

Anexo 5.1 – Posible solución Sesión 4

Optimización del código

Revisa el código actual e identifica áreas de mejora.

Genera el código necesario para manejar la lógica común de entrada y salida.

Implementación de nuevas características

Añade un contador de tiempo para simular el cobro por tiempo de estacionamiento.

Implementa un sistema de alarma cuando el parqueadero está casi lleno.

Pruebas y refinamiento final

Realiza pruebas exhaustivas de todas las funcionalidades.

Ajusta el código según sea necesario basándote en los resultados de las pruebas.

```
# Python code
#
MAX_ESPACIOS = 9
CASI_LLENO = 2
puestosDisponibles = MAX_ESPACIOS
tiempoTotal = 0
basic.show_string("'" +
str(puestosDisponibles))
pins.servo_write_pin(AnalogPin.P15, 90)
pins.servo_write_pin(AnalogPin.P16, 90)

def mover_barrera(pin, abrir):
    angulo = 0 if abrir else 90
    pins.servo_write_pin(pin, angulo)
    basic.pause(5000)
def manejar_vehiculo(es_entrada):
    global puestosDisponibles, tiempoTotal
    if es_entrada and puestosDisponibles > 0:
        mover_barrera(AnalogPin.P15, True)
        puestosDisponibles += -1
```

```
basic.show_arrow(ArrowNames.NORTH)
elif not es_entrada and puestosDisponibles
< MAX_ESPACIOS:
    mover_barrera(AnalogPin.P16, True)
    puestosDisponibles += 1
    basic.show_arrow(ArrowNames.SOUTH)
    tiempoTotal += 1 # Simula cobro por
tiempo
else:
    basic.show_icon(IconNames.NO)
    basic.pause(3000)
    actualizar_display()
    if puestosDisponibles <= CASI_LLENO:
        basic.show_icon(IconNames.DIAMOND)
        basic.pause(1000)
```

```
def on_button_pressed_a():
    manejar_vehiculo(True)
```

```
def on_button_pressed_b():
    manejar_vehiculo(False)
```

```
input.on_button_pressed(Button.A, on_
button_pressed_a)
input.on_button_pressed(Button.B, on_
button_pressed_b)
```

```
def on_forever():
    global tiempoTotal
    basic.pause(60000) # Espera 1 minuto
    tiempoTotal += 1
    if tiempoTotal % 5 == 0: # Cada 5 minutos
        basic.show_string("T:" + str(tiempoTotal))
        basic.pause(1000)
        actualizar_display()
```

```
basic.forever(on_forever)
```

```
pins.servo_write_pin(pin, 90)
```

```
def actualizar_display():
    basic.show_number(puestosDisponibles)
```