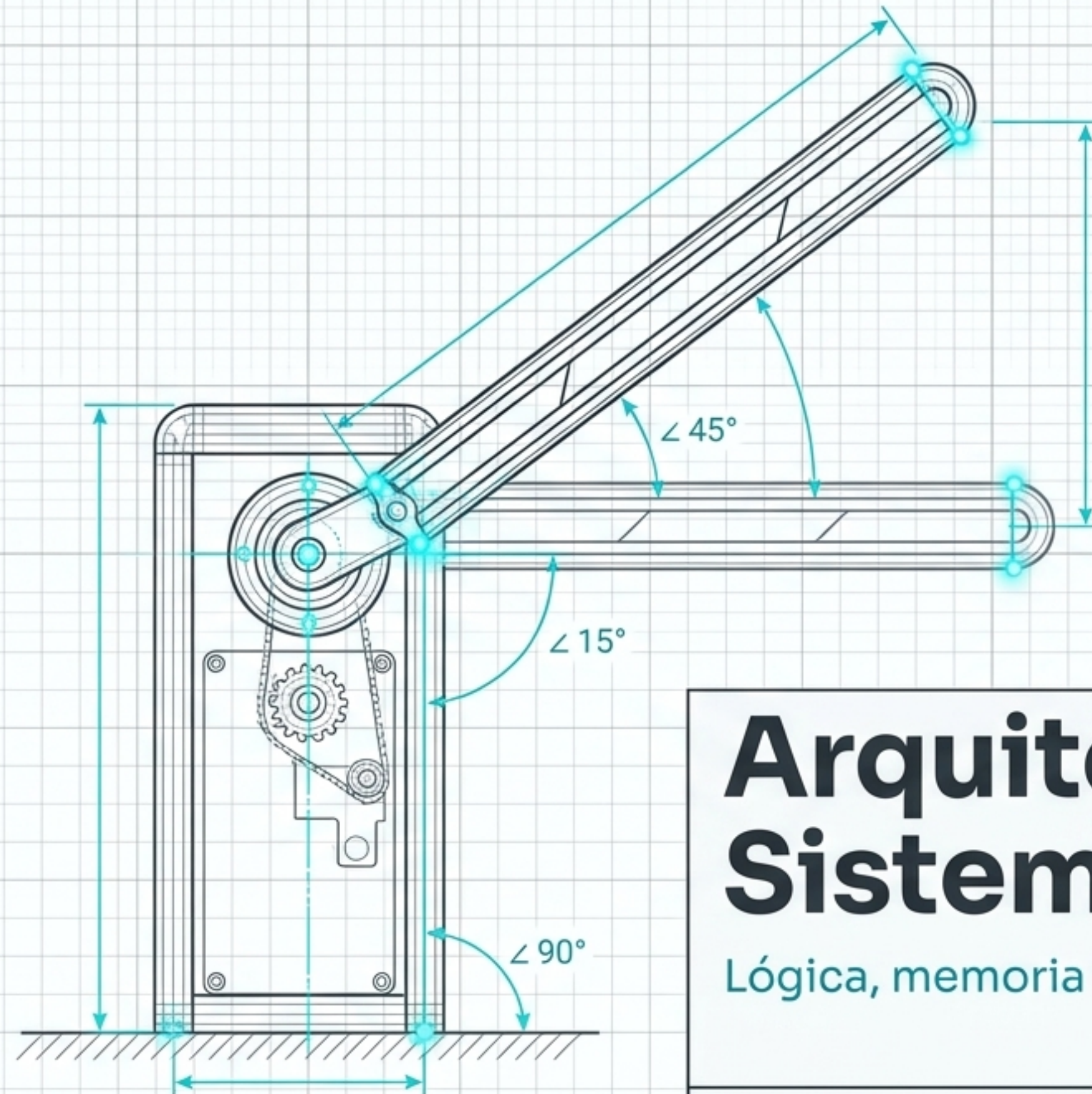




Arquitectura de un Sistema Automatizado

Lógica, memoria y control para un parqueadero inteligente.

The Modern Engineering Workbench

Capa 1: Interacción Física



Simular el hardware. Conectar eventos reales (botones) con respuestas mecánicas (servomotores).

Capa 2: Gestión de Datos



Crear y actualizar variables (**puestosDisponibles**) para rastrear el espacio en tiempo real.

Capa 3: Toma de Decisiones



Implementar condicionales (**Si/Sino**) para que el sistema reaccione autónomamente ante múltiples estados.

The Modern Engineering Workbench

ENTRADAS (Eventos Físicos)

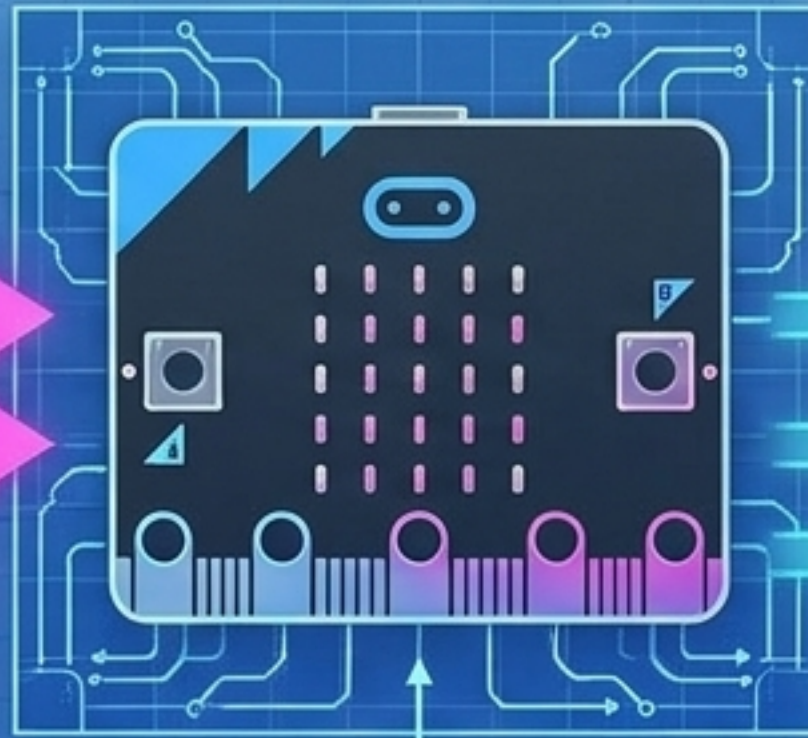


Botón A:
Ingreso de Vehículo



Botón B: Salida de Vehículo

Unidad Lógica Central



Unidad Lógica Central

Procesa reglas y memoria

SALIDAS (Actuadores y UI)



Pin P15: Barrera de Entrada



Pin P16: Barrera de Salida



Pantalla: Disponibilidad y Alertas

The Modern Engineering Workbench

Capa 1: Mecánica de Entrada y Salida

al presionar Botón A



girar servo P15 a 0°



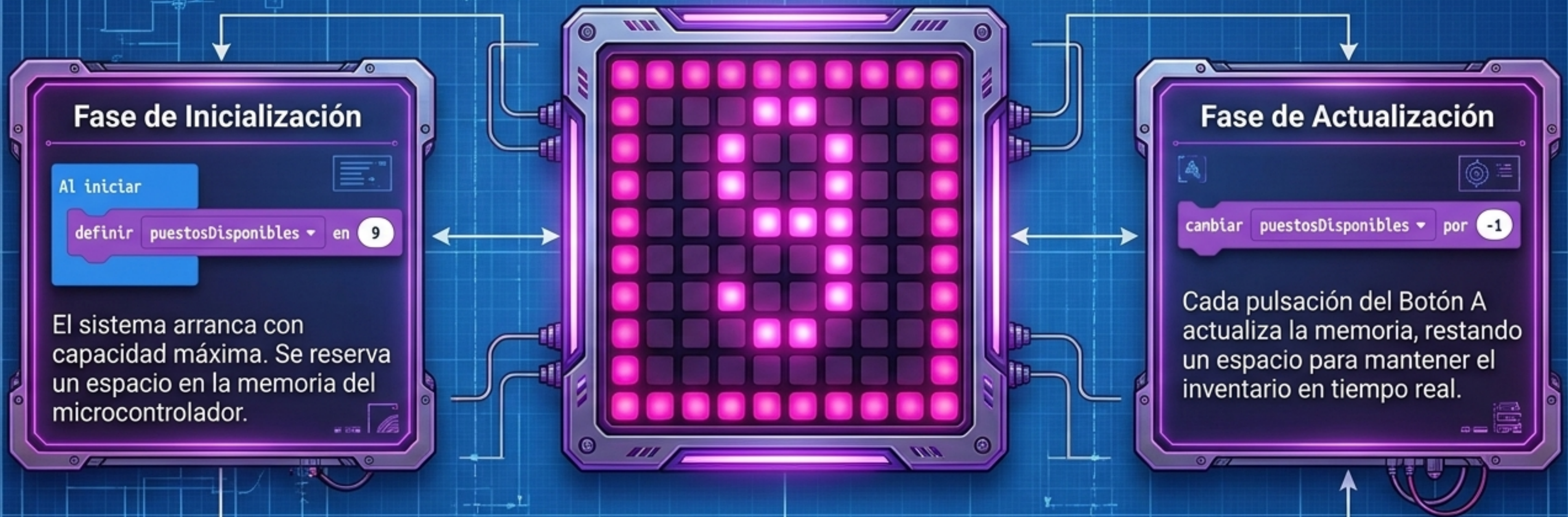
esperar 5 segundos



girar servo P15 a 90°

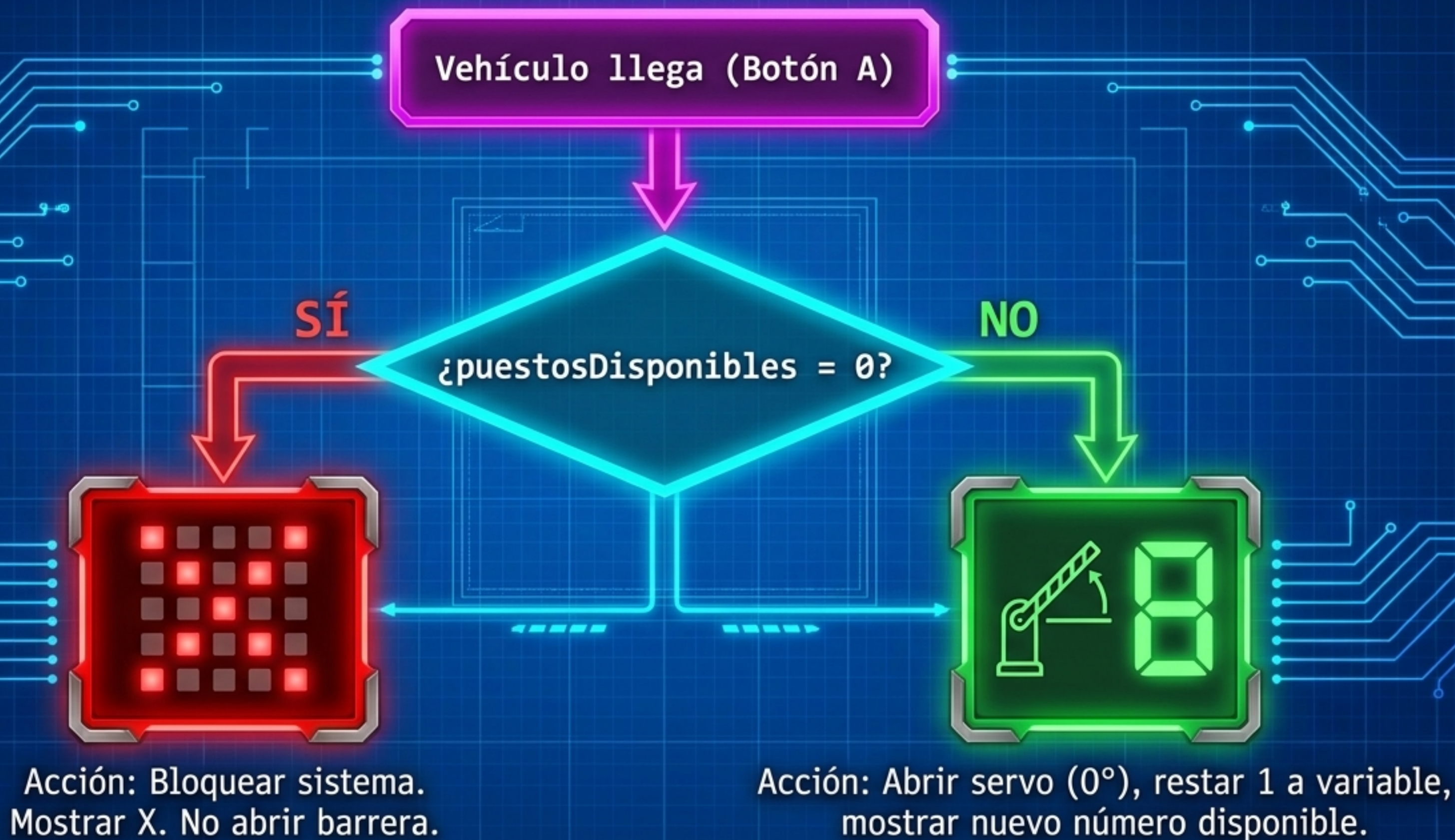


Capa 2: Memoria Dinámica



**Capacidad Inicial - Vehículos Ingresados
= puestosDisponibles**

Capa 3: El Peaje Lógico (Condicionales)



Síntesis: El Algoritmo Ensamblado



1 **El Gatillo**
Inicia el evento físico.

2 **El Juez**
Evalúa si quedan
puestos (> 0).

5 **La Restricción**
Protege contra el
sobrecupo.

al presionar Botón A

si... $\text{puestosDisponibles} \leq 9$ y $\text{puestosDisponibles} > 0$

girar servo en pin P15 a 0 grados

pausa (ms) 5000

girar servo en pin P15 a 90 grados

cambiar $\text{puestosDisponibles}$ por -1

mostrar número $\text{puestosDisponibles}$

sino

mostrar icono X

pausa (ms) 3000

mostrar número $\text{puestosDisponibles}$

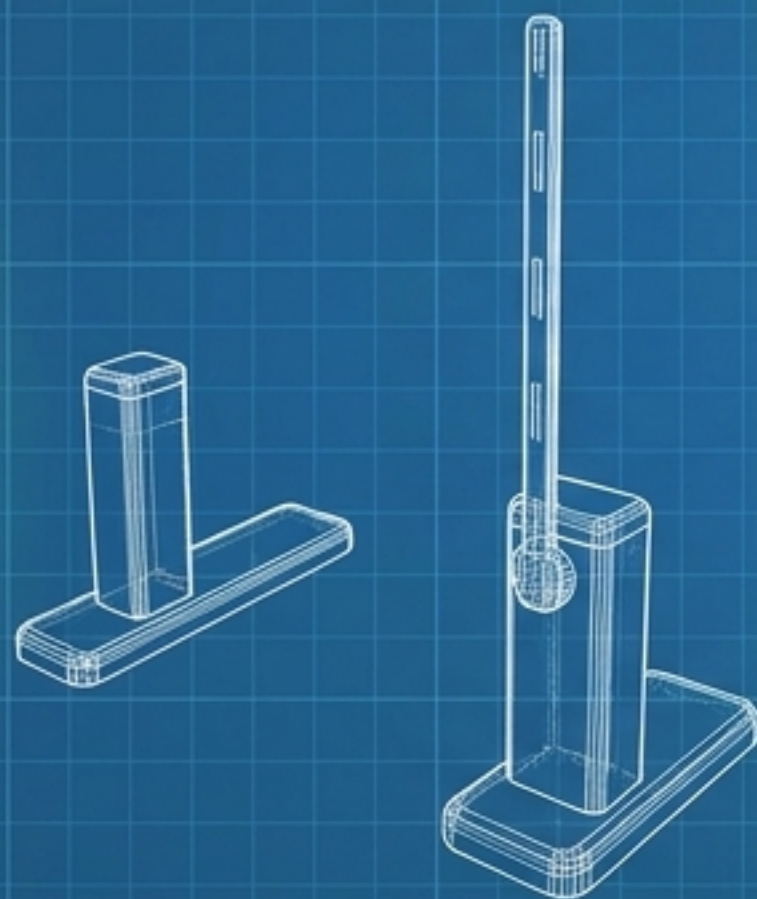
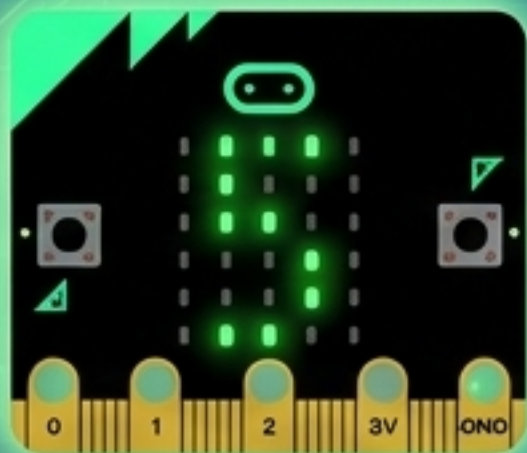
3 **La Ejecución**
Abre la barrera
mecánicamente.

4 **La Actualización**
Mantiene la memoria
sincronizada.

Los Dos Estados del Sistema

ESTADO: DISPONIBLE

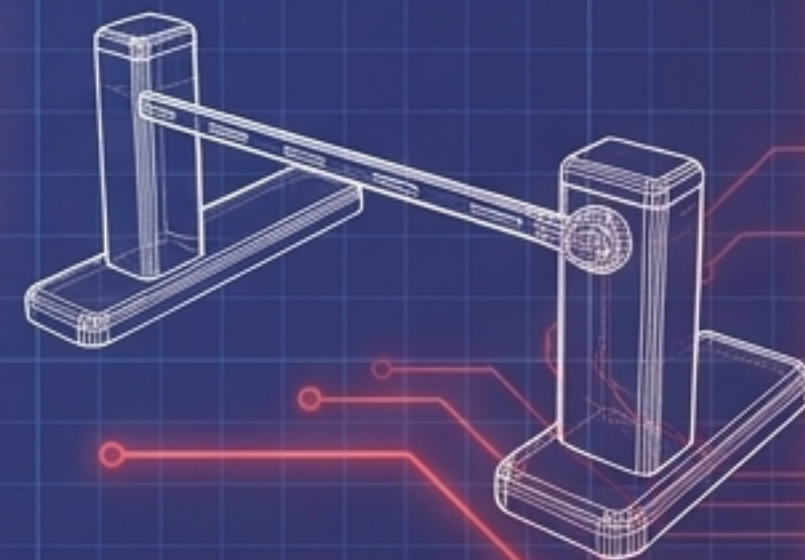
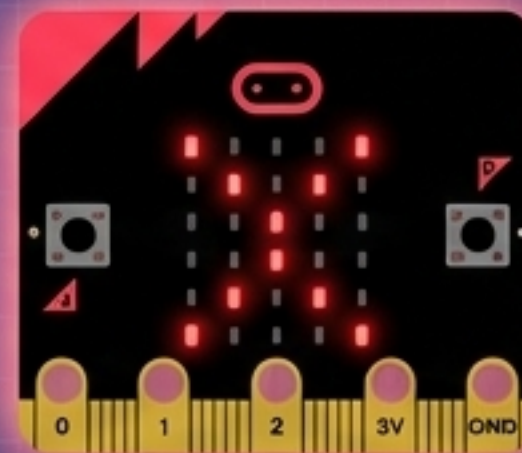
Condición: `puestosDisponibles > 0` => [VERDADERA]



Resultado: Acceso permitido. El servo gira y el contador matemático se actualiza.

ESTADO: LLENO

Condición: `puestosDisponibles = 0` => [VERDADERA]



Resultado: Acceso denegado. El servo permanece estático para evitar sobrecupo.

Matriz de Traducción: Lógica Universal

Bloques Visuales



girar servo en pin **P15** a **0** grados

mostrar número **puestosDisponibles**

Código Python

```
if puestosDisponibles <= 9 and  
    puestosDisponibles > 0:  
else:
```

```
pins.servo_write_pin(AnalogPin.P15, 0)
```

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

"El lenguaje cambia; la arquitectura del pensamiento se mantiene."

Ingeniería Crítica: Análisis de Vulnerabilidades (Casos de Borde)

RIESGO IDENTIFICADO	CONSECUENCIA EN EL SISTEMA	SOLUCIÓN ARQUITECTÓNICA
Falsos Ingresos (Usuario presiona Botón A, pero el vehículo no entra).	Desfase de inventario. El sistema marca menos cupos de los que realmente existen en el nivel físico.	Integrar un sensor de proximidad ultrasónico en la barrera en lugar de depender de un botón manual.
Desbordamiento Negativo (El contador baja de 0).	Errores matemáticos en la pantalla LED y fallos en la lógica de apertura y cierre.	Validación estricta en el bloque condicional inicial asegurando que $(puestosDisponibles > 0)$ antes de restar.

Checkpoint del Sistema



Implementar condicionales interactivos (Si/Sino).



Rastrear estados con memoria dinámica.



Proteger el sistema contra vulnerabilidades y sobrecupos.

NOTA DIAGNÓSTICA: Si el sistema falla en alguna de estas métricas durante la simulación, revise la Capa Lógica (Diapositiva 6) o consulte a su arquitecto principal (Docente).