

J. Ángel Velázquez Iturbide

Directrices de Documentación de Programas ScratchJr

Número 2025-03

**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: Directrices de Documentación.....	2

Directrices de Documentación de Programas 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. El entorno de programación del lenguaje ScratchJr no permite almacenar y cargar ficheros. Esto plantea un problema para evaluar en una asignatura o curso los programas desarrollados por profesores en formación. La alternativa que presentamos consiste en elaborar un documento que explica el programa desarrollado de forma estructurada pero comprensible para profesores en formación, generalmente sin un perfil técnico. Las directrices sugieren que se siga la estructura del programa en páginas, personajes e interacciones entre éstos. Para cada página, personaje o interacción, se presenta una captura de pantalla (y quizá alguna información adicional) y una explicación breve de su función. Las directrices se ilustran con la documentación de cinco programas de ejemplo.

Palabras clave: Programación basada en bloques, ScratchJr, Educación Infantil, formación de profesores, documentación de programas.

1 Introducción

ScratchJr es un lenguaje de programación basado en bloques concebido para niños de 5 a 7 años [1]. Su entorno de programación es muy sencillo e intuitivo, pero no permite almacenar y cargar ficheros. Esto plantea un problema para evaluar en una asignatura o curso los programas desarrollados por profesores en formación.

En este informe presentamos una alternativa consistente en que los profesores, en lugar de enviar su programa, elaboren un documento que explique su programa desarrollado. Las directrices permiten explicar un programa de forma estructurada. Está inspirado en el modelo de Bloques de Schulte [2], pero adaptado a ScratchJr y simplificado para profesores en formación, generalmente sin un perfil técnico. Las directrices sugieren que se siga la estructura del programa en páginas, personajes e interacciones entre éstos.

Este informe se inscribe dentro de un esfuerzo para mejorar la formación del profesorado de Educación Infantil en programación mediante ScratchJr. Otras aportaciones didácticas al aprendizaje de ScratchJr han sido ejercicios de programación específicos para ScratchJr [3], la descripción de la semántica dinámica de ScratchJr [4] y patrones de programación en ScratchJr [5]. Todas estas aportaciones didácticas se han utilizado en el Máster Oficial en Competencia Digital Docente y Pensamiento Computacional [6][7].

Las directrices de documentación se encuentran en el Apéndice, ilustradas con cinco ejemplos.

Agradecimientos. Este trabajo se ha financiado con el Proyecto PROGRAMA (ref. PID2022-137849OB-I00), financiado por MICIU/AEI/10.13039/501100011033 y por FEDER, UE.

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, <https://doi.org/10.1145/2485760.2485785>
2. Schulte, C.: Block Model: An educational model of program comprehension as a tool for a scholarly approach to teaching. En: Proceedings of the Fourth International Workshop on Computing Education Research (ICER'08). ACM DL, 2008, págs. 149-160, <https://doi.org/10.1145/1404520.1404535>
3. Velázquez Iturbide, J.Á.: Designing exercises for block-based languages: The case of ScratchJr. En: Proceedings TEEM 2022: Tenth International Conference on Technological Ecosystems for Enhancing Multiculturality, García-Peñalvo, F. J., y García-Holgado, A. (eds.), Lecture Notes in Educational Technology, Springer, Singapore (2023) 50-59, https://doi.org/10.1007/978-981-99-0942-1_5
4. Velázquez Iturbide, J.Á.: Reglas de comportamiento del lenguaje basado en bloques ScratchJr. En: Serie de Informes Técnicos DLSII-URJC, no. 2024-01, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2021, <https://hdl.handle.net/10115/30710>
5. Velázquez Iturbide, J.Á.: Patrones de programación para el lenguaje basado en bloques ScratchJr. En: Serie de Informes Técnicos DLSII-URJC, no. 2024-02, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2021, <https://hdl.handle.net/10115/30711>
6. Paredes-Velasco, M., Velázquez-Iturbide, J. Á.: Una asignatura para la formación del profesorado en programación mediante lenguajes basados en bloques. *Actas de las JENUI*, 7 (julio 2022) 337-344, https://aenui.org/actas/pdf/JENUI_2022.pdf
7. Paredes-Velasco, M., Velázquez-Iturbide, J. Á.: Mejora de una asignatura para la formación del profesorado en programación basada en bloques. *Actas de las JENUI*, 8 (julio 2023) 269-276, https://aenui.org/actas/pdf/JENUI_2023_034.pdf

Apéndice: Directrices de Documentación

1. Documentación de programas

La actividad de programar no conlleva solamente el desarrollo de un programa en un lenguaje de programación concreto, es decir, lo que los informáticos denominan **codificación**. Como cualquier otra actividad de diseño o de ingeniería, conlleva ciertas tareas, tan importantes o más que la propia codificación. Por ejemplo, el desarrollo de un programa ScratchJr exigirá realizar también las siguientes tareas:

- **Diseño.** Invención de una idea general o esbozo del programa a desarrollar y posterior concreción de los detalles de dicha idea.
- **Pruebas.** Ejecución del programa tantas veces como sea necesario para comprobar que el programa funciona correctamente, es decir, que hace lo que se espera.
- **Depuración.** En caso de que el programa funcione incorrectamente, análisis del comportamiento del programa y arreglo de las partes que fallan.
- **Documentación.** Explicación del diseño y del programa desarrollados, de forma que el propio programador u otro puedan comprender el programa en una lectura futura.

En la asignatura se pide desarrollar y documentar un programa ScratchJr. A continuación, se explican las pautas a seguir en la asignatura para documentar un programa ScratchJr, ilustradas con algunos ejemplos sencillos.

2. Pautas de documentación de programas ScratchJr

En cualquier programa ScratchJr podemos distinguir cinco ámbitos:

1. Programa. Es el todo que hemos desarrollado.
2. Páginas. Un programa puede tener entre 1 y 4 páginas, cada una con un escenario y unos personajes distintos. Aunque un personaje aparezca en varias páginas, deben considerarse copias distintas del mismo personaje, ya que cada una tiene su propio programa.
3. Personajes. Una página puede contener un número discrecional de personajes, incluso cero. Cada personaje tiene su propio programa, que puede consistir en uno o más guiones. Aunque un personaje aparezca varias veces, deben considerarse copias distintas del mismo personaje, ya que cada uno tiene su propio programa.
4. Guiones. Un personaje puede tener un número discrecional de guiones, incluso cero.
5. Bloques. Cada guion consta de uno o más bloques. Son los elementos básicos que permiten crear un programa.

La documentación de un programa ScratchJr se organizará siguiendo dos ideas principales:

- Ámbitos de un programa. La documentación explicará tres de los cinco ámbitos antes mencionados: programa, páginas y personajes. Se comenzará por el ámbito más amplio, el programa, y después se irán explicando las páginas y los personajes de cada página y sus relaciones.
- Aspectos de un programa. Un programa puede explicarse desde varios puntos de vista:

- Estructura. Describe el código del programa.
- Comportamiento. Describe qué sucede durante la ejecución del programa.
- Resumen. Resume qué hace el programa en un vocabulario no técnico.

En general, el comportamiento de los programas ScratchJr es fácil de deducir a partir del propio programa. Por tanto, el punto de vista del comportamiento se omitirá. Si se considera necesario explicar el comportamiento de partes de un programa, pueden añadirse explicaciones para esas partes.

Un programa se documentará mediante una tabla con la siguiente estructura. El número de filas variará, dependiendo de la complejidad del programa:

	Estructura	Resumen
Programa	Número de páginas y llamadas entre ellas.	Resumen de qué hace el programa.
Página 1	Fondo y personajes.	Resumen de qué hace la página.
Relaciones P1 (opcional)	Si se considera conveniente, estado inicial de los personajes. Relaciones existentes entre los personajes y con otras páginas.	Resumen del propósito de las interacciones entre personajes.
Personaje 1	Guiones del personaje y contenido de los bloques “Decir”.	Resumen de qué hace el personaje.
Personaje 2	...	...
...	...	...
Página 2	...	...
...	...	...

Veamos algunas aclaraciones adicionales:

- En caso de que el programa tenga una sola página, se fusionan las filas “Programa” y “Página 1” porque serían redundantes.
- Un resumen debe escribirse en una o muy pocas frases.
- Cada celda puede rellenarse con texto, figuras, diagramas o cualquier otro formato. Algunas celdas de la columna de estructura incluirán capturas de pantalla para que la documentación sea más vívida:
 - Celda de estructura de una página. Declara el nombre e incluye una captura de pantalla (a tamaño reducido) del fondo y de cada personaje. Conviene incluir también una captura de la situación inicial de los personajes en el escenario, en cuyo caso puede omitirse la primera captura de pantalla del fondo, por sencillez.
 - Celda de estructura de un personaje. Contendrá una captura de pantalla de sus guiones. Si hay bloques “Decir” cuyo contenido completo no se ve, también se escribirán las frases completas.
- La fila “Relaciones” de cada página puede omitirse si los personajes se ejecutan de forma independiente y no hay saltos de página. Si los personajes se sincronizan mediante eventos, de nuevo se incluirán capturas de pantalla de los personajes para representar dichas relaciones, como veremos en los ejemplos.

A continuación, presentamos 5 ejemplos de documentación de programas ScratchJr, de más sencillo a más complejo. Los tres primeros son relativamente sencillos, mientras

que el cuarto es un programa complejo. El quinto programa es sencillo, pero su documentación se complica al sincronizarse sus dos personajes mediante esperas de tiempo.

1. Programa con una sola página y dos personajes que se ejecutan de forma independiente.
2. Programa con tres páginas y un solo personaje.
3. Programa con una página y dos personajes que se sincronizan mediante mensajes.
4. Programa con cuatro páginas y cuatro personajes que se sincronizan mediante diversos eventos.
5. Programa con una página y dos personajes que se sincronizan mediante esperas de reloj.

En cada ejemplo se incluyen aclaraciones adicionales sobre las pautas de documentación.

Las documentaciones presentadas deben tomarse como formas concretas de seguir las pautas, pero otras son posibles. Lo importante es que un programa se documente siguiendo sus ámbitos y sus aspectos, de forma que finalmente su descripción esté clara.

Ejemplo 1. Es una animación sencilla, de una página, en la que dos personajes bailan por separado. Su documentación se presenta en la Figura 1.

En este ejemplo, se detalla qué dice el personaje Pingüino (“¡Bailemos!”), ya que el bloque “Decir” de su guion solamente muestra los primeros caracteres de la frase.

Figura 1. Documentación del Ejemplo 1.

	Estructura	Resumen
Programa / página 1	Fondo: Teatro Personajes: Gato Pingüino Estado inicial: 	Dos personajes bailan, de forma independiente, en el escenario de un teatro.
Gato	 	Baila.
Pingüino	  Dice “¡Bailemos!”.	Anima y después arranca a bailar.

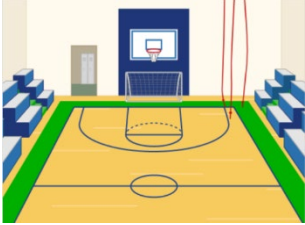
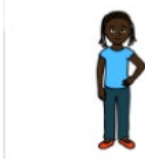
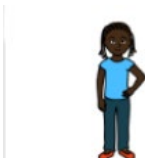
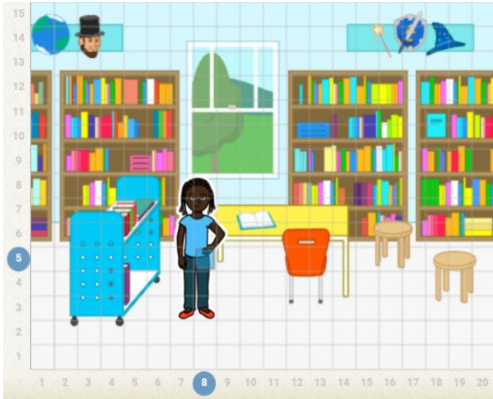
Ejemplo 2. Es una presentación formada por tres páginas, en la que un personaje nos presenta lugares distintos. Su documentación se presenta en la Figura 2.

La celda de estructura del programa declara el número de páginas del programa y su secuencia de ejecución. En general, las páginas de un programa pueden identificarse de forma sencilla con los nombres Página 1 o P1, Página 2 o P2, Página 3 o P3, y Página 4 o P4.

La situación de la niña en la página 2 no es relevante, ya que no anda, pero sí lo es en la página 1 y sobre todo en la página 3. En esta, la niña se acerca a la librería como si fuera a coger un libro, para lo cual se sube y se baja de un taburete. Para conseguir este efecto, debemos ajustar su posición inicial y el número de pasos que da. Para evitar redundancias, la celda “Estructura” de esta página no muestra inicialmente el fondo “Biblioteca” pero lo presenta en el estado inicial del escenario, con la biblioteca de fondo y la niña situada en una posición adecuada. Se ha activado la cuadrícula para que se aprecien mejor las posiciones.

Figura 2. Documentación del Ejemplo 2.

	Estructura	Resumen
Programa	3 páginas: P1 → P2 → P3	Una niña recorre distintas partes de su colegio.
Página 1	- Fondo: Aula - Personajes: Niña  - Estado inicial: 	La niña va a un aula.
Niña		Anda.

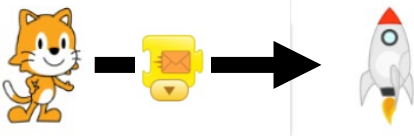
Página 2	– Fondo: Gimnasio  – Personajes: Niña 	La niña va al gimnasio.
Niña	 Dice “¡Hora de hacer gimnasia!”.	Nos habla y se pone a saltar.
Página 3	– Fondo: Biblioteca – Personajes: Niña  – Estado inicial: 	La niña muestra la biblioteca.

Niña	 Dice “Me subo al taburete para llegar a los libros”.	Se acerca a una estantería, sube a un taburete, nos habla y se baja del taburete.
-------------	---	--

Ejemplo 3. Es un programa sencillo, de una sola página, con dos personajes que se sincronizan mediante un mensaje. Su documentación se presenta en la Figura 3.

La sincronización de los personajes se explica en la fila “Relaciones”. La celda “Estructura” muestra los personajes sincronizados y la dirección y el color de los mensajes (el gato envía un mensaje naranja al cohete).

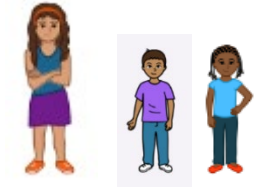
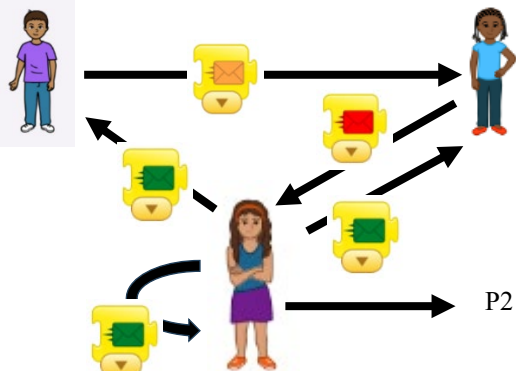
Figura 3. Documentación del Ejemplo 3.

	Estructura	Resumen
Programa / página 1	- Fondo: Luna - Personajes: Gato  Cohete  - Estado inicial: 	El gato cuenta atrás y despegue el cohete.
Relaciones P1		El gato cuenta atrás, tras lo cual despegue el cohete.
Gato	 	Cuenta atrás desde 3 hasta 1.
Cohete	 	Despegue y desaparece.

Ejemplo 4. Es un programa complejo, son cuatro páginas, cuatro personajes y sincronizaciones complejas. Su documentación se presenta en la Figura 4.

Obsérvese que es importante la posición inicial de los personajes en cada página. En las páginas 1, 2 y 4, es importante para conseguir un comportamiento visualmente coherente. Por ejemplo, en las páginas 2 y 4, los personajes deben situarse a la altura de la puerta del autobús para entrar o salir. También es importante que las personas no sean visibles inicialmente. En la página 3, es importante la posición inicial para que los personajes Profesora y Niña choquen.

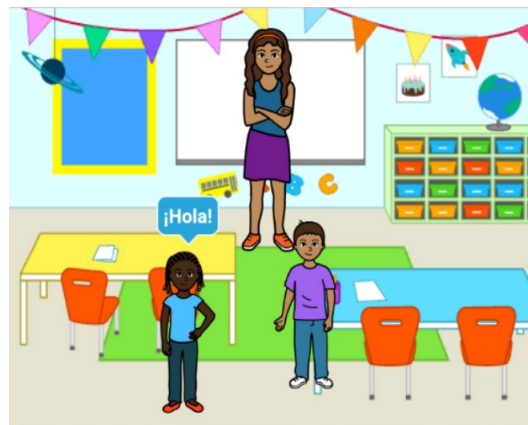
Figura 4. Documentación del Ejemplo 4.

	Estructura	Resumen
Programa	Cuatro páginas: P1 → P2 → P3 → P4	El programa presenta varias escenas de un día de excursión: reunión en el aula, subir al autobús, pasar el día en el campo y bajar del autobús.
Página 1	- Fondo: Aula - Personajes: Profesora Niño Niña  - Estado inicial: 	Escena en el aula, en la que se reúnen la profesora y los alumnos para salir de excursión.
Relaciones P1	Eventos: 	Los personajes se saludan en turnos mediante mensajes. Finalmente, la profesora activa a todos los personajes para que salgan de la clase y produce el salto a la página siguiente.

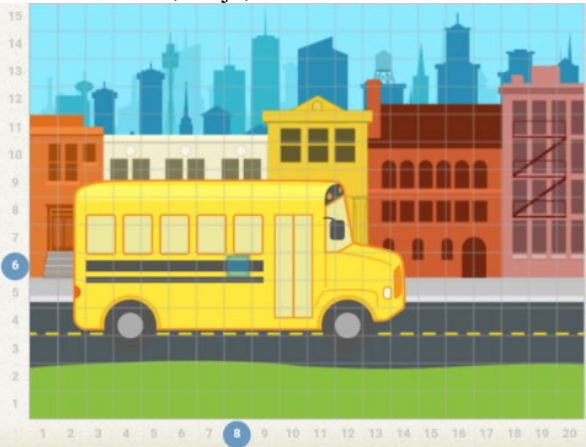
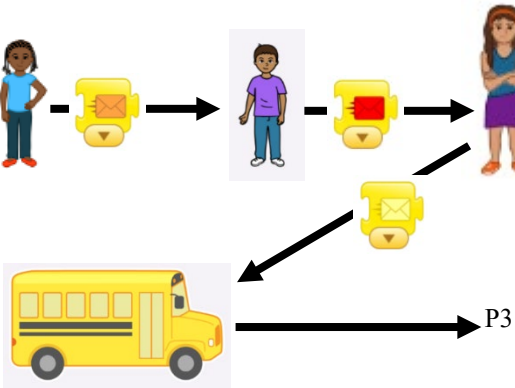
Profesora	 Dice: “¿Todos listos para la excursión?”, “¡Pues vamos!”	Anuncia que ya se van todos.
Niño	 Dice “Buenos días”	Al principio saluda y al final se va.
Niña		Saluda y al final se va.

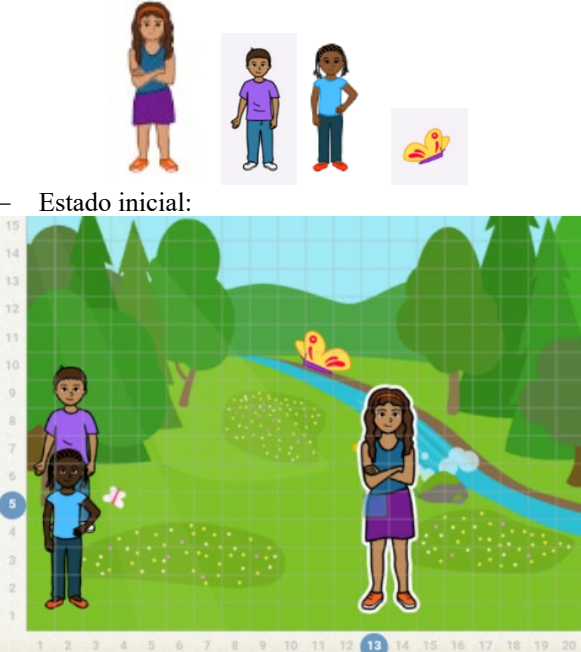
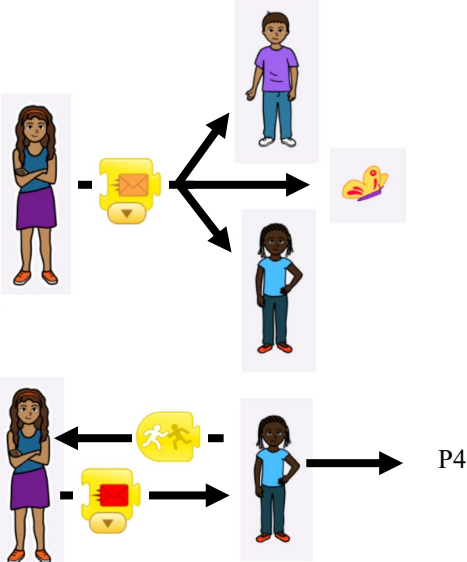
Si se desea, puede mostrarse el comportamiento de los personajes mediante una secuencia de capturas en las que pueden verse las distintas posiciones de los personajes o sus conversaciones. Por ejemplo:

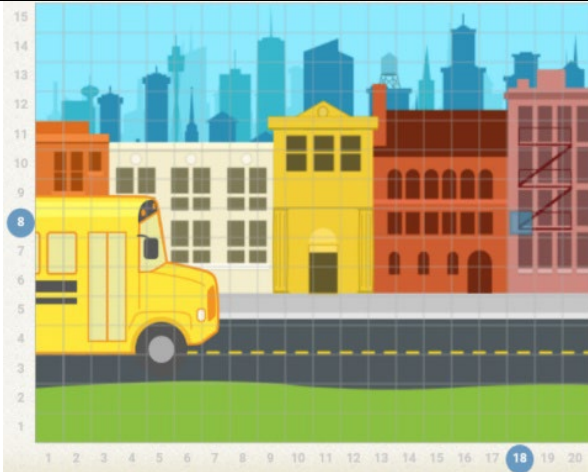
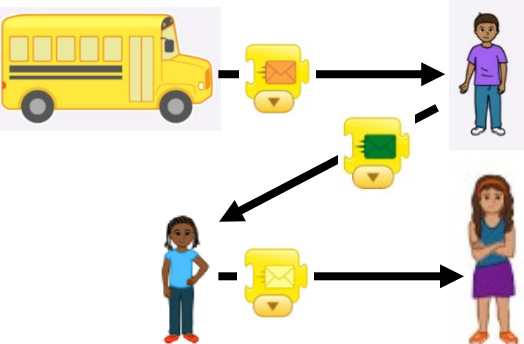
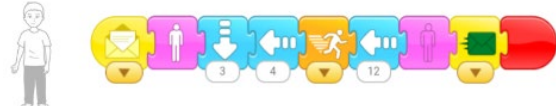




Veamos la documentación de las páginas restantes.

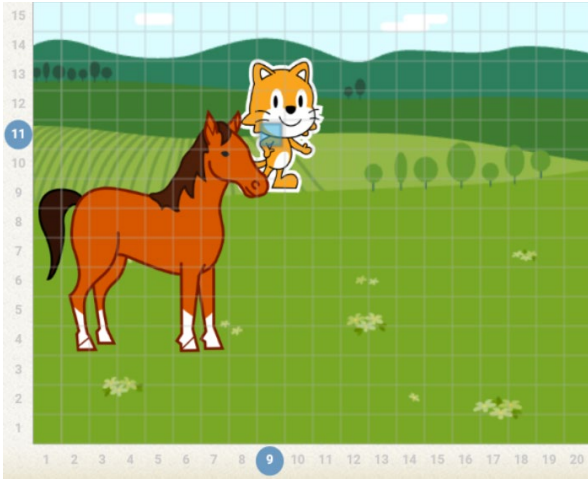
Página 2	– Fondo: Ciudad – Personajes: Profesora Niño Niña Autobús  – Estado inicial: Personas en $x=1$, abajo, ocultas 	Escena en la calle, en la que suben al autobús y salen de viaje.
Relaciones P2		Los personajes se sincronizan mediante mensajes para ir subiendo en orden al autobús, que arranca y produce un salto a la siguiente página.
Profesora	 Dice “Podemos irnos”	Dice que ya pueden salir y sube al autobús.
Niño		Sube al autobús.

Niña		Sube al autobús.
Autobús		Arranca y acelera.
Página 3	– Fondo: Río – Personajes: Profesora Niño Niña Mariposa  – Estado inicial:	Escena en el campo, donde la profesora enseña una mariposa a los niños.
Relaciones P3		Hay dos interacciones entre los personajes: – La profesora llama la atención, mediante un mensaje, de los niños sobre una mariposa. Los niños corren tras la mariposa, que intenta escapar. – La niña tropieza con la profesora, que, al sentirlo, le dice que tenga más cuidado. La niña se disculpa y produce un salto de página.

		
Relaciones P4		Los personajes se sincronizan mediante mensajes de forma que primero llega el autobús y después se bajan todos del autobús en orden.
Profesora	 Dice “¡Hasta mañana!”	Se baja del autobús y se despide de los niños.
Niño		Se baja del autobús y se va con prisa.
Niña		Se baja del autobús y se va con prisa.
Autobús		Llega a la puerta del colegio.

Ejemplo 5. Es un programa con una sola página, con dos personajes que se coordinan mediante esperas de tiempo. Su documentación se presenta en la Figura 5.

Figura 5. Documentación del Ejemplo 5.

	Estructura	Resumen
Programa / página 1	- Fondo: Campo - Personajes: Gato  Caballo  - Estado inicial: 	El gato pide al caballo que le deje montar y éste accede, tras lo cual dan un corto paseo hacia la derecha.
Relaciones		Los personajes se sincronizan mediante esperas de tiempo para hablar y para que el gato monte al caballo y paseen.
Gato	  Dice “¿Puedo montarte?”	Pide al caballo que le deje montar y, tras recibir permiso, lo hace.
Caballo	 	Accede a la petición del gato y se mueve a la derecha para pasear al gato.

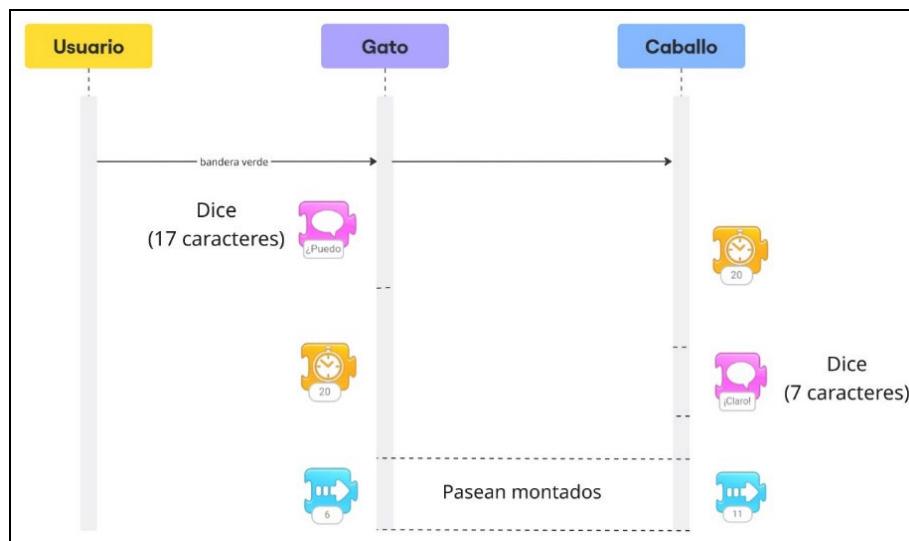
La sincronización del gato y del caballo mediante esperas de tiempo no es difícil, pero debe cuidarse su posición inicial y el uso del tiempo (las esperas de reloj y el tiempo que

dura la ejecución de cada bloque “Decir”, que depende de la longitud del texto), probando el programa hasta que se comporte como se desea.

Por un lado, ambos personajes esperan el mismo tiempo. Lo hacen en momentos sucesivos para aparentar una conversación. Por otro lado, hay que sincronizarles para que el gato quede situado encima del caballo y se muevan juntos, dando la sensación de que está montado. Como el gato está inicialmente situado a la derecha del caballo, da menos pasos. Los pasos que el caballo da de más duran lo mismo que el tiempo que el gato habla más que el caballo (dice una frase más larga).

El siguiente diagrama muestra la ejecución a lo largo del tiempo de los bloques de los dos personajes. En la parte superior, puede observarse que se identifican los tres “actores”: el usuario del programa (que pulsa el icono de bandera verde) y los dos personajes. No haría falta incluir al usuario, pero así se muestran todos los eventos del programa.

El usuario comienza pulsando el icono de bandera verde, evento que activa a ambos personajes. Debajo de cada actor se muestran sus acciones en el tiempo, de arriba a abajo.



El diagrama se ha creado con una herramienta de diseño, pero bastaría con dibujarlo a mano. Cualquier forma de explicación textual o visual es válida. Por ejemplo, una versión manual del diagrama es:

