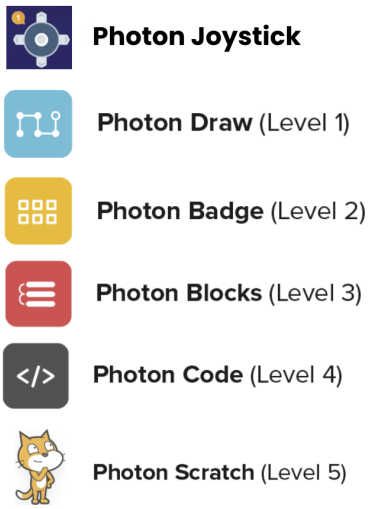
2. Niveles de programación de photon

1. Introducción a las distintas interfaces para programar PHOTON

Photon puede controlarse mediante distintas interfaces que permiten avanzar desde la acción directa hasta formas más estructuradas de programación. Esta progresión facilita que el [illegible]no más adecuado para cada actividad y para las características del



De menor a mayor nivel de abstracción, podemos establecer la siguiente secuencia orientativa:

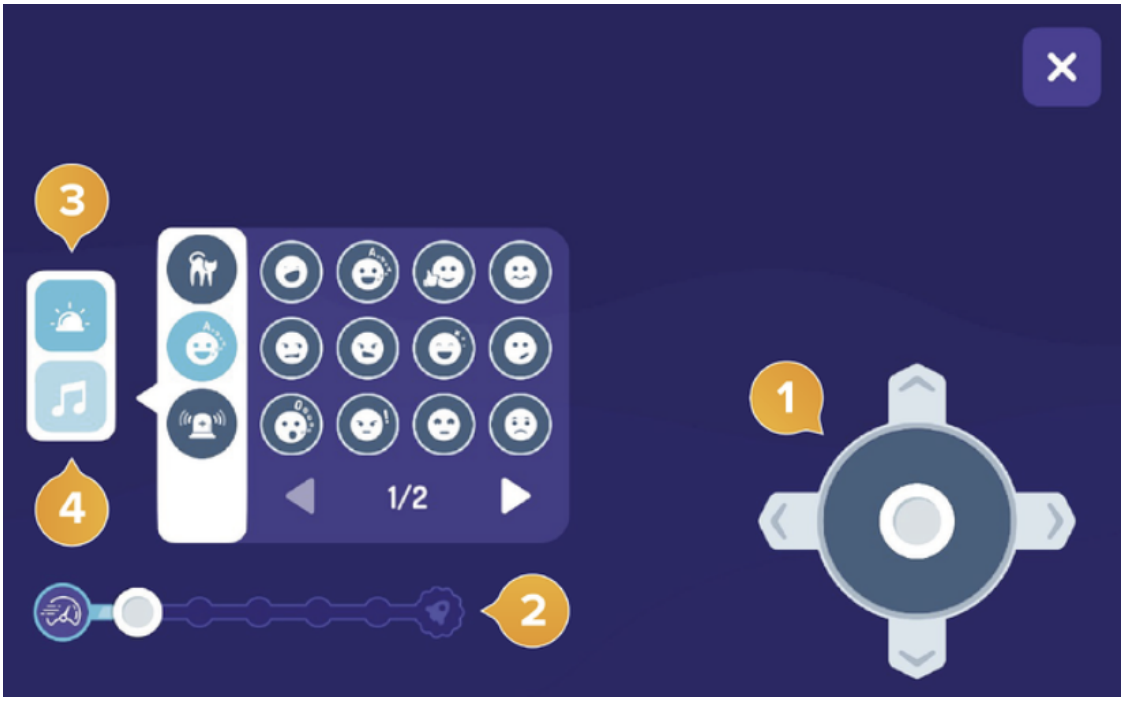
1. Photon Joystick: control directo e inmediato.
2. Photon Draw: programación mediante el dibujo de una trayectoria.
3. Photon Badge: secuenciación de instrucciones mediante iconos.
4. Photon Blocks: programación estructurada mediante bloques.
5. Photon Code: aproximación al código y uso de variables.
6. Scratch y otros lenguajes: ampliación hacia entornos externos de programación.

Esta secuencia no debe aplicarse de forma automática. Un mismo grupo puede utilizar Joystick para una actividad de comunicación, Badge para trabajar secuencias y Blocks para resolver un reto

con sensores. La interfaz debe elegirse por su utilidad pedagógica y no únicamente por su supuesta dificultad.

La aplicación Photon EDU permite, además, personalizar las acciones visibles dentro de una interfaz. El profesorado puede limitar las opciones disponibles y ofrecer únicamente los comandos necesarios para una actividad. Esta posibilidad resulta especialmente relevante desde una mirada inclusiva, porque reduce la sobrecarga visual y facilita una progresión ajustada.

2. Photon Joystick: control directo del robot



Joystick es la interfaz más básica. El alumnado controla el desplazamiento de Photon mediante un mando virtual en la pantalla. También puede modificar su velocidad, seleccionar el color de los ojos y las antenas y reproducir diferentes sonidos.

Las acciones se ejecutan inmediatamente. No es necesario diseñar previamente una secuencia ni pulsar un botón para ejecutar un programa.

¿Qué aprendizajes favorece?

Joystick permite realizar una primera exploración del robot y comprender la relación entre una acción en la pantalla y una respuesta física:

- “Desplazo el mando hacia delante y Photon avanza”.
- “Giro el mando y Photon cambia de dirección”.
- “Selecciono un color y se iluminan sus ojos y antenas”.

Esta respuesta inmediata permite trabajar la relación causa–efecto, la orientación espacial, el control inhibitorio, la coordinación óculo-manual y el seguimiento de instrucciones sencillas.

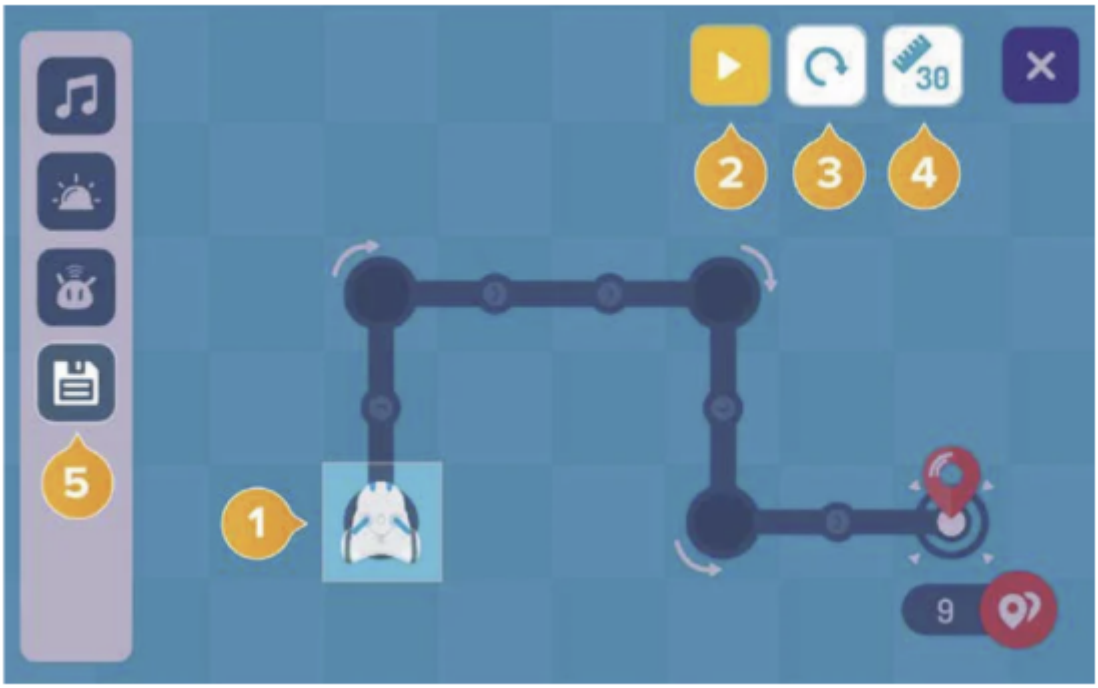
También resulta útil en actividades cuyo objetivo principal no es aprender programación. Por ejemplo, puede emplearse para acercar Photon a un compañero, elegir una tarjeta, recorrer diferentes espacios o participar en una dinámica comunicativa.

En Joystick se **controla** el robot, pero no se construye todavía un algoritmo. El alumnado toma decisiones durante el movimiento, sin anticipar necesariamente toda la ruta.

> Mirada inclusiva: Joystick reduce la demanda de planificación y memoria de trabajo, pero requiere cierto control motor sobre la pantalla. Para facilitar el acceso se puede:

1. Reducir la velocidad del robot.
2. Utilizar una tablet de mayor tamaño.
3. Estabilizar el dispositivo sobre una mesa o soporte.
4. Permitir que una persona indique la dirección y otra maneje el mando.
5. Dividir la actividad en decisiones muy sencillas: avanzar, parar o girar.

3. Photon Draw: dibujar para programar



Draw permite programar el recorrido de Photon dibujando una trayectoria directamente sobre la pantalla. El alumnado arrastra el dedo desde la imagen del robot y representa el camino que quiere que siga.

La trayectoria no se ejecuta de forma inmediata. Primero se dibuja y después se pulsa el botón de reproducción para que Photon realice el recorrido.

En determinados puntos del camino se pueden añadir acciones, como reproducir un sonido, modificar el color de los ojos y las antenas o esperar hasta que se active uno de los sensores del robot.

También puede ajustarse la longitud que representa cada paso. La configuración inicial se corresponde con los campos de las alfombras educativas oficiales.

¿Qué aprendizajes favorece?

Draw introduce una diferencia fundamental respecto a Joystick: el alumnado debe **anticipar una trayectoria** antes de comprobar el resultado.

Esto permite trabajar:

1. Planificación espacial.
2. Relación entre representación gráfica y movimiento real.
3. Estimación de distancias.

4. Orientación y direccionalidad.
5. Predicción y comprobación de resultados.
6. Revisión del error.

Es una interfaz especialmente adecuada para dibujar formas, recorrer mapas, representar el desplazamiento de un personaje o conectar elementos distribuidos por una alfombra.

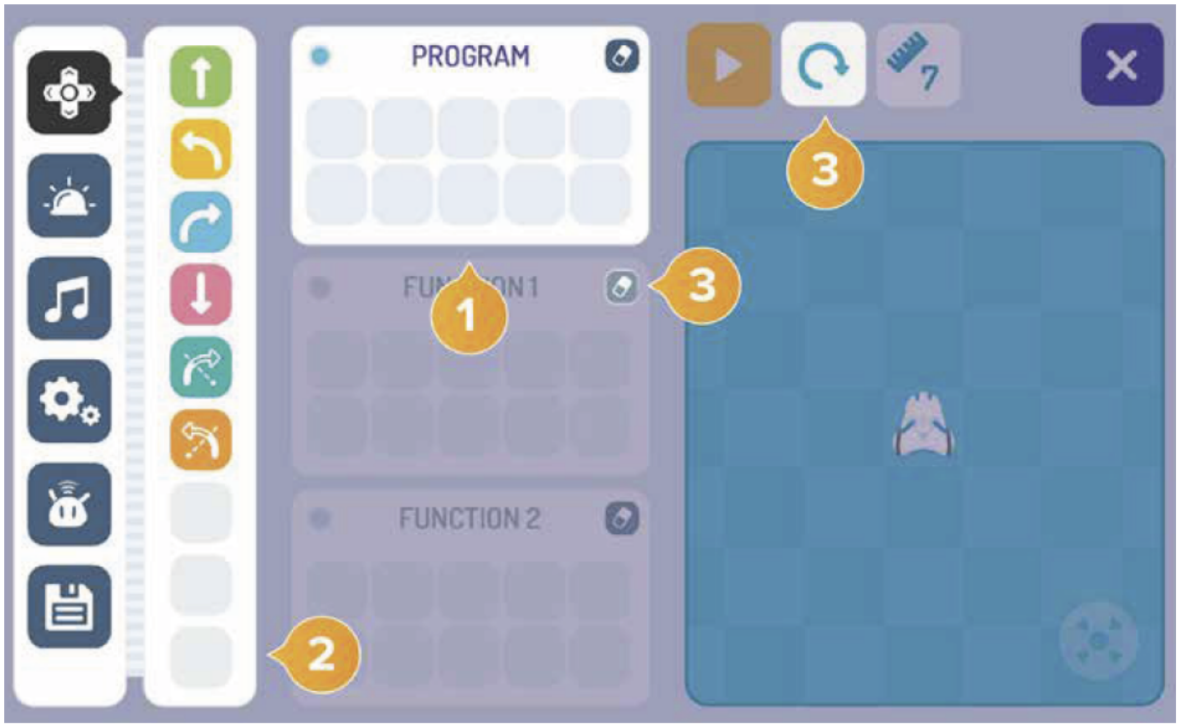
En Draw la programación se representa como un **camino continuo**, no como una lista de instrucciones independientes. Esto facilita una aproximación intuitiva, aunque puede hacer menos visibles los pasos concretos que componen el algoritmo.

> Mirada inclusiva: Draw puede ser muy accesible para alumnado que comprende mejor la información visual y global. Sin embargo, dibujar una trayectoria precisa puede generar barreras motrices o visoespaciales.

Para facilitar la participación se puede:

1. Utilizar un lápiz táctil grueso.
2. Dibujar caminos cortos y sencillos.
3. Marcar previamente el inicio y el destino.
4. Proporcionar una plantilla visual del recorrido.
5. Permitir que una persona describa el camino y otra lo dibuje.
6. Comparar el recorrido dibujado con flechas colocadas físicamente en el suelo.

4. Photon Badge: programar mediante símbolos



Badge introduce la programación secuencial mediante iconos o insignias. El alumnado selecciona instrucciones de movimiento, colores, sonidos o sensores y las coloca en una zona de programación.

Photon ejecuta las instrucciones en el mismo orden en el que aparecen. La interfaz permite construir secuencias de hasta diez acciones.

Las instrucciones se añaden arrastrándolas y pueden eliminarse individualmente. También se puede ajustar la longitud de cada paso, guardar el programa y recuperarlo posteriormente.

¿Qué aprendizajes favorece?

Badge hace visible el algoritmo como una secuencia formada por acciones diferenciadas:

- “Avanza”.
- “Gira”.
- “Cambia de color”.
- “Reproduce un sonido”.
- “Espera una interacción”.

El alumnado puede leer la secuencia de izquierda a derecha, anticipar el comportamiento del robot y localizar en qué instrucción se produce un error.

Esta interfaz permite trabajar especialmente:

1. Orden temporal.
2. Secuenciación.
3. Anticipación de acciones.
4. Correspondencia entre símbolo y respuesta.
5. Memoria de trabajo.
6. Detección y corrección de errores.

Badge supone el paso desde la representación global de Draw hacia una representación **discreta y secuenciada**. Cada símbolo corresponde a una instrucción concreta.

El límite de diez acciones reduce la complejidad y ayuda a trabajar con programas breves y comprensibles.

> Mirada inclusiva: Su lenguaje icónico reduce la dependencia de la lectura y la escritura, por lo que puede resultar adecuado para edades tempranas, alumnado con discapacidad intelectual o personas que utilizan apoyos visuales.

Para mejorar la accesibilidad se puede:

1. Mostrar solo las categorías necesarias.
2. Utilizar tarjetas físicas con los mismos iconos que aparecen en la aplicación.
3. Construir primero la secuencia sobre la mesa.
4. Leer la programación en voz alta antes de ejecutarla.
5. Reducir inicialmente el programa a dos o tres instrucciones.
6. Asignar diferentes roles: seleccionar el icono, colocarlo, anticipar la acción y comprobar el resultado.

5. Photon Blocks: programación estructurada

Blocks permite crear programas mediante bloques de instrucciones que se colocan bajo un bloque inicial. Photon ejecuta el programa de arriba abajo.

A diferencia de Badge, el número de bloques no está limitado y las instrucciones ofrecen más posibilidades de configuración. Se pueden determinar distancias, ángulos de giro, duración de las acciones y comportamientos diferenciados para los ojos y las antenas.

También permite introducir estructuras propias de la programación, como repeticiones, condiciones e instrucciones vinculadas a los sensores. Entre sus posibilidades se encuentra la función de seguir una línea negra mediante los sensores de contraste del robot.

¿Qué aprendizajes favorece?

Blocks permite pasar de las secuencias lineales a programas más complejos. Con esta interfaz se trabajan:

1. Algoritmos con mayor número de instrucciones.
2. Repeticiones y reconocimiento de patrones.
3. Condiciones del tipo “si ocurre esto, realiza esta acción”.
4. Uso de sensores y eventos.
5. Parametrización de distancias, tiempos y ángulos.
6. Depuración y optimización de programas.

Por ejemplo, para dibujar un cuadrado no es necesario colocar cuatro veces la misma secuencia. El alumnado puede identificar el patrón “avanzar y girar” e introducirlo dentro de una repetición.

Blocks permite programar no solo **qué debe hacer** Photon, sino también **cuándo, durante cuánto tiempo, cuántas veces o bajo qué condición** debe hacerlo.

Es la interfaz en la que el pensamiento computacional se hace más explícito, especialmente en las dimensiones de descomposición, reconocimiento de patrones, diseño de algoritmos y depuración.

> Mirada inclusiva:La ampliación de posibilidades también aumenta la cantidad de información en pantalla y la demanda de planificación. Para reducir barreras se puede:

1. Limitar previamente los bloques disponibles.
2. Introducir una sola categoría nueva en cada actividad.
3. Utilizar códigos de color y tarjetas físicas equivalentes.
4. Entregar parte del programa ya construida.
5. Dividir el reto en pequeños subproblemas.
6. Usar bloques de repetición únicamente después de experimentar corporalmente el patrón.
7. Distribuir roles dentro del grupo: planificación, programación, ejecución y comprobación.

6. Photon Code: acercamiento al código escrito

Photon Code presenta una estructura semejante a Photon Blocks, pero las instrucciones aparecen representadas mediante fragmentos de código en lugar de bloques puramente gráficos.

Esta interfaz permite, además, crear y modificar variables. Una variable puede almacenar un dato que posteriormente se utiliza o cambia durante la ejecución del programa.

¿Qué aprendizajes favorece?

Code actúa como puente entre la programación visual y los lenguajes de programación escritos. Permite que el alumnado empiece a reconocer:

1. La estructura formal de una instrucción.
2. La relación entre un bloque y su representación en código.
3. El uso de valores y parámetros.
4. La creación y modificación de variables.
5. La importancia de la precisión sintáctica y lógica.

Particularidad principal

La diferencia principal con Blocks no está necesariamente en lo que hace el robot, sino en **cómo se representa el programa**. El alumnado deja de apoyarse exclusivamente en iconos y empieza a interpretar fragmentos más próximos al lenguaje informático.

> Mirada inclusiva: Code exige más competencia lectora, mayor capacidad de abstracción y atención a detalles formales. No debe presentarse como una evolución obligatoria para todo el alumnado.

Puede facilitarse su acceso mediante:

1. Comparación simultánea entre un bloque y su código equivalente.
2. Programas breves y parcialmente completados.
3. Resaltado de las partes que cambian en cada instrucción.
4. Explicación visual de las variables.
5. Trabajo cooperativo por parejas.
6. Uso de ejemplos relacionados con programas ya realizados en Blocks.

7. Scratch y la ampliación hacia otros entornos

Photon EDU incorpora una interfaz específica de Scratch. Photon Coding también ofrece una versión de Scratch con bloques dedicados al robot.

En ordenadores, Photon Magic Bridge permite ampliar el trabajo a Scratch, MakeCode, JavaScript y Python mediante el adaptador correspondiente. De esta forma, Photon puede utilizarse como dispositivo físico dentro de entornos de programación más generales.

¿Qué aprendizajes favorece?

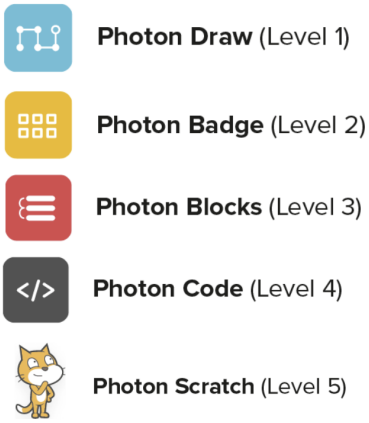
Scratch permite integrar el comportamiento físico de Photon con otros elementos digitales, como personajes, escenarios, sonidos o eventos. Esto abre la puerta a proyectos más amplios:

1. Historias interactivas en las que se coordinan un personaje digital y el robot.
2. Juegos que combinan acciones en pantalla y movimientos físicos.
3. Proyectos interdisciplinarios.
4. Programación por eventos.
5. Uso progresivo de condiciones, variables y operadores.

Photon deja de ser el único centro del programa y pasa a formar parte de un **sistema más amplio** , en el que pueden intervenir elementos físicos y digitales.

Desde una perspectiva inclusiva, conviene introducir estas posibilidades de forma gradual, evitando que la cantidad de bloques, menús y elementos en pantalla desplace el objetivo educativo de la actividad.

8. Comparación de las interfaces



En resumen, ¿cuál de todas las interfaces me conviene más usar?

En la siguiente tabla puedes hacer una comparación rápida de la forma de programación y habilidades requeridas para su uso:

Interfaz	Forma de control o programación	Anticipación requerida	Principal potencial didáctico	Posible barrera
Joystick	Control directo mediante mando virtual	Baja	Causa–efecto, orientación y exploración	Precisión motriz
Draw	Dibujo de una trayectoria	Media	Planificación espacial y predicción	Coordinación visomotriz
Badge	Secuencia de iconos, hasta diez acciones	Media	Orden, algoritmos breves y depuración	Memoria de trabajo
Blocks	Bloques configurables y estructuras de control	Alta	Patrones, bucles, condiciones y sensores	Sobrecarga visual
Code	Fragmentos de código y variables	Alta	Transición hacia programación escrita	Lectura y abstracción
Scratch	Bloques dentro de un entorno digital más amplio	Alta	Proyectos interactivos y programación por eventos	Complejidad de la interfaz

9. ¿Qué interfaz debemos elegir?

La selección no debería basarse solamente en la edad. Conviene considerar cuatro elementos:

1. **El objetivo de aprendizaje.** Si queremos trabajar comunicación, puede bastar Joystick o Badge. Si queremos comprender repeticiones, será más adecuado Blocks.
2. **La experiencia previa.** El alumnado necesita conocer el significado de las instrucciones antes de combinarlas en programas complejos.
3. **Las barreras de acceso.** Una interfaz aparentemente sencilla puede resultar difícil por sus demandas motrices, visuales, lectoras o cognitivas.
4. **El grado de autonomía esperado.** En algunas actividades el docente puede preparar el programa y el alumnado activar el robot; en otras, el alumnado debe diseñarlo completamente.

Una progresión posible sería comenzar experimentando el movimiento mediante Joystick, anticipar recorridos con Draw, construir secuencias con Badge, introducir estructuras mediante Blocks y aproximarse posteriormente a Code o Scratch.

No obstante, es posible regresar a una interfaz anterior cuando resulte más eficaz. Utilizar Joystick después de haber trabajado Blocks no supone retroceder: puede ser la opción más adecuada para una dinámica social, una actividad motriz o una experiencia de causa–efecto.

> Las interfaces de Photon no representan únicamente distintos grados de dificultad técnica. Cada una propone una manera diferente de pensar, representar y controlar una acción.

> El valor educativo no reside en llegar cuanto antes a la interfaz más avanzada, sino en seleccionar aquella que permita al alumnado comprender el reto, tomar decisiones, participar y comprobar el efecto de sus acciones.

Revision #3
Created 2026-06-22 20:15:45 CEST by Jorge CATEDU
Updated 2026-06-22 22:26:22 CEST by Jorge CATEDU