

Sesión

1

Aprendizajes esperados

Duración sugerida

Al final de esta sesión verifica que puedas:



Simular un sistema en *Tinkercad*.



Crear y manipular variables en *Python* dentro del entorno *Tinkercad*.



Configurar pines virtuales para controlar servomotores en *Tinkercad*.



Material para la clase

- ☐ Anexo 1.1
- ☐ Computador con acceso a internet



Lo que sabemos, lo que debemos saber



Esta sección corresponde al 40% de avance de la sesión

Observa las Figuras 1, 2 y 3 y responde:



¿Has visto alguno de estos sistemas?
¿Cuál es su función?

Figura 1. Fila virtual



Figura 2. Control de turnos en centro de atención

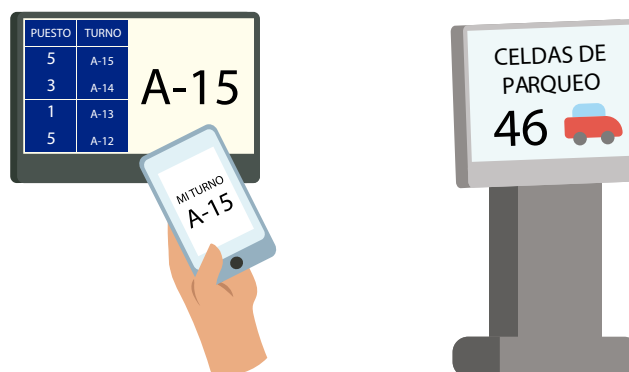


Figura 3. Disponibilidad celdas de parqueo



Anexo

Anexo 1.1

Tu reto es crear un sistema de parqueadero inteligente utilizando una *micro:bit* como unidad de control central. El sistema debe gestionar un parqueadero con 9 espacios disponibles inicialmente. Utilizando dos servomotores conectados a los pines P0 y P18 de la *micro:bit* para simular las barreras de entrada y salida respectivamente. La pantalla led integrada de la *micro:bit* se usará para mostrar constantemente el número de espacios disponibles. Los botones A y B de la *micro:bit* simularán la llegada y salida de vehículos; el botón A representará un vehículo entrando, mientras que el botón B representará un vehículo saliendo.

El sistema debe ser capaz de:

1. Disminuir el contador de espacios y abrir la barrera de entrada (mover el servomotor en P0 a 0 grados) cuando se presiona el botón A, siempre y cuando haya espacios disponibles.
2. Aumentar el contador de espacios y abrir la barrera de salida (mover el servomotor en P18 a 0 grados) cuando se presiona el botón B, siempre que el parqueadero no esté vacío.
3. Mostrar un *icono de No* en la pantalla led cuando se intenta ingresar a un parqueadero lleno o salir de un parqueadero vacío.
4. Mantener las barreras abiertas por 5 segundos antes de cerradas (volver a 90 grados).
5. Actualizar y mostrar el número de espacios disponibles después de cada operación. Tu reto incluye implementar esta lógica, asegurando que el sistema funcione de manera fluida y maneje correctamente todos los casos posibles, incluidos los casos límite cuando el parqueadero está lleno o vacío.

Para ir más lejos, adicionalmente, el sistema debe incorporar un conjunto de led para mejorar la señalización visual. Se utilizarán cuatro led en total: un par de led (rojo y verde) para cada barrera (entrada y salida). El led rojo permanecerá encendido mientras la barrera correspondiente esté abajo y se apagará cuando la barrera esté arriba. El led verde, por el contrario, se encenderá cuando la barrera esté arriba y se apagará cuando esté abajo. Esta característica proporcionará una indicación visual clara del estado de cada barrera, mejorando la experiencia del usuario y la seguridad del sistema.



Figura 1: fila virtual para comprar boletas o productos cuando hay muchos(as) clientes intentando entrar a una página de internet y así no sobrepasar su capacidad.

Figura 2: diferentes formas de llevar los turnos en un centro de atención, por ejemplo, la sala de espera de un consultorio o en un centro de servicio al cliente. Así, las y los clientes conocen el tiempo de espera y se respeta la capacidad de atención de las personas que trabajan.

Figura 3: un anuncio de la cantidad de celdas de parqueo disponibles por piso. Así se controla la capacidad del parqueadero y se evita que muchas personas esperen a que se libere un cupo.



¿Notas algo en común en todos los ejemplos?
¿Cómo crees que funciona?

Durante esta guía vas a simular el funcionamiento de un parqueadero. A través de este proyecto vas a fortalecer tus habilidades en programación, a adquirir conocimientos nuevos y comprender mucho mejor cómo funcionan este tipo de sistemas tan utilizados en el día a día. Antes de continuar, lee el reto que se encuentra en el Anexo 1.1.

Como sabes, la *micro:bit* es una pequeña computadora programable que puede controlar diversos componentes y mostrar información. En nuestro parqueadero inteligente, la usaremos para llevar la cuenta de los espacios disponibles en un parqueadero y controlar las barreras de entrada y salida.

En nuestro proyecto, esto implica que debemos programar la *micro:bit* para mostrar números en su pantalla led y configurar los pines para controlar los servomotores de las barreras, entre otras cosas. Sin embargo, no siempre contamos con suficientes elementos físicos como servomotores, cables o baterías. Así que toda la solución del proyecto la realizarás en un entorno simulado llamado *Tinkercad*.

Anexo

Anexo 1.2

Versión en bloques de Tinkercad



Versión en Python

```
def on_button_pressed_A():
    global puestasDisponibles
    if puestasDisponibles == 9 and puestasDisponibles > 0:
        pins.servo.write_pulsewidths(PUL, 0)
        basic.pause(5000)
        pins.servo.write_pulsewidths(PUL, 90)
        puestasDisponibles -= 1
        basic.show_number(puestasDisponibles)
    else:
        basic.show_text("¡Complétame!")
        basic.pause(5000)
        basic.show_number(puestasDisponibles)
    input_on_button_pressed(button_A, on_button_pressed_A)
```

Tinkercad es un simulador de circuitos electrónicos basado en navegador que admite microcontroladores Arduino Uno, placas *micro:bit* o chips ATtiny. El código se puede crear utilizando CodeBlocks gráficos, fragmentos de código que se pueden organizar fácilmente con el ratón o código basado en texto.

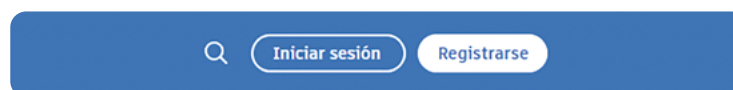
En años anteriores aprendiste que las simulaciones son útiles para ahorrar tiempo, dinero y ejecutar experimentos que no podrías hacer desde tu salón. Durante esta guía aprenderás que simular sistemas también resulta muy útil ya que puedes hacer muchas pruebas sin provocar daños permanentes, además de que cuentas con un catálogo muy amplio de elementos, sin necesidad de invertir dinero.

Para ingresar a *Tinkercad* sigue estas instrucciones:

Paso 1: Entrar a *Tinkercad*

- Accede a *Tinkercad* siguiendo el enlace o el QR.
- Haz clic en Iniciar sesión como se presenta en la *Figura 4*.

Figura 4. Iniciar sesión en *Tinkercad*



- Elige Estudiantes con código de clase como se presenta en la *Figura 5*.

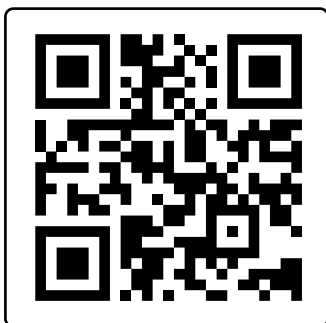
Figura 5. Selección de Estudiantes con código de clase



Nota

Se recomienda que tu docente cree una clase en *Tinkercad* siguiendo el Anexo 1.2

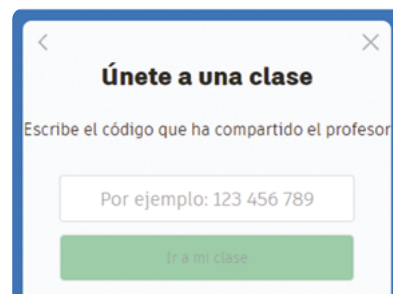
Enlace



 Tinkercad

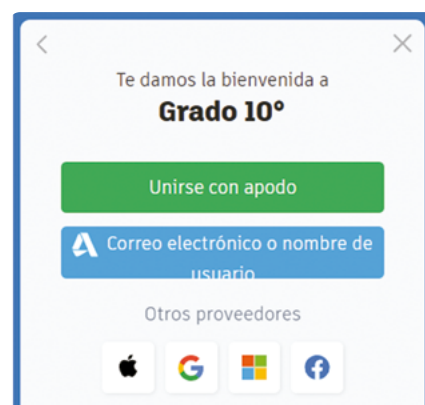
- ☐ Ingresa el código que te comparta tu docente.

Figura 6. Unirse a clase con código



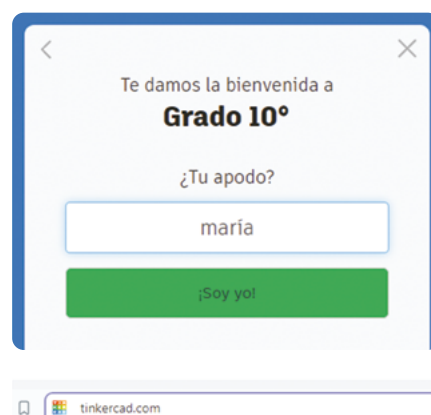
- ☐ Elige Unirse con apodo, a menos que tu docente indique algo diferente.

Figura 7. Ingreso de nombre de identificación en la clase



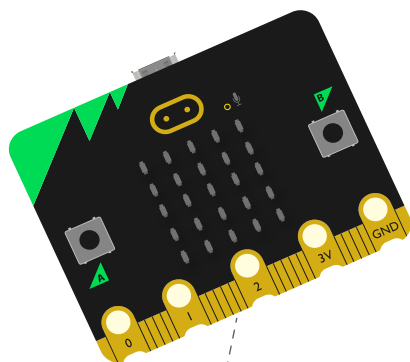
- ☐ Escribe el alias según las indicaciones de tu docente y haz clic en el botón ¡Soy yo!

Figura 8. Ingreso al curso con botón: ¡Soy yo!



- Crea una nueva simulación de circuito haciendo clic en Crear y luego en circuito.

Figura 9. Creación de nueva simulación



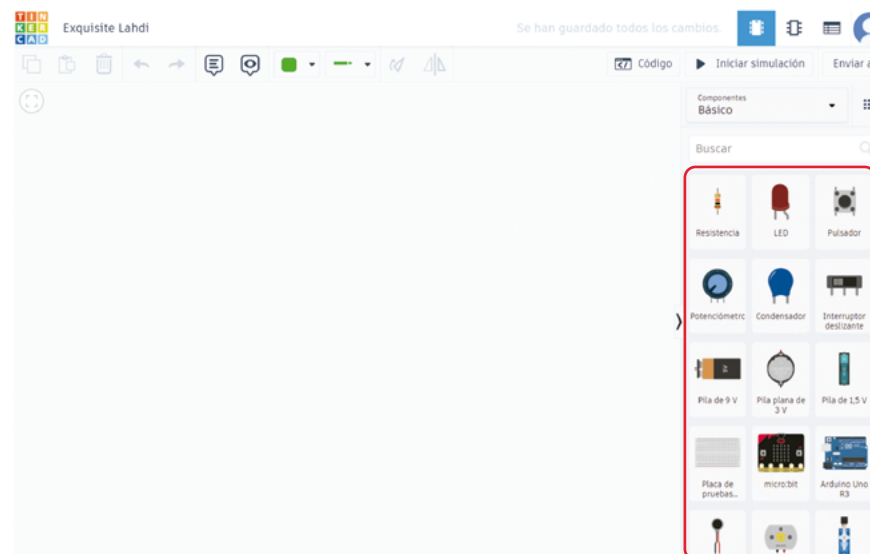
- Tómate unos minutos para explorar la interfaz como se presenta en la Figura 10: componentes, editor de código, simulador. Los componentes de la derecha pueden ser arrastrados y colocados en el área de trabajo.



¿Cuáles reconoces?

¿Cuáles crees que serán útiles en la resolución del reto?

Figura 10. Exploración de interfaz gráfica de Tinkercad



Manos a la obra**Conectadas**

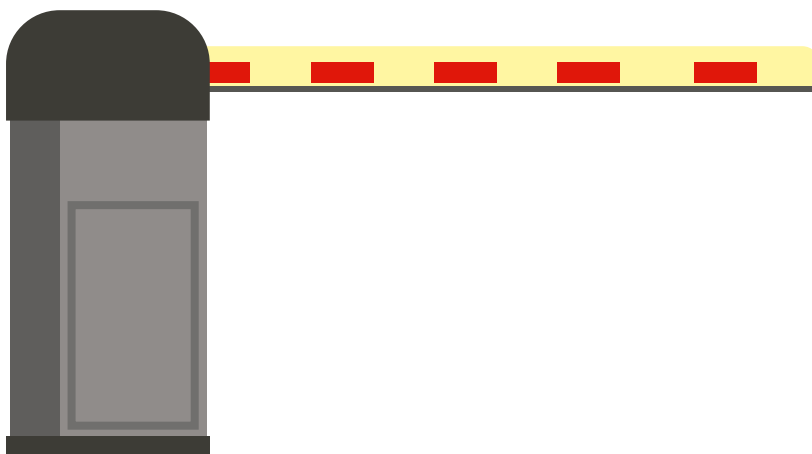
Esta sección corresponde al 80% de avance de la sesión

Ya habiendo explorado el entorno de *Tinkercad*. Vas a completar los dos primeros pasos hacia la simulación del parqueadero: 1. Realizar las conexiones para configurar los servomotores y 2. Crear variables y mostrarlas en la pantalla de led.

Paso 1. Configuración de servomotores

Es posible que en grados anteriores hayas realizado algunas conexiones de sensores y actuadores. En este caso, vas a realizar una conexión de dos servomotores y sus baterías a la *micro:bit*. Los servomotores son los elementos que controlan el movimiento de la barra de entrada al parqueadero.

Figura 11. Control ingreso parqueadero



Tal como en una conexión real, necesitas varios materiales:

- ☐ *Micro:bit* con salidas
- ☐ 2 servomotores
- ☐ Placa de pruebas
- ☐ Baterías o pilas de 1.5V

- Añade una *Micro:bit* al espacio de trabajo. Para esto haz clic en la opción buscar y escribe *micro:bit*.

Figura 12. Búsqueda de dispositivos y elementos en *Tinkercad*



- Selecciona la opción *micro:bit* con salida

Figura 13. Selección de componente *micro:bit*



- Añade dos servomotores al circuito en *Tinkercad*. Búscalos en la sección Componentes:

Figura 14. Selección de servomotores

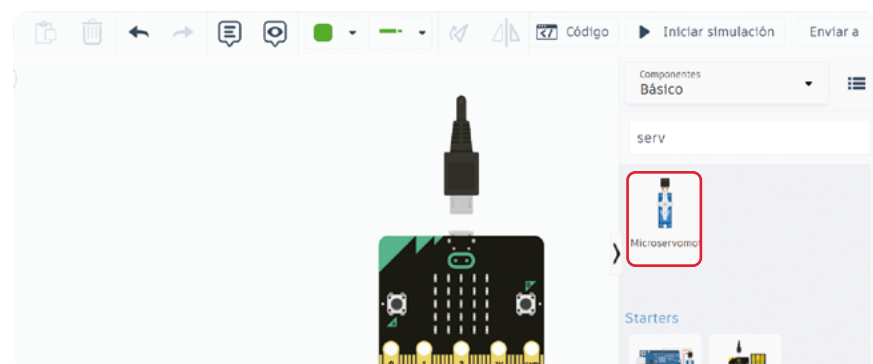


Figura 16. Conexión de pin de señal

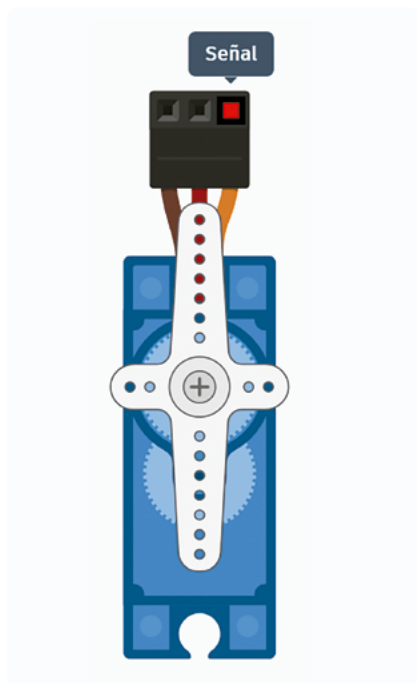
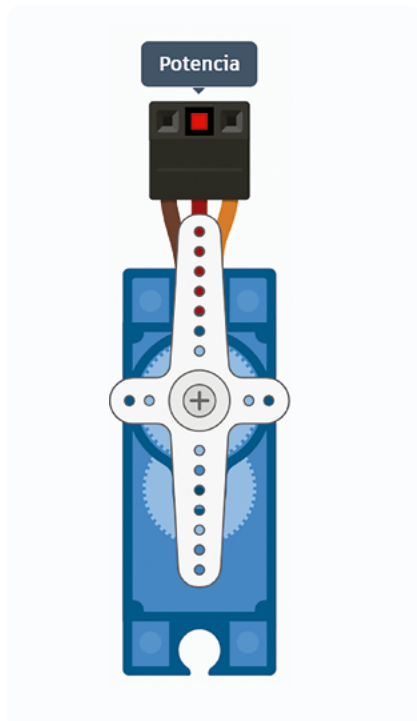


Figura 17. Conexión de pin de potencia



- Busca y añade el resto de los materiales al espacio de trabajo. Al añadir las baterías, selecciona Recuento = 2 baterías, como se ve en la Figura 15.

Figura 15. Selección de baterías



- Al pasar el ratón por los cables de los servomotores, puedes identificar cuál cable es la señal, potencia, y tierra.
- Conéctalos a los pines P15 y P16 de la *micro:bit* simulada. Para esto, lleva los cables de Señal al P15 o P16, el cable Potencia al polo positivo y el cable Tierra al polo negativo, siguiendo la imagen. Puedes cambiar el color de los cables para tener un entorno organizado.

Figura 19. Selección cableado

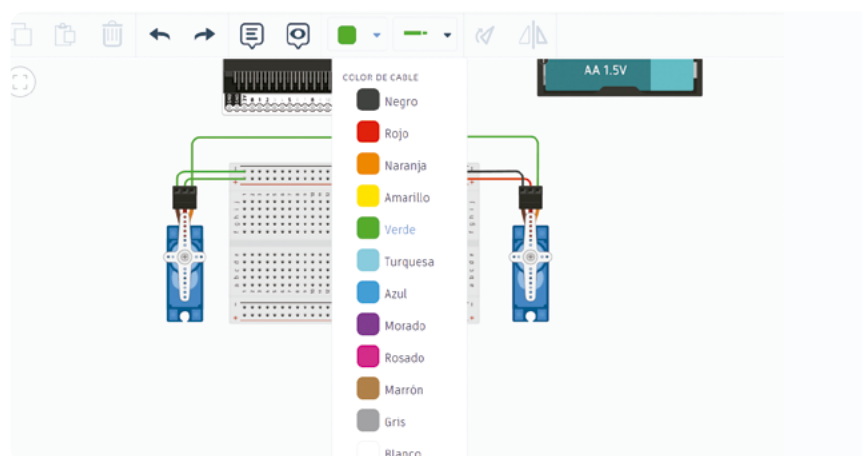


Figura 18. Conexión de pin de tierra

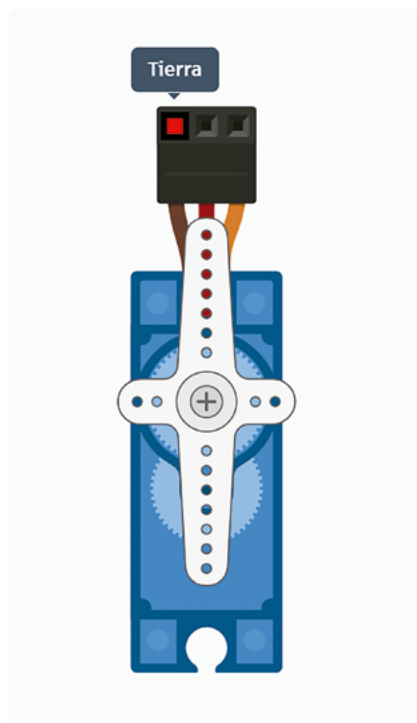


Figura 20. Conexión de pines

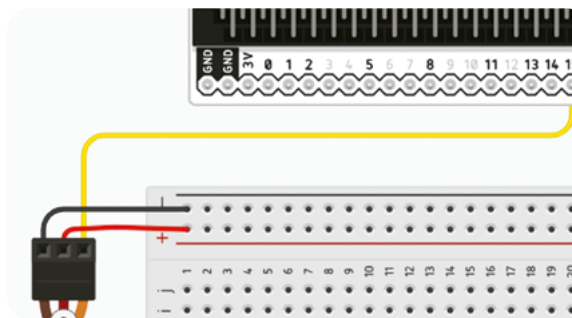
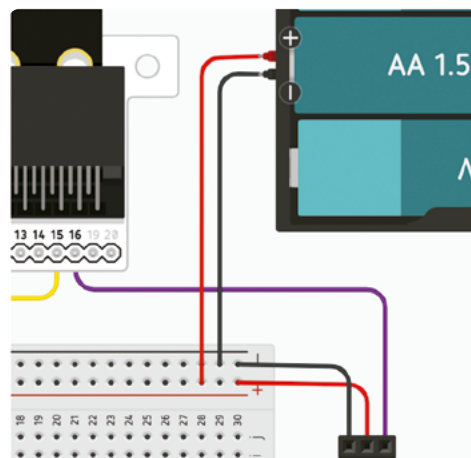
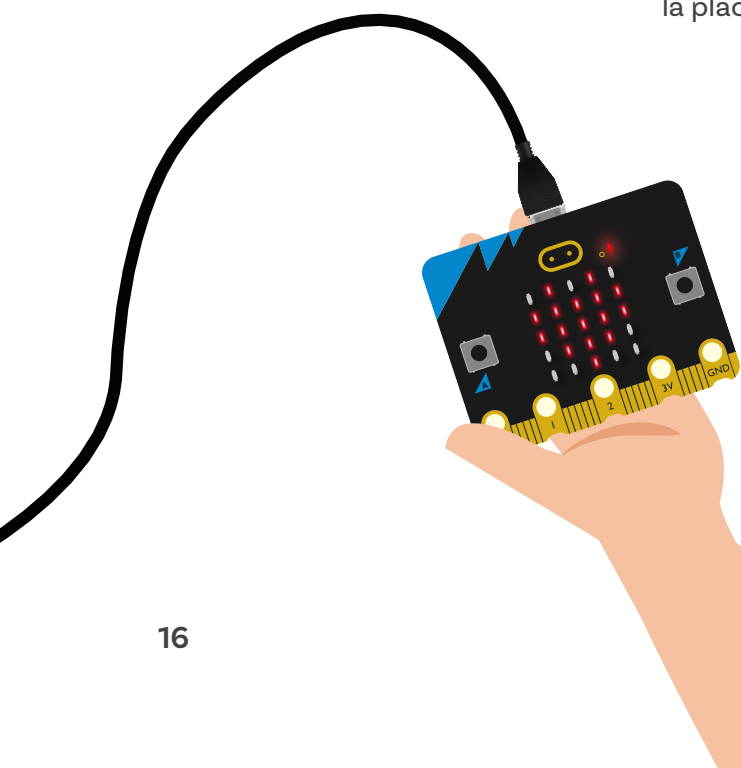


Figura 21. Conexión de baterías

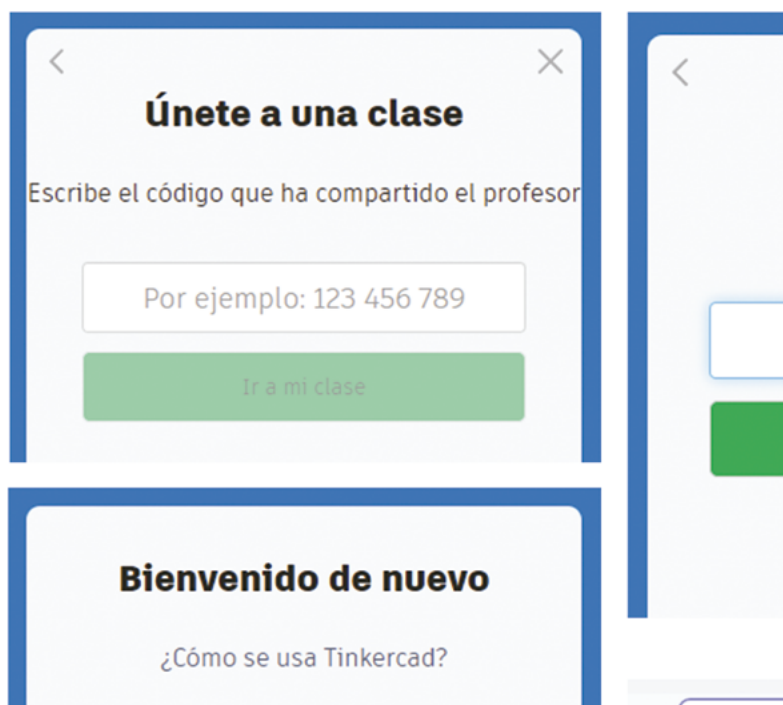


- Por último, conecta la *micro:bit*, llevando el pin de alimentación (3V) al polo positivo y el pin de Tierra (GND) al polo negativo de la placa de pruebas.



En este punto el espacio de trabajo debe verse como en la *Figura 22*.

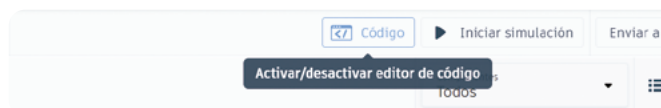
Figura 22. Conexión completa del circuito



Paso 2: Programar la *micro:bit*

- Haz clic en el botón Código para ingresar al editor de código.

Figura 23. Ingreso a editor de código en Tinkercad

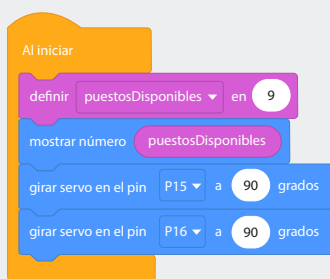


¿Qué ves? ¿Cómo podrías hacer que se vea una flecha en la pantalla al iniciar la simulación?

Es probable que reconozcas el lenguaje de programación por bloques con el que se puede programar la *micro:bit* desde *Tinkercad*. Haz un par de pruebas para explorar sus funciones.

○ Ahora observa estos los códigos de la *Figura 24*.

Figura 24. Visualización de programación en *Tinkercad*



```
puestosDisponibles = 9
```

```
basic.show_number(puestosDisponibles)
```

```
pins.servo_write_pin(AnalogPin.P15, 90)
```

```
pins.servo_write_pin(AnalogPin.P16, 90)
```

¿Qué hacen? Escribe tu predicción.

¿En qué se diferencian?

¿Cuántos puestos disponibles tiene el parqueadero?



Tinkercad ofrece la opción de programar las simulaciones en bloques y el texto. En este caso, el lenguaje de programación en texto es *Python*.

Figura 25. Opciones de visualización en bloques o en código



Recrea el código en tu simulador y pruébalo.



¿Qué sucede? ¿Era lo que esperabas?

Este código inicia nuestro simulador:

- ☐ Crea una variable llamada *puestosDisponibles* y la inicia en 9.
- ☐ Usa `basic.show_number()` para mostrar el valor en la pantalla led simulada.
- ☐ Usa `pins.servo_write_pin()` para controlar los servomotores.
- ☐ Establece la posición inicial de los servomotores a 90 grados.

Realiza las pruebas y verifica el funcionamiento:

- ☐ Experimenta cambiando el valor de *puestosDisponibles* y ejecuta la simulación para ver cómo se actualiza la pantalla.
- ☐ Cambiando los ángulos de giro del servomotor.

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:

- 1 ¿Puedes simular un sistema en *Tinkercad*?
 - ☐ Sí
 - ☐ Parcialmente
 - ☐ Aún no
- 2 ¿Puedes crear y manipular variables en *Python* dentro del entorno *Tinkercad*?
 - ☐ Sí
 - ☐ Parcialmente
 - ☐ No
- 3 ¿Puedes configurar pines virtuales para controlar servomotores en *Tinkercad*?
 - ☐ Sí
 - ☐ Parcialmente
 - ☐ Aún no

Al inicio de la sesión leíste este párrafo:



En años anteriores aprendiste que las simulaciones son útiles para ahorrar tiempo, dinero y ejecutar experimentos que no podrías hacer desde tu salón.

¿Cómo evidencias estas afirmaciones durante el desarrollo de la actividad?

Para cerrar la actividad te proponemos que completes la siguiente tabla y escribas una breve reflexión sobre las consecuencias que tendrían estos experimentos si utilizaras conexiones reales.

Descripción	Figura	Efecto
Conectar la tierra y el pin de alimentación de la <i>micro:bit</i> al polo negativo de la placa de pruebas	Figura 25	
Agregar 4 baterías	Figura 26	
Conectar los servomotores directamente sobre la <i>micro:bit</i>	Figura 27	

Tu reflexión:
