



**Reto IoT - “Ventinteligente” - Entregable 4**

**Equipo 5**

**Juan Daniel Rodríguez Oropeza A01411625**

**Jorge Claudio González Becerril A01412375**

**Luis Angel Ramiro Amaro A01411763**

**Luis Carlos Olave Vera A01233810**

**Monterrey, Nuevo León, México**

**Diciembre 03, 2021**

**Dra. Lorena Guadalupe Gómez Martínez**

**Ing. David Alonso Cantú Delgado**

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

**Implementación de Internet de las Cosas**

Todos participaron de manera activa constante durante el desarrollo del proyecto. En este documento también se adjuntan evidencias de los avances en cada sección del proyecto.

## **Product BackLog**

### **Software**

- Modelo ER - Juan Daniel y Jorge Becerril
- Modelo Relacional - Juan Daniel
- Queries de Creación de Tablas e Inserción de Datos - Juan Daniel, Jorge Becerril, Luis Ramiro, y Luis Olave
- Importar los datos de los sensores de CSV a MySQL - Jorge Becerril
- Queries de Consulta - Juan Daniel, Jorge Becerril, y Luis Ramiro
- Limpieza de Datos del Sensor de Temperatura - Juan Daniel
- Desarrollar la Web API - Luis Olave

### **Hardware**

- Realizar el circuito y códigos para las mediciones con los sensores de temperatura, corriente y voltaje junto con la activación de los LEDs que representan las velocidades del ventilador. - Juan Daniel
- Adaptación del Circuito en Proteus al Arduino en físico - Luis Olave

También hay videos al final del documento que muestran evidencia del funcionamiento del proyecto.

El CSV que usamos también viene anexado en la entrega en Canvas.

## **Funcionalidad**

Se tendrá un sensor para temperatura y otro para la cantidad de energía consumida (voltaje, potencia, corriente) por el abanico, éstos almacenarán los datos cada hora.

De acuerdo a la temperatura, se encenderá un ventilador a determinada velocidad:

- Apagado: Temperatura < 21 °C
- Velocidad 1: 21 °C ≤ Temperatura < 25 °C
- Velocidad 2: 25 °C ≤ Temperatura < 30 °C
- Velocidad 3: Temperatura > 30 °C

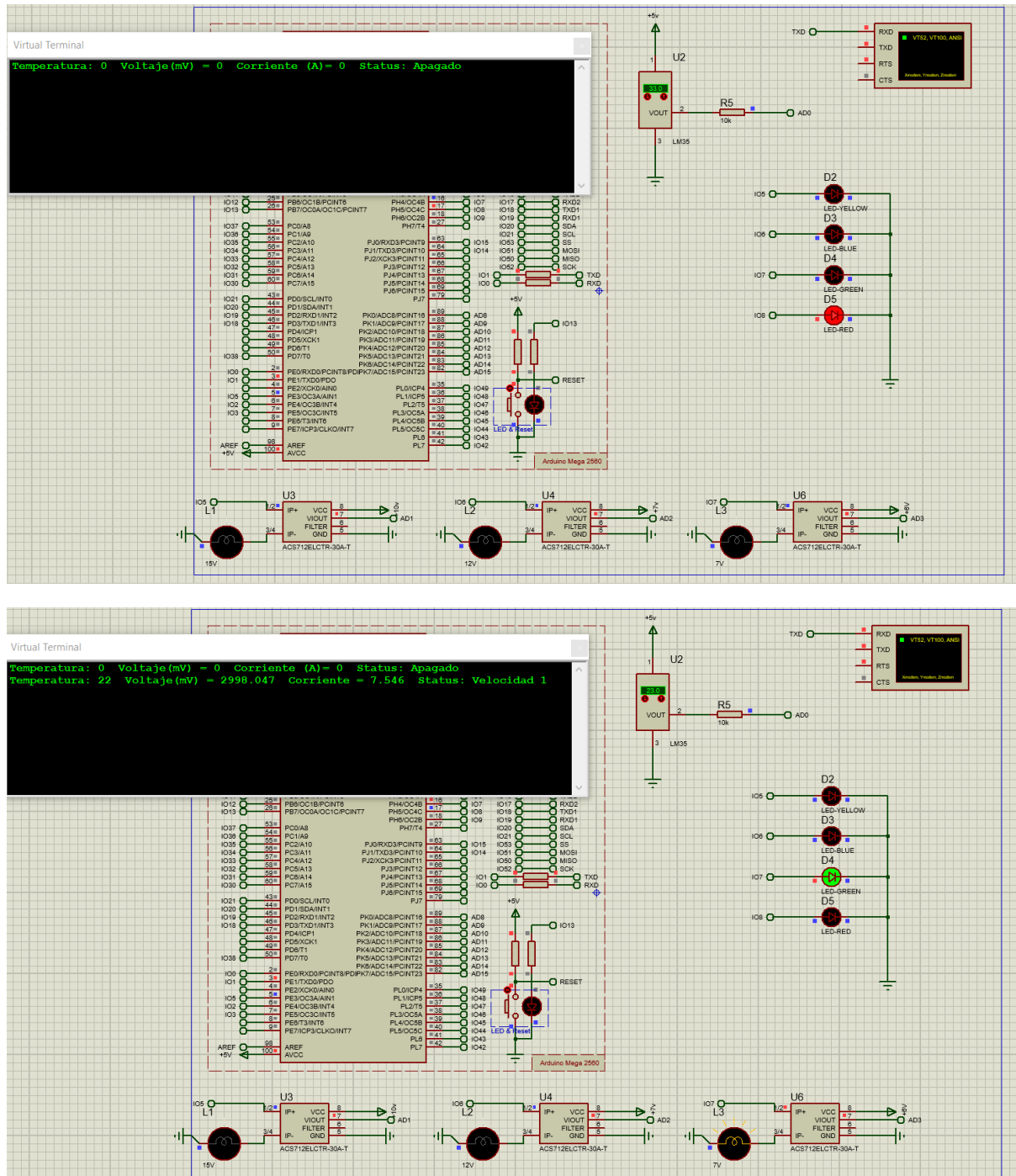
## HARDWARE

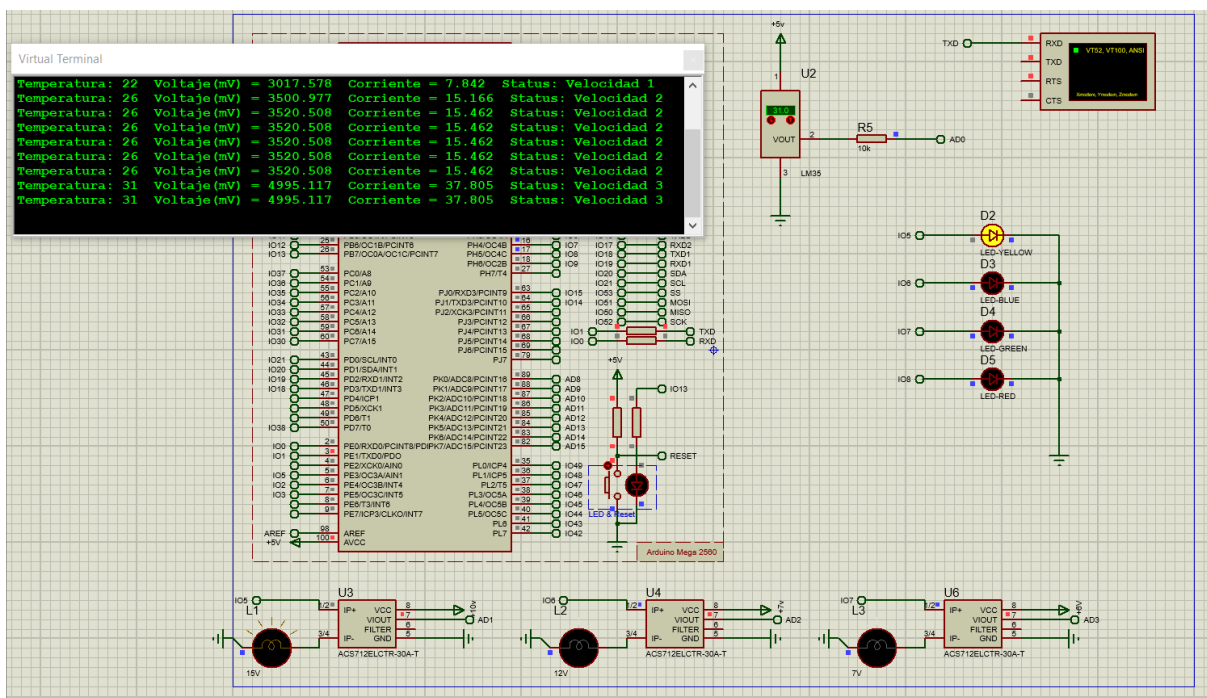
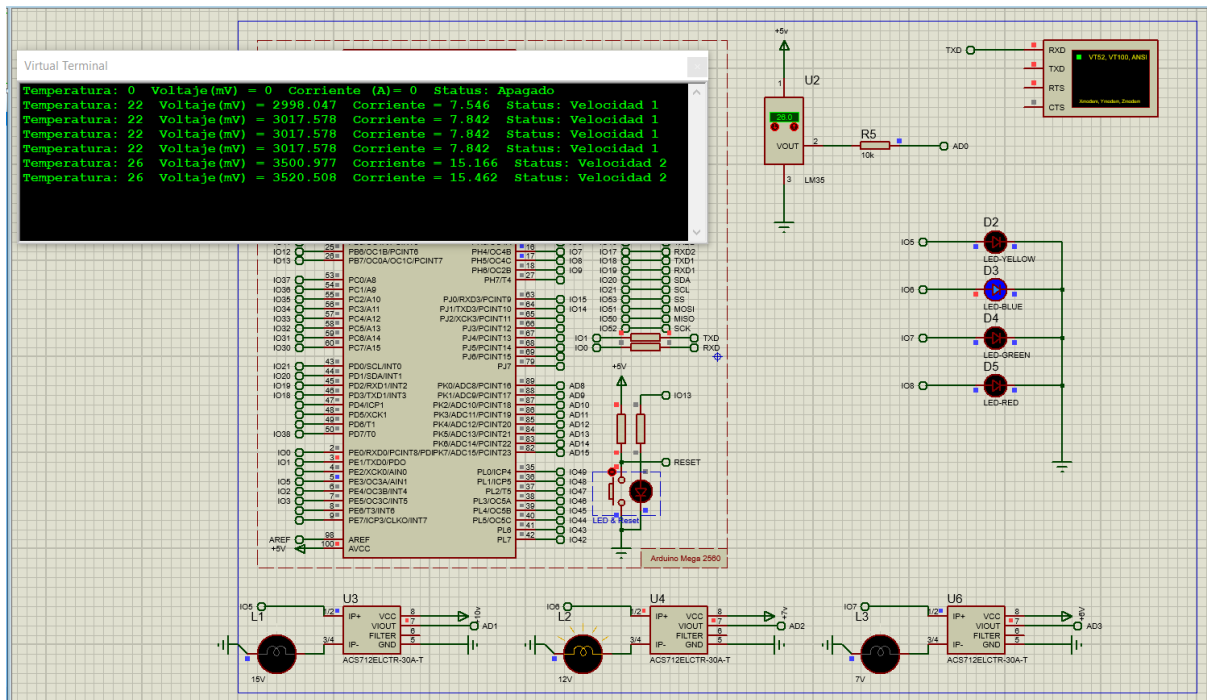
Video donde se describe el funcionamiento de tu aplicación de IoT:

<https://drive.google.com/file/d/1-2arz4rvnipezfYaStMSncvz9DrdEqx/view?usp=sharing>

El código viene en el zip.

Capturas del circuito y código de los sensores en Proteus:






```
main.ino
1  /* Main.ino file generated by New Project wizard
2  *
3  * Created: Thu Sep 5 2013
4  * Processor: ATmega2560
5  * Compiler: Arduino AVR
6  */
7
8  /* Equipo 5
9  Juan Daniel Rodríguez Oropeza A01411625
10 Jorge Claudio González Becerril A01412375
11 Luis Angel Ramiro Amaro A01411763
12 Luis Carlos Olave Vera A01233810
13 */
14
15 // Pines A del Arduino
16 int analogPin = A0;
17 int currentPin = A1;
18 int currentPin2 = A2;
19 int currentPin3 = A3;
20
21 // Pines de los LED
22 int pinVel3 = 5; // Velocidad 3
23 int pinVel2 = 6; // Velocidad 2
24 int pinVel1 = 7; // Velocidad 1
25 int pinApag = 8; // Apagado
26
27 // Variables para la lectura de Voltaje y Corriente
28 int sensitivity = 66;
29 int adcValue = 0;
30 int adcValue2 = 0;
31 int adcValue3 = 0;
32 int offsetVoltage = 2500;
33 double adcVoltage = 0;
34 double adcVoltage2 = 0;
35 double adcVoltage3 = 0;
36 double currentValue = 0;
37 double currentValue2 = 0;
38 double currentValue3 = 0;
39
40 void setup()
41 { // put your setup code here, to run once:
42   Serial.begin(9600);
43   pinMode(pinVel3, OUTPUT);
```

```
main.ino x
41 { // put your setup code here, to run once:
42   Serial.begin(9600);
43   pinMode(pinVel3, OUTPUT);
44   pinMode(pinVel2, OUTPUT);
45   pinMode(pinVel1, OUTPUT);
46   pinMode(pinApag, OUTPUT);
47 }
48
49 void loop()
50 { // put your main code here, to run repeatedly:
51   // Lectura de Temperatura
52   int a=analogRead(analogPin);
53   int temp= a*4.88/10;
54   Serial.print("Temperatura: ");
55   Serial.print(temp);
56
57   // Lectura de Voltaje y Corriente para Velocidad 3
58   adcValue = analogRead(currentPin);
59   adcVoltage = (adcValue / 1024.0) * 5000;
60   currentValue = ((adcVoltage - offsetVoltage) / sensitivity);
61
62   // Lectura de Voltaje y Corriente para Velocidad 2
63   adcValue2 = analogRead(currentPin2);
64   adcVoltage2 = (adcValue2 / 1024.0) * 5000;
65   currentValue2 = ((adcVoltage2 - offsetVoltage) / sensitivity);
66
67   // Lectura de Voltaje y Corriente para Velocidad 1
68   adcValue3 = analogRead(currentPin3);
69   adcVoltage3 = (adcValue3 / 1024.0) * 5000;
70   currentValue3 = ((adcVoltage3 - offsetVoltage) / sensitivity);
71
72   // Velocidad 3
73   if (temp >= 30){
74     digitalWrite(pinVel3, HIGH); // Velocidad 3
75     digitalWrite(pinVel2, LOW); // Velocidad 2
76     digitalWrite(pinVel1, LOW); // Velocidad 1
77     digitalWrite(pinApag, LOW); // Apagado
78
79     Serial.print(" Voltaje(mV) = ");
80     Serial.print(adcVoltage, 3);
81
```

```
main.ino x
76     digitalWrite(pinVel1, LOW); // Velocidad 1
77     digitalWrite(pinApag, LOW); // Apagado
78
79     Serial.print(" Voltaje(mV) = ");
80     Serial.print(adcVoltage, 3);
81
82     Serial.print(" Corriente = ");
83     Serial.print(currentValue, 3);
84
85     Serial.println(" Status: Velocidad 3");
86 }
87 // Velocidad 2
88 else if (temp >= 25 && temp < 30){
89     digitalWrite(pinVel3, LOW); // Velocidad 3
90     digitalWrite(pinVel2, HIGH); // Velocidad 2
91     digitalWrite(pinVel1, LOW); // Velocidad 1
92     digitalWrite(pinApag, LOW); // Apagado
93
94     Serial.print(" Voltaje(mV) = ");
95     Serial.print(adcVoltage2, 3);
96
97     Serial.print(" Corriente = ");
98     Serial.print(currentValue2, 3);
99
100    Serial.println(" Status: Velocidad 2");
101 }
102 // Velocidad 1
103 else if (temp >= 21 && temp < 25){
104     digitalWrite(pinVel3, LOW); // Velocidad 3
105     digitalWrite(pinVel2, LOW); // Velocidad 2
106     digitalWrite(pinVel1, HIGH); // Velocidad 1
107     digitalWrite(pinApag, LOW); // Apagado
108
109     Serial.print(" Voltaje(mV) = ");
110     Serial.print(adcVoltage3, 3);
111
112     Serial.print(" Corriente = ");
113     Serial.print(currentValue3, 3);
114
115     Serial.println(" Status: Velocidad 1");
116 }
```

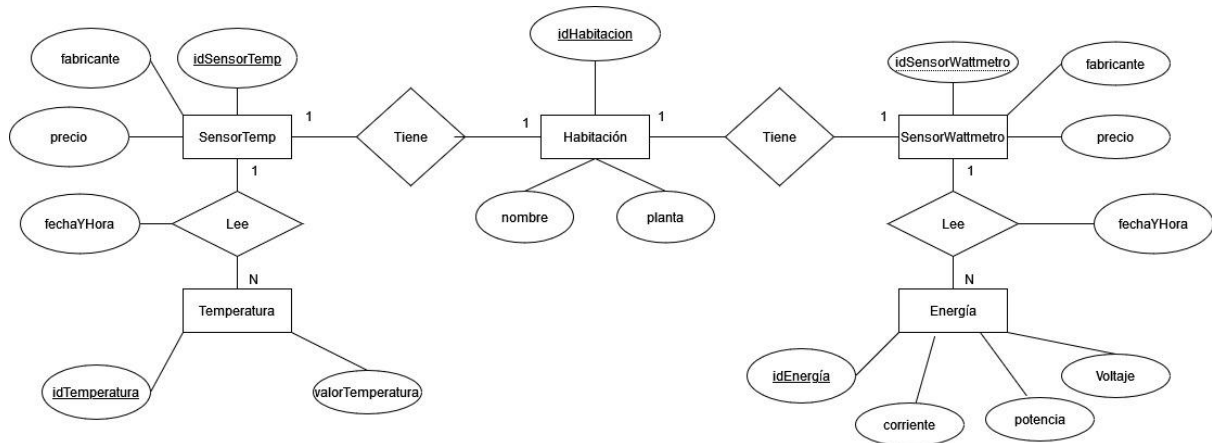
```
main.ino 
92     digitalWrite(pinApag, LOW); // Apagado
93
94     Serial.print(" Voltaje(mV) = ");
95     Serial.print(adcVoltage2, 3);
96
97     Serial.print(" Corriente = ");
98     Serial.print(currentValue2, 3);
99
100    Serial.println(" Status: Velocidad 2");
101 }
102 // Velocidad 1
103 else if (temp >= 21 && temp < 25){
104     digitalWrite(pinVel3, LOW); // Velocidad 3
105     digitalWrite(pinVel2, LOW); // Velocidad 2
106     digitalWrite(pinVel1, HIGH); // Velocidad 1
107     digitalWrite(pinApag, LOW); // Apagado
108
109     Serial.print(" Voltaje(mV) = ");
110     Serial.print(adcVoltage3, 3);
111
112     Serial.print(" Corriente = ");
113     Serial.print(currentValue3, 3);
114
115     Serial.println(" Status: Velocidad 1");
116 }
117 // Apagado
118 else{
119     digitalWrite(pinVel3, LOW); // Velocidad 3
120     digitalWrite(pinVel2, LOW); // Velocidad 2
121     digitalWrite(pinVel1, LOW); // Velocidad 1
122     digitalWrite(pinApag, HIGH); // Apagado
123
124     Serial.print(" Voltaje(mV) = 0");
125     Serial.print(" Corriente (A)= 0");
126     Serial.println(" Status: Apagado");
127 }
128
129 delay(2000);
130 }
131
```

El archivo y código de Proteus viene incluido en el .zip.



## SOFTWARE

### Modelo Entidad Relación



### Modelo Relacional

Temperatura (idTemperatura, valorTemperatura, fechaYHora, idSensorTemp\_fk)

PK (idTemperatura)

FK (idSensorTemp\_fk) references SensorTemp (idSensorTemp)

Degree(Temperatura) = 4      Number of FKs = 1

SensorTemp (idSensorTemp, fabricante, precio, idHabitacion\_fk)

PK (idSensorTemp)

FK (idHabitacion\_fk) references Habitacion (idHabitacion)

Degree(SensorTemp) = 4      Number of FKs = 1

Habitacion (idHabitacion, nombre, planta)

PK (idHabitacion)

Degree(Habitacion) = 3      Number of FKs = 0

SensorWattmetro (idSensorWattmetro, fabricante, precio, idHabitacion\_fk)

PK (idSensorWattmetro)

FK (idHabitacion\_fk) references Habitacion (idHabitacion)

Degree(SensorWattmetro) = 2      Number of FKs = 1

Energía (idEnergía, corriente, voltaje, potencia, fechaYHora, idSensorWattmetro\_fk)

PK (idEnergía)

FK (idSensorWattmetro\_fk) references SensorWattmetro (idSensorWattmetro)

Degree(Energía) = 6          Number of FKs = 1

**Queries de Creación de Tablas en Base de Datos (también se encuentra en el zip como txtx).**

BEGIN;

CREATE DATABASE Equipo5\_IOT;

```
CREATE TABLE Habitacion (  
    id_habitacion bigint(20) NOT NULL AUTO_INCREMENT,  
    nombre VARCHAR(15) NULL,  
    planta VARCHAR(15) NULL,  
    PRIMARY KEY (id_habitacion)  
);
```

```
CREATE TABLE SensorTemp (  
    id_sensor_temp bigint(20) NOT NULL AUTO_INCREMENT,  
    fabricante VARCHAR(30) NULL,  
    precio float NULL,  
    id_habitacion_fk bigint(20) NULL,  
    PRIMARY KEY (id_sensor_temp),  
    FOREIGN KEY (id_habitacion_fk) REFERENCES Habitacion (id_habitacion)  
);
```

```
CREATE TABLE SensorWattmetro (  
    id_sensor_wattmetro bigint(20) NOT NULL AUTO_INCREMENT,  
    fabricante VARCHAR(30) NULL,  
    precio float NULL,  
    id_habitacion_fk bigint(20) NULL,  
    PRIMARY KEY (id_Sensor_Wattmetro),  
    FOREIGN KEY (id_habitacion_fk) REFERENCES Habitacion (id_habitacion)  
);
```

```
CREATE TABLE Temperatura (  
    id_temperatura bigint(20) NOT NULL AUTO_INCREMENT,  
    valor_temperatura float NULL,  
    fecha_y_hora timestamp NOT NULL DEFAULT current_timestamp,  
    id_sensor bigint(20) NULL,  
    PRIMARY KEY (id_temperatura),  
    FOREIGN KEY (id_sensor) REFERENCES SensorTemp (id_sensor_temp)  
);
```

```
CREATE TABLE Energia (  
    id_energia bigint(20) NOT NULL AUTO_INCREMENT,  
    corriente float NULL,  
    voltaje float NULL,  
    potencia float NULL,
```

```

fecha_y_hora timestamp NOT NULL DEFAULT current_timestamp,
id_sensor bigint(20) NULL,
    PRIMARY KEY (id_energia),
    FOREIGN KEY (id_sensor) REFERENCES SensorWattmetro (id_sensor_wattmetro)
);

```

INSERT INTO Habitacion VALUES

```

(1, "Cocina", "Baja"),
(2, "Recamara", "Alta"),
(3, "Recibidor", "Baja"),
(4, "Sala", "Baja"),
(5, "Estudio", "Alta");

```

INSERT INTO SensorTemp VALUES

```

(1, "LG", 1000.0, 1),
(2, "Samsung", 1200.0, 2),
(3, "Sony", 900.0, 3),
(4, "Sharp", 950.0, 4),
(5, "HP", 860.0, 5);

```

INSERT INTO SensorWattmetro VALUES

```

(1, "LG", 1000.0, 1),
(2, "Samsung", 1200.0, 2),
(3, "Sony", 900.0, 3),
(4, "Sharp", 950.0, 4),
(5, "HP", 860.0, 5);

```

INSERT INTO Energia VALUES

```

(1, 0.3, 100, 0.05, "2021-11-25 12:05:02", 1),
(2, 0.4, 110, 0.06, "2021-11-25 13:05:43", 2),
(3, 0.5, 120, 0.07, "2021-11-25 14:05:34", 3),
(4, 0.3, 100, 0.05, "2021-11-25 16:06:02", 1),
(5, 0.5, 110, 0.06, "2021-11-25 18:25:43", 1),
(6, 0.6, 130, 0.08, "2021-11-28 15:24:23", 2),
(7, 0.3, 100, 0.05, "2021-11-28 18:22:05", 2),
(8, 0.8, 150, 0.09, "2021-11-28 19:14:58", 1),
(9, 0.2, 90, 0.05, "2021-11-29 04:25:35", 3),
(10, 0.55, 120, 0.07, "2021-11-29 07:24:12", 2),
(11, 0.36, 106, 0.054, "2021-11-29 10:09:48", 1),
(12, 0.39, 110, 0.06, "2021-11-29 11:29:30", 2),
(13, 0.6, 130, 0.075, "2021-11-29 01:56:38", 1),
(14, 0.54, 127, 0.068, "2021-11-29 00:29:30", 2),
(15, 0.2, 96, 0.048, "2021-11-29 05:19:53", 3),
(16, 0.6, 129, 0.072, "2021-11-29 09:13:09", 3),
(17, 0.5, 136, 0.61, "2021-11-29 15:58:00", 2),
(18, 0.9, 149, 0.096, "2021-11-29 09:43:25", 3),
(19, 0.67, 132, 0.08, "2021-11-29 11:46:20", 1),

```

```
(20, 0.45, 113, 0.057, "2021-11-29 19:36:45", 3),
(21, 0.2, 90, 0.04, "2021-11-29 09:30:01", 1);
```

```
COMMIT;
```

### Importar CSV (también se encuentra en el zip como txt).

```
-- Borra la fila con los nombres de las columnas.
```

```
DELETE FROM temperatura_csv
WHERE id_temperatura = "id_temperatura";
```

```
INSERT INTO temperatura
SELECT
CONVERT (id_temperatura, unsigned int) AS id_temperatura,
CONVERT (valor_temperatura, float) AS valor_temperatura,
STR_TO_DATE(fecha_y_hora, "%Y-%m-%d %T") AS fecha_y_hora,
CONVERT (id_sensor, unsigned int) AS id_sensor
FROM temperatura_csv;
```

### Limpiar datos del Sensor de Temperatura (también se encuentra en el zip como txt).

```
UPDATE temperatura
SET valor_temperatura=NULL
WHERE valor_temperatura=0
```

### Base de datos

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño
<input type="checkbox"/> energía	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	21	InnoDB	utf8mb4_general_ci	32.0 KB
<input type="checkbox"/> habitacion	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	16.0 KB
<input type="checkbox"/> sensortemp	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	32.0 KB
<input type="checkbox"/> sensorwattmetro	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	32.0 KB
<input type="checkbox"/> temperatura	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	654	InnoDB	utf8mb4_general_ci	80.0 KB
<input type="checkbox"/> temperatura_csv	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	654	InnoDB	utf8_general_ci	64.0 KB
6 tablas	Número de filas	1,344	InnoDB	utf8mb4_general_ci	256.0 KB

### Queries de Consulta (también se encuentran el zip como txt).

#### Obtener la temperatura (°C) máxima con fecha y hora, y nombre de habitación.

```
SELECT t.fecha_y_hora, h.nombre, t.valor_temperatura AS max_temperatura
FROM temperatura AS t
JOIN sensortemp AS st ON st.id_sensor_temp = t.id_sensor
JOIN habitacion AS h ON h.id_habitacion = st.id_habitacion_fk
WHERE t.valor_temperatura = (
    SELECT MAX(t2.valor_temperatura)
    FROM temperatura AS t2
);
```

fecha_y_hora	nombre	max_temperatura
2021-11-29 21:34:55	Cocina	25.4

**Obtener la temperatura (°C) media.**

```
SELECT AVG(valor_temperatura)
FROM Temperatura
AS temperatura_promedio;
```

AVG(valor_temperatura)
22.46552852847712

**Obtener la temperatura (°C) más baja con fecha y hora, y nombre de habitación.**

```
SELECT t.fecha_y_hora, h.nombre, t.valor_temperatura AS min_temperatura
FROM temperatura AS t
JOIN sensortemp AS st ON st.id_sensor_temp = t.id_sensor
JOIN habitacion AS h ON h.id_habitacion = st.id_habitacion_fk
WHERE t.valor_temperatura = (
    SELECT MIN(t2.valor_temperatura)
    FROM temperatura AS t2
);
```

fecha_y_hora	nombre	min_temperatura
2021-11-25 17:10:15	Cocina	22
2021-11-25 17:10:20	Cocina	22
2021-11-25 17:10:25	Cocina	22
2021-11-25 17:10:30	Cocina	22
2021-11-25 17:10:35	Cocina	22
2021-11-25 17:10:40	Cocina	22
2021-11-25 17:10:45	Cocina	22
2021-11-25 17:10:50	Cocina	22
2021-11-25 17:10:55	Cocina	22
2021-11-25 17:11:00	Cocina	22
2021-11-25 17:11:05	Cocina	22
2021-11-25 17:11:10	Cocina	22
2021-11-25 17:11:15	Cocina	22
2021-11-25 17:11:20	Cocina	22

**Obtener el voltaje (V) máximo con fecha y hora, y nombre de habitación.**

```
SELECT e.fecha_y_hora, h.nombre, e.voltaje AS max_voltaje
FROM energia AS e
JOIN sensorwattmetro AS sw ON sw.id_sensor_wattmetro = e.id_sensor
JOIN habitacion AS h ON h.id_habitacion = sw.id_habitacion_fk
WHERE e.voltaje = (
    SELECT MAX(e2.voltaje)
    FROM energia AS e2
);
```

fecha_y_hora	nombre	max_voltaje
2021-11-28 19:14:58	Cocina	150

#### Obtener el voltaje (V) promedio.

```
SELECT AVG(voltaje)
FROM Energia
AS voltaje_promedio;
```

**AVG(voltaje)**

116.57142857142857

#### Obtener la corriente (A) máxima con fecha y hora, y nombre de habitación.

```
SELECT e.fecha_y_hora, h.nombre, e.corriente AS max_corriente
FROM energia AS e
JOIN sensorwattmetro AS sw ON sw.id_sensor_wattmetro = e.id_sensor
JOIN habitacion AS h ON h.id_habitacion = sw.id_habitacion_fk
WHERE e.corriente = (
    SELECT MAX(e2.corriente)
    FROM energia AS e2
);
```

fecha_y_hora	nombre	max_corriente
2021-11-29 09:43:25	Recibidor	0.9

#### Obtener la corriente (A) promedio.

```
SELECT AVG(corriente)
FROM Energia
AS corriente_promedio;
```

**AVG(corriente)**

0.4695238165912174

#### Obtener la potencia (kW) máxima con fecha y hora, y nombre de habitación.

```
SELECT e.fecha_y_hora, h.nombre, e.potencia AS max_potencia
FROM energia AS e
JOIN sensorwattmetro AS sw ON sw.id_sensor_wattmetro = e.id_sensor
JOIN habitacion AS h ON h.id_habitacion = sw.id_habitacion_fk
WHERE e.potencia = (
    SELECT MAX(e2.potencia)
    FROM energia AS e2
);
```

fecha_y_hora	nombre	max_potencia
2021-11-29 15:58:00	Recamara	0.61

**Obtener la potencia (kW) promedio.**

```
SELECT AVG(potencia)
FROM Energia
AS potencia_promedio;
```

AVG(potencia)
0.09000000091535705

**Obtener la temperatura máxima de cada hora de cada día, su tiempo exacto y nombre de habitación.**

```
SELECT DATE(t.fecha_y_hora) AS fecha, TIME(t.fecha_y_hora) AS hora, h.nombre AS
habitacion, MAX(t.valor_temperatura) AS max_temperatura
FROM temperatura AS t
JOIN sensortemp AS st ON st.id_sensor_temp = t.id_sensor
JOIN habitacion AS h ON h.id_habitacion = st.id_habitacion_fk
GROUP BY DAY(t.fecha_y_hora), HOUR(t.fecha_y_hora);
```

fecha	hora	habitacion	max_temperatura
2021-11-25	17:10:15	Cocina	22
2021-11-25	18:39:37	Cocina	23.8
2021-11-28	13:37:26	Cocina	NULL
2021-11-29	21:31:56	Cocina	25.4

**Obtener la cantidad de temperaturas registradas por día.**

```
SELECT DATE(t.fecha_y_hora) AS fecha, COUNT(*) AS temperaturas_registradas
FROM temperatura AS t
GROUP BY DATE(t.fecha_y_hora);
```

fecha	temperaturas_registradas
2021-11-25	17
2021-11-28	39
2021-11-29	598

**Obtener la temperatura promedio por día con su nombre de habitación.**

```
SELECT DATE(t.fecha_y_hora) AS fecha, h.nombre AS Habitacion,
AVG(t.valor_temperatura) AS promedio_temperatura
FROM temperatura AS t
JOIN sensortemp AS st ON st.id_sensor_temp = t.id_sensor
JOIN habitacion AS h ON h.id_habitacion = st.id_habitacion_fk
GROUP BY DAY(t.fecha_y_hora);
```

fecha	Habitacion	promedio_temperatura
2021-11-25	Cocina	22.14117633595186
2021-11-28	Cocina	NULL
2021-11-29	Cocina	22.474749242980344

**Obtener la cantidad de temperaturas registradas cada hora de cada día.**

```
SELECT DATE(t.fecha_y_hora) AS fecha, HOUR(t.fecha_y_hora) AS hora, COUNT(*) AS
temperaturas_registradas
FROM temperatura as t
GROUP BY DATE(t.fecha_y_hora), HOUR(t.fecha_y_hora);
```

fecha	hora	temperaturas_registradas
2021-11-25	17	14
2021-11-25	18	3
2021-11-28	13	39
2021-11-29	21	598

**Obtener la corriente, potencia y voltaje promedio por día y por habitación.**

```
SELECT DATE(e.fecha_y_hora) AS Fecha, h.nombre AS Habitacion, AVG(e.corriente) AS
Promedio_corriente,
AVG(e.potencia) AS Promedio_potencia, AVG(e.voltaje) AS Promedio_voltaje
FROM energia AS e
JOIN sensorwattmetro AS sw ON sw.id_sensor_wattmetro = e.id_sensor
JOIN habitacion AS h ON h.id_habitacion = sw.id_habitacion_fk
GROUP BY DAY(e.fecha_y_hora), Habitacion;
```

Fecha	Habitacion	Promedio_corriente	Promedio_potencia	Promedio_voltaje
2021-11-25	Cocina	0.36666667461395264	0.053333333338300387	103.33333333333333
2021-11-25	Recamara	0.4000000059604645	0.05999999865889549	110
2021-11-25	Recibidor	0.5	0.07000000029802322	120
2021-11-28	Cocina	0.800000011920929	0.09000000357627069	150
2021-11-28	Recamara	0.45000001786139343	0.06499999947845936	115
2021-11-29	Cocina	0.45750001445412636	0.062250000424683094	114.5
2021-11-29	Recamara	0.4950000047683716	0.20200000423938036	123.25
2021-11-29	Recibidor	0.4699999988079071	0.06459999978542327	115.4

**Obtener la máxima corriente, potencia y voltaje por día y habitación.**

```
SELECT DATE(e.fecha_y_hora) AS Fecha, h.nombre AS Habitacion, MAX(e.corriente) AS
Max_corriente,
MAX(e.potencia) AS Max_potencia, MAX(e.voltaje) AS Max_voltaje
FROM energia AS e
JOIN sensorwattmetro AS sw ON sw.id_sensor_wattmetro = e.id_sensor
JOIN habitacion AS h ON h.id_habitacion = sw.id_habitacion_fk
GROUP BY DAY(e.fecha_y_hora), Habitacion;
```



Fecha	Habitacion	Max_corriente	Max_potencia	Max_voltaje
2021-11-25	Cocina	0.5	0.06	110
2021-11-25	Recamara	0.4	0.06	110
2021-11-25	Recibidor	0.5	0.07	120
2021-11-28	Cocina	0.8	0.09	150
2021-11-28	Recamara	0.6	0.08	130
2021-11-29	Cocina	0.67	0.08	132
2021-11-29	Recamara	0.55	0.61	136
2021-11-29	Recibidor	0.9	0.096	149

**Obtener el total de corriente, potencia y voltaje por día y habitación.**

```
SELECT DATE(e.fecha_y_hora) AS Fecha, h.nombre AS Habitacion, SUM(e.corriente) AS
Suma_corriente,
SUM(e.potencia) AS Suma_potencia, SUM(e.voltaje) AS Suma_voltaje
FROM energia AS e
JOIN sensorwattmetro AS sw ON sw.id_sensor_wattmetro = e.id_sensor
JOIN habitacion AS h ON h.id_habitacion = sw.id_habitacion_fk
GROUP BY DAY(e.fecha_y_hora), Habitacion;
```

Fecha	Habitacion	Suma_corriente	Suma_potencia	Suma_voltaje
2021-11-25	Cocina	1.100000023841858	0.1600000001490116	310
2021-11-25	Recamara	0.4000000059604645	0.05999999865889549	110
2021-11-25	Recibidor	0.5	0.07000000029802322	120
2021-11-28	Cocina	0.800000011920929	0.090000000357627869	150
2021-11-28	Recamara	0.9000000357627869	0.12999999895691872	230
2021-11-29	Cocina	1.8300000578165054	0.249000000169873238	458
2021-11-29	Recamara	1.9800000190734863	0.8080000169575214	493
2021-11-29	Recibidor	2.3499999940395355	0.3229999989271164	577

### Videos Adicionales:

#### Creación de la Base de Datos:

<https://drive.google.com/file/d/1OMPuH2mtHnDF9pLZgdf-WgcRbSZgZqzb/view?usp=sharing>

#### Limpieza de los Datos del Sensor de Temperatura (comparación con los datos sin limpieza):

<https://drive.google.com/file/d/1RYIc6hNZSDenAspLIF66FS2T4g6xcW2v/view?usp=sharing>