

Plan de estudios CAL-ScratchJr Para 1er Grado (CAL-ScratchJr-1)



**Usando el enfoque de Programación como otro idioma (CAL)
desarrollado por Grupo de investigación DevTech**

Plan de Estudios de 1er Grado

TABLA DE CONTENIDOS

La pedagogía

Organización del plan de estudios: Ideas poderosas

Habilidades ScratchJr

Diseño del plan de estudio integrado

Estimulación

Materiales

La gestión del aula

Tamaño de grupos

Alineación del marco académico

Evaluaciones

Lecciones

Lección 1: Hablando en código

Lección 2: El orden importa

Lección 3: ¡Ingresa a ScratchJr!

Lección 4: Tu primer programa

Lección 5: Programa la canción Hokey-Pokey

Lección 6: Re-programa el Hokey-Pokey

Lección 7: Piensa como un creador

Lección 8: Meterse en el personaje

Lección 9: Cuenta un cuento

Lección 10: Amplía tu caja de herramientas para contar historias

Lección 11: ¿Cuál es tu historia?

Lección 12: ¡Encuentra ese error!

Lección 13: ¿Puedes oírme?

Lección 14: Volver a empezar

Lección 15: ¡Necesita velocidad!

Lección 16: ¡Hagamos el baile de las estatuas!

Lección 17: ¿Puedes repetir eso?

Lección 18: Haz que suceda

Lección 19: Tu Proyecto Final I

Lección 20: Tu Proyecto Final II

Lección 21: Tu Proyecto Final III

Lección 22: Your Final Project IV

Lección 23: Nuestra comunidad de programación

Lección 24: Muestra de los Proyectos Finales

El plan de estudios Programación como otro idioma (CAL) para ScratchJr (CAL-ScratchJr) presenta ideas poderosas de la informática, en conversación con la alfabetización de una manera lúdica, estructurada y apropiada para el desarrollo. El enfoque Coding as Another Language (CAL), desarrollado por la profesora Marina Umaschi Bers y miembros de su grupo de investigación DevTech, entiende el aprendizaje de la informática como una nueva alfabetización para el siglo XXI que ayuda a los jóvenes estudiantes a desarrollar nuevas formas. de pensar en sí mismos y en el mundo.

La programación como otro idioma (CAL) para ScratchJr (CAL-ScratchJr) del [DevTech Research Group](#) tiene una licencia [Creative Commons Reconocimiento No comercial-Compartir por igual 3.0 Licencia no exportada..](#)

Con esta licencia, puede utilizar y adaptar este trabajo, pero debe atribuir el trabajo al Grupo de investigación de DevTech. **No se puede** utilizar ni adaptar este trabajo con fines comerciales.

Este trabajo está financiado por el Departamento de Educación de EE. UU. (Subvención # U411C190006) y por la Fundación Scratch.

© 2020, DevTech Research Group.

Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) by the [DevTech Research Group](#) is licensed under a [Creative Commons Attribution NonCommercial-ShareAlike 3.0 Unported License](#). Under this license, you may use and adapt this work, but you must attribute the work to the DevTech Research Group. You **may not** use or adapt this work for commercial purposes.

This work is funded by the US Dept of Education (grant #U411C190006) and by the Scratch Foundation.

© 2020, DevTech Research Group.

La Pedagogía

El plan de estudios CAL se implementa siguiendo el enfoque pedagógico desarrollado por la profesora Marina Bers que se puede encontrar en sus libros *Coding as a Playground and Beyond Coding: Teaching Human Values to Children*. Esta pedagogía tiene cuatro marcos principales:

Programación como área de juegos:

Al involucrar a los niños en una experiencia de aprendizaje de las ciencias de la computación, damos la bienvenida al juego. A través del juego podemos incidir en todas las áreas del desarrollo humano: cognitivo, socioemocional, lingüístico, moral, físico e incluso espiritual. El patio de juegos de la programación, en contraste con el parque de la programación, promueve oportunidades para la exploración abierta, la creación de proyectos personalmente significativos, la imaginación, la resolución de problemas, la resolución de conflictos y la colaboración. El patio de juegos de la programación involucra a los niños en seis comportamientos que también podemos encontrar en el patio de juegos regular: creación de contenido, creatividad, elecciones de conducta, comunicación, colaboración y construcción de comunidad. Estos 6C se explican con más detalle en el marco teórico de PTD (Desarrollo tecnológico positivo) en el libro de Marina Bers, y se destacan en las actividades a lo largo del plan de estudios con sus respectivos íconos:



CREACIÓN Y CONTENIDO diseñando un programa ScratchJr y programando sus comportamientos. El proceso de diseño de ingeniería de la construcción y el pensamiento computacional involucrado en la programación fomentan la competencia en informática y fluidez tecnológica.



CREATIVIDAD haciendo y programando proyectos personalmente significativos, resolviendo problemas de formas creativas y lúdicas e integrando diferentes medios como materiales reciclables, artes y manualidades y un lenguaje de programación tangible. Los proyectos finales de ScratchJr que representan un tema que se encuentra en el plan de estudios general de la primera infancia son una manera maravillosa de involucrar a los niños en el proceso creativo de aprendizaje.



COLABORACIÓN al involucrar a los niños en un entorno de aprendizaje que promueve el trabajo en equipo, compartiendo recursos y preocupándose unos de otros mientras trabajan con sus programas ScratchJr. La colaboración se define aquí como obtener o brindar ayuda con un proyecto, programar juntos, prestar o pedir prestado materiales o trabajar juntos en una tarea común. Mientras trabajan en sus proyectos finales, los niños crean una red de colaboración: una herramienta que se utiliza para fomentar la colaboración y el apoyo. Los niños pueden escribir o dibujar “tarjetas de agradecimiento” a los otros niños con los que más han colaborado.



COMUNICACIÓN a través de mecanismos que promueven un sentido de conexión entre pares o con adultos. Por ejemplo, los círculos tecnológicos, cuando los niños dejan de trabajar, comparten su creación ScratchJr y explican su proceso de aprendizaje. Los círculos tecnológicos presentan una buena oportunidad para la resolución de problemas como comunidad. Algunos maestros invitan a todos los niños a sentarse juntos en el área de la alfombra para esto. Cada salón de clases tendrá sus propias rutinas y expectativas en torno a las discusiones de grupo y los tiempos de círculo, por lo que se anima a los maestros a adaptar lo que ya funciona en su clase para los círculos tecnológicos en este plan de estudios.



CONSTRUCCIÓN COMUNITARIA a través de oportunidades escalonadas para formar una comunidad de aprendizaje que promueva la contribución de ideas. Los proyectos finales realizados por los niños se comparten con la comunidad a través de una jornada de puertas abiertas, un día de demostración o una exhibición. Estas jornadas de puertas abiertas brindan oportunidades auténticas para que los niños compartan y celebren el proceso y los productos tangibles de su aprendizaje con familiares y amigos. A cada niño se le da la oportunidad no solo de ejecutar su programa, sino de desempeñar el papel de maestro mientras explican a su familia cómo construyeron, programaron y resolvieron los problemas.

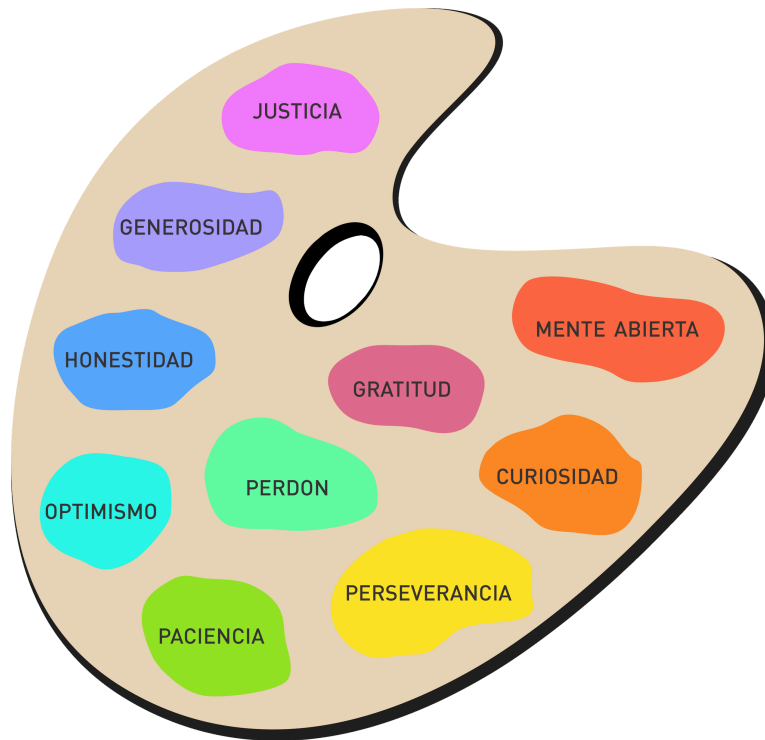


ELECCIONES DE CONDUCTA que brindan a los niños la oportunidad de experimentar con preguntas de "qué pasaría si" y las posibles consecuencias, y de provocar el examen de valores y la exploración de los rasgos de carácter mientras trabajan con ScratchJr. Como programa desarrollado siguiendo el enfoque de PTD, el enfoque en el aprendizaje de la codificación es tan importante como ayudar a los niños a desarrollar una brújula interior para guiar sus acciones de una manera justa y responsable.





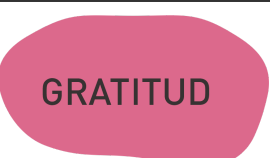


La programación como otro idioma: Caracterizar la programación como una actividad STEM es limitante. En cambio, si posicionamos la enseñanza y el aprendizaje de la programación como una nueva forma de pensar y expresarnos, estamos en el dominio del lenguaje. Dominar un sistema simbólico de representación con funciones comunicativas y expresivas abre muchas oportunidades. Aprender a programar se convierte en una actividad creativa y expresiva para producir algo significativo y que se pueda compartir, y no solo un conjunto de habilidades para la resolución de problemas. La pedagogía CAL promueve la exploración de las similitudes y diferencias entre los lenguajes naturales y artificiales para el proceso de creación, su sintaxis y gramática, y su potencial para empoderar a las personas. Cuando la programación se enseña como un lenguaje, y no solo como STEM, entra en juego la dimensión humana.

La programación como puente: La programación es un acto semiótico, una actividad de creación de significado que utiliza y combina símbolos para representar ideas abstractas. Cuando aprendemos a programar, aprendemos un nuevo idioma. Los lenguajes pueden crear o destruir, y pueden servir para construir puentes o muros. En el campo de la programación, la intención es construir puentes. La pedagogía CAL propone que al aprender el lenguaje artificial de las máquinas, también podemos aprender el lenguaje humano que nos sirve para interactuar con los demás, para conectarnos de manera profunda y crear relaciones significativas. La metáfora de la programación como puente promueve el diálogo y los encuentros significativos con los demás y sitúa la programación como un aprendizaje socioemocional y no solo como una actividad cognitiva.

La programación como paleta de virtudes: Cualquier actividad humana involucra valores humanos. Tomar decisiones para participar en algunos comportamientos y no en otros. Comprender y asumir responsabilidades y consecuencias. El patio de juegos de la programación, a través de la pedagogía CAL, brinda una oportunidad intencional para enseñar y aprender valores. La metáfora de una paleta de virtudes recuerda la paleta del pintor. Al igual que un artista que hace su paleta con nuevos colores y los mezcla y combina, la programación también tiene una paleta de virtudes dinámicas que utiliza. En el campo de la programación, diez de estos valores se exploran explícitamente: curiosidad, perseverancia, mentalidad abierta, optimismo, honestidad, paciencia, generosidad, gratitud, perdón, equidad. Sin embargo, se pueden agregar otros nuevos. La programación creativa puede ser un camino para el desarrollo del carácter, para explorar la dimensión socioemocional y la dimensión ética del aprendizaje. En definitiva, entender que nuestras acciones, como las acciones de cualquiera que crea, tienen consecuencias.



Valor	Definición	Actividades
CURIOSIDAD	Un fuerte deseo de saber o aprender algo; búsqueda de la novedad.	Estrella y un deseo
PERSEVERANCIA	Persistencia en un curso de acción a pesar de la dificultad o adversidad; determinación en la consecución de objetivos; firmeza de propósito; creencia de que podemos mejorar.	Tablero de depuración Soy CODIFICADOR
PACIENCIA	La capacidad de aceptar o tolerar retrasos, problemas o sufrimiento sin enojarse o enojarse.	Tablero de depuración Soy CODIFICADOR

 OPTIMISMO	Esperanza y confianza sobre el futuro o un resultado exitoso; esperando lo mejor.	Estrella y un deseo Cruzando la habitación
 HONESTIDAD	La cualidad de ser honorable; rectitud; nobleza; integridad	Tablero de depuración Transmitiendo las reglas
 JUSTICIA	Trato imparcial y justo sin favoritismos ni discriminación; justicia	Transmitiendo las reglas Intercambio
 GENEROSIDAD	La cualidad de ser amable; dar y recibir, ayudarse a sí mismo y a los demás	Hacer una Web (en persona) Intercambio
 GRATITUD	La cualidad de estar agradecido; disposición a mostrar aprecio; ser consciente de las cosas buenas	Estrella y un deseo Hacer una Web (en persona) Intercambio
 PERDON	La acción de perdonar y aceptar, dando una segunda oportunidad a uno mismo y a los demás.	Soy CODIFICADOR
 MENTE ABIERTA	La cualidad de estar dispuesto a considerar ideas, opiniones y sentimientos nuevos o diferentes a los suyos; flexibilidad para adoptar diferentes perspectivas.	Tablero de depuración Cruzando la habitación

Organización del plan de estudios: Ideas poderosas

El plan de estudios está organizado en torno a ideas poderosas tanto de la informática como de la alfabetización. El término **idea poderosa** se refiere a un concepto o habilidad central dentro de una disciplina que es simultáneamente útil personalmente, **está** intrínsecamente interconectado con otras disciplinas y tiene sus raíces en el conocimiento intuitivo que un niño ha interiorizado durante un largo período de tiempo.

Las **ideas poderosas de la ciencia de la computación** que se abordan en este plan de estudios incluyen: algoritmos, proceso de diseño, representación, depuración, estructuras de control, modularidad y hardware / software. Las **ideas poderosas de la alfabetización** que se pondrán en conversación con estas poderosas ideas de la informática son: secuenciación, proceso de escritura, correspondencia alfabeto y letra-sonido, edición y conciencia de la audiencia, recursos literarios, conciencia fonológica y herramientas de comunicación y lenguaje. (ver Tabla 1).

Tabla 1: Ideas poderosas

Ideas poderosas de la informática	Ideas poderosas de la alfabetización	Conectando las ideas poderosas
Algoritmos <i>Lecciones relevantes: 3, 7, 8, 12, 19, 20, 21, 22</i>	Secuenciación <i>Lecciones Relevantes: 3, 7, 8, 9, 11, 12, 19, 20, 21, 22</i>	El énfasis en “el orden es importante” y que las tareas complejas se pueden dividir en instrucciones paso a paso de una manera lógica..
Proceso de diseño <i>Lecciones relevantes: 9, 10, 11, 12, 13, 14, 16, 21, 22, 23, 24</i>	Proceso de escritura <i>Lecciones Relevantes: 9, 10, 12, 13, 14, 16, 21, 22, 23, 24</i>	Procesos creativos, iterativos, cíclicos que implican imaginar, planificar, hacer, revisar y compartir, con diferentes puntos de partida.
Representación <i>Lecciones Relevantes: 5, 6, 7, 8, 13, 16, 17, 18</i>	Correspondencia entre el alfabeto y el sonido de las letras <i>Lecciones Relevantes: 5, 6, 14</i>	Los símbolos tienen diferentes atributos (color, forma, sonido, etc.) para representar algo más.

Depuración <i>Lecciones Relevantes: 14, 22, 23</i>	Edición y conocimiento del público <i>Lecciones Relevantes: 3, 14, 17, 21, 22, 23</i>	Análisis, pruebas y evaluación sistemáticos para mejorar la comunicación con la audiencia destinataria (computadora o persona). Siempre que ocurre una falta de comunicación, el programador o escritor usa una variedad de estrategias para resolver el problema.
Estructuras de Control <i>Lecciones Relevantes: 15, 16, 18, 19</i>	Recursos literarios <i>Lecciones Relevantes: 15, 18, 19</i>	Estrategias avanzadas para comunicar un conjunto de ideas mediante repetición, patrones, condicionales y eventos.
Modularidad <i>Lecciones Relevantes: 6, 7, 8, 11, 12, 16, 18, 19, 21</i>	Conocimiento fonológico <i>Lecciones Relevantes: 2, 7, 8, 14, 16</i>	Descomponer o dividir una tarea compleja en tareas más pequeñas y reutilizar esos nuevos módulos.
Hardware/Software <i>Lecciones Relevantes: 1, 2, 4, 17</i>	Herramientas de comunicación y lenguaje <i>Lecciones Relevantes: 2, 4, 11, 17</i>	Comunicar ideas abstractas a través de medios tangibles. Al igual que el hardware y el software funcionan juntos, la expresión de pensamientos a través del lenguaje requiere un medio para comunicarse con el mundo exterior, como la palabra hablada o escrita.

Habilidades ScratchJr

Las habilidades más importantes de ScratchJr utilizadas en cada lección se enumeran a continuación (consulte la Tabla 2). Tenga en cuenta que esta no es una lista completa porque cada actividad debe ser creativa y, por lo general, abierta. Esta tabla está destinada a indicar las habilidades sin las que sería difícil completar una lección. A los niños siempre se les anima a usar cualquier bloque o habilidad que aprendan en clase o por su cuenta en cualquier proyecto.

Tabla 2: Habilidades de ScratchJr

Habilidades ScratchJr	Lecciones relevantes
Comenzar al presionar bandera verde	Lessons 4*, 5
Comenzar al presionar	Lesson 14*
Comenzar al tocar	Lesson 14*
Comenzar al recibir mensaje	Lesson 18*
Bloque enviar mensaje	Lesson 18*
Mover bloque a la derecha	Lessons 4*, 5
Mover bloque a la izquierda	Lessons 4*, 5
Bloque subir	Lessons 4*, 5
Mover bloque hacia abajo	Lessons 4*, 5
Bloque girar a la derecha	Lessons 4*, 5
Bloque girar a la izquierda	Lessons 4*, 5
Bloque de salto	Lessons 4*, 5
Bloque ir al inicio	Lessons 15*
Bloque decir	Lessons 13*
Bloque crecer	Lesson 10*
Bloque disminuir	Lesson 10*
Bloque restablecer tamaño	Lesson 10*
Bloque ocultar	Lesson 10*
Bloque mostrar	Lesson 10*
Bloque sonido Pop	Lesson 6*
Bloque reproducir sonido grabado	Lesson 13*
Bloque esperar	Lessons 16*

Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) by the [DevTech Research Group](#) is licensed under a [Creative Commons Attribution NonCommercial-ShareAlike 3.0 Unported License](#). Under this license, you may use and adapt this work, but you must attribute the work to the DevTech Research Group. You **may not** use or adapt this work for commercial purposes.

This work is funded by the US Dept of Education (grant #U411C190006) and by the Scratch Foundation.

© 2020, DevTech Research Group.

Bloque parar	Lesson 14*
Bloque fijar velocidad	Lessons 15*
Bloque repetir	Lesson 17*
Bloque final	Lessons 4*, 5
Bloque repetir por siempre	Lesson 6*
Bloque ir a la página	Lesson 10*, 19, 20
Guardar el nombre del proyecto	Lessons 3*, 23
Herramienta de cuadrícula	Lessons 5*, 15
Modo presentación	Lesson 4*
Bandera verde	Lesson 4*
Herramienta Agregar texto	Lessons 8*, 9
Herramienta de edición de pintura sobre personajes y fondo	Lesson 8*
Comenzando un nuevo proyecto	Lesson 3*
Agregar caracteres	Lessons 4*, 8, 21, 22
Agregar fondos	Lessons 9*, 19, 20, 22
Añadir página	Lessons 9*, 19, 20, 22
Creando un programa	Lesson 4*
Eliminar un personaje	Lessons 4*
Eliminar una página	Lessons 9*
Copiar guiones	Lesson 20*, 21
Ajuste de parámetros	Lessons 5*, 10, 17
Programación paralela	Lesson 16*

* This lesson contains the introduction or tutorial for the associated ScratchJr concept.

Diseño del plan de estudio integrado

El enfoque CAL permite a los niños hacer conexiones entre la programación y la alfabetización, así como entre la programación y las matemáticas. Este plan de estudios anima a los niños a expresar sus pensamientos, ideas y aprendizajes a través de las actividades de ScratchJr. El plan de estudios está diseñado para un total de 18 horas, pero se puede adaptar a entornos de aprendizaje particulares. Cada lección contiene una variedad de actividades, que incluyen:

- Actividades para introducir o reforzar conceptos de forma lúdica.
- Círculos tecnológicos de inicio y cierre para discutir, compartir y reflexionar sobre actividades y conceptos.
- Desafíos estructurados con ScratchJr para presentar las ideas poderosas de la informática.
- Exploraciones expresivas con ScratchJr para practicar la habilidad de una manera creativa y abierta.
- Juegos / actividades de tiempo desenchufado para promover las interacciones sociales y el movimiento.
- Tiempo de palabra para presentar las ideas poderosas de la alfabetización.

La culminación de la unidad es un proyecto abierto para compartir con familiares y amigos. Así como los niños pequeños pueden leer libros apropiados para su edad, la programación de computadoras se puede hacer accesible proporcionándoles a los niños pequeños las herramientas adecuadas, como ScratchJr. Este plan de estudios proporciona integración entre la informática y la programación en el contexto de la alfabetización. Los niños aprenderán a explorar dos libros, Biografía de la primera mujer programadora: “[Ada Lovelace: Pequeña y grande](#)” por María Isabel Sánchez Vegara e ilustraciones por Zafouko Yamamoto y [Donde viven los monstruos](#) de Maurice Sendak para escribir creativos y divertidos programas en ScratchJr.

Estimulación

Esta es una unidad curricular de 18 horas dividida en lecciones de aproximadamente 45 minutos. Algunos niños pueden beneficiarse de una mayor división de las actividades en pasos más pequeños o de más tiempo para explorar cada nuevo concepto antes de pasar al siguiente, ya sea en el contexto de la exploración libre o con desafíos diseñados por el maestro. Cada una de las ideas poderosas de la informática en este plan de estudios se puede expandir fácilmente a una unidad de estudio que ampliará el plan de estudios y permitirá a los niños explorar una variedad de actividades diferentes.

Materiales

Dado que este plan de estudios se basa en ScratchJr, el material principal necesario para los niños son iPads, Androids o Chromebooks (consulte aquí <https://www.ScratchJr.org/about/faq> para dispositivos compatibles con ScratchJr) para que los niños puedan codificar. Además, hay páginas de bloques de ScratchJr que se pueden imprimir para ayudar con la comprensión del niño. Se proporciona más información en las lecciones que utilizan estas páginas. Este plan de estudios también utiliza los libros “[Ada Lovelace: Pequeña y grande](#)” por María Isabel Sánchez Vegara e ilustraciones por Zafouko Yamamoto ([Audiolibro](#)) y [Donde viven los monstruos](#) de Maurice Sendak.

Otros materiales utilizados en el plan de estudios son manualidades económicas y materiales reciclados. El uso de manualidades y materiales reciclados, una práctica que ya es común en otros dominios de la educación infantil, brinda oportunidades para que los niños usen materiales con los que ya se sienten cómodos.

La gestión del aula

Durante el plan de estudios de CAL, los niños estarán cantando, moviéndose, hablando entre ellos, mostrándose el trabajo y haciendo preguntas. La pedagogía CAL ofrece orientación para desarrollar comportamientos positivos en el aula a través de estas y otras actividades relacionadas con la codificación y la tecnología. El marco de Desarrollo Tecnológico Positivo incluye opciones de conducta, que alientan a los niños a tomar decisiones positivas para ellos mismos y los miembros de su comunidad. Además, a través de la enseñanza de la programación como una paleta de virtudes, se anima a los maestros a incorporar el lenguaje de las virtudes y fomentar la discusión sobre las virtudes dentro de las actividades del aula. Sugerimos usar este lenguaje para guiar a los niños a tomar decisiones positivas durante las actividades de programación. En la descripción de nuestra pedagogía anterior se incluye más información sobre el desarrollo tecnológico positivo y la programación como paleta de virtudes, incluidas las actividades para promoverlas.

Tamaños de grupo

El plan de estudios se refiere al trabajo en grupo completo versus el trabajo en grupo pequeño o individual. La viabilidad de los grupos pequeños depende de la cantidad de maestros disponibles para supervisar los grupos y de las capacidades de la tecnología, que pueden estar limitadas por varias razones. Sin embargo, se debe hacer un esfuerzo para permitir que los niños trabajen en grupos tan pequeños como sea posible, para que tengan la oportunidad de participar activamente. Si los grupos pequeños no son posibles, se deben fomentar las sesiones de trabajo individuales durante el tiempo de clase. Los profesores pueden estar presentes durante toda la clase mientras todos trabajan en su propio

dispositivo y estarán disponibles para responder preguntas según sea necesario. Al mismo tiempo, el plan de estudios incluye numerosas oportunidades para promover conversaciones que se enriquecen con múltiples voces, puntos de vista y experiencias. Algunas clases pueden tener estas discusiones en grupo. Otras clases pueden dividirse en grupos más pequeños para permitir que más niños tengan la oportunidad de hablar y mantener la concentración. Ya sea en grupos pequeños o en discusiones de toda la clase, se debe enfatizar la comunicación en todo momento. Al aprender a programar, los niños aprenden a expresar sus ideas. Poder verbalizar esas ideas y conectarse con otros niños es una parte integral del plan de estudios.

Alineación del marco académico

El plan de estudios CAL ScratchJr fomenta las habilidades técnicas generales, el pensamiento computacional y las habilidades de ingeniería de los niños pequeños. También apoya las habilidades fundamentales de matemáticas, lectura y artes del lenguaje que se enseñan comúnmente en las aulas de la primera infancia. El plan de estudios está específicamente alineado con los [Marcos de Ciencias de la Computación K-12](#), los [Estándares ISTE para Estudiantes](#), así como los [Marcos Básicos Comunes de Matemáticas y Artes del Lenguaje en Inglés \(ELA\)](#). La siguiente tabla proporciona ejemplos de cómo las lecciones del plan de estudios 1er grado CAL ScratchJr están alineadas con los estándares y marcos particulares de 1er grado.

Tabla 3: Alineación del marco académico

Estándares estudiantiles ISTE	Marcos de Ciencias de la Computación K-12	Estándares básicos comunes de ELA de primer grado	Estándares básicos comunes de matemáticas de primer grado
Estándares Estudiantiles ISTE # 1 Alumno empoderado: 1a, 1c Lecciones relevantes: 5, 21, 22	K – 2.Sistemas informáticos. Dispositivos Lecciones relevantes: 1, 2, 3, 4 K – 2.Sistemas informáticos.Hardware y software Lecciones relevantes: 1, 3, 4, 5, 13	CCSS.ELA-LITERAC Y.RL.1.1 Lecciones relevantes: 7, 8, 9, 19 CCSS.ELA-LITERAC Y.RL.1.2 Lecciones relevantes: 7, 8, 9, 19	CCSS.MATH.CONTE NT.1.NBT.A.1 Lecciones relevantes: 5, 6, 16 CCSS.MATH.CONTE NT.1.G.A.1 Lecciones relevantes: 10, 22
Estándares Estudiantiles ISTE # 4 Diseñador innovador: 4a, 4b, 4c, 4d Lecciones relevantes: 5, 6, 7, 8, 10, 11, 19, 20, 21, 22, 23, 24	K – 2.Sistemas informáticos. Solución de problemas Lecciones relevantes: 10, 11, 12, 22	CCSS.ELA-LITERAC Y.RL.1.3 Lecciones relevantes:: 7, 8, 9, 13, 22	
Estándares Estudiantiles ISTE # 5 Pensador computacional - 5a, 5c, 5d Lecciones relevantes: 2, 5, 6, 7, 9, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23	K – 2.Algoritmos y programación.Control Lecciones relevantes: 13, 14, 15, 16, 17, 18	CCSS.ELA-LITERAC Y.RL.1.7 Lecciones relevantes: 9, 13, 22	
Estándares Estudiantiles ISTE # 6 Comunicador creativo 6b, 6c, 6d Lecciones relevantes:	K – 2.Algoritmos y programación. Algoritmo Lecciones relevantes: 2, 5, 6, 9, 16, 17, 19 20, 21, 22, 23 K – 2.Algoritmos y programación.	CCSS.ELA-LITERAC Y.SL.1.4 Lecciones relevantes: 13, 24 CCSS.ELA-LITERAC Y.SL.1.1 Lecciones relevantes: 6, 9, 11 22, 24	

21, 22, 23, 24	Modularidad Lecciones relevantes: 20, 21		
----------------	---	--	--

Evaluaciones

El plan de estudios CAL entiende la evaluación como un componente crítico del proceso de enseñanza y aprendizaje. Es una oportunidad para detener el trabajo y evaluar cómo van las cosas. La evaluación brinda oportunidades para revisar ideas, contenido y habilidades, para ajustar y modificar las prácticas de instrucción y para recopilar la información necesaria para comunicarse con los padres. Puede servir para eliminar brechas de rendimiento y modificar estrategias pedagógicas. Las evaluaciones CAL se basan en la observación de los comportamientos de los niños, la escucha de las historias y el razonamiento de los niños y el análisis del trabajo que producen. El plan de estudios CAL incluye tanto evaluaciones formativas que se integran a lo largo de las lecciones (estas se llaman Verificación de comprensión y se pueden encontrar en las lecciones 4, 9, 14, 16, 17, 18) como evaluaciones sumativas ([Muestra lo que sabes](#)) que se llevan a cabo al final del plan de estudios. Además, los proyectos de los niños se evalúan siguiendo la rúbrica del proyecto ScratchJr para identificar las habilidades de programación, así como la expresividad y el propósito de los proyectos creados. ([Rúbrica](#) para profesores, que incluye instrucciones, etc).

Cuando se usa CAL en el contexto de proyectos de investigación, hay otras tres formas de evaluación que se usan comúnmente.

Primero, la evaluación de etapas de la programación (CSA) que ocurre antes de comenzar el plan de estudios y después de su finalización. La CSA evalúa el progreso en el aprendizaje de los lenguajes de programación ScratchJr ([Bers, 2019](#)). Esta evaluación se lleva a cabo uno a uno haciéndole al niño preguntas interactivas y de programación abierta. La evaluación examina las cinco etapas de programación (emergente, codificación y decodificación, fluidez, nuevos conocimientos y determinación) que atraviesan los niños al participar en el plan de estudios CAL. CSA se administra como un juego antes y después del plan de estudios de CAL y tarda menos de 45 minutos en completarse.

En segundo lugar, [TechCheck \(Relkin, de Ruiter, Bers, 2020\)](#) es una evaluación "desconectada" del pensamiento computacional (TC) que presenta a los niños desafíos análogos a los que surgen en el curso de la programación informática, pero que no requiere experiencia en programación para completar. La evaluación investiga seis dominios de CT descritos por [Bers \(2018\)](#) como apropiados para el desarrollo de los niños pequeños (algoritmos, modularidad, hardware / software, depuración, representación y estructuras de control). El formato es de opción múltiple y la evaluación se puede administrar individualmente o en grupos en 20 minutos o menos. TechCheck se administra antes y después del plan de estudios CAL.

En tercer lugar, la [lista de verificación de PTD](#) evalúa seis comportamientos positivos ("seis C") respaldados por programas como CAL que implementan tecnologías educativas. Estas son comunicación, colaboración, construcción de comunidad, creación de contenido, creatividad y elección de conducta ([Bers, 2012; Bers, 2020](#)). La lista de verificación de PTD se divide en seis

secciones, cada una de las cuales representa un comportamiento descrito en el marco de PTD. Se puntúa en una escala Likert de 5 puntos basada en aproximadamente 20-30 minutos de observación. La lista de verificación de PTD se utiliza en varios puntos a lo largo del plan de estudios de CAL para evaluar hasta qué punto los entornos de aprendizaje / facilitadores y los comportamientos de los niños están contribuyendo al desarrollo tecnológico positivo.

Además, se pueden recopilar evaluaciones de matemáticas y alfabetización, así como otras evaluaciones estándar.

Lecciones

Lección 1: Hablando en código

Ideas poderosas de la informática	Representación
Ideas poderosas de alfabetización	Herramientas de comunicación y lenguaje
Desarrollo tecnológico positivo (PTD)	Comunicación
Paleta de virtudes	Mente abierta
Los niños podrán	<ul style="list-style-type: none"> Definir el lenguaje como método estructurado de comunicación. Identificar los lenguajes de programación como lenguajes utilizados para comunicarse con las computadoras.
Vocabulario	<ul style="list-style-type: none"> Lenguajes de programación: lenguaje utilizado por las computadoras. Código: texto escrito en un lenguaje de programación para decirle a una computadora qué hacer. Idiomas: métodos basados en reglas para compartir información e ideas ScratchJr: lenguaje de programación con imágenes.
Preparación del maestro	<ul style="list-style-type: none"> Leer el plan de la lección Imprima el diario de diseño completo o el diario de diseño de la lección 1 para cada niño. Tenga el video Hola alrededor del mundo abierto y listo para ser reproducido. <p>Otras opciones de video en español:</p> <p>Los saludos del mundo</p> <p>Canción de las Naciones Unidas por la Paz del mundo</p> <ul style="list-style-type: none"> Proyecto del maestro: Cree un proyecto simple con no más de 4 bloques o proyecto de extracción realizado en Desarrollo profesional (PD) (10 minutos).
Introducción <ul style="list-style-type: none"> ¿Cuántos idiomas sabes? <i>(Tiempo sugerido: 5 minutos)</i> <ul style="list-style-type: none"> Pregunte a los niños qué idiomas conocen. Luego, hagan una lluvia de ideas como clase sobre todos los idiomas que hayan escuchado. 	

- Una vez que los niños no puedan pensar en más idiomas, jueguen al [Hola alrededor del mundo](#). Pregúnteles a los niños, ¿había algún idioma nuevo en el que no habíamos pensado?
- Discuta con los niños que hay muchos idiomas, pero todos se usan para lo mismo: ¡compartir pensamientos e ideas!

Iniciando el círculo tecnológico

- **Lenguajes de programación** (*Tiempo sugerido: 5 minutos*)
 - Discuta qué es un **idioma**, por qué necesitamos idiomas y qué sucede cuando las personas no hablan el mismo idioma.
 - Explique que las computadoras también usan idiomas.
 - Las computadoras usan lenguajes llamados **lenguajes de programación**.
 - Si los niños no han oído hablar de lenguajes de programación, explíqueles que diferentes tipos de computadoras como iPhones o Alexa usan diferentes lenguajes.
 - ¿Qué ejemplos de lenguajes de programación han escuchado los niños? (¿C, JAVA, etc.?)

NOTA: Si los niños preguntan acerca de los parlantes inteligentes, etc., explique que con Siri, Alexa, Google, etc., las computadoras en realidad no hablan inglés, pero alguien tradujo el inglés al idioma de la computadora.

Tiempo de ScratchJr

Desafío de estructura:

- **Observar este proyecto de ScratchJr** (*Tiempo sugerido: 10 minutos*)
 - Comparta que conoce un lenguaje de programación llamado "**ScratchJr**".
 - Comparta su proyecto ScratchJr con la clase.
 - *Nota: asegúrese de mostrar el código y no solo el escenario.*
 - Haga a los niños preguntas orientadoras sobre cómo explorar la aplicación y lo que notan en el **código** (por ejemplo, colores, símbolos, palabras).
 - *Señale el área de programación de la pantalla:* ¿Cuántos colores ve aquí? *Toca cada color para mostrar todos los bloques:* ¿Qué crees que significan los colores?
 - *Seleccione un bloque de inicio y un bloque de movimiento:* ¿Por qué algunos bloques tienen una forma diferente a otros?
 - *Seleccione dos bloques de movimiento diferentes:* ¿Tiene alguna suposición sobre lo que significan los símbolos en estos bloques? ¿En qué se diferencian?
 - *Señale el área de preparación:* ¿Qué más ve en otras partes de la pantalla?
 - Explique que la próxima vez los niños aprenderán a programar con ScratchJr.

Tiempo de palabras

- **Diarios de diseño iniciales** (*Tiempo sugerido: 15 minutos*)

Nota: Los diarios de diseño se enviarán a los niños con anticipación, les pediremos a los maestros o padres que los impriman para que los niños los tengan listos para la clase.

- Proporcione diarios de diseño individuales a cada niño y explique que se utilizarán para escribir ideas, dibujar, planificar y recordar cosas. Los programadores utilizan diarios de diseño para realizar un seguimiento de sus proyectos. Cada niño se convertirá en programador y obtendrá su propio diario de diseño.
- Los niños comenzarán escribiendo sus nombres en la portada y luego responderán las indicaciones de la Lección 1:
 - Escribe una lista de todos los **lenguajes humanos** de los que has oído hablar ...
 - Escribe qué idioma o idiomas hablan en casa ...
 - ¿Cuál es el nombre del lenguaje de programación que aprenderemos?

Cerrando el círculo tecnológico

- **Compartir en el diario de diseño** (*Tiempo sugerido: 10 minutos*)
 - Los niños comparten lo que han escrito en sus diarios de diseño.

Lesson 2: El orden importa

Ideas poderosas de la informática	Algoritmos
Ideas poderosas de alfabetización	Secuenciación
Desarrollo tecnológico positivo (PTD)	Comunicación
Paleta de virtudes	Optimismo, curiosidad
Los niños podrán	<ul style="list-style-type: none">● Comparar y contrastar humanos y computadoras en términos de sus características y el lenguaje utilizado para comunicarse con ellos.● Identificar el papel del orden en los algoritmos.● Utilizar los conocimientos previos para escribir un libro de instrucciones.
Vocabulario	<ul style="list-style-type: none">● Orden: la forma en que se configura una lista de cosas.● Algoritmo: instrucciones para que una computadora resuelva un problema en un orden particular.
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección● Imprima una copia por niño del archivo Palabras recortadas y corte cada palabra.● Imprima el Diario de diseño de la lección 2 para cada niño o consulte el Diario de diseño completo.

Introducción

- **¿Tiene sentido la oración?** (*Tiempo sugerido: 5 minutos*)
 - Recuerde a los niños que ScratchJr es un lenguaje de programación: un lenguaje que usamos para hablar con las computadoras. Explique que todos los diferentes tipos de idiomas necesitan que las palabras se pongan en un cierto **orden** para que tengan sentido.
 - Use las [palabras recortadas](#) de la oración ("El gato está en la alfombra"), revuélvalas y lea en voz alta / haga que alguien lea la oración revuelta (por ejemplo, "alfombra la en está gato El")
 - Discuta si esto tiene sentido y pida a los niños que pongan las palabras en el orden correcto.
 - Explique que el lenguaje debe estar en el orden correcto para ser entendido.

Iniciando el círculo tecnológico

- **¿Qué es un algoritmo?** *(Tiempo sugerido: 5 minutos)*
 - Explique que el lenguaje con las computadoras es muy parecido a dar instrucciones a un amigo. Tienes que decirle a la computadora exactamente qué hacer, en el orden correcto, para que funcione. Las instrucciones para computadoras se llaman **algoritmos**.
 - Explica qué es un algoritmo.
 - Una secuencia de pasos en el orden correcto.
 - Reúna ejemplos de actividades que se deben realizar en un orden determinado.
 - Por ejemplo, cepillarse los dientes, ponerse las medias y los zapatos.

Tiempo de ScratchJr

Desafío de estructura:

- **¿Qué está haciendo el gatito?** *(Tiempo sugerido: 10 minutos)*
 - Explique que en ScratchJr, los bloques de programación son algoritmos, y que los personajes de ScratchJr representan la computadora con la que estamos hablando. ¡Nuestros programas le dicen al gato qué hacer!
 - Comparta su proyecto ScratchJr nuevamente con la clase.
 - Haga que los niños miren algunos programas diferentes y adivinen lo que las instrucciones le dicen al gato que haga, luego ejecuten los programas para que los niños puedan ver si sus conjeturas son correctas.
 - Recuerde a los niños que los algoritmos tienen que ver con el orden de los pasos. ¡Elija un voluntario para cambiar el orden de uno de los programas y adivine cómo cambiará lo que hace el gato!

Tiempo desconectado

- **Programa al maestro/a** *(Tiempo sugerido: 10 minutos)*
 - ¡Díganles a los niños que la maestra ahora será la computadora y ellos podrán programarlos! ¡Recuerden que deben decir todos los pasos en el orden correcto!
 - Los niños serán responsables de dirigir verbalmente a su maestra a destinos especiales en el aula (por ejemplo, a una biblioteca o un armario) o hacer una tarea (por ejemplo, hacer un sándwich).
 - Las instrucciones que los niños le den a la maestra deben ser específicas. Por ejemplo, los niños no deben simplemente decir "Adelante". En su lugar, deberían decir: "Avanza ____ pasos". o "Pon la mermelada en el pan" debe ser "Abre el frasco de mermelada y usa tu cuchillo para colocar la mermelada en el pan"
 - Cuando las secuencias de instrucciones no funcionan (tal vez porque la cantidad de pasos que se tomaron fue incorrecta), los niños deben modificar sus instrucciones.
 - Analice lo importante que es ser específico y lo importante que es el orden en la programación.

Tiempo de palabras

- [Cómo diseñar un diario como libro](#) *(Tiempo sugerido: 10 minutos)*
 - En el Diario de diseño de la lección 2, los niños escriben instrucciones para un hermano menor que no sabe cómo cepillarse los dientes. Recuérdeles que deben ser específicos y poner todos los pasos en el orden correcto, ¡como cuando estaban programando al maestro!

Cerrando el círculo tecnológico

- **Compartir en el diario de diseño** *(Tiempo sugerido: 5 minutos)*
 - Pida a uno o dos voluntarios que compartan lo que escribieron en su Diario de diseño.
 - Discuta las diferencias entre los libros de instrucciones, ya que todos se cepillan los dientes de manera un poco diferente.

Oportunidades de diferenciación

- **Aprendizaje virtual**
 - Los niños le dan instrucciones al maestro para hacer algo (por ejemplo, "levante el brazo"), y el maestro hace preguntas de aclaración (por ejemplo, "¿Qué brazo? ¿A qué altura?") Para que los niños sean lo más precisos y ordenados posible.

Lesson 3: ¡Ingrese a ScratchJr!

Ideas poderosas de la informática	Representación, hardware / software
Ideas poderosas de alfabetización	Herramientas de comunicación y lenguaje
Desarrollo tecnológico positivo (PTD)	Creación de contenido, desarrollo comunitario
Paleta de virtudes	Optimismo, curiosidad, equidad
Los niños podrán	<ul style="list-style-type: none"> ● Manejar una tableta de forma segura. ● Abrir la aplicación ScratchJr en una tableta. ● Navegar por los conceptos básicos de la interfaz ScratchJr. ● Guardar un proyecto con un nombre.
Vocabulario	<ul style="list-style-type: none"> ● Programador: alguien que escribe código y crea programas de computadora. ● Vocabulario de interfaz <ul style="list-style-type: none"> ○ Escenario: dónde está el personaje y ocurre la acción en ScratchJr. ○ Secuencia de comandos de programación: bloques ScratchJr encajados juntos. ○ Área de programación: donde el programa está escrito en ScratchJr. ○ Paleta de bloques: dónde están los bloques en ScratchJr. ● Hardware: los objetos que usamos cuando usamos computadoras, como la cpu, tableta, teclado o mouse. ● Software: los programas de computadora que usamos cuando usamos las computadoras, como la aplicación ScratchJr.
Preparación del maestro	<ul style="list-style-type: none"> ● Lea el plan de la lección. ● Lea acerca de las personas que aparecen en la presentación de diapositivas Imágenes de programadores. Ábralo antes de que comience la clase. ● Memorice la letra de la canción de limpieza. Abre el enlace y prepárate para jugarlo al final de la clase. ● Familiarícese con la Guía de interfaz de ScratchJr. Ábrelo en otra pestaña y prepárate para proyectarlo.

Introducción

- **¿Quiénes son estas personas?** (Tiempo sugerido: 5 minutos)
 - ¡Ilustre que los **programadores** pueden ser de diferentes orígenes (género, raza, etnia, edad) con imágenes (incluido uno de usted)!
 - Pregunte a los niños: "¿Qué crees que tienen estas personas en común?"
 - Respuesta: ¡Todos son programadores!
 - Explique que hoy, ¡todos serán programadores!

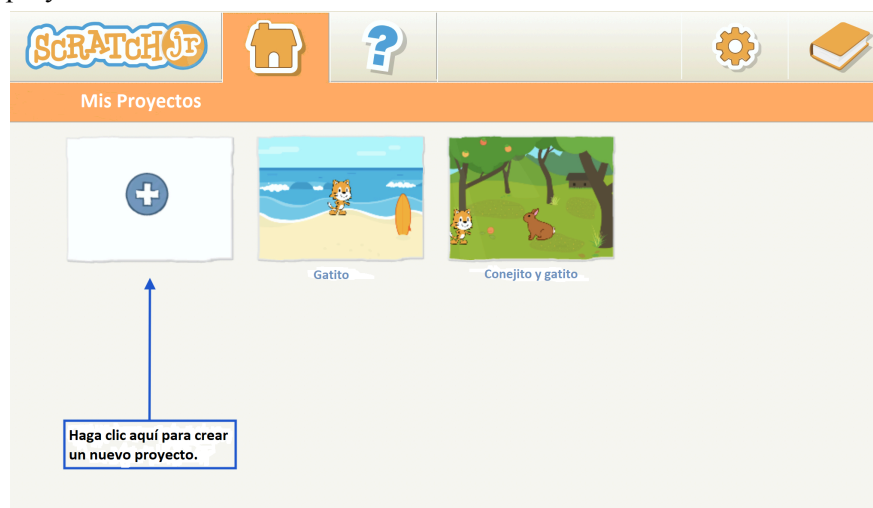
Iniciando el círculo tecnológico

- **Inicio de ScratchJr** (Tiempo sugerido: 10 minutos)
 - Presente la tableta y explique cómo manejarla de manera segura con las dos manos.
 - Muestre a los niños cómo abrir la aplicación ScratchJr.
 - Muestre a los niños cómo crear un nuevo proyecto. Para comenzar, proyecte una tableta en una pantalla para que todos los niños puedan ver la siguiente introducción a la interfaz de ScratchJr.

1. Abra la aplicación ScratchJr.



2. Haga clic en el signo más para abrir un nuevo proyecto en "Mis proyectos".



3. Muestre a los niños el **escenario** (donde ocurre el programa) y el **área de programación** (donde el programa está escrito en orden).



Tiempo de ScratchJr

Desafío de estructura:

- **Explore ScratchJr** (*Tiempo sugerido: 10 minutos*)
 - Explore la plataforma con los niños mientras descubren partes de ScratchJr e introducen vocabulario importante (escenario, área de programación, guión de programación, etc.). Utilice la [Guía de interfaz de ScratchJr](#) a continuación para señalar las características clave de la interfaz.



- **Guardar proyectos** (*Tiempo sugerido: 5 minutos*)
 - Muestre a los niños cómo guardar un proyecto con un nombre siguiendo los pasos a continuación:
 1. Abra el proyecto que desea compartir.

2. Toque el rectángulo amarillo en la esquina superior derecha de la pantalla para ver la pantalla de información del proyecto.



3. Escribe un nombre específico para este proyecto (por ejemplo, "Fiesta de baile"). Si los niños aún están desarrollando sus habilidades de mecanografía, puede invitarlos a escribir algo simple como su propio nombre, y puede cambiar el nombre del proyecto cuando lo reciba.



Explorando expresiones:

- **Juego libre** (*Tiempo sugerido: 10 minutos*)
 - Los niños juegan y exploran más la interfaz de ScratchJr.
 - Los niños practican guardar su proyecto con su nombre.

Cerrando el círculo tecnológico

- [Canción de la limpieza](#) (*Tiempo sugerido: 5 minutos*)

- Presente la canción de limpieza de tabletas y explique cómo almacenar tabletas de manera segura en el aula o virtualmente.

(con la melodía de Itsy Bitsy Araña)
Apagamos las tabletas y las guardamos
Nos divertimos mucho, pero el día se acabó.
usamos ambas manos y caminamos sin correr
Guardamos las tabletas y la canción se terminó

Oportunidades de diferenciación

- **Aprendizaje virtual**

- Es posible que los niños no puedan guardar su tableta y permanecer frente a la cámara. Si se quedan en su asiento, se recomienda que los niños aplaudan para redirigir la atención a la lección.

Lesson 4: Tu primer programa

Ideas poderosas de la informática	Representación
Ideas poderosas de alfabetización	Correspondencia entre el alfabeto y el sonido de las letras
Desarrollo tecnológico positivo (PTD)	Creación de contenido, comunicación
Paleta de virtudes	Mente abierta, curiosidad
Los niños podrán	<ul style="list-style-type: none"> ● Definir símbolos como imágenes que representan o significan algo. ● Usar una tableta con pantalla táctil para tocar, arrastrar y soltar bloques de manera efectiva. ● Identificar los bloques de inicio, fin y movimiento en ScratchJr. ● Ordenar los bloques de inicio, fin y movimiento para crear programas en ScratchJr ● Crear un nuevo personaje en ScratchJr
Vocabulario	<ul style="list-style-type: none"> ● Símbolo: elementos o imágenes que tienen un significado. ● Bloque de programación: una sola instrucción para un carácter ScratchJr. ● Personaje: una persona o un animal en una historia.
Preparación del maestro	<ul style="list-style-type: none"> ● Leer el plan de la lección ● Imprima o tenga el documento Imágenes de letreros listo para proyectar. ● Imprima o tenga el documento de búsqueda del tesoro listo para proyectar. ● Imprima las hojas con los bloques ScratchJr y recorte el título de cada página. ● Imprima la Verificación de comprensión de la Lección 4, para cada niño o abra las diapositivas de Verificación de la comprensión de la lección 4.
Introducción <ul style="list-style-type: none"> ● Búsqueda del tesoro de símbolos (<i>Tiempo sugerido: 5 minutos</i>) <ul style="list-style-type: none"> ○ Muestre las imagenes de letreros <ul style="list-style-type: none"> ■ Pregunte a los niños si saben lo que significa cada signo. ■ ¿Cómo lo saben? 	

- Muestre una foto o dos de una calle y realice una [búsqueda del tesoro](#) en busca de **símbolos**.
 - Pida a los niños que miren alrededor de la imagen y busquen otros letreros en la calle.

Iniciando el círculo tecnológico

- **Símbolos de ScratchJr** (*Tiempo sugerido: 10 minutos*)
 - Pregúntele a los niños “¿Cómo sabes qué significan los bloques ScratchJr? ¿Qué pistas puedes encontrar sobre lo que hacen? ”
 - Formas: *levante un bloque de inicio, un bloque de acción y un bloque de finalización*. Discuta las formas de los bloques y pregunte si las formas significan algo.
 - Colores: discuta el significado de los colores del bloque. Explique que cada color representa un cierto tipo de bloque (por ejemplo, azul = movimiento, verde = sonido, rosa = apariencia).
 - Imágenes: ¿Qué nos dicen las imágenes?
 - Discuta la importancia de la bandera verde y el bloque final. Estos bloques tienen propósitos específicos dentro del lenguaje de programación. Los bloques de inicio son como letras mayúsculas al principio de una oración y los bloques de finalización son como puntos. ¡Todos los bloques en el medio son las palabras de tu oración!

Tiempo desconectado

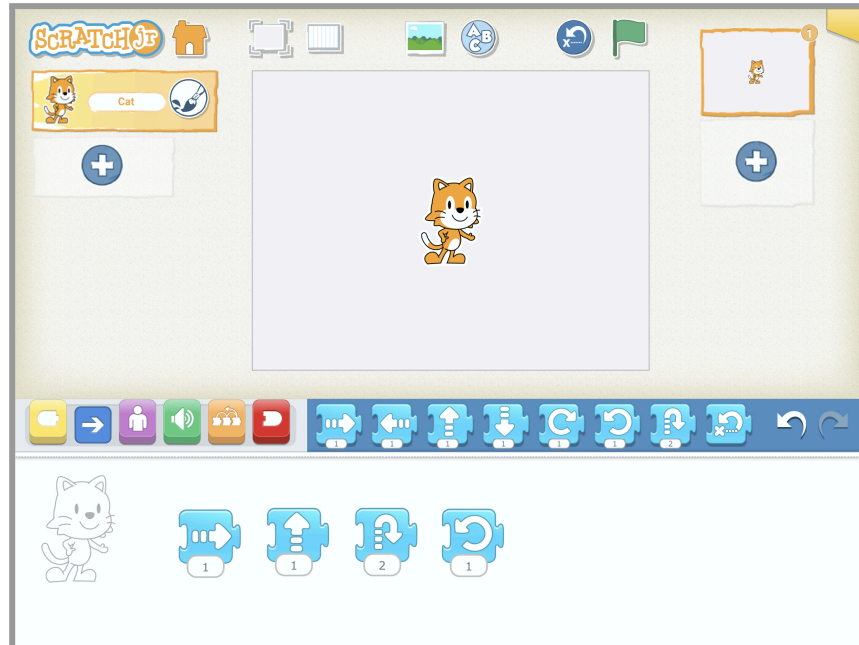
- **El programador dice** (*Tiempo sugerido: 5 minutos*)
 - Esta actividad se juega como el juego tradicional “Simón dice”, en el que los niños repiten una acción si Simón dice que hagan algo. Utilice los [recortes de bloques de ScratchJr](#) para que esta actividad se centre en las instrucciones paso a paso y los asuntos relacionados con el pedido.
 - Presente brevemente cada instrucción de programación y lo que significa (use solo los bloques numerados en la sección Preparación del maestro para esta lección).
 - Haga que la clase se ponga de pie. Sostenga un gran bloque ScratchJr recortado a la vez y diga: "El programador dice _____". Cada instrucción debe incluirse más de una vez.
 - Ampliación opcional 1: los niños se turnan como “programadores” y dan instrucciones a sus compañeros.
 - Expansión opcional 2: Incluya bloques de inicio y finalización para cada instrucción, por lo que cada instrucción será "El programador dice _____", con un programa completo.

Tiempo de ScratchJr

Desafío de estructura:

¿Cómo hacer un programa? (*Tiempo sugerido: 10 minutos*)

- Muestre bloques de movimiento simples a la clase e introduzca cómo hacer un programa en ScratchJr siguiendo los pasos a continuación:
 1. Arrastra bloques al área de programación y tócalos para mostrar cómo se mueve el gatito. Algunos bloques de movimiento importantes son izquierda, derecha, arriba, abajo, rotar a la izquierda / derecha y saltar.

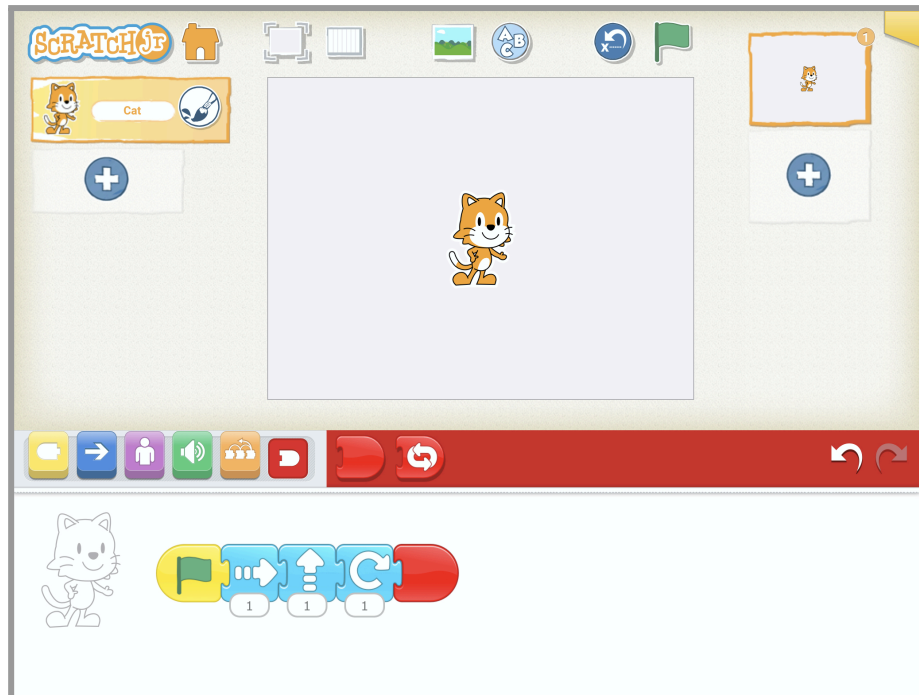


2. Muestre cómo unir bloques de programación para hacer una secuencia de movimientos.



3. Demuestre cómo crear un programa usando los bloques Inicio y Fin que se ejecuta haciendo clic en la Bandera Verde. Tenga en cuenta que los bloques Inicio y Fin son necesarios cuando se usa el modo de

presentación o cuando se ejecuta un programa con varias partes (para que comiencen simultáneamente).



A continuación, muestre a los niños cómo agregar un nuevo personaje siguiendo los pasos a continuación:

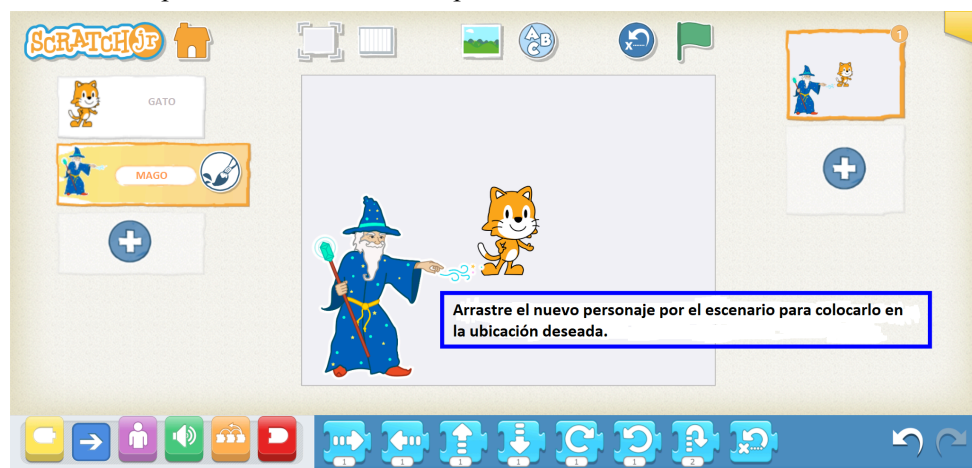
1. Toque el signo más en el lado izquierdo de la pantalla.



2. Seleccione el carácter deseado y toque la marca de verificación para continuar.



3. Arrástrelo por el área del escenario para colocarlo en la ubicación deseada.



- Para eliminar un personaje, presione y mantenga el dedo en el cuadro con el nombre del personaje. Aparecerá una "x" roja en la parte superior del cuadro de caracteres.



Verificar la comprensión de la Lección 4: antes de comenzar un proyecto, verifique que sus alumnos comprendan los nuevos conceptos que acaban de aprender. Lea cada pregunta a los niños y pídeles que respondan con un pulgar hacia arriba para decir "sí" o un pulgar hacia abajo para "no". Deténgase y vuelva a explicar los conceptos según sea necesario.

Explorando expresiones:

- **¡Danza del gato! (Tiempo sugerido: 10 minutos)**
 - ¡Dícales a los niños que usarán todos los bloques de movimiento para programar el baile del gato!
 - Los niños crearán un programa utilizando bloques aprendidos hasta ahora (Bloque de inicio, Bloques de movimiento, Bloque de finalización).

Cerrando el círculo tecnológico

- **Aprendiendo el baile del gato (Tiempo sugerido: 5 minutos)**
 - Solicite voluntarios para compartir sus proyectos.
 - Haga que los otros niños hagan cumplidos sobre el proyecto que se está compartiendo.
 - Discuta los procedimientos para compartir (por ejemplo, escuchar mientras otros están presentando).
 - Cuando comparta proyectos, recuerde a los niños que muestren a la clase su código y su trabajo.

Oportunidades de diferenciación

- **Desafío extra**
 - Para los niños que necesitan más desafíos, bríndeles las siguientes indicaciones:
 - Agrega otro personaje a la fiesta de baile del gato.

- ¿Puedes hacer que hagan el mismo baile?
 - ¿Puedes hacer que hagan diferentes bailes?
- **Aprendizaje virtual**
 - Para compartir proyectos, haga que los niños sostengan la tableta frente a la pantalla.

Lesson 5: Programa la canción Hokey-Pokey

Ideas poderosas de la informática	Algoritmos, proceso de diseño
Ideas poderosas de alfabetización	Secuenciación, conciencia fonológica
PTD	Creación de contenido, creatividad
Paleta de virtudes	Paciencia, mentalidad abierta
Los niños podrán	<ul style="list-style-type: none">• Usar la cuadrícula en ScratchJr• Hacer una lluvia de ideas y planificar un programa• Crear un programa en ScratchJr para acompañar un baile• Cambiar parámetros en ScratchJr
Vocabulario	<ul style="list-style-type: none">• Cuadrícula: líneas que se cruzan entre sí para formar muchos cuadrados.• Lluvia de ideas: discusión en grupo para tener una idea• Parámetro: una instrucción numérica que le dice a ScratchJr cuántas veces debe hacer algo
Preparación del maestro	<ul style="list-style-type: none">• Leer el plan de la lección.• Familiarízate con la letra del Hokey-Pokey y ten el video abierto y listo para reproducir durante la clase.• Tenga la presentación de diapositivas Hokey-Pokey abierta y lista para proyectar o imprimir.• Imprima recortes de bloques de ScratchJr y recorte los símbolos.

Introducción

- **Baila el Hokey-Pokey** (*Tiempo sugerido: 5 minutos*)
 - Explique a los niños que hoy crearán un programa en ScratchJr para hacer el Hokey-Pokey. Cante y baile el [Hokey-Pokey](#) como clase para asegurarse de que todos lo conozcan y lo recuerden.

Tiempo de palabras

- **Leer en voz alta el Hokey-Pokey** (*Tiempo sugerido: 5 minutos*)
 - El maestro lee en voz alta el [gráfico escrito de Hokey Pokey](#) e introduce la parte de ScratchJr al final.

[La mano derecha aquí.](#)

la mano derecha allá,
la mano derecha aquí sacudiéndola muy bien.
Harás el hocki-pocki
saltando en tu lugar y así lo sabrás bailar.

2) mano izquierda
3) pie derecho
4) pie izquierdo
5) cabeza
6) nariz
7) todo el cuerpo

¡Ahora vamos a bailar con nuestro gatito!

Con tu gatito aquí,
con tu gatito allá,
con tu gatito aquí sacudiéndolo muy bien.
Harás el hocki-pocki
saltando en su lugar y así lo sabrá bailar.

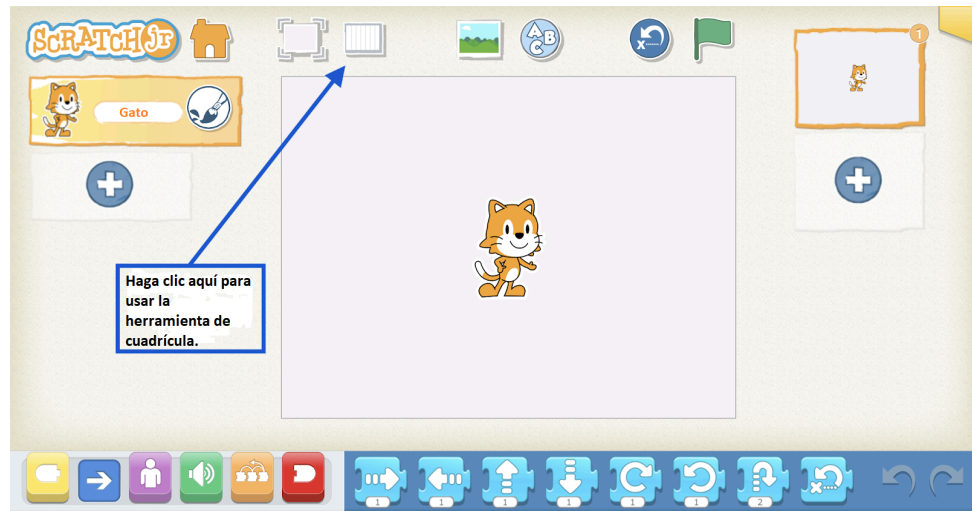
Iniciando el círculo tecnológico

- **Hokey-Pokey en ScratchJr** (*Tiempo sugerido: 5 minutos*)
 - Señale el Hokey-Pokey escrito para guiar su discusión con los niños.
 - Analice cómo puede traducir los movimientos en bloques ScratchJr.
 - Muestre los [recortes de bloques de ScratchJr](#) para ayudar a pensar en los diferentes movimientos que los niños podrían usar para programar el Hokey-Pokey.
 - Pregúnteles a los niños qué podrían hacer con la línea “haz el Hokey-Pokey”, ¿podría ser lo que quieran!

Tiempo de ScratchJr

Desafío de estructura:

- **La herramienta Cuadrícula** (*Tiempo sugerido: 5 minutos*)
 - ScratchJr tiene muchas herramientas diferentes para ayudar a crear y modificar sus proyectos. Presente la herramienta de **cuadrícula** para guiar a los niños con la cantidad de pasos de ScratchJr que da el gato mientras baila el Hokey-Pokey.

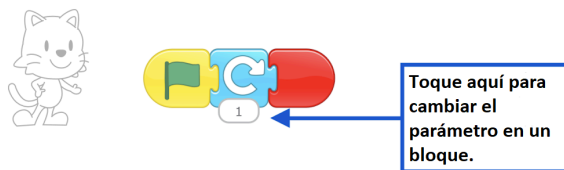
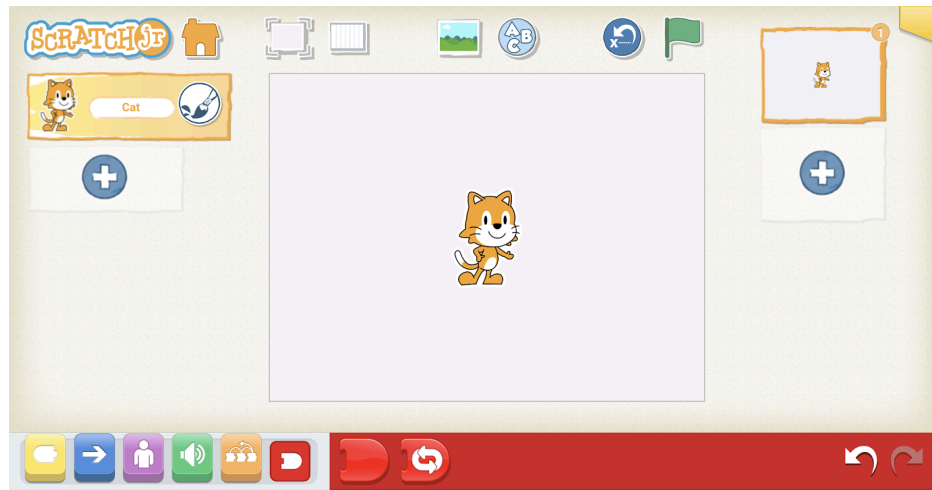


○

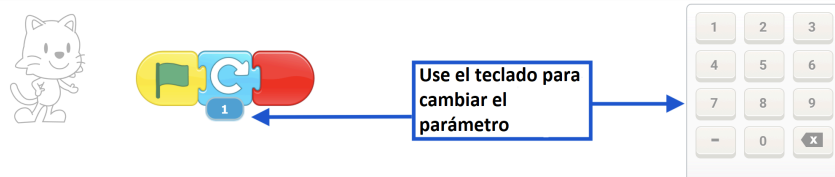
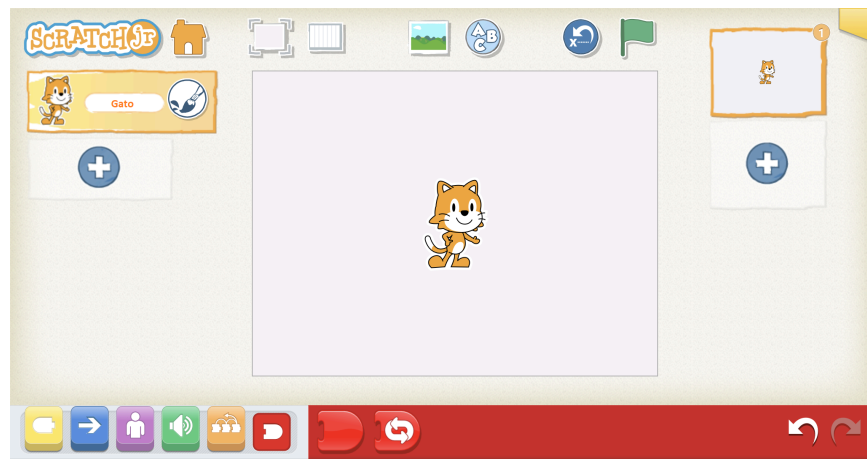


- Para eliminar la herramienta de cuadrícula, haga clic en el mismo icono nuevamente
- ¿Qué es un parámetro? (Tiempo sugerido: 10 minutos)
 - Explica qué es un **parámetro** ...
 - Le dice a una computadora cuántas veces debe realizar esa acción
 - Analice por qué los parámetros son útiles ...

- Hacen que nuestro código sea más eficiente (por ejemplo, en lugar de tener 12 bloques de turno, podemos simplemente cambiar el parámetro en un bloque de turno al número 12)
- Explique cómo funciona en ScratchJr siguiendo los pasos a continuación:
 - Seleccione el bloque en el que desea cambiar el parámetro. Toque la burbuja con un número para usar el teclado para cambiar el número.



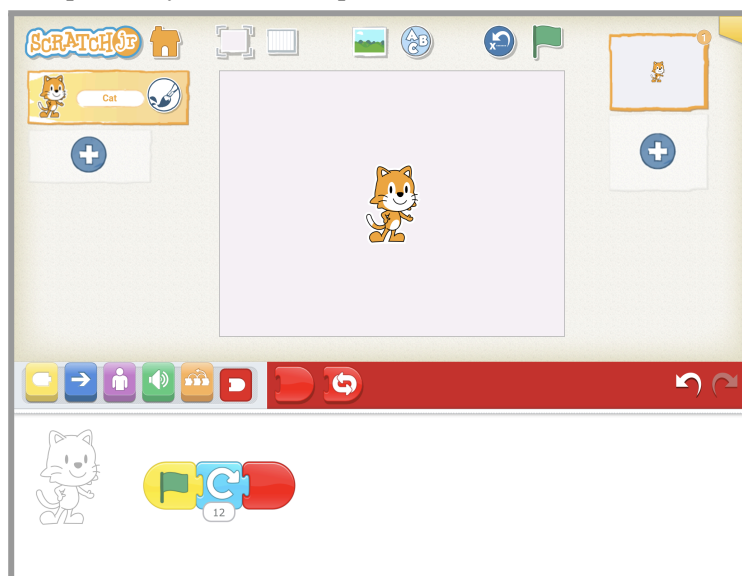
- El teclado aparecerá en el lado derecho de la pantalla.



- Utilice el teclado para cambiar el parámetro (número máximo de dos dígitos).



- Una vez que haya ingresado el nuevo parámetro, toque en cualquier lugar de la pantalla y la clave desaparecerá.



- ¡Ahora devuélvemelo al Hokey-Pokey! Pregúntele a los niños: ¿Cómo podría mostrar la parte de la canción que lo agitan o da la vuelta?

Explorando expresiones:

- **Programando el Hokey-Pokey** (*Tiempo sugerido: 10 minutos*)
 - Saque las tabletas y recuérdelas a los niños las reglas o procedimientos. Dígales a los niños que vamos a programar al gato en ScratchJr para hacer el Hokey-Pokey. Haga que varios niños compartan sus estrategias de programación. Los niños programarán al gato para que haga el Hokey-Pokey. Dígales a los niños que también se pondrán a trabajar en sus programas en la próxima clase, así que está bien si no terminan.

Cerrando el círculo tecnológico

Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) by the [DevTech Research Group](#) is licensed under a [Creative Commons Attribution NonCommercial-ShareAlike 3.0 Unported License](#). Under this license, you may use and adapt this work, but you must attribute the work to the DevTech Research Group. You **may not** use or adapt this work for commercial purposes.

This work is funded by the US Dept of Education (grant #U411C190006) and by the Scratch Foundation.

© 2020, DevTech Research Group.

- **Compartir creaciones** (*Tiempo sugerido: 5 minutos*)
 - Los niños comparten cómo representaron el “batido” en su programa.
 - Comente que hay varias formas de expresar lo mismo.

Oportunidades de diferenciación

- **Actividades adicionales**
 - Juegue un juego corto en el que los niños puedan explorar cómo funciona la herramienta de cuadrícula. Haga que los niños programen a su gato para que se mueva 1 espacio, luego 2 espacios y así sucesivamente ...

Lesson 6: Re-programa la canción Hokey-Pokey

Ideas poderosas de la informática	Algoritmos, proceso de diseño
Ideas poderosas de alfabetización	Secuenciación, conciencia fonológica, edición y conciencia de la audiencia
PTD	Creación de contenido
Paleta de virtudes	Perseverancia, mentalidad abierta
Los niños podrán	<ul style="list-style-type: none"> ● Utilizar el bloque Repetir para siempre en ScratchJr ● Utilizar el bloque Pop en ScratchJr
Vocabulario	<ul style="list-style-type: none"> ● Revisar: volver a visitar y realizar cambios en algo.
Preparación del maestro	<ul style="list-style-type: none"> ● Leer el plan de la lección. ● Imprima el gráfico de anclaje de Hokey-Pokey o ábralo para que esté listo para proyectar al comienzo de la clase. ● Tenga listo el video de Hokey-Pokey antes de la clase. ● Vea proyectos de ejemplo para familiarizarse con la actividad del proyecto.

Introducción

- **Baila el Hokey-Pokey** (*Tiempo sugerido: 5 minutos*)
 - Canten y bailen el Hokey-Pokey nuevamente como clase para asegurarse de que todos lo conozcan y lo recuerden.

*La mano derecha aquí,
 la mano derecha allá,
 la mano derecha aquí sacudiéndola muy bien.
 Harás el hocki-pocki
 saltando en tu lugar y así lo sabrás bailar.*

*2) mano izquierda
 3) pie derecho
 4) pie izquierdo
 5) cabeza
 6) nariz
 7) todo el cuerpo*

¡Ahora vamos a bailar con nuestro gatito!

*Con tu gatito aquí,
con tu gatito allá,
con tu gatito aquí sacudiéndolo muy bien.
Harás el hocki-pocki
saltando en su lugar y así lo sabrá bailar.*

Iniciando el círculo tecnológico

- **Agregando al Hokey-Pokey** (*Tiempo sugerido: 5 minutos*)
 - Explique a los niños que **revisarán** sus programas Hokey-Pokey agregando más bloques y terminando sus programas.

Tiempo de ScratchJr

Desafío de estructura:

- **Introducir el bloque Pop** (*Tiempo sugerido: 5 minutos*)
 - Dígales a los niños que este nuevo bloque podría ser divertido para agregar cuando el gato "hace el Hokey-Pokey". Demuestre que al agregar el bloque Pop en el código de un personaje, se producirá un ruido "pop". Si no escuchas un sonido de "pop" cuando ejecutas el código de tu personaje, ¡verifica el volumen en tu dispositivo!



- **Introducir el bloque Repetir por siempre** (*Tiempo sugerido: 5 minutos*)
 - ¡Dígales a los niños que pueden usar un bloque nuevo para hacer que el gato baile el Hokey-Pokey por siempre! Demuestre que al agregar el bloque Repetir para siempre al final del código de un personaje, ese personaje ejecutará el código continuamente.



Explorando expresiones:

- **Re-programa el Hokey-Pokey** (*Tiempo sugerido: 15 minutos*)
 - Saque las tabletas y recuérdelas a los niños las reglas o procedimientos.

- Los niños pueden seguir programando su Hokey-Pokey. Anime a los niños a incorporar los bloques Pop y Repetir por siempre en sus programas.

Cerrando el círculo tecnológico

- **Compartir creaciones** (*Tiempo sugerido: 10 minutos*)
 - Los niños comparten sus proyectos revisados en grupos pequeños.
 - Los niños explican cómo cambiaron o mejoraron sus programas.
 - Revise los procedimientos para compartir.
 - Por ejemplo, escuchar mientras otros están presentando.

Oportunidades de diferenciación

- **Desafío extra**
 - Para los niños que necesitan más desafíos o andamios para practicar parámetros, bríndeles las siguientes indicaciones:
 - Si cantamos el Hokey-Pokey muy lentamente, ¿cuántos pasos necesitarías que hiciera el gato?
 - Si cantamos el Hokey-Pokey muy rápido, ¿cuántos pasos necesitarías que hiciera el gato?

Lección 7: Piensa como un creador

Ideas poderosas de la informática	Algoritmos, proceso de diseño
Ideas poderosas de alfabetización	Proceso de escritura, secuenciación
PTD	Creación de contenido, creatividad
Paleta de virtudes	Paciencia, perseverancia
Los niños podrán	<ul style="list-style-type: none">● Identificar los pasos del proceso de diseño.● Aplicar los pasos del proceso de diseño a un desafío de ingeniería.
Vocabulario	<ul style="list-style-type: none">● Proceso de diseño:<ul style="list-style-type: none">○ Preguntar: para usar una pregunta○ Imagina: hacer una idea en tu mente○ Planificar: decidir qué hacer y cómo hacerlo.○ Crear: hacer algo○ Prueba: para ver si funciona tu idea y cómo funciona○ Mejorar: hacer algo mejor○ Compartir: para mostrarles a los demás lo que hiciste.● Ciclo: algo que se repite.
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección.● Tenga una pestaña abierta con la presentación de diapositivas del gráfico de anclaje para el proceso de diseño lista para ser proyectada● Reúna materiales reciclados para hacer manualidades en el aula o en casa..

Introducción

- **Canción del proceso de diseño** (*Tiempo sugerido: 5 minutos*)
 - Presente que hoy los niños aprenderán sobre el **proceso de diseño**, pero primero aprenderán la canción del proceso de diseño. Muestre el [gráfico de anclaje del proceso de diseño](#) mientras reproduce la canción para que los niños la escuchen. ¡Toque la canción Design Process por segunda vez e invite a los niños a cantar con ellos!

(canta con la melodía de "estrellita, estrellita")

Pregunto e imagino, planeo y creo.

Pruebo y mejoro y comparto lo que hacemos.

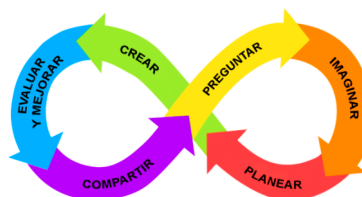
(Repetir)

Iniciando el círculo tecnológico

- **El proceso de diseño** (*Tiempo sugerido: 10 minutos*)
 - Explique el proceso de diseño utilizando el [gráfico de anclaje del proceso de diseño](#).

Proceso de diseño

Al realizar proyectos, los ingenieros siguen una serie de pasos denominados proyectos de diseño. Se trata de 6 pasos: PREGUNTAR, IMAGINAR, PLANIFICAR, CREAR, PROBAR Y MEJORAR y COMPARTIR. El diseño de proyectos es un ciclo, no hay un punto oficial de inicio o final. puede comenzar en cualquier paso, retroceder y avanzar entre los pasos y, o repetir el ciclo una y otra vez.



Proceso de diseño de la canción

(Con la melodía de "Twinkle, Twinkle")

Pregunte e imagine, planifique y cree,
Pruebe y mejore y comparta lo que hacemos.
(Repetir)

Plan de estudios de jardín de infantes de CAL ScratchJr
Lección 9: Piense como un Creador
Calentamiento: Canción del proceso de diseño
Apertura del círculo tecnológico: introduzca el proceso de diseño

- Repase el proceso de diseño del proyecto Osito de peluche de la última clase.
 - **Pregunte:** ¿Cómo podemos programar este baile en ScratchJr?
 - **Imagínese:** Lluvia de ideas de movimientos.
 - **Plan:** Decidimos los parámetros que queremos usar y etiquetamos a nuestro personaje.
 - **Crear:** Crea el personaje y el programa.
 - **Probar y mejorar:** ¿Fueron correctos los parámetros? ¿Necesitamos hacer nuestro personaje más grande?
 - **Compartir:** ¡Compartimos al final de la clase!

Tiempo desconectado

- **Tu propio proceso de diseño** (*Tiempo sugerido: 25 minutos*)
 - Explique que el proceso de diseño no es solo para ScratchJr, ¡es para hacer cualquier cosa! ¡Hoy van a poder elegir lo que diseñan usando el Proceso de Diseño!
 - Pregunte: Dé a los niños una pauta de cómo hacer algo. *Esta puede ser una buena oportunidad para conectarse con otros temas del aula / temas que tienen oportunidades para la artesanía (ejemplo: si el aula está estudiando mariposas, haga la pregunta: ¿Cómo podríamos hacer una mariposa con materiales reciclados?)*
 - Imagínese: Divídanse en grupos pequeños para intercambiar ideas para sus proyectos.
 - Planificar: los niños piensan en ideas con su grupo pequeño.
 - Crear: ¡Dé tiempo a los niños para que utilicen manualidades y materiales reciclados para hacer su creación!

- Pruebe y mejore: haga una pausa y pregunte a los niños si sus creaciones siguen su plan. ¿Hay algún cambio que quieran hacer ahora que empezaron a hacer su proyecto?
- Compartir: recorra y comparta creaciones en clase.

Cerrando el círculo tecnológico

- **¡Es un ciclo!** (*Tiempo sugerido: 5 minutos*)
 - Pregunte a los niños si ver los proyectos de todos y trabajar en sus propios proyectos les dio alguna idea de cosas nuevas que quieren hacer.
 - Recuerde a los niños que el proceso de diseño es un **ciclo**. ¡Compartir solo conduce a más preguntas y más creación y luego a más compartir!

Oportunidades de diferenciación

- Aprendizaje virtual: Informe a los padres con anticipación que los niños necesitarán materiales para manualidades para esta lección.

Lesson 8: Meterse en el personaje

Ideas poderosas de la informática	Proceso de diseño
Ideas poderosas de alfabetización	Proceso de escritura
PTD	Creación de contenido, creatividad
Paleta de virtudes	Mente abierta
Los niños podrán	<ul style="list-style-type: none"> ● Identificar similitudes entre escritura y programación. ● Identificar a los personajes de una historia. ● Diseñar un nuevo personaje en ScratchJr. ● Usar la herramienta de texto en ScratchJr para agregar texto a un proyecto.
Vocabulario	<ul style="list-style-type: none"> ● Título: nombre de un libro o proyecto ● Autor: una persona que escribe un libro.
Preparación del maestro	<ul style="list-style-type: none"> ● Leer el plan de la lección. ● Tenga una pestaña abierta con la presentación de diapositivas del gráfico de anclaje para el proceso de diseño lista para ser proyectada ● Tenga acceso al libro Ada Lovelace “Pequeña y Grande”, ya sea en papel o digital y listo para ser proyectado.

Introducción

- **Canción del proceso de diseño** (*Tiempo sugerido: 5 minutos*)
 - Vuelva a mostrar el [gráfico de anclaje para el proceso de diseño](#) en la pantalla y cante la canción del proceso de diseño.

[\(canta con la melodía de "estrellita, estrellita"\)](#)

[Pregunto e imagino, planeo y creo.](#)

[Pruebo y mejoro y comparto lo que hacemos.](#)

[\(Repetir\)](#)

Iniciando el círculo tecnológico

- **Proceso de diseño y de escritura** (*Tiempo sugerido: 10 minutos*)
 - Discutir cómo los **autores** atraviesan un proceso similar con su escritura..
 - Muestre a los niños el [proceso de diseño y el cuadro de anclaje del proceso de escritura](#). Explique a los niños que ambos son procesos creativos que requieren

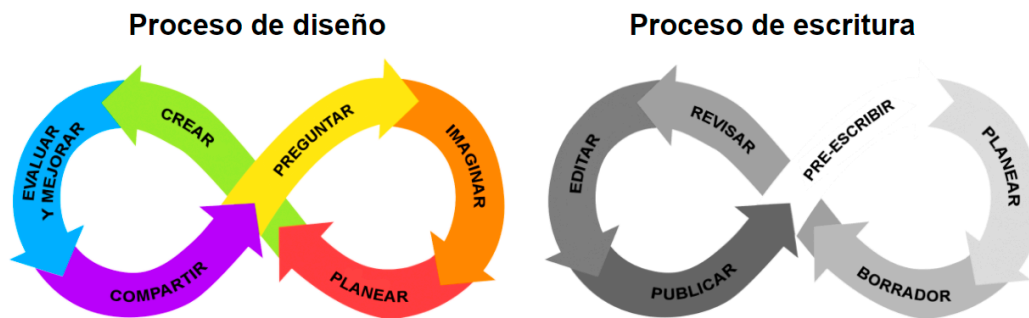
Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) by the [DevTech Research Group](#) is licensed under a [Creative Commons Attribution NonCommercial-ShareAlike 3.0 Unported License](#). Under this license, you may use and adapt this work, but you must attribute the work to the DevTech Research Group. You **may not** use or adapt this work for commercial purposes.

This work is funded by the US Dept of Education (grant #U411C190006) and by the Scratch Foundation.

© 2020, DevTech Research Group.

imaginación, planificación, creación, revisión, retroalimentación e intercambio. Tanto los programadores como los escritores convierten las ideas en proyectos que se comparten con otros. Pregunte a los niños qué otras actividades requieren un proceso (por ejemplo, cocinar, pintar, mejorar en un deporte, etc.).

- Discuta qué es igual y diferente entre programadores y autores. Proceso de escritura: así como los programadores usan el proceso de diseño para diseñar y crear proyectos, los escritores usan el proceso de escritura para generar ideas, escribir un borrador, hacer revisiones y compartir su escritura con otros. El proceso de escritura también es un ciclo: no hay un punto de inicio o finalización oficial, ¡y puedes avanzar y retroceder entre los pasos!



Tiempo de palabras

- Lea “[Ada Lovelace: Pequeña y grande](#)” (Tiempo sugerido: 10 minutos)
 - Haga una pausa en varios puntos durante el libro para verificar la comprensión.

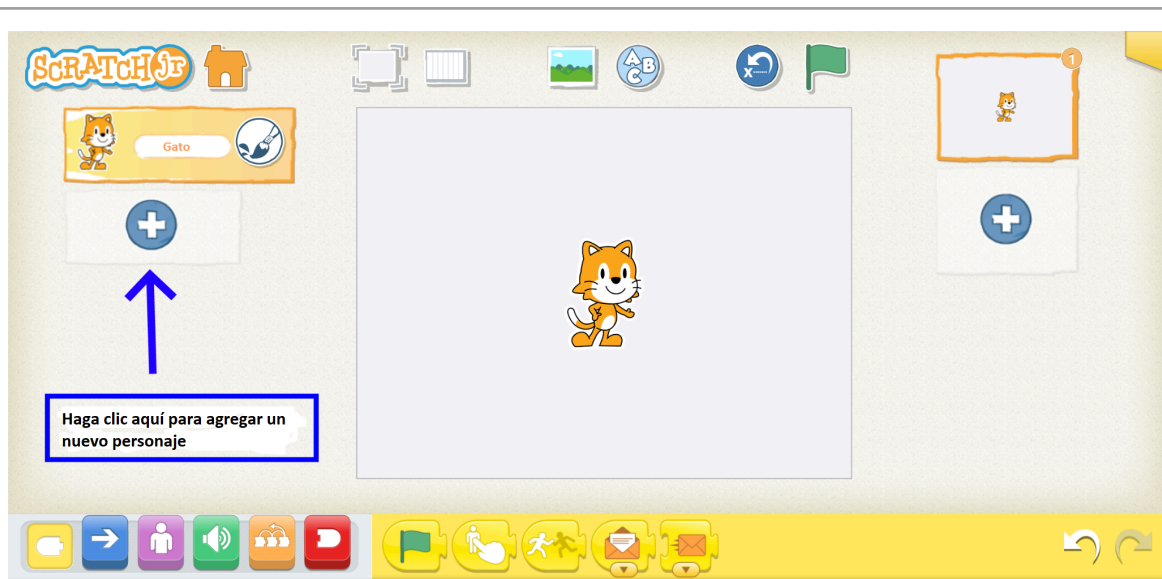
Tiempo de ScratchJr

Desafío de estructura: (Tiempo sugerido: 5 minutos)

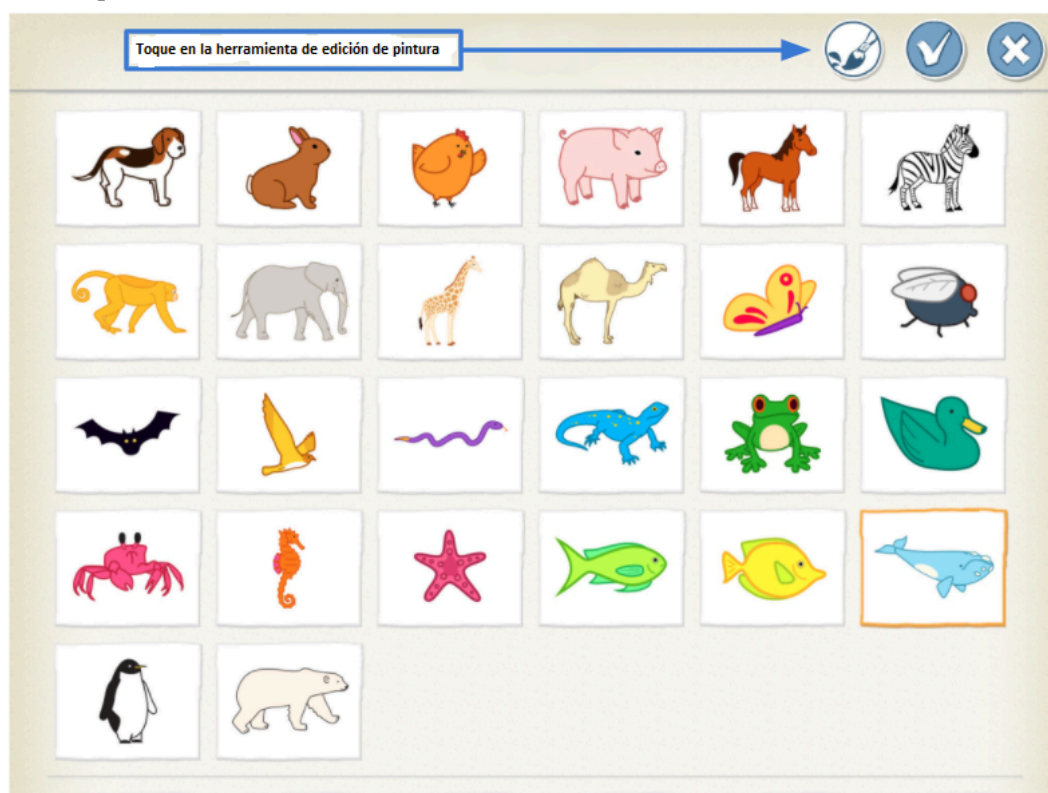
1. La herramienta Paint Editor

¡Dígalas a los niños que ahora harán su propia Ada en ScratchJr! Presente la [Guía del Editor Paint](#), siguiendo los siguientes pasos a continuación:

1. Toque el signo más para agregar un carácter

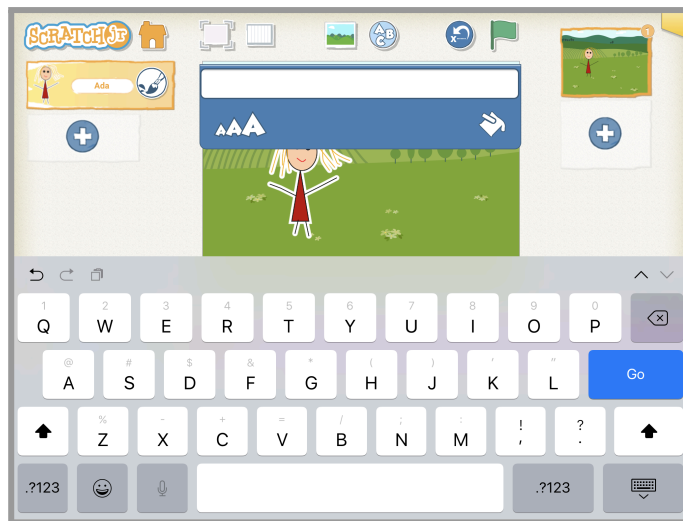


2. Selecciona un personaje para editar o crear uno desde cero usando el Editor de pintura.



3. Con la [guía](#) a continuación, dirija a los niños a través del Editor de pintura.

2. Con el teclado del dispositivo, los niños pueden agregar texto.



3. Cuando terminan, pueden arrastrar el cuadro de texto por la pantalla.



Nota: los caracteres se pueden mover y programar; Los fondos no pueden.

Explorando expresiones:

1. Haz tu propia Ada (Tiempo sugerido: 10 minutos)

- Muestre algunas escenas del cuento para que los niños las utilicen como referencia.
 - Haga preguntas a los niños sobre Ada para recopilar detalles.
 - Por ejemplo, "¿De qué color es el cabello de Ada?"
- Los niños comenzarán a diseñar su propia Ada usando la herramienta Paint Editor.

Cerrando el círculo tecnológico

- **Compartir creaciones** (Tiempo sugerido: 5 minutos)
 - Los niños comparten sus proyectos con la clase.

Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) by the [DevTech Research Group](#) is licensed under a [Creative Commons Attribution NonCommercial-ShareAlike 3.0 Unported License](#). Under this license, you may use and adapt this work, but you must attribute the work to the DevTech Research Group. You **may not** use or adapt this work for commercial purposes.

This work is funded by the US Dept of Education (grant #U411C190006) and by the Scratch Foundation.

© 2020, DevTech Research Group.

Lección 9: Cuenta un cuento

Ideas poderosas de la informática	Algoritmos
Ideas poderosas de alfabetización	Secuenciación
PTD	Construcción de comunidad, colaboración
Paleta de virtudes	Justicia, paciencia
Los niños podrán	<ul style="list-style-type: none">● Recuperar y ordenar escenas de una historia.● Identificar la importancia del orden al contar una historia.● Agregar una nueva página a un proyecto en ScratchJr.● Cambiar el fondo en ScratchJr.
Vocabulario	<ul style="list-style-type: none">● Páginas: donde se escribe o dibuja un libro o una historia.● Fondo: la parte de una imagen que se encuentra detrás del objeto principal.● Escenario: donde tiene lugar una historia.
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección.● Imprima la Verificación de la comprensión de la lección 9 a cada niño o abra las diapositivas de la verificación de la lección 9.

Introducción

- **¡Actuar!** (*Tiempo sugerido: 10 minutos*)
 - Hable sobre 4 eventos diferentes que sucedieron en Ada Lovelace y luego forme grupos de 3-5 niños. Cada grupo representará una escena en el orden en que sucedió en la historia. Deje que cada grupo sepa qué escena tienen y permítales practicar cómo actuarán la escena. Luego, ¡vuelva a reunirse y haga que cada grupo represente su escena! Haga que los grupos hagan sus escenas fuera de orden de la historia.

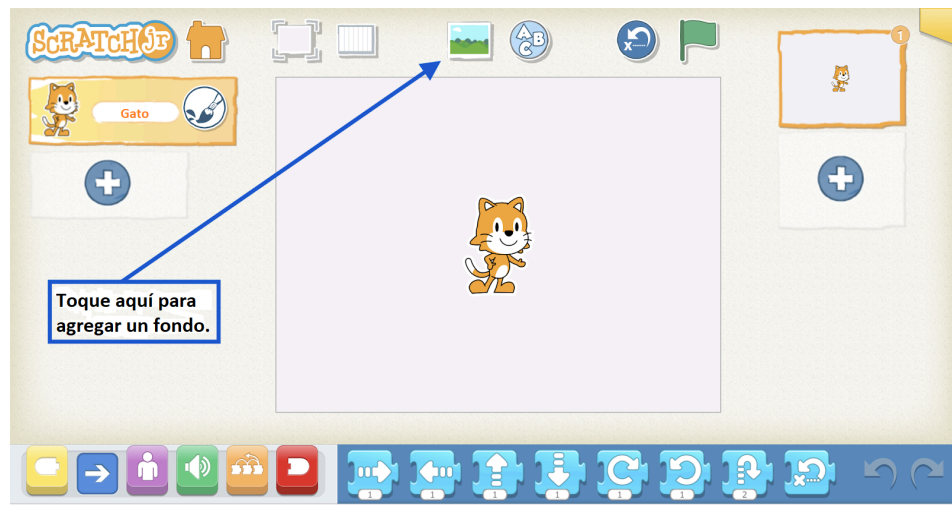
Iniciando el círculo tecnológico

- **Repasando escenas** (*Tiempo sugerido: 5 minutos*)
 - Habla sobre las partes de cada **escena** (los personajes, la acción, etc.).
 - También hubo un **escenario** (o tiempo / lugar) que no vimos mientras actuamos.
 - Todas estas piezas (personajes, escenario, acción) son importantes para contar una historia.

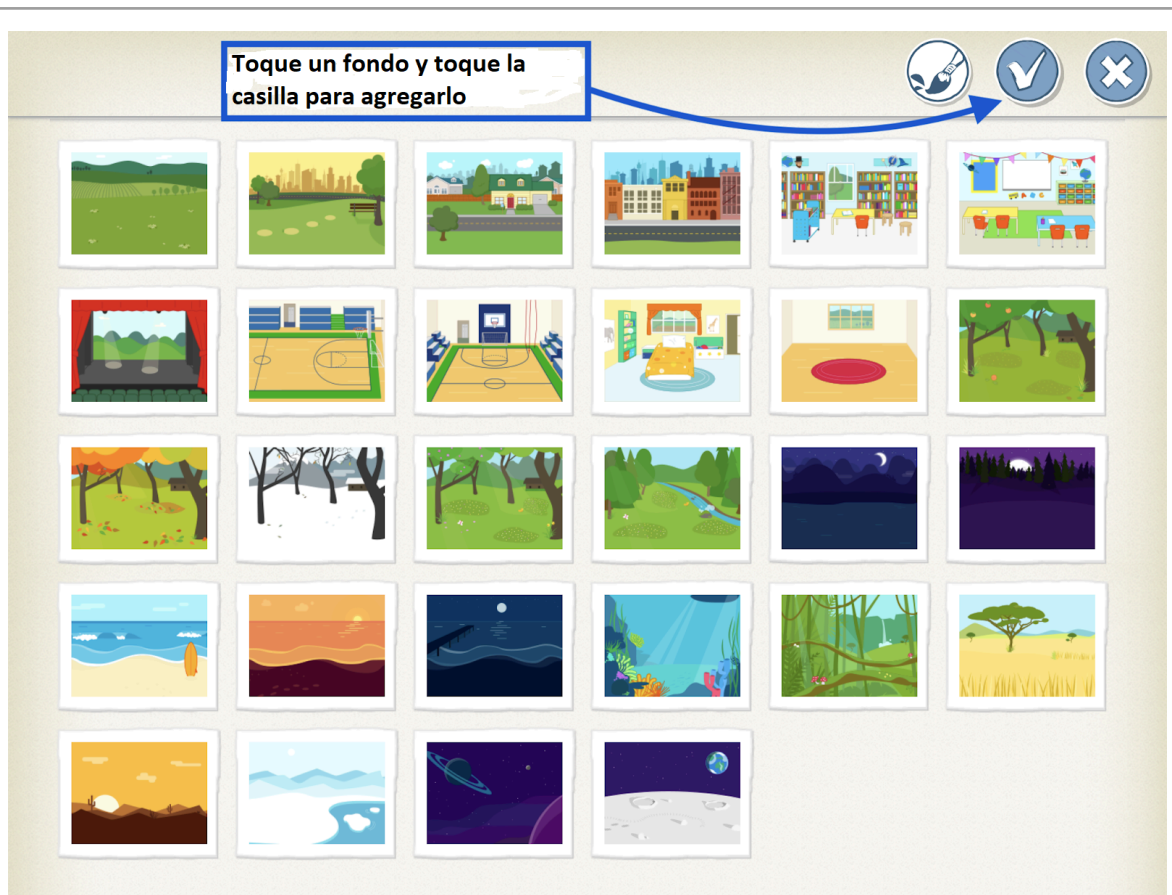
Tiempo de ScratchJr

Desafío de estructura:

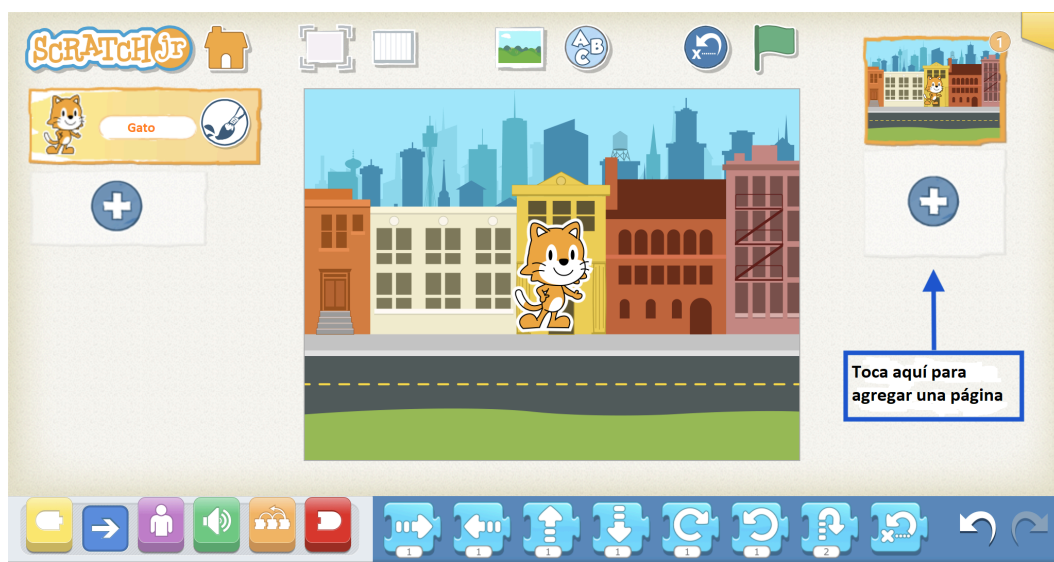
- **Cambiar la configuración y agregar una página** (*Tiempo sugerido: 5 minutos*)
 - Introduzca la configuración de cambio y agregue el bloque de página siguiendo los pasos a continuación:
 1. Muestre a los niños cómo cambiar el **fondo** en ScratchJr. Permítales ajustar el fondo en ScratchJr tocando el botón Agregar fondo.



2. Seleccione un fondo y toque el cheque para agregarlo a su página.



3. ¡Agregaste un fondo! Ahora agreguemos una página:



- Para eliminar una página, mantenga presionado el dedo en la página numerada que desea eliminar. Aparecerá una "x" roja en la parte superior del bloque de página numerado. Toque la "x" para eliminar la página. Vea el ejemplo a continuación:



Lección 9 Verificación de la comprensión: antes de comenzar un proyecto, verifique que sus hijos comprendan los nuevos conceptos que acaban de aprender. Lea cada pregunta a los niños y pídeles que respondan con un pulgar hacia arriba para decir "sí" o un pulgar hacia abajo para "no". Deténgase y vuelva a explicar los conceptos según sea necesario.

Explorando expresiones:

- **Haciendo tu historia** (*Tiempo sugerido: 15 minutos*)
 - Muestre algunas de las escenas del libro para que los niños vean la historia.
 - Usando su personaje Ada creado en la Lección 8, haga que los niños agreguen diferentes partes de la vida de Ada para crear una historia.
 - Haga que los niños practiquen el uso de la herramienta de texto para crear una página de título con el título de una historia y el nombre del autor.

Tiempo desconectado

- **El orden importa** (*Tiempo sugerido: 5 minutos*)
 - Dígalos a los niños que su historia solo tiene una página en este momento, ¡pero eventualmente tendrá muchas escenas como el libro! Comparta con la clase algunas escenas del libro.
 - Discuta con la clase si todo está en el orden correcto y si no es así, ¿cómo cambiaría eso la historia / cómo lo arreglaría?
 - Discuta cómo las historias deben tener un comienzo, un desarrollo y un final, ¡en ese orden!

Cerrando el círculo tecnológico

- **¿Cuándo es tu escena?** (*Tiempo sugerido: 5 minutos*)
 - Después de hablar sobre el principio, el medio y el final, pregunte a los niños dónde se ubica en la historia la escena que hicieron. ¿Es su escena desde el principio de la historia? ¿O el medio? ¿O al final?

Lección 10: Amplía tu caja de herramientas para contar historias

Ideas poderosas de la informática	Proceso de diseño, representación, depuración
Ideas poderosas de alfabetización	Proceso de escritura, edición y conciencia de la audiencia
PTD	Creatividad, Creación de contenido
Paleta de virtudes	Honestidad, Perdón
Los niños podrán	<ul style="list-style-type: none">● Usar el bloque Ir a la página en ScratchJr● Utilizar los bloques Crecer, Reducir, Ocultar y Mostrar en ScratchJr● Reflexionar sobre su proyecto utilizando las indicaciones del diario de diseño.
Vocabulario	<ul style="list-style-type: none">● Editar: hacer cambios para mejorar algo
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección.● Imprima el Diario de diseño de la Lección 10 para cada niño o consulte el Diario de diseño completo.● Vea proyectos de ejemplo para familiarizarse con la actividad del proyecto.

Introducción

- **El programador dice - Edición de tamaño** (*Tiempo sugerido: 5 minutos*)
 - Instruya a los niños en un juego de "El programador dice" usando las indicaciones:
 - Hazte lo más grande posible.
 - Hazte lo más pequeño posible.
 - Hazte lo más alto posible.
 - Escóndete de la cámara.

Iniciando el círculo tecnológico

- **Edades de Ada** (*Tiempo sugerido: 5 minutos*)
 - Muestre algunas escenas de la historia para mostrar lo que sucedía en la vida de Ada a diferentes edades. A lo largo de la historia, Ada va creciendo. La historia comienza con Ada de niña imaginando volar y a los diecisiete años, su infancia llegó a su fin. En su juventud se hizo amiga del señor Babbage.

- A medida que pasaban los años Ada crecía y se convertía en la primera programadora.
- Explique que hoy todos aprenderán cómo hacer que las cosas crezcan, se encojan, se escondan y se muestren en ScratchJr.

Tiempo de ScratchJr

Desafío de estructura:

- **Ir a la página** (*Tiempo sugerido: 5 minutos*)
 - Presentar el bloque Ir a la página
 - Para continuar una historia en otra página, agregue el bloque rojo Fin con una imagen de la página siguiente a uno de sus programas. Esto crea una función del programa "Pasar página". Siga los pasos a continuación:

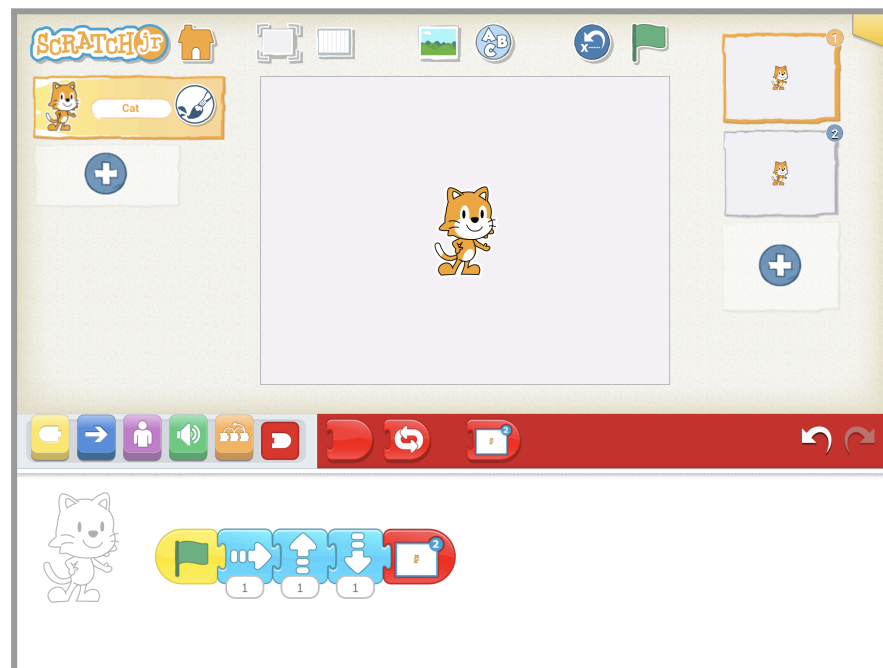
1. Agregar una nueva página



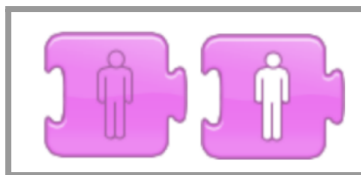
2. Una vez que haya agregado una nueva página, el bloque Ir a la página aparecerá en su paleta de bloques de extremo rojo.



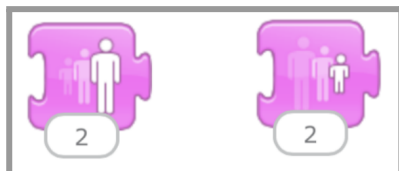
3. Agregue el bloque Ir a la página al final de su programa en la primera página para que las páginas cambien después de que finalice el programa.



- **Ocultar / Mostrar y Crecer / Reducir** (*Tiempo sugerido: 5 minutos*)
 - Introducir bloques Ocultar / Mostrar: ¡Demuestre que agregar un bloque Ocultar al código de un personaje los hará desaparecer! Puede hacer que reaparezcan usando un bloque Mostrar.



- Luego, usa los bloques Crecer / Contraer para mostrar cómo puede verse un personaje a medida que crece durante su vida.



- Parámetros en bloques de tamaño: dígalos a los niños que los bloques de tamaño también tienen parámetros, al igual que los bloques de movimiento. En lugar de contar los pasos de ScratchJr, estos parámetros le dicen cuántas veces debe crecer.
- Presente el bloque Restablecer tamaño: demuestre que al agregar un bloque Restablecer tamaño al código de un personaje, su personaje volverá a su tamaño original.



- ¡Ahora tráiganlo de vuelta a la historia! Digamos que podemos usar estos bloques para agregar páginas a nuestra historia y mostrar a Ada creciendo en nuestra nueva página. Dígalos a los niños que programarán una nueva parte de sus historias de Ada, ¡cuando Ada sea pequeña!

Explorando expresiones:

- **Agregar una nueva página con bloques morados** (*Tiempo sugerido: 10 minutos*)
 - Haga que los niños creen una nueva página para contar la historia de Ada creciendo, anímalos a agregar bloques morados (Ocultar / Mostrar, Crecer / Reducir, Restablecer tamaño) para mostrar que ella es más joven en la primera parte de la historia.

Tiempo de palabras

- **Diario de diseño de la lección 10** (*Tiempo sugerido: 10 minutos*)
 - En sus Diarios de diseño, los niños responderán a estas indicaciones sobre más cambios en su historia de Ada:
 - ¿Hay algo que desee cambiar o agregar?
 - ¿Qué más detalles puedes agregar a tu historia para mejorarla?

Cerrando el círculo tecnológico

- **Compartir el diario de diseño** (*Tiempo sugerido: 5 minutos*)
 - Los niños comparten lo que han escrito en sus diarios de diseño.

Lección 11: ¿Cuál es tu historia?

Ideas poderosas de la informática	Proceso de diseño, depuración
Ideas poderosas de alfabetización	Proceso de escritura, edición y conciencia de la audiencia
PTD	Comunicación
Paleta de virtudes	Perseverancia, Perdón
Los niños podrán	<ul style="list-style-type: none"> Definir y celebrar el proceso de depuración. Identificar estrategias de depuración y edición. Localizar errores o "errores" en el código ScratchJr.
Vocabulario	<ul style="list-style-type: none"> Depuración: para encontrar problemas en un programa de computadora e intentar resolver el problema de diferentes maneras.
Preparación del maestro	<ul style="list-style-type: none"> Leer el plan de la lección Tenga una pestaña abierta para la presentación de diapositivas del gráfico de anclaje para el proceso de diseño lista para ser proyectada. Abra el gráfico de anclaje de la lista de estrategias y esté listo para proyectar o imprimir. Proyecto del profesor que se depurará: cree un proyecto simple.

Introducción

- **Canción del proceso de diseño** (*Tiempo sugerido: 5 minutos*)
 - [Gráfico de anclaje para el proceso de diseño](#) que se muestra en el aula o en la pantalla
 - Resalte en Prueba y Mejora

(canta con la melodía de "estrellita, estrellita")
Pregunto e imagino, planeo y creo.
Pruebo y mejoro y comparto lo que hacemos.
(Repetir)

Iniciando el círculo tecnológico

- **Probar, depurar y mejorar** (*Tiempo sugerido: 10 minutos*)
 - Muestre la tabla del proceso de diseño nuevamente y recuerde a los niños los pasos para hacer un Proyecto ScratchJr: ¡Pregunte, Imagínesse, Planifique, Cree, Pruebe y Mejore, Comparta! ¡Hoy aprenderemos cómo probar y mejorar!

- Dígales a los niños que debemos probar nuestros programas para ver si siguen nuestro plan, y si no lo hacen ... ¡Debemos mejorar!
- **¿Qué es la depuración?**
 - **Depuración** es una palabra que se usa en ciencias de la computación para describir cuando las personas encuentran **errores** en sus programas de computadora y tratan de resolverlos de diferentes maneras. Si bien la palabra "error" se usó en otros campos científicos, la palabra "depuración" se atribuye a la almirante Grace Hopper, quien en la década de 1940 encontró una polilla atascada dentro de la computadora (¡las computadoras solían ser tan grandes!). Lo que provocó un error en el sistema. Pudo resolver el error eliminando el error, de ahí la palabra "depuración".
 - Discuta con los niños acerca de los problemas que surgen durante la programación. Los programadores pasan por el proceso repetitivo de depuración para encontrar y solucionar problemas.
 - Como clase, cree una lista de estrategias para encontrar errores en los programas. Gráfico de anclaje de la lista de estrategias. Regrese a la [lista para agregar más estrategias](#) cuando sea necesario.

Tiempo de ScratchJr

Desafío de estructura:

- **Depuración del proyecto de la maestra/o** (*Tiempo sugerido: 10 minutos*)
 - Levante su proyecto de error y explique a los niños cuál era su plan (por ejemplo, quería que el gato salte dos veces y luego pase a ser perro). Luego, muestre a los niños el proyecto y señale algunas cosas que no están funcionando como usted quería.
 - Haga que los niños se turnen para levantar la mano y depurar cosas en sus proyectos. Asegúrese de detener y reproducir el programa después de cada cambio. Dígales a los niños que está probando. ¡Recuerden que las pruebas fueron parte de nuestro proceso de diseño!

Explorando expresiones:

- **Edite su proyecto** (*Tiempo sugerido: 10 minutos*)

Haga que los niños revisen sus proyectos. Han editado su historia en su diario de diseño escribiendo sobre cosas que cambiarían o agregarían. Ahora es el momento de que los niños editen sus proyectos.

 - Haga que los niños revisen sus diarios de diseño y sus programas.
 - ¿Qué partes de su programa no coinciden con la historia que ha escrito ahora? Intente arreglar su programa para que coincida mejor con su historia cambiada.
 - Forme parejas o grupos pequeños para discutir si alguien encontró errores en su proyecto ScratchJr.

Cerrando el círculo tecnológico

- **Reflexión sobre depuración** (*Tiempo sugerido: 10 minutos*)

- Revisar el proceso de diseño: la depuración es parte del proceso de creación.
- Presente la idea de agregar estrategias para depurar y editar al Gráfico de anclaje de la lista de estrategias.

Lección 12: ¡Encuentra ese error!

Ideas poderosas de la informática	Depuración
Ideas poderosas de alfabetización	Edición y conciencia de la audiencia, conciencia fonológica
PTD	Comunicación
Paleta de virtudes	Perdón, perseverancia, paciencia
Los niños podrán	<ul style="list-style-type: none"> ● Definir y celebrar el proceso de depuración. ● Identificar estrategias de depuración y edición. ● Localizar errores o "errores" en el código ScratchJr. ● Solucionar errores en ScratchJr ● Localizar errores en el texto en español (por ejemplo, errores de ortografía, letras escritas incorrectamente).
Vocabulario	
Preparación del maestro	<ul style="list-style-type: none"> ● Leer el plan de la lección ● Haga que la presentación de diapositivas de ejemplos de error se abra en una nueva pestaña y esté lista para proyectar. ● Haga que la presentación de diapositivas de proyectos con error se abra en una nueva pestaña y esté lista para proyectar. ● Imprima el Diario de diseño de la lección 12 para cada niño o consulte el Diario de diseño completo.

Introducción

- **Ejemplos de errores** (*Tiempo sugerido: 5 minutos*)
 - Muestre [ejemplos de errores](#) en el lenguaje de programación ScratchJr y en el idioma español en la clase.
 - Pregúnteles a los niños qué hay de malo en estos programas, palabras u oraciones.
 - ¿Cómo podemos solucionarlos?

Iniciando el círculo tecnológico

- **Probar y mejorar** (*Tiempo sugerido: 5 minutos*)
 - Analice ejemplos de pruebas y mejoras.
 - Al programar, una computadora encuentra un error y no funciona.

- En alfabetización, una persona se encuentra con un error y, a veces, puede leerlo de todos modos, pero a veces es demasiado difícil y no se puede leer.
- Explique la diferencia entre cambiar el aspecto y depurarlo.
 - Por ejemplo, cambiar de apariencia: cambiar a gato de naranja a azul.
 - Por ejemplo, depuración: cambiar el programa para que el gato se mueva en la dirección correcta.

Tiempo de ScratchJr

Desafío de estructura:

- **Proyectos con errores** (*Tiempo sugerido: 15 minutos*)
 - Estos [proyectos de ScratchJr tienen algunos errores](#). Recrear estos proyectos o crear uno propio para compartir con la clase. Cada proyecto contiene un mensaje con lo que el programador quería que sucediera y con lo que realmente está sucediendo. Estos proyectos deben depurarse para que funcionen:
 - Quiero que el gato y el conejo salten para recoger manzanas de los árboles, pero sólo el gato salta para recoger manzanas y el conejo se mueve hacia la derecha.
 - Quiero que el gato desaparezca y luego se haga más grande y reaparezca, pero el gato desaparece, reaparece y luego se hace más pequeño.
 - Cuando los niños encuentren errores, participen en la celebración del proceso de depuración.
 - Aula: choca los cinco y alienta!
 - Virtual - Aplausos y pulgares arriba.

Tiempo de palabras

- **Diario de diseño de la lección 12** (*Tiempo sugerido: 10 minutos*)
 - ¡Haga que los niños abran sus Diarios de diseño en la Lección 12 y practiquen la búsqueda y corrección de errores por su cuenta! Díales que marquen con un círculo cada vez que encuentren un error y luego escriban su solución en el cuadro al lado.

Cerrando el círculo tecnológico

- **Reflexión sobre depuración** (*Tiempo sugerido: 10 minutos*)
 - Piense en estrategias para depurar la escritura.
 - Por ejemplo, volver a leer su escritura para encontrar errores, hacer que un amigo mire su escritura

Lección 13: ¿Puedes oírme?

Ideas poderosas de la informática	Hardware software
Ideas poderosas de alfabetización	Herramientas de comunicación y lenguaje
PTD	Comunicación
Paleta de virtudes	Mente abierta, paciencia
Los niños podrán	<ul style="list-style-type: none"> ● Identificar diferentes medios para expresar emociones (por ejemplo, expresiones faciales, movimientos corporales, ruidos, palabras, etc.). ● Usar el bloque decir en ScratchJr ● Grabar un clip de sonido con éxito utilizando la grabadora de sonido en ScratchJr. ● Utilizar bloques de sonido grabados en un programa ScratchJr.
Vocabulario	<ul style="list-style-type: none"> ● Grabar: usar su tableta para guardar su voz para escucharla más tarde ● Sonido: un ruido
Preparación del maestro	<ul style="list-style-type: none"> ● Leer el plan de la lección. ● Abra la presentación de diapositivas Sentir caras y prepárese para proyectarla. ● Abra la presentación de diapositivas de Herramientas de comunicación y prepárese para proyectarla.

Introducción

- **¡Dime cómo te sientes realmente!** *(Tiempo sugerido: 5 minutos)*
 - Comparta imágenes de [3 caras de sentimientos](#) (sonriendo, llorando, enojado) y luego pregunte a los niños qué nos dicen estas caras sonrientes.

Iniciando el círculo tecnológico

- **Herramientas de comunicación** *(Tiempo sugerido: 5 minutos)*
 - Muestre el cuadro de anclaje de [Herramientas de comunicación](#) y hable sobre todas las formas en que podemos comunicar nuestros pensamientos y sentimientos.
 - Palabras
 - Voces
 - Discuta la diferencia entre palabras y voces, ¡hable sobre cómo la forma en que suenan las voces le dice algo!

- Caras
- Cuerpos
- ¡Pregúnteles si pueden pensar en otras formas! ¿ScratchJr? ¿Números? ¿Dibujos?

Tiempo desconectado

- **Actividad de expresión** (*Tiempo sugerido: 10 minutos*)
 - Juegue un juego de charadas, donde los niños representan un sentimiento y otros niños adivinan lo que el niño está tratando de expresar (por ejemplo, el niño tiene el ceño fruncido, los brazos cruzados, las cejas flexionadas = enojado / molesto).
 - Déle al niño un mensaje cuando sea su turno (p. Ej., "Muestre a la clase que está feliz con su cuerpo", "Muestre a la clase que está triste con su voz (pero no puede decir 'Estoy triste, '"" Muestre a la clase que es pícaro con su cara)").
 - ¡Dale a cada niño un turno para ser el personaje expresivo!

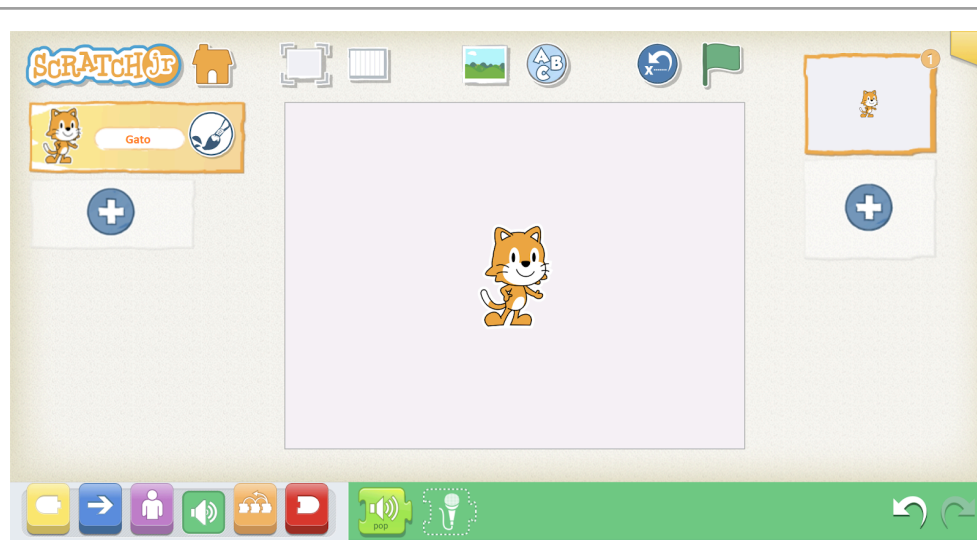
Tiempo de ScratchJr

Desafío de estructura:

- **Presentar el bloque "Decir"** (*Tiempo sugerido: 5 minutos*)
 - Los niños pueden usar el bloque "Decir" para que su personaje diga algo. Toque el bloque "Decir" para usar el teclado del dispositivo para agregar texto.

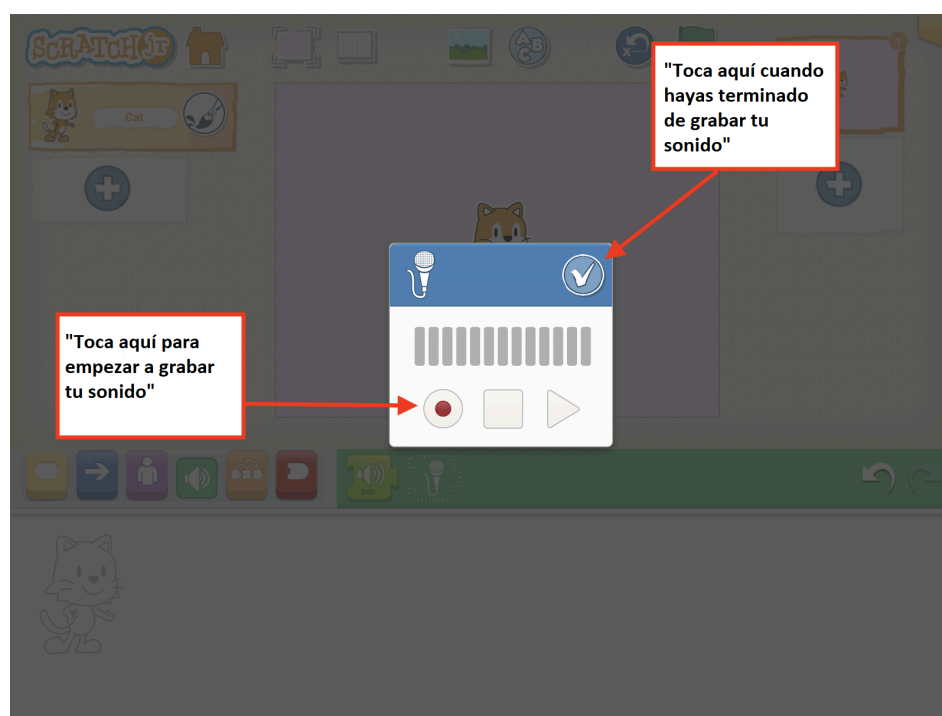


- **Reproducir el sonido grabado** (*Tiempo sugerido: 5 minutos*)
 - Uso del bloque Reproducir grabadora de sonido:
 1. En el área de programación, haga clic en el símbolo de **sonido** verde. Dentro de ese menú, haga clic en el bloque de micrófono con el borde punteado para grabar el sonido.

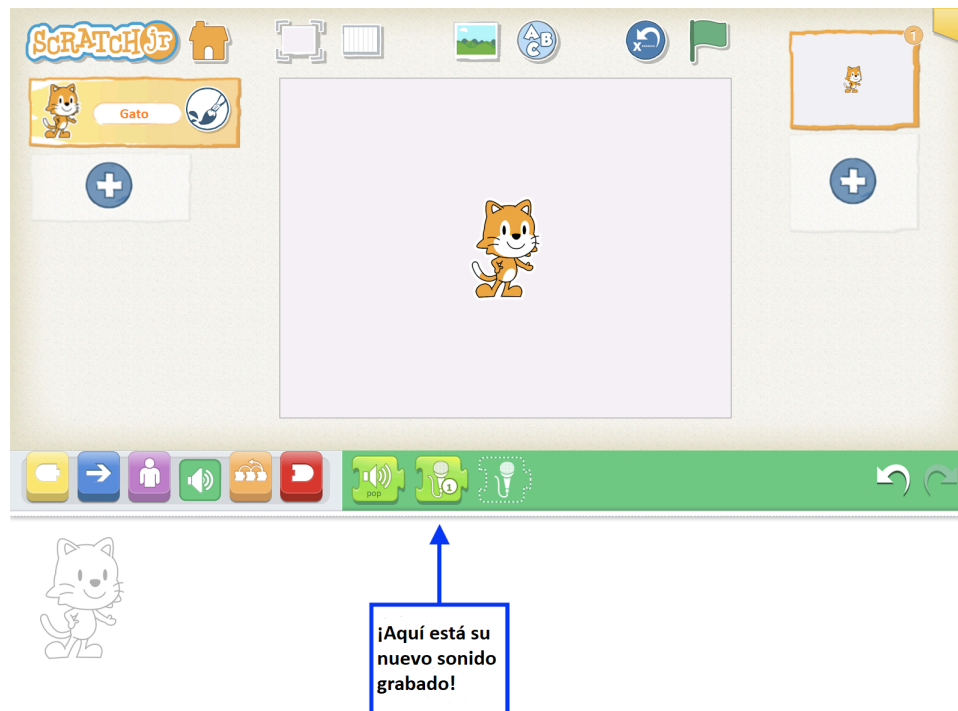


¡Toque aquí para grabar su propio sonido!

2. Toque el punto rojo para grabar su sonido. Cuando haya terminado de grabar, presione el mismo punto para detener la grabación. Luego, toque la marca de verificación.



3. ¡Ahora tiene su propio sonido grabado para usar en su programa! Arrastre el nuevo bloque que apareció en la paleta (el micrófono con un número al lado) al área de programación. Tóquelo y deje que los niños se escuchen a sí mismos.



Explorando expresiones:

- **Déjame escuchar cómo te sientes** (*Tiempo sugerido: 10 minutos*)
 - Haga que el niño programe un personaje expresivo usando el gato en ScratchJr
 - Haga que los niños practiquen el uso del bloque Grabador de sonido y el bloque "Decir"

Cerrando el círculo tecnológico

- **Compartir creaciones** (*Tiempo sugerido: 5 minutos*)
 - Algunos niños comparten sus programas ScratchJr usando el bloque de grabadora de sonido.

Oportunidades de diferenciación

- **Desafío extra**
 - Para los niños que necesitan más desafíos o andamios, déles las siguientes indicaciones:
 - ¿De cuántas formas se puede crear un programa que exprese la misma idea?
 - Intente utilizar herramientas de comunicación, incluidos sonidos, palabras, movimientos o la herramienta de pincel.

Lección 14: Volver a empezar.

Ideas poderosas de la informática	Estructuras de Control
Ideas poderosas de alfabetización	Recursos literarios
PTD	Creación de contenido
Paleta de virtudes	Curiosidad, paciencia
Los niños podrán	<ul style="list-style-type: none">● Identificar situaciones condicionales en la vida cotidiana.● Usar los bloques Comenzar al presionar y comenzar al tocar en ScratchJr.● Usar el bloque Stop en ScratchJr.
Vocabulario	<ul style="list-style-type: none">● Condicional: algo que solo sucederá si sucede algo más primero.
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección● Imprima la Verificación de la comprensión de la lección 14 a cada niño o abra las diapositivas de la verificación de la comprensión de la lección 14 para proyectar.

Introducción

- **Cuatro esquinas** (*Tiempo sugerido: 10 minutos*)
 - Designe cuatro esquinas del aula, una alfombra o un espacio abierto como espacio de juego.
 - Un niño (o el maestro) se para en el centro del salón y cierra los ojos. Cada niño tiene que ir a una de las cuatro esquinas. Cuando todos están en una esquina, la persona en el centro señala en una dirección aleatoria y abre los ojos. Todos en esa esquina están fuera. Esto se repite, con la persona en el centro cerrando los ojos, todos eligiendo una esquina, la persona en el centro eligiendo una esquina, y todos en esa esquina están afuera. Continúe hasta que quede un ganador o 2-3 niños.
 - Si el tiempo lo permite, el ganador del juego anterior puede estar en el centro para una segunda ronda.

Iniciando el círculo tecnológico

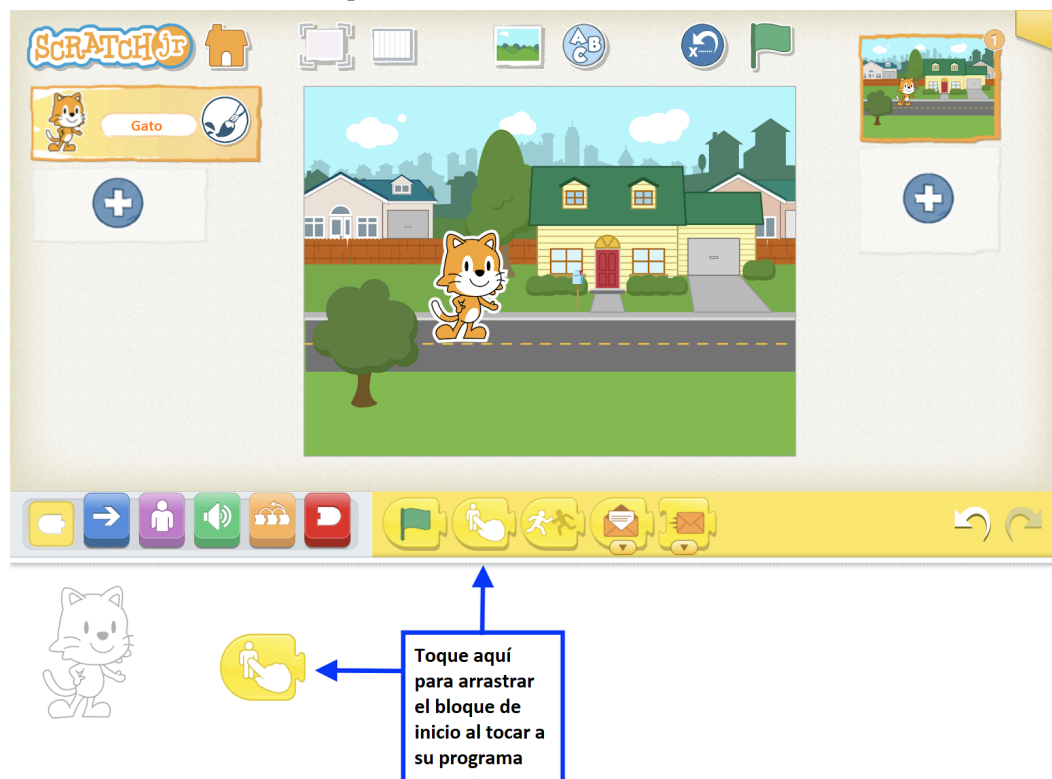
- **¿Qué es un condicional?** (*Tiempo sugerido: 5 minutos*)

- Reflexiona sobre el juego de las cuatro esquinas. Si se señalaba su esquina, estaba fuera. Para determinar si estaba a salvo o fuera, tenía que verificar hacia dónde apuntaba la persona en el centro. A esto se le llama **condicional**.
- Los condicionales son cuando una cosa tiene que suceder antes de que sucedan las siguientes. Analice los siguientes ejemplos:
 - ¿Qué revisa antes de ponerse una campera? (Si hace frío o llueve)
 - ¿Qué tiene que pasar antes de hablar en clase? (Que el maestro te llame)
 - ¿Qué tiene que pasar antes de comer? (Lávese las manos)
 - ¿Qué verifica antes de que el auto pueda avanzar? (Que la luz del semáforo sea verde)

Tiempo de ScratchJr

Desafío de estructura:

- **Comience al tocar / presionar & Stop** (*Tiempo sugerido: 10 minutos*)
 - Presentar el Bloque comenzar al presionar
 - Demuestre que el uso de este bloque al comienzo de un programa hace que un personaje represente su programa solo después de que haya tocado el personaje en la pantalla. Muestre esto en el modo de presentación para demostrar claramente la diferencia entre comenzar al presionar y comenzar al presionar la bandera verde.



- Introduzca el bloque stop y el bloque comenzar al tocar
 - Demuestre cómo usar el bloque Stop y el bloque comenzar al tocar en ScratchJr. El bloque stop se utiliza para finalizar todos los programas que se ejecutan para un carácter en particular, excepto el programa que

contiene el bloque de stop en sí. El bloque comenzar al tocar se usa para hacer que el programa de un personaje comience después de que otro personaje choca con el personaje. Para enseñar estos bloques, el maestro debe tener dos personajes. El primer carácter tiene dos programas: uno que se repite para siempre y otro que se detiene cuando se golpea. El segundo carácter debe tener una secuencia que se repita para siempre. Por ejemplo:



Verificación de la comprensión de la lección 14: antes de comenzar un proyecto, verifique que los niños comprendan los nuevos conceptos que acaban de aprender. Lea cada pregunta a los niños y pídeles que respondan con un pulgar hacia arriba para decir "sí" o un pulgar hacia abajo para decir "no". Deténgase y vuelva a explicar los conceptos según sea necesario.

Explorando expresiones:

- **Juego libre** (*Tiempo sugerido: 15 minutos*)

- Los niños pueden explorar con los bloques comenzar al tocar, Stop y comenzar a presionar.
 - Refuerce el proceso de depuración de prueba y error mientras usa el bloque comenzar al tocar.

Cerrando el círculo tecnológico

- **Compartiendo creaciones** (*Tiempo sugerido: 5 minutos*)
 - Los niños comparten cómo usaron los bloques comenzar al tocar, Stop y comenzar al presionar durante el juego libre.

Lección 15: ¡Necesita velocidad!

Ideas poderosas de la informática	Estructuras de Controls
Ideas poderosas de alfabetización	Recursos literarios
PTD	Creación de contenido, elecciones de conducta
Paleta de virtudes	Justicia
Los niños podrán	<ul style="list-style-type: none">● Utilizar el bloque de velocidad establecido en ScratchJr.● Utilizar el bloque Ir al inicio en ScratchJr.
Vocabulario	<ul style="list-style-type: none">● Lento: moverse de una manera que requiere más tiempo● Rápido: moverse de una manera que requiere menos tiempo● Regresar: volver a donde comenzaste
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección.● Papel cuadriculado grande.

Introducción

- **Cambiando la velocidad** (*Tiempo sugerido: 5 minutos*)
 - Haga que los niños aplaudan a diferentes velocidades (**rápido / lento**). Luego haga que los niños traten de decir el trabalenguas, "El hipopótamo Hipo está con hipo, ¿quién le quita el hipo al hipopótamo Hipo?", tanto lento como rápido.

Iniciando el círculo tecnológico

- **¿Qué hace que una carrera sea una carrera?** (*Tiempo sugerido: 5 minutos*)
 - Analice con los niños las reglas y los elementos de una carrera.
 - ¿Cuáles son los elementos centrales de una carrera (por ejemplo, punto de partida, distancia de la carrera, etc.)?

Tiempo de ScratchJr

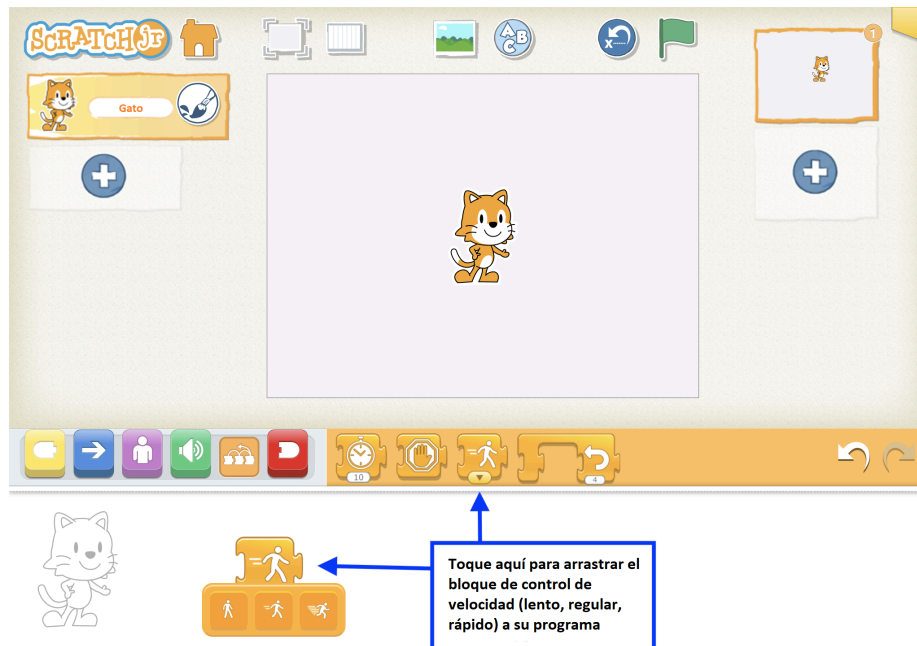
Desafío de estructura:

1. **Bloque de velocidad** (*Tiempo sugerido: 5 minutos*)

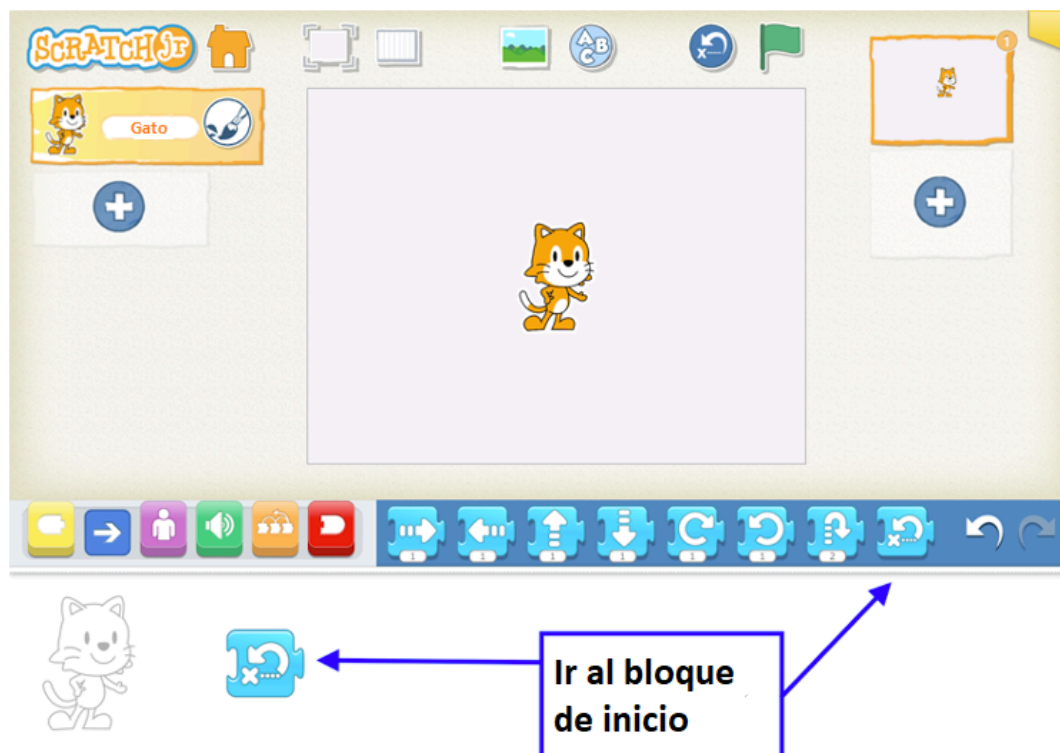
Presente el bloque de velocidad.

 - Usando el bloque de Velocidad, se puede hacer que los personajes de ScratchJr se muevan a diferentes velocidades (rápido, regular, lento, también llamado “correr,

caminar, gatear”). Puede presentar el bloque de programación de velocidad y demostrar cómo usarlo en el programa ScratchJr.



2. Ir al Inicio (Tiempo sugerido: 5 minutos)



Introduzca el bloque Ir al inicio. Este bloque será útil para los niños ya que sus corredores deberán **regresar** al punto de partida después de la carrera.

Explorando expresiones:

- **Es una carrera** (*Tiempo sugerido: 10 minutos*)
 - Dígalos a los niños que programarán su propia carrera. Antes de que comiencen a programar, decida los mismos tres personajes que usarán para su carrera. Dígale a los niños al final que van a contar qué personaje ganó en cada carrera y verán cuántas victorias obtuvo cada personaje.
 - Los niños deben usar la cuadrícula para asegurarse de que todos los personajes comiencen en el mismo lugar. Revisa las reglas de una carrera:
 - Para ser justos: todos los corredores comienzan en el mismo lugar y viajan la misma distancia hasta la línea de meta.

- Los corredores deben volver a la salida (usar el bloque Ir a la salida) cuando terminen la carrera.



Tiempo desconectado

- **Comparando ganadores** (*Tiempo sugerido: 10 minutos*)
 - Muestre a los niños un gráfico de barras simple con una etiqueta de "Victorias" en el eje Y (vertical) y una etiqueta de "Personajes" en el eje X (horizontal). Escriba también los tres personajes que la clase eligió para sus carreras a lo largo del eje X (horizontal).
 - Haga que cada niño juegue su carrera para la clase. Después de cada acción, registre la ganancia en el gráfico de barras.
 - Una vez que todos hayan compartido, examine el gráfico de barras y explique que esta es una forma en que podemos comparar cuántas veces ganó cada personaje.

Cerrando el círculo tecnológico

- **¿Qué más podemos comparar?** (*Tiempo sugerido: 5 minutos*)
 - Dígalos a los niños que una **gráfica**, como la que acabamos de hacer, se puede usar para contar y comparar muchas cosas. Dé algunos ejemplos (por ejemplo, cuántos niños tienen mascotas en casa, cuántos días hay sol y cuántos días llueve esta semana) y luego pida a algunos voluntarios que compartan algo que puedan contar y ¡comparar!

Oportunidades de diferenciación

- **Desafío extra**
 - Si los niños terminaron con su programa de carrera, pídeles que practiquen como comentaristas deportivos anunciando la carrera para que estén listos para compartir la historia de su programa (practique en parejas si otros también terminaron)



Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) by the [DevTech Research Group](#) is licensed under a [Creative Commons Attribution NonCommercial-ShareAlike 3.0 Unported License](#). Under this license, you may use and adapt this work, but you must attribute the work to the DevTech Research Group. You **may not** use or adapt this work for commercial purposes.

This work is funded by the US Dept of Education (grant #U411C190006) and by the Scratch Foundation.

© 2020, DevTech Research Group.

Lección 16: ¡Hagamos el baile de las estatuas!

Ideas poderosas de la informática	Estructuras de Control
Ideas poderosas de alfabetización	Recursos literarios
PTD	Construcción de comunidad, creación de contenido
Paleta de virtudes	Honestidad, Paciencia
Los niños podrán	<ul style="list-style-type: none"> • Usar y ajustar el bloque de espera en ScratchJr. • Reflexionar sobre trabajos anteriores. • Crear programas paralelos en ScratchJr.
Vocabulario	<ul style="list-style-type: none"> • Espere: permanecer en su lugar antes de comenzar. Esperar se refiere al proceso de quedarse. • Pausa: para detenerse en su lugar antes de reiniciar. Pausar se refiere al acto de detenerse. • Programación paralela: cuando dos o más programas se ejecutan al mismo tiempo
Preparación del maestro	<ul style="list-style-type: none"> • Leer el plan de la lección • Abra la presentación de diapositivas con música para el baile de las estatuas, para elegir cualquier canción y reproducirla en voz alta. • Imprima una copia por niño los recortes de bloques de de ScratchJr y recorte el encabezado de cada página. • Imprima la Verificación de comprensión de la lección 16 a cada niño o abra la presentación de la Verificación de comprensión de la lección 16, para proyectar.

Introducción

- **Baile de estatuas** (*Tiempo sugerido: 5 minutos*)
 - Utilizando música para el [baile de estatuas](#) o su música preferida para el aula, juegue al baile de estatuas. El baile de estatuas es un gran juego para hacer que los niños se muevan y comprometan su creatividad.
 - Cuando suena la música, los niños bailan y cuando la música se **detiene**, deben congelarse inmediatamente. Como maestro, controle la música y presione pausa periódicamente para que los niños se congelen. Recuerde reforzar las normas de la clase en torno a la seguridad y la precaución con los cuerpos.
 - Opcional: puede mezclarlo con música rápida y música lenta para incorporar las ideas de los bloques introducidos durante la última lección.

Iniciando el círculo tecnológico

- **¿Qué es el baile de estatuas?** (*Tiempo sugerido: 5 minutos*)
 - Discuta cómo hacer que un personaje de ScratchJr baile como antes estuvieron bailando. ¡Cuando bailamos hacemos muchos movimientos diferentes al mismo tiempo! Por ejemplo, podemos saltar y avanzar al mismo tiempo, ¡o podemos girar y mover nuestras manos al mismo tiempo!
 - En ScratchJr, ¿cómo podemos programar a nuestros personajes para que realicen diferentes movimientos al mismo tiempo? Dé a los niños la oportunidad de intercambiar ideas.
 - Luego, discuta las reglas y los elementos de un baile de estatuas (bailar mientras suena la música, congelarse cuando la música se detiene)
 - En ScratchJr, ¿cómo podríamos programar a los personajes para que esperen cuando la música se detenga?

Tiempo de ScratchJr

Desafío de estructura:

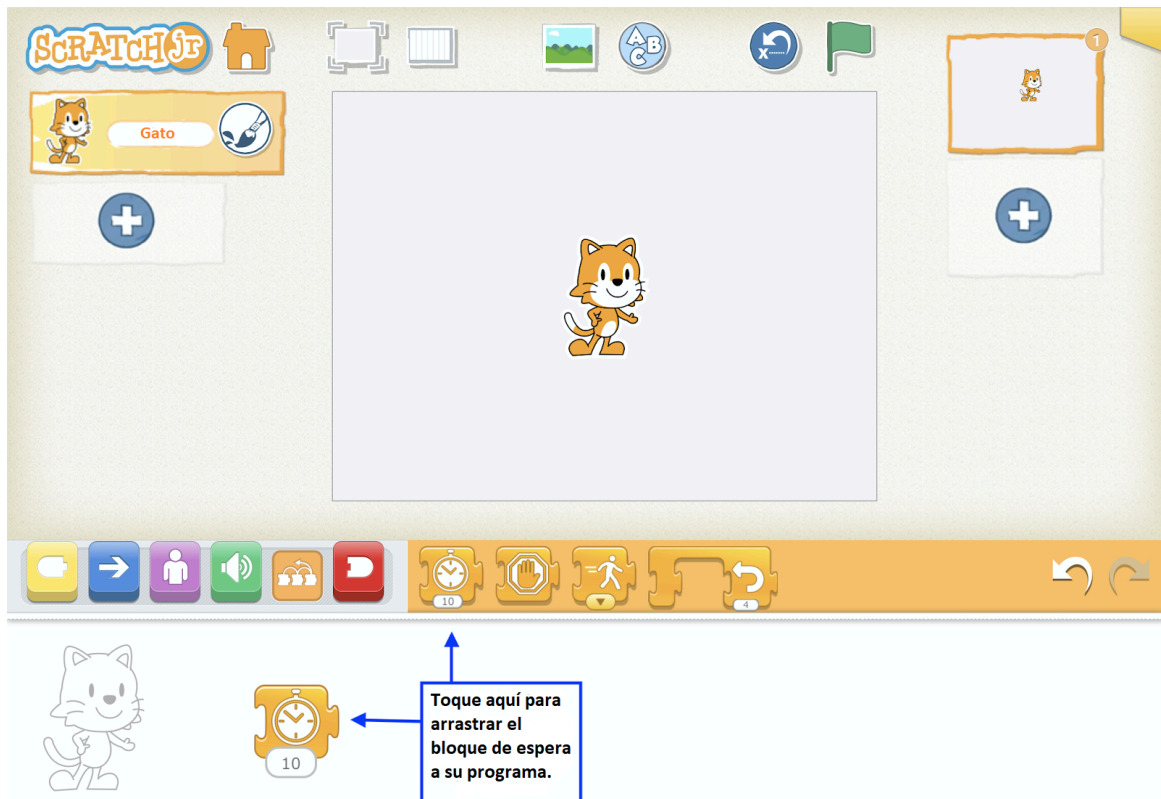
1. Programación paralela (*Tiempo sugerido: 5 minutos*)

Introducir **programación paralela**: significa que dos programas están sucediendo al mismo tiempo. Un ejemplo de programa paralelo sería el siguiente:



- Los programas paralelos anteriores permiten que el gato se mueva hacia la derecha al mismo tiempo que el gato se mueve hacia arriba, por lo que el gato se moverá en diagonal.
- **Bloque de espera** (*Tiempo sugerido: 5 minutos*)
 - Introduce el bloque de espera.

- El bloque Esperar programa a tu personaje para que tome un descanso o pausa dentro del programa. El número se refiere a DÉCIMOS (1/10) de segundo, no a segundos. Por lo tanto, un bloque de espera con un parámetro de 10 números pausará el programa durante 1 segundo.



Lección 16 Verificar la comprensión: antes de comenzar un proyecto, verifique que sus hijos comprendan los nuevos conceptos que acaban de aprender. Lea cada pregunta a los niños y pídeles que respondan con un pulgar hacia arriba para decir "sí" o un pulgar hacia abajo para decir "no". Deténgase y vuelva a explicar los conceptos según sea necesario.

Explorando expresiones:

- **Programa tu propio baile de estatuas** (*Tiempo sugerido: 15 minutos*)
 - Haga que los niños programen su propio baile de estatuas.
 - Anime a los niños a usar la programación paralela para hacer que los personajes bailen en diferentes direcciones.
 - Recuerde a los niños que el bloque de espera puede hacer que los personajes se congelen y luego sigan bailando.
 - Recuerde a los niños que es importante que todos los personajes se congelen al mismo tiempo.

Cerrando el círculo tecnológico

- **Compartir creaciones** (*Tiempo sugerido: 10 minutos*)

Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) by the [DevTech Research Group](#) is licensed under a [Creative Commons Attribution NonCommercial-ShareAlike 3.0 Unported License](#). Under this license, you may use and adapt this work, but you must attribute the work to the DevTech Research Group. You **may not** use or adapt this work for commercial purposes.

This work is funded by the US Dept of Education (grant #U411C190006) and by the Scratch Foundation.

© 2020, DevTech Research Group.

- Los niños comparten sus creaciones. Indicaciones:
 - ¿Qué fue fácil o difícil de este proyecto?
 - Si su programa tuviera música, ¿sería música rápida o lenta?
 - ¿Qué harías de manera diferente si pudieras volver atrás y editar?

Oportunidades de diferenciación

- **Desafío extra**
 - Para los niños que necesitan más desafíos o andamios, pregúnteles cómo pueden incluir los bloques Comenzar al presionar, comenzar al tocar y velocidad en sus programas.

Lección 17: ¿Puedes repetir eso?

Ideas poderosas de la informática	Algoritmos, estructuras de control
Ideas poderosas de alfabetización	Dispositivos literarios, herramientas de comunicación y lenguaje
PTD	Creación de contenido
Paleta de virtudes	Mente abierta, curiosidad, perseverancia
Los niños podrán	<ul style="list-style-type: none"> ● Identificar patrones repetidos. ● Utilizar el bloque de repetición en ScratchJr. ● Reconocer que un programa se puede representar utilizando múltiples enfoques.
Vocabulario	<ul style="list-style-type: none"> ● Repetir: hacer algo de nuevo ● Bucle: algo que se repite ● Patrón: algo que sucede de forma repetida
Preparación del maestro	<ul style="list-style-type: none"> ● Leer el plan de la lección. ● Abra la presentación de diapositivas Repetir actividad y esté listo para proyectar. ● Imprima la Verificación de la comprensión de la lección 17 a cada niño o abra la presentación en diapositivas para proyectar la lección.

Introducción

- **Las ruedas del autobús** (*Tiempo sugerido: 5 minutos*)
 - Cantar y bailar “Las ruedas del autobús” en la clase

*Las ruedas del autobús girando van
 Girando van
 Girando van
 Las ruedas del autobús girando van
 Girando van
 Girando van
 Por la ciudad.*

*Los limpiaparabrisas hacen swish, swish, swish;
 Swish, swish, swish;*

Swish, swish, swish.
Los limpiaparabrisas hacen swish, swish, swish,
por la ciudad.

- Después de cantar dos versos de la canción, discuta con la clase las repeticiones que observaron en la canción.
 - ¿Qué línea **repetimos** varias veces?
 - Pida a los niños que se imaginen que le están enseñando a su amigo la letra de la canción. ¿Existe una manera más fácil de decirles todas las palabras sin decir las mismas palabras tantas veces?
 - ¡Dile a los niños que la hay! Puede decirle a su amigo que “Repita” la línea un cierto número de veces. Como, por ejemplo, "¡Repita 'girando va, girando va' 3 veces!"

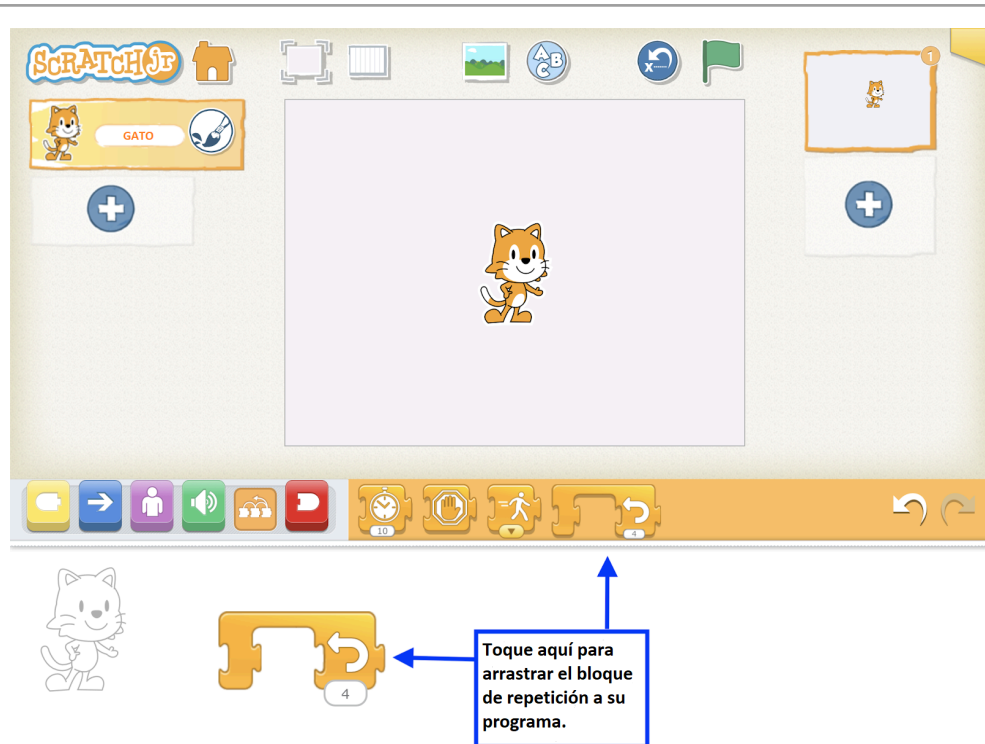
Iniciando el círculo tecnológico

- **Patrones y bucles de repetición** (*Tiempo sugerido: 5 minutos*)
 - Crea un **patrón** de aplausos (rápido y lento), que también consiste en repetición y ritmo.
 - Pida a los niños que sigan el ritmo de las palmas que se acaba de demostrar.
 - Para mayor desafío: pida a los niños que se ofrezcan como voluntarios para crear su propio patrón para que el resto de la clase lo siga.

Tiempo de ScratchJr

Desafío de estructura:

- **Bloque de repetición** (*Tiempo sugerido: 15 minutos*)
 - Un bloque de repetición puede hacer que nuestros códigos sean más eficientes, lo que puede ahorrarnos tiempo y espacio. Por ejemplo, si queremos que un personaje se oculte y se muestre 20 veces, ¡imagina cuánto tiempo nos llevará arrastrar cada bloque 20 veces! En su lugar, podemos usar un bloque Repetir, que repetirá el código para ocultar y mostrar 20 veces.



- Repetir bucles permite que se repitan todos los bloques dentro del bucle. Se repiten tantas veces como el parámetro numérico (en la imagen de arriba los bloques se repetirán 4 veces).
- [Repetir actividad](#)
 - Establezca una conexión con la alfabetización: hay diferentes formas de comunicar el mismo mensaje, como hacer una llamada, enviar mensajes de texto y enviar cartas.
 - Muestre una serie de programas como los que se muestran a continuación y pida a los niños que creen un nuevo programa que haga lo mismo, pero usando un bloque de repetición.
 - Analice las diferentes formas en que cada programa maneja la repetición.



Lección 17 Verifique la comprensión: antes de comenzar un proyecto, verifique que sus alumnos comprendan los nuevos conceptos que acaban de aprender. Lea cada pregunta a los niños y pídeles que respondan con un pulgar hacia arriba para decir "sí" o un pulgar hacia abajo para decir "no". Deténgase y vuelva a explicar los conceptos según sea necesario.

Explorando expresiones:

- **Juego libre con bloques de repetición** (*Tiempo sugerido: 15 minutos*)
 - ¡Haga que los niños piensen en diferentes patrones para los que pueden usar bloques de repetición! Déjelos explorar por su cuenta, revisando periódicamente para asegurarse de que estén usando los bloques Repetir.

Cerrando el círculo tecnológico

- **Compartiendo creaciones** (*Tiempo sugerido: 5 minutos*)
 - Los niños comparten su experiencia de juego libre. Utilice estas indicaciones guiadas:
 - ¿Cómo usaste el bloque Repetir en tu programa?
 - ¿Experimentó algún desafío?

Lección 18: Haz que suceda

Ideas poderosas de la informática	Estructuras de Control
Ideas poderosas de alfabetización	Recursos literarios
PTD	Creación de contenido
Paleta de virtudes	Curiosidad, mentalidad abierta
Los niños podrán	<ul style="list-style-type: none">● Identificar usos para los bloques Enviar mensaje y comenzar a recibir mensajes en proyectos ScratchJr● Usar los bloques Enviar mensaje y Comenzar en mensaje en ScratchJr
Vocabulario	<ul style="list-style-type: none">● Mensaje: algo que se comunica por escrito, hablado o mediante señales.
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección.● Abre el video de la carrera de relevos en una nueva pestaña.● Imprima la verificación de la Lección 18 o abra las diapositivas de la verificación de la Lección 18.

Introducción

- **¡Pasar el bastón!** (*Tiempo sugerido: 5 minutos*)
 - Reproduzca el [video de la carrera de relevos](#) y explique a los niños que cuando los corredores compiten en equipo, se pasan un palo llamado porra o bastón para señalar que pueden detenerse y la siguiente persona puede ir y comenzar a correr.

Iniciando el círculo tecnológico

- **¡El bastón que baila!** (*Tiempo sugerido: 10 minutos*)
 - ¡Elige un objeto de la clase para que sea el palo o bastón de baile! Explíqueles a los niños que el bastón de baile será como el bastón de la carrera de relevos, ¡solo que les dirá cuándo empezar a bailar en lugar de competir! ¡Haga que los niños se pongan de pie y se pongan tontos! Cada niño tendrá un turno para bailar y hacer movimientos divertidos, pero SOLO cuando obtenga el bastón de baile.
 - ¡Enfatice que los niños solo pueden comenzar a bailar una vez que alguien más les entregue el bastón de baile!

Tiempo de ScratchJr

Desafío de estructura:

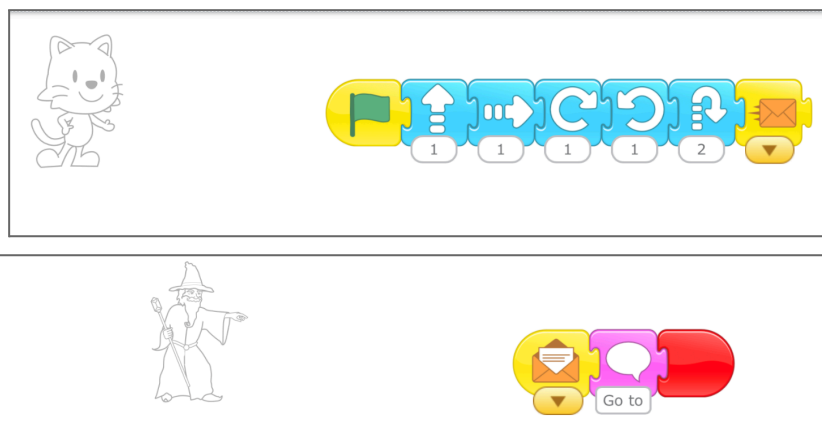
Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) by the [DevTech Research Group](#) is licensed under a [Creative Commons Attribution NonCommercial-ShareAlike 3.0 Unported License](#). Under this license, you may use and adapt this work, but you must attribute the work to the DevTech Research Group. You **may not** use or adapt this work for commercial purposes.

This work is funded by the US Dept of Education (grant #U411C190006) and by the Scratch Foundation.

© 2020, DevTech Research Group.

- **Enviar mensaje** (*Tiempo sugerido: 10 minutos*)

- Al igual que los bastones de las carreras y nuestro bastón de baile, ScratchJr tiene una herramienta que nos permite pasar algo de un programa a otro para decirle que comience. ¡Se llaman bloques de **mensajes**!
- Enviar mensajes: en ScratchJr, los personajes pueden enviarse mensajes entre sí para indicar cuándo iniciar sus programas. El bloque **Iniciar en mensaje** se puede utilizar en lugar del bloque Bandera verde. Vea el programa de ejemplo a continuación:



Verificación de la comprensión de la lección 18: Antes de comenzar un proyecto, verifique que sus alumnos comprendan los nuevos conceptos que acaban de aprender. Lea cada pregunta a los niños y pídales que respondan con un pulgar hacia arriba para decir "sí" o un pulgar hacia abajo para decir "no". Deténgase y vuelva a explicar los conceptos según sea necesario.

Explorando expresiones:

- **Hacer un mensaje** (*Tiempo sugerido: 10 minutos*)

- Los niños hacen su propio proyecto utilizando bloques de mensajes de al menos 2 caracteres.

Cerrando el círculo tecnológico

- **Compartir creaciones** (*Tiempo sugerido: 10 minutos*)

- Los niños comparten sus creaciones con la clase.

Oportunidades de diferenciación

- **Aprendizaje virtual**

- ¡En lugar de un objeto que se pase, elija una palabra divertida! Los niños bailararán y luego dirán la palabra divertida y el nombre de otro niño para pasárselo a ese niño.

Lección 19: Tu proyecto final I

Ideas poderosas de la informática	Algoritmos, proceso de diseño
Ideas poderosas de alfabetización	Secuenciación, proceso de escritura
PTD	Creación de contenido
Paleta de virtudes	Mente abierta, curiosidad
Los niños podrán	<ul style="list-style-type: none"> ● Identificar las escenas inicial, intermedia y final de <i>Dónde están las cosas salvajes</i> ● Planificar los proyectos finales utilizando las indicaciones del diario de diseño ● Comenzar a programar proyectos finales
Vocabulario	<ul style="list-style-type: none"> ● Escenas: una parte de una historia completa
Preparación del maestro	<ul style="list-style-type: none"> ● Leer el plan de la lección. ● Imprima el Diario de diseño de la lección 19 para cada niño o consulte el Diario de diseño completo. ● Tenga el libro Donde viven los monstruos disponible en formato digital o impreso. ● Vea proyectos de ejemplo para familiarizarse con la actividad del proyecto.

Introducción

- **"Donde viven los monstruos"** (Tiempo sugerido: 10 minutos)
 - Lea [Donde viven los monstruos](#) en la clase

Iniciando el círculo tecnológico

- **"Donde viven los monstruos" Eventos** (Tiempo sugerido: 5 minutos)
 - Revise *Donde viven los monstruos*.
 - Recuerde el orden de los eventos en la historia, Dónde viven los monstruos. Muestre ejemplos del libro.
 - Usando 3 escenas de la historia, pregunte a los niños a dónde pertenece cada evento en la historia.
 - "¿Qué sucede al principio, en el medio y al final?"

Tiempo de palabras

- **Planificación del cuento “Donde viven los monstruos”** (*Tiempo sugerido: 10 minutos*)
 - En sus [Diarios de diseño](#), los niños planificarán su historia. La hoja de planificación tiene tres secciones, para el principio, medio y final de la historia con líneas para escribir y espacio para dibujar.

Tiempo de ScratchJr

Explorando expresiones:

- **Programa el cuento Donde viven los monstruos** (*Tiempo sugerido: 15 minutos*)
 - Los niños programarán una escena desde el principio, el medio y el final de la historia, Donde viven los monstruos, para crear un proyecto ScratchJr de 3 páginas usando el bloque Ir a la página.

Cerrando el círculo tecnológico

- **¿Algún problema?** (*Tiempo sugerido: 5 minutos*)
 - Los niños pueden compartir cualquier desafío que puedan tener con el grupo para obtener ayuda.

Lección 20: Tu proyecto final II

Ideas poderosas de la informática	Algoritmos, proceso de diseño, modularidad
Ideas poderosas de alfabetización	Secuenciación, conciencia fonológica
PTD	Creatividad, creación de contenido
Paleta de virtudes	Optimismo, generosidad, gratitud
Los niños podrán	<ul style="list-style-type: none">● Crear programas sincronizados● Copiar un guión de un personaje a otro en ScratchJr● Planificar finales alternativos para una historia
Vocabulario	<ul style="list-style-type: none">● Sincronización: varias personas o personajes haciendo lo mismo al mismo tiempo
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección.● Abra el video del Baile sincronizado en una nueva pestaña y téngalo listo para jugar.

Introducción

- **Pasos uno y dos** (*Tiempo sugerido: 5 minutos*)
 - Los niños verán un video de [robots bailarines sincronizados](#).

Iniciando el círculo tecnológico

- **¿Que es un baile sincronizado?** (*Tiempo sugerido: 5 minutos*)
 - Hable sobre cómo los bailarines tienen que aprender todos los pasos y hacer exactamente los mismos pasos para que sus bailes parezcan **sincronizados**, ¡casi como si tuvieran los mismos bloques en su programa!
 - Discuta utilizando las siguientes indicaciones:
 - ¿Cómo podemos hacer que todos los personajes tengan el mismo programa?
 - ¿Hay alguna parte de “Donde viven los monstruos” que podamos encajar en un baile sincronizado?

Tiempo de ScratchJr

Desafío de estructura:

- **Copiando el guion** (*Tiempo sugerido: 5 minutos*)
 - Introduzca la copia de guiones en ScratchJr.

- Los niños pueden arrastrar un guión de un personaje a otro. ¡Entonces ambos personajes tendrán el mismo guión!



Explorando expresiones:

- **Programa el “El salvaje Rumpus” dentro de su cuento “Donde viven los monstruos”**
(Tiempo sugerido: 20 minutos)
 - Para el proyecto final, los niños usarán el bloque Ir a la página y tendrán 3 páginas para su historia.
 - Para un proyecto final más guiado, haga que los niños programen un Wild Rumpus como su final y haga que todos los personajes bailen el mismo baile usando los guiones de copia.

Cerrando el círculo tecnológico

- **¿Qué pasa si ...?** (Tiempo sugerido: 10 minutos)
 - Como clase, discuta la posibilidad de historias o finales alternativos. Por ejemplo, ¿qué pasa si la historia termina de manera diferente? ¿Y si Max no se iba a casa? ¿Y si Max no extrañaba su hogar?

Lección 21: Tu proyecto final III

Ideas poderosas de la informática	Proceso de diseño, modularidad
Ideas poderosas de alfabetización	Proceso de escritura, conciencia fonológica
PTD	Creación de contenido, creatividad
Paleta de virtudes	Curiosidad, paciencia, optimismo
Los niños podrán	<ul style="list-style-type: none">● Planificar finales alternativos para una historia utilizando indicaciones del diario de diseño● Crear un final alternativo para una historia en sus proyectos ScratchJr
Vocabulario	<ul style="list-style-type: none">● Final alternativo: un posible final de una historia que es diferente a la original.
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección.● Imprima el Diario de diseño de la lección 21 para cada niño o consulte el Diario de diseño completo.● Abra la presentación de diapositivas de preguntas sobre personajes adicionales en una nueva pestaña y téngala lista para presentarla.

Introducción

- **¿Adivina quién?** (*Tiempo sugerido: 5 minutos*)
 - Juega un juego de ¿adivina quién?
 - Si conoce a los niños lo suficientemente bien, elija uno en el aula y dé pistas sobre ellos.
 - O use personajes de uno de los libros o ScratchJr.
 - Por ejemplo, Ada Lovelace: Pequeña y grande.
 - Por ejemplo, el gato naranja, le gusta que lo programen

Iniciando el círculo tecnológico

- **Final alternativo con un personaje extra** (*Tiempo sugerido: 10 minutos*)
 - Revise la discusión de la última clase sobre los finales alternativos. Discuta la idea de qué pasaría si la historia cambiara en estos dos escenarios:
 - "¿Y si hubiera nuevos personajes?"
 - "¿Qué pasa si el final de la historia cambia?"
 - Introducción al proyecto final.

- ¡Los niños cambiarán la última página de sus proyectos creados en la lección 19-20 para crear un nuevo final y agregarán un nuevo invitado adicional para unirse a Wild Rumpus!

Tiempo de palabras

- **Planificación del proyecto final** (*Tiempo sugerido: 10 minutos*)
 - En sus [diarios de diseño](#), los niños planearán cambios en la última página y su carácter adicional.
 - Cambios en la última página: ¿qué podrías cambiar?
 - Nuevo personaje de bonificación: ¿quién debería ser tu invitado de bonificación?

Tiempo de ScratchJr

Explorando expresiones:

- **Programación del proyecto final** (*Tiempo sugerido: 10 minutos*)
 - Los niños programarán para cambiar solo la última página de su historia. Los niños también invitarán a un nuevo personaje de ScratchJr a unirse a Wild Rumpus usando el guión de copia de otro personaje.

Cerrando el círculo tecnológico

- **Preguntas sobre personajes extra** (*Tiempo sugerido: 10 minutos*)
 - Pida a los niños que creen preguntas para saber quién es el personaje de bonificación; utilice el [cuadro de anclaje para las preguntas](#) de carácter de bonificación y guarde las respuestas para la siguiente lección.
 - Por ejemplo, ¿tu personaje adicional es imaginario o real?

Lección 22: Tu proyecto final IV

Ideas poderosas de la informática	Proceso de diseño, depuración, algoritmos
Ideas poderosas de alfabetización	Proceso de escritura
PTD	Creación de contenido, creatividad
Paleta de virtudes	Honestidad, Generosidad
Los niños podrán	<ul style="list-style-type: none"> ● Recordar las habilidades de ScratchJr aprendidas a lo largo del plan de estudios. ● Revisar un proyecto ScratchJr existente agregando un nuevo personaje a “Donde viven los monstruos” ● Brindar y recibir comentarios constructivos de sus compañeros utilizando el mensaje "2 estrellas y un deseo".
Vocabulario	
Preparación del maestro	<ul style="list-style-type: none"> ● Leer el plan de la lección. ● Lea la información de Muestra lo que sabes. ● Imprima una copia para cada niño de la Hoja del estudiante Muestra lo que sabes (disponible para impresión a doble cara aquí: Hoja del estudiante Muestra lo que sabe - Doble cara). ● Abra las diapositivas del profesor Muestra lo que sabes en una nueva pestaña antes de la clase para proyectar a la clase. ● Busque la clave de respuestas Muestra lo que sabes para puntuar los resultados.

¡Muestra lo que sabes! (Tiempo sugerido: 20 minutos)

- Comience la clase con una verificación rápida de los conocimientos de los niños. Esta es la evaluación sumativa del plan de estudios. Reparta una copia de la [Hoja del alumno Muestra lo que sabes](#) a cada niño y proyecte las [diapositivas del maestro](#) en la pizarra. Lea cada pregunta en voz alta y dé a los niños aproximadamente 2 minutos por pregunta. Para obtener más información, consulte la [Hoja de información Muestra lo que sabes](#).

Iniciando el círculo tecnológico

- **Agregando algo más a su proyecto** (Tiempo sugerido: 5 minutos)

- Siempre que los niños terminen con “Muestra lo que sabes” permítales regresar a los proyectos y continuar agregando **detalles** y revisando. Dígales a los niños que la próxima clase será cuando compartan sus proyectos terminados.

Tiempo de ScratchJr

Explorando expresiones:

- **Programación del proyecto final** (*Tiempo sugerido: 15 minutos*)
 - Los niños trabajarán en sus proyectos:
 - Agregando el título de la página
 - Añadiendo un personaje adicional - copiando el guión
 - ¡Añadiendo detalles!

Cerrando el círculo tecnológico

- **2 estrellas y un deseo** (*Tiempo sugerido: 5 minutos*)
 - En parejas o en grupos pequeños, haga que los niños discutan sus proyectos usando el mensaje de retroalimentación “*2 estrellas y un deseo*”:
 - 2 estrellas son dos cosas que salieron bien.
 - 1 Deseo es una cosa que desea hacer mejor.

Lección 23: Nuestra comunidad de programación

Ideas poderosas de la informática	Proceso de diseño, algoritmos
Ideas poderosas de alfabetización	Proceso de escritura
PTD	Colaboración, Community Building, Comunicación
Paleta de virtudes	Gratitud, Generosidad
Los niños podrán	<ul style="list-style-type: none">● Reconocer la ayuda que recibieron y brindaron a otros.● Expresar gratitud a través de una tarjeta de agradecimiento.
Vocabulario	<ul style="list-style-type: none">● Gratitud: estar agradecido
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección● Imprima una copia por niño de la página web de colaboración.● Imprima el Diario de diseño de la lección 23 para cada niño o consulte el Diario de diseño completo

Introducción

- **Historia de una oración** (*Tiempo sugerido: 10 minutos*)
 - El maestro comienza la historia con: "Érase una vez un programador".
 - Los niños y los maestros van de un lado para otro y cada uno agrega una oración a la historia, que finalmente termina con una historia de la clase.

Iniciando el círculo tecnológico

- **Manos amigas** (*Tiempo sugerido: 5 minutos*)
 - Discuta que los niños pueden haberse ayudado unos a otros y necesitaron ayuda durante los proyectos o actividades. Trabajar juntos y apoyarse mutuamente es parte de ser una comunidad de programación y es la forma en que los programadores trabajan juntos. Discuta la importancia de la **gratitud** y cómo es importante mostrar aprecio y agradecer a las personas que lo ayudan. La actividad de introducción de la historia no sería la misma si todos los niños no contribuyesen a la historia.

Tiempo de ScratchJr

Desafío de estructura:

- **Elección del nombre del proyecto** (*Tiempo sugerido: 5 minutos*)

- Invite a los niños a nombrar sus proyectos finales siguiendo los pasos a continuación.

1. Toque el rectángulo amarillo en la esquina superior derecha de la pantalla para nombrar su proyecto y ver la pantalla de información del proyecto.



2. Escriba un nombre específico para este proyecto (p. Ej., "Nombre del niño_nivel de grado"). Vea el ejemplo a continuación:



Tiempo de palabra

- [Web de colaboración](#) (Tiempo sugerido: 10 minutos)
 - Muestre una red de **colaboración** en el salón de clases en la pared con imágenes de los niños y pida a los niños que dibujen líneas o unan hilo a las imágenes de

otros niños que los ayudaron. Los niños deben comenzar a pensar en quién los ayudó con sus proyectos antes de subir a la pizarra.

- **Tarjetas de agradecimiento** (*Tiempo sugerido: 10 minutos*)
 - Los niños se decidirán por dos compañeros que les ayudaron y les escribirán tarjetas de agradecimiento.

Cerrando el círculo tecnológico

- **Enviando gratitud** (*Tiempo sugerido: 5 minutos*)
 - Los niños entregarán tarjetas de agradecimiento a sus compañeros.

Lección 24: Presentación del proyecto final

Ideas poderosas de la informática	Proceso de diseño, algoritmos
Ideas poderosas de alfabetización	Proceso de escritura, secuenciación
PTD	Construcción comunitaria
Paleta de virtudes	Gratitud, Generosidad
Los niños podrán	<ul style="list-style-type: none">● Hablar y expresarse frente a la clase a través de preguntas guiadas de un profesor.● Compartir sus creaciones finales de ScratchJr.
Vocabulario	
Preparación del maestro	<ul style="list-style-type: none">● Leer el plan de la lección.● Imprima el Diario de diseño de la lección 24 o consulte el Diario de diseño completo.

Introducción

- **“Wild Rumpus” baila** (*Tiempo sugerido: 5 minutos*)
 - Como clase, baila como si estuvieras en el Wild Rumpus del libro “Donde viven los monstruos”.

Muestra del Proyecto final

 (*Tiempo sugerido: 20 minutos*)

- Los profesores pueden hacer que los niños envíen sus proyectos finales por correo electrónico y accedan a los proyectos en su iPad/tablet. El maestro compartirá la pantalla de su iPad/tablet con la clase para mostrar el proyecto de cada niño y tener tiempo para que el niño explique su trabajo.
- *Nota: Los profesores pueden grabar en pantalla los proyectos de ScratchJr y subirlos a un powerpoint para compartir la pantalla desde su computadora, si es necesario.*
- Utilice las preguntas guiadas:
 - ¿Quién es tu invitado adicional?
 - ¿Cómo cambiaste el final de la historia?
 - ¿Cuáles son algunos de los bloques que usaste en tu programa?

Tiempo de palabras

- **Diario de diseño de la lección 24** (*Tiempo sugerido: 10 minutos*)

- En sus [Diarios de diseño](#), los niños escribirán una reflexión para responder a las indicaciones:
 - Escribe sobre algo que hayas aprendido ...
 - Escribe sobre tu parte favorita de ScratchJr ...

Cerrando el círculo tecnológico

- **El futuro de la programación** (*Tiempo sugerido: 10 minutos*)
 - Aplauda y felicita a los niños por todo su gran trabajo tanto en sus proyectos finales como a lo largo de esta clase.
 - Discuta el futuro de la programación con toda la clase con estas preguntas guiadas:
 - ¿Qué es lo más lindo que han aprendido en ScratchJr?
 - ¿En qué proyecto les gustaría trabajar a continuación? ¿O qué habilidad querrían aprender a continuación?
 - Mirando más allá de ScratchJr, ¿qué pieza de tecnología querrían ayudar a inventar o programar?
 - Explique que ahora que han terminado de tomar esta clase, ¡son el futuro de la programación! ¡Que a medida que crezcan seguirán aprendiendo cómo pueden tomar las ideas que tienen en el cerebro en este momento y hacerlas realidad!
 - Pida a los niños que canten y bailen una canción de limpieza por última vez.

(con la melodía de Itsy Bitsy Araña)
Apagamos las tabletas y las guardamos
Nos divertimos mucho, pero el día se acabó.
usamos ambas manos y caminamos sin correr
Guardamos las tabletas y la canción se terminó