

INFORME

Documento de informe del proyecto de sistema de riego.

INTEGRANTES:

Calderone Pablo

Colatrilla Lucio

Nasello Ignacio

Saldías Valentín

Quincke Juan Cruz

COMISIÓN: TUP 8



Índice

1. Introducción.....	3
1.1. Problemática.....	3
1.2. Solución Propuesta.....	3
1.3. De que se trata y cómo funciona el proyecto.....	4
1.4. Arduino.....	4
2. Desarrollo y diseño.....	5
2.1. Diagrama en bloque.....	5
2.2. Circuito Esquemático.....	5
2.3. Materiales y costos.....	6
3. Componentes principales.....	6
3.1. LDR.....	6
3.2. Sensor de humedad.....	7
3.3. Sensor de temperatura.....	9
3.4. Display.....	10
4. Ejecución.....	11
4.1. Diagrama de Gantt.....	11
4.2. Imágenes del proyecto.....	11
5. Programa.....	11
6. Información de la placa principal Arduino UNO.....	14
7. Conclusión.....	17

1.Introducción:

En este informe se encuentra toda la información respecto a la ejecución del proyecto “Sistema de riego” realizado por alumnos de la Tecnicatura de Programación en la materia de “Sistemas de Procesamiento de Datos”. Este mismo consta de un sistema de riego automatizado realizado con una placa Arduino UNO en el cual contiene un sensor de temperatura, un medidor de humedad, un LDR, además de dos motores de corriente continua, un motor representa la ventilación y el otro el riego, los cuales, los datos de estos componentes se exhiben en una pantalla LCD, que también se muestra su nivel de luminosidad mediante un led.

1.1. Problemática

Una de las principales problemáticas que se encontró con motivo de realizar el proyecto fue la sequía que se produjo desde 2019, Argentina y varios países vecinos venían experimentando condiciones de sequía. En los últimos cuatro meses de 2022, se recibió solo el 44% de la precipitación media, lo que se traduce en el menor valor de 35 años.

1.2. Solución Propuesta

La solución que se propuso en este proyecto es la realización de un sistema de riego para poder enfrentar las prolongadas sequías que se presentan como consecuencia de intensos veranos.

Los sistemas de riego constituyen una opción a través de la cual no solo contribuye al ahorro del agua, haciendo más efectiva su distribución, sino que facilita otros procesos como la fumigación de las plantas o la irrigación de fertilizantes.

Esta solución fue pensada para el ahorro de tiempo ya que, el sistema trabaja de una manera independiente y no hace falta estar constantemente pendiente del trabajo.

1.3. De que se trata y cómo funciona el proyecto

Un sistema de riego es un sistema totalmente pensado y diseñado para suministrar agua de manera controlada a las plantas en jardines, todo tipo de cultivo agrícola o áreas verdes. Estos sistemas tienen como objetivo mantener el suelo húmedo de manera constante y eficiente para promover el crecimiento saludable de las plantas.

Consiste en manipular y controlar la temperatura, humedad y cantidad de luz que recibe el cultivo.

Mediante los datos obtenidos de los sensores que conforman este proyecto, podemos activar los ventiladores, toda la iluminación del campo y el riego automáticamente, siempre dependiendo las condiciones que se debe cumplir para el mejor funcionamiento del sistema.

Ejemplo de funcionamiento del sistema:

- Cuando la temperatura sea mayor a 30 grados se activará el sistema de ventilación.
- Cuando la humedad sea menor a un 70%, se activará el sistema de riego.
- Cuando el LDR detecte menos sensibilidad de luz, se encenderá la iluminación.
- Todos los valores obtenidos de los sensores, se mostrarán en el display.

1.4. Arduino

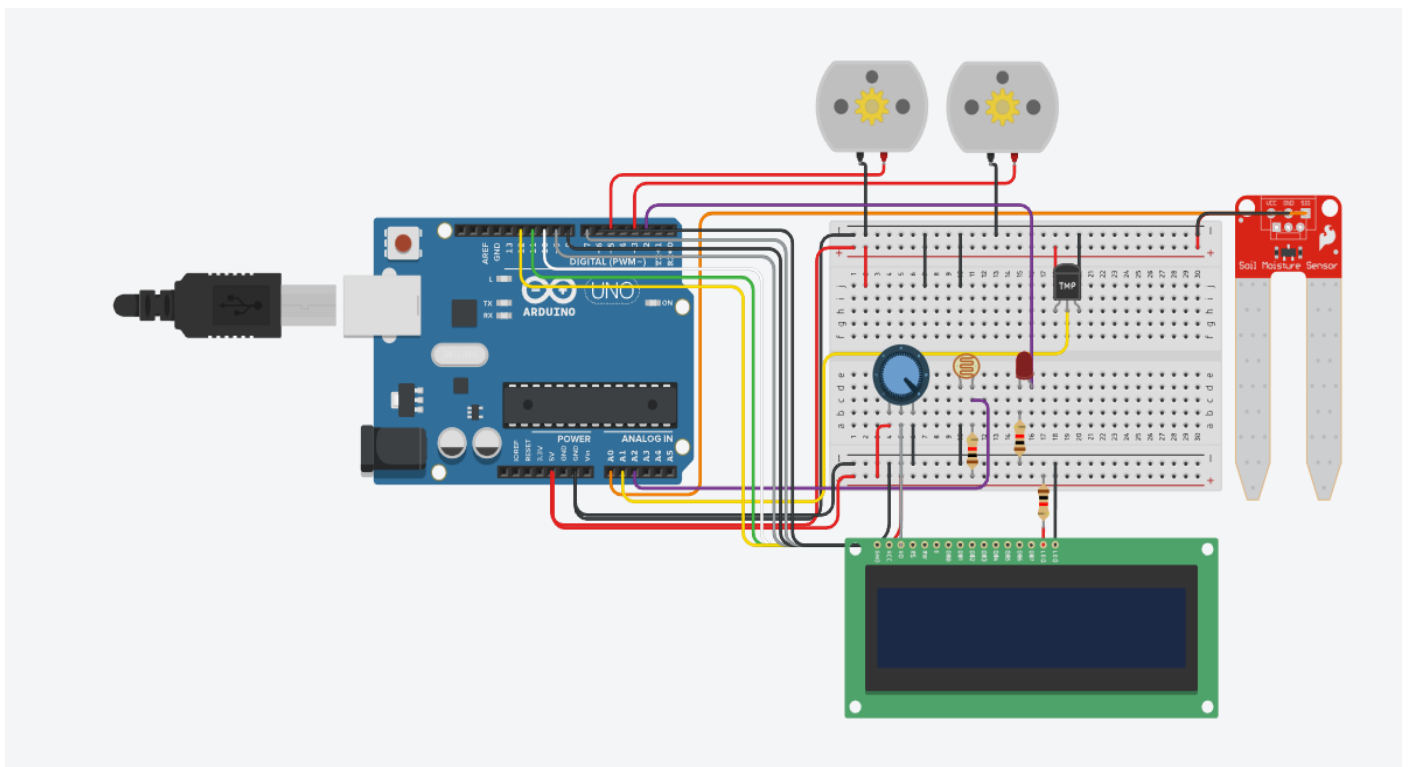
Para la realización del proyecto se utilizó el programa de Arduino, este es una plataforma electrónica de código abierto cuyos principios son contar con software y hardware de fácil uso, básicamente lo que permite esta herramienta es la generación de infinidad de tipos de microordenadores de una sola placa, que luego pueden tener una amplia variedad de usos según la necesidad de la persona que lo cree.

2. Desarrollo y diseño

2.1 Diagrama en bloque



2.2. Circuito esquemático



2.3. Materiales y costos

<i>Productos</i>	<i>Precio</i>	<i>Cantidad</i>	<i>Total</i>
Arduino Uno R3	\$ 11.790	1	\$ 11.790
Cables macho - macho	\$ 60	20	\$ 1.200
Display LCD Serie L2C	\$ 5.300	1	\$ 5.300
LDR	\$ 200	1	\$ 200
Led's de 5mm	\$ 93	1	\$ 93
Motores de CC	\$ 30.000	2	\$ 60.000
Potenciómetro	\$ 850	1	\$ 850
Resistencias 220 ohm	\$ 80	3	\$ 180
Sensor de humedad	\$ 1.410	1	\$ 1.410
Sensor de temperatura	\$ 3.000	1	\$ 3.000
<i>Total</i>			\$ 84023

3. Materiales principales

3.1. LDR

- ¿Qué es y cómo funciona?

El LDR es un componente electrónico que tiene una resistencia variable que cambia con la intensidad de la luminosidad, básicamente es una fotorresistencia sensible a la luz. Su resistencia disminuye a medida que aumenta la intensidad de la luz, y cuando el nivel de la misma disminuye la resistencia aumenta.

En presencia de la luz intensa su valor es aproximadamente de unos 500 Ω y en plena oscuridad su resistencia es de unos 200 K Ω a 10 Ω . La resistencia de un LDR es inversamente proporcional a la intensidad de la luz que incide en ella.



3.2. Sensor de humedad

- ¿Para qué sirven y como es el funcionamiento de los sensores de humedad del suelo?

Los sensores de humedad miden una señal eléctrica que calcula la cantidad de agua en el suelo, lo cual permite determinar el volumen de agua almacenado en este después de un riego o una lluvia, calcula el consumo de agua por el cultivo en un día o una semana y determina la eficiencia del riego.

- Ejemplo de funcionamiento:

Por ejemplo, un valor promedio de humedad de 28,5% medido entre 0 y 20 cm de profundidad indica que en esa capa de 20 cm de suelo hay una lámina de 57 mm ($28,5 \times 20/10$), pero no indica si están disponibles o no para las plantas.

Los sensores de humedad son diferentes a los de potencial mátrico, porque estos últimos lo que determinan es el agua en el suelo disponible para la planta.

- Como se desarrollan y como operan los sensores de humedad:

Los sensores de humedad no miden directamente esta variable, sino que se basan en las técnicas como la reflectometría de dominio (TDR), la capacitancia (FDR) y la conductividad eléctrica aparente (CE), que son mediciones que se deben convertir a valores de humedad volumétrica mediante una calibración. Comúnmente se conocen como sensores de humedad TDR o FDR.

- ¿Como podemos explayar esta información para programar riegos?

Cuando se mide la humedad del suelo para programar los riegos se debe conocer el rango óptimo de humedad en el cual las plantas no se afectan por exceso o déficit de agua. Esta información es específica para cada tipo de suelo y corresponde al valor de humedad a un potencial mátrico entre -20 kPa y -75 kPa.

En ese sentido, para un potencial mátrico de -75 kPa, el cual es el valor de umbral para el inicio del riego, la humedad para un suelo arenoso es del 12%; para un suelo arcilloso, del 28%; y para uno franco arcilloso, del 33% aproximadamente.

- Su instalación:

La profundidad de instalación de los sensores de humedad depende de la longitud de los electrodos y de la dirección de colocación. Por ejemplo, un sensor con electrodos de 20 cm de longitud instalados verticalmente desde la superficie mide la humedad media entre 0 y 20 cm de profundidad. Se mide la humedad entre 20 y 40 cm.

Si los sensores se instalan horizontalmente se mide la humedad media a la profundidad que se instalen: un sensor de 20 cm de longitud instalado horizontalmente a 30 cm, mide la humedad media a esa profundidad del suelo.

Su instalación es sencilla y rápida: cavar un orificio con barreno a la profundidad requerida, insertar el sensor vertical u horizontal, tapar de nuevo el orificio y compactarlo ligeramente para evitar el flujo preferencial de agua en el sitio.

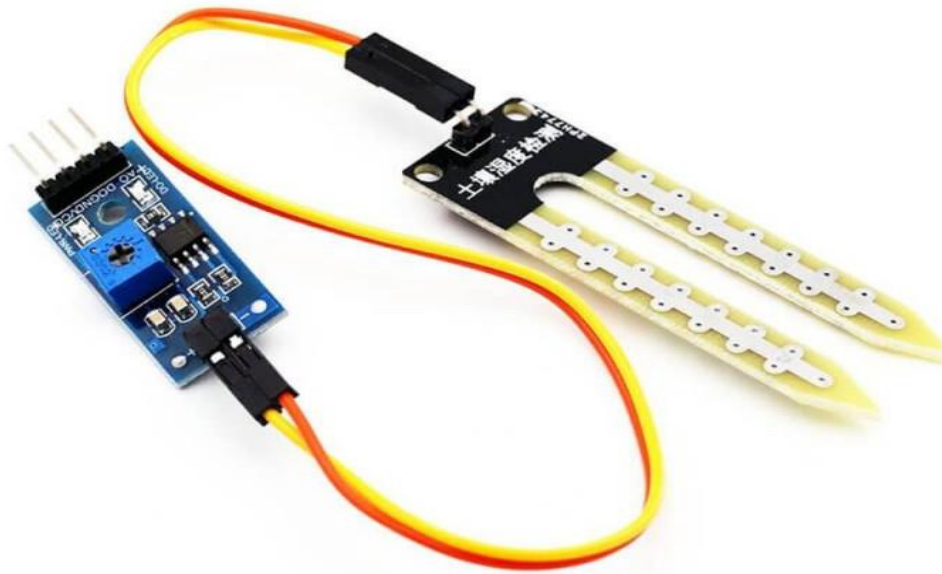
- Ventajas y limitaciones:

Ventajas:

- ✓ Permite mediciones con poca alteración del suelo.
- ✓ Requiere poco mantenimiento.
- ✓ Las mediciones se pueden realizar de manera continua durante varios ciclos del cultivo.
- ✓ No requiere instalaciones especiales.

Limitaciones:

- Para una buena precisión se requiere varias mediciones debido a la variabilidad de los suelos.
- Alto costo de los equipos.



3.3. Sensor de temperatura

Un sensor de temperatura convierte directamente la temperatura en corriente eléctrica (mv), que son luego acondicionados o modificados por equipos eléctricos o electrónicos.

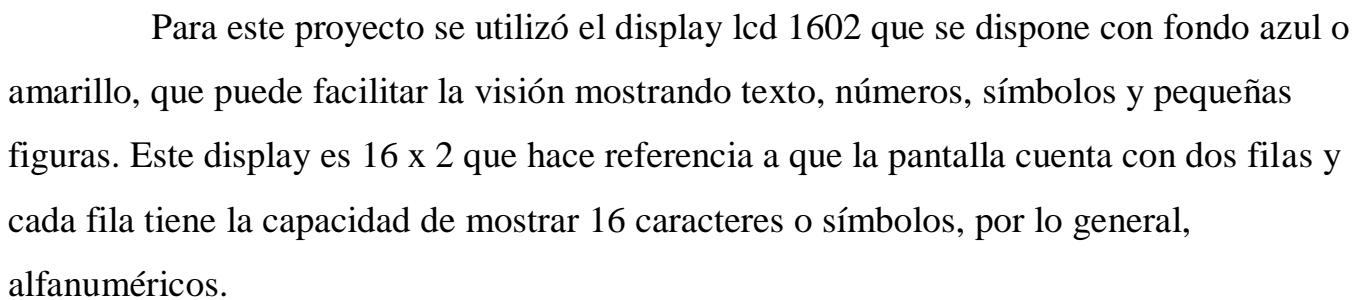
Qué tipo de sensor de temperatura se utilizó para este proyecto:

Para este proyecto, utilizamos el TMP36, que es un sensor de temperatura que en su salida nos proporciona una lectura en grados centígrados de precisión y de bajo voltaje. Trabaja en un rango de temperatura de -40°C a $+125^{\circ}\text{C}$, y también con un voltaje de funcionamiento de 2.7 V a 5.5 V.

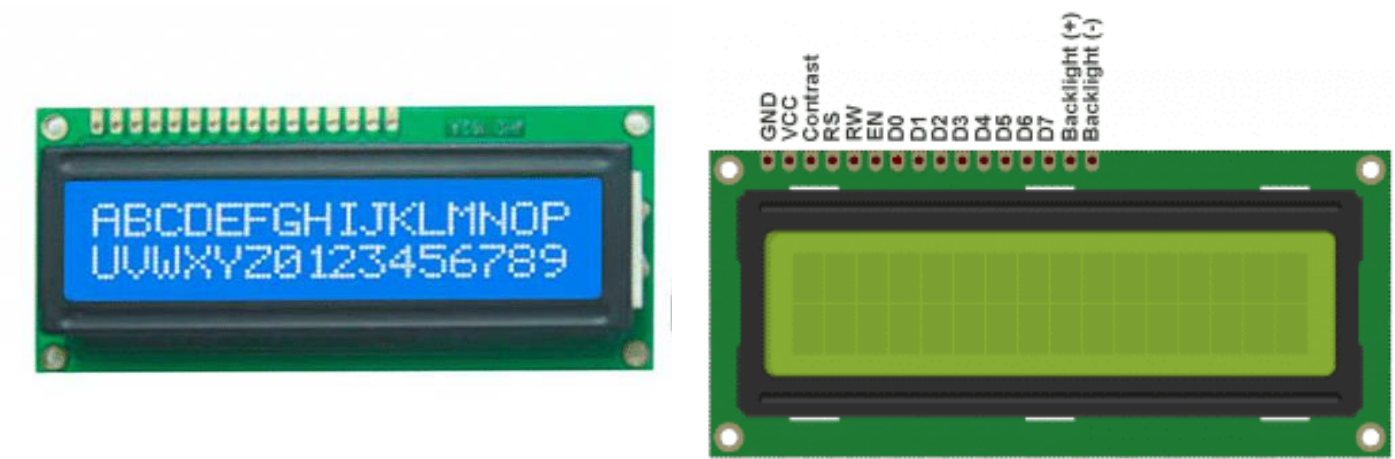
Este sensor se utiliza para medir la temperatura de los sistemas de control ambiental, protección térmica, control de procesos industriales, alarmas contra incendios, gestión térmica de la CPU, etc.

ESPECIFICACIONES Y CARACTERÍSTICAS:

-



Para llegar a conectarlos a la placa Arduino UNO necesitaremos una resistencia de 220 Ω o un potenciómetro de 10 K Ω .



3.Ejecución

4.1 Diagrama de Gantt

Tiempo total trabajado:

ACTIVIDADES	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
Capacitación en componentes					
Capacitación en programacion					
Construcción del proyecto					
Elecciones de grupos					
Exposición del proyecto					
Informe final					
Programación de arduino					

4.2. Imágenes del proyecto

5.Programa – (Scroll)

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(10,11,12,9,8,7,6);

int motor1=5;

int motor2=3;

float temp=0;

int LedRojo=2;

void setup()

{

pinMode(motor1,OUTPUT);

pinMode(motor2,OUTPUT);

Serial.begin(9600);

lcd.begin(16,2);

}

void loop()

{

int humedad=analogRead(A0);

float valor=analogRead(A1);

float ldr = analogRead(A2);

temp=(((5valor100)/1024)-50);

Serial.println("La humedad es:");

Serial.println(humedad);

Serial.println("La temperatura es:");

Serial.println(temp);

Serial.println("LUZ:");

Serial.println(ldr);

if (humedad>700)

{

digitalWrite(motor1,HIGH);
```

```
}else if(humedad<700)
{
digitalWrite(motor1,LOW);
}
if (temp>30)
{
digitalWrite(motor2,HIGH);}else if(temp<30)
{
digitalWrite(motor2,LOW);
}
if(ldr <700)
{
digitalWrite(LedRojo,LOW);
}else if (ldr>700)
{
digitalWrite(LedRojo,HIGH);
}
lcd.setCursor(0, 0);
lcd.print("Temp:");
lcd.print(temp);
lcd.print(" C |");
lcd.print(" Luz:");
lcd.print(ldr);
lcd.setCursor(1, 1);
lcd.print("Hum: ");
lcd.print(humedad);
if(humedad<200){
lcd.print(" |Muy seco");}
else if(humedad>200 && humedad < 400){
```

```
lcd.print(" |Seco");}  
  
else if(humedad>400 && humedad < 600){  
    lcd.print(" |Humedo");}  
  
else if(humedad>600){  
    lcd.print(" |Muy humedo");}  
  
lcd.scrollDisplayLeft();  
delay(250);
```

5.Programa – (Clear)

```
#include <LiquidCrystal.h>  
  
LiquidCrystal lcd(10,11,12,9,8,7,6);  
  
int motor1=5;  
int motor2=3;  
float temp=0;  
int LedRojo=2;  
  
void setup()  
{  
    pinMode(motor1,OUTPUT);  
    pinMode(motor2,OUTPUT);  
    Serial.begin(9600);  
    lcd.begin(16,2);  
    lcd.clear();  
}  
  
void loop()  
{
```

```
int humedad=analogRead(A0);
```

```
float valor=analogRead(A1);
```

```
float ldr = analogRead(A2);
```

```
temp=(((5valor100)/1024)-50);
```

```
Serial.println("La humedad es:");
```

```
Serial.println(humedad);
```

```
Serial.println("La temperatura es:");
```

```
Serial.println(temp);
```

```
Serial.println("LUZ:");
```

```
Serial.println(ldr);
```

```
if (humedad<=700)
```

```
{
```

```
digitalWrite(motor1,HIGH);
```

```
}else if(humedad>700)
```

```
{
```

```
digitalWrite(motor1,LOW);
```

```
}
```

```
if (temp>=30)
```

```
{
```

```
digitalWrite(motor2,HIGH);
```

```
}else if(temp<30)
```

```
{
```

```
digitalWrite(motor2,LOW);
```

```
}
```

```
if(ldr <=700)
{
digitalWrite(LedRojo,LOW);

}else if (ldr>700)
{
digitalWrite(LedRojo,HIGH);
}
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("Temp:");
lcd.print(temp);
lcd.print(" C");
```

```
lcd.setCursor(0, 1);
lcd.print("Luz:");
lcd.print(ldr);
```

```
delay(2500);
lcd.clear();
delay(500);
```

```
lcd.setCursor(0, 0);
lcd.print("Hum: ");
lcd.print(humedad);
```

```
lcd.setCursor(0,1);
```



```
if(humedad<=200){  
  lcd.print("Muy seco");  
}  
else if(humedad>200 && humedad <= 400){  
  lcd.print("Seco");  
}  
else if(humedad>400 && humedad <= 600){  
  lcd.print("Humedo");  
}  
else if(humedad>600){  
  lcd.print("Muy humedo");  
}  
  
delay(1000);  
lcd.clear();  
delay(500);  
  
}
```

Enviar mensaje a #bienvenida-y-reglas

6. Información de la placa principal Arduino UNO

- Arduino es una placa con un microcontrolador de la marca Atmel y con toda la circuitería de soporte, que incluye, reguladores de tensión, un puerto USB (En los últimos modelos, aunque el original utilizaba un puerto serie) conectado a un módulo adaptador USB-Serie que permite programar el microcontrolador desde

cualquier PC de manera cómoda y también permite hacer pruebas de comunicación con el propio chip.

¿De que dispone Arduino UNO?

- 14 pines configurables como entrada y salida y a los que puede conectarse cualquier dispositivo que sea capaz de transmitir o recibir señales digitales de 0 y 5 V.
- Dispone de entradas y salidas analógicas; Mediante las entradas analógicas podemos obtener datos de sensores en forma de variaciones continuas de un voltaje. Las salidas analógicas suelen utilizarse para enviar señales de control en forma de señales PWM.
- Cada uno de los 14 pines digitales se puede usar como entrada o como salida. Funcionan a 5 V, cada pin puede suministrar hasta 40 mA.

La intensidad máxima de entrada también es de 40 mA. Cada uno de los pines digitales dispone de una resistencia de pull-up interna de entre $20\text{K}\Omega$ y $50\text{K}\Omega$ que está desconectada, salvo que nosotros indiquemos lo contrario. Arduino también dispone de 6 pines de entrada analógicos que trasladan las señales a un conversor analógico/digital de 10 bits.

Pines especiales de entrada y salida:

- RX y TX: Se usan para transmisiones serie de señales TTL.
- Interrupciones externas: Los pines 2 y 3 están configurados para generar una interrupción en el atmega. Las interrupciones pueden dispararse cuando se encuentra un valor bajo en estas entradas y con flancos de subida o bajada de la entrada.
- PWM: Arduino dispone de 6 salidas destinadas a la generación de señales pwm de hasta 8 bits.
- SPI: Los pines 10, 11, 12 y 13 pueden utilizarse para llevar a cabo comunicaciones SPI, que permiten trasladar información full dúplex en un entorno Maestro/Esclavo.
- I²C: Permite establecer comunicaciones a través de un bus I²C. El bus I²C es un producto de Phillips para interconexión de sistemas embebidos.

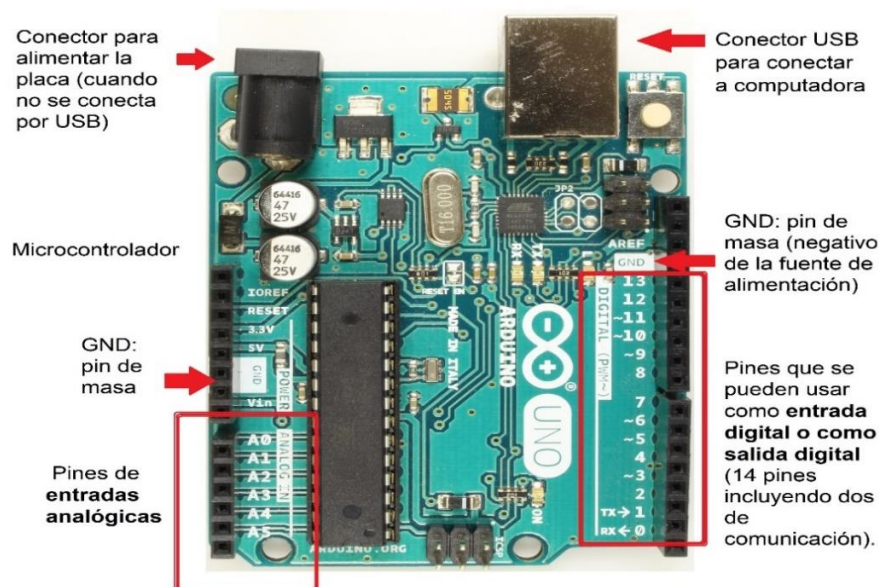
Actualmente se puede encontrar una gran diversidad de dispositivos que utilizan esta interfaz, desde pantallas LCD, memorias EEPROM, sensores, etc.

Como se puede alimentar un Arduino UNO

Arduino UNO puede alimentarse directamente a través del propio cable USB o mediante una fuente de alimentación externa, como puede ser un transformador de tamaño reducido, por ejemplo, una pila de 9V. Los límites están entre los 6 y los 12 V. Como única restricción hay que saber que, si la placa se alimenta con menos de 7V, la salida del regulador de tensión a 5V puede dar menos que este voltaje y si sobrepasamos los 12V, probablemente dañaremos la placa.

La alimentación puede conectarse mediante un conector de 2,1mm con el positivo en el centro o directamente a los pines Vin y GND marcados sobre la placa.

Hay que tener en cuenta que podemos medir el voltaje presente en el Jack directamente desde Vin (voltaje de entrada). En el caso de que el Arduino esté siendo alimentado mediante el cable USB, ese voltaje no podrá monitorizarse desde aquí.





Notebook corriendo
Arduino IDE para poder
programar Arduino



Cable de conexión, para
transferir el programa y
para alimentar
momentáneamente la
placa Arduino



Microcontrolador, que
adentro tiene memoria
FLASH que guarda el
programa

7. Conclusión:

Nuestra conclusión se basa que nos pareció un buen desafío a la hora de realizarlo, con todo el trabajo que realizamos pudimos obtener conocimientos de los diferentes programas que nos brinda la materia (Arduino, Tinkercad), al momento de utilizarlos como herramientas de producción y diseño. Sin dejar de lado todo el proceso de programar un código para después explayarlo en un programa.