

Grado 11° Guía 4











Pasar rápido





Estudiantes







MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Julián Molina Gómez Ministro TIC

Luis Eduardo Aguiar Delgadillo Viceministro (e) de Conectividad

Yeimi Carina Murcia Yela Viceministra de Transformación Digital

Óscar Alexander Ballen Cifuentes **Director (e) de Apropiación de TIC**

Alejandro Guzmán **Jefe de la Oficina Asesora de Prensa**

Equipo Técnico Lady Diana Mojica Bautista Cristhiam Fernando Jácome Jiménez Ricardo Cañón Moreno

Consultora experta Heidy Esperanza Gordillo Bogota

BRITISH COUNCIL

Felipe Villar Stein Director de país

Laura Barragán Montaña Directora de programas de Educación, Inglés y Artes

Marianella Ortiz Montes Jefe de Colegios

David Vallejo Acuña Jefe de Implementación Colombia Programa

Equipo operativo

Juanita Camila Ruiz Díaz Bárbara De Castro Nieto Alexandra Ruiz Correa Dayra Maritza Paz Calderón Saúl F. Torres Óscar Daniel Barrios Díaz César Augusto Herrera Lozano Paula Álvarez Peña

Equipo técnico

Alejandro Espinal Duque Ana Lorena Molina Castro Vanesa Abad Rendón Raisa Marcela Ortiz Cardona Juan Camilo Londoño Estrada

Edición y coautoría versiones finales

Alejandro Espinal Duque Ana Lorena Molina Castro Vanesa Abad Rendón Raisa Marcela Ortiz Cardona

Edición Juanita Camila Ruiz Díaz Alexandra Ruiz Correa

British Computer Society – Consultoría internacional

Niel McLean Jefe de Educación

Julia Adamson **Directora Ejecutiva de Educación**

Claire Williams **Coordinadora de Alianzas**

Asociación de facultades de ingeniería - ACOFI

Edición general Mauricio Duque Escobar

Coordinación pedagógica Margarita Gómez Sarmiento Mariana Arboleda Flórez Rafael Amador Rodríguez

Coordinación de producción Harry Luque Camargo

Asesoría estrategia equidad Paola González Valcárcel

Asesoría primera infancia Juana Carrizosa Umaña

Autoría

Arlet Orozco Marbello Harry Luque Camargo Isabella Estrada Reyes Lucio Chávez Mariño Margarita Gómez Sarmiento Mariana Arboleda Flórez Mauricio Duque Escobar Paola González Valcárcel Rafael Amador Rodríguez Rocío Cardona Gómez Saray Piñerez Zambrano Yimzay Molina Ramos

PUNTOAPARTE EDITORES

Diseño, diagramación, ilustración, y revisión de estilo

Impreso por Panamericana Formas e Impresos S.A., Colombia

Material producido para Colombia Programa, en el marco del convenio 1247 de 2023 entre el Ministerio de Tecnologías de la Información y las Comunicaciones y el British Council

Esta obra se encuentra bajo una Licencia Creative Commons Atribución-No Comercial 4.0 Internacional. https:// creativecommons.org/licenses/ by-nc/4.0/

ⓒ (i) (S) CC BY-NC 4.0

"Esta guía corresponde a una versión preliminar en proceso de revisión y ajuste. La versión final actualizada estará disponible en formato digital y puede incluir modificaciones respecto a esta edición"

Prólogo

Estimados educadores, estudiantes y comunidad educativa:

En el Ministerio de Tecnologías de la Información y las Comunicaciones, creemos que la tecnología es una herramienta poderosa para incluir y transformar, mejorando la vida de todos los colombianos. Nos guia una visión de tecnología al servicio de la humanidad, ubicando siempre a las personas en el centro de la educación técnica.

Sabemos que no habrá progreso real si no garantizamos que los avances tecnológicos beneficien a todos, sin dejar a nadie atrás. Por eso, nos hemos propuesto una meta ambiciosa: formar a un millón de personas en habilidades que les permitan no solo adaptarse al futuro, sino construirlo con sus propias manos. Hoy damos un paso fundamental hacia este objetivo con la presentación de las guías de pensamiento computacional, un recurso diseñado para llevar a las aulas herramientas que fomenten la creatividad, el pensamiento crítico y la resolución de problemas.

Estas guías no son solo materiales educativos; son una invitación a imaginar, cuestionar y crear. En un mundo cada vez más impulsado por la inteligencia artificial, desarrollar habilidades como el pensamiento computacional se convierte en la base, en el primer acercamiento para que las y los ciudadanos aprendan a programar y solucionar problemas de forma lógica y estructurada.

Estas guías han sido diseñadas pensando en cada región del país, con actividades accesibles que se adaptan a diferentes contextos, incluyendo aquellos con limitaciones tecnológicas. Esta es una apuesta por la equidad, por cerrar las brechas y asegurar que nadie se quede atrás en la revolución digital. Quiero destacar, además, que son el resultado de un esfuerzo colectivo: más de 2.000 docentes colaboraron en su elaboración, compartiendo sus ideas y experiencias para que este material realmente se ajuste a las necesidades de nuestras aulas. Además, con el apoyo del British Council y su red de expertos internacionales, hemos integrado prácticas globales de excelencia adaptadas a nuestra realidad nacional.

Hoy presentamos un recurso innovador y de alta calidad, diseñado en línea con las orientaciones curriculares del Ministerio de Educación Nacional. Cada página de estas guias invita a transformar las aulas en espacios participativos, creativos y, sobre todo, en ambientes donde las y los estudiantes puedan desafiar estereotipos y explorar nuevas formas de pensar.

Trabajemos juntos para garantizar que cada estudiante, sin ·importar dónde se encuentre, tenga acceso a las herramientas necesarias para imaginar y construir un futuro en el que todos seamos protagonistas del cambio. Porque la tecnología debe ser un instrumento de justicia social, y estamos comprometidos a que las herramientas digitales ayuden a cerrar brechas sociales y económicas, garantizando oportunidades para todos.

Con estas guias, reafirmamos nuestro compromiso con la democratización de las tecnologías y el desarrollo rural, porque creemos en el potencial de cada región y en la capacidad de nuestras comunidades para liderar el cambio.

Julián Molina Gómez Ministro de Tecnologías de la Información y las Comunicaciones Gobierno de Colombia



Guía de íconos



Guía 4

Lógica, programación y depuración) Prácticas de datos Computación física

Aprendizajes de la guía

Con las actividades de esta guía se espera que puedas avanzar en:



Usar la entrada (datos o señales) para producir una salida (acciones o respuestas).



Programar máquinas de estados, como semáforos, utilizando la *micro:bit* para manejar diferentes transiciones secuenciales en un sistema.

Resumen de la guía

Al finalizar esta guía de 5 sesiones de trabajo, se habrá aprendido sobre el concepto de retroalimentación y su aplicación, este concepto es esencial para la creación de sistemas automatizados. A través de diversas actividades, se avanzará en el uso de la *micro:bit* en el entorno *Tinkercad* y su programación para controlar diferentes dispositivos, desde semáforos para mejorar el tráfico en la ciudad hasta sistemas de alarma para la seguridad del hogar. Estos conocimientos preparan para entender y contribuir al diseño de ciudades más inteligentes y ecológicas.

Resumen de las sesiones

Sesión 1

Se explorarán los conceptos de entrada, salida y retroalimentación al simular un sistema de puertas automáticas que responderá a la presencia de personas, usando una actividad desconectada en grupo.

Sesión 2

Se aprenderá a programar un semáforo con la *micro:bit* usando LED, entendiendo cómo funcionan las máquinas de estados y la programación secuencial para mejorar el flujo de tráfico en la ciudad del futuro.

Aprendizajes de la guía

Implementar contadores y acumuladores en la programación para realizar seguimientos de eventos, como un conteo en un semáforo para carreras de velocidad.



Utilizar la variación de voltaje para controlar actuadores como motores y LED.

ġ

Integrar sensores (como sensor de proximidad o PIR y fotorresistencias) con actuadores (como motores y parlantes) para crear sistemas automatizados.

Sesión 3

Se utilizarán condicionales anidadas en bucles, contadores y acumuladores para reprogramar el semáforo a uno de carreras adaptando su funcionamiento a diferentes situaciones y necesidades.

Sesión 4

Diseñarán un sistema de alarma de seguridad con un sensor PIR, un piezoeléctrico y un LED, y explorarán cómo ajustar el sonido y la luz mediante la variación de voltaje.

Sesión 5

Se aprenderá a controlar la velocidad de un motor DC con una fotorresistencia y un transistor TIpin 120, aprendiendo a utilizar la función MAP y sensores para optimizar la eficiencia energética.

Si se requiere

- La Guía 1 del grado 5 introduce el manejo del lenguaje de programación en bloques MakeCode.
- La Guía 2 del grado 7 introduce el manejo de arreglos en programación.
- C La Guía 3 del grado 6 introduce el manejo de variables en MαkeCode.
- La Guía del grado 10 introduce el manejo de micro:bit en el entorno Tinkercad.

Sobre los materiales

Esta unidad trabaja sobre computación física, lo cual implica aprender a interactuar, desde un programa de computador, con el entorno usando sensores y actuadores. Existen dos opciones para lograr este objetivo:

- En caso de contar con los materiales: implementar dispositivos físicos reales, incluyendo la *micro:bit* junto con diversos sensores y actuadores, para proporcionar experiencias reales. Esto requerirá la disponibilidad de los materiales solicitados para cada práctica.
- O En caso de no contar con los materiales: explorar la computación física mediante el simulador *Tinkercad* que ofrece una alternativa al no depender de componentes electrónicos reales. Ten en cuenta que el foco de la guía no es aprender de circuitos eléctricos sino de computación física.

En esta guía se optó por la segunda alternativa. Sin embargo, en un contexto donde se cuente con recursos suficientes y tengas aprendizajes previos, también podría implementarse la primera opción. Además, si tu docente desea profundizar, se recomienda realizar una planificación previa utilizando el simulador. Esto permitirá familiarizarse con las herramientas y el contenido, facilitando la orientación en clase y asegurando un uso efectivo del simulador para desarrollar las competencias en computación física.





Conexión con otras áreas

A continuación se presentan algunos puntos de conexión con otras áreas:

Ciencias Naturales

ဆိုနိ

 La guía promueve aprendizajes relacionados con circuitos eléctricos, esto te brinda una base para comprender algunos principios físicos fundamentales detrás de las conexiones.

Matemáticas

La guía permite aplicar conceptos como O proporciones y ecuaciones, promoviendo el pensamiento lógico-matemático.

Ingeniería

La interacción con sensores, actuadores y O microcontroladores introduce conceptos básicos de diseño e implementación de sistemas automatizados, brindando una perspectiva inicial sobre la robótica y la ingeniería de control.

Ciencias Sociales

 Las discusiones sobre el impacto de la
 automatización y los sistemas inteligentes en la sociedad promueven una visión crítica sobre la tecnología y sus aplicaciones éticas. Grado 11º Guía 4



Sesión 1

Aprendizajes esperados

Al final de esta sesión verifica que puedas:



Entender los conceptos de entrada y salida en la realimentación de sistemas automatizados.

Reconocer cómo la información de entrada se procesa para generar acciones de salida.

Identificar cómo la realimentación permite ajustar y mejorar el comportamiento del sistema automatizado.

Material para la clase

- O Objetos o fichas hechas a mano que representen personas.
- Anexo 1.1, Anexo 1.2 con los 2 vehículos recortados y el plano del peaje, Anexo 1.3 con las instrucciones de la actividad.

Duración sugerida



15% 70%

15%







Anexo

Anexo 1.1

Aneso 11 Reto

Los alternas unalimentados sos comunos en com los hos neneros. Con hádingia, los algungia chan es ho homenatania la conpositiva de los seres relovanos mantenes conclusiones de arma estables a parar de los cambions en sus internas, como la preseria de los cambions en sus internas. No ejumpio, el sudo, com os desares en la mangen de un outro autócnos, en una espanta de los campo hamas esta tempenaturas altas, por altando la unificipación y al manteritrivinto, por altando la unificipación y al manteritrivintos, por alternas han tempenaturas.



digital incorporan sistemas realimentados, disrilados para responder de manera autónoma a cambios en su entorno. Un ejempio cotidiano es el verdilador de un computador, que se enclende autonáticamente al detectar un aumento en la temperatura interna, protegiendo así los componentes de aletroinem de debe.

Esta minera lógica se aplica a tecnologias más avanzadas. Un don viltzado en aplicidaras anaturable de precisión opera con múltiplas atternas realmentados para realizar tanana complejas de manera eficiente. Lo nómos cetas autorabitas, autornolísas, autoreanos, poestas autorabitas, autornolísas, parte en alternas los desarras realizantes para realizados los desarras realizantes aos esenciales para los attes tencionados los desarciales y a der los atestas tencionados los desarciales y a der los atestas tencionados los desarciales y a der desartes tencionados los desarcos y segar de antes tencionados estas tencionados y segar de antes tencionados estas desarciales para estas tencionados estas desarciales para de antes tencionados estas desarciales de antes desarciales para de antes tencionados estas desarciales de antes desarciales para de antes tencionados estas desarciales de antes desarciales para de antes tencionados de antes desarciales de antes de antes desarciales de antes desarciales de antes de ant



En este contexto, el redo que se perpone es simular un sistema automaticado similar al que reglas el país de vahíciose com o dejo TCA de la perpeis electricitoria. En este sistema, la falançama se activa automáticamente consola con servico detecta al TCA, permitiendo al paro del vahícios. Es un validada no construito TCA, la taliangama no as enclas, ingeliados su anone. Esta és queició barida la spontambidad de esparimentar de manara práctica con los alternas inselmentados aplemismas porserios en situaciones el mande en del contrara se enclas seguidadas y quiebabidads.

Lo que sabemos,

lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

En esta primera sesión se establecerán las bases para convertirnos en arquitectos(as) de la ciudad del futuro, una ciudad donde la tecnología podría trabajar para el bienestar de sus habitantes. Comenzaremos explorando cómo los conceptos de entrada, salida y realimentación son esenciales para el diseño de sistemas inteligentes. Mediante una actividad desconectada simularemos el funcionamiento de las talanqueras automáticas en los peajes de la ciudad del futuro, analizando cómo responden a la presencia de vehículos equipados con un chip o TAG para optimizar la velocidad y seguridad en el paso de los usuarios. Este será nuestro primer paso hacia la creación de una ciudad más eficiente e interconectada.

Empecemos por explorar el reto de esta guía, el cual se presenta en el *Anexo 1.1.*

22

¿Cómo podríamos aplicar lo que ya sabemos sobre sensores y actuadores para diseñar el sistema automatizado del peaje?

¿Qué conceptos o habilidades adicionales necesitamos aprender para programar la detección del TAG y el control de la talanquera con la micro:bit?

No dudes en construir un esquema que refleje lo que entiendes y lo que te falta por aprender para resolverlo.

Hasta ahora sabemos que muchos sistemas en nuestro entorno diario funcionan automáticamente, sin necesidad de intervención humana constante. Sabemos que las puertas automáticas de los centros comerciales se abren cuando alguien se acerca y se cierran después de que la persona ha pasado. También hemos observado que para que estos sistemas funcionen correctamente, deben "saber" cuándo actuar y cómo responder a diferentes situaciones. **Figura 2.** Dispositivo TAG pegado a la parte interna del vidrio panorámico del vehículo



Esto nos lleva a comprender que las máquinas y dispositivos requieren algún tipo de información para decidir cuándo realizar una acción, es decir, lo que llamamos "entrada" y "salida".

En esta sesión aprenderás qué partes tienen estos sistemas automáticos e identificarás sus entradas, sus salidas y la realimentación.

En un sistema automático, la **entrada** se define como lo que permite recoger la información que un sistema automático necesita para tomar decisiones.

Para este ejercicio vas a simular en grupos un sistema de peaje automático como el que se encuentra en muchas carreteras.

Figura 1. Lectura de TAG en peaje



Cuando un vehículo se aproxima teniendo pegado en su vidrio delantero un código denominado TAG, *Figura 2*, el lector del peaje lo lee y procede a levantar la talanquera, ver *Figura 1*. Si el vehículo no dispone de ese dispositivo, la talanquera no se levantará. Además, la talanquera no debe bajarse hasta que el vehículo haya pasado para no dañarlo.

La **entrada** del sistema, en este caso, sería la detección del dispositivo en el vidrio delantero del vehículo.



Anexo 1.3



La **salida** es la orden para que la talanquera abra o cierre y la **realimentación** permite saber si el vehículo ya pasó o no. De hecho, existe una segunda realimentación y es saber si la talanquera abrió o cerró una vez el sistema dé la orden correspondiente.

En esta actividad descubrirás cómo funcionan los conceptos de entrada, salida y realimentación simulando un sistema de talanqueras en un peaje que responde de forma automática cuando un vehículo con el TAG válido se acerca.

Imaginemos que estamos en un peaje y una talanquera se abre automáticamente al detectar la presencia de un TAG en un vehículo. Ustedes jugarán diferentes roles para ver cómo una simple señal de entrada puede hacer que la talanquera se abra (salida) y cómo el sistema verifica que esté realmente se abrió o cerró (realimentación). Al final de esta actividad, entenderán mejor cómo estas ideas se utilizan para diseñar y mejorar sistemas automatizados.

Usando los dos vehículos recortados del Anexo 1.2, que muestra también un plano de un peaje, simularás con tu grupo el funcionamiento de la talanquera siguiendo las instrucciones dadas en el Anexo 1.3.

Glosario

- Entrada: la información que el sistema recibe para iniciar su operación. En el caso de la talanquera del peaje, la entrada es la detección de un TAG o chip del vehículo por medio de un sensor, como se ve en la Figura 1.
- **Salida:** la acción que el sistema ejecuta en respuesta a la entrada recibida. En este caso, la salida es la activación o no de la talanquera, permitiendo o impidiendo el paso del vehículo.
- 5Å

Realimentación: la información que verifica si el sistema funcionó según lo esperado. En el ejemplo del peaje, la realimentación puede ser la confirmación de que la talanquera se abrió correctamente cuando el TAG fue detectado o que permaneció cerrada cuando no se detectó ningún TAG. Anexos

Anexo 1.2







Manos a la obra

Desconectadas



Organízate en grupo con otras 4 personas. Sigue las instrucciones de tu docente.

Realicen la actividad siguiendo las instrucciones del Anexo 1.3. En los roles hay estudiantes que podrán ver y otros que no. Quienes no puedan ver deben cerrar los ojos o cubrirlos con una venda.

Realicen la actividad al menos cinco veces, asegurándose de que cada estudiante participe en cada rol al menos una vez.

Al terminar, discutan sobre las siguientes situaciones. Piensen en formas de resolverlas:

- \bigcirc La talanquera no abre cuando se le ordena.
- O La talanquera no cierra cuando se le ordena.
- O El lector del TAG no funciona.
- El sensor de paso del vehículo avisa antes de que termine de pasar el vehículo.
- El vehículo decide dar reversa y no pasar después de que el lector ya avisó que había llegado un vehículo con TAG.

Traten de encontrar soluciones y, si tienen tiempo, traten de simularlas cambiando los mensajes y el algoritmo del computador.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes de la sesión de forma individual respondiendo las preguntas de forma que mejor reflejen tu progreso:



¿Puedes entender los conceptos de entrada, salida y realimentación en sistemas automatizados?

Sí
 Parcialmente
 Aún no

¿Puedes reconocer cómo la información de entrada se procesa para generar acciones de salida?
Sí
Parcialmente
Aún no
¿Puedes identificar cómo la realimentación permite ajustar y mejorar el comportamiento del sistema automatizado?
Sí
Parcialmente
Aún no

Sí si tus respuestas fueron "Parcialmente" o "Aún no", regresa a los puntos de la actividad relacionados con entrada, salida

a los puntos de la actividad relacionados con entrada, salida y realimentación. Si después de esto todavía tienes dudas, acude a tu docente por apoyo adicional.

Las preguntas que te proponemos ahora para pensarlas en grupo y luego compartir las respuestas con el salón:

ŝŝ	
	l
	l

Si regresamos al reto, ¿para qué serviría lo que hemos aprendido en esta sesión?

¿Se te viene a la mente otro sistema automático que puedas describir en sus componentes de funcionamiento?

Utiliza este espacio final para elaborar un esquema que resuma parte de lo aprendido, incluyendo, por ejemplo, definiciones de términos clave y ejemplos del uso de los conceptos trabajados en la actividad desconectada.





Sesión 2

Aprendizajes esperados

Al final de esta sesión verifica que puedas:



Aplicar bloques de programación secuencial para el control de máquinas de estados.



Realizar conexiones entre la *micro:bit* y actuadores externos (LED) en el entorno *Tinkercad*.



Simular código de programación y montajes electrónicos con la *micro:bit* desde el entorno *Tinkercad*.

Material para la clase

- Acceso a Tinkercad cuenta de estudiante o personal.
- O Anexo 2.1.

Ð

Educación

Duración sugerida





15%



Sesión 2 Estudiantes

Figura 1. Semáforo

Lo que sabemos,

lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Ahora que entendemos cómo funcionan los sistemas automatizados, es hora de mejorar el flujo de tránsito en la ciudad del futuro. En esta sesión aprenderemos a programar un semáforo utilizando la *micro:bit* y algunos LED, para garantizar que tanto los vehículos como los peatones se muevan de manera segura y ordenada. Cada semáforo actuará como un guardián que previene accidentes y atascos, haciendo nuestras calles más seguras para todos los ciudadanos(as).

Un semáforo es un ejemplo de **máquina de estados**, es decir, un dispositivo cuyo comportamiento se define por varios estados, transiciones entre estos y acciones que se llevan a cabo en respuesta a ciertos eventos o condiciones. Por ejemplo, una lavadora es una máquina de estados, ya que inicia con el llenado, luego pasa al lavado, continúa con el enjuague y, finalmente, al secado. De manera similar, un semáforo funciona como una máquina de estados, y sus estados y transiciones pueden describirse de la siguiente manera, ver *Figura 1*:

O Estado 1

luz roja prendida, luz amarilla apagada, luz verde apagada (5 segundos).

O Estado 2

luz roja prendida, luz amarilla prendida, luz verde apagada (2 segundo).

O Estado 3

luz roja apagada, luz amarilla apagada, luz verde encendida (5 segundos).

O Estado 4

luz roja apagada, luz amarilla encendida, luz verde apagada (2 segundos).

		Tinker ~ Gale	ría Proyectos	Aulas	Recursos ~	۹ <table-cell></table-cell>
0	E C	41	Ent	ernow		
María						+ Crear
۱	🕅 Diseños 3D					Diseño 3D
Ver distintivos						Circuitos
fà Inicio						Bloques de código
💍 Clases						
🗄 Diseños	Surprising Gaaris					

Figura 2. Opción de crear circuitos en Tinkercad

Iniciemos con la programación en *Tinkercad* de un semáforo clásico de 4 estados, para lo cual primero debes dirigirte a www.tinkercad.com y abrir tu cuenta, una vez allí haz clic en el botón Crear y luego en la opción Circuitos como se muestra en la *Figura 2*.

Con base a tus experiencias anteriores con *Tinkercad*, en este simulador encontrarás una gran zona gris a la que llamaremos "Lienzo" y una barra vertical a la izquierda donde encontrarás los componentes. Dicha barra cuenta con una opción de filtro y un buscador de componentes para facilitar encontrar el componente requerido.

Glosario

- Máquina de estados: un sistema con estados finitos, transiciones y acciones, cambiando entre estados según condiciones o eventos. Por ejemplo, un semáforo o una lavadora.
- Actuadores: dispositivos que transforman la energía en movimiento, luz o cualquier otra acción física, permitiendo que un sistema automatizado interactúe con su entorno. Por ejemplo, motores, LED, parlantes, entre otros.
 - **Pin GND:** conector en la *micro:bit* que sirve como referencia de polo negativo para completar los circuitos eléctricos.

Figura 4. Conexión de resistencia a pin 0



Manos a la obra

Conectadas



Esta sección corresponde al 85% de avance de la sesión

Sigue los pasos para hacer este montaje y simulación en el entorno *Tinkercad*:

Paso 1: busca entre los componentes la *micro:bit*, haz clic sostenido sobre ella y llévala al lienzo como se muestra en la *Figur*α 3.

Figura 3. Seleccionar *micro:bit* en la lista de componentes en *Tinkercad*



Paso 2: busca entre los componentes la "Resistencia" y conecta al pin 0 tal como lo muestra la imagen *Figura 4.*

En caso de que tengas la *micro:bit*, cuentes con los componentes y quieras hacer montajes físicos, puedes usar cables caimán, tal como se muestra en la *Figura 5*.

O simplemente pela los extremos de un cable conductor y amarra las partes descubiertas a las terminales. No te preocupes por la electricidad en estos montajes, la corriente y el voltaje son demasiado pequeños para hacerte daño.

Para efectos de esta guía no usaremos cables-caimán en Tinkercad.

Figura 5. Conexión de resistencia con caimanes



Grado 11º Guía 4

Figura 6. Conexión de led a resistencia



Figura 7. Cátodo del LED



Figura 8. Opción de cambio de color de cableado





Paso 3: busca entre los componentes el LED, arrástralo al lienzo y hazlo girar oprimiendo la tecla "R" o dando clic sobre la herramienta.

Cuando esté girado conéctalo con la resistencia, tal como lo muestra la *Figura* 6.

Paso 4: haz un clic sostenido en la terminal del LED llamada "cátodo" como en la Figura 7 y extiende el cable hasta el pin GND de la *micro:bit.* Puedes cambiar el color del cable dando clic sobre el cuadrado de colores, ver Figura 8.

Paso 5: siguiendo el mismo procedimiento realiza el montaje de los otros dos LED, puedes cambiar el color del led en las opciones emergentes que aparecen cuando das clic sobre alguno de ellos, ver *Figura* 9.

Figura 9. Conexiones de LEDS a los pines del micro:bit



Una vez terminado el montaje vamos a iniciar con la programación, para lo cuál debes hacer clic en la opción "Código".

Una vez en la zona de programación encontrarás una caneca donde puedes desechar los bloques que no vas a utilizar, *Figura 10*.



Figura 10. Opción para desechar los bloques



Al igual que en *Scratch*, encontrarás los bloques con los mismos colores de la paleta a la que pertenecen. Igualmente, los bloques se encajan uno debajo o dentro del otro.

Paso 6: busca en la paleta "Output" o "Salida" la opción pin O-HIGH y conéctalo dentro del bucle "Forever" o "Para siempre". Esta opción hará que nuestro LED rojo se encienda todo el tiempo. Puedes comprobarlo haciendo clic sobre el botón como se ve en la *Figura 11*.

Figura 11. Bloque con instrucciones "Output"



Paso 7: detén la simulación y programa el primer estado del semáforo, rojo conectado a pin 0 en HIGH o encendido, amarillo conectado a pin 1 en LOW o apagado y verde conectado a pin 2 también en LOW. Coloca el tiempo de espera para cada uno de los estados, ver *Figura* 12.



Figura 12. Programación de escrituras digitales en pines



Paso 8: programa los otros tres estados restantes teniendo en cuenta que no todos tienen los mismos tiempos. Simula el montaje con el código terminado y observa su comportamiento.

El código completo lo puedes observar en el Anexo 2.1.

¿Funciona cada uno de los cuatro estados? ¿Qué tendrías que hacer si quisiéramos hacer que dure dos veces más el semáforo en rojo que en verde?

Observa que este semáforo se programa de manera secuencial, un estado debajo del otro, cuando acaba uno inicia el otro.

Antes de irnos

22



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes de la sesión. Elije la opción de respuesta que mejor describa lo que alcanzaste.

 ¿Puedes reconocer el entorno de simulación de circuitos electrónicos Tinkercad?

Sí
 Parcialmente
 Aún no

¿Puedes conectar LEDs a la micro:bit y encenderlos y apagarlos de forma secuencial?
 Sí
 Parcialmente
 Aún no
 ¿Puedes simular montajes electrónicos programados en micro:bit desde el entorno Tinkercad?
 Sí
 Parcialmente
 Aún no

En caso que tus respuestas a hayan sido "Parcialmente" o "Aún no", discute con tu compañera o compañero de grupo el trabajo realizado sobre la simulación realizada en Tinkercad, verifica la solución que dieron a uno de los ejercicios planteados y consulta con tu docente las preguntas que aún tengas.

Las preguntas que te proponemos ahora pensar las siguientes preguntas:

ŝŝ

Si regresamos al reto, ¿para qué serviría lo que hemos aprendido en esta sesión?

Algunos semáforos son inteligentes, dado que pueden saber si hay vehículos o no esperando. ¿Podrías indicar en palabras lo que habría que hacer para que este programa funcione?

Utiliza este espacio final para crear un esquema que resuma lo aprendido durante la actividad conectada, incluyendo definiciones de términos clave y ejemplos del uso de los conceptos aplicados al programar el semáforo.





Sesión 3

Aprendizajes esperados

Al final de esta sesión verifica que puedas:



Usar contadores en la programación de temporizadores.

Usar acumuladores que permitan, a partir de conteos, filtrar datos para la ejecución de procesos en un programa.



Automatizar máquinas de estados a partir de contadores y acumuladores dentro de condicionales y bucles anidados.

Material para la clase

O Tinkercad - cuenta de estudiante o personal.

Duración sugerida



15% 70% 15%







Lo que sabemos,

lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

En la ciudad del futuro también habrá eventos emocionantes como carreras de velocidad. Usando el mismo semáforo que programamos anteriormente, aprenderemos a adaptarlo para que funcione como un semáforo de carreras, contando los segundos hasta dar la señal de salida. Con la ayuda de contadores y acumuladores, veremos cómo modificar y personalizar los sistemas para adaptarse a diferentes necesidades, demostrando así la versatilidad y adaptabilidad de la tecnología en nuestra ciudad ideal.

Haremos una variación del semáforo creado en la sesión pasada, para lo cual se recomienda realizar un nuevo montaje en *Tinkercad*.



¿Has notado que en las carreras de coches hay semáforos que indican a los corredores cuándo es el momento de partir?

Estos semáforos no funcionan con los cuatro estados que encontramos en las calles de la ciudad; en cambio, operan con tres estados: comienzan con la luz roja encendida, luego se enciende la luz amarilla, y finalmente, en un tercer estado, se enciende la luz verde, señalando el inicio de la carrera. Vamos a programar este tipo de semáforo usando condicionales dentro de bucles y un par de variables (contador y acumulador) que nos ayudarán a llevar un conteo en el arreglo de LED de la *micro:bit*.

Los **contadores** y **acumuladores** son conceptos de programación utilizados para realizar un seguimiento de operaciones o condiciones dentro de un programa. Los **contadores** son variables que se usan para contar el número de veces que ocurre un evento específico; se incrementan o aumentan (o, en algunos casos, se decrementan o disminuyen) cada vez que se produce dicho evento. Los **acumuladores**, por otro lado, son variables que se emplean para sumar (o acumular) valores durante la ejecución de un programa.

A diferencia de los contadores, que solo cuentan, los acumuladores mantienen un total al que se agregan valores continuamente.

Piensa en el caso de una caja registradora en un supermercado: cada vez que un artículo pasa por el escáner, se incrementa un contador que lleva el número total de artículos comprados. Al mismo tiempo, el precio de cada artículo se suma a un acumulador que mantiene el total a pagar.



Figura 1. Montaje de semaforo



Figura 2. Programación de variables en bloque "On Start"



Figura 3. Bloque de repetición en bloque "Forever"



Manos a la obra

Conectadas



Esta sección corresponde al 85% de avance de la sesión

Sigue los pasos teniendo en cuenta que vamos a usar un montaje electrónico idéntico al del primer semáforo, pero esta vez nos enfocaremos en automatizar los estados del semáforo de carreras usando un contador y un acumulador dentro de bucles y condicionales anidadas.

Paso 1: haz exactamente el mismo montaje del semáforo que creaste en la práctica anterior, ver *Figura 1.*

Paso 2: en la zona de código busca la paleta "Variables" y crea las variables "acumulador" y "contador". Conéctalas a la función "On start" o "Al empezar" y déjalas en cero como puede ver en la *Figura 2*.

Con la variable contador llevaremos un conteo de 0 a 3 en el **display o matriz de LED** de la *micro:bit* que indicará el cambio de estado del semáforo, mientras que con el **acumulador** vamos a ir sumando los conteos, por esta razón ambas variables deben iniciar en cero.

Paso 3: en la paleta control busca la opción "repeat while (true)" o "repetir mientras (verdadero)", anida este bucle con el bucle "forever" o "para siempre", ver *Figura 3*.

Paso 4: ahora busca en las paletas "Math" y "Variables" y completa la sentencia condicional "repetir mientras que contador sea menor o igual que 3", conéctala en "true" o verdadero, ver *Figura 4*.

Figura 4. Programación condicional con operadores



Paso 5: anida una condicional doble al bucle "while" y como sentencia condicional pregunta si el acumulador es mayor que 3. En caso de ser cierto pin 0 donde está conectado el LED rojo se encenderá, en caso contrario permanecerá apagado o en "LOW", ver *Figurα 5*.

Figura 5. Programación completa de condicional



Paso 6: crea otra condicional donde se encienda el LED amarillo conectado a pin 1 cuando el acumulador sea mayor a 6. Conéctala dentro del ciclo "while" o mientras, debajo de la primera condicional , como se observa en la *Figura 6*.

Figura 6. Programación de otro condicional



Paso 7: vuelve a crear otra condicional donde preguntes si el acumulador es mayor a 9, en caso de ser cierto activará en HIGH a pin 2 donde se encuentra conectado el LED verde. Conéctala debajo de las otras dos condicionales.



Figura 7. Programación de otro condicional

Paso 8: crea una nueva condicional doble donde preguntes si el acumulador es menor que 9. En caso de que sea cierto, mostrará el valor de la variable contador de 0 a 3 en el display o matriz de LED de la *micro:bit* y sumará un 1 al contador. En caso contrario, se mostrará un ícono de chulito en el display o matriz de LED de la *micro:bit*.

Figura 8. Creación de condicional doble



Figura 10. Filas y Columnas en el display o matriz de LED de la *micro:bit* **Paso 9:** fuera del ciclo "while" o "mientras que" asigna al acumulador la suma de sí mismo más el contador y posteriormente reinicia el contador a cero.



Figura 9. Suma de variable y reinicio de contador a a cero

sh	ow numbe	contad	or			
cha	ange co	tador 🔻	by 1			
lse						
sho	w icon					
						ۍ
et	acumul	ador 🔻	to acum	ulador	(+ 🔻	contador
et	contad	or 🔻 to	0			

Simula el semáforo y notarás que el contador no supera el número 3 iniciando con el cero, debido a que fue declarado con cero desde la función "on start" o "al iniciar" y posteriormente cada vez que alcanza el valor de 3 se sale del ciclo "while" y vuelve a retomar el valor de cero.

ŝ

¿Qué relación encuentras entre el contador y el acumulador?

¿Qué ocurriría con el programa si el contador o el acumulador se inician con un número diferente a cero?

Para ir más lejos

Los **bucles anidados** son usados en nuestra cotidianidad pues se trata de repeticiones que dependen de otras. Un ejemplo de estos bucles anidados es nuestra percepción y medición del tiempo. Piensa en un reloj. Podemos decir que la unidad básica de tiempo son los segundos, que al irse contando desde cero y llegar hasta 59 hacen el incremento de los minutos en un minuto, tras lo cual los segundos se reinician. Igualmente ocurre cuando se completan 60 minutos, pues se reinician los minutos a 0 y se incrementan en 1 las horas. También podríamos seguir anidando los ciclos del tiempo y decir que al completarse 24 horas se incrementan en 1 los días, al llegar a 30 o 31 días se incrementan los meses y al completar 12 meses se incrementan los años.

Tu reto consiste en utilizar los bucles anidados y los contadores para rellenar gradualmente el arreglo de LED de la *micro:bit*. Para solucionarlo, crea dos variables "Fila" y "Columna" para ir llenando el display o matriz de LED de la *micro:bit Figura 10*, usa la *Figura 11* como pista para completar el código.



Figura 11. Bucles anidados y contadores

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes de la sesión de forma individual respondiendo las preguntas de forma que mejor reflejen tu progreso:

¿Puedes aplicar contadores para el registro de veces que se ejecuta un evento en un programa?

- O Sí
-) Parcialmente
- 🔵 🛛 Aún no

2 ¿Puedes acumuladores para totalizar la cantidad de conteos y discriminar con condicionales que nuevos eventos ejecutar de acuerdo con el total acumulado?

- 🔾 Sí
-) Parcialmente
- 🔵 Aún no

 ¿Puedes usar condicionales, bucles anidados, contadores y acumuladores para temporizar acciones que se ejecutan desde la micro:bit?

-) Sí
-) Parcialmente
-) Aún no

Si tus respuestas fueron "Parcialmente" o "Aún no", discute con tu compañera o compañero de grupo el trabajo realizado relacionado con los bucles anidados, verifica la solución que dieron a uno de los ejercicios planteados y consulta con tu docente las preguntas que aún tengas. Ahora te proponemos la siguiente pregunta



Si regresamos al reto, ¿para qué serviría lo que hemos aprendido en esta sesión?

Imaginemos un caso de la vida cotidiana.

Imagina que estás diseñando un semáforo para una ciudad. Considerando lo aprendido sobre contadores, acumuladores y bucles anidados, ¿cómo lo utilizarías para optimizar el tiempo de cambio de luces en función del flujo de tráfico?

Discute brevemente cómo adaptarías el código para ajustar automáticamente los tiempos de cada luz en función de la cantidad de vehículos detectados.



Aprovecha este espacio final para hacerte un esquema en que resumas algo de lo que aprendiste, por ejemplo, colocando las definiciones de palabras y algún ejemplo de uso de los tipos de variables **contador** y **acumulador** en **bucles anidados**.







Aprendizajes esperados

Al final de esta sesión verifica que puedas:



Aplicar lecturas digitales para producir cambios en salidas analógicas.

Aplicar cambios graduales en actuadores, usando la escritura analógica.

Conectar correctamente sensores externos a la *micro:bit.*

Material para la clase

O Tinkercad - cuenta de estudiante o personal.

Duración sugerida



15%

70%

15%









Lo que sabemos,

lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

La seguridad es una prioridad en la ciudad del futuro, por lo que en esta sesión diseñaremos un sistema de alarma utilizando un **sensor de movimiento PIR**, un **piezo o parlante** y un LED. Este sistema de seguridad detectará intrusos y nos alertará con una alarma sonora. Al aprender sobre la variación de voltaje y la escritura analógica, podremos ajustar el tono y el volumen de la alarma, haciendo que nuestros hogares y edificios sean más seguros. Este proyecto demostrará cómo la tecnología puede protegernos y brindarnos tranquilidad en nuestra vida diaria.

En el mundo de la automatización, los sensores, actuadores y las lecturas y escrituras analógicas se integran para crear sistemas eficientes y precisos. Los sensores recopilan datos del entorno como la temperatura o la luminosidad y estos datos suelen ser señales analógicas que representan valores físicos variables. Las lecturas analógicas se convierten en datos digitales que la *Micro:bit* puede procesar, mientras que la escritura analógica se utiliza para controlar actuadores, como motores o luces LED, ajustando la energía suministrada de manera eficiente.

La **lectura analógica** se refiere a la capacidad de la *micro:bit* para medir voltajes variables en sus pines de entrada, a través de sensores. Estas lecturas permiten captar información del voltaje cambiante, a diferencia de las lecturas digitales, que solo detectan estados de encendido o apagado (HIGH o LOW). Por otro lado, la **escritura analógica** se refiere a la capacidad de la *micro:bit* para generar señales de salida de voltaje variable en sus pines, lo que permite modificar la acción de un actuador, como aumentar o disminuir la velocidad de un motor, el brillo de una bombilla o como en este proyecto cambiar el tono de sonido de un parlante o **piezo**.

Un ejemplo excelente de esta integración es un sistema de climatización inteligente en un hogar. Un sensor de temperatura mide continuamente la temperatura ambiente y estos datos se leen como señales analógicas, que luego se convierten en valores digitales para controlar la velocidad del ventilador y la potencia del compresor en un aire acondicionado. Si la temperatura es muy alta, el sistema incrementa la velocidad del ventilador y la capacidad de enfriamiento del compresor para enfriar la habitación más rápidamente. Este sistema asegura un control preciso y eficiente del clima, manteniendo el confort y optimizando el consumo energético.



Figura 1. Selección de componentes en Tinkercad



Figura 2. Conexión de LED a pin 0 de la *micro:bit*



Figura 3. Conexión de piezo a la *micro:bit*



Manos a la obra

Conectadas



Esta sección corresponde al 85% de avance de la sesión

Para aplicar sensores y actuadores vamos a iniciar esta sesión con una práctica que simule el funcionamiento de una alarma que detecta el movimiento y emite un sonido que alerte la presencia. sigue los pasos:

Paso 1: crea un nuevo circuito desde *Tinkercαd*, haz clic en la opción "componentes" y luego selecciona la opción "All" o "Todos", *Figura 1*.

Paso 2: busca en la sección de componentes "output" o "salida" el LED e instálalo tal como en la práctica del semáforo, ver *Figur*α 2.

No olvides la resistencia.

Paso 3: allí mismo en la sección "output" encontrarás un elemento llamado "piezo", el cual es un dispositivo que funciona de forma semejante a un altavoz que emite sonidos de agudos a graves, dependiendo de la señal eléctrica que fluye por él. Después variaremos el sonido con una señal de voltaje variable. Instala el *piezo*, de acuerdo con la imagen, el polo negativo conectado a GND y el positivo a pin 1, ver Figura 3.

Paso 4: en la sección "input" o "entrada" de los componentes busca el sensor PIR o "PIR sensor", que es un sensor digital que al detectar movimiento emite una señal de corriente eléctrica constante hasta que deja de detectar movimiento. Instálalo tal como se ve en la *Figura 4.*

El terminal "power" o "potencia" del sensor debe ir conectado a 3V, la terminal "ground" o "tierra" conectada a GND y "signal" o "señal" a pin 2.

Ahora comencemos con la programación.

Figura 4. Conexión de Sensor PIR a la *micro:bit*



Paso 5: crea un par de variables llamadas "Movimiento" y "PWM". Con la primera vamos a guardar el estado del sensor. Si se detecta movimiento tomará el valor de 1 y en caso contrario valdrá 0. La variable "PWM", por otro lado, nos permitirá escuchar cómo cambia el sonido al variar el voltaje en el parlante o piezo.

Busca en la paleta "Input" la lectura del pin digital pin 2 y conéctalo a la variable "movimiento" dentro del ciclo "forever" o "Para siempre", como se ve en la *Figura 5*.

Figura 5. Creación y programación de variables



Ahora la variable "Movimiento" toma la lectura de pin 2, es decir la señal emitida por el sensor PIR.

Paso 6: desde la paleta "control" selecciona y conecta una condicional doble, luego desde la paleta "math" selecciona y arrastra a la pregunta de la condicional una igualdad y pregunta ¿La variable Movimiento es igual a 1?, ver *Figura 6*.





En caso de que "Movimiento" sea igual a 1, quiere decir que el sensor PIR está detectando movimiento, así que encenderemos el LED conectado a "pin O" pasándolo en una escritura digital a "HIGH". Por otra parte, si la sentencia es verdadera también vamos a hacer sonar el piezo. Sin embargo, fíjate que para cambiar el sonido debemos modificar la señal eléctrica, para lo cual usaremos una escritura analógica que inicialmente vamos a dejar con un valor de 1.

En caso de que "Movimiento" sea 0 el Led debe estar en "LOW" y el *piezo* en 0 como se ve en la *Figura 7.*

Figura 7. Programación para "Movimiento" en O



Ahora vamos a usar la variable "PWM" para hacer que el sonido del piezo cambie cada que haya movimiento.

Figura 8. Acción de incremento de variable "PWM"

Paso 7: en la paleta variable selecciona y conecta en el lado verdadero de la condicional la opción "Change (PWM) by (1)" y donde estaba el 1 en la escritura analógica ensambla el valor de "PWM". Esto hará que cada vez que el sensor detecte movimiento se incremente en 1 el valor de la variable "PWM" y al escribirse en la salida del piezo hace que cambie el sonido, ver *Figura 8*.

Reinicia el valor de "PWM" a cero en caso de que el sensor no detecte movimiento, como se ve en la *Figura* 9.





Paso 8: simula el montaje dando clic en "Start simulation" o "Iniciar simulación".



Mientras se ejecuta la simulación, haz clic sobre el sensor PIR y arrastra el círculo azul para simular el movimiento que detectará la alarma, como se observa en la *Figura 8.*



¿Notas el cambio de sonido al simular movimiento en el montaje?

La variable PWM sirve para ajustar el voltaje variable de la salida, aunque en este ejemplo lo usamos para modificar continuamente la señal y así provocar cambios en el sonido.



¿Cómo se te ocurre que puedes usar sensores y actuadores para solucionar un problema en tu contexto?

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes de la sesión. Elije la opción de respuesta que mejor describa lo que alcanzaste.



O Sí

1

- Parcialmente
- 🔘 Aún no

¿Puedes hacer lecturas analógicas de sensores externos y controlar actuadores externos desde la escritura analógica?
Sí
Parcialmente
Aún no
¿Puedes conectar sensores y actuadores externos a la micro:bit desde el simulador Tinkercad?
Sí
Parcialmente
Aún no

Si tus respuestas fueron "Parcialmente" o "Aún no", discute con tu compañera o compañero de grupo el trabajo realizado sobre sensores y actuadores, verifica la solución que dieron a uno de los ejercicios planteados y consulta con tu docente las preguntas que aún tengas.

A continuación te proponemos algunas preguntas para discutir en grupo y luego compartir:

0	6
\mathcal{D}	0
ى	0

Si regresamos al reto, ¿para qué serviría lo que hemos aprendido en esta sesión?

¿Qué ventajas tendría trabajar con lecturas y escrituras análogas frente a lecturas y escrituras digitales de encendido y apagado?

Aprovecha este espacio final para crear un esquema que sintetice lo aprendido, incorporando definiciones de términos clave y ejemplos relacionados con los conceptos aplicados al programar una alarma que, al detectar movimiento, ajusta el tono de un piezo mediante **escritura analógica** en la *micro:bit*.







Aprendizajes esperados

Al final de esta sesión se espera que puedas:



Conectar correctamente motores eléctricos a la *micro:bit.*



Controlar la velocidad de un motor desde la *micro:bit.*

Material para la clase

O Tinkercad - cuenta de estudiante o personal.













Anexo



Lo que sabemos,

lo que debemos saber

Esta sección corresponde al

15% de avance de la sesión

////

En la última sesión, nos enfocaremos en la eficiencia energética, un pilar fundamental para la ciudad del futuro. Usaremos la *micro:bit* para controlar la velocidad de un **motor DC** mediante un **transistor** y una **fotorresistencia**, ajustando el voltaje variable en función de la luz disponible. Este montaje nos enseñará cómo podemos utilizar sensores para ahorrar energía y reducir el impacto ambiental, haciendo de la ciudad del futuro un lugar más sostenible y amigable con el planeta. A través de esta actividad, culminaremos nuestra travesía de aprendizaje, viendo cómo todas las piezas encajan para construir una ciudad verdaderamente inteligente.

Es el momento de intentar resolver el reto. Dale una mirada al *Anexo 1.1.*

En la práctica anterior aprendimos que, para ajustar el voltaje variable a un actuador o salida, como un piezo, un parlante o un zumbador, es necesario usar una escritura analógica que permita ajustar los valores entre 0 y 1023. Cuando la señal está en 0, no pasa electricidad y cuando está en 1023, permite el paso de toda la electricidad disponible. Sin embargo, hay actuadores que no funcionan con los 3 voltios que provee la *micro:bit* y requieren baterías externas. Este es el caso de los motores eléctricos, que necesitan un mayor voltaje para alcanzar su máxima velocidad de 16,082 RPM (revoluciones por minuto).

Para hacer funcionar correctamente un motor eléctrico pequeño con la señal de la *micro:bit* (3V), necesitamos amplificar dicha señal a un voltaje mayor (9V). Para ello, utilizaremos un transistor TIP 120. Los transistores funcionan como pequeños interruptores que pueden encender y apagar componentes eléctricos y también amplificar la electricidad. Solo necesitan una pequeña cantidad de corriente en la base para permitir que una carga eléctrica mayor fluya entre el colector y el emisor. Al proveer la base con una salida desde un pin de la *micro:bit*, se abrirá una compuerta interna que permitirá que los 9 voltios de la batería fluyan a través del motor eléctrico. Por ahora, lo importante es reconocer las tres terminales o conexiones del transistor: la base, el colector y el emisor. También, ten en cuenta que la señal o corriente eléctrica emitida desde los pines de la *micro:bit* controlará gradualmente el flujo de corriente o voltaje variable entre la batería y el motor, a través del transistor.

Glosario

Motor DC: dispositivo que convierte energía eléctrica en movimiento rotatorio utilizando corriente directa.

Transistor TIP 120: componente electrónico que actúa como interruptor digital para controlar el flujo de corriente en un circuito. También amplifica el voltaje que reciben los componentes conectados a su salida.

Fotorresistencia: componente electrónico cuya resistencia varía en función de la cantidad de luz que recibe.

Manos a la obra Conectadas



Esta sección corresponde al 85% de avance de la sesión

Aprendamos a conectar los motores eléctricos a la *micro:bit* con un transistor TIP 120. Sigue los pasos:

Paso 1: busca los componentes a continuación y arrástralos al lienzo. Usa el buscador (Lupa para buscar componentes) para encontrar el transistor TIP 120, pues hay otros componentes que se le parecen mucho, como se puede ver en la *Figura 1*.



Figura 2. Alimentación de componentes



45

Paso 2: cada que realices un montaje con baterías externas recuerda que debemos conectar el polo negativo de la batería y todos los polos negativos de los componentes. Conecta el polo negativo de la batería al emisor y luego el emisor al GND de la *micro:bit.*

También conecta el polo positivo de la batería a alguna de las terminales del motor, ver *Figura 2*.

Paso 3: conecta la otra terminal o conexión del motor al colector del transistor y por último la base del transistor al pin 2 de la *micro:bit,* ver *Figura 3.*

Figura 3. Conexiones de componentes



Iniciemos ahora con la programación considerando que vamos a observar el cambio de la velocidad del motor o el incremento de sus RPM (revoluciones por minuto) en el simulador. También podrías observar directamente en un montaje físico, si cuentas con los componentes. **Paso 4:** crea una variable llamada "PWM", conéctala en la función "on start" o "al iniciar" con el valor de cero. Por otra parte, anida un bucle "repeat while (true)" o "repetir mientras que (verdadero)" dentro del bucle "forever" o "por siempre". Conecta como pregunta condicional del bucle ¿es la variable PWM menor que 1023? ver *Figura 4*.



Figura 4. Programación condicional con variable PWM

Paso 5: dentro del bucle anidado, asigna el valor de la variable "PWM" a la escritura análoga de pin 2, que en este caso regula la entrada de electricidad a la base del transistor TIP 120 con una señal que varía entre 0 a 3 voltios. También dentro del bucle anidado incrementa el valor de "PWM" en 1 y agrega una espera de 10 milisegundos.

Ya por fuera del bucle anidado, asigna un valor de O a la escritura analógica de pin 2 para detener el motor y muestra un ícono que indique que el proceso se detuvo, como se ve en la *Figura 5*.

while **v** repeat PWM < 🔻 1023 write analog pin P2 🔻 to PWM 1 change PWM 🔻 by milliseconds 🔻 wait 10 **J** show icon write analog pin **P2** to 0

Figura 5. Asignación de valor de 0 a la escritura analógica de pin 2

Prueba parcialmente el montaje simulando.



Fíjate en las RPM (revoluciones por minuto) del motor:



;Se incrementan? ;El motor acelera o disminuye su velocidad?

Hasta este punto solamente hemos visto el efecto de la variable "PWM" sobre la escritura analógica. Ahora vamos a ver la relación entre los sensores, la lectura analógica (read), el PWM y la escritura analógica (write) en torno a la función "map". Sigue los pasos sobre este mismo montaje. **Paso 6:** busca entre los componentes "input" o "entrada" la fotorresistor o fotocelda LDR cuyo propósito es medir la intensidad de la luz. El fotorresistor funciona como una válvula que deja pasar mayor o menor cantidad de electricidad entre sus conectores dependiendo de la luz disponible. Si hay mucha luz, pasará más electricidad que en ausencia de ella. Conecta una resistencia en alguna de sus conectores como se muestra en la *Figura 6*.

Figura 6. Conexión de resistencia a fotoresistor



Paso 7: extiende un cable desde el punto de conexión de la resistencia con la terminal del fotorresistor hasta el pin 0, ver *Figur*α 7.

49



Figura 7. Conexión de fotoresistencia a pin cero

Paso 8: conecta la terminal de la resistencia al GND de la *micro:bit* y la terminal suelta del fotorresistor a los 3V, ver *Figura 8*.

Figura 8. Conexión de fotoresistencia a transistor



Ya tenemos el montaje, ahora vamos a programar nuevamente. Te sugiero que elimines el contenido del bucle "forever" o "para siempre" que habíamos programado para incrementar el PWM del motor.



On start	
set PWM 🔻 to 🛛	
forever	

Paso 9: crea dos variables más, una llamada "Sensor" y la otra llamada "Display". La primera nos servirá para guardar la lectura analógica del fotorresistor conectado a pin 0 y la segunda la usaremos para visualizar en el display o matriz de LED de la *micro:bit* el nivel de luz captada por el fotorresistor.

Dentro del bucle "forever" asigna desde la paleta Input o entrada la lectura analógica de pin 0 a la variable "Sensor".



Figura 10. Programación valor variable "Sensor"

Paso 10: conecta la asignación de la variable PWM debajo de la asignación del sensor, también conecta a la escritura analógica de pin 2 el valor de la variable PWM para controlar la velocidad del motor.

Figura 11. Escritura analógica de pin 2

OTEV	er				
set	Sensor 🔻	to	rea	ad analog pin	P0 🗸
set	PWM 🔻 to	0			
vrite	e analog pin	P2	•	to PWM	

Paso 11: selecciona desde la paleta "math" o "matemáticas" la función "map" y asigna al valor que tomará la variable PWM cada vez que se repita el ciclo "forever" o "para siempre". La función map opera de la siguiente manera: en el espacio llamado map se agrega la variable que tiene el rango que deseamos transformar. En este caso la variable sensor, por efectos de la resistencia eléctrica, vale 7 cuando carece totalmente de luz y 679 cuando el fotorresistor capta la mayor cantidad de luz.

Los otros dos espacios son para el valor mínimo y máximo del rango que queremos como salida, es decir que en este caso cuando el Fotorresistor no tiene luz y marca 7, este se transforma en una salida equivalente a 0, y cuando el fotorresistor tiene luz plena y marca 679, la salida equivale a 1023. Es importante tener en cuenta que el valor de salida se asigna en este caso a la variable "PWM", ver *Figura 11*.



En este punto ya puedes hacer la simulación, haz clic sobre el fotorresistor mientras se ejecuta la simulación y desplaza la barra hacia la izquierda para ausencia de luz y a la derecha para luz plena. Notarás que el motor incrementa sus revoluciones por minuto o RPM cuando el fotorresistor adquiere más luz.

Paso 12: convierte ahora el rango del PWM a 5 niveles , ver *Figura* 12, que grafique en pantalla el estado del fotorresistor. Busca la opción "plot graft ()()" o "trazar gráfico de barras desde () hasta ()" en la paleta de "output" o "salida".

Figura 12. Rango del PWM a 5 niveles



Reflexiona sobre lo siguiente:

ୖୄ	¿Qué aparatos o artefactos crees tú que usan la función "map" para funcionar correctamente?
	¿Qué artefactos piensas que usan señales de voltaje variable para modificar sus estados?
	¿Cómo se te ocurre que puedes usar sensores y actuadores para solucionar un problema en tu contexto?
	¿Qué otra situación se te ocurre que es susceptible para aplicar automatización en la ciudad del futuro?

Antes de irnos

(

(



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes de la sesión. Elije la opción de respuesta que mejor describa lo que alcanzaste.

1	¿Puedes conectar sensores y actuadores externos a la <i>micro:bit</i> desde el simulador <i>Tinkercad</i> ?
	 Sí Parcialmente Aún no
2	¿Puedes hacer lecturas analógicas de sensores externos y controlar actuadores externos desde la escritura analógica?
	 Sí Parcialmente Aún no
3	¿Transformar proporcionalmente rangos con el uso de la función "MAP"?
	 Sí Parcialmente Aún no
4	¿Controlar con la <i>micro:bit</i> motores eléctricos que requieran baterías externas a través de transistores?
	 Sí Parcialmente Aún no
Si	tus respuestas fueron "Parcialmente" o "Aún

no", regresa a explorar las conexiones de sensores y actuadores. Si después de esto todavía tienes dudas, acude a tu docente por apoyo adicional. Ahora en grupos de 3 o 4 integrantes, deberán idear una mejora o extensión creativa del sistema del parqueadero inteligente utilizando los conceptos previamente abordados. Por ejemplo:

- O Un sistema que envíe alertas a un celular.
- Un parqueadero con luces de colores de acuerdo a la disponibilidad.
- O Integración de sensores de humo para emergencias.

Una vez finalicen de generar ideas de mejoramiento, los equipos van a preparar una breve presentación (1-2 minutos) para explicar su mejora.

Para finalizar, van a **diseñar un esquema** que resuma lo aprendido durante la actividad. En el esquema, **incorpora definiciones** de los términos clave, como **escritura analógica, lectura analógica** y **función map** y muestra cómo estos conceptos se aplican al **controlar la velocidad de un motor DC.** Usa ejemplos concretos para ilustrar cómo **el fotorresistor** se utiliza para medir la luz y cómo estos valores se transforman mediante la **función map** para ajustar la velocidad del motor. El diagrama debe ayudarte a visualizar las interacciones entre los componentes del sistema y el flujo de datos entre ellos, mientras refuerzas los conceptos clave que has trabajado en esta actividad.



Anexo 1.1 Reto

Los sistemas realimentados son comunes en casi todos los campos. En biología, un ejemplo claro es la homeostasis, la capacidad de los seres vivos para mantener condiciones internas estables a pesar de los cambios en su entorno, como la temperatura. Este proceso utiliza información del entorno para ajustar funciones internas. Por ejemplo, el sudor, como se observa en la imagen de un rostro sudoroso, es una respuesta del cuerpo humano ante temperaturas altas, permitiendo la refrigeración y el mantenimiento de una temperatura corporal adecuada.



Casi todos los artefactos tecnológicos que cuentan con componentes electrónicos o tecnología digital incorporan sistemas realimentados, diseñados para responder de manera autónoma a cambios en su entorno. Un ejemplo cotidiano es el ventilador de un computador, que se enciende automáticamente al detectar un aumento en la temperatura interna, protegiendo así los componentes electrónicos de daños.

Esta misma lógica se aplica a tecnologías más avanzadas. Un dron utilizado en agricultura sostenible de precisión opera con múltiples sistemas realimentados para realizar tareas complejas de manera eficiente. Lo mismo ocurre con robots, automóviles, ascensores, puertas automáticas, aspiradoras robot y hornos, entre otros dispositivos. La lista es extensa, ya que los sistemas realimentados son esenciales para garantizar el funcionamiento autónomo y seguro de estas tecnologías.



En este contexto, el reto que se propone es simular un sistema automatizado similar al que regula el paso de vehículos con un chip o TAG en los peajes electrónicos. En este sistema, la talanquera se activa automáticamente cuando un sensor detecta el TAG, permitiendo el paso del vehículo. Si un vehículo no cuenta con TAG, la talanquera no se activa, impidiendo su avance. Este ejercicio brinda la oportunidad de experimentar de manera práctica cómo los sistemas realimentados optimizan procesos en situaciones del mundo real, destacando su relevancia y aplicabilidad.

GIAUUTI GIAIG T

Anexo 1.2 Vehículos y peajes



Anexo 1.3 Instrucciones

Cada rol maneja los siguientes mensajes y respeta las siguientes condiciones:

Rol	Acciones posibles y condición	Mensajes que puede pasar
Estudiante 1 El vehículo	Puede ver, no puede hablar. Acerca un vehículo de los 3 lentamente.	
Estudiante 2 El sensor del tag	Puede ver, solo puede decir un mensaje.	Vehículo con TAG
Estudiante 3 El computador	No puede ver. Debe ejecutar el algoritmo que se muestra a continuación. Solo puede decir los siguientes mensajes.	Abrir talanquera Cerrar talanquera
Estudiante 4 La talanquera	No puede ver ni hablar. Levanta la talanquera cuando escucha " abrir talanquera " y la cierra cuando escucha " cerrar la talanquera ".	
Estudiante 5 Sensor de paso	Puede ver. Envía mensajes cuando el vehículo pasó.	Vehículo ya pasó



Grado 11º Guía 4

Anexo 2.1 Semáforo



orever							
digital	write	pin	P0	•	to	HIGH	•
digital	write	pin	P1	•	to	LOW	•
digital	write	pin	Р2	•	to	LOW	•
wait	5	secs 🔻					
digital	write	pin	P0	-	to	HIGH	
digital	write	pin	Р1		to	HIGH	~
digital	write	pin	Р2		to	LOW	~
wait	2	secs 🔻					
digital	write	pin	P0	•	to	LOW	~
digital	write	pin	P1	•	to	LOW	~
digital	write	pin	Р2	•	to	HIGH	~
wait	5	secs 🔻					
digital	write	pin	P0		to	LOW	~
digital	write	pin	Ρ1		to	HIGH	~
digital	write	pin	Р2	-	to	LOW	~
wait	2	secs 🔻					







