

J. Ángel Velázquez Iturbide

**Patrones de Programación para el
Lenguaje Basado en Bloques
ScratchJr**

Número 2024-02

**Serie de Informes Técnicos DLSI1-URJC
ISSN 1988-8074
Departamento de Informática y Estadística
Universidad Rey Juan Carlos**

Índice

1	Introducción	1
	Agradecimientos.	2
	Referencias.....	2
	Apéndice: Catálogo de Patrones de Programación para ScratchJr	3

Patrones de Programación para el Lenguaje Basado en Bloques ScratchJr

J. Ángel Velázquez Iturbide

Departamento de Informática y Estadística, Universidad Rey Juan Carlos,
C/ Tulipán s/n, 28933, Móstoles, Madrid
angel.velazquez@urjc.es

Resumen. Se presenta una colección de patrones de programación para el lenguaje basado en bloques ScratchJr. La colección se desarrolló tras realizar un análisis cualitativo de programas publicados por los autores del lenguaje ScratchJr. Posteriormente se evaluó con profesores matriculados en un máster de educación y con profesores del lenguaje en el Grado de Educación Infantil. La colección de patrones de programación está dirigida tanto a profesores de Educación Infantil como a sus formadores.

Palabras clave: Programación basada en bloques, ScratchJr, Educación Infantil, patrones de programación, formación de profesores.

1 Introducción

ScratchJr es un lenguaje de programación basado en bloques concebido para niños de 5 a 7 años [1]. Pueden encontrarse descripciones informales de la dinámica del lenguaje, normalmente mediante ejemplos [2][3][4][5], así como una descripción estructurada en un informe técnico [6]. Aunque en principio basta con un conocimiento del lenguaje de programación para el desarrollo de programas, el conocimiento de formas comunes de código facilita dicho desarrollo. En concreto, los patrones de programación resultan útiles para comprender el uso de elementos de un lenguaje de programación. La colección de patrones de programación está dirigida tanto a profesores de Educación Infantil como a sus formadores.

En este informe describimos una colección de patrones de programación para el lenguaje de programación ScratchJr. Se han obtenido como resultado de un análisis cualitativo de programas ScratchJr publicados por los propios creadores del lenguaje [7]. Posteriormente, se han evaluado tanto por parte de alumnos de un máster de formación del profesorado como de profesores del lenguaje en el Grado de Educación Infantil. La evaluación se encuentra descrita en detalle en otro informe técnico [8].

La colección de patrones de programación se encuentra en el Apéndice. Se estructuran en tres conjuntos: efectos visuales, patrones de movimiento, y patrones de control.

Agradecimientos. Este trabajo se ha financiado con los proyectos de investigación PROGRAMA de la Universidad Rey Juan Carlos (ref. M3035) y del Ministerio de Ciencia e Innovación (ref. PID2022-137849OB-I00).

Referencias

1. Flannery, L.P., Kazakoff, E.R., Bontá, P., Silverman, B., Bers, M.U., Sullivan, A.: Designing ScratchJr: Support for early childhood learning through computer programming. En: Proceedings of the 12th International Conference on Interaction Design and Children (IDC'13). ACM DL, 2013, págs. 1-10, DOI [10.1145/2485760.2485785](https://doi.org/10.1145/2485760.2485785)
2. ScratchJr – Aprende – Guía de bloques. <https://www.scratchjr.org/pdfs/block-descriptions.pdf>
3. ScratchJr – Enseña – Actividades. <https://www.scratchjr.org/teach/activities>
4. Bers, M.U., Resnick, M.: The Official ScratchJr Book: Help Your Kids Learn to Code. No Start Press (2016)
5. Bers, M.U., Sullivan, A.: ScratchJr Coding Cards: Creative Coding Activities. No Start
6. Velázquez Iturbide, J.Á.: Reglas de comportamiento del lenguaje basado en bloques ScratchJr. En: Serie de Informes Técnicos DLSI1-URJC, no. 2024-01, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2021.
7. Velázquez Iturbide, J.Á.: Identificación de patrones elementales para ScratchJr. En: Serie de Informes Técnicos DLSI1-URJC, no. 2023-02, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2023.
8. Velázquez Iturbide, J.Á.: Evaluación de patrones de programación y reglas de comportamiento de ScratchJr. En: Serie de Informes Técnicos DLSI1-URJC, no. 2024-03, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2021.

Apéndice: Catálogo de Patrones de Programación para ScratchJr

Los patrones de programación son patrones de código que resultan útiles para comprender el uso de elementos de un lenguaje de programación y el desarrollo de programas, garantizando un uso seguro de dichos elementos y un buen estilo de programación. Por ejemplo, un patrón de programación para un lenguaje imperativo textual puede presentar el recorrido de un vector, que permite realizar diversos tratamientos (p.ej., calcular la suma de sus elementos o cualquier otro cálculo).

Los siguientes patrones pretenden ayudar a los profesores de Educación Infantil o Primaria a desarrollar programas estructurados y que tengan el comportamiento esperado. Los bloques fijos de cada patrón se presentan con su forma concreta, aunque quizá sin incluir sus elementos variables (p.ej. el número de pasos de un bloque Mover). Los bloques variables se presentan en una caja de texto.

Efectos visuales

Con frecuencia se necesita conseguir algún efecto visual, normalmente ligado a un movimiento o acción de un personaje. El bloque disparador varía según cuando queramos que se inicie el movimiento.

- **Bamboleo.** Se produce un giro a izquierda y a derecha.

Ejemplos:

- Vuelo (continuo) de un murciélago.



- Oscilación de un melocotón maduro en un árbol, seguida de su caída.



- **Resaltado.** Consiste en aumentos y disminuciones continuos y modestos de tamaño.

Ejemplos:

- Movimiento sinuoso de una serpiente.



- Centelleo (continuo) de una estrella.



- **Inclinación.** Sirve para mostrar el acto de que un alimento o bebida se gire para que un personaje se lo tome. Después puede volver a su posición original, quizá disminuido de tamaño, o desaparecer.

Ejemplos:

- Comer un melocotón. Se inclina para morderlo y después desaparece.



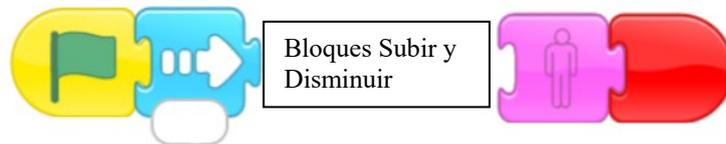
- Beber de un vaso. Se inclina, manteniéndose en esa posición mientras se bebe, volviendo después a su posición vertical.



Patrones de Movimiento

Los patrones de movimiento proporcionan esquemas para conseguir comportamientos visuales o sonoros de los personajes.

- **Patrón “Entrar en algún sitio”.** El personaje se mueve, normalmente en horizontal, y entra o se esconde en algún sitio, desapareciendo. Para darle mayor realismo, puede moverse hacia el lugar, normalmente hacia arriba, e incluso puede disminuir de tamaño antes de desaparecer. Los pasos a la derecha y arriba dependerán de cada situación y el tamaño se reducirá al mínimo antes de hacerle desaparecer:



Ejemplo:

- Va andando por la acera y entra en una tienda:



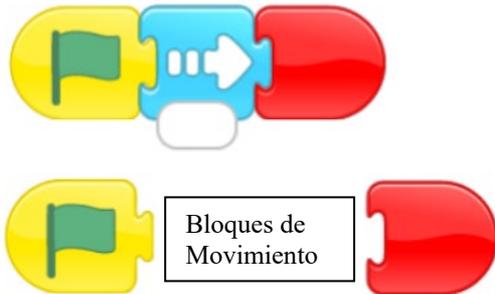
Variantes. Aunque menos frecuente, pueden producirse comportamientos inversos al anterior, como bajar o aparecer (en lugar de subir o desaparecer, respectivamente). Deben cambiarse los bloques de movimiento o apariencia, o su orden relativo, para obtener el comportamiento deseado.

Ejemplo:

- Se pone el Sol:



- **Patrón “Movimiento compuesto”**. El personaje se mueve como resultado de realizar en paralelo los movimientos más sencillos de dos guiones. Un guion realiza un movimiento continuo, normalmente en horizontal, mientras que el otro guion realiza otro movimiento distinto. El patrón consta de dos guiones correspondientes a un mismo personaje:



Ejemplos:

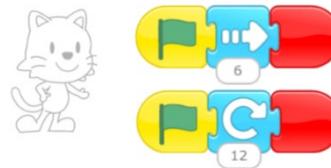
- Tira un balón de baloncesto a canasta:



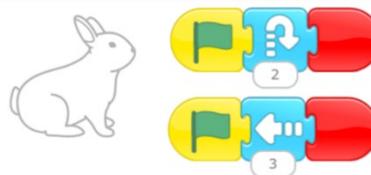
- Eleva un balón de fútbol, que después sigue rondando:



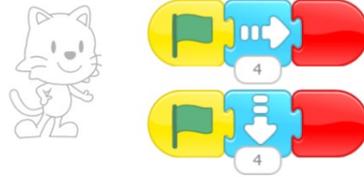
- Flota en el espacio:



- Un conejo avanza saltando:



- Avance en diagonal, normalmente para acercarse o alejarse:



Variante: Patrón “Efecto multimedia compuesto”. El patrón puede generalizarse para que, en lugar de realizar únicamente un movimiento compuesto, puedan combinarse bloques de movimiento, apariencia o sonido, obteniendo otros efectos multimedia.

Ejemplo:

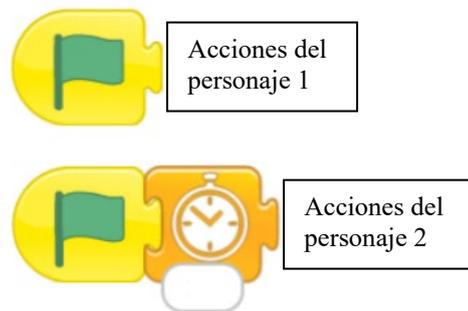
- Efecto coordinado de aparecer/desaparecer junto con un sonido grabado. Hay que llamar la atención sobre la dificultad de que las operaciones de sonido se realicen al mismo tiempo que otras, que exigen que tarden exactamente el mismo tiempo:



Patrones de control

Los patrones de control proporcionan esquemas de control del flujo de ejecución del programa.

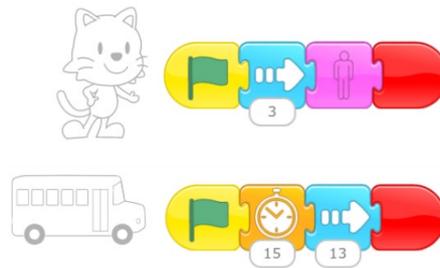
- **Patrón “Sincronización de dos personajes mediante el reloj”.** Dos personajes cuya ejecución comienza al mismo tiempo pueden sincronizarse mediante el reloj si uno de ellos espera un cierto tiempo. De esta forma, da tiempo a que el otro personaje realice una acción previa. El patrón consta de dos guiones correspondientes a dos personajes:



Es un patrón válido, pero en general es desaconsejable este tipo de sincronización, siendo preferible el envío de mensajes.

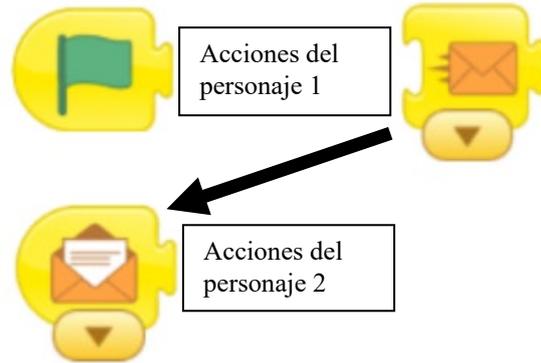
Ejemplo:

- El autobús espera a que Kitty se acerque y suba, tras lo cual el autobús parte.



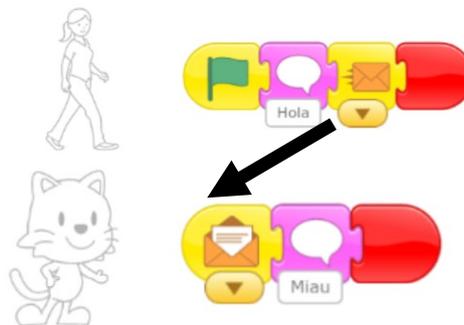
Variante: “Sincronización múltiple mediante el reloj”. El patrón puede generalizarse de forma que los personajes sincronicen más acciones mediante la inclusión de más bloques “Esperar” que garanticen el tiempo de comienzo de cada acción. Por ejemplo, el primer personaje puede ejecutar un bloque “Esperar” tras ejecutar sus instrucciones, de forma que dé tiempo a que el segundo personaje realice las suyas. Incluso la planificación temporal puede implicar a más personajes, pero el uso de medidas absolutas de tiempo la hace insegura.

- Patrón “Sincronización de dos personajes mediante un mensaje”. Un personaje realiza unas acciones, tras lo cual envía un mensaje a un segundo personaje, indicándole que realice sus acciones. El patrón consta de dos guiones correspondientes a dos personajes:



Hemos resaltado con una flecha el sentido temporal del envío de mensaje.
Ejemplo:

- Una mujer habla al gato, que responde:



El patrón puede generalizarse de varias formas.

Variante 1: Patrón “Sincronización en cadena de varios personajes”. Se sincronizan varios personajes en cascada, es decir, de forma que realizan sus acciones en orden secuencial. Basta repetir el patrón anterior para cada cesión de turno, usando un mensaje de un color distinto para cada cesión de turno.

Ejemplo:

- Sea un profesor, que saluda a tres niños, que contestan uno detrás de otro. En el programa siguiente, los niños contestan tal y como aparecen de arriba abajo. La conversación transcurre así (los nombres son inventados, para concretar):

Maestro: “Hola”

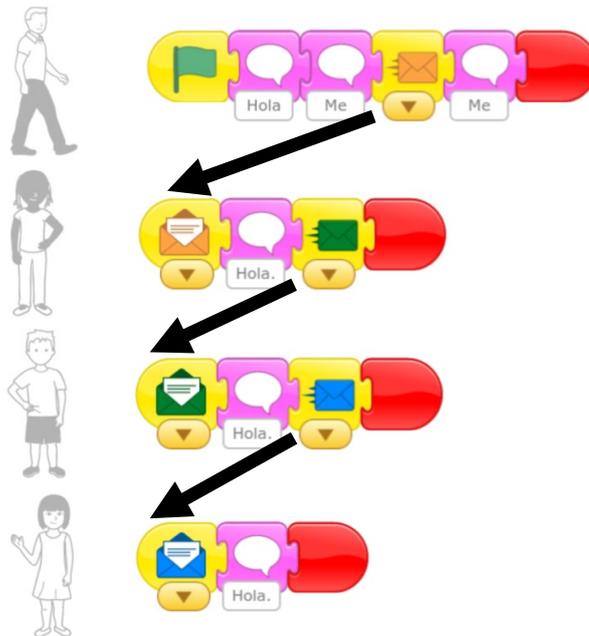
Maestro: “Me llamo Antonio. ¿Y vosotros?”

Niña negra: “Hola. Me llamo Pilar”

Niño: “Hola. Me llamo Jaime”

Niña blanca: “Hola. Me llamo Juana”

Maestro: “Me alegro de conocerlos”



Variante 2: Patrón “Sincronización simultánea de varios personajes”. En este caso, el personaje emisor envía el mismo mensaje a todos los receptores, que realizan sus acciones simultáneamente.

Ejemplo:

- Sea el mismo ejemplo anterior, pero con los tres niños presentándose simultáneamente. La conversación transcurre así (los nombres son inventados, para concretar):

Maestro: “Hola”

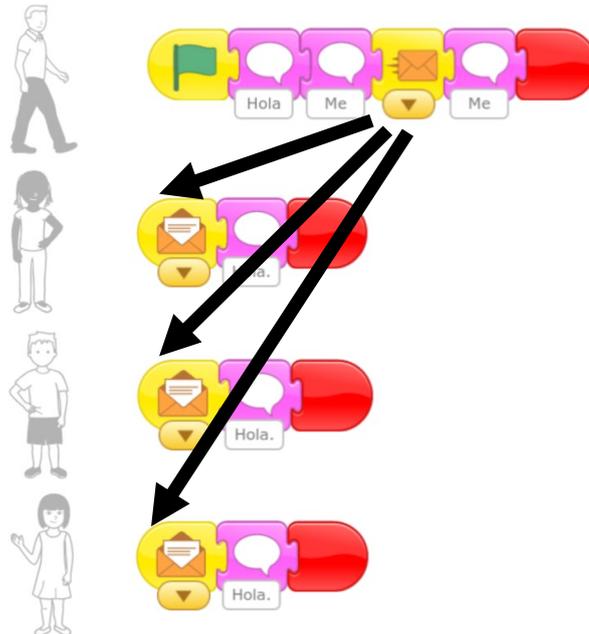
Maestro: “Me llamo Antonio. ¿Y vosotros?”

“Hola. Me llamo Pilar”

En paralelo: “Hola. Me llamo Jaime”

“Hola. Me llamo Juana”

Maestro: “Me alegro de conocerlos”

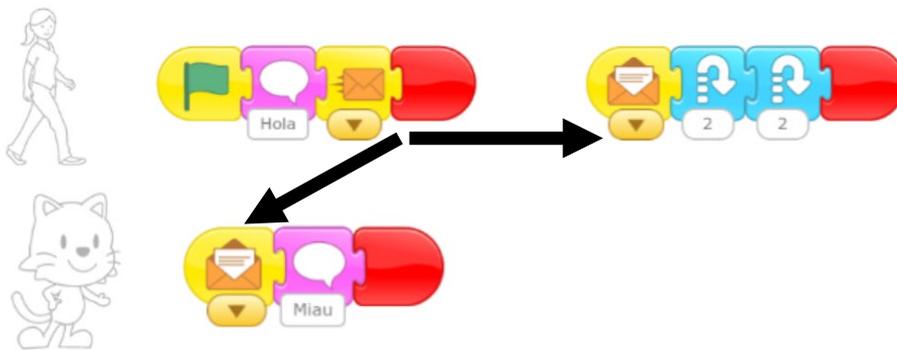


Variante 3: Patrón “Sincronización sin que el personaje emisor espere”.

En todos los casos anteriores, el personaje emisor envía un mensaje y se queda esperando a que todos los personajes terminen de realizar las acciones activadas. En caso de que se quiera que el personaje emisor no se quede parado, deberá enviarse a sí mismo el mismo mensaje que envía al primer personaje, debiendo tener un guion que reciba este mensaje.

Ejemplo:

- Sea el primer ejemplo, de una mujer que habla al gato. El siguiente programa permite que, tras saludar al gato y mientras éste contesta, ella dé un par de saltos:



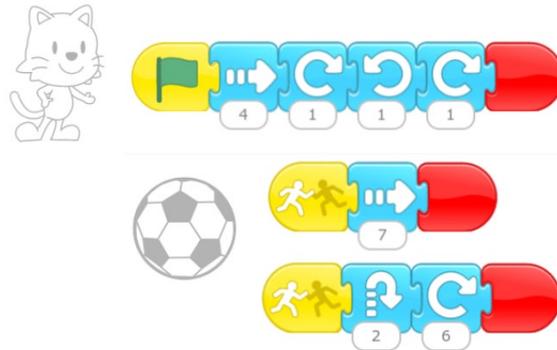
- **Patrón “Sincronización de dos personajes mediante choque”.** Un personaje se mueve y choca con un segundo personaje. Éste detecta el choque y se mueve para que no vuelva a producirse el evento de choque. El patrón consta de dos guiones correspondientes a dos personajes:



Las acciones del primer personaje deben contener bloques de movimiento para que el choque se produzca. Asimismo, al menos un personaje debe moverse para separarse y que no vuelva a lanzares un evento de choque.

Ejemplos:

- El gato avanza hasta tocar el balón, que resulta lanzado y termina rodando.



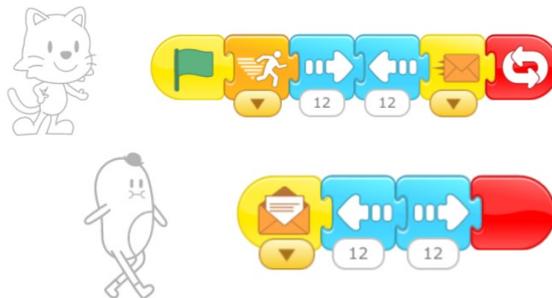
En un programa pueden usarse varios patrones, combinándolos como resulte conveniente. En este ejemplo, se ha combinado el patrón “Sincronización de dos personajes mediante choque” con el patrón “Movimiento compuesto”, que permite producir el lanzamiento y rodamiento del balón.

- Kitten y Tic juegan a que Kitten se acerca a Tic y le toca, retrocede y Tic se acerca a él, tocándole, repitiéndose indefinidamente.



En este ejemplo, se repite indefinidamente la acción mutua de movimiento-choque-movimiento. Para ello, los dos personajes se sincronizan mediante choques y esperas de tiempo. Este ejemplo permite mostrar los riesgos de usar el reloj para la sincronización de personajes. Los tiempos de espera admiten diversos valores, pero si el tiempo de espera de Kitty se reduce a 30 o menos, la repetición de los movimientos deja de estar coordinada.

Una alternativa consiste en usar mensajes, a costa de que los dos personajes no anden simultáneamente:



- **Patrón “Parar a un personaje”.** A veces, se desea que un personaje termine su ejecución. Una forma sencilla de avisar de ello es tocando al personaje. Se consigue el efecto si existe un guion que detecta el evento de tocar, que ejecuta un bloque “parar”. En el patrón, presentamos un solo guion con acciones, pero el personaje puede tener varios guiones.



Ejemplo:

- El gato está avanzando continuamente pero, tras ser tocado por el usuario, se para.



Variante: “Cambiar el comportamiento de un personaje”. En lugar de parar un personaje, puede cambiarse el comportamiento del personaje si, tras el bloque “Parar”, se incluyen otras acciones.

Ejemplo:

- El gato está avanzando continuamente pero, tras ser tocado por el usuario, pasa a saltar continuamente.



- **Patrón “Parar a varios personajes”.** Se trata de un **patrón compuesto** por dos patrones de programación, “Parar a un personaje” y “Sincronización simultánea de varios personajes”.

La acción de parar puede extenderse a varios personajes si se toca a uno de ellos, el cual para sus guiones y envía un mensaje a los demás personajes. Estos personajes tendrán un guion con el bloque parar. La acción queda más limpia si el toque se realiza sobre un personaje extra, cuya única función sea centralizar la parada de los demás.

Ejemplo:

- El gato y la cebra están moviéndose. Al ser tocada la estrella, envía un mensaje al gato y a la cebra para que se paren y hablen brevemente.

