

**CÓDIGO 21**  
TECNOLOGÍAS CREATIVAS



# Programar para aprender: orientaciones para el profesorado de Primaria



Gobierno de Navarra  
Departamento de Educación

# Programar para aprender: orientaciones para el profesorado de Primaria

1. -¿Por qué programar software en Primaria y Secundaria?
  - 1.1.- Leer y escribir en el s.XXI
  - 1.2.- La competencia digital
  - 1.3.- Creatividad e interacción: Scratch (by MIT)
- 2.- ¿Qué dice el currículo?
  - 2.1.- 4º y 5º de Primaria
    - 4º de Primaria
    - 5º de Primaria
  - 2.2.- Proyectos de centro y transversalidad
- 3.- Recomendaciones de trabajo
- 4.- Recursos para el profesorado
  - 4.1.- Sesiones de clase listas para usar en Scratch
    - 4.1.1.- Veinte sesiones de 60 minutos
    - 4.1.2.- Siete unidades graduadas
  - 4.2.- Sesiones de clase listas para usar: no tengo ordenadores
  - 4.3.- Otros recursos
    - 4.3.1.- Para empezar: tutoriales, comunidad, recursos
    - 4.3.2.- Proyectos y experiencias
    - 4.3.3.- Formación
    - 4.3.4.- Actividades específicas para el currículo de Matemáticas
    - 4.3.5.- Otras webs y proyectos que debes conocer
    - 4.3.6.- Si te va la robótica educativa

Los sistemas educativos de nuestro entorno se encuentran en la necesidad de efectuar cambios sustanciales para adaptar los aprendizajes a los modelos de competencias propios del siglo XXI. Hay un acuerdo bastante extenso y documentado en que la sociedad del conocimiento requiere de profesionales con unos perfiles distintos, que se formen y se capaciten en sistemas educativos adecuados.

Así, desde el ámbito europeo, el Plan de Acción *Apertura a la Educación (Opening up education)* elaborado por la Comisión Europea, considera que “es el momento oportuno para estimular métodos educativos y formas de aprendizaje de gran calidad e innovadores utilizando las nuevas tecnologías y los contenidos digitales... aumentando la motivación de los alumnos y la eficiencia de los procesos de aprendizaje”. (Cfr. *Dictamen del Comité de las Regiones. Apertura de la Educación. 2014/C. 126/6*). Además, la Agenda Digital Europea ha querido dar la máxima prioridad a esta línea de trabajo incluyendo, dentro de los siete pilares de la Agenda, uno específico dirigido a “Mejorar la alfabetización, las destrezas y la inclusión digitales” (VI), y líneas de trabajo convergentes en otros pilares.

Por otra parte, las previsiones apuntan a que en los próximos años toda Europa va a precisar personal formado en estas áreas, con lo que ello supone para el crecimiento general basado en el conocimiento (Cfr. *Ibíd.*).

En los primeros niveles formativos es preciso renovar las asignaturas científicas y tecnológicas: sus contenidos, sus métodos y su potencial motivador para el estudio y la formación académica y humana, así como abrirlas a la indagación sistemática, la curiosidad, la creatividad, el trabajo en equipo, el afán de conocimiento riguroso y otras habilidades y competencias de gran importancia para una formación abierta a un futuro enmarcado en la sociedad del conocimiento.

Por ello, el **Departamento de Educación del Gobierno de Navarra** promovió un *Plan de Innovación Educativa en Ciencia y Tecnología* (<http://goo.gl/aVBwGA>) que impulsase el talento en el alumnado en dichos ámbitos y en todas las etapas mediante una renovación metodológica, temática y

organizativa del aprendizaje, dirigida a la enseñanza de la Ciencia y la Tecnología en los centros escolares de Navarra.

Fruto de ese Plan, se inició desde el verano de 2012 la colaboración con la Universidad Pública de Navarra (UPNA) y el Planetario de Pamplona, en forma de cursos de formación a profesorado y del establecimiento de un centro de recursos y una red de centros que trabajan en robótica educativa. La experiencia en esta red de centros nos hizo conscientes de que era posible y deseable ampliar las posibilidades temáticas al ámbito de la programación de software unida al desarrollo de múltiples competencias, no estrictamente científicas o tecnológicas. De esta forma, los nuevos currículos han introducido en Navarra las destrezas de programación de software en Primaria, y lo harán también en Secundaria. La información centralizada de todo este proyecto puede accederse desde la web de Código 21: <http://codigo21.educacion.navarra.es>

Quiero agradecer a todo este profesorado, que ha acogido de forma tan positiva la iniciativa, el esfuerzo que está haciendo en formación y experimentación. En el contexto tan complicado que nos ha tocado compartir, este esfuerzo es aún más valioso y más valorado.

Deseo, para concluir, que estas orientaciones que presentamos, y el resto de iniciativas con las que esperamos apoyar los esfuerzos del profesorado, sean de verdad instrumentos útiles en el camino que comenzamos.

D. José Iribas Sánchez de Boado  
Consejero de Educación  
Gobierno de Navarra

# 1. -¿Por qué programar software en Primaria y Secundaria?

Antes de empezar a trabajar con estas orientaciones y con los materiales online que ponemos a tu disposición a través de la web de Código 21 (<http://codigo21.educacion.navarra.es>), queremos explicar bien por qué es importante que nuestro alumnado pueda programar código. Por lo tanto, lee bien este apartado y consulta los enlaces que te proponemos.

## 1.1.- Leer y escribir en el s.XXI

En una famosa conferencia en TEDx<sup>1</sup> el profesor Mitch Resnick, responsable del Lifelong Kindergarten Group<sup>2</sup> del MIT MediaLab, afirmaba que la generación llamada “nativos digitales” era sobre todo una generación de consumidores de tecnología, con mucha familiaridad en la interacción con los medios digitales pero con escasa capacidad de expresión en estos medios. El mensaje que lanzaba era claro, utilizando un símil con la lectura y escritura: nuestros jóvenes pueden leer, pero no escribir con las TIC.

Esta idea era la que subyacía en el desarrollo que su grupo de investigación había realizado: un entorno gráfico donde, desde edades muy tempranas, los chicos y chicas pudieran expresarse con los códigos que están acostumbrados a consumir. Este entorno, del que luego hablaremos, es Scratch. Para Mitch Resnick y su grupo habría que lograr en nuestro alumnado una capacidad de expresión en los medios digitales similar a la que consiguen con la escritura. No para obtener al final de sus años de educación “escritores profesionales”, sino para permitirles utilizar esa *nueva forma de escritura* para aprender conceptos y competencias propias del siglo XXI, igual que utilizan la escritura digamos convencional para aprender en la escuela.

Es en este sentido en el que, desde el Departamento de Educación del Gobierno de Navarra, se pretende también orientar la introducción de contenidos básicos sobre desarrollo de software en el sistema educativo. Buscamos que sea una destreza transversal, más allá de la informática, una herramienta de expresión y producción de nuestro alumnado a través de todo tipo de proyectos: historias, juegos, tutoriales de todo tipo, actividades que den cuerpo a conceptos matemáticos o físicos, actividades que conecten el mundo real y el virtual. Queremos ofrecer contextos significativos y motivadores de aplicación y reflexión sobre la ciencia y tecnología como se está haciendo ya en muchos lugares a través de Scratch y propuestas similares<sup>3</sup>.

También aspiramos a que las alumnas y los alumnos que hoy están formándose en Navarra puedan comprender en profundidad el entorno digital en el que viven y puedan ser personas capaces de contribuir a ese entorno digital con madurez y sentido crítico.

Como venimos explicando, una primera respuesta a la pregunta que sirve de título a este apartado sería que aprendemos programación para aprender a expresarnos en los lenguajes propios de nuestro siglo y a interpretarlos con sentido crítico pero, además, existe evidencia científica que demuestra -en grupos que han aprendido a programar desde Infantil- la mejora de resultados en pruebas de matemáticas, razonamiento y resolución de problemas, el impacto positivo en la creatividad y respuesta emocional de niños con dificultades de aprendizaje, así como en el desarrollo de las habilidades cognitivas y socio-emocionales. También se ha demostrado que el alumnado que aprende a programar en edades tempranas tiene menos estereotipos de género en relación a las carreras STEM -Ciencias, Tecnología, Ingeniería y Matemáticas- y menos reticencias para continuar sus estudios y profesiones en estas disciplinas.

Por lo tanto, queremos programar para aprender, para expresarnos, para comprender mejor los contenidos científicos y tecnológicos, para perder el miedo a crear y compartir, y para que todos tengamos las mismas oportunidades de acceder a una alfabetización propia del siglo XXI.

---

<sup>1</sup> <http://goo.gl/yplmoJ>. El vídeo está en inglés pero pueden activarse los subtítulos en muchos idiomas

<sup>2</sup> <http://ilk.media.mit.edu/>

<sup>3</sup> La comunidad de usuarios de Scratch ha publicado y compartido más de 6.000.000 de proyectos. Puede accederse a sus estadísticas de uso en <http://scratch.mit.edu/statistics/>

## 1.2.- La competencia digital

Además de todo lo que venimos explicando, desde la mirada de los proyectos de programación, en Europa se ha producido un salto muy importante en la definición de lo que se entiende como competencia digital.

La tradición escolar en la enseñanza de las TIC venía recogiendo fundamentalmente el aprendizaje de herramientas específicas de software (procesadores, hojas de cálculo...), con lo que se fomentaban las destrezas operacionales. Sin embargo, en el contexto de una sociedad del conocimiento, la Comisión Europea afirma que es necesario fomentar habilidades, conocimientos y actitudes diversas: ser crítico y reflexivo con el uso que hacemos de las tecnologías, ser consciente de sus posibilidades y riesgos, ser capaces de evolucionar con ellas, ser capaces de crear con ellas...

La Comisión Europea ha desarrollado, por ello, un Marco de Competencia Digital (<http://goo.gl/O8BaSZ>) que recoge estas nuevas necesidades y que pretende ser, como lo fue el Marco Común Europeo de Referencia para las Lenguas realizado por el Consejo de Europa (<http://goo.gl/fCC3YE>), un documento que permita aunar criterios, esfuerzos, reconocimientos y evaluaciones en el campo digital.

El documento de la Comisión ha sido utilizado por el INTEF y las CCAA para realizar una propuesta de Competencia Digital Docente, que puede consultarse en esta dirección: <http://goo.gl/AwxIRd>. Ambos documentos comparten la división en cinco dimensiones y las operaciones básicas en cada una de ellas. De forma muy resumida, estas son las definiciones de cada dimensión:

1. **Información:** identificar, localizar, recuperar, almacenar, organizar y analizar la información digital, evaluando su finalidad y relevancia.
2. **Comunicación:** comunicar en entornos digitales, compartir recursos a través de herramientas en línea, conectar y colaborar con otros a través de herramientas digitales, interactuar y participar en comunidades y redes; conciencia intercultural.
3. **Creación de contenido:** Crear y editar contenidos nuevos (textos, imágenes, videos...), integrar y reelaborar conocimientos y contenidos previos, realizar producciones artísticas, contenidos multimedia y programación informática, saber aplicar los derechos de propiedad intelectual y las licencias de uso.
4. **Seguridad:** protección personal, protección de datos, protección de la identidad digital, uso de seguridad, uso seguro y sostenible.
5. **Resolución de problemas:** identificar necesidades y recursos digitales, tomar decisiones a la hora de elegir la herramienta digital apropiada, acorde a la finalidad o necesidad, resolver problemas conceptuales a través de medios digitales, resolver problemas técnicos, uso creativo de la tecnología, actualizar la competencia propia y la de otros.

Como iremos viendo en los siguientes apartados, la introducción de destrezas de programación y la interacción con contenidos multimedia a través de herramientas como Scratch, son potentes líneas de trabajo para desarrollar varias de las dimensiones de la Competencia Digital.

En línea con los grupos de investigación del MIT (<http://ilk.media.mit.edu/>) y Harvard (<http://scratched.gse.harvard.edu/>) queremos fomentar una programación creativa que conecte los intereses y gustos del alumnado con proyectos en los que pueden ser protagonistas y que se expresan en los mismos lenguajes que están “acostumbrados a consumir”. Por encima de la maestría técnica -insistimos en que no queremos formar programadores profesionales- pretendemos que, por medio de la programación con herramientas como Scratch y en contextos significativos, puedan trabajar la creatividad, la expresión, el trabajo colaborativo y en equipo, la comunicación, la modelización y resolución de problemas, el proceso que va desde empezar con una idea hasta llegar a desarrollar un proyecto, la experimentación con nuevas ideas, la perseverancia cuando las cosas no funcionan bien, sin frustrarse o aprendiendo a tolerar la frustración.

### **1.3.- Creatividad e interacción: Scratch (by MIT)**

Hay muchas herramientas que pueden utilizarse para fomentar este enfoque de la programación informática que busca desarrollar de manera amplia las competencias digitales y otras competencias que tienen que ver con la creatividad, la perseverancia o la resolución de problemas. Desde el Departamento de Educación del Gobierno de Navarra hemos optado por recomendar para los cursos de Primaria y los primeros de Secundaria Scratch, puesto que es un lenguaje de programación gratuito (al que puede accederse desde <http://scratch.mit.edu>) y muy orientado a la creación de historias interactivas, de juegos o animaciones, en un sencillo entorno gráfico de bloques. Desde la versión 2.0 incluye bloques que permiten interactuar con la realidad a través de cámaras web, por ejemplo.

El programa es una herencia de Logo al que se le han añadido parte de las posibilidades que ofrecían las herramientas de autor multimedia de hace algunos años. El resultado es una herramienta interactiva que permite programar de manera visual y con feedback inmediato, y que también permite “contar historias” mezclando imagen, audio, texto e interacción por medio de teclado, ratón u otros dispositivos.

Además, cuenta con una comunidad de usuarios que no hace más que crecer desde 2007 (4.349.094 en el momento de hacer esta guía), y que ha publicado y compartido, como ya se ha dicho más arriba, cerca de 7.000.000 de proyectos (<http://scratch.mit.edu/statistics/>). El hecho de tener una comunidad tan grande y activa, con tanto desarrollo publicado, así como a dos de las universidades más importantes del mundo trabajando en el proyecto (MIT y Harvard) hace que podamos acceder con mucha facilidad a ejemplos ya realizados, a tutoriales y a guías, como veremos.

Por otra parte, basados en la lógica de Scratch, hay otros entornos como BYOB-Snap (<http://snap.berkeley.edu/>), Enchanting (<http://goo.gl/XvDHgP>) que también son ampliamente utilizados y, en algunos casos, conectan la programación en pantalla con proyectos de robótica educativa.

En definitiva, utilizando Scratch y los programas de su familia, no vamos a ser los primeros ni los únicos, y tampoco vamos a estar solos. Incluso en Navarra: en el momento de redactar esta guía hay 30 centros trabajando en robótica educativa y utilizando alguno de estos lenguajes y, si sumamos los centros que se encuentran representados en la formación online que ha comenzado en el mes de noviembre, esperamos que al acabar el curso se hayan realizado proyectos sencillos de programación en un total de 90 centros. Puedes conocer en qué centros hay personas formándose o están trabajando en programación con Scratch o en robótica educativa en este mapa: <http://goo.gl/MsekH1>.

## 2.- ¿Qué dice el currículo?

Los nuevos currículos en Navarra quieren impulsar el trabajo de estas destrezas desde el enfoque que viene explicándose en la presente guía. No detalla cómo ha de hacerse, ni el tipo concreto de proyecto. El texto legal recogido en el **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 (BON número 174, de 5 de septiembre de 2014) <http://goo.gl/5iVWrk> ofrece unos estándares de evaluación dentro del Currículo de Matemáticas de 4º y 5º de Primaria que descomponen las tareas que deberán dominarse pero no obliga en la manera de alcanzarlo. Esto deja abiertas muchas posibilidades de trabajo, como veremos.

Por otra parte, deja abierta la posibilidad de establecer itinerarios más intensivos en este área de la programación informática gracias al punto 7 del artículo 7.

Los borradores de los nuevos currículos de Secundaria también recogen esta transversalidad, así como la utilización de herramientas tecnológicas como Scratch y otras en el currículo de Matemáticas de toda la ESO así como en asignaturas más específicamente informáticas.

### 2.1.- 4º y 5º de Primaria

Dentro de 4º se ha introducido un contenido del Bloque 1. Procesos, métodos y actitudes en matemáticas, en el sentido de ampliar el uso de las Tecnologías de la Información y la Comunicación que deben hacerse en ese curso. La ampliación del contenido recoge los lenguajes y herramientas de programación, en general, aunque se recomienda el trabajo con Scratch, como hemos mencionado en el apartado **¿Por qué programar software en Primaria y Secundaria?**

Este contenido conlleva la necesidad de añadir un criterio de evaluación y un descriptor, donde se trata de establecer el nivel que el alumnado debería alcanzar en 4º de Primaria. Las mismas operaciones se han realizado en el Currículo de 5º de Primaria.

Las siguientes tablas muestran los requerimientos:

#### 4º de Primaria:

##### BLOQUE 1. PROCESOS, MÉTODOS Y ACTITUDES EN MATEMÁTICAS

Contenidos	Criterios de evaluación	Estándares de aprendizaje evaluables
“Integración de las tecnologías de la información y la comunicación, así como de los lenguajes y herramientas de programación en el proceso de aprendizaje.”	14. Utilizar herramientas y lenguajes de programación para modelizar y resolver problemas.	14.1. De manera guiada, realiza un proyecto de programación donde hay que describir el algoritmo, descomponer el problema en partes más pequeñas y codificarlo con un lenguaje de programación visual formal (como Scratch).

¿Qué implicaciones tienen estos contenidos? Scratch se puede ver bien como una herramienta que utiliza el docente para crear materiales y trabajarlos interactivamente en clase, o bien como una herramienta que pueden utilizar los alumnos para crear materiales (utilizando imagen, audio y programación interactiva).

Si queremos enfocarnos en la resolución de problemas, necesitamos primero “modelizarlos en el entorno Scratch” y a partir de ahí ver cómo resolverlos (diseñando el escenario del problema, el algoritmo de resolución y el interfaz de usuario, es decir la visualización de resultados y la forma de interacción).

## PROGRAMAR PARA APRENDER: ORIENTACIONES PARA EL PROFESORADO DE PRIMARIA

---

Si estamos en Matemáticas, debemos estudiar el currículo y decidir qué partes pueden adaptarse mejor a este planteamiento.

Debemos tener en cuenta que para que los estudiantes sean capaces de trabajar con Scratch, necesitaremos en algún momento un periodo inicial de formación (más técnica) de los alumnos en relación al propio entorno de programación. Según vayan adquiriendo destrezas en cuanto al uso de Scratch y de su programación, serán capaces de utilizarlo como herramienta de modelización y resolución de problemas.

Por todo esto, un planteamiento posible es que en 4º de Primaria la actividad sea más dirigida, el profesor/a crea materiales, los muestra para reforzar conceptos y “de paso” va enseñando a utilizar Scratch (motivando y animando a utilizarlo). Mostrar proyectos que han realizado otros grupos de edades similares y dejar un tiempo para que jueguen y experimenten con ellos, accediendo por ejemplo a la web de proyectos de Scratch y recorrer algunos de sus “Estudios” (colecciones de proyectos) <http://goo.gl/EbHwAa>, puede ser también una fuente de motivación.

### 5º de Primaria:

#### BLOQUE 1. PROCESOS, MÉTODOS Y ACTITUDES EN MATEMÁTICAS

Contenidos	Criterios de evaluación	Estándares de aprendizaje evaluables
“Integración de las tecnologías de la información y la comunicación, así como de los lenguajes y herramientas de programación en el proceso de aprendizaje.”	14. Utilizar herramientas y lenguajes de programación para modelizar y resolver problemas.	14.1. Diseña y realiza proyectos de programación donde se utilizan secuencias de comandos, bucles, condicionales, variables, así como distintas formas de entrada y salida de datos (interacción con el ordenador).

Siguiendo con lo planteado en el punto anterior, en 5º de Primaria se pueden trabajar proyectos que incluyan problemas relacionados con el currículo de Matemáticas. Por un lado, el objetivo es que sean capaces de trabajar de esta manera, con esta herramienta y con este entorno y lenguaje de programación. Por otro lado, se trata también de empezar a tener un contacto con los aspectos básicos de programación como pueden ser secuencias de comandos, bucles, condicionales, variables, así como distintas formas de entrada y salida de datos (interacción con el ordenador).

Tal y como se ha comentado anteriormente, no se trata de formar a programadores, pero sí se trata (a medio plazo) de que nuestros estudiantes cuando acaben Primaria sean capaces de modelizar y resolver problemas con este tipo de herramientas, y a la vez sepan los aspectos básicos de programación.

Los creadores de Scratch lo definen como una herramienta “*low-floor/high-ceiling/wide-walls*”, literalmente en inglés quiere decir “con suelo bajo, techo alto y muros anchos”, y metafóricamente quiere decir que es una herramienta que se puede empezar a utilizar a edades tempranas, que da cabida a muchos tipos de actividades, y que nunca acabamos de crear y aprender con ella. Por ello entendemos que esta metodología de trabajo con este tipo de herramienta nos puede permitir a medio plazo tener un itinerario integrado en el currículo de Primaria y que cumpla los objetivos de trabajar currículo, competencias y al mismo tiempo dotar a nuestros estudiantes de unos conocimientos básicos de programación, pensando sobre todo en desarrollar el pensamiento formal.



## **2.2.- Proyectos de centro y transversalidad**

El Decreto también ha previsto en su Artículo 7. Elementos transversales, punto 7, que “los centros educativos establecerán a través de proyectos acciones destinadas a la adquisición y mejora de las destrezas básicas de programación informática.”

A la vista de este artículo, los centros podrían comenzar itinerarios desde cualquier curso de Primaria, o incidir en programación orientada a proyectos de robótica educativa (con Beebots en primer ciclo de primaria y NXT a partir del segundo ciclo), por ejemplo. Será cada equipo docente, en función de su contexto, formación y plan estratégico, el que decida cómo puede acometer mejor el proyecto.

Por otra parte, desde el punto de vista horizontal y transversal pueden acometerse proyectos interdisciplinarios de creación de historias interactivas, o videojuegos, o felicitaciones interactivas, por ejemplo, donde las diferentes asignaturas aportan una parte de los conocimientos y habilidades necesarios para abordarlos: en Lengua se puede preparar el guión, los diálogos...; en Plástica pueden preparar los fondos, los disfraces de los personajes; en Música pueden preparar la ambientación musical; en Matemáticas pueden programar la interacción...

Para mantenerte al día sobre los proyectos que van realizándose en Navarra suscríbete a la web <http://codigo21.educacion.navarra.es>

## 3.- Recomendaciones de trabajo

Antes de que comiences a planificar tus clases, queremos ofrecerte algunas indicaciones generales.

En primer lugar, todo lo que esta guía o los recursos que ponemos a tu alcance proponen son solamente propuestas generales cuyo único interés es facilitar la puesta en marcha de los proyectos de aula. No significa que sea esta la única manera de hacerlo y, por tanto, hay total libertad para trabajar con otros lenguajes o entornos.

Por otra parte, somos conscientes de que hay que ir poco a poco y por ello pensamos que no es necesario que todo el equipo docente tenga que formarse en un inicio. Quizás sea suficiente con que una persona lo haga y realice los proyectos en las clases del mismo nivel, o puedan agruparse clases en un determinado momento.

Tal y como hemos comentado previamente el objetivo es trabajar a tres niveles:

1. Programación: secuencias de comandos, bucles, condicionales, variables, así como distintas formas de entrada y salida de datos (interacción con el ordenador).
2. Temario de Matemáticas de Primaria.
3. Competencias transversales de Primaria.

Las siguientes guías disponibles en castellano e inglés muestran cómo se pueden trabajar los puntos 1 y 3:

- <http://goo.gl/avIP1e>
- <http://goo.gl/o0ANlz>
- <http://goo.gl/KZr3z8>

El punto 2, las actividades propias del temario de Matemáticas, deben coordinarlo y decidirlo los equipos de profesores, con apoyo de los equipos directivos. Es posible que no sea necesario que todos los tutores tengan que formarse. Con una adecuada coordinación y apoyo podría algún docente especializarse en este tipo de proyectos y actividades y realizarlas en los diferentes grupos. En cualquier caso, sí es conveniente contar con un plan de trabajo de centro, con actividades secuenciadas en el tiempo (lo ideal sería para toda la Primaria, aunque inicialmente se centrará en 4º y 5º). Es deseable, así mismo, algún tipo de **observación/monitorización/evaluación** de las actividades.

A continuación se exponen las indicaciones más interesantes, a partir de la experiencia internacional y propia:

- **Poner énfasis en el desarrollo de la creatividad.** Aprovechar la potencia multimedia del entorno Scratch para plantear retos y proyectos que exploten aspectos creativos, desde múltiples inteligencias y estilos de aprendizaje.
- **Adaptarse al contexto.** Para lograr un aprendizaje significativo (de currículo, de desarrollo de competencia, de técnica de programación, etc.) es importante crear situaciones didácticas adaptadas al contexto del alumnado. Es importante que se identifiquen con la actividad y sean quienes van descubriendo sus necesidades para avanzar en ella (concepto de programación que no conocen, contenidos que no comprenden, falta de dominio de alguna de las competencias que se les demanda).
- **Trabajar la dimensión oral y el trabajo en grupo.** La explicación de sus propios procesos de aprendizaje y la gestión de equipos de trabajo son valores añadidos que enriquecen unos aspectos transversales que a menudo se quedan desatendidos.

- **Promover la reflexión y despertar el sentido crítico sobre Internet y productos de software como videojuegos.** Es vital que los alumnos y alumnas sean conscientes de que es necesario contrastar la información que encuentran por Internet...esto seguramente no lo verán claro hasta que sean más mayores pero se les puede pedir que de manera regular consensúen lo que encuentran en Internet, por ejemplo que busquen siempre al menos 3 referencias para ver si dicen lo mismo, etc... En cuanto a los videojuegos, pueden tomar conciencia de la forma en que han sido contruidos, de los recursos que utilizan para mantenernos activos en el juego, y reflexionar sobre los juegos que consumen.
- **Remezclar, reutilizar: es lo que a veces hacemos también los adultos.** La ingeniería inversa nos permite comprender cómo funcionan mecanismos complejos como pueda ser un programa informático y, una vez asimilado, ampliar y aplicar en diferentes proyectos o extender uno existente. Cuando abordemos una actividad de este tipo habrá que tratar de forma explícita el respeto a los derechos de autor, a los que se citará de forma adecuada, y el conocimiento de las diferentes licencias de uso.
- **Trabajar sin ordenadores: Unplugged (desenchufado).** Un programa es la codificación de un algoritmo, que a su vez es la secuencia de instrucciones o reglas bien definidas que resuelven un problema planteado (el algoritmo) Sin ordenadores podemos simular el movimiento de un robot en un escenario determinado donde hay que seguir un itinerario de inicio a fin, escribimos el programa con flechas o con cualquier otro recurso y un alumno puede interpretar ese programa y hacer exactamente lo que haría el robot para ver si la solución planteada es correcta. De esta manera, programamos en papel y simulamos lo que haría el robot. Luego podríamos codificarlo en un lenguaje y utilizar un robot, virtual o físico. En el apartado Recursos enlazamos a una guía con actividades para contextos en los que no hay ordenadores disponibles.
- **Buscar audiencia para los proyectos de los alumnos.** Además de ser capaces de explicar sus creaciones entre ellos o a los profesores, es importante que de vez en cuando lo cuenten a terceros. Por ejemplo a los padres o a los otros compañeros. Eventos como el Scratch Day (<http://day.scratch.mit.edu/>) o la Code Week (<http://codeweek.eu/>) permiten compartir los logros alcanzados y darle más significado si cabe al aprendizaje por medio de la divulgación científica. También puede ser interesante abrir un "Estudio" en la web de Scratch donde pongamos a disposición de otros profesores o alumnos nuestros proyectos y observar si se visitan o no, si se remezclan, si sirven de inspiración a otros. Podemos ver ejemplos de estudios en esta dirección: <http://goo.gl/fUAcaq>
- **Tener en cuenta la inclusión de género.** A veces todo lo relacionado con informática, robótica, o tecnología en general atrae más a los chicos que a las chicas... Esto se refleja, por ejemplo, en que en Grados de ingeniería informática o industrial encontramos más hombres que mujeres; sin embargo, queremos favorecer que ambos géneros trabajen de la misma manera y que colaboren, ya que los dos puntos de vista, naturalmente diferentes, se complementan y favorecen la creatividad además de otras competencias sociales necesarias para nuestra sociedad. Está demostrado que una formación de este tipo que llegue a todos, a chicos y a chicas, contribuye de forma decisiva a destruir estereotipos de género en cuanto a ocio, profesiones o gustos.
- **Aprovechar las posibilidades multilingües.** Tanto la herramienta Scratch como las actividades se pueden trabajar en diferentes idiomas; la forma de trabajo no cambia, pero sí cambia el idioma vehicular. Es una forma de aprender idiomas a través de los contenidos que los centros, en función de sus diversos proyectos lingüísticos, podrán considerar.

## 4.- Recursos para el profesorado

El objetivo de esta sección es mostrar algunas puertas de entrada imprescindibles para el campo de la programación en Primaria y, en concreto, para el programa Scratch. Aunque hay mucho más en Internet, hemos preferido referenciar solamente lo esencial para que sea más fácil comenzar. Buscando en Internet podrás encontrar muchos más materiales, recursos, videotutoriales e ideas...

### 4.1.- Sesiones de clase listas para usar en Scratch

El equipo que lidera Scratch, ha desarrollado dos guías con sesiones listas para usar. Estas guías, junto a los materiales que enlazamos desde la web del proyecto <http://codigo21.educacion.navarra.es> pueden proporcionarte todo lo que necesitas para comenzar a trabajar con Scratch en tu aula:

#### 4.1.1.- Veinte sesiones de 60 minutos

El enlace que ofrecemos a continuación lleva a un pdf con una introducción a la computación creativa con Scratch, utilizando un enfoque basado en el diseño.

La guía está organizada en una serie de 20 sesiones de 60 minutos de duración, y detalla los planes de sesión, los materiales que habría que entregar, los proyectos y los vídeos que se mostrarían en clase.

Las 20 sesiones se organizan en los siguientes temas:

- introduction
- arts
- stories
- games
- final project

La guía está pensada para ser neutral en términos de curso y asignaturas, de manera que pueda adecuarse a múltiples contextos y necesidades.

Puede accederse al material en inglés aquí: <http://goo.gl/UK2xVy>

Y a la traducción en español, aquí: <http://goo.gl/LdeoMu>

#### 4.1.2.- Siete unidades graduadas

Esta guía es una colección de ideas, estrategias y actividades para una experiencia creativa de introducción a la programación mediante el lenguaje Scratch. Las actividades están diseñadas para apoyar la familiaridad y el aumento de la fluidez con la creatividad computacional y el pensamiento computacional.

En particular, las actividades estimulan la exploración de conceptos clave de pensamiento computacional (secuencia, loops, paralelismo, eventos, condicionales, operadores, datos) y de prácticas de pensamiento computacional (experimentación y la iteración, pruebas y depuración, la reutilización y la remezcla, la abstracción y la modularización).

Está organizada en 7 unidades didácticas perfectamente documentadas e incluso minutadas, listas para usar.

Puedes acceder desde aquí: <http://goo.gl/zVpr1o>

La versión original en inglés está aquí: <http://goo.gl/7t7NfS>

## **4.2.- Sesiones de clase listas para usar: no tengo ordenadores**

Hay muchos recursos en la web para hacer actividades de programación ¡sin ordenadores!. Uno de los más interesantes es: <http://csunplugged.org/>

En él podrás acceder a una completa guía del profesor, con numerosas actividades, traducida al español: <http://goo.gl/Or4kjf>

El proyecto cuenta, también, con un canal de vídeo: <https://www.youtube.com/user/csunplugged>

También puedes acceder a un ejemplo preparado para la Hora del Código, en Estados Unidos, disponible en versión española: <http://goo.gl/7AwvLc>

## **4.3.- Otros recursos**

### **4.3.1.- Para empezar: tutoriales, comunidad, recursos**

- Para empezar, es muy conveniente acudir al sitio oficial y a su sección de ayuda: <http://scratch.mit.edu/help/>
- Dentro de esa sección encontrarás, por ejemplo, videotutoriales: <http://scratch.mit.edu/help/videos/>
- El otro gran lugar de referencia para empezar es la comunidad de profesorado: <http://scratched.gse.harvard.edu/>

### **4.3.2.- Proyectos y experiencias**

- También aquí, la principal referencia es la propia web de Scratch: <http://scratch.mit.edu/explore/projects/featured/>
- Desde la web de Programamos.es podemos acceder a materiales interesantes: <http://programamos.es/materiales/> y también a experiencias en Primaria: <http://programamos.es/category/primaria/>

### **4.3.3.- Formación**

- En el aula virtual del PNTE (Departamento de Educación) se ofrece un curso online con 150 plazas: <http://goo.gl/3LOfPP>. Se está realizando la primera edición y se convocarán varias ediciones por curso, hasta que se cubra la demanda. Las convocatorias se publicarán en: <http://goo.gl/aFPwKy> y también las difundiremos a través de <http://goo.gl/VHyrWM>
- Materiales del curso de verano 2014, Departamento de Educación y UPNA (Universidad Pública de Navarra):
  - Antonio Ruiz: <http://goo.gl/B1ch6v>
  - Jesús Moreno y José Ignacio Huertas: <http://programamos.es/abpscratch/>

### **4.3.4.- Actividades específicas para el currículo de Matemáticas**

En la web Código 21 encontrarás unos cuantos recursos específicos, como: <http://goo.gl/nvpD1y>.

No obstante, queremos resaltar:

- Puedes acceder a ejemplos de uso de Scratch para cubrir el currículo de Matemáticas en este archivo de la profesora Miren Percáz: <http://goo.gl/mCmtWL>
- Para Primaria, el profesor Antonio Ruiz tiene estas propuestas: <http://goo.gl/DktGfQ>
- Otro ejemplo: <http://goo.gl/1idJxN>

#### 4.3.5.- Otras webs y proyectos que debes conocer

Para los más pequeños, de **Educación Infantil**, el equipo de Scratch ha desarrollado **Scratch Jr**:

- <http://www.scratchjr.org/>
- <http://youtu.be/mXbOMQ-OWWU>
- <http://programamos.es/category/scratch-jr/>

**Code.org** Es una iniciativa sin ánimo de lucro, que a su vez organiza en todo el mundo “La hora del código” <http://hourofcode.com/es> apoyada por numerosas instituciones y empresas tecnológicas <http://hourofcode.com/es/partners> con el objetivo de aumentar la alfabetización computacional entre los escolares de todo el mundo. Acoge materiales y propuestas desde Infantil hasta Secundaria.

Desde aquí, puede accederse a materiales de formación para alumnado de Primaria, por ejemplo, para su uso en familia o en clase, entre ellos:

- Curso 1 <http://studio.code.org/s/course1> : esta diseñado para permitir a los lectores tempranos crear programas de computadora que les ayuden a aprender a colaborar con otros, desarrollar habilidades de solución de problemas, y persistir en tareas difíciles. Para el final de este curso, los estudiantes crean su propio juego personalizado o historia que pueden compartir. Recomendado para primeros cursos de Primaria.
- Curso 2 <http://studio.code.org/s/course2> : está diseñado para estudiantes que pueden leer y no tienen experiencia previa en programación. En este curso los estudiantes crearán programas para resolver problemas y desarrollar juegos interactivos o historias que pueden compartir. Recomendado para primeros cursos de Primaria.
- Curso 3 <http://studio.code.org/s/course3> : está diseñado para estudiantes que hayan tomado el Curso 2. Los estudiantes profundizarán más en temas de programación introducidos en los cursos anteriores que dan soluciones flexibles a problemas más complejos. Para el final del curso, los estudiantes crean historias y juegos interactivos que pueden compartir con cualquiera. Recomendado a partir de 4º de Primaria.

**Girls who code:** Institución norteamericana sin ánimo de lucro que trata de cerrar la brecha de género que existe en los sectores de la tecnología y la ingeniería. <http://girlswhocode.com/>

**Waterbearlang:** Herramienta para facilitar el aprendizaje de la programación. No es tan adecuada para utilizar en Primaria, pero puede utilizarse en cursos superiores. <http://waterbearlang.com/>

**Codecademy:** Cursos de programación sobre varios lenguajes que se utilizan mucho en la actualidad: HTML5, Javascript, Python, Ruby, PHP...El sitio no está pensado para público escolar. <http://www.codecademy.com>

Otras herramientas para enseñar programación a niños:

- HopScotch: <http://www.gethopscotch.com/>
- Alice: <http://www.alice.org/>
- Tynker: <http://www.tynker.com/>
- Hackety: Hack: <http://hackety.com/>
- Kodable: <http://www.kodable.com/>
- Stencyl: <http://www.stencyl.com/>
- RoboMind: <http://www.robomind.net/en/index.html>
- Etoys: <http://www.squeakland.org/>

- Move the Turtle: <http://movetheturtle.com/>
- Cargo Bott: <http://twolivesleft.com/>
- Daisy the Dinosaur: <http://www.daisythedinosaur.com/>
- Para más mayores: <https://www.codehunt.com/> (juego de Microsoft para aprender a programar)
- Juego de mesa: Kickstarter Robot Turtles: <http://goo.gl/y9UcoK>

#### **4.3.6.- Si te va la robótica educativa**

- Las novedades en Navarra las encontrarás en <http://codigo21.educacion.navarra.es>
- Web de Planetario de Pamplona, que alberga el Centro de Recursos de Robótica Educativa:
  - Artículos con etiqueta robótica educativa: <http://pamplonetario.org/tag/robotica-educativa/>
  - Reserva de robots del Centro de Recursos de Robótica Educativa: <http://goo.gl/hVy1sN>
- LEGO: <http://goo.gl/h8wIEz>

Más allá de los robots de LEGO:

- Robots Play-i: <http://goo.gl/nUiRjz>
- Robots de madera Kibo: <http://goo.gl/41osJO>
- Otro tipo de robots: Moway: <http://goo.gl/Dhkh4e>
- Robot de Arduino: <http://goo.gl/hvPw64>

Toda la información actualizada la encontrarás en: <http://codigo21.educacion.navarra.es/>

