

Juguemos en la granja Marianito

Grado sugerido: Sexto

Rubén Darío Soler López

Publicado en el Banco Virtual de Recursos de Colombia Programa en el año 2025.



Este material se comparte bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0). Puede copiar y redistribuir el material en cualquier medio o formato, siempre que dé el crédito adecuado al autor, no lo use con fines comerciales, y no remezcle, transforme o cree a partir del material.

Para más información, consulte la licencia completa en [Deed - Attribution-NonCommercial-NoDerivatives 4.0 International - Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Para contactar al autor/a de este recurso, escriba a: rudasoler@gmail.com

RUBENDARIO SOLER LOPEZ- IETA MARIANO MELENDRO

JUGUEMOS EN LAS GRANJA DE MARIANITO

Este documento presenta instrucciones para la programación de un video juego basado en experiencias de la Granja Mariano Melendro

Duración	<p><i>El proyecto tiene una duración de 3 sesiones, cada una dividida en 2 horas desconectadas y 2 horas conectadas.</i></p> <p><i>En este apartado se abordarán las conectadas</i></p>
Objetivo y descripción del proyecto	<p><i>Desarrollar el pensamiento computacional en estudiantes del grado 6 mediante la construcción de un videojuego en Scratch</i></p>
Lista de materiales	<ul style="list-style-type: none">- <i>Computadores con Scratch</i>- <i>Tablero y marcadores</i>- <i>Cuaderno, Lápices, Colores y Lapiceros</i>
Características del problema para tener en cuenta en la solución.	<p><i>Los estudiantes de la institución reciben formación técnica agropecuaria, la cual debe orientarse hacia procesos de tecnificación y sostenibilidad ambiental. No obstante, en la práctica persisten acciones que contradicen estos principios, en parte debido a las exigencias operativas de sostener la granja escolar. Además, en los grados 5º y 6º, el trabajo práctico de campo aún no alcanza un nivel significativo de profundidad. Por ello, se hace necesario fortalecer la formación teórica en torno al manejo de la granja, incorporando enfoques pedagógicos contemporáneos que promuevan prácticas sostenibles y proporcione una base sólida para los grados superiores. En este sentido, el desarrollo de un videojuego temático permite introducir estos contenidos de manera lúdica e innovadora, facilitando la apropiación de conocimientos tecnológicos y agroecológicos desde edades tempranas.</i></p>
Pasos para desarrollar el proyecto	<p>Nivel 1: Portada y animación del juego</p> <ul style="list-style-type: none">• <i>Programar la pantalla de inicio con:</i>○ <i>Título del juego: !DONDE MARIANITO!</i>

- *Botón de Iniciar juego: ! START!*
- *Animación breve con personajes mostrando el contexto del juego*
- *Habilidades: animaciones, eventos, escenarios, mensajes.*

Nivel 2: ¡Rápido! Recolactemos semillas!

- *El personaje debe recolectar semillas de maíz antes de que baje el sol y se oscurezca*
- *Elementos: tiempo límite, colisiones, cambio de disfraces.*
- *Habilidades: control de movimiento, condiciones, variables de conteo.*

Nivel 3: ¡Detengamos los Insectos!

- *Aparecen Insectos Aleatoriamente que avanzan en el espacio*
- *El jugador debe evitar tocarlas mientras no para de trabajar por el bien de la granja buscando recursos y mejorando sus habilidades*
- *Habilidades: lógica de aparición aleatoria, detección de colisiones, efectos visuales.*

¡Ahora tú, crea el siguiente Nivel

Nivel 4: La carrera por los huevos

- *Juego tipo plataforma horizontal. El personaje debe esquivar obstáculos (serpientes) y llegar hasta los huevos a tiempo*
- *Animación de victoria al llegar.*
- *Habilidades: scroll horizontal, detección de obstáculos, niveles de vida o intentos.*

Adaptaciones	Adaptaciones para Población Rural sin Acceso a Internet
	<p>Para el proyecto "Juguemos en la Granja de Marianito", se pueden implementar adaptaciones para poblaciones rurales sin acceso a Internet:</p> <ul style="list-style-type: none"> • Acceso a Materiales de Scratch: Utilizar la versión de escritorio de Scratch, preinstalada en los computadores. Esto permite a los estudiantes trabajar sin internet. Los equipos pueden ser de una sala de informática o prestados para las sesiones. • Recursos Audiorivisuales Offline: Descargar y almacenar videos o animaciones localmente. Proyectarlos o verlos directamente en las pantallas de los equipos sirve de inspiración y contexto del juego. • Guías Impresas y Actividades Desconectadas: Diseñar guías impresas con pasos clave y pseudocódigo para reforzar la comprensión. Las horas desconectadas se usan para bocetar, planificar el juego en papel o simular dinámicas sin computador. • Enfoque en Problemas Locales y Recursos: Adaptar los conceptos del juego para enfatizar la formación agropecuaria. Discutir temas como rotación de cultivos o identificación de plagas. Los ejemplos se basan en experiencias diarias de la granja escolar.

ANEXO № 1
SECUENCIA DIDÁCTICA: Creando la Entrada a la Granja de Marianita

	<i>Indique el o los aprendizajes que busca desarrollar en las/los estudiantes durante la sesión de clase</i>
Aprendizaje(s) esperado(s)	<ul style="list-style-type: none"> • Comprender y aplicar conceptos básicos de programación visual como escenarios, personajes y eventos. • Diseñar y programar una pantalla de inicio con título y botón de inicio. • Crear una animación sencilla que introduzca el contexto del juego

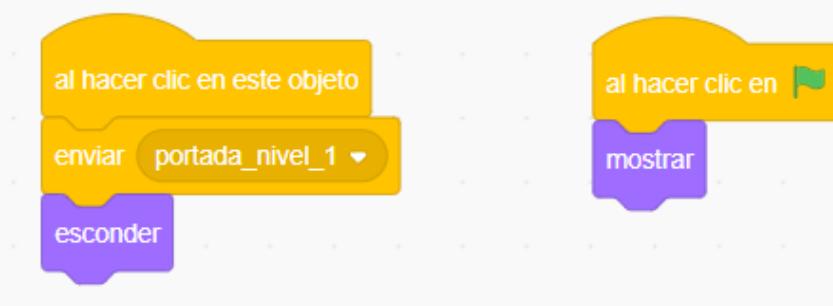
Materiales requeridos	<ul style="list-style-type: none"> Computadores con Scratch (versión de escritorio si no hay Internet). Tablero y marcadores. Cuaderno, lápices, colores y lapiceros.
Conocimientos previos requeridos	<ul style="list-style-type: none"> Familiaridad básica con el entorno de Scratch (interfaz de usuario, bloques de código). Conceptos básicos de dibujo y diseño para bocetar personajes y fondos. Entendimiento de lo que es un videojuego y sus elementos principales (título, botón de inicio, personajes)
<p>Actividad(es) a desarrollar</p> <p><i>Indique las acciones que realizarán el/la docente y sus estudiantes y las indicaciones si el trabajo se debe realizar de forma individual, en parejas o grupal.</i></p>	

Inicio (15 minutos - Desconectado):

- Docente:** Presenta el proyecto "Juguemos en la Granja de Marianita" y el objetivo de desarrollar un videojuego en Scratch. Lo primero que se desea lograr es crear la animación de la portada del juego
- Estudiantes (Individual):** Los estudiantes aportan sus ideas de como seria el posible juego, Lo escriben en el cuaderno
- Desarrollo (75 minutos - Conectado):**
 - Docente:** Indica a los estudiantes los siguientes pasos,
 - Como cargar los fondos y el objeto START a la aplicación Scratch
 - Estudiantes (Individual):** El estudiante carga los fondos Bichos, Fondo_1, Fondo_2 y Fondo_3 y en el espacio código pone los siguientes bloques:



De igual manera al cargar el objeto START le agrega el siguiente código:



Ver el vídeo que se encuentra en el anexo donde se amplia la información:

- **Cierre (30 minutos - Desconectado/Conectado):**
- **Docente:** Se acerca a cada grupo y verifica que se muestre lo indicado, si no corrige el código y explica su funcionalidad
- **Estudiantes (Grupal):**
 - Cada estudiante describe lo que observo en la actividad y explica de que otras formas se podría animar.
 - Se asigna como tarea pensar en la siguiente etapa del juego: la recolección de semillas.

Tiempo estimado: 120 minutos

Adaptaciones

- **Población rural sin internet:** Utilizar la versión de escritorio de Scratch. Los diseños de los estudiantes pueden ser dibujados directamente en cuadernos antes de pasarlos a Scratch. El docente puede llevar ejemplos de animaciones pre-creadas en Scratch en una memoria USB para mostrar como referencia. Si los computadores son limitados, los estudiantes pueden trabajar en parejas en un solo equipo.

Actividades evaluativas	
<p>Los estudiantes, crean animaciones propias que cambien de fondos usando el material propio de Scrath, deben usar un objeto que permita el cambio de fondos.</p>	
Referencias	<p><i>Guía para la enseñanza del pensamiento computacional.</i> <i>Coding for Kids. Brithis Council</i></p> <p>https://es.scribd.com/document/444448786/Fichas-coding-for-kids-V2-2-2-pdf</p>

ANEXO. Video con la explicación del código para la Portada
https://drive.google.com/file/d/1IrsA6lak0XtybvqGSxuydtkq9S_lgHs/view?usp=sharing

ANEXO 2.
SECUENCIA DIDÁCTICA: Recolectemos las Semillas!!

	<p><i>Indique el o los aprendizajes que busca desarrollar en las/los estudiantes durante la sesión de clase</i></p>
Aprendizaje(s) esperado(s)	<ul style="list-style-type: none"> • Implementar un sistema de juego con tiempo límite Basado en el movimiento en vertical del sol. • Programar las colisiones entre el puntero del ratón y objetos (semillas). • Utilizar variables (SEMILLAS) para llevar el conteo de elementos recolectados. • Aplicar el cambio de disfraces en la semilla al tocar los bordes
Materiales requeridos	<ul style="list-style-type: none"> • Computadores con Scratch (versión de escritorio si no hay internet). • Tablero y marcadores. • Cuaderno, lápices, colores y lapiceros.
Conocimientos previos requeridos	<ul style="list-style-type: none"> • Familiaridad básica con el entorno de Scratch (interfaz de usuario, bloques de código y su funcionalidad en la lógica del juego). • Entendimiento de lo que es un videojuego y sus elementos principales (título, botón de inicio, personajes)

Actividad(es) a desarrollar

Indique las acciones que realizarán el/la docente y sus estudiantes y las indicaciones si el trabajo se debe realizar de forma individual, en parejas o grupal.

Inicio (15 minutos - Desconectado)

Docente: Comienza revisando con los estudiantes lo que aprendieron y programaron en el Nivel 1. Luego, presenta el reto del Nivel 2: “¡Rápido! Recolecemos semillas antes de que el sol se ponga”. Explica que el objetivo será programar la caída del sol como si siguiera un movimiento de caída libre y que el jugador deberá recolectar semillas de maíz antes de que esto ocurra.

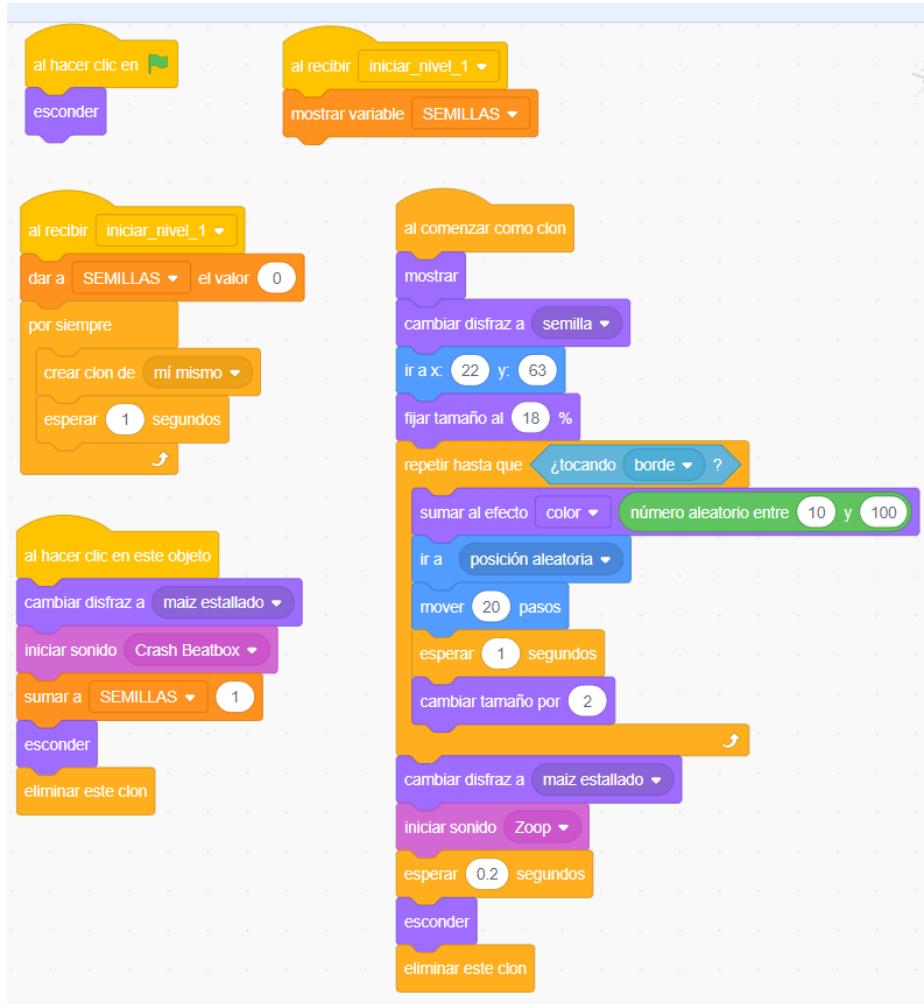
Estudiantes: En parejas o grupos pequeños, discutan ideas sobre cómo simular el movimiento del sol, cómo hacer que las semillas aparezcan y se muevan, y cómo el jugador podría “tomarlas” con el puntero. Propongan formas creativas para que las semillas desaparezcan al ser recolectadas.

Desarrollo (75 minutos - Conectado)

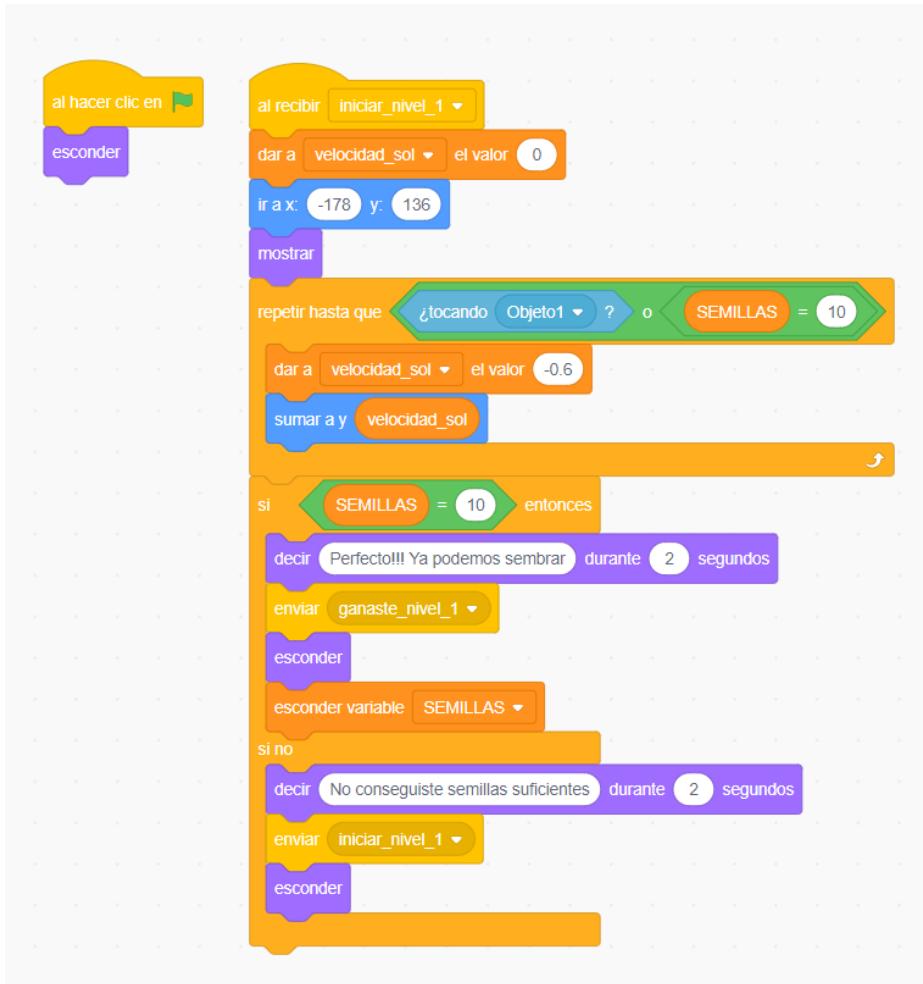
Docente: Enseña cómo crear una variable llamada *velocidad* para simular la gravedad. Muestra cómo esta variable puede aumentar gradualmente dentro de un bucle para lograr el efecto de caída del sol. Luego, explica cómo programar la interacción entre el jugador y las semillas usando la detección de colisiones, específicamente cuando el puntero o personaje toca una semilla de maíz.

Demuestra cómo hacer que las semillas desaparezcan (ocultarse) o cambien de disfraz al ser tocadas. Introduce una nueva variable llamada *conteo de semillas* que aumente en uno cada vez que el jugador recolecta una. Supervisa el avance de los estudiantes mientras aplican estos conceptos en su propio código.

Programación del Objeto Sol:



Programación del objeto semillas:



- **Estudiantes (Individual):** Programan el movimiento de su personaje. Crean sprites para las semillas de maíz y los duplican. Implementan la lógica de colisión para que, al tocar una semilla, esta desaparezca y el contador de semillas aumente. Añaden un temporizador que detiene el juego o cambia el escenario cuando el tiempo se agota. Experimentan con el cambio de disfraces para el personaje o las semillas

Ver el vídeo que se encuentra en el anexo:

Cierre (30 minutos - Desconectado/Conectado):

- **Docente:** Invita a los estudiantes a probar sus juegos y a mostrar cómo funciona el movimiento del sol y la recolección de semillas.
- **Estudiantes (Grupal):** Comparten sus avances y los desafíos encontrados. Se discuten las "habilidades" desarrolladas: control de movimiento, condiciones, variables de conteo, tiempo límite, colisiones y cambio de disfraces. Se anticipa el próximo desafío: evitar a las hormigas arrieras.

Tiempo estimado: 120 minutos

.

Adaptaciones

- **Población rural sin Internet:** Utilizar la versión de escritorio de Scratch. Los diseños de los estudiantes pueden ser dibujados directamente en cuadernos antes de pasarlos a Scratch. El docente puede llevar ejemplos de animaciones pre-creadas en Scratch en una memoria USB para mostrar como referencia. Si los computadores son limitados, los estudiantes pueden trabajar en parejas en un solo equipo.

Actividades evaluativas

El docente verificará que los estudiantes hayan:

- Implementado una variable SEMILLAS que afecte el juego (ej. terminando el juego o cambiando el escenario).
- Programado la detección de colisiones entre el personaje y las semillas.
- Utilizado una variable para contar las semillas recolectadas.
- Aplicado cambios de disfraces al recolectar semillas.
- Se observará la lógica de las condiciones ("si toca la semilla...") y el uso adecuado de las variables

Referencias

Guía para la enseñanza del pensamiento computacional.
Coding for Kids. British Council

<https://es.scribd.com/document/444448786/Fichas-coding-for-kids-V2-2-2-pdf>

ANEXO

<https://drive.google.com/file/d/1mJslWs4LtwbIt7Hh0CwtqBU6C0YHuggx/view?usp=sharing>

ANEXO 3.
SECUENCIA DIDÁCTICA: Evitemos los Insectos!!

Aprendizaje(s) esperado(s)	<p><i>Indique el o los aprendizajes que busca desarrollar en las/los estudiantes durante la sesión de clase</i></p> <ul style="list-style-type: none"> • Programar la aparición aleatoria de objetos (Insectos) en el escenario. • Implementar la detección de colisiones para evitar obstáculos. • Aplicar efectos visuales simples para mejorar la interactividad del juego. • Integrar los elementos de los niveles anteriores para un juego más complejo.
Materiales requeridos	<ul style="list-style-type: none"> • Computadores con Scratch (versión de escritorio si no hay internet). • Tablero y marcadores. • Cuaderno, lápices, colores y lapiceros.
Conocimientos previos requeridos	<ul style="list-style-type: none"> • Conocimiento de cómo crear y programar múltiples sprites. • Comprensión de variables y condicionales (Si/Entonces). • Haber completado satisfactoriamente los Niveles 1 y 2 del proyecto.
Actividad(es) a desarrollar	<p><i>Indique las acciones que realizarán el/la docente y sus estudiantes y las indicaciones si el trabajo se debe realizar de forma individual, en parejas o grupal.</i></p>
Inicio (15 minutos - Desconectado):	<ul style="list-style-type: none"> • Docente: Presenta el Nivel 3: "¡Detengamos los Insectos!", donde los estudiantes deben evitar que los insectos toquen al jugador. Se relaciona con la importancia de proteger los cultivos. • Estudiantes (Individual/Grupal): Discuten cómo podrían hacer que los insectos aparezcan en diferentes lugares de la pantalla y cómo programar al jugador para que las evite. Consideran qué sucede si un

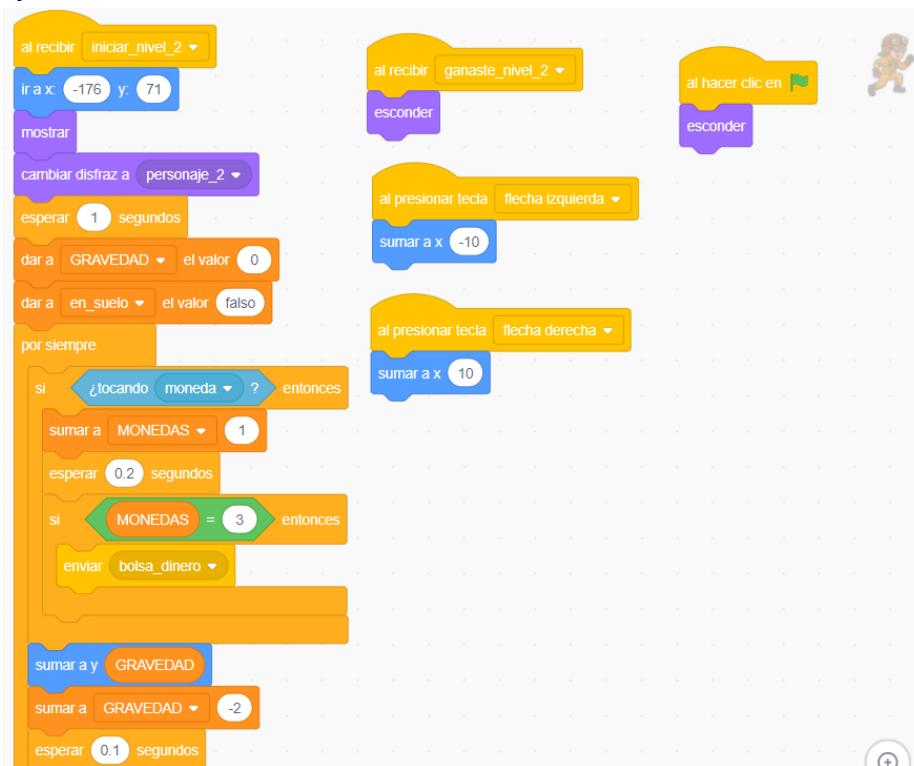
insecto toca al jugador (ej. el juego se reinicia).

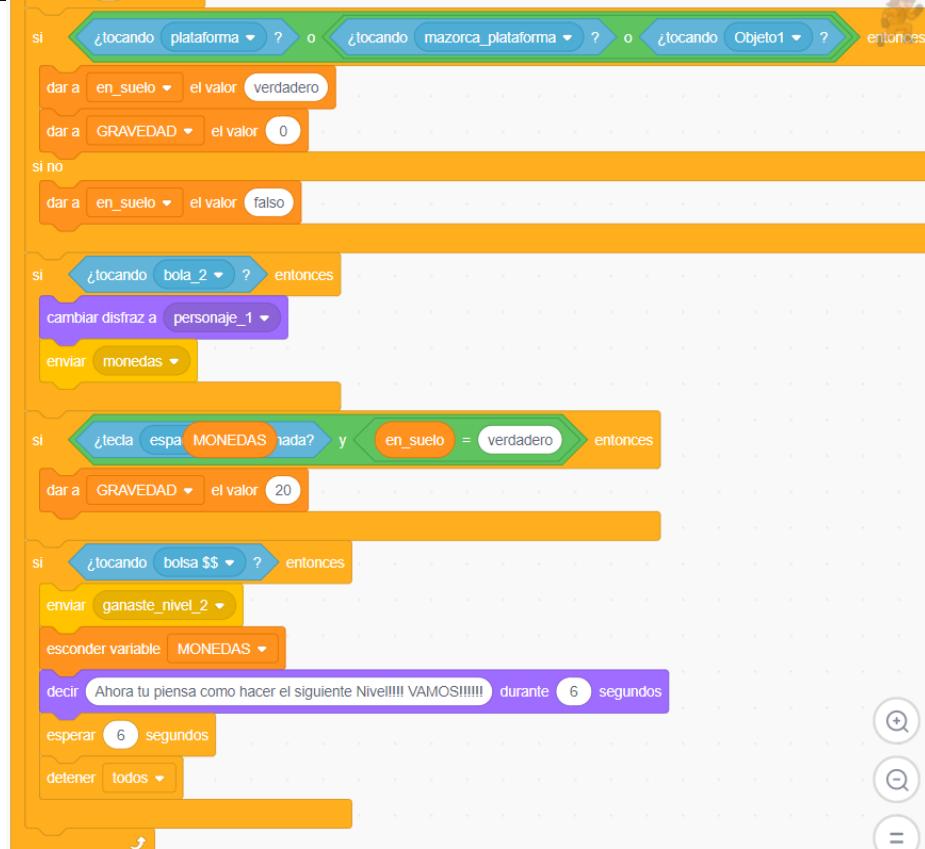
- **Desarrollo (75 minutos - Conectado):**

- **Docente:** Demuestra cómo clonar sprites (para las monedas) y cómo hacer que aparezcan aleatoriamente los insectos en el escenario y se muevan. Explica cómo programar la detección de colisiones entre el jugador y las monedas así como la bolsa de dinero, y qué acción debe ocurrir (ej. reiniciar posición). Introduce conceptos de efectos visuales (ej. cambiar de disfraz).
- **Estudiantes (Individual):** Crean el sprite del insecto. Programan su aparición aleatoria y movimiento. Implementan la lógica de colisión con el jugador, asegurándose de que el jugador las evite. También el movimiento y las físicas del personaje con las plataformas que le permiten llegar hasta las monedas y la bolsa de dinero.

Programación de Objetos del Nivel 2

Personaje



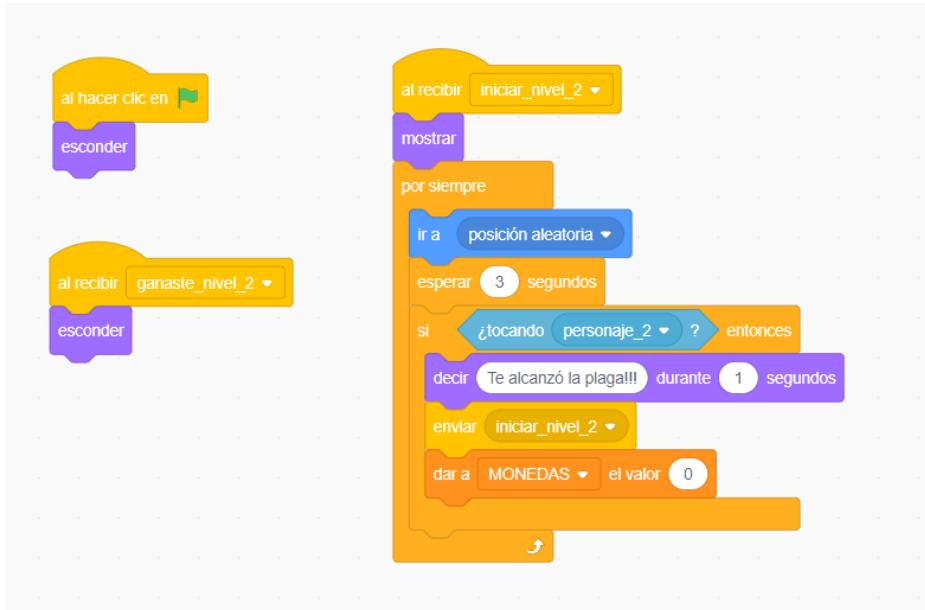


```

when green flag clicked
  if (touched) [plataforma v? or 0 or mazorca_plataforma v? or 0 or Objeto1 v?]
    then
      set [en_suelo v] to [verdadero]
      set [GRAVEDAD v] to [0]
    else
      set [en_suelo v] to [falso]
    end
  end
  if (touched) [bola_2 v?]
    then
      change costume to [personaje_1 v]
      send [monedas v] to [all]
    end
  end
  if (touched) [espaciado MONEDAS vada? and en_suelo = verdadero]
    then
      set [GRAVEDAD v] to [20]
    end
  end
  if (touched) [bolsa $$ v?]
    then
      say [Ahora tu piensa como hacer el siguiente Nivel!!!! VAMOS!!!!] for [6] seconds
      wait [6] seconds
      stop [all v]
    end
  end
end

```

Insecto 1

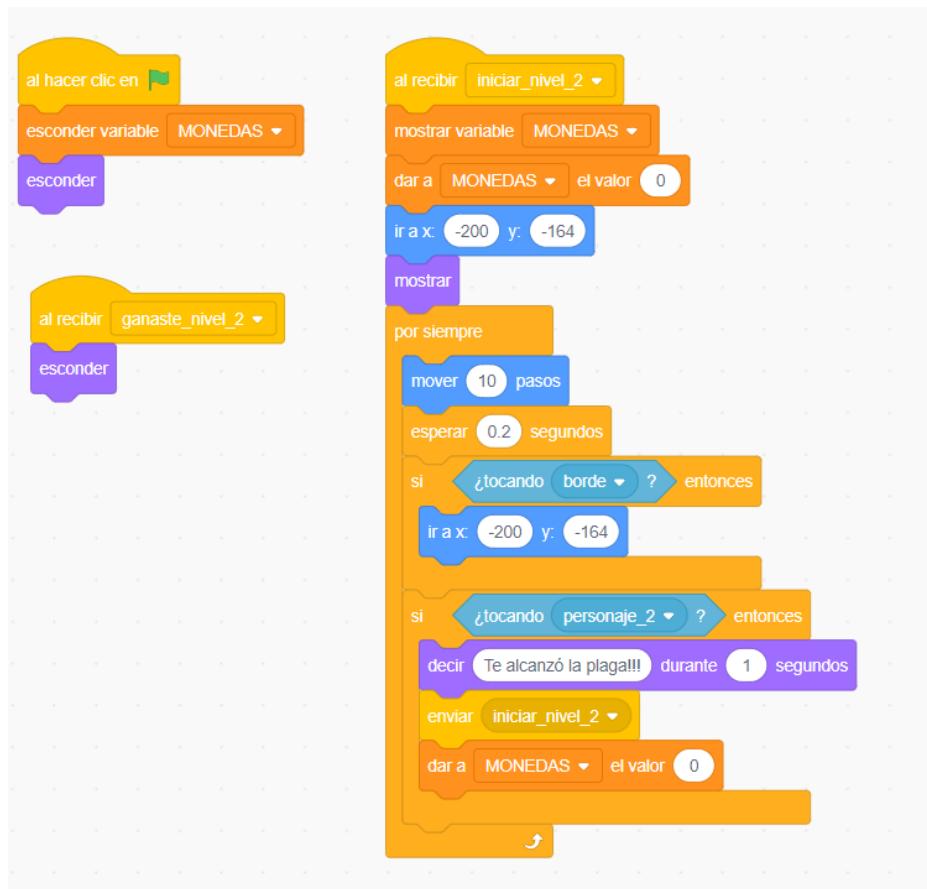


```

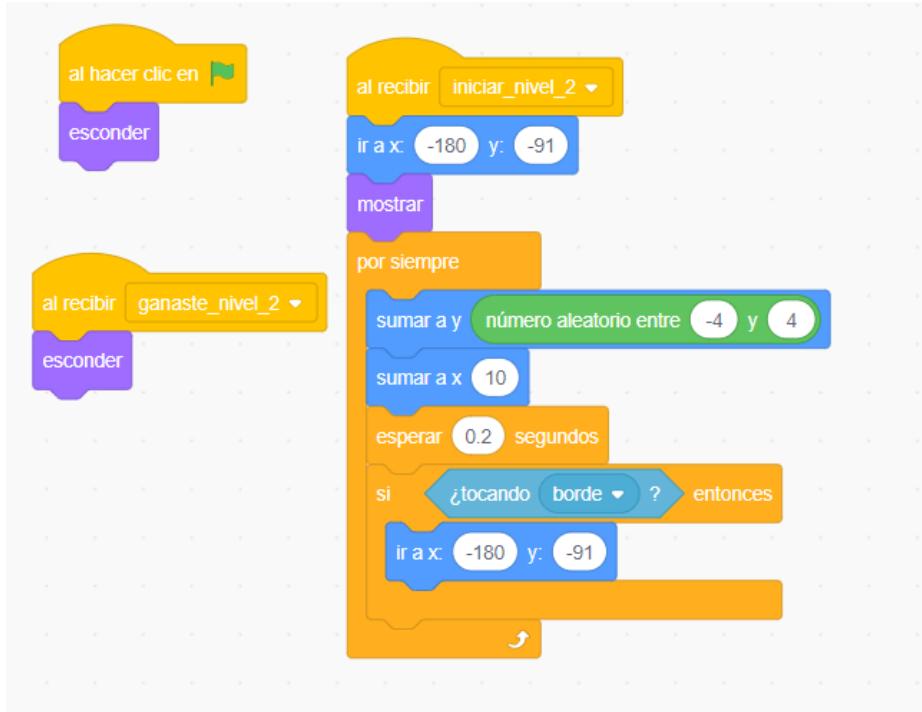
when green flag clicked
  hide
when [ganaste_nivel_2 v] received [ ]
  then
    show
    forever
      go to [random position v]
      wait [3] seconds
      if (touched) [personaje_2 v?]
        then
          say [Te alcanzó la plaga!!!] for [1] seconds
          send [iniciar_nivel_2 v] to [all]
          set [MONEDAS v] to [0]
        end
      end
    end
end

```

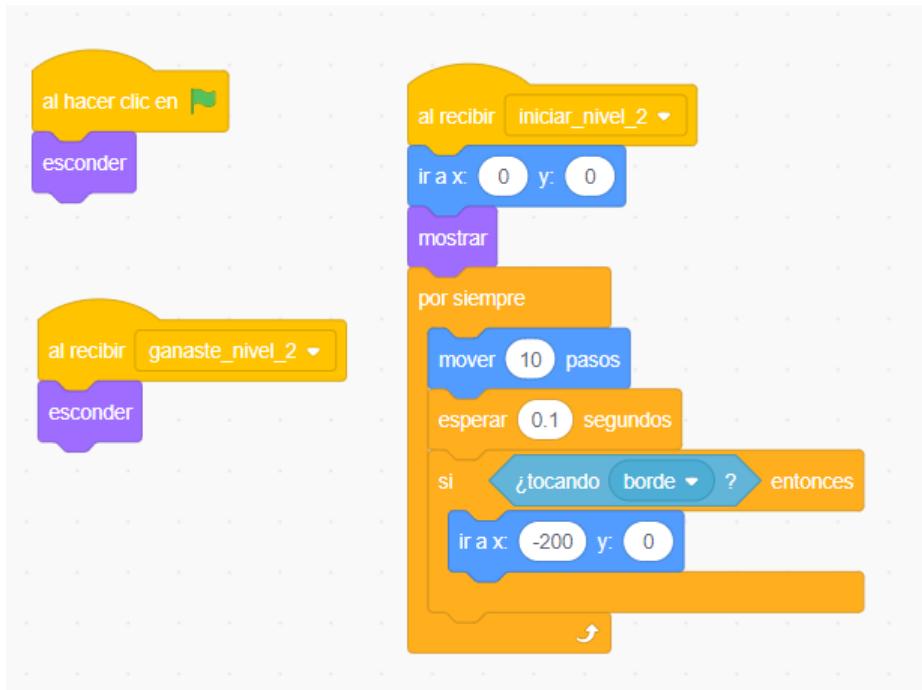
Insecto 2



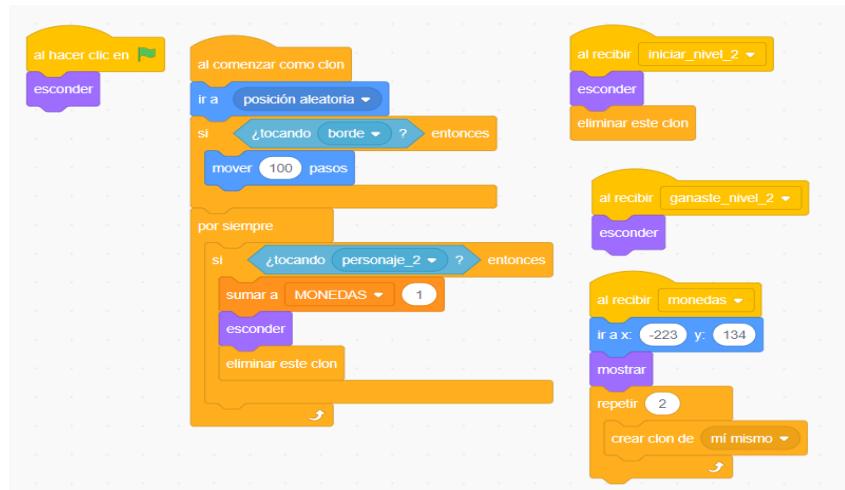
Plataforma 1



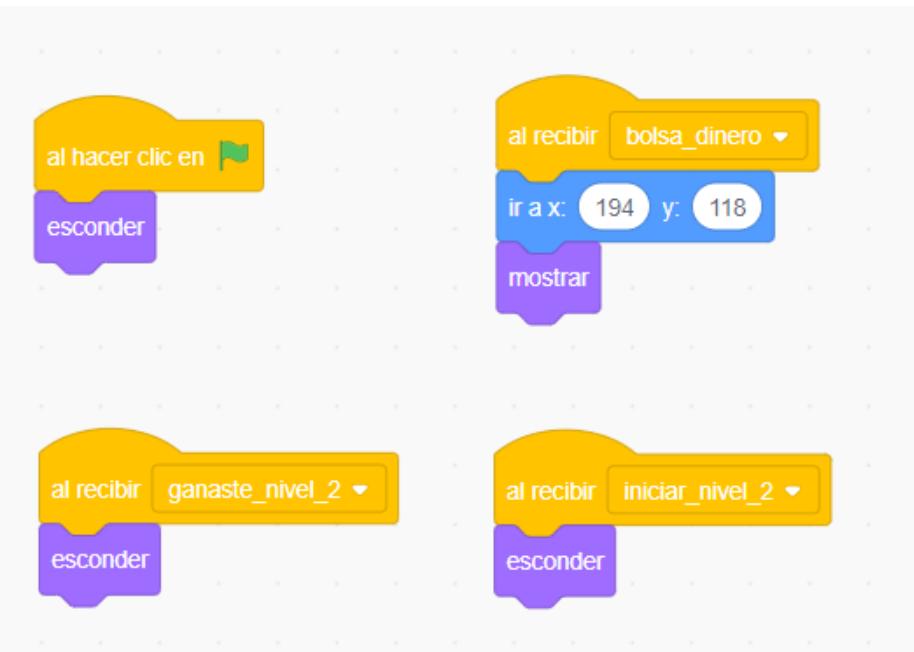
Plataforma 2



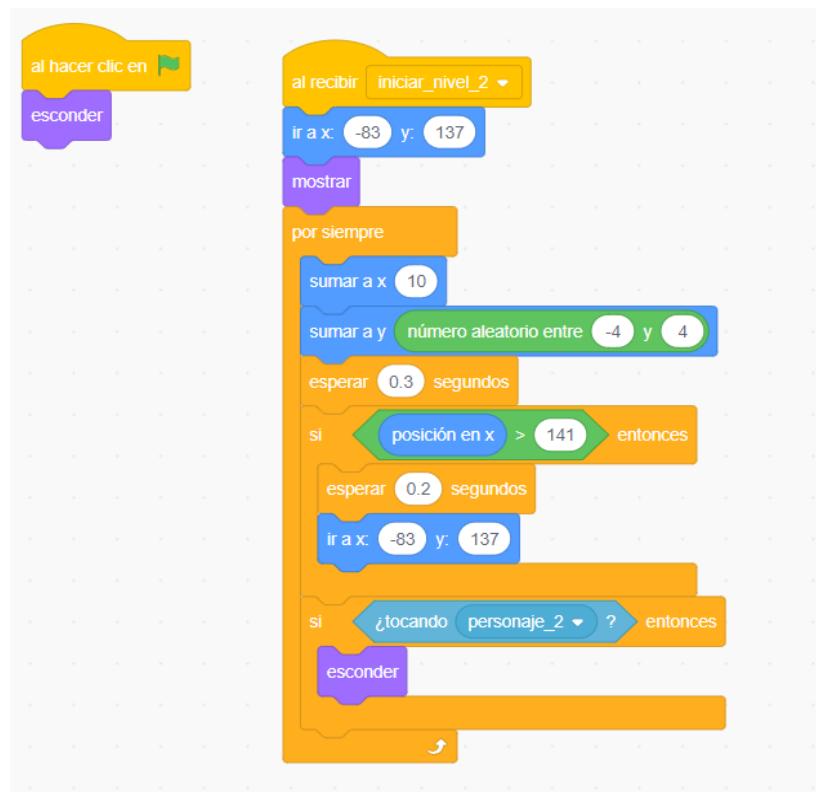
Monedas



Bolsa de dinero



Bola



- **Cierre (30 minutos - Desconectado/Conectado):**

- **Docente:** Guía una sesión de prueba y depuración, donde los estudiantes prueban sus juegos y se ayudan mutuamente a solucionar errores.
- **Estudiantes (Grupal):** Presentan la versión final de su juego con los tres niveles integrados. Se hace un resumen de las "habilidades" adquiridas: lógica de aparición aleatoria, detección de colisiones, efectos visuales. Se anima a los estudiantes a pensar en un posible Nivel 4 ("La carrera por los huevos").

Tiempo estimado: 120 minutos

<p>Adaptaciones</p> <ul style="list-style-type: none"> • Población rural sin Internet: Utilizar la versión de escritorio de Scratch. Los diseños de los estudiantes pueden ser dibujados directamente en cuadernos antes de pasarlos a Scratch. El docente puede llevar ejemplos de animaciones pre-creadas en Scratch en una memoria USB para mostrar como referencia. Si los computadores son limitados, los estudiantes pueden trabajar en parejas en un solo equipo. 	
<p>Actividades evaluativas</p> <p>El docente evaluará la capacidad de los estudiantes para:</p> <ul style="list-style-type: none"> • Programar la aparición y movimiento aleatorio de las hormigas. • Implementar la lógica de detección de colisiones para que el jugador evite las hormigas y el movimiento en el escenario sobre las plataformas • Integrar satisfactoriamente los tres niveles del juego (portada, recolección de semillas y evadir hormigas). • Explicar la lógica de su código para la aparición aleatoria y las colisiones 	
Referencias	<p><i>Guía para la enseñanza del pensamiento computacional. Coding for Kids. Brithis Council</i></p> <p>https://es.scribd.com/document/444448786/Fichas-coding-for-kids-V2-2-2-pdf</p>

ANEXO

<https://drive.google.com/file/d/1UW5liE6bEtPv0-QKzPlgmTBYDUtzA3Xj/view?usp=sharing>