

# unir

UNIVERSIDAD  
INTERNACIONAL  
DE LA RIOJA

**Universidad Internacional de La Rioja**  
**Facultad de Educación**

---

## APRENDER A PROGRAMAR EN LA ESCUELA PARA LA MEJORA COMPETENCIAL EN EDUCACIÓN PRIMARIA

*Estudio teórico y pautas de formación docente*

---

Trabajo fin de grado presentado por: Olga Terrés Pérez

Titulación: Maestro/a en Educación Primaria

Línea de investigación: Propuesta de Intervención

Directora: Dra. M<sup>a</sup> Rocío Díaz Gómez.

Ciudad: Barcelona

Junio de 2015

Firmado por: Olga Terrés Pérez

CATEGORÍA TESAURO: 1.1.1 Medios audiovisuales y nuevas tecnologías aplicadas a la educación; 1.3.3 La formación continua del profesorado; 1.5.3 Etapas educativas; 1.7.4 Tecnologías de la Información y la Comunicación (TIC).

## **Resumen**

El presente trabajo de fin de grado pretende contribuir a la divulgación de la situación actual de la incorporación de la enseñanza de la programación de ordenadores en la enseñanza primaria. Para ello se revisa la legislación educativa vigente en España al respecto así como las directrices y propuestas de la Unión Europea, complementando esta información con algunas iniciativas destacables en este terreno. Tras un resumen de los principales beneficios que, según los expertos, aporta este tipo de aprendizaje, se presenta el software de uso más generalizado en estas iniciativas en el ámbito de la educación primaria.

Así mismo, el trabajo aporta una propuesta metodológica de intervención para la capacitación del profesorado de primaria en la docencia de la programación con el objeto de que puedan incorporar estos conocimientos a su práctica docente así como tener una opinión crítica fundamentada sobre este tipo de propuestas.

## **Palabras clave**

Programación, Código, Scratch, Pensamiento Computacional, Educación Primaria

## ÍNDICE DE CONTENIDOS

<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1. JUSTIFICACIÓN .....	1
1.2. OBJETIVOS E HIPÓTESIS .....	2
1.2.1. <i>Objetivo general</i> .....	2
1.2.2. <i>Objetivos específicos</i> .....	2
1.2.3. <i>Hipótesis de partida</i> .....	3
1.3. METODOLOGÍA DEL TRABAJO.....	3
<b>2. MARCO TEÓRICO EN TORNO AL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA PROGRAMACIÓN EN EDUCACIÓN PRIMARIA.....</b>	<b>4</b>
2.1. MARCO CONCEPTUAL Y TERMINOLÓGICO EN TORNO A LA PROGRAMACIÓN EN EDUCACIÓN PRIMARIA .....	4
2.1.1. <i>Definiciones y traducción de términos</i> .....	4
2.1.2. <i>Desarrollo de la Informática</i> .....	5
2.1.3. <i>Internet y la expansión de las Tecnologías de la Información y la Comunicación (TIC)</i> .	6
2.1.4. <i>Usuarios de los medios digitales: De consumidores a creadores</i> .....	6
2.2. ANÁLISIS DE LA ENSEÑANZA DE LA PROGRAMACIÓN EN LA NORMATIVA LEGAL DE EDUCACIÓN PRIMARIA .....	8
2.2.1. <i>Posicionamiento de la Unión Europea al respecto</i> .....	8
2.2.1.1. <i>Competencias Clave para el aprendizaje en la Unión Europea</i> .....	9
2.2.2. <i>Legislación educativa vigente en España para la educación Primaria</i> .....	10
2.2.2.1. <i>Legislación autonómica en esta materia</i> .....	10
2.2.2.2. <i>Objetivos y Competencias Básicas en la legislación española</i> .....	10
2.3. BENEFICIOS DEL APRENDIZAJE DE LA PROGRAMACIÓN EN LA ETAPA DE EDUCACIÓN PRIMARIA .....	11
2.3.1. <i>Motivación: Impulso creativo</i> .....	11
2.3.2. <i>Perspectiva constructivista del aprendizaje: Aprender haciendo</i> .....	12
2.3.3. <i>Teoría de las inteligencias múltiples: Aprendizaje transversal</i> .....	12
2.3.4. <i>Pequeños creadores sin miedo al error</i> .....	13
2.3.5. <i>Apuesta por las competencias sobre los contenidos</i> .....	13
2.3.6. <i>El pensamiento computacional</i> .....	14
2.3.7. <i>Fomento del trabajo colaborativo</i> .....	14
2.3.8. <i>Retos para el profesorado ante la enseñanza en el medio digital</i> .....	15

2.4. SOFTWARE Y APLICACIONES PARA ENSEÑAR A PROGRAMAR EN EDUCACIÓN PRIMARIA .....	15
2.4.1. <i>Los orígenes de la programación para niños: El lenguaje Logo</i> .....	15
2.4.2. <i>El lenguaje de uso más extendido actualmente en Educación Primaria: Scratch</i> .....	16
2.4.2.1. Origen y principales características .....	16
2.4.2.2. Funcionalidad standard .....	17
2.4.2.3. Funcionalidad ampliada hacia la robótica educativa .....	19
2.4.3. <i>Otros lenguajes de programación adecuados para la etapa de primaria</i> .....	20
2.5. ALGUNAS EXPERIENCIAS Y PROYECTOS DESTACABLES RELACIONADOS CON LA TEMÁTICA .....	21
2.5.1. <i>Propuestas o experiencias de alcance europeo</i> .....	21
2.5.1.1. Iniciativas en el marco de la Comisión Europea .....	21
2.5.1.1.1. Estrategia Europa 2020: El origen de la Agenda Digital Europea .....	21
2.5.1.1.2. Propuestas de la Agenda Digital Europea en materia educativa .....	22
2.5.1.2. Iniciativas privadas de alcance europeo .....	24
2.5.1.2.1. Proyecto CodeClubWorld desde el Reino Unido .....	24
2.5.1.2.2. Proyecto CoderDojo desde Irlanda .....	26
2.5.2. <i>Propuestas o experiencias en España</i> .....	27
2.5.2.1. Propuesta Código 21 del Gobierno de Navarra .....	27
2.5.2.2. Asociación Programamos .....	28
2.6. CONCLUSIONES DEL APARTADO .....	30
<b>3. PAUTAS METODOLÓGICAS DE INTERVENCIÓN PARA APRENDER A PROGRAMAR EN EL AULA DE EDUCACIÓN PRIMARIA .....</b>	<b>31</b>
3.1. INTRODUCCIÓN .....	31
3.2. CONTEXTO DE INTERVENCIÓN .....	32
3.3. OBJETIVOS DE LA INTERVENCIÓN .....	33
3.4. DESTINATARIOS DE LA INTERVENCIÓN .....	33
3.5. DESARROLLO DE LA PROPUESTA DE INTERVENCIÓN .....	33
3.5.1. <i>Relación de Sesiones y Actividades</i> .....	33
3.5.1.1. Sesión 1: Aprender a programar ¿Para qué? .....	33
3.5.1.2. Sesión 2: ¿Qué es Scratch? .....	34
3.5.1.3. Sesión 3: Visión general de entorno de programación de Scratch .....	35
3.5.1.4. Sesión 4: Tipos de Bloques apilables (Instrucciones Scratch) .....	35
3.5.1.5. Sesión 5: Trabajando con Scratch: Hacer una composición visual .....	36
3.5.1.6. Sesión 6: Trabajando con Scratch: Hacer una composición audiovisual .....	36
3.5.1.7. Sesión 7: Trabajando con Scratch: Conversando con mi mascota .....	37
3.5.1.8. Sesión 8: Trabajando con Scratch: El gato se sabe las tablas de multiplicar .....	37
3.5.1.9. Sesión 9: Trabajando con Scratch: Dibujando figuras geométricas .....	38

3.5.1.10. Sesión 10: Comunidad Scratch: Compartir proyectos en la red .....	38
3.5.1.11. Sesión 11: Comunidades educativas virtuales: Apoyo al profesorado en la red .....	39
3.5.1.12. Sesión 12: Conclusiones .....	39
<b>3.5.2. Temporalización y Recursos .....</b>	<b>40</b>
3.5.2.1. Temporalización .....	40
3.5.2.2. Recursos materiales .....	40
3.5.2.3. Recursos profesionales .....	41
3.5.2.4. Recursos didácticos .....	41
<b>3.5.3. Metodología .. ..</b>	<b>41</b>
<b>3.5.4. Evaluación .</b> .....	<b>42</b>
<b>4. CONCLUSIONES, LIMITACIONES Y PROSPECTIVA.....</b>	<b>42</b>
4.1. CONCLUSIONES GENERALES .....	42
4.2. LIMITACIONES DEL ESTUDIO.....	44
4.3. FUTURAS LÍNEAS DE INVESTIGACIÓN .....	45
<b>5. FUENTES BIBLIOGRÁFICAS .....</b>	<b>46</b>
5.1. REFERENCIAS .....	46
5.2. BIBLIOGRAFÍA RECOMENDADA.....	51

## ÍNDICE DE TABLAS E IMÁGENES

TABLA 1. FASES DEL PROCESO INVESTIGADOR Y TEMPORALIZACIÓN (ADAPTADA DE RODRÍGUEZ Y OTROS, 1996) .....	4
TABLA 2. INSTRUCCIONES DE USO MÁS FRECUENTE EN EL LENGUAJE SCRATCH (ELABORACIÓN PROPIA) .....	18
TABLA 3. LENGUAJES DE PROGRAMACIÓN MÁS UTILIZADOS ACTUALMENTE CON FINES EDUCATIVOS (ELABORACIÓN PROPIA) .....	20
IMAGEN 1. CAPTURA DE LA PANTALLA INICIAL DEL PROGRAMA SCRATCH (ELABORACIÓN PROPIA)...	17
IMAGEN 2. PROGRAMA SCRATCH MUY SENCILLO PARA MOVER UN HELICÓPTERO CON LAS FLECHAS DEL TECLADO(ELABORACIÓN PROPIA) .....	19
IMAGEN 3. CAPTURA PANTALLA DE LA WEB ‘ALL YOU NEED IS CODE’ DE LA INICIATIVA EUROPEA DE PROGRAMACIÓN (ELABORACIÓN PROPIA) .....	23
IMAGEN 4. CAPTURA DE PANTALLA DE LA WEB DE LA EUROPEAN SCHOOLNET ACADEMY (ELABORACIÓN PROPIA) .....	24

IMAGEN 5. CAPTURA DE PANTALLA DE LA WEB DE CODECLUBWORLD (ELABORACIÓN PROPIA).....	25
IMAGEN 6. CAPTURA DE PANTALLA DE LA WEB DE CODERDOJO (ELABORACIÓN PROPIA) .....	26
IMAGEN 7. CAPTURA DE PANTALLA DE LA WEB DE CÓDIGO 21 DEL GOBIERNO DE NAVARRA (ELABORACIÓN PROPIA) .....	27
IMAGEN 8. CAPTURA DE PANTALLA DE LA WEB DE PROGRAMAMOS (ELABORACIÓN PROPIA) .....	29

# 1. INTRODUCCIÓN

## 1.1. Justificación

En los últimos años estamos asistiendo a una creciente proliferación de iniciativas que promueven la incorporación de la enseñanza de la programación en el ámbito escolar.

Al mismo tiempo observamos que salvo los investigadores y profesionales directamente involucrados en estos proyectos, el resto de docentes y de la comunidad educativa en general desconocen en gran medida tanto las motivaciones como los contenidos y el alcance de estas propuestas.

Este trabajo trata de aportar una visión sintética sobre el origen, la naturaleza y el grado de desarrollo actual de estas iniciativas, cuya idea esencial es que los niños pasen de ser simples ‘usuarios’ a ser también ‘creadores’ en el medio digital, y esto básicamente por dos razones, por lo que puede aportar este tipo de aprendizajes en el desarrollo de otras capacidades cognitivas como son el razonamiento lógico y la creatividad por ejemplo y también por la demanda creciente de profesionales en el sector de las tecnologías de la información y la comunicación.

Cabría añadir el indudable aspecto lúdico de este tipo de aprendizajes así como el factor de proximidad en relación al medio social y a los conocimientos previos de los niños.

En las teorías constructivistas del aprendizaje, ampliamente aceptadas en nuestro medio educativo actual como soporte teórico válido de los procesos de enseñanza-aprendizaje, hay elementos clave que tenemos que admitir que no se han llevado a un grado óptimo de desarrollo todavía. Podemos citar por ejemplo la importancia de la conexión entre la práctica docente y el medio social en el que se desenvuelven los niños, la toma en consideración de sus intereses cotidianos, la importancia de valorar y aprovechar en la escuela los conocimientos que los propios niños aportan, la concepción de que todo aprendizaje significativo se sustenta en la participación activa del propio sujeto, etc.

En esa línea se podrían añadir otras aportaciones más recientes a las teorías de la educación, con un amplio grado de aceptación también entre la comunidad científica educativa y que están cuestionando la propia función de la escuela. En un mundo que cambia a una velocidad creciente se está empezando a considerar del todo necesario que la escuela aporte a los niños elementos que fomenten su creatividad (entendida como la capacidad de dar respuestas originales y eficientes a los problemas a los que se enfrentan), su autoconfianza y en definitiva su capacidad de aprender a aprender.

Finalmente me gustaría destacar lo que considero uno de los principales retos desde el punto de vista educativo de todo este movimiento, que no para de extenderse como tampoco para de extenderse el grado de tecnificación de nuestra sociedad: No debería perderse de vista en ningún momento que la tecnología siempre ha de estar al servicio de la educación, entendida ésta como el ideal universal que persigue que todas las personas alcancen el mayor grado posible de desarrollo y de bienestar afectivo, cognitivo y social en el marco de una sociedad que brinde a todos las mismas oportunidades.

## 1.2. Objetivos e hipótesis

### 1.2.1. Objetivo general

Realizar una propuesta de intervención metodológica en torno a la enseñanza de la programación en educación primaria con la finalidad de contribuir a la divulgación de los actuales movimientos, sean gubernamentales o no, que promueven la incorporación de este tipo de enseñanza en la escuela, mejorando así la participación activa y crítica de la comunidad educativa en estos procesos.

### 1.2.2. Objetivos específicos

- Conocer el posicionamiento de la Unión Europea, como nuestro marco de referencia cultural y legislativo, en relación a la incorporación de la programación al currículum escolar de primaria.
- Conocer las iniciativas más avanzadas que se están desarrollando en torno a esta cuestión en la actualidad, a nivel europeo y en nuestro país, tanto desde instancias públicas como privadas.
- Contribuir a mejorar la capacidad crítica de las escuelas ante las diversas propuestas en este sentido.
- Contribuir a divulgar el software actualmente más extendido para la enseñanza de la programación en la educación primaria.

### 1.2.3. Hipótesis de partida

Aprender a programar –aprender a codificar-, puede aportar a los niños la competencia en el medio digital que necesitan para llegar a ser sujetos de pleno derecho en ese medio que es ya parte fundamental de la realidad que les envuelve.

El profesor John Naughton, autor de lo que se considera un auténtico ‘manifiesto’ por la educación en ciencias de la computación en el siglo XXI, se dirige al secretario de estado de educación del Reino Unido en estos términos “*Creemos que todos los niños deberían tener la oportunidad de aprender ciencias de la computación, empezando en la escuela (...) Enseñamos física básica a cada niño, no con el objetivo principal de educar físicos si no porque todos ellos viven en un mundo gobernado por sistemas físicos. De la misma manera, todos los niños deberían aprender un poco de informática desde temprana edad porque van a vivir en un mundo en el que la computación está en todas partes*” (The Guardian , 2012).

## 1.3. Metodología del trabajo

- Para la elaboración del marco teórico:
  - Identificar el origen, la filosofía y las bases teóricas en las que se apoyan las propuestas de incorporar el aprendizaje de la programación al currículum escolar.
  - Conocer las propuestas más destacadas en actualidad de nuestro entorno, tanto por parte de las autoridades educativas como lanzadas desde instancias no gubernamentales.
  - Recopilar la normativa legal vigente al respecto en nuestro país y en la Unión Europea.
- Para la elaboración de la propuesta de intervención educativa:
  - Identificar el grado de conocimiento e interés por parte del equipo docente.
  - Conocer la capacidad tecnológica de la escuela en cuanto a equipos informáticos, software y mantenimiento de ambos elementos.
  - Seleccionar de entre las distintas propuestas aquella que más se adecúe a los intereses del centro y adaptarla sus necesidades y posibilidades concretas.

Partiendo de las fases propuestas por Rodríguez y otros (1996), seguidamente presentamos la actividades realizadas y la temporalización que hemos seguido:

<b>FASES DEL PROCESO INVESTIGADOR Y TEMPORALIZACIÓN</b>		
<b>Fase preparatoria</b>	Elección de la temática y presentación de la propuesta de TFG. Planificación del trabajo. Exposición de objetivos e hipótesis. Planteamiento de apartados del marco teórico.	30 de marzo 2015  8 de abril 2015
<b>Fase exploratoria e investigadora</b>	Localización de fuentes bibliográficas, revisión y selección de las mismas.	31 marzo-6 mayo 2015
<b>Fase analítica</b>	Análisis de la información obtenida durante la fase anterior. Elaboración del marco teórico.	8 abril- 6 mayo 2015
<b>Fase conclusiva</b>	Elaboración de propuesta partiendo del análisis previo. Elaboración de conclusiones, limitaciones y prospectiva. Presentación y defensa del TFG.	20 mayo -3 junio 3 – 17 junio Septiembre 2015

TABLA 1. FASES DEL PROCESO INVESTIGADOR Y TEMPORALIZACIÓN (ADAPTADA DE RODRÍGUEZ Y OTROS, 1996)

## 2. MARCO TEÓRICO EN TORNO AL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA PROGRAMACIÓN EN EDUCACIÓN PRIMARIA

### 2.1. Marco conceptual y terminológico en torno a la programación en educación primaria

#### 2.1.1. Definiciones y traducción de términos

La RAE (2012a) define ‘*Informática*’ como el ‘*Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores*’. En el medio angloparlante Informática se traduce por ‘*Computer Science*’.

En cuanto al término ‘*Programa*’, en su acepción Informática, se define como . ‘*Conjunto unitario de instrucciones que permite a un ordenador realizar funciones diversas, como el tratamiento de textos, el diseño de gráficos, la resolución de problemas matemáticos, el manejo de bancos de datos, etc.*’ (RAE 2012b).

En inglés el término programa se suele traducir por ‘*Code*’ (asimilando los términos programar y codificar) y la programación o elaboración de programas mediante ordenadores se traduce habitualmente por ‘*Coding*’.

### 2.1.2. Desarrollo de la Informática

La ciencia de la computación o informática, que emergió en torno a la segunda mitad del siglo XX, no ha parado de desarrollarse hasta ocupar un lugar insustituible en nuestra vida cotidiana como medio gestión, producción y consumo de todo tipo de información.

El formato en que se presenta la información manejada ha experimentado una notable transformación. Desde la gestión inicial de simples bloques de textos y de cifras hasta el procesamiento de toda clase de elementos audiovisuales característicos de nuestro tiempo y que envuelven nuestra realidad cotidiana.

Además de la información que se maneja, también han evolucionado enormemente los dispositivos físicos que soportan estos procesos (hardware), de manera que junto a los tradicionales ordenadores -computadores en el medio angloparlante-, conviven en la actualidad muy diversos dispositivos que procesan la información de manera digital como tablets o teléfonos móviles.

El software o conjunto de programas que se ejecutan en estos dispositivos ha seguido también una evolución constante diversificándose enormemente hasta formar un espeso entramado de código que se superpone uno sobre otro, desde el que gestiona más directamente los recursos del hardware (sistema operativo) hasta todos aquellos ejecutables que procesan y proveen la información que maneja el usuario. Los lenguajes de programación, especialmente aquellos destinados a crear estos programas más cercanos al consumidor, o incluso aptos para que los utilicen usuarios avanzados, han evolucionado también de manera que casi todos ellos presentan actualmente una interface gráfica, es decir un entorno donde se dispone de una serie de objetos elementales visualmente identificables y de unas determinadas reglas para combinarlos formando secuencias de instrucciones o código.

Por último señalar también la importancia de los avances en las conexiones entre ordenadores como fue el caso de la aparición en los 70 de las llamadas redes locales como Ethernet<sup>1</sup> que por primera vez permitieron conectar ordenadores ubicados en una misma oficina o edificio permitiendo compartir recursos como los dispositivos de almacenamiento de memoria y las impresoras. Y por supuesto la trascendencia de la posterior aparición de Internet que permitió la conexión entre ordenadores alejados geográficamente.

---

<sup>1</sup> Ceruzzi, P. E. (2008). 'Historia de la informática'. Open Mind. Recuperado de [https://www.bbvaopenmind.com/wp-content/uploads/static/pdf/109-127\\_PAUL\\_E.\\_CERUZZI\\_ESP\\_R.pdf](https://www.bbvaopenmind.com/wp-content/uploads/static/pdf/109-127_PAUL_E._CERUZZI_ESP_R.pdf)  
APRENDER A PROGRAMAR EN LA ESCUELA PARA LA MEJORA COMPETENCIAL EN EDUCACIÓN PRIMARIA *Estudio teórico y pautas de formación docente*

### 2.1.3. Internet y la expansión de las Tecnologías de la Información y la Comunicación (TIC)

Una vez inventados los ordenadores hace ya más de medio siglo y tras el enorme avance que representó la aparición de los ordenadores personales en los primeros años 80<sup>2</sup>, la siguiente revolución llegó con la invención de Internet en los 90. Internet es un sistema de comunicación en red que puede conectar de forma descentralizada infinidad de redes gracias a que todas comparten un mismo protocolo de comunicación para entenderse. El profesor Castells (2002) pone el acento en el aspecto cultural más que en el tecnológico de este invento en el sentido de que a diferencia de los medios de comunicación anteriores, fácilmente manipulables por el poder, Internet es creado con el propósito de ser un instrumento de comunicación horizontal, global y libre.

Aunque paradójicamente su embrión fueron unas investigaciones financiadas por el Departamento de Defensa de EUA hacia finales de los 60, su formato actual es un logro principalmente de proyectos de colaboración entre varias universidades norteamericanas hacia finales de los 80. Los recursos más populares de internet son la 'web' (World Wide Web) que permite la consulta remota de archivos, el correo electrónico, las conversaciones en línea o la telefonía móvil.

En resumen, este invento genial ha permitido por primera vez en la historia el *milagro* de que todas las personas del mundo, estén donde estén, puedan intercambiar y compartir todo tipo de información a través de ordenadores u otros dispositivos electrónicos.

A su vez, este avance tecnológico en el campo de las comunicaciones ha permitido la creación y difusión de una creciente cantidad de recursos educativos al alcance de todos. Precisamente esa velocidad vertiginosa a la que crecen los contenidos disponibles en la red hace cada vez más necesaria la reflexión permanente acerca de las pautas adecuadas para seleccionar y filtrar de forma eficiente y de acuerdo con nuestros objetivos pedagógicos todo lo que podemos encontrar en este universo en expansión.

El acrónimo TIC (Tecnologías de la Información y la Comunicación) es un término genérico usado hasta ahora para denominar a los recursos disponibles en la red en esos ámbitos. Más recientemente, en el medio educativo se está utilizando más el término TAC (Tecnologías del Aprendizaje y la Comunicación) para referirse de modo más específico a los recursos propiamente educativos.

### 2.1.4. Usuarios de los medios digitales: De consumidores a creadores

---

<sup>2</sup> El PC de IBM apareció en 1981 y el Macintosh 'Mac' de Apple en 1984.

Como podemos observar hoy en día casi todos somos usuarios de medios digitales, tanto adultos como niños somos por tanto *consumidores* de código (programas). En el caso de los niños hay que destacar que cada vez más este medio les ha acompañado desde sus primeras etapas de desarrollo. Pensemos en el uso casi masivo de videoconsolas y juegos dirigidos a todas las edades empezando por las más tempranas. O por ejemplo pensemos en el amplio acceso que tienen los más pequeños a poder hacer, visualizar, compartir o manipular fotos y vídeos con dispositivos como teléfonos móviles o ipads que, aunque no sean suyos propiamente sino de los jóvenes o adultos con los que conviven, sí suelen estar a su alcance para este tipo de actividades.

Ahora pensemos por un momento en la fuente de creación de todos esos programas que utilizamos habitualmente, y cada vez más, para realizar todo tipo de tareas como gestiones bancarias o comerciales por internet, producir y compartir de forma inmediata toda clase de mensajes de texto, imágenes o audiovisuales, todos los programas usados en las empresas, en el mundo educativo o en el mundo de ocio. ¿Quién crea todo este software?

Históricamente los programadores profesionales, expertos en informática y especializados en uno o varios lenguajes de programación, han sido los creadores exclusivos de código. Pero según Comisión Europea (2014) la explosión que está representando el incremento continuo y exponencial de software, especialmente el destinado a los ámbitos educativo y de ocio en general (del que toda la población es susceptible de ser usuaria) ha hecho que la demanda de profesionales programadores se haya disparado y las previsiones a medio y largo plazo sean de mayor crecimiento. Y esto es así aún teniendo en cuenta la enorme producción de código libre creado por entusiastas de la programación para ser distribuido gratuitamente.

Hablando en términos globales, hay en estos momentos una gran falta de programadores en el mercado laboral. Este es sin duda un factor clave –que no se oculta, sino todo lo contrario- en el actual interés de gobiernos y empresas en promocionar la introducción de niños y adolescentes en el aprendizaje de la programación de computadores, incluyendo propuestas concretas para incluir la enseñanza-aprendizaje de la programación en el currículum escolar como el caso de Código 21 (2015a).

Pero además de la demanda directa de profesionales de la programación, no cabe duda de que esta inmersión masiva en el medio digital nos demanda a todos ciertos aprendizajes, cierta ‘alfabetización digital’, que vamos incorporando a medida que vamos utilizando nuevos productos. Por esta razón, si consideramos imprescindible preparar a nuestros niños para desenvolverse y ser

parte activa de la sociedad, la educación no puede dejar de abordar su formación como usuarios y potenciales creadores en el medio digital<sup>3</sup>.

## 2.2. Análisis de la enseñanza de la programación en la normativa legal de educación primaria

### 2.2.1. Posicionamiento de la Unión Europea al respecto

El 25 de Julio de 2014, la vice-presidenta de la Comisión Europea de la Agenda Digital, la neerlandesa Neelie Kroes, en coordinación con la Comisión Europea de Educación, envió a todos los ministros de educación de los países miembros de la UE una carta<sup>4</sup> que empezaba haciendo referencia al grave asunto del desempleo juvenil como uno de los mayores retos de Europa, por otra parte se hacía eco de la manifiesta falta de profesionales en el mercado de las nuevas tecnologías de la información y la comunicación llegando incluso a cifrar en 900.000 el déficit de profesionales necesarios para cubrir este mercado en 2020. Tras este preámbulo concluía *“Promover las habilidades en programación en Europa es parte de la solución.”*

Y más adelante añadía *“Para enfrentarse a este reto nos gustaría garantizar que nuestros niños están mejor equipados para trabajar y vivir en la era digital. Hace menos de un año nosotros, como comisionados de Educación y de la Agenda Digital, hemos lanzado de forma conjunta la iniciativa ‘Opening up Education’, donde, entre otras cosas, abogamos por incrementar la capacitación en habilidades digitales para todos.”* *“..Entre estas habilidades digitales cada niño debería tener la oportunidad de desarrollar destrezas básicas de programación en orden a llegar a ser un ciudadano digital informado y con recursos propios.”*

Finalmente animaba a los interlocutores a desarrollar localmente de forma subsidiaria las iniciativas recogidas en la plataforma ‘Opening up Education’ dando soporte a la formación digital de profesores, niños y familias, proporcionando a los estudiantes la oportunidad de aprender a programar en la escuela.

---

<sup>3</sup> Inevery Crea (2014c). No podemos dejar solo al docente en el aprendizaje de la programación. Entrevista con Antolín García de la asociación *Con Más Futuro*. Recuperado de <http://ineverycrea.net/comunidad/ineverycrea/recurso/no-podemos-dejar-solo-al-docente-en-el-aprendizaje/3e93253e-e2e4-4196-9afc-120f05740ba4>

<sup>4</sup> Comisión Europea (2014). Carta de las responsables de las Comisiones de Agenda Digital y de Educación a los responsables de educación de los estados miembros. Recuperado de [http://ec.europa.eu/information\\_society/newsroom/image/kroes\\_vassiliou\\_letter\\_6597.pdf](http://ec.europa.eu/information_society/newsroom/image/kroes_vassiliou_letter_6597.pdf)  
APRENDER A PROGRAMAR EN LA ESCUELA PARA LA MEJORA COMPETENCIAL EN EDUCACIÓN PRIMARIA  
*Estudio teórico y pautas de formación docente*

En este sentido se emplazaba a los gobiernos destinatarios a una futura rendición de cuentas respecto al grado de realización de este tipo de iniciativas en el marco de los ‘ET 2020 Working Groups’ (Grupos de Trabajo Educación y Formación 2020).

Más adelante, en un apartado específico sobre la Agenda Digital Europea, se retoman estas propuestas.

### *2.2.1.1. Competencias Clave para el aprendizaje en la Unión Europea*

En la Recomendación 2006/962/EC del Parlamento Europeo y del Consejo<sup>5</sup> se considera que los ciudadanos deben estar preparados para adaptarse de una forma flexible a un mundo globalizado y que cambia a gran velocidad. En este sentido se insta a los estados miembros a que el sistema educativo garantice que los jóvenes adquieren todas aquellas competencias que se consideren clave para el desarrollo de su vida de adultos.

El texto define las competencias clave como *una combinación de conocimientos, capacidades y actitudes adecuadas al contexto. Las competencias clave son aquéllas que todas las personas precisan para su realización y desarrollo personales, así como para la ciudadanía activa, la inclusión social y el empleo.*

Competencias Clave enunciadas en la Recomendación 2006/962/EC del Parlamento Europeo y del Consejo:

1. Comunicación en la lengua materna
2. Comunicación en lenguas extranjeras
3. Competencia matemática y competencias básicas en ciencia y tecnología
4. Competencia digital
5. Aprender a aprender
6. Competencias sociales y cívicas
7. Sentido de la iniciativa y espíritu de empresa
8. Conciencia y expresión culturales

Adicionalmente al carácter integrador e interdisciplinar que caracteriza el ejercicio de la programación de ordenadores, podemos destacar como más genuinamente ligadas al aprendizaje de la programación las competencias digital, de aprender a aprender y de sentido de la iniciativa y espíritu de empresa.

---

<sup>5</sup> Recomendación 2006/962/EC del Parlamento Europeo y del Consejo. Recuperado de [http://www.mcu.es/cine/docs/Novidades/Recomendacion\\_Parlamento\\_Europeo\\_Consejo\\_Aprendizaje\\_permanente.pdf](http://www.mcu.es/cine/docs/Novidades/Recomendacion_Parlamento_Europeo_Consejo_Aprendizaje_permanente.pdf)  
APRENDER A PROGRAMAR EN LA ESCUELA PARA LA MEJORA COMPETENCIAL EN EDUCACIÓN PRIMARIA 9  
*Estudio teórico y pautas de formación docente*

## 2.2.2. Legislación educativa vigente en España para la educación Primaria

La legislación vigente que se regula la etapa de la educación primaria en España es la siguiente:

- Ley Orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa. (LOMCE)
- Real Decreto 126/2014, de 28 de febrero, por el que se establece el currículo básico de la Educación Primaria. (desarrolla la LOMCE)

En el texto de estas disposiciones legales no aparece ninguna referencia a la enseñanza o aprendizaje de la programación en la etapa de Educación Primaria.

En cambio, aunque no aplique en el marco del presente trabajo, sí aparece una referencia a la enseñanza de la programación en el currículo de la Educación Secundaria<sup>6</sup>, concretamente entre los contenidos del 4º Curso de la asignatura de Tecnología, Bloque 4 Control y Robótica, se incluye el *‘Uso del ordenador como elemento de programación y control’*.

### 2.2.2.1. Legislación autonómica en esta materia

Si bien la legislación estatal española vigente que establece el currículum de educación primaria (Real Decreto 126/2014) no hace ninguna referencia al aprendizaje de la programación, nos encontramos con dos comunidades autónomas que sí han legislado en esta materia, se trata de la Comunidad de Madrid y la de Navarra, estos son los respectivos decretos autonómicos:

- Decreto 89/2014, de 24 de julio, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el Currículo de la Educación Primaria.
- Decreto Foral 60/2014, de 16 de julio, por el que se establece el currículo de las enseñanzas de Educación primaria en la Comunidad Foral de Navarra.

### 2.2.2.2. Objetivos y Competencias Básicas en la legislación española

Los Objetivos de la Educación Primaria están detallados en el artículo 7 del Real Decreto 126/2014. Por su vinculación expresa con la temática tratada podemos destacar estos tres:

\* Desarrollar hábitos de trabajo individual y de equipo, de esfuerzo y de responsabilidad en el estudio, así como actitudes de confianza en sí mismo, sentido crítico, iniciativa personal, curiosidad, interés y creatividad en el aprendizaje, y espíritu emprendedor.

\* Iniciarse en la utilización, para el aprendizaje, de las Tecnologías de la Información y la Comunicación desarrollando un espíritu crítico ante los mensajes que reciben y elaboran.

---

<sup>6</sup> Real Decreto 1631/2006, de 29 de diciembre, por el que se establecen las enseñanzas mínimas correspondientes a la Educación Secundaria Obligatoria (desarrolla la LOE)

\* Utilizar diferentes representaciones y expresiones artísticas e iniciarse en la construcción de propuestas visuales y audiovisuales.

En cuanto a las llamadas Competencias Básicas o Competencias Clave, el Real Decreto 126/2014 que desarrolla la LOMCE manifiesta su alineamiento con la Recomendación 2006/962/EC del Parlamento Europeo y del Consejo sobre este punto.

En relación a las competencias básicas en la legislación española, encontramos que éstas se definen como *‘las capacidades para aplicar de forma integrada los contenidos propios de cada enseñanza y etapa educativa, con el fin de lograr la realización adecuada de actividades y la resolución eficaz de problemas complejos’*. Real Decreto 126/2014. Como también se indica en este texto legal la enseñanza orientada a las competencias es por definición una enseñanza transversal e integradora donde los aprendizajes deben abordarse desde las distintas áreas del conocimiento y tanto desde los ámbitos formales como informales del medio educativo.

Competencias Básicas enunciadas en el artículo 2 del Real Decreto 126/2014 que desarrolla la LOMCE:

1. Comunicación lingüística
2. Competencia matemática y competencias básicas en ciencia y tecnología
3. Competencia digital
4. Aprender a aprender
5. Competencias sociales y cívicas
6. Sentido de iniciativa y espíritu emprendedor.
7. Conciencia y expresiones culturales

Observamos que la legislación educativa vigente en España recoge también las competencias clave más directamente relacionadas con el aprendizaje de la programación, vistas en el apartado 2.2.1.1.

## **2.3. Beneficios del aprendizaje de la programación en la etapa de educación primaria**

### **2.3.1. Motivación: Impulso creativo**

El simple hecho de utilizar un ordenador en lugar del tradicional libro de texto suele resultar motivador para los pequeños. Por su experiencia personal tienden a asociar este medio con actividades lúdicas y en general responden positivamente a una propuesta de cierta acción frente a la habitual escucha pasiva. Pero si el cambio se queda sólo en el medio y el mensaje sigue siendo una sucesión de contenidos presentados de una manera más o menos elaborada su atención también

acabará por decaer un poco. Esto es lógico porque en realidad el cambio con respecto al método tradicional es mínimo, sólo hay que preguntar a cualquier niño de primaria si quiere juegos de ordenador ‘educativos’ para su cumpleaños.

La diferencia entre aprender mediante un ordenador (utilizar programas didácticos) y aprender a crear programas, crear videojuegos, hacer composiciones artísticas o encontrar la secuencia de instrucciones precisa para dibujar una figura geométrica, por ejemplo, son cosas bien distintas. Programar es un ejercicio eminentemente creativo y la motivación intrínseca para las actividades creativas a priori no tiene límites.

### 2.3.2. Perspectiva constructivista del aprendizaje: Aprender haciendo

Desde las aportaciones de Piaget<sup>7</sup> sobre el desarrollo cognitivo humano sabemos que el aprendizaje es un proceso activo de construcción del conocimiento y que la mejor manera de potenciarlo es a través de la experimentación con la realidad. Es la acción la que conforma el pensamiento y no al revés según Piaget, en un proceso constante por el que nuestros conocimientos se van reorganizando y ampliando a medida que asimilamos nuevos conocimientos.

Pues bien, junto a la creación ejercida directamente en el mundo físico, que es insustituible para el aprendizaje, la creación de software puede aportar ese mismo potencial en el medio digital, cuyo dominio sabemos que es también esencial para desenvolvernó en nuestro mundo actual.

Por tanto, podemos decir que la propuesta de enseñar a programar en la escuela se enmarca en la línea de las teorías constructivistas del aprendizaje donde el alumno es protagonista activo del proceso y todo nuevo conocimiento se construye sobre el anterior que así se reelabora y amplía.

### 2.3.3. Teoría de las inteligencias múltiples: Aprendizaje transversal

Gracias a la Teoría de las inteligencias múltiples enunciada por Gardner<sup>8</sup> sabemos que la inteligencia humana no es una capacidad monolítica ni unívoca, ni siquiera es constante a lo largo del tiempo. Al contrario, se presenta en forma de talentos o capacidades distintas y con una gran diversidad entre las personas. Por tanto, si tenemos diversas inteligencias no podemos enseñar a todos los niños de la misma manera, sino que se necesita una educación que se adapte al individuo, una educación personalizada. En una entrevista para el programa Redes de TVE con motivo de su

---

<sup>7</sup> Monografías (2015). Teoría cognitiva de Piaget. Recuperado de <http://www.monografias.com/trabajos76/teoria-cognitiva-piaget/teoria-cognitiva-piaget2.shtml#lateoradeb>

<sup>8</sup> Gardner, H. (1998), *Inteligencias múltiples. La teoría en la práctica*, Barcelona, Paidós.

premio Príncipe de Asturias de Ciencias sociales 2011, Gardner (2012) afirma que la revolución digital está brindando en la actualidad la posibilidad de materializar su propuesta de educación personalizada enunciada hace más de veinte años. Destaca también la facilidad que brinda el medio digital para el trabajo colaborativo al mismo tiempo que respeta los distintos ritmos o estilos de aprendizaje.

Partiendo de que no hay aprendizaje sin motivación Bahón (2014) insiste en que la presentación de los contenidos en la escuela debe ser semejante a la forma en que estos se presentan (y requieren respuesta) en la vida real. Dado que en la vida real los conocimientos propios de distintas disciplinas se presentan mezclados apuesta por el aprendizaje transversal, es decir, por el aprendizaje por proyectos.

Pues bien, se puede afirmar que aprender a programar conduce a un aprendizaje por proyectos. Crear un programa es crear un pequeño proyecto en el sentido de que contando con unos objetos que emulan elementos de la realidad (o de la fantasía) el niño desarrolla diversos razonamientos y destrezas para actuar sobre ellos y conseguir que hagan cosas.

#### 2.3.4. Pequeños creadores sin miedo al error

Según afirma Robinson (2011) todos los niños son extraordinariamente creativos en sus primeros años pero lamentablemente la escuela tradicional suele acabar con ese talento innato en buena medida porque estigmatiza el error.

A su vez Bahón (2014) sostiene directamente que castigar el error va en contra del aprendizaje y que de ese modo ayudamos a que los niños vinculen el aprendizaje con sentimientos negativos cuando debería ser justamente lo contrario.

El aprovechamiento del error para avanzar en el proceso emprendido es un aspecto muy sustancial de la programación. El resultado de la secuencia de instrucciones (código) puede apreciarse de forma inmediata y si no es el resultado esperado se puede proceder a cambiar el código y volver a probarlo inmediatamente. Esta enorme facilidad para ejercer la aproximación y el ensayo-error en la solución a los problemas desestigmatiza el error que en cambio aparece como un elemento más de los que conforman el proceso creativo. Sin duda esto genera autoconfianza en los niños, en su capacidad de corregir sus propios errores y sobretodo de aprender de ellos.

#### 2.3.5. Apuesta por las competencias sobre los contenidos

Como vemos, estamos asistiendo en los últimos años a un cuestionamiento generalizado del modelo educativo tradicional. Uno de los aspectos más criticados es la todavía excesiva preeminencia de la función de la escuela como mera transmisora de conocimientos. Lo que se propone es que la enseñanza basada en la asimilación de contenidos (vinculados normalmente a una única área de conocimiento) ceda espacio en la escuela a la enseñanza basada en la realización de proyectos (de carácter esencialmente interdisciplinar) con el fin de que los alumnos alcancen lo que se consideran competencias básicas.

Bahón (2014) afirma que en muchos casos los currículos están desfasados y que la adquisición de competencias debería tener un papel mucho más relevante en la escuela. En definitiva, debemos apostar claramente por enseñar a los niños cómo aprender ante un futuro desconocido.

### 2.3.6. El pensamiento computacional

En el año 2006 la Dra. Jeannette Wing publicó en la revista digital *Communications of the ACM*<sup>9</sup> un artículo denominado ‘*Computational thinking*’<sup>10</sup> (‘*El pensamiento computacional*’) donde caracterizaba como pensamiento computacional a aquel que formula los problemas de manera que puedan ser resueltos mediante algoritmos lógicos y por tanto a través de un programa informático que aplique estos algoritmos. La idea es que este tipo de pensamiento contribuye de manera importante al aprendizaje de las matemáticas y las ciencias. En Programamos (2015b) se recogen las características del pensamiento computacional como son organizar y analizar la información de forma lógica, representar la información mediante modelos abstractos que faciliten las simulaciones, plantear soluciones en forma secuencia ordenada de pasos o algoritmos, identificar e implementar distintas posibles soluciones, etc.

En la misma línea el profesor Antonio Ángel Ruíz, experto en lenguaje Scratch (autor del blog educativo <http://jueduco.blogspot.com.es/>), afirma en Inevery Crea (2014a) que el aprendizaje de la programación impulsa el razonamiento lógico-matemático.

### 2.3.7. Fomento del trabajo colaborativo

---

<sup>9</sup> ACM Association for Computing Machinery, fundada en 1947 en USA, como la primera sociedad científica y educativa sobre la computación. Recuperado de [http://es.wikipedia.org/wiki/Association\\_for\\_Computing\\_Machinery](http://es.wikipedia.org/wiki/Association_for_Computing_Machinery)

<sup>10</sup> Wing, J. (2006). ‘Computational thinking’, *Communications of the ACM*. Recuperado de <https://www.cs.cmu.edu/~CompThink/papers/Wing06.pdf>

También merece la pena destacar –porque podría parecer paradójico- la incidencia directa del aprendizaje de la programación en la adquisición de las competencias sociales y cívicas ya que los entornos digitales que se utilizan actualmente para enseñar programación a los niños, como es el caso del lenguaje Scratch que veremos seguidamente, promueven el trabajo colaborativo facilitando enormemente que los alumnos compartan sus creaciones con los compañeros y también en la red. Ciertamente esta cualidad es generalizable al conjunto de los aprendizajes TIC cuando se realizan de forma online, pero adquiere especial interés en los programas creados por los niños ya que se son un tipo de actividad que se presta especialmente a compartir ideas y esfuerzos para conseguir objetivos.

### 2.3.8. Retos para el profesorado ante la enseñanza en el medio digital

En el contexto del aprendizaje personalizado que brindan las nuevas tecnologías, Gardner nos señala en TVE REDES (2012) que el papel tradicional del profesor en este entorno debe cambiar también. Los maestros deben asumir que ya no son la fuente de toda información y convertirse esencialmente en guías, en orientadores del proceso de aprendizaje de sus alumnos. De la aceptación de que existen distintos tipos de inteligencia se desprende que no todos aprendemos de la misma manera ni tampoco al mismo ritmo.

Este cambio radical del papel de los docentes no es tarea fácil, sin duda es un gran reto incluso para aquellos maestros más favorables a poner al servicio de la mejora de la enseñanza los recursos de las nuevas tecnologías. En Inevery Crea (2014b) un grupo de profesores de la Comunidad de Madrid comprometidos con el uso de las nuevas tecnologías en el aula expresan sus opiniones sobre las principales dificultades a las que se enfrentan en su tarea como son la falta de programas de formación y reciclaje para ellos y el insuficiente equipamiento tecnológico de los centros.

## 2.4. Software y aplicaciones para enseñar a programar en educación primaria

### 2.4.1. Los orígenes de la programación para niños: El lenguaje Logo

El lenguaje Logo fue el primer lenguaje de programación desarrollado específicamente para ser utilizado por niños. Fue creado por el matemático y experto en inteligencia artificial Seymour Papert a finales de los años 60. Papert trabajaba en ese tiempo para el MIT<sup>11</sup> pero pocos años antes

---

<sup>11</sup> Massachusetts Institute of Technology

había trabajado en Ginebra con Jean Piaget, psicólogo y padre del enfoque constructivista del desarrollo cognitivo.

Logo permite crear gráficos dando instrucciones precisas a una imaginaria tortuga que se desplaza en una superficie obedeciendo las órdenes que recibe. Las sentencias del lenguaje Logo, disponible en varios idiomas, son muy semejantes al lenguaje humano. El MIT ha continuado creando nuevas versiones de este lenguaje como es el caso de StarLogo TNG (2008) que dispone de un entorno gráfico en 3D.

## 2.4.2. El lenguaje de uso más extendido actualmente en Educación Primaria: Scratch

### 2.4.2.1. Origen y principales características

El lenguaje de programación Scratch también fue desarrollado el MIT, esta vez en colaboración con la empresa LEGO en el año 2007. Concretamente Scratch es una creación del Lifelong Kindergarten Group del MIT Media Lab. Se trata de un software abierto, por tanto accesible para todos, con el que puede trabajarse de forma online<sup>12</sup> o bien descargarse de la web del producto<sup>13</sup>. La aplicación puede instalarse sobre los sistemas operativos Mac, Windows y Linux.

Scratch tiene una interface totalmente gráfica donde las distintas sentencias del lenguaje (instrucciones, variables, contadores, etc.) son piezas móviles y encajables, denominadas ‘Bloques’, que al unirse conforman la secuencia de instrucciones del programa. Esta es una diferencia sustancial con el lenguaje Logo donde los niños debían conocer y escribir las instrucciones del programa para conseguir el movimiento deseado en la ‘tortuga’.

Otras características relevantes son la enorme facilidad de la plataforma online para compartir en los proyectos realizados con Scratch y la funcionalidad que permite incorporar a los programas imágenes de objetos creados en otras aplicaciones. Por otra parte es una aplicación totalmente multilingüe, que permite trabajar en diversas lenguas y cambiar de una a otra con suma facilidad.

Para aprovechar toda la funcionalidad de la plataforma de Scratch online hay que registrarse como usuario, que es un trámite sencillo y rápido. Todos los usuarios de la plataforma Scratch online disponen de un espacio exclusivo donde guardar sus trabajos (proyectos) y de la funcionalidad que permite compartirlas en la red. Desde cualquier ordenador conectado a internet podemos trabajar y compartir nuestros proyectos de forma inmediata con cualquier otro usuario de la red.

---

<sup>12</sup> Scratch (2015a). Plataforma online del lenguaje de programación Scratch. Recuperado de <https://scratch.mit.edu/>

<sup>13</sup> Scratch (2015b). Web para descarga lenguaje Scratch, versión 2. Recuperado de <https://scratch.mit.edu/scratch2download/>

### 2.4.2.2. Funcionalidad standard

En el área de trabajo de Scratch podemos identificar inmediatamente estos 3 tipos de elementos: el escenario (inicialmente en blanco), los personajes (inicialmente el gato característico) y las instrucciones del lenguaje (bloques apilables de distintas categorías).

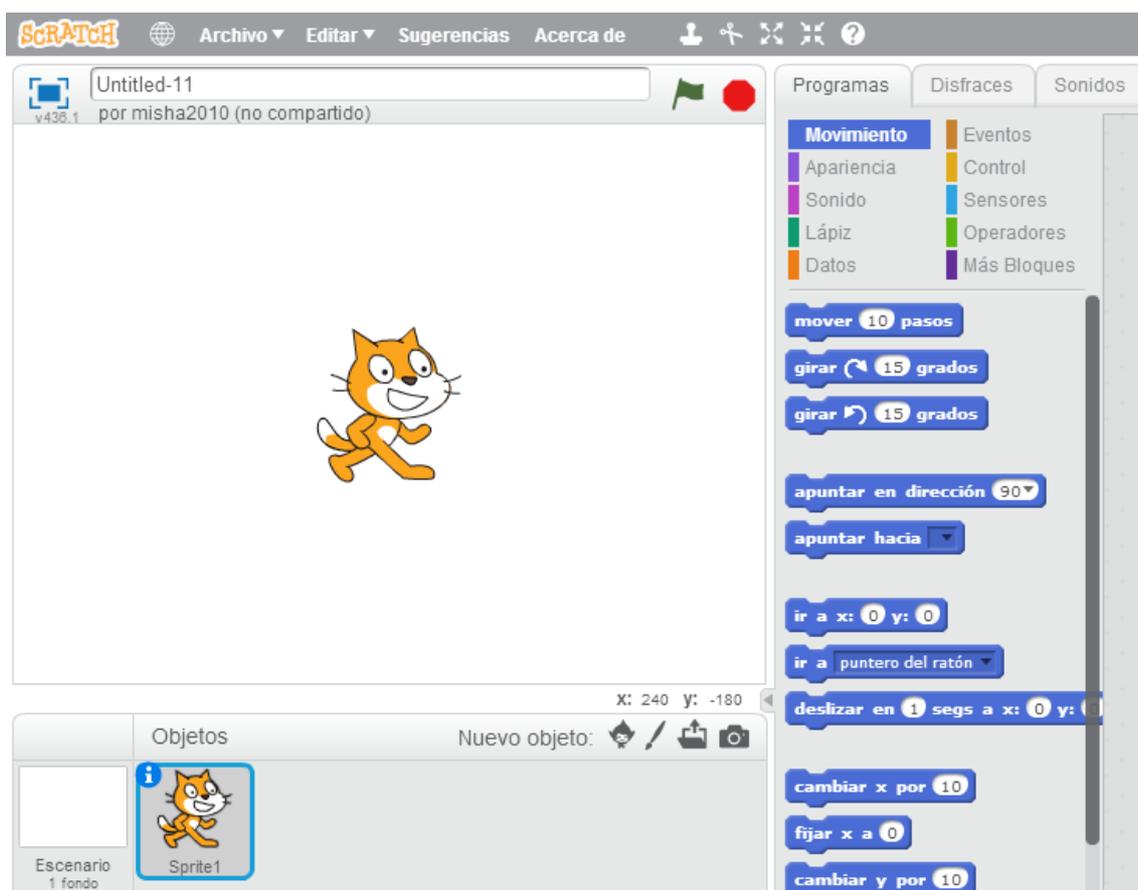


IMAGEN 1. CAPTURA DE LA PANTALLA INICIAL DEL PROGRAMA SCRATCH (ELABORACIÓN PROPIA)

Como hemos dicho las instrucciones o sentencias del lenguaje Scratch son también pequeños objetos gráficos con forma de piezas apilables que contienen el texto de la acción que realizan. Estos objetos apilables, convenientemente combinados y en la secuencia adecuada, constituyen el código del programa de manera que determinan las acciones a realizar por los personajes en el escenario activo.

Los distintos tipos de instrucciones (de movimiento, de apariencia, de sonido, de operaciones, de control de la secuencia, etc.) tienen diferentes colores para facilitar su identificación.

Tipos de instrucciones de Scratch de uso más frecuente			
<p>Programas   Disfraces   Sonid</p> <p><b>Movimiento</b></p> <ul style="list-style-type: none"> <li>mover 10 pasos</li> <li>girar 15 grados</li> <li>girar 15 grados</li> <li>apuntar en dirección 90</li> <li>apuntar hacia</li> <li>ir a x: 0 y: 0</li> <li>ir a puntero del ratón</li> <li>deslizar en 1 segs a x: 0 y: 0</li> <li>cambiar x por 10</li> <li>fijar x a 0</li> <li>cambiar y por 10</li> </ul>	<p>Programas   Disfraces   Sonid</p> <p><b>Apariencia</b></p> <ul style="list-style-type: none"> <li>decir Hello! por 2 segundos</li> <li>decir Hello!</li> <li>pensar Hmm... por 2 segundos</li> <li>pensar Hmm...</li> <li>mostrar</li> <li>esconder</li> <li>cambiar disfraz a costume2</li> <li>siguiente disfraz</li> <li>cambiar fondo a backdrop1</li> <li>cambiar efecto color por 25</li> <li>establecer efecto color a 0</li> </ul>	<p>Programas   Disfraces   Sonid</p> <p><b>Sonido</b></p> <ul style="list-style-type: none"> <li>tocar sonido meow</li> <li>tocar sonido meow y esperar</li> <li>detener todos los sonidos</li> <li>tocar tambor 1 durante 0.25</li> <li>silencio por 0.25 pulsos</li> <li>tocar nota 60 durante 0.5 pul</li> <li>fijar instrumento a 1</li> <li>cambiar volumen por -10</li> <li>fijar volumen a 100 %</li> <li>volumen</li> <li>cambiar tempo por 20</li> </ul>	<p>Programas   Disfraces   Sonid</p> <p><b>Eventos</b></p> <ul style="list-style-type: none"> <li>al presionar</li> <li>al presionar tecla espacio</li> <li>al hacer clic en este objeto</li> <li>cuando el fondo cambie a backdr</li> <li>cuando volumen del sonido sea &gt;</li> <li>al recibir message1</li> <li>enviar message1</li> <li>enviar message1 y esperar</li> </ul>

TABLA 2. INSTRUCCIONES DE USO MÁS FRECUENTE EN EL LENGUAJE SCRATCH (ELABORACIÓN PROPIA)

Al crear un programa con Scratch se van arrastrando y soltando las sucesivas instrucciones (bloques), que van encajando como piezas de LEGO, lo que le convierte en un lenguaje idóneo para trabajar con niños.

En la imagen 2 podemos ver un ejemplo de programa muy sencillo que permite mover un helicóptero en las cuatro direcciones cardinales utilizando las flechas del teclado.



IMAGEN 2. PROGRAMA SCRATCH MUY SENCILLO PARA MOVER UN HELICÓPTERO CON LAS FLECHAS DEL TECLADO (ELABORACIÓN PROPIA)

El entorno de programación de Scratch dispone de una amplia colección de imágenes de objetos/personajes y de escenarios, pero una de las mayores potencialidades de este lenguaje es que podemos incorporar imágenes creadas por nosotros.

#### 2.4.2.3. Funcionalidad ampliada hacia la robótica educativa

Existen ampliaciones de la funcionalidad estándar de Scratch que le permiten actuar sobre pequeños dispositivos móviles dotados de sensores (pequeños *robots*).

Podemos destacar dos iniciativas en este sentido, una desarrollada en colaboración con la empresa LEGO (robots de Lego programables) y otra iniciativa en colaboración con el Proyecto Arduino (hardware sin patente y de bajo coste). La empresa LEGO tiene en el mercado dos packs para crear robots programables, uno para primaria (LEGO Wedo) y otro más adecuado para secundaria (LEGO Mindstorm), que aunque son programables con un lenguaje propio de LEGO también lo son mediante el lenguaje Scratch.

## 2.4.3. Otros lenguajes de programación adecuados para la etapa de primaria

En la siguiente tabla se relacionan los lenguajes de programación más utilizados actualmente con fines educativos. Aunque suelen ser más idóneos para unas etapas educativas que para otras, la mayoría de ellos se pueden utilizar también en primaria.

Lenguaje	Página Web	Etapas preferente	Software libre
Scratch (MIT)	<a href="http://scratch.mit.edu">http://scratch.mit.edu</a>	Primaria/ Secundaria	Sí
Scratch Junior (MIT)	Para tablets (Android o Ipad) <a href="http://www.scratchjr.org/">http://www.scratchjr.org/</a>	Infantil	Sí
Logo (Open Start Logo) (MIT)	Hay múltiples versiones de este lenguaje desde la original creada por el MIT en 1967. <a href="http://el.media.mit.edu/logo-foundation/">http://el.media.mit.edu/logo-foundation/</a>	Primaria	Sí
Ligth-Bot	Para resolver puzzles usando la lógica de programación <a href="http://lightbot.com/">http://lightbot.com/</a>	Primaria	No (menos de 5€)
Minecraft (Mojang)	En realidad es un videojuego pero del tipo 'mundo abierto' <a href="https://minecraft.net/">https://minecraft.net/</a>	Primaria	No (menos de 20€)
Small Basic (Microsoft)	<a href="https://www.microsoft.com/es-es/download/details.aspx?id=46392">https://www.microsoft.com/es-es/download/details.aspx?id=46392</a>	Primaria	Sí
Kodu (Microsoft)	Para crear videojuegos. <a href="http://www.kodugamelab.com/">http://www.kodugamelab.com/</a>	Primaria	Sí
Simple	<a href="http://www.simplecodeworks.com/website.html">http://www.simplecodeworks.com/website.html</a>	Primaria	Sí
Alice (Carnegie Mellon Univ.)	Basado en Java, diseñado para enseñar programación en entorno 3D. <a href="http://www.alice.org/index.php">http://www.alice.org/index.php</a>	Secundaria	Sí
Python (Python Software Fd.)	<a href="https://www.python.org/">https://www.python.org/</a>	Secundaria	Sí

APP Inventor (MIT & Google Labs)	Para crear APPs para móviles / tablets Android. <a href="http://appinventor.mit.edu/explore/">http://appinventor.mit.edu/explore/</a>	Secundaria	Sí
HTML5 & CSS	Para crear páginas web. <a href="http://www.w3.org/TR/html5/">http://www.w3.org/TR/html5/</a> y <a href="http://www.w3.org/Style/CSS/">http://www.w3.org/Style/CSS/</a>	Secundaria	Sí

TABLA 3. LENGUAJES DE PROGRAMACIÓN MÁS UTILIZADOS ACTUALMENTE CON FINES EDUCATIVOS (ELABORACIÓN PROPIA)

## 2.5. Algunas experiencias y proyectos destacables relacionados con la temática

Atendiendo al alcance del presente trabajo, las iniciativas que se muestran a continuación abarcan o se ocupan específicamente de la etapa de educación primaria.

### 2.5.1. Propuestas o experiencias de alcance europeo

A continuación se presentan las iniciativas de alcance europeo más relevantes en la promoción de la enseñanza de la programación a niños y jóvenes, separando por un lado las propuestas que tienen origen en la propia Comisión Europea (apartado 2.5.1.1.) y por otro lado las iniciativas y proyectos de origen privado (apartado 2.5.1.2).

#### 2.5.1.1. Iniciativas en el marco de la Comisión Europea

##### 2.5.1.1.1. Estrategia Europa 2020: El origen de la Agenda Digital Europea

En Marzo de 2010 la Comisión Europea lanzó una ambiciosa iniciativa que trataba de poner la vista más allá de la superación de la crisis económica en que se hallaba inmersa, se trata de la llamada Estrategia Europa 2020<sup>14</sup> cuyo objetivo principal era recuperar el empleo y elevar la calidad de vida de los ciudadanos europeos a lo largo de los próximos 10 años.

Sus objetivos concretos, agrupados en las áreas de empleo, investigación e innovación, cambio climático y energía, educación y lucha contra la pobreza, debían convertirse en objetivos nacionales evaluables (con un seguimiento semestral) para cada uno de los estados miembros.

<sup>14</sup> Estrategia Europa 2020 (2010a). Plataforma de comunicación de la Comisión Europea sobre el seguimiento de las propuestas de la Estrategia Europa 2020 en los distintos países de la Unión. Recuperado de [http://ec.europa.eu/europe2020/index\\_es.htm](http://ec.europa.eu/europe2020/index_es.htm)

En el terreno de la educación establece el siguiente objetivo, *‘El porcentaje de abandono escolar debería ser inferior al 10 % y al menos el 40 % de la generación más joven debería tener estudios superiores completos’*. Estrategia Europa 2020 (2010b).

En armonía con sus objetivos la Estrategia Europa 2020 lanzó una serie de iniciativas concretas para impulsar los avances en estos temas considerados prioritarios. En relación con la temática del presente trabajo destacamos la siguiente: *‘«Una agenda digital para Europa», con el fin de acelerar el despliegue de internet de alta velocidad y beneficiarse de un mercado único digital para las familias y empresas’*. Estrategia Europa 2020 (2010b).

#### *2.5.1.1.2. Propuestas de la Agenda Digital Europea en materia educativa*

La Agenda Digital para Europa<sup>15</sup> se propone alcanzar un mercado único digital basado en el acceso generalizado a internet con banda ancha a altas velocidades en 2020, así como incrementar en un 50% o más el número de hogares europeos conectados a internet.

Asimismo según Uría Menéndez (2010), la Agenda Digital se propone potenciar el uso de las TIC y de internet como soporte esencial a la actividad económica y social al constatar que en Europa hay una insuficiente cualificación profesional en las TIC y un déficit general en la alfabetización digital (llamada ‘brecha digital’) que excluye a muchos ciudadanos de la sociedad y de la economía digitales. Por esta razón la Agenda considera esencial educar a los ciudadanos europeos en este terreno.

En el año 2014 la Agenda Digital Europea (en coordinación con la Comisión Europea de Educación) anunció el lanzamiento de la *‘Opening Up Education Initiative’*<sup>16</sup> (Iniciativa de Educación Abierta) para la promoción del aprendizaje de las TIC en la escuela, especialmente mediante el uso de software educativo abierto (Open Education Resources OER).

En una carta<sup>17</sup> conjunta enviada en Julio de 2014 por representantes de la Comisión de la Agenda Digital y de la Comisión de Educación a todos los responsables de educación de los estados miembros, se hace un llamamiento a despertar su conciencia sobre la importancia de dotar de herramientas de programación a los niños europeos así como de formar a los profesores en cómo impartir estas enseñanzas.

---

<sup>15</sup> Agenda Digital Europea (2010). Página web de la Comisión Europea de la Agenda Digital. Recuperado de <http://ec.europa.eu/digital-agenda/en>

<sup>16</sup> Opening Up Education Initiative (2014). Iniciativa de la Agenda digital Europea para promover la innovación en el aprendizaje. Recuperado de <http://www.openeducationeuropa.eu/es/initiative>

<sup>17</sup> Comisión Europea (2014). Carta de las responsables de las Comisiones de Agenda Digital y de Educación a los responsables de educación de los estados miembros. Recuperado de [http://ec.europa.eu/information\\_society/newsroom/image/kroes\\_vassiliou\\_letter\\_6597.pdf](http://ec.europa.eu/information_society/newsroom/image/kroes_vassiliou_letter_6597.pdf)

Con este objetivo específico de promover la enseñanza de la programación en el medio escolar la Agenda Digital ha desarrollado la ‘*European Coding Initiative*’ (Iniciativa Europea de Programación) que en Marzo de 2015 ha lanzado una nueva web<sup>18</sup> para dar soporte a estudiantes, profesores y adultos en general interesados en aprender a programar.



IMAGEN 3. CAPTURA PANTALLA DE LA WEB ‘ALL YOU NEED IS CODE’ DE LA INICIATIVA EUROPEA DE PROGRAMACIÓN (ELABORACIÓN PROPIA)

En el apartado dedicado a los profesores, esta web ofrece diversos tutoriales sobre entornos de programación y entornos digitales en general.

Estos son algunos de los tutoriales disponibles actualmente para profesores de primaria:

- Ligth-bot, para resolver puzzles usando la lógica de la programación.
- Scratch Junior, versión simplificada del Scratch para usarse en tablets por los más pequeños.
- Small Basic, entorno de Microsoft para enseñar a programar a niños a partir de 8 años.
- Kodu, entorno de Microsoft para crear videojuegos en el ordenador.

Por otra parte, esta iniciativa europea para la programación trabaja en colaboración con la ‘*European Schoolnet Academy*’<sup>19</sup> (Academia Europea en la Red), organización especializada en ofrecer cursos gratuitos y masivos (Massive Open Online Curses MOOCs) para profesores de primaria y secundaria sobre esta temática entre otras.

<sup>18</sup> European Coding Initiative (2014). Iniciativa de la Agenda Digital Europea para la promoción del aprendizaje de la programación en el ámbito educativo. Recuperado de <http://www.allyouneediscode.eu/>

<sup>19</sup> European Schoolnet Academy (2014). Escuela online que ofrece formación en innovación a profesores de primaria y secundaria. Recuperado de <http://www.europeanschoolnetacademy.eu/>



IMAGEN 4. CAPTURA DE PANTALLA DE LA WEB DE LA EUROPEAN SCHOOLNET ACADEMY (ELABORACIÓN PROPIA)

Aunque la Schoolnet Academy está basada en Bruselas constituye una red que conecta los Ministerios de Educación de todos los estados miembros de la UE.

A modo de ejemplo podemos citar los cursos actualmente disponibles en al Schoolnet Academy:

- Creative use of Tablets in Schools
- How to teach computing for primary teachers
- Games in Schools 2nd Round

### *2.5.1.2. Iniciativas privadas de alcance europeo*

#### *2.5.1.2.1. Proyecto CodeClubWorld desde el Reino Unido*

El proyecto CodeClub<sup>20</sup> es una iniciativa privada surgida en el Reino Unido en 2012 con el propósito de brindar al mayor número posible de niños, especialmente de educación primaria, la oportunidad de aprender a programar a través de una iniciativa también innovadora en sí misma. La propuesta es que profesionales, estudiantes avanzados o cualquier persona aficionada al mundo de la programación de ordenadores dedique voluntariamente una hora semanal de su tiempo a enseñar a niños a programar en el marco de una actividad extraescolar.

<sup>20</sup> CodeClub (2012). Web central del Proyecto CodeClub. Recuperado de <http://codeclubworld.org/>  
 APRENDER A PROGRAMAR EN LA ESCUELA PARA LA MEJORA COMPETENCIAL EN EDUCACIÓN PRIMARIA  
*Estudio teórico y pautas de formación docente*

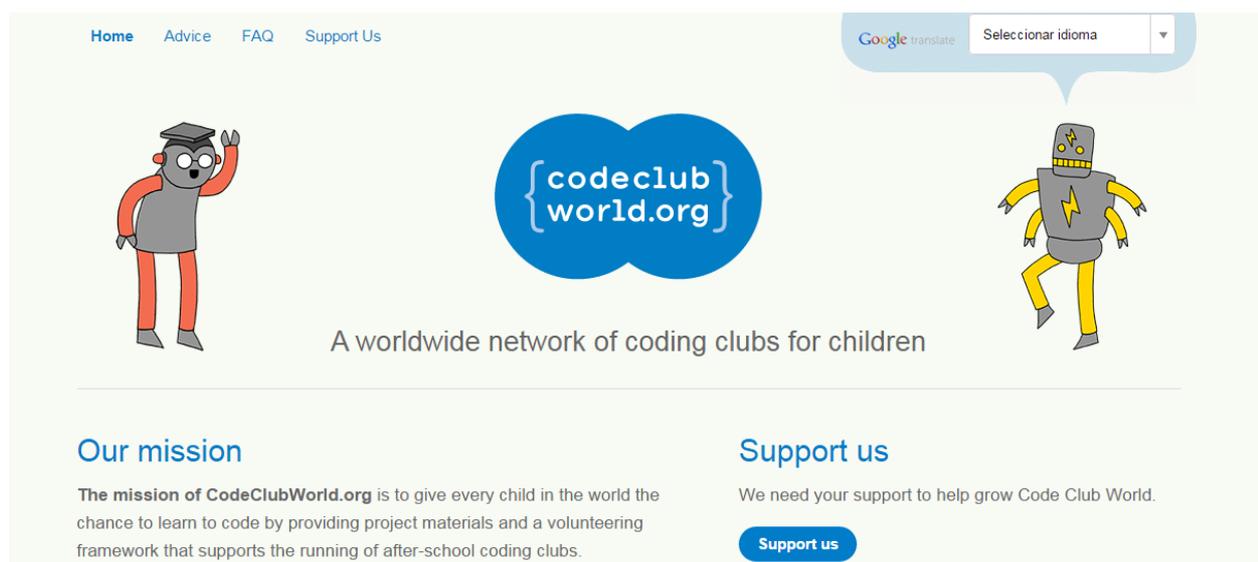


IMAGEN 5. CAPTURA DE PANTALLA DE LA WEB DE CODECLUBWORLD (ELABORACIÓN PROPIA)

El emplazamiento físico puede ser una escuela, una biblioteca, un centro cívico o similar que disponga de algunos ordenadores y conexión a internet. Las personas interesadas en organizar esta actividad deben registrar su ubicación y sus datos en un registro global de ‘CodeClubs’ o Clubs de Código o Clubs de Programación.

Para las edades propias de la educación primaria el proyecto apuesta por enseñar a programar mediante el lenguaje Scratch (lenguaje visual), pero para edades más avanzadas también está previsto que se enseñen otros lenguajes como HTML y CSS (más específicos para el desarrollo de páginas web) y el Python (lenguaje con editor de instrucciones).

La organización global provee de todo el material necesario para llevar a cabo estos aprendizajes, que puede descargarse de la web. En este punto cabe destacar que al ser Scratch un lenguaje abierto y gratuito todos los clubs del mundo son potencialmente creadores de material educativo.

La idea original partió de dos mujeres vinculadas profesionalmente al mundo de las nuevas tecnologías, Clare Sutcliffe<sup>21</sup> y Linda Sandvik, que plantearon la necesidad de que el currículum escolar, además de enseñar a los niños a ser usuarios de software, debía enseñarles a ser ellos mismos creadores de software ya que este aprendizaje les iba a situar en una mejor posición para desenvolverse en un medio social y laboral eminentemente tecnológico. En su opinión la programación es una actividad divertida y creativa que fortalece el pensamiento lógico y la capacidad de resolver problemas. Además, este aprendizaje en la edad escolar permitiría a muchos niños descubrir un mundo con un gran futuro laboral ya que hay un importante déficit de

<sup>21</sup> TED (2012a). Conferencia de Clare Sutcliffe, co-fundadora del Proyecto CodeClub. “Hacking the future”. Recuperado de [https://www.youtube.com/watch?v=ng7sf2\\_peFg&feature=youtu.be](https://www.youtube.com/watch?v=ng7sf2_peFg&feature=youtu.be)  
 APRENDER A PROGRAMAR EN LA ESCUELA PARA LA MEJORA COMPETENCIAL EN EDUCACIÓN PRIMARIA 25  
*Estudio teórico y pautas de formación docente*

programadores disponibles para las demandas del creciente sector industrial de las nuevas tecnologías.

#### 2.5.1.2.2. Proyecto CoderDojo desde Irlanda

El proyecto CoderDojo<sup>22</sup> es también una iniciativa privada sin ánimo de lucro y financiada por sus patrocinadores, generalmente empresas del sector tecnológico. De forma similar a los CodeClubs, los CoderDojos son espacios de enseñanza liderados por voluntarios para la difusión del aprendizaje de la programación en la infancia y la juventud.



IMAGEN 6. CAPTURA DE PANTALLA DE LA WEB DE CODERDOJO (ELABORACIÓN PROPIA)

Esta iniciativa nació en Irlanda en el año 2011 a partir de las inquietudes en este sentido de varios profesionales del mundo de los medios tecnológicos como diseño industrial, marketing, etc.

Desde Abril de 2015 CoderDojo cuenta con Microsoft entre sus patrocinadores.

Entre los lenguajes de programación con los que trabajan están Scratch, HTML, Python, Ruby, Java y Minecraft, así como proyectos para hardware Arduino.

Una característica propia de los CoderDojo es su apuesta por involucrar también a los padres, de hecho se recomienda que asistan acompañando a los menores de 13 años.

<sup>22</sup> CoderDojo (2011). Web central del Proyecto CoderDojo. Recuperado de <https://coderdojo.com/>

## 2.5.2. Propuestas o experiencias en España

### 2.5.2.1. Propuesta Código 21 del Gobierno de Navarra

Código 21<sup>23</sup> es un proyecto liderado por el Departamento de Educación de la Comunidad de Navarra que se propone el aprendizaje de la programación, de la robótica educativa y de otras tecnologías emergentes. Se trata de un proyecto abierto a docentes, alumnos y familias en general. Mediante su plataforma en internet se invita a todos los miembros de la comunidad educativa y personas interesadas en general a conocerlo, difundirlo y colaborar en el mismo de forma activa.

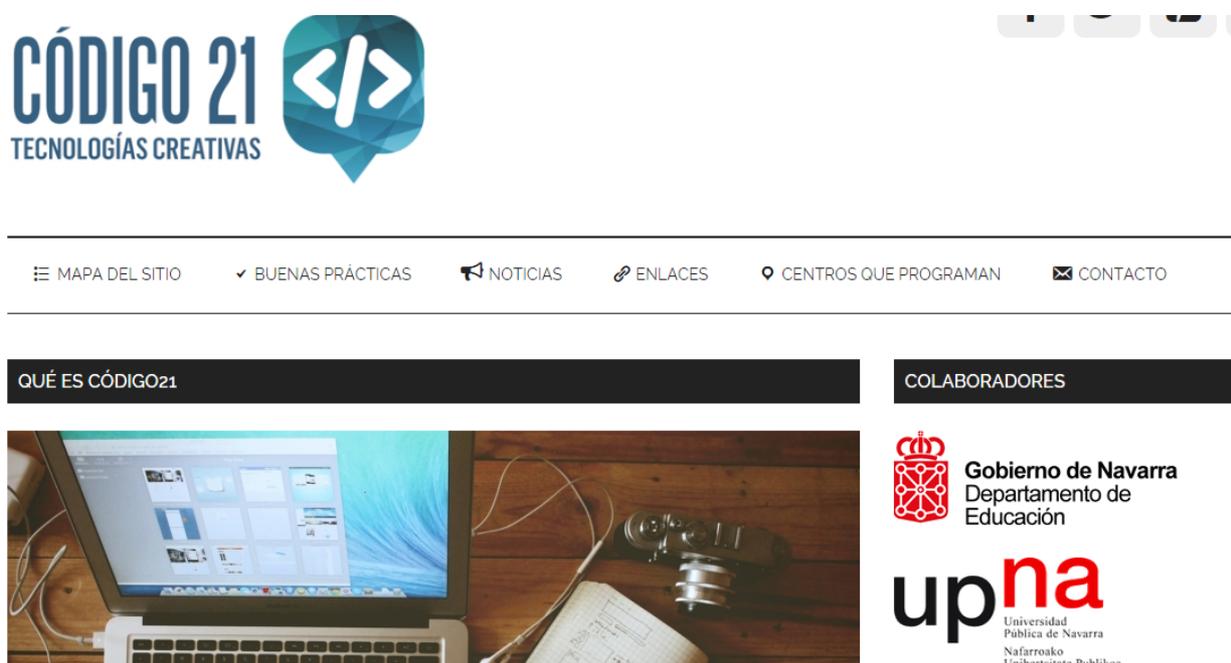


IMAGEN 7. CAPTURA DE PANTALLA DE LA WEB DE CÓDIGO 21 DEL GOBIERNO DE NAVARRA (ELABORACIÓN PROPIA)

El proyecto Código 21 de la Comunidad de Navarra se inspira, entre otras, en las propuestas de un organismo tan relevante en esta materia como es el Lifelong Kindergarten del MIT Media Lab (creadores del lenguaje Scratch). La idea principal es que aunque los niños de hoy en día sean usuarios habituales de las nuevas tecnologías (“nativos digitales”), fundamentalmente son consumidores pasivos del medio digital y no creadores, en general no usan ese medio para expresarse ellos mismos. El profesor Mitch Resnick, responsable de este departamento del MIT, lo resume gráficamente al decir que “*nuestros jóvenes pueden leer, pero no escribir con las TIC*”<sup>24</sup>,

<sup>23</sup> Código 21 (2015a). Web del Proyecto Código 21 del Gobierno de Navarra para el aprendizaje de la programación en la escuela. Recuperado de <http://codigo21.educacion.navarra.es/>

<sup>24</sup> TED (2012b). Conferencia de Mitch Resnick, especialista en educación y tecnología del MIT. “Let’s teach kids to code”. Recuperado de [http://www.ted.com/talks/mitch\\_resnick\\_let\\_s\\_teach\\_kids\\_to\\_code](http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code)

en definitiva Resnick se muestra escéptico con la idea de que los niños son nativos digitales y no necesitan formación en este terreno, al contrario, considera fundamental ayudarles a alcanzar la suficiente ‘fluidez’ en el medio digital para que puedan usarlo para aprender otras cosas y también como medio de expresión de sus propias ideas y sentimientos. Como podemos observar son las mismas ideas que impulsan los planes de la European Coding Initiative (Agenda Digital Europea) en esta materia, así como las iniciativas antes comentadas de CodeClubs y CoderDojo.

El Proyecto Código 21 fue presentado en Enero de 2015<sup>25</sup> por el consejero de Educación del Gobierno de Navarra. La propuesta inicial tiene el objetivo de formar en programación (mediante el lenguaje Scratch), así como en robótica, a alumnos de de 4º y 5º de Primaria de 90 centros educativos de la Comunidad. En la presentación del proyecto participó también Jesús Moreno, cofundador de la Asociación Programamos y colaborador del INTEF (Instituto Nacional de Tecnologías Educativas y Formación del Profesorado del Ministerio).

En el acto se presentó la web del proyecto así como dos guías docentes. La primera de ellas, denominada ‘*Informática Creativa*’<sup>26</sup> es la traducción al español del documento ‘Creative Computing’ de la Harvard Graduate School of Education y constituye una guía práctica para aprender Scratch a través de ejemplos que se pueden utilizar como material educativo para sesiones de clase.

El otro documento es una guía para el profesorado que se titula ‘*Programar para aprender: Orientaciones para el profesorado de primaria*’<sup>27</sup> e incluye con orientaciones y recursos para los docentes. Esta guía contiene la referencia al texto legal que establece el currículo de la educación primaria de la Comunidad Foral de Navarra, el Decreto Foral 60/2014, de 16 de julio, que incorpora las herramientas de programación en el contenido del Área de Matemáticas para los cursos 4º y 5º de primaria.

### 2.5.2.2. Asociación Programamos

La Asociación Programamos<sup>28</sup> es una comunidad virtual sin ánimo de lucro fundada por un equipo de profesionales españoles expertos en informática y en docencia que comparten la idea de que el

---

<sup>25</sup> Código 21 (2015b). Presentación del Proyecto Código 21 del Gobierno de Navarra. Recuperado de <http://codigo21.educacion.navarra.es/2015/01/12/presentacion-del-proyecto-codigo-21-de-programacion-educativa-en-el-que-participan-90-centros/>

<sup>26</sup> Código 21 (2015c). Guía ‘Informática Creativa’ del Proyecto Código 21 del Gobierno de Navarra. Recogido de <http://codigo21.educacion.navarra.es/profesorado/guia-informatica-creativa/>

<sup>27</sup> Código 21 (2015d). Guía ‘Programar para aprender’ del Proyecto Código 21 del Gobierno de Navarra. Recogido de <http://codigo21.educacion.navarra.es/profesorado/guia-programar-para-aprender-orientaciones-para-el-profesorado-de-primaria/>

<sup>28</sup> Programamos (2015a). Web de la Asociación Programamos. Recuperado de <http://programamos.es/>

pensamiento computacional<sup>29</sup> favorece el aprendizaje significativo de la matemáticas y las ciencias en general, además de promover la creatividad y las ganas de aprender.

El pensamiento computacional viene a ser aquel que formula los problemas, del tipo que sean, de modo que podrían ser resueltos mediante la secuencia de instrucciones adecuada como lo haría el programa de ordenador que lo podría resolver. Este tipo de pensamiento favorece el análisis, la concentración, la abstracción, el uso razonado de algoritmos, la motivación, etc.

*“El objetivo principal con el que nace este proyecto es tratar de modificar la forma en la que los estudiantes se relacionan con la tecnología, pasando de ser consumidores a creadores tecnológicos a través de la programación de videojuegos y el desarrollo de aplicaciones para dispositivos móviles.”* Programamos (2015a).

FAQ Quiénes somos Suscribirse Contacta

Programamos  
Videojuegos y 'apps'

El proyecto Materiales Blog Servicios Comunidad

## Aprender a programar:

Imaginar, razonar, crear, colaborar, compartir, emprender... y además

¡Es muy divertido y satisfactorio!

**“Todo el mundo debería saber programar, desde niños hasta mayores”**

IMAGEN 8. CAPTURA DE PANTALLA DE LA WEB DE PROGRAMAMOS (ELABORACIÓN PROPIA)

La Asociación se propone promover el pensamiento computacional desde edades muy tempranas y para llevar a cabo estas iniciativas apuesta especialmente por estos dos entornos de programación, el lenguaje Scratch para la creación de videojuegos en las primeras etapas educativas y el lenguaje APP Inventor para la creación de apps para móviles, más orientado a secundaria.

<sup>29</sup> Programamos (2015b). Posición de la Asociación Programamos sobre el pensamiento computacional. Recuperado de <http://programamos.es/que-es-el-pensamiento-computacional/>  
 APRENDER A PROGRAMAR EN LA ESCUELA PARA LA MEJORA COMPETENCIAL EN EDUCACIÓN PRIMARIA 29  
 Estudio teórico y pautas de formación docente

## 2.6. Conclusiones del apartado

Se puede afirmar que la función de transmisión pasiva de conocimientos por parte de la escuela ha quedado en parte desfasada por el acceso masivo a las nuevas tecnologías que ponen esos mismos conocimientos al alcance de todos, conocimientos cuyo contenido y forma de presentación pueden ser además permanentemente actualizados y mejorados gracias en gran parte al trabajo desinteresado de un número creciente de comunidades de docentes, de padres y de personas interesadas en general que comparten sus recursos en la red. El profesor, liberado así de esta tarea ingente (que pasa a ser una tarea compartida por la comunidad educativa en general), puede dedicarse más a funciones que sólo puede hacer el/ella en exclusiva, como son guiar, orientar, inspirar, contagiar entusiasmo, realizar las adaptaciones curriculares pertinentes para su grupo de alumnos, etc.

En este contexto general, vemos que aprender a programar no es un elemento más, sino que aporta una diferencia muy relevante con respecto a los demás recursos tecnológicos educativos disponibles. Programar es por definición crear algo nuevo, algo que no existía antes. Sea una composición artística, una animación audiovisual, la representación de un problema matemático y su resolución o la presentación de cualquier conocimiento o contenido educativo.

Los entornos de programación adaptados a la etapa escolar permiten que el profesor haga propuestas y que los alumnos (con su ayuda) construyan, inventen, ensayen, aprendan de sus errores, compartan y sobre todo aprendan a confiar en sus capacidades.

Ahora bien, para que todo ese potencial, toda esa ingente cantidad de recursos (la mayoría de acceso libre) que tenemos disponibles en la red llegue a nuestros alumnos hacen falta entre otros estos dos elementos imprescindibles:

- \* Que los centros estén suficientemente equipados tecnológicamente incluyendo el correspondiente servicio de mantenimiento periódico tanto de dispositivos físicos como de software.
- \* Que los profesores estén formados en esta temática y tengan la oportunidad de reciclarse permanentemente.

Según las opiniones manifestadas por docentes en Inevery Crea (2014b) estos dos elementos no están suficientemente atendidos en nuestro sistema educativo actual ni mucho menos. Habiendo además una gran desigualdad entre las situaciones de los distintos centros debido a que las mejoras en este terreno responden muchas veces al interés y dedicación personal de algunos docentes.

Finalmente comentar que aunque el presente trabajo se ocupa de la etapa de educación primaria, son también numerosas las iniciativas que aplican en otras etapas educativas. A modo de ejemplo podemos comentar las siguientes:

- \* El proyecto PRIMO<sup>30</sup>, iniciativa privada procedente del Reino Unido dirigida a la etapa de educación infantil. Se trata de un pequeño robot en forma de cubo que los niños mueven (programan) mediante la manipulación física de pequeñas piezas que encajan en un tablero.
- \* El proyecto Codemadrid<sup>31</sup>, iniciativa del Gobierno de la Comunidad de Madrid destinada a dar la formación necesaria a los profesores de enseñanza secundaria de las asignaturas de Tecnología e Informática para que puedan impartir la nueva asignatura de Programación.
- \* El programa mSchools<sup>32</sup>, iniciativa en la que colaboran la Fundación Mobile World, la GSMA (organización de operadores de móviles), el Gobierno de Cataluña y el Ayuntamiento de Barcelona para incorporar al currículum de 4º ESO una nueva asignatura optativa de programación de aplicaciones para móviles así como procurar a los profesores interesados la formación necesaria.

### 3. PAUTAS METODOLÓGICAS DE INTERVENCIÓN PARA APRENDER A PROGRAMAR EN EL AULA DE EDUCACIÓN PRIMARIA

#### 3.1. Introducción

Como hemos constatado a lo largo del presente trabajo, aún existe un desconocimiento notable por parte de la comunidad educativa en general y del profesorado de primaria en particular, acerca de la filosofía, los objetivos y los contenidos que proponen los distintos movimientos que promueven la incorporación de la enseñanza de la programación en la educación primaria.

---

<sup>30</sup> PRIMO (2013). Vídeo ejemplo del Proyecto PRIMO. Robots educativos programables mediante la manipulación física de fichas. Recuperado de [https://www.youtube.com/watch?v=W0sE1j\\_kqY](https://www.youtube.com/watch?v=W0sE1j_kqY)

<sup>31</sup> Codemadrid (2014). Web de la plataforma de la Consejería de Educación de Madrid para la capacitación digital del profesorado. Recuperado de <http://codemadrid.org/>

<sup>32</sup> mSchools (2012). Iniciativa mixta público-privada en Cataluña con la fundación Mobile World para promover la incorporación en las aulas de la tecnología móvil. Recuperado de <http://www.xtec.cat/web/projectes/mschools>

Esto se debe a que a pesar de las directrices claras en este sentido desde las más altas instancias europeas competentes estas ideas tienen aún un reflejo apenas incipiente en el currículum escolar en nuestro país. Por esta razón a día de hoy estas iniciativas están llamando a la puerta de las escuelas en forma de propuestas extraescolares ofrecidas por empresas privadas. De este modo, estos aprendizajes llegarán sólo a un pequeño colectivo de alumnos -los matriculados en las actividades extraescolares-, y por otra parte, al ser desempeñados por profesionales de la enseñanza ajenos al colectivo de profesores éstos seguirán al margen y por tanto sin incorporar estos aprendizajes a su tarea docente.

Como hemos escuchado opinar en Inevery Crea (2014b) a los docentes interesados y concienciados de la importancia de la alfabetización digital de los niños y de los valores del pensamiento computacional en su desempeño escolar, para que todos los alumnos puedan beneficiarse de estos aprendizajes es imprescindible contar con la participación activa de los profesores.

Por lo tanto, dado que en la actualidad no están generalizados en nuestro país, o al menos no de forma suficiente, programas de capacitación digital para el profesorado, y menos en la materia específica de la enseñanza de la programación, la presente propuesta de intervención educativa se propone trabajar en favor de esta causa y consiste en la realización de un taller de programación con lenguaje Scratch para los profesores de una escuela de primaria con el objeto de capacitarles para que, en un paso posterior, ellos mismos puedan impartir talleres de programación a sus alumnos o bien sencillamente integren este conocimiento en su bagaje docente y puedan usarlo como instrumento de aprendizaje transversal de las diversas áreas del currículum.

La intervención se propone como un punto de partida en la capacitación en programación del profesorado del centro, de manera que aunque se realice inicialmente con un mínimo número de profesores (aunque lo aconsejable es que sea el mayor número posible), este pequeño grupo inicial puede después extender esta formación al resto del colectivo de forma que esta cadena de aprendizaje y enseñanza alcance finalmente al alumnado.

### **3.2. Contexto de intervención**

La presente propuesta se puede realizar en cualquier espacio de la escuela que disponga de ordenadores, aunque lo más propio sería el aula de Informática del centro si dispone de ella, ya que así se garantiza que el software básico de los equipos (que después serán utilizados por el alumnado) está en perfecto estado de actualización. Además como veremos más adelante estos equipos quedarán con el software del lenguaje Scratch instalado.

### 3.3. Objetivos de la intervención

- Mejorar la capacitación digital del profesorado de primaria, específicamente en el área de la programación de computadores, mediante el aprendizaje del lenguaje Scratch.
- Mejorar la capacidad crítica del profesorado de primaria ante las diversas propuestas que puedan llegar a la escuela en este ámbito, tanto si proceden de las autoridades educativas como desde iniciativas privadas.
- Promover la colaboración en la red del colectivo de profesores de primaria, compartiendo todo tipo de recursos y experiencias en esta y otras materias de interés común.

### 3.4. Destinatarios de la intervención

Los destinatarios de la presente propuesta de intervención educativa son el colectivo de profesores de un centro de enseñanza primaria con el propósito de que posteriormente, a su vez, puedan transmitir estos conocimientos y destrezas a sus alumnos -que son los destinatarios últimos de toda propuesta educativa-. Así mismo, la presente propuesta también podría destinarse a colectivos de padres interesados en la temática.

### 3.5. Desarrollo de la propuesta de intervención

#### 3.5.1. Relación de Sesiones y Actividades

En este apartado se detallan las doce sesiones que integran la presente propuesta de intervención metodológica. Tras unas sesiones introductorias al tema se pasa al aprendizaje sistemático del software propuesto (lenguaje Scratch) tratando de combinar teoría y práctica. En las últimas sesiones se incide en el carácter global de este tipo de propuestas tecnológicas y en los recursos disponibles en la red para los profesores interesados.

## 3.5.1.1. Sesión 1: Aprender a programar ¿Para qué?

Sesión 1: Aprender a programar ¿Para qué?	
Objetivos:	<ul style="list-style-type: none"> <li>• Conocer las opiniones de los expertos más relevantes en esta temática.</li> <li>• Promover la propia opinión al respecto.</li> </ul>
Actividades:	<ol style="list-style-type: none"> <li>1. Visionar los siguientes documentos: <ul style="list-style-type: none"> <li>• Conferencia de Mitch Resnick ‘<i>Enseñemos a los niños a programar</i>’. TED (2012b)</li> <li>• Entrevista a Antonio A. Ruíz ‘<i>Scratch, estupenda herramienta para trabajar las inteligencias múltiples</i>’. Inevery Crea (2014<sup>a</sup>)</li> <li>• Entrevista a Antolín García ‘<i>No podemos dejar solo al docente en el aprendizaje de la programación</i>’. Inevery Crea (2014c)</li> </ul> </li> <li>2. Debate abierto sobre el tema, examinar nuestro punto de partida e intercambiar opiniones acerca de las manifestaciones escuchadas.</li> </ol>
Metodología:	Metodología expositiva y promoción del libre debate.
Evaluación:	Auto reflexión sobre el grado de conocimiento inicial en la temática y las aportaciones recibidas en la sesión.

## 3.5.1.2. Sesión 2: ¿Qué es Scratch?

Sesión 2: ¿Qué es Scratch?	
Objetivos:	<ul style="list-style-type: none"> <li>• Conocer los elementos básicos de la programación a través de Scratch.</li> <li>• Saber acceder al lenguaje Scratch en modo Offline y Online.</li> </ul>
Actividades:	<ol style="list-style-type: none"> <li>1. Breve introducción teórica a la programación y a los lenguajes de programación. Destacar una diferencia sustancial entre los lenguajes propios de adultos como Basic o Java y los lenguajes creados para niños, los primeros suelen usar un lenguaje similar al humano aunque algo encriptado y los segundos tienen interfaces gráficas como Scratch.</li> <li>2. Instalar en el ordenador el software del lenguaje Scratch desde la web oficial del MIT <a href="https://scratch.mit.edu/scratch2download/">https://scratch.mit.edu/scratch2download/</a></li> <li>3. Acceder a la plataforma online de Scratch <a href="https://scratch.mit.edu/">https://scratch.mit.edu/</a></li> </ol>
Metodología:	Combinación de metodología expositiva y actividad guiada.

Evaluación:	Poder trabajar con Scratch de forma online y también usando el software instalado.
-------------	--

### 3.5.1.3. Sesión 3: Visión general de entorno de programación de Scratch

Sesión 3: Visión general del entorno de programación de Scratch	
Objetivo:	<ul style="list-style-type: none"> <li>Identificar los elementos principales del entorno gráfico de programación de Scratch.</li> </ul>
Actividades:	<ol style="list-style-type: none"> <li>Revisar los elementos principales: Escenario, Personajes/Objetos y Bloques apilables (Instrucciones del lenguaje).</li> <li>Breve explicación sobre la esencia de un ‘proyecto’ Scratch (los programas Scratch se denominan proyectos): Esencialmente es una secuencia de instrucciones que determinan las acciones que van a realizar unos personajes en el marco de un determinado escenario.</li> <li>Probar las distintas opciones para cambiar el escenario y los personajes: Capturar elementos de la biblioteca interna, subir elementos desde nuestro ordenador, crear nosotros mismos tanto escenarios como objetos o personajes mediante un editor gráfico incorporado al lenguaje.</li> </ol>
Metodología:	Combinación de metodología expositiva y actividad guiada.
Evaluación:	Saber cambiar tanto el escenario de fondo como sustituir o añadir nuevos personajes y objetos a nuestro proyecto.

### 3.5.1.4. Sesión 4: Tipos de Bloques apilables (Instrucciones Scratch)

Sesión 4: Tipos de Bloques apilables (Instrucciones Scratch)	
Objetivo:	<ul style="list-style-type: none"> <li>Conocer los distintos tipos de Bloques de Scratch.</li> </ul>
Actividad:	<p>Breve explicación sobre la funcionalidad propia de los distintos tipos de bloques seguida de la experimentación con ejemplos de cada uno de ellos. Cada tipo de bloque se identifica con un color.</p> <ul style="list-style-type: none"> <li><u>De Movimiento</u>: Permiten situar objetos en un punto específico del escenario mediante coordenadas x/y, así como desplazarse en cualquier dirección, girar un determinado número de grados, etc.</li> </ul>

	<ul style="list-style-type: none"> <li>• <u>De Apariencia</u>: Permiten cambiar el aspecto de objetos y personajes como el color o el tamaño así como asociarles la expresión de un texto que puede aparecer dentro de una ‘nube de diálogo’ o de una ‘nube de pensamiento’.</li> <li>• <u>De Sonido</u>: Permiten reproducir cualquier sonido que grabemos (viene incorporado el maullido de un gato). También podemos reproducir cualquier sonido musical indicando instrumento y altura de sonido.</li> <li>• <u>Función Lápiz</u>: Hace que los objetos dejen un trazo de su desplazamiento por el escenario. Combinado con los bloques de movimiento permite dibujar figuras geométricas.</li> <li>• <u>Bloques de Eventos y de Control</u>: Que permiten establecer la secuencia de ejecución en los programas que contienen más de una secuencia de instrucciones.</li> </ul>
Metodología:	Experimentación inicialmente guiada y después libre.
Evaluación:	Saber qué tipo de bloque debemos usar para conseguir que un personaje u objeto realice un determinado tipo de acción.

### 3.5.1.5. Sesión 5: Trabajando con Scratch: Hacer una composición visual

Sesión 5: Trabajando con Scratch: Hacer una composición visual	
Objetivo:	<ul style="list-style-type: none"> <li>• Saber hacer una composición gráfica con Scratch.</li> </ul>
Actividades:	<ol style="list-style-type: none"> <li>1. Ver un ejemplo de tarjeta de felicitación hecha con Scratch y analizar su código (bloques de instrucciones).</li> <li>2. Hacer una copia de dicho proyecto y a continuación cambiar libremente el código para realizar una composición personal.</li> </ol>
Metodología:	Secuencia de análisis, reproducción y experimentación.
Evaluación:	Saber crear pequeñas composiciones visuales artísticas con Scratch.

### 3.5.1.6. Sesión 6: Trabajando con Scratch: Hacer una composición audiovisual

Sesión 6: Trabajando con Scratch: Hacer una composición audiovisual	
Objetivo:	<ul style="list-style-type: none"> <li>Saber hacer una composición audiovisual con Scratch.</li> </ul>
Actividad:	Modificar el proyecto realizado en la sesión anterior e incorporarle sonido.
Metodología:	Experimentación inicialmente guiada y después libre.
Evaluación:	Saber crear pequeñas composiciones audiovisuales artísticas.

### 3.5.1.7. Sesión 7: Trabajando con Scratch: Conversando con mi mascota

Sesión 7: Trabajando con Scratch: Conversando con mi mascota	
Objetivo:	<ul style="list-style-type: none"> <li>Saber solicitar y tratar la información introducida desde el teclado para interactuar con los personajes.</li> </ul>
Actividades:	<ol style="list-style-type: none"> <li>Ver un ejemplo de proyecto Scratch que contenga un diálogo con el usuario y analizar su código.</li> <li>Hacer una copia de dicho proyecto y a continuación cambiar libremente el código para realizar una composición libre.</li> </ol>
Metodología:	Secuencia de análisis, reproducción y experimentación.
Evaluación:	Saber cómo entrar información a un programa desde el teclado.

### 3.5.1.8. Sesión 8: Trabajando con Scratch: El gato se sabe las tablas de multiplicar

Sesión 8: Trabajando con Scratch: El gato se sabe las tablas de multiplicar	
Objetivo:	<ul style="list-style-type: none"> <li>Saber manejar los operadores lógicos y aritméticos del lenguaje Scratch.</li> </ul>
Actividades:	<ol style="list-style-type: none"> <li>Crear diversas listas de bloques (procedimientos) que calculen y muestren las tablas de multiplicar, después hacer que el gato pregunte un número del 1 al 9 y utilizar esa entrada para lanzar la ejecución de la lista que corresponda.</li> <li>Hacer variaciones sobre este programa utilizando los demás operadores aritméticos.</li> </ol>
Metodología:	Experimentación guiada y experimentación libre.
Evaluación:	Saber utilizar los operadores lógicos y aritméticos de Scratch.

### 3.5.1.9. Sesión 9: Trabajando con Scratch: Dibujando figuras geométricas

Sesión 9: Trabajando con Scratch: Dibujando figuras geométricas	
Objetivo:	<ul style="list-style-type: none"> <li>Saber utilizar la función del lápiz de Scratch.</li> </ul>
Actividades:	<ol style="list-style-type: none"> <li>Probar todas las posibilidades que ofrece la función lápiz (color, grosor, etc.) trazar líneas al tiempo que desplazamos un objeto por el escenario combinando los bloques de mover y girar.</li> <li>Realizar procedimientos (listas de bloques) que dibujen las figuras geométricas planas más simples (triángulo, cuadrado, rombo, etc.).</li> </ol>
Metodología:	Experimentación guiada y experimentación libre.
Evaluación:	Realizar trazos geométricos (líneas y curvas) con Scratch.

### 3.5.1.10. Sesión 10: Comunidad Scratch: Compartir proyectos en la red

Sesión 10: Comunidad Scratch: Compartir proyectos en la red	
Objetivo:	<ul style="list-style-type: none"> <li>Saber utilizar la funcionalidad de compartir proyectos Scratch en la red mediante la plataforma online.</li> </ul>
Actividades:	<ol style="list-style-type: none"> <li><u>Función ‘Explorar’</u>: Acceder a los proyectos realizados y subidos a la plataforma de Scratch por otros usuarios en cualquier parte del mundo. Están catalogados por etiquetas del tipo ‘Animación’, ‘Juego’, ‘Música’, etc. Podemos editarlos, probarlos, modificarlos y descargarlos en nuestro ordenador.</li> <li><u>Función ‘Compartir’</u>: Subir nuestros proyectos a la plataforma de Scratch para compartirlos con todo el mundo. En el momento de subir un proyecto a la plataforma nos pide acompañarlo de una mínima información: Breves instrucciones sobre su uso, Comentarios y Etiqueta para clasificarlo (‘Animación’, ‘Juego’, etc.)</li> </ol>
Metodología:	Experimentación guiada y experimentación libre.
Evaluación:	Saber descargar y cargar proyectos en la plataforma online de Scratch.

## 3.5.1.11. Sesión 11: Comunidades educativas virtuales: Apoyo al profesorado en la red

Sesión 11: Comunidades educativas virtuales: Apoyo al profesorado en la red	
Objetivo:	<ul style="list-style-type: none"> <li>• Conocer algunas de las comunidades virtuales especializadas en soporte al profesorado en las TIC en general y en programación en particular.</li> </ul>
Actividad:	<p>Acceder a la web de las siguientes organizaciones, conocer sus propuestas, recursos que ofrecen, eventos que promueven, etc.:</p> <ul style="list-style-type: none"> <li>• Asociación ‘Programamos’: <a href="http://programamos.es/">http://programamos.es/</a></li> <li>• Plataforma Inevery Crea: <a href="http://ineverycrea.net">http://ineverycrea.net</a></li> <li>• Blog del profesor Antonio A. Ruíz: <a href="http://jueduco.blogspot.com.es/">http://jueduco.blogspot.com.es/</a></li> <li>• Asociación de padres y educadores ‘Con Más Futuro’: <a href="http://www.conmasfuturo.es/">http://www.conmasfuturo.es/</a></li> <li>• Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado del Ministerio de Educación: <a href="http://educalab.es/intef">http://educalab.es/intef</a></li> <li>• Academia Europea en la red: <a href="http://www.europeanschoolnetacademy.eu/">http://www.europeanschoolnetacademy.eu/</a></li> <li>• Web especializada de la Agenda Digital Europea ‘All you need is code’: <a href="http://www.allyouneediscode.eu/">http://www.allyouneediscode.eu/</a></li> <li>• Proyecto CoderDojo: <a href="https://coderdojo.com/">https://coderdojo.com/</a></li> <li>• Proyecto CodeClub: <a href="http://codeclubworld.org/">http://codeclubworld.org/</a></li> </ul>
Metodología:	Experimentación guiada y experimentación libre.
Evaluación:	Acceder a las plataformas propuestas y saber registrarse en ellas.

## 3.5.1.12. Sesión 12: Conclusiones

Sesión 12: Conclusiones	
Objetivos:	<ul style="list-style-type: none"> <li>• Reflexionar sobre las ventajas educativas de la programación y la importancia de la formación del profesorado en esta materia.</li> <li>• Formular conclusiones sobre el taller realizado.</li> </ul>
Actividad:	<p>Debate abierto para la reflexión sobre:</p> <ul style="list-style-type: none"> <li>• Principios metodológicos y ventajas educativas de enseñar a programar en primaria.</li> </ul>

	<ul style="list-style-type: none"> <li>• Posibilidades de incidir con Scratch en la adquisición de las distintas Competencias Básicas (interdisciplinariedad).</li> </ul>
Metodología:	Debate abierto.
Evaluación:	Auto evaluación acerca de la utilidad del taller realizado y en su caso del grado de suficiencia alcanzado para poder incorporar en su discurso docente los conocimientos adquiridos.

### 3.5.2. Temporalización y Recursos

#### 3.5.2.1. Temporalización

La presente propuesta de intervención educativa se extendería a un trimestre (12 semanas) a razón de una sesión semanal de una hora de duración.

#### 3.5.2.2. Recursos materiales

La propuesta de intervención requiere disponer de un espacio físico con ordenadores. Así como también de un proyector o pizarra digital. Un equipamiento idóneo sería en torno a 10 ordenadores. Pueden tener cualquiera de los sistemas operativos más habituales, Mac, Windows o Linux.

Respecto al navegador de internet, según la plataforma online de Scratch (2015a) puede ser Chrome 35 o posterior, Firefox 31 o posterior o Internet Explorer 9 o posterior, incluyendo también Adobe Flash Player versión 10.2 o posterior.

Es preciso también contar con un servicio mínimo de mantenimiento tanto de los equipos físicos como del software básico instalado. Esto último implica una actualización periódica del sistema operativo y de los programas estándar necesarios para la gestión de contenidos gráficos y la navegación por la red.

El lenguaje Scratch dispone de una plataforma on-line muy potente que nos permite trabajar liberándonos de la tarea de instalar y mantener actualizado su software en todos y cada uno de los equipos. En dicha plataforma podemos registrarnos como usuarios y gestionar un espacio propio para almacenar nuestras producciones, sean proyectos finalizados o en construcción, pruebas, etc. Además, podemos compartir con los demás aquellos proyectos que nos interese así como capturar proyectos realizados por otros usuarios. También es posible instalar el programa en los equipos lo que nos permitirá trabajar sin conexión a internet en caso necesario. Scratch es un software de uso libre y gratuito.

### 3.5.2.3. Recursos profesionales

Para impartir este taller de programación se requiere tener una preparación básica en TICs y un conocimiento amplio del lenguaje Scratch así como una capacitación docente en educación primaria. También es aconsejable estar al corriente de los cambios legislativos en el currículum de nuestro país, a nivel estatal y autonómico, así como de las nuevas propuestas de la Agenda Digital Europea. Del mismo modo es conveniente estar informado de las iniciativas privadas que puedan surgir en esta temática. En cuanto a la disponibilidad temporal se requiere una dedicación de 1 hora semanal durante 12 semanas.

### 3.5.2.4. Recursos didácticos

En cuanto a lo que habitualmente entendemos por los recursos ‘didácticos’, en nuestro caso se trata de proyectos Scratch realizados por otros usuarios y disponibles en la red a través de la plataforma online de Scratch. Dado que éste es un lenguaje de código abierto<sup>33</sup> y gratuito, cualquier persona puede crear código nuevo y colgarlo en internet, por lo que la fuente de programas que nos sirvan de inspiración y de ejemplo puede ser inagotable y muy diversa.

Consideramos como algo relevante en la presente propuesta el hecho de que se utilice un material educativo -software en este caso- de origen totalmente colaborativo. Pensemos que los proyectos pueden ser creado o mejorado por cualquier usuario de Scratch en el mundo.

### 3.5.3. Metodología

Como hemos podido ver en el planteamiento de las actividades, se proponen diversas dinámicas a nivel metodológico con la finalidad de lograr alcanzar los objetivos y desarrollar satisfactoriamente el proceso de enseñanza-aprendizaje.

\* Metodología expositiva y actividad guiada: Introducción conceptual a los elementos presentados y propuesta de experimentación.

\* Experimentación guiada y experimentación libre: Tras introducir los nuevos elementos de la sesión se propone un ejercicio modelo que posteriormente será ampliado por los alumnos.

\* Secuencia de análisis, reproducción y experimentación: Partiendo de un proyecto Scratch

---

<sup>33</sup> Junto al formato ‘ejecutable’ de los programas también están accesibles las propias instrucciones del lenguaje de programación, el llamado ‘código’ o ‘código fuente’ del programa.

disponible en la red que incluya los elementos a enseñar, se analiza su contenido y se proponen cambios y adaptaciones.

\* Debate abierto: Revisión crítica de los conceptos y destrezas adquiridos bajo el enfoque de su uso posterior como recurso educativo para la adquisición de las competencias básicas propias de la educación primaria.

#### 3.5.4. Evaluación

Como puede observarse en el detalle de las actividades se propone un método de evaluación específico en cada caso para verificar que se han alcanzado los objetivos de la sesión. La evaluación es por tanto continua a lo largo de la intervención.

Por otra parte, teniendo presente que los alumnos a los que se dirige la propuesta son profesores de educación primaria y teniendo en cuenta también el carácter eminentemente práctico del contenido de la intervención, el método de evaluación predominante es la autoevaluación. En el caso de las aportaciones teóricas se propone una revisión crítica y puesta en común valorativa de esas aportaciones. En cuanto a las actividades más propiamente experimentales, el marcado carácter auto evaluativo de la programación aporta de forma natural el instrumento evaluativo idóneo en este caso, el éxito en la realización de los experimentos propuestos.

Finalmente, respecto a la evaluación global de la propuesta metodológica se propone que el instructor recoja de forma sistemática, en un documento de trabajo, las aportaciones en este sentido que alcancen cierto consenso en el debate de la última sesión dedicada precisamente a extraer conclusiones. Además de ello, el instructor tomará buena cuenta de aquellos conceptos o propuestas experimentales que han tenido mayor dificultad en su asimilación o realización para mejorar su presentación en una futura intervención.

## 4. CONCLUSIONES, LIMITACIONES Y PROSPECTIVA

### 4.1. Conclusiones generales

A continuación, a modo de síntesis y reflexión, se ofrece una revisión de los objetivos del presente trabajo y el grado de cumplimiento que se haya podido alcanzar de los mismos.

*Objetivo general de realizar una propuesta de intervención metodológica en torno a la enseñanza de la programación en educación primaria con la finalidad de contribuir a la divulgación de los actuales movimientos, sean gubernamentales o no, que promueven la incorporación de este tipo de enseñanza en la escuela, mejorando así la participación activa y crítica de la comunidad educativa en estos procesos.*

El presente trabajo ha pretendido aportar una fundamentación teórica al tema de la incorporación de la enseñanza de la programación de ordenadores en la escuela primaria. En el apartado del marco teórico se hace una revisión de las principales iniciativas que hay en la actualidad en Europa y en España, promovidas tanto desde instancias públicas como privadas, así como de su grado de implementación.

Posteriormente se ha planteado una propuesta de intervención educativa en forma de taller de programación para profesores de enseñanza primaria basada en el lenguaje actualmente más utilizado en todo el mundo con esta finalidad, el lenguaje Scratch desarrollado por el MIT y que es de acceso libre gratuito en la red. Además de la formación específica en programación en Scratch la intervención pretende divulgar la filosofía que inspira este tipo de propuestas que se resumen en una serie de ventajas educativas ampliamente detalladas a lo largo del presente trabajo.

*Objetivo específico de conocer el posicionamiento de la Unión Europea, como nuestro marco de referencia cultural y legislativo, en relación a la incorporación de la programación al currículum escolar de primaria.*

A lo largo del trabajo hemos podido constatar la apuesta decidida de la Unión Europea a favor de la incorporación de la programación al currículum escolar a todos los niveles. A través de la Agenda Digital se promueven constantes iniciativas y se aportan recursos en la red con esta finalidad. Así mismo este organismo hace un seguimiento de la implementación de sus propuestas en los diferentes países de la Unión en el marco del compromiso que representa la Estrategia Europa 2020.

*Objetivo específico de conocer las iniciativas más avanzadas que se están desarrollando en torno a esta cuestión en la actualidad, a nivel europeo y en nuestro país, tanto desde instancias públicas como privadas.*

No ha sido posible realizar una reseña completa de las iniciativas legislativas en los países europeos por exceder la extensión del presente trabajo pero sí hemos podido constatar la casi unanimidad en

las autoridades educativas a favor de esta causa. Otra cosa es el grado de realización que es aún incipiente en términos generales.

Sí hemos podido destacar algunas de las principales iniciativas en esta temática de origen europeo pero que se están extendiendo a nivel global como por ejemplo CoderDojo o CedeClub.

También hemos podido constatar que si bien la LOMCE no incorpora este contenido en el currículum de educación primaria, si lo están empezando a hacer algunas Comunidades Autónomas como hemos visto el caso de Navarra y Madrid.

*Objetivo específico de contribuir a mejorar la capacidad crítica de las escuelas ante las diversas propuestas en este sentido.*

Una de las ideas que inspira la propuesta de intervención metodológica que se aporta en este trabajo es justamente contribuir a mejorar la capacidad crítica de las escuelas ante la multitud de informaciones, cada vez más numerosas, que están recibiendo desde distintos estamentos y organizaciones. Que los profesores estén formados en el tema resulta esencial para que puedan liderar o acompañar estos cambios haciendo que prevalezca por encima de todo el objetivo de la excelencia en la enseñanza.

*Objetivo específico de contribuir a divulgar el software actualmente más extendido para la enseñanza de la programación en la educación primaria.*

Precisamente en este objetivo se fundamenta la decisión de utilizar el lenguaje Scratch en la propuesta de intervención metodológica del presente trabajo. Es uno de los lenguajes por el que apuestan todas las organizaciones que se dedican a la difusión de la enseñanza de la programación para niños y existen miles de proyectos Scratch de todo tipo disponibles en la red que pueden ser usados como modelos para aprender ampliando y modificando su código.

## **4.2. Limitaciones del estudio**

A continuación se ofrecen las principales dificultades que han limitado el presente trabajo de final de grado:

- En relación a la elaboración del marco teórico sin duda la falta inicial de experiencia en la búsqueda de fuentes bibliográficas en la red ha condicionado el trabajo por la gran cantidad

de tiempo dedicado a esta búsqueda inicial.

- Otro aspecto, más específico del tema tratado y que añade dificultad a la hora de intentar recoger la máxima información posible, es la enorme variedad y sobre todo cantidad de proyectos e iniciativas en una temática de tanta actualidad.
- También destacar que por esa misma razón, al ser un tema tan novedoso e incipiente, el marco legal vigente no responde a las propuestas e iniciativas que, sobre todo lideradas desde la Unión Europea, se siguen en nuestro país y en los de nuestro entorno. Esa falta generalizada de legislación en la materia, aporta cierta inseguridad a la investigación del tema.
- En relación a la propuesta de intervención metodológica presentada, la limitación es clara ya que no ha podido llevarse a cabo por falta de tiempo disponible para realizarla en un centro educativo.

### 4.3. Futuras líneas de investigación

- En relación al marco teórico: Investigación sistemática a nivel europeo de las mejoras que se puedan constatar en el rendimiento escolar en aquellos centros educativos que incorporen la enseñanza de la programación. El estudio debería recoger específicamente los posibles avances en la adquisición de las Competencias Clave recogidas en la Recomendación 2006/962/EC del Parlamento Europeo y del Consejo. También debería recoger información relativa al nivel de capacitación de los docentes involucrados y a los recursos tecnológicos disponibles. Otro aspecto relevante a estudiar es la fórmula por la que se opta para la introducción de estos aprendizajes, como actividad extraescolar o como parte integrante del currículum, y en caso de ser esta segunda opción, destacar en lo posible si la programación se incorpora como una asignatura específica -de forma análoga a las otras áreas específicas del currículum- o bien se incorpora en la medida de lo posible en el conjunto de asignaturas a modo de instrumento transversal de enseñanza.
- En relación a la intervención metodológica: Llevar a la práctica la propuesta presentada en este trabajo para la capacitación del profesorado de enseñanza primaria en programación con lenguaje Scratch. Sería de gran interés hacer un seguimiento del impacto de su posterior desempeño docente en la adquisición de competencias básicas de sus alumnos. Estos resultados así como las aportaciones de los docentes en relación a su práctica podrían

contribuir enormemente a la mejora de los planes de capacitación para docentes en el área tecnológica en general y en la enseñanza de la programación en particular.

## 5. FUENTES BIBLIOGRÁFICAS

### 5.1. Referencias

Agenda Digital Europea (2010). Página web de la Comisión Europea de la Agenda Digital. Recuperado de <http://ec.europa.eu/digital-agenda/en>

Bahón, J. (2014). Conferencia ‘Las Inteligencias Múltiples en el aula. Técnicas y herramientas inteligentes’. Recuperado de <https://www.youtube.com/watch?v=WZQMo3kO4uM>

Castells, M. (2002). ‘La dimensión cultural de internet’ en Debates Culturales, Barcelona, Universitat Oberta de Catalunya. Recuperado de <http://www.uoc.edu/culturaxxi/esp/articulos/castells0502/castells0502.html>

Ceruzzi, P. E. (2008). ‘Historia de la informática’. Open Mind. Recuperado de [https://www.bbvaopenmind.com/wp-content/uploads/static/pdf/109-127\\_PAUL\\_E\\_CERUZZI\\_ESP\\_R.pdf](https://www.bbvaopenmind.com/wp-content/uploads/static/pdf/109-127_PAUL_E_CERUZZI_ESP_R.pdf)

CodeClub (2012). Web central del Proyecto CodeClub. Recuperado de <http://codeclubworld.org/>

Codemadrid (2014). Web de la plataforma de la Consejería de Educación de Madrid para la capacitación digital del profesorado. Recuperado de <http://codemadrid.org/>

CoderDojo (2011). Web central del Proyecto CoderDojo. Recuperado de <https://coderdojo.com/>

Código 21 (2015a). Web del Proyecto Código 21 del Gobierno de Navarra para el aprendizaje de la programación en la escuela. Recuperado de <http://codigo21.educacion.navarra.es/>

Código 21 (2015b). Presentación del Proyecto Código 21 del Gobierno de Navarra. Recuperado de <http://codigo21.educacion.navarra.es/2015/01/12/presentacion-del-proyecto-codigo-21-de-programacion-educativa-en-el-que-participan-90-centros/>

Código 21 (2015c). Guía 'Informática Creativa' del Proyecto Código 21 del Gobierno de Navarra. Recogido de <http://codigo21.educacion.navarra.es/profesorado/guia-informatica-creativa/>

Código 21 (2015d). Guía 'Programar para aprender' del Proyecto Código 21 del Gobierno de Navarra. Recogido de <http://codigo21.educacion.navarra.es/profesorado/guia-programar-para-aprender-orientaciones-para-el-profesorado-de-primaria/>

Comisión Europea (2014). Carta de las responsables de las Comisiones de Agenda Digital y de Educación a los responsables de educación de los estados miembros. Recuperado de [http://ec.europa.eu/information\\_society/newsroom/image/kroes\\_vassiliou\\_letter\\_6597.pdf](http://ec.europa.eu/information_society/newsroom/image/kroes_vassiliou_letter_6597.pdf)

Decreto Foral 60/2014, de 16 de julio, por el que se establece el currículo de las enseñanzas de Educación primaria en la Comunidad Foral de Navarra.

Decreto 89/2014, de 24 de julio, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el Currículo de la Educación Primaria.

Estrategia Europa 2020 (2010a). Plataforma de comunicación de la Comisión Europea sobre el seguimiento de las propuestas de la Estrategia Europa 2020 en los distintos países de la Unión. Recuperado de [http://ec.europa.eu/europe2020/index\\_es.htm](http://ec.europa.eu/europe2020/index_es.htm)

Estrategia Europa 2020 (2010b). Comunicación de la Comisión Europea acerca de la iniciativa Europa 2020. Recuperado de <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:2020:FIN:ES:PDF>

European Coding Initiative (2014). Iniciativa de la Agenda Digital Europea para la promoción del aprendizaje de la programación en el ámbito educativo. Recuperado de <http://www.allyouneediscode.eu/>

European Schoolet Academy (2014). Escuela online que ofrece formación en innovación a profesores de primaria y secundaria. Recuperado de <http://www.europeanschoolnetacademy.eu/>

Inevery Crea (2014a). Scratch, estupenda herramienta para trabajar inteligencias múltiples. Entrevista con Antonio A. Ruiz. Recuperado de <https://www.youtube.com/watch?v=ZhiSGADmj4c>

Inevery Crea (2014b). Tertulia de profesores de la Comunidad de Madrid sobre la integración de las TIC en la escuela. Recuperado de <http://ineverycrea.net/comunidad/ineverycrea/recurso/si-o-no-a-introducir-el-aprendizaje-de-la-programa/a352ea64-55b4-43d2-bd64-84136766228f>

Inevery Crea (2014c). No podemos dejar solo al docente en el aprendizaje de la programación. Entrevista con Antolín García de la asociación *Con Más Futuro*. Recuperado de <http://ineverycrea.net/comunidad/ineverycrea/recurso/no-podemos-dejar-solo-al-docente-en-el-aprendizaje/3e93253e-e2e4-4196-9afc-120f05740ba4>

Ley Orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa (LOMCE).

Monografías (2015). Teoría cognitiva de Piaget. Recuperado de <http://www.monografias.com/trabajos76/teoria-cognitiva-piaget/teoria-cognitiva-piaget2.shtml#lateoradeb>

mSchools (2012). Iniciativa mixta público-privada en Cataluña con la fundación Mòvil World para promover la incorporación en las aulas de la tecnología móvil. Recuperado de <http://www.xtec.cat/web/projectes/mschools>

Opening Up Education Initiative (2014). Iniciativa de la Agenda digital Europea para promover la innovación en el aprendizaje. Recuperado de <http://www.openeducationeuropa.eu/es/initiative>

PRIMO (2013). Vídeo ejemplo del Proyecto PRIMO. Robots educativos programables mediante la manipulación física de fichas. Recuperado de [https://www.youtube.com/watch?v=WOsE1j\\_kqaY](https://www.youtube.com/watch?v=WOsE1j_kqaY)

Programamos (2015a). Web de la Asociación Programamos. Recuperado de <http://programamos.es/>

Programamos (2015b). Posición de la Asociación Programamos sobre el pensamiento computacional. Recuperado de <http://programamos.es/que-es-el-pensamiento-computacional/>

RAE (2012a). Diccionario de la lengua española, 22ª edición. Entrada correspondiente al término ‘Informática’. Recuperado de <http://lema.rae.es/drae/?val=inform%C3%A1tica>

RAE (2012b). Diccionario de la lengua española, 22ª edición. Entrada correspondiente al término ‘Programa’. Recuperado de <http://lema.rae.es/drae/?val=programa>

Real Decreto 126/2014, de 28 de febrero, por el que se establece el currículo básico de la Educación Primaria. (desarrolla la LOMCE)

Real Decreto 1631/2006, de 29 de diciembre, por el que se establecen las enseñanzas mínimas correspondientes a la Educación Secundaria Obligatoria. (desarrolla la LOE)

Recomendación 2006/962/EC del Parlamento Europeo y del Consejo. Recuperado de [http://www.mcu.es/cine/docs/Novedades/Recomendacion\\_Parlamento\\_Europeo\\_Consejo\\_Aprendizaje\\_permanente.pdf](http://www.mcu.es/cine/docs/Novedades/Recomendacion_Parlamento_Europeo_Consejo_Aprendizaje_permanente.pdf)

Rodríguez, G. y otros (1996). *Metodología de la investigación cualitativa*. Málaga: Aljibe.

Scratch (2015a). Plataforma online del lenguaje de programación Scratch. Recuperado de <https://scratch.mit.edu/>

Scratch (2015b). Web para descarga lenguaje Scratch, versión 2. Recuperado de <https://scratch.mit.edu/scratch2download/>

StarLogo TNG (2008). Web de acceso a las distintas versiones de Star Logo. Recuperado de <http://education.mit.edu/starlogo/>

TED (2012a). Conferencia de Clare Sutcliffe, co-fundadora del Proyecto CodeClub. “Hacking the future”. Recuperado de [https://www.youtube.com/watch?v=ng7sf2\\_peFg&feature=youtu.be](https://www.youtube.com/watch?v=ng7sf2_peFg&feature=youtu.be)

TED (2012b). Conferencia de Mitch Resnick, especialista en educación y tecnología del MIT. “Let’s teach kids to code”. Recuperado de [http://www.ted.com/talks/mitch\\_resnick\\_let\\_s\\_teach\\_kids\\_to\\_code](http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code)

The Guardian (2012). Manifiesto por la enseñanza de ciencias de la computación en el siglo 21. Carta abierta del profesor John Naughton de la Open University al secretario de estado de educación del Reino Unido. Recuperado de <http://www.theguardian.com/education/2012/mar/31/manifiesto-teaching-ict-education-minister>

TVE REDES (2011). Entrevista a Ken Robinson titulada ‘El sistema educativo es anacrónico’. Recuperado de <http://www.rtve.es/television/20110304/redes-sistema-educativo-anacronico/413516.shtml>

TVE REDES (2012). Entrevista a Howard Gardner titulada ‘De las inteligencias múltiples a la educación personalizada’. Recuperado de <http://www.rtve.es/television/20111209/inteligencias-multiples-educacion-personalizada/480968.shtml>

Uría Menéndez (2010). Foro de Actualidad Jurídica 27-2010. Documento sobre la Agenda Digital Europea. Recuperado de <http://www.uria.com/documentos/publicaciones/2759/documento/foro01.pdf?id=2546>

Wing, J. (2006). ‘Computational thinking’, Communications of the ACM. Recuperado de <https://www.cs.cmu.edu/~CompThink/papers/Wing06.pdf>

## 5.2. Bibliografía recomendada

Gardner, H. (1998). *Inteligencias múltiples. La teoría en la práctica*, Barcelona, Paidós.

ISURI ARTE (2010). Guía didáctica de Scratch para profesores. Recuperado de <http://es.scribd.com/doc/143370634/Scratch-Guia-Didactica-para-Profesores>

López García, J.C. (2013). Guía de referencia de Scratch 2.0. Eduteka. Recuperado de <http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf>

TED (2006). Conferencia de Ken Robinson. ‘Las escuelas matan la creatividad’. Recuperado de <https://www.youtube.com/watch?v=nPB-41q97zg>