

Sesión

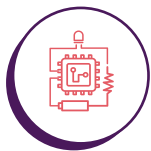
2

Aprendizajes esperados

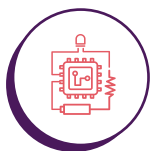
Al final de esta sesión verifica que puedas:



Implementar condicionales en la simulación de un sistema en *Tinkercad*.



Actualizar y mostrar el contador de espacios disponibles en la pantalla led simulada.



Reconocer y programar múltiples estados del sistema.

Duración sugerida



Material para la clase

- Anexo 2.1



Enlace



¿Qué es un servomotor y qué hace?

Lo que sabemos, lo que debemos saber



Esta sección corresponde al 20% de avance de la sesión

En la sesión anterior configuramos nuestro entorno en Tinkercad y verificamos cómo mostrar información básica en la *micro:bit* simulada. Además, programamos el estado inicial de dos servomotores.



¿Puedes recordar su configuración inicial?
Nombra las diferentes conexiones con las que cuentan los servomotores.

Antes de continuar, observa el video sugerido para conocer más acerca de este elemento.

Ahora vamos a dar vida a nuestro parqueadero implementando la lógica de entrada de vehículos.



¿Has pensado alguna vez cómo un sistema automatizado decide si dejar entrar un vehículo a un parqueadero?
¿Qué factores crees que debe considerar?

En esta sesión, programaremos la *micro:bit* para que responda cuando se presiona el botón A, simulando la llegada de un vehículo. Nuestro sistema deberá decidir si hay espacio disponible, actualizar el contador y controlar la barrera de entrada.

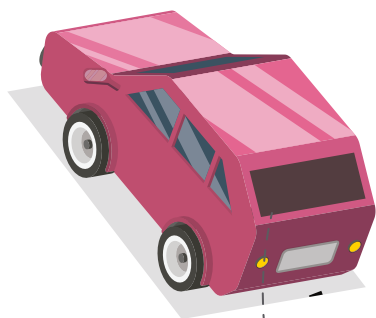


Figura 1. Configuración entradas A y B



Manos a la obra Conectadas



Esta sección corresponde al 60% de avance de la sesión

Sigue las indicaciones que se te dan a continuación

Paso 1: Configuración de los botones

- En *Tinkercad*, asegúrate de que las instrucciones para leer los botones A y el B de la *micro:bit* estén agregados al entorno de trabajo. Para esto, selecciona la opción Básico en donde encontrarás el bloque necesario, arrástralo 2 veces al entorno de trabajo y cambia uno a B. Debe quedar como en la *Figura 1*.
- El botón A manejará el servomotor de la izquierda (P15, entrada al parqueadero) y el B el de la derecha (P16, salida del parqueadero).

Paso 2: Lógica de entrada de vehículos

- Crea un programa que maneje la entrada de vehículos (botón A). Al presionar el botón A el servomotor correspondiente debe girar a 0 grados por 5 segundos y devolverse a su posición inicial de 90 grados.

Figura 2. Programación acciones con entradas A y B

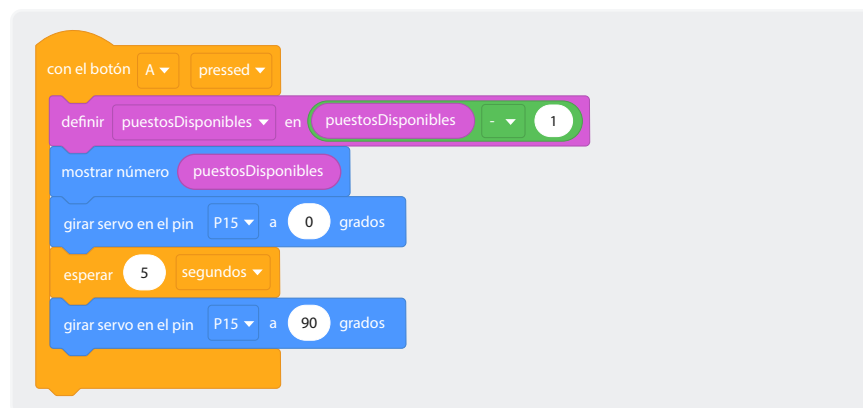


Figura 4. Programación condicional



- Implementa la lógica para decrementar la variable *puestosDisponibles*. La idea es que, al ingresar un vehículo, se disminuya en 1 los puestos disponibles.

Figura 3. Programación variable



Paso 3: Manejo de casos especiales

- Implementa una condición para verificar si el parqueadero está lleno.
- Muestra un ícono de X en la pantalla led cuando no hay espacios disponibles.
- Asegúrate de que la barrera no se abra cuando el parqueadero esté lleno.

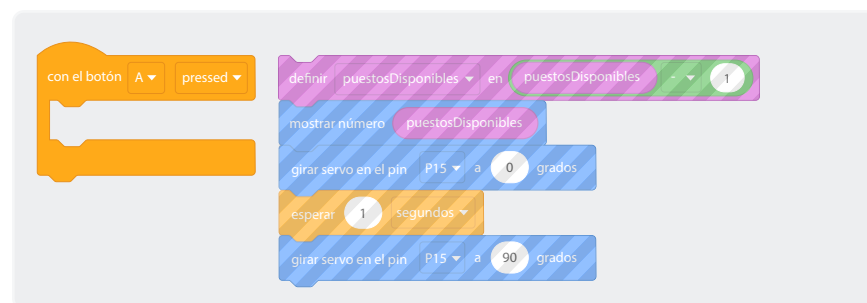


Para hacerlo, vamos a usar el bloque Si – Sino que se encuentra en la opción Control. Este bloque nos permitirá ejecutar o no ejecutar una acción dependiendo de una comparación (valor booleano) que puede dar como resultado verdadero o falso.



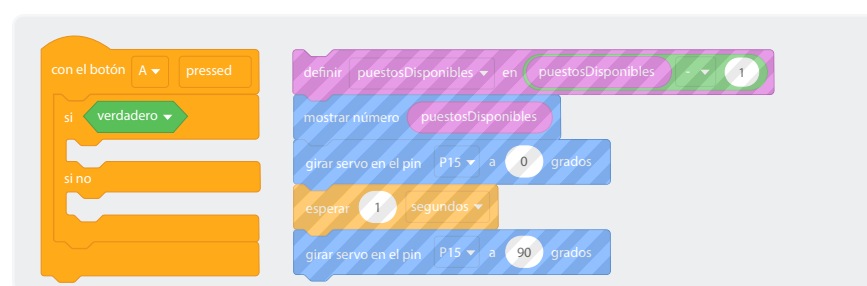
Antes de arrastrar ese bloque al entorno de trabajo y colocarlo dentro del bloque “Botón A” de sacar los bloques que se encuentran en él. Debe quedar como se presenta en la *Figura 5*.

Figura 5. Extraer comandos de entrada de botón A



Ahora, arrastra el bloque “Si – Sino” dentro del bloque “Con el botón A” a fin de que quede como se presenta en la *Figura 6*.

Figura 6. Ingresar condicional a entrada de botón A



Acto seguido, arrastra el código que sacaste del bloque “Con el botón A” a la parte del “si no”, debe quedar como la *Figura 7*.

Figura 7. Ingresar comandos a condicional

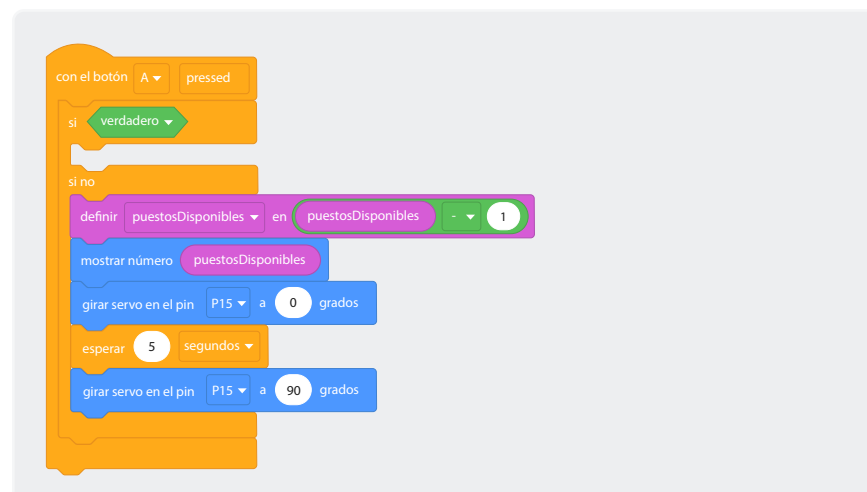
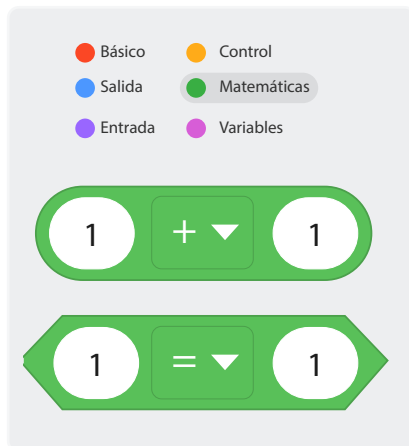


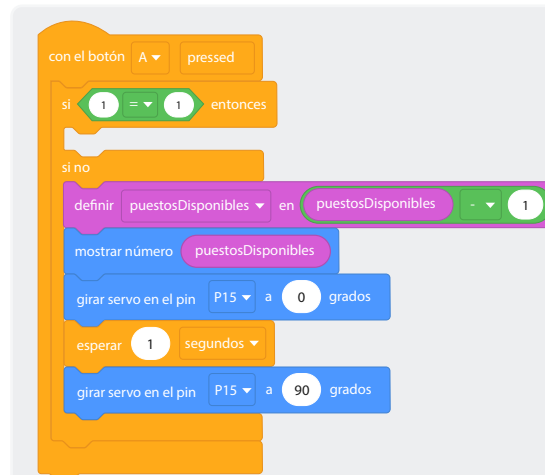
Figura 8. Utilización de operadores



Ahora procederemos a programar el condicional, para esto, iremos a la opción Matemáticas y seleccionamos el bloque que se resalta con rojo en la imagen, arrástralo al entorno de trabajo y colócalo en el bloque Si reemplazando donde dice verdadero.

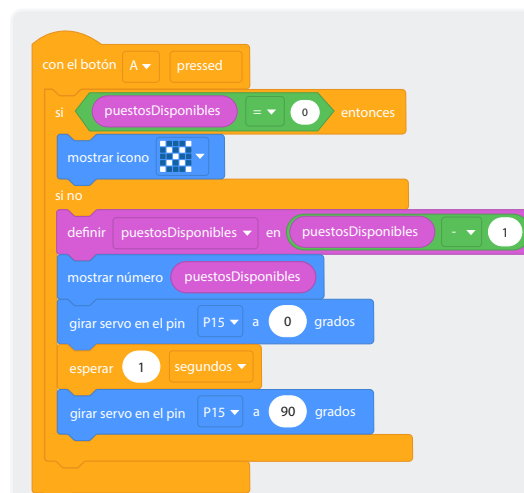
Debe quedar como se presenta en la *Figura 9*.

Figura 9. Ingresar operadores en condicional



Ahora vamos a hacer la comparación. Para esto utilizaremos la variable “puestosDisponibles” y la colocaremos en el condicional, no sin antes cambiar el 1 que está ubicado a la derecha por un 0. Este cambio permitirá que cuando la variable llegue a 0, se muestre el ícono “X” que representa que no hay más puestos en el parqueadero. El programa debe quedar como se presenta en la *Figura 10*.

Figura 10. Presentación de programa completado



Anexo

Anexo 2.1

Versión en bloques de Tinkercad



Versión en Python

```
def on_button_pressed_A():
    global puntosDisponibles
    if puntosDisponibles > 0:
        print("Puntos disponibles: " + str(puntosDisponibles))
        basic.pause(2000)
        puntosDisponibles -= 1
        basic.show_number(puntosDisponibles)
    else:
        basic.show_message("No hay puntos")
        basic.pause(2000)
    repeat_until_button_pressed_A(on_button_pressed_A)
```

Una vez hayas terminado podrás examinar el Anexo 2.1 para observar otra solución al programa. Identifica por lo menos tres diferencias en los códigos y explica lo que hacen. Luego responde:



¿Ambos programas cumplen la misma función?
¿Dirías que un programa es mejor que el otro? ¿Por qué?

Prepárate para compartir tus respuestas.

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 refleje tu progreso:

1

¿Puedes implementar condicionales en la simulación de un sistema en Tinkercad?

- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no

2

¿Puedes actualizar y mostrar el contador de espacios disponibles en la pantalla led simulada?

- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no

3 ¿Puedes reconocer y programar múltiples estados del sistema?

- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no



Si tus respuestas a las preguntas anteriores fueron “Parcialmente” o “Aún no”, regresa a simular condicionales en la simulación de TinkerCAD y reconocer múltiples estados del sistema. Si después de esto, todavía tienes dudas, acude a tu docente por apoyo adicional.

Antes de cerrar la sesión, te proponemos las siguientes preguntas. Responde y discute con tu grupo.

- ☐ ¿Qué riesgos presenta un sistema que solo usa un botón para la entrada?
- ☐ ¿Cómo podrías mejorar la seguridad sin complicar demasiado el sistema?
- ☐ Si un usuario presiona el botón, pero no entra, ¿qué consecuencias tiene para los otros usuarios?
- ☐ ¿Por qué es importante validar que el contador no baje de cero?
- ☐ ¿Qué otras variables necesitarías agregar para hacer el sistema más robusto?

