



Componente o Característica	Descripción de Funcionalidad	Pin o Variable Asociada	Condición de Activación	Acción del Sistema	Objetivo de Aprendizaje	Fuente
Servo de Entrada	Controla la barrera física para permitir el ingreso de vehículos al parqueadero.	AnalogPin.P15 / Variable <code>puestosDisponibles</code>	Presión del botón A y <code>puestosDisponibles > 0</code> .	Mueve el servo a 0 grados para abrir y luego vuelve a 90 grados.	Comprender el control de actuadores analógicos y lógica condicional.	[1]
Servo de Salida	Controla la barrera física para permitir la salida de vehículos y registra el tiempo.	AnalogPin.P16 / Variable <code>puestosDisponibles</code>	Presión del botón B y <code>puestosDisponibles < MAX_ESPACIOS</code> .	Mueve el servo a 0 grados para abrir y suma tiempo al cobro.	Implementar lógica de control para procesos de salida y flujo de datos.	[1]
Servomotores (Barreras)	Control físico de acceso al parqueadero mediante el movimiento de barreras.	No especificado en la fuente	Detección de vehículo o disponibilidad de puestos.	Movimiento del servo para abrir o cerrar barreras.	Evaluar y mejorar el diseño general del sistema.	[2]
Led Rojo	Sistema de alarma visual para indicar que no hay disponibilidad en el parqueadero.	Pin (P1) / Variable <code>puestosDisponibles</code>	<code>puestosDisponibles = 0</code>	Escribir señal digital Alta (ALTA) en el pin P1 para encender el led.	Implementar características adicionales al sistema y utilizar técnicas avanzadas de programación.	[2]
Led Naranja	Sistema de alarma visual para indicar baja disponibilidad de espacios.	Pin (P2) / Variable <code>puestosDisponibles</code>	<code>puestosDisponibles > 0</code> y <code>puestosDisponibles <= 3</code>	Escribir señal digital Alta (ALTA) en el pin P2 para encender el led.	Optimizar el código para mejorar su eficiencia y legibilidad.	[2]
Sistema de Alarma (Casi Lleno)	Alerta visual que indica cuando el parqueadero tiene pocos espacios disponibles.	Variable <code>CASI_LLENO</code> (valor 2)	<code>puestosDisponibles <= CASI_LLENO</code> .	Muestra un icono de diamante en la matriz de LEDs.	Aplicar alarmas visuales basadas en umbrales de variables.	[1]
Contador de Tiempo / Cobro	Simulación de evolución hacia sistemas reales con integración de pagos y cálculo de costos.	Variable <code>tiempoTotal</code> / Variable de costo	Cada 60,000 ms (1 minuto), al salir un vehículo o escalamiento a sistema real.	Incrementa <code>tiempoTotal</code> , muestra el valor cada 5 minutos y simula integración con pagos.	Evaluar las implicaciones de escalar un proyecto a un sistema real y simular cobros mediante variables.	[2][1]
Display Matriz LED	Muestra información relevante como puestos disponibles, flechas de dirección o alertas.	<code>puestosDisponibles</code> / <code>actualizar_display()</code>	Cambios en el estado del parqueadero o temporizador activo.	Muestra números, cadenas de texto (T:) o iconos (NO, diamante).	Gestionar la interfaz de usuario mediante retroalimentación visual.	[1]
Optimización de Código	Refactorización para manejar la lógica común de entrada y salida de forma eficiente.	<code>manejar_vehiculo(es_entrada)</code>	Identificación de redundancias en el código original.	Uso de funciones parametrizadas para reducir líneas de código.	Mejorar la eficiencia y legibilidad del código de programación.	[1]
Bloque de Espera (Pausa)	Control de tiempo para la visualización de las señales y parpadeos.	Bloque de espera / <code>basic.pause</code>	Después de una instrucción de escritura digital.	Crear la sensación de parpadeo esperando 1 segundo (1000 ms).	Analizar la equivalencia entre programación por bloques y Python.	[2]

[1] Anexo 5.1 – Posible solución Sesión 4.pdf

[2] Guía 3. Parqueaderos - Sesión 4.pdf