

# Pozos Provenzales

Erick José André Villatoro Revolorio, Deivid Alexander Lux Revolorio,  
Diego Andrés Obín Rosales, Bryan Eduardo Gonzalo Méndez Quevedo,  
Elder Anibal Pum Rojas

*Universidad de San Carlos de Guatemala, Facultad de Ingeniería*

## I. INTRODUCCIÓN

En los últimos años, el país de Canadá decidió implementar su propio sistema de pozos, mejor conocidos como Pozos Provenzales. Dichos pozos permiten utilizar energía renovable y además proveer al interior de una casa para su uso en invierno y en provenza para el verano.

Guatemala por su parte, no cuenta con un sistema parecido, por el contrario, utilizamos pozos comunitarios manejados por EMPAGUA. Por lo que, sabiendo la necesidad que existe en estos departamentos, se decide contratar personal especializado que instale un intercambiador de calor que no es más que un sistema de tubos enterrados que conducen el aire del exterior al interior de la casa y se desarrolle un dispositivo capaz de medir y reportar las distintas magnitudes relacionadas al análisis de temperatura.

Como ya es de costumbre hoy día, los datos generados y almacenados por cualquier dispositivo deben poder ser monitoreados, visibles y de fácil comprensión para cualquier tipo de usuario, desde el más experto hasta el más inexperto, por ello se le solicita también integrar una interfaz que permita interpretar las magnitudes de una forma gráfica y animada. El diseño del dispositivo para su implementación e integración de los módulos, sensores y placa arduino queda a discreción del equipo desarrollador, pero debe poder soportar la intemperie al 100%, independientemente del clima ya que estará en el exterior.

## I. OBJETIVOS

### A. Generales

- Diseñar un dispositivo que solucione una necesidad de los tiempos actuales con respecto al uso de pozos provenzales.

### B. Específicos

- Diseñar un dispositivo que sea una alternativa que solucione la necesidad de un calefactor o aire acondicionado en el interior de la casa.
- Implementar una aplicación en Processing que permite visualizar magnitudes físicas digitalizadas para una comprensión de datos humanamente legible.
- Aprender a desarrollar una solución mediante la correcta implementación del framework de IoT.

- Sensor de humedad en tierra HL-69
- Sensores de temperatura DHT11
- Fotorresistencia
- Estaño
- Cable de Protoboard
- Ventilador
- Tubo PVC
- Maceta
- Casa de madera
- Sensor de Luz BH1750
- Codos de tubo PVC
- Bolsas de tierra

### b. Listado de Materiales digitales:

- API de ingreso de parámetros
- Componentes digitales de diseño
- Framework de React para las gráficas y diseño para ver resultados finales.
- Aplicación en Processing para los resultados

## B. Magnitudes físicas a medir

- Porcentaje de humedad (%)

$$Humedad = \frac{5}{1024} lectura$$

*Ecuación 1: Valor que sale del sensor HL-69*

- Temperatura en grados centígrados (C°)

## I. DESARROLLO DE LA PRÁCTICA

### A. Bill of Material

#### a. Listado de Materiales físicos:

- Arduino Mega 2560
- Sensor de gases MQ135

$$\text{Volt} = \frac{5}{1024} \text{ lectura}$$

*Ecuación 2: Temperatura que sale del sensor DHT11 tanto interior como exterior*

- CO2 en el ambiente

$$\text{CO2} = \frac{5}{1024} \text{ lectura}$$

*Ecuación 1: Valor que sale del sensor MQ135*

- Sensor de Luz (lumen /m<sup>2</sup>)

- Noche: 0.001-0.02
- Luz de medianoche: 0.02-0.3
- Interior nublado: 5-50
- Exterior nublado: 50-500
- Soleado Interior: 100-1000
- Bajo el sol de verano: alrededor de 10<sup>6</sup>
- Recomendado para lectura de libros: 50-60;
- Estándar de video casero: 1400.

### C. Funciones Principales:

- **Medición temperatura:**

El dispositivo será capaz de medir la temperatura en dos ambientes distintos, ambos en grados centígrados. Los ambientes serán dentro de la casa y fuera de la casa, respectivamente.

- **Cantidad de luz en el ambiente:**

El prototipo debe facilitar al usuario final la visualización de la cantidad de luz en el ambiente en una localidad determinada. Para ello, se requiere la implementación de un sensor capaz de obtener la luz que está siendo capturada en el momento.

- **Cantidad humedad en el ambiente:**

La estación deberá ser capaz de medir en porcentaje cuál es la humedad en el entorno en ese instante, el porcentaje deberá ser calculado y justificado por el desarrollador, demostrando su razonamiento.

- **Medición de calidad de aire:**

Se deberá realizar la medición de CO2 del ambiente para determinar la calidad del aire.

### D. Procedimiento

- **Sensores**

- Arduino Mega 2560

| Tamaño       | Lectura Sensor   | Instalación  | Microcontrolador | Medición voltaje |
|--------------|--|--|------------------|------------------|
| 7 x 7 x 6 mm | Pines de E/S digitales: 54 (de los cuales 14 proporcionan salidas PWM)<br>Pines de entrada analógica: 16 | Maqueta<br><br>Objetivo: áreas interiores, exteriores. | ATMEGA2560       | 7 a 12V          |

- link datasheet:

<https://www.robotshop.com/media/files/PDF/ArduinoMega2560Datasheet.pdf>



- Precio: Q195.00

- **Sensor DHT11**

| Tamaño                      | Lectura Sensor    | Instalación   | Rango de Medición  | Unidad de Medida                       |
|-----------------------------|-------------------|---|--|--|
| 3.5 * 1.4cm / 1.37 * 0.55 " | Digital analógica | Maqueta<br><br>Objetivo: áreas interiores, exteriores | temperatura: 0 a 50 °C<br>humedad: 20% a 90% RH<br>Porcentaje de humedad (%) | Temperatura en grados centígrados (C°) |

- link datasheet:

[http://image.dfrobot.com/image/data/DFR0067/DFR0067\\_DS\\_10\\_en.pdf](http://image.dfrobot.com/image/data/DFR0067/DFR0067_DS_10_en.pdf)



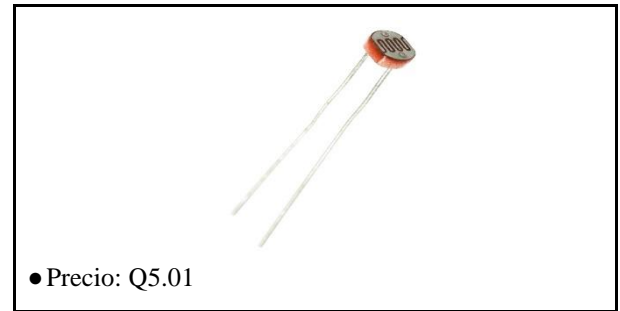
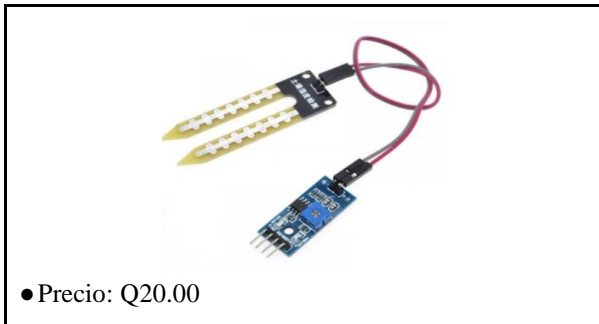
- Precio: Q25.00

- **Sensor HL-69**

| Tamaño         | Lectura Sensor    | Instalación  | Rango de Medición         | Unidad de Medida |
|----------------|-------------------|--|---------------------------|------------------|
| 60 x 20 x 5 mm | Digital analógica | Objetivo: áreas exteriores, intemperie. que permita percibir la humedad en la tierra | Porcentaje de humedad (%) | Valor absoluto   |

- link datasheet:

<https://www.ti.com/lit/ds/symlink/lm393-n.pdf>



#### ● Sensor de gases MQ135

| Tamaño          | Medición voltaje | Instalación   | Rango de Medición   | Unidad de Medida                             |
|-----------------|------------------|---|---|--|
| 32 x 22 x 24 mm | 5 voltios DC     | Anemómetro<br><br>Objetivo: áreas exteriores, intemperie que permite percibir la velocidad del viento | temperatura: -20 a 70 °C<br>humedad: 0% a 95% RH<br>Porcentaje de humedad (%)<br>Concentración oxígeno (%) 2% a 21% | Concentración del oxígeno en el ambiente (%) |

#### ● link datasheet:

<https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf>



● Precio: Q31.00

#### ● Fotorresistencia (Luz)

| Tamaño | Medición voltaje | Instalación  | Rango de Medición  | Unidad de Medida |
|--------|------------------|--|--|------------------|
| 5mm    | 150 V máximo     | Objetivo: áreas exteriores, intemperie que permite percibir la información de la luz en el ambiente. | Longitud de onda 540 nm<br>Resistencia a la oscuridad 200k Ohms<br>Resistencia a Lux 5k – 10k Ohms | Lux en Kilo Ohms |

#### ● link datasheet:

<https://www.electan.com/datasheets/cebek/CE-C2795.pdf>

#### ● Conectividad

- La transmisión de datos se realiza por medio del Arduino Mega que envía los datos a través del puerto serial a la PC, esto ocurre de forma cableada vía USB.
- Los datos recibidos en la PC son almacenados en una Base de Datos alojada en la Nube y este a su vez son procesados en la aplicación Processing a través de una imagen.
- Datos que deben de enviar desde la unidad al PC:
  - Medición de la temperatura, tanto dentro de la casa como fuera de ella, ambos en grados centígrados
  - Humedad en la tierra en porcentaje
  - Cantidad de Luz en el ambiente
  - Cantidad de CO2 en el ambiente
- Datos mostrados y evaluados serán visibles en la parte del Dashboard, que por medio de gráficas se podrá visualizar el cambio de los distintos sensores y así poder ver el funcionamiento del pozo.

#### ★ Dashboard

Con el objetivo de poder visualizar y evaluar la medición de las magnitudes en tiempo real, se le solicita al desarrollador elaborar una aplicación en con un Framework, el objetivo es representar de forma gráfica todos los datos que sean recolectados

En la Figura 3 se puede observar un vista previa de cómo se vería el funcionamiento del prototipo y los datos que serán visibles para el usuario, por lo que la conectividad del puerto serial será a través de un servidor que estará al pendiente de las peticiones en la visualización del cliente o frontend.

#### ★ Arduino

El envío de datos al servidor se realizó de esta manera:

```

void loop()
{
  //TEMPERATURA//
  DHT.read11(DHT11_PIN);
  float temp1 = DHT.temperature;
  //TEMPERATURA 2//
  DHT.read11(DHT11_PIN2);
  float temp2 = DHT.temperature;
  //CO2 //
  MQ135 gasSensor = MQ135(A0);
  float ppm = gasSensor.getPPM();
  // Humedad en la tierra //
  float humedad =
analogRead(TIERA_HUMEDAD);
  float porcentajeHum =
map(humedad, 0, 1023, 100, 0);
  // Luminocidad //
  double lux =
calcular_int_lum(analogRead(LDR_PIN
));

  DynamicJsonDocument doc(1024);
  doc["temp1"] = temp1;
  doc["temp2"] = temp2;
  doc["lumin"] = lux;
  doc["humedad"] = porcentajeHum;
  doc["co2"] = ppm;
  serializeJson(doc, Serial);
  Serial.println();
  delay(2000);
}

```

Así mismo se envía los datos a la Base de Datos de la siguiente manera:

```

import { MongoClient,
ServerApiVersion } from 'mongodb';

// Lo siguiente fue realizado con
Atlas MongoDB, una base de datos en
la nube
const uri =
"mongodb+srv://DiiAns23:andres23@cluster0.pg03o.mongodb.net/myFirstDatabase?retryWrites=true&w=majority";

// Se crea la conexion a Mongo

```

```

let mongoClient = new
MongoClient(uri, { useNewUrlParser:
true, useUnifiedTopology: true,
serverApi: ServerApiVersion.v1 });

const insertData = async (data) =>{
  try {
    if(!mongoClient){
      console.log('El cliente
de mongo no existe')
      return
    }
    mongoClient.connect((err,
db) => {
      if(err) {
        console.log('Error
al conectar')
        console.error(err);
      }

      if (!db) {
        console.log('No se
encontro la base de datos')
        return;
      }
      let dbo =
db.db('practical');
      dbo.collection('data').
insertOne( data, (err, res) => {
        if(err) {
          console.error(e
rr);
        } else {
          console.log(`Ob
jeto ${res.insertedId}
insertado.`);
        }
      })
    })
  } catch (err){
    db.close();
    console.error(err)
    console.log('error en la
bdd')
  }
}

const getLastItems = async () => {

```

```

    await MongoClient.connect();
    const db =
MongoClient.db('practical1');
    const data =
db.collection('data');
    const array = await
data.find({}).toArray();
    return array.slice(-20);
}

export { insertData, getLastItems
};

```

Este a su vez es recibido y almacenado correctamente en la Base de Datos alojada en la nube.

### ★ Entorno del Objeto

El prototipo simula una casa en la vida real, donde puede medir distintas cosas dentro de la casa y además puede medir en el exterior, como en los siguientes escenarios:

- Patio
- Parque

### ★ Tamaño del objeto

La base del prototipo está instalada sobre una maceta y esta mide un aproximado de 60 x 26 cm. Y la altura cuenta con 25 cm aproximadamente.



Figura 1: Dimensiones del prototipo

### • Bocetos del Prototipo

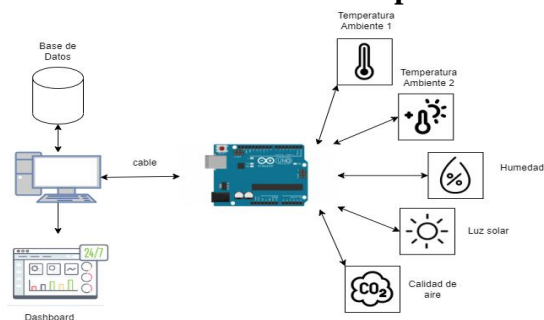


Figura 2: Boceto general del funcionamiento de todo el prototipo



Figura 3: Prototipo de la estación meteorológica

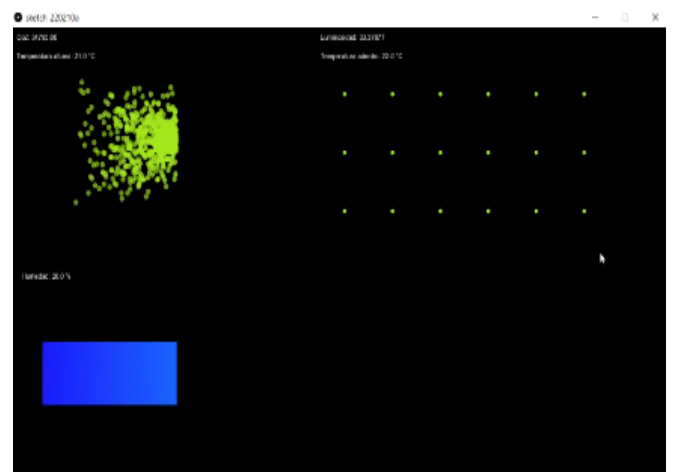


Figura 4: Flujo de datos representados en Processing

### • Analítica

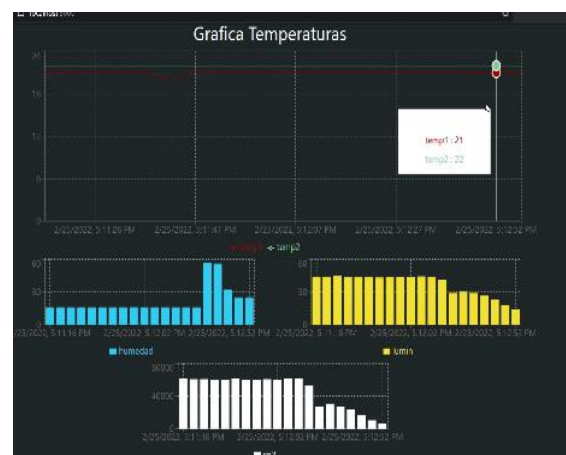


Figura 5: Gráfica obtenida desde el Dashboard, donde podemos ver Humedad y Luminosidad.

- **Base de Datos**

| Nombre  | Tipo de Dato                        |
|---------|-------------------------------------|
| _id     | autoincrementable, int, primary key |
| Temp1   | Int32                               |
| Temp2   | Int32                               |
| Lumin   | Double                              |
| Humedad | Int32                               |
| Co2     | Double                              |
| Time    | String                              |

Tabla 1: Modelo de Base de Datos para insertar valores

| Column1_id               | Column1.temp1 | Column1.temp2 | Column1.lumin | Column1.humedad | Column1.co2 | Column1.time          |
|--------------------------|---------------|---------------|---------------|-----------------|-------------|-----------------------|
| 62193f0e7c8b2544d45e90a  | 22            | 23            | 2.769205      | 8               | 266.3746    | 2/25/2022, 2:44:56 PM |
| 62193f0e7c8b2544d45e90b  | 22            | 23            | 2.195795      | 7               | 179.8013    | 2/25/2022, 2:45:01 PM |
| 62193f0e7c8b2544d45e90c  | 22            | 23            | 2.010002      | 8               | 146.5883    | 2/25/2022, 2:45:06 PM |
| 62193f0e7c8b2544d45e90d  | 22            | 23            | 1.984078      | 8               | 139.4213    | 2/25/2022, 2:45:11 PM |
| 62193f0e7c8b2544d45e90e  | 23            | 23            | 2.16879       | 8               | 169.7043    | 2/25/2022, 2:45:16 PM |
| 62193f0e7c8b2544d45e90f  | 22            | 23            | 2.141934      | 8               | 165.6786    | 2/25/2022, 2:45:21 PM |
| 62193f0e7c8b2544d45e910  | 22            | 23            | 2.561227      | 8               | 228.8429    | 2/25/2022, 2:45:26 PM |
| 62193f0e7c8b2544d45e911  | 22            | 23            | 2.010002      | 8               | 146.5883    | 2/25/2022, 2:45:31 PM |
| 62193f0e7c8b2544d45e912  | 22            | 23            | 1.385201      | 8               | 71.81744    | 2/25/2022, 2:45:36 PM |
| 62193f0e7c8b2544d45e913  | 23            | 23            | 2.36113       | 8               | 199.9277    | 2/25/2022, 2:45:41 PM |
| 62193f0e7c8b2544d45e914  | 22            | 23            | 2.985225      | 8               | 308.3232    | 2/25/2022, 2:45:46 PM |
| 62194000e7c8b2544d45e915 | 22            | 23            | 2.250279      | 8               | 182.2123    | 2/25/2022, 2:45:52 PM |
| 62194000e7c8b2544d45e916 | 22            | 23            | 2.333184      | 7               | 195.3849    | 2/25/2022, 2:45:57 PM |
| 62194000e7c8b2544d45e917 | 22            | 23            | 3.242135      | 32              | 369.3383    | 2/25/2022, 2:46:02 PM |
| 62194000e7c8b2544d45e918 | 23            | 23            | 3.176889      | 33              | 355.0542    | 2/25/2022, 2:46:07 PM |
| 62194014e7c8b2544d45e919 | 22            | 23            | 3.080299      | 33              | 334.4167    | 2/25/2022, 2:46:12 PM |
| 62194019e7c8b2544d45e91a | 22            | 23            | 2.953864      | 33              | 308.3232    | 2/25/2022, 2:46:17 PM |
| 6219401ee7c8b2544d45e91b | 22            | 23            | 3.341285      | 33              | 362.1428    | 2/25/2022, 2:46:22 PM |
| 62194023e7c8b2544d45e91c | 23            | 23            | 3.04844       | 33              | 308.3232    | 2/25/2022, 2:46:27 PM |
| 62194028e7c8b2544d45e91d | 22            | 23            | 2.417504      | 33              | 199.9277    | 2/25/2022, 2:46:32 PM |

Figura 6: Tabla con datos reales del Modelo de BD

### a. Análisis Descriptivo:

- ¿Cuál es la temperatura en este momento dentro de mi casa y fuera de mi casa?
- ¿Cuánta humedad se encuentra en el ambiente conforme a la temperatura?
- ¿Cuánta luz se encuentra en el ambiente?
- ¿Cuál es la cantidad de CO2 que hay en el ambiente?

### b. Análisis de Diagnóstico:

- ¿Cuál es la temperatura del día de hoy dentro de mi casa?
- Conforme a la humedad en el ambiente puede pronosticar que tanto incrementa o disminuye los valores del sensor.
- Ya sea de día o de noche, incluso si hay gases cerca, podemos determinar ese tipo de información con nuestro dispositivo.

- **Conocimientos aplicados**

### a. Conexión de la DB en la nube, realizada con MongoDB:

```
import { MongoClient,
ServerApiVersion } from 'mongodb';
```

```
// Lo siguiente fue realizado con
Atlas MongoDB, una base de datos en
la nube
```

```
const uri =
"mongodb+srv://DiiAns23:andres23@cluster0.pg03o.mongodb.net/myFirstDatabase?retryWrites=true&w=majority";
```

```
// Se crea la conexion a Mongo
let mongoClient = new
MongoClient(uri, { useNewUrlParser:
true, useUnifiedTopology: true,
serverApi: ServerApiVersion.v1 });
```

### b. Ingresar datos a la DB:

```
const insertData = async (data) =>{
  try {
    if(!mongoClient){
      console.log('El cliente
de mongo no existe')
      return
    }
    mongoClient.connect((err,
db) => {
      if(err) {
        console.log('Error
al conectar')
        console.error(err);
      }

      if (!db) {
        console.log('No se
encontro la base de datos')
        return;
      }
      let dbo =
db.db('practical1');
      dbo.collection('data').
insertOne( data, (err, res) => {
        if(err) {
          console.error(e
rr);
        } else {
          console.log(`Ob
jeto ${res.insertedId}
insertado.`);
        }
      })
    })
  }
```

```

    })
  } catch (err){
    db.close();
    console.error(err)
    console.log('error en la
bdd')
  }
}

```

#### c. Obtener los últimos 20 datos de la DB:

```

const getLastItems = async () => {
  await mongoClient.connect();
  const db =
mongoClient.db('practical1');
  const data =
db.collection('data');
  const array = await
data.find({}).toArray();
  return array.slice(-20);
}

export { insertData, getLastItems
};

```

#### d. Obtener los últimos 20 datos desde un endpoint en Node:

```

app.get('/getData', async (req,
res) => {
  const data = await
getLastItems();
  res.json(data);
});

```

#### e. Obtener el último dato desde un endpoint en Node:

```

app.get('/getCurrentData', (req,
res) => {
  res.json(currentData);
});

```

- **Link del repositorio de github**

- [https://github.com/Villa01/ACE2\\_1S22\\_G-18/tree/main](https://github.com/Villa01/ACE2_1S22_G-18/tree/main)

- **Link del video en youtube**

- [https://www.youtube.com/watch?v=055xAiI\\_Iic](https://www.youtube.com/watch?v=055xAiI_Iic)

### III. REFERENCIAS

- <https://riunet.upv.es/bitstream/handle/10251/28783/Control%20de%20velocidad%20de%20un%20motor%20DC%20mediante%20la%20utilizaci%C3%B3n%20de%20un%20sensor%20magn%C3%A9tico%20y%20microcontrol.pdf?sequence=1>
- <http://atlas.ideam.gov.co/basefiles/TEXTO---Direccion-del-Viento.pdf>
- M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: <http://www.ctan.org/texarchive/macros/latex/contrib/supported/IEEEtran/>
- <https://lavozdelmuro.net/sabes-que-es-un-pozo-canadiense-puede-ahorrarte-hasta-un-70-en-la-factura-de-la-luz/>