

Aprendizaje del pensamiento computacional: operaciones, progresión y recursos

Manejar con precisión las operaciones que hay debajo de cualquier actividad de programación, saber qué es exigible a cada edad y por dónde se rompen las sesiones.

- 1 Las operaciones del pensamiento computacional y la depuración de errores
- 2 Progresión de los aprendizajes por edad y etapa educativa
- 3 Aportación diferencial de los recursos desconectados, los lenguajes de bloques y la robótica
- 4 Delimitación del enfoque STEAM frente a la actividad manipulativa sin contenido
- 5 Factores de fracaso de una sesión y su relación con el diseño didáctico

2.1. Las operaciones del pensamiento computacional y la depuración de errores



Las operaciones que hay debajo: descomponer, reconocer patrones, abstraer, algoritmizar y depurar.

Abre cualquier material escolar sobre pensamiento computacional y encontrarás lo mismo: cuatro recuadros de colores —descomponer, reconocer patrones, abstraer, algoritmos— y, a veces, un quinto con la depuración. La lista se repite tanto que parece un resultado de investigación. No lo es. Y saber de dónde sale importa, porque de ahí cuelga lo que pones en tu programación y lo que contestas cuando alguien pregunta qué se trabaja en esa sesión. Este epígrafe aclara el origen de esas listas y desarrolla las cinco operaciones con ejemplos de aula, incluida la que casi ningún material desarrolla.

2.1.1. Ninguna de las listas que circulan procede de un experimento

Lo primero es desactivar una falsa precisión: no existe ningún marco que enumere cuatro operaciones y añada la depuración como quinta. Esa partición es una simplificación didáctica, y las fuentes reales no coinciden ni en el número ni en el estatus de sus elementos.

La guía para docentes de Computing at School (Csizmadia y otros, 2015) enumera cinco conceptos —pensamiento algorítmico, descomposición, generalización, abstracción y evaluación— y deja la depuración aparte, como técnica. Barefoot enumera seis conceptos y además cinco enfoques, entre ellos depurar, «finding and fixing errors». El informe del JRC de la Comisión Europea (Bocconi y otros, 2016) enumera seis conceptos nucleares —abstracción, pensamiento algorítmico, automatización, descomposición, depuración y generalización— y es el único que pone la depuración al mismo nivel que la abstracción. Brennan y Resnick (2012) reparten siete conceptos, cuatro prácticas y tres perspectivas, con la depuración entre las prácticas. Y las cuatro piedras angulares que reproducen casi todos los materiales escolares proceden de la divulgación de la BBC y **no incluyen la depuración**.

Cuatro, cinco, seis o siete según quién escriba, y ninguna procede de un experimento: las dos primeras son materiales de formación docente, la del JRC es un informe de política construido sobre revisión y consulta a expertos, y Brennan y Resnick es una ponencia de congreso apoyada en treinta y una entrevistas a usuarios de Scratch. Aquí se trabaja con cinco porque es el reparto más manejable en el aula: decisión de este curso, no consenso del campo. La depuración entra porque el JRC la pone en igualdad con las demás y porque casi ningún material la desarrolla.

ATENCIÓN · «LAS CUATRO PIEDRAS ANGULARES Y LA DEPURACIÓN»

Es la formulación más repetida del sector y no la sostiene ninguna fuente: usarla es citar un material divulgativo como si fuera un marco. Se puede trabajar con cinco operaciones —aquí se hace—, diciendo que la agrupación es didáctica y no un hallazgo. Y un matiz más: **«reconocer patrones» no es una operación autónoma en los marcos originales**; en Computing at School y en el JRC está dentro de la generalización. Separarla es otra decisión de este curso.

De dónde salen las listas y con cuál trabaja este curso

Las listas que circulan (ninguna procede de un experimento)

Computing at School, 2015: cinco conceptos; depurar es técnica

Barefoot: seis conceptos y cinco enfoques; depurar es enfoque

JRC, 2016: seis conceptos nucleares, con la depuración entre ellos

Brennan y Resnick, 2012: siete conceptos, cuatro prácticas, tres perspectivas

BBC: cuatro piedras angulares, sin depuración

Las cinco operaciones de este curso (decisión didáctica)

Descomponer

Reconocer patrones

Abstraer

Algoritmizar

Depurar

Qué hace el alumnado en cada una

Parte el problema en trozos resolubles por separado

Ve lo que se repite y lo nombra

Decide qué información sobra

Escribe pasos que otro ejecuta sin interpretar

Localiza el error, propone la causa y la comprueba

Dos avisos que van con la figura

«Reconocer patrones» está dentro de la generalización en los marcos originales

Ningún marco enumera cuatro operaciones más la depuración

Fuente: elaboración propia a partir de Csizmadia y otros (Computing at School, 2015), Barefoot, Bocconi y otros (JRC, 2016) y Brennan y Resnick (2012). La agrupación en cinco operaciones es una decisión de este curso.

FIGURA 2.1

2.1.2. Descomponer y reconocer patrones

Descomponer es partir un problema en trozos que se puedan resolver por separado. Computing at School lo define como pensar los objetos «in terms of their component parts», y nada de eso exige un ordenador: es lo que haces al planificar una excursión y lo que hace un alumno de segundo cuando escribe cómo se lava los dientes y descubre que «coger el cepillo» son tres acciones y no una. La señal de que está trabajada es que el alumnado pueda decir qué trozo resuelve ahora y cuál viene después.

Reconocer patrones es ver lo que se repite y llamarlo por su nombre: advertir similitudes para predecir y crear reglas, en palabras de Barefoot. En un recorrido por la cuadrícula del suelo, el patrón es «tres pasos y giro» repetido cuatro veces. El salto ocurre cuando el alumnado deja de escribir la repetición entera y la nombra: ahí aparece el bucle, con o sin máquina delante.

2.1.3. Abstraer: la más difícil y la que menos se trabaja

Abstraer es decidir qué información es irrelevante. Computing at School lo define como hacer algo más comprensible «through reducing the unnecessary detail», y Barefoot, como identificar lo importante y dejar fuera lo que no se necesita. Dicho así parece sencillo; en el aula es la más difícil de las cinco, porque exige quitar, y quitar da miedo.

El ejemplo que mejor funciona es un plano del metro frente a un mapa de la ciudad: miente en las distancias a propósito, porque para lo que sirve —saber dónde cambiar de línea— esa información sobra. Y es la que menos se trabaja porque no deja producto vistoso: no hay cartulina de la abstracción. Para hacerla visible, pide el descarte por escrito: qué datos del enunciado no vas a usar, y por qué.

2.1.4. Algoritmizar: que otro lo ejecute sin interpretarlo

Algoritmizar es escribir la secuencia de pasos de modo que otro pueda ejecutarla sin interpretarla. Barefoot define el algoritmo como «an exact sequence of instructions, or set of rules, for performing a task», y la palabra que hace el trabajo es *exact*. La prueba de aula no cuesta nada: un equipo escribe las instrucciones y otro las ejecuta al pie de la letra, sin arreglar lo que falte. Cuando el segundo se queda parado porque en el paso cuatro no dice hacia dónde girar, la clase acaba de descubrir por qué el detalle importa, y sin enchufar nada. Ahí se abre la quinta operación: el algoritmo falla y hay que encontrar dónde.

2.1.5. Depurar: lo que hay, y no más

Depurar es localizar el error, formular una hipótesis sobre su causa, probar y volver a probar. Es la operación con más evidencia acumulada de las cinco, así que aquí se da lo que hay y no más.

El metaanálisis de Sun, Yang y Becker (2024), sobre 62 tamaños de efecto procedentes de 18 artículos, sitúa el efecto medio de las intervenciones sobre la destreza de depuración en $g = 0,64$, con un intervalo de confianza del 95 % entre 0,32 y 0,96 [E]. El intervalo es ancho: hay mucha variación entre estudios, y eso es parte del resultado. Lo que mejor funciona es la **instrucción sistemática del procedimiento ($g = 0,93$)**; lo que no funciona en absoluto es mejorar los **mensajes de error ($g = 0,04$, efecto nulo)**. Los autores declaran heterogeneidad alta y posible sesgo de publicación. La lectura para tu aula: no se trata de dar mejores avisos, sino de enseñar el procedimiento.

Y el único ensayo con grupo de control sobre enseñanza explícita de un procedimiento de depuración en un aula escolar que este curso ha podido comprobar es pequeño: Michaeli y Romeike (2019), 28 alumnos en el contraste principal, de 15 a 16 años, con una intervención de unos diez minutos dentro de una sesión de noventa y en lenguaje textual [E]. Da efectos significativos en autoeficacia y en rendimiento depurando, y sus autores escriben que la depuración está infrarrepresentada en el aula y en la investigación. No autoriza a generalizar a Infantil, a Primaria ni a los entornos por bloques.

2.1.6. Por qué la depuración es contenido y no contratiempo

Nadie lo ha medido, y aun así se sostiene [P]. Cuando el docente se acerca, ve el fallo y lo arregla, la sesión termina con el producto funcionando y el aprendizaje sin ocurrir: el alumnado se lleva un recorrido correcto y ninguna estrategia. Si la depuración es contenido, entra en la planificación como todo lo demás —minutos asignados, procedimiento nombrado en voz alta— y el error deja de ser el imprevisto que se come el final de la clase. Los demás puntos de rotura de la sesión son del epígrafe 2.5; los instrumentos, del **curso 07**.

CASO DE AULA · PRIMARIA, TERCER CICLO, MATEMÁTICAS

Dos equipos escriben las instrucciones para dibujar una figura en una cuadrícula y las intercambian. El ejecutor las sigue literalmente y la figura sale abierta por un lado. El docente ve el fallo desde su sitio: falta el último tramo. Si lo dice, la sesión se salva y el aprendizaje se pierde. Si pregunta «¿en qué paso deja de ir bien?», el equipo vuelve a leer, señala la línea y propone una causa. Ese minuto es el contenido.

Aplicación práctica

1. **Localiza las cinco operaciones en tres actividades que ya usas.** Coge tres actividades de tu programación —de cualquier área, con o sin dispositivo— y haz una tabla de cinco columnas: **lo haces solo**. Marca en cada una qué operaciones aparecen de verdad y con qué gesto del alumnado se manifiestan. Las casillas vacías son el resultado útil: casi siempre faltan abstraer y depurar. Guárdalo como [operaciones-en-mis-actividades.md](#). **Una hora.**
2. **Rediseña una actividad para que la depuración sea el contenido.** Elige la de esas tres con más margen y reescribe su desarrollo con tres cosas por escrito: **lo haces solo**. En qué momento se provoca el error, cuántos minutos se reservan para localizarlo y qué hace el alumnado en esos minutos —leer, señalar el paso, proponer la causa, probar—. Añade qué se quita de la sesión para que quepan. Guárdalo como [actividad-con-depuracion.md](#). **Una hora.**
3. **Escribe tu guion de tres preguntas.** Redacta las tres preguntas exactas que harás cuando veas el error antes que el alumnado, para no arreglárselo: **lo haces solo**. Una para localizar el paso, otra para formular la causa y otra para decidir cómo comprobarla. Pruébalas en una sesión y anota debajo cuál no funcionó y por qué. Guárdalo como [guion-de-tres-preguntas.md](#). **Una hora.**

2.2. Progresión de los aprendizajes por edad y etapa educativa

Qué puede hacer un niño de cuatro años, uno de ocho y uno de catorce.

El error más caro de esta materia no es comprar mal: es pedir de más. Se lleva al aula de cuatro años una secuencia con un bucle dentro, la sesión se rompe, el docente concluye que «esto no funciona a esta edad» y el material vuelve al armario dos cursos. Lo mismo pasa arriba, cuando se le pide a 1.º de ESO que maneje variables como si las trajera aprendidas de Matemáticas. Este epígrafe recorre lo que sostiene cada tramo separando dos cosas: lo que **prescribe el currículo español**, que es obligación, y lo que **han medido unos pocos estudios**, que es información con su muestra y sus límites. Ninguna de las dos es una ley del desarrollo.

2.2.1. Infantil: la secuencia corta, y el dato de los cuatro años que sí existe

El Real Decreto 95/2022 no le pide a esta etapa ni algoritmos ni robots: esas palabras no están en la norma. Su criterio 2.5 del segundo ciclo pide «Programar secuencias de acciones o instrucciones para la resolución de tareas analógicas y digitales, desarrollando habilidades básicas de pensamiento computacional». Secuencia de acciones, con el cuerpo o con tarjetas, sin exigencia de lectura ni de escritura.

Sobre lo que de verdad sostiene un niño de cuatro años hay un dato, y es más modesto que el que circula. Sullivan y Bers (2015) evaluaron por tareas individuales a sesenta niños de Pre-K a 2.º tras ocho semanas con un kit de robótica tangible de bloques de madera, sin pantalla, a una hora por semana [E]. El grupo de Pre-K —quince niños, en torno a los cuatro años— dominó la secuencia corta, con una media de 5,2 sobre 6, y **no llegó a bucles, condicionales ni sensores**; en la secuencia larga bajó a 3,4 sobre 6, con más apoyo adulto y más repetición. Es un estudio sin grupo de control: da techo, no eficacia.

ATENCIÓN · «HAY ESTUDIOS QUE DICEN QUE A LOS CUATRO AÑOS YA PROGRAMAN»

La frase procede del resumen de Bers y otros (2014), que escribe «children as young as four». Pero la muestra de ese trabajo son 53 alumnos de kindergarten, de cinco y seis años. El dato de cuatro años verificable es el de Sullivan y Bers (2015), y dice casi lo contrario de lo que se le hace decir: secuencia corta sí, bucles no. Si citas una cifra en un claustro, cita esa, con su muestra de quince niños delante.

2.2.2. Primaria: la progresión la escribe la norma con tres verbos

Aquí no hace falta buscar marcos: la progresión está en el currículo. Real Decreto 157/2022, anexo II, Matemáticas, bloque D (Sentido algebraico), primer ciclo: «Estrategias para la interpretación de algoritmos sencillos». Anexo II, Matemáticas, bloque D, segundo ciclo: «Estrategias para la interpretación y modificación de algoritmos sencillos». Anexo II, Matemáticas, bloque D, tercer ciclo: «Estrategias para la interpretación, modificación y creación de algoritmos sencillos». Interpretar, modificar, crear, en ese orden.

Los paréntesis de ejemplos de la norma marcan el otro escalón: los bucles y los patrones repetitivos aparecen en el segundo ciclo, y las instrucciones anidadas y los condicionales solo en el tercero, con la creación propia. Pedir a primer ciclo que cree un algoritmo es pedir lo que el currículo no pide; pedir condicionales en segundo ciclo, también.

El dato empírico añade una noticia que ningún folleto cuenta. En Sullivan y Bers (2015), Infantil, 1.º y 2.º rindieron de forma equivalente entre sí en las tareas avanzadas, con medias de 4,5 a 6,0 sobre 6; los bucles con parámetro se movieron entre 4,36 y 5,0, y los de sensor entre 3,69 y 4,55; los condicionales solo los abordó 2.º. La única diferencia significativa por curso apareció en la secuencia larga ($p = 0,00023$). Entre los cinco y los ocho años el rendimiento en estas tareas apenas se mueve: el salto grande está antes del tramo, no dentro.

Qué sostiene cada tramo de edad

4 años	Pre-K: secuencia corta Media de 5,2 sobre 6 en secuencia corta y 3,4 en la larga, sin bucles ni condicionales (n = 15).
5-8 años	Infantil a 2.º: apenas se mueve Infantil, 1.º y 2.º rinden de forma equivalente; los condicionales solo los aborda 2.º.
1.er ciclo	La norma pide interpretar «Estrategias para la interpretación de algoritmos sencillos». Crear no se pide aquí.
2.º ciclo	Interpretar y modificar Aparecen los bucles y los patrones repetitivos, siempre como ejemplos entre paréntesis.
3.er ciclo	Interpretar, modificar y crear Entran las instrucciones anidadas y los condicionales, y la creación propia.
12-16 años	ESO: algoritmia y variables Bloque «Pensamiento computacional, programación y robótica»; los bloques aventajan al texto solo al principio.

Fuente: elaboración propia a partir del Real Decreto 95/2022, el Real Decreto 157/2022 y el Real Decreto 217/2022, y de Sullivan y Bers (2015) y Weintrop y Wilensky (2017 y 2019).

FIGURA 2.2

2.2.3. Secundaria: la variable, la algoritmia y un salto que no se gana

El Real Decreto 217/2022 lleva el contenido a un bloque propio de Tecnología y Digitalización, titulado «Pensamiento computacional, programación y robótica», con la algoritmia y los diagramas de flujo, los sistemas de control programado y la robótica entre sus saberes. En Bachillerato el currículo estatal establece Tecnología e Ingeniería I y II, y lo que de esto entre allí depende de cada decreto.

Dos avisos para ese tramo. El primero es la variable: Grover y Basu (2017) documentan que el alumnado de 11 a 14 años arrastra de Matemáticas la idea de que una variable es una letra que representa un número desconocido y fijo, y por eso interpreta mal instrucciones del tipo «repetir (NúmeroDeVeces)» [E]. En programación es una caja que guarda un valor que cambia: dos conceptos distintos, y se enseñan como tales.

El segundo es el salto al texto, y lo miden los dos experimentos que verás enteros en el epígrafe 2.3: la ventaja inicial de los bloques sobre el texto se había igualado al cabo de quince semanas, y lo único que no se igualó fue el interés por seguir. Sesenta alumnos de un centro selectivo: una pista, no una ley. Pero deshace la idea de que el texto sea un escalón que se sube y se gana.

2.2.4. Los tres marcos de progresión, y que son extranjeros

Fuera del currículo español hay tres referencias de progresión verificables, y las tres son extranjeras [R]. El **currículo inglés de Computing (2013)**, del Departamento de Educación del Reino Unido, coloca en Key Stage 1 (5-7 años) «create and debug simple programs»; en Key Stage 2 (7-11) «use sequence, selection, and repetition in programs; work with variables»; y en Key Stage 3 (11-14) dos lenguajes, «at least one of which is textual». El *K-12 Computer Science Framework* (2016), estadounidense, se declara guía conceptual y no estándares ni currículo, y reparte en cuatro bandas: K-2, 3-5, 6-8 y 9-12. Los **estándares CSTA (2017)**, de una asociación profesional estadounidense, organizan los niveles 1A, 1B, 2 y 3A para todo el alumnado y reservan el 3B para estudios avanzados.

El contraste que interesa es este: el inglés sitúa secuencia, selección, repetición y variables entre los siete y los once años, antes de donde los coloca el reparto español, que deja los condicionales para el tercer ciclo. No es un modelo a importar: es una vara para ver que el techo por edad no es un hecho natural, sino una decisión curricular que cada país toma distinta.

2.2.5. El techo razonable y las señales de que has pedido de más

El techo de cada tramo no está medido en ninguna parte: se fija con dos preguntas: qué verbo usa tu currículo para esa edad —interpretar, modificar o crear— y cuántos elementos nuevos entran a la vez en la misma sesión.

Las señales de que se ha pedido de más se reconocen sin instrumento ninguno: el alumnado copia la solución del equipo de al lado en lugar de probar; pregunta «¿está bien?» antes de ejecutar nada; abandona al segundo intento fallido; o termina con un producto que funciona y no sabe explicar por qué. Cuando aparezcan, la corrección no es bajar la exigencia del curso entero: es quitar de esa sesión un elemento nuevo —el aparato, el concepto o el formato de trabajo— y dejar los otros dos.

CASO DE AULA · 1.º DE ESO, TECNOLOGÍA Y DIGITALIZACIÓN

Un grupo de 1.º lleva tres sesiones interpretando secuencias de pasos y diagramas de flujo. El docente propone crear un programa con una variable dentro porque «ya la traen de Matemáticas». La mitad la usa como incógnita fija y la otra mitad se bloquea. No es fallo del alumnado: es el arrastre que documentan Grover y Basu. Se recupera volviendo al verbo: se les da el programa hecho y se les pide que lo interpreten en voz alta y cambien un valor.

Aplicación práctica

1. **Construye la escala de progresión de tu etapa.** Abre el currículo de tu comunidad y copia, para tu curso y para el anterior y el posterior, el saber básico y el criterio que tocan programación o pensamiento computacional: **lo trabajas con material público**. Debajo de cada uno escribe tres columnas —qué se pide con la letra de la norma, qué es techo a esa edad y qué sería pedir de más—, y en la de techo anota la fuente cuando la haya. Guárdalo como [escala-de-progresion.md](#). **Una hora y media.**
2. **Contrasta tu escala con uno de los tres marcos extranjeros.** Elige el currículo inglés de Computing (2013), el *K-12 Computer Science Framework* (2016) o los estándares CSTA (2017) y pon en dos columnas, edad por edad, lo que coloca cada uno: **lo trabajas con material público**. Marca los dos o tres puntos donde no coinciden —bucles, condicionales, variables— y escribe en una línea qué harías con esa diferencia, recordando que ese marco no te obliga. Guárdalo como [contraste-marco-extranjero.md](#). **Una hora.**
3. **Revisa una propuesta tuya que pedía más de lo que la edad sostiene.** Busca en tus materiales del curso pasado una sesión que se rompió y descríbela en diez líneas: qué pediste, con qué material y qué pasó: **lo haces solo**. Cuenta cuántos elementos nuevos había a la vez, compárala con tu escala del documento 1 y reescribe la consigna quitando uno solo de esos elementos. Guárdalo como [revision-de-sesion-rota.md](#). **Una hora.**

2.3. Aportación diferencial de los recursos desconectados, los lenguajes de bloques y la robótica



Qué enseña el papel, qué enseña el bloque y qué solo enseña la máquina.

La pregunta llega siempre igual: «¿esto lo hago en papel o reservo el aula?». Y casi siempre se contesta por lo que hay: se reserva si queda hueco y se hace en papel si no queda. Hay otra manera. Cada escalón —el papel, la programación por bloques, el robot— añade algo que el anterior no tiene y cobra un precio de gestión que no está escrito en ninguna programación. Saber qué añade y qué cuesta cada uno te permite decidir en septiembre, con el calendario delante, y no en la puerta del aula de informática.

2.3.1. La pregunta que sí se puede contestar

«¿Cuál de los tres enseña más?» no tiene respuesta verificada, y el epígrafe 1.4 ya explicó por qué: las dos cifras del metaanálisis comparativo de 2025 van cada una contra su propio grupo de control, no proceden de una prueba de superioridad y no se restan. Aquí no se compara eficacia. Se comparan dos cosas que sí puedes mirar tú: **qué añade cada escalón que el anterior no tenía y qué cuesta gestionarlo antes de que empiece la sesión.**

Recuerda de dónde salen los tres. En el currículo aparecen dentro de un paréntesis de ejemplos y unidos por una disyuntiva —recursos analógicos o digitales—: obligatorio es el saber básico, y eso se vio con la letra de la norma en el epígrafe 1.2. De ahí la regla de este curso: se empieza por lo desenchufado porque el papel sostiene el saber básico entero, y el escalón se añade cuando aporta algo que el papel no puede dar.

2.3.2. El papel: lenguaje, discusión, cuerpo y coste cero

Lo que el papel añade no lo añade ningún otro escalón. El algoritmo se dice en voz alta, se discute, se reescribe encima y se ejecuta con el cuerpo en el suelo del aula. No hay cuentas que crear ni reserva que pedir: el coste de gestión es cero y la semana la eliges tú.

Y hay un resultado que lo mantiene en pie, el del epígrafe 1.4: empezar por lo desenchufado no cuesta comprensión de conceptos, y el grupo que empezó así salió con más autoeficacia y usó por su cuenta más bloques distintos [E]. Treinta y cinco niños de un colegio: una pista, con la muestra delante.

Lo que el papel no da es el ejecutor. Quien ejecuta es un compañero, y un compañero interpreta: entiende lo que querías escribir. Este curso sostiene que, sin una máquina que ejecute literalmente lo escrito, la depuración auténtica no tiene dónde ocurrir [P]. Nadie ha comparado la depuración en los dos entornos: es un razonamiento, no un hallazgo.

2.3.3. El bloque: el retorno inmediato y la mochila que trae

Lo que añade el entorno por bloques es eso: una máquina que hace lo que pone, al instante. El error aparece solo, sin que tengas que señalarlo, y ahí la depuración deja de ser un contratiempo y se convierte en el contenido de la sesión. Y probar es barato: se cambia un bloque y se ejecuta otra vez.

La mochila que trae es de gestión: dispositivo disponible, a menudo cuenta de usuario —y la decisión sobre cuentas de menores es del centro y no tuya, como se trabaja en el epígrafe 3.2— y un hueco en un aula que rara vez es la tuya. Trae también una demanda lectora que su propio diseño reconoce: los autores del entorno por bloques pensado para prelectores escribieron que la sintaxis textual resulta frustrante para quien todavía aprende a leer, y por eso mezclaron iconos y palabras [R]. No es un dato medido, sino la razón de diseño declarada por quien lo construyó; por dónde rompe la sesión se desarrolla en el epígrafe 2.5.

ATENCIÓN · SUBIR DE ESCALÓN NO REGALA APRENDIZAJE

Weintrop y Wilensky compararon dos entornos de programación, bloques y texto, con sesenta alumnos de un centro selectivo. En el trabajo de **2017**, a las cinco semanas, los bloques aventajaban al texto: 66,6 % frente a 58,8 %. En el de **2019**, a las quince semanas y en lenguaje textual, la ventaja ya no estaba: 64,9 % frente a 64,7 % [E]. Lo que no se igualó fue el interés por seguir. No compara papel con pantalla, pero desactiva la idea de fondo: un escalón no se sube y se gana.

2.3.4. El robot: el error visible en el espacio y los minutos que nadie ha contado

El robot añade algo que ni el papel ni la pantalla dan: el error ocurre en el espacio y lo ve el grupo entero. Se sale de la alfombra, choca con la pata de la mesa, gira noventa grados donde había que girar ciento ochenta. No hace falta explicar que hay un fallo.

El coste también es distinto: baterías, emparejamientos, piezas que se pierden y turnos para que no haya cinco manos sobre el mismo aparato. De esos minutos no existe un solo dato publicado; es experiencia de aula y no se le pone cifra [P]. Y el robot no rebaja el techo de la edad: a los cuatro años el techo es la secuencia corta con aparato o sin él, como viste en el epígrafe 2.2.

Sobre lo que el aparato produce por sí mismo, el epígrafe 1.3 ya dio las cifras de los dos metaanálisis de 2024 y explicó por qué conviven. Aquí basta una de ellas: medida dentro de la educación STEM, la robótica no mejora de forma significativa el pensamiento computacional ($g = 0,079$; $p = 0,434$) [E]. El resto de las cifras —el efecto global, el de rendimiento de aprendizaje y el del metaanálisis que aísla los robots— está en el 1.3, y allí queda dicho por qué lo que separa un resultado del otro es el recorte y el diseño de la actividad, no el aparato. Para este epígrafe basta la consecuencia: el robot no añade aprendizaje por estar encendido. Lo que añade es el error visible, y eso es un argumento de diseño, no de eficacia.

Qué añade y qué cuesta gestionar cada escalón

	Papel	Bloques	Robot
Qué añade que el anterior no tiene	Lenguaje, discusión y cuerpo	Ejecución literal e inmediata	Error visible en el espacio real
Quién ejecuta el algoritmo	Un compañero, que interpreta	La máquina, sin interpretar	La máquina, moviéndose
Dónde aparece el error	Lo señala el grupo al discutir	En la pantalla, al ejecutar	En el suelo, y lo ve todo el grupo
Qué cuesta gestionar antes de empezar	Nada: sin cuentas ni reservas	Dispositivo, cuentas y reserva de aula	Baterías, emparejamientos, piezas y turnos
Qué exige además del contenido	Turno de palabra y espacio en el aula	Lectura: su propio diseño la reconoce	Reparto de papeles y sitio despejado
Qué dice la evidencia verificada	Empezar sin enchufar no cuesta comprensión de conceptos	La ventaja sobre el texto se iguala a las quince semanas en lenguaje textual	Dos metaanálisis de 2024 con recortes distintos y resultados distintos

Fuente: elaboración propia a partir de Hermans y Aivaloglou (2017), Weintrop y Wilensky (2017 y 2019), Hong (2024), Ouyang y Xu (2024), Wang, Fan y Yu (2025) y la razón de diseño declarada de los entornos por bloques para prelectores.

FIGURA 2.3

2.3.5. Cuándo se sube, cuándo se baja y quién manda en el calendario

Subes cuando lo que falta es lo que el escalón añade: ejecución literal para que haya error propio, o error visible en el espacio. No subes para mejorar el rendimiento, porque eso no está medido así: el metaanálisis de 2025 encuentra equivalencia en Educación Primaria, en materias STEM y en intervenciones de más de cuatro semanas, y ventaja de lo enchufado en Secundaria y en intervenciones cortas [E]. Con esas cifras delante, subir lo decides tú y no el dato.

Bajas cuando el coste de gestión se come la sesión: aula ocupada, dispositivos sin carga, piezas que faltan. Bajar no es renunciar al saber básico, es cambiar de soporte para el mismo contenido, y por eso la contingencia se escribe antes —el protocolo entero está en el epígrafe 3.3—.

Y una regla de calendario: el aula de informática está ocupada muchas más semanas de las que crees. Por eso la secuencia se diseña primero en papel, entera y con su anclaje curricular, y el escalón se inserta donde el material esté disponible. Se recorta el escalón; no se recorta el saber básico.

CASO DE AULA · PRIMARIA, SEGUNDO CICLO, MATEMÁTICAS

Una secuencia de cinco sesiones sobre instrucciones y bucles se diseña entera para el suelo del aula, con tarjetas y un alumno de ejecutor. En la tercera semana hay tabletas libres dos días: ahí entra la versión por bloques del mismo algoritmo, para que el error lo produzca la máquina. Si esos dos días caen en la excursión, la secuencia no se cae: se da entera en papel y el escalón se pospone al segundo trimestre.

Aplicación práctica

1. **Construye la tabla de los tres escalones para tu aula.** Tres columnas —papel, bloques, robot— y cuatro filas: qué añade que el anterior no tiene, qué cuesta gestionar antes de empezar, dónde aparece el error y qué exige a tu alumnado además del contenido: **lo haces solo**. Rellénala con lo que hay de verdad en tu centro, no con lo que dice el catálogo, y anota en cada celda de coste las semanas en que ese escalón está disponible. Guárdalo como [tres-escalones-mi-aula.md](#). **Una hora.**
2. **Escribe tu criterio de subida y de bajada.** Media página con dos listas: **lo haces solo**. En la primera, las tres condiciones que tienen que darse para subir un escalón en una secuencia tuya; en la segunda, las tres señales que te hacen bajarlo, con lo que pasa a hacer el grupo. Añade debajo el calendario del aula de informática de tu centro este curso, semana a semana, marcando las ocupadas. Guárdalo como [criterio-de-escalon.md](#). **Una hora.**
3. **Escribe la misma actividad en dos versiones.** Toma una que ya uses y redáctala dos veces, en papel y en pantalla, con los mismos pasos y el mismo saber básico: **lo trabajas con material público**. Debajo, en cinco líneas, anota qué cambia entre las dos —quién ejecuta, dónde aparece el error, qué hace el resto del grupo, qué tienes que preparar antes— y cuál de las dos darías la semana que viene y por qué. Guárdalo como [misma-actividad-dos-versiones.md](#). **Una hora.**

2.4. Delimitación del enfoque STEAM frente a la actividad manipulativa sin contenido



Qué es STEAM cuando es algo y qué es cuando es una cartulina con luces.

En el pasillo hay una exposición. Se llama «reto STEAM» y son doce cartulinas con una maqueta encima, un circuito con una bombilla que enciende al cerrar la pinza y un rótulo pintado a mano, y la memoria del curso dirá que se ha trabajado por proyectos. Si preguntas qué Matemáticas hay dentro, la respuesta honrada es que ninguna; y el rótulo no decide nada, porque la maqueta funciona igual sin él. Hay un criterio para distinguir cuándo esas cinco letras describen un trabajo real y cuándo son una etiqueta puesta encima, y sirve también para decirlo sin despreciar el trabajo de nadie.

2.4.1. El problema empieza en la definición, y está documentado

Antes de discutirlo con nadie, mira en qué terreno pisas. La revisión integradora de Perignat y Katz-Buonincontro (2019), sobre cuarenta y cuatro artículos publicados entre 2007 y 2018 —dieciocho empíricos, catorce descriptivos y doce de marco pedagógico—, encuentra «a myriad of definitions of the STEAM concept in general, a variety of interpretations for the “A” in STEAM»: no hay definición común ni acuerdo sobre qué abarcan «las artes» dentro de esos marcos. Es una revisión [R], no un experimento.

Y hace el hallazgo más útil para un docente: bajo la misma etiqueta conviven al menos cinco formas distintas de integrar disciplinas —transdisciplinar, interdisciplinar, multidisciplinar, cruzada e integración artística—. Cuando dos personas discuten si lo del pasillo «es STEAM» no suelen estar en desacuerdo sobre los hechos: usan dos de esos cinco modelos sin saberlo.

2.4.2. Tres niveles de integración, que es la simplificación que usa este curso

Esos cinco modelos son categorías de investigación y en una reunión de coordinación no sirven. Este curso los agrupa en tres niveles, y lo dice como lo que es: una decisión didáctica propia [P], no una clasificación tomada de ninguna fuente.

Nivel 1: las áreas se tocan y no se mezclan. Cada materia hace lo suyo sobre un tema común: en Tecnología se monta el circuito, en Plástica se pinta el rótulo y en Matemáticas, con suerte, se mide algo. Es lo que suele haber detrás de una semana cultural, y coordinarla ya cuesta trabajo. Lo que no cabe es llamarlo integración.

Nivel 2: las áreas comparten un problema. Hay una pregunta única y cada área aporta algo sin lo cual no se resuelve: la maqueta no se sostiene si no se calcula, el circuito no enciende si no se entiende, y el rótulo no es adorno sino la manera de que un visitante entienda qué hace el aparato. Para subir del nivel 1 al 2 hace falta una sola cosa, y no es material: que el problema se formule antes de repartir las áreas.

Nivel 3: el problema no pertenece a ninguna área. Es el que menos se ve: el alumnado encuentra algo que no viene etiquetado —el agua que se acumula en el patio, el ruido del comedor— y decide él qué conocimientos hay que traer. Hace falta tiempo lectivo de más de un área, acuerdo entre profesorado y tolerancia a que el resultado no sea el previsto; sin esas tres condiciones, planificarlo acaba en un nivel 1 con mala conciencia.

Tres niveles de integración, y qué hace falta para subir

STEAM: qué hace cada área dentro del trabajo

Nivel 1. Se tocan y no se mezclan

- Un tema común, cada área por su lado
- Es lo típico de una semana cultural
- Coordinar ya cuesta trabajo
- No se puede llamar integración

Nivel 2. Comparten un problema

- Una sola pregunta para todas las áreas
- Sin lo que aporta cada una no se resuelve
- Para subir: formular antes de repartir
- No hace falta material nuevo

Nivel 3. El problema no es de nadie

- Algo del centro que no viene etiquetado
- El alumnado decide qué saberes trae
- Para subir: tiempo de más de un área
- Y acuerdo entre profesorado

Prueba del algodón

- Se quita una letra y se mira qué pasa
- Si el reto sigue igual, esa letra decoraba
- Sin medir ni ajustar, no hay matemáticas
- Se pasa por escrito, letra a letra

Fuente: elaboración propia a partir de Perignat y Katz-Buonincontro (2019) y Winner, Goldstein y Vincent-Lancrin (OCDE, 2013).

FIGURA 2.4

2.4.3. De dónde sale la A, y qué se puede afirmar de ella con honradez

La A se añadió al STEM con un argumento instrumental, y es el que vas a oír: que el arte mejora el aprendizaje de las demás materias. Detrás hay una revisión que casi siempre se te citará a medias, el informe *Art for Art's Sake?* de la OCDE (Winner, Goldstein y Vincent-Lancrin, 2013), que revisó la evidencia empírica sobre el impacto de la educación artística —música, plástica, teatro y danza, enseñanza integrada con las artes y estudio artístico fuera del horario— en los resultados del alumnado [R].

Resultado por resultado, y son cuatro. No hay evidencia de que la formación en artes plásticas mejore destrezas verbales o matemáticas. No la hay todavía de que la educación musical tenga impacto causal sobre las puntuaciones en matemáticas. No la hay de vínculo entre la formación teatral y las destrezas académicas generales. Y el puñado de estudios experimentales multiartísticos no muestra impacto significativo sobre el rendimiento. Donde el estudio es correlacional, avisan los autores, no puedes descartar explicaciones que no sean causales.

Hay una excepción, acotada y fuerte, y la das igual que lo anterior: el informe sí encuentra evidencia sólida de que **el teatro en forma de dramatización de relatos en el aula refuerza las destrezas verbales**. Decir que el arte no transfiere nada es tan inexacto como decir que mejora las matemáticas.

Nada de esto te autoriza a sacar el arte del reto: el propio informe deja escrita la razón de que esté, y es que el valor de las artes para la experiencia humana basta para justificar su presencia en el currículo, haya o no transferencia. Retiras la justificación falsa, no la materia.

ATENCIÓN · «EL ARTE MEJORA LAS MATEMÁTICAS»

Es la frase que sostiene media docena de proyectos y se te cae con un dato: lo revisado por la OCDE en 2013 no encuentra impacto causal de la música sobre las puntuaciones de Matemáticas ni de la plástica sobre lo verbal o lo matemático. Si defiendes la A por lo que el arte aporta al trabajo, no pierdes esa discusión; si la defiendes con esa frase, sí.

2.4.4. Lo que no existe: una cifra de eficacia del STEAM

No se ha localizado ningún metaanálisis de eficacia de la educación STEAM equivalente a los que este manual usa para la programación, y por eso aquí no vas a encontrar ninguna cifra de eficacia del STEAM. Lo que hay es la revisión de Perignat y Katz-Buonincontro (2019), que documenta «an overall lack of reported learning outcomes in the areas of creativity, problem-solving, and arts education» y remata con la observación más incómoda: «Creativity is rarely, if ever, measured or assessed as part of STEAM education». La creatividad es la justificación más repetida de todo esto y casi nunca se mide. En el espacio editorial español, Bautista (2021) parte del mismo diagnóstico —la promoción del STEAM descansa sobre fundamentos empíricos relativamente débiles— y presenta un monográfico planteado para reforzarlos: quien dice que la evidencia es escasa no son los detractores, son quienes lo investigan.

La misma revisión añade un diagnóstico de aula que el bloque 4 recoge: los docentes se quedan a menudo con el producto final y pierden de vista el proceso creativo. Qué se mira y cuándo es del epígrafe 4.1; con qué instrumento se recoge, del **curso 07**.

2.4.5. La prueba del algodón

Queda el criterio práctico, y es oficio [P]: quitas una letra y miras qué pasa. Si quitas la A y tu reto sigue funcionando igual, la A era decoración. Si quitas la M y nadie tiene que medir, estimar ni ajustar, no había matemáticas: había una manualidad con instrucciones. Se pasa letra a letra y por escrito, porque de cabeza siempre sale aprobado.

No sirve para descalificar el trabajo de nadie, sino para saber qué se está evaluando: un nivel 1 bien hecho es una semana cultural decente, lo que no puede es figurar en la programación como si cubriera criterios de tres áreas. Cómo se construye un reto en el que las cinco letras trabajen, y cómo se ancla a un criterio de evaluación que ya existe, es el epígrafe 3.4.

CASO DE AULA · PRIMARIA, TERCER CICLO, CONOCIMIENTO DEL MEDIO Y PLÁSTICA

El reto era «construid una maqueta del sistema solar y explicadla». Sin la A se entendía igual; sin la M, también, porque nadie usaba la escala. Se cambió el encargo a «que quepa en el pasillo, que las distancias sean proporcionales y que un visitante de Infantil entienda qué ve». Ahora la M decide el tamaño de cada bola y la A, qué se rotula. Mismo material y mismas horas.

Aplicación práctica

1. **Pasa la prueba del algodón a un reto STEAM ajeno.** Busca uno publicado en la web de un centro, en una revista de profesorado o en un banco de recursos público: **lo trabajas con material público**. Escribe debajo cinco líneas, una por letra, con la misma pregunta: si quito esta letra, ¿qué deja de funcionar? Marca cuáles resisten y cierra con una frase sobre qué cambiarías para que la primera que sobra dejara de sobrar. Guárdalo como **prueba-del-algodon.md**. Una hora.
2. **Clasifica tres propuestas de tu centro en los tres niveles.** Coge tres trabajos que ya existan sobre papel —la semana cultural, un proyecto de tu programación y una propuesta de otro equipo—: **lo haces solo**. Sitúa cada uno en el nivel 1, el 2 o el 3 con una frase que lo justifique, y

añade qué le falta para subir: formular el problema antes del reparto, tiempo de más de un área o acuerdo entre profesorado. Guárdalo como [niveles-de-integracion.md](#). **Una hora y media.**

3. **Escribe tu respuesta a «el arte mejora las Matemáticas».** Media página dirigida a alguien concreto del claustro, con sus palabras: **lo haces solo**. Tiene que contener qué revisó el informe de la OCDE de 2013, los resultados que no encuentra, la excepción acotada de la dramatización de relatos sobre las destrezas verbales y por qué el arte se queda en el reto de todos modos. Que no termine desacreditando a nadie. Guárdalo como [respuesta-arte-y-matematicas.md](#). **Una hora.**

2.5. Factores de fracaso de una sesión y su relación con el diseño didáctico



Por dónde se rompe una sesión, y por qué casi nunca es por el robot.

La sesión estaba preparada. El material funcionaba, el reto era bueno y el grupo entró con ganas. Y a los cuarenta minutos hay dos equipos que no han entendido qué se pedía, uno que lleva el rato viendo trabajar a un compañero y ninguno que haya localizado un error propio. La explicación de la sala de profesores es siempre la misma: falló el material. Casi nunca es eso. Las sesiones se rompen por cinco sitios identificables, todos anteriores al aparato y previsibles la semana de antes. Van aquí en orden, cada uno con su contramedida y con el respaldo que tiene esa contramedida.

2.5.1. Tres cosas nuevas a la vez

El primer punto de rotura es el más común y el que menos se ve: pedirle al alumnado que resuelva un problema que no conoce, con una herramienta que no conoce y con un procedimiento que tampoco conoce. Las tres novedades compiten por la misma capacidad limitada.

Aquí hay que decir qué se está haciendo. El mecanismo es el de la memoria de trabajo, y su estudio es propiedad del **curso 17**: allí se remite. Lo publicado es que resolver problemas por análisis de medios y fines consume capacidad de procesamiento que deja de estar disponible para construir esquemas, y lo demostró Sweller (1988) con experimentos de geometría y álgebra, no de programación. No hay ninguna medición verificada de carga cognitiva en una sesión escolar de robótica. Así que este curso no cita un resultado: **aplica un mecanismo general a un caso que nadie ha medido**, y lo dice.

La revisión de referencia sobre aprendizaje de la programación es la de Robins, Rountree y Rountree (2003), y su literatura es universitaria, cosa que hay que decir. Añade la distinción que ordena la contramedida: el principiante falla más por falta de estrategia que de conocimiento. Sabe qué es un bucle y no sabe cuándo usarlo. Y hay un supuesto que carga la sesión sin que nadie lo note: que la variable «ya la saben de Matemáticas». El epígrafe 2.2 documenta lo contrario, y ese arrastre gasta capacidad en cada instrucción de repetición que el alumnado lee mal.

La contramedida cabe en una línea: **estrena una cosa por sesión**. Herramienta nueva, problema conocido; problema nuevo, herramienta conocida. Y el procedimiento —cómo se prueba, cómo se depura— se enseña aparte, antes, en papel.

2.5.2. La lectura que el entorno no anuncia

El segundo punto es la demanda lectora de los entornos por bloques. Un bloque es un texto corto, y quien todavía descifra gasta en leerlo lo que necesitaba para pensar. Eso no está medido en ningún estudio verificado por este curso, así que no se escribe «está demostrado». Lo que sí hay es la razón de diseño que declararon por escrito quienes construyeron el entorno de bloques para prelectores: mezclaron iconos y palabras precisamente porque la sintaxis resulta poco intuitiva y frustrante para los novatos, y eso «se agrava en los niños pequeños que todavía están aprendiendo a leer».

Y hay que quitar la coartada de que el alumnado «ya viene sabiendo» porque ha crecido con pantallas: por qué eso no se sostiene es del **curso 04**. Aquí basta la consecuencia: manejar una interfaz no es leer una consigna. Lo mismo con «programar reconfigura el cerebro», los hemisferios, las ventanas que se cierran y la pirámide de retención: se nombran para descartar, y

su desmontaje es del **curso 17**. Contramedidas: consigna en ícono y palabra, leída en voz alta; tarjetas del reto con dibujo además de texto; y los cuatro o cinco bloques que se usarán, en la pizarra antes de encender nada.

2.5.3. Las manos, en Infantil

El tercer punto es motriz, y en Infantil decide la sesión entera. Arrastrar un bloque a un sitio exacto, encajar dos piezas o pulsar una secuencia sin saltarse un paso son destrezas finas sin terminar a los cuatro años, y la sesión se va en eso.

Aquí el dato del epígrafe 2.2 vale, con la cautela de que mide secuenciación y no destreza fina: a los cuatro años se domina la secuencia corta, se cae en la larga y no se llega a bucles ni a condicionales, con quince niños de muestra. La contramedida es de diseño: secuencias de tres o cuatro pasos, piezas grandes y el cuerpo antes que el dedo.

CASO DE AULA · EDUCACIÓN INFANTIL, SEGUNDO CICLO

Veintiún niños de cinco años, una cuadrícula dibujada con cinta en el suelo y tarjetas de flecha. Hay que llevar al muñeco a la casilla del árbol. Tres equipos lo resuelven con cuatro tarjetas; el cuarto coloca seis y no se aclara. La maestra no corrige la secuencia: retira dos tarjetas y pide que la recorran andando. El error aparece solo, y aparece con los pies. La contramedida no fue explicar mejor: fue acortar el reto.

2.5.4. Uno con el teclado y los demás mirando

El cuarto punto es el reparto de papeles. Sin gestión explícita, el equipo se organiza solo y siempre igual: quien va más suelto toma el mando y los demás pasan a espectadores. El que menos sabe es el que menos toca.

Lo medido es sobre parejas ante un ordenador, no sobre turnos ante un robot, y la diferencia importa. Denner y otros (2014), con 320 alumnos de Secundaria, encontraron ventaja de la programación por parejas, con un detalle aprovechable: quien más gana es quien menos sabe. Y otro contraintuitivo: quien mejor se llevaba con su pareja obtuvo menos conocimiento de programación, señal de que la pareja cómoda se reparte el trabajo en vez de discutirlo. El trabajo que buscó una mejora de actitud en las alumnas y no la encontró documenta, en cambio, la práctica que aquí sirve: los papeles rotaban con temporizador y el alumnado se cambiaba de sitio en cuanto sonaba. El protocolo completo, en el epígrafe 3.3.

La deriva por sexos dentro de ese reparto es propiedad del **curso 16**: allí se remite y aquí se para. La contramedida es la rotación temporizada, con los papeles escritos y visibles y con el reloj mandando en lugar del docente.

2.5.5. El reloj, y la depuración que no llega

El quinto punto es el tiempo, y es la causa número uno de que una sesión termine sin que nadie haya depurado nada. Los minutos se amontonan al principio —repartir, arrancar, emparejar— y la depuración, que es donde ocurre el aprendizaje, está al final: cuando se come el margen, lo que se cae es justo lo que se venía a hacer.

Cuántos minutos se van en eso no lo ha medido nadie: es la afirmación más repetida del ramo y la que no tiene un dato detrás, así que se escribe como experiencia de aula. Las contramedidas son de oficio y valen para cualquier material: el reto, escrito y sobre la mesa antes de que entre el grupo; el material, repartido y encendido antes; y un corte fijo, a falta de diez minutos, para la puesta en común del error. El protocolo de sesión con material compartido es del epígrafe 3.3.

Los cinco puntos de rotura, en su orden



Fuente: elaboración propia.

FIGURA 2.5

2.5.6. Cada contramedida, con su etiqueta

Casi todo lo anterior es oficio, y lo sostiene mejor decirlo. Las tres etiquetas del epígrafe 1.3 lo reparten. [E], evidencia empírica: el techo de los cuatro años, la ventaja de la pareja para quien menos sabe, el resultado nulo sobre la actitud y la rotación temporizada tal como quedó documentada en ese mismo trabajo. [R], revisión o marco: la distinción entre conocimiento y estrategia y la razón de diseño declarada del entorno para prelectores. [P], recomendación profesional: estrenar una cosa por sesión, leer la consigna en voz alta, acortar la secuencia y cortar a falta de diez minutos. Una contramedida [P] se defiende igual de bien ante el claustro, siempre que no se presente como hallazgo.

ATENCIÓN · PONER UNA CIFRA DE CARGA COGNITIVA A TU SESIÓN

Circula en presentaciones y suena técnico: «una sesión de robótica satura la memoria de trabajo en un tanto por ciento». No hay ninguna medición verificada de carga cognitiva en una sesión escolar de robótica, y Sweller no midió programación. Se nombra el mecanismo, se remite al curso 17 y se dice que se está aplicando. Una cifra inventada tumba el argumento delante del primero que la busque.

Aplicación práctica

1. **Prevé los cinco puntos de rotura de una sesión tuya.** Coge una sesión que vayas a dar este trimestre y recorre los cinco puntos en orden: **lo haces solo**. En cada uno, escribe por dónde se rompería en tu grupo —qué alumnos, qué momento, qué consigna— y la contramedida, redactada como una acción y no como un propósito. Guárdalo como [puntos-de-rotura.md](#). **Una hora y media.**
2. **Etiqueta tus contramedidas.** Repasa las cinco que acabas de escribir y ponle a cada una [E], [R] o [P], volviendo al epígrafe para ver qué respalda cada cosa: **lo haces solo**. Las que queden en [P] no se quitan: se reescriben empezando por «en mi aula funciona» y no por «está demostrado». Guárdalo como [contramedidas-etiquetadas.md](#). **Una hora.**

3. **Escribe el plan de reparto de papeles.** Define cuántos papeles hay, qué hace cada uno, cada cuánto rotan y quién avisa del cambio: **lo haces solo**. Incluye a quién pones con quién y qué haces con el alumno que se queda fuera en cuanto te descuidas. Guárdalo como [reparto-de-papeles.md](#). **Una hora.**