



Tecnológico de Estudios Superiores de Valle de Bravo

Ingeniería En Sistemas Computacionales

Sistemas Embebidos

Manual de prácticas

M. En I.S.C. César Primero Huerta

Valle De Bravo, Estado De México, marzo, 2023





ÍNDICE

COMPETENCIA ESPECÍFICA DE LA ASIGNATURA	3
PRÁCTICAS	3
PRÁCTICA 1: TABLAS DE VERDAD CON INTEGRADOS	3
PRÁCTICA 2: CREACIÓN DE UNA RED NEURONAL	7
PRÁCTICA 3: DISEÑO DE PROTOTIPO.	9
PRÁCTICA 4: RECOLECCIÓN DE DATOS.....	13
PRÁCTICA 5: INTEGRACIÓN DEL MODELO DE LA RED NEURONAL ENTRENADA	16
PRÁCTICA 6: SERVIDOR EN PYTHON	17
PRÁCTICA 7: DESARROLLO DE UN WEB SOCKET.....	20
REFERENCIAS.....	25



COMPETENCIA ESPECÍFICA DE LA ASIGNATURA

- Proporcionar a los estudiantes del curso conocimientos y habilidades relacionados con la tecnología de los procesadores, microcontroladores, dispositivos programables de última generación y sus aplicaciones, a partir de los conceptos de la Arquitectura RISC.
- Conocer la organización y características de los elementos que integran a los sistemas embebidos, incluyendo hardware y software.
- Identificar dispositivos y sus características, para el desarrollo de sistemas embebidos.
- Conocer las técnicas utilizadas para el desarrollo de aplicaciones utilizando sistemas embebidos.

PRÁCTICAS

Práctica 1: Tablas de verdad con integrados

Se debe realizar un multiplicador de 3 bits haciendo uso de compuertas lógicas como son la compuerta nor, and, xor, entre otras. Para esto se hará uso del programa proteus, que nos permite realizar circuitos de manera más fácil haciendo uso del computador. Para esto primeramente se hizo un mapa de Karnaugh, para el cual se utilizó una tabla con en la cual se colocaron los distintos valores y resultados que puede tener el circuito.



	A2	A1	A0	B2	B1	B0	Z5	Z4	Z3	Z2	Z1	Z0
9	0	0	1	0	0	1	0	0	0	0	0	1
10	0	0	1	0	1	0	0	0	0	0	1	0
11	0	0	1	0	1	1	0	0	0	0	1	1
12	0	0	1	1	0	0	0	0	0	1	0	0
13	0	0	1	0	0	1	0	0	0	1	0	1
14	0	0	1	1	1	0	0	0	0	1	1	0
15	0	0	1	1	1	1	0	0	0	1	1	1
16	0	1	0	0	0	0	0	0	0	0	0	0
17	0	1	0	0	0	1	0	0	0	0	1	0
18	0	1	0	0	1	0	0	0	0	1	0	0
19	0	1	0	0	1	1	0	0	0	1	1	0
20	0	1	0	1	0	0	0	1	0	0	0	0
21	0	1	0	1	0	1	0	1	0	0	1	0
22	0	1	0	1	1	0	0	1	1	0	0	0
23	0	1	0	1	1	1	0	1	1	1	1	0
24	0	1	1	0	0	0	0	0	0	0	0	0
25	0	1	1	0	0	1	0	0	0	0	1	1
26	0	1	1	0	1	0	0	0	0	1	1	0
27	0	1	1	0	1	1	0	0	1	0	0	1
28	0	1	1	1	0	0	0	0	1	1	0	0
29	0	1	1	1	0	1	0	0	1	1	1	1
30	0	1	1	1	1	0	0	1	0	0	1	0
31	0	1	1	1	1	1	0	1	0	1	0	1
32	1	0	0	0	0	0	0	0	0	0	0	0
33	1	0	0	0	0	1	0	0	0	1	0	0
34	1	0	0	0	1	0	0	0	1	0	0	0
35	1	0	0	0	1	1	0	0	1	1	0	0
36	1	0	0	1	0	0	0	1	0	0	0	0
37	1	0	0	1	0	1	0	1	0	1	0	0
38	1	0	0	1	1	0	0	1	1	0	0	0
39	1	0	0	1	1	1	0	1	1	1	0	0

Figura 1 Tabla de estados para la realización del mapa Karnaugh

Con esto se hizo uso de la aplicación Boolean Algebra con el cual se pudo realizar el mapa de manera más sencilla. Con esto se obtuvieron las distintas operaciones para realizar el circuito.

Quedando de la siguiente manera

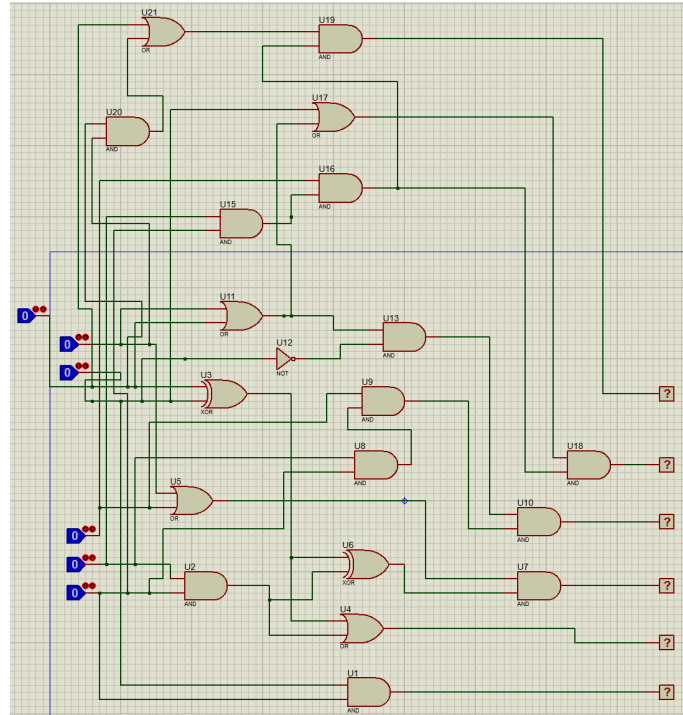


Figura 2 Circuito

En este se pueden ver 6 entradas puesto que cada bit tiene dos entradas de datos, saliente como resultado 6 salidas, para las entradas se usaron logicstate, que pueden mandar señales de 0 y 1 respectivamente, así como en la salida se usaron logicprobe, los cuales reciben y muestran una señal de 0 y 1 respectivamente.

Resultado

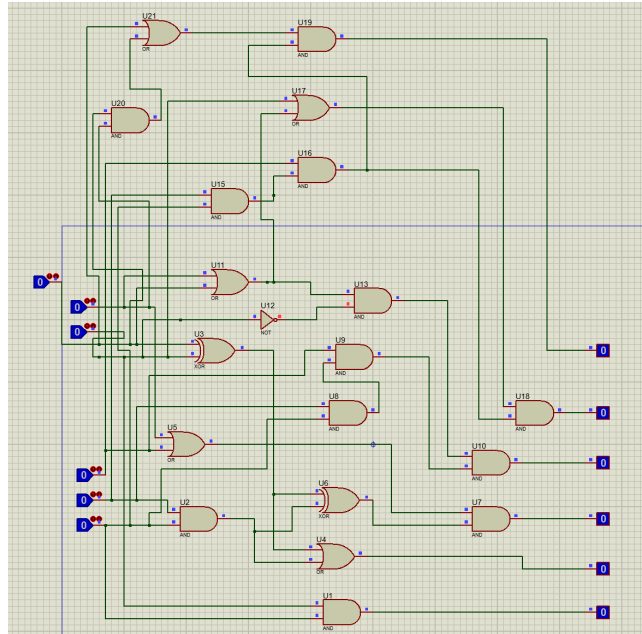


Figura 3 Resultado de operación $0*0$

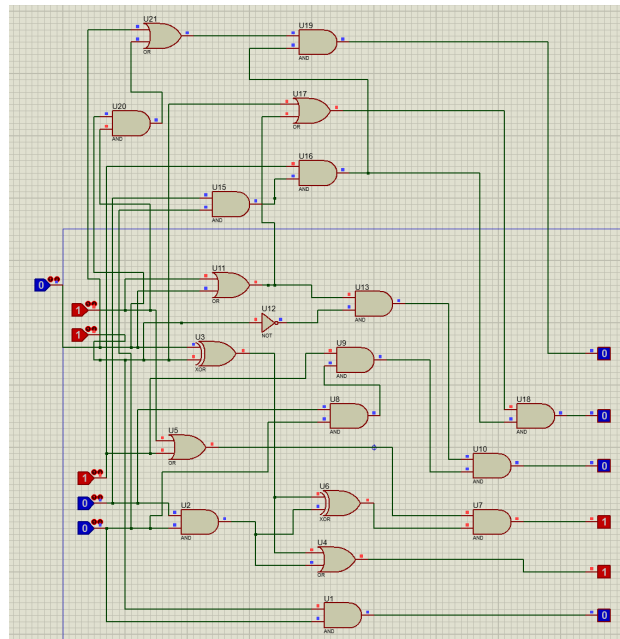


Figura 4 Resultado de operación $3*4$



Práctica 2: Creación de una Red Neuronal

Para el desarrollo de la práctica es necesario generar un archivo nuevo de Numpy

Importar librerías de Python.

```
[1] 1 import pandas as pd
     2 from sklearn.neural_network import MLPClassifier
     8 from sklearn.model_selection import train_test_split
     4 from sklearn.metrics import accuracy_score
```

Figura 5 Librerías de Python a Utilizar

Importar el archivo csv a analizar para su desarrollo.

```
1 df = pd.read_csv('/content/datosarirc(1).csv')
2 df
```

Figura 6 Archivo CSV

En esta sección se deben separar y ordenar los datos los cuales sean tipo input (entrada) y cuáles son los targets (clases)

```
[14] 1 X = df.drop("Target", axis=1)
     2 y = df["Target"]
```

Figura 7 Separación de Datos

Se dividieron los datos en un grupo de conjuntos para su entrenamiento y a su ejecución de prueba.

```
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figura 8 Prueba de la Red Neuronal





Creación del modelo.

```
model = MLPClassifier(hidden_layer_sizes=(10, 10), max_iter=1000, random_state=42)
```

Figura 9 Modelo de la Red Neuronal

Se entrena el modelo con los datos considerados de entrenamiento.

```
[28] 1 model.fit(X_train, y_train)

MLPClassifier
MLPClassifier(hidden_layer_sizes=(10, 10), max_iter=1000, random_state=42)
```

Figura 10 Datos de Entrenamiento para la Red Neuronal

Para lo siguiente se evalúa la precisión del modelo, se calcula la matriz.

```
1 y_pred = model.predict(X_test)

1 accuracy = accuracy_score(y_test, y_pred)
2 print("Precisión: {:.2f}%".format(accuracy*100))

Precisión: 100.00%

1 from sklearn.metrics import confusion_matrix

1 y_pred = model.predict(X_test)

1 cm = confusion_matrix(y_test, y_pred)
2 print(cm)

[[36 0]
 [ 0 6]]

1 import seaborn as sns
```

Figura 11 Precisión y Confusión de matriz para la Red Neuronal

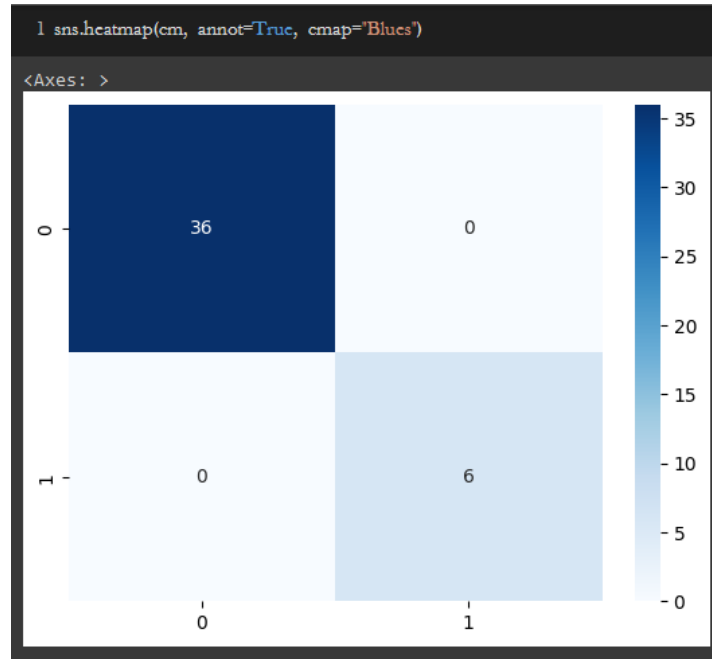


Figura 12 Graficación de la Red Neuronal

Práctica3: Diseño de prototipo.

Para la conexión de los distintos sensores se hicieron uso de los siguientes diagramas de conexión.

Sensor MQ9

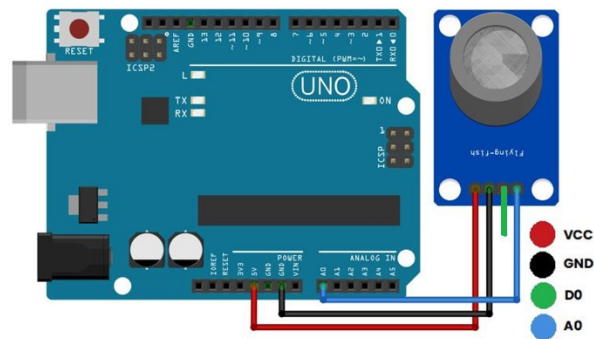


Figura 13 Diagrama de Conexión del Sensor MQ9

(MQ-9 Detector de Gas de Combustible, s.f.)

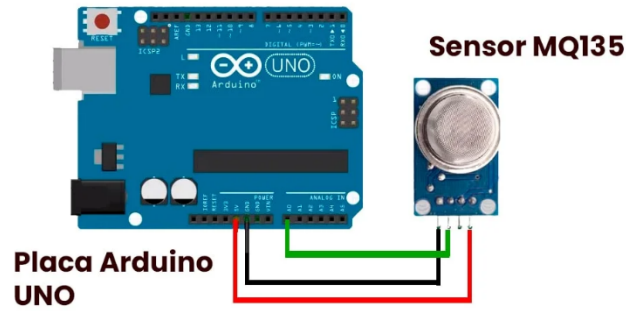


Figura 14 Diagrama de Conexión del Sensor MQ135

(Medidor de CO2 con Arduino, 2021)

Sensor de flama

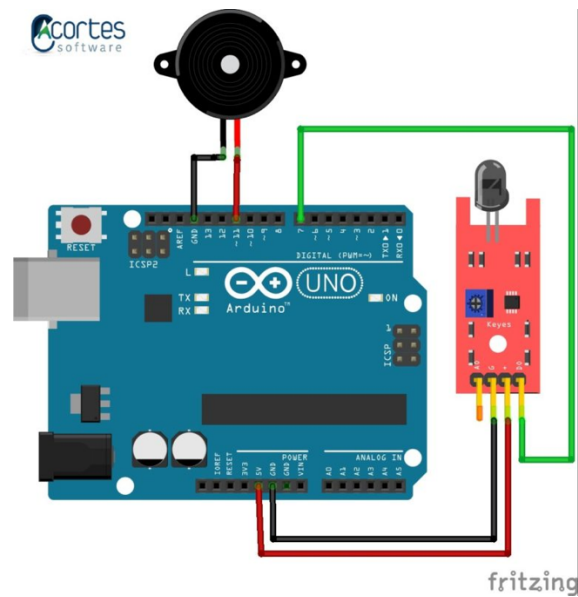


Figura 15 Diagrama de Conexión del Sensor de Flama

(Proyecto 14 – Módulo Sensor de Llama, 2021)

Sensor de temperatura DHT11

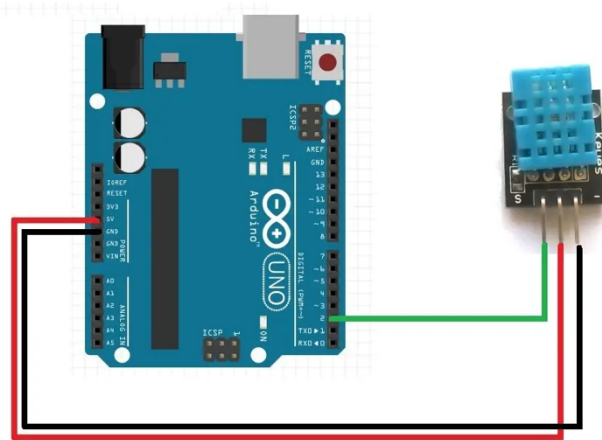


Figura 16 Diagrama de Conexión del Sensor de Temperatura DHT11

(DHT11 Sensor de humedad y temperatura en 10 minutos, s.f.)

Una vez conectados todos los sensores el prototipo queda de la siguiente manera

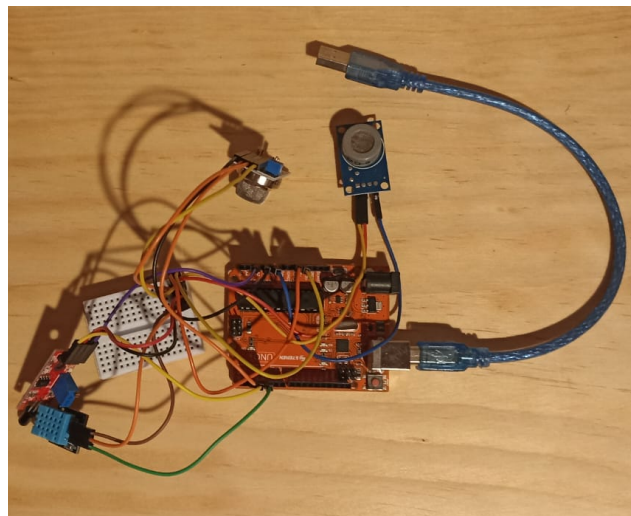


Figura 17 Prototipo Con Sensores Conectados

El código que se utilizo es el siguiente.



```

    // Incluimos librería
    #include <DHT.h>
    // Definimos el pin digital donde se conecta el sensor
    #define DHTPIN 4
    // Dependiendo del tipo de sensor
    #define DHTTYPE DHT11
    // Inicializamos el sensor DHT11
    DHT dht(DHTPIN, DHTTYPE);

    #include "MQ135.h"
    #define ANALOGPIN A1 // Define Analog PIN on Arduino Board
    #define RZERO 206.85 // Define RZERO Calibration Value
    MQ135 gasSensor = MQ135(ANALOGPIN);

    #define MQ1 (0) //define la entrada analogica para el sensor
    #define RL_VALOR (5) //define el valor de la resistencia mde carga en kilo ohms
    #define RAL (9.83) // resistencia del sensor en el aire limpio / R0, que se deriva de la
    // tabla de la hoja de datos
    #define GAS_LP (0)
    String inputstring = ""; //Cadena recibida desde el PC
    float LPCurve[3] = {2.3,0.21,-0.47};
    float Ro = 10;

    int led=13;
    int digitalPin = 3; // Pin digital del módulo
    int analogPin = A2; // Pin analógico del módulo
    int digitalVal; // valor digital
    int analogVal; // valor analógico
    
```

Figura 18 Declaración de variables y pines a utilizar

```

    void setup()
    {
        Serial.begin(9600);
        float rzero = gasSensor.getRZero();
        delay(150);
        //Serial.print("MQ135 RZERO Calibration Value : ");
        //Serial.println(rzero);

        //Serial.println("Iniciando ...");
        //configuracion del sensor
        //Serial.print("Calibrando...\n");
        Ro = Calibracion(MQ1); //Calibrando el sensor. Por favor de asegurarse que el
        // sensor se encuentre en una zona de aire limpio mientras se calibra
        //Serial.print("Calibracion finalizada...\n");
        //Serial.print("Ro=");
        //Serial.print(Ro);
        //Serial.print("kohm");
        //Serial.print("\n");

        pinMode(led, OUTPUT);
        pinMode(digitalPin, INPUT);

        dht.begin();
    }
    
```

Figura 19 Sección Setup





```
void loop() {  
  float ppm = gasSensor.getPPM();  
  delay(150);  
  //Serial.print("CO2 ppm value : ");  
  //valor del co2  
  Serial.print(ppm);  
  Serial.print(",");  
  //Serial.print("LP:");  
  //valor del gas  
  Serial.print(porcentaje_gas(lecturaMQ(MQ1)/Ro,GAS_LP) );  
  //Serial.print( "ppm" );  
  
  Serial.print(",");  
  //Serial.print( "\n");  
  
  //valor flama  
  digitalVal = digitalRead(digitalPin);  
  if(digitalVal == HIGH){ // si hay llama  
    digitalWrite(led, HIGH); // encender LED  
  }else{  
    digitalWrite(led, LOW); // apagar LED  
  }  
  // lectura del pin analogico  
  analogVal = analogRead(analogPin);  
  Serial.print(analogVal); // mostrar el valor por el monitor serial  
  delay(150);  
  
  //temperatura  
  // Esperamos 5 segundos entre medidas  
  delay(150);  
  
  // Leemos la humedad relativa  
  float h = dht.readHumidity();  
  // Leemos la temperatura en grados centígrados (por defecto)  
  float t = dht.readTemperature();  
  // Leemos la temperatura en grados Fahrenheit  
  float f = dht.readTemperature(true);  
  
  // Comprobamos si ha habido algún error en la lectura  
  if (isnan(h) || isnan(t) || isnan(f)) {  
    Serial.println("Error obteniendo los datos del sensor DHT11");  
    return;  
  }  
}
```

Figura 20 Sección Loop de Sensores: MQ9, MQ135, DHT11, FLAME

Práctica 4: Recolección de datos

Los datos recolectados se almacenaron dentro de un archivo de tipo CSV, esto haciendo uso del programa Excel, puesto que este cuenta con un apartado que permite recibir datos de dispositivos externos y almacenarlos en un archivo de manera automática.





Para esto se activó el complemento *Microsoft Data Streamer for Excel*, el cual es el complemento que permite que dispositivos externos se conecten a Excel y puedan enviarle datos directamente al programa. (Excel 365: Conectar Arduino con Excel usando Data Streamer, 2021)

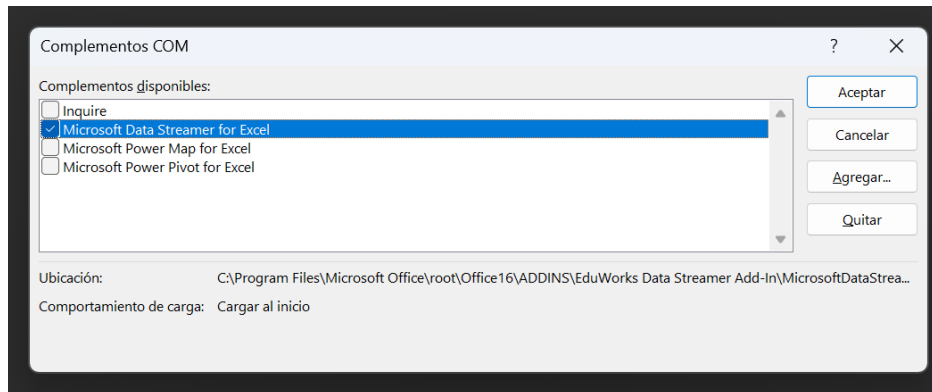


Figura 21 Ventana del listado de complementos de los puertos COM para el programa Excel

Con este apartado activado hace más fácil la recolección de datos, puesto que, en otro contexto se debería de hacer uso de programas externos que complicarían esta tarea.

Entrada de datos (desde Arduino Uno (COM7))
 Los datos procedentes del origen de datos actual aparecerán abajo según se reciban.

Datos actuales										
TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10
22:51:56.63	341.19	0	1000	23	22.2	71.96	:1.0869.94			

Información histórica										
TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10
22:51:56.63	341.19	0	1000	23	22.2	71.96	:1.0869.94			

Figura 22 Entrada de Datos de los Sensores

Para estas pruebas lo que se realizó fue presentar 3 estados, uno donde este sin ninguna anomalía en el ambiente, el segundo estado habría gas LP en el aire y el tercero sería la presencia de una flama, esto haciendo uso de un encendedor, pues este nos permite tener los 3 estados de una manera más controlada, para evitar algún incidente.



Datos actuales

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7
22:54:59.85	369.34	0	1003	25	22.3	72.14	21.2470.23

Información histórica

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7
22:54:59.85	369.34	0	1003	25	22.3	72.14	21.2470.23
22:54:44.58	376.64	0	1005	25	22.3	72.14	21.2470.23
22:54:29.30	369.34	0	1000	24	22.3	72.14	21.2170.18
22:54:14.03	369.34	0	1000	25	22.2	71.96	21.1370.03
22:53:58.76	406.96	0	1000	25	22.1	71.78	21.0269.83
22:53:43.49	384.05	0	1004	24	22.1	71.78	20.9969.79
22:53:28.22	362.14	0	1000	24	22.1	71.78	20.9969.79
22:53:12.95	348.07	0	1004	24	22.2	71.96	21.1069.98
22:52:57.67	348.07	0	1002	24	22.2	71.96	21.1069.98
22:52:42.40	362.14	0	1003	24	22.2	71.96	21.1069.98
22:52:27.13	341.19	0	1002	24	22.2	71.96	21.1069.98
22:52:11.85	341.19	0	1000	23	22.2	71.96	21.0869.94
22:51:56.63	341.19	0	1000	23	22.2	71.96	21.0869.94

Figura 23 Entrada de Datos Caso 1

Datos actuales

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7
22:58:03.12	3393.54	0	944	24	23.1	73.58	21.0971.77

Información histórica

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7
22:58:03.12	3393.54	0	944	24	23.1	73.58	21.0971.77
22:57:47.84	9832.46	0	1014	24	22.6	72.68	21.5470.78
22:57:32.57	246475.35	0	1004	24	22.3	72.14	21.2170.18
22:57:17.30	1109102.87	-18061	1004	24	22.3	72.14	21.2170.18
22:57:02.03	553360.25	18	1002	24	22.3	72.14	21.2170.18
22:56:46.76	12592.53	0	1005	24	22.3	72.14	21.2170.18
22:56:31.48	664871	0	1004	24	22.2	71.96	21.1069.98

Figura 24 Entrada de Datos Caso 2 y Caso3

Con esto se genera un archivo CSV, en los cuales solo se coloca el nombre de cada columna, para acomodar los datos, según el sensor y el valor que este entregue. Estos datos se pueden dejar así, siendo que los sensores MQ usan medidas de ppm (partícula por millón), mientras los otros sensores no, por lo cual se deben de normalizar datos para poder tener una medida entre estos datos, para así, poder tener un rango más cómodo para trabajar.



Práctica 5: Integración del Modelo de la Red Neuronal Entrenada

Se importaron las bibliotecas necesarias. pandas es utilizada para manipular y analizar datos tabulares, numpy proporciona funcionalidades para operaciones numéricas eficientes, MLPClassifier es la clase que proporciona la implementación de redes neuronales en scikit-learn, train_test_split se utiliza para dividir los datos en conjuntos de entrenamiento y prueba, accuracy_score es una métrica de evaluación para calcular la precisión del modelo y StandardScaler se utiliza para estandarizar los datos.

```
import pandas as pd
import numpy as np
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
```

Figura 25 Importar Librerías de Python

se utiliza el método drop() del DataFrame df para eliminar una columna llamada "Clase". El parámetro axis=1 se utiliza para indicar que se elimine una columna (en lugar de una fila). El resultado de esta operación se asigna a la variable X, que representa las características o variables independientes.

```
X = df.drop("Clase", axis=1)
y = df["Clase"]
```

Figura 26 Eliminar Columna

La función train_test_split() toma como entrada las características (X) y la variable objetivo (y) y los divide en cuatro conjuntos: X_train, X_test, y_train y y_test. X_train y y_train son los conjuntos de entrenamiento que se utilizarán para entrenar el modelo.





X_test y y_test son los conjuntos de prueba que se utilizarán para evaluar el rendimiento del modelo en datos no vistos. El parámetro test_size=0.2 indica que el 20% de los datos se utilizarán para el conjunto de prueba, mientras que el 80% restante se utilizarán para el conjunto de entrenamiento. El parámetro random_state=42 se utiliza para asegurar que la división de los datos sea reproducible. Al proporcionar un valor fijo (en este caso, 42), se obtendrá la misma división de los datos cada vez que se ejecute el código.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figura 27 Entrenamiento de la Red

Se utiliza la función accuracy_score() de scikit-learn para calcular la precisión del modelo. Toma como entrada los valores reales de la variable objetivo (y_test) y los valores predichos por el modelo (y_pred). El resultado se asigna a la variable accuracy.

```
accuracy = accuracy_score(y_test, y_pred)  
print("Precisión: {:.2f}%".format(accuracy*100))
```

```
Precisión: 73.56%
```

Figura 28 Precisión del Modelo

Práctica 6: Servidor en Python

Se creó un servidor en Python para sacar los valores de la url generada desde el servidor creado en php para validarlos en la red neuronal y que devuelva el valor de la simulación.



```
servidor.py - C:\Users\Corey\Downloads\servidor.py (3.11.3)
File Edit Format Run Options Window Help
#Importar las librerias para la api
import pandas as pd
import numpy as np
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
import math

from fastapi import FastAPI
app = FastAPI()

#Colocar el codigo de la red neuronal
#Prueba de la extraccion de una red neuronal
# Leer los datos
df = pd.read_csv('C:\\Users\\Corey\\Downloads\\datos22.csv')
X = df.drop("Clase", axis=1)
y = df["Clase"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = MLPClassifier(hidden_layer_sizes=(10, 10), max_iter=1000, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

#mq9=float(input("Ingresa un valor para mq2 -> "))
#mq135=float(input("Ingresa un valor para mq135 -> "))
#dth11=float(input("Ingresa un valor para dth11 -> "))
#flame=float(input("Ingresa un valor para flame-> "))

#colocar la url para enviar los valores
@app.get("/red_test")
def setValues(mq2:float,mq135:float,dth11:float,flame:float):
    #Llamada a la red neuronal
    red_model = model.predict([[mq2,mq135,dth11,flame]])
    return str(red_model)
    #return {"Hello World!" + str(red_model)}
#print(red_model)
```

Figura 29 Servidor en Python

Para verificar el funcionamiento de que el servidor está recibiendo correctamente los parámetros y los está simulando en la base de datos, se debe hacer una prueba con el servidor local donde se mandan los parámetros por medio del método get y la red neuronal regresa el resultado de la evaluación.

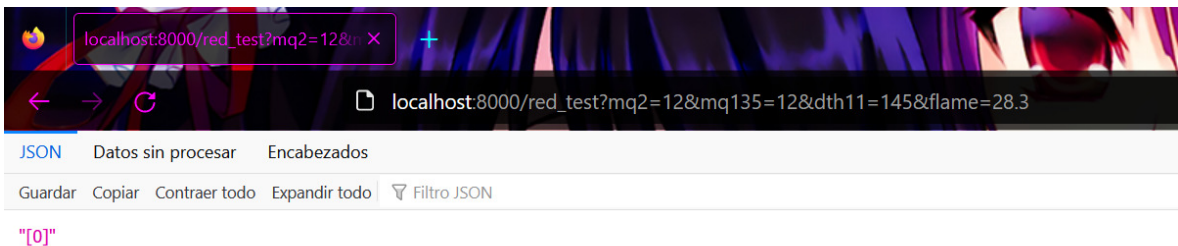


Figura 30 Verificar Parámetros en el Servidor





Para poder verificar el funcionamiento del servidor se debe ejecutar el siguiente comando que permite levantar el servidor.

```
corey@Legna_Riay MINGW64 ~/Downloads
$ uvicorn servidor:app --port=8080
INFO: Started server process [1176]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://127.0.0.1:8080 (Press CTRL+C to quit)
C:\Users\Corey\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:439: UserWarning:
have valid feature names, but MLPClassifier was fitted with feature names
warnings.warn(
INFO: 127.0.0.1:50383 - "GET /red_test?mq2=12&mq135=12&dth11=145&flame=28.3 HTTP/1.1" 200 OK
```

Figura 31 Verificar Parámetros en el Servidor

Práctica 7: Desarrollo de un Web Socket

Creación de la base de datos para guardar los registros.

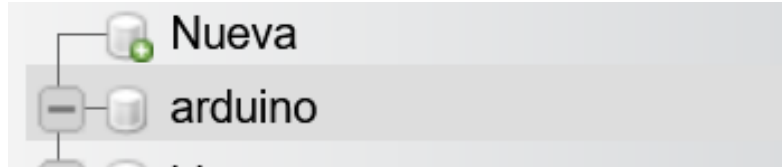


Figura 32 Crear una nueva base de datos

Estructura de la tabla, en esta se guardarán los datos y el resultado de la red neuronal.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(11)			No	Ninguna			Cambiar Eliminar Más
2	mq135	double			Sí	NULL			Cambiar Eliminar Más
3	mq2	double			Sí	NULL			Cambiar Eliminar Más
4	ky026	double			Sí	NULL			Cambiar Eliminar Más
5	dht11t	double			Sí	NULL			Cambiar Eliminar Más
6	dht11f	double			Sí	NULL			Cambiar Eliminar Más
7	dht11c	double			Sí	NULL			Cambiar Eliminar Más
8	tipo	int(11)			Sí	NULL			Cambiar Eliminar Más
9	fecha	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Cambiar Eliminar Más

Figura 33 Tabla a utilizar para almacenar

Se creo otra tabla que se conectara con la aplicación y se enviaran los datos del Arduino.



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 id	int(11)			No	Ninguna		AUTO_INCREMENT	
<input type="checkbox"/>	2 mq135	double			Sí	NULL			
<input type="checkbox"/>	3 mq2	double			Sí	NULL			
<input type="checkbox"/>	4 ky026	double			Sí	NULL			
<input type="checkbox"/>	5 dht11t	double			Sí	NULL			
<input type="checkbox"/>	6 dht11f	double			Sí	NULL			
<input type="checkbox"/>	7 dht11c	double			Sí	NULL			
<input type="checkbox"/>	8 resultado	int(11)			Sí	NULL			
<input type="checkbox"/>	9 create_at	datetime			Sí	NULL			
<input type="checkbox"/>	10 update_at	datetime			Sí	NULL			

Figura 34

Se creo un nuevo proyecto con el comando

composer create-project --prefer-dist laravel/laravel nombre.

```
$ composer create-project --prefer-dist laravel/laravel incendios
Creating a "laravel/laravel" project at "./incendios"
Info from https://repo.packagist.org: #StandWithUkraine
Cannot use laravel/laravel's latest version v10.2.2 as it requires php ^8.1 which is not satisfied by your platform.
Installing laravel/laravel (v9.5.2)
- Installing laravel/laravel (v9.5.2): Extracting archive
Created project in C:\Users\alber\Documents\Diagnostico\incendios
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/deprecations (v1.0.0)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/instantiator (1.5.0)
- Locking doctrine/lexer (2.1.0)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (3.2.5)
- Locking fakerphp/faker (v1.22.0)
- Locking filp/whoops (2.15.2)
- Locking fruitcake/php-cors (v1.2.0)
- Locking graham-campbell/result-type (v1.1.1)
- Locking guzzlehttp/guzzle (7.7.0)
- Locking guzzlehttp/promises (2.0.0)
- Locking guzzlehttp/psr7 (2.5.0)
- Locking guzzlehttp/uri-template (v1.0.1)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v9.52.7)
- Locking laravel/pint (v1.5.0)
- Locking laravel/sail (v1.22.0)
- Locking laravel/sanctum (v3.2.5)
```

Figura 35 Creación del proyecto

Instalar predis con el siguiente comando





```

alber@Huawei MINGW64 ~/Documents/Diagnostico/incendios
$ composer require predis/predis
bash: composer: command not found

alber@Huawei MINGW64 ~/Documents/Diagnostico/incendios
$ composer require predis/predis
./composer.json has been updated
Running composer update predis/predis
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking predis/predis (v2.1.2)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Downloading predis/predis (v2.1.2)
- Installing predis/predis (v2.1.2): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

85 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
    
```

Figura 36 Instalación de Predis

Realizar la migración con la base de datos

```

alber@Huawei MINGW64 ~/Documents/Diagnostico/incendios
$ php artisan migrate

 INFO  Preparing database.

s >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

added 204 packages in 12s
    
```

Figura 37 Migración de la base de datos

Realizar la instalación de laravel echo server

```

alber@Huawei MINGW64 ~/Documents/Diagnostico/incendios
$ npm install -g laravel-echo-server
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

changed 204 packages in 39s

17 packages are looking for funding
run `npm fund` for details
    
```

Figura 38 Instalación laravel echo server





Realzar la configuración de laravel echo server

```
alber@Huawei MINGW64 ~/Documents/Diagnostico/incendios
$ laravel-echo-server init
? Do you want to run this server in development mode? Yes
? Which port would you like to serve from? 6001
? Which database would you like to use to store presence channel members? redis
? Enter the host of your Laravel authentication server. http://localhost
? Will you be serving on http or https? http
? Do you want to generate a client ID/Key for HTTP API? Yes
? Do you want to setup cross domain access to the API? No
? What do you want this config to be saved as? laravel-echo-server.json
appId: 8898d0a1156e2292
key: 5602f26794349f26b2709f22c9dd4157
Configuration file saved. Run laravel-echo-server start to run server.
```

Figura 39 Configuración

Esta configuración se guardó dentro del archivo *laravel.echo-server.json*, al abrirlo se observa lo siguiente.

```
1 {
2   "authHost": "http://localhost",
3   "authEndpoint": "/broadcasting/auth",
4   "clients": [
5     {
6       "appId": "8898d0a1156e2292",
7       "key": "5602f26794349f26b2709f22c9dd4157"
8     }
9   ],
10  "database": "redis",
11  "databaseConfig": {
12    "redis": {},
13    "sqlite": {
14      "databasePath": "/database/laravel-echo-server.sqlite"
15    }
16  },
17  "devMode": true,
18  "host": null,
19  "port": "6001",
20  "protocol": "http",
21  "socketio": {},
22  "secureOptions": 67108864,
23  "sslCertPath": "",
24  "sslKeyPath": "",
25  "sslCertChainPath": "",
26  "sslPassphrase": "",
27  "subscribers": {
28    "http": true,
29    "redis": true
30  },
31  "apiOriginAllow": {
32    "allowCors": false,
33    "allowOrigin": "",
34    "allowMethods": "",
35    "allowHeaders": ""
36  }
37 }
```

Figura 40



Se instalo laravel-echo y socket.io

```
alber@Huawei MINGW64 ~/Documents/Diagnostico/incendios
$ npm install laravel-echo

added 1 package, and audited 23 packages in 2s

4 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

alber@Huawei MINGW64 ~/Documents/Diagnostico/incendios
$ npm install socket.io-client

added 9 packages, and audited 32 packages in 5s

4 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Figura 41



Referencias

- *AWS AMAZON.* (s.f.). Obtenido de ¿Qué es una red neuronal?: <https://aws.amazon.com/es/what-is/neural-network/>
- *Industria 4.0, I. A.* (22 de Octubre de 2019). *atriainnovation.* Obtenido de Qué son las redes neuronales y sus funciones: <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>
- *mathworks.* (1994-2023). Obtenido de ¿Qué es una red neuronal?: <https://la.mathworks.com/discovery/neural-network.html>
- *Compuertas Lógicas.* (s.f.). Obtenido de Logicbus: <https://www.logicbus.com.mx/compuertas-logicas.php>
- *Mapas de Karnaugh.* (s.f.). Obtenido de UNAM CUATITLAN: http://suayed.cuautitlan.unam.mx/uapas/08_Mapas_Karnaugh/
- *MULTIPLICADOR DE 2 BITS.* (16 de Mayo de 2020). Obtenido de YouTube: <https://www.youtube.com/watch?v=E4GiJalVHpo&t=123s>
- *¿Probabilidad de que ocurra un incendio en mi vivienda?* (01 de Marzo de 2020). Obtenido de EXTIHOUSE: <https://extinhouse.es/probabilidad-de-que-ocurra-un-incendio-en-mi-vivienda/>
- *ARDUINO BASED FIRE DETECTION AND CONTROL SYSTEM.* (2020). *International Journal of Eengineering Applied Sciences and Technology*, 447-453.
- *Arduino UNO R3 [A000066] Placa de microcontrolador.* (s.f.). Obtenido de Amazon: https://www.amazon.com.mx/Arduino-Org-A000066-R3-microcontrolador-a000066/dp/B008GRTSV6/ref=asc_df_B008GRTSV6/?tag=gledskshopmx-





20&linkCode=df0&hvadid=339643974669&hvpos=&hvnetw=g&hvrnd=10630271858
713892161&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmidl=&hvloci

- *Bomba De Achique Sumergible Seaflo 350 Gph.* (s.f.). Obtenido de Mercado Libre: https://articulo.mercadolibre.com.mx/MLM-764067237-bomba-de-achique-sumergible-seaflo-350-gph-_JM?matt_tool=18267311&matt_word=&matt_source=google&matt_campaign_id=15700894205&matt_ad_group_id=154310708868&matt_match_type=&matt_network=g&matt_device=c&mat
- *DHT11 Sensor de humedad y temperatura en 10 minutos.* (s.f.). Obtenido de HetPro-Store: <https://hetpro-store.com/TUTORIALES/sensor-dht11/>
- *Electroválvula Válvula Agua ½ Solenoide 12v, Arduino, Pic.* (s.f.). Obtenido de Mercado Libre: https://articulo.mercadolibre.com.mx/MLM-803405853-electrovalvula-valvula-agua-solenoide-12v-arduino-pic-_JM?matt_tool=91188883&matt_word=&matt_source=google&matt_campaign_id=15698047816&matt_ad_group_id=143431914600&matt_match_type=&matt_network=g&matt_d
- *Excel 365: Conectar Arduino con Excel usando Data Streamer.* (28 de Enero de 2021). Obtenido de YouTube: <https://www.youtube.com/watch?v=UzP8gdRsM8Q>
- *Forest Fire Detection Using Arduino Based WSN.* (2021). *International Conference on Communication and Information Processing* , 1-6.





- Intelligent fire detection and alert system using labVIEW. (2019). *International Journal of Electrical and Computer Engineering*, 1842-1849.
- IoT based forest fire detection system. (2018). *International Journal of Engineering & Technology*, 124-126.
- IOT fire detection system using sensor with Arduino. (2019). *ausrevista*, 74-78.
- *Medidor de CO2 con Arduino*. (03 de Mayo de 2021). Obtenido de medidos de CO2: <https://medidordeco2.info/medidor-de-co2-arduino/>
- *MQ-9 Detector de Gas de Combustible*. (s.f.). Obtenido de UNIT electronics: <https://uelectronics.com/producto/mq-9-detector-de-gas-de-combustible/>
- *Proyecto 14 – Módulo Sensor de Llama*. (02 de Febrero de 2021). Obtenido de Acortes Software: <https://acortes.co/proyecto-14-modulo-sensor-de-llama/>
- *Sensor de flama* . (02 de Julio de 2020). Obtenido de Talos Electronic: <https://www.taloselectronics.com/blogs/tutoriales/sensor-de-flama>
- *Sensor de temperatura y humedad relativa DHT11*. (s.f.). Obtenido de NAYLAMP: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/57-sensor-de-temperatura-y-humedad-relativa-dht11.html>
- *Sensor MQ-135 gas Calidad de aire*. (s.f.). Obtenido de NAYLAMP: <https://naylampmechatronics.com/sensores-gas/73-sensor-mq-135-gas-calidad-aire.html>



- ¿Probabilidad de que ocurra un incendio en mi vivienda? (01 de Marzo de 2020). Obtenido de EXTIHOUSE: <https://extinhouse.es/probabilidad-de-que-ocurra-un-incendio-en-mi-vivienda/>
- ARDUINO BASED FIRE DETECTION AND CONTROL SYSTEM. (2020). *International Journal of Eengineering Applied Sciences and Technology*, 447-453.
- *Arduino UNO R3 [A000066] Placa de microcontrolador.* (s.f.). Obtenido de Amazon: https://www.amazon.com.mx/Arduino-Org-A000066-R3-microcontrolador-a000066/dp/B008GRTSV6/ref=asc_df_B008GRTSV6/?tag=gledskshopmx-20&linkCode=df0&hvadid=339643974669&hvpos=&hvnetw=g&hvrnd=10630271858713892161&hvppone=&hvptwo=&hvqmt=&hvdev=c&hvdvcm dl=&hvloci
- *Bomba De Achique Sumergible Seaflo 350 Gph.* (s.f.). Obtenido de Mercado Libre: https://articulo.mercadolibre.com.mx/MLM-764067237-bomba-de-achique-sumergible-seaflo-350-gph-_JM?matt_tool=18267311&matt_word=&matt_source=google&matt_campaign_id=15700894205&matt_ad_group_id=154310708868&matt_match_type=&matt_network=g&matt_device=c&mat
- *DHT11 Sensor de humedad y temperatura en 10 minutos.* (s.f.). Obtenido de HetPro-Store: <https://hetpro-store.com/TUTORIALES/sensor-dht11/>
- *Electroválvula Válvula Agua ½ Solenoide 12v, Arduino, Pic.* (s.f.). Obtenido de Mercado Libre: <https://articulo.mercadolibre.com.mx/MLM-803405853-electrovalvula-valvula-agua-solenoide-12v-arduino-pic->





_JM?matt_tool=91188883&matt_word=&matt_source=google&matt_campaign_id=156
98047816&matt_ad_group_id=143431914600&matt_match_type=&matt_network=g&m
att_d

- *Excel 365: Conectar Arduino con Excel usando Data Streamer.* (28 de Enero de 2021).
Obtenido de YouTube: <https://www.youtube.com/watch?v=UzP8gdRsM8Q>
- *Forest Fire Detection Using Arduino Based WSN.* (2021). *International Conference on Communication and Information Processing* , 1-6.
- *Intelligent fire detection and alert system using labVIEW.* (2019). *International Journal of Electrical and Computer Engineering*, 1842-1849.
- *IoT based forest fire detection system.* (2018). *International Journal of Engineering & Technology*, 124-126.
- *IOT fire detection system using sensor with Arduino.* (2019). *ausrevista*, 74-78.
- *Medidor de CO2 con Arduino.* (03 de Mayo de 2021). Obtenido de medidos de CO2:
<https://medidordeco2.info/medidor-de-co2-arduino/>
- *MQ-9 Detector de Gas de Combustible.* (s.f.). Obtenido de UNIT electronics:
<https://uelectronics.com/producto/mq-9-detector-de-gas-de-combustible/>
- *Proyecto 14 – Módulo Sensor de Llama.* (02 de Febrero de 2021). Obtenido de Acortes
Software: <https://acortes.co/proyecto-14-modulo-sensor-de-llama/>
- *Sensor de flama* . (02 de Julio de 2020). Obtenido de Talos Electronic:
<https://www.taloselectronics.com/blogs/tutoriales/sensor-de-flama>





- *Sensor de temperatura y humedad relativa DHT11.* (s.f.). Obtenido de NAYLAMP:
<https://naylampmechatronics.com/sensores-temperatura-y-humedad/57-sensor-de-temperatura-y-humedad-relativa-dht11.html>
- *Sensor MQ-135 gas Calidad de aire.* (s.f.). Obtenido de NAYLAMP:
<https://naylampmechatronics.com/sensores-gas/73-sensor-mq-135-gas-calidad-aire.html>