7°3°

SMART PARKING SYSTEM

Name: Tiago Josue Ilacqua Vazquez

School: “EESTN°2 Tres de Febrero”

7°3°

SMART PARKING SYSTEM

A supermarket aims to modernize its parking lot to improve vehicle entry and exit processes. The goal is to discourage potential access to the parking lot when there are no available spaces. Additionally, since the parking lot is outdoors, the lighting system must be automatically controlled to operate only under specific ambient light conditions.

To achieve this, a comprehensive control system is proposed, which includes the following:

---

1. EXTERNAL VISUAL INDICATOR

The system must provide a clear indication or alert to drivers, from a distance of no less than 50 meters, about the availability of parking spaces. This will help drivers decide whether or not to approach the parking lot. This functionality will be implemented using the VISUAL INDICATOR described later.

---

2. SENSORS, BARRIERS, AND TRAFFIC LIGHTS AT ENTRY AND EXIT POINTS

ENTRY POINT:

- The system must detect the presence of a vehicle and welcome the driver via an LCD display.
- It should then activate the green light on the traffic signal and raise the barrier.
- After 30 seconds, the barrier must lower automatically, and the red light should activate.

EXIT POINT:

- The system must detect the vehicle's presence and thank the driver for their visit via an LCD display.
- It should then activate the green light on the traffic signal and raise the barrier.
- After 30 seconds, the barrier must lower automatically, and the red light should activate.

The vehicle's presence will be detected using magnetic sensors placed just before the entry and exit barriers. Confirmation from the driver will be required by pressing a button.

---

3. AUTOMATIC PARKING LOT LIGHTING

The system must control the turning on and off of the parking lot's lighting, ensuring it operates ONLY when the ambient light intensity is below 20 LUX. For this purpose, an appropriate light sensor must be installed.

---

4. CENTRAL CONTROL PANEL

The control panel located in the Monitoring Center must include the following elements, ensuring clear information display:

- **3 (THREE) LUMINOUS INDICATORS:**
  - **RED:** No spaces available.
  - **YELLOW:** Fewer than 5 spaces available.
  - **GREEN:** Spaces available.
- **1 (ONE) 16X2 LCD DISPLAY:** Displays the number of available parking spaces.
  Whenever a vehicle enters or exits, the system must update the number of available spaces on the LCD display and activate the corresponding luminous indicator.

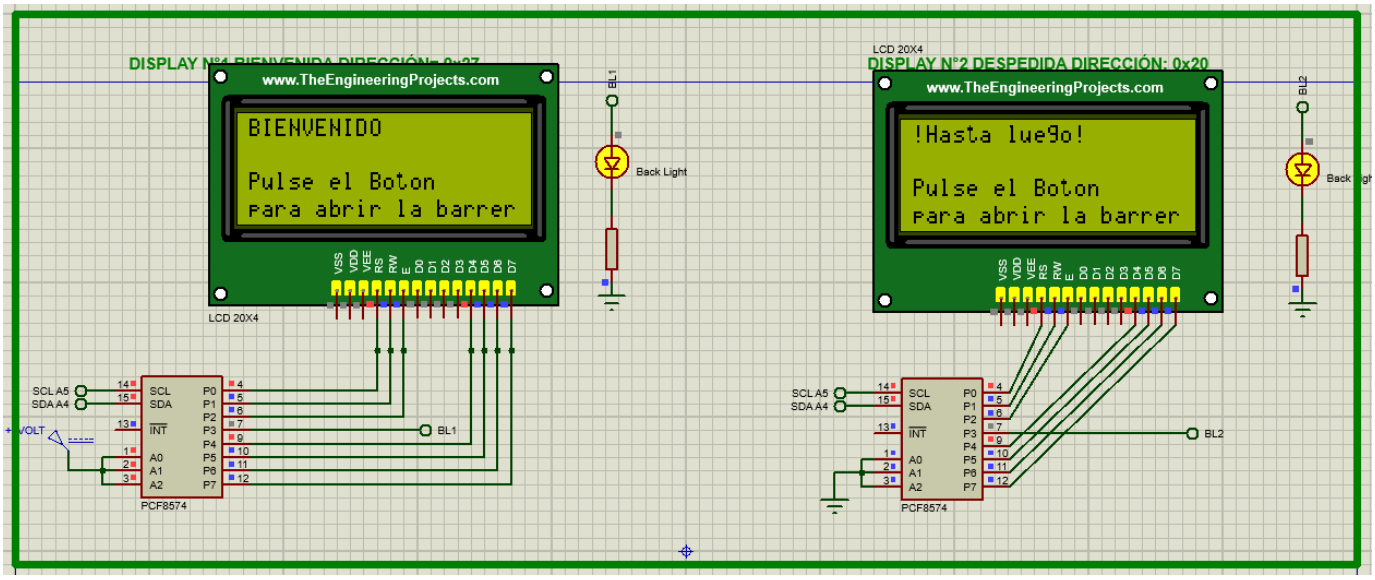
TECHNOLOGICAL PROPOSAL

All devices required to fulfill the project will be structured using an ARDUINO board. The project will include a detailed technical report outlining the program used, electrical circuits, and a complete simulation of the control system.

WELCOME AND FAREWELL DISPLAYS

Using I2C, the system initializes two I2C LCD displays with different addresses, connected to the SCL and SDA pins of the PCF8574 controller on the Arduino.

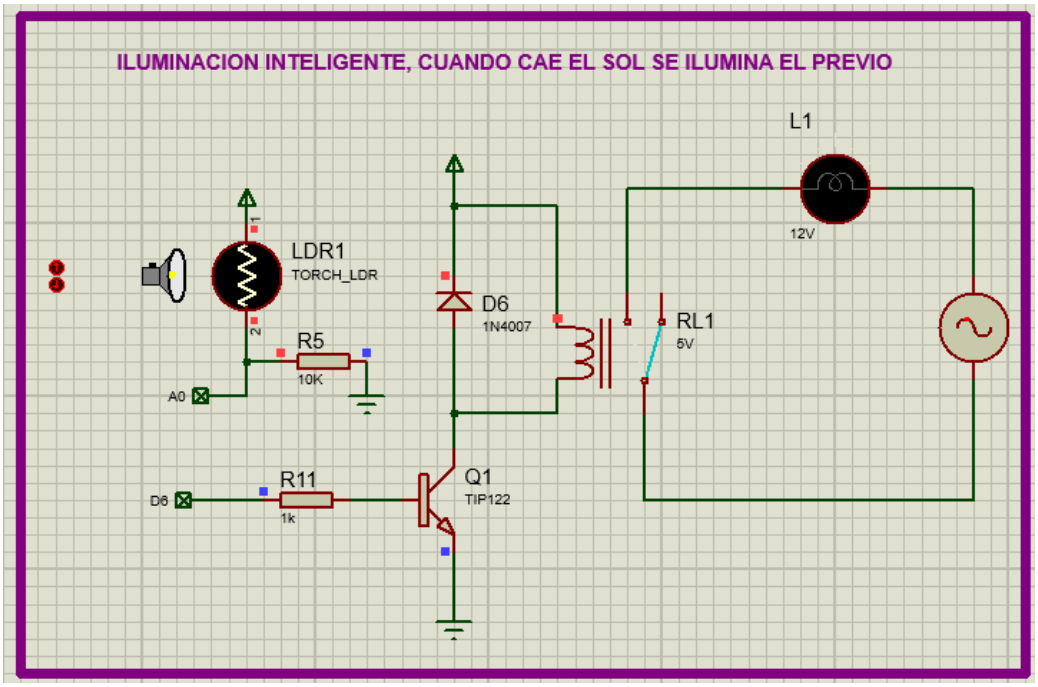
- **LCD1:** Address 0x27, with 20 columns and 4 rows.
- **LCD2:** Address 0x20, with 20 columns and 4 rows.



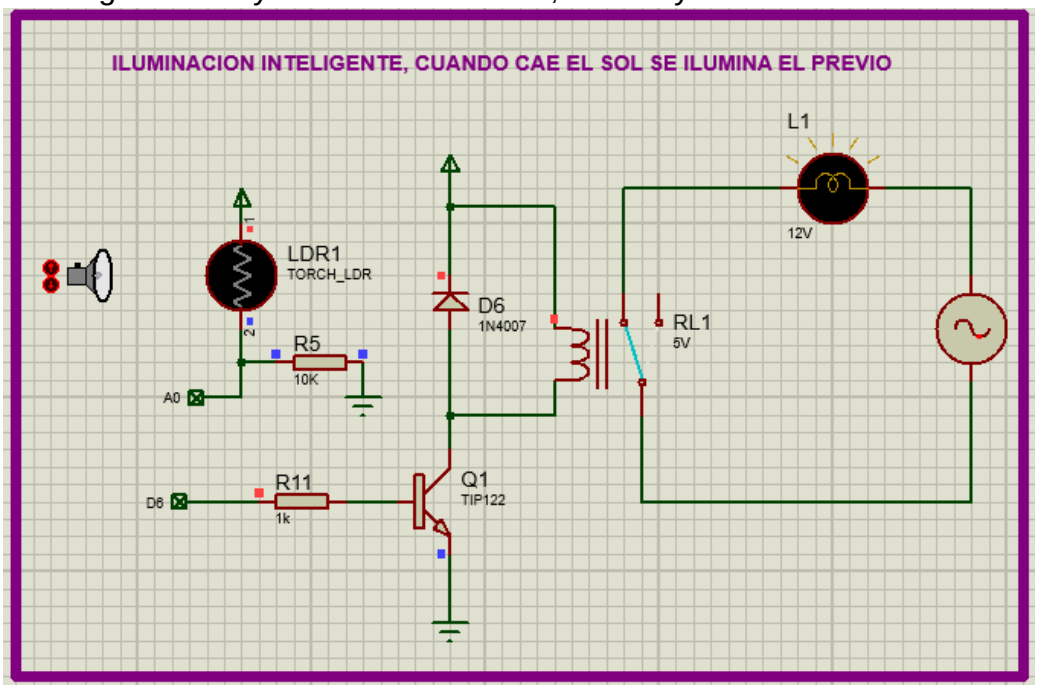
SMART LIGHTING IN THE PARKING LOT

Using an LDR and an analog reading from pin A0, the system reads and processes the sensed data. A relay will be configured with its normally open (NO) output connected to the high-power lighting system and its coil connected to digital pin D6.

If the light intensity exceeds 220 lux, the relay will turn off.

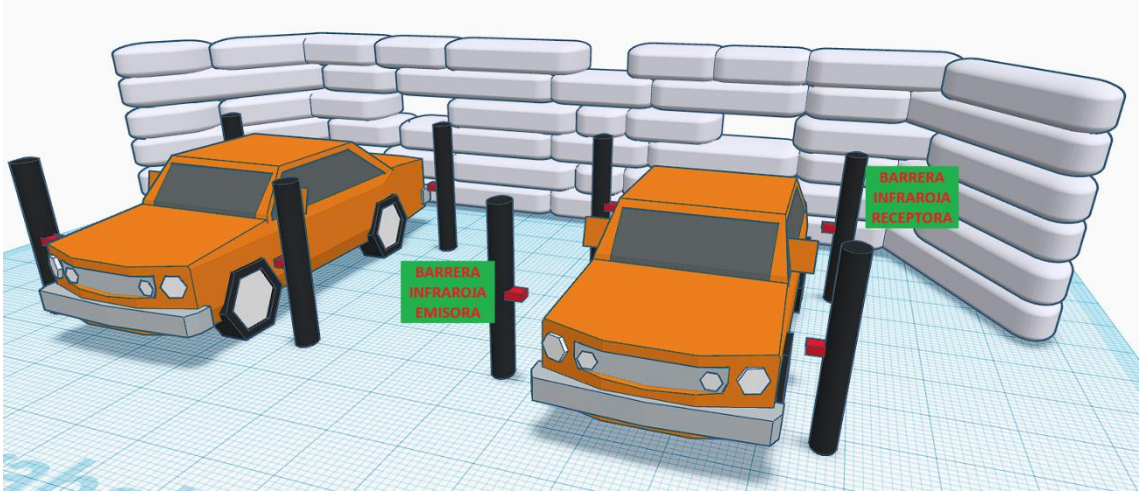


If the light intensity is less than 220 lux, the relay will turn on.



PARKING SPACE SENSORS

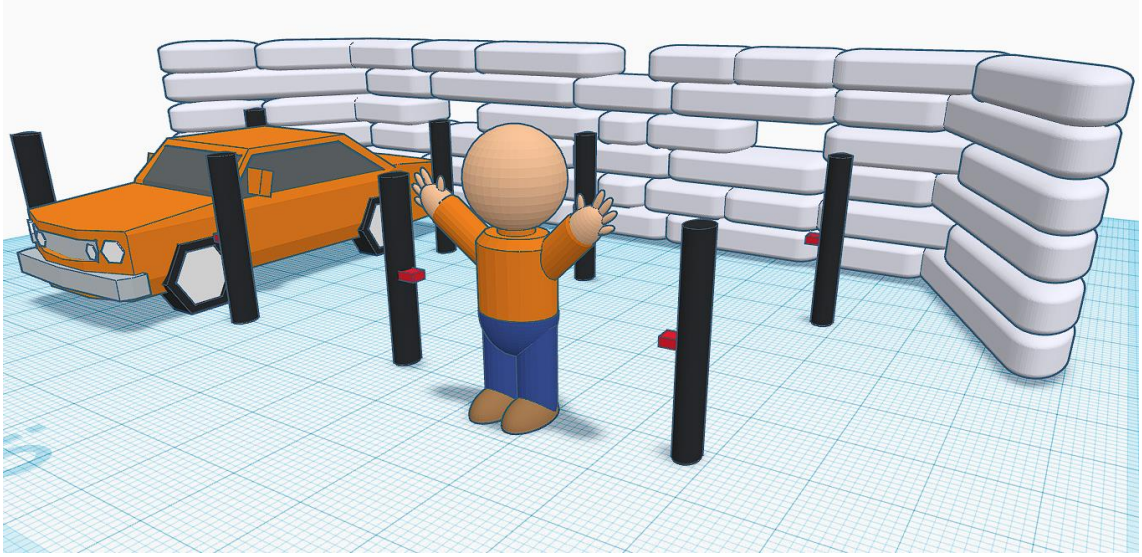
Using two infrared barriers (Emitter-Receiver) at each end of the parking space, the system will detect whether a car is parked or not.



IN PROTEUS WE WILL SEE

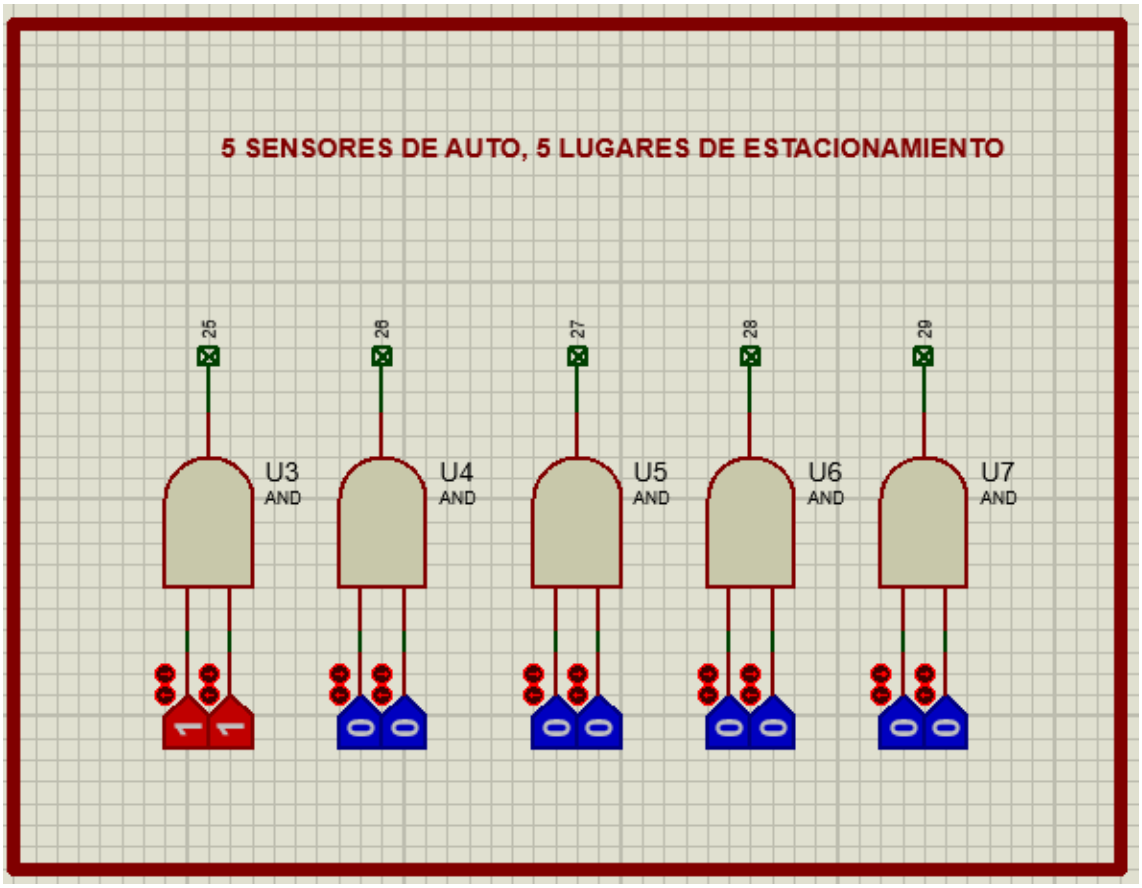
An AND gate simplifying the sensor signals and optimizing control lines. Additionally, two sensors were used to ensure that if an object like a shopping cart or a person crosses, the program does not register it as an unavailable

parking space.



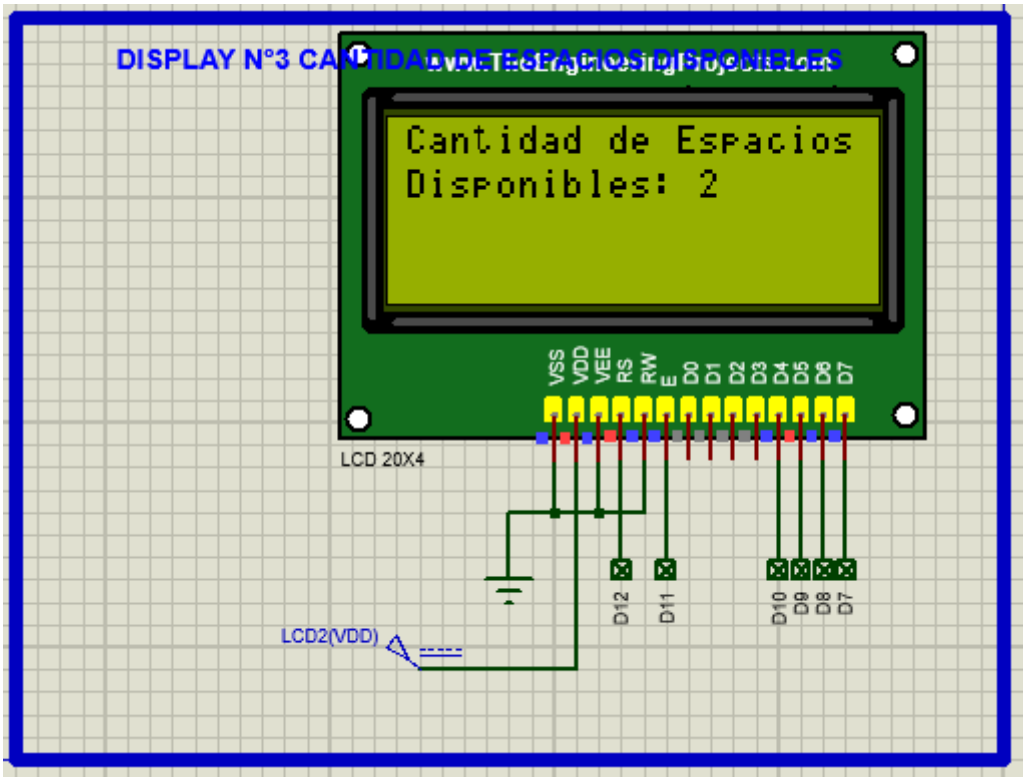
The condition must be that both sensors are interrupted by an object. If this condition is met, a pulse will be sent to the Arduino to indicate that parking space N1 is occupied. In this case, we simulate a total of 5 parking spaces, meaning 5 AND gates.

The sensor variables will be stored in the array:
int sensores[] = {25, 26, 27, 28, 29};



CONTROL PANEL

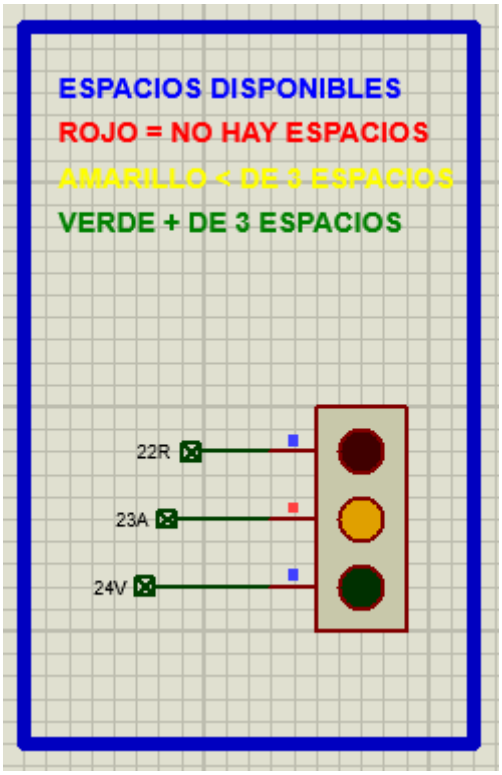
On a 20x4 Crystal LCD display configured with the **LiquidCrystal** library, the number of available parking spaces will be displayed. It will also show the array of available spaces through a logic-based mathematical calculation: espaciosDisponibles = numSensores - espaciosOcupados;



EXTERNAL VISUAL INDICATOR

This three-color traffic light (RED, YELLOW, and GREEN) will activate based on the number of available parking spaces:

- **RED LED**: No spaces available.
- **YELLOW LED**: Two or fewer spaces available.
- **GREEN LED**: Three or more spaces available.

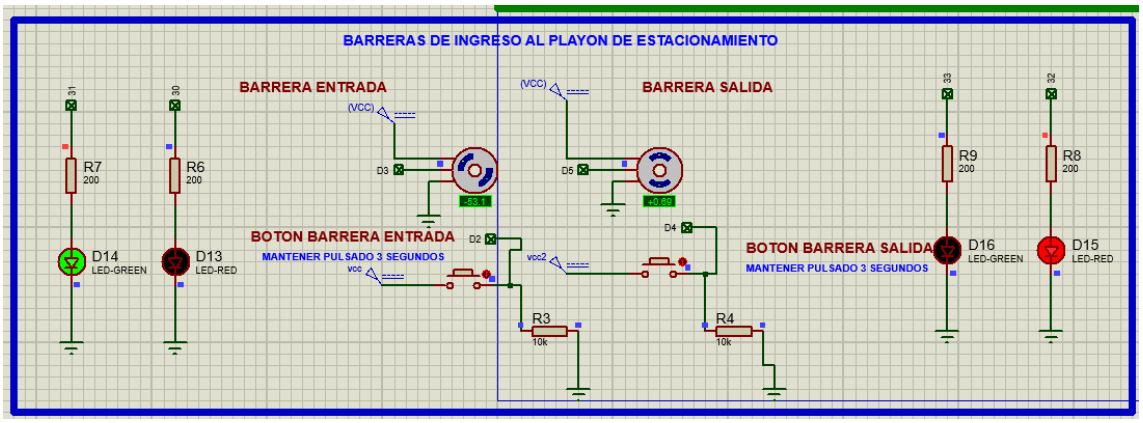


ENTRY TO THE PARKING LOT USING BARRIERS WITH SERVOMOTORS

The system consists of two servomotors: one for the entrance barrier and another for the exit barrier. When a button is pressed, the barriers will lift to **90 degrees** for **30 seconds** and then return to the initial position of **0 degrees**.

- The entrance barrier will only activate if parking spaces are available; otherwise, it cannot be triggered.
- While no button is pressed, the **RED indicators** will remain on.

But if the barrier is opening and remains open, the **GREEN indicator** will turn on and the **RED indicator** will turn off.



Arduino program:

```
#include <LiquidCrystal_I2C.h>
#include <LiquidCrystal.h>
#include <Servo.h>

// Define the pins for the buttons and servos
#define buttonPin 2
#define servoPin 3
#define buttonPin2 4
#define servoPin2 5

// Initialize two Servo objects to control two servos
Servo servo;
Servo servo2;

// DECLARATION OF THE ADDRESSES SET IN PCF8574
// Initialize two I2C LCD screens at different I2C addresses
LiquidCrystal_I2C lcd1(0x27, 20, 4);  // LCD1 at address 0x27 with 20
columns and 4 rows
LiquidCrystal_I2C lcd2(0x20, 20, 4);  // LCD2 at address 0x20 with 20
columns and 4 rows

// Initialize an LCD without I2C interface using specific pins
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

// Configuration of pins for parking space sensors
const int sensores[] = {25, 26, 27, 28, 29};  // Pins for the sensors
const int numSensores = 5;                    // Total number of sensors
const int ledRojo = 22;                        // Pin for the red LED
const int ledAmarillo = 23;                    // Pin for the yellow LED
const int ledVerde = 24;                       // Pin for the green LED
const int ledRojoBarreraEntrada = 30;          // Pin for the red LED at
the entry barrier
const int ledVerdeBarreraEntrada = 31;         // Pin for the green LED
at the entry barrier
const int ledRojoBarreraSalida = 32;           // Pin for the red LED at
the exit barrier
```

```
const int ledVerdeBarreraSalida = 33;          // Pin for the green LED
at the exit barrier

int luzRele = 6;                               // Pin for the light relay
int a, b, c;                                   // Auxiliary variables

void setup() {
  Serial.begin(9600);                          // Start serial
communication
  servo.attach(servoPin);                      // Attach the main servo
pin
  servo2.attach(servoPin2);                    // Attach the second servo
pin

  pinMode(luzRele, OUTPUT);                    // Set the relay pin as
output
  pinMode(buttonPin, INPUT_PULLUP);            // Set button 1 as input
with pull-up resistor
  pinMode(buttonPin2, INPUT_PULLUP);           // Set button 2 as input
with pull-up resistor

  pinMode(ledRojo, OUTPUT);                    // Set red LED as output
  pinMode(ledAmarillo, OUTPUT);                // Set yellow LED as
output
  pinMode(ledVerde, OUTPUT);                   // Set green LED as output
  pinMode(ledRojoBarreraEntrada, OUTPUT);      // Set entry barrier red
LED as output
  pinMode(ledVerdeBarreraEntrada, OUTPUT);     // Set entry barrier green
LED as output
  pinMode(ledRojoBarreraSalida, OUTPUT);       // Set exit barrier red
LED as output
  pinMode(ledVerdeBarreraSalida, OUTPUT);      // Set exit barrier green
LED as output

  lcd.begin(20, 4);                            // Initialize the standard
LCD with 20 columns and 4 rows

  // Set the sensor pins as inputs
  for (int i = 0; i < numSensores; i++) {
    pinMode(sensores[i], INPUT);               // Set each sensor as
input
  }

  lcd1.init();                                 // Initialize the first
I2C LCD
  lcd2.init();                                 // Initialize the second
I2C LCD
```

```
  lcd1.backlight();                              // Turn on the backlight
of the first LCD
  delay(300);
  lcd2.backlight();                              // Turn on the backlight
of the second LCD
  delay(300);

  // Welcome messages on both LCDs
  lcd1.setCursor(0, 0);
  lcd1.print("WELCOME");
  lcd1.setCursor(0, 2);
  lcd1.print("Press the Button");
  lcd1.setCursor(0, 3);
  lcd1.print("to open the barrier");
  delay(1000);

  lcd2.setCursor(0, 0);
  lcd2.print("Goodbye!");
  lcd2.setCursor(0, 2);
  lcd2.print("Press the Button");
  lcd2.setCursor(0, 3);
  lcd2.print("to open the barrier");
  delay(1000);
}

void loop() {
  int posi = 90;                                 // Initial position of the
first servo
  int buttonState = digitalRead(buttonPin);      // Read the state of
button 1
  int posi2 = 90;                                // Initial position of the
second servo
  int buttonState2 = digitalRead(buttonPin2);  // Read the state of
button 2

  // Control of the first servo when button 1 is pressed

  int espaciosOcupados = 0;
  // Count how many sensors are activated (occupied spaces)
  for (int i = 0; i < numSensores; i++) {
    if (digitalRead(sensores[i]) == HIGH) { // If the sensor is activated
      espaciosOcupados++;
    }
  }

  int espaciosDisponibles = numSensores - espaciosOcupados;

  // Control of the LEDs based on the number of available spaces
  if (espaciosDisponibles == 0) {
```

```
    digitalWrite(ledRojo, HIGH);     // Red LED if no spaces are
available
    digitalWrite(ledAmarillo, LOW);
    digitalWrite(ledVerde, LOW);
  } else if (espaciosDisponibles <= 2) {
    digitalWrite(ledAmarillo, HIGH); // Yellow LED if few spaces are
available
    digitalWrite(ledRojo, LOW);
    digitalWrite(ledVerde, LOW);
  } else {
    digitalWrite(ledVerde, HIGH);    // Green LED if many spaces are
available
    digitalWrite(ledRojo, LOW);
    digitalWrite(ledAmarillo, LOW);
  }

  a = analogRead(A0);                // Read the light sensor value
  b = map(a, 0, 1023, 0, 255);       // Map the light value
  Serial.println(b);
  if (b < 220) {                     // If the light is low, turn on the
relay
    digitalWrite(luzRele, HIGH);
  }
  if (b > 220) {                     // If the light is high, turn off
the relay
    digitalWrite(luzRele, LOW);
  }

  // Display the number of available spaces on the standard LCD
  lcd.setCursor(0, 0);
  lcd.print("Available Spaces");
  lcd.setCursor(0, 1);
  lcd.print("Available: ");
  lcd.print(espaciosDisponibles);     // Display the number of spaces
  delay(2000);
  lcd.clear();                        // Clear the screen to update

//If the entry button for parking is pressed, move the first servo from
0° to 90° and back, as long as there is at least 1 space available
//A red LED will turn on when the barrier stays down and a green LED will
turn on when the barrier is opening and stays open.
digitalWrite(ledRojoBarreraEntrada, HIGH);
if (buttonState == HIGH) {
    servo.write(posi);
  digitalWrite(ledRojoBarreraEntrada, LOW);

    for (posi = 90; posi >= 0; posi -= 1 & espaciosDisponibles >= 1 ) {
      servo.write(posi);
      delay(27);
```

```
      Serial.println(posi);
      digitalWrite(ledVerdeBarreraEntrada, HIGH);
    }
    for (posi = 0; posi <= 90; posi += 1) {
      servo.write(posi);
      delay(27);
      Serial.println(posi);
      digitalWrite(ledVerdeBarreraEntrada, HIGH);
    }
    digitalWrite(ledVerdeBarreraEntrada, LOW);
    }


  //If the exit button is pressed, move the second servo from 0° to 90°
and back.
  //A red LED will turn on when the barrier stays down and a green LED
will turn on when the barrier is opening and stays open.
  digitalWrite(ledRojoBarreraSalida, HIGH);
  if (buttonState2 == HIGH) {
    servo2.write(posi2);
    digitalWrite(ledRojoBarreraSalida, LOW);

    // Move the second servo from 0° to 90° and back
    for (posi2 = 90; posi2 >= 0; posi2 -= 1) {
      servo2.write(posi2);
      delay(27);
      Serial.println(posi2);
      digitalWrite(ledVerdeBarreraSalida, HIGH);
    }
    for (posi2 = 0; posi2 <= 90; posi2 += 1) {
      servo2.write(posi2);
      delay(27);
      Serial.println(posi2);
    }
    digitalWrite(ledVerdeBarreraSalida, LOW);
  }

}
```