeo tal como se muestra en la Figura 6.4, nos seguirá a través de la pantalla.

Además el juego incluye instrucciones de animación por las cuales el murciélago moverá las alas mientras vuela y música que sonará constantemente de fondo.

Los conceptos de programación a los que se le presta atención en este proyecto son:

- **Programación de varios objetos.** En este proyecto, por un lado, se programa el murciélago y, por otra, el fondo para incluir la música, por lo que cada una de estas actividades se programará en un objeto diferente del programa.
- **Uso de periféricos externos.** El pilar fundamental de este proyecto se incluye el uso de la cámara de vídeo y de las diferentes instrucciones que posee Scratch para programar con ella tal y como se muestra en la Figura 6.5. El

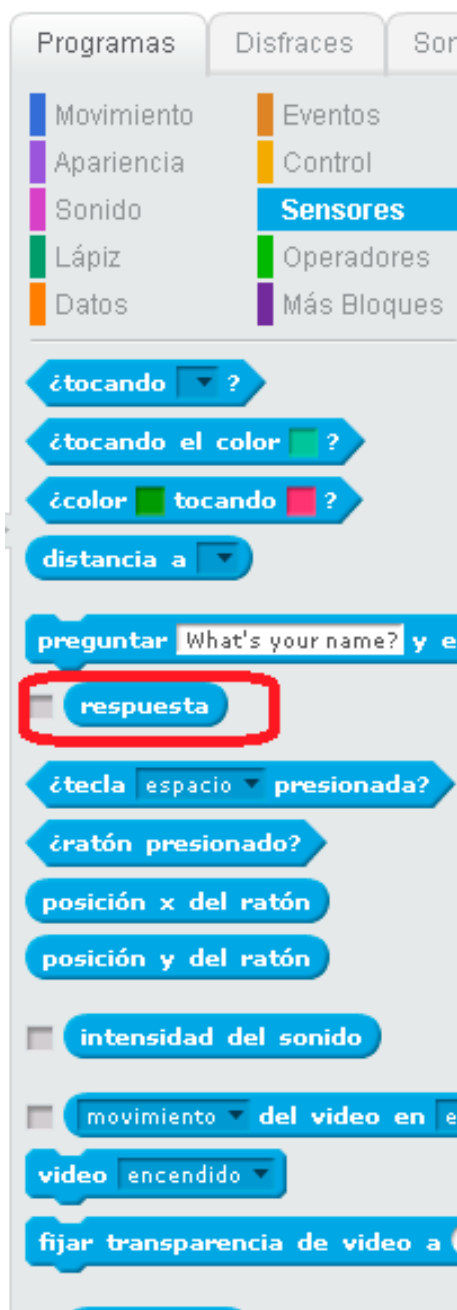


Figura 6.2: Ubicación de la variable «respuesta»



Figura 6.3: Instrucciones condicionales utilizadas

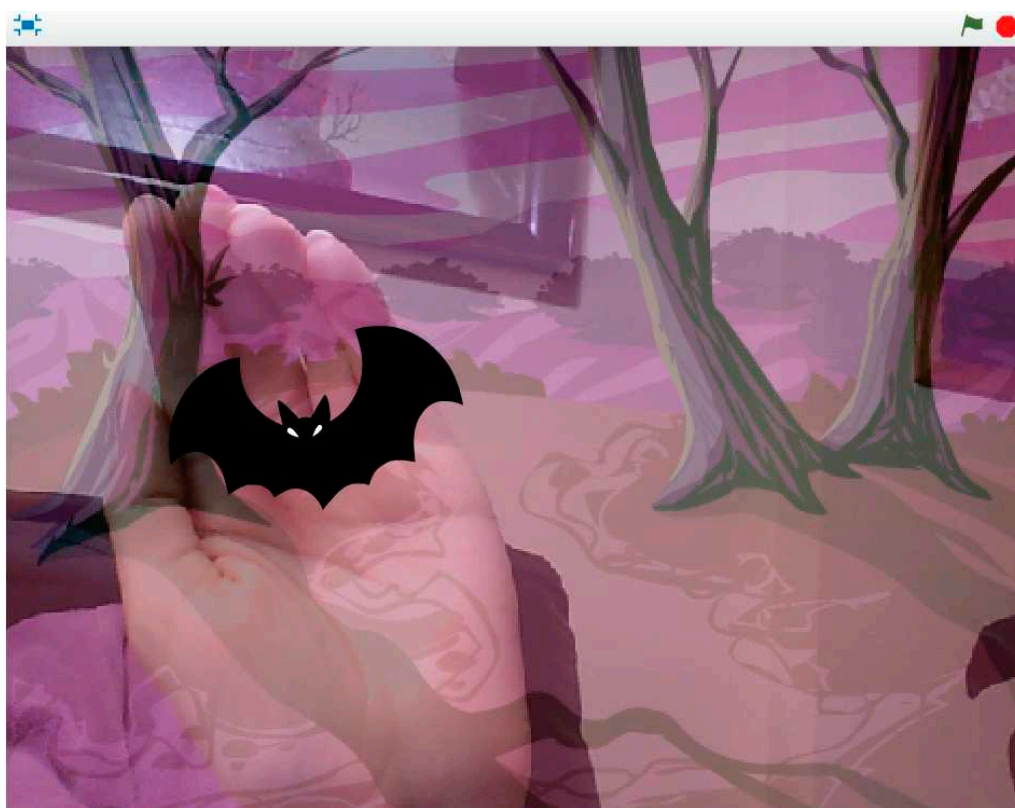


Figura 6.4: Una mascota especial en ejecución



Figura 6.5: Instrucciones utilizadas en el objeto murciélago

alumno debe elegir entre las opciones que dichas instrucciones poseen a la hora de hacer que el programa funcione correctamente y el murciélago siga el movimiento de la mano. Mediante este proyecto, se demuestra que la programación no es algo que solo puede tener interacción con el mundo real a través del teclado sino que puede recibir y dar información a través de otros muchos dispositivos.

- **Bucles como elementos de repetición.** En este proyecto se incluye el uso de bucles tal y como se muestra en la Figura 6.5 y en la Figura 6.6. En esta última podemos observar el código que permite que la música suene durante toda la ejecución del proyecto sin detenerse. Mediante ese programa, el alumno puede entender de forma sencilla cómo funcionan los bucles y cuál es su función dentro de la programación.
- **Música y animación.** El proyecto requiere tanto música por un lado como animación del murciélago por otro. El alumno aprenderá a manejar ambos conceptos mediante Scratch interactuando con los disfraces y con las bibliotecas de sonido e imagen.

6.1.3. ¡Asústame!

Este proyecto es un ejemplo de interacción con algunos sensores de la tarjeta PicoBoard. El programa, tal como aparece en la Figura 6.7, nos muestra a un gato por pantalla que reaccionará ante eventos tales como los sonidos fuertes o la falta de luz (en este último caso el fondo cambiará por uno más oscuro).

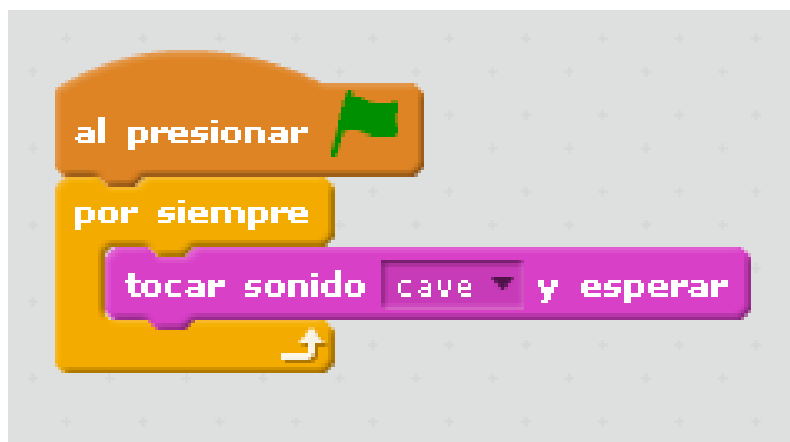


Figura 6.6: Ejemplo de bucle simple

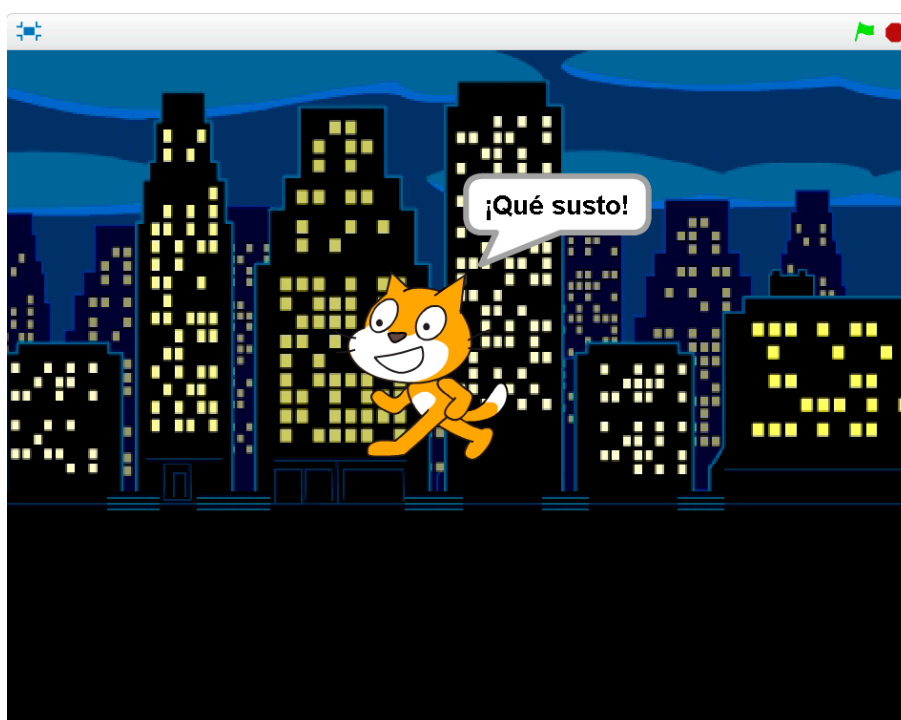


Figura 6.7: ¡Asústame! en ejecución

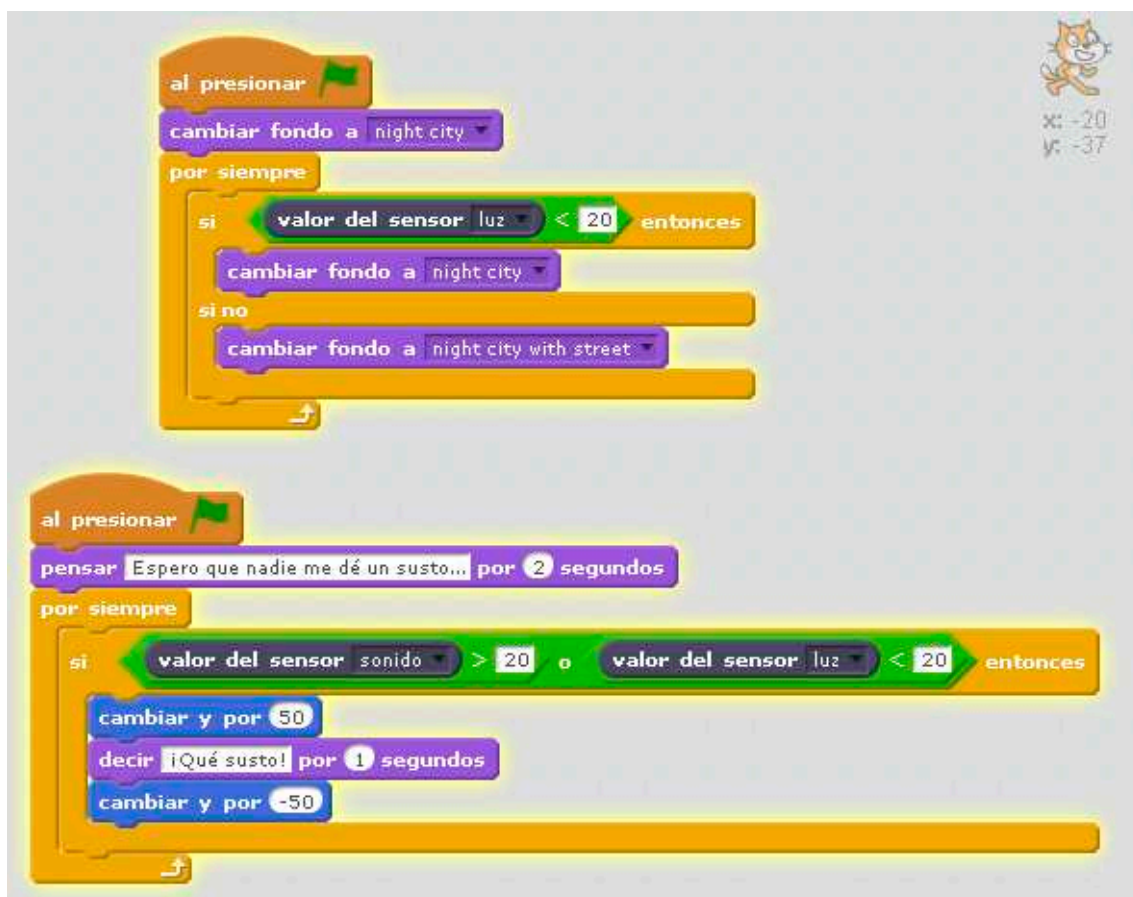


Figura 6.8: Uso de instrucciones para la PicoBoard (en color gris)

Los conceptos en los que hace énfasis este proyecto son:

- La tarjeta PicoBoard.** En este proyecto se incorpora el uso de la placa de sensores PicoBoard (véase la Figura 6.8). En concreto se usarán los sensores de sonido y de luz. El alumno aprenderá a distinguir los diferentes componentes de la PicoBoard así como interpretar los datos que reciban de la tarjeta para poder entender el uso de las instrucciones que proporciona Scratch para programarla.
- Paralelismo.** Este proyecto incluye un concepto importante dentro de la programación como es el paralelismo. Según muestra la Figura 6.8, dentro de un mismo objeto (en este caso el gato) se han incluido dos bloques paralelos que se ejecutarán a la vez: uno se encargará de recibir la información de los sensores de la PicoBoard para determinar si el objeto debe saltar o no y el otro se encargará de cambiar el fondo del proyecto si el sensor con el que el usuario ha interactuado es el de luz.
- Bucles y condiciones.** En este proyecto se une el uso de bucles con la utilización de condiciones. Una vez el alumno ya haya aprendido, gracias a los

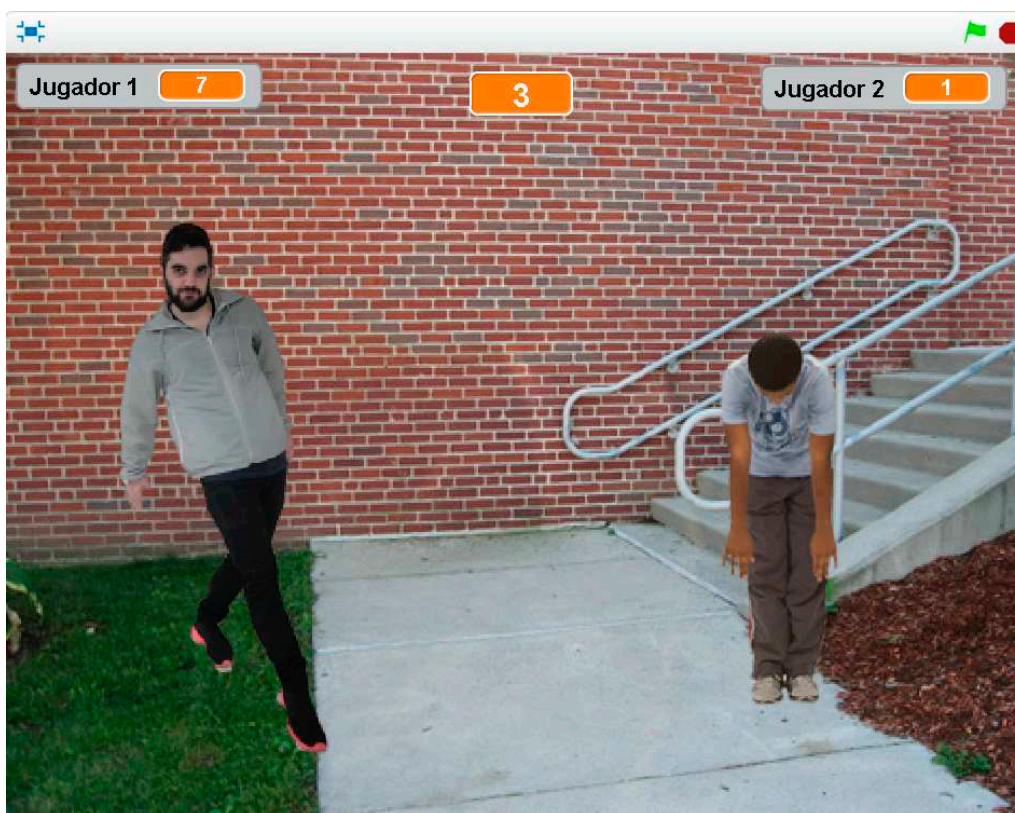


Figura 6.9: La batalla de baile en ejecución

anteriores proyectos, a usar este tipo de instrucciones por separado, podrá ser capaz de combinarlas y empezar a trabajar creando programas algo más complejos como el de este proyecto. Además, dentro de la instrucción condicional se incluye una estructura de operadores más compleja para que el alumno no trabaje únicamente con operadores únicos sino que sea capaz de formar estructuras lógicas más avanzadas.

6.1.4. La batalla de baile

En este caso se trata de crear un juego para dos jugadores. Cada uno de los jugadores deberá presionar una tecla del teclado para hacer que su personaje se mueva. Ganará el jugador que al finalizar el tiempo haya conseguido más movimientos con su personaje. La Figura 6.9 muestra el juego en ejecución. Este proyecto, al tratarse del más complejo, podría ser adecuado para trabajar en grupo.

Los conceptos destacables en este proyecto son:

- **Uso de bucles y condicionales.** Este proyecto vuelve a combinar ambos tipos de instrucciones para su creación, como muestra de ejemplo la Figura 6.10.

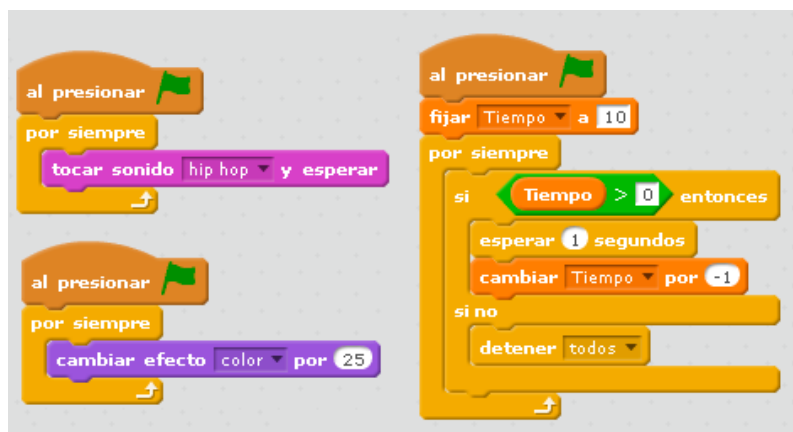


Figura 6.10: Implementación del fondo del proyecto

- Uso de variables.** Se utilizan tanto una variable de tiempo que actúa como cronómetro dentro del juego como variables de marcador de jugador para ir contando el número de veces que presionan la tecla correspondiente de movimiento para cada uno de ellos. El uso de una de ellas dentro del juego puede observarse en la Figura 6.11.
- Programación de varios objetos.** Se vuelve a hacer hincapié en este concepto ya que tanto cada jugador como el fondo deberán ser programados por separado.
- Trabajo en grupo.** Este proyecto sirve para hacerle ver al alumno que los programas más complejos deben conseguirse mediante el trabajo en grupo, en el que cada miembro aporta su parte para crear un proyecto mucho más complejo de lo que podrían conseguir por separado.

6.2 Actividades para educación secundaria

Las actividades orientadas a secundaria son algo más complejas por el nivel general que ya tienen los alumnos en esta parte de su educación ya que aunque no se imparte específicamente programación dentro de la asignatura de informática, sí que se incluye diseño de páginas HTML por lo que se explora un poco este tipo de lenguaje.

6.2.1. ¡Adivina el número!

En este primer proyecto el ordenador pensará un número al azar entre 0 y 10 y el jugador dispondrá de cuatro intentos para adivinarlo (véase la Figura 6.12). En cada uno de estos intentos el ordenador le dará pistas, indicándole si el número introducido es mayor o menor que el que ha pensado.

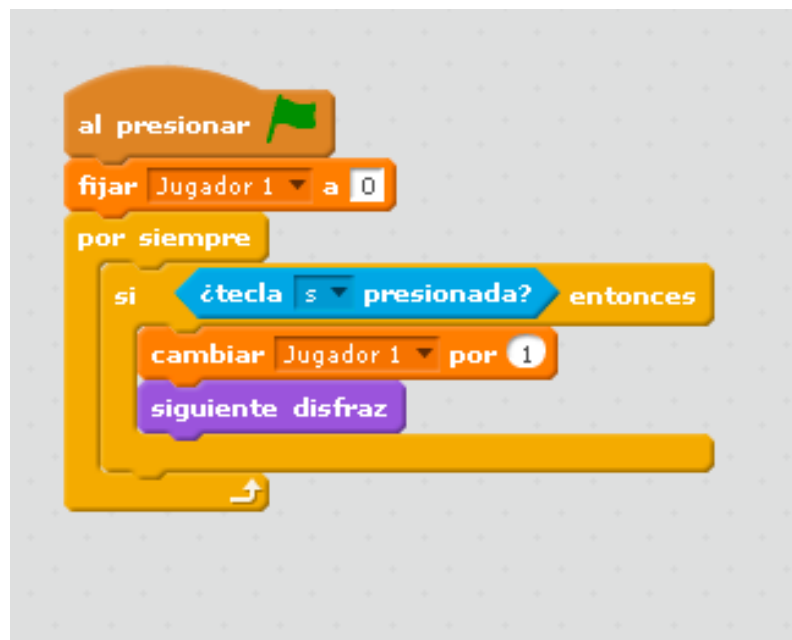


Figura 6.11: Código de implementación del Jugador 1



Figura 6.12: ¡Adivina el número! en ejecución

Los conceptos introducidos en este proyecto son:

- **Estructura de un programa.** Al no incorporar paralelismo o programación de otros objetos dentro del mismo proyecto, este ejemplo resulta útil para mostrar el funcionamiento de un programa, la secuencialidad de instrucciones y la transformación de una idea inicial a un resultado que el ordenador sea capaz de procesar (pensamiento computacional).
- **Utilización de variables.** Como muestra la Figura 6.13, el programa maneja diversas variables, tanto creadas por el usuario (en color naranja) como propias de Scratch para manejar respuestas o sensores (en azul). El alumno aprenderá a diferenciar ambas y a manejar las instrucciones que utilizan para poder implementarlas correctamente en el juego.
- **Operadores y funciones aleatorias.** Este proyecto incluye los operadores básicos de igualdad para las condiciones así como la función de aleatoriedad para poder determinar que el ordenador elija un número al azar entre 0 y 10. Su implementación puede observarse en la instrucción de color verde que muestra la Figura 6.14.
- **Instrucciones condicionales y bucles.** El proyecto incluye por una parte varias instrucciones condicionales para poder determinar si el número introducido por el jugador es menor, mayor o igual y cómo actuar ante cada caso. Además, para que el juego se ejecute cuatro veces, se incluye un bucle de repetición en el que el alumno puede observar de forma clara que el programa se repetirá tantas veces como número incluya en el bucle (instrucción *repetir* de la Figura 6.13).

6.2.2. Explota la pelota

En este proyecto el usuario deberá hinchar una pelota en el menor tiempo posible mediante el uso de un micrófono hasta hacer que explote (véase la Figura 6.15).

Los conceptos de programación en este proyecto son:

- **Uso de periféricos.** En este proyecto se incluye la interacción con periféricos externos. En este caso el alumno aprende a manejar la información recibida desde el micrófono así como el conjunto de instrucciones que dispone Scratch para la programación de sensores (instrucciones de color azul de la Figura 6.16).
- **Variables, instrucciones y operadores complejos.** La programación del objeto pelota requiere del uso de variables y de operadores más complejos

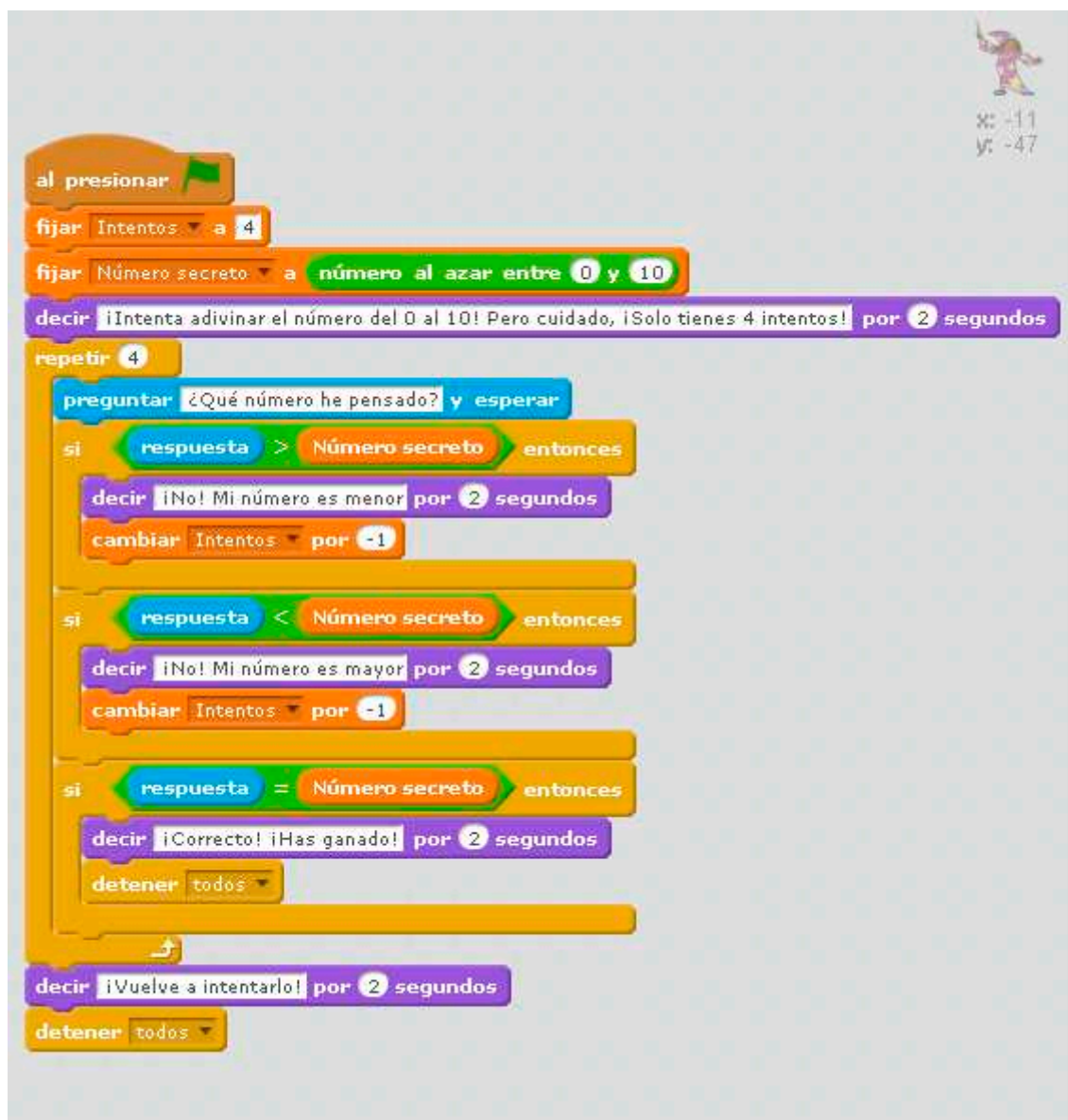


Figura 6.13: Código del objeto mago



Figura 6.14: Instrucción de aleatoriedad en Scratch

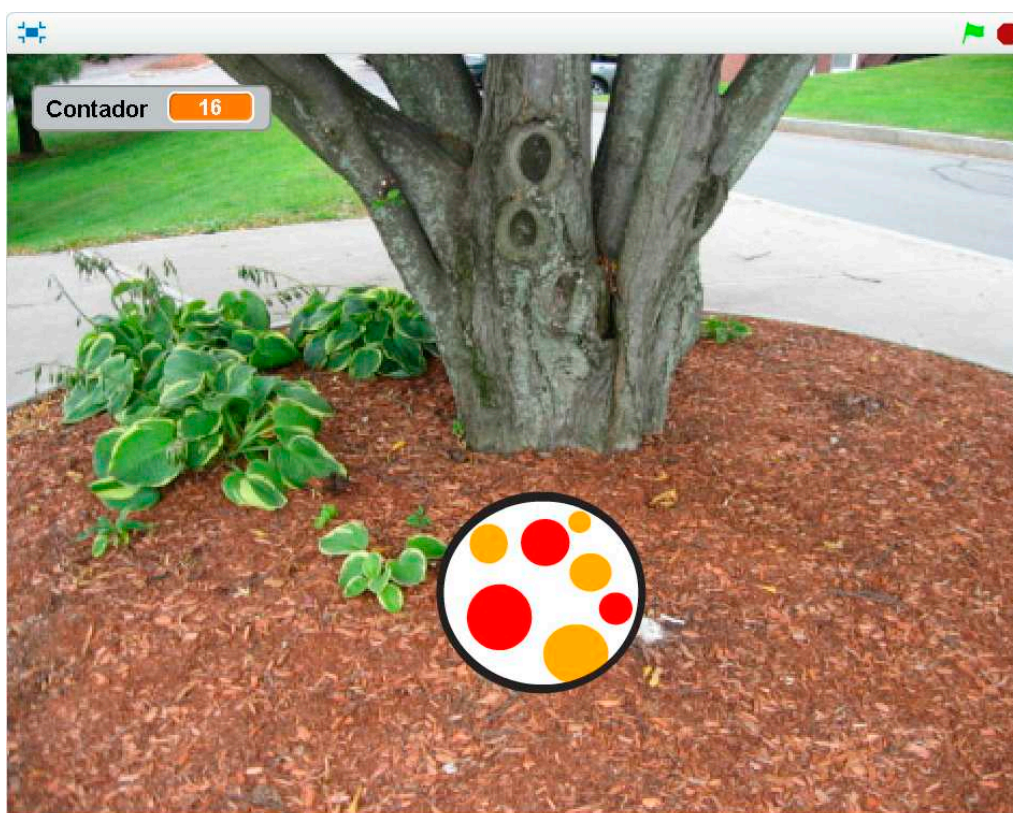


Figura 6.15: Explota la pelota en ejecución

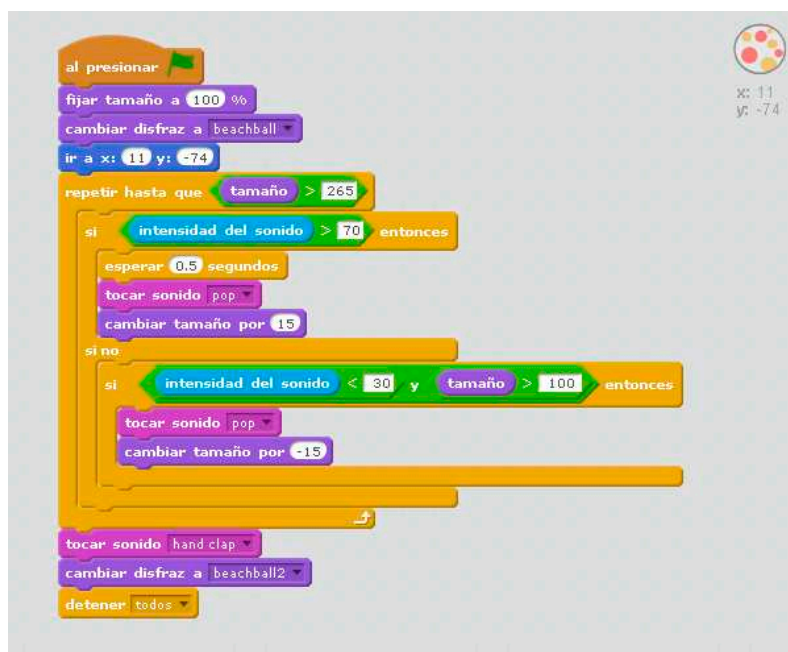


Figura 6.16: Programa del objeto pelota

para crear las estructuras lógicas necesarias para las instrucciones condicionales. Además, el alumno deberá controlar las instrucciones tanto de apariencia como de sonido para poder determinar el tamaño de la pelota, cambiar su disfraz a la hora de que explote y que a su vez suene un sonido determinado.

- **Uso de bucles e instrucciones condicionales.** El proyecto vuelve a utilizar la combinación de bucles e instrucciones condicionales para su implementación tal como muestra la Figura 6.16.
- **Programación de varios objetos.** Para poder crear el contador (cuyo código se muestra en la Figura 6.17), además de programar el objeto pelota, el alumno tendrá que programar el fondo del juego.

6.2.3. ¡Atrápame si puedes!

Este proyecto muestra a un tiburón que debe perseguir a un pez controlado por el usuario durante diez segundos (véase la Figura 6.18). Si el tiburón llega a comerse el pez, el jugador pierde, mientras que si consigue huir de él hasta que se acabe el tiempo, gana.

En este proyecto podemos encontrar los siguientes conceptos:

- **Paralelización.** Cada uno de los objetos del programa requiere de paralelización (como muestra la Figura 6.19 con la programación del objeto pez)



Figura 6.17: Programa del contador en el fondo

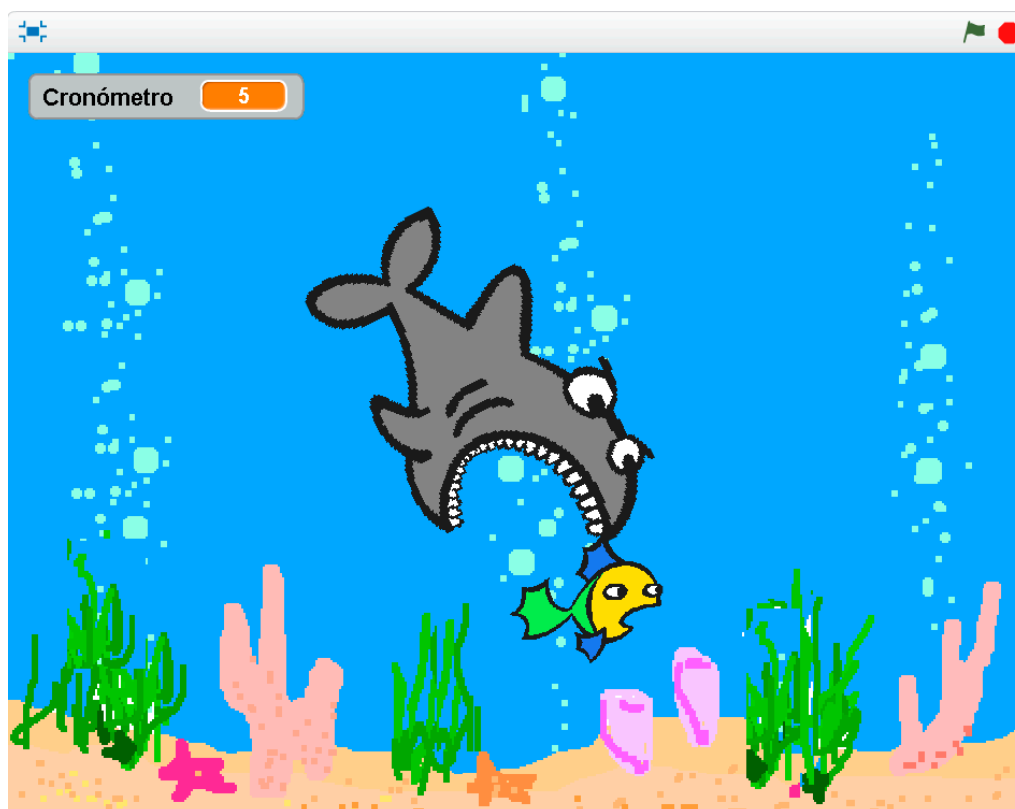


Figura 6.18: ¡Atrápame si puedes! en ejecución

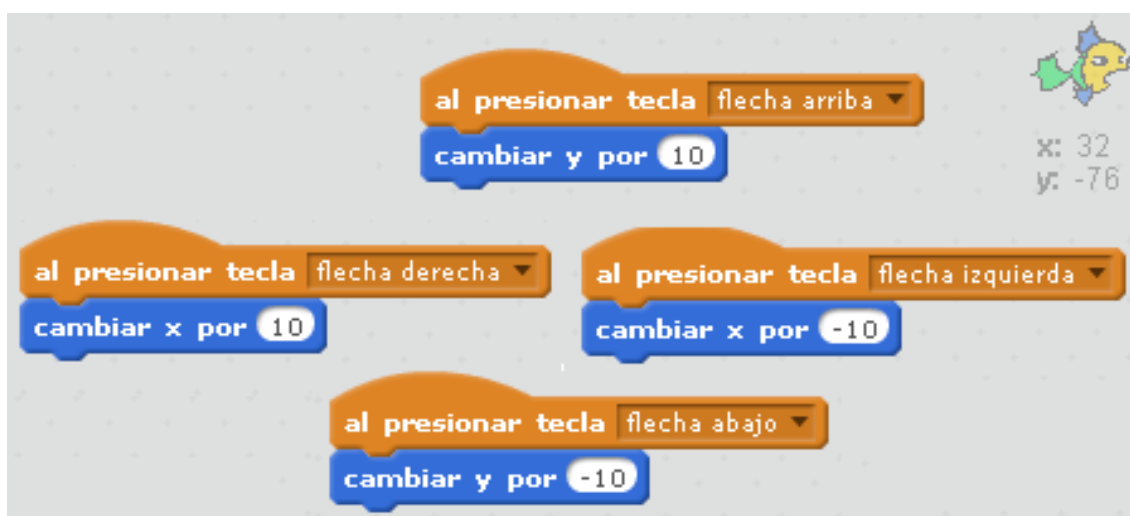


Figura 6.19: Programación del objeto pez

por lo que el alumno acabará entendiendo el concepto si quiere llevar el proyecto a cabo.

- **Programación de variables.** El juego necesita una variable para contar los segundos que necesita el jugador hasta que pueda ganar.
- **Programación de varios objetos.** Cada uno de los personajes del juego, así como el fondo, necesitan ser programados con las funciones a desempeñar.
- **Combinación de condicionales y bucles.** Como en proyectos anteriores, éste combina el uso de bucles con instrucciones condicionales. Además, incluye conceptos como la condición de parada, es decir, una instrucción o conjunto de instrucciones que se utilizan como condición para terminar un bucle infinito tal y como muestra la Figura 6.20 con la instrucción que corresponde a la condición de parada resaltada en rojo.

6.2.4. Scratch Invaders

Este proyecto sigue una dinámica muy parecida al juego *Space Invaders*. Al ser algo más complejo de implementar, es ideal para ser trabajado en grupo.

El jugador controla una nave con la que irá desplazándose por la pantalla disparando misiles hacia los marcianos que se moverán acercándose hacia él. En esta versión se dispondrá de un cronómetro que irá descontando hasta llegar a cero. Si el jugador consigue acabar con todos los enemigos antes de que se acabe el tiempo, habrá ganado (véase la Figura 6.21).

La nave se controlará haciendo uso de algunos sensores de la PicoBoard. En este caso se utilizará el deslizador para moverla por la pantalla y el botón para disparar el misil.



Figura 6.20: Condición de parada en la variable de tiempo

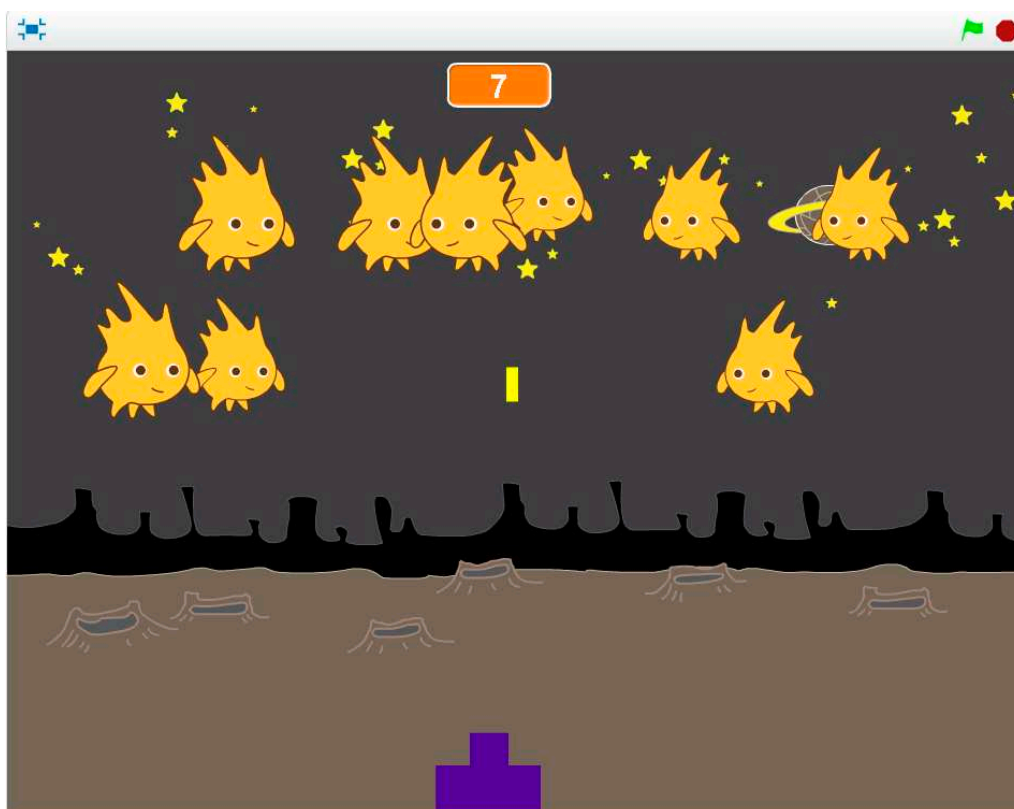


Figura 6.21: Scratch Invaders en ejecución

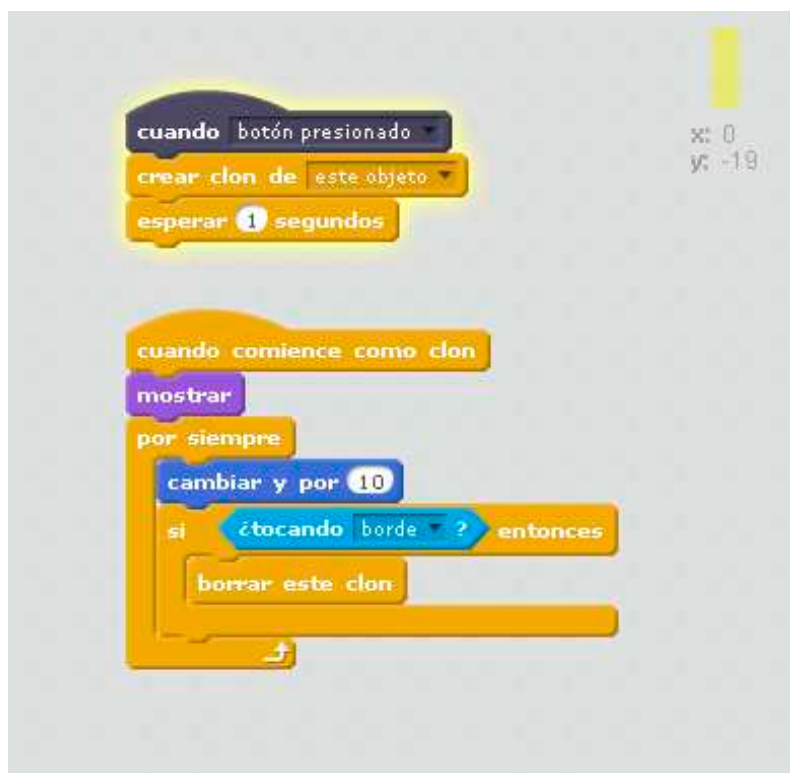


Figura 6.22: Implementación del objeto misil. Uso de instrucciones relacionadas con la PicoBoard

Los conceptos que se incluyen en este proyecto son:

- **Trabajo en grupo.** Al tratarse de un trabajo algo más complejo que el resto de proyectos puede ser realizado en grupo y así desarrollar esta habilidad tan importante en la programación.
- **Uso de la tarjeta PicoBoard.** En este proyecto se hace uso de dos sensores dentro de la PicoBoard. Por un lado el deslizador que nos permitirá mover la nave (véase la Figura 6.22) y por otro el botón que al ser presionado, hará que se dispare el misil (véase la Figura 6.23).
- **Programación de varios objetos.** En este proyecto se deben programar, por una parte, el misil, la nave y el marciano (que luego deberá clonarse tantas veces como marcianos se quiera) y, por otra, el fondo, para que lleve la cuenta del tiempo que queda hasta que se acabe el juego.
- **Paralelismo.** Para poder crear este proyecto, el alumno deberá utilizar el paralelismo en todos los objetos que lo componen.
- **Uso de condicionales y bucles.** De nuevo el uso de estas instrucciones vuelve a ser esencial por lo que afianza su conocimiento en el alumno.

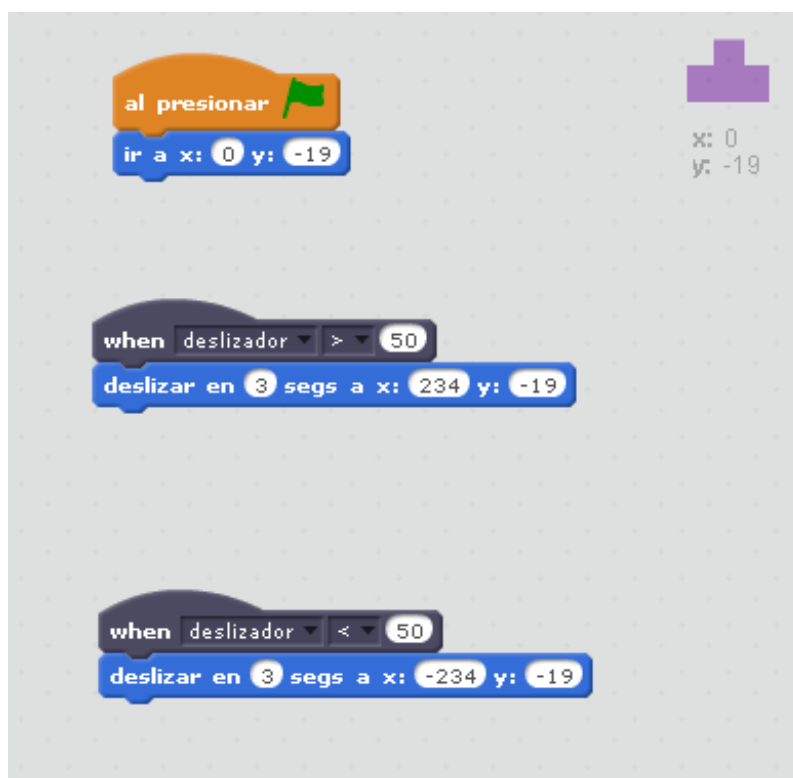


Figura 6.23: Implementación del objeto nave. Uso de instrucciones relacionadas con la PicoBoard

- **Interacción con los bloques de Scratch.** En este juego se hace un variado uso de todos los tipos de instrucciones que posee Scratch, tales como movimiento, efectos, operadores, variables, etc., por lo que ayuda a conocer el entorno de trabajo del programa y a dominar su estructura.

6.3 Actividades para bachillerato

En este último apartado se propondrán los proyectos más complejos debido al nivel que ya han alcanzado los estudiantes así como el hecho de que aquellos que cursan la asignatura Tecnologías de la Información y la Comunicación darán dentro de la misma un bloque de introducción a la programación. Los proyectos incluidos abarcarán apartados propios de la asignatura como el uso de listas o paso de mensajes.

6.3.1. Flappy Scratch

Este proyecto crea un juego muy parecido al famoso *Flappy Birds*. El jugador manejará mediante el botón izquierdo del ratón a un gato que deberá sortear una serie de obstáculos sin fallar ni una vez. Cada vez que supere uno de éstos, se le sumará un punto al marcador. El juego terminará en el momento en que el gato no consiga pasar a través de un obstáculo o llegue a tocar el suelo. El juego en ejecución puede verse en la Figura 6.24.

Los conceptos sobre programación destacables en este proyecto son:

- **Funciones de aleatoriedad, variables y coordenadas.** En el juego el gato solo se mueve de arriba a abajo mientras que es el escenario el que parece moverse hacia él avanzando y creando obstáculos diferentes. Para conseguir este efecto, se dispone de un solo objeto obstáculo al cual se le crearán diferentes disfraces tal como muestra la Figura 6.25.

Una vez tenemos el objeto obstáculo con varios disfraces, para lograr la sensación de que el paisaje se va moviendo desplazaremos el obstáculo hasta la posición final del eje X, posición más a la izquierda que abarca la pantalla del juego, para que una vez llegue ahí, desaparezca y vuelva a aparecer en la posición más alta del eje X que correspondería al lado opuesto de la pantalla (posición más a la derecha). Para lograr que cada vez sea un obstáculo diferente el que aparezca, antes de que el objeto vuelva a aparecer en el lado del eje X más hacia la derecha de la pantalla, incluiremos una instrucción que haga que cambie de disfraz entre todos los posibles dibujados de forma aleatoria. El código que implementa todos estos pasos es el que se muestra en la Figura 6.26.



Figura 6.24: Flappy Scratch en ejecución

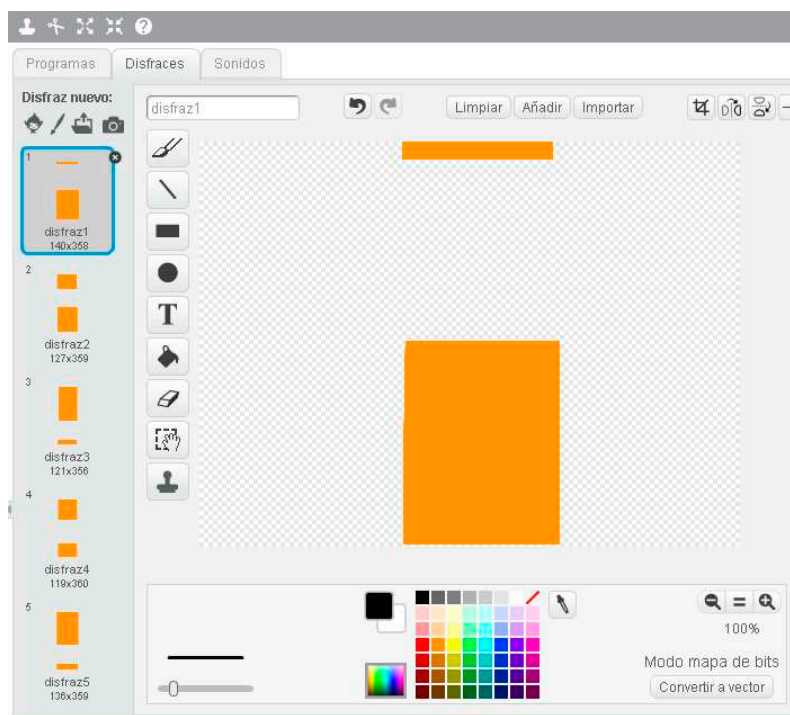


Figura 6.25: Disfraces del objeto obstáculo

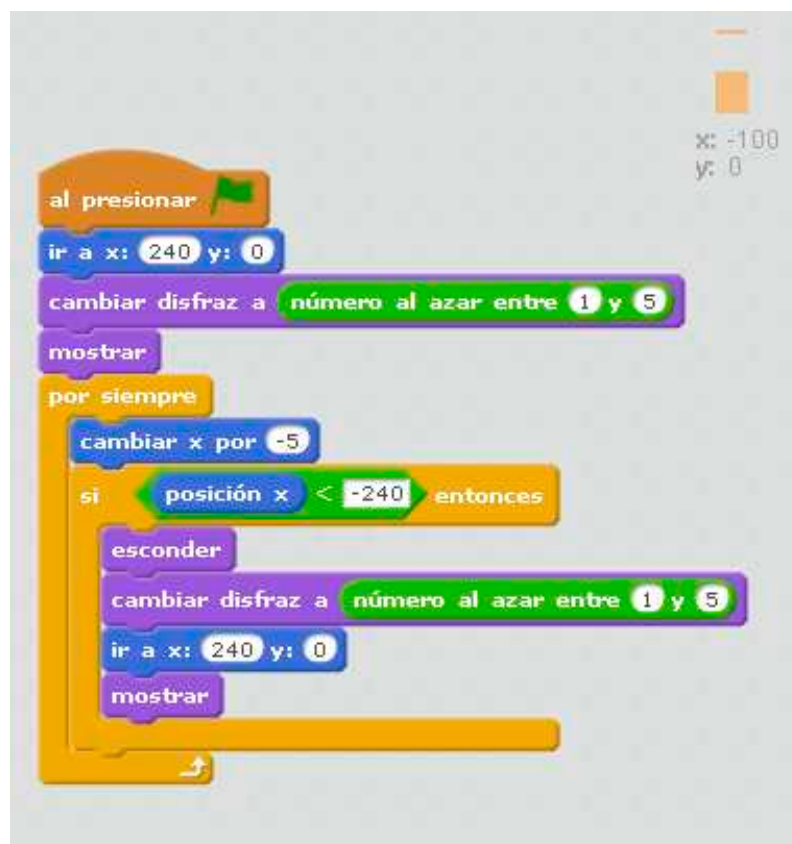


Figura 6.26: Código del objeto obstáculo

- **Programación de varios objetos.** En este proyecto en concreto deberá programarse el objeto gato, que es el que domina el jugador, y el objeto obstáculo, que irá desplazándose por el fondo dando la sensación de que el escenario va moviéndose.
- **Paralelismo.** El objeto gato necesita hacer varias actividades simultáneamente y de forma paralela. Por un lado, deberá elevarse si se aprieta el botón izquierdo del ratón y deberá descender si no se aprieta. A su vez, ha de tener un control de la puntuación por la cual, si supera un obstáculo, se le debe sumar un punto a su marcador. Finalmente deberá comprobarse cuándo pierde, dándose ese caso cada vez que el gato toque el color naranja (tanto el suelo como los obstáculos son de este color). El código para implementar este objeto se muestra en la Figura 6.27.
- **Condicionales y bucles.** El uso de este tipo de instrucciones vuelve a ser fundamental en este proyecto.

6.3.2. Scratch Test

Este proyecto implementa una aplicación test (véase la Figura 6.28). El programa le hará una serie de preguntas al jugador. Si no responde correctamente o se acaba el tiempo antes de que conteste a todas, perderá.

En este proyecto podríamos destacar los siguientes conceptos:

- **Uso de listas y variables.** Debido a que en bachillerato, dentro de la asignatura Tecnología de la Información y la Comunicación, se estudia el concepto de lista, resulta útil incorporarlo en este proyecto ya sea a modo de introducción o como ejemplo práctico. En este caso se manejarán dos listas, una que mostrará las preguntas y otra que contiene las respuestas, que será la que usaremos para comparar con lo introducido por el usuario. En la Figura 6.29 se muestra el contenido de cada una de estas listas así como las instrucciones que posee Scratch para manejarlas (resaltadas en rojo).

Asimismo, el programa necesitará, por un lado, la variable de tiempo, otra de respuesta para almacenar el resultado introducido por el usuario y una variable «Pregunta» que almacena el número de la pregunta que se va a mostrar por pantalla para así comparar la respuesta del usuario con el elemento de la lista de respuestas que corresponda con ese mismo número (véase la Figura 6.30).

- **Paso de mensajes.** En este proyecto se incluye un concepto algo más complejo dentro de la programación como es el paso de mensajes. En este caso, lo utilizaremos para indicarle al juego que debe terminarse. Para ello en el escenario implementaremos una variable tiempo que irá descontando hasta llegar a cero; una vez esto ocurra, enviará un mensaje al objeto principal

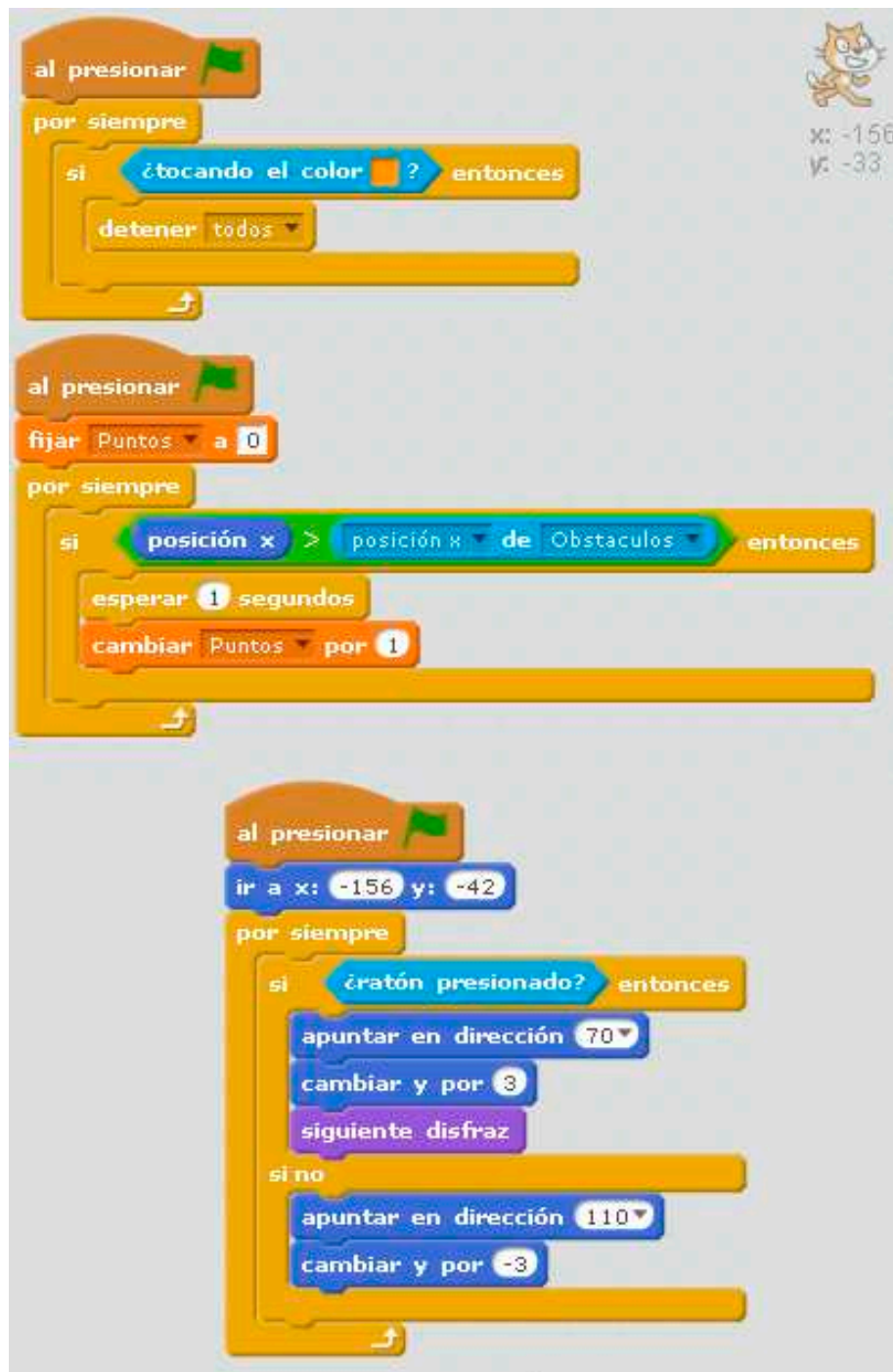


Figura 6.27: Código del objeto gato

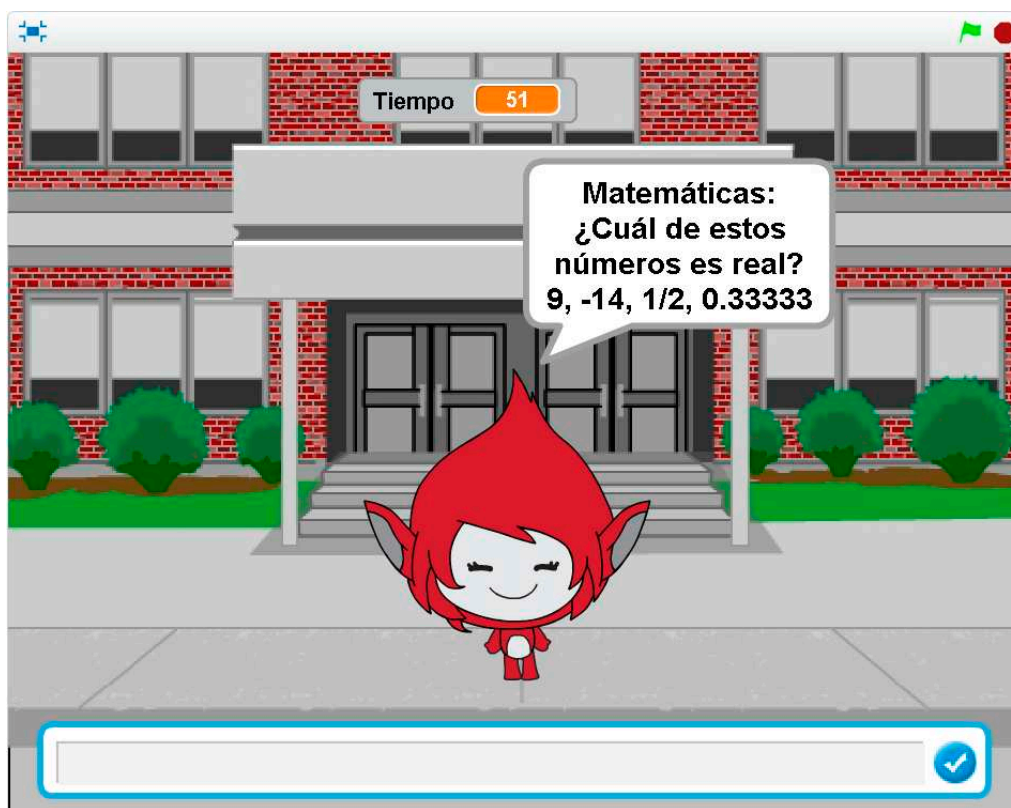


Figura 6.28: Scratch Test en ejecución

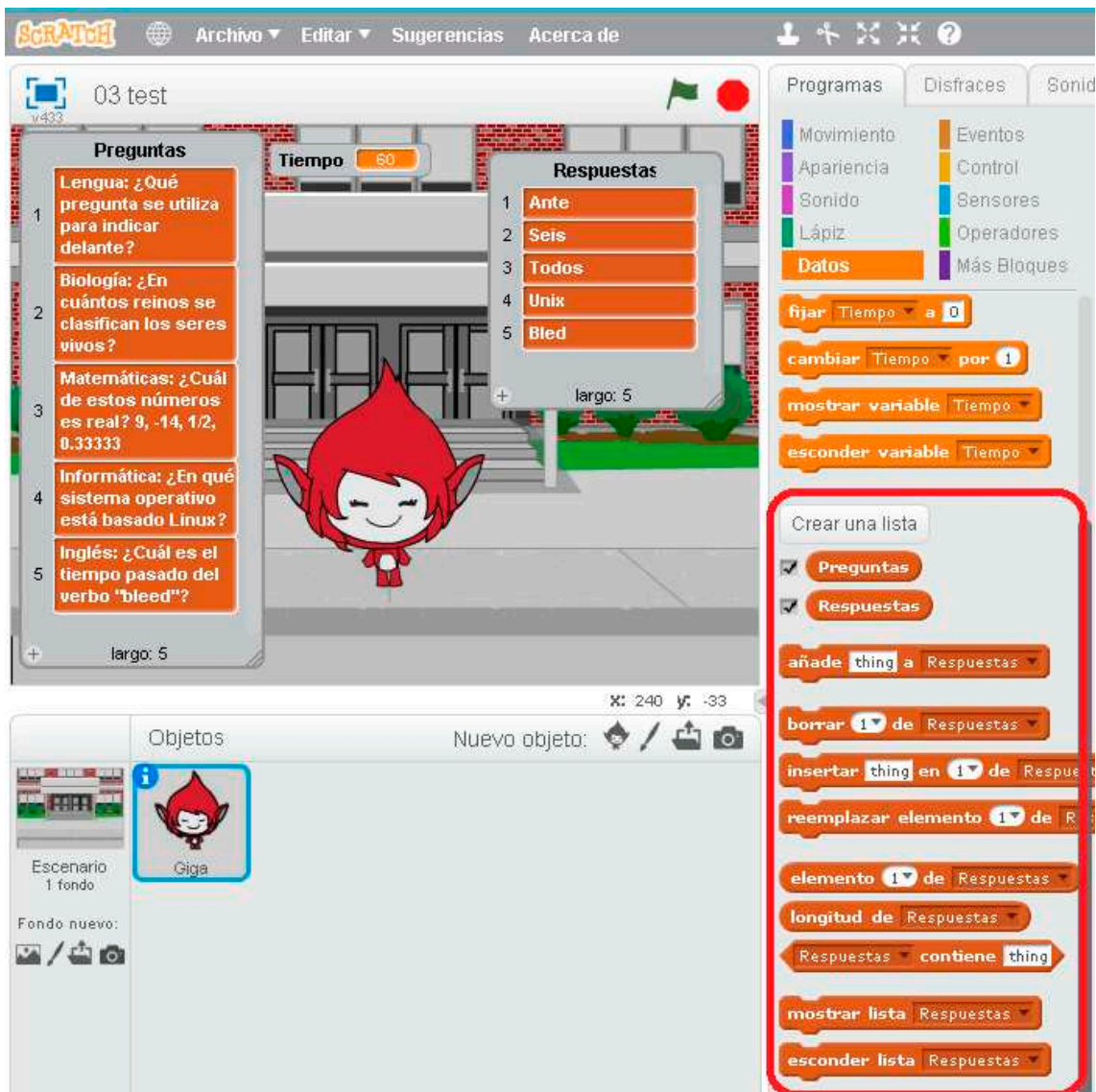


Figura 6.29: Listas en Scratch

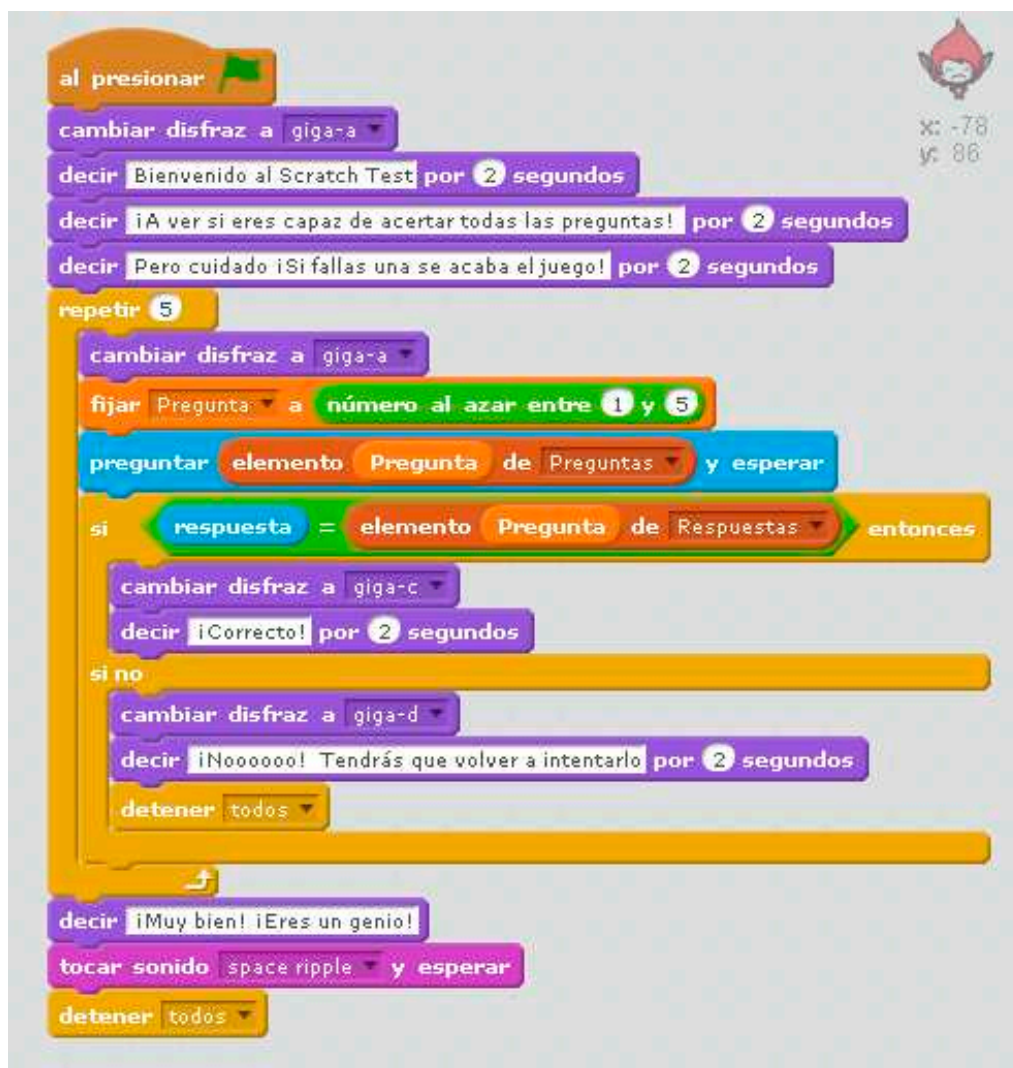


Figura 6.30: Implementación con listas y variables



Figura 6.31: Envío de mensaje en Scratch

(véase la Figura 6.31) que el objeto principal del juego recibirá para seguidamente, terminar mostrándole un mensaje al jugador indicándole que el tiempo ha acabado y ha perdido (véase la Figura 6.32).

- **Paralelismo.** Giga, el objeto principal de este proyecto, posee paralelización ya que, por un lado, debe interactuar con el usuario, mientras que por otro debe estar pendiente de recibir el mensaje por parte del escenario que le indique que el tiempo ha acabado.
- **Programación de varios objetos.** Tanto el escenario como el objeto principal (Giga) deben programarse en este proyecto.

6.3.3. Pong

Este proyecto implementa el famoso juego Pong en su versión para dos jugadores tal como indica la Figura 6.33. Cada uno de los jugadores controlará una de las paletas que utilizarán para lanzar una pelota de uno a otro. Cada vez que un jugador falle al devolver la pelota, el juego sumará un tanto al marcador del jugador contrario. Al acabarse el tiempo, ganará el jugador que tenga la puntuación más alta.

Al tratarse de un juego para dos jugadores y ser algo más complejo en programar, este proyecto es ideal para realizarlo en grupo. Los conceptos presentes en este proyecto son:



Figura 6.32: Recepción de mensaje en Scratch

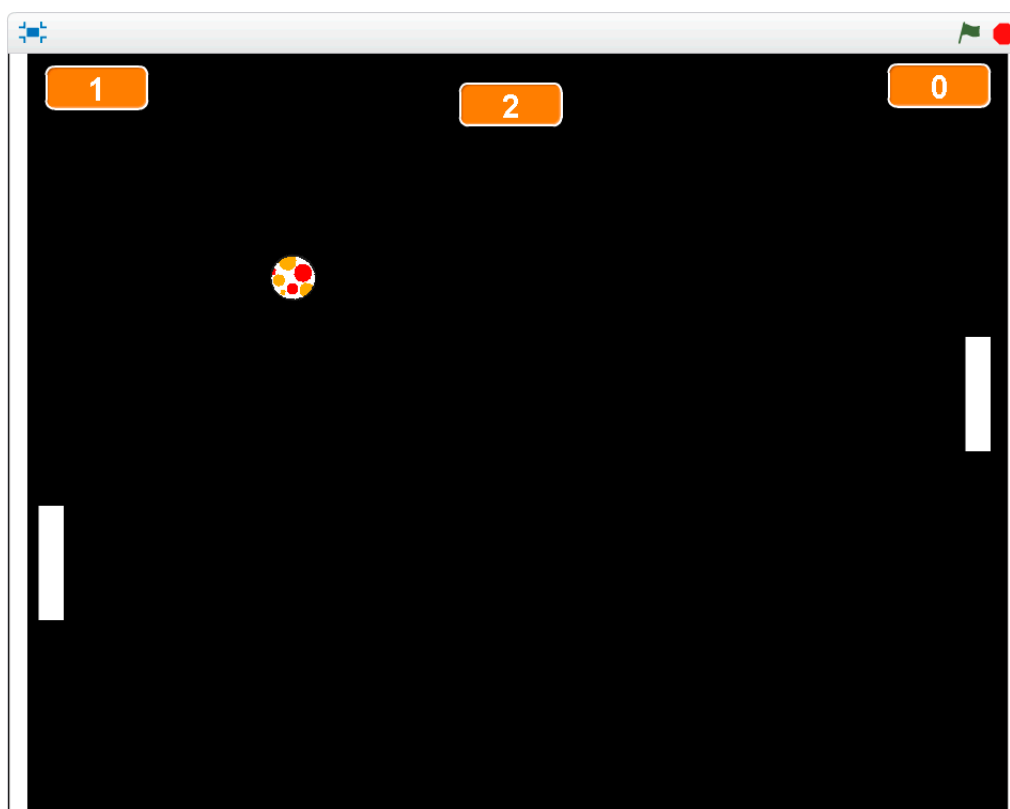


Figura 6.33: Pong en ejecución

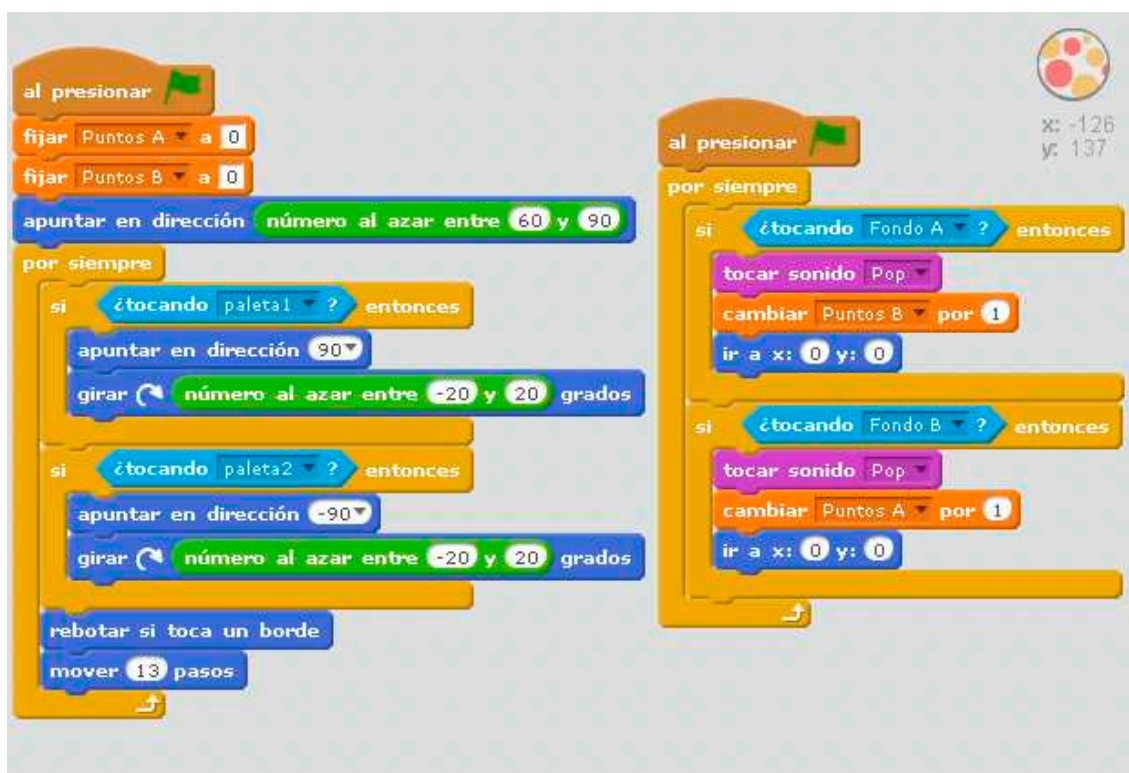


Figura 6.34: Código del objeto pelota

- **Trabajo en grupo.** Este proyecto puede servir para incentivar el trabajo en grupo y la colaboración.
- **Función de aleatoriedad, variables y sensores.** El juego necesita, por un lado variables que controlen el tiempo así como las puntuaciones de los dos jugadores, aleatoriedad en la dirección que debe tener la pelota una vez rebote en cualquiera de las dos paletas y sensores que detecten por un lado si lo que la pelota ha tocado es una paleta o es el fondo que queda por detrás de ella. En este último caso se le sumaría un punto al jugador contrario y la pelota volvería al centro de la pantalla (véase el código de la Figura 6.34).
- **Paralelismo.** El programa usa paralelismo tanto en el código del escenario, que por un lado debe controlar el tiempo y por otro mostrar la pantalla de fin y terminar el programa cuando se requiera, como con el código del objeto pelota, que deberá implementar su movimiento así como las instrucciones necesarias para sumar puntos a los marcadores. El código de este último objeto se muestra en la Figura 6.34.
- **Paso de mensajes.** Dentro del código incluido en el escenario, puede introducirse de nuevo el concepto de paso de mensaje. En el escenario se encuentra la variable de tiempo que irá descontando hasta llegar a cero. Una vez sea cero, sonará un sonido para indicar que el juego ha acabado así como una pantalla de fin y una orden de terminación del programa. Todas



Figura 6.35: Código incluido en el escenario

estas instrucciones pueden ser introducidas dentro de una misma instrucción condicional pero también pueden ser llamadas mediante el paso de un mensaje dentro del mismo escenario, tal como muestra la Figura 6.34, para así reforzar este concepto.

- **Programación de varios objetos.** Tanto el objeto pelota como el escenario necesitan ser implementados. En este proyecto, a su vez, se incluyen objetos que no requieren ninguna programación como son las dos barreras que se colocan detrás de cada una de las paletas y que sirven para indicarle al programa que la pelota no ha sido atrapada por el jugador.
- **Condicionales y bucles.** Por último, el uso de bucles e instrucciones condicionales es necesario en este proyecto para lograr que funcione correctamente.

6.3.4. El adivinador de frutas

En este proyecto se hará uso de la tarjeta PicoBoard. En particular, empleará el sensor de resistencia eléctrica medible desde las pinzas de cocodrilo. En este juego se utilizarán cuatro frutas que, por su resistencia eléctrica y conectándolas mediante las pinzas, el programa reconocerá mostrando por pantalla la fruta escogida tal como muestran las Figuras 6.36 y 6.37.

Los conceptos sobre programación incluidos en este proyecto son:



Figura 6.36: Foto de la fruta

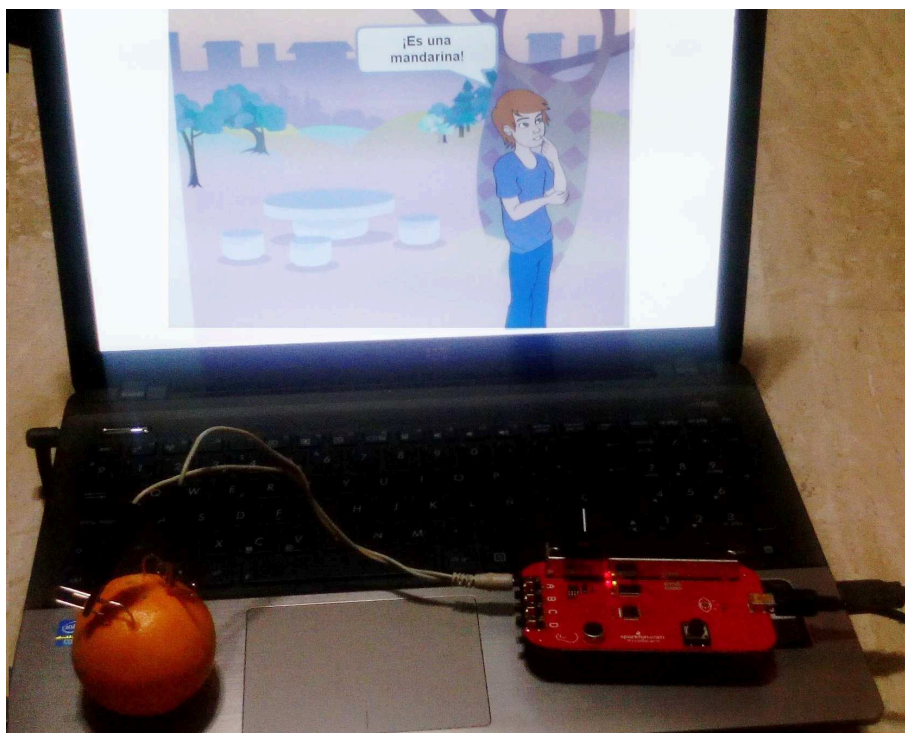


Figura 6.37: Mensaje con el nombre de la fruta

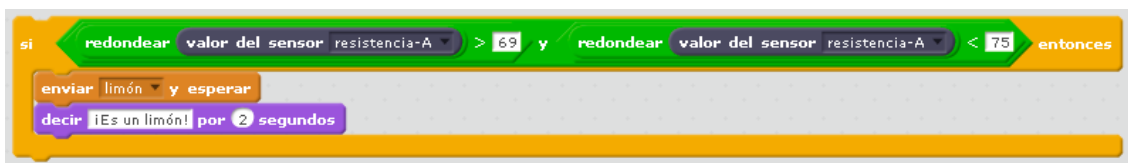


Figura 6.38: Código para adivinar el limón



Figura 6.39: Código del objeto limón

- **Interacción con la tarjeta PicoBoard.** En este caso haremos uso del sensor de resistencia y las pinzas de cocodrilo. Para poder medir la fruta, deberán conectarse las pinzas a un trozo de cable que será introducido dentro de la misma. Una vez nos devuelva el valor de su resistencia, el programa analizará dicho valor comparándolo entre una serie de rangos, ya que dependiendo de la distancia a la que se introduzcan las pinzas, este valor puede variar ligeramente, y finalmente enviará un mensaje a la fruta que corresponda. El código para implementar una de las comprobaciones se muestra en la Figura 6.38.
- **Paso de mensajes.** Una vez el objeto principal (en este caso, un niño) averigüe de qué fruta se trata, enviará una llamada al objeto fruta que corresponda. Una vez la reciba, deberá mostrarse por pantalla durante un tiempo y ocultarse de nuevo (véase la Figura 6.39).
- **Uso de condicionales y bucles.** En el proyecto vuelven a incluirse el uso de estos dos conceptos, ya que el objeto principal está constantemente ejecutándose, por lo que necesitará el uso de bucles y las instrucciones condicionales serán necesarias para poder determinar qué tipo de fruta se ha conectado con las pinzas. Además, se añade una condición extra que comprueba si las dos pinzas están tocándose para mostrar un texto por pantalla indicándolo tal como muestra la Figura 6.40.
- **Paralelismo.** Cada una de las frutas necesita ejecutar dos códigos de forma paralela, uno que las oculta nada más empezar el programa y otro que hace que se muestren si reciben el mensaje desde el objeto principal.
- **Programación de varios objetos.** En este proyecto cada uno de los objetos debe programarse por separado para que funcione correctamente. Por un

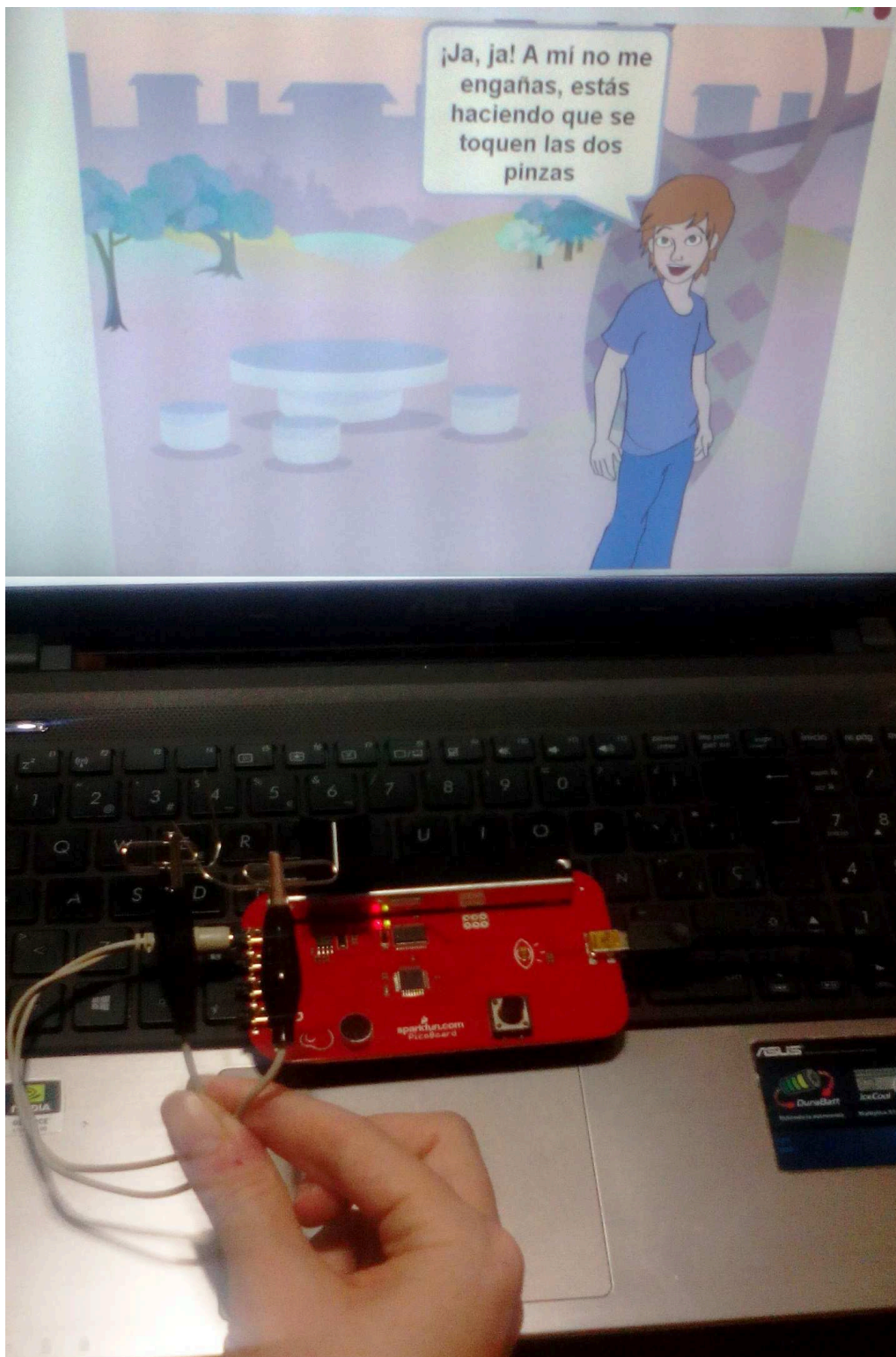


Figura 6.40: Programa en ejecución

lado el objeto principal que interactúa con el usuario y por el otro cada una de las frutas incluidas.

CAPÍTULO 7

Integración en el Museo de Informática

En el presente capítulo analizaremos la experiencia de incorporar este tipo de actividades educativas dentro del Museo de Informática. Para ello, se ha incorporado un taller de Scratch en donde alumnos de secundaria y bachiller realizan uno de los proyectos expuestos en este trabajo (en concreto el proyecto de ¡Atrápame si puedes!), el cual se ha adaptado al nivel de los asistentes.

A lo largo del capítulo se explicará los objetivos de este taller. Asimismo se expondrán los datos estadísticos recogidos del mismo, proporcionados en forma de encuesta, que cada uno de los asistentes ha rellenado al finalizarlo.

7.1 Oferta de la actividad

El Museo de Informática ha incorporado a su oferta de actividades en este curso académico 2014/15 los medios e infraestructura necesarios para realizar una actividad nueva llamada Taller de Scratch tales como su promoción mediante su página web como puede observarse en la Figura 7.1. En este taller se proponen actividades como las expuestas en este trabajo para jóvenes desde primaria hasta bachillerato que no hayan cursado anteriormente asignaturas de programación (véase la Figura 7.2).

7.2 Organización de la actividad

Durante las dos horas de duración de la actividad, se les introduce los conceptos y la importancia de la programación en la actualidad mediante un primer apartado de audiovisuales para acabar explicando el entorno de trabajo de Scratch. Finalmente, la actividad se centra en la resolución de un proyecto como



The screenshot shows the website for the Museo de Informática at the Universitat Politècnica de València. The header includes the museum's logo, the university name, and the text "etsinf". A navigation menu is located below the header. The main content area is titled "Taller de Scratch: información y protocolo a seguir". It features the Scratch logo, the Scratch cat character, and a graphic with the text "por siempre imagina programa comparte". Below this, there is a list of requirements for attending the workshop, including the need for a prior appointment, contact information, and details about the group size and programming knowledge. A small image of a museum display is also visible on the left side of the page.

museo informàtica UNIVERSITAT POLITÈCNICA DE VALÈNCIA etsinf Valencià | Castellano

¡Un museo dentro de una... tarjeta perforada!

INICIO EL MUSEO ACTIVIDADES HORARIOS Y VISITAS DIDÁCTICA SABER MÁS DONACIONES ENLACES CONTACTO

Taller de Scratch: información y protocolo a seguir

por siempre
imagina
programa
comparte

SCRATCH

Para el taller de Scratch hace falta solicitar **cita previa** por correo electrónico, especificando claramente **"Solicitud taller de Scratch – Nombre del centro u organización"** en el asunto del correo e incluyendo la siguiente información:

- Nombre del centro educativo u organización que efectúa la petición
- Responsable/s del grupo, teléfono de contacto y DNI de uno de los responsables (**IMPORTANTE a efectos de reserva**)
- Titulación (ESO, Bachillerato, Ciclos Formativos) y curso de los alumnos
- Número de alumnos, nombre, apellidos y DNI (**IMPORTANTE a efectos de reserva**)
- Idioma preferido (castellano/valenciano)
- Conocimientos de Scratch (el taller puede realizarse sin tener conocimientos previos de programación o de Scratch)

Los grupos están **limitados** a un máximo de **40** personas (2 personas por ordenador). El taller de Scratch dura aproximadamente **2 horas** e incluye una breve **presentación**, la **proyección** de un documental sobre **la importancia de saber programar en la actualidad**, y la explicación de la plataforma **Scratch**, una herramienta que permite de forma sencilla familiarizarse con la programación y su entorno. A su vez, se plantearán sencillas actividades que los alumnos podrán resolver utilizando **Scratch** como la **creación de juegos** o **animaciones**. El taller de **Scratch** comienza en el **Punto de Encuentro** situado justo en la puerta de entrada del edificio 1E. **No es necesario poseer conocimientos previos de programación o de la herramienta Scratch**. Sólo se necesita motivación y ganas de aprender.

Figura 7.1: Información del taller en la web del museo

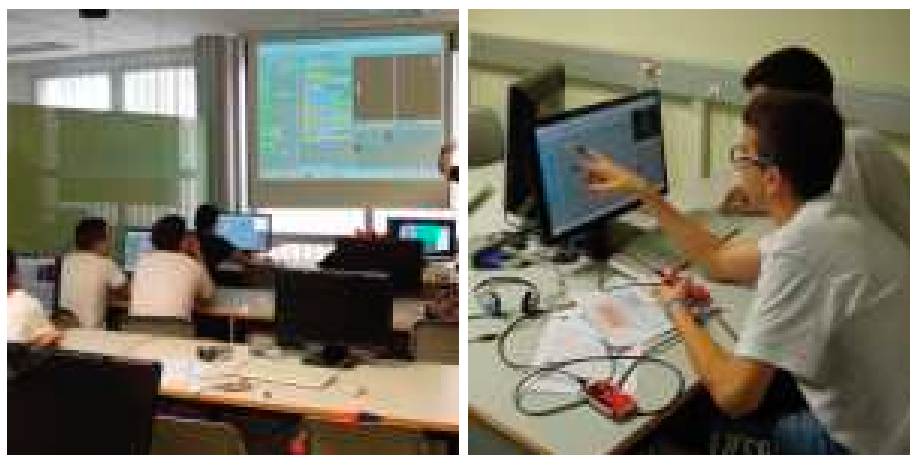


Figura 7.2: Taller de Scratch

los planteados en este trabajo que los alumnos programan por su cuenta, con una guía de ayuda y aportando la información didáctica necesaria para su elaboración mediante un documento escrito que se les entrega al principio de la actividad con el objetivo a seguir y los pasos necesarios así como los conceptos de programación con los que va relacionado. Parte de su contenido puede observarse en la Figura 7.3.

Durante su puesta en marcha en septiembre de 2014 hasta marzo de 2015, han pasado por este taller más de 450 alumnos¹ de alrededor de 20 instituciones educativas diferentes de la Comunidad Valenciana². De estas instituciones, el 77.5 % pertenecen a la provincia de Valencia, y el 22.5 % restante se divide entre Castellón y Alicante de forma equitativa (11.25 % para cada uno). Asimismo, de entre todos los grupos de asistentes, no se ha encontrado ninguno perteneciente a la educación primaria, quedando repartidos entre secundaria (un 47.4 %) y bachillerato (un 52.6 %).

Una vez finalizado el Taller de Scratch, a cada grupo que acude se le pide que rellene una encuesta como la de la Figura 7.4 a fin de obtener realimentación sobre lo que han aprendido y poder determinar si el objetivo del taller se ha conseguido.

Esta encuesta totalmente anónima se divide en tres bloques que agrupan todos los objetivos del Taller de Scratch y un apartado final de opinión de los asistentes que quieran aportar algún detalle que no esté incluido en ella. Estos bloques son los siguientes:

- **Organización del taller.** En esta parte se recoge el grado de satisfacción del asistente a la actividad en relación con la organización tanto de los recursos de infraestructura y de medios tecnológicos que ha dispuesto el museo así como con la duración general de la actividad.

¹462 alumnos totales a mes de marzo de 2015

²22 instituciones educativas totales a mes de marzo de 2015



Así pues, sabiendo esto, el movimiento de nuestro pez es muy fácil de hacer. Primero haremos doble clic sobre el pez para poder acceder a él. Al hacerlo nos aparecerá una pantalla como la siguiente:



Tal y como se muestra en la imagen, aparece dentro de **Eventos** la instrucción **al presionar tecla específica**. Si pulsamos el menú desplegable, podremos seleccionar la opción "flecha arriba" y arrastrarla al área gris de la derecha.

A continuación explicaremos los pasos para crear el movimiento hacia arriba con la flecha hacia arriba.



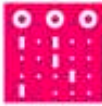
Al cuadrar estos dos bloques, el primero comprobará si se ha presionado la flecha hacia arriba mientras que el segundo (que puedes encontrar dentro de **Movimiento**) cambiará el valor de la Y por 5.

Actividad:

¿Eres capaz de programar el resto de movimientos? ¡Pruébalo!




Figura 7.3: Boletín del Taller de Scratch




**museo
informática**

¡Un museo dentro de una... tarjeta perforada!



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

etsinf



Encuesta sobre el Taller de Scratch

Organización del taller

| | ☹️ | 😊 | 😄 |
|--|----|---|---|
| La forma en que se ha organizado la actividad me parece adecuada | | | |
| Los recursos informáticos usados para llevar a cabo la actividad son adecuadas | | | |
| La duración me parece ajustada a los objetivos de la actividad | | | |

Desarrollo de la actividad

| | ☹️ | 😊 | 😄 |
|---|----|---|---|
| Las explicaciones de la persona que ha presentado y dirigido la actividad han sido de gran utilidad | | | |
| El video "Todo el mundo debería saber programar" me ha resultado atractivo y divulgativo | | | |
| Los ejemplos estudiados han resultado interesantes y proporcionados para iniciarse en la programación | | | |
| El boletín que describe los ejemplos de la actividad está bien estructurado y resulta de ayuda | | | |
| Creo que esta actividad ha servido para ilustrar el concepto de "pensamiento computacional" (<i>computational thinking</i>) | | | |

Valoración global

| | ☹️ | 😊 | 😄 |
|---|----|---|---|
| Considero que la actividad me ha servido de ayuda y motivación para estudiar asignaturas sobre programación | | | |
| Pienso que el "pensamiento computacional" (<i>computational thinking</i>) es una destreza indispensable para el siglo XXI | | | |
| Recomendaría esta experiencia educativa a personas sin conocimientos sobre programación | | | |

Tu opinión nos interesa

¡Muchas gracias por tu colaboración!

Figura 7.4: Encuesta sobre el Taller de Scratch del Museo de Informática

- **Desarrollo de la actividad.** En este apartado el asistente evalúa si las explicaciones dadas, la ayuda aportada así como los ejemplos y actividades realizados mediante el boletín que se le ha facilitado al inicio de la misma son útiles y le han servido de ayuda para ilustrar el concepto de pensamiento computacional y afianzar una base inicial en programación.
- **Valoración global.** Finalmente, en este bloque se recogen las valoraciones sobre los objetivos principales de la actividad: incitar a que los alumnos se inicien en la programación, concienciar sobre la importancia del pensamiento computacional en la actualidad y averiguar si la actividad es capaz de cumplir su objetivo de enseñar bases de programación a personas que no han programado nunca.
- **Tu opinión nos interesa.** Este último apartado se añade para recoger aquellas opiniones, críticas o aspectos que el asistente al taller quiera resaltar y que no se encuentren plasmados en la encuesta.

7.3 Resultados obtenidos

Mediante la recogida de información por parte de los asistentes al Taller de Scratch que ofrece el Museo de Informática, se han realizado una serie de tablas y gráficos estadísticos para comprobar si los objetivos de las actividades y el tipo de proyectos propuestos en el presente trabajo son adecuados para los alumnos que asisten, basándose en el estado actual de la educación, los resultados obtenidos y las opiniones de los asistentes.

7.3.1. Organización del taller

Los datos globales obtenidos de entre todas las encuestas recogidas, se han analizado diferenciando entre los tres posibles grupos. Para cada una de las secciones (organización global, desarrollo de la actividad, y valoración global) se ha obtenido el porcentaje global de satisfacción dividiendo el total de encuestas contestadas por la puntuación que han obtenido cada uno de los apartados, es decir, el número de esas encuestas que han estado de acuerdo, en desacuerdo o indiferentes respectivamente. Finalmente se ha obtenido como indica la Figura 7.5, que un 65% de los alumnos que han realizado el taller han encontrado que la organización era correcta frente a solo un 2% que le ha parecido que no estaba bien organizado. El 33% restante cree que la organización de la actividad no ha sido ni buena ni mala. Así pues, más de la mitad de los encuestados ha estado de acuerdo con la organización del taller que el Museo de Informática ha llevado a cabo.

Analizando cada uno de los objetivos a conseguir dentro del desarrollo de la actividad de forma más detallada tal como aparece reflejado en la Figura 7.6, cabe



Figura 7.5: Grado de satisfacción global con la organización del taller

| | En desacuerdo | Indiferente | De acuerdo |
|--|---------------|-------------|------------|
| Organización del taller | | | |
| La forma en la que se ha organizado la actividad me parece adecuada | 1% | 11% | 88% |
| Los recursos informáticos usados para llevar a cabo la actividad son adecuadas | 1% | 23% | 76% |
| La duración me parece ajustada a los objetivos de la actividad | 4% | 32% | 64% |

Figura 7.6: Grado de satisfacción detallado con la organización del taller

destacar que un 88 % ha encontrado adecuada la organización de la actividad frente solo a un 1 % que no la ha encontrado bien organizada y un 11 % que le ha parecido indiferente. Dentro de los recursos utilizados, sigue habiendo una mayoría de asistentes satisfechos con un 76 % frente al 1 % que no ha estado de acuerdo y un 23 % que no está ni de acuerdo ni en desacuerdo. Finalmente, y relativo a la duración de la actividad, sigue destacando el porcentaje que está de acuerdo, sitúandose en un 64 % frente a un 4 % que no ha estado de acuerdo y un 32 % que no le ha parecido ni bien ni mal.

7.3.2. Desarrollo de la actividad

En relación con el desarrollo de la actividad y tal como muestra de forma global el gráfico de la Figura 7.7, un 75 % de los asistentes ha considerado que la actividad es positiva y está de acuerdo con ella frente a un 2 % que no lo ha estado y un 22 % que no se decanta ni por uno ni por otro. Esto significa que un



Figura 7.7: Grado de satisfacción global con el desarrollo de la actividad

promedio de siete de cada diez asistentes han encontrado la actividad útil y han aprendido con los proyectos planteados en ella y la ayuda ofrecida por el boletín.

A su vez, tal y como nos muestra la Figura 7.8, el porcentaje de asistentes que ha aceptado positivamente la actividad sigue siendo muy superior a los que opinan lo contrario. Por ejemplo el 73 % piensa que los proyectos con Scratch propuestos han resultado útiles frente al 3 % al que no le ha resultado útil o el 72 % que cree que esta actividad sirve para ilustrar el concepto de pensamiento computacional frente al 2 % que opina que no lo ilustra.

7.3.3. Valoración global

Finalmente, de forma global la valoración general que se tiene sobre el Taller de Scratch ofrecido por el Museo de Informática es altamente positiva con 7 de cada 10 asistentes aproximadamente que así lo valora.

Tal como muestra la Figura 7.9, un 73 % valoran positivamente la actividad frente a un 3 % que opina lo contrario y un 24 % a los cuales les ha parecido indiferente.

Cabe destacar, como puede verse en la Figura 7.10, que más de las mitad de asistentes, en concreto un 64 %, considera que la actividad del Taller de Scratch les ha servido como motivante y ayuda para estudiar en un futuro asignaturas sobre programación, así como el 73 % que recomendaría la actividad a personas sin conocimientos previos en esta memoria.

| | En desacuerdo | Indiferente | De acuerdo |
|---|----------------------|--------------------|-------------------|
| Desarrollo de la actividad | | | |
| Las explicaciones de la persona que ha presentado y dirigido la actividad han sido de gran utilidad | 2% | 12% | 86% |
| El vídeo "Todo el mundo debería saber programar" me ha resultado atractivo y divulgativo | 2% | 24% | 74% |
| Los ejemplos estudiados han resultado interesantes y proporcionados para iniciarse en la programación | 3% | 24% | 73% |
| El boletín que describe los ejemplos de la actividad está bien estructurado y resulta de ayuda | 3% | 26% | 71% |
| Creo que esta actividad ha servido para ilustrar el concepto de "pensamiento computacional" (<i>computational thinking</i>) | 2% | 26% | 72% |

Figura 7.8: Grado de satisfacción detallado con el desarrollo de la actividad

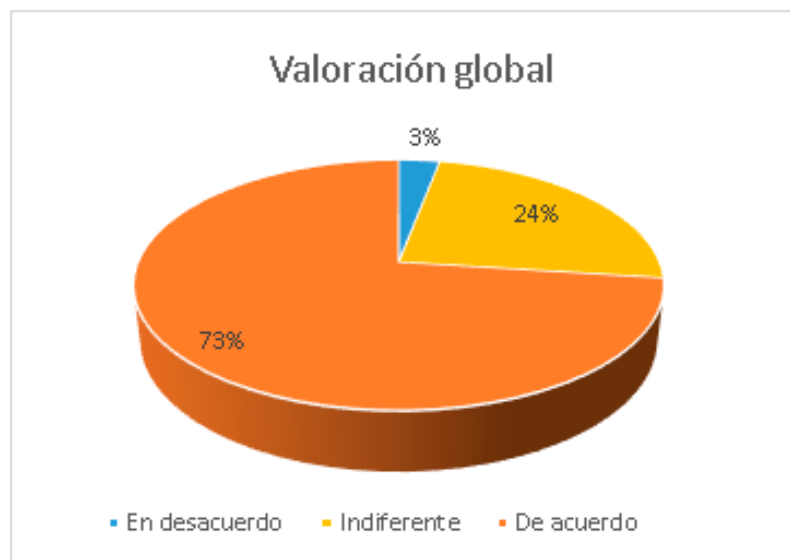


Figura 7.9: Grado de satisfacción general

| | En desacuerdo | Indiferente | De acuerdo |
|---|--------------------------|--------------------|-----------------------|
| Valoración global | | | |
| Considero que la actividad me ha servido de ayuda y motivación para estudiar asignaturas sobre programación | 5% | 31% | 64% |
| Pienso que el "pensamiento computacional" (<i>computational thinking</i>) es una destreza indispensable para el siglo XXI | 2% | 27% | 71% |
| Recomendaría esta experiencia educativa a personas sin conocimientos sobre programación | 3% | 24% | 73% |

Figura 7.10: Grado de satisfacción general detallado

7.3.4. Tu opinión nos interesa

Dentro del apartado de opiniones destacan mayormente las positivas, en las que los asistentes resaltan que la actividad les resulta divertida y aprenden mucho. También se sorprenden al ver que la programación es algo más sencillo de lo que tenían en mente. Finalmente, como aspecto negativo, suelen mencionar la duración del taller, la mayoría porque se les ha pasado demasiado rápido, aunque también existen asistentes a los que se les ha hecho demasiado largo.

7.4 Conclusiones

A partir de los datos obtenidos y analizados, puede concluirse que, de manera global, el Taller de Scratch del Museo de Informática es una actividad que desarrolla con éxito cada uno de sus objetivos. Además, es un gran método de difusión de la importancia de la programación, tal como lo demuestra la gran cifra de asistentes que han pasado en los apenas ocho meses en que lleva realizándose. Igualmente, la gran mayoría de estos asistentes, se divierten mientras aprenden conceptos de programación, cumpliendo así con los objetivos de la actividad.

CAPÍTULO 8

Conclusiones y trabajo futuro

En este último capítulo se plantean unas consideraciones finales obtenidas a partir del trabajo desarrollado, realizando un resumen final y analizando si los objetivos propuestos al principio del mismo se han cumplido. Finalmente, se aporta una serie de ideas que podrían ser útiles para futuros trabajos que usaran el presente como base para su desarrollo.

8.1 Consideraciones finales

Mediante el presente trabajo se ha intentado crear un abanico de actividades que abarquen todo el sistema educativo español anterior a la universidad y que puedan desarrollarse dentro del Taller de Scratch que ofrece el Museo de Informática de la Escola Tècnica Superior d'Enginyeria Informàtica. Para ello se necesitaba consultar, por una parte, el nivel que los alumnos de nuestro sistema educativo tienen en cada una de sus etapas de aprendizaje sobre esta materia así como indagar sobre la duración adecuada de la actividad y los medios disponibles para ella.

En la realización de este trabajo, se ha puesto especial énfasis en la programación como arma creativa en un mundo totalmente tecnológico y en el componente de motivación a la hora de aprender a dominarla. Para ello, se han consultado tanto libros como material audiovisual que han aportado una idea concreta sobre la situación tecnológica de la sociedad en la que vivimos así como la experiencia que tienen los jóvenes ante esta tecnología que, como se ha demostrado es desproporcionada ya que viven en un mundo totalmente conectado pero no son capaces de escribir en él y adaptarlo a sus necesidades mediante el uso de la programación.

Tal y como se ha puesto en evidencia, la escasa formación en materias de programación está presente dentro de nuestro sistema educativo actual por lo que sigue habiendo un vacío entre la importancia de dominar esta habilidad y su integración en escuelas e institutos.

Para llevar a cabo la creación de dichas actividades didácticas, se ha optado por la plataforma Scratch y su entorno de trabajo, y ha quedado demostrado que, por la gran variedad de información y bibliografía que posee, por la actividad constante que dedican sus desarrolladores en mejorar el proyecto y por su compatibilidad con todos los sistemas operativos, incluyendo una versión en línea, además de por disponer de gran variedad de periféricos compatibles, ha resultado ser, en nuestra opinión, la elección correcta. Además, el componente social de la plataforma, añade aún más potencial a esta herramienta, ya que consigue que sus usuarios se formen mejor y se motiven más, compartan sus experiencias con otros usuarios, aprendan de ellos al ver sus creaciones, o colaboren entre ellos para crear sus proyectos gracias al sitio web que Scratch posee.

Por ello el presente trabajo, con su propuesta de actividades, representa una buena opción para empezar a introducir una base en programación y difundir su importancia en la gente más joven. Asimismo, elimina viejas ideas sobre esta habilidad, como la de que se trata de una materia muy compleja que necesita mucha formación anterior, que programar es algo que una persona crea aislada, sin tener contacto con un grupo, ni compartir ideas y conocimiento, o que programar con periféricos externos, como puede ser una cámara de vídeo o un micrófono, es algo muy complejo y difícil.

Según los resultados obtenidos, puede afirmarse que las actividades que se han propuesto en este trabajo, resultarán muy útiles una vez se incorporen definitivamente dentro del Taller de Scratch del museo, ya que se basan en la experiencia que la actividad ha tenido hasta el momento (8 meses) y que ha conllevado un alto grado de aceptación por parte de los alumnos asistentes y les ha motivado, en la gran mayoría de los casos, un impulso por aprender a programar así como a concienciarse de la importancia de esta habilidad mientras se divierten. A su vez, la difusión de la actividad utilizando la plataforma del Museo de Informática ha sido un éxito ya que como ha quedado demostrado, han pasado durante el poco tiempo que lleva operativa más de 450 alumnos de diferentes centros y niveles.

Finalmente, creemos que el presente trabajo puede resultar útil no solo dentro del ámbito del Museo de Informática, sino de cara a cualquier docente que quiera iniciar a sus alumnos en programación, ya que las actividades propuestas parten de la base de que no hay un conocimiento previo y pueden ser realizadas en cualquier aula informática donde se instale la plataforma Scratch, totalmente gratuita, o que tenga conexión a internet para poder acceder a ella desde su página web.

8.2 Trabajo futuro

Como trabajo futuro a llevar a cabo a partir de este proyecto, podría encontrarse la ampliación de las actividades, haciéndolas más complejas e incluso de más de una sesión para que el profesor pudiera continuarlas desde el aula de clase. También sería interesante, una vez se llevasen a cabo definitivamente, recoger

datos de los asistentes para ajustar las actividades en caso de que algún concepto no quedara claro o tuvieran que remodelarse. A su vez, podrían crearse actividades más complejas para asistentes que tengan un conocimiento más avanzado de programación, como puede ser por ejemplo, alumnos de ciclos formativos, de primeros cursos universitarios, o grupos de alumnos que ya hayan realizado anteriormente el taller.

Finalmente, podría modificarse el apartado de la web del Museo de Informática relativo a este taller, para incluir, en cada uno de los niveles educativos, las actividades propuestas y que pudiera ser el profesor o el responsable del grupo de asistentes, el que eligiera la actividad que más le interesara según los conceptos de programación que quisiera que se estudiaran, o los periféricos que quisiera que se usaran.

Bibliografía

- [1] Jessica Chiang y Sergio van Pul. *Scratch 2.0 Game Development HOTSHOT*. PACKT Publishing, United Kingdom, 2014.
- [2] Ines Dussel y Luis Alberto Quevedo. *Educación y nuevas tecnologías: los desafíos pedagógicos ante el mundo digital*. Fundación Santillana, Buenos Aires, 2010. <<http://www.virtualeduca.org/ifd/pdf/ines-dussel.pdf>>[Consulta: marzo de 2015]
- [3] Sergio Garrido. *Scratch para niños... y no tan niños*. Amazon Media EU S.à.r.l., Francia, 2015.
- [4] Arturo Gómez, Eva Parramón, Teófilo Antúnez y Eutiquiano Merino. *Informática, 4º E.S.O.*. Editorial Donostiarra, San Sebastián, 2013.
- [5] Arturo Gómez, Eva Parramón y Teófilo Antúnez. *Tecnologías de la Información y la Comunicación, bachillerato*. Editorial Donostiarra, San Sebastián, 2013.
- [6] *Guía de referencia de Scratch 2.0*. EDUTEKA, Colombia, 2013. <<http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf>>[Consulta: marzo de 2015]
- [7] Lev Manovich. *El lenguaje de los nuevos medios de comunicación. La imagen en la era digital*. Paidós Comunicación, Barcelona, 2006.
- [8] Seymour Papert. An Exploration in the Space of Mathematics Education. *International Journal of Computers for Mathematical Learning*, vol. 1, núm. 1, pp. 95-123, 1996.
- [9] Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Nueva York, segunda edición en 1994.
- [10] Majed Marji. *Learn to Program with Scratch: a visual introduction to programming with games, art, science and math*. no starch press, San Francisco, 2014.
- [11] Mark Prensky. *Digital natives, digital immigrants*. On the Horizon, MCB University Press, 2001.

-
- [12] Máximo Prudencio. Una herramienta lúdica de iniciación a la programación, SCRATCH. *Linux Magazine*, núm. 28, pp. 78-82, 2013.
- [13] Mitchel Resnick, Karen Brennan. New frameworks for studying and assessing the development of computational thinking. *Proceedings of the American Educational Research Association (AERA) annual conference*, pp. 1-25, 2012.
- [14] Mitchel Resnick, John Maloney, Andrés Monroy Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman y Yasmin Kafai. Scratch: Programming for all. *Communications of the ACM*, vol. 52, núm. 11, pp. 60-67, 2009.
- [15] Mitchel Resnick. *Lifelong Kindergarten*. Cultures of Creativity, LEGO Foundation, 2013. <<http://web.media.mit.edu/~mres/papers/CulturesCreativityEssay.pdf>>[Consulta: marzo de 2015]
- [16] Sean McManus. *Scratch Programming*. Easy Steps, United Kingdom, 2013.
- [17] Gabriel Serrano. *Programación Scratch para niños*. Amazon Media EU S.à.r.l., Francia, 2015.
- [18] The LEAD Project. *Super Scratch Programming Adventure!: Learn to Program By Making Cool Games*. Hong Kong Federation of Youth Groups, Canadá, 2010.
- [19] Jeannette M. Wing. Computational Thinking. Viewpoint. *Communications of the ACM*, vol. 49, núm. 3, pp. 33-35, marzo 2006.