



**Tecnológico  
de Monterrey**

Angel Corrales Sotelo (A01562052)  
Guillermo Martínez Montes (A00825023)  
Jesús Palomino Hurtado (A01638492)

***Instituto Tecnológico y de Estudios Superiores de Monterrey***

***“Examen Integrador Módulo 3”***

**Implementación de internet de las cosas**

12 de abril de 2021

# Problemática y Solución

A lo largo del tiempo la humanidad ha desarrollado tecnologías cada vez más avanzada, estas a su vez nos dan muchas facilidades en nuestras actividades del día a día, pero también generan una gran problemática la cual es el uso constante de energía eléctrica. Ya es muy bien sabido que la forma en que se generan estas energías no es 100% limpia y causan un gran impacto en la contaminación del planeta.

Es por esto que nos hemos planteado la creación de un producto que procure el cuidado de la energía eléctrica a la vez de que ayuda en nuestro día a día, basándonos en el objetivo 7: Energía asequible y no contaminante y el objetivo 11: Ciudades y comunidades sostenibles de la ONU.

Nuestra propuesta es la activación y desactivación de aparatos que utilicen energía eléctrica a través de sensores de movimiento los cuales se encargaron de prender y apagar los aparatos de una habitación o pasillo si es que se detecta algún movimiento en la zona, esto para evitar el desperdicio de energía en el caso de que un cuarto se encuentra vacío.

## Datos

Los datos que se utilizan para el prototipo de tinkercad son la distancia del sujeto al sensor, la cual según el estado apaga o prende el led correspondiente. Mientras que en la base de datos se están registrando los datos de tiempo con respecto al encendido y apagado de cada foco.

## Procedimiento

Comenzamos obteniendo los datos a través de un sensor de proximidad que ayudarán a determinar, a través de código, si un foco debe encenderse o mantenerse apagado. Con esto se generará una base de datos con los tiempos en que estuvieron encendidos y que más tarde será utilizada para calcular la energía eléctrica utilizada, así como para la creación y muestra de un dashboard que contendrá gráficos para facilitar la visualización de dicha información al usuario.

## Salidas

Para las salidas tenemos el tiempo total de encendido de los focos, además de que se generará el dato de energía consumida por cada foco, todo esto se mostrará en un dashboard generado en una página de Oracle APEX.

# Estructura base de datos

## *Diccionario de datos*

Field	Constraint	Data type	Description
codl	Primary Key, NOT NULL	int	codigo de luz, auto generado
nombrel	NOT NULL	VARCHAR(100)	nombre de luz
codr	Primary Key, NOT NULL	int	codigo de registro, auto generado
fechaOn	NOT NULL	datetime	fecha en la que se enciende la luz
fechaOff	NOT NULL	datetime	fecha en la que se apaga la luz

## Comandos base de datos

### **Crear base de datos**

CREATE DATABASE Examen Integrador Módulo 3

### **Crear tabla Luces**

CREATE TABLE Luces (codl INT NOT NULL AUTO\_INCREMENT, nombrel VARCHAR(100) NOT NULL, PRIMARY KEY(codl));

### **Insertar valores a tabla Luces**

INSERT INTO Luces (nombrel) VALUES ('Pasillo 1'),('Pasillo 2'),('Pasillo 3'), ('Pasillo 4'), ('Pasillo 5');

### **Crear tabla de registro de estados de Luces**

CREATE TABLE registroLuces (codr INT NOT NULL AUTO\_INCREMENT, codl INT NOT NULL, fechaOn DATETIME NOT NULL, fechaOff DATETIME NOT NULL, PRIMARY KEY(codr), FOREIGN KEY (codl) REFERENCES Luces(codl));

### **Crear valores a tabla de registro de estados de Luces**

INSERT INTO registroLuces (codl, fechaOn, fechaOff) VALUES (1, '04-12-21 20:27:11', '04-12-21 20:27:57'), (2, '15-12-21 12:25:07', '15-12-21 12:25:59'),(3, '15-12-21 16:48:12', '15-12-21 16:48:50'), (4, '04-12-21 20:28:05', '04-12-21 20:28:55'), (5, '10-06-21 16:48:12', '10-06-21 16:48:50')

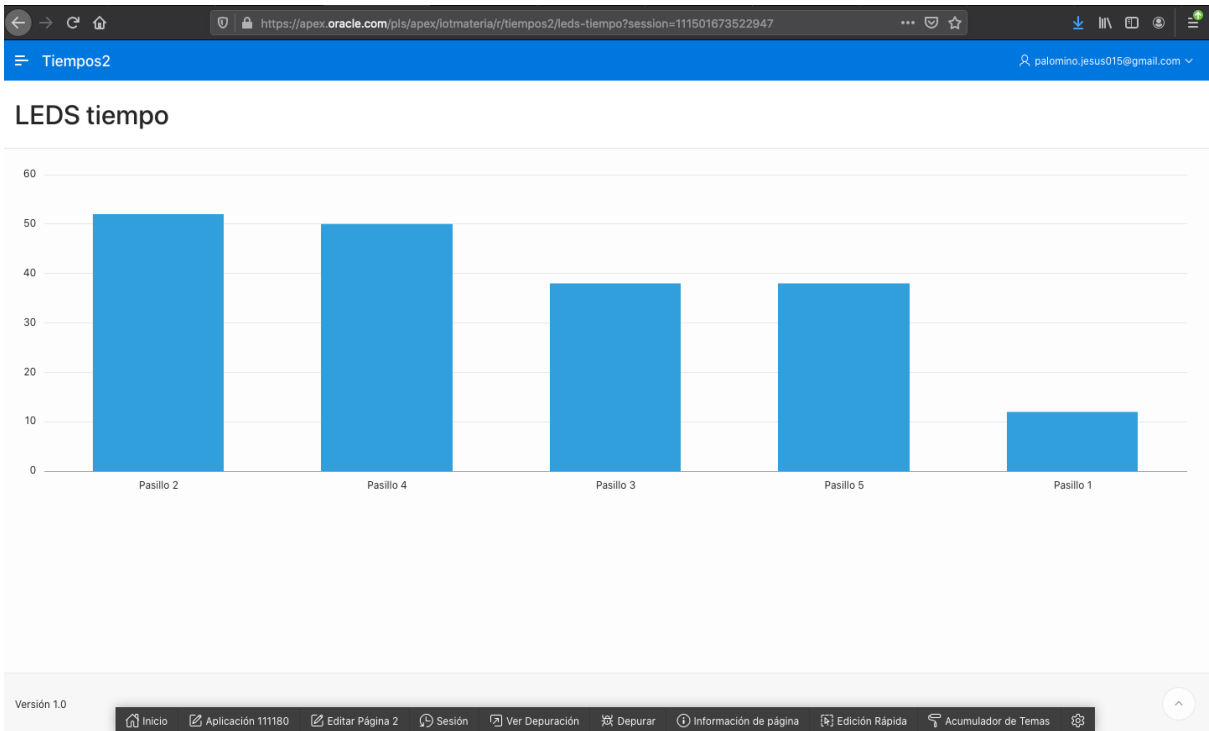
Contar el tiempo total de encendido de cada Luz

SELECT nombrel as 'Nombre luz', SUM(fechaOff-fechaOn) AS 'Total prendido en segundos'  
FROM Luces, registroLuces WHERE Luces.codl=registroLuces.codl GROUP by nombrel

Vista de Datos

	Tabla	Datos	Índices	Modelo	Restricciones	Permisos	Estadísticas	Valores por Defecto de Interfaz de Usuario	Disparadores	Dependencias	SQL	REST	Consulta
	Consulta	Contar Filas	Insertar Fila	Cargar Datos									
	EDITAR		CODL		NOMBREL								
			1		Pasillo 1								
			2		Pasillo 2								
			5		Pasillo 5								
			4		Pasillo 4								
			3		Pasillo 3								

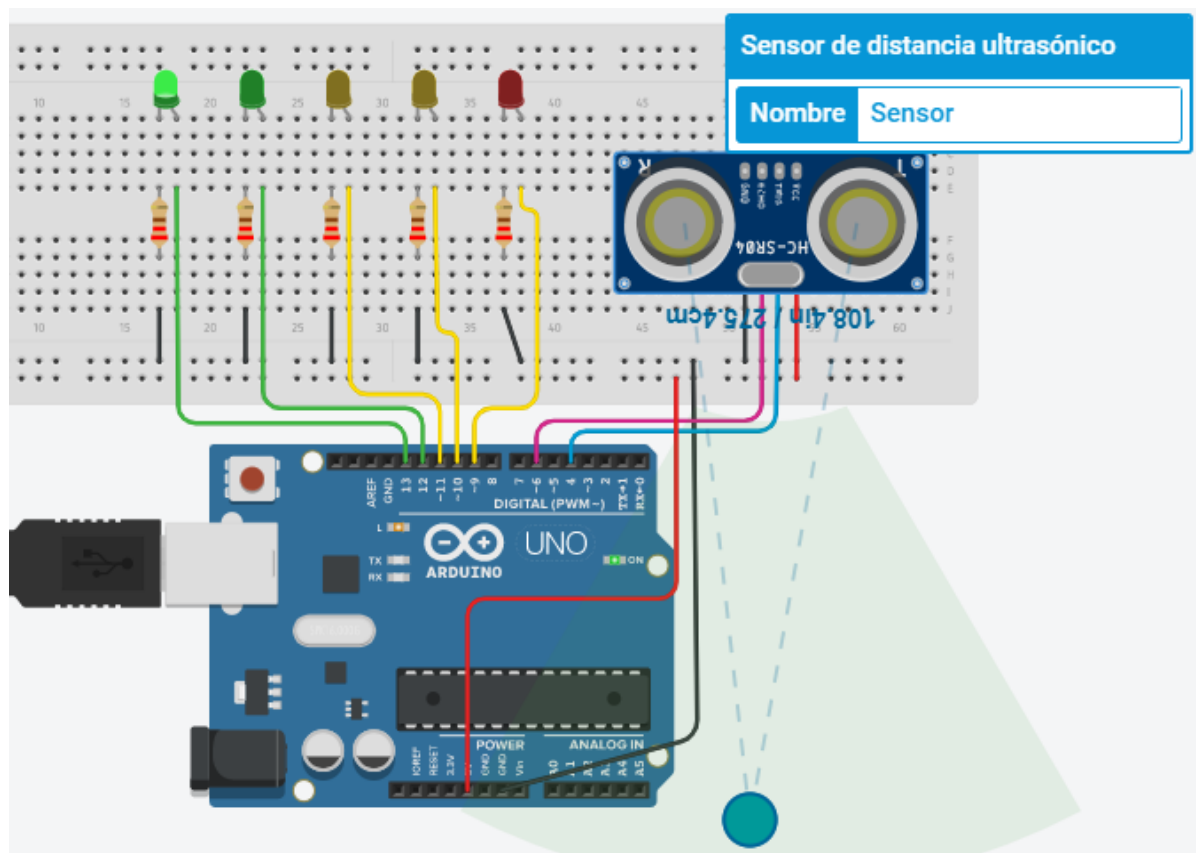
	Tabla	Datos	Índices	Modelo	Restricciones	Permisos	Estadísticas	Valores por Defecto de Interfaz de Usuario	Disparadores	Dependencias	SQL	REST	Consulta
	Consulta	Contar Filas	Insertar Fila	Cargar Datos									
	EDITAR	CODR	CODL	FECHAON		FECHAOFF							
		25	3	15/12/21 16:48:12,000000		15/12/21 16:48:50,000000							
		26	5	10/06/21 16:48:12,000000		10/06/21 16:48:50,000000							
		22	1	04/12/21 20:27:45,000000		04/12/21 20:27:57,000000							
		23	4	04/12/21 20:28:05,000000		04/12/21 20:28:55,000000							
		3	2	15/12/21 12:25:07,000000		15/12/21 12:25:59,000000							



# Simulación Tinkercad




Como parte de la solución, se realizó un código en el que se encienden cinco leds según la distancia que detecta un sensor, simulando cinco luces de un pasillo. Fue hecho en la herramienta Tinkercad, en un circuito de Arduino UNO dado la falta de un nodeESP en dicha plataforma. A continuación, se muestran imágenes de dicho circuito, así como un serial donde se imprime información relevante, junto con el código que permitió su funcionamiento.

Circuito de 5 LEDS desarrollado en Tinkercad




Serial donde se imprime cuando un LED se enciende y apaga, así como el tiempo total que estuvo encendido por razones de ahorro de energía como es presentado en la solución

Texto



1 (Arduino Uno R3)

 Monitor en serie

Led 1 encendido  
Led 1 apagado, estuvo encendido 1.27 segundos  
Led 2 encendido  
Led 2 apagado, estuvo encendido 0.90 segundos  
Led 3 encendido  
Led 3 apagado, estuvo encendido 0.44 segundos  
Led 4 encendido  
Led 4 apagado, estuvo encendido 2.31 segundos  
Led 5 encendido  
Led 3 encendido  
Led 5 apagado, estuvo encendido 0.53 segundos  
Led 1 encendido  
Led 3 apagado, estuvo encendido 0.12 segundos  
Led 1 encendido

Por último, se presenta el código utilizado para llevar a cabo esta simulación.

```
1 //Declaración de pines
2 int pingPin = 6; // pin connected to Echo Pin in the ultrasonic distance sensor
3 int trigPin = 4; // pin connected to trig Pin in the ultrasonic distance sensor
4 int led1 = 13;
5 int led2 = 12;
6 int led3 = 11;
7 int led4 = 10;
8 int led5 = 9;
9 //Variable para determinar el cambio de encendido/apagado
10 bool led1c = true;
11 bool led2c = true;
12 bool led3c = true;
13 bool led4c = true;
14 bool led5c = true;
15 //Variables para calcular los tiempos de encendido
16 unsigned long l1i = 0;
17 unsigned long l1f = 0;
18 unsigned long l2i = 0;
19 unsigned long l2f = 0;
20 unsigned long l3i = 0;
21 unsigned long l3f = 0;
22 unsigned long l4i = 0;
23 unsigned long l4f = 0;
24 unsigned long l5i = 0;
25 unsigned long l5f = 0;
26 //Variables del tiempo de encendido de un led
27 float t1 = 0;
28 float t2 = 0;
```

```

29 float t3 = 0;
30 float t4 = 0;
31 float t5 = 0;
32
33 //Declaración del serial y uso de pines
34 void setup()
35 {
36     Serial.begin(3600);
37     pinMode(pingPin, INPUT); //Set the connection pin output mode Echo pin
38     pinMode(trigPin, OUTPUT); //Set the connection pin output mode trog pin
39     pinMode(led1, OUTPUT);
40     pinMode(led2, OUTPUT);
41     pinMode(led3, OUTPUT);
42     pinMode(led4, OUTPUT);
43     pinMode(led5, OUTPUT);
44     delay(1000); //delay 1000ms
45 }
46
47 /*Función loop en la que se corre el Arduino donde se
48 verificarán las distancias y determinará si un le debe estar
49 apagado o encendido*/
50 void loop()
51 {
52     int cm = ping(pingPin);
53     //Serial.println(String(cm) + " cm.");
54     //Distancia en la que prenderá cada led
55     int Limite1 = 300;
56
57     int Limite2 = 240;
58     int Limite3 = 180;
59     int Limite4 = 120;
60     int Limite5 = 60;
61
62     //LED1
63     if ( cm > Limite2 && cm <= Limite1){
64         //Se enciende el led y se imprime en el serial
65         digitalWrite ( led1 , HIGH) ;
66         if(led1c == true){
67             led1c = false;
68             l1i = millis();
69             Serial.print("Led 1 encendido \n");
70         }
71     }
72     else{
73         //Se apaga el led y se imprime el tiempo encendido en el serial
74         digitalWrite( led1 , LOW) ;
75         if(led1c == false){
76             led1c = true;
77             l1f = millis();
78             t1 = (l1f-l1i)/1000.0f;
79             Serial.print("Led 1 apagado, estuvo encendido ");
80             Serial.print(t1);
81             Serial.print(" segundos \n");
82         }
83     }

```

```

83 //LED2
84 if ( cm > Limite3 && cm <= Limite2){
85     //Se enciende el led y se imprime en el serial
86     digitalWrite ( led2 , HIGH);
87     if(led2c == true){
88         led2c = false;
89         l2i = millis();
90         Serial.print("Led 2 encendido \n");
91     }
92 }
93 else{
94     //Se apaga el led y se imprime el tiempo encendido en el serial
95     digitalWrite( led2 , LOW);
96     if(led2c == false){
97         led2c = true;
98         l2f = millis();
99         t2 = (l2f-l2i)/1000.0f;
100     Serial.print("Led 2 apagado, estuvo encendido ");
101     Serial.print(t2);
102     Serial.print(" segundos \n");
103 }
104 }
105 //LED3
106 if ( cm > Limite4 && cm <= Limite3){
107     //Se enciende el led y se imprime en el serial
108     digitalWrite ( led3 , HIGH) ;
109     if(led3c == true){
110         led3c = false;
111         l3i = millis();
112         Serial.print("Led 3 encendido \n");
113     }
114 }
115 else{
116     //Se apaga el led y se imprime el tiempo encendido en el serial
117     digitalWrite( led3 , LOW) ;
118     if(led3c == false){
119         led3c = true;
120         l3f = millis();
121         t3 = (l3f-l3i)/1000.0f;
122     Serial.print("Led 3 apagado, estuvo encendido ");
123     Serial.print(t3);
124     Serial.print(" segundos \n");
125 }
126 }
127 //LED4
128 if ( cm > Limite5 && cm <= Limite4){
129     //Se enciende el led y se imprime en el serial
130     digitalWrite ( led4 , HIGH) ;
131     if(led4c == true){
132         led4c = false;
133         l4i = millis();
134         Serial.print("Led 4 encendido \n");
135     }
136 }
137 else{
138     //Se apaga el led y se imprime el tiempo encendido en el serial

```



```

139     digitalWrite( led4 , LOW) ;
140     if(led4c == false){
141         led4c = true;
142         l4f = millis();
143         t4 = (l4f-l4i)/1000.0f;
144         Serial.print("Led 4 apagado, estuvo encendido ");
145         Serial.print(t4);
146         Serial.print(" segundos \n");
147     }
148 }
149 //LED5
150 if ( cm <= Limite5){
151     //Se enciende el led y se imprime en el serial
152     digitalWrite ( led5 , HIGH) ;
153     if(led5c == true){
154         led5c = false;
155         l5i = millis();
156         Serial.print("Led 5 encendido \n");
157     }
158 }
159 else {
160     //Se apaga el led y se imprime el tiempo encendido en el serial
161     digitalWrite( led5 , LOW) ;
162     if(led5c == false){
163         led5c = true;
164         l5f = millis();
165         t5 = (l5f-l5i)/1000.0f;
166         Serial.print("Led 5 apagado, estuvo encendido ");
167
168         Serial.print(t5);
169         Serial.print(" segundos \n");
170     }
171     delay(100);
172 }
173
174 //Función para calcular distancia del sensor ultrasónico
175 int ping(int pingPin)
176 {
177     // establish variables for duration of the ping,
178     // and the distance result in inches and centimeters:
179     long duration, cm;
180     // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
181     // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
182     pinMode(trigPin, OUTPUT);
183     digitalWrite(trigPin, LOW);
184     delayMicroseconds(2);
185     digitalWrite(trigPin, HIGH);
186     delayMicroseconds(5);
187     digitalWrite(trigPin, LOW);
188
189     pinMode(pingPin, INPUT);
190     duration = pulseIn(pingPin, HIGH);
191
192     // convert the time into a distance
193     cm = microsecondsToCentimeters(duration);
194     return cm ;

```

```
195 }  
196  
197 long microsecondsToCentimeters(long microseconds)  
198 {  
199     // The speed of sound is 340 m/s or 29 microseconds per centimeter.  
200     // The ping travels out and back, so to find the distance of the  
201     // object we take half of the distance travelled.  
202     return microseconds / 29 / 2;  
203 }
```

---