



# Propuesta didáctica: Un laberinto de polinización



**Descripción:** Propuesta donde los estudiantes diseñan y programan un laberinto para una abeja en Scratch Jr., aprendiendo sobre polinización, programación y trabajo colaborativo de forma creativa y práctica.

**Fecha de creación:** noviembre 2024

**Ciclo:** 2do

**Tramo:** 3

**Grado:** 3ro

**Espacio:** Científico - Matemático. Técnico - tecnológico

**Unidad curricular:** Ciencias del ambiente (Biología). Ciencias de la computación y tecnología educativa.

**Competencia general:** Pensamiento Computacional

**Competencias específicas:***Biología:*

CE5. Recopila y organiza información, almacena datos y emplea diferentes programas para identificar y resolver problemas socioambientales simples vinculados a la salud.

CE5. Explora, de forma colaborativa, problemas computacionales simples, siguiendo secuencias, en situaciones lúdicas y cotidianas, para dar respuestas a interrogantes planteadas.

*Ciencias de la computación y tecnología educativa:*

CE5. Explora, de forma colaborativa, problemas computacionales simples, siguiendo secuencias, en situaciones lúdicas y cotidianas, para dar respuestas a interrogantes planteadas.

**Contenido:***Biología:*

Reproducción sexuada en vegetales. Polinización.

Introducción a la programación por bloques, características del lenguaje de programación y su relación con otros lenguajes.

*Ciencias de la computación y tecnología educativa:*

Introducción a la programación por bloques, características del lenguaje de programación y su relación con otros lenguajes.

**Criterios de Logro:***Biología:*

Recopila y organiza información sobre las etapas de la reproducción con mediación docente y las socializa a través de diferentes lenguajes.

Utiliza, colaborativamente, la programación en la realización de animaciones y productos lúdicos o curriculares.

*Ciencias de la computación y tecnología educativa:*

Utiliza, colaborativamente, la programación en la realización de animaciones y productos lúdicos o curriculares.

**Metas de aprendizaje:**

(La/s meta/s de aprendizaje se situará/n a la realidad del grupo a cargo del docente)

Los estudiantes:

- Organizarán información sobre la polinización para socializarla a través de un juego.
- Explorarán el lenguaje de programación por bloques para realizar un juego animado.

**Plan de aprendizaje:**

Esta propuesta didáctica tiene como objetivo fortalecer el aprendizaje de los estudiantes sobre el proceso de polinización y el rol fundamental de los polinizadores en los ecosistemas. Antes de llevar a cabo esta actividad en Scratch Jr., se recomienda que hayan introducido previamente el concepto de polinización en el aula. Es ideal que los niños comprendan cómo algunas plantas dependen de insectos, como las abejas, para trasladar el polen de una flor a otra, permitiendo la producción de frutos y semillas.

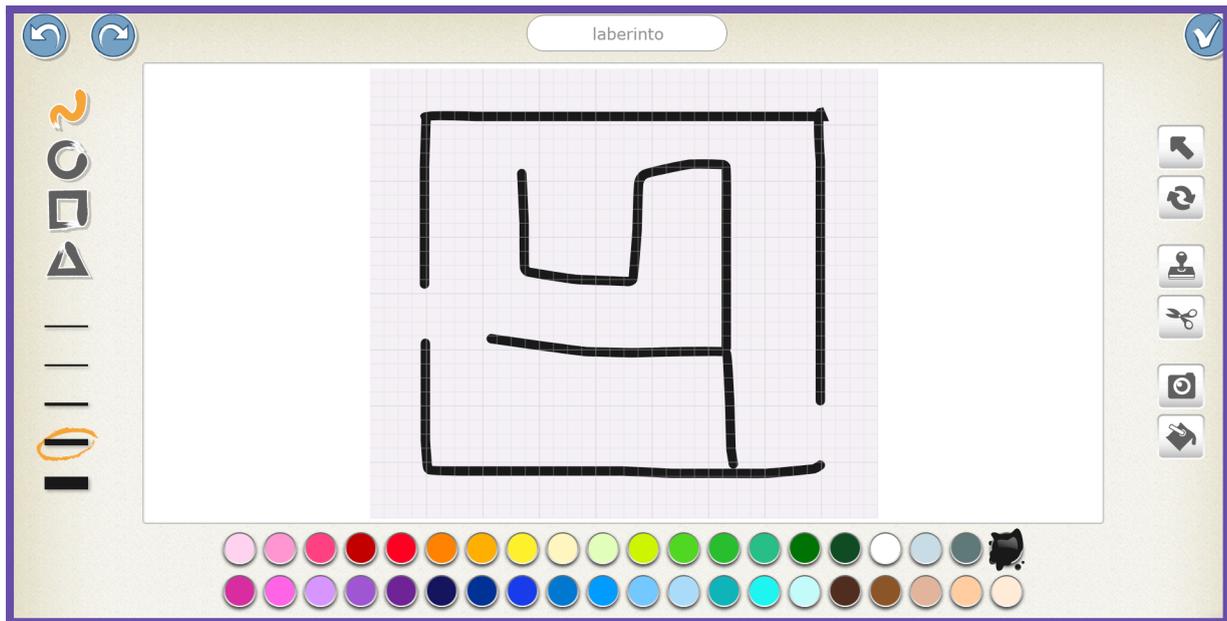
La actividad en Scratch Jr. refuerza este conocimiento de manera práctica y lúdica, mediante un laberinto en el cual deberán ayudar a una abeja a encontrar diferentes flores para cumplir con su misión de polinización. Al tener esta base teórica, los estudiantes podrán participar activamente en el juego, comprendiendo la importancia de la labor de la abeja y las dificultades que enfrenta en su entorno.

**Actividad 1 Diseño**

En esta actividad inicial, los estudiantes trabajarán en equipos para planificar y diseñar los elementos principales de su juego en Scratch Jr. La planificación incluye decidir cómo serán el laberinto, la abeja, las flores y los botones con flechas que usarán para mover a la abeja por el juego.

Es recomendable dividir la clase en grupos pequeños para que cada uno de ellos realice una breve lluvia de ideas para decidir cómo quieren que luzcan cada uno de los elementos: ¿De qué color será el laberinto? ¿Cómo será la abeja (colores, tamaño, características)? ¿Cuántas flores habrá y de qué colores?

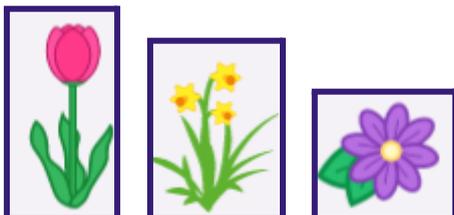
Es conveniente que cada equipo dibuje un boceto de cómo imaginan el laberinto. Pueden decidir si tendrá caminos rectos, curvas o atajos. Además, pueden decidir los colores que usarán para las paredes y el fondo.



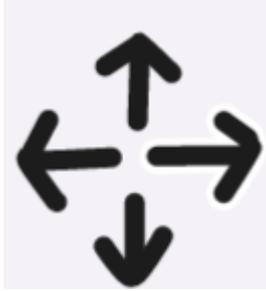
Además, elegirán los colores y el diseño de su abeja. Pueden hacer un dibujo sencillo con franjas amarillas y negras, añadir detalles como alas y antenas.



Los equipos decidirán cuántas flores colocarán en el laberinto y qué colores o formas tendrán. Pueden diseñarlas para que sean de diferentes tamaños y colores, haciendo el juego más visual y atractivo.



Para mover a la abeja, los estudiantes deben crear flechas en cuatro direcciones (arriba, abajo, izquierda y derecha). Pueden decidir el tamaño, color y si le agregan detalles que representen su función (por ejemplo, un cuadrado o círculo que la rodea como si fuera un botón).



Al realizar las actividades de diseño y programación del laberinto, es fundamental fomentar en los estudiantes la idea de representación y abstracción. Al dibujar los personajes y elementos del juego, los niños están trabajando una habilidad poderosa: la capacidad de concentrarse en lo relevante y representar un ser o un objeto de manera efectiva. Esta actividad no solo les ayuda a expresar sus ideas creativamente, sino que también les permite abstraer las características esenciales de lo que están creando, promoviendo así su comprensión conceptual. Invitar a los estudiantes a reflexionar sobre qué elementos son necesarios para representar adecuadamente a una abeja o una flor, por ejemplo, puede enriquecer su proceso de aprendizaje y fortalecer su pensamiento crítico.

## Actividad 2: Programación

En esta actividad, los estudiantes programarán los elementos de su juego en Scratch Jr. utilizando el método de ensayo y error para descubrir cómo mover la abeja a través del laberinto y lograr los objetivos establecidos.

Es conveniente que el docente explique brevemente los objetivos de la programación, recordando que:

Al hacer clic en cada flecha, la abeja deberá moverse.

Si la abeja toca los bordes del laberinto, debe regresar a su punto de inicio.

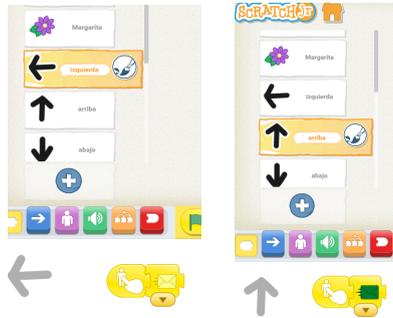
Al tocar las flores, se debería emitir un mensaje visual, o auditivo

El juego termina cuando la abeja toca un botón final (círculo), que cambia el fondo para mostrar un mensaje de “¡GANASTE!”.

- Para programar las flechas de movimiento:

Cada equipo programa las flechas de movimiento. La flecha "arriba" envía un mensaje de color específico a la abeja para que suba, la flecha "abajo" la hace bajar, y así sucesivamente para las direcciones izquierda y derecha.

Los estudiantes experimentan enviando mensajes de prueba para verificar que la abeja responda correctamente.



- Programación del laberinto

Si la abeja lo toca, envía otro mensaje.



- Programación de la Abeja:

La abeja recibe mensajes de las flechas y se mueve en la dirección correspondiente.



Los estudiantes deben programar a la abeja para que vuelva al punto de origen si recibe el mensaje del laberinto.



- Programación de las Flores:

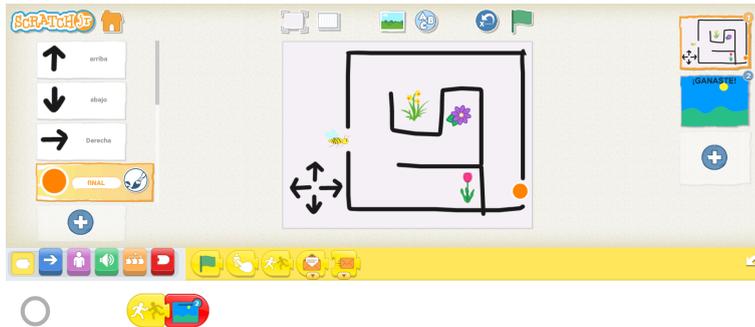
Cada vez que la abeja toca una flor, los estudiantes programan un mensaje que muestra algo o se escuche algo. Pueden experimentar con diferentes opciones de retroalimentación visual o auditiva en Scratch Jr.



- Programación del Botón Final:

Se programa un botón final, que podría ser un círculo al final del laberinto.

Cuando la abeja toca este botón, el fondo cambia a una pantalla que muestra “¡GANASTE!”.



Cada vez que un equipo logra un objetivo (por ejemplo, movimiento de la abeja o interacción con las flores), muestra su logro a los demás, compartiendo cómo lo hicieron.

El docente facilita estas socializaciones, motivándolos con preguntas como: “¿Qué tuvieron que cambiar para que funcione?”, “¿Qué creen que podrían mejorar?”, o “¿Qué harían diferente si lo intentan otra vez?”.

Apoya a los equipos a través de preguntas o consejos si los estudiantes lo solicitan. Sin embargo, se promueve que los estudiantes exploren y resuelvan problemas de manera autónoma antes de pedir ayuda.

### Actividad 3: Depuración

En esta actividad, revisarán y mejorarán su juego para asegurarse de que todos los elementos funcionen correctamente. La depuración es una etapa esencial en la programación y ayuda a los estudiantes a desarrollar habilidades para identificar y resolver problemas de forma independiente.

El docente explicará qué es la depuración y su importancia, enfatizando que incluso los mejores programadores revisan su código para corregir errores.

Se anima a los estudiantes a que vean la depuración como una oportunidad de mejora y no como un error, usando frases como: “Buscar formas de hacer que el juego sea aún mejor”.

Los equipos juegan su propio laberinto y observan si algo no funciona como esperaban. Pueden tomar nota de problemas específicos, como:

La abeja no se mueve en la dirección correcta cuando se presionan las flechas.

La abeja no regresa al punto de inicio al tocar los bordes.

Las flores no emiten el mensaje “¡Bien!” cuando son tocadas.

El botón final no cambia el fondo para mostrar “¡GANASTE!”.

La docente puede guiar a los equipos con preguntas como: “¿Qué sucede cuando la abeja toca los bordes?” o “¿Qué pasa al tocar el botón final?”.

Cada equipo intenta modificar su programación para solucionar los problemas encontrados. Es importante fomentar la cooperación y el aprendizaje compartido al animar a los equipos a que den sugerencias entre ellos si encuentran problemas similares.

Los estudiantes realizan pruebas para ver si sus cambios resolvieron el problema y ajustan según sea necesario.

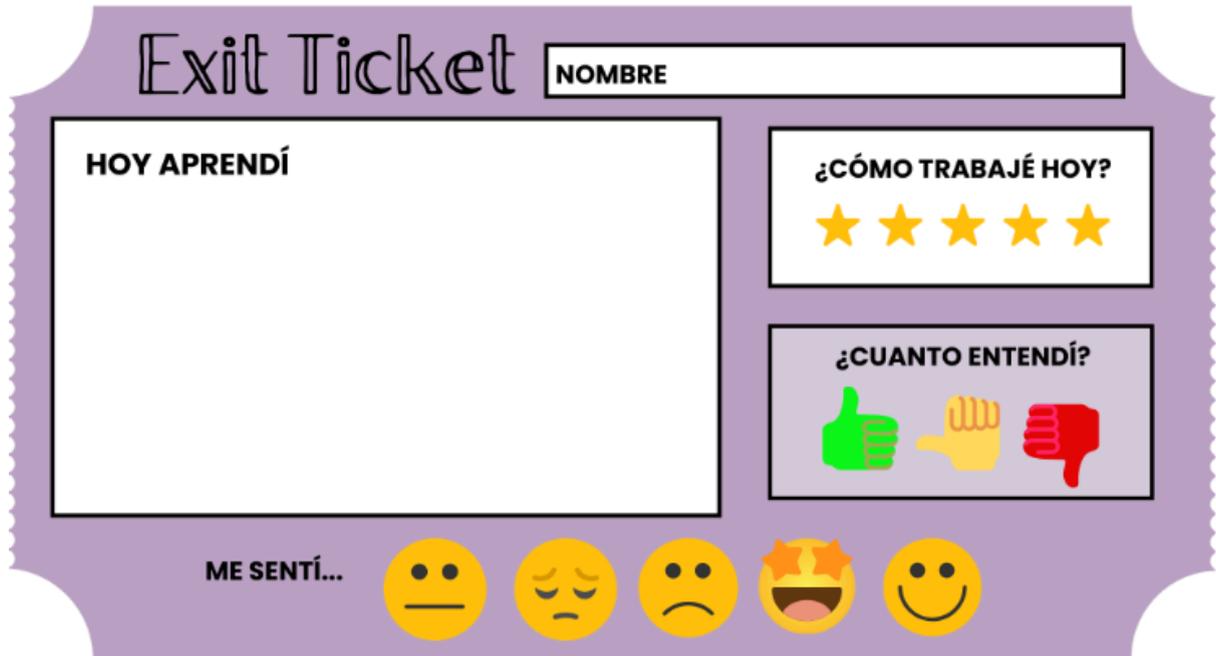


[Video de ayuda para docentes](#)

#### **Actividad 4: Metacognición**

En esta actividad final, los estudiantes reflexionarán sobre su proceso de aprendizaje a lo largo de las actividades de diseño, programación y depuración del juego de laberinto. Esta reflexión los ayudará a desarrollar habilidades metacognitivas, permitiéndoles analizar sus fortalezas, áreas de mejora y las estrategias que utilizaron para superar desafíos.

Se propone realizarla a través de la rutina [Exit ticket](#)



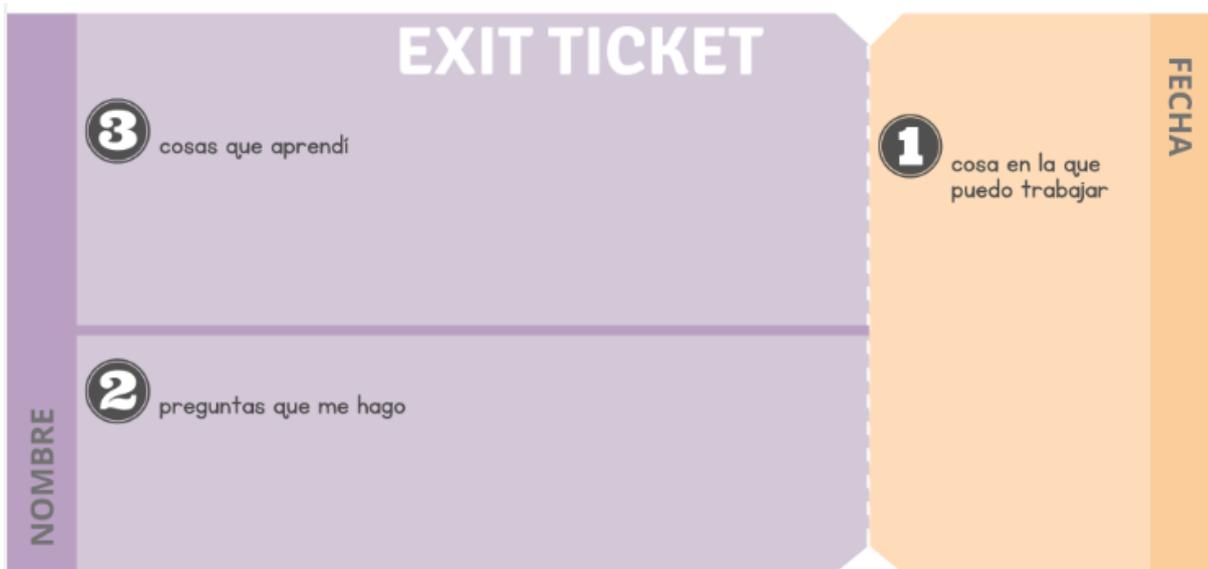
**Exit Ticket** NOMBRE \_\_\_\_\_

**HOY APRENDÍ**

**¿CÓMO TRABAJÉ HOY?**  
★ ★ ★ ★ ★

**¿CUANTO ENTENDÍ?**  
👍 👎 👏

**ME SENTÍ...** 😐 😞 😓 🤩 😊



**EXIT TICKET**

**3** cosas que aprendí

**2** preguntas que me hago

**1** cosa en la que puedo trabajar

NOMBRE \_\_\_\_\_ FECHA \_\_\_\_\_

### Sugerencias didácticas y de evaluación:

### Sugerencias para trabajar polinización:

- Los estudiantes pueden construir flores grandes de papel o materiales reciclables y representar el polen con pequeñas bolitas de papel o algodón. Usando "abejas" hechas con limpiapipas o algodón, simulan el transporte de polen de una flor a otra, entendiendo visualmente el papel de los insectos y otros polinizadores.
- Organizar un juego de roles donde algunos sean flores y otros insectos (abejas, mariposas). Los insectos deben recoger "polen" (representado por



polvo de talco o confeti) de una flor y llevarlo a otra. Esto ayuda a los estudiantes a comprender la importancia de los polinizadores en el proceso de reproducción de las plantas.

- Llevarlos al patio o jardín escolar para observar plantas en floración y buscar insectos alrededor de ellas.
- Explicar el ciclo de vida de una planta, desde la semilla hasta la floración, destacando el momento en que ocurre la polinización. Los estudiantes pueden realizar un dibujo o una línea de tiempo con cada etapa, identificando el papel de la polinización en la producción de semillas y frutos.

### ***Sugerencias para trabajar programación:***

- Antes de que los estudiantes empiecen a programar, mostrar un ejemplo básico de cómo la abeja debe moverse en el laberinto. Esto ayuda a que los estudiantes comprendan cómo se espera que funcione el juego, dando una idea clara de las metas.
- Animarlos a probar y ajustar sus programaciones. Explicar que cometer errores es una parte natural del proceso de aprendizaje y que cada error los ayuda a comprender mejor el código y el funcionamiento de su juego.
- Planear momentos para que los equipos compartan sus avances y hallazgos. Esto fomenta la cooperación y permite que se inspiren entre ellos. También puede darles ideas para resolver problemas que podrían estar encontrando.
- Cuando un equipo enfrenta dificultades, en lugar de resolver el problema por ellos, hacer preguntas que los lleven a descubrir la solución por sí mismos. Preguntas como “¿Qué pasa si cambias el color del mensaje?” o “¿Cómo puedes hacer que la abeja vuelva al inicio?” promueven el pensamiento crítico.
- Sugerir que mantengan sus bloques de código organizados y utilicen un color específico para cada dirección de movimiento en los mensajes. Esto les ayudará a evitar confusión y a reconocer patrones en su programación.
- Recordarles que prueben su juego cada vez que hagan un cambio en el código. Esto les permitirá ver el efecto de cada ajuste en tiempo real y facilita el proceso de depuración, detectando errores en el momento.
- Aunque el juego tiene ciertas metas claras, dar libertad para que personalicen el laberinto o añadan detalles (como sonidos o decoraciones adicionales) si ya han terminado la base del proyecto. Esto puede motivarlos y hacer la actividad aún más divertida.
- Enfatizar que la programación es más efectiva cuando se colabora. Recordar que compartan ideas y se dividan las tareas según sus habilidades, como diseño, prueba o depuración.
- Ayudarlos a comprender la importancia del orden en el código. Reflexionar con ellos que el juego solo funcionará correctamente si los pasos están en el orden adecuado, de modo que cada acción tenga un propósito y una secuencia lógica.



Una posible rúbrica de evaluación sería la siguiente:

Criterio	Nivel básico	Nivel intermedio	Nivel Avanzado
<b>Colaboración en el equipo</b>	Los miembros del equipo trabajan de forma individual, con poca interacción entre ellos	Algunos miembros del equipo colaboran, pero la participación no es equitativa.	Todos los miembros del equipo participan activamente y colaboran entre sí.
<b>Comunicación</b>	La comunicación es mínima y poco clara; no se comparten ideas ni se escucha a los demás.	La comunicación es ocasional; se comparten algunas ideas, pero no se escucha a todos los miembros del equipo.	La comunicación es fluida y efectiva; todos los miembros del equipo comparten ideas y se escuchan mutuamente.
<b>Programación y resolución de problemas</b>	El equipo no reflexiona sobre los errores y tiene dificultades para solucionar problemas; no se realizan pruebas efectivas.	El equipo reflexiona sobre algunos errores y trata de solucionar problemas, pero no siempre logra el objetivo.	El equipo reflexiona sobre los errores, realiza pruebas efectivas y soluciona dificultades de manera colaborativa para alcanzar los objetivos de diseño y programación.
<b>Responsabilidad</b>	Los miembros del equipo no asumen responsabilidades y dependen de otros para completar las tareas.	Algunos miembros asumen responsabilidades, pero hay dependencia en otros para completar las tareas.	Cada miembro del equipo asume responsabilidades y contribuye de manera activa y equitativa en el proceso.



**ANEP**

DIRECCIÓN GENERAL  
DE EDUCACIÓN  
INICIAL Y PRIMARIA

DIVISIÓN  
PLANEAMIENTO  
EDUCATIVO

Departamento  
de Tecnologías Educativas  
aplicadas y virtualidad

**Autor: Maestra Contenidista Graciela Oyhenard**

**Licenciamiento:** [Atribución-NoComercial-CompartirIgual 4.0 Internacional \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Imagen de portada: Foto de Pixabay:

<https://www.pexels.com/es-es/foto/abeja-marron-y-negra-sobre-nectar-de-flor-amarilla-460961/>