



**CODING FOR  
INCLUSION**

[www.codinc.fun](http://www.codinc.fun)



# COD1NC: METODOLOGÍA

Introducción.....	3
Objetivos de aprendizaje.....	6
Objetivos de aprendizaje para maestros y profesores.....	6
Objetivos de aprendizaje para estudiantes.....	7
Cuadro de objetivos de aprendizaje para estudiantes.....	11
Objetivos de aprendizaje para la educación primaria.....	12
Objetivos de aprendizaje para alumnos.....	12
Cuadro de objetivos de aprendizaje para los estudiantes.....	14
Marco teórico.....	15
Habilidades del siglo xxi.....	15
Pensamiento computacional.....	16
Pensamiento creativo.....	17
¿Por qué es necesario aprender a programar?.....	18
Principios pedagógicos básicos y facilitación.....	20
Cuando el proyecto despierta recelos.....	23
Fichas.....	25
Materiales.....	25
Apéndice de Referencias.....	25

## INTRODUCCIÓN

Esta metodología está dirigida al profesorado de educación secundaria y primaria y a los dinamizadores.

En las páginas que siguen se explica en qué se basan estos métodos pedagógicos que presentamos, y cómo se pueden aplicar a un grupo de estudiantes de secundaria de entre quince y dieciocho años para que se sientan implicados y puedan hacer una formación básica sobre programación informática, así como otras actividades pertenecientes a las áreas de la ciencia, la tecnología, la ingeniería, el arte y las matemáticas (STEAM, por sus siglas en inglés) con alumnos de primaria de entre ocho y diez años.

En el proyecto CODINC, nuestros formadores enseñan los principios de la programación a los estudiantes y el profesorado de secundaria en una serie de talleres en el aula cuya duración es de diez horas en total. Después de esta formación, los estudiantes deberán transmitir los conocimientos y las habilidades que han adquirido a un grupo de alumnos de primaria, acompañados por los maestros de estas escuelas y nuestros dinamizadores.

Por su parte, al profesorado de secundaria este proyecto le servirá para aumentar su capital científico, mejorar sus habilidades pedagógicas y transmitir lo aprendido sobre programación básica a los alumnos de primaria.

CODINC es un proyecto centrado en la promoción de la inclusión social, a través de la formación entre iguales, sobre programación informática y otras actividades de STEAM. Los estudiantes y alumnos a los que va dirigido el proyecto son jóvenes y niños de barrios con unos índices de exclusión social superiores a los del resto de la ciudad. En muchos países se publican datos que permiten identificar cuáles son las zonas y las escuelas más desfavorecidas y con un grado de exclusión más elevado.

CODINC busca ayudar a los estudiantes de entre quince y dieciocho años procedentes de entornos desfavorecidos a aumentar su autoestima, guiándoles a través del descubrimiento de su propio potencial de creación, innovación y acompañamiento. Nuestra propuesta para alcanzar este objetivo se vehicula a través de una formación en la que aprenden a programar y crear sus proyectos de STEAM y en la que, posteriormente, intercambian los roles y hacen ellos mismos de formadores de alumnos más pequeños. Esto es de gran utilidad para aquellos jóvenes que estén en un momento de definición personal, buscando su encaje e identidad en el mundo de los adultos. Es posible que los estudiantes de entornos más desfavorecidos experimenten que no encuentran su lugar y se sientan

carentes de la confianza y el apoyo de los adultos para descubrir su potencial creativo e innovador.

En CODINC hemos adaptado y ampliado la metodología inclusiva de Capital Digital, un proyecto diseñado y llevado a cabo por nuestro socio, MAKS VZW, en barrios de Bruselas (Bélgica) clasificados como muy desfavorecidos.

Capital Digital es un proyecto innovador basado en el aprendizaje entre iguales que tiene como objetivo el formar a jóvenes de entornos desfavorecidos en el mundo de la programación y en otras actividades de STEAM (ciencia, tecnología, ingeniería, arte y matemáticas), para que posteriormente ellos mismos hagan de dinamizadores con sus compañeros más pequeños. Con este proyecto se formaron jóvenes de entre quince y dieciocho años, procedentes de contextos desfavorables, para que pudieran enseñar programación a niños de entre diez y doce años. Estos jóvenes dinamizadores aprendieron a motivar a los más pequeños a conocer y poner en práctica tales actividades a través de juegos y dinámicas. Para los dinamizadores, este proyecto constituyó una primera experiencia laboral que les ayudó a ganar confianza, y al mismo tiempo a clarificar sus horizontes a la hora de elegir qué querían estudiar.

Asimismo, Capital Digital les ayudó a establecer una relación constructiva entre ellos y a disfrutar de su papel como educadores de los más pequeños. Por añadidura, a través del proyecto pudieron desarrollar capacidades tales como el pensamiento crítico, la creatividad, las habilidades digitales o la colaboración entre compañeros, además de aumentar su capital científico. Sin embargo, lo más importante es que la metodología del aprendizaje entre iguales que presenta Capital Digital tiene un fuerte potencial inclusivo y fomenta la educación en STEAM y la inclusión de jóvenes desfavorecidos, tanto dentro como fuera del aula.

Como decíamos, CODINC es una adaptación y ampliación de la metodología y el manual de formación de Capital Digital, en el sentido de que se aplica en otro contexto educativo, la educación formal en las escuelas, y del cual se está haciendo una prueba piloto en cinco países europeos: Bélgica, Chipre, Alemania, Italia y España. Los objetivos específicos del proyecto son los siguientes:

1. Profundizar y mejorar la capacidad del profesorado para fomentar la educación en las áreas STEAM entre jóvenes procedentes de contextos desfavorecidos siguiendo una metodología inclusiva basada en el aprendizaje entre iguales.
2. Conseguir que estos jóvenes se empoderen a través de la adquisición y el desarrollo de competencias informáticas y otras capacidades, como la colaboración entre compañeros, la resolución de problemas, la confianza en uno mismo y la creatividad, todo ello por medio de un programa de formación en programación basado en el aprendizaje entre iguales.
3. Impulsar el desarrollo de una comunidad europea de aprendizaje centrada en el lema «Codinc: coding for inclusion» (programación para la inclusión), compuesta de varios

actores y presente en diferentes sectores (formación y educación formal y no formal) que puedan mantener los resultados obtenidos en el proyecto y amplificar los beneficios.

En Europa existe la necesidad de desarrollar las competencias digitales de la ciudadanía, y en muchos países el sistema educativo todavía no está del todo adaptado al uso de las nuevas tecnologías. Y esta necesidad es aún mayor en los barrios con unos índices de exclusión social más elevados, donde justamente se pueden utilizar las nuevas tecnologías para impulsar la inclusión. Aunque en estas áreas los estudiantes de primaria y secundaria son usuarios activos de Internet y de los dispositivos móviles, el profesorado de las escuelas aún debe potenciar sus habilidades para poderles hacer un buen acompañamiento en el uso de las TIC. La programación informática en especial, y el STEAM de manera general, ofrecen nuevas vías para la educación basada en los intereses de los niños y los jóvenes, en el sentido de permitirles ser creadores, y no solo consumidores de contenidos digitales.

El proyecto CODINC dirige particularmente a:

- Alumnado de educación primaria de entre ocho y doce años y alumnado de educación secundaria de entre quince y dieciocho años, todos procedentes de barrios desfavorecidos. Al primer grupo lo llamamos *alumnos*, y al segundo, *estudiantes*.
- Profesorado de primaria y profesorado de secundaria. Al primer grupo lo llamaremos *maestros*, y al segundo, *profesores*.
- Formadores que trabajen con jóvenes en contextos formales, no formales e informales (escuelas, telecentros, centros para jóvenes, ONG, centros de formación, etc.).
- Otros actores clave: escuelas, familias, organizaciones de la sociedad civil, entidades locales, ONG, instituciones educativas y formativas, autoridades públicas, etc.

En esta metodología describimos, por un lado, el programa de formación sobre programación informática y habilidades pedagógicas, de diez horas de duración, destinado a los estudiantes; y, por otro, el programa de formación dirigido a los alumnos y los estudiantes en el rol de dinamizadores. Veréis que hay muchas similitudes entre la formación de los estudiantes y la de los alumnos. Esto obedece a que la formación que reciben los estudiantes es sobre los mismos módulos que estos después han de enseñar a los alumnos. La formación para estudiantes y profesores, además de centrarse en la programación, lo hace también en métodos pedagógicos, técnicas de evaluación y juegos dinámicos. Formar a los estudiantes para que aprendan a comprender conceptos con rapidez les hace ganar confianza y creer en sus habilidades, a la vez que les permite fortalecer su autoestima y perder el miedo a enseñar cosas que no acaban de entender del todo.

¡Buena lectura!

## OBJETIVOS DE APRENDIZAJE

Con la metodología de CODINC, los estudiantes de secundaria y vosotros, maestros y profesores, llevaréis a cabo una serie de actividades de aprendizaje a través de las cuales pondréis en práctica su pensamiento computacional y sus habilidades sociales y creativas.

Los objetivos de aprendizaje de esta metodología están divididos en dos secciones, una para la educación primaria y otra para la educación secundaria. Cada sección contiene los objetivos para los alumnos o estudiantes y los objetivos para vosotros, maestros y profesores.

Siempre cabe consultar las parrillas de objetivos de aprendizaje para ver rápidamente los objetivos que se asocian a una actividad en particular, por ejemplo.

### Objetivos de aprendizaje para maestros y profesores

Como maestro o profesor, lo primero que harás es adquirir conocimientos sobre pensamiento computacional y programación informática y, después, reforzar tus habilidades pedagógicas.

Al terminar las actividades que os proponemos en esta metodología sabréis responder a las siguientes preguntas:

- ¿Qué es el pensamiento computacional y por qué es un elemento clave en la sociedad?
- ¿Cuáles son las competencias del siglo XXI que deben adquirir vuestros alumnos o estudiantes?
- ¿De qué manera las actividades de programación fomentan el pensamiento computacional, la creatividad y la capacidad de resolución de problemas?
- ¿Qué habilidades sociales y pedagógicas son esenciales para los estudiantes de manera que puedan dinamizar actividades de programación con sus compañeros más pequeños?

Como profesionales de la educación, aprenderéis a llevar a cabo, conducir y dinamizar actividades de formación en las materias siguientes:

- Programación informática.
- Informática analógica.
- Ejercicios de creación y colaboración.
- Ejercicios de diseño.
- Ejercicios pedagógicos y de dinamización.

Los profesores de secundaria reciben la formación junto con sus estudiantes, y los maestros de primaria reciben una formación aparte.

## Objetivos de aprendizaje para estudiantes

En esta formación, los estudiantes llevarán a cabo una completa serie de actividades que les permitirá alcanzar diversos objetivos de aprendizaje en los siguientes campos:

- Pensamiento computacional.
- Habilidades sociales y colaborativas.
- Habilidades creativas.
- Habilidades de diseño y de resolución de problemas.
- Habilidades pedagógicas y de dinamización de grupos.

Los resultados de la formación, sin embargo, pueden ser muchos más, ya que los estudiantes deberán poner en práctica el lenguaje verbal y visual de manera creativa, e incluso deberán utilizar conceptos matemáticos. De hecho, el proyecto final que tendrán que hacer podéis relacionarlo con alguna asignatura, como historia, geografía, arte, etc.

## Pensamiento computacional

Los objetivos de aprendizaje relativos al pensamiento computacional se dividen en dos grupos: los conceptos que deben dominar los estudiantes y los ejercicios cognitivos correspondientes.

En esta metodología se establecen nueve conceptos clave sobre pensamiento computacional, gracias a los cuales los estudiantes llegarán a dominar el ejercicio de la programación informática y a crear sus propios juegos y aplicaciones.

Concepto	Descripción
Algoritmos	Conjuntos de instrucciones dirigidas a un ordenador: por ejemplo, para crear un juego o incluso para cocinar un plato.
Secuencias	Instrucciones que se dan en un orden específico, una tras otra, y que el ordenador ejecuta de principio a fin.
Bucles y repeticiones	Repetición de un subconjunto de instrucciones varias veces o de manera infinita.
Eventos y selección	Indicación del momento en que tiene que pasar algo; por ejemplo, el gato debe comenzar a moverse cuando se pulse el botón «Iniciar» en la aplicación.
Operadores lógicos y condicionales	Elementos que permiten que el ordenador tome alguna decisión; especifican, por ejemplo, que si pasa A el ordenador debe hacer Y, o que si pasa B el ordenador no debe hacer Z.
Operadores matemáticos	Algoritmos que utilizan operaciones matemáticas (multiplicación,

	división, etc.); por ejemplo, para reducir gradualmente la velocidad de un objeto.
Variables y gestión de datos	Espacios en los que se pueden almacenar cifras y letras; por ejemplo, para mostrar en ellos la puntuación actual del jugador.
Funciones	Subconjuntos de códigos que se pueden reaprovechar dando una orden determinada; por ejemplo, para hacer avanzar al protagonista de la aplicación.

Relacionados con estos conceptos hay seis ejercicios cognitivos que permiten poner en práctica las habilidades y conocimientos de programación y desarrollar el pensamiento computacional. Estos ejercicios no solo son útiles en el campo de la tecnología, sino que tienen una alta aplicabilidad en otros ámbitos del conocimiento y de la vida misma, como la resolución de problemas de manera creativa.

Ejercicio cognitivo	Descripción
Estrategias de trabajo incremental e iterativo	Fragmentar el trabajo en varios pasos que se siguen progresivamente y, al mismo tiempo, volver a repetir pasos ya realizados para mejorar la formación. La programación no es un proceso lineal, sino una espiral que se va repitiendo.
Verificación y depuración de errores	Poner a prueba (verificar) los juegos o proyectos creados y ver qué aspectos se pueden mejorar, identificando problemas (errores) y encontrando estrategias para resolverlos (depurarlos).
Reutilización y remezcla	Aprender de los proyectos de los compañeros, reutilizando fragmentos de código o procesos cognitivos y remezclando soluciones y proyectos para crear algo nuevo.
Abstracción	Traducir en patrones abstractos lo que se va aprendiendo durante la formación.
Modularización	Dividir los proyectos en partes más pequeñas (movimiento, velocidad, puntuación, etc.) para reutilizarlas después.
Recopilación y gestión de la información	Conocer fuentes de información y saber buscar otras, sean o no en línea, para encontrar soluciones a los problemas.

### Habilidades sociales y colaborativas.

La metodología de CODINC apuesta con fuerza por el aprendizaje colaborativo y entre iguales, en el que los mayores (estudiantes de secundaria) trabajan conjuntamente para enseñar programación a los más pequeños (alumnos de primaria).

Vuestros estudiantes trabajarán en sus habilidades colaborativas de la siguiente forma. Para empezar, (1) funcionarán como un equipo y aprenderán juntos, (2) aprenderán a negociar entre ellos y también con los alumnos a los que formarán. Enseñar no solo significa tener que conocer la materia impartida, sino disponer también de habilidades pedagógicas que permitan comprender la mentalidad de los más pequeños. Por lo tanto, también aprenderán a (3) describir sus procesos mentales con un lenguaje que sea comprensible para quienes han de recibir la formación, sean estos alumnos de primaria u otros estudiantes de



secundaria como ellos. Finalmente, (4) aprenderán unos de otros y aprenderán también del grupo de primaria en una serie de dinámicas en las que hay que resolver problemas de manera colaborativa.

Cada estudiante hará de formador con un grupo de cinco alumnos. No se trata de impartir lecciones sobre programación, sino de guiarlos en su descubrimiento y en la resolución de problemas. La función de los estudiantes y los profesores es acompañar a los más pequeños en la búsqueda de soluciones para los problemas planteados. Y esto lo hacen con las fichas de trabajo: el estudiante explica a los alumnos qué es lo que hay que hacer y les da trucos y consejos para resolver los problemas.

Pongamos un ejemplo: el Scratch. En la primera sesión los alumnos descubren los aspectos técnicos de este lenguaje de programación. Se les pide que elijan un personaje y que lo hagan caminar, bailar, hablar y cantar, y así es como descubren los principios básicos de la programación. En la segunda sesión dedicada al Scratch se les pide que hagan algunos ejercicios a partir de las fichas. En la tercera y última sesión de Scratch, se les pide que diseñen un juego, y esto no solo significa que tienen que pensar en el diseño, sino también en el contenido que deberá tener. Es decir, a la hora de diseñar un juego hay que pensar cómo se ganará, cómo se perderá y también, por ejemplo, qué personajes habrá. Esto conlleva analizar el problema que haya podido surgir.

La metodología que se sigue en esta formación es la misma para estudiantes y para alumnos, a fin de que los primeros puedan conocer de primera mano lo que tendrán que enseñar y hacer los segundos. De este modo, los estudiantes pueden encontrar la mejor manera de transmitir los conocimientos y acompañar a los alumnos en su proceso de aprendizaje.

En estos talleres también se pone énfasis en la necesidad del refuerzo positivo: hay que felicitar a los alumnos cuando hagan bien las cosas. Si existe la necesidad de amonestar a alguien, es mejor aislar el problema y explicarle con todos los detalles que sean necesarios por qué se le está castigando. El castigo sólo se hará necesario cuando un niño viole repetidamente las normas establecidas después de haberle avisado con anterioridad varias veces. Es preferible no dejar pasar el problema y abordarlo en el momento en que surja para que el niño entienda qué es lo que ha pasado.

### **Habilidades creativas**

A lo largo de la formación también se potencian las habilidades creativas y se ponen en práctica los tres aspectos principales de la creatividad que establece la teoría de la creatividad sistemática de Lego: (1) explorar las posibilidades, las ideas y los proyectos de los demás; (2) combinar las habilidades, las ideas y los proyectos que ya tenemos para crear algo nuevo, y (3) transformar las propiedades que ya existen en cosas nuevas.

## Habilidades de diseño y de resolución de problemas

La metodología que presentamos incorpora las habilidades de diseño y de resolución de problemas de la manera siguiente. Los estudiantes (1) aprenden a identificar problemas nuevos y a descomponerlos, (2) adquieren estrategias para generar ideas nuevas por medio de lluvias de ideas y, posteriormente (3) implementan estas ideas. Asimismo, (4) evalúan sus soluciones y reflexionan sobre el impacto que tienen. Finalmente, (5) buscan maneras de mejorar las soluciones y vuelven a evaluar el problema en una iteración del proceso.

El método de ensayo y error es fundamental en este proyecto y es la manera de reforzar el pensamiento computacional. En el Makey Makey, por ejemplo, se explican los principios básicos del juego y entonces los alumnos ya se ponen a jugar y a hacer cosas.

## Habilidades pedagógicas y de dinamización

El último grupo de objetivos que los estudiantes alcanzarán en esta formación es el relativo a las habilidades pedagógicas y de dinamización de grupos. La consecución de estos objetivos se trabajará a lo largo de las diez horas que duran los talleres de formación, pero sobre todo durante su labor como dinamizadores con los alumnos de primaria: (1) dinamizando grupos y actividades; (2) gestionando la motivación del grupo y de alumnos concretos y proponiendo juegos dinámicos para levantar el ánimo; (3) evaluando actividades, procesos y la tarea de uno mismo como dinamizador; (4) definiendo normas de conducta colectivas, y (5) encontrando y aplicando métodos y habilidades de organización práctica.

Las normas básicas de conducta se establecen al comenzar la formación con los alumnos. Las normas son aprobadas después de haberlas debatido y consensuado entre todos: alumnos, estudiantes y maestros. En un cartel bien visible se escriben las cosas que hay que hacer y las cosas que no se pueden hacer en clase, y si todo el mundo está de acuerdo se firma el documento. A los estudiantes les pedimos que pongan énfasis en la formulación positiva de estas normas, la mayor parte de las cuales seguramente irán apareciendo en el debate conjunto.

1. Tratamos a los demás como nos gustaría que nos trataran a nosotros.
2. Respetamos el material de los demás cuando lo tengamos que utilizar.
3. Reímos con todos, pero no nos reímos de nadie.
4. Somos responsable de nuestro aprendizaje.
5. Llegamos puntuales.
6. Escuchamos al dinamizador y a los compañeros para encontrar soluciones.
7. No molestamos a los que están trabajando.
8. Utilizamos un lenguaje apropiado para que todos nos entiendan.

Este es un debate que tenemos con los estudiantes en el inicio de su formación, y posteriormente ellos también lo han de tener con el grupo de alumnos cuando dinamicen los talleres.

Sin embargo, en este segundo caso es importante discutir con los maestros de los alumnos cómo quieren gestionar estas normas de conducta y ver si en algún punto entran en conflicto con las normas de conducta de la escuela. De todos modos, creemos que es esencial que la definición de las normas corra a cargo de los estudiantes y los alumnos; en este sentido, durante los talleres son los estudiantes los que dinamizan las actividades y velan por el bienestar de los alumnos, y los maestros solo deben intervenir cuando haya problemas graves.

### Cuadro de objetivos de aprendizaje para estudiantes

Las diferentes actividades que propone la metodología de CODINC están enfocadas a trabajar varios de estos objetivos de aprendizaje. A lo largo de las sesiones programadas se alcanzan todos los objetivos planteados. Puede encontrar una descripción general de los diversos objetivos y las actividades relacionadas en el cuadro de objetivos de aprendizaje para estudiantes (véase el apéndice).

## Objetivos de aprendizaje para la educación primaria

### Objetivos de aprendizaje para alumnos

En esta formación los alumnos alcanzarán una gran variedad de objetivos de aprendizaje en los siguientes campos: (1) pensamiento computacional, (2) habilidades sociales y colaborativas, (3) habilidades creativas y (4) habilidades de diseño y de resolución de problemas.

Los resultados de la formación, sin embargo, pueden ser muchos más, ya que los estudiantes deberán poner en práctica el lenguaje verbal y visual de manera creativa, e incluso deberán utilizar conceptos matemáticos. De hecho, el proyecto final que tendrán que hacer podéis relacionarlo con alguna asignatura, como historia, geografía, arte, etc.

### Pensamiento computacional

Los objetivos de aprendizaje relativos al pensamiento computacional se dividen en dos grupos: los conceptos que deben dominar los estudiantes y los ejercicios cognitivos correspondientes.

En esta metodología se establecen nueve conceptos clave sobre pensamiento computacional, gracias a los cuales los estudiantes llegarán a dominar el ejercicio de la programación informática y a crear sus propios juegos y aplicaciones.

Concepto	Descripción
Algoritmos	Conjuntos de instrucciones dirigidas a un ordenador: por ejemplo, para crear un juego o incluso para cocinar un plato.
Secuencias	Instrucciones que se dan en un orden específico, una tras otra, que el ordenador ejecuta de principio a fin.
Bucles y repeticiones	Repetición de un subconjunto de instrucciones varias veces o de manera infinita.
Eventos y selección	Indicación del momento en que tiene que pasar algo; por ejemplo, el gato debe comenzar a moverse cuando se pulse el botón «Iniciar» en la aplicación.
Operadores lógicos y condicionales	Elementos que permiten que el ordenador tome alguna decisión; especifican, por ejemplo, que si pasa A el ordenador debe hacer Y, o que si pasa B el ordenador no debe hacer Z.
Operadores matemáticos	Algoritmos que utilizan operaciones matemáticas (multiplicación, división, etc.); por ejemplo, para reducir gradualmente la velocidad de un objeto.
Variables y gestión de datos	Espacios en los que se pueden almacenar cifras y letras; por ejemplo, para mostrar en ellos la puntuación actual del jugador.
Funciones	Subconjuntos de códigos que se pueden reaprovechar dando una orden determinada; por ejemplo, para hacer avanzar al protagonista de la aplicación.

Relacionados con estos conceptos hay seis ejercicios cognitivos que permiten poner en práctica las habilidades y conocimientos de programación y desarrollar el pensamiento computacional. Estos ejercicios no solo son útiles en el campo de la tecnología, sino que tienen una alta aplicabilidad en otros ámbitos del conocimiento y de la vida misma, como la resolución de problemas de manera creativa.

Ejercicio cognitivo	Descripción
Estrategias de trabajo incremental e iterativo	Fragmentar el trabajo en varios pasos que se siguen progresivamente y, al mismo tiempo, repetir pasos ya realizados para mejorar la formación. La programación no es un proceso lineal, sino una espiral que se va repitiendo.
Verificación y depuración de errores	Poner a prueba (verificar) los juegos o proyectos creados y ver qué aspectos se pueden mejorar, identificando problemas (errores) y encontrando estrategias para resolverlos (depurarlos).
Reutilización y remezcla	Aprender de los proyectos de los compañeros, reutilizando fragmentos de código o procesos cognitivos y remezclando soluciones y proyectos para crear algo nuevo.
Abstracción	Traducir en patrones abstractos lo que se va aprendiendo durante la formación.
Modularización	Dividir los proyectos en partes más pequeñas (movimiento, velocidad, puntuación, etc.) para reutilizarlas después.
Recopilación y gestión de la información	Conocer fuentes de información y saber buscar otras, sean o no en línea, para encontrar soluciones a los problemas.

### Habilidades sociales y colaborativas.

La metodología de CODINC cree firmemente en la enseñanza colaborativa, dentro de la cual los alumnos exploran, experimentan y crean un proyecto final de manera conjunta. Como resultado, se alcanzan y se ponen en práctica un amplio conjunto de habilidades sociales y colaborativas.

Los alumnos (1) trabajan juntos para crear un juego como proyecto final, proceso durante el cual aprenden a (2) negociar los unos con los otros y a (3) describir sus procesos mentales. Además, aprenden a (4) presentar sus proyectos a los compañeros y a (5) hacer y recibir críticas y comentarios. Finalmente, su aprendizaje no es un camino que recorren solos, sino que (6) es una experiencia compartida.

### Habilidades creativas

A lo largo de la formación también se potencian las habilidades creativas y se ponen en práctica los tres aspectos principales de la creatividad que establece la teoría de la creatividad sistemática de Lego: (1) explorar las posibilidades, las ideas y los proyectos de los demás; (2) combinar las habilidades, las ideas y los proyectos que ya tenemos para crear algo nuevo, y (3) transformar las propiedades que ya existen en cosas nuevas.

## **Habilidades de diseño y de resolución de problemas**

La metodología que presentamos incorpora las habilidades de diseño y de resolución de problemas de la manera siguiente. Los estudiantes (1) aprenden a identificar problemas nuevos y a descomponerlos, (2) adquieren estrategias para generar ideas nuevas por medio de lluvias de ideas y, posteriormente (3) implementan estas ideas. Asimismo, (4) evalúan sus soluciones y reflexionan sobre el impacto que tienen. Finalmente, (5) buscan maneras de mejorar las soluciones y vuelven a evaluar el problema en una iteración del proceso.

## **Cuadro de objetivos de aprendizaje para los estudiantes**

Las diferentes actividades que propone la metodología de CODINC están enfocadas a trabajar varios de estos objetivos de aprendizaje. A lo largo de las sesiones programadas se alcanzan todos los objetivos planteados. Puede encontrar una descripción general de los diversos objetivos y las actividades relacionadas en el cuadro de objetivos de aprendizaje para alumnos (véase el apéndice).

## MARCO TEÓRICO

### Habilidades del siglo XXI

El Foro Económico Mundial (FEM) hizo un estudio de las habilidades más buscadas en 2020 en comparación con 2015. Según esta investigación:

La creatividad se convertirá en una de las tres habilidades más solicitadas en los trabajadores. Con la avalancha de nuevos productos, nuevas tecnologías y nuevas formas de trabajar, los trabajadores deberán ser más creativos si quieren beneficiarse de estos cambios.<sup>1</sup>

Paralelamente, la escucha activa, actualmente considerada una habilidad esencial, desaparecerá de la lista de las diez habilidades más demandadas, al tiempo que la inteligencia emocional, que hoy en día no figura en la lista, pasará a ser una de las más importantes.

El FEM también reveló que el 65% de los niños de seis años que empiezan la escuela este año acabarán teniendo un trabajo que todavía no existe. Esto supone concebir la educación y el aprendizaje desde una dimensión totalmente diferente.

En Europa, la Organización para la Cooperación y el Desarrollo Económico (OCDE) estableció una lista de habilidades que niños, jóvenes y adultos deberán tener para poder moverse con facilidad en la sociedad del mañana: son las habilidades del siglo XXI. Son habilidades tales como la resolución de problemas, la colaboración, la creatividad o el pensamiento computacional, que están interconectadas y que deberían formar parte de la educación de toda la ciudadanía a fin de permitirnos nuestro desarrollo en el futuro y estar preparados para el aprendizaje permanente.

Concretamente, las habilidades que define la OCDE son:

1. Pensamiento crítico
2. Pensamiento creativo
3. Resolución de problemas
4. Pensamiento computacional
5. Alfabetización informacional
6. Habilidades informáticas
7. Alfabetización mediática
8. Comunicación
9. Colaboración
10. Habilidades sociales y culturales
11. Autorregulación

---

<sup>1</sup> GRAY, A. (2016). «The 10 skills you need to thrive in the Fourth Industrial Revolution», a *World Economic Forum*. <<https://www.weforum.org/agenda/2016/01/the-10-skills-you-need-to-thrive-in-the-fourth-industrial-revolution>>.

La metodología de CODINC se basa en el desarrollo de estas habilidades del siglo XXI y se centra sobre todo en el pensamiento computacional, la resolución de problemas, el pensamiento creativo, la colaboración y la comunicación. El resto de habilidades también están presentes en el proyecto, ya que, como decíamos, todas están relacionadas entre sí.

## Pensamiento computacional

### ¿Qué es el pensamiento computacional?

Cuando oímos hablar por primera vez del pensamiento computacional lo que nos viene a la cabeza quizás tiene que ver con la informática. Sin embargo, es un concepto mucho más amplio. El término lo estableció Jeanette Wing en 2006 en la Carnegie Mellon University.

En Internet podéis encontrar una gran cantidad de información sobre el tema, pero se recomienda en especial este [vídeo](#) de Google, en el que se explica brevemente qué es el pensamiento computacional.

No se trata únicamente de una habilidad, sino de una serie de conceptos, herramientas, aplicaciones y estrategias mentales que se utilizan para resolver problemas. Se puede utilizar tanto para descubrir el genoma humano como para analizar la obra de Shakespeare, y al mismo tiempo cabe aplicarlo sin utilizar el ordenador. Pero entonces ¿qué es?

El pensamiento computacional es una manera de encarar la resolución de problemas. Significa detectar un problema, aislarlo y encontrar la manera de abordarlo haciendo uso de nuestros conocimientos y habilidades informáticas. (Diane Main, directora del departamento de Aprendizaje, Innovación y Diseño de la Upper School, en la Harker School, California).

Google divide el pensamiento computacional en cuatro aspectos:

1. Descomposición: dividir un problema en partes más pequeñas.
2. Reconocimiento de patrones: buscar diferencias y similitudes entre las diversas partes para poder hacer predicciones.
3. Abstracción: buscar los principios básicos de las diversas partes y los patrones que siguen los problemas.
4. Diseño de algoritmos: elaborar una serie de instrucciones paso a paso para resolver los diferentes problemas.

### ¿Por qué es tan importante el pensamiento computacional?

En una sociedad como la nuestra, cada vez más compleja y más basada en la tecnología, es clave que los alumnos y los estudiantes aprendan a pensar de manera crítica y aprendan a crear su propia experiencia digital. Más que simples consumidores de tecnología digital, queremos que se conviertan en productores de esta tecnología y que lo hagan desde una



actitud crítica. El pensamiento computacional, de este modo, forma parte de la alfabetización mediática.

La brecha digital, tal como la conocemos hoy en día, es decir, entre quienes tienen acceso a la tecnología y quienes no la tienen, pasará a ser entre quienes saben crear tecnología y quienes no. Es aquí donde radica la importancia del pensamiento computacional. En un lado estarán los que sepan controlar el uso que hacen de la tecnología y generar soluciones para cubrir sus necesidades, y por el otro lado otra estarán los que consumen tecnología sin mentalidad crítica.

En este sentido, un contacto temprano con el pensamiento computacional en la escuela puede animar a los más pequeños a decantarse por áreas de estudio más relacionadas con STEAM. Esto puede ser especialmente positivo para las niñas, ya que un buen acompañamiento en su primera incursión en estos campos puede ayudarlas a sentirse cómodas.

Aunque también está presente el factor profesional. El mercado laboral cada vez necesitará más y más trabajadores con formación en TIC y programación; de hecho, ya existe esta demanda no cubierta. Sin embargo, el pensamiento computacional no será el único requisito para todas estas tareas relacionadas con la tecnología y la programación, sino que, ahora que cada vez hay más tareas «simples» que quedan resueltas gracias a la robótica y la inteligencia artificial, la necesidad de trabajadores con habilidades creativas y de resolución de problemas también será creciente.

## Pensamiento creativo

Esta habilidad resulta complicada de definir. Para empezar: ¿qué es la creatividad? Hay tres actividades principales que conforman lo que entendemos por creatividad:<sup>2</sup>

**Combinar:** generar ideas y cosas nuevas, sorprendentes y valiosas combinando ideas y proyectos que ya existen.

**Explorar:** ampliar los conocimientos que tenemos sobre una materia concreta a partir de ideas y cosas nuevas, sorprendentes y valiosas.

**Transformar:** cambiar la manera en que vemos o entendemos el mundo a partir de ideas y cosas nuevas, sorprendentes y valiosas.

En este conjunto de **habilidades del siglo XXI**, el pensamiento creativo se puede dividir en habilidades más concretas, tales como (1) conocer y utilizar técnicas creativas, (2) salir de la ruta principal, (3) saber ver nuevas conexiones y combinaciones, (4) atreverse a asumir

<sup>2</sup> ACKERMANN, E. (et al.) (2010). *Defining systematic creativity in the digital realm*. Billund: Lego Learning Institute. <<https://westminsterresearch.westminster.ac.uk/item/905xw/defining-systematic-creativity-in-the-digital-realm>>.

riesgos (calculados), (5) entender los errores como posibilidades de aprendizaje y (6) tener una actitud abierta e investigadora.

## ¿Por qué es necesario aprender a programar?

Hay varias maneras de trabajar estas habilidades del siglo XXI, específicamente el pensamiento computacional y la creatividad. Sin embargo, pensamos que uno de los mejores abordajes es enseñando a los alumnos y los estudiantes a programar: es una práctica que estimula su capacidad para identificar, extrapolar y crear patrones. Si se aprende desde una edad temprana, la programación informática ayuda a entender mejor los sistemas y la manera en que están diseñados, ya que se trata de analizar problemas que van surgiendo y buscar soluciones innovadoras. De igual modo, programando también se conocen nuevas herramientas que permiten crear y construir por uno mismo.

En otra vertiente, comprender cómo funciona el código y los principios que rigen los lenguajes de programación significa tener mucho ganado a la hora de buscar trabajo. Es importante recalcar, sin embargo, que los beneficios que tiene saber programar no se limitan al mercado profesional.

Aprender programación informática permite a los alumnos de primaria devenir creadores y productores en el mundo digital. En vez de consumir tecnología diariamente de una manera acrítica, pueden empezar a construir su propio futuro digital; en vez de ser usuarios de aplicaciones, pueden ser desarrolladores de aplicaciones. Ya sea por motivos profesionales o por pura diversión, aprender a programar supone una ventaja enorme.

## ¿Por qué es tan importante que los niños y jóvenes de barrios desfavorecidos aprendan a programar?

¿Basta con que los sectores de población más desfavorecidos tengan ordenadores para reducir la brecha salarial que los separa de las clases más acomodadas? Muchos estudios señalan que no, que también es necesario que sepan utilizar estos dispositivos.

El acceso a la información y a una adecuada alfabetización digital son factores sociales y económicos clave que pueden dar lugar a una nueva estratificación de la sociedad. Conocida como *brecha informacional*, esta división no es ni mucho menos un concepto nuevo, pero resulta muy probable que gane peso en la era digital. La omnipresencia de las TIC y de la tecnología digital es incuestionable. La información cada vez es más accesible, en Internet y desde dispositivos digitales. Hay muchos estudios que aseguran que la brecha digital que conocíamos hasta ahora, la fractura socioeconómica respecto del acceso a la tecnología, ya ha empezado a disminuir y acabará por desaparecer a medida que vaya madurando el proceso de digitalización. Aun así, a pesar de la creencia popular de que la digitalización y la generalización del acceso a Internet conducirá a un nivel de conocimientos igual en todo el

mundo, los datos más recientes indican diferencias sociales significativas en la manera en que la gente usa las TIC. Es lo que conocemos como la *segunda brecha digital*. Buscar, procesar y evaluar información en este mundo digital es un proceso que requiere un nuevo conjunto de habilidades informacionales. En este sentido, las clases sociales más acomodadas tienen más recursos y mayor capital cultural, y esto les permite tener un conocimiento más vasto de las nuevas tecnologías, y en consecuencia, obtener un mejor y más rápido acceso a la información.<sup>3</sup>

De este modo, los estudiantes procedentes de entornos más favorecidos pueden sacar más partido a la tecnología que los estudiantes de clases más bajas, los cuales utilizan los ordenadores básicamente para jugar a videojuegos, consultar las redes sociales y comprar productos en línea.

Por este motivo, potenciar las habilidades digitales de los grupos más desfavorecidos es una cuestión urgente. Debemos reducir la brecha digital e impulsar la integración y la movilidad social. El hecho de crear productos digitales desde móviles y ordenadores cambia la relación entre el aprendiz con estos dispositivos. Cuando producimos contenido digital, nos convertimos en consumidores proactivos y pasamos a entender los dispositivos electrónicos como herramientas de aprendizaje permanente. Este es otro de los objetivos que se alcanzan indirectamente en esta formación.

### **La programación informática como herramienta de empoderamiento para los grupos más desfavorecidos**

Otorgar a los estudiantes más desfavorecidos una función en la escuela es darles voz y permitir que se sientan parte integrante de la comunidad educativa. Con el método de la enseñanza entre iguales mejoran su bienestar y al mismo tiempo pueden ofrecer herramientas y recursos de empoderamiento a sus compañeros más pequeños. Los jóvenes provenientes de países del Tercer Mundo o llegados a nuestro país en condición de refugiados normalmente no tienen acceso al mismo capital científico ni social que los jóvenes autóctonos, por lo que se enfrentan a mayores índices de abandono escolar y baja autoestima.

Para revertir esta situación, con el proyecto CODINC se les ofrece una formación que pretende aumentar justamente su capital científico y social y prepararlos para hacer de dinamizadores en programación y otras actividades de STEAM a sus compañeros. Este rol de dinamización les ayuda a aumentar la confianza en sí mismos y les permite ser actores en el mundo digital y protagonistas en su comunidad. Del mismo modo, CODINC también puede

<sup>3</sup> AZZOLINI, D. y SCHIZZEROTTO, A. (2017). «The second digital divide in Europe: A cross-national study on students' digital reading and navigation skills», *FBK-IRVAPP Working Papers*, núm. 2017-02. Trento: Research Institute for the Evaluation of Public Policies; Bruno Kessler Foundation. <https://irvapp.fbk.eu/wp-content/uploads/2017/09/FBK-IRVAPP-Working-Paper-No.-2017-02.pdf>.

servir para mejorar la relación entre el profesorado y los estudiantes en el aula, ya que los primeros tienen la oportunidad de ver a los segundos encarando la enseñanza y el aprendizaje desde otro punto de vista, e interpretando un papel educativo totalmente diferente de lo habitual. Asimismo, esto abre la puerta a la cooperación y la cohesión social en el aula.

## Principios pedagógicos básicos y facilitación

Los principios pedagógicos básicos en que se basa la metodología de CODINC se inspiran a su vez en las cuatro *P* que utiliza Mitch Resnick, profesor del LifeLong Kindergarten Group, en el MIT Media Lab, y creador del lenguaje de programación visual Scratch: *projects*, *passion*, *peers* y *play*.

Vea este [vídeo](#), en el que se explican resumidamente estos conceptos y el aprendizaje creativo.

### **Projects (proyectos)**

Tanto los alumnos de primaria como los estudiantes de secundaria están más motivados y aprenden mejor cuando trabajan en proyectos que los animan y que son importantes para ellos. Por tanto, en esta formación no se dan lecciones ni se siguen ejemplos, sino que se construye un proyecto personal. Los alumnos crean un juego que han ideado y diseñado ellos mismos, del que acaban, con razón, estando orgullosos.

### **Passion (entusiasmo)**

Aprender es divertido, o al menos debería serlo. En esta serie de talleres, uno de los objetivos principales tanto para los alumnos como para los estudiantes es que lo pasen bien y que disfruten con lo que hacen. Cuando un tema o un proyecto nos entusiasma, nos sentimos más motivados, nos implicamos más y aprendemos más cosas.

### **Peers (compañeros)**

En CODINC se pone en práctica la enseñanza entre iguales, desde la idea de que trabajar y jugar en colectivo es uno de los procesos de aprendizaje más potentes que hay. Así, los estudiantes no sólo forman a los alumnos, sino que también aprenden de ellos. Al fin y al cabo, aprender de manera individual, además de ser una tarea lenta y limitada, es más aburrido.

### **Play (juego)**

La formación no consta de la serie de sesiones teóricas habituales. Lo que se busca es jugar con la tecnología, con los conceptos y los unos con los otros. Por medio de juegos, dinámicas y experimentación activa, los alumnos aprenden más que sentados en la mesa y escuchando lo que dice el maestro. El ambiente de aprendizaje ideal es aquel donde uno puede jugar y pasárselo bien, no un contexto tradicional o restrictivo.

Este movimiento que apuesta por la programación informática y la educación creativa bebe de los principios teóricos de pedagogos como Jean Piaget y Paulo Freire en las siguientes cuestiones:

### **1. Los niños son aprendices por naturaleza**

El deseo que tienen los alumnos de adquirir conocimientos y habilidades que les permitan comprender y vivir en el mundo que los rodea es tan fuerte como su necesidad de conseguir comida, calor, amor y comodidad.

¿Cómo lo hacen? Aprenden observando detalladamente la cultura humana en la que están inmersos, jugando en ella, participando e imitando lo que ven. Ven lo que hace la gente mayor y con más experiencia que ellos y sienten el impulso de reproducirlo. De hecho, se enrabietan cuando no se les deja hacer las cosas que los demás hacen: ven la danza de la vida y ellos quieren bailar también, no quedarse al margen. Los niños siguen un método científico: elaboran hipótesis y teorías, las ponen a prueba y después las revisan o las abandonan según van viendo.

El maestro o educador puede contribuir a su aprendizaje siempre que dé apoyo y libertad para que hagan lo que quieren hacer y les ayude a investigar lo que quieren investigar, pero solo cuando ellos quieran, pidan o necesiten claramente esta intervención. Las intervenciones no deseadas o no solicitadas no le hacen ningún favor al aprendizaje; al contrario, no son más que un obstáculo.<sup>4</sup>

### **2. Facilitación de actividades de programación y de actividades analógicas**

Crear y construir puede ser todo un reto. Si queremos estimular la creatividad y la innovación, debemos ayudar a los estudiantes a entender este reto como una parte más del proceso de aprendizaje, como un trance que conseguirán pasar cuando vayan adquiriendo práctica. La enseñanza de la programación informática contribuye a identificar las etapas del proceso y observar las reacciones de los estudiantes y de los dinamizadores, así como las habilidades adquiridas en cada una de las etapas.

Esta metodología toma la idea del docente como transmisor de conocimiento en un contexto formal, y la transforma en la concepción del docente como dinamizador del aprendizaje de sus alumnos en un contexto más relajado.

Asimismo, propone cambiar la clásica evaluación final de los conocimientos adquiridos por una serie de evaluaciones parciales a partir de productos creados por los alumnos y estudiantes. Estas evaluaciones son incrementales y acaban uniéndose en un proyecto final.

---

<sup>4</sup> ACKERMANN, E. y FALBEL, A. (1998), *What we mean by learning*, LEGO Dacta. <<http://learn.media.mit.edu/lcl/resources/readings/what-we-mean-by-learning.pdf>>.

Durante las actividades de los talleres, tanto en las de programación como en las analógicas, se les anima a probar varias cosas para ver si funcionan. Se pone en práctica el método de ensayo y error, que obviamente puede conllevar resultados equivocados, pero también puede conducir a la repetición y la innovación.

Todos somos capaces de plantearnos retos, superarlos y mejorar, y en este proceso generamos un espacio seguro en el que el error no es más que una posibilidad. A diferencia de lo que ocurre en una evaluación convencional, un experimento fallido no es un mal final, sino una invitación a volver a intentarlo.

Cuando a los alumnos y estudiantes se les deja libertad de invención y creación, comienzan a entender la tecnología como un medio para resolver problemas reales y para extrapolar lo que van aprendiendo a otros campos.

Los recursos que ofrecemos están pensados como herramientas de aprendizaje práctico que se pueden utilizar para trabajar habilidades necesarias en la economía creativa y digital, tales como el pensamiento crítico, la colaboración, la comunicación, la curiosidad, la invención o la resolución de problemas.

### **Como profesor, ¿tengo que saber programar antes de que empiecen las actividades?**

En estos talleres, nuestros formadores enseñan los principios de la programación a los estudiantes y el profesorado de secundaria en una serie de talleres en el aula cuya duración es de diez horas en total. Posteriormente, estos estudiantes harán su formación en una escuela de primaria.

Los maestros de primaria recibirán también una formación para poder facilitar la tarea de los estudiantes con los alumnos. Por tanto, no, no es necesario que sepáis programar antes de comenzar la formación. Eso sí, es clave que vengáis con ganas de experimentar y probar el trabajar en el aula con una metodología diferente. Si alguien termina antes de tiempo, proponedles mejorar el proyecto (añadir una puntuación, aumentar la dificultad, cambiar el tema, etc.) o crear otro más difícil.

Cuando todo el mundo haya terminado, dadles diez minutos para que cada uno pueda poner a prueba los proyectos del resto y luego poner en común lo que han aprendido hasta ahora.

## CUANDO EL PROYECTO DESPIERTA RECELOS

Cuando se puso en práctica el proyecto Capital Digital, en el que se basa la metodología que utilizamos en CODINC, las escuelas se mostraron favorables a la incorporación de actividades gratuitas sobre programación informática si la entidad responsable del proyecto se encargaba de aportar los equipos y los materiales necesarios. Antes de comenzar la formación es importante, pues, que los formadores de CODINC, junto con los maestros y los centros de primaria, establezcan una buena correlación entre las actividades que se llevarán a cabo y los objetivos de aprendizaje que habrá que alcanzar.

### En cuanto a los maestros de primaria

Los maestros aceptaron la propuesta de aplicar el proyecto Capital Digital una vez vieron cuáles eran las actividades y el entusiasmo que mostraban los alumnos en su realización.

A raíz de la popularidad que últimamente ha adquirido la programación informática, y dada la alta demanda de actividades de este tipo por parte de las familias, en aquella experiencia también se observó que a las direcciones de los centros les gustaba poder ofertar este tipo de talleres. Las metodologías que se utilizan y los objetivos que se trabajan aumentan el prestigio de los centros, y eso es muy positivo para aquellas escuelas que se ubican en barrios considerados marginales. Es primordial que los formadores hablen con la dirección de cada uno de los centros antes de comenzar el proyecto CODINC y firmen un acuerdo por escrito en el que la escuela exprese su apoyo al proyecto.

### Evitemos el caos en clase: un poco de organización

Los maestros temen que se genere un caos en las aulas porque, para ellos, la falta de organización es un impedimento para el aprendizaje de sus alumnos. Por eso es necesario que todo esté bien estructurado: los ejercicios, la programación de las actividades y el espacio físico en el aula. En la formación que hacemos a los estudiantes hablamos de estos aspectos. Para ese día concreto ¿se puede modificar la disposición de las mesas en el aula? Si resulta que no, ¿cómo podemos hacer que los alumnos trabajen en grupo con las mesas separadas? ¿Hay otra aula donde podamos hacer las actividades? Es fundamental hablar de todo eso con los maestros antes de empezar la formación.

### Planificar los talleres antes de empezar

Convendría que planteáramos en bloques de dos o tres horas las actividades de programación, en vez de en sesiones de cincuenta minutos, que suele ser lo habitual en la mayoría de centros de secundaria.

De este modo, lo idóneo sería distribuir las diez horas de formación a los estudiantes en cinco bloques de dos horas, o en dos bloques de tres horas y dos de dos. Las actividades con los alumnos de primaria son mucho más fáciles de planificar.

## **Alumnos de primaria**

### **Haced una demostración práctica de las actividades**

Presentad las actividades de programación explicando a los alumnos qué es lo que harán en cada una y cómo irá avanzando el proyecto. Permitidles que practiquen para que vean que lo que les proponéis es divertido. La mayoría de los alumnos se animan cuando se hacen proyectos en el aula y se les saca de la rutina que suponen las clases diarias.

Preguntad a los estudiantes qué reticencias tienen a la hora de trabajar con alumnos de primaria y hablad de ello. Si tenéis la ocasión, intentad preguntarles con qué tipo de escuela prefieren trabajar. Si no, asignadlos a una y explicadles los motivos por los que necesitan su ayuda para enseñarles programación a los niños.

### **Los jóvenes pueden ser referentes para los más pequeños**

En la mayoría de familias desfavorecidas los vínculos entre familiares son muy fuertes y los hermanos mayores están encantados de ayudar a los más pequeños. Explicadles que pueden ser un referente para los más pequeños y que pueden ayudarles a descubrir y adentrarse en el mundo de la programación informática.

Anexo 1. Cuadro de objetivos de aprendizaje para los estudiantes



## FICHAS

[https://docs.google.com/document/d/1wEyC7jfkImP7ks15qr\\_cx7pFo7kqc5Y0X0lhqYg5U6c/edit#](https://docs.google.com/document/d/1wEyC7jfkImP7ks15qr_cx7pFo7kqc5Y0X0lhqYg5U6c/edit#)

LightBot (aplicación en línea).

Hour of Code

- Con el tema de [La guerra de las galaxias](#).
- Con el tema de [Minecraft](#).
- Con el tema de [Frozen](#).

## MATERIALES

Elemento	Precio unitario aprox. (€)	Cantidad	Precio total aprox. (€)
Makey Makey	55	10	550
Ordenadores o portátiles		10	
Tabletas (opcional)	100	10	1000
Materiales para pintar y dibujar			
Materiales para escribir			

## APÉNDICE DE REFERENCIAS

ANANIADOU, K. y CLARO, M. (2009). «21st century skills and competences for new millennium learners in OECD countries», *OECD Education Working Papers*, núm. 41, París: OECD Publishing. <[http://www.oecd.org/officialdocuments/publicdisplaydocumentpdf/?cote=edu/wkp\(2009\)20&doclanguage=en](http://www.oecd.org/officialdocuments/publicdisplaydocumentpdf/?cote=edu/wkp(2009)20&doclanguage=en)>.

BRENNAN, K. y RESNICK, M. (2012). *Using artifact-based interviews to study the development of computational thinking in interactive media design*. Paper presentado en el congreso anual de la American Educational Research Association (AERA), celebrado en Vancouver,

Canadá.

<[https://web.media.mit.edu/~kbrennan/files/Brennan\\_Resnick\\_AERA2012\\_CT.pdf](https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf)>.

«Curriculum», a *Code Club*. Raspberry Pi Foundation.

<<https://codeclubprojects.org/en-GB/curriculum/>>.

AZZOLINI, D. y SCHIZZEROTTO, A. (2017). «The second digital divide in Europe: A cross-national study on students' digital reading and navigation skills», *FBK-IRVAPP Working Papers*, núm. 2017-02. Trento: Research Institute for the Evaluation of Public Policies, Bruno Kessler Foundation. <<https://irvapp.fbk.eu/wp-content/uploads/2017/09/FBK-IRVAPP-Working-Paper-No.-2017-02.pdf>>.