



Universidad de Valladolid

TRABAJO FIN DE MÁSTER

MÁSTER EN PROFESOR DE EDUCACIÓN
SECUNDARIA OBLIGATORIA Y
BACHILLERATO, FORMACIÓN PROFESIONAL
Y ENSEÑANZAS DE IDIOMAS

Especialidad de Tecnología e Informática

**METODOLOGÍA Y TECNOLOGÍA DOCENTE
PARA LA ENSEÑANZA DE
PROGRAMACIÓN**

**EDUCATIONAL METHODOLOGY AND
TECHNOLOGY FOR PROGRAMMING
TEACHING**

Autor:

D. Jose Antonio Da Rocha Padilla

Tutor:

Dña. Carmen Hernández Díez

Valladolid, Diciembre de 2019

*A mi familia
y amigos,
y para mi futura persona favorita.
Gracias.*

Agradecimientos

*No fracasé, sólo descubrí 999 maneras de cómo
no hacer una bombilla.*

Thomas Alva Edison

Antonio Garrigues decía que “*Los discursos largos estarán muy pronto tipificados como delito en el código penal*”, por lo que intentaré no extenderme demasiado con los agradecimientos a las personas que me han ayudado con este trabajo.

Parafraseando a *Marcus Aurelius Antoninus*, ponerse manos a la obra es tener ya la mitad del trabajo hecho. Y en este sentido, debo estar muy agradecido a mi tutor del *practicum* por su ayuda y acompañamiento durante toda mi estancia en el IES Ribera de Castilla, que me lo hizo todo mucho más fácil. Dicho esto, muchas gracias, Gustavo.

También quiero dar las gracias a mis tutores de este trabajo, Santiago y Carmen, por la paciencia que han tenido conmigo y la ayuda que me han brindado con documentación que me ha facilitado mucho el trabajo.

De la misma forma, quiero agradecer a muchas personas (algunas anónimas y otras no tanto), que nunca serán conscientes del nacimiento de este trabajo, por la ayuda que me han aportado con los artículos publicados que me han servido de base para este trabajo.

Como no podía ser de otra forma, quería dar las gracias a mis compañeros de mastergobios por los buenos ratos que hemos compartido en los últimos 8 meses y de los que tanto he aprendido.

Y como los últimos serán los primeros, dejo para el final esas personas que están siempre detrás, sin dejarse notar, pero que sin ellos, todo sería más complicado. Gracias familia y, en especial, Lorena, por aguantarme (o no aguantarme en mi ausencia) este último año.

Resumen

La mayoría de los buenos programadores programan no porque esperan que les pagen o que el público los adore, sino porque programar es divertido.

Revista Grid de la IEEE (Instituto de Ingenieros Electricistas y Electrónicos)

Esta memoria refleja el producto del trabajo final del Máster en profesor de Educación Secundaria Obligatoria y bachillerato, formación profesional y enseñanzas de idiomas con especialidad Informática y Tecnología de la Universidad de Valladolid.

A lo largo del trabajo se presenta el concepto de **pensamiento computacional** y se justifica su importancia en la sociedad actual; a continuación se presentan algunas iniciativas existentes en la educación no formal. En el siguiente punto, ya se analiza la situación actual del sistema educativo europeo y se pone de ejemplo el enfoque de algunos países como Reino Unido o Finlandia, antes de analizar la situación actual española. Para finalizar, se han mostrado algunas herramientas con las que acercar el pensamiento computacional a los alumnos, como **Scratch** o **App Inventor** y se han presentado dos propuestas de actividades a realizar en secundaria.

La distribución de la presente memoria es la siguiente: En los capítulos 2 y 3 se presenta el pensamiento computacional y su importancia. En el capítulo 4 se presentan las iniciativas existentes dentro de la educación no formal. En el capítulo 5 se analiza la educación formal y su situación actual en Europa. En el capítulo 6 se presentan dos actividades para desarrollar dos contenidos de dos materias diferentes a través de las nuevas tecnologías. Finalmente, se exponen las conclusiones y la bibliografía.

Abstract

*The computer was born to solve problems that
did not exist before.*

Bill Gates

This paper is the product of the final work of the Master in Compulsory Secondary Education and High School, professional training and languages with a specialization in Computer Science and Technology from the University of Valladolid.

Throughout this paper the concept of **computational thinking** is presented and its importance in today's society is justified; after that, there are some existing initiatives in non-formal education. In the next point, current situation of the European education system is analyzed and the approach of some countries such as the United Kingdom or Finland is set as an example, before analyzing the current Spanish situation. Finally, some tools are shown to bring computational thinking to students, such as **Scratch** or **App Inventor** and two activities are proposed to be carried out in secondary school.

The content is distributed is as follows: In chapters 2 and 3, computational thinking and its importance are presented. Chapter 4 presents the existing initiatives within non-formal education. Chapter 5 analyses how formal education is and its current situation in Europe. In chapter 6, two activities are presented to develop contents from two different subjects through new technologies. Finally, the conclusions and the bibliography are presented.

Índice

Agradecimientos	v
Resumen	vi
Abstract	vii
1. Introducción y Objetivos	1
1.1. Contexto	1
1.2. Objetivos	2
En el próximo capítulo	3
2. El Pensamiento Computacional	4
2.1. Habilidades y actitudes	5
En el próximo capítulo	6
3. Importancia del pensamiento computacional	7
En el próximo capítulo	8
4. Educación no formal: Iniciativas existentes	9
4.1. Mundiales	9
4.1.1. CoderDojo	9
4.1.2. Code.org	9
4.1.3. Bebras	10
4.1.4. CS Unplugged	10
4.1.5. Code Club	10
4.1.6. Made with Code	10
4.1.7. Olimpiada Informática Internacional	10
4.1.8. Google Code Jam	10
4.1.9. Microsoft Hacking STEM	10
4.1.10. Microsoft MakeCode	11
4.2. Europeas	11
4.2.1. Semana de la Programación de la UE	11

4.2.2.	All you need is code	11
4.2.3.	Barefoot Computing	11
4.2.4.	Computing At School	12
4.2.5.	Code it Like a Girl	12
4.3.	Españolas	12
4.3.1.	Programamos	12
4.3.2.	Grupo KGBL3 - Dr. Scratch	13
4.3.3.	Olimpiada Informática Española	13
4.3.4.	Creando Código - ScolaTIC	13
4.3.5.	CLOQQ	13
4.4.	Castilla y León	14
4.4.1.	La escuela TIC	14
4.4.2.	Crea y Explora	14
	En el próximo capítulo	15
5.	Educación Formal: Situación actual	16
5.1.	Caso de Reino Unido	17
5.2.	Caso de Finlandia	18
5.3.	Currículo actual en España	18
5.3.1.	Materias troncales	18
5.3.2.	Materias de libre configuración autonómica	19
	En el próximo capítulo	20
6.	Propuesta	21
6.1.	Enfoque	21
6.2.	Herramientas	22
6.2.1.	Herramientas basadas en bloques	22
6.2.2.	Herramientas basadas en texto	25
6.2.3.	Scratch	26
6.2.4.	App Inventor	29
6.2.5.	Comparativa	30
6.3.	Actividades	30
6.3.1.	El cuerpo humano	30
6.3.2.	El Teorema de Pitágoras	38
7.	Conclusiones	44
	Referencias	46
	Índice alfabético	50
	Glosario de acrónimos	53

Índice de figuras

3.1. Tendencia de la búsqueda del concepto Pensamiento Computacional en el periodo 2014 - 2019 (Google Trends)	8
5.1. Distribución de los países según el proceso de revisión del currículo.	17
6.1. Ejemplo de estructura de un código realizado en una herramienta basada en bloques.	22
6.2. Blockly.	23
6.3. Scratch.	23
6.4. App Inventor.	23
6.5. Alice.	23
6.6. Python.	25
6.7. Scratch.	26
6.8. Ejemplo de un código desarrollado con Scratch.	27
6.13. Variables booleanas en Scratch.	28
6.14. Hilos en Scratch.	28
6.15. Eventos en Scratch.	28
6.9. Instrucciones en Scratch.	28
6.10. Bucles en Scratch.	28
6.11. Condiciones en Scratch.	28
6.12. Variables en Scratch.	28
6.16. App Inventor.	29
6.17. Fases en el desarrollo de una proyecto con App Inventor.	29
6.18. Ejemplo de un código desarrollado con App Inventor.	30
6.19. Visualización de la solución realizada (inicio).	35
6.20. Visualización de la solución realizada (fin).	35
6.21. Ejemplo solución para el comportamiento del hígado.	36
6.22. Ejemplo solución para el comportamiento del intestino grueso.	36
6.23. Ejemplo solución para el comportamiento del intestino delgado.	36
6.24. Ejemplo solución para el comportamiento del estómago.	37
6.25. Ejemplo solución para el comportamiento del esófago.	37

6.26. Ejemplo solución para el comportamiento de la boca y las glándulas salivales.	37
6.27. Ejemplo solución para el comportamiento del páncreas.	37
6.28. Ejemplo visual de solución para el ejercicio con App Inventor.	42
6.29. Ejemplo de solución para la aplicación del Teorema de Pitágoras.	43

Índice de Tablas

6.1. Ejemplos de herramientas basadas en bloques.	24
6.2. Ejemplos de herramientas basadas en texto.	25
6.3. Ejemplos de herramientas basadas en texto (cont.).	26
6.4. Temporalización de la actividad El cuerpo humano.	34
6.5. Temporalización de la actividad Teorema de Pitágoras.	41

Capítulo 1

Introducción y Objetivos

Controlar la complejidad es la esencia de la programación

Brian Kernigan

RESUMEN: Este capítulo presenta al lector las bases del trabajo, mostrándole el contexto en el que se encuentra y los objetivos que se plantean.

1.1. Contexto

El ritmo con el que avanza la sociedad cada vez es mayor y, por ello, las nuevas generaciones necesitan estar preparadas para adaptarse a conceptos y herramientas que están aún por descubrirse. Para ello, necesitan una serie de habilidades que les permitan desenvolverse ante estos nuevos retos que les aparezcan donde el PC (*Computational Thinking* - Pensamiento Computacional) y la programación se ven como un camino para el desarrollo de estas habilidades.

En un principio puede pensar que el PC, y la informática, es una materia propia para quienes se van a dedicar a emplear ordenadores, pero esta percepción dista mucho de la realidad. El PC va de buscar soluciones a problemas reales. Y también va de interactuar con las máquinas para darles la inteligencia para resolver estos problemas.

Fue en los años ochenta cuando se introdujeron los primeros lenguajes de programación en la educación (véase Logo), principalmente en Estados Unidos, pero es en la actualidad cuando se ha impulsado sobremanera el desarrollo del PC y de la programación como herramienta para desarrollar aplicaciones móviles o páginas web, entre otras. A partir de ello, se han demostrado los beneficios cognitivos que aporta el hecho de aprender a pensar de esta forma.

El PC conlleva, por un lado, una forma determinada de pensar, analizar, organizar y relacionar ideas con el fin de resolver retos y, por otro, la identificación y resolución

de errores que se puedan encontrar en dicho proceso. Todo ello requiere de un elevado grado de comprensión del problema de partida, de manera que el alumno sea capaz de convertir su idea en instrucciones entendibles por el sistema informático.

El aprendizaje del PC aporta importantes beneficios al desarrollo del razonamiento lógico, el pensamiento abstracto y algorítmico y la resolución de problemas complejos, lo cual a su vez prepara para solventar otras problemáticas presentes en la vida real.

Por otro lado, que un problema pueda tener múltiples soluciones, incentiva que el alumno quiera experimentar y fomenta su autonomía, creatividad y motivación.

En lo relacionado con el aspecto económico-laboral, desde hace varios años existe una fuerte demanda de programadores, lo cual vincula el aprendizaje de la programación informática con una clara oportunidad de empleo.

Además, la presencia de los contenidos de programación en el ámbito educativo desde edades tempranas es importante con el fin de evitar estereotipos de género en relación con las carreras STEM (*Science, Technology, Engineering and Mathematics* - Ciencias Naturales, Tecnología y Matemáticas).

Sin embargo, en España aún no se ha dado el paso para apostar por estas disciplinas, apenas aparecen en el currículo actual, como podrá verse más adelante, y cuando aparece se limita a asignaturas optativas, permitiendo que en España un estudiante pueda cursar toda la educación obligatoria sin haber recibido ningún tipo de formación en relación a estas disciplinas tan importantes en los años venideros (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, 2018).

En 2015, sólo un 16 % de los estudiantes de secundaria afirmaban utilizar sus dispositivos electrónicos para programar, y menos del 25 % conocían algún lenguaje de programación (Fundación Española para la Ciencia y la Tecnología, Google y Everis, 2016).

Por otro lado, según este mismo informe, se puede afirmar que la mayoría de alumnos mostraban interés en estudiar estas materias, por lo que es muy recomendable promover iniciativas que acerquen estos contenidos a los estudiantes y faciliten que estos se inicien en los mismos.

1.2. Objetivos

El objetivo principal de este trabajo es **extraer una serie de propuestas para que el PC llegue de una mejor forma a las aulas**. Para conseguirlo, se han definido una serie de objetivos parciales que serán los que sirvan de guía en el camino.

El primero de los objetivos será *ilustrar la importancia de la programación y, sobre todo, del PC en secundaria*.

Una vez se conozcan estos, el segundo objetivo será *describir el currículo actual en lo relativo a la didáctica de contenidos relacionados con el PC y la programación*.

Realizado este análisis, el último objetivo será ya la *definición de las propuestas con las que intentar acercar el PC y la programación a los alumnos de secundaria*.

Una parte importante de estas recomendaciones estará en la **metodología** con la

que se transmitan los contenidos, que deberá poner especial énfasis en que el alumnado entienda la filosofía y los conceptos en los que se basa la programación, sus ventajas y herramientas de trabajo. Para ello, debe tener un carácter muy práctico y atractivo, basado en las nuevas tecnologías con las que atraer al alumnado.

En el próximo capítulo...

En este capítulo se ha centrado al lector en el ámbito de este trabajo y los objetivos del mismo.

En el siguiente capítulo se va a explicar en qué consiste esta disciplina para, posteriormente, justificar su importancia en la sociedad actual.

Capítulo 2

El Pensamiento Computacional

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

Steve Jobs

RESUMEN: Este capítulo presenta al lector las bases del trabajo, mostrándole la definición de Programación, qué es la programación y la importancia que tiene, así como una justificación de la necesidad de modificar la metodología empleada para impartir el currículo y, con ello, conseguir llegar mejor a los alumnos.

Wing definió el PC como una forma de pensamiento que implica la formulación de un problema y su solución de modo que esta última sea representada para que un sistema de información pueda llevarla a cabo de manera eficiente (Wing, 2006).

Asimismo, Wing insiste en que el PC es una actitud y una habilidad que todo el mundo, no sólo programadores, es necesario que tenga.

A partir de esta definición se extrajeron dos aspectos clave:

- El PC es una forma de pensamiento, independiente de la tecnología.
- El PC implica múltiples habilidades para diseñar una solución que debe ser ejecutada por un ordenador, una persona o una combinación de ambas.

Con ello, se creó un gran debate mundial sobre la necesidad de llevar el PC a las aulas y un gran número de investigaciones aparecieron al respecto creando nuevas definiciones.

En 2009, la CSTA (*Computer Science Teachers Association*) y la ISTE (*International Society for Technology in Education*) definieron el PC en una serie de características mínimas del mismo:

- Formula un problema de una forma que permite usar un ordenador u otras herramientas para resolverlo.

- Organiza y analiza la información de manera lógica.
- Representa los datos a través de abstracciones como modelos y simulaciones.
- Es capaz de automatizar las soluciones a través del pensamiento algorítmico en una serie de pasos.
- Identifica, analiza e implementa posibles soluciones con el objetivo de conseguir la combinación de pasos y recursos más eficiente y efectiva.
- Permite generalizar y transferir este proceso de resolución del problema a otro tipo de problemas.

Por otro lado, la *Royal Society* británica definió en 2012 el PC como el proceso de reconocer los aspectos de la computación en el mundo que nos rodea y aplicar las herramientas y técnicas de las CS (*Computing Science*) para comprender los sistemas y procesos naturales y artificiales.

En 2016, la CSTA publicó lo que definió como los estándares de las CS hasta el nivel K-12, donde definió el PC como la metodología de resolución de problemas que expande el ámbito de las CS a todas las disciplinas, ofreciendo una forma distinta de analizar y desarrollar soluciones a problemas que se pueden resolver computacionalmente.

Esta definición pone el foco en la abstracción, automatización y análisis, dejando al PC como núcleo de una disciplina más amplia que son las CS.

2.1. Habilidades y actitudes

Existe una gran cantidad de investigaciones acerca de las ventajas de introducir el PC en los sistemas educativos. En general, todos ellos comparten que esta forma de pensar permite pensar de una manera diferente para resolver los problemas del día a día, analizarlos desde una perspectiva diferente, desarrollar la capacidad de descubrir y crear e innovar o comprender lo que ofrece la tecnología (Lee et al., 2011) (Allan et al., 2010).

Se puede extraer que el PC implica seis habilidades:

- **Abstracción:** Proceso de eliminar los detalles innecesarios para conseguir un proceso más fácil de entender. Es importante ser capaz de abstraerse y elegir el foco principal adecuadamente, simplificando el problema y sin perder lo que sea importante (Csizmadia et al., 2015).
- **Pensamiento algorítmico:** Forma de llegar a la solución a través de una sucesión clara de pasos (Csizmadia et al., 2015).
- **Automatización:** Tarea con la que se instruye al ordenador a realizar una serie de tareas repetitivas de manera rápida y eficiente (Lee et al., 2011).

- **Descomposición:** Forma de pensar en forma de un problema como conjunto de sub-problemas más pequeños. Cada sub-problema se puede comprender, solucionar, desarrollar y evaluar de manera independiente. Con ello se consigue que cada problema complejo sea más sencillo de resolver, que las situaciones nuevas se comprendan mejor y que los sistemas grandes se pueden diseñar más fácilmente (Csizmadia et al., 2015).
- **Depuración:** Aplicación sistemática del análisis y evaluación usando habilidades como el testeo, trazabilidad y el pensamiento lógico para predecir y verificar los resultados (Csizmadia et al., 2015).
- **Generalización:** Está asociada con la identificación de patrones, similitudes y conexiones y ser capaz de explotarlas. Es una manera rápida de resolver problemas nuevos basándose en soluciones a problemas antiguos y la experiencia obtenida. De igual forma, algoritmos que resuelve problemas específicos se pueden adaptar para solucionar problemas similares (Csizmadia et al., 2015).

Algunos autores afirman que el PC no se caracteriza sólo por las habilidades, sino que también se caracteriza por actitudes, tal y como se muestra a continuación:

- **Confianza:** Confianza en uno mismo a la hora de afrontar un problema nuevo y complejo (Barr et al., 2011) (Weintrop et al., 2016).
- **Perseverancia:** Ser perseverante a la hora de trabajar con problemas difíciles (Barr et al., 2011) (Weintrop et al., 2016) (Woollard, 2016).
- **Creatividad:** Ser creativo a la hora de proponer soluciones a problemas distintos (Woollard, 2016).
- **Trabajo en equipo:** Habilidad para comunicarse y trabajar en equipo para resolver los problemas más fácilmente y alcanzar una solución común (Barr et al., 2011) (Woollard, 2016).
- **Sin miedo:** No tener miedo a jugar, probar y equivocarse... (Woollard, 2016).

En el próximo capítulo...

En este capítulo se ha introducido al lector de forma muy básica en los conceptos básicos y características del PC.

Con todo ello, en el siguiente capítulo se pretenderá justificar la necesidad actual de trasladar el PC a las aulas de secundaria.

Capítulo 3

Importancia del pensamiento computacional

Los ordenadores en sí mismos, y el software aún no desarrollado, va a desarrollar la forma en que aprendemos.

Steve Jobs

RESUMEN: Este capítulo presenta al lector la importancia que tiene el PC así como las ventajas que tiene el aprendizaje del mismo.

Partiendo de la definición y las características que la literatura otorga al PC, en este capítulo se intenta explicar la importancia del mismo en el siglo XXI.

En la actualidad, la vida de cualquier persona está rodeada de tecnología, tanto en la carretera, en los hospitales, oficinas, fábricas. . . , es un mundo digital que se debe comprender para poder avanzar.

Si bien es cierto que basta realizar una búsqueda en Google para ver la tendencia de la importancia de este concepto a lo largo de los últimos diez años, tal y como se muestra en la figura 3.1.

Y el motivo es que el PC ha ganado tanto crédito por ser una habilidad transversal, como leer o escribir, que será usada por todo el mundo para mediados del siglo XXI (Wing, 2017).

Puede establecerse un listado con las ventajas del aprendizaje del PC (Playcode Academy, 2017):

1. El PC, a través de los lenguajes de programación, **es la forma de expresión más empleada en el mundo**, usada por miles de millones de máquinas. Si una persona sabe expresarse a través de un lenguaje de programación, conocerá las estructuras del PC y apenas le costará esfuerzo aprender otro lenguaje de programación de nivel similar.

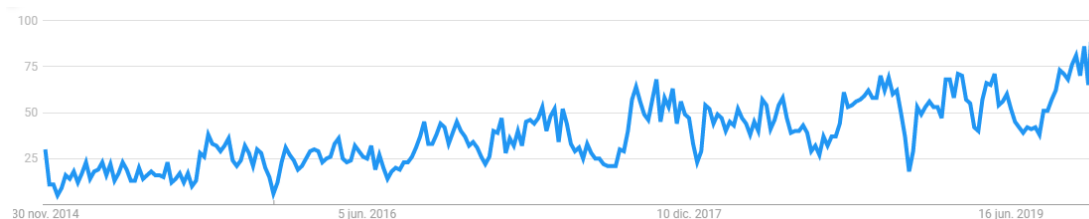


Figura 3.1: Tendencia de la búsqueda del concepto Pensamiento Computacional en el periodo 2014 - 2019 (Google Trends)

2. A través del PC **cambia el consumo de tecnología**. Al entender cómo funciona una máquina, se aprende más de ella, y se es un consumidor más sabio además de poder ser también un creador de contenido.
3. Un alumno que conoce el PC **puede crear tecnología** para su consumo, montar su propio robot, videojuego, aplicación móvil... según sus intereses en cada momento. Es muy adecuado tanto para la niña que quiere ser ingeniera como para el niño que quiere ser bailarín. Ambos se terminarán beneficiando en el futuro.
4. Si un alumno puede crear tecnología, **podrá llevar sus ideas al mundo real**, ya sea un robot, una página web o un modelo 3D que imprimirá después.
5. Con la programación aprendes a **resolver problemas**. Hay un objetivo a conseguir, con multitud de soluciones, y a base de un proceso iterativo de mejora constante, se puede ir mejorando hasta lograr la solución final.
6. Con el PC se aprende a **pensar de manera lógica, ordenada y estructurada**. Se consigue adquirir una serie de procesos mentales internos, que les hace mejores estudiantes en general y que son muy beneficiosos en todas las disciplinas.
7. **La programación es útil y fácil** en la actualidad. Tanto en el mundo profesional como en el personal se emplea *software*, por eso es muy útil saber programar, y más en la actualidad donde es muy accesible.

En el próximo capítulo...

En este capítulo se ha mostrado al lector la importancia que tiene el PC para la sociedad actual.

A partir de este momento se va a analizar la situación actual del currículo en cuanto a educación formal y no formal. Para comenzar, en el próximo capítulo se van a enumerar brevemente las distintas iniciativas existentes dentro de la educación no formal.

Capítulo 4

Educación no formal: Iniciativas existentes

La formación en informática no puede convertir a nadie en experto programador, de la misma forma que estudiar pinceles y pigmentos puede hacer a alguien un experto pintor

Eric Raymond

RESUMEN: Más allá de los esfuerzos por adaptar el currículo en la educación formal, existe un amplio abanico de opciones entre las que elegir dentro de la educación informal y no formal tanto a nivel internacional, nacional o regional.

En este capítulo se muestran algunas iniciativas que se han encontrado en la red y que merece la pena destacar.

4.1. Mundiales

4.1.1. CoderDojo

Club gratuito para niños entre 7 y 17 años organizado por voluntarios. Niños, padres, mentores y otros juegan con tecnología y aprenden a programar. Hay más de 2.056 grupos en más de 106 países alcanzando los 58.000 niños.

Más información en: <https://coderdojo.com/es-ES>

4.1.2. Code.org

Organización sin ánimo de lucro dedicada a expandir el acceso a las CS y la participación de la mujer y las minorías. Más de 263 millones de personas han probado ya la **Hora de Código** en todo el mundo y 11 millones de alumnos han usado **Code Studio**.

Más información en: <http://code.org/>

4.1.3. Bebras

Iniciativa que busca promover las CS entre profesores y alumnos de todas las edades, así como acercarlas al público en general. Esta iniciativa organiza retos algorítmicos en línea para muchos países.

Más información en: www.bebas.org

4.1.4. CS Unplugged

Colección de actividades gratuitas para aprender a través de los juegos y los puzzles. La iniciativa permite a los alumnos jóvenes introducirse en los primeros conceptos de CS.

Más información en: www.codeclubworld.org

4.1.5. Code Club

Comunidad de expertos voluntarios que comparten su pasión por la tecnología con los niños y profesores de todo el mundo. Ofrece lecciones gratuitas para profesores de primaria y organiza clubs de niños para aprendizaje de programación.

Más información en: www.codeclubworld.org

4.1.6. Made with Code

Iniciativa respaldada por Google que busca acercar a las mujeres al aprendizaje de la programación y cerrar la brecha existente en la industria tecnológica. Ofrece recursos, vídeos y actividades para motivar a las chicas a realizar actividades de programación.

Más información en: www.madewithcode.com

4.1.7. Olimpiada Informática Internacional

Concurso de programación algorítmica para alumnos y alumnas de educación secundaria (ESO, bachillerato o grado medio) que se viene celebrando anualmente en un país entre los países miembro.

Más información en: <https://ioinformatics.org/>

4.1.8. Google Code Jam

Competición online organizada por Google para alumnos de todos los niveles con el objetivo de resolver problemas algorítmicos de manera eficiente.

Más información en: <https://codingcompetitions.withgoogle.com/codejam>

4.1.9. Microsoft Hacking STEM

Es un conjunto de proyectos formados por actividades de STEM que permiten a los docentes guiar a sus alumnos para construir y crear instrumentos científicos y herramientas basadas en proyectos para visualizar datos a través de las STEM.

Cada proyecto está diseñado para cultivar experiencias en el aula resolviendo problemas del mundo real. En estas actividades los alumnos adoptan el rol de ingenieros de software, eléctricos, mecánicos y científicos de datos.

Más información en: www.microsoft.com/es-xl/education/education-workshop/activity-library.aspx

4.1.10. Microsoft MakeCode

Este proyecto ofrece a los estudiantes proyectos divertidos con resultados inmediatos y editores tanto con bloques como con texto para estudiantes de distintos niveles.

Ofrece un simulador interactivo que da información instantánea sobre cómo se está ejecutando su programa y permite depurar el código. También incorpora un editor de bloques para los más pequeños o aquellos que no tiene experiencia en programación, con bloques de colores que se pueden arrastrar y soltar para crear programas.

Por último, cuando estén listos, los estudiantes podrán pasar a un editor completo de JavaScript con fragmentos de código, información sobre herramientas y detección de errores para ayudarles.

Más información en: www.microsoft.com/es-es/makecode

4.2. Europeas

4.2.1. Semana de la Programación de la UE

Movimiento llevado a cabo por voluntarios que promueven la programación en sus países como embajadores. Esta iniciativa europea apareció en 2013 y ya tiene más de 150.000 personas organizando más de 4.200 eventos anuales en los distintos países de la UE (Unión Europea).

Más información en: <https://codeweek.eu>

4.2.2. All you need is code

Esta iniciativa reúne un grupo de empresas para promover la programación y el PC en todos los niveles educativos, así como en entornos informales.

Más información en: www.allyouneediscode.eu

4.2.3. Barefoot Computing

Programa que apoya a los profesores de primaria británicos para llevar a cabo el currículo en lo relacionado a programación.

Más información en: www.barefootcomputing.org

4.2.4. Computing At School

Esta iniciativa busca fijar una figura de liderazgo y guía estratégico para todas las personas implicadas en la educación de la programación en los colegios.

Más información en: www.computingatschool.org.uk

4.2.5. Code it Like a Girl

Este programa griego organiza talleres para familiarizar a las mujeres con la programación.

Más información en: <http://codeitlikeagirl.com>

4.3. Españolas

4.3.1. Programamos

Asociación sin ánimo de lucro cuyo objetivo fundamental es promover el desarrollo del PC desde edades tempranas a través de la programación de videojuegos y aplicaciones para móviles en todas las etapas escolares, desde educación Infantil hasta la universidad. Entre sus iniciativas destacan:

- **Curso gratuito básico para aprender a programar con el lenguaje de programación Scratch.** Se trata de una introducción a la programación informática de videojuegos usando la herramienta Scratch. Con este curso se aprenden los elementos fundamentales para programar videojuegos de una forma totalmente práctica. Comienza con vídeos que muestran una introducción a la herramienta, así como distintos aspectos básicos de programación. A continuación, se muestra cómo crear paso a paso un clásico videojuego de naves, para lo que se aprende a realizar movimientos, detectar colisiones, realizar disparos, ...
- **Comunidad Programamos - Red social.** Es una red social de programación en la que se pueden compartir recursos, interaccionar con otros docentes y colaborar con otros compañeros.
- **Recursos para robótica en Infantil y Primaria con robots programables.** Se han creado diferentes tableros (rincón de las emociones, rincón de los cuentos, rincón de los más pequeños, rincón del arte...), que permiten trabajar diferentes aspectos emocionales, artísticos, literarios, aritméticos. . . , al tiempo que se desarrolla el PC de los estudiantes, haciendo uso de robots tipo Beebots o Escornabots, que pueden ser programados mediante los botones que incorporan en la parte superior del dispositivo.
- **Listas gratuitas de vídeos para aprender a programar en diferentes lenguajes de programación.** Como Scratch, Snap! o App Inventor. Todos los vídeos

son gratuitos, y cada vídeo se acompaña de un artículo en el blog en el que se discuten aspectos concretos de los mismos.

Más información en: <https://programamos.es>

4.3.2. Grupo KGBL3 - Dr. Scratch

Grupo formado por investigadores de la Universidad Rey Juan Carlos y la UNED (Universidad Nacional de Educación a Distancia), que ha desarrollado diversas investigaciones que muestran que el desarrollo del PC a través de la programación tiene un impacto positivo en el aprendizaje de distintas disciplinas, como las matemáticas, los idiomas, las ciencias o la narrativa.

Entre otras cosas, el grupo KGBL3 trabaja en el desarrollo de herramientas para evaluar el PC, como son **Dr. Scratch**, una aplicación web gratuita que permite evaluar el nivel de PC demostrado al programar un proyecto Scratch, y el **Test de Pensamiento Computacional**, un instrumento que permite medir el grado de desarrollo de ciertos aspectos del PC para estudiantes de Primaria y Secundaria.

Más información en: <https://kgbl111.github.io>

4.3.3. Olimpiada Informática Española

Concurso de programación algorítmica para alumnos y alumnas de educación secundaria (ESO (Educación Secundaria Obligatoria), bachillerato o grado medio) que se viene celebrando anualmente desde 1997. Como premio principal los ganadores son invitados a participar en la Olimpiada Informática Internacional.

Más información en: <https://olimpiada-informatica.org>

4.3.4. Creando Código - ScolaTIC

Iniciativa de Telefónica que ofrece una oferta completa para el docente de Infantil, Primaria y Secundaria con proyectos de programación, robótica e impresión 3D diseñados para aplicar directamente en el aula adaptados a las materias y currículo de cada etapa. Incluye la formación que el profesor necesita alrededor de estas materias.

Los docentes disponen de un repositorio de proyectos y propuestas didácticas curriculares listas para ser implementadas en el aula, que complementan la programación que cada docente lleva a cabo.

Más información en: www.scolartic.com

4.3.5. CLOQQ

CLOQQ (Crea Lo Que Quiera) es una iniciativa social de fun-learning para que los niños den rienda suelta a su imaginación y comprendan mejor el mundo que les rodea a través de tecnologías creativas. Utilizándolas desarrollan habilidades para resolver pro-

blemas inesperados, trabajan en equipo y fortalecen su autoconfianza sacando adelante sus propios proyectos.

Ofrece tanto actividades presenciales (talleres, eventos, etc.) como un completo entorno online y de contenidos para que niñas y niños puedan crear sus propios videojuegos, apps, robots, animaciones y mucho más.

Más información en: <https://cloqq.com>

4.4. Castilla y León

4.4.1. La escuela TIC

La Consejería de Educación de la Junta de Castilla y León pone en marcha un programa denominado “Escuelas de verano”, que pretende definir nuevas actuaciones y experiencias formativas intensivas y de especialización. A través de él, la Administración de Castilla y León proyecta actualizar y mejorar las competencias profesionales del profesorado mediante cursos intensivos y especializados referentes a las líneas prioritarias definidas por la Junta de Castilla y León.

En concreto, la escuela TIC (Tecnologías de la Información y las Comunicaciones) prevé 2 itinerarios: en primer lugar, cursos sobre robótica y programación, dirigidos a profesorado de Secundaria de Tecnología e Informática; y, en segundo lugar, cursos sobre diseño e impresión 3D, dirigidos al profesorado en general.

Más información en: <http://bocyl.jcyl.es/boletines/2017/09/12/pdf/BOCYL-D-12092017-1.pdf>

4.4.2. Crea y Explora

En el proyecto **Crea** se forma al profesorado en técnicas de impresión 3D, mientras que en **Explora** se trabaja todo lo referente a la integración de técnicas de realidad virtual en el aula.

A través de estas iniciativas se promueve el uso de los últimos avances tecnológicos y se pretende formar a los participantes en metodologías innovadoras que mejoren sus competencias profesionales utilizando las últimas tecnologías, así como conocer sus principales aplicaciones didácticas. Además, se trata de fomentar el desarrollo de las competencias STEAM (*Science, Technology, Engineering, Arts and Maths* - Ciencias Naturales, Tecnología, Arte, Diseño y Matemáticas), incidiendo de forma especial en ciencias, tecnología e ingeniería. Del mismo modo que los proyectos Ingenia, en Crea y Explora el profesorado recibe una formación presencial, trabaja de forma colaborativa con el resto de participantes y desarrolla un proyecto en el aula con sus alumnos.

Más información en: www.educa.jcyl.es/profesorado/es/formacion-profesorado/proyectos-relacionados-formacion-permanente-profesorado/competencia-digital/convocatorias/proyectos-innovacion-educativa-tic-crea-explora-ingen-5a82c

En el próximo capítulo...

A lo largo de este capítulo se ha llevado a cabo un breve recorrido por las iniciativas existentes dentro de la educación no formal.

Llegados a este punto, en el próximo capítulo se continúa el recorrido analizando la situación actual de la educación formal a nivel europeo.

Capítulo 5

Educación Formal: Situación actual

¿Sabes cuál es el problema? Imaginate el algoritmo y no programarlo.

Paul Huanca

RESUMEN: Se están llevando a cabo multitud de iniciativas para incluir los conceptos del pensamiento computacional en los distintos currículos educativos del mundo (Bocconi et al., 2016). En este capítulo, se va a hacer un breve repaso de los currículos europeos para finalizar haciendo hincapié en el currículo actual del sistema educativo en Castilla y León.

Como se ha visto en capítulos previos, son muchos los motivos para incluir el PC en la educación formal. En algunos países se han incluido asignaturas de programación, con el objetivo erróneo de desarrollar las habilidades de resolución de problemas y el PC en general (Webb et al., 2017).

Otros países, como Austria, han decidido ver las asignaturas de informática como una forma de resolución de problemas, análisis de procesos y dependencias entre ellos.

La inserción del PC ha sido vista en otros países como una forma de cubrir la brecha existente entre el currículo y las necesidades de los alumnos y la sociedad en general. Países como Reino Unido han hecho una remodelación completa del currículo haciendo énfasis en las habilidades digitales (Webb et al., 2015).

Aunque las estrategias adoptadas por los distintos países europeos han sido muy variadas, se pueden agrupar en cuatro bloques según los enfoques adoptados para integrar el PC en el currículo:

1. Países que han comenzado un proceso de revisión y ajuste del currículo a nivel nacional.
2. Países que han comenzado procesos de revisión y ajuste del currículo a nivel regional.

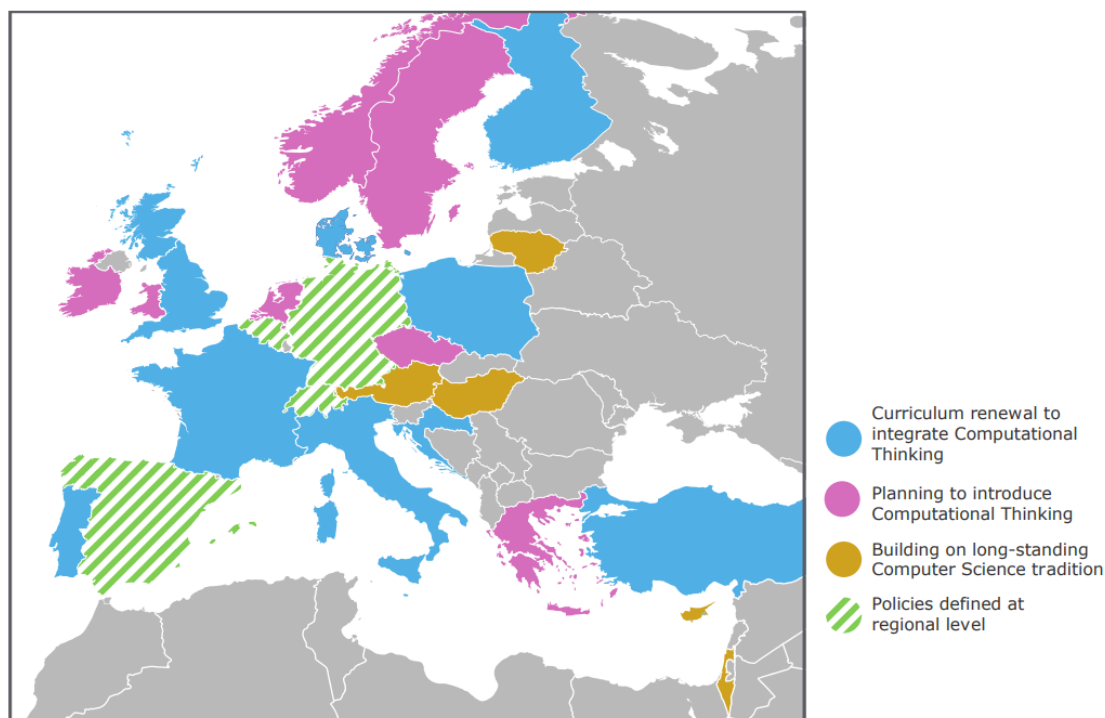


Figura 5.1: Distribución de los países según el proceso de revisión del currículo.

3. Países que aún no han comenzado a introducir el PC en su sistema educativo obligatorio pero se están preparando para hacerlo en breve.
4. Países que tienen ya una tradición en CS en escuelas de educación secundaria y buscan expandir este hacia los niveles inferiores.

En la figura 5.1 puede verse la distribución de estos cuatro bloques.

5.1. Caso de Reino Unido

Reino Unido es un caso de éxito siendo uno de los pioneros en Europa implantando el PC y la programación de manera obligatoria en el currículo de primaria y secundaria desde el año 2004.

Este país apostó por “una educación informática de calidad que prepare a los alumnos para emplear el PC y la creatividad en el futuro para comprender y cambiar el mundo”. Esta decisión sentó las bases de las reformas educativas de muchos otros países del continente europeo, no sólo poniendo el foco en la informática, sino también en el PC.

5.2. Caso de Finlandia

Finlandia fue uno de los primeros países europeos en implantar de manera obligatoria en sus currículos el pensamiento algorítmico y la programación (desde 2016). Sin embargo, a diferencia de Reino Unido, aplicó un enfoque totalmente distinto, definiendo la programación y el PC como actividades transversales en todo el currículo desde la educación primaria.

En los primeros cursos, los alumnos aprenden acerca de resolver los problemas paso a paso para llegar a una solución global. Más adelante, comienzan a utilizar herramientas de programación visuales para, en los últimos cursos de la educación básica, llegar a ver algoritmos y comprobar la utilidad de estos.

Este enfoque implica aplicar la programación y el PC como un medio y no como un fin, esto es, como una herramienta.

5.3. Currículo actual en España

En España el currículo se está adaptando a nivel regional, como se ve en la figura 5.1, por lo que depende mucho de cada comunidad autónoma.

A nivel nacional, dentro del sistema educativo de la ESO se pueden encontrar contenidos relacionados con la programación, la robótica y el PC en 3º y 4º de ESO.

Por un lado, en cuanto a las materias troncales se refiere, tal y como se recoge en la Orden EDU/362/2015, de 4 de mayo, la única materia es Tecnología en 4º de ESO dentro de la opción de enseñanzas aplicadas.

5.3.1. Materias troncales

5.3.1.1. 4º de ESO: Tecnología

La asignatura de Tecnología pretende que los alumnos observen en su entorno los objetos y los avances que les rodean y vean en ellos el resultado de un proceso que abarca la ciencia y la técnica, el pensamiento científico y las habilidades prácticas.

A lo largo de la historia de la humanidad los desarrollos tecnológicos han cambiado en gran medida la forma de vida, dando respuesta a una necesidad, a un anhelo o a una idea. En la educación Secundaria, esta materia busca que los estudiantes comprendan la relación del ser humano con el mundo creado por el hombre, valoren la tecnología como un proceso ligado íntimamente al ingenio, emprendimiento y habilidad humana.

La materia organiza los contenidos en seis bloques (**Tecnologías de la información y de la comunicación, Instalaciones en viviendas, Electrónica, Control y Robótica, Neumática e hidráulica y Tecnología y Sociedad**) que permiten avanzar en aspectos básicos que permiten analizar problemas tecnológicos concretos.

5.3.2. Materias de libre configuración autonómica

Por otro lado, en cuanto a las materias de libre configuración autonómica, tal y como se recoge en la Orden EDU/589/2016, de 22 de junio, las materias que incluyen contenidos relacionados con la programación, la robótica y el PC son:

5.3.2.1. 3º de ESO: Control y robótica

Esta materia engloba todos los pasos necesarios para resolver un problema tecnológico real, abarcando el conjunto de actividades pedagógicas dirigidas al diseño, la fabricación y montaje de un robot, las cuales se complementan con la elaboración del programa informático que permita el control del mismo.

De modo singular, se aplica al caso cada vez más presente de la impresora 3D, un tipo específico de robot que cumple una función esencial dentro de la filosofía DIY (*Do It Yourself*), que engloban procesos de inteligencia, así como de creación colectiva a través de la compartición de códigos, prototipos y modelos.

La programación como herramienta de control es utilizada en numerosos campos técnicos y sistemas de información, y es necesario conocerla para poder comprender y controlar la tecnología que nos rodea. Saber programar es fundamental para automatizar el funcionamiento de los robots y que éstos puedan interrelacionar con el entorno.

Por otro lado, para lograr el control de un robot es necesario aplicar conocimientos de múltiples disciplinas durante el diseño de la estructura, para dar movimiento al robot y conseguir que se adapte y comunique con la información del entorno. Por lo tanto, a través de esta materia se integran conocimientos relacionados con las matemáticas, ciencias experimentales y tecnologías de la información y la comunicación, los cuales toman una mayor significación al ser orientados hacia la resolución de un problema tecnológico.

Los contenidos se han agrupado en cuatro bloques:

1. **Bloque 1: Sistemas automáticos de control.** Busca comprender los tipos de sistemas de control, los componentes que lo forman y sus características principales.
2. **Bloque 2: Fundamentos de robótica.** Busca distinguir y conocer las características de las señales analógicas y digitales y el funcionamiento y propiedades de los componentes electrónicos.
3. **Bloque 3: Programación y control.** Incluye los conocimientos necesarios para programar usando algoritmos, diagramas de flujo, tipos de variables, estructuras de repetición, secuenciales y condiciones orientadas al control de robots.
4. **Bloque 4: Proyectos de robótica.** En este último bloque se unen los conceptos de los tres primeros bloques. El alumno aprende los elementos básicos de un robot, los diseña, proyecta y construye empleando software libre, con el que programa el mismo.

5.3.2.2. 4º de ESO: Programación informática

El aprendizaje de la programación representa un nuevo medio de alfabetización, que facilita la comprensión del mundo tecnológico en el que está inmersa la sociedad actual y permite intervenir en él y adoptar un papel activo en lugar del actual papel pasivo que tiene gran parte de la sociedad actual como usuarios de esta tecnología.

Los contenidos se han agrupado en tres bloques:

1. **Bloque 1: Introducción a la programación.** Busca que el alumno comprenda las bases del PC, mostrándole distintas formas de resolver problemas complejos. Además, se muestra un esbozo de lo que son los lenguajes de programación y qué lógica llevan detrás.
2. **Bloque 2: Entornos de programación gráfica mediante bloques.** Busca proporcionar un primer contacto con la programación usando herramientas sencillas que contienen algunos de los elementos básicos de los lenguajes de alto nivel. Con ellos, el alumno aprende a realizar aplicaciones móviles (APPs), animaciones y videojuegos atractivos con los que aprende a través de entornos gráficos donde cada acción viene determinada por un bloque que implica una serie de instrucciones de movimiento, acciones condicionales, etc.
3. **Bloque 3: Lenguajes de programación mediante código.** Es un primer paso a que el alumno tenga las habilidades necesarias para afrontar el aprendizaje de lenguajes de programación por código.

En el próximo capítulo...

A lo largo de este capítulo se ha llevado a cabo un breve recorrido por la situación actual de los sistemas educativos europeos.

Asimismo, se ha profundizado en el currículo de Castilla y León viendo las asignaturas relacionadas con el PC y la programación.

Llegados a este punto, se da por finalizada la definición de la situación actual para comenzar en el próximo capítulo con el planteamiento de las propuestas de actividades relacionadas con el PC.

Capítulo 6

Propuesta

Primero resuelve el problema. Entonces, escribe el código

John Johnson

RESUMEN: Este capítulo busca presentar distintos tipos de herramientas con las que presentar al alumno contenidos relacionados con el pensamiento computacional. Además, se plantean dos actividades con dos herramientas diferentes para interiorizar diferentes conceptos.

6.1. Enfoque

El PC se puede aplicar en un abanico muy amplio de disciplinas fuera de la informática o las CS. Este se puede usar para resolver problemas de prácticamente cualquier forma o color.

Véase por ejemplo cómo se puede aplicar el PC al análisis morfosintáctico de una oración, en la que se busca identificar y extraer patrones para diferentes estructuras sintácticas.

De la misma forma se puede aplicar para potenciar la creatividad de los alumnos en clase pasando de ser meros consumidores de tecnología a creadores de herramientas sociales.

Por toda esta variedad de aplicaciones, el NCR (*National Council for Research*) afirmó en 2010 que el PC es una habilidad que se espera que toda persona tenga. Además, definió que cada alumno debe adquirir esta habilidad de forma transparente junto con otras materias y que los profesores deben conseguir esto empleando las estrategias adecuadas (Yadav et al., 2014).

Este tipo de motivos son los que me hacen decantarme por un enfoque similar al elegido por países como Finlandia, donde el PC y la programación son materias trans-

versales que se reparten entre el resto de materias, fijando en la programación de estas una serie de contenidos a tratar.

6.2. Herramientas

La enseñanza de los conceptos de PC y programación requieren de herramientas que reduzcan las barreras de entrada y consigan que las actividades sean accesibles para los alumnos de secundaria e, incluso, sean atractivas y motivadoras para ellos (Piedade et al., 2019).

El pionero en estas herramientas fue Seymour Papert en los años 60, creando Logo, un lenguaje de programación para niños. Este es un lenguaje basado en texto, e introduce interfaces basados en bloques en la década de los 90. Otros lenguajes más recientes han sido Alice o Scratch, ya basados en bloques, que permiten que el alumno se centre en crear y experimentar sin preocuparse por la sintaxis u otros problemas del lenguaje (Papadakis et al., 2017).

Principalmente, las herramientas que permiten iniciar a los alumnos en el PC y la programación son los lenguajes basados en bloques o los textuales, los cuales tienen sus ventajas y desventajas (Piedade et al., 2019).

6.2.1. Herramientas basadas en bloques



Figura 6.1: Ejemplo de estructura de un código realizado en una herramienta basada en bloques.

Este tipo de herramientas presenta menos barreras de entrada para los alumnos, les permite llegar a conseguir su objetivo de forma más rápida y sencilla. Además de que



Figura 6.2: Blockly.



Figura 6.3: Scratch.



Figura 6.4: App Inventor.



Figura 6.5: Alice.

pueden llegar a leer, entender y aprender mucho más fácil sin tener que preocuparse por la sintaxis del lenguaje, ya que no hay que escribir nada. De esta forma, el alumno se puede centrar sólo y en exclusiva en lo que quiere hacer y sólo en eso.

Asimismo, con estas herramientas los resultados son inmediatos, se pueden ver según se va avanzando.

Por otro lado, si se combinan estas herramientas con una pantalla táctil o con una cámara web se permite que el alumno interactúe, permitiéndolo que juegue con el código de su programa y lo haga más suyo.

Algunos ejemplos dentro de este bloque son Scratch, App Inventor o Alice o el reciente Blockly, que muestran la conexión de los distintos elementos del programa. Son herramientas orientadas para el mundo infantil, pero que tienen un gran potencial más allá.

Para las actividades planteadas, se va a poner el foco en Scratch y App Inventor por ser los más conocidos y con mayor fuerza en la actualidad.

No obstante, en la tabla 6.1 se muestran algunas opciones con su descripción (Scratch Wiki, 2019):

Herramienta	Descripción
BYOB/Snap!	Es un Scratch modificado y más avanzado. En su versión 4.0 ha sido reescrito a JavaScript y se ha diferenciado bastante de Scratch.
Blockly	Herramienta creada por Google que convierte en tiempo real una aplicación por bloques en texto.
App Inventor	Herramienta creada inicialmente por el MIT y continuada por Google que permite crear una aplicación Android con un interfaz simple similar a Scratch.
Stencyl	Herramienta similar a Scratch con funcionalidades más avanzadas.
Gamefroot	Herramienta de creación de juegos online.
Pocket Code	Herramienta visual para móviles Android, iOS y Windows Phone, tablets y navegadores HTML 5.
Hopscotch	Herramienta para móviles iOS similar a Scratch pero más simple y sencilla de usar.
GameSalad	Herramienta muy simple para creación de juegos de manera muy sencilla y rápida.
GameMaker Studio	Herramienta para creación de video juegos por inexpertos. Para alumnos con mayores conocimientos, ofrece librerías que permiten agregar funciones de mayor complejidad.
Tynker	Herramienta orientada a la creación de juegos de manera sencilla. Permite la conexión con herramientas externas como Minecraft.

Tabla 6.1: Ejemplos de herramientas basadas en bloques.

6.2.2. Herramientas basadas en texto

A diferencia de las herramientas basadas en bloques, este tipo de herramientas son más complejas para comenzar. No obstante, son auténticos lenguajes de programación profesionales.

La principal ventaja de estos es que, de cara al futuro, si el alumno aprende un lenguaje como Python (que es de los más sencillos), permite entender los conceptos de programación a un nivel mucho más profundo.



Figura 6.6: Python.

Existe un gran abanico de opciones como se puede ver en las tablas 6.2 y 6.3 (Scratch Wiki, 2019):

Herramienta	Descripción
Java	Originalmente creado y popularizado por ser independiente de la máquina. Hoy en día es un lenguaje de propósito general.
Python	Lenguaje de propósito general con el foco puesto en la legibilidad. Tiene librerías para comunicarse con Scratch.
C	Lenguaje originalmente popular por su portabilidad. Hoy en día se usa a menudo en sistemas operativos y sistemas embebidos.
C++	Lenguaje basado en C que ofrece programación orientada a objetos.
C#	Lenguaje creado por Microsoft basado en C.
ActionScript	Lenguaje de propósito general para Adobe Flash. Ofrece un gran potencial a nivel gráfico y de animaciones.

Tabla 6.2: Ejemplos de herramientas basadas en texto.

Herramienta	Descripción
Lua	Lenguaje muy ligero muy popular en la creación de videojuegos.
JavaScript	Lenguaje usado para aplicaciones web interactivas.
Logo	Es el predecesor de Scratch. Ofrece un único personaje, una tortuga, usada para dibujar una historia.
QB64	Es un lenguaje muy similar a Python y BASIC. Ofrece una amplia variedad de aplicaciones y sencillo de aprender.
Ruby / Ruby on Rails	Lenguaje de programación orientado a objetos muy usado para creación de páginas web.

Tabla 6.3: Ejemplos de herramientas basadas en texto (cont.).

6.2.3. Scratch



Figura 6.7: Scratch.

Scratch es una herramienta gratuita creada por el MIT (*Massachusetts Institute of Technology*) diseñada para que cualquier persona pueda iniciarse en los conceptos del PC y, en concreto, en la programación. Con ella, el alumno podrá crear animaciones, juegos, etc. para crear su propia historia y compartirlo con el resto de compañeros (Meerbaum-Salant et al., 2013) (Papadakis et al., 2016).

Scratch (tanto en su versión online como en su versión instalable) permite animar los elementos o “**sprites**” para que realicen acciones y comportamientos de forma que reaccionen de una determinada manera ante unos eventos seleccionados.

Estas acciones o comportamientos tienen forma de puzzle que el alumno tiene que encajar (como se ve en la figura 6.8). De esta forma, para el alumno se asemeja a resolver

un puzzle encajando las piezas, siendo algo atractivo para este, que lo ve como un juego con el que aprende mientras juega. Esta motivación elimina una de las barreras de entrada más importantes para los alumnos que se inician en este tipo de herramientas.



Figura 6.8: Ejemplo de un código desarrollado con Scratch.

Existe una gran cantidad de acciones y comportamientos, que se agrupan en las siguientes categorías:

- **Apariencia:** Cambia la visualización del sprite, su tamaño, color ...
- **Control:** Condiciones if-else, forever, repeat, stop.
- **Datos:** Crear variables y se asignan.
- **Eventos:** Maneja los eventos que disparan determinadas acciones en un bloque.
- **Lápiz:** Dibuja controlando el tamaño de pincel.
- **Movimiento:** Mueve el sprite y lo gira por la pantalla.
- **Operadores:** Operadores matemáticos, generadores de números aleatorios ...
- **Sensores:** Los *sprites* pueden interactuar con su entorno.
- **Sonido:** Hace sonar un sonido.

En las figuras figuras 6.9 a 6.12 se muestran algunos casos de bloques de Scratch empleados para representar elementos del PC:

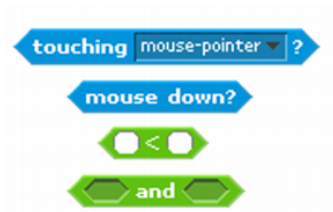


Figura 6.13: Variables booleanas en Scratch.



Figura 6.14: Hilos en Scratch.



Figura 6.15: Eventos en Scratch.



Figura 6.9: Instrucciones en Scratch.



Figura 6.10: Bucles en Scratch.



Figura 6.11: Condiciones en Scratch.



Figura 6.12: Variables en Scratch.

6.2.4. App Inventor



Figura 6.16: App Inventor.

App Inventor es una herramienta gratuita creada inicialmente por el MIT y continuada por Google que está orientada a la creación de aplicaciones Android de manera limitada y muy sencilla (Papadakis et al., 2016).

Con App Inventor el alumno puede, de manera visual y con un conjunto de utilidades básicas, enlazar bloques con los que crear su App.

En primer lugar, el alumno deberá llevar a cabo el diseño de su App, para lo que tendrá que seleccionar y arrastrar una serie de elementos como botones, imágenes, campos de texto, etc. sobre el entorno de trabajo.

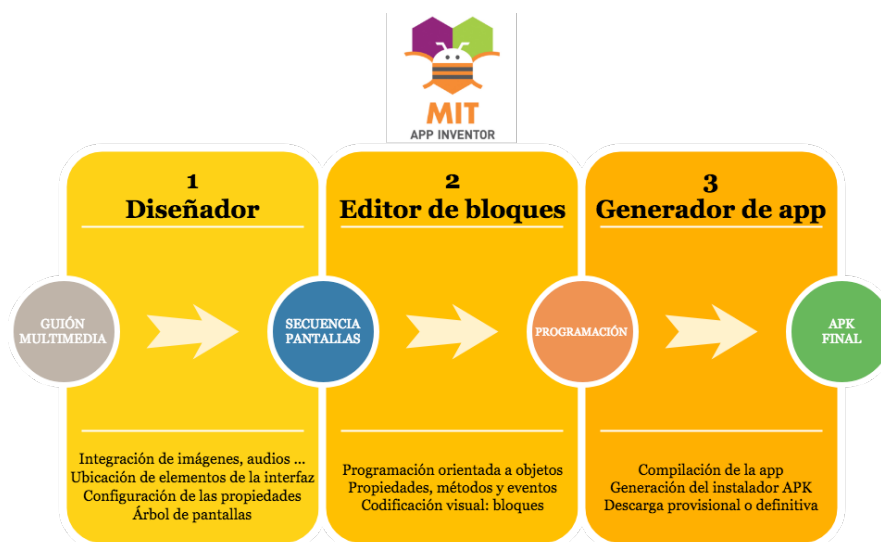


Figura 6.17: Fases en el desarrollo de una proyecto con App Inventor.

Una vez diseñada su App, ya estará en disposición de ir al editor de bloques y escoger

el comportamiento que quiera asignar a cada elemento. La App aparecerá paso a paso en el dispositivo móvil según se van añadiendo bloques a la misma o en un emulador a través de la aplicación AI Companion, también gratuita.

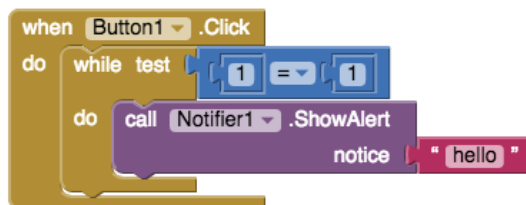


Figura 6.18: Ejemplo de un código desarrollado con App Inventor.

Como se puede ver, es muy similar a la forma de trabajar en Scratch con la diferencia de la portabilidad al dispositivo Android.

6.2.5. Comparativa

Scratch y App Inventor ha podido comprobarse que son más efectivos en distintas dimensiones del PC por lo que pueden usarse de forma complementaria (Papadakis et al., 2017).

Los proyectos realizados con Scratch profundizan más en conceptos como el paralelismo o el control del flujo, mientras que los realizados con App Inventor lo hacen en la interacción con el usuario y la representación de la información.

Es por ello que parece que cada lenguaje es más indicado para distintas áreas, por lo que se pueden ver como herramientas complementarias, puesto que la combinación permite que el alumno alcance un nivel más completo y global.

6.3. Actividades

Para este proyecto se van a proponer una actividad basada en Scratch y otra en App Inventor de distintas disciplinas para tratar de justificar que la programación debe ser tratada desde distintas materias.

6.3.1. El cuerpo humano

6.3.1.1. Introducción

Con esta actividad se va a trabajar la asignatura de Biología y Geología de 1º de ESO, en concreto se pretenden conocer los órganos del cuerpo humano. Además, a través de esta actividad también se quieren mostrar las condiciones if-else.

6.3.1.2. Contextualización de la unidad didáctica

Como he indicado, esta actividad se va a realizar desde la asignatura de Biología y Geología de 1º de ESO, donde, siguiendo el Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y el Bachillerato, en el bloque 4 de **Las personas y la salud, promoción de la salud** incluye entre los contenidos a trabajar la *Organización general del cuerpo humano: células, tejidos, órganos, aparatos y sistemas*.

6.3.1.3. Centro educativo: contextualización

Se trata de un instituto situado en el casco urbano de la ciudad de Valladolid, capital de la comunidad autónoma de Castilla y León, contando con una población de 300.000 habitantes, aproximadamente.

La ciudad de Valladolid cuenta con una gran oferta cultural y gastronómica, y está muy bien comunicado por tren y carretera con Madrid, así como con el resto de ciudades de la Comunidad.

El contexto social es correcto, en un barrio de nivel medio-alto, con gran dedicación al sector servicios.

El centro es un instituto de tamaño grande con enseñanza de ESO y Bachillerato, con un total de 50 profesores y 500 alumnos. La edad del profesorado es media/alta y con gran experiencia aunque anclado en lo tradicional.

6.3.1.4. Duración

Esta actividad está prevista para llevarse a cabo den 1º ESO durante la tercera evaluación del curso y tendrá una duración de tres sesiones de 50 minutos.

6.3.1.5. Criterios de evaluación

Siguiendo el Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y el Bachillerato, dentro del bloque 4 de “Las personas y la salud, promoción de la salud” con esta actividad se van a tratar los siguientes criterios de evaluación y estándares de aprendizaje evaluables.

■ Criterios de evaluación

- 17. Identificar los componentes de los aparatos digestivo, circulatorio, respiratorio y excretor y conocer su funcionamiento.
- Reconocer y diferenciar los órganos de los sentidos y los cuidados del oído y la vista.
- * Emplear las condiciones if-else para su cometido

■ Estándares de aprendizaje evaluables

- 17.1. Conoce y explica los componentes de los aparatos digestivo, circulatorio, respiratorio y excretor y su funcionamiento.
- 18.1. Especifica la función de cada uno de los aparatos y sistemas implicados en la funciones de relación.
- 18.2. Describe los procesos implicados en la función de relación, identificando el órgano o estructura responsable de cada proceso.
- 18.3. Clasifica distintos tipos de receptores sensoriales y los relaciona con los órganos de los sentidos en los cuales se encuentran.
- * El alumno sabe emplear las condiciones if-else para su cometido.

* Se han añadido estos criterios de evaluación relativos al aprendizaje propio del PC que se quiere transmitir en esta actividad.

6.3.1.6. Metodología

La metodología que se pretende seguir es puramente activa donde los alumnos aprendan haciendo, basando todo el contenido en un aprendizaje orientado a proyectos.

Siguiendo esta forma de trabajo, se plantea un ejercicio con el que los alumnos desarrollen un ejercicio de ubicación de los órganos del cuerpo humano con el objeto de comprobar que saben ubicarlos correctamente y, además, puedan interiorizar el concepto de condicional.

Puede dividirse el proceso en 4 fases:

- **Instrucción:** El profesor explica una introducción sobre los órganos del cuerpo humano.
- **Presentación:** El profesor presenta el ejercicio y hace un ejemplo aprovechando para explicar el concepto de condicional.
- **Realización:** Los alumnos realizan el ejercicio.
- **Evaluación:** Los alumnos presentan sus ejercicios y el profesor realiza algunas preguntas para poder evaluarlo.

Esta actividad implica trabajo en equipo en todas las fases excepto en la inicial de instrucción.

6.3.1.7. Medidas de atención a la diversidad

La atención a la diversidad es el conjunto de actuaciones educativas dirigidas a dar respuesta a las diferentes capacidades, ritmos y estilos de aprendizaje, motivaciones e intereses, situaciones sociales, culturales, lingüísticas y de salud del alumnado.

Tan pronto como se detecten dificultades de aprendizaje en un alumno se pondrán en marcha medidas de carácter ordinario, adecuando la programación a las necesidades del alumnado, adaptando actividades, metodología o temporalización y, en su caso, realizando adaptaciones no significativas del currículo.

Están incluidas las adaptaciones curriculares tanto para el alumnado con necesidades educativas especiales como con altas capacidades intelectuales.

Alumnado con necesidades educativas especiales Son aquellos alumnos que requieran durante un periodo de su escolarización o a lo largo de toda la etapa, determinados apoyos y atenciones educativas específicas derivadas de discapacidad o trastornos graves de conducta, de acuerdo con el correspondiente dictamen de escolarización.

La identificación y valoración de las necesidades educativas especiales requerirá la realización de una evaluación psicopedagógica.

La Concreción Curricular dentro del Proyecto Educativo del centro ha de identificar qué necesidades específicas tiene su alumnado para que el profesorado pueda ajustar el currículo a las características del grupo-aula en la programación.

El departamento de Orientación pondrá a disposición del profesorado los informes de la evaluación psicopedagógica de los alumnos con necesidades educativas especiales para que una vez conocidas sus capacidades, el profesorado puede adaptar la programación a las necesidades del alumno.

Para que este alumnado pueda alcanzar el máximo desarrollo de sus capacidades personales y los objetivos de la etapa, se establecerán dentro de los principios de inclusión y normalidad, las medidas organizativas y curriculares, que aseguren su adecuado progreso y el máximo logro de los objetivos.

El plan de trabajo individualizado para este alumnado concretará las medidas de compensación y de estimulación, así como las materias en las que precise adaptación curricular, especificando las tareas a realizar por cada profesional.

En el caso del alumnado con necesidades educativas especiales que requiera adaptaciones curriculares significativas, éstas se elaborarán con el nivel de exigencia de la calidad y cantidad del resultado final y de la información que sean capaces de transmitir.

Las adaptaciones significativas de los elementos del currículo se realizarán buscando el máximo desarrollo posible de las competencias; la evaluación continua y la promoción tomarán como referente los elementos fijados en dichas adaptaciones. En cualquier caso, el alumnado con adaptaciones curriculares significativas deberá superar la evaluación final para poder obtener el título correspondiente.

Alumnado con altas capacidades intelectuales La atención educativa al alumnado con altas capacidades intelectuales se desarrollará, en general, a través de medidas de adecuación del currículo, de enriquecimiento y/o de ampliación curricular, con la finalidad de promover un desarrollo equilibrado de las distintas capacidades establecidas en los objetivos de la etapa, así como de conseguir un desarrollo pleno y equilibrado de sus potencialidades y de su personalidad.

Sesión	Duración	Actividad
Sesión 1	5 min.	Vídeo sobre el cuerpo humano.
	15 min.	Debate sobre el vídeo.
	30 min.	Explicación magistral sobre los órganos del cuerpo humano
Sesión 2	10 min.	Presentación de la actividad a realizar.
	20 min.	Explicación sobre un ejemplo en Scratch de las condicionales.
	20 min.	Trabajo en la actividad por parte de los alumnos.
Sesión 3	30 min.	Trabajo en la actividad por parte de los alumnos.
	20 min.	Presentación de las actividades realizadas por los alumnos.

Tabla 6.4: Temporalización de la actividad El cuerpo humano.

6.3.1.8. Temporalización

6.3.1.9. Materiales

En la asignatura no se usará ningún libro texto. El seguimiento de la asignatura se realizará a través de una plataforma LMS (Moodle) en la que están la guía de los conceptos clave de cada Unidad, los materiales necesarios para cada práctica, así como los trabajos entregados por los alumnos.

Se proporcionarán textos relacionados con los contenidos que se van a desarrollar en la unidad, acompañados de cuestiones sobre la lectura que susciten la reflexión, la opinión y el debate sobre los aspectos y los conocimientos relacionados con la misma.

También se proporcionarán imágenes, gráficas y fotografías que permiten visualizar lo que se está explicando en el texto.

Por otro lado, también se visualizarán vídeos aclarativos sobre los puntos tratados en el texto central.

Además, se realizarán y/o adjuntarán cuadros, tablas, esquemas que recojan esquemáticamente algunos contenidos.

En cuanto a los recursos del aula, se hará uso de:

- Ordenadores con acceso a Internet y un navegador instalado o la aplicación Scratch.
- Proyector
- Conexión a Internet

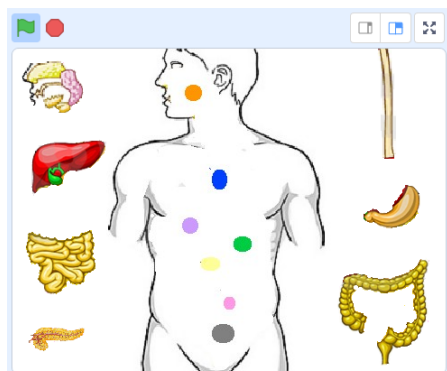


Figura 6.19: Visualización de la solución realizada (inicio).

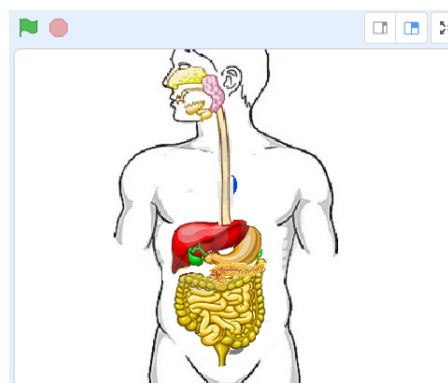


Figura 6.20: Visualización de la solución realizada (fin).

6.3.1.10. Desarrollo de la actividad

Este proyecto se centra principalmente en la actividad planteada a los alumnos, que desarrollan entre las sesiones 2 y 3.

Materiales y recursos Los materiales para esta actividad son los mismos que de forma global:

- Ordenadores con acceso a Internet y un navegador instalado o la aplicación Scratch.
- Proyector para mostrar la actividad de cada grupo de alumnos
- Conexión a Internet

Descripción Una vez explicados los órganos del cuerpo humano en la primera sesión, la actividad consiste en que los alumnos, por grupos de 4, seleccionen el hígado, intestinos grueso y delgado, estómago, esófago, páncreas, boca y glándulas salivales y hagan un proyecto Scratch para ubicarlos en el cuerpo similar al mostrado en las figuras 6.19 y 6.20.

A continuación se muestra un ejemplo de solución similar al que deben realizar los alumnos en esta actividad.

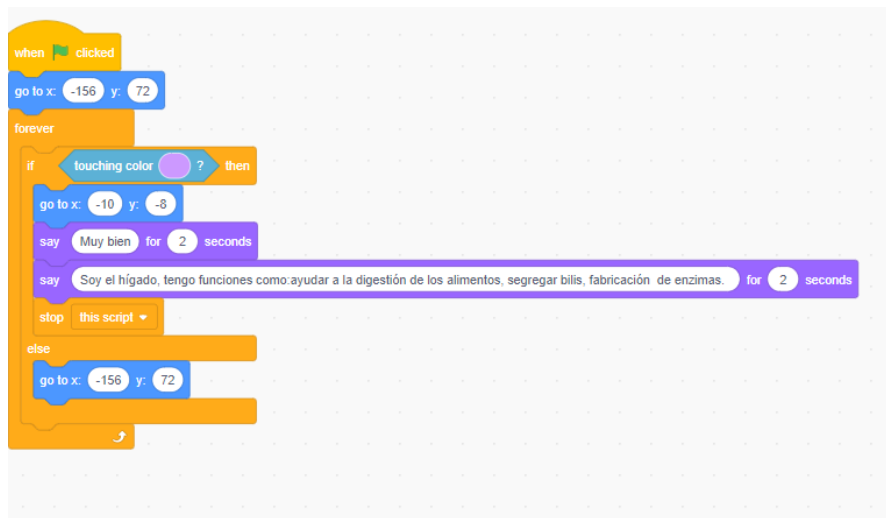


Figura 6.21: Ejemplo solución para el comportamiento del hígado.

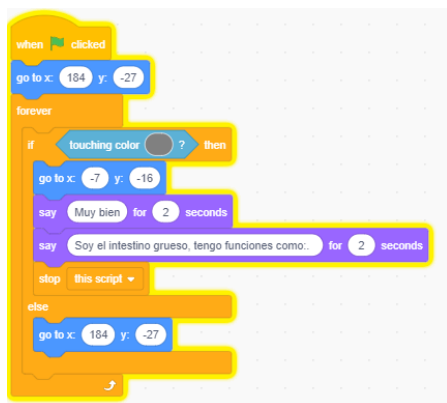


Figura 6.22: Ejemplo solución para el comportamiento del intestino grueso.

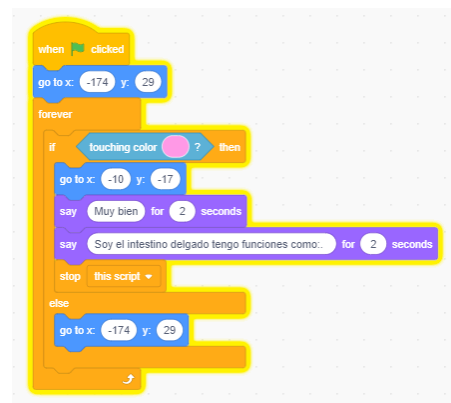


Figura 6.23: Ejemplo solución para el comportamiento del intestino delgado.

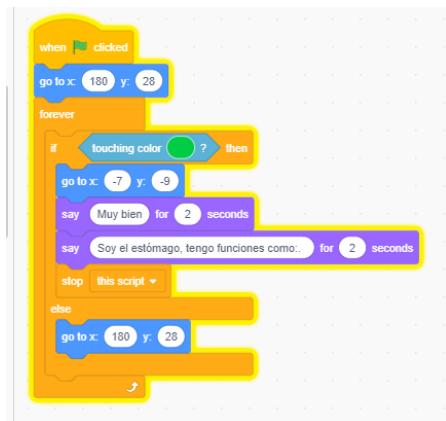


Figura 6.24: Ejemplo solución para el comportamiento del estómago.

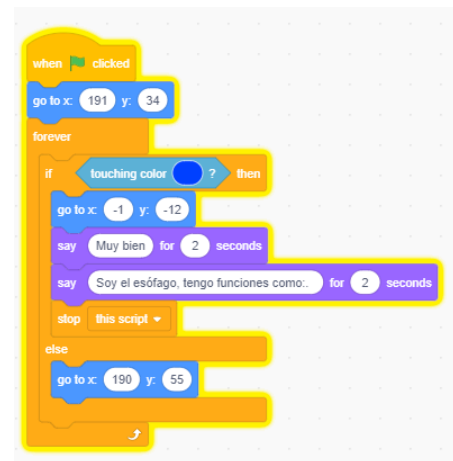


Figura 6.25: Ejemplo solución para el comportamiento del esófago.

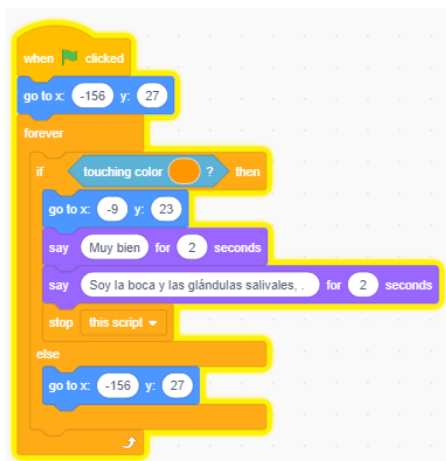


Figura 6.26: Ejemplo solución para el comportamiento de la boca y las glándulas salivales.

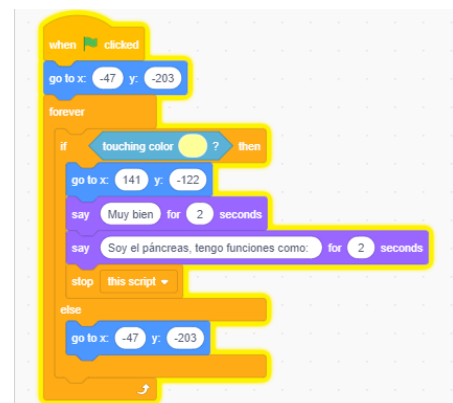


Figura 6.27: Ejemplo solución para el comportamiento del páncreas.

6.3.1.11. Evaluación

Para el caso de esta unidad, la calificación se repartirá entre la actividad presentada y la implicación vista en el aula.

Como se ha indicado en la temporalización, durante la presentación de la actividad se harán preguntas complementarias para comprobar que todos los conceptos han quedado claros y se tendrá en cuenta para la calificación.

Asimismo, al final de la evaluación estos conceptos aparecerán en el examen de la asignatura.

- **Actividad:** Supondrá un 60 % de la calificación. Dentro de este porcentaje se tendrá en cuenta si cumplen todos los criterios propuestos en la actividad:
 - 15 % Conoce y explica los componentes de los aparatos digestivo, circulatorio, respiratorio y excretor y su funcionamiento.
 - 15 % Especifica la función de cada uno de los aparatos y sistemas implicados en la funciones de relación.
 - 15 % Describe los procesos implicados en la función de relación, identificando el órgano o estructura responsable de cada proceso.
 - 15 % El alumno sabe emplear las condiciones if-else para su cometido.
- **Presentación:** Supondrá un 30 % de la calificación. Dentro de este porcentaje se tendrá en cuenta si consiguen transmitir lo que han hecho, si se desenvuelven con soltura ... de la siguiente forma:
 - 20 % Cómo se desenvuelven a la hora de hablar. 10 % será la nota global del grupo y el otro 10 % la nota de cada alumno.
 - 10 % Hablan utilizando estructuras complejas y vocabulario variado de un niño de su edad. 5 % será la nota global del grupo y el otro 5 % la nota de cada alumno.
- **Implicación:** Supondrá un 10 % de la calificación. Dentro de este porcentaje se tendrá en cuenta también el comportamiento, participación, etc.

6.3.2. El Teorema de Pitágoras

6.3.2.1. Introducción

Con esta actividad se va a trabajar la asignatura de Matemáticas orientadas a las enseñanzas aplicadas de 4º de ESO, en concreto se pretende aplicar el teorema de Pitágoras. Además, a través de esta actividad también se quiere aprovechar para emplear el concepto de evento en App Inventor.

6.3.2.2. Contextualización de la unidad didáctica

Como he indicado, esta actividad se va a realizar desde la asignatura de Matemáticas orientadas a las enseñanzas aplicadas de 4º de ESO, donde, siguiendo el Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y el Bachillerato, en el bloque 3 de **Geometría** incluye entre los contenidos a trabajar *los Teoremas de Tales y Pitágoras*.

6.3.2.3. Centro educativo: contextualización

Se trata de un instituto situado en el casco urbano de la ciudad de Valladolid, capital de la comunidad autónoma de Castilla y León, contando con una población de 300.000 habitantes, aproximadamente.

La ciudad de Valladolid cuenta con una gran oferta cultural y gastronómica, y está muy bien comunicado por tren y carretera con Madrid, así como con el resto de ciudades de la Comunidad.

El contexto social es correcto, en un barrio de nivel medio-alto, con gran dedicación al sector servicios.

El centro es un instituto de tamaño grande con enseñanza de ESO y Bachillerato, con un total de 50 profesores y 500 alumnos. La edad del profesorado es media/alta y con gran experiencia aunque anclado en lo tradicional.

6.3.2.4. Duración

Esta actividad está prevista para llevarse a cabo den 1º ESO durante la tercera evaluación del curso y tendrá una duración de tres sesiones de 50 minutos.

6.3.2.5. Criterios de evaluación

Siguiendo el Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y el Bachillerato, dentro del bloque 3 de “Geometría” con esta actividad se van a tratar los siguientes criterios de evaluación y estándares de aprendizaje evaluables.

■ Criterios de evaluación

- 1. Calcular magnitudes efectuando medidas directas e indirectas a partir de situaciones reales, empleando los instrumentos, técnicas o fórmulas más adecuadas, y aplicando, así mismo, la unidad de medida más acorde con la situación descrita.
- * Recoger los campos de entrada y emplearlos para calcular y mostrar la respuesta deseada.

■ Estándares de aprendizaje evaluables

- 1.4. Calcula medidas indirectas de longitud, área y volumen mediante la aplicación del teorema de Pitágoras y la semejanza de triángulos.
- * El alumno sabe recoger los campos de entrada y emplearlos para calcular y mostrar la respuesta deseada.

* Se han añadido estos criterios de evaluación relativos al aprendizaje propio del PC que se quiere transmitir en esta actividad.

6.3.2.6. Metodología

La metodología que se pretende seguir es puramente activa donde los alumnos aprendan haciendo, basando todo el contenido en un aprendizaje orientado a proyectos.

Siguiendo esta forma de trabajo, se plantea un ejercicio con el que los alumnos desarrollen un ejercicio de aplicación del teorema de Pitágoras con el objeto de calcular la longitud de la hipotenusa correctamente y, además, puedan afianzar el concepto de evento.

Puede dividirse el proceso en 4 fases:

- **Instrucción:** El profesor explica una introducción sobre el teorema de Pitágoras.
- **Presentación:** El profesor presenta el ejercicio.
- **Realización:** Los alumnos realizan el ejercicio.
- **Evaluación:** Los alumnos presentan sus ejercicios y el profesor realiza algunas preguntas para poder evaluarlo.

Esta actividad implica trabajo en equipo en todas las fases excepto en la inicial de instrucción.

6.3.2.7. Medidas de atención a la diversidad

La atención a la diversidad es el conjunto de actuaciones educativas dirigidas a dar respuesta a las diferentes capacidades, ritmos y estilos de aprendizaje, motivaciones e intereses, situaciones sociales, culturales, lingüísticas y de salud del alumnado.

Tan pronto como se detecten dificultades de aprendizaje en un alumno se pondrán en marcha medidas de carácter ordinario, adecuando la programación a las necesidades del alumnado, adaptando actividades, metodología o temporalización y, en su caso, realizando adaptaciones no significativas del currículo.

Están incluidas las adaptaciones curriculares tanto para el alumnado con necesidades educativas especiales como con altas capacidades intelectuales.

Alumnado con necesidades educativas especiales Son aquellos alumnos que requieran durante un periodo de su escolarización o a lo largo de toda la etapa, determinados apoyos y atenciones educativas específicas derivadas de discapacidad o trastornos graves de conducta, de acuerdo con el correspondiente dictamen de escolarización.

La identificación y valoración de las necesidades educativas especiales requerirá la realización de una evaluación psicopedagógica.

La Concreción Curricular dentro del Proyecto Educativo del centro ha de identificar qué necesidades específicas tiene su alumnado para que el profesorado pueda ajustar el currículo a las características del grupo-aula en la programación.

El departamento de Orientación pondrá a disposición del profesorado los informes de la evaluación psicopedagógica de los alumnos con necesidades educativas especiales para que una vez conocidas sus capacidades, el profesorado puede adaptar la programación a las necesidades del alumno.

Para que este alumnado pueda alcanzar el máximo desarrollo de sus capacidades personales y los objetivos de la etapa, se establecerán dentro de los principios de inclusión y

normalidad, las medidas organizativas y curriculares, que aseguren su adecuado progreso y el máximo logro de los objetivos.

El plan de trabajo individualizado para este alumnado concretará las medidas de compensación y de estimulación, así como las materias en las que precise adaptación curricular, especificando las tareas a realizar por cada profesional.

En el caso del alumnado con necesidades educativas especiales que requiera adaptaciones curriculares significativas, éstas se elaborarán con el nivel de exigencia de la calidad y cantidad del resultado final y de la información que sean capaces de transmitir.

Las adaptaciones significativas de los elementos del currículo se realizarán buscando el máximo desarrollo posible de las competencias; la evaluación continua y la promoción tomarán como referente los elementos fijados en dichas adaptaciones. En cualquier caso, el alumnado con adaptaciones curriculares significativas deberá superar la evaluación final para poder obtener el título correspondiente.

Alumnado con altas capacidades intelectuales La atención educativa al alumnado con altas capacidades intelectuales se desarrollará, en general, a través de medidas de adecuación del currículo, de enriquecimiento y/o de ampliación curricular, con la finalidad de promover un desarrollo equilibrado de las distintas capacidades establecidas en los objetivos de la etapa, así como de conseguir un desarrollo pleno y equilibrado de sus potencialidades y de su personalidad.

6.3.2.8. Temporalización

Sesión	Duración	Actividad
Sesión 1	10 min.	Juego donde haya que aplicar el Teorema de Pitágoras.
	10 min.	Debate sobre lo realizado.
	30 min.	Explicación magistral sobre el Teorema de Pitágoras
Sesión 2	5 min.	Presentación de la actividad a realizar.
	45 min.	Trabajo en la actividad por parte de los alumnos.
Sesión 3	15 min.	Trabajo en la actividad por parte de los alumnos.
	35 min.	Presentación de las actividades realizadas por los alumnos.

Tabla 6.5: Temporalización de la actividad Teorema de Pitágoras.

6.3.2.9. Materiales

En la asignatura no se usará ningún libro texto. El seguimiento de la asignatura se realizará a través de una plataforma LMS (Moodle) en la que están la guía de los

conceptos clave de cada Unidad, los materiales necesarios para cada práctica, así como los trabajos entregados por los alumnos.

Se proporcionarán textos relacionados con los contenidos que se van a desarrollar en la unidad, acompañados de cuestiones sobre la lectura que susciten la reflexión, la opinión y el debate sobre los aspectos y los conocimientos relacionados con la misma.

También se proporcionarán imágenes, gráficas y fotografías que permiten visualizar lo que se está explicando en el texto.

Por otro lado, también se visualizarán vídeos aclarativos sobre los puntos tratados en el texto central.

Además, se realizarán y/o adjuntarán cuadros, tablas, esquemas que recojan esquemáticamente algunos contenidos.

En cuanto a los recursos del aula, se hará uso de:

- Ordenadores con acceso a Internet y la aplicación App Inventor instalada.
- Proyector
- Conexión a Internet

6.3.2.10. Desarrollo de la actividad

Este proyecto se centra principalmente en la actividad planteada a los alumnos, que desarrollan entre las sesiones 2 y 3.

Materiales y recursos Los materiales para esta actividad son los mismos que de forma global:

- Ordenadores con acceso a Internet y la aplicación App Inventor instalada.
- Proyector para mostrar la actividad de cada grupo de alumnos
- Conexión a Internet

Descripción Una vez explicado el Teorema de Pitágoras a los alumnos en la primera sesión, la actividad consiste en que los alumnos, por grupos de 4, hagan una aplicación con App Inventor similar a la mostrada en la figura 6.28, donde se recoge el valor de los catetos y se muestra el valor de la hipotenusa al pulsar en el botón.

The image shows a simple web application interface. At the top, there is a label "Introduce el valor del cateto" above a text input field. Below this, there is another label "Introduce el valor del otro cateto" above a second text input field. At the bottom center, there is a button labeled "calcular".

Figura 6.28: Ejemplo visual de solución para el ejercicio con App Inventor.

En la figura 6.29 se muestra un ejemplo de solución similar al que deben realizar los alumnos en la actividad.

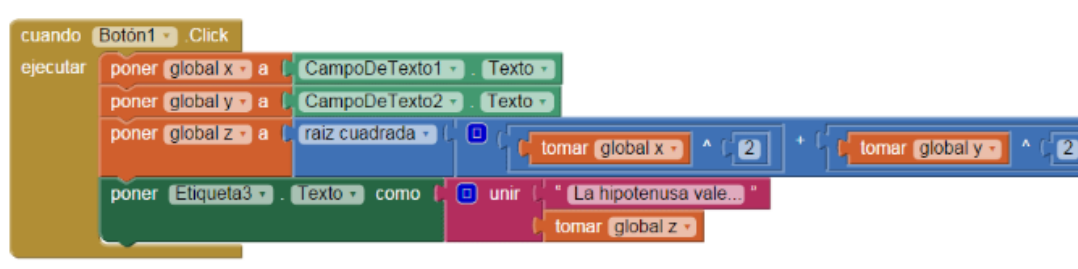


Figura 6.29: Ejemplo de solución para la aplicación del Teorema de Pitágoras.

6.3.2.11. Evaluación

Para el caso de esta unidad, la calificación se repartirá entre la actividad presentada y la implicación vista en el aula.

Como se ha indicado en la temporalización, durante la presentación de la actividad se harán preguntas complementarias para comprobar que todos los conceptos han quedado claros y se tendrá en cuenta para la calificación.

Asimismo, al final de la evaluación estos conceptos aparecerán en el examen de la asignatura.

- **Actividad:** Supondrá un 60 % de la calificación. Dentro de este porcentaje se tendrá en cuenta si cumplen todos los criterios propuestos en la actividad:
 - 45 % Calcula medidas indirectas de longitud, área y volumen mediante la aplicación del teorema de Pitágoras y la semejanza de triángulos.
 - 15 % El alumno sabe recoger los campos de entrada y emplearlos para calcular y mostrar la respuesta deseada.
- **Presentación:** Supondrá un 30 % de la calificación. Dentro de este porcentaje se tendrá en cuenta si consiguen transmitir lo que han hecho, si se desenvuelven con soltura ... de la siguiente forma:
 - 20 % Cómo se desenvuelven a la hora de hablar. 10 % será la nota global del grupo y el otro 10 % la nota de cada alumno.
 - 10 % Hablan utilizando estructuras complejas y vocabulario variado de un niño de su edad. 5 % será la nota global del grupo y el otro 5 % la nota de cada alumno.
- **Implicación:** Supondrá un 10 % de la calificación. Dentro de este porcentaje se tendrá en cuenta también el comportamiento, participación, etc.

Capítulo 7

Conclusiones

Más vale un bit de reflexión que un megabyte de programación.

Fuente desconocida.

A lo largo de este documento se ha hecho especial hincapié en el ritmo con el que avanza la sociedad, sobre todo a nivel tecnológico y la necesidad que tiene de estar preparada para poder afrontar los retos que están por llegar.

Es por todo ello que el PC es una base que debe estar presente en todos los jóvenes que representan el futuro y que lo necesitarán para conseguir seguir avanzando y no quedarse atrás.

Vistos los evidentes beneficios que aporta (razonamiento lógico, pensamiento abstracto y algorítmico, resolución de problemas complejos . . .) para la vida real en todas las disciplinas, el foco y principal preocupación de este documento es la forma de llevarlo al sistema educación actual europeo y, más concretamente, español. Esto, tal y como se ha podido ver, es el auténtico reto educativo al que se está enfrentando en la actualidad la educación formal en la gran mayoría de países.

Dentro de la educación no formal, respaldadas por empresas tecnológicas importantes, se han mostrado múltiples iniciativas que están llevando el pensamiento computacional hasta los más jóvenes.

Sin embargo, la educación formal está unos cuantos más atrás, aunque avanzando poco a poco, como se ha podido ver en los currículos educativos de auténticos pioneros como Reino Unido o Finlandia, que lo han hecho de formas totalmente diferentes, el primero fijando una asignatura con contenidos a lo largo de todos los cursos de primaria y secundaria y, el otro, repartiendo los contenidos de manera transversal entre el resto de materias.

Por otro lado, una vez mostrado cómo están intentando cambiar algunos países que lo están haciendo con éxito, se ha visto la situación actual del currículo en España para ver dónde se encuentra el punto de partida desde el que hay que comenzar a trabajar.

Finalmente, se han planteado dos actividades con las dos herramientas basadas en

bloques más utilizadas en la actualidad, principalmente en la educación no formal, como son Scratch y App Inventor. Con estas se ha tratado de poner algunos ejemplos de cómo emplear en cualquier materia este tipo de herramientas y complementarlo con conceptos de pensamiento computacional con el objetivo de acercar el pensamiento computacional y la programación a los alumnos de secundaria de manera transversal.

Referencias

- ALLAN, W., COULTER, B., DENNER, J., ERICKSON, J., LEE, I., MALYN-SMITH, J. y MARTIN, F. Computational thinking for youth. 2010.
- BARR, D., HARRISON, J. y CONERY, L. Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, vol. 38(6), páginas 20–23, 2011.
- BOCCONI, S., CHIOCCARIELLO, A., DETTORI, G., FERRARI, A. y ENGELHARDT, K. Developing computational thinking in compulsory education. 2016. Disponible en <https://playcodeacademy.com/10-ventajas-programacion-informatica-para-ninos/> (último acceso, Diciembre, 2019).
- BROWN, N. C. C. y WILSON, G. Ten quick tips for teaching programming. *PLoS Computational Biology*, 2018. Disponible en <https://doi.org/10.1371/journal.pcbi.1006023> (último acceso, Diciembre, 2019).
- CONSEJERÍA DE EDUCACIÓN DE LA JUNTA DE CASTILLA Y LEÓN. Orden EDU/362/2015, de 4 de mayo, por la que se establece el currículo y se regula la implantación, evaluación y desarrollo de la educación secundaria obligatoria en la comunidad de castilla y león. 2015. Disponible en <https://www.educa.jcyl.es/es/resumenbocyl/orden-edu-362-2015-4-mayo-establece-curriculo-regula-implan> (último acceso, Diciembre, 2019).
- CONSEJERÍA DE EDUCACIÓN DE LA JUNTA DE CASTILLA Y LEÓN. Orden EDU/589/2016, de 22 de junio, por la que se regula la oferta de materias del bloque de asignaturas de libre configuración autonómica en tercer y cuarto curso de educación secundaria obligatoria, se establece su currículo y se asignan al profesorado de los centros públicos y privados en la comunidad de castilla y león. 2016. Disponible en <http://bocyl.jcyl.es/boletines/2016/06/27/pdf/B0CYL-D-27062016-4.pdf> (último acceso, Diciembre, 2019).
- CSIZMADIA, A., CURZON, P., DORLING, M., HUMPHREYS, S., NG, T., SELBY, C. y WOOLLARD, J. Computational thinking. a guide for teachers. *Computing at School*, 2015.

- DEEK, F., JONES, J., MCCOWAN, D., STEPHENSON, C. y VERNO, A. A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee. *Computer Science Teacher Association*, 2003. Disponible en http://eduteka.icesi.edu.co/articulos/Ciencia_Computacion (último acceso, Diciembre, 2019).
- DUFFANY, L., JEFFREY. Choice of Language for an Introduction to Programming Course. *Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology*, 2014. Disponible en <http://www.laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP138.pdf> (último acceso, Diciembre, 2019).
- FUNDACIÓN ESPAÑOLA PARA LA CIENCIA Y LA TECNOLOGÍA, GOOGLE Y EVERIS. Educación en Ciencias de la Computación en España 2015. 2016. Disponible en <https://www.fecyt.es/es/publicacion/educacion-de-las-ciencias-de-la-computacion-en-espana> (último acceso, Diciembre, 2019).
- IBÁÑEZ, M. B., DI-SERIO, N. y DELGADO KLOOS, C. Gamification for Engaging Computer Science Students in Learning Activities: A Case Study. *IEEE Transactions on Learning Technologies*, vol. 7(3), páginas 291–301, 2014. Disponible en <https://ieeexplore.ieee.org/document/6827214> (último acceso, Diciembre, 2019).
- INSTITUTO NACIONAL DE TECNOLOGÍAS EDUCATIVAS Y DE FORMACIÓN DEL PROFESORADO. Programación, Robótica y Pensamiento Computacional en el aula. 2018. Disponible en <http://code.intef.es/wp-content/uploads/2018/10/Ponencia-sobre-Pensamiento-Computacional.-Informe-Final.pdf> (último acceso, Diciembre, 2019).
- LEE, I., MARTIN, F., DENNER, J., COULTER, B., ALLAN, W., ERICKSON, J., MALYN-SMITH, J. y WERNER, L. Computational thinking for youth in practice. *Inroads*, vol. 2(1), páginas 32–37, 2011.
- LEÓN, J. M. ¿qué lenguaje es más efectivo para desarrollar el pensamiento computacional: Scratch o app inventor? 2019. Disponible en <https://programamos.es/que-lenguaje-es-mas-efectivo-para-desarrollar-el-pensamiento-computacional-scratch-o-app> (último acceso, Diciembre, 2019).
- LÓPEZ PERNAS, S., GORDILLO, A., BARRA, E. y QUEMADA, J. Examining the Use of an Educational Escape Room for Teaching Programming in a Higher Education Setting. *IEEE Access*, vol. 7, páginas 31723–31737, 2019. Disponible en <https://ieeexplore.ieee.org/document/8658086> (último acceso, Diciembre, 2019).
- MEERBAUM-SALANT, O., ARMONI, M. y BEN-ARI, M. M. Learning computer science concepts with scratch. *Computer Science Education*, vol. 23(3), páginas 239–264, 2013. Disponible en <https://doi.org/10.1080/08993408.2013.832022> (último acceso, Diciembre, 2019).

- PAPADAKIS, S. J., KALOGIANNAKIS, M., ORFANAKIS, V. y ZARANIS, N. Using scratch and app inventor for teaching introductory programming in secondary education. a case study. *International Journal of Technology Enhanced Learning*, vol. 8(3), 2016.
- PAPADAKIS, S. J., KALOGIANNAKIS, M., ORFANAKIS, V. y ZARANIS, N. The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. *International Journal of Web-Based Learning and Teaching Technologies*, vol. 12(4), páginas 58–77, 2017.
- PARK, Y. y SHIN, Y. Comparing the effectiveness of scratch and app inventor with regard to learning computational thinking concepts. 2019. Disponible en <https://www.mdpi.com/2079-9292/8/11/1269/pdf> (último acceso, Diciembre, 2019).
- PIEDEDE, J., DOROTEA, N., FERRENTINI, F. S. y PEDRO, A. A cross-analysis of block-based and visual programming apps with computer science student-teachers. 2019. Disponible en <https://www.mdpi.com/2227-7102/9/3/181> (último acceso, Diciembre, 2019).
- PLAYCODE ACADEMY. 10 ventajas de los niños que deciden aprender a programar. 2017. Disponible en <https://playcodeacademy.com/10-ventajas-programacion-informatica-para-ninos/> (último acceso, Diciembre, 2019).
- POSTOLOVSKI, T. How to Teach Programming. 2018. Disponible en <https://www.codementor.io/npostolovski/how-to-teach-programming-pvpgdtoed> (último acceso, Diciembre, 2019).
- RIOJAS, M., LYSECKY, S. y ROZENBLIT, J. Educational Technologies for Precollege Engineering Education. *IEEE Transactions on Learning Technologies*, vol. 5(1), páginas 20–37, 2011. Disponible en <https://ieeexplore.ieee.org/document/5744071> (último acceso, Diciembre, 2019).
- RUBY, I. y DAVIDSON, A.-L. The promise of the "learn to code" movement. 2018. Disponible en <https://theconversation.com/the-promise-of-the-learn-to-code-movement-107836> (último acceso, Diciembre, 2019).
- SCRATCH WIKI. Alternatives to scratch. 2019. Disponible en https://en.scratch-wiki.info/wiki/Alternatives_to_Scratch (último acceso, Diciembre, 2019).
- SHRIYA, S., ASHWINI, B., CHANDRAN, A. y SOMAN, K. P. Computational Thinking leads to computational learning: Flipped classroom experiments in linear algebra. *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2015. Disponible en <https://ieeexplore.ieee.org/document/7193021> (último acceso, Diciembre, 2019).

- SILVA MACEDA, G., ARJONA VILLICAÑA, P. D. y CASTILLO BARRERA, F. E. More Time or Better Tools? A Large-Scale Retrospective Comparison of Pedagogical Approaches to Teach Programming. *IEEE Transactions on Education*, vol. 59(4), páginas 274–281, 2016. Disponible en <https://ieeexplore.ieee.org/document/7440799> (último acceso, Diciembre, 2019).
- SUN, Q., WU, J., RONG, W. y LIU, W. Formative assessment of programming language learning based on peer code review: Implementation and experience report. *Tsinghua Science and Technology*, vol. 24(4), páginas 423–434, 2019. Disponible en <https://ieeexplore.ieee.org/document/8660528> (último acceso, Diciembre, 2019).
- WEBB, M., DAVIS, N., BELL, T., KATZ, Y. J., REYNOLDS, N., SYS?O, M. M. y CHAMBERS, D. P. Towards deeper understanding of the roles of cs informatics in the curriculum. 2015.
- WEBB, M., DAVIS, N., BELL, T., KATZ, Y. J., REYNOLDS, N., SYS?O, M. M. y CHAMBERS, D. P. Computer science in k-12 school curricula of the 21st century: Why, what and when? *Education and Information Technologies*, vol. 22(2), 2017. Disponible en <https://doi.org/10.1007/s10639-016-9493-x> (último acceso, Diciembre, 2019).
- WEINTROP, D., BEHESHTI, E., HORN, M., ORTON, K., JONA, K., TROUILLE, L. y WILENSKY, U. Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, vol. 25(1), páginas 127–147, 2016.
- WESTBROOK, D. S. A multiparadigm language approach to teaching principles of programming languages. *FIE'99 Frontiers in Education*, 1999. Disponible en <https://ieeexplore.ieee.org/document/839221> (último acceso, Diciembre, 2019).
- WING, J. M. Computational thinking. *Commun*, vol. 49, páginas 33–35, 2006.
- WING, J. M. Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, vol. 25(2), páginas 7–14, 2017.
- WOOLLARD, J. Ct driving computing curriculum in england. *CSTA Voice*, vol. 12(1), páginas 4–5, 2016.
- YADAV, A., MAYFIELD, C., ZHOU, N., HAMBRUSCH, S. E. y KORB, J. T. Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, vol. 14(1), 2014. Disponible en <http://dx.doi.org/10.1145/2576872> (último acceso, Diciembre, 2019).

Índice alfabético

- abstracción, 5
- ActionScript, 25
- actitud, 4–6
- Al Companion, 30
- algoritmo, 19
- Alice, 22, 23
- All you need is code, 11
- análisis, 6, 16
- analizar, 2
- Android, 29
- apariencia, 27
- App Inventor, 13, 23, 29, 30
- automatización, 5
- autonomía, 2

- Barefoot Computing, 11
- barrera, 22, 23, 27
- Bebras, 10
- biología y geología, 30
- Blockly, 23
- bloque, 22
- bucle, 28
- BYOB/Snap, 23

- C, 25
- C++, 25
- C#, 25
- característica, 7
- CLOQQ, 13
- Code Club, 10
- Code it Like a Girl, 12
- Code.org, 9
- CoderDojo, 9
- Computing At School, 12
- condición, 28, 30
- condicional, 32

- confianza, 6
- consumo, 8
- contenido, 8
- contexto, 31, 38
- control, 27
- control del flujo, 30
- Crea y Explora, 14
- creador, 8, 21
- Creando Código - ScolaTIC, 13
- creatividad, 2, 6, 14, 17, 21
- criterios de evaluación, 31, 39
- CS Unplugged, 10
- cuerpo humano, 30
- currículo, 2, 16–18, 32

- dato, 27
- definición, 5, 7
- depuración, 6
- desarrollo, 35, 42
- descomposición, 6
- disciplina, 5, 19, 21
- diversidad, 32, 40
- duración, 31, 39

- educación formal, 16
- El cuerpo humano, 30
- empleo, 2, 8
- equipo, 6
- Escuela TIC, 14
- estereotipo, 2
- estrategia, 21
- estructurado, 8
- evaluación, 6, 32, 37, 43
- evento, 27, 28
- experimentar, 2

- Finlandia, 18
- Gamefroot, 23
- GameMaker Studio, 23
- GameSalad, 23
- generalización, 6
- Google Code Jam, 10
- habilidad, 5–7, 16, 18, 21
- herramienta, 1–5, 22, 27, 29
- hilo, 28
- Hopscotch, 23
- importancia, 7
- iniciativa, 2
- instrucción, 28, 32
- interacción, 30
- Java, 25
- JavaScript, 26
- KGBL3, 13
- lápiz, 27
- lógica, 8
- lenguaje, 7
- Logo, 1, 22, 26
- Lua, 25
- móvil, 30
- Made with Code, 10
- material, 34, 41
- metodología, 3, 5, 14, 32, 39
- Microsoft Hacking STEM, 10
- Microsoft MakeCode, 11
- miedo, 6
- motivación, 2, 22
- movimiento, 27
- objetivo, 2
- Olimpiada Informática Española, 13
- Olimpiada Informática Internacional, 10
- operador, 27
- organizar, 2
- paralelismo, 30
- paso, 5
- pensamiento abstracto, 2
- pensamiento algorítmico, 2, 5
- pensamiento científico, 18
- pensamiento computacional, 1, 2, 4, 7, 13, 16, 18, 19, 27
- pensamiento lógico, 6
- pensar, 2
- perseverancia, 6
- Pocket Code, 23
- presentación, 32
- problema, 4–6, 8, 19
- proceso mental, 8
- profesional, 25
- programación, 1–3, 7, 8, 12–14, 16–20, 22, 26
- Programamos, 12
- Python, 25
- QB64, 26
- razonamiento, 2
- realidad virtual, 14
- realización, 32
- Reino Unido, 17
- relacionar, 2
- representación, 30
- resultado, 23
- reto, 2
- robótica, 12–14, 18, 19
- robot, 19
- Ruby / Ruby on Rails, 26
- Scratch, 12, 13, 22, 23, 26, 30
- secundaria, 2, 13, 14, 17, 22
- Semana de la Programación de la UE, 11
- sensor, 27
- sintaxis, 23
- sistema de información, 4
- solución, 1, 2, 4–6, 8
- sonido, 27
- sprite, 26
- STEAM, 14
- STEM, 2, 11

Stencyl, 23

técnica, 5

tecnología, 7, 14, 18, 21

tecnologías de la información, 19

temporalización, 34, 41

tendencia, 7

Teorema de Pitágoras, 38

Tynker, 23

variable, 28

videojuego, 12, 14

Glosario de acrónimos

CLOQQ	Crea Lo Que Quiera
CS	<i>Computing Science</i>
CSTA	<i>Computer Science Teachers Association</i>
DIY	<i>Do It Yourself</i>
ESO	Educación Secundaria Obligatoria
ISTE	<i>International Society for Technology in Education</i>
MIT	<i>Massachusetts Institute of Technology</i>
NCR	<i>National Council for Research</i>
PC	<i>Computational Thinking</i> - Pensamiento Computacional
STEAM	<i>Science, Technology, Engineering, Arts and Maths</i> - Ciencias Naturales, Tecnología, Arte, Diseño y Matemáticas
STEM	<i>Science, Technology, Engineering and Mathematics</i> - Ciencias Naturales, Tecnología y Matemáticas
TIC	Tecnologías de la Información y las Comunicaciones
UE	Unión Europea
UNED	Universidad Nacional de Educación a Distancia