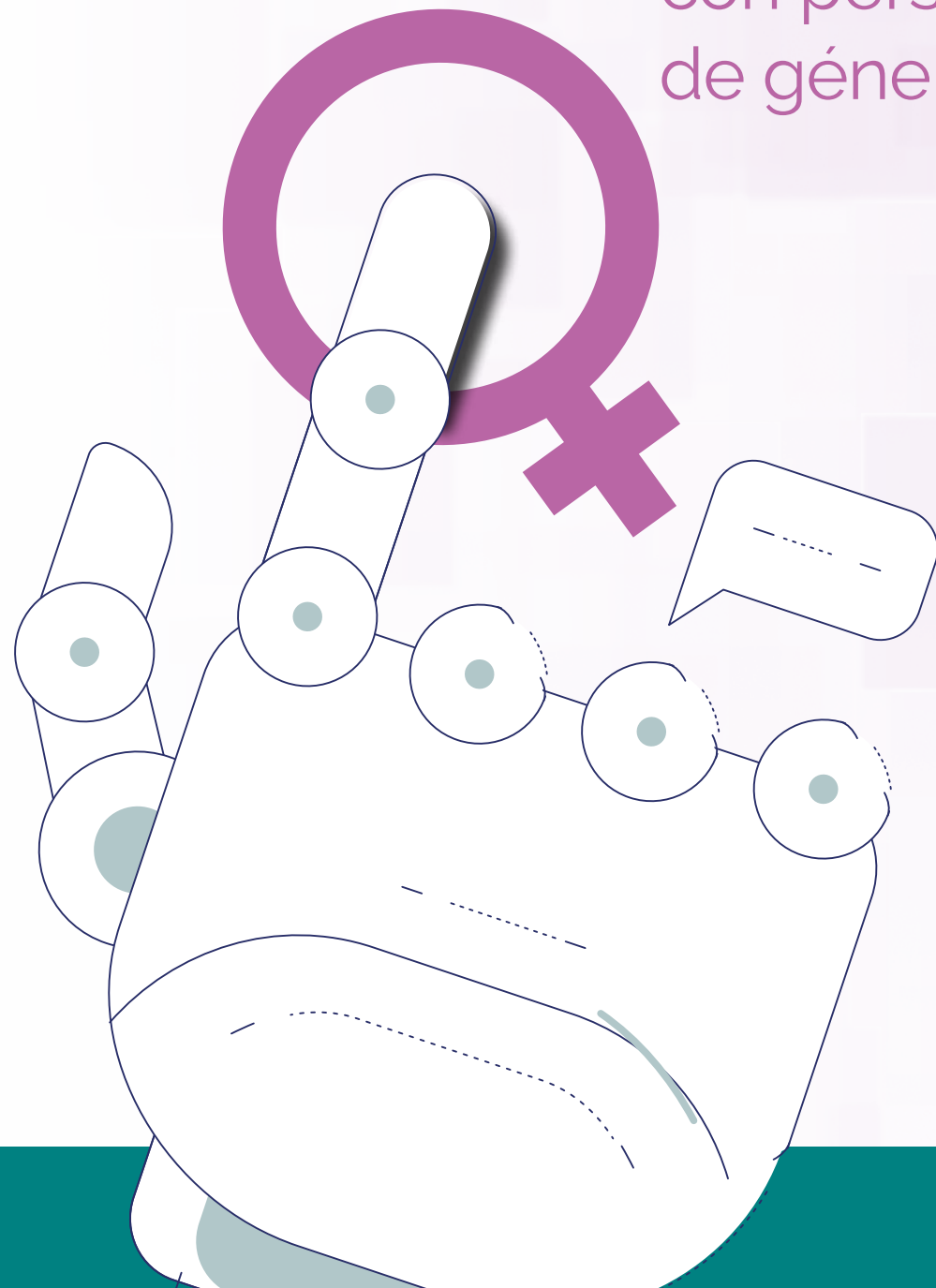


Iniciación
a la **Robótica**
y Programación
con perspectiva
de género



Iniciación a la Robótica y Programación con perspectiva de género © 2025, por Jorge Lobo Martínez, es una obra derivada del MOOC Iniciación a la Robótica y Programación con perspectiva de género, elaborado para INTEF y el Instituto de las Mujeres, publicado bajo licencia Creative Commons Reconocimiento-CompartirIgual (CC BY-SA).

Esta obra se comparte bajo la misma licencia.

Edita: Instituto de las Mujeres (Ministerio de Igualdad)

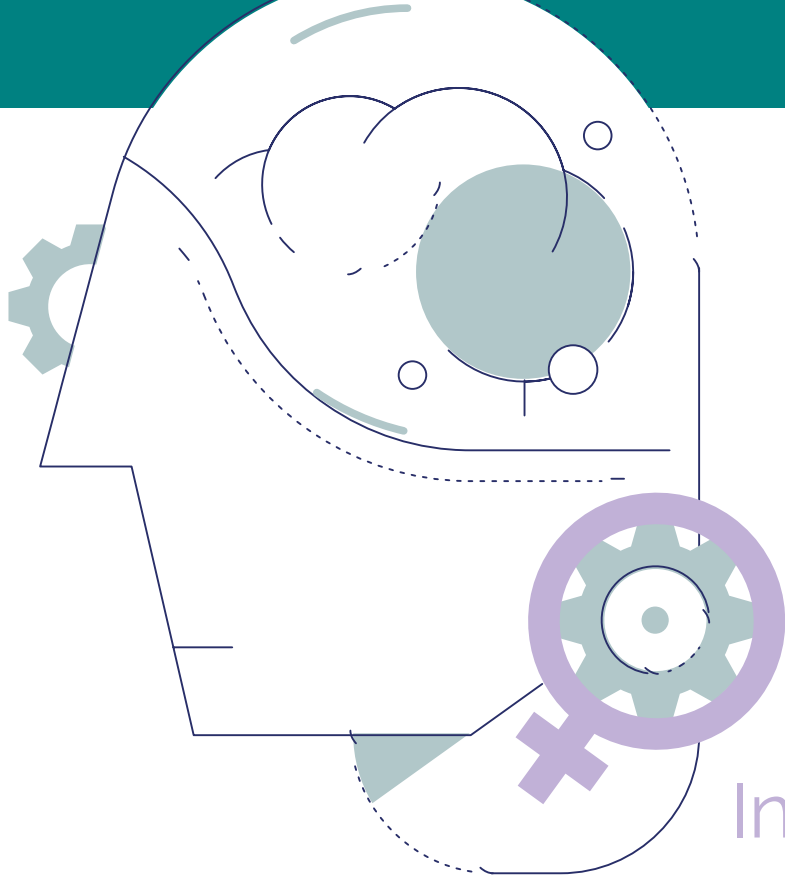
Imprime: Blanca Impresores SL

NIPO: 050-26-030-3

eNIPO: 050-26-031-9

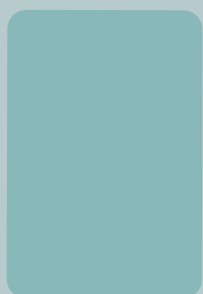
D.L.: M-14127-2026

El Instituto de las Mujeres no se hace responsable de las opiniones expresadas en este estudio ni de los resultados obtenidos en el mismo.



Iniciación
a la **Robótica**
y Programación
con perspectiva
de género

JORGE LOBO



INDICE DE CONTENIDOS

- INTRODUCCIÓN..... 7
- MÓDULO 1: REDUCIENDO LA BRECHA DE GÉNERO EN ACTIVIDADES STEM.....9
 - PARTE 1: ALGUNOS DATOS9
 - 1. Algunos datos.....9
 - 2. Eliminando sesgos de género.....10
 - 3. Aumentando la visibilidad de modelos femeninos.....11
 - 4. Creando entornos inclusivos.....18
 - 5. Para ampliar información19
 - PARTE 2: QUÉ ES EL PENSAMIENTO COMPUTACIONAL20
 - 1. ¿Qué es el pensamiento computacional?.....20
 - 2. Papert y Wing, dos enfoques del pensamiento computacional.....20
 - 3. Definición de pensamiento computacional.....24
 - 4. Los pilares del pensamiento computacional.....26
- MÓDULO 2: ACTIVIDADES UNPLUGGED29
 - 1. ¿A qué llamamos actividades desenchufadas?29
 - 2. Beneficios del pensamiento computacional desenchufado29
 - 3. El marco conceptual de Brennan-resnick.....30
 - 3.1. Brennan y Resnick30
 - 3.2. El marco Brennan-Resnick32
 - 4. Actividades desenchufadas.....33
 - 4.1. Secuencias33
 - 4.2. Bucles.....36
 - 4.3. Paralelismo.....42
 - 4.4. Eventos.....45
 - 4.5. Condicionales.....49
 - 4.6. Datos y operadores.....52
 - 4.7. Repositorio de actividades desenchufadas58
- MÓDULO 3: SCRATCH59
 - 1. Scratch, un entorno de programación por bloques.....59
 - 2. Historias de Scratch.....60
 - 3. Practicamos con Scratchjr.....62

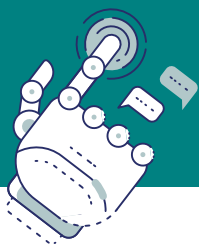
3.1. Interfaz de ScratchJr.....	64
3.2. Bloques de ScratchJr.....	66
3.3. Editor de imagen de ScratchJr.....	70
3.4. Programando en ScratchJr.....	71
4. Practicamos por ScratchJr.....	73
4.1. Interfaz de Scratch.....	74
4.2. Programando en Scratch.....	74
4.3. Ideas para utilizar Scratch en el aula.....	76
5. Creando un entorno inclusivo en Scratch y Scratchjr.....	77

MÓDULO 4: INTELIGENCIA ARTIFICIAL.....81

1. ¿Qué es la inteligencia artificial?.....	81
2. Breve historia de la IA.....	82
3. La IA generativa.....	83
4. Preocupaciones éticas del uso de la IA.....	84
5. IA en el aula.....	86
5.1. Decálogo de la IA en educación.....	87
5.2. Recursos de IA en el Aula.....	89
5.3. Herramientas para crear modelos de machine learning.....	90
5.4. Creando modelos de machine learning en Learning ML.....	93

MÓDULO 5: ROBOTS DE SUELO Y PLACAS EDUCATIVAS95

1. ROBOTS DE SUELO.....	95
1.1. Algunos modelos de robots de suelo.....	97
1.1.1. Beebot.....	97
1.1.2. Ratón robot code & go.....	99
1.1.3. Super-Doc.....	100
1.1.4. Matatastudio Coding Set.....	101
1.1.5. Tale-Bot.....	102
1.1.6. Escornabot.....	104
1.2. Cómo hacer actividades más inclusivas con robots de suelo.....	105
2. PLACAS DE ROBÓTICA EDUCATIVA.....	106
2.1 Arduino.....	107
2.2. Micro:bit.....	110
2.3. Makey Makey.....	112
2.4 Echidna.....	113



INTRODUCCIÓN



Son innumerables los estudios y artículos académicos que revelan las ventajas educativas de la introducción de la robótica en edades tempranas, así como también lo son las experiencias que se han llevado a cabo en centros educativos. Sin embargo, a pesar del esfuerzo y las iniciativas para fomentar las vocaciones STEAM (ámbito de ciencias, tecnología, ingeniería, artes y matemáticas por sus siglas en inglés) no ha tenido un gran impacto en el porcentaje de mujeres graduadas en estos campos, siendo aún menor en aquellos que no tienen conexión directa con los cuidados.

En un mundo donde la tecnología y la robótica están cada vez más presentes en nuestras vidas, es fundamental garantizar que estos avances reflejen valores de igualdad y equidad de género. Este manual te acompañará a la hora de introducirte en el mundo de la Programación y Robótica educativa a la vez que te propondrá reflexionar sobre cómo abordar la brecha de género en actividades STEAM, mediante un enfoque múltiple que incluye iniciativas para fomentar el interés y la participación de las niñas desde pequeñas, eliminar los sesgos de género en la programación y la robótica educativa, dar visibilidad a modelos femeninos en campos STEAM y crear entornos inclusivos que valoren la diversidad y promuevan la igualdad.

Iniciación a la Robótica y Programación con perspectiva de género nace con el objetivo de ofrecer herramientas prácticas y recursos educativos que permitan trabajar la Robótica y la Programación desde un punto de vista inclusivo. Está dirigido principalmente a docentes y personas educadoras que deseen iniciar su práctica en este ámbito, integrando de forma consciente la perspectiva de género en su actividad docente. Los diferentes capítulos del manual forman una secuencia que parte de un análisis de la situación actual de las mujeres en el ámbito STEAM y la necesidad de visibilizar referentes femeninos que ayuden a descubrir y revertir estereotipos de género. A continuación, se introduce el pensamiento computacional, recuperando las aportaciones fundamentales de Seymour Papert y Jeannette Wing, y analizando los pilares básicos sobre los que se sustenta. Desde ahí, se proponen actividades desenchufadas como una forma accesible para iniciarse de forma sencilla y accesible, sin necesidad de dispositivos digitales. Después, se analizan herramientas para la programación por



bloques como una estupenda manera de iniciarse en este campo de forma visual, práctica y fácil. También, se presenta una introducción a la inteligencia artificial con herramientas diseñadas específicamente para trabajar con el alumnado, y se concluye con una primera aproximación a la robótica educativa centradas en el uso de robots de suelo y descubriendo algunas de las numerosas placas de robótica educativa.

Este manual no solo pretende proporcionar recursos y propuestas metodológicas para iniciarse en la programación y la robótica, también propone reflexionar sobre cómo los sesgos de género pueden estar presentes, de forma explícita o implícita, en los contenidos educativos y las dinámicas del aula. Esperamos que la lectura pueda ayudarte a descubrir nuevas herramientas y a repensar la práctica docente para tratar de hacer nuestras clases más inclusivas y respetuosas, además de explorar cómo los estereotipos de género pueden influir en el diseño y la implementación de tecnologías, y cómo pueden ser superados.



Niñas en pie de ciencia, del canal de Youtube [Canal Ministerio de Educación, FP y Deportes](#), bajo licencia [Youtube Estandar](#)



MÓDULO 1: REDUCIENDO LA BRECHA DE GÉNERO EN ACTIVIDADES STEM



PARTE 1: ALGUNOS DATOS

1. Algunos datos

Según diferentes informes, desde los más antiguos como [*Descifrar el código: La educación de las niñas y las mujeres en ciencias, tecnología, ingeniería y matemáticas \(STEM\)* \(UNESCO, 2017\)](#) hasta los más recientes como [*La brecha de género en la Educación STEM* \(Martin et al., 2022\)](#) las niñas van siendo alejadas de estas disciplinas a lo largo de su trayectoria educativa por diferentes factores individuales, familiares y de los pares, escolares y sociales, lo que limita sus oportunidades para ingresar en estos campos como adultas.

La brecha de género en disciplinas STEAM en España comienza en la Educación Primaria y continúa hasta el ámbito laboral.

El informe [*Mujeres en STEM. Desde la educación básica hasta la carrera laboral*, \(Cobrerros et al. 2024\)](#) revela esta brecha de género en disciplinas STEAM con algunos datos de interés:

En Educación Primaria, las niñas muestran una probabilidad un 15% menor que los niños de considerar las matemáticas como su materia preferida, y entre un 8 y un 9% menor de considerarse buenas, aprender rápido, o disfrutar con esta asignatura. Piensan con mayor probabilidad que son aburridas y difíciles.

- En Bachillerato, aunque hay más mujeres que completan con éxito estudios STEM, solo el 52% elige estas áreas, frente al 65% de los hombres.
- En la educación superior, las mujeres matriculadas en grados universitarios STEM no llegan al 50% en casi ningún caso, y en Matemáticas (36%), Física (27%), Telecomunicaciones (23%), o Informática (13%) son especialmente bajas.



- En la Formación Profesional, solo el 7% de las mujeres se gradúan en STEM, frente al 52% de los hombres. En la mayoría de los grados STEM la proporción hombres/mujeres es de prácticamente 9 a 1.
- En el ámbito laboral, solo el 5,5% de las mujeres trabajan en campos STEM, ocupando únicamente el 25% de los puestos en esta área.

Algunas ideas propuestas para cerrar estas brechas son:

- Fomentar la participación en áreas STEM a través actividades enmarcadas en estas áreas.
- Eliminar sesgos de las actividades y de los materiales, haciéndolos más flexibles y adaptables.
- Incrementar el acceso a referentes STEM femeninos, tanto del pasado como actuales.

2. Eliminando sesgos de género

El profesorado, su trato con el alumnado y sus creencias y expectativas sobre la habilidad de ellos y ellas en las diferentes asignaturas, tienen un impacto determinante en su rendimiento y elección de futuros itinerarios educativos.

Hay investigaciones que muestran que el profesorado presenta sesgos que pueden tener implicaciones en la brecha de género en áreas STEM. En la investigación Exploring teachers' and students' gender role bias and students' confidence in STEM fields (Hand et al., 2017), se pusieron de manifiesto estos sesgos: tanto el profesorado como el alumnado atribuían más características tradicionalmente asociadas al género masculino a las personas dedicadas a la ciencia y características tradicionalmente asociadas al género femenino para las dedicadas a humanidades. Además, el profesorado y el alumnado también expresaron su creencia de que los niños tienden a ser mejores que las niñas en disciplinas STEM.

3. Aumentando la visibilidad de modelos femeninos

Dar a conocer referentes femeninos en campos como la Programación, la Robótica y la inteligencia artificial (IA) representa un papel fundamental en la reducción de la brecha de género en áreas STEM en la educación. Como se ha comentado previamente, una de las razones que se han identificado para explicar la escasa participación de las mujeres en carreras STEM es la falta de modelos femeninos a seguir en estos campos. La representación y divulgación de mujeres referentes en estas áreas puede inspirar a niñas y jóvenes a seguir sus pasos y a interesarse e identificarse con estas disciplinas. La presencia de referentes femeninos también ayuda a reducir los estereotipos y sesgos de género arraigados, promoviendo la idea de que hombres y mujeres son igualmente capaces en disciplinas técnicas y científicas, desterrando la idea de que ciertas áreas STEM son eminentemente masculinas.

- EL EFECTO MATILDA

El efecto Matilda es un término utilizado para describir el fenómeno por el que las mujeres no reciben el crédito que merecen por sus contribuciones científicas, siendo sus logros ignorados, minimizados o atribuidos erróneamente a hombres.

El término se originó en honor a [Matilda Joselyn Gage](#), la primera mujer dedicada al ámbito científico que denunció este aspecto negativo en la sociedad.



NO MORE MATILDAS, del canal de Youtube [#NoMoreMatildas](#), bajo licencia [Youtube Estandar](#)



- Algunos referentes femeninos en Ciencias de la computación, Robótica e IA



Ada Lovelace (1815-1852): considerada la primera programadora de la historia, Ada Lovelace desarrolló el primer algoritmo destinado a ser procesado por una máquina, estableciendo así los fundamentos de la programación informática.

Ada Byron daguerreotype by Antoine Claudet 1843 or 1850, por Antoine Claudet, Public domain, via Wikimedia Commons.



Grace Hopper (1906-1992): pionera en el campo de la informática, Grace Hopper fue una destacada científica de la computación que contribuyó al desarrollo del primer compilador para un lenguaje de programación y al desarrollo del lenguaje COBOL.

Commodore Grace M. Hopper, USN (covered), por James S. Davis, Public domain, via Wikimedia Commons.



Heidy Lamar (1914-2000): actriz e inventora austriaco-estadounidense, Heidy Lamar es conocida por su invención de la tecnología de "salto de frecuencia", que sienta las bases para tecnologías modernas como el WiFi y el Bluetooth.

Hedy Lamarr in "The Heavenly Body", por Employee(s) of MGM, Public domain, via Wikimedia Commons.



Katherine Johnson (1918-2020): matemática y científica de la NASA, Katherine Johnson fue fundamental en los primeros vuelos espaciales tripulados por humanos. Realizó cálculos críticos para la seguridad de las misiones espaciales, incluido el vuelo de John Glenn en 1962.

Katherine Johnson 1983 por NASA, Public domain, via Wikimedia Commons.



Fran Allen (1932-2020): pionera en el campo de la optimización de compiladores y programación paralela, Fran Allen fue la primera mujer en recibir el Premio Turing, el más prestigioso en el campo de la informática, por sus contribuciones a la teoría y práctica de la compilación.

Frances E. Allen receiving the Erna Hamburger Distinguished Lecture Award at the EPFL, por Rama, CC BY-SA 2.0 fr, via Wikimedia Commons.



Margaret Hamilton (1936): conocida por su trabajo en el programa espacial Apolo de la NASA, Margaret Hamilton fue una ingeniera de software clave en el desarrollo del software de guía de vuelo de la misión Apolo 11, que llevó al primer alunizaje humano en la Luna.

Photograph of Margaret Hamilton por Daphne Weld Nichols, CC BY-SA 3.0, via Wikimedia Commons.



Lynn Conway (1938): ingeniera eléctrica y pionera en diseño de microchips, Lynn Conway es conocida por su trabajo en la arquitectura de computadoras y el diseño de chips VLSI. Fue fundamental en el desarrollo del diseño de chips en la década de 1970 y ha sido una defensora de los derechos de las personas transgénero en la ciencia y la tecnología.

Lynn Conway July 2006, por Charles Rogers, CC BY-SA 2.5, via Wikimedia Commons.



Barbara Liskov (1939): destacada informática y profesora en el MIT, Barbara Liskov es conocida por su trabajo en lenguajes de programación, sistemas distribuidos y programación orientada a objetos. Fue una de las primeras mujeres de los Estados Unidos en conseguir un doctorado en ciencias de la computación, y recibió el Premio Turing en 2008.

Barbara Liskov speaking at OOPSLA 2009 conference por Mirko Raner, CC BY-SA 3.0, via Wikimedia Commons.





Barbara Grosz (1948): destacada investigadora en inteligencia artificial y lingüística computacional, Barbara Grosz es conocida por su trabajo en diálogos computacionales y sistemas multiagente. Fue la primera mujer en recibir el Premio AAAI por contribuciones sobresalientes a la inteligencia artificial.

[Barbara Grosz](#), por Bengt Oberger, CC BY-SA 4.0, via Wikimedia Commons.



Anita Borg (1949-2003): científica informática y activista por la igualdad de género en tecnología, Anita Borg fue la fundadora del Instituto de Investigación para las Mujeres y la Tecnología de la Información y la autora de "Mujeres en la Tecnología".

[Anita Borg \(January 17, 1949 – April 6, 2003\)](#), por autor/a desconocida, Copyright (usado bajo criterio de Fair Use), via Wikimedia Commons.



Radia Perlman (1951): conocida como la "Madre de Internet", Radia Perlman es una ingeniera de redes informáticas que desarrolló el algoritmo de árbol de expansión de puentes (STP), fundamental para la estabilidad de las redes Ethernet.

[Radia Perlman](#), por Scientist-100 at English Wikipedia, Public domain, via Wikimedia Commons.



Carme Torras Genís (1956): matemática y doctora en informática, es experta en robótica, visión por computador e inteligencia artificial. Profesora de investigación en el Instituto de Robótica e Informática Industrial (CSIC-UPC). Compagina la investigación científica con la literatura.

[PNR-TNC_CarmeTorras](#), por [Mussklprozz](#), CC BY-SA 4.0, via Wikimedia Commons.



Amparo Alonso Betanzos (1957): catedrática en la Universidad de A Coruña y presidenta de la Asociación Española para la Inteligencia Artificial (AEPIA) entre 2013 y 2021. En 2020 entró a formar parte recién constituido del [Consejo Asesor de Inteligencia Artificial](#) del Ministerio de Asuntos Económicos y Transformación Digital.

[Retrato de Amparo Alonso tomado antes da súa participación no encontro Big Data Coruña 2017](#), por Marcusfernandez, CC BY-SA 4.0, via Wikimedia Commons.



Carla Gomes (1960): científica informática y experta en inteligencia artificial, Carla Gomes es conocida por su trabajo en la aplicación de técnicas de IA a problemas complejos en áreas como la ecología, la sostenibilidad y la biología computacional. Es pionera en la integración de la IA en la toma de decisiones en políticas ambientales.

[Professor Carla Gomes, giving a conference presentation](#), por Dennis Hamilton, CC BY-SA 2.0, via Wikimedia Commons.



Cynthia Breazeal (1967): pionera en el campo de la robótica social, Cynthia Breazeal es conocida por su trabajo en el desarrollo de robots interactivos y emocionales. Es la fundadora del Laboratorio de Robótica Personal en el MIT.

[Cynthia Breazeal with Jibo](#), por Cynthia Breazeal, CC BY-SA 4.0, via Wikimedia Commons.



Nuria Oliver (1970): ingeniera en telecomunicaciones doctora por el MIT. Es una de las mayores divulgadoras científicas en España. Es conocida por su trabajo en modelos computacionales de comportamiento humano, inteligencia artificial, interacción persona-máquina, informática móvil y *big data* para bien social.

[Portrait of Nuria Oliver](#), por Ester Pérez Broto, CC BY-SA 2.0 via Wikimedia Commons.





Elena García Armada (1971): ingeniera industrial e investigadora del CSIC. Desarrolló el primer exoesqueleto pediátrico del mundo, que ayuda a niños con parálisis a caminar. Su empresa Marsi-Bionics ha sido premiada por su innovación y contribución a la robótica asistencial. Su labor combina ciencia, tecnología y compromiso social.

[Elena García Armada en los Reconocimientos 8 de Marzo](#). Día de la Mujer (2018), por [Cristina Cifuentes](#), CC BY-SA 2.0 via Wikimedia Commons.



Ayanna Howard (1972): ingeniera robótica e investigadora en inteligencia artificial. Ayanna Howard es conocida por su trabajo en el diseño y desarrollo de sistemas robóticos autónomos y accesibles. Su investigación se centra en el desarrollo de tecnologías robóticas para mejorar la calidad de vida de las personas con discapacidad. Además, es defensora de la diversidad en STEM y ha recibido premios por su liderazgo en ingeniería.

[Ayanna M. Howard - SnoMote](#), por Ayanna M. Howard, CC BY-SA 3.0, via Wikimedia Commons.



Reshma Saujani (1975): política y abogada, Reshma Saujani es conocida por ser fundadora y directora ejecutiva de [Girls Who Code](#), una organización que se enfoca en cerrar la brecha de género en la tecnología y la programación al proporcionar oportunidades educativas a jóvenes mujeres.

[2018 Prathamusa Reshma Saujani](#), por Fuzheado, CC BY-SA 4.0, via Wikimedia Commons.



Fei-Fei Li (1976): investigadora en inteligencia artificial y visión por computadora, Fei-Fei Li es conocida por su trabajo en el desarrollo de algoritmos de reconocimiento de imágenes. Es cofundadora del Laboratorio de IA y Visión de la Universidad de Stanford.

[Fei-Fei Li speaking at AI for Good 2017](#), por ITU Pictures, CC BY-SA 2.0, via Wikimedia Commons.



Timnit Gebru (1980): científica de datos y ética de la inteligencia artificial conocida por su trabajo en equidad y justicia en algoritmos de IA. Gebru ha investigado temas como el sesgo algorítmico y la diversidad en la tecnología. También es cofundadora de Black in AI, una organización para promover el aumento de la presencia y la inclusión de personas negras en el campo de la IA.

[Timnit Gebru crop](#), por TechCrunch, CC BY-SA 2.0, vía Wikimedia Commons.



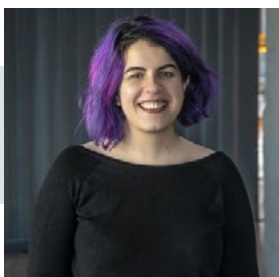
Katherine Maher (1983): especialista en tecnología, desarrollo y democracia, Katherine Maher es conocida por haber sido directora ejecutiva de la Fundación Wikimedia, la organización sin ánimo de lucro detrás de Wikipedia. Maher es una defensora del acceso abierto al conocimiento y de la diversidad en la tecnología.

[Katherine Maher in 2016](#), por Myleen Hollero, CC BY-SA 4.0, vía Wikimedia Commons.



Margaret Mitchell (1986): científica de datos y ética de la inteligencia artificial, Mitchell es conocida por su trabajo en la comprensión del lenguaje natural y la ética en el aprendizaje automático. También es una defensora de la transparencia y la equidad en los algoritmos de IA.

[Margaret Mitchell 2017S](#), por Mujeresconciencia, CC BY-SA 4.0, vía Mujeresconciencia.



Nerea Luis Minguenza (1988): doctora en Inteligencia Artificial y cofundadora del evento [T3chFest](#). Investiga en algoritmos explicables y sesgos de género en sistemas automáticos. Es conocida por su compromiso con la visibilización de mujeres en la tecnología y la educación tecnológica accesible para todos.

[Nerea Luis](#), por [mujerescreativas](#), CC BY-SA 4.0, vía [mujerescreativas](#).





Joy Buolamwini (1989): investigadora en inteligencia artificial y activista por la ética en la IA, Joy Buolamwini es reconocida por su trabajo en la identificación y mitigación de sesgos algorítmicos y discriminación facial en sistemas de reconocimiento facial. Es fundadora del Proyecto Algoritmo Justo y cofundadora del Laboratorio de Equidad Algorítmica de la Universidad de Stanford.

[Joy Buolamwini](#), 2018, por Niccolò Caranti, CC BY-SA 4.0, via Wikimedia Commons.

4. Creando entornos inclusivos

Además de las medidas mencionadas anteriormente, como revisar y monitorizar nuestro trato con el alumnado para tratar de detectar si **nuestros propios sesgos** pueden estar transmitiendo creencias y expectativas que contribuyan a mantener la brecha de género entre nuestro alumnado, y **aumentar la exposición a referentes femeninos**, existen algunos otros principios a considerar para que nuestras actividades de Programación y Robótica sean más inclusivas.

- **Dotar a los proyectos de un carácter inclusivo:** seleccionar proyectos o desafíos que no estén vinculados a estereotipos de género y que sean igualmente atractivos para todo el alumnado. Por ejemplo, en lugar de centrarse en la construcción de robots de combate o automóviles, considerar proyectos relacionados con la resolución de problemas del mundo real. Y, además, dándole la oportunidad al alumnado de decidir sobre sus propias creaciones, ofreciendo actividades más abiertas, orientadas a trabajar las competencias, destrezas y contenidos de una forma más libre.
- **Fomentar el trabajo en equipo diverso:** promover la formación de equipos diversos en términos de género, habilidades, conocimientos, etc., ayuda a evitar que se reproduzcan dinámicas de género en el aula y fomenta la colaboración entre estudiantes con diferentes enfoques y experiencias. Este principio requiere que el profesorado facilite que todas las voces sean escuchadas y valoradas durante las actividades mediante normas de participación, creación de un ambiente seguro para compartir ideas y una metodología que permita oportunidades equitativas para que todas las personas puedan contribuir al proyecto.

- **Lenguaje inclusivo:** utilizar un lenguaje inclusivo en las explicaciones, instrucciones y materiales de la actividad, evitando el uso de términos o ejemplos que refuercen estereotipos de género, permitirá asegurar de que todas las personas se sientan representadas y bienvenidas.
- **Refuerzo positivo y reconocimiento de logros igualitario:** reconocer y reforzar los logros de todo el alumnado de manera equitativa, evitando favoritismos o prejuicios basados en el género ayudará a crear ese entorno seguro en el que todo el alumnado pueda mejorar y crecer en sus habilidades.
- **Contextualización inclusiva:** presentar las actividades y desafíos de manera que sean relevantes y accesibles para todo el alumnado, evitando estereotipos de género, eligiendo ámbitos que interesen a una mayoría del alumnado y relacionar la Programación y la Robótica con contextos cotidianos o del mundo real.
- **Referentes diversos:** relacionar las actividades con el trabajo de referentes tanto femeninos como masculinos mostrando al alumnado que estas disciplinas están abiertas a todas las personas.

5. Para ampliar información

- **No more Matildas** (<https://www.nomorematildas.com/>) Asociación sin ánimo de lucro que persigue dar visibilidad a la mujer en la ciencia, y crear contenido para recuperar a las científicas para la historia.
- **Mujeres STEAM** (<https://mujeresteam.es/>) Proyecto para la incorporación de las mujeres en los sectores de ciencia, tecnología, ingeniería, arte y matemáticas.
- **Vocaciones STEAM** (https://www.ciencia.gob.es/Secc-Servicios/Igualdad/vocaciones_stem.html) Recursos del Ministerio de Ciencia, Innovación y Universidades para fomentar entre las niñas y adolescentes las vocaciones científicas desde las materias relacionadas con los ámbitos STEAM.
- **11 de febrero** (<https://11defebrero.org/>) Web de la iniciativa ciudadana para conmemorar el Día Internacional de la Mujer y la Niña en la Ciencia.



PARTE 2: QUÉ ES EL PENSAMIENTO COMPUTACIONAL

1. ¿Qué es el pensamiento computacional?

El pensamiento computacional es un enfoque de resolución de problemas que se utiliza comúnmente en la informática y la programación, aunque también se puede aplicar a una amplia variedad de situaciones no informáticas.

En términos generales, el pensamiento computacional implica descomponer un problema complejo en partes más pequeñas y manejables, identificar patrones y relaciones entre los datos, y diseñar algoritmos para solucionar el problema de manera eficiente y efectiva.

El pensamiento computacional a menudo se describe como una combinación de habilidades analíticas, lógicas y creativas que se utilizan para abordar problemas complejos. Estas habilidades incluyen la capacidad de abstraer información importante, la capacidad de formular y probar hipótesis, la capacidad de diseñar y evaluar soluciones, y la capacidad de comunicar ideas de manera clara y concisa.

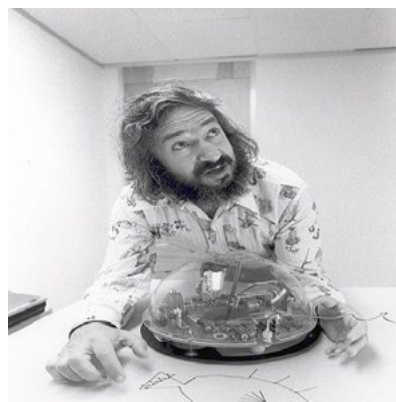
El pensamiento computacional se considera una habilidad importante en una amplia variedad de campos, no solo en la informática y la programación, sino también en la ciencia, la ingeniería, la medicina, la educación y muchos otros.

2. Papert y Wing, dos enfoques del pensamiento computacional

- PAPERT

Se considera la figura de Seymour Papert (1928) como referente del pensamiento computacional. Fue un matemático, informático, educador y uno de los pioneros en la investigación del aprendizaje y la educación con dispositivos informáticos.

Basándose en las ideas constructivistas de Piaget este destacado científico computacional desarrolló la teoría del "construccionismo".



[Seymour Papert](#), por Gius195, bajo licencia [CC BY SA](#)

Ambos enfoques, el constructivismo de Piaget y el construccionismo de Papert, tienen como objetivo principal fomentar el aprendizaje activo y la construcción del conocimiento por parte del estudiante.

Por un lado, las aportaciones de Piaget se centran en la idea de que el aprendizaje debe estar basado en la experiencia y la interacción con el entorno, y que el estudiante debe ser un agente activo en su propio proceso de aprendizaje. Por otro lado, Papert introduce la idea de que la construcción de objetos físicos y la programación pueden ser excelentes herramientas para desarrollar ese aprendizaje. En el construccionismo además se destaca la importancia de los entornos de aprendizaje que fomenten la exploración y el descubrimiento, y que permitan al alumnado aprender a su propio ritmo y según sus propios intereses.

Es decir, el construccionismo de Papert se basa en las teorías constructivistas de Piaget, pero se enfoca en la creación de objetos físicos y la programación como herramientas para el aprendizaje activo, ambas teorías se complementan.

Papert también fue el creador del lenguaje de programación LOGO, que estaba diseñado para enseñar a los niños y las niñas a programar. Asimismo, Papert trabajó en el MIT (Instituto Tecnológico de Massachusetts) durante muchos años, fue miembro fundador del Media Lab del MIT y autor de varios libros, entre ellos "Mindstorms: Children, Computers, and Powerful Ideas", un clásico de la educación y la tecnología, y en el aparece por primera vez el término pensamiento computacional. Para Papert, que no llegó a definirlo, el Pensamiento Computacional sería el resultado de su enfoque construccionista de la educación.



[Antecedentes del pensamiento computacional: Seymour Papert](#), del canal de Youtube [Canal ISEP](#), bajo licencia [Youtube Estandar](#)



- WING



Jeannette Marie Wing (1956), es una teórica informática e ingeniera estadounidense. Ha sido una importante promotora del término pensamiento computacional y la autora de la primera definición de este.

En 2006 la doctora Wing lo definió en un artículo queriendo describir cómo piensa un científico de las ciencias de la computación y lo beneficios que esta forma de pensar podría tener para todo el mundo.

[Jeannette Wing, Davos 2013](#), por [Cropbot](#), bajo licencia [CC BY SA](#)

En su artículo [Computational Thinking](#), de 2006, Wing afirma que el pensamiento computacional es relevante en muchos ámbitos más allá de la programación y la informática, y que puede ser utilizado para resolver problemas en disciplinas como la Biología, la Física y la Economía. Además, explica por qué se trata de un proceso mental que involucra la formulación de problemas, la organización de datos, la abstracción y la creación de algoritmos para resolver problemas, una habilidad esencial en el momento histórico que vivimos.

También argumenta que el pensamiento computacional debería ser enseñado como una habilidad fundamental en la educación, desde la educación básica hasta la universidad.

En el citado artículo Wing resalta los siguientes aspectos sobre el pensamiento computacional:

- Pensar como un informático o una informática es más que ser capaz de programar un ordenador.
- Es una habilidad fundamental, es decir, algo que todo ser humano debe tener interiorizado para funcionar en la sociedad moderna.
- Es una forma de pensar propia de los humanos, no de los ordenadores, es un pensamiento creativo e imaginativo. Después se utilizarán los disposi-

tivos informáticos para apoyar esas soluciones creadas, pero la clave es nuestra inteligencia para abordar problemas.

- Complementa y combina el pensamiento matemático y de ingeniería.
- Se basa en ideas, no en "cosas". Da importancia a los conceptos computacionales que usamos en diferentes aspectos de la vida, no solo al *software* y al *hardware*.
- Su finalidad sería: "Para todo el mundo y en todas partes". Wing indica que el pensamiento computacional será una realidad cuando esté tan integrado, tan asimilado en nuestro modo de hacer las cosas, que ya no se vea como algo especial ni como una disciplina formal que deba enseñarse o explicarse de manera explícita.
- PAPER vs. WING

Con respecto al término pensamiento computacional del que hablaba Papert, este se centraba en el aprendizaje de la programación informática como un medio para desarrollar habilidades de pensamiento creativo y resolución de problemas con ordenadores. Él creía que la programación podía ser una herramienta poderosa para fomentar la exploración, la experimentación y el descubrimiento en niños y niñas, y que esta experiencia podía transferirse a otros aspectos de la vida. Papert ya decía que el pensamiento computacional no se trata solo de aprender a programar ordenadores, sino de usar la programación como una forma de aprender a pensar de manera creativa y estructurada sobre problemas complejos enfatizando la importancia de la construcción activa del conocimiento a través de la experiencia. En lugar de simplemente aprender sobre conceptos teóricos, los niños y las niñas pueden aprender haciendo, construyendo y experimentando con sistemas informáticos y desarrollar una comprensión más profunda y significativa de los conceptos de la informática y cómo aplicarlos a situaciones del mundo real.

Una idea fundamental en "Mindstorms: Children, Computers, and Powerful Ideas", de Papert, es "construir" por medios computacionales, versiones concretas de conceptos matemáticos o de la física más bien abstractos. De esta forma se construyen modelos mentales que permiten comprender el mundo, es decir, los ordenadores proporcionan una referencia concreta para que se entiendan mejor los conceptos abstractos.



En cambio, el enfoque de Wing parece, por el contrario, centrarse más en el modo de pensar que en el apoyo de medios informáticos. Aunque uso del ordenador y la programación están de fondo, resta importancia a dichos medios

En 2022, el artículo [The effectiveness of unplugged activities and programming exercises in computational thinking education: A Meta-analysis](#), de Feng Li et al., indaga acerca de la efectividad de cada uno de estos enfoques, y concluye que:

- Ambos enfoques son válidos para desarrollar el pensamiento computacional.
- El enfoque mediante actividades de programación resultó ser más eficaz.
- Mejor integrarlo en las diferentes materias que trabajarlo de forma aislada.
- Las actividades desconectadas deben enfocarse más a Educación Primaria y las de programación a Educación Secundaria.

3. Definición de pensamiento computacional

La **Dra. Wing** fue la primera persona en definir el concepto pensamiento computacional como: *Proceso de pensamiento para formular un problema y sus soluciones de manera que las soluciones sean representadas de una forma que pueden ser llevadas a un agente de procesamiento de información.*

Desde entonces han ido surgiendo otras definiciones y matizaciones. Algunas de ellas son:

Royal Society: *El proceso de reconocimiento de aspectos de la informática en el mundo que nos rodea, y aplicar herramientas y técnicas de la informática para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales.*

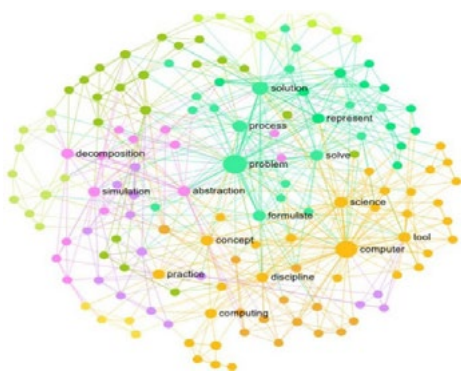
Alfred V. Aho: *El pensamiento computacional es el proceso que permite formular problemas de forma que sus soluciones pueden ser representadas como secuencias de instrucciones y algoritmos.*

Tedre y Denning: *Expresión popular que se refiere a un conjunto de ideas computacionales y hábitos mentales que las personas en disciplinas informáticas*

adquieren a través de su trabajo en el diseño de programas, software, simulaciones y cálculos realizados por máquinas.

Como se puede observar, muchas personas y entidades han propuesto diferentes definiciones para el pensamiento computacional, y no hay consenso al respecto. En algunos casos incluso se confunde el pensamiento computacional con la programación o la robótica.

En 2019, Jesús Moreno-León, Gregorio Robles, Marcos Román-González y Juan David Rodríguez García, publicaron el artículo [No es lo mismo: un análisis de red de texto sobre definiciones de pensamiento computacional](#) para estudiar su relación con la programación informática.



Este artículo analiza diferentes definiciones de pensamiento computacional para encontrar características comunes y entender mejor su relación con la programación.

A través de un análisis de red, se identificaron palabras clave y las conexiones entre ellas. Las palabras más importantes en estas definiciones, que actúan como puntos de enlace entre diferentes conceptos, son “problema”, “ordenador”, “solución” y “proceso”.

[Visual representation of the main topics and influential keywords in CT definitions](#), por Gregorio Robles, Marcos Román-González y Juan David Rodríguez García bajo licencia [CC BY SA NC](#)

El equipo resalta que en este análisis ni “programación”, ni “codificación” aparecen como palabras clave centrales en las definiciones estudiadas, pero en la práctica educativa suelen utilizarse casi como palabras sinónimas, probablemente porque la mayoría de las actividades diseñadas para desarrollar el pensamiento computacional utilizan actividades de programación. El artículo propone también una nueva definición de pensamiento computacional basada en los datos analizados:



La capacidad de formular y representar problemas para resolverlos utilizando herramientas y conceptos de la informática, como la abstracción y la descomposición.

Esta definición podría ser útil para aclarar que el pensamiento computacional es una habilidad cognitiva, mientras que la programación es solo uno de los métodos para desarrollarla.

4. Los pilares del pensamiento computacional

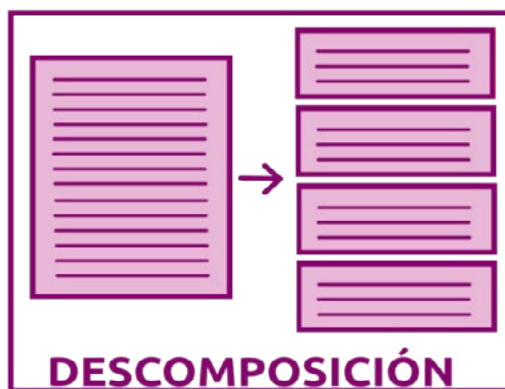
Una vez definido el concepto de pensamiento computacional, se presentan sus cuatro pilares básicos:

- **Descomposición:** el proceso de dividir un problema en partes más pequeñas y manejables. Esto implica analizar un problema complejo y dividirlo en tareas más simples y fáciles de entender.

Por ejemplo, si quieres organizar un viaje en coche a una ciudad que no conoces, puedes dividir el problema en tareas más pequeñas y manejables, como buscar posibles rutas en un mapa, planear las paradas en el camino, hacer reservas para el alojamiento, buscar información de los lugares y monumentos más importantes de ese sitio y organizar las actividades de cada día. Al descomponer el problema en partes más pequeñas, puedes hacer que planear el viaje sea algo menos abrumador, más manejable y disfrutable.

- **Abstracción:** el proceso de enfocarse en los aspectos importantes de un problema y eliminar los detalles innecesarios.

Por ejemplo, en algunas versiones de los planos del metro encontramos las diferentes líneas representadas de ma-



[Descomposición](#), por [lobotic](#), bajo licencia [CC BY](#)



[Abstracción](#), por [lobotic](#), bajo licencia [CC BY](#)

nera esquemática. Es decir, las distancias y la posición de las estaciones no son fieles a la realidad, pero nos da una visión clara de conjunto que nos facilita la tarea a la hora de elegir la mejor ruta para llegar a nuestro destino.

- **Reconocimiento de patrones:** la capacidad de encontrar patrones, tendencias y relaciones en la información. Esto implica encontrar similitudes y diferencias entre datos, identificar secuencias y establecer relaciones entre los datos.

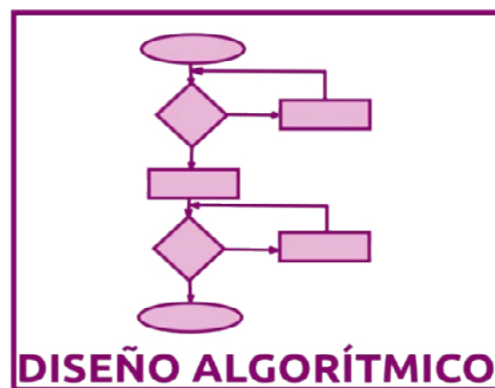
Por ejemplo, en algunos deportes, como el baloncesto, es habitual que el equipo técnico estudie grabaciones de los partidos de su rival para descubrir esos patrones y prever las acciones del equipo contrario. Al observar los movimientos de los jugadores pueden identificar patrones en la forma en que se pasan la pelota, corren o se posicionan en el campo, y de esa forma preparar a su equipo para anticipar las jugadas y tomar decisiones más rápidas y efectivas a fin de obstaculizar la estrategia del rival y aprovechar sus rutinas en su propio beneficio.



[ReconocerPatrones](#), por [lobotic](#),
bajo licencia [CC BY](#)

Diseño algorítmico: el proceso de desarrollar un conjunto de pasos lógicos para resolver un problema. Esto significa diseñar y desarrollar algoritmos que puedan ser implementados y ejecutados por una computadora para resolver el problema.

Por ejemplo, un algoritmo que utilizamos muy a menudo es el de la suma: para sumar dos números, se escriben uno debajo del otro. La cifra de las unidades del primer número se escribe debajo de la cifra de las unidades del segundo número. Las cifras de las decenas se colocan una debajo de la otra, y así sucesivamente para cada cifra. Luego, se suma la columna de las unidades. Si la suma es menor a 10, se escribe el resultado debajo de la línea. Si es mayor o igual a 10, se escribe



[DiseñoAlgorítmico](#), por [lobotic](#),
bajo licencia [CC BY](#)



sólo la cifra de las unidades del resultado debajo de la línea, y se lleva la cifra de las decenas a la siguiente columna.

Este algoritmo *que* hemos utilizado miles de veces a lo largo de nuestra vida nos facilita lograr el objetivo, es decir, calcular la suma.

El pensamiento computacional, apoyándose en estos cuatro pilares, permite afrontar problemas complejos de una forma más estructurada y lógica. Esto puede suponer un apoyo para la resolución de problemas en diversas áreas de la vida cotidiana, desde planificar cómo hacer la compra hasta elegir la mejor ruta para llegar a un destino o entender con precisión cómo seguir una receta de cocina. Las habilidades desarrolladas gracias al pensamiento computacional ayudan a abordar situaciones de forma eficiente y potencian tanto la creatividad como el análisis lógico. Esto la convierte en una herramienta fundamental para la educación y el desarrollo de competencias clave en la sociedad.



MÓDULO 2: ACTIVIDADES UNPLUGGED

1. ¿A QUÉ LLAMAMOS ACTIVIDADES DESENCUFADAS?

El Pensamiento Computacional Desenchufado (*Computational Thinking Unplugged* en inglés), trata de enseñar los principios del pensamiento computacional sin la necesidad de dispositivos electrónicos, como ordenadores, tabletas, etc. Es una forma de aprender a "resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática", como decía J. Wing, utilizando actividades sin tecnología.

El pensamiento computacional desenchufado enseña al alumnado los principios y habilidades de la programación y la informática, mediante actividades y juegos con materiales físicos (como cartas, papel, lápices, tableros, etc.) para favorecer la comprensión de conceptos como: la descomposición, la abstracción, el reconocimiento de patrones y la creación de algoritmos.

Este tipo de actividades permiten al alumnado interactuar de manera práctica con los conceptos abstractos que se usan en la programación, de una manera más tangible y accesible.

2. BENEFICIOS DEL PENSAMIENTO COMPUTACIONAL DESENCUFADO

El Pensamiento Computacional Desenchufado, además de introducir conceptos fundamentales de la informática en el aula, pueden ayudar al alumnado a pensar de manera estructurada y lógica, fomentando habilidades como la resolución de problemas, el trabajo en equipo y la creatividad. Algunos de los beneficios de trabajar este tipo de actividades en el aula son los siguientes:

- **Accesibilidad:** no se necesita tecnología. Uno de los principales problemas que enfrentan los centros educativos es la limitación de recursos, por lo que las actividades desenchufadas permiten trabajar los conceptos que se aprenderían con tecnología, pero usando únicamente materiales básicos como lápiz, papel, cartas o actividades físicas.



- **Sencillez:** estas actividades son fáciles de implementar y requieren pocos recursos fáciles de conseguir.
- **Fomenta el razonamiento lógico:** ayuda al alumnado a estructurar su pensamiento de manera lógica y fomenta habilidades esenciales como el pensamiento crítico, la resolución de problemas o los procesos de toma de decisiones.
- **Trabajo en equipo:** muchas de las actividades están diseñadas para desarrollarlas en pareja o en pequeño grupo, lo que fomenta la colaboración y el trabajo en equipo.
- **Conceptos de programación:** facilitan la adquisición de los principales conceptos utilizados en este campo.
- **Inclusividad y adaptabilidad:** las actividades desenchufadas suelen ser fácilmente adaptables a diferentes edades, niveles y estilos de aprendizaje.

3. EL MARCO CONCEPTUAL DE BRENNAN-RESNICK

El marco conceptual de Brennan-Resnick es una herramienta que nos puede ayudar a entender mejor el pensamiento computacional, cuáles son sus características, qué conceptos y procedimientos engloba, cómo abordar la evaluación de este... es decir, permite pasar de una definición abstracta y conceptual (sobre la cual, como hemos visto previamente, no hay consenso) a una definición más operativa.

3.1. Brennan y Resnick

[Mitchel Resnick](#) es profesor del [MIT Media Lab](#) y director del grupo [Lifelong Kindergarten](#), un grupo de trabajo que se enfoca en crear experiencias de aprendizaje que fomentan la creatividad, la colaboración y la resolución de problemas. Resnick y el grupo Lifelong Kindergarten son principalmente conocidos por su trabajo en [Scratch](#), un entorno de programación visual diseñado para enseñar al alumnado los conceptos fundamentales de la informática de manera divertida y accesible. Su enfoque pedagógico está basado en el constructivismo, influenciado por las ideas de [Seymour Papert](#), y defiende que el alumnado aprende mejor cuando crea y juega con su propio conocimiento.

Este enfoque enlaza con la idea de la creación de "productos significativos" que proponía Papert: creaciones que el alumnado produce a lo largo de un proceso de aprendizaje activo y que tienen valor porque están relacionadas con sus intereses, necesidades y experiencias, y no son simplemente una tarea para cumplir con un objetivo extrínseco.



Brennan-Resnick es una obra derivada de [Scratch@MIT 2012 - Day 1](#), por [ScratchEd Team](#), bajo licencia [CC BY](#)

[Karen Brennan](#) es profesora de práctica en la [Escuela de Posgrado en Educación de la Universidad de Harvard \(HGSE\)](#) y directora del [Creative Computing Lab](#). Además, está afiliada al departamento de [Ciencias de la Computación](#) de la universidad. Su trabajo de investigación se centra en el diseño de experiencias de aprendizaje en informática para alumnado de Educación Primaria y Educación Secundaria.

El enfoque de las experiencias de aprendizaje diseñadas por Brennan está basado en el construccionismo, de Papert, tratando de promover la autonomía del alumnado y la creación de comunidades a través de la personalización, la creación, el intercambio y la reflexión. Sus actividades de investigación y diseño se estructuran en tres áreas clave: investigar cómo el profesorado de Educación Primaria y Educación Secundaria diseña experiencias de aprendizaje en Informática, crear estructuras de apoyo para que el profesorado adopte enfoques construccionistas en la enseñanza de informática, y estudiar el desarrollo de la fluidez computacional en el alumnado más joven.



Antes de unirse a HGSE, Brennan completó su doctorado en el [MIT Media Lab](#), donde formó parte del equipo que desarrolla el entorno de programación [Scratch](#).

El marco Brennan-Resnick

El marco Brennan-Resnick clasifica las características observables del desarrollo del pensamiento computacional en tres grupos:

- **Conceptos:** son las ideas clave que se aprenden cuando se desarrolla el pensamiento computacional; conceptos que se aplican en la programación informática, es decir, **qué aprendemos**.
- **Prácticas:** se enfocan en los procesos y habilidades que se desarrollan, es decir, se relacionan con **cómo aprendemos**.
- **Perspectivas:** se refiere a **cómo nos relacionamos** con la tecnología, con otras personas y con el mundo, y a cómo aprendemos a usar esa tecnología de manera crítica y consciente.

Esta herramienta facilita el proceso de enseñanza-aprendizaje del pensamiento computacional, ya que define **qué aprender, cómo hacerlo y cómo expresarse y relacionarse** mediante la tecnología de forma crítica.

MARCO CONCEPTUAL DE BRENNAN-RESNIK		
CONCEPTOS	PRÁCTICAS	ACTITUDES
Secuencias	Desarrollo iterativo e incremental	Expresar
Bucles	Probar y depurar	Conectar
Paralelismo	Reutilizar y remezclar	Cuestionar
Eventos	Abstraer y modularizar	
Condicionales		
Operadores		
Datos		

Para profundizar en la comprensión del pensamiento computacional y su aplicación en el aula, a continuación, se propondrán una serie de actividades desenchufadas relacionadas con los conceptos del marco Brennan-Resnik y las diferentes fuentes de las que se han extraído, con el objetivo de dotar de un amplio banco de recursos para este tipo de actividades.

4. ACTIVIDADES DESENCUFADAS

4.1. Secuencias

El concepto de secuencia se refiere al orden lógico en el que se deben ejecutar unas instrucciones para resolver un problema computacionalmente, es decir, organizar acciones paso a paso para alcanzar un objetivo.

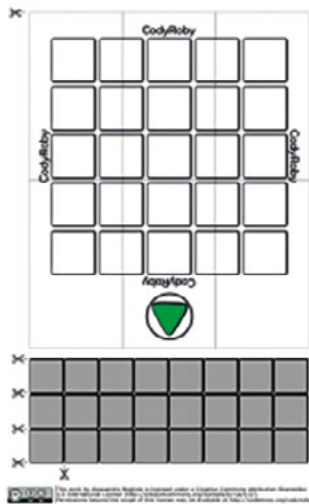
- CODY ROBY

[Cody Roby](#) es un conjunto de materiales *DIY* (del inglés, *Do It Yourself*, hacerlo uno/a mismo/a) y una serie de juegos que utilizan dichos materiales para desarrollar el pensamiento computacional de forma desenchufada.

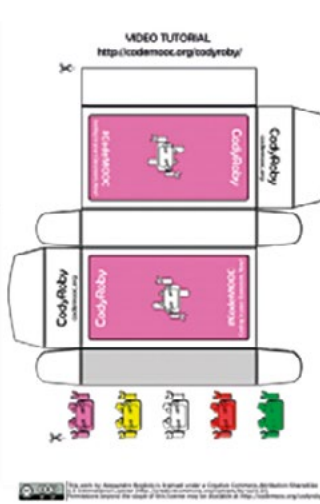
El kit de inicio imprimible incluye fichas de robot, una baraja de cartas con flechas para programar sus movimientos y un tablero, todo descargable para imprimir y recortar:

- **Tablero:** se suele utilizar uno de 5x5 casillas, pero puede crearse uno tan grande como necesites.
- **Cartas:** las básicas son "avanzar", "giro a la derecha" y "giro a la izquierda", aunque hay cartas especiales para juegos más avanzados.
- **Fichas:** en el kit vienen fichas de robots de distintos colores para representar a Roby.
- **Caja:** también incluye el desarrollo de una caja para guardar las tarjetas. No es necesaria para desarrollar las actividades, pero puede ser un buen complemento.

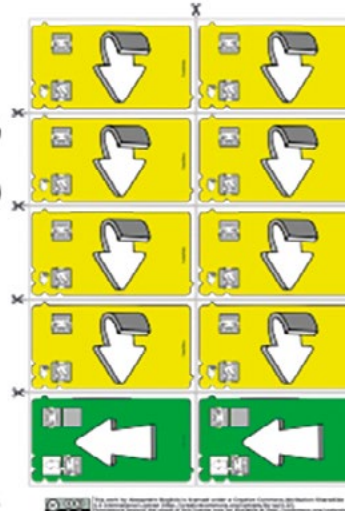




[Board](#), por Alessandro Bogliolo, bajo licencia [CC BY SA](#)



[BoxEC](#), por Alessandro Bogliolo, bajo licencia [CC BY SA](#)



[Turn-left-cards](#), por Alessandro Bogliolo, bajo licencia [CC BY SA](#)

Las actividades propuestas están basadas en la interacción entre dos personas con diferentes roles: **Cody**, quien da las instrucciones, y **Roby**, quien las ejecuta. Las instrucciones se representan mediante tres cartas básicas: avanzar, girar a la derecha y girar a la izquierda. En el siguiente vídeo se muestra cómo preparar los materiales y los juegos propuestos.



[How to build and use the CodyRoby starter kit](#), del canal de YouTube [bricoteens](#), bajo licencia [Youtube Estandar](#)

Además de estos juegos propuestos, este material puede utilizarse para realizar actividades de secuenciación de recorridos, hacer que el robot salga de un punto y llegue a otro pasando por determinadas casillas o evitando otras. Esto lo convierte en un recurso muy versátil y adaptable a cualquier temática o asignatura.

Propuestas de actividades con perspectiva de género para Cody Roby

Profesiones sin estereotipos

- **Etapas:** Educación Infantil.
- **Objetivo:** desmitificar profesiones tradicionalmente asociadas a un género.
- **Actividad:** crear un tablero con imágenes de profesiones diversas (ingeniera, enfermera, científica, cocinera...).
- **Desarrollo:** el alumnado debe programar a Roby para llegar a una casilla específica que tenga una profesión asignada. Al llegar, se propondrán preguntas sencillas para debatir sobre esa profesión, reflexionando si tienen estereotipos asociados y fomentando la idea de que todas las personas pueden acceder a cualquier profesión sin importar su género (¿Qué hacen las personas que realizan este trabajo? ¿Conocéis a alguna mujer que trabaje en esta profesión? ¿Y algún hombre? ¿Qué creéis que hay que aprender para llevar a cabo este trabajo? ¿A quién le gustaría trabajar en esta profesión? ...).
- **Reflexión final:** se hablará sobre la importancia de elegir una profesión en función de los intereses personales, y no por motivos de género.

Historias inclusivas con Roby

- **Etapas:** Educación Primaria (1º, 2º, 3º).
- **Objetivo:** crear historias con personajes diversos, sin estereotipos de género. **Desarrollo:** diseñar un tablero con diferentes casillas que representen elementos de una historia (personajes, objetos, escenarios...). El alumnado programará a Roby para avanzar por el tablero, integrando el elemento de la casilla en la que termina su movimiento en una historia que irán creando colectivamente, asegurándose de que sea inclusiva y no reproduzca estereotipos de género. Antes de comenzar, se leerán o re-



memorarán ejemplos de cuentos tradicionales y se debatirá sobre cómo podrían ser más inclusivos.

- **Reflexión final:** se hablará sobre la importancia de representar la diversidad en cuentos y relatos, destacando que todos los personajes, independientemente de su género, pueden ser valientes, inteligentes o líderes en sus historias.

En busca de referentes

- **Etapas:** Educación Primaria (4º, 5º, 6º).
- **Objetivo:** conocer figuras femeninas que han sido referentes en algún campo. **Desarrollo:** diseñar fichas del tamaño de la cuadrícula con imágenes de figuras femeninas que han sido referentes en algún campo y colocarlas en distintas casillas. El alumnado programará a Roby para llegar a cada casilla, donde descubrirá quién es el referente y qué hizo (por ejemplo, mediante un código QR por detrás de cada ficha). Al llegar a la casilla, se puede iniciar un breve diálogo con preguntas, como ¿Qué creéis que hace esta persona? Tras conocer su historia, se puede reflexionar mediante preguntas: ¿Por qué es importante lo que hizo? o ¿Cómo podemos ayudar nosotros a que haya igualdad?
- **Reflexión final:** hablar sobre la importancia de conocer a estas mujeres y cómo sus acciones han cambiado el mundo, destacando que todas las personas podemos contribuir a la igualdad en nuestra vida cotidiana.

4.2. Bucles

El bucle es un concepto del pensamiento computacional que consiste en repetir una misma secuencia varias veces para resolver un problema o una parte de este. Permite simplificar o clarificar el proceso además de ahorrar tiempo. Para poder trabajar con bucles es necesario identificar patrones.

- BAILANDO EN BUCLE

[Bailando en bucle](#), de la [recopilación de actividades desenchufadas](#) de la asociación [Programamos](#), es una estupenda actividad para enseñar el concepto de bucle de manera divertida y práctica, utilizando algo cotidiano y lúdico: un baile.

Ayudará a entender que, al identificar patrones de repetición en una secuencia de movimientos, se pueden agrupar las acciones que se repiten bajo un bucle, simplificando así el proceso.

Para iniciar, se pide a una persona voluntaria que dé una vuelta alrededor de una mesa o silla. Una vez que lo haga, se vuelve a dar la misma instrucción varias veces, exactamente con las mismas palabras. Tras la cuarta repetición, se pregunta al grupo si no habría sido más sencillo decir directamente que diera cuatro vueltas. Esta sencilla dinámica les ayudará a comprender que repetir instrucciones de manera literal puede ser tedioso, y que existen formas más eficientes de hacerlo, como los bucles.

Con esta idea en mente, se introduce el baile. Se reparte una ficha con los pasos a seguir y se practica con el grupo. Poco a poco, el alumnado irá interiorizando los movimientos y comenzará a notar los patrones de movimientos que se repiten.



Captura de pantalla de [Bailando en bucle](#), por [Programamos](#), bajo licencia [CC BY SA](#)



Captura de pantalla de [Bailando en bucle](#), por [Programamos](#), bajo licencia [CC BY SA](#)



Después se pide que trabajen en parejas para localizar las partes del baile que se repiten y se explica cómo se podrían usar bucles para representar esas repeticiones de manera más eficiente. Para cerrar la actividad, se fomenta la reflexión con preguntas como:

¿Qué sería más fácil: añadir más pasos al baile o simplemente cambiar el número de repeticiones en un bucle? Esto ayudará al alumnado a entender la utilidad de los bucles, especialmente en situaciones donde una acción debe repetirse muchas veces, mostrando cómo simplificar y optimizar procesos en cualquier ámbito de la vida.

Propuestas de actividades con perspectiva de género con bucles

Antón Pirulero

- **Etapas:** Educación Infantil.
- **Objetivo:** fomentar la igualdad de género asignando roles diversos y repetitivos en un juego colectivo.
- **Desarrollo:** se jugará una variación del juego de Antón Pirulero. Con todas las personas que van a jugar sentadas en círculo, una de ellas representará el papel de la "madre".

3 Antón Pirulero.
2 Cada cual que atienda su juego.
Y quien no lo aprenda
pagará una prenda.

Cada participante tendrá un papel con el "algoritmo" para cantar la canción: "Antón, Antón, Antón Pirulero. Cada cual, cada cual, ue atienda a su juego y quien no lo atienda, pagará una prenda":

Mientras cantan, irán dando golpecitos con el dedo en cada palabra de la canción, dando en el caso de Antón, tres golpes y en el de cada cual, dos. La "madre", desde el centro del círculo, irá señalando a cada participante y le transmitirá un gesto que represente una tarea. A partir de entonces, en lugar de los golpecitos en la tarjeta cambiará su "bucle" reproduciendo el gesto hasta que la "madre" transmita otro gesto a otra persona.

- **Reflexión final:** tras preguntar por las tareas que ha representado cada participante, se comentarán y se destacará que, al igual que en el juego, en la vida real todas las personas pueden aprender y realizar cualquier tarea.

Gincana: Misiones por la Igualdad

- **Etapas:** Educación Primaria (1º, 2º, 3º).
- **Objetivo:** desarrollar conciencia sobre la igualdad en la vida diaria a través de pequeñas pruebas que consisten en realizar bucles de ejercicios físicos.
- **Desarrollo:** dividido en pequeños grupos, el alumnado realizará de forma rotativa una serie de pruebas para ganar en cada estación una tarjeta con un lema que promueva la igualdad. Cada estación estará dirigida por una persona adulta que ayudará a interpretar el bucle si es necesario y controlará el correcto desarrollo de la actividad. Cuando un grupo termine en una estación, pasará a la siguiente. El desafío es realizar cada bucle todo el grupo a la vez, de forma sincronizada.
- **Reflexión final:** los grupos presentarán las tarjetas que han conseguido y discutirán cómo pueden promover la igualdad en su entorno escolar y familiar.



Ejemplos de pruebas y lemas:

El talento no tiene género



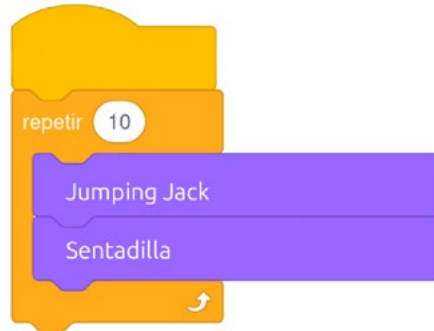
Iguals en derechos, iguals en oportunidades



Juntas y juntos, hacemos un mundo mejor



La igualdad comienza en nuestras propias acciones



Investigando en bucle

- **Etapas:** Educación Primaria (4º, 5º, 6º).
- **Objetivo:** resolver puzles con bucles para que una investigadora complete su trabajo hasta el gran descubrimiento.
- **Desarrollo:** el alumnado tendrá que resolver diferentes puzles de progra-

mación mediante un bucle de instrucciones. Para ello, tendrán los retos cuadrícula. En una casilla comenzará la investigadora. En otras casillas aparecerán diferentes descubrimientos, inventos o avances que esa investigadora haya logrado. El alumnado tendrá que descubrir y apuntar, mediante flechas, el bucle de instrucciones que le permitirá pasar por todas esas casillas, comenzando desde la investigadora. En cada reto aparecerá una breve biografía de la científica.

- **Reflexión final:** se comentará cómo la constancia es clave en el trabajo científico, y que cualquier persona, sin importar su género, puede lograr grandes descubrimientos. Además, se reflexionará sobre cómo históricamente las mujeres han enfrentado un techo de cristal que ha limitado su acceso y reconocimiento en el ámbito científico, y hoy en día es fundamental seguir promoviendo la igualdad de oportunidades para romper esas barreras.

Ada Lovelace

			Creación del primer algoritmo	
Publicación de sus notas			Visión de la computación más allá de los cálculos	

Ada Lovelace (1815-1852) fue una pionera de la informática, conocida por crear el primer algoritmo diseñado para una máquina, la **Máquina Analítica** de Charles Babbage. Fue visionaria al imaginar aplicaciones de las computadoras más allá de los cálculos matemáticos, como la creación de música y arte. Su legado quedó plasmado en las bases de la programación moderna. Su trabajo es considerado el primer ejemplo de software y destaca por su capacidad para prever el futuro de la computación.

Repetir (3) veces

↳ ↑ ↑ ↑ → □

Repetir (3) veces

↳ Repetir (3) veces

↳ ↑ □ □ □

↳ → □ □ □ □



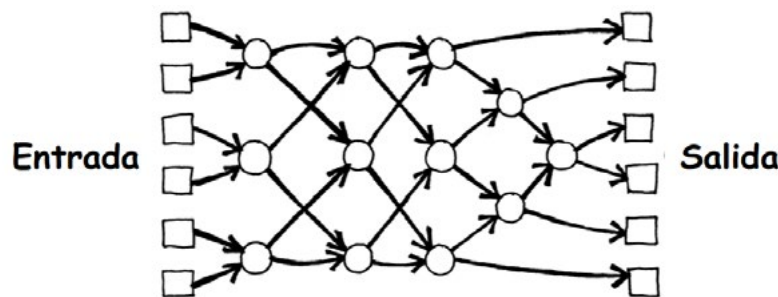
4.3. Paralelismo

El paralelismo es un concepto del pensamiento computacional que consiste en ejecutar varias tareas simultáneamente. Esto permite dividir un problema en partes independientes que se resuelven al mismo tiempo, facilitando tanto la programación como la depuración de esta en caso de que existan errores.

- GÁNALE AL RELOJ: REDES DE ORDENAMIENTO

Esta actividad se recoge en uno de los pilares básicos del pensamiento computacional desenchufado, el libro *CS Unplugged*, que puedes descargar en la web [Classic CS Unplugged](#), donde encontrarás muchos otros recursos. También puedes visitar la [nueva web](#) del proyecto.

Esta actividad muestra cómo las computadoras ordenan datos utilizando redes de ordenamiento, un sistema que permite realizar comparaciones en paralelo para optimizar el tiempo de resolución. Se trabajan conceptos de comparación, clasificación y paralelismo mediante un juego en el exterior que implica movimiento. Para empezar, hay que dibujar en el suelo, con tiza, una red de ordenamiento como la de la imagen.



[Red de ordenamiento](#), por Matt Powell, bajo licencia [CC BY-NC-SA](#)

A continuación, se reparten tarjetas con números al alumnado, que se agrupará en equipos de seis personas y se colocará en la posición inicial en la red con su tarjeta. Entonces comienzan a avanzar por las líneas, y al llegar a un punto de comparación (representado por un círculo), se muestran sus tarjetas. Quien tenga el número menor, avanza hacia arriba; quien tenga el mayor, hacia abajo. Todo el equipo repite el proceso, avanzando a la vez, hasta completar la red. La actividad incluye diferentes propuestas para dar mayor variedad y diversidad de dificultades.

Al finalizar la actividad, se propone destacar cómo las tareas en paralelo permiten optimizar procesos, tanto en computación como en la vida diaria, fomentando el trabajo colaborativo.

Propuestas de actividades con perspectiva de género con paralelismo

Macedonias paralelas

- **Etapas:** Educación Infantil.
- **Objetivo:** introducir el concepto de paralelismo y la corresponsabilidad mediante la preparación colaborativa de una receta sencilla.
- **Desarrollo:** se formarán tres tipos de grupos:
 - Una persona trabajará sola.
 - Dos personas colaborarán.
 - Cinco personas trabajarán juntas.

Cada grupo recibirá la misma cantidad de frutas ya peladas y cortadas en trozos grandes. Tendrán que corlarlas en trocitos pequeños con cuchillos desechables para preparar una macedonia en el menor tiempo posible. Se observará cómo los grupos más grandes completan la tarea más rápido.

- **Reflexión final:** se destacará que el trabajo en equipo y la cooperación permiten realizar tareas más rápido y de manera más eficiente, al igual que el reparto equitativo de tareas en la vida real... ¡mientras disfrutáis de la macedonia!

Investigando juntas

- **Etapas:** Educación Primaria (1º, 2º, 3º).
- **Objetivo:** introducir el concepto de paralelismo en programación y la importancia de la colaboración en el trabajo científico.
- **Desarrollo:** en parejas, el alumnado programará los movimientos de dos científicas, como Marie Curie y su hija Irène Joliot-Curie, pioneras en los descubrimientos relacionados con la radiactividad, a través de una cuadrícula. Deberán recoger pistas para completar su investigación y llegar a



la casilla de meta. Cada científica comienza en una de las esquinas inferiores. Cada pareja pensará y escribirá primero en una secuencia creada con flechas (avanzar, girar derecha y girar izquierda) para que cada una de las científicas recopile todas las pistas. Después, pensarán y escribirán la secuencia para recopilar las pistas que hay en su mitad de la cuadrícula. Luego, compararán las secuencias en solitario y en paralelo y escribirán una breve conclusión.

- **Reflexión final:** se analizarán las diferencias entre trabajar individualmente y hacerlo en equipo, destacando cómo el paralelismo de las tareas en conjunto puede mejorar la eficiencia, al igual que en la ciencia real, donde las investigaciones y los descubrimientos se logran con la colaboración entre diferentes personas.

Caligrafía colaborativa

- **Etapas:** Educación Primaria (4º, 5º, 6º).
- **Objetivo:** aplicar el concepto de paralelismo para fomentar la cooperación y la igualdad de género, utilizando un texto literario de una autora representativa.
- **Desarrollo:** el alumnado se agrupará en equipos de 4 personas. Es preferible que los equipos estén distanciados unos de otros para evitar ver lo que hace el resto. A unos equipos se les proporcionará una copia de un texto y un folio en blanco, a otros una copia del texto con líneas bien visibles dividiendo el texto en cuatro partes, y un foliocortado en cuatro partes del mismo tamaño que las partes señaladas en la copia del texto (uno a cada persona del grupo). Se les indicará que la tarea es copiar el texto lo más rápido posible, con buena letra y que cada persona del grupo debe participar copiando una parte.
- **Reflexión final:** se compararán los tiempos de cada equipo y se hablará de los procesos que han seguido para completar la tarea, destacando cómo el trabajo en paralelo (con la división de tareas) puede ser más rápido que el trabajo individual. Se reflexionará sobre cómo en la vida real las tareas compartidas permiten lograr más en menos tiempo, y también sobre la importancia de la colaboración sin importar el género de las personas que trabajan juntas.

4.4. Eventos

Un evento es una acción o suceso que hace que ocurra algo en un programa, es decir, se trata de una señal que indica que debe ejecutarse una secuencia de instrucciones solo cuando ocurre algo.

- EVERYBODY DANCE NOW

[Everybody Dance Now](#) es una actividad desenchufada perteneciente a la recopilación de [actividades desenchufadas de la Hora del Código](#) de Code.org, una plataforma educativa sin ánimo de lucro que promueve el aprendizaje de la programación y las ciencias de la computación que ofrece cursos y recursos gratuitos para enseñar conceptos básicos de programación. Además de esa recopilación, podéis encontrar otras actividades desenchufadas de Code.org [aquí](#).

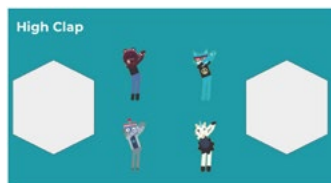
Esta actividad consiste en usar un mando de controlador remoto de papel para crear la coreografía de un baile. Comienza explicando el concepto de evento y poniendo ejemplos de eventos que provocan acciones en el mundo físico, como pulsar un interruptor para que se encienda la luz. Después, propone aprender una serie de pasos con los que el alumnado creará un baile con un mando a distancia con botones de diferentes formas. Para ello, cuenta con unos "tutoriales" de una página con una ilustración del movimiento en la que se ve claramente el color y la forma asociada y una explicación de los pasos que lo compone.



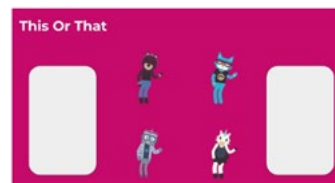
[Star](#), por [Code.org](#), bajo licencia [CC BY-NC-SA](#)



[Dab](#), por [Code.org](#), bajo licencia [CC BY-NC-SA](#)



[High Clap](#), por [Code.org](#), bajo licencia [CC BY-NC-SA](#)



[This or that](#), por [Code.org](#), bajo licencia [CC BY-NC-SA](#)





Una vez aprendidos los pasos, se elige una persona para dirigir la coreografía y un grupo de baile para ejecutarla. Con música de fondo, la persona a cargo de la coreografía pulsa los botones a intervalos para cambiar los movimientos. El grupo de baile debe reaccionar rápidamente, manteniendo el ritmo y siguiendo las indicaciones.

[The Big Event1](#), por [Code.org](#), bajo licencia [CC BY-NC-SA](#)

Propuestas de actividades con perspectiva de género con eventos

Juguetes para todos, juguetes para todas

- **Etapas:** Educación Infantil.
- **Objetivo:** introducir el concepto de evento y reflexionar sobre los estereotipos de género asociados a los juguetes.
- **Desarrollo:** el alumnado se agrupará en un círculo y se le mostrarán diferentes fotografías de juguetes: una muñeca, un coche, una comba, un balón, bloques de construcción...
- El grupo decidirá un gesto y un sonido para representar cada juguete, por ejemplo, para la muñeca, el gesto de mecer y el sonido "ea, ea"; para el coche, el gesto de conducir y el sonido "brum, brum"; para la comba, el gesto de saltar y como sonido el propio ruido del salto; para el balón, el gesto de chutar y el sonido "¡gol!"; para los bloques de construcción, el gesto de apilar con el sonido "clic, clic". La actividad comenzará practicando todos los gestos y sonidos en grupo, asociándolos correctamente a cada juguete. A continuación, se jugará de forma dinámica: se mostrará una imagen y el alumnado deberá realizar el gesto y sonido correspondientes. La actividad se irá complicando al aumentar la velocidad y cambiar rápidamente entre juguetes, o introduciendo nuevas fotografías.

- **Reflexión final:** al finalizar el juego, se preguntará si se han divertido realizando todos los gestos y sonidos. A continuación, se mostrará un catálogo de juguetes y se iniciará un debate: ¿Por qué algunos juguetes están etiquetados para niños y otros para niñas?
- ¿Es correcto que estén agrupados por género? Se fomentará la reflexión sobre la importancia de elegir juguetes en función de los intereses personales y no de estereotipos, destacando que cualquier persona puede jugar con cualquier juguete, independientemente de su género.

¡Controla tu cuento!

- **Etapas:** Educación Primaria (1º, 2º, 3º)
- **Objetivo:** fomentar la creatividad, trabajar el concepto de eventos y reflexionar sobre los roles de género en los relatos.
- **Desarrollo:** el alumnado se sentará en círculo. El/la docente o un alumno o alumna dirigirá la actividad utilizando un panel con botones hechos con cartulina que representen diferentes eventos que modificarán la historia colectiva. Cada botón tiene una función específica:
 - **Botón de personaje:** introduce un nuevo personaje.
 - **Botón de escenario:** indica que el personaje protagonista debe cambiar de lugar o escenario.
 - **Botón de misterio:** cambia el tono a una atmósfera intrigante.
 - **Botón de acción:** hace que la historia pase a ser más emocionante.
 - **Botón de miedo:** introduce elementos de suspense o miedo.
 - **Botón de humor:** cambia el tono a algo gracioso.
 - **Botón de cambio de género:** cambia el género del personaje protagonista.



La historia comenzará con una frase sencilla que establezca un punto de partida. Cada estudiante, siguiendo el orden del círculo, aportará un corto párrafo para continuarla cuando se pulse uno de los botones, modificando así el desarrollo del relato.

- **Reflexión final:** al finalizar la historia, se analizará cómo cambiaron los eventos y qué papel jugaron los diferentes personajes. Se reflexionará especialmente sobre el **cambio de género** del protagonista y si esto afectó o no a la historia, reflexionando si esos cambios fueron determinados por estereotipos de género. Tras destacar que el género no define las capacidades ni las aventuras que puede vivir una persona, se puede iniciar un nuevo relato más igualitario.

La Carrera Científica

- **Etapas:** Educación Primaria (4º, 5º, 6º)
- **Objetivo:** reflexionar sobre las dificultades generales y específicas que enfrentan hombres y mujeres en el ámbito científico, fomentando la empatía y el pensamiento crítico.
- **Desarrollo:** para llevar a cabo esta actividad, se preparará con cinta adhesiva de color una cuadrícula de 5x5 o 6x6 en el suelo que representará el recorrido de una carrera científica. En la cuadrícula se marcará un punto de inicio y otro de meta. Se colocarán estratégicamente folios con frases que implican obstáculos en el desarrollo de una carrera científica (falta de recursos para el proyecto, experimentos fallidos, problemas técnicos con los equipos, necesidad de repetir investigaciones...).

Un alumno comenzará el recorrido llevando un antifaz que le impedirá ver. El resto de la clase guiará a la persona con el antifaz, indicando cómo moverse por las casillas con las instrucciones **avanza, derecha e izquierda**, tratando de que llegue a la meta sorteando los obstáculos.

Una vez que el primer alumno haya llegado a la meta, el turno será para la siguiente persona del grupo. Esta vez será **una alumna** la que llevará el antifaz. En su recorrido, además de los obstáculos generales, se añadirán obstáculos específicos que reflejan las dificultades que las mujeres enfrentan en el ámbito científico. (te cuestionan por ser mujer, tienes que demostrar más que tus compañeros, te interrumpen en reuniones, no hay mujeres referentes en tu área...)

- **Reflexión final:** al acabar este segundo recorrido, se llevará a cabo una reflexión grupal sobre la experiencia vivida. Se le preguntará al alumnado cómo se sintió al enfrentarse a los obstáculos de la carrera científica, o viendo cómo se enfrentaba su compañera. Se reflexionará sobre las diferencias que hay entre los obstáculos que afectan a hombres y mujeres, y cómo estas barreras pueden influir en el desarrollo de las científicas. También se les pedirá que piensen en formas en las que se podrían mejorar las condiciones para las mujeres en la ciencia, favoreciendo la igualdad de oportunidades. Una vez hecho esto, se propondrá romper los obstáculos específicos de las mujeres en ciencia antes de continuar la actividad.

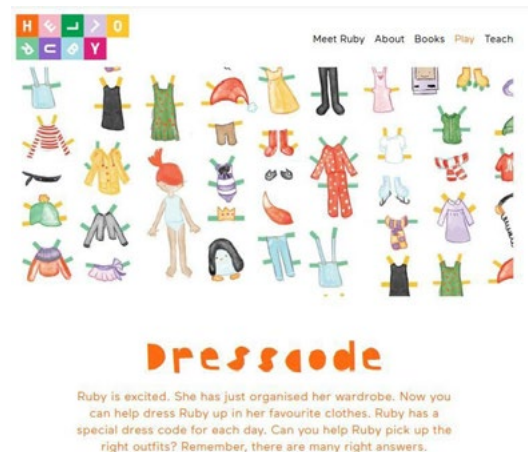
4.5. Condicionales

Un condicional es una instrucción que permite tomar decisiones según una condición específica. Indica que una acción se ejecutará solo si se cumple cierta situación. Con condicionales y bucles se pueden resolver un enorme abanico de problemas de programación.

- DRESSCODE

[Dresscode](#) es una actividad desenhufada diseñada por [Linda Liukas](#), recogida en su web [Hello Ruby](#).

En esta actividad, el alumnado ayudará a Ruby, la protagonista de los libros de Linda Liukas, a vestirse según las reglas de cada día. A través de muñecas de papel y prendas recortables, el alumnado se acercará a las estructuras condicionales de forma tangible.



Captura de pantalla de [Hello Ruby](#)

La propuesta comienza explicando que Ruby tiene un código de vestimenta especial para cada día. Después, se trabaja el concepto de selección con condicionales con la siguiente estructura:

Si [EVENTO], Ruby usa [TIPO DE ROPA]; si no, elige [OTRO TIPO DE ROPA].



El alumnado recortará y combinará diferentes prendas para vestir a Ruby según las condiciones dadas, creando bloques con los recortables donde se representará gráficamente cada decisión.

Esta actividad ayuda a desarrollar la lógica, la toma de decisiones y la comprensión de estructuras como “**si... entonces... si no...**”.

Propuestas de actividades con perspectiva de género con condicionales

Elige tu camino

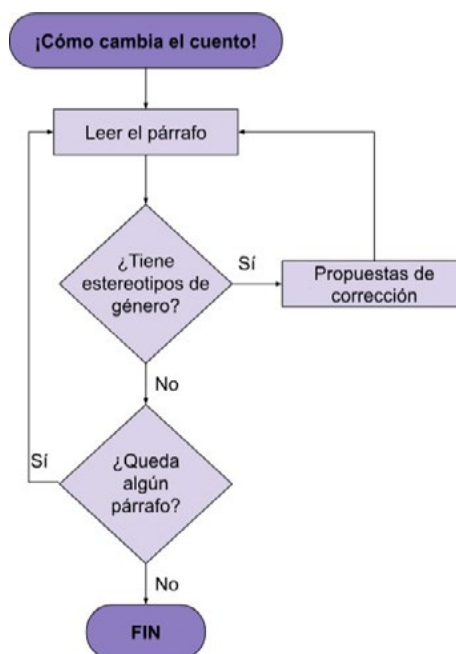
- **Etapas:** Educación Infantil.
- **Objetivo:** introducir el concepto de condicionales mediante elecciones basadas en intereses, fomentando la reflexión sobre la igualdad de opciones para todos y todas.
- **Desarrollo:** se creará un recorrido en el suelo con casillas grandes que representen un camino. En un punto intermedio habrá una bifurcación con un cartel que simboliza un condicional “Si/Si no”, en el que se podrán poner diferentes iconos o pictogramas para preguntar si una actividad les gusta o no, y con flechas que indican qué camino tocar en caso de que les guste o no.
- El alumnado caminará por el recorrido, eligiendo el camino en cada bifurcación según sus preferencias. Después de cada elección se pedirá al alumnado que observe su grupo y también el otro grupo. A continuación, se le preguntará a algunas personas por qué les gusta o no les gusta esa actividad. Después de unas cuantas elecciones se puede complicar la actividad ampliando el camino y poniendo en él varios condicionales para tener más de dos grupos al final del recorrido.



- **Reflexión final:** se conversará con el alumnado sobre las elecciones que hicieron destacando que no hay actividades “de niños” ni “de niñas” y que todo el mundo debe sentirse libre para elegir lo que más le gusta sin prejuicios.

¡Cómo cambia el cuento!

- **Etapas:** Educación Primaria (1º, 2º, 3º).
- **Objetivo:** trabajar el concepto de condicional mediante la creación colaborativa de un cuento sin estereotipos de género.
- **Desarrollo:** se leerá al grupo un cuento clásico. Después se preguntará si se han detectado estereotipos de género. Tras esta reflexión, se presentará el algoritmo de:
- ¡Cómo cambia el cuento!, que consiste en repetir cada una de las partes hasta que la mayoría de la clase esté de acuerdo con las modificaciones.



- **Reflexión final:** al finalizar la actividad se reflexionará sobre los cambios realizados y su influencia en la historia, centrándose en la importancia de cuestionar los roles asignados a los personajes en las historias, destacando cómo estos pueden influir en nuestra visión del mundo.

¿Cuánto colaboras?

Etapas: Educación Primaria (4º, 5º, 6º).

Objetivo: trabajar el concepto de condicional mediante la creación colaborativa de un *test* para medir la colaboración en tareas domésticas, fomentando la reflexión sobre la importancia de un reparto justo de tareas.



Desarrollo: el grupo se dividirá en parejas y cada una creará un condicional "Si/Si no", en el que se otorgará un punto si realiza regularmente una tarea doméstica, o una "prenda" si no lo hace.

Una vez hayan terminado, se formará un gran programa con todos los condicionales y el alumnado lo "ejecutará" a la vez que el/la docente lo va leyendo, apuntando sus puntos en una hoja de papel si realizan cada tarea que se menciona o pagando la "prenda" propuesta si no lo hacen.

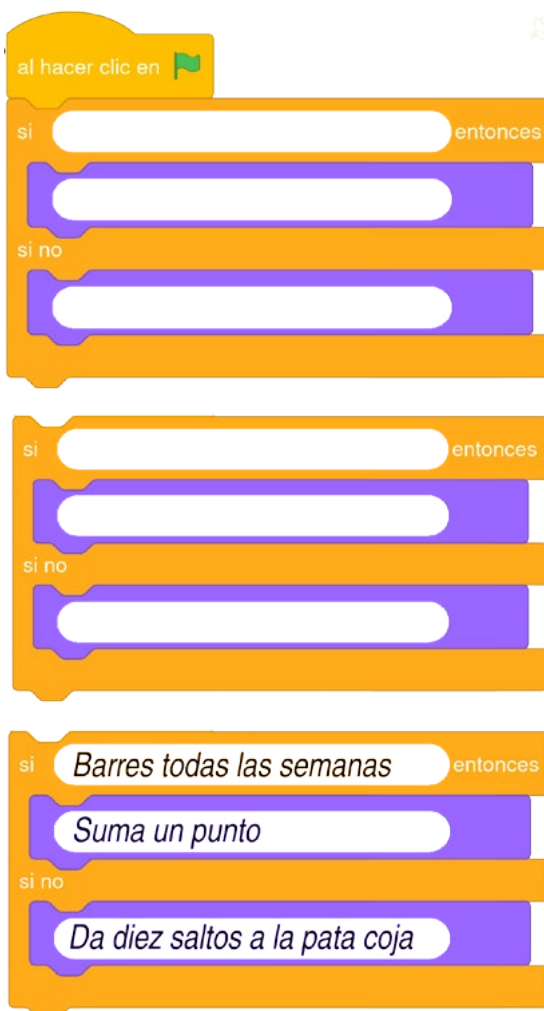
Reflexión final: al finalizar la actividad, se analizarán los resultados del *test* y se discutirán en grupo destacando la importancia de la corresponsabilidad en el hogar, reflexionando sobre cómo las tareas domésticas no tienen género y son responsabilidad de todas las personas que conviven.

4.6. Datos y operadores

Los **datos** son información que los programas utilizan para realizar acciones o tomar decisiones, como números, palabras o imágenes. Los **operadores** permiten trabajar con los datos mediante operaciones matemáticas y lógicas. En combinación, permiten que los programas realicen cálculos y comparaciones que facilitan la toma de decisiones.

MUJERES DE CIENCIA (IA)

Mujeres de ciencia es una actividad desenchufada perteneciente a la recopilación [Ejemplos de actividades Pensamiento Computacional Desconectado de INTEF](#)



para acercar al alumnado al concepto de aprendizaje automático (*machine learning*) a través de imágenes de científicas con características fácilmente identificables para construir un modelo de datos. Además, pretende fomentar la reflexión sobre el papel de las mujeres en la ciencia y destacar figuras femeninas históricas y ficticias.

La actividad comienza con una introducción al aprendizaje automático, explicando que el aprendizaje automático genera reglas para resolver problemas a partir de los datos proporcionados durante un proceso de entrenamiento.

					
Nombre	Hipatia	Amy	Katherine Johnson	Dra. Eleanor	Dra. Ryan Stone
¿Tiene gafas?	NO	SÍ	sí	NO	NO
¿Tiene pelo rizado?	SÍ	NO	sí	NO	NO
¿Pelo corto?	NO	NO	NO	NO	SÍ
¿Viste de astronauta?	NO	NO	NO	SÍ	SÍ
¿Tiene la piel blanca?	SÍ	SÍ	NO	SÍ	SÍ
¿Color de pelo?	Castaño	Castaño	Castaño	Rubio	Moreno

Imagen (5). por INTEF, bajo licencia CC BY-NC-SA

Después, se presenta a cinco mujeres relacionadas con la ciencia, a partir de cuyas imágenes y una batería de preguntas sencillas basadas en características físicas o de contexto, se crea una tabla que funciona como el *dataset* del modelo de aprendizaje automático.

El alumnado realizará entonces un proceso que simula el reconocimiento de patrones que realiza una IA: se introducen imágenes de las mismas personas o personajes y se aplican preguntas del modelo. Al comparar las respuestas de estas imágenes con las del conjunto de datos etiquetado se podrá identificar a quién corresponde cada una.

Esta actividad resulta muy apropiada para introducir de forma sencilla el concepto de aprendizaje automático y facilita que el alumnado comprenda cómo una IA



“aprende” a partir de ejemplos y genera reglas para clasificar nueva información. Se ha querido mantener esta actividad tal y como se encuentra en el repositorio, pero dado que la clasificación se apoya principalmente en rasgos físicos o de apariencia externa, que no son los más deseables a la hora de clasificar personas, se propone una adaptación de la actividad, construyendo el conjunto de datos a partir de características relacionadas a la actividad científica y al contexto de las mujeres representadas, sustituyendo las preguntas basadas en rasgos físicos por otras como:

- ¿Es un personaje ficticio?
- ¿Su trabajo está relacionado con las matemáticas o la informática?
- ¿Su trabajo está relacionado con la exploración espacial?
- ¿Realiza investigación científica teórica?
- ¿Realiza trabajo científico aplicado o experimental?
- ¿Desarrolla su labor en el espacio?

De este modo, la actividad no solo acerca al alumnado al funcionamiento del aprendizaje automático, sino que refuerza el objetivo de visibilizar a las mujeres en la ciencia desde lo que hacen y aportan.

Propuestas de actividades con perspectiva de género con datos y operadores

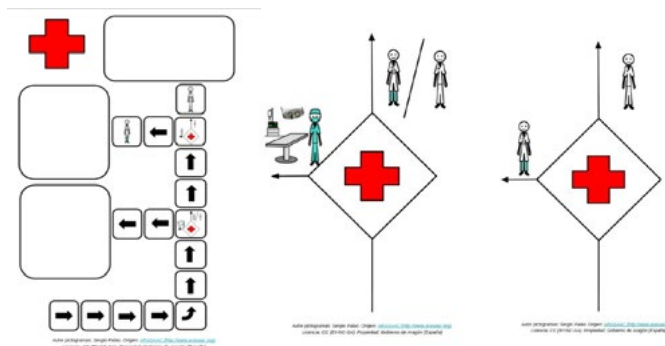
El hospital de mascotas

- **Etapas:** Educación Infantil.
- **Objetivo:** introducir los conceptos de datos y operadores a través de un juego de rol ambientado en un hospital, promoviendo la igualdad de género en profesiones.
- **Desarrollo:** se comenzará explicando que estamos en un hospital donde todos los niños y niñas serán quienes traigan a sus mascotas o pertenecerán al equipo médico del hospital.

Se hablará de que no todas las dolencias son iguales, por lo que el/la recepcionista tendrá que completar la ficha del paciente para que se le atienda lo mejor posible. Los campos para rellenar con unas etiquetas que se encontrarán clasificadas en cajas (variables), serán:

- Profesional que debe tratarlo (doctor/a, enfermero/a, cirujano/a...)
- Parte del cuerpo (las partes que se hayan trabajado en clase: cabeza, patas, ojos...)
- Dolencia (grave, media, leve).

En el suelo habrá marcado un recorrido con carteles en las intersecciones para llevar a las mascotas hasta el/la especialista que les debe atender (permitiendo así trabajar el concepto de condicional).



La persona que trae a la mascota recorrerá el camino siguiendo las indicaciones de la ficha hasta llegar a la sala donde atenderán a su mascota. En cada sala habrá dos personas que tratarán a las mascotas que les lleguen siguiendo el protocolo indicado:

- Si la dolencia es grave, vendarán la zona afectada.
- Si es media, pondrán una inyección.
- Si es leve, pondrán una tirita.

Se rotarán los roles hasta que todo el grupo haya sido personal sanitario en algún momento.

- **Reflexión final:** se resaltarà que se puede ser doctor/a, enfermero/a, cirujano/a sin importar el género. Se pueden introducir figuras como Marie



Curie o Florence Nightingale para mostrar referentes femeninas que han contribuido al avance de la medicina.

Tendiendo puentes hacia la igualdad

- **Etapas:** Educación Primaria (4º, 5º, 6º).
- **Objetivo:** trabajar los conceptos de datos y operadores a través de los logros de figuras históricas que promovieron la igualdad de género y de derechos.
- **Desarrollo:** tras explicar brevemente los conceptos de datos y operadores, se plantea al alumnado el reto de construir un puente hacia la igualdad, utilizando datos sobre figuras históricas y operadores básicos. El puente se representa en un tablero con casillas numeradas, que simbolizan los bloques que deben completar correctamente para avanzar en la construcción.

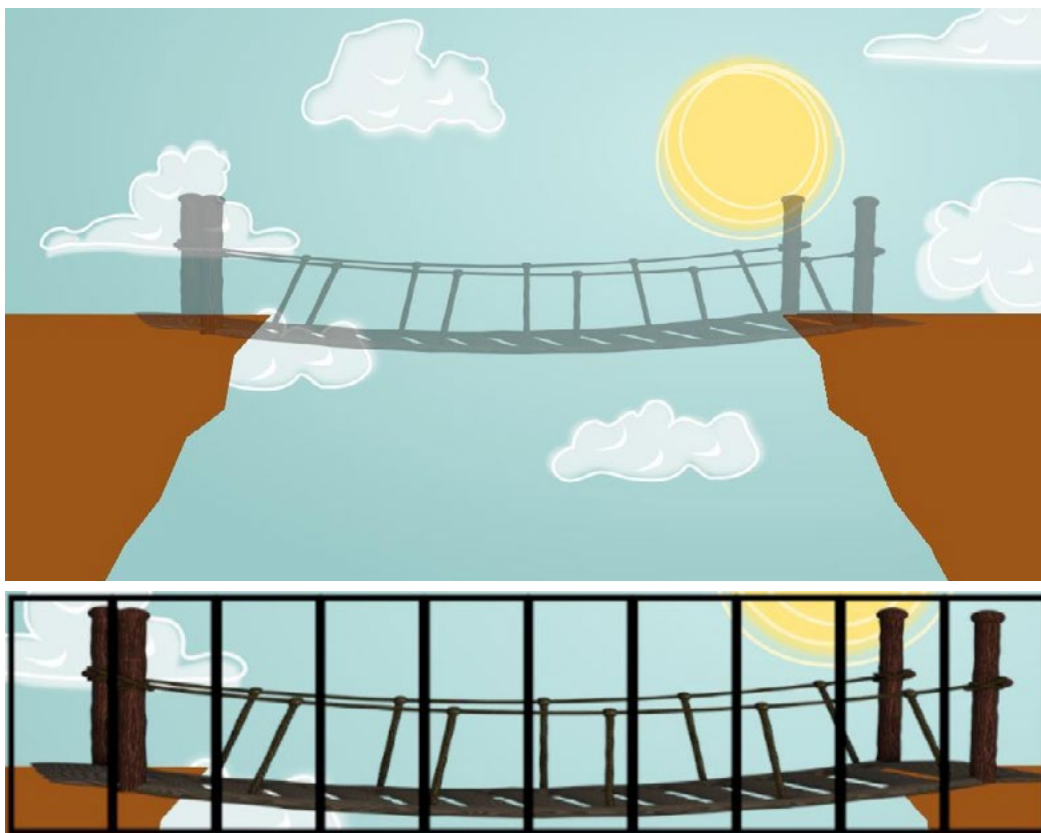
En un póster se mostrará el algoritmo para construir el puente. Se indicará al alumnado que deben conseguir diez hitos para tender un puente hacia la igualdad. Para lograr un hito, se debe relacionar correctamente una mujer con su contribución al avance en los derechos de las mujeres y la igualdad.

Se mostrarán después las tarjetas con nombres de mujeres y avances. Por turnos, el alumnado escogerá una tarjeta de cada lista y con *Bluetooth* las pegará al póster. Se preguntará al alumnado si creen que es correcto, y si



creen que no lo es, se le pedirá que indiquen cuál es el avance en derechos al que contribuyó la mujer elegida. Si es correcto, se sumará un punto a la variable puente y se añadirá una casilla de puente en el tablero.

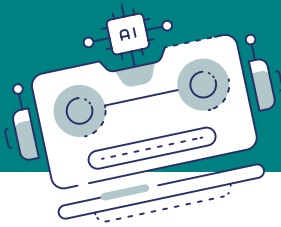
Reflexión final: tras completar el puente hacia la igualdad, el alumnado reflexionará sobre la importancia de los logros alcanzados por las figuras históricas mencionadas y cómo estos contribuyeron a construir un mundo más justo. Se discutirá cómo esos avances han impactado en nuestras vidas actuales y qué queda aún por hacer para seguir tendiendo puentes hacia la igualdad.



4.7. Repositorio de actividades desenchufadas

A lo largo de este bloque se han proporcionado diferentes actividades de varios repositorios diferentes. En este último punto de actividades desenchufadas, se presenta el listado a fin de facilitar el acceso a los mismos.

- [Cody Roby.](#)
- [Recopilación de actividades desenchufadas](#) de la asociación [Programamos.](#)
- [Classic CS Unplugged.](#)
- [CS Unplugged.](#)
- [Code.org CS Fundamentals Unplugged.](#)
- [Hello Ruby](#)
- [Ejemplos de actividades Pensamiento Computacional Desconectado de INTEF](#)



MÓDULO 3: SCRATCH

1. SCRATCH, UN ENTORNO DE PROGRAMACIÓN POR BLOQUES

[Scratch](#) es un entorno de programación por bloques creado específicamente para introducir al alumnado, especialmente al más joven, en el mundo de la programación de una manera accesible, creativa y divertida.

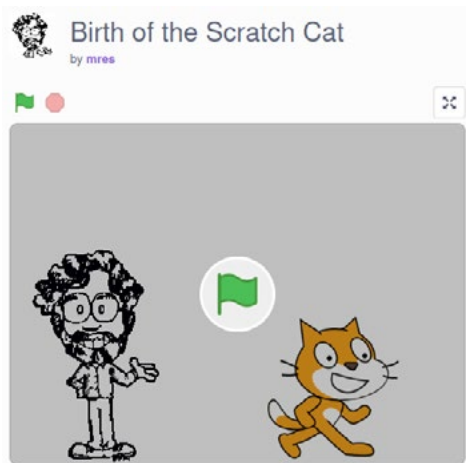
Esta herramienta tiene una interfaz sencilla que permite al alumnado crear historias interactivas, animaciones, juegos, música y mucho más, uniendo y anidando bloques de colores con las diferentes instrucciones en lugar de escribir código como en un lenguaje de programación tradicional.

Scratch es muy versátil, se pueden incluir actividades basadas en este entorno de programación en diferentes áreas, materias y grupos de edad, ya sea mediante creaciones del profesorado para enriquecer nuestras clases con otras formas de presentar o practicar los contenidos, o para que el alumnado pueda crear sus propios proyectos adquiriendo un rol más activo y adentrándose en el mundo de la programación.

El nombre "Scratch" se deriva de la técnica de *cratching*". Al igual que los y las *DJs* mezclan diferentes sonidos para crear algo nuevo, Scratch permite combinar bloques de código y elementos multimedia para crear sus propios proyectos como videojuegos, Animaciones, música y simulaciones...

El personaje principal de Scratch es un gato, y fue elegido por la conexión entre la palabra *scratch* (arañazo en una de sus acepciones) y el animal. En [este proyecto de Scratch](#) creado por Resnik se cuenta la historia de la famosa mascota.





Creación propia con recursos de [Scratch](#) bajo licencia [CC BY SA](#)



Captura de pantalla del proyecto [Birth of the Scratch Cat](#), de [mres](#)

2. HISTORIAS DE SCRATCH

Scratch fue desarrollado en el [MIT Media Lab](#) del Instituto de Tecnología de Massachusetts (MIT) por el grupo [Lifelong Kindergarten](#) bajo la dirección de [Mitchel Resnick](#).

Resnick, diseñó Scratch basándose en el trabajo de Papert y su visión constructora de la educación. Su intención era crear una herramienta que permitiera a niñas y niños no solo aprender a programar, sino también desarrollar habilidades de resolución de problemas y fomentar la creatividad y la colaboración.



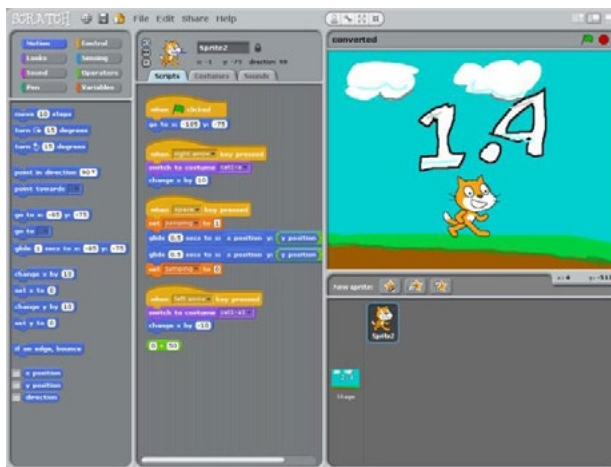
[Mitchel Resnick](#), por [Joi Ito](#), bajo licencia [CC BY SA](#)

Scratch se lanzó oficialmente en mayo de 2007 como una aplicación de escritorio. Con esta primera versión, además de crear juegos, historias interactivas y animaciones, las usuarias y usuarios ya podían subir sus proyectos al sitio web de Scratch para compartirlos con la comunidad. Esta posibilidad de compartir y colaborar en línea, ver el trabajo de otras personas para aprender unas de otras

y "remezclar" proyectos existentes, es decir, modificar proyectos de otras personas para hacer trabajos derivados, fue una de las características más revolucionaria del proyecto.

Esta primera versión ya incluyó en su última revisión, la versión 1.4, bloques para controlar motores y sensores de los kits de robótica de Lego.

El lanzamiento de Scratch 2.0 en 2013, supuso una importante expansión de la plataforma por una característica que facilitaba enormemente su uso, acceso y difusión: esta versión permitía programar directamente desde un navegador web, sin necesidad de descargar la aplicación. Esto impulsó el crecimiento de la comunidad, aumentando rápidamente el número de proyectos creados y compartidos. Scratch se popularizó rápidamente como una herramienta ideal para enseñar programación y pensamiento computacional en el aula.



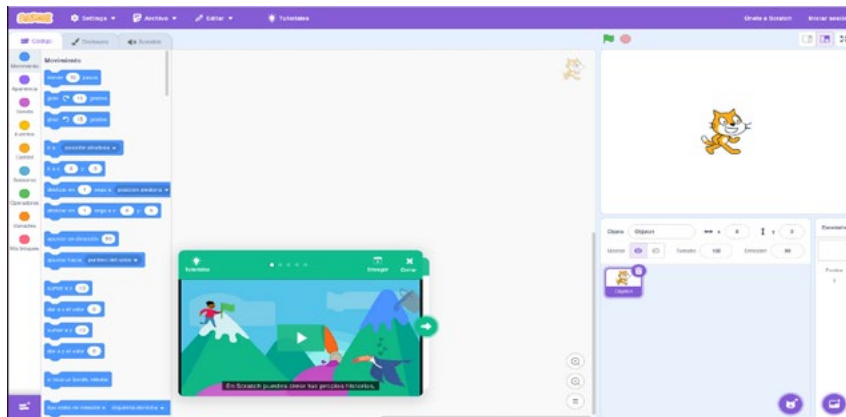
1.4 Editor, por [Evanzap](#), bajo licencia [CC BY SA](#)

Algo que sorprendió a educadores y educadoras que ya estaban trabajando en robótica educativa fue la ausencia de una extensión para controlar placas [Arduino](#), referente del *hardware* libre y utilizadas en muchísimos centros educativos.

Esa ausencia, que actualmente no se ha solucionado y no parece que vaya a corregirse en un futuro, se ha resuelto con *forks* de Scratch como [EchidnaScratch](#) o [mBlock](#), porque otra de las grandes virtudes de Scratch es que es *software* libre, por lo que diferentes personas y entidades han podido usar su base para crear nuevas herramientas adaptadas a necesidades específicas.

Actualmente, Scratch es la mayor plataforma de programación para niñas y niños. Está disponible en todo el mundo de forma gratuita, se usa en más de 200 países y territorios diferentes, y está disponible en más de 70 idiomas.





[Scratch layout](#), por [Kenny2scratch](#), bajo licencia [CC BY SA](#)

3. PRACTICAMOS CON SCRATCHJR

Existen muchas plataformas de programación basadas en bloques, pero una que queremos resaltar junto a Scratch es [ScratchJr](#).

ScratchJr se lanzó inicialmente como *app* para iPad en 2014, llegando a sistemas Android en 2015. La aplicación, como Scratch, tiene galerías de personajes y fondos variados, pudiendo además editarlos con una herramienta incluida en la misma aplicación. Destaca la posibilidad de incluir fotografías en los objetos mediante la cámara del dispositivo, especialmente en algunos personajes sin cara que están diseñados precisamente para este fin, lo que permite al alumnado convertirse el protagonista de sus proyectos.



Captura de pantalla de [ScratchJr](#).

Esta aplicación, inspirada en Scratch, sigue la filosofía de la anterior, pero facilita algunos aspectos de uso, resultando más accesible a alumnado de menos edad, estando recomendado para edades entre 5 y 7 años.

Por un lado, al ser una aplicación para dispositivos móviles, se facilita el uso gracias al mayor manejo y precisión que permite una pantalla táctil frente al uso del ratón. Además, ScratchJr cuenta con una interfaz más sencilla, en la que hay menos bloques de código que en Scratch y son de un tamaño más grande, lo que facilita las acciones de arrastrar y soltar los bloques. Otra ventaja es que en ScratchJr no es necesario saber leer, los bloques no tienen texto, únicamente iconos fáciles de reconocer.

La organización de bloques en la programación, en lugar de ser de arriba hacia abajo como en Scratch, es de izquierda a derecha, la dirección a la que el alumnado está más acostumbrado por ser la dirección de la lectoescritura. Por último, ScratchJr se ejecuta localmente y no tiene una comunidad en línea donde compartir los proyectos, siendo por ello un entorno más controlado.

Al igual que Scratch, ScratchJr invita a crear historias interactivas, animaciones o juegos sencillos donde se pueden utilizar personajes y escenarios prediseñados o personalizarlos de forma muy sencilla; y a pesar de que sus opciones son bastante limitadas por tener pocos bloques y pocas posibilidades de cambio de escenario, es una puerta de entrada ideal a los entornos de programación por bloques

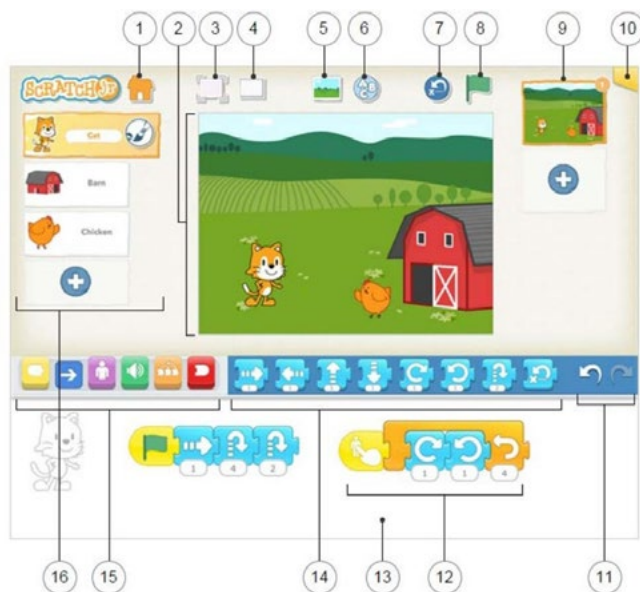


ScratchJr Interface, por [PrincessPandaWiki](#), bajo licencia [CC BY SA](#)



3.1. Interfaz de ScratchJr

Como hemos visto, la interfaz de ScratchJr. es sencilla y accesible. A continuación, veremos los diferentes iconos y espacios de esta:



Obtenido de [ScratchJr interface Guide](#)., por [ScratchJr](#), bajo licencia [CC BY SA](#)

- 1) **Guardar:** guarda el proyecto actual y sale a la página de inicio.
- 2) **Escenario:** aquí es donde ocurre la acción en el proyecto. Para eliminar un personaje, mantén pulsado sobre él.
- 3) **Modo de presentación:** expande el escenario a pantalla completa.
- 4) **Cuadrícula:** activar (y desactivar) la cuadrícula de coordenadas x-y.
- 5) **Cambiar fondo:** selecciona o crea una imagen de fondo para el escenario.
- 6) **Añadir texto:** escribe títulos y etiquetas en el escenario.
- 7) **Restablecer personajes:** restablece todos los personajes a sus posiciones iniciales en el escenario (arrastra los personajes para configurar nuevas posiciones iniciales).

- 8) Bandera verde:** inicia todos los *scripts* de programación que comienzan con un bloque de "Comenzar con la bandera verde" tocando aquí.
- 9) Páginas:** selecciona entre las páginas de tu proyecto o toca el signo más (+) para añadir una nueva página. Cada página tiene su propio conjunto de personajes y un fondo. Para eliminar una página, mantén pulsado sobre ella. Para reordenar las páginas, arrástralas a nuevas posiciones.
- 10) Información del proyecto:** cambia el título del proyecto y consulta cuándo fue creado.
- 11) Deshacer y rehacer:** si cometes un error, toca "Deshacer" para retroceder, revirtiendo la última acción. Toca "Rehacer" para revertir el último "Deshacer".
- 12) Script de programación:** une bloques para crear un *script* de programación, indicando al personaje qué hacer. Toca en cualquier parte del *script* para ejecutarlo. Para eliminar un bloque o *script*, arrástralo fuera del área de programación. Para copiar un bloque o *script* de un personaje a otro, arrástralo a la miniatura del personaje.
- 13) Área de programación:** aquí es donde conectas los bloques de programación para crear *scripts*.
- 14) Paleta de bloques:** este es el menú de bloques de programación. Arrastra un bloque al área de programación y tócalo para ver lo que hace.
- 15) Categorías de bloques:** aquí es donde puedes seleccionar una categoría de bloques de programación:
- a. Bloques de eventos (amarillo).
 - b. Movimiento (azul).
 - c. Apariencia (morado).
 - d. Sonidos (verde).
 - e. Control (naranja).
 - f. Bloques de finalización (rojo).
- 16) Personajes:** selecciona entre los personajes de tu proyecto o toca el signo más (+) para añadir uno nuevo. Una vez seleccionado un personaje, puedes editar sus *scripts*, tocar su nombre para renombrarlo o tocar el pincel para



editar su imagen. Para eliminar un personaje, mantén pulsado sobre él. Para copiar un personaje a otra página, arrástralo a la miniatura de la página.

3.2. Bloques de ScratchJr

ScratchJr utiliza bloques de programación sencillos que el alumnado puede arrastrar y conectar para controlar a los personajes y de esta manera crear animaciones, juegos o contar historias interactivas. Estos bloques están organizados por categorías (movimiento, apariencia, sonido...) y cada categoría tiene un color asociado, facilitando así su localización.

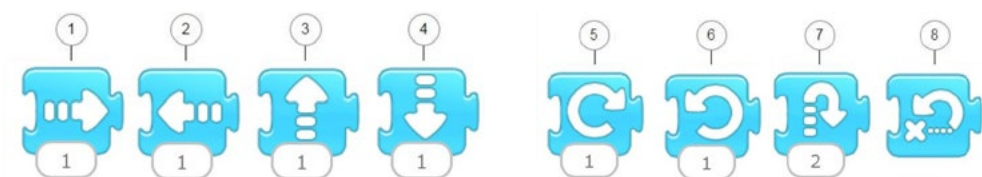
- BLOQUES DE EVENTOS (AMARILLO)



Obra derivada de [Scratch Jr Paint Editor Guide](#) y [Block descriptions](#),
por [ScratchJr](#), bajo licencia [CC BY SA](#)

- 1) **Al presionar bandera verde:** la secuencia de comandos se inicia al pulsar la bandera verde.
- 2) **Comenzar al pulsar:** la secuencia de comandos se inicia al pulsar sobre el personaje.
- 3) **Comenzar al tocar:** la secuencia de comandos se inicia cuando otro personaje toca al personaje.
- 4) **Comenzar con mensaje:** la secuencia de comandos se inicia cuando se envía un mensaje del color indicado.
- 5) **Enviar mensaje:** envía un mensaje del color indicado.

- BLOQUES DE MOVIMIENTO (AZUL)

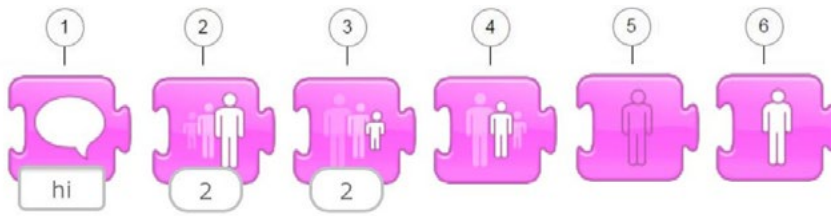


Obra derivada de [Scratch Jr Paint Editor Guide](#) y [Block descriptions](#),
por [ScratchJr](#), bajo licencia [CC BY SA](#)

- 1) **Mover a la derecha:** mueve el personaje el número de cuadrículas indicado hacia la derecha.
- 2) **Mover a la izquierda:** mueve el personaje el número de cuadrículas indicado hacia la izquierda.
- 3) **Subir:** Mueve el personaje el número de cuadrículas indicado hacia arriba.
- 4) **Bajar:** Mueve el personaje el número de cuadrículas indicado hacia abajo.
- 5) **Girar a la derecha:** gira el personaje la cantidad indicada en el sentido de las agujas del reloj. Para una vuelta completa, especificar 12 en la cantidad.
- 6) **Girar a la izquierda:** gira el personaje en la cantidad indicada en el sentido contrario a las agujas del reloj. Para una vuelta completa, especificar 12 en la cantidad.
- 7) **Saltar:** Mueve el personaje el número de cuadrículas indicado hacia arriba y después hacia abajo.
- 8) **Ir al inicio:** Vuelve a situar al personaje en la posición inicial. (Para establecer una nueva posición inicial, arrastra el personaje hasta el lugar deseado).



- BLOQUES DE APARIENCIA (MORADO)



Obra derivada de [Scratch Jr Paint Editor Guide](#) y [Block descriptions](#).
por [ScratchJr.](#) bajo licencia [CC BY SA](#)

- 1) **Decir:** aparece el mensaje indicado en un bocadillo por encima del personaje.
- 2) **Crecer:** aumenta el tamaño del personaje.
- 3) **Disminuir:** reduce el tamaño del personaje.
- 4) **Restablecer tamaño:** devuelve el personaje a su tamaño original.
- 5) **Ocultar:** desvanece el personaje hasta que lo hace invisible.
- 6) **Mostrar:** el personaje aparece gradualmente hasta que es completamente visible.

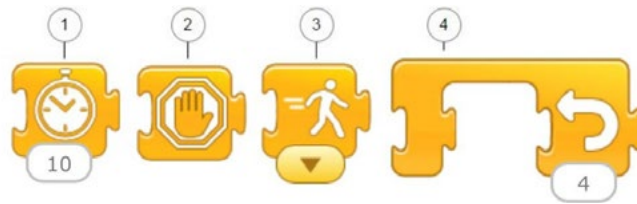
- BLOQUES DE SONIDOS (VERDE)



Obra derivada de [Scratch Jr Paint Editor Guide](#) y [Block descriptions](#).
por [ScratchJr.](#) bajo licencia [CC BY SA](#)

- 1) **Pop:** reproduce el sonido "pop".
- 2) **Reproducir sonido grabado:** reproduce un sonido grabado por el usuario.

- BLOQUES DE CONTROL (NARANJA)



Obra derivada de [Scratch Jr Paint Editor Guide](#) y [Block descriptions](#),
por [ScratchJr](#), bajo licencia [CC BY SA](#)

- 1) **Esperar:** pausa la secuencia de comandos durante el tiempo indicado (en décimas de segundo).
- 2) **Parar:** detiene todas las secuencias de comandos del personaje.
- 3) **Fijar velocidad:** cambia la velocidad a la que se ejecutan determinados bloques.
- 4) **Repetir:** ejecuta los bloques un número indicado de veces.

- BLOQUES DE FINALIZACIÓN (ROJO)



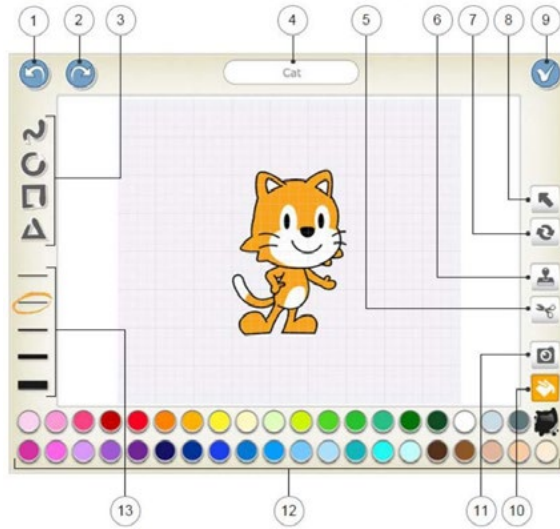
Obra derivada de [Scratch Jr Paint Editor Guide](#) y [Block descriptions](#),
por [ScratchJr](#), bajo licencia [CC BY SA](#)

- 1) **Finalizar:** indica el final de la secuencia de comandos (pero no la afecta de ningún modo).
- 2) **Repetir indefinidamente:** ejecuta la secuencia de comandos una y otra vez.
- 3) **Ir a la página:** cambia a la página del proyecto indicada.



3.3. Editor de imagen de ScratchJr

Una de las características más interesantes de ScratchJr es la posibilidad de personalizar los objetos y escenarios. La aplicación tiene integrado un editor de imágenes que permite crearlas desde cero o modificar las existentes.



Obtenido de [Scratch Jr Paint Editor Guide](#), por [ScratchJr](#), bajo licencia [CC BY SA](#)

- 1) **Deshacer:** revierte el cambio más reciente.
- 2) **Rehacer:** revierte el último "Deshacer".
- 3) **Forma:** elige una forma para dibujar: línea, círculo/elipse, rectángulo o triángulo.
- 4) **Nombre del personaje:** edita el nombre del personaje.
- 5) **Cortar:** después de seleccionar la herramienta de cortar, puedes tocar un personaje o forma para eliminarlo del lienzo.
- 6) **Duplicar:** después de seleccionar la herramienta de duplicar, puedes tocar un personaje o forma para crear una copia de él.
- 7) **Rotar:** después de seleccionar la herramienta de rotar, puedes rotar un personaje o forma alrededor de su centro.

- 8) **Arrastrar:** después de seleccionar la herramienta de arrastrar, puedes mover un personaje o forma en el lienzo. Si tocas una forma, puedes editarla arrastrando los puntos que aparecen.
- 9) **Guardar:** guarda los cambios y sal del editor de pintura.
- 10) **Relleno:** después de seleccionar la herramienta de relleno, puedes tocar cualquier sección de un personaje o forma para rellenarla con el color seleccionado.
- 11) **Cámara:** después de seleccionar la herramienta de cámara, puedes tocar cualquier sección de un personaje o forma, y luego tocar el botón de la cámara para rellenar la sección con una nueva foto tomada con la cámara.
- 12) **Color:** selecciona un nuevo color para usar al dibujar y rellenar formas.
- 13) **Grosor de la línea:** cambia el grosor de las líneas en las formas que estás dibujando.

3.4. Programando en ScratchJr

Ahora que conoces la aplicación es hora de probarla. Mira estos dos vídeos que te ayudarán a crear tu primer proyecto.



[Scratch Jr: Descargando la app y Conociendo la Interfaz, 2022](#), del canal de Youtube [PERO PROFE](#) bajo licencia [Youtube Estandar](#)





[Scratch Jr: creando mi primer juego interactivo](#), del canal de Youtube [PERO PROFE](#) bajo licencia [Youtube Estandar](#)

Con estos dos vídeos puedes empezar a crear con ScratchJr., pero quizás te apetezca seguir viendo algunos ejemplos. A continuación, encontrarás dos listas de vídeos con los que podrás ampliar tu conocimiento sobre esta plataforma:

- [SCRATCH JR: Aprendiendo a crear animaciones y juegos para principiantes.](#)
- [Curso de Scratch Jr. para Maestros.](#)
- **IDEAS PARA UTILIZAR SCRATCHJR EN EL AULA**

Seguro que a lo largo de esta unidad se te han ido ocurriendo diferentes ideas de cómo utilizar ScratchJr. en el aula, pero si te apetece ver cómo lo han hecho docentes de muchos y diferentes lugares puedes visitar [ScratchJr Connect](#), una base de datos gratuita y supervisada para compartir recursos, lecciones y proyectos de ScratchJr; en la que cualquier persona se puede registrar de forma gratuita y presentar un recurso o proyecto para que, después de ser aprobado, se agregue a la base de datos.

4. Practicamos por ScratchJr

Scratch se encuadra en el enfoque de **suelo bajo, techo alto y paredes anchas** basado en el construccionismo de S. Papert.

M. Resnick y su equipo aplicaron estas ideas al desarrollo de Scratch con el objetivo de que fuera un entorno de aprendizaje accesible para todo el mundo (suelo bajo), que favoreciera el ampliar el conocimiento y su aplicación a quienes avancen más (techo alto), y que permitiera diversas formas de abordar problemas y proyectos (paredes anchas). En el caso de Scratch podríamos ver este enfoque, por ejemplo, en estos aspectos:

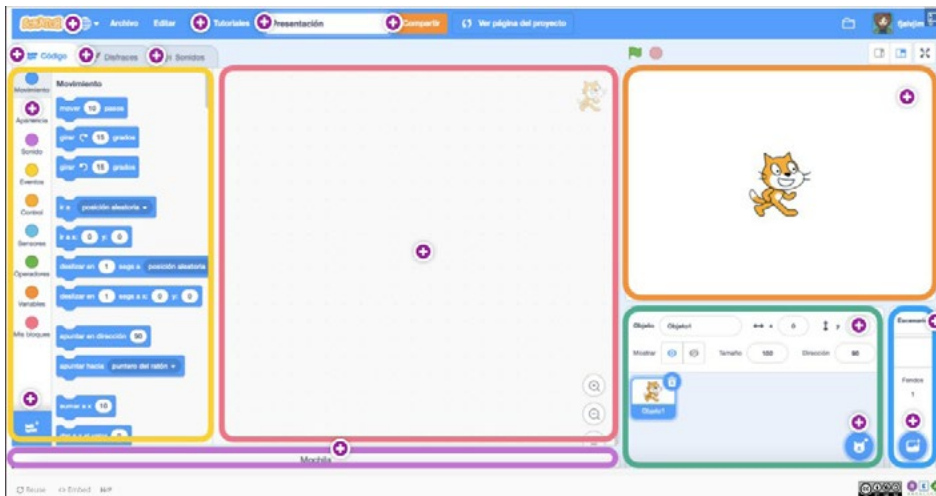
- **Suelo bajo:** Scratch tiene una interfaz visual intuitiva que permite que el alumnado empiece a programar con facilidad. Los bloques de código son fáciles de coger, arrastrar y soltar, y la forma de anidarlos es sencilla. Además, no es necesario tener conocimientos previos de programación para empezar a trabajar con este programa: un proyecto sencillo, como hacer que un personaje se mueva o cambie de color, es accesible para principiantes y suficientemente llamativo como para animar a seguir usando la herramienta.
- **Techo alto:** aunque Scratch es fácil de usar, ofrece una gran profundidad. El alumnado más avanzado puede crear proyectos complejos, como videojuegos interactivos, simulaciones científicas, o historias animadas con múltiples escenas. Scratch incluye la creación de bloques personalizados, uso de variables, sensores y otras opciones que permiten trabajar en proyectos más complejos y aprender nuevos conceptos.
- **Paredes anchas:** Scratch permite múltiples formas de abordar un proyecto, por lo que es ideal para facilitar un aprendizaje más diverso. El alumnado puede elegir entre un gran abanico de tipos de proyectos y hacer que sean más o menos complejos. Además, cada estudiante puede personalizar su proyecto según sus intereses eligiendo personajes, escenarios, sonidos, creando sus propios sonidos o imágenes, etc., lo que amplía las posibilidades creativas y permite que cada persona encuentre su forma de expresarse.



4.1. Interfaz de Scrath

La interfaz de Scratch es muy intuitiva y está diseñada para facilitar la programación a través de bloques de colores que se arrastran y sueltan, uniéndose y anidándose.

A la izquierda, encontramos la paleta de bloques, que agrupa los mismos por categorías (como Movimiento, Apariencia, Control...). En el centro está el área de programación, donde se conectan los bloques unos a otros para crear secuencias de programación. A la derecha, se encuentra el escenario, que muestra la animación o interacción de los personajes, es decir, donde vemos el resultado de nuestro proyecto. Debajo del escenario están los menús donde se pueden añadir o crear personajes y fondos. A continuación, puedes ver la interfaz, sus espacios y una breve explicación.



[Interfaz de Scratch](#), por [F. Javier Álvarez](#), bajo licencia [CC BY-SA](#)

4.2. Programando en Scratch

Los bloques de Scratch, además de ser la mayoría muy intuitivos, son descriptivos, tienen texto con el comando que representan, por lo que es muy sencillo utilizarlos.

Como ya hemos comentado, las posibilidades de Scratch son enormes, incluyendo bloques que permiten crear proyectos muy completos y de enorme complejidad, pero por tratarse este de un curso de iniciación, nos centraremos únicamente en los básicos de algunos tipos de bloques.



Introducción a Scratch 3.0: Primeros pasos



[Primeros pasos con Scratch 3.0](#), del canal de Youtube [INTEF](#) bajo licencia [CC BY](#)



Introducción a Scratch 3.0: Creación de objetos



[Crear objetos. Introducción a Scratch 3.0](#), del canal de Youtube [INTEF](#) bajo licencia [CC BY](#)



Introducción a Scratch 3.0: Propiedades de los objetos



[Propiedades de los objetos en Scratch 3.0](#), del canal de Youtube [INTEF](#) bajo licencia [CC BY](#)





[Cómo agregar bocadillos de diálogo y ejecutar disfraces. Scratch 3.0.](#)
del canal de Youtube [INTEF](#) bajo licencia [CC BY](#)

Con estos vídeos puedes empezar a crear proyectos sencillos con Scratch, pero si te apetece seguir profundizando, a continuación encontrarás dos recursos con los que podrás ampliar tu conocimiento sobre esta plataforma:

- [Curso abierto: Pensamiento computacional en el aula con Scratch.](#)
- [Lista de Youtube: Curso tutorizado Pensamiento computacional en el aula con Scratch.](#)

4.3. Ideas para utilizar Scratch en el aula

Al igual que en el caso de ScratchJr, existen diferentes recursos que puedes consultar para ver propuestas de aplicación de Scratch en el aula. Algunos de ellos son: Scratch Educator Guides: Scratch ofrece una sección para educadores donde se pueden encontrar planes de lecciones adaptadas a diferentes niveles educativos y temáticas. Además, incluye tutoriales en vídeo para enseñar diferentes conceptos de Scratch, tanto básicos como más avanzados. También se pueden descargar materiales como guías o fichas de actividades, algunas de las cuales se pueden usar directamente en el aula.

Enlace: <https://scratch.mit.edu/educators/>

ScratchEd: se trata de una comunidad creada por el equipo de Scratch donde los y las docentes pueden compartir ideas, recursos, proyectos y experiencias sobre el uso de Scratch en el aula. Aquí podrás encontrar diferentes recursos, como casos de éxito, planes de actividades, ejemplos de proyectos... La comunidad también organiza eventos y talleres.

Enlace: <http://scratched.gse.harvard.edu/>

Scratch Wiki: enorme repositorio de información sobre el uso de Scratch donde se puede encontrar guías, consejos, información, ejemplos de proyectos...

Enlace: <https://en.scratch-wiki.info/>

Code Club: es una plataforma gratuita que ofrece proyectos organizados por niveles de dificultad para enseñar programación con Scratch. Incluye guías paso a paso para crear juegos, animaciones o historias interactivas.

Enlace: <https://projects.raspberrypi.org/en/collections/scratch>

CS First: una iniciativa de Google que ofrece actividades orientadas a la enseñanza de Ciencias de la Computación utilizando Scratch. Incluye guías bien organizadas con temáticas diversas, muy prácticas para integrar Scratch en diferentes áreas curriculares. Enlace: <https://csfirst.withgoogle.com/s/es/home>

5. CREANDO UN ENTORNO INCLUSIVO EN SCRATCH Y SCRATCHJR

Como ya vimos el módulo 1, una parte importante a la hora de crear un entorno inclusivo en el aula recae sobre la propia actitud del docente o la docente, por lo que es imprescindible permanecer alerta y autoevaluarse continuamente.

Procura, tanto a la hora de programar tus ejemplos como a la de proponer proyectos al alumnado, que estos sean inclusivos y libres de estereotipos y que permitan diseñar animaciones, historias o juegos que no estén asociadas a roles de género tradicionales. Algunas temáticas con muchas posibilidades son aquellas relacionadas con la naturaleza, el arte, la resolución de problemas sociales o medioambientales o historias sobre su entorno cercano o familiar, ofreciendo opciones para que cada estudiante elija según su interés.



Si eliges agrupar al alumnado para trabajar en equipo, una opción muy recomendable en trabajos de programación, intenta formar equipos equilibrados en términos de género y habilidades, asegurándote de que cada estudiante tenga la oportunidad de expresar sus opiniones en un entorno receptivo y seguro con propuestas de funcionamiento, como el establecimiento de turnos de palabra, la asignación de roles rotativos dentro del equipo para que todos sus miembros tengan la oportunidad de liderarlo, o contribuir a diferentes partes del proyecto.

No olvides utilizar un lenguaje inclusivo. Es importante evitar términos o ejemplos que refuercen estereotipos. Anima al alumnado a crear personajes y escenarios que representen diversidad, tanto en género como en cultura, para que todos y todas se sientan reflejadas.

Da retroalimentación y reconocimiento de manera equitativa, asegurándote de destacar los logros de todo el mundo. Señala y celebra tanto los avances técnicos como los creativos, valorando la participación de todo el alumnado.

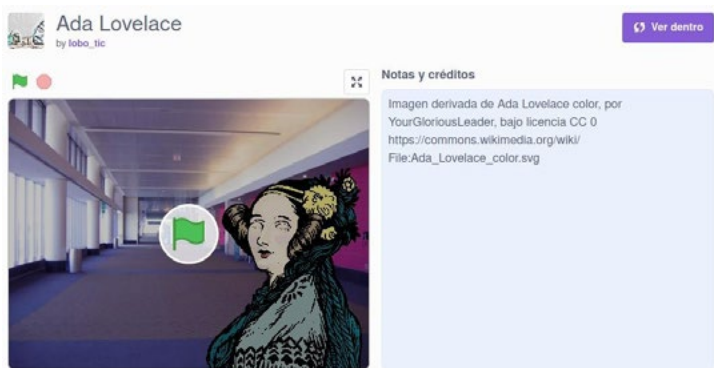
Contextualiza de manera inclusiva, es decir, relaciona las actividades con situaciones del día a día que sean cercanas para todo el alumnado. Podrías pedirles que programen historias sobre su comunidad, familia, o temas sociales que importen a todos y todas, como el medio ambiente.

Y, por supuesto, no olvides utilizar referentes diversos. Presenta ejemplos que sean mujeres que trabajan en programación y robótica, mostrando que todos y todas pueden sobresalir en estas áreas. Una propuesta de trabajo podría ser crear proyectos basados en estos referentes, como en los siguientes ejemplos:



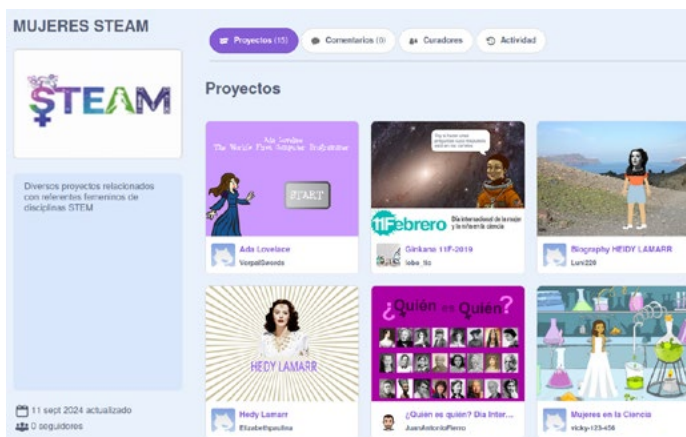


Proyecto Ada Lovelace con Scratch Jr, del canal de Youtube [Jorge Lobo](#) bajo licencia [CC BY](#)



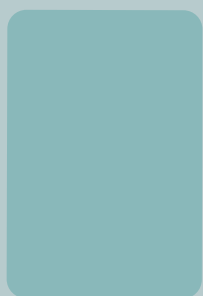
[Ada Lovelace](#), por [Lo-botic](#), bajo licencia [CC BY SA](#)

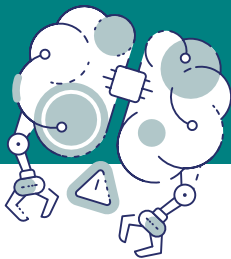
Si te has quedado con ganas de más, puedes visitar el [estudio Mujeres STEAM](#) de Scratch con más proyectos, algunos muy completos e interesantes.



Captura de pantalla de [Scratch](#).







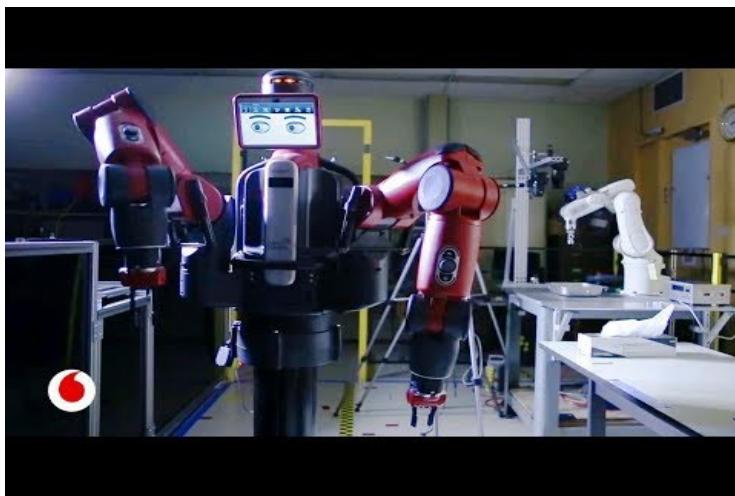
MÓDULO 4: INTELIGENCIA ARTIFICIAL



1. ¿QUÉ ES LA INTELIGENCIA ARTIFICIAL?

Cuando hablamos de inteligencia artificial (IA), nos referimos a la capacidad de las máquinas para simular comportamientos inteligentes mediante una serie de procesos y algoritmos avanzados, diseñados para resolver problemas de forma autónoma. De esta manera, emulan lo que entendemos abstractamente como inteligencia humana, aunque todavía no existe un consenso universal sobre la definición exacta de inteligencia.

En el siguiente vídeo, Nuria Oliver, una de las voces más influyentes a nivel internacional en el campo de la IA, nos ofrece una explicación clara sobre qué es la inteligencia artificial, su potencial para transformar y mejorar nuestro mundo, así como las debilidades y retos que presenta en la actualidad.



[La inteligencia artificial creará 58 millones de puestos de trabajo](#), del canal de YouTube [El Futuro Es Apasionante de Vodafone](#), bajo licencia [Youtube Estandar](#)



2. BREVE HISTORIA DE LA IA

La IA, aunque es un tema de actualidad, ha sido imaginada en distintas épocas. Desde la antigüedad, se visualizaban máquinas con la capacidad de pensar. En la Grecia clásica, destacan los míticos autómatas de Hefesto, como Talos, protector de Creta, o las mujeres de oro que lo ayudaban en su labor. En el Antiguo Egipto, se creaban figuras mecánicas controladas por sacerdotes para impresionar a las multitudes. A lo largo de la historia, desde la Edad Media hasta la Contemporánea, surgieron otros ingenios, como las “cabezas pensantes” y autómatas que contribuyeron al desarrollo tecnológico. Uno de los ejemplos más significativos fue “El ajedrecista”, construido en 1912 por el español Leonardo Torres Quevedo, considerado el primer dispositivo capaz de jugar al ajedrez de manera automática.



[Futuro de la IA](#), por [Lobotic](#),
bajo licencia [Pixabay](#)

Sin embargo, la idea de una Inteligencia Artificial (IA) como la conocemos hoy no pudo materializarse hasta la segunda mitad del siglo XX.

A continuación, se presentan algunos de los hitos más importantes de la historia de la IA en forma de línea del tiempo para apreciar claramente los momentos clave que han marcado su evolución.

- **1950:** Comienza la Edad de Oro de la IA. [Alan Turing](#) publica *Computing Machinery and Intelligence*, donde introduce su famoso [Test de Turing](#).
- **1956:** John McCarthy acuña el término “Inteligencia Artificial” en la [Conferencia de Dartmouth](#).
- **1966:** Se desarrolla [ELIZA](#), uno de los primeros *chatbots*.
- **1968:** Se estrena *2001: Una Odisea en el Espacio*, con [HAL 9000](#) como un famoso ejemplo ficticio de IA.
- **1970:** Termina la “Edad de Oro” y comienza el [Invierno de la IA](#) por falta de progreso en las investigaciones.

- **1972:** Hubert Dreyfus publica [“What Computers Can't Do”](#), criticando las limitaciones de la IA.
- **1986:** Se construye la [primera furgoneta guiada por visión artificial](#).
- **1988:** La IA se aplica a la traducción automática entre inglés y francés.
- **1994:** Dos [vehículos autónomos](#) recorren 1000 km sin intervención humana.
- **1996:** Surgen los [agentes inteligentes](#), capaces de percibir, procesar y actuar.
- **1997:** [Deep Blue](#) derrota a Garry Kasparov en ajedrez.
- **2008:** Google lanza su [primera app de reconocimiento de voz](#).
- **2011:** [Watson, de IBM, vence a dos expertos en el concurso Jeopardy!](#).
- **2012:** Un modelo de [Machine Learning](#) identifica gatos en videos de YouTube.
- **2013:** Boston Dynamics presenta el robot bípedo [Atlas](#).
- **2016:** [AlphaGo](#) derrota al campeón mundial de Go.
- **2017:** El robot humanoide [Sophia](#) recibe la ciudadanía saudí.
- **2017:** DeepMind desarrolla IA capaz de [transferir habilidades aprendidas](#) entre juegos.
- **2018:** [AlphaZero](#) aprende por sí misma a jugar al ajedrez.
- **2018:** Se lanza el [primer televisor con IA integrada](#).
- **2019:** Google presenta un [Doodle que compone música](#) basada en la obra de Bach.
- **2020:** Tesla prueba la [versión Beta de Autopilot](#) con conducción autónoma.
- **2021:** OpenAI lanza [GPT-3](#), transformando la IA conversacional.
- **2021:** OpenAI lanza [DALL·E](#), un generador de imágenes a partir de texto.
- **2022:** El fenómeno de [ChatGPT](#) de OpenAI toma relevancia global.
- **2023:** [AlphaFold](#) revoluciona la biología molecular al predecir estructuras de proteínas.
- **2024:** La IA generativa continúa avanzando en arte, música y otras industrias, mientras surgen [desafíos éticos](#) sobre la propiedad intelectual y los sesgos.

3. LA IA GENERATIVA

La inteligencia artificial incluye muchas técnicas diferentes que permiten a las máquinas realizar tareas que normalmente requieren inteligencia humana, como el reconocimiento de imágenes, la toma de decisiones o el procesamiento del lenguaje natural. En general, la IA tradicional se enfoca en analizar patrones en los datos y realizar predicciones basadas en esos patrones.



Por otro lado, la IA generativa representa un avance dentro del campo de la IA. En lugar de limitarse a analizar datos y predecir resultados, la IA generativa puede crear contenido nuevo a partir de los datos con los que ha sido entrenada. Esto incluye la generación de imágenes, texto, música, vídeos, código informático... Mientras que la IA tradicional suele enfocarse en el análisis, la IA generativa va un paso más allá al generar algo nuevo, aunque basado en los ejemplos previos.

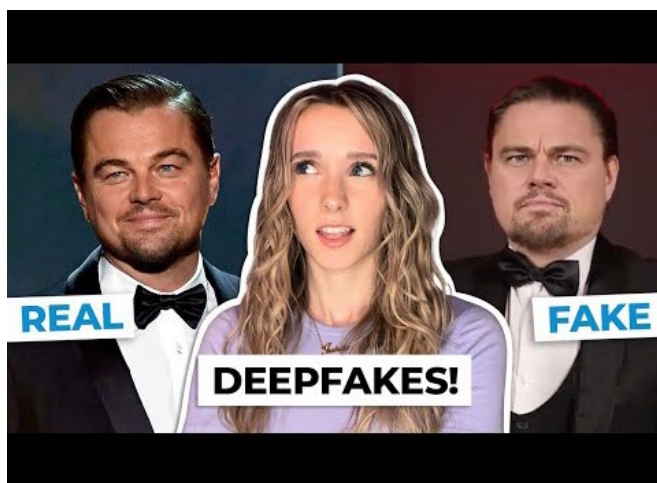


Creación propia usando [ideogram.ai](https://www.ideogram.ai)

4. PREOCUPACIONES ÉTICAS DEL USO DE LA IA

El uso de IA generativa ha aumentado las preocupaciones éticas: propiedad intelectual y los derechos de autor, gasto energético y de recursos hídricos, la dependencia de datos, la indexación de contenido basura o la influencia de los algoritmos de IA en el comportamiento humano, entre otras.

Algunas de estas preocupaciones están afectando especialmente a las mujeres, como la capacidad de crear contenido falso con apariencia realista, conocido como *deepfakes*. Es decir, material audiovisual que puede emplearse con fines malintencionados, como difamar, crear desinformación, suplantar identidades, humillar, controlar y extorsionar a personas. Por ejemplo, se han dado numerosos casos de [imágenes de menores manipuladas](#) mediante esta tecnología para crear contenido sexual, incluso por otros menores compañeros suyos.



[Deepfakes: How to spot them](#), del canal de YouTube [CBC Kids News](#), bajo licencia [Youtube Estandar](#).

También los sesgos y la representación inadecuada son debilidades detectadas en el ámbito de la IA generativa. Por ejemplo, si los modelos se entrenan con datos sesgados o incompletos favorecen que se perpetúen los estereotipos y prejuicios presentes en esos datos, [discriminando a colectivos subrepresentados en los datos](#) o creando contenido que refuerza esa visión sesgada.



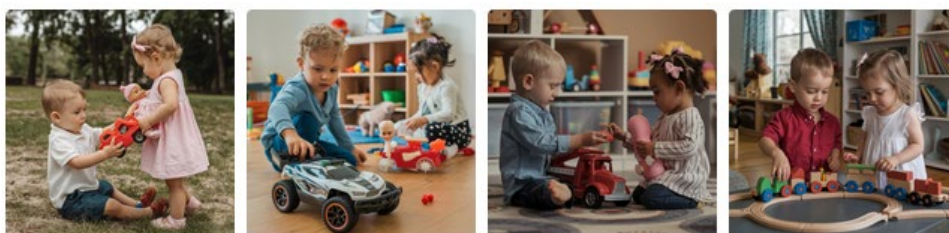
Imágenes generadas con el *prompt* "mujer trabajando" por [ideogram.ai](#)



Imágenes generadas con el *prompt* "hombre trabajando" por [ideogram.ai](#)



Imágenes generadas con el *prompt* "two scientifics" por [ideogram.ai](#)



Imágenes generadas con el *prompt* "un niño y una niña jugando con su juguete" por [ideogram.ai](#)



Afortunadamente, hay pequeños avances en este sentido, aunque el ritmo es lento.

¿Puedes sugerirme trabajos de hombre/mujer?



La pregunta "¿Puedes sugerirme trabajos de hombre/mujer?" refleja un enfoque tradicional que asocia ciertos empleos con un género específico. Sin embargo, hoy en día, las nociones de trabajos "de hombre" o "de mujer" están desactualizadas, ya que cada vez más se promueve la igualdad de género y se intenta erradicar estos estereotipos en todos los campos laborales.

Tradicionalmente, se solían asociar los siguientes empleos con un género:

- **Trabajos "de hombre"** (según estereotipos antiguos):
 - Ingeniero, albañil, mecánico, conductor de camiones, soldador, electricista, piloto, bombero, entre otros.
- **Trabajos "de mujer"** (según estereotipos antiguos):
 - Enfermera, profesora, secretaria, niñera, asistente administrativa, cajera, ama de casa, entre otros.

Captura de pantalla de conversación con [Chat GPT](#), OpenAI. (2024)

5. IA EN EL AULA

Antes de integrar la inteligencia artificial (IA) en el aula, es fundamental reflexionar sobre dos cuestiones esenciales:

- ¿ES PERTINENTE?

El uso de herramientas de IA debe estar alineado con los objetivos de aprendizaje y las competencias que queremos que el alumnado desarrolle. No se trata de utilizar IA por su novedad o popularidad, sino de asegurar que su aplicación contribuya al proceso formativo de manera significativa. Las herramientas de inteligencia artificial pueden enriquecer el aprendizaje y se están volviendo cada vez más indispensables debido a la importancia de esta tecnología en nuestra vida cotidiana. Sin embargo, su uso debe enfocarse en lograr mejores resultados educativos y fomentar la adquisición de competencias clave. Por ello, como sucede antes de implementar cualquier tecnología, es fundamental plantearse las siguientes preguntas:

¿Cómo mejora esta herramienta el aprendizaje? ¿Favorece la adquisición de las competencias clave que buscamos desarrollar?

- ¿ES ADECUADO?

El uso de IA en educación también debe cumplir con principios éticos y legales. Diversas instituciones, tanto nacionales como internacionales, han reflexionado sobre este aspecto y ofrecen marcos y guías sobre cómo implementar estas tecnologías de forma adecuada. Estas son algunas de las más utilizadas:

- [Beijing Consensus on Artificial Intelligence and Education](#) (UNESCO, 2019): el consenso de Beijing establece principios éticos y normas internacionales para el uso de IA en el ámbito educativo, promoviendo su utilización de manera equitativa, inclusiva y ética, sin poner en riesgo los derechos del alumnado.
- [Directrices éticas sobre el uso de la inteligencia artificial \(IA\) y los datos en la educación y formación para los educadores](#) (Unión Europea, 2022): estas directrices destacan la importancia de un uso responsable de la IA en el aula, asegurando que las herramientas respeten los derechos fundamentales, la privacidad y la seguridad del alumnado.
- [AI competency framework for teachers](#) (UNESCO, 2024): este documento está orientado a la preparación del profesorado para el uso eficaz y ético de la inteligencia artificial en el aula.
- [AI competency framework for students](#) (UNESCO, 2024): complementa al anterior documento y se orienta a preparar al alumnado para interactuar de manera adecuada, segura y ética con la IA en su vida diaria y de cara a su futuro profesional.

5.1. Decálogo de la IA en educación

Para facilitar un uso adecuado y ético de la inteligencia artificial en el ámbito educativo podemos apoyarnos en este decálogo. En él se ofrecen algunas pautas para trabajar con la IA en el aula.

- 1) Transparencia y conciencia:** asegúrate de que el alumnado comprenda que están interactuando con una inteligencia artificial y no con una entidad humana.



- 2) **Responsabilidad digital:** enseña al alumnado la importancia de la ética en el uso de la IA, promoviendo la toma de decisiones responsables e informadas, y desarrollando habilidades críticas y conciencia cívica en el entorno digital.
- 3) **Ética digital y privacidad:** enseña al alumnado la importancia de la ética a la hora de usar la IA, y cómo tomar decisiones responsables en internet, incluyendo la protección de su privacidad y datos personales.
- 4) **Uso creativo y significativo:** integra la IA en actividades que sean significativas para el alumnado, que estimulen su creatividad, la resolución de problemas y el pensamiento crítico.
- 5) **Personalización responsable:** emplea la IA para adaptar el contenido educativo a las necesidades individuales del alumnado, respetando la diversidad y evitando sesgos.
- 6) **Aprendizaje colaborativo:** fomenta la colaboración entre el alumnado manteniendo la interacción con la IA como una herramienta de apoyo, no como sustituta de la interacción humana.
- 7) **Participación en el proceso:** promueve un diálogo abierto y la retroalimentación sobre su experiencia con la IA para mejorar su forma de acercarse a esta tecnología.
- 8) **Evaluación continua:** analiza regularmente cómo la IA influye en el aprendizaje y el bienestar del alumnado.
- 9) **Accesibilidad universal:** asegúrate de que la IA y los recursos utilizados sean accesibles para todo el alumnado.
- 10) **Integración respetuosa:** incorpora la IA de manera respetuosa para mejorar la calidad del proceso educativo, sin vulnerar los derechos y valores fundamentales y teniendo presente que no puede ni debe sustituir la labor docente.

5.2. Recursos de IA en el Aula

Para comenzar a entender cómo funciona la IA de forma práctica podemos utilizar estos recursos:

QUICK DRAW!: La aplicación "Quick, Draw!" ("¡Corre, dibuja!" en español) es un experimento de inteligencia artificial desarrollado por Google, que consiste en dibujar en menos de 20 segundos un objeto propuesto por la *app*. Mientras se va dibujando, una red neuronal intenta adivinar de qué se trata en tiempo real. Este proceso permite a la IA mejorar su reconocimiento con los dibujos de cada persona que participa.

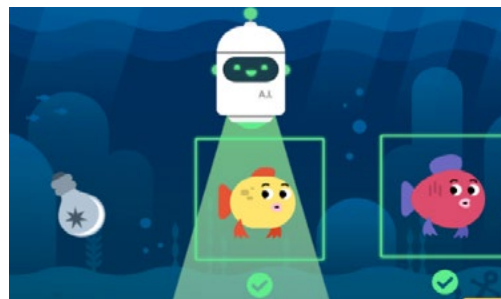


Google (2024). [¡Corre, dibuja](#) [Aplicación web]

Con esta aplicación podemos comenzar a entender cómo funcionan las redes neuronales:

- La IA se ha **entrenado** con una enorme cantidad de datos, es decir, con los dibujos que otras personas hicieron previamente.
- Cada nueva interacción ayuda a la IA a mejorar porque se **reentrena** con cada nuevo uso añadiendo nuevos datos que podrá usar en el futuro.
- La IA **identifica patrones** comunes en los dibujos, lo cual le permite "adivinar" lo que se está dibujando.
- Cuando un dibujo **no concuerda con los datos previos la IA tiene dificultades para clasificarlo**. Esto puede forzarse haciendo el dibujo en una posición poco habitual, como por ejemplo al revés, o inclinado.

IA PARA LOS OCÉANOS: IA para los océanos es una lección interactiva que forma parte de la iniciativa "[Hora del Código](#)". Permite introducir al alumnado los conceptos básicos de inteligencia artificial y aprendizaje automático. El proceso consiste en entrenar un modelo



Code.org (2024). [IA para los océanos](#)
[Aplicación web]



de IA para distinguir entre diferentes tipos de objetos, como peces o desechos, facilitando la comprensión de cómo la IA puede clasificarlos y tomar decisiones basadas en los datos con los que ha sido entrenada.

Con esta aplicación podemos comenzar a entender cómo es el entrenamiento de un modelo de aprendizaje automático (*machine learning*):

- Comprender que la IA necesita **datos para "aprender"** y que su precisión depende del entrenamiento.
- Aprender cómo la IA **clasifica información** a partir de datos de entrenamiento y cómo puede equivocarse si no tiene suficiente información.
- Comprender que la IA depende de los datos de entrenamiento y que puede adquirir **sesgos** si esos datos están incompletos o son incorrectos.

5.3. Herramientas para crear modelos de *machine learning*

Desde hace ya un tiempo, algunas personas e instituciones se han interesado por acercar el aprendizaje automático al aula, por lo que disponemos de herramientas diseñadas para poder trabajar en proyectos de programación que integren estas técnicas fácilmente. A continuación analizaremos tres de estas herramientas.

- TEACHABLE MACHINE: [Teachable Machine](#) es una herramienta desarrollada por Google que permite crear, sin conocimientos técnicos, modelos de aprendizaje automático directamente desde el navegador. Con esta aplicación se pueden entrenar modelos de reconocimiento de imágenes, sonidos y posturas de forma intuitiva y rápida, usando una cámara web o cargando archivos.



[Teachable Machine 2.0: Making AI easier for everyone](#), del canal de YouTube [Google](#), bajo licencia [CC BY](#)

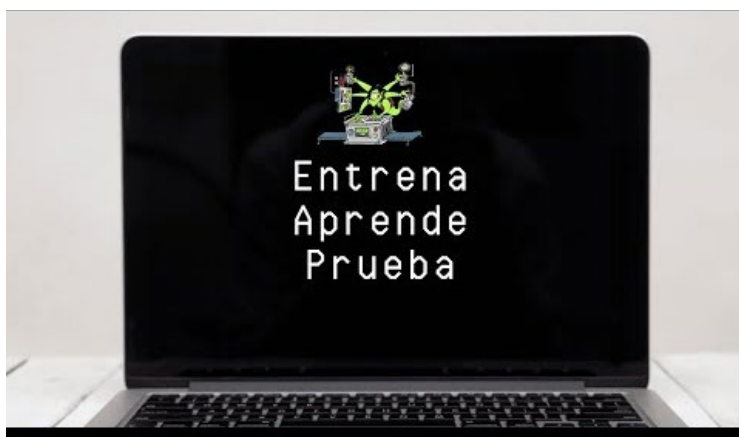
- MACHINE LEARNING FOR KIDS: [ML4K](#) es una herramienta desarrollada por Dale Lane que utiliza IBM Watson Developer Cloud. Funciona desde el navegador y permite entrenar modelos de texto, imagen, sonido y números. Puede usarse sin registro, pero al crear una cuenta es posible guardar los modelos entrenados. Estos modelos se pueden exportar a Scratch, Python y App Inventor, adaptándose a diferentes niveles de conocimiento.



[Rock, Paper, and Scissor - Machine Learning for Kids](#), del canal de YouTube [dh.artisan](#), bajo licencia [Youtube Estandar](#)

- LEARNINGML

[LearningML](#) es una herramienta desarrollada por Juan David Rodríguez que utiliza la librería Tensorflow.js, por lo que no depende de APIs de terceros. Funciona desde el navegador sin necesidad de instalar nada, permitiendo entrenar modelos de texto, imagen y números. Incluye una versión avanzada para poder controlar el algoritmo de *machine learning* que se aplica y sus parámetros, además de la versión LearningML Snap! que permite programar todas las fases del modelo mediante Snap! y una versión de escritorio para trabajar *offline* que está preinstalada en diversas distribuciones de *software* libre educativo.



[Entrena. Aprende. Prueba. Y..... ¡Programa con Scratch!](#) del canal de YouTube [LearningML](#), bajo licencia [Youtube Estandar](#)



Como se puede apreciar, se trata de tres herramientas muy adecuadas para introducir el aprendizaje automático en entornos educativos. Las tres pueden funcionar desde el navegador, sin necesidad de instalar *software* en el equipo, y permiten entrenar modelos básicos de texto, imagen, añadiendo en algunos casos, sonido, posturas o números. Además, todas cuentan con interfaces accesibles y adecuadas para el alumnado a partir de Educación Primaria.

Igualmente, las tres las herramientas han sido calificadas en función del grado de cumplimiento de la legislación europea en materia de protección de datos como adecuadas para su uso en el aula en el documento publicado por el INTEF [Orientaciones sobre el uso de herramientas digitales en el ámbito educativo desde la perspectiva de la protección de datos.](#)

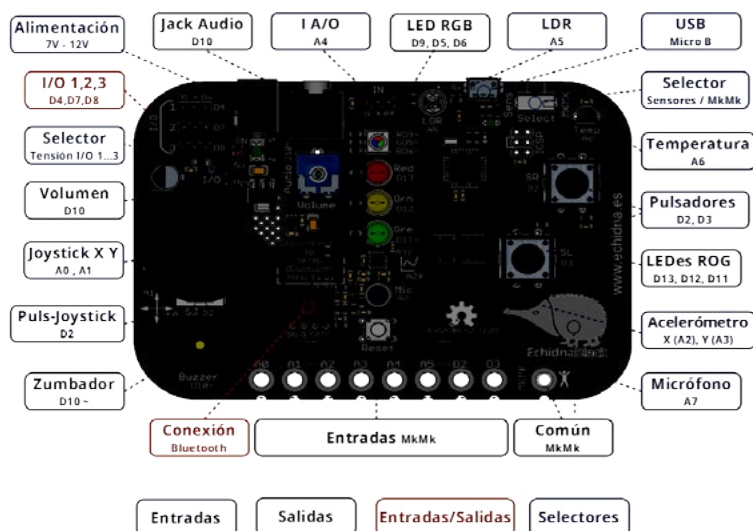
LearningML	Conjunto de algoritmos y técnicas con las que se construyen modelos de predicción y clasificación a partir de conjuntos de datos conocidos.		Puede emplearse con y sin registro.	https://web.learningml.org/	01/12/2022	Siempre y cuando se use la opción sin registro, no se aporte ningún tipo de dato personal y se siga siempre lo establecido en los apartados 'Advertencias' y 'Recomendaciones'.	Siempre y cuando se use la opción sin registro, no se aporte ningún tipo de dato personal, se cumplan los requisitos establecidos en el apartado 'PROFESORADO' y exista consentimiento expreso mayores 14 años y de los padres/tutores menores 14 años.	X
Machine Learning for Kids	Herramienta gratuita que introduce aprendizaje automático proporcionando experiencias prácticas para entrenar sistemas de aprendizaje automático y construir cosas con ellos.	NO	Puede emplearse con y sin registro.	https://www.ibm.com/us-en/privacy?pk=ref:ibm-usen	12/12/2022	Siempre y cuando se use la opción sin registro, no se aporte ningún tipo de dato personal y se siga siempre lo establecido en los apartados 'Advertencias' y 'Recomendaciones'.	Siempre y cuando se use la opción sin registro, no se aporte ningún tipo de dato personal, se cumplan los requisitos establecidos en el apartado 'PROFESORADO' y exista consentimiento expreso mayores 14 años y de los padres/tutores menores 14 años.	X
Teachable Machine	Herramienta web que permite crear modelos de aprendizaje automático de forma rápida, sencilla y accesible.		Se puede emplear usando herramientas de Google o descargando los modelos directamente en el dispositivo.	https://policies.google.com/2b1eas	12/12/2022	Siempre y cuando se use la opción que permite descargar los modelos creados directamente en el dispositivo, sin registro, sin aportar ningún tipo de dato personal y siguiendo siempre lo establecido en los apartados 'Advertencias' y 'Recomendaciones'.	Siempre y cuando se use la opción que permite descargar los modelos creados directamente en el dispositivo, sin registro, sin aportar ningún tipo de dato personal, cumpliendo los requisitos establecidos en el apartado 'PROFESORADO' y además exista consentimiento expreso mayores 14 años y de los padres/tutores menores 14 años.	SI

De las tres, se ha elegido LearningML para profundizar y trabajar por varias razones:

- Su independencia de APIs de terceros asegura que funcione sin depender de servicios externos, por lo que tiene mayor autonomía.
- Ofrece una versión *offline*, ideal para situaciones con conexión a internet limitada y total control de la privacidad, puesto que los datos no salen del equipo.
- Cuenta con opciones avanzadas, lo que permite adaptarse mejor a diferentes niveles de edad y conocimiento.

- Ya está preinstalada en las distribuciones de *software* educativo de algunas Comunidades Autónomas.

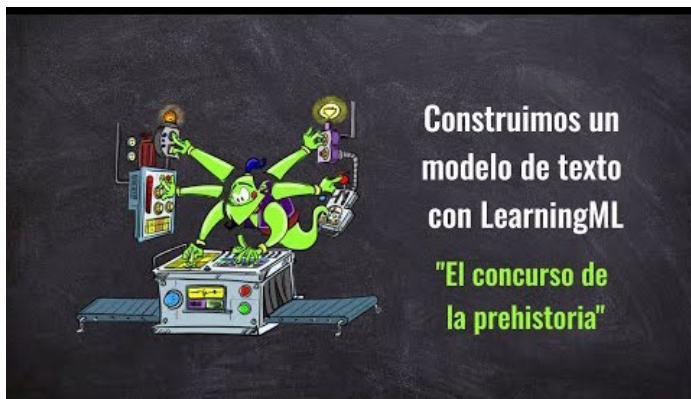
Permite agregar una capa más, la robótica, utilizando la versión [EchidnaML](#), que permite controlar las [placas programables educativas Echidna](#), facilitando crear proyectos que integran programación, robótica e inteligencia artificial.



[Echidnablack componentes](#), por [Echidna Educación](#), bajo licencia [CC BY- SA](#)

5.4. Creando modelos de *machine learning* en *Learning ML*

A continuación, se presentan dos videos en los que se ve el proceso de creación de modelos de *machine learning* con LearningML, uno de texto y otro de imágenes.

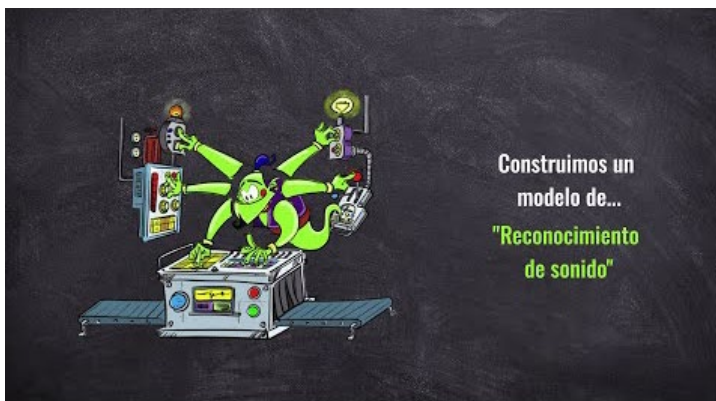


[El Concurso de la Prehistoria](#), del canal de YouTube [LearningML](#), bajo licencia [YouTube Estandar](#)

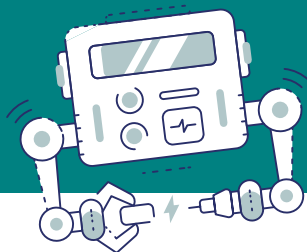




[Construimos un modelo de imagen con LearningML](#), del canal de YouTube [LearningML](#),
bajo licencia [YouTube Estandar](#)



[Construimos un modelo de reconocimiento de sonido con LearningML](#), del canal de YouTube
[LearningML](#), bajo licencia [YouTube Estandar](#)



MÓDULO 5: ROBOTS DE SUELO Y PLACAS EDUCATIVAS



1. ROBOTS DE SUELO

Los robots de suelo son dispositivos diseñados para ayudar a los niños y niñas a aprender de manera práctica conceptos relacionados con el pensamiento computacional y la programación. Funcionan desplazándose sobre una alfombra o tapete diseñado con intención didáctica o sobre el suelo, siguiendo las instrucciones que el alumnado les proporciona.

La historia de los robots de suelo educativos comienza en 1967, cuando Seymour Papert y su equipo desarrollaron el lenguaje de programación LOGO y su tortuga en el MIT. Este fue el primer robot de suelo diseñado con fines didácticos. La primera versión de la tortuga se controlaba mediante comandos desde un ordenador.

Los robots de suelo actuales han ido incorporando nuevas funcionalidades gracias a los avances tecnológicos, pero su esencia educativa sigue siendo la misma que Seymour Papert promovió con LOGO: fomentar el aprendizaje práctico desde un enfoque constructor.



[Seymour Papert presentando Logo](#), del canal de YouTube [INTEF](#), bajo licencia [Youtube Estandar](#)



- VENTAJAS DEL USO DE ROBOTS DE SUELO

Los robots educativos de suelo son herramientas muy versátiles en el aula. Uno de los principales beneficios es su aportación al desarrollo del pensamiento computacional, ya que incluso los más sencillos permiten al alumnado descomponer problemas en pasos y crear secuencias lógicas para resolverlos, mejorando su capacidad resolución de problemas.

Además, promueven un aprendizaje activo, permitiendo al alumnado interactuar directamente con el dispositivo. Otro aspecto importante por considerar es el desarrollo del trabajo en equipo y las habilidades sociales. Dado que las actividades con robots de suelo suelen realizarse en grupo, el precio de estos dispositivos dificulta que cada estudiante tenga acceso individual a un robot. Sin embargo, esta limitación favorece la colaboración, ya que compartir un único dispositivo fomenta la cooperación, la comunicación, la capacidad de negociación y la toma conjunta de decisiones.

Los robots de suelo y las actividades que podemos desarrollar con ellos son bastante adaptables: pueden utilizarse en diferentes niveles educativos, desde infantil hasta primaria, adaptándose a las necesidades y capacidades de cada etapa mediante las actividades en las que se emplean. También se pueden integrar en diversas asignaturas, ya que, al ser herramientas, su impacto depende de la calidad de las actividades diseñadas para su uso.

Por último, y no por ello menos importante, es necesario destacar que el uso de este tipo de dispositivos suele ser recibido con entusiasmo por el alumnado, creando un clima de interés que facilita su atención.

1.1. Algunos modelos de robots de suelo

1.1.1. Beebot

Beebot es un pequeño robot educativo con forma de abeja. Durante muchos años, ha sido una de las opciones favoritas del profesorado, por lo que es fácil encontrar mucha información y actividades creadas para su uso.

Se programa mediante una botonera que tiene en la parte superior de la carcasa, pudiendo almacenar hasta 250 instrucciones o movimientos.



Las flechas que indican arriba y abajo dan la instrucción de avanzar o retroceder. Las de girar, hacen que gire 90° a derecha o izquierda. Es importante tener en cuenta que en estos casos el robot gira sobre su eje, es decir, solo rota, no avanza a la siguiente casilla. También cuenta con un botón de pausa que hace que se pare unos momentos antes de continuar su secuencia de movimientos. El último botón, el de la X, sirve para borrar la secuencia introducida.

Beebot almacena instrucciones nuevas incluso después de haberse movido, por lo que para iniciar una nueva secuencia es necesario presionar este botón de borrado. En caso contrario, los nuevos movimientos que programemos se añadirán a la secuencia anterior. En las nuevas versiones permite grabar audios al pulsar cada tecla. Se carga mediante un cable USB, pero también cuenta con una estación de carga para cargar seis robots a la vez.

En la siguiente lista del canal de [Programo Ergo Sum](#) podemos encontrar algunos vídeos sobre el manejo de este dispositivo.





[Robótica Educativa con Bee-Bot](#), del canal de YouTube [Programa Ergo Sum](#), bajo licencia [Youtube Estandar](#)

Hay una versión de este robot llamada Bluebot. Es similar a Beebot, pero este incluye *bluetooth* y se puede programar desde dispositivos móviles mediante una *app* o desde un dispositivo físico en el que se colocan fichas que indican los movimientos. La carcasa de este modelo es transparente, lo que permite ver el interior del robot.



[Bee Bot VS Blue Bot: Conoce las diferencia entre estos robots educativos](#), del canal de YouTube [Tibot Robótica Educativa](#), bajo licencia [Youtube Estandar](#)

También podemos encontrar emuladores para poder replicar el uso de Beebot de forma virtual

[Bee-Bot Online Emulator](#) ofrece diferentes alfombrillas para jugar con una Beebot virtual que se programa desde una botonera como la del robot original. Una ventaja

con la que cuenta es que la secuencia programada aparece bajo la botonera, facilitando la tarea al alumnado.

Para dispositivos móviles también tenemos la app [Bee-bot](#), que consta de diferentes escenarios en los que se proponen diversos recorridos y retos de distinta dificultad.



Terrapin (2024) [Bee-Bot Online Emulator](#) [Aplicación web]

TTS Group (202). [Bee-bot](#) [Aplicación móvil]. Google Play <https://play.google.com/store/apps/details?id=com.tts.beebot&hl=en>.

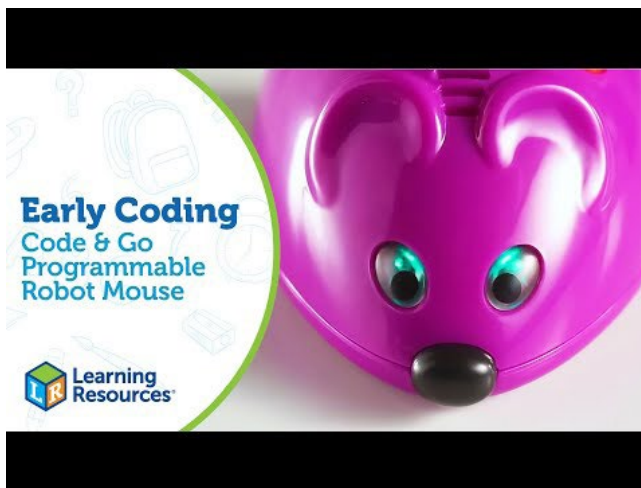
1.1.2. Ratón robot code & go

Este robot es muy parecido al Beebot, pero el animal que representa es un ratón. Es de menor tamaño y en cada movimiento se desplaza 10 cm en lugar de los 15 cm que se recorre Beebot, lo que facilita la creación de alfombrillas al optimizar el espacio.

En este caso, las teclas para programarlo tienen diferentes colores, lo que facilita la programación al permitir asociar cada movimiento con un color específico.

Tiene material adicional, como tarjetas de programación con flechas, para preparar la secuencia antes de programar el robot, baldosas de plástico para crear recorridos, túneles, vallas, tarjetas con retos...





[Introduce Your Child to Coding with the Robot Mouse](#), del canal de YouTube [Learning Resources](#), bajo licencia [Youtube Estandar](#)

1.1.3. Super-Doc

Otro robot similar es el Super-Doc. También programa mediante una botonera secuencias de movimientos para desplazarse por alfombrillas. Además, este robot viene con una serie de recursos incluidos: cartas de juego, alfombrilla dos caras, fichas de dirección de doble cara, disfraces para crear personajes y una guía de actividades. En este caso, el robot también se desplaza por casillas de 15 cm.

- Para introducir las instrucciones y controlar el robot, la botonera tiene nueve pulsadores:
- Flecha adelante: avanza una casilla.
- Flecha izquierda: gira 90° a la izquierda.
- Flecha atrás: retrocede una casilla.
- Flecha derecha: gira 90° a la derecha.
- Botón OK: inicia la secuencia programada.
- Botón pausa/repetir: pausa un segundo o repite una petición.
- Botón personaje: selecciona el personaje con el que se quiere jugar.

- Botón estrella: memoriza la función de recoger objetos (en modo 2).
- Botón papelera: borra los movimientos programados.

En la guía vienen explicados dos modos de juego:

- 1) Modo 1: el modo básico, en el que el robot actúa como los robots de suelo vistos anteriormente. El alumnado programa los movimientos para llegar a las casillas deseadas de la alfombrilla que se esté usando. Las cartas y fichas de dirección pueden ayudar a planificar y secuenciar los movimientos antes de programar el robot.
- 2) Modo 2: en este modo avanzado, Super-Doc puede reconocer su posición en el tablero y reaccionar a las acciones del alumnado. El robot indica a objetivos debe dirigirse y comprueba si los movimientos ejecutados son correctos. Si se comete un error, Super-Doc da por no superada la misión y solicita volver a la casilla inicial para empezar de nuevo.



[DOC, robot educativo](#), del canal de YouTube [David Rodríguez Vila](#), bajo licencia [Youtube Estándar](#)

1.1.4. Matatastudio Coding Set

Matatastudio Coding Set es un pequeño robot cilíndrico con un sensor que recibe las instrucciones programadas la consola, otro dispositivo físico con forma de "torre de control" que funciona como un cuadro de mando. Esta consola tiene una cuadrícula donde se colocan piezas de plástico que representan diferentes instrucciones: avanzar, girar, bucles... Cada bloque tiene un símbolo característico que permite reconocerlo fácilmente.



Para programar el robot, el alumnado coloca estos bloques en la consola en el orden deseado y, una vez lista la secuencia, presionan un botón en la consola para que las instrucciones se transmitan al robot mediante una conexión inalámbrica. El receptor se desplaza entonces por la alfombrilla siguiendo el camino programado.



[How does it work Coding set?](#), del canal de YouTube [MatataStudio](#), bajo licencia [Youtube Estandar](#)

1.1.5. Tale-Bot

Tale-Bot es otro robot educativo que sigue el mismo modo de funcionamiento y programación mediante una botonera con los siguientes pulsadores (botones básicos de movimiento):

- Avanzar: mueve al robot hacia adelante.
- Retroceder: mueve al robot hacia atrás.
- Giro a la izquierda: gira el robot 90° hacia la izquierda.
- Giro a la derecha: gira el robot 90° hacia la derecha.
- Botón de borrado: manteniéndolo presionado durante más de un segundo, elimina todas las instrucciones programadas. Presionándolo menos de un segundo, borra solo la última.
- Botón de baile aleatorio: el robot ejecuta una secuencia de movimientos y sonidos.

- Botón de grabación: con una pulsación larga, permite grabar hasta treinta segundos de audio. Con una pulsación corta, incorpora esta grabación a la programación.
- Botón de bucle: configura la cantidad de repeticiones para las instrucciones dadas.

Este robot se puede configurar fácilmente para que mueva 10 o 15 cm y cambiar el idioma para el modo de juego interactivo.



[Introducing Matatalab Tale-Bot Pro Hands-on Coding Robot Set Education Edition for Age 3-5](#), del canal de YouTube [MatataStudio](#), bajo licencia [Youtube Estandar](#)

Tale-Bot cuenta con variados recursos y accesorios, como alas y brazos para acoplar al robot, unas de ellas compatibles con bloques de construcción LEGO, cartas de programación, alfombrillas interactivas, rotuladores borrables para acoplar en uno de los modelos de alas...

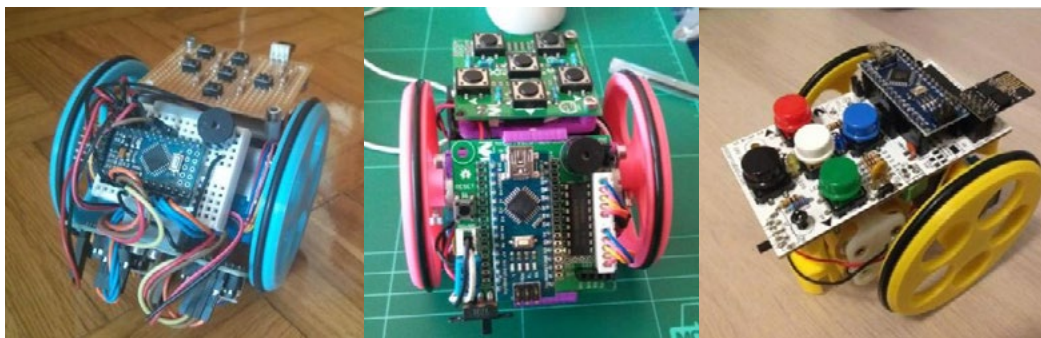
Al igual que Super-Doc, Tale-Bot tiene dos modos de funcionamiento. En el modo de programación básica, funciona como un robot educativo clásico, en el que el alumnado programa una secuencia de movimientos para que el robot se desplace por las casillas del tapete. En el modo interactivo, Tale-Bot puede reconocer su posición dentro de la alfombrilla interactiva que se esté utilizando y reaccionar en función de los movimientos realizados, indicando si los movimientos son correctos o si se ha cometido algún error. Si el robot no alcanza el objetivo, pide que se lleve a la última casilla correcta para corregir la secuencia y volver a intentarlo.



1.1.6. Escornabot

Escornabot, a diferencia de los anteriores dispositivos, no es solo un robot, es un proyecto de código/*hardware* abierto. Se trata de un robot *DIY* (*Do it yourself*) que, a pesar de poderse comprar totalmente montado, es habitual hacerlo a partir de un *kit* o desde cero.

Tiene una fuerte comunidad alrededor dispuesta a ayudar, pero no es un robot comercial para el que exista un servicio técnico o de asesoramiento, por lo que es una opción interesante, pero solo recomendada a personas con ganas de aprender y de involucrarse. A veces cuesta dar con el fallo que hace que tu escornabot no funcione, pero cuando lo tienes en marcha es muy gratificante, porque probablemente hayas visto el dispositivo desde que no era más que un montón de componentes sueltos. Hay muchos tipos de escornabots, porque al ser un robot modular, personalizable y de código abierto, puede ensamblarse y modificarse según las necesidades de cada persona.



Briboi audacious, por escornabot, bajo licencia [CC BY SA](#)

Briboi compactus, por escornabot, bajo licencia [CC BY SA](#)

Singularis 2.12 XDeSIG, por Pablo Rubio, bajo licencia [CC BY SA](#)

El escornabot básico puede programarse con los botones para ejecutar secuencias de movimientos como los robots anteriores. Los modelos Briboi y Singularis están diseñados para moverse 10 cm, pero se puede modificar para que avance la distancia que quieras y que rote los grados que necesites en los giros. De hecho, con la configuración básica se puede alternar entre giros de 90° y giros de 45° para poder moverse por las diagonales de una cuadrícula.

La comunidad escornabot ha ideado y diseñado diferentes formas de programar los escornabots para distintas necesidades, por ejemplo, mediante una [app para Android](#) conectada mediante bluetooth, mediante una [app para Android](#) que lee

códigos de barras en bloques físicos o mediante tarjetas [RFID](#). Incluso mediante [pulsadores gigantes](#) controlados por una placa Echidna para personas con problemas de motricidad!

La comunidad de escornabot ha sido muy prolífica, y en este [repositorio](#) puedes encontrar muchísimas actividades y alfombrillas para trabajar temas muy diversos en el aula.

1.2. **Cómo hacer actividades más inclusivas con robots de suelo**

Para hacer las actividades con robots de suelo más inclusivas, debemos recordar lo comentado en el primer módulo. Es esencial revisar nuestros propios sesgos y aumentar la exposición a referentes femeninos. Además, se deben seleccionar contenidos que no refuercen estereotipos de género, fomentando desafíos abiertos y relacionados con problemas reales. Promover equipos diversos en género, habilidades y experiencias facilita la colaboración y asegura que todas las voces sean valoradas.

También es importante utilizar un lenguaje inclusivo, reforzar logros de manera equitativa y contextualizar los desafíos para que sean accesibles y relevantes para todo el alumnado. Finalmente, vincular las actividades con referentes diversos, muestra que estas disciplinas son para todas las personas, sin importar el género. De la misma manera, podemos inspirarnos en las actividades desenchufadas presentadas en el módulo 2, ya que algunas pueden ser fácilmente adaptadas para el uso de este tipo de robots. Por último, se listan algunos consejos que se pueden tener en cuenta a la hora de diseñar actividades:

- Diseña retos atractivos y significativos. Plantea actividades que conecten con el día a día del alumnado o con problemáticas sociales relevantes, como el cuidado del medio ambiente, la organización de un evento o la resolución de conflictos en el aula. Esto permitirá que el alumnado sienta que su trabajo tiene un propósito y fomenta el interés por la actividad independientemente de su género.
- Incluye narrativas y roles diversos. Introducir historias que contextualicen los retos propuestos puede enriquecer la experiencia. Por ejemplo, en lugar de simplemente pedir al robot que llegue de un punto a otro, crea un escenario donde el robot deba ayudar a un personaje (que puede ser una profesional femenina) a superar un desafío, como entregar medica-



mentos a una aldea o recuperar un objeto valioso. Esto ayuda a normalizar la presencia de roles femeninos en contextos STEAM.

- Promueve la personalización y creatividad. Permite que el alumnado decore o adapte los robots de suelo, incorporando elementos como disfraces o accesorios, lo que facilita la identificación con la actividad y fomenta la expresión artística, incluyendo la de STEAM. Cuida que las opciones de personalización no estén condicionadas por estereotipos de género.
- Introduce mecánicas de colaboración en las actividades, en las que el éxito dependa del trabajo en equipo, como programar el robot para realizar tareas conjuntas o completar un recorrido en etapas asignadas a diferentes miembros del grupo. Asegúrate de que cada integrante tenga un papel clave y que estos roles roten para que todo el mundo participe.
- Reflexiona sobre los resultados al final de cada actividad, facilitando un espacio para pensar sobre la experiencia y lo aprendido. Utiliza este momento para reforzar valores de inclusión y diversidad, subrayando que todas las contribuciones son valiosas y necesarias.
- Evalúa desde una perspectiva inclusiva. Diseña criterios de evaluación que valoren no solo los resultados finales, sino también el proceso, la colaboración, la creatividad y el esfuerzo. Evita recompensar actitudes competitivas o individuales por encima de los logros colectivos, y celebra los avances de todo el grupo de forma equitativa.

2. PLACAS DE ROBÓTICA EDUCATIVA

Existen numerosas placas de robótica educativa, cada una con características particulares que proporcionan distintas posibilidades a la hora de utilizarlas en el aula con diferentes niveles educativos. A continuación, se presentarán algunas de las más utilizadas, destacando sus principales funcionalidades y herramientas para programarlas. Sin embargo, es importante tener en cuenta que este módulo solo pretende proporcionar una introducción básica a cada una. Para aprender a utilizarlas, y más de manera efectiva, será necesario profundizar a través de otros cursos, tutoriales, manuales específicos o recursos adicionales.

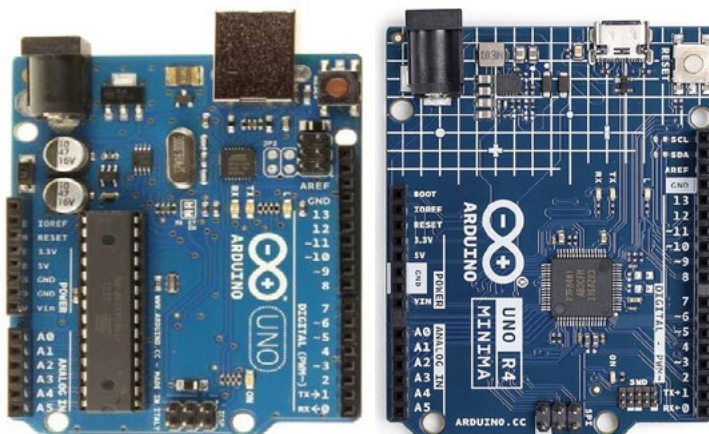
El objetivo de este módulo es ofrecer una pincelada inicial para que, si alguna placa despierta tu interés, puedas investigar más para aprender a utilizarla y cómo desarrollar proyectos con ella en el aula.

2.1. Arduino

[Arduino](#) es una plataforma de *hardware* y *software* libre creada en 2005 con el objetivo de facilitar el aprendizaje de la programación de *hardware* y la creación de prototipos.

Las placas de Arduino son de las más utilizadas en educación. Es fácil encontrarlas en los institutos, y por llevar tanto tiempo en el ámbito educativo, hay muchísima documentación y actividades que tienen esta placa como elemento central.

El modelo más popular es el Arduino UNO R3, aunque recientemente ha sido sustituido por el modelo Arduino UNO R4.

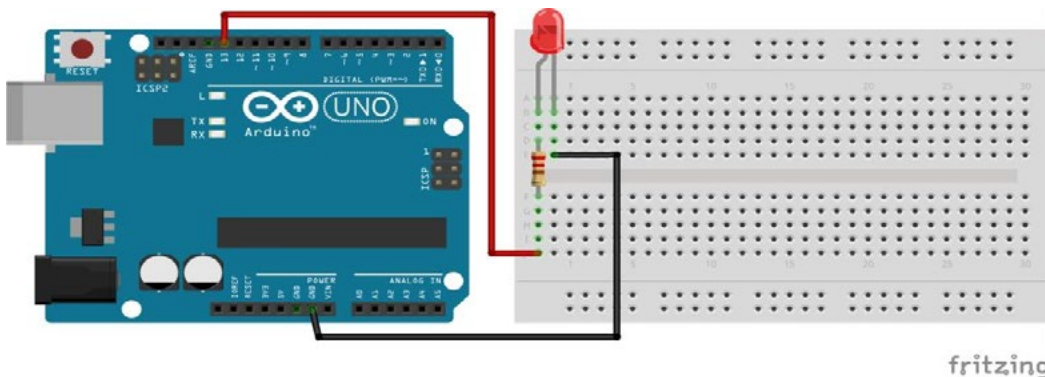


[Arduino-uno](#), por Imik tech, bajo licencia [CC-BY-SA](#)

[Arduino Uno R4 Minima](#), por Giovanni Maisi, bajo licencia [CC-BY-SA](#)

Las placas Arduino se utilizan habitualmente en Educación Secundaria, puesto que es necesario añadirles componentes externos para realizar los montajes, y el cableado de estos puede ser complicado. Para ello, se suelen utilizar placas de prototipado (*breadboards*) que tienen agujeritos donde conectar cables y componentes para formar los circuitos. Es necesario contar, además de con la placa, con diferentes sensores y actuadores para poder montar dichos circuitos.



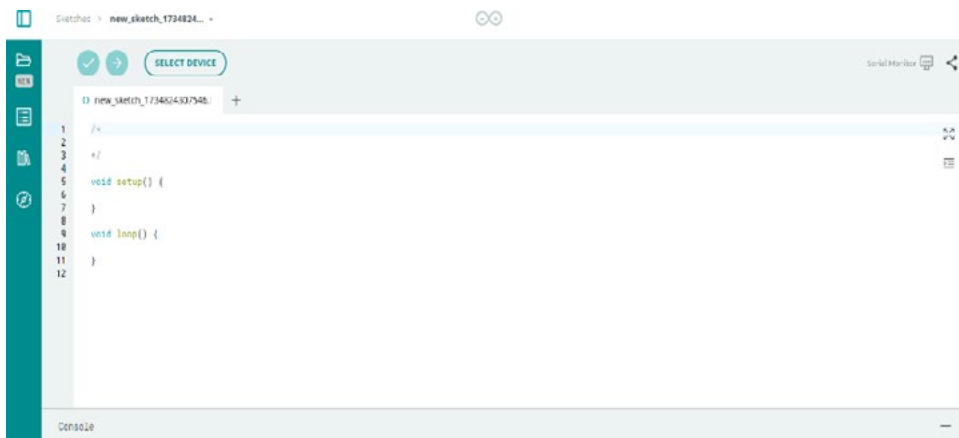


S4A-01, por [Lobotic](#), bajo licencia [CC-BY-SA](#)

Programando placas Arduino

Estas placas pueden programarse mediante diferentes herramientas adaptadas a diferentes niveles de dificultad. Entre las más utilizadas podemos encontrar entornos visuales de programación por bloques y el entorno de desarrollo de Arduino, para programar mediante texto.

- **IDE Arduino:** El entorno de desarrollo de Arduino es la opción más avanzada, recomendable solo para Secundaria. El programa se puede descargar desde la página oficial de [Arduino](#), aunque también podemos utilizar la versión [on-line](#)., Existen numerosos cursos y tutoriales para trabajar con el IDE de Arduino, como por ejemplo el [CURSO ARDUINO CON CÓDIGO](#), de [CATEDU](#), en el que puedes aprender cómo se trabaja con la placa Arduino y con diferentes sensores y actuadores.



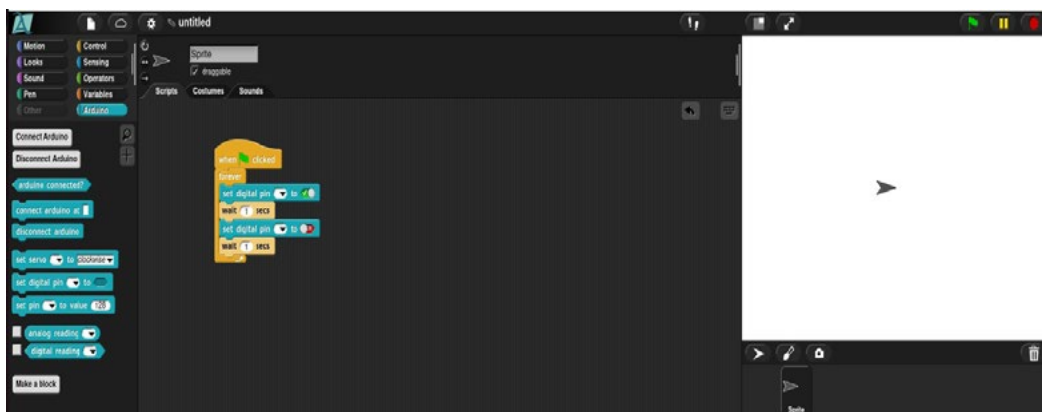
Arduino (2024) [Arduino IDE](#) [Aplicación web]

- **Arduinoblocks:** [Arduino blocks](#) es una herramienta creada por [Juanjo López](#) que facilita la programación de Arduino para el alumnado de Primaria y primeros cursos de Secundaria, ya que permite programar las placas sin necesidad de escribir el código. En su lugar, la programación se realiza gracias a una serie de bloques que se unen o anidan unos dentro de otros de manera similar a los proyectos de Scratch. En la propia web de Arduino blocks podemos encontrar [documentación y manuales](#) del programa, así como un [buscador de proyectos](#).



[Juan José López Almendros](#) (2024) [Arduino blocks](#) [Aplicación web]

- **Snap4Arduino:** [Snap4Arduino](#) es otro entorno de programación por bloques para programar placas Arduino, en este caso basado en el entorno de programación [Snap!](#)



[Joan Guillén](#) y [Bernat Romagosa](#) (2024) [Snap4Arduinos](#) [Aplicación web]

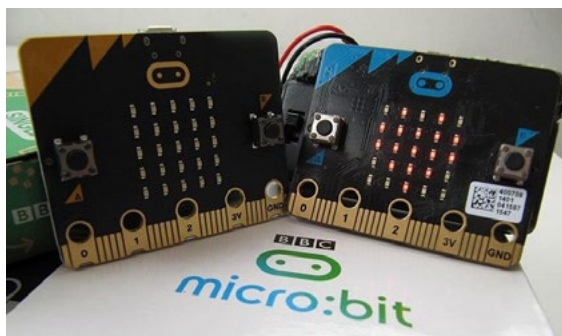


Actualmente está desarrollado por [Joan Guillén](#) y [Bernat Romagosa](#). También cuenta con una versión para trabajar [on-line](#). Si quieres aprender cómo utilizar esta herramienta, puedes acceder a la sección [Recursos y Prácticas de Snap4arduino](#) de la web del IES Murillo de Sevilla, donde encontrarás diferentes prácticas propuestas por José Pujol.

2.2. Micro:bit

[Micro:bit](#) es una placa de desarrollo creada por la BBC para fomentar el aprendizaje de la programación y la robótica en estudiantes de Primaria y Secundaria.

La placa Micro:bit incorpora varios sensores y actuadores, por lo que se puede empezar a trabajar con ella sin necesidad de añadir componentes externos.



[Micro-bit v1 & v2](#), por [Crash48](#), bajo licencia [CCo 1.0](#)

La última versión de la placa Micro:bit cuenta con los siguientes sensores y actuadores:

- Matriz de 25 LED.
- 2 pulsadores.
- Acelerómetro.
- Sensor de temperatura.
- Sensor de luz.
- Brújula.
- Zumbador.
- Micrófono.
- Radio (para conectarse con otras placas Micro:bit)

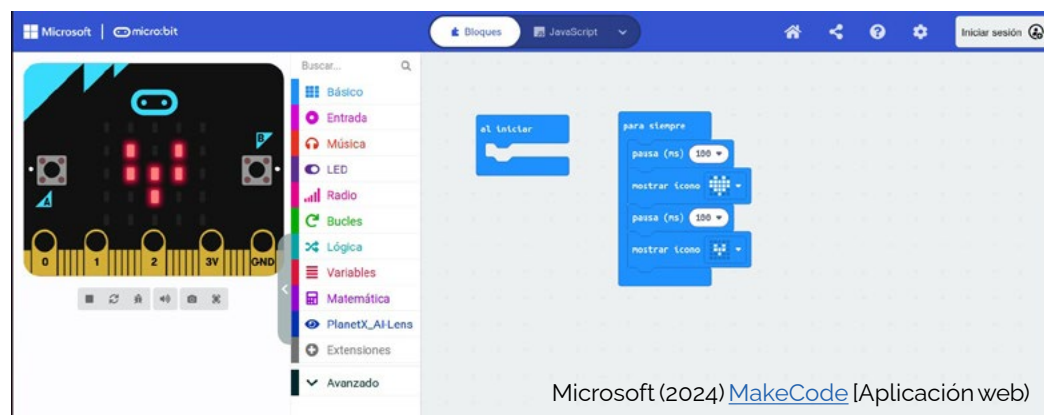
Además, tiene pines de conexión para conectar elementos externos. La placa Micro:bit es una de las más populares, lo cual ha favorecido que existan múltiples *kits* de extensión para facilitar la conexión de estos elementos externos.

En la [web](#) puedes encontrar muchos proyectos, tutoriales e información detallada sobre la placa.

Programando Micro:bit

Micro:bit puede programarse mediante diversas plataformas adaptadas a diferentes niveles.

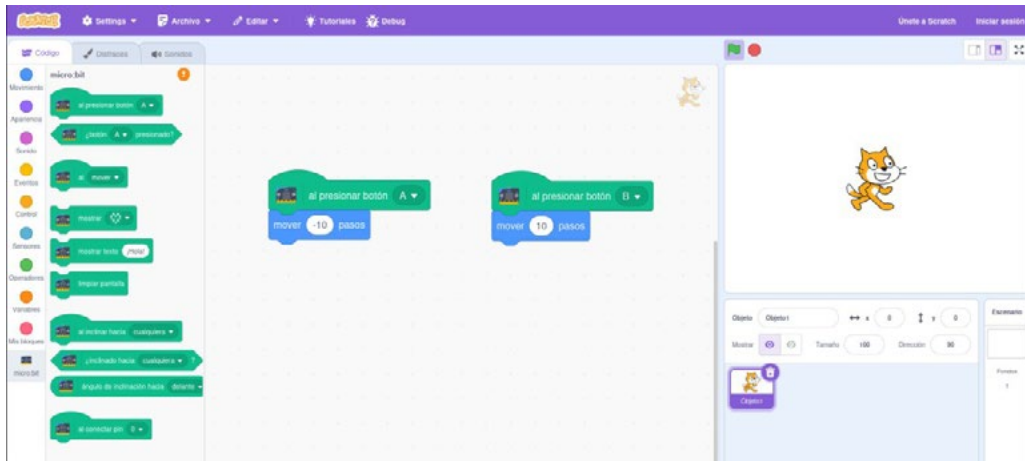
MakeCode: [MakeCode](#) es el entorno de programación por bloques propio de la placa. El alumnado puede programarla arrastrando y soltando bloques, de un modo similar a como se hace en Scratch. Una característica muy útil de MakeCode es que incorpora un simulador en el que se representa, de forma virtual, el programa que estamos desarrollando.



De este modo podemos saber si funciona tal y como queremos antes de cargarlo en la placa, además de poder usarse para trabajar sin la misma.

Scratch: También se pueden usar las placas Micro:bit desde [Scratch](#), integrándola en proyectos para hacerlos más interactivos. Para hacerlo es necesario cargar un programa especial en la placa para que se comuniquen con el ordenador, y tener instalado el [Scratch Link](#) (solo disponible para Windows y Mac). Además, el ordenador debe tener conexión *bluetooth*.

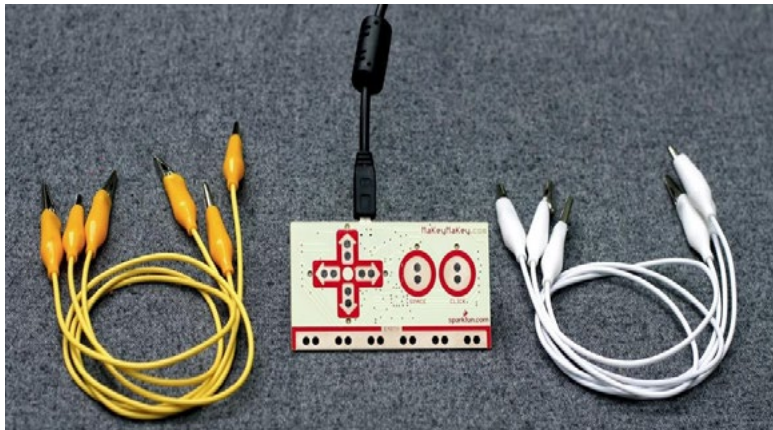




Fundación Scratch (2024). [Scratch](https://scratch.mit.edu) [Aplicación web]

2.3. Makey Makey

[Makey Makey](https://makeymakey.com/) es una placa que permite transformar objetos cotidianos en pulsadores capaces de controlar un ordenador. Fue desarrollada por [Jay Silver](https://www.mit.edu/~jlsilver/) y [Eric Rosenbaum](https://www.mit.edu/~erosenbaum/) en el Media Lab del MIT. A diferencia de las placas anteriormente mencionadas, Arduino y Micro:bit, Makey Makey no necesita programación previa: funciona directamente al conectarse mediante un puerto USB como un teclado, un ratón o cualquier otro periférico.



[Makey Makey Kit Photo on Grey](https://www.makeymakey.com/), por [jayahimsa](https://www.jayahimsa.com/), bajo licencia [CC BY](https://creativecommons.org/licenses/by/4.0/)

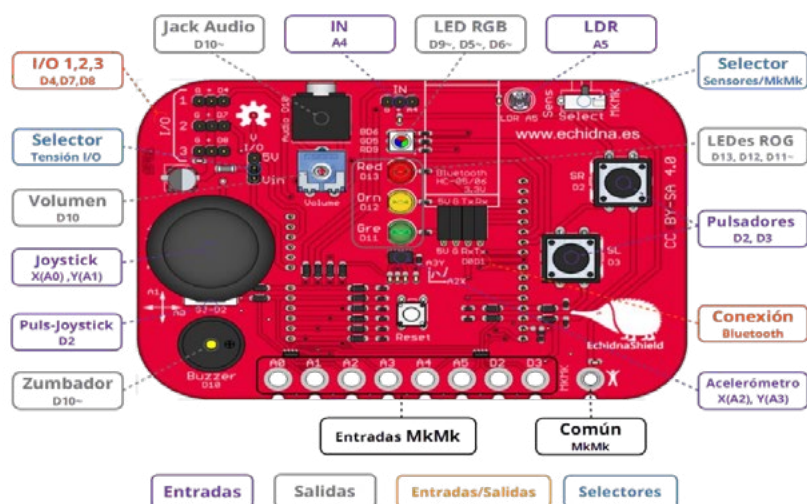
Una vez conectada, cualquier cosa que conduzca mínimamente la electricidad puede convertirse en un pulsador, permitiendo controlar juegos, aplicaciones... o incluso escribir como un teclado.

Aunque Makey Makey no necesita programación para funcionar, se puede combinar con entornos de programación como Scratch para explorar nuevas ideas y pensar nuevos modos de controlar nuestros proyectos de forma creativa. Scratch cuenta con bloques específicos para trabajar con la placa Makey Makey, pero puesto que sus conectores son detectados como teclas del teclado, se puede usar directamente en los proyectos con los bloques de pulsar la tecla correspondiente a sus conectores. En la web puedes encontrar un [curso](#) y diversos ejemplos de [actividades](#) para usar Makey Makey.

2.4. Echidna

[Echidna Educación](#) es un proyecto de *hardware* y *software* libre para facilitar el aprendizaje de la programación de sistemas físicos en Primaria y Secundaria. Tiene dos tipos de placas diseñadas por Xabier Rosas, profesor de Ciclos Formativos; Jose Pujol, profesor de Secundaria; y Jorge Lobo, maestro de Primaria, a partir de las experiencias y necesidades detectadas en sus aulas.

La primera, [Echidna Shield](#), es un escudo para Arduino, es decir, una placa con sensores y actuadores que se acopla a un Arduino. La segunda, [Echidna Black](#), es una placa autónoma, no se necesita Arduino, e incorpora algunos sensores más que la Echidna Shield. Ambas placas incluyen un modo MkMk para usar 8 conectores transformar objetos cotidianos en pulsadores como la placa Makey Makey.



Esquema de componentes [Echidna Shield](#), por [Echidna Educación](#), bajo licencia [CC-BY-SA](#)





Esquema de componentes Echidna Black, por [Echidna Educación](#), bajo licencia [CC-BY-SA](#)

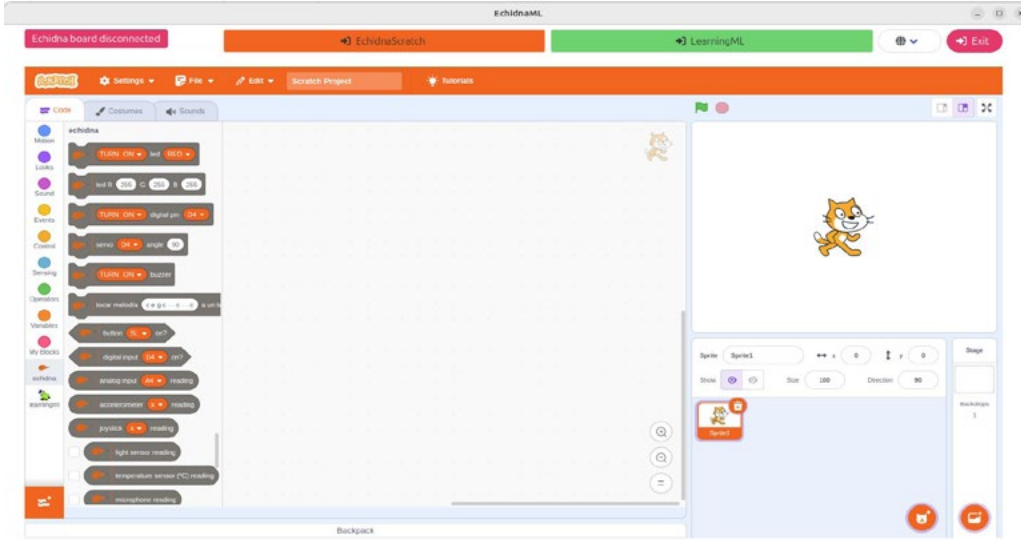
En la web del proyecto se puede encontrar mucha información sobre las placas y su uso didáctico como, por ejemplo [actividades](#) para Primaria, actividades para Secundaria con entornos gráficos, actividades para Secundaria con IDE de Arduino, y muchas otras actividades desarrolladas en [talleres](#) o explicadas en su [blog](#).

Programando placas Echidna

Al igual que las placas Arduino, las placas Echidna se pueden programar con IDE de [Arduino](#), y su versión [on-line](#), [Arduino blocks](#) o [Snap4Arduino](#), que además cuenta con una extensión de bloques específicos para Echidna.

- EchidnaML

Además, Echidna Educación tiene un *software* específico basado en Scratch: [EchidnaML](#), desarrollado por [Juan David Rodríguez](#). Este *software* incluye bloques específicos para estas placas y que incluye [LearningML](#), permitiendo integrar de esta forma la Programación, la Robótica y la Inteligencia Artificial en un mismo proyecto de forma sencilla.



[Echidna Educación](#) (2024). [EchidnaML](#) [Aplicación de escritorio



