

Introducción a la Programación



Secuencias de Trabajo para Taller de Ciclo Básico

Nº 6

Relé - Relay. Y algo más...

Índice:

Pagina 1: Introducción: Organización de la secuencia - Algunos Conceptos: Corriente Eléctrica. Magnetismo. Electromagnetismo. Relé o Relay.

Pagina 10: Actividad Explicativa: Realización del circuito y la programación. Situación Problema: **“Sistema de Iluminación Automática con Detección de Movimiento”**

Pagina 12: Actividad Explicativa: Realización del circuito y la programación. Situación Problema: **“Optimización del Sistema de Iluminación Automática con Detección de Movimiento”**

Pagina 16: Problemática circuital y de programación. (Pensamiento Computacional - Método de Pólya) Situación Problema: **“Control de temperatura ambiental para un Invernadero Automatizado”**

Pagina 21: Problemática circuital y de programación. (Pensamiento Computacional - Método de Pólya) Situación Problema: **“Control de Temperatura en un Invernadero con visualización en display LCD”**

Pagina 28: Problemática circuital y de programación. (Pensamiento Computacional - Método de Pólya) Situación Problema: **Sistema de Riego Automático Inteligente para Plantas de Interior”**

Pagina 35: Problemática circuital y de programación. (Pensamiento Computacional - Método de Pólya) Situación Problema: **“Control Inteligente de Iluminación para Habitaciones Eficientes”**

Pagina 42: Situaciones Problemáticas Tecnológicas. “Problemáticas Abiertas” Metodología Proyectual. **“Proyecto Tecnológico - Método Proyectual”**

Pagina 47: Situación Problema Tecnológica. “Problemática Abierta” (Metodología Proyectual) Título de la problemática: **“Invernadero automatizado”**

Pagina 61: Situación Problema Tecnológica. “Problemática Abierta” (Metodología Proyectual) Título de la Problemática: **“Casa Inteligente con Automatización y Control”**

Pagina 75: El proyecto tecnológico y la evaluación de capacidades profesionales básicas: Capacidades Básicas. Capacidades Profesionales Básicas. La evaluación de capacidades profesionales básicas en el proyecto tecnológico.

Secuencias de Trabajo para Taller de Ciclo Básico

Secuencia de Trabajo Nº 6 (Para estudiantes de primer ciclo de ETP)

“Relay o Relé”

Esta secuencia de trabajo estará organizada con las siguientes actividades:

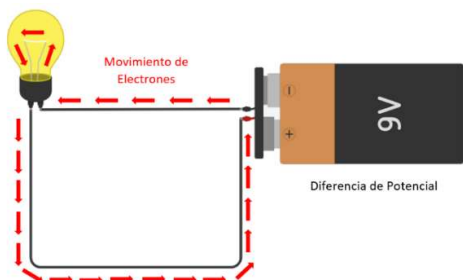
1. **Explicativa:** Realización del circuito y la programación.
2. **Explicativa:** Realización del circuito y la programación.
3. **Problemática circuital y de programación.** (Pensamiento Computacional - Método de Pólya)
4. **Problemática circuital y de programación.** (Pensamiento Computacional - Método de Pólya)
5. **Problemática circuital y de programación.** (Pensamiento Computacional - Método de Pólya)
6. **Problemática circuital y de programación.** (Pensamiento Computacional - Método de Pólya)
7. **Situación Problema Tecnológica. “Problemática Abierta”** (Metodología Proyectual). *
8. **Situación Problema Tecnológica. “Problemática Abierta”** (Metodología Proyectual). *

*En estas dos últimas actividades se pretende que las y los estudiantes de taller del primer ciclo (Ciclo Básico), resuelvan una problemática articulando lo distintos saberes desarrollados durante su formación, a través de la metodología proyectual y que este proceso de como resultado la fabricación de un prototipo maqueteizado y automatizado.

Comenzamos con algunos conceptos:

Corriente Eléctrica

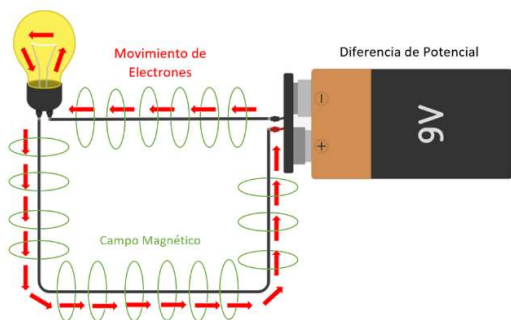
La corriente eléctrica es el flujo de electrones a través de un conductor. En un circuito eléctrico, la corriente se origina cuando hay una diferencia de potencial (voltaje) entre dos puntos, lo que crea un campo eléctrico que impulsa a los electrones a moverse. La unidad de medida de la corriente eléctrica es el amperio (A).



La corriente eléctrica es el flujo ordenado de partículas cargadas eléctricamente, generalmente electrones, a través de un conductor. Estos electrones son partículas subatómicas con carga negativa que orbitan alrededor del núcleo de los átomos. La corriente eléctrica es el movimiento de estos electrones a lo largo de un circuito, lo que genera un flujo de carga eléctrica.

Conceptos Básicos de Electromagnetismo

El electromagnetismo es una rama de la física que estudia la relación entre electricidad y magnetismo. Una corriente eléctrica que fluye a través de un conductor crea un campo magnético alrededor del conductor. La intensidad de este campo magnético depende de la cantidad de corriente que fluye y de la distancia al conductor.



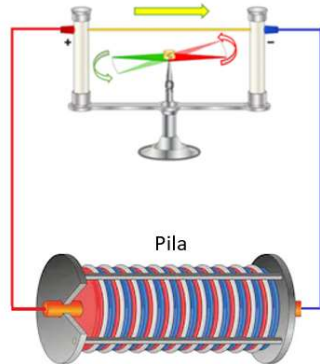
Cuando una corriente eléctrica fluye a través de un conductor, se genera un fenómeno conocido como "efecto magnético de la corriente" o "campo magnético de la corriente". Este fenómeno se basa en los principios del electromagnetismo y fue descubierto por el científico danés Hans Christian Oersted en 1820. La corriente eléctrica es un flujo ordenado de electrones con carga negativa. Debido a su movimiento, estos electrones generan un campo magnético a su alrededor. El campo magnético se forma en círculos concéntricos alrededor del conductor, como si fuera un patrón de anillos concéntricos que se expanden desde el conductor.

Experiencia de Hans Christian Oersted

En 1820, Hans Christian Oersted, un físico y químico danés, estaba impartiendo una conferencia sobre electricidad en la Universidad de Copenhague. Durante su conferencia, estaba realizando una demostración con un circuito eléctrico simple que consistía en una pila, un cable conductor y una aguja magnética (brújula) colocada cerca del cable.

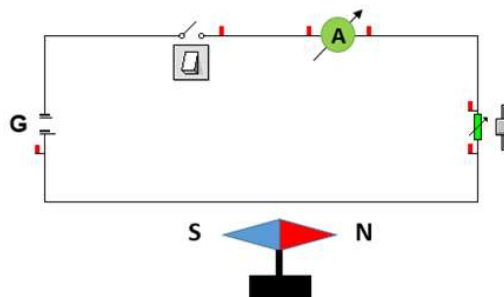
Mientras Oersted pasaba corriente eléctrica a través del cable, notó que la aguja magnética se desviaba de su dirección habitual y se alineaba perpendicularmente al cable. Este fenómeno fue sorprendente, ya que hasta ese momento no se había demostrado ninguna conexión entre la electricidad y el magnetismo. Oersted había descubierto que una corriente eléctrica en movimiento generaba un campo magnético a su alrededor, lo que causaba la deflexión de la aguja magnética.

Cuando Oersted conectó los extremos del cable a los terminales de una batería, la aguja giró hasta adoptar una orientación este-oeste, casi perpendicular a los cables. Las cargas en reposo no afectan a un imán, pero aquí estaba la prueba de que las cargas en movimiento (la corriente) ejercen realmente una extraña fuerza sobre un imán.

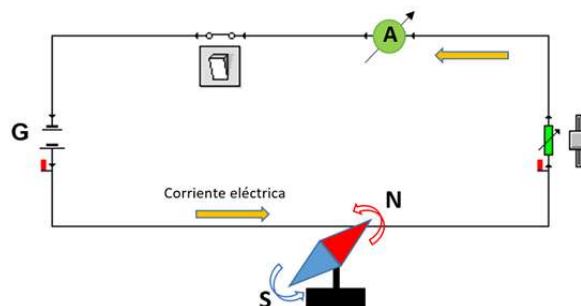


La experiencia de Oersted básicamente consiste en lo siguiente:

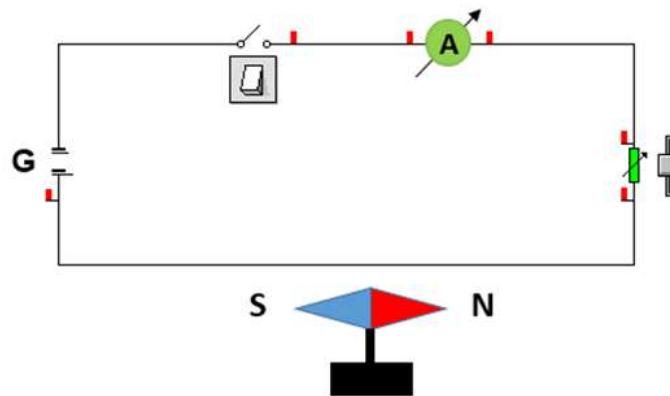
Se arma un circuito eléctrico formado por un generador, un conductor metálico, un interruptor, un amperímetro y un reóstato. Una parte del conductor, dispuesta en línea recta, se sitúa por encima de una brújula y paralela al eje de la aguja magnética que señala la dirección N-S:



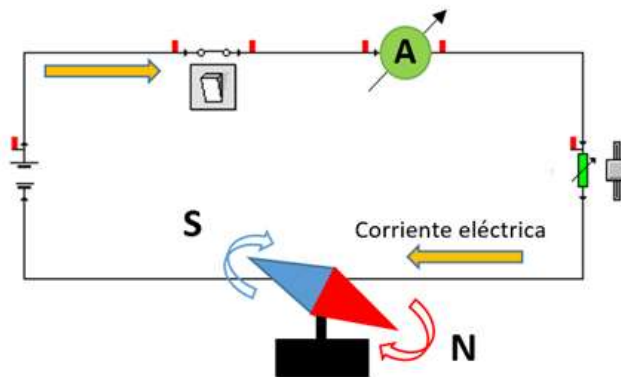
Cuando se acciona el interruptor para que pase la corriente eléctrica, la aguja imantada de la brújula se desvía y tiende a colocarse perpendicular a la dirección del conductor, acercándose tanto más a esa posición cuanto mayor sea la intensidad de la corriente, la que se regula con el reóstato y se mide con el amperímetro:



Si se abre el circuito, de modo que cese la corriente, la aguja magnética vuelve a su posición primitiva.



Cuando se cambia la conexión en los bornes del generador para que la corriente circula en sentido contrario, se invierte el giro que sufre la aguja imantada de la brújula:



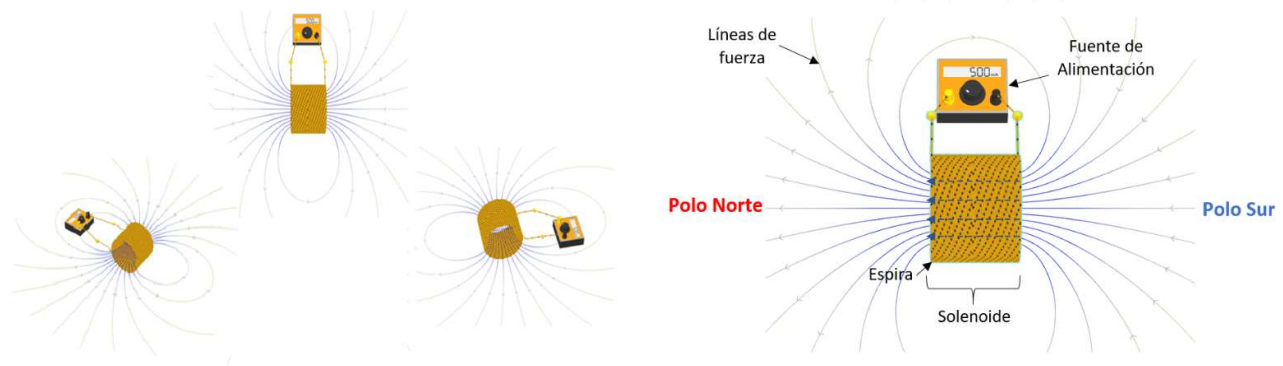
En consecuencia, se puede establecer que: En la región próxima a un conductor por el que pasa una corriente eléctrica existe un campo magnético. El sentido del campo magnético depende del sentido que tiene la corriente en el conductor. Es decir, la aguja dejaba de alinearse con el campo magnético de la tierra al estar cerca del conductor de corriente eléctrica. Esto solamente podría explicarse si la corriente que pasaba a través del conductor creaba su propio campo magnético: finalmente se confirmaba que la electricidad y el magnetismo eran dos fenómenos altamente relacionados.

Campo magnético generado por un solenoide

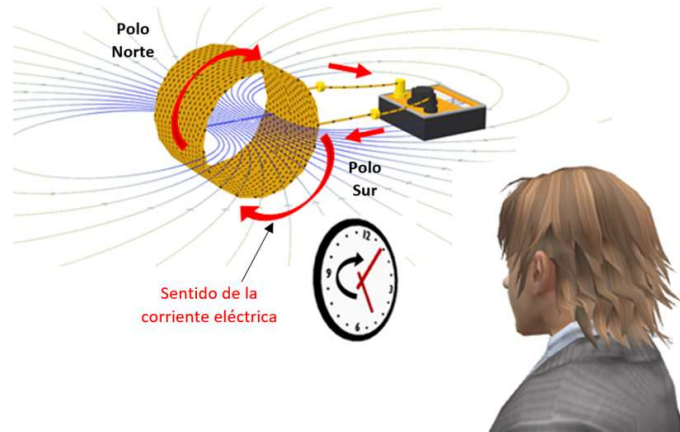
El solenoide es una bobina constituida por varias espiras de alambre arrolladas en forma cilíndrica y aislada entre sí:



Cada espira puede considerarse como un conductor circular equivalente a un pequeño imán. Como estos pequeños imanes están colocados uno a continuación del otro, forman entre todos un imán de grandes dimensiones, con sus correspondientes polos norte y sur. En consecuencia, el campo magnético de un solenoide está dado por los campos creados por cada una de las espiras:

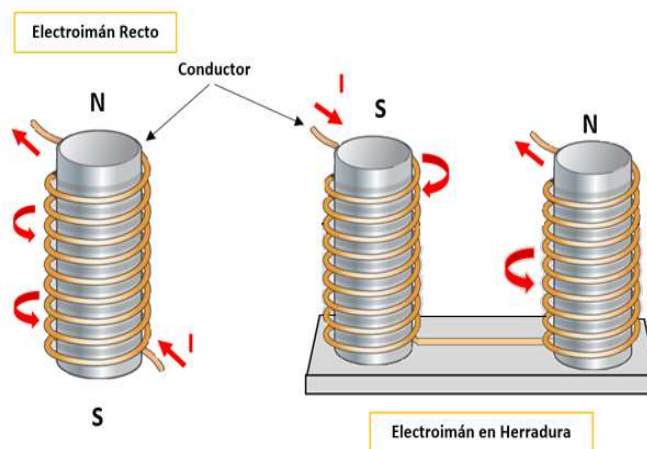


El campo magnético generado por un solenoide es similar al que crea un imán con forma de barra. Para determinar a qué polo magnético equivale una cara del solenoide, se tiene en cuenta lo siguiente: si frente a esa cara se observa que la corriente gira en el sentido de las agujas del reloj, se está delante del polo sur y la cara opuesta es el polo norte (regla de las agujas del reloj):



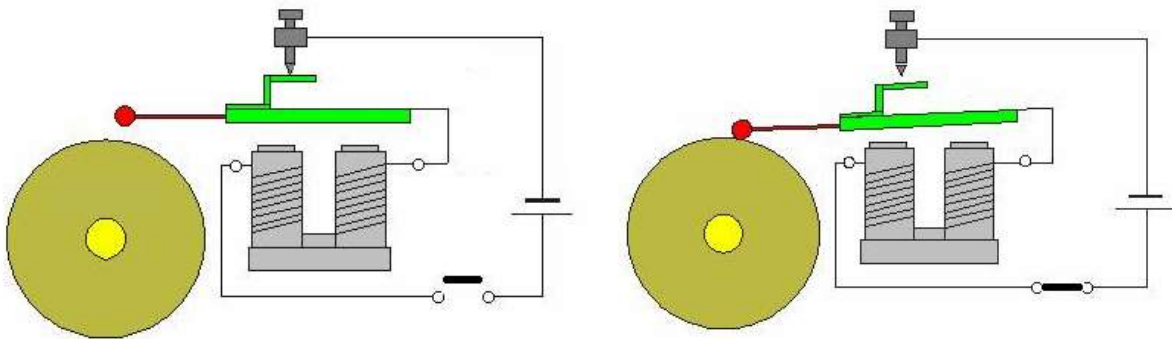
Electroimán

Si se introduce en un solenoide una barra de hierro (núcleo de hierro), ésta se magnetiza, constituyendo un electroimán:



El electroimán es básicamente un arrollamiento de alambre por el que se hace circular corriente eléctrica. Alrededor de esa bobina de alambre se va a generar un campo magnético cuya intensidad será proporcional a la corriente que circule y a la cantidad de vueltas que tenga ese bobinado. Generalmente se pone dentro de esa bobina de alambre un núcleo de hierro laminado que sirve para “orientar” a ese campo magnético en la dirección que lo necesitemos. Imaginemos una grúa en un depósito de chatarra: la grúa tiene en su extremo un electroimán en lugar de un gancho. Cuando se posa sobre un trozo de chatarra el operador activa el imán y puede levantarlo con facilidad. Al desactivar ese imán puede soltar la carga. Así de simple y así de útil.

El timbre de una casa, el motor eléctrico, esas puertas de los negocios en los que nos abren desde adentro por medio de un pulsador, las alarmas... infinidad de dispositivos eléctricos, electrónicos y electromecánicos funcionan gracias a los electroimanes. Vayamos a conocer juntos un tipo especial de implementación de los mismos: el relé o relay.

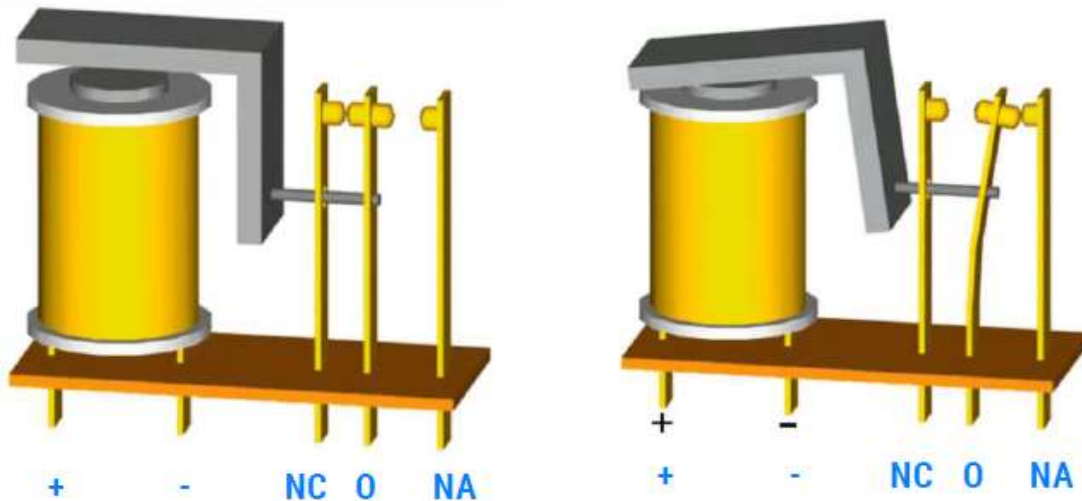


El relé o relay

Uno de los dispositivos principales de la automatización. Tiene muchas variantes y medidas, pero el funcionamiento es prácticamente el mismo.

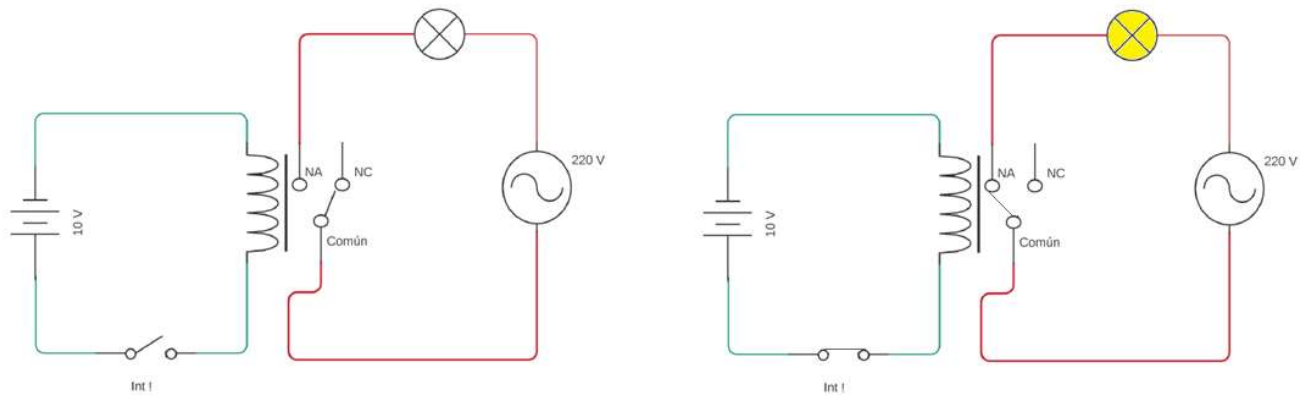
En la figura vemos un electroimán que se alimenta desde los terminales indicados con + y -. Hay una pieza metálica gris en forma de "L" y tres contactos NA, O y NC:

Vemos que al estar en reposo el contacto "O" (Común) está tocando con el NC. Es decir que hay conectividad entre ellos. Por eso se llama contacto NC, porque está "normalmente cerrado". Se toma la condición sin corriente circulando. Al pasar corriente por los bornes + y - la cosa cambia: la bobina del electroimán atrae a la pieza en forma de "L" y el común ahora hace contacto con el normalmente abierto (NA) y deja de hacerlo con el NC.



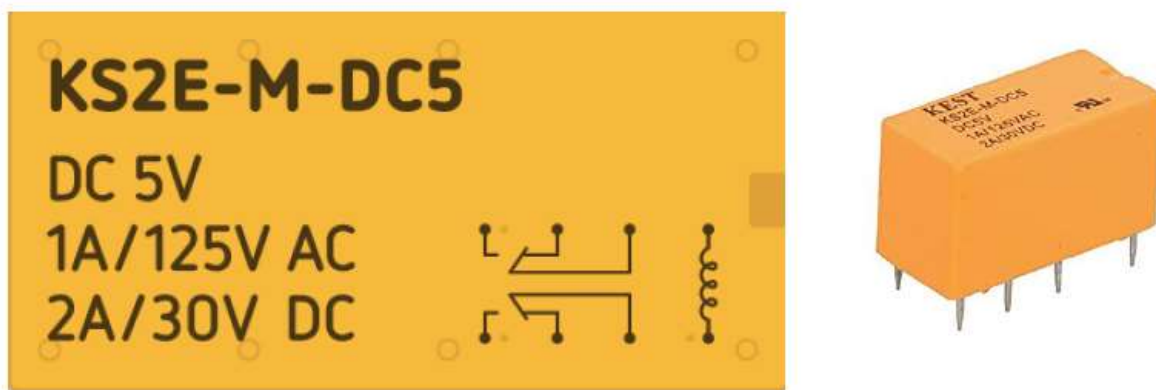
Podemos separar un circuito de automatización en dos partes: Comando y Potencia. Esto nos brinda seguridad, claridad y prolijidad en nuestros montajes. Lo explicaremos con símbolos:

Para más claridad se han usado colores diferentes: la parte verde es simplemente una fuente de baja tensión (10 V, inofensiva) que, a través del Int1, alimenta a la bobina de un relé. A su vez los contactos del relé (ya en la parte roja) conectan o desconectan una lámpara a 220 V (peligrosa). Si se cierra el Int1 el relé conectará el NA y la lámpara se encenderá.



La parte de Potencia (la roja) está físicamente separada de la de Comando (verde). Las personas que operen ese circuito van a trabajar siempre del lado verde del mismo, nunca van a operar sobre tensión peligrosa. Y cualquier problema que surgiera en el circuito de potencia será aislado por el relé que lo mantendrá desvinculado del de comando.

A continuación realizaremos la conexión de nuestro primer relé a través de nuestro protoboard. Tenemos que tener en cuenta que el relé comercial tiene el aspecto de un cubito de plástico opaco con unos terminales que salen del mismo. Tenemos que identificar en cada modelo de relé qué terminales corresponden a la bobina y cuáles a los contactos NA y NC. Tanto en el caso del relé virtual de nuestro simulador como en un relé real, vamos a usar el modelo KS2E-M_DC5.

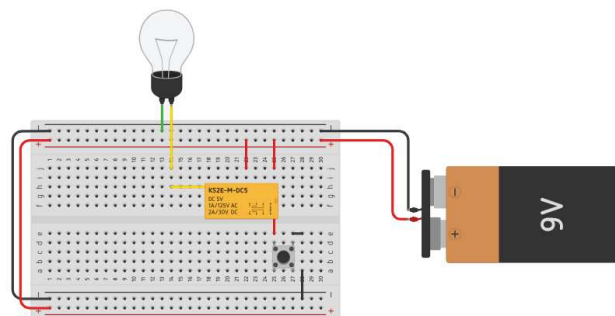


Veamos detenidamente el esquema que tiene impreso sobre la cara superior. Los dos contactos de la derecha son los de la bobina. Por ahí no tenemos problema. Ahora viene la novedad: la mayoría de los relés tienen dos juegos de NA / NC. Aprovechando el espacio proveen dos contactos en lugar de uno. Por lo tanto en el sector superior se conecta uno y en el inferior, otro. Esto se llama “relé doble inversor” y son totalmente independientes eléctricamente pero se mueven al unísono.

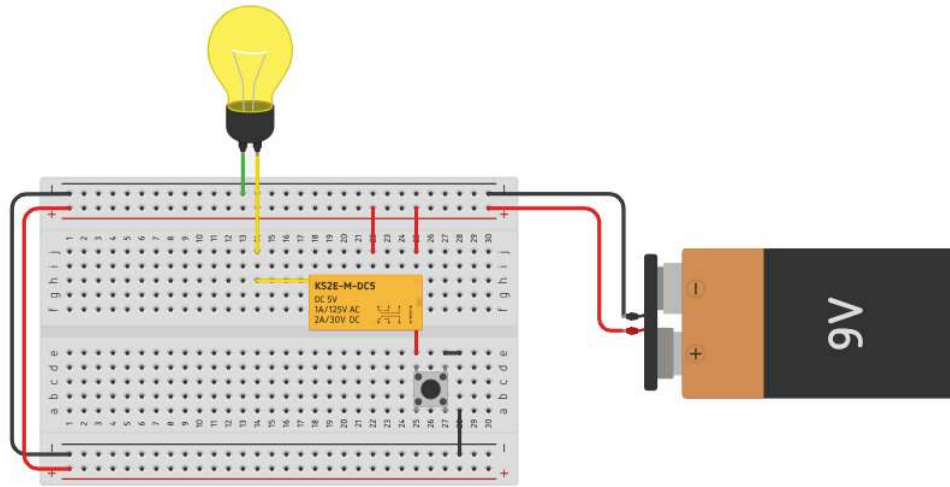
Ahora veamos nuestra conexión:

Para nuestra experiencia utilizaremos:

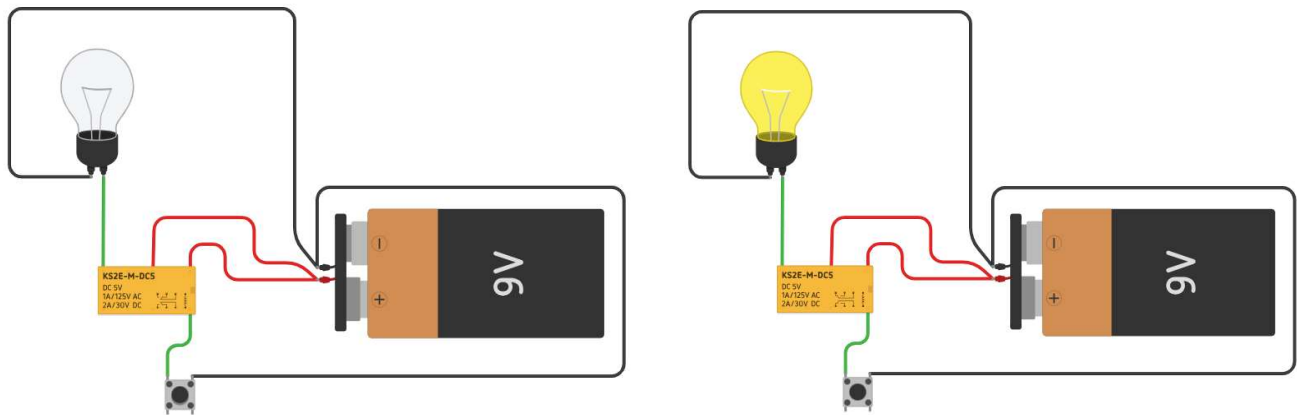
- 1 Bateria de 9 volt
- 1 Pulsador
- 1 Lámpara de 9 volt
- 1 Relé
- 1 Protoboard
- Cables de conexión



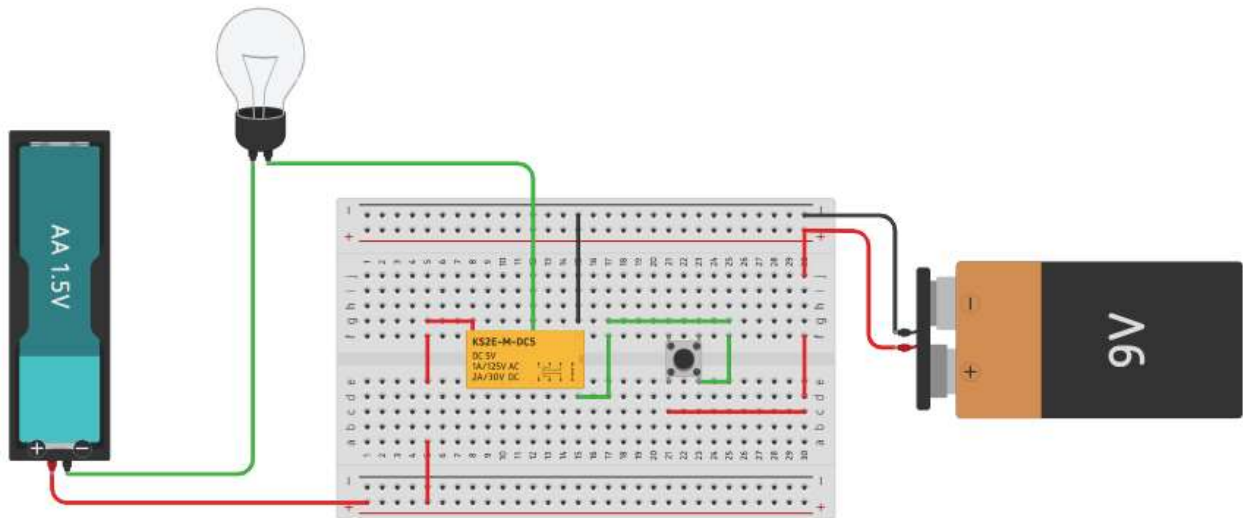
Al accionar el interruptor la corriente circula y al pasar por la bobina se genera un campo magnético y se acciona el relé y la lámpara se enciende. Si bien en este esquema los circuitos de comando y potencia poseen la misma fuente de alimentación, nos sirve para comprender el conexionado.



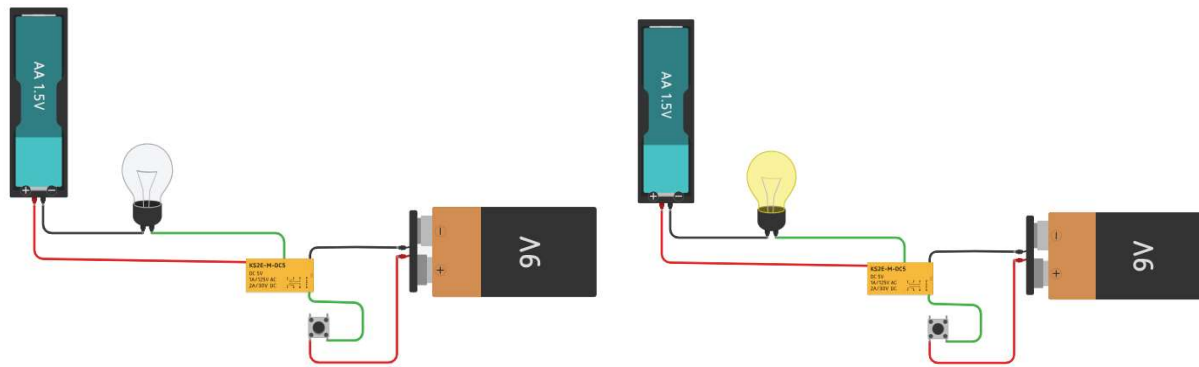
Veamos el conexionado sin el protoboard para entender mejor el circuitado.



Ahora utilizaremos circuitos de comando y potencia con distinta fuente de alimentación, para poder entender mejor la función del relé.

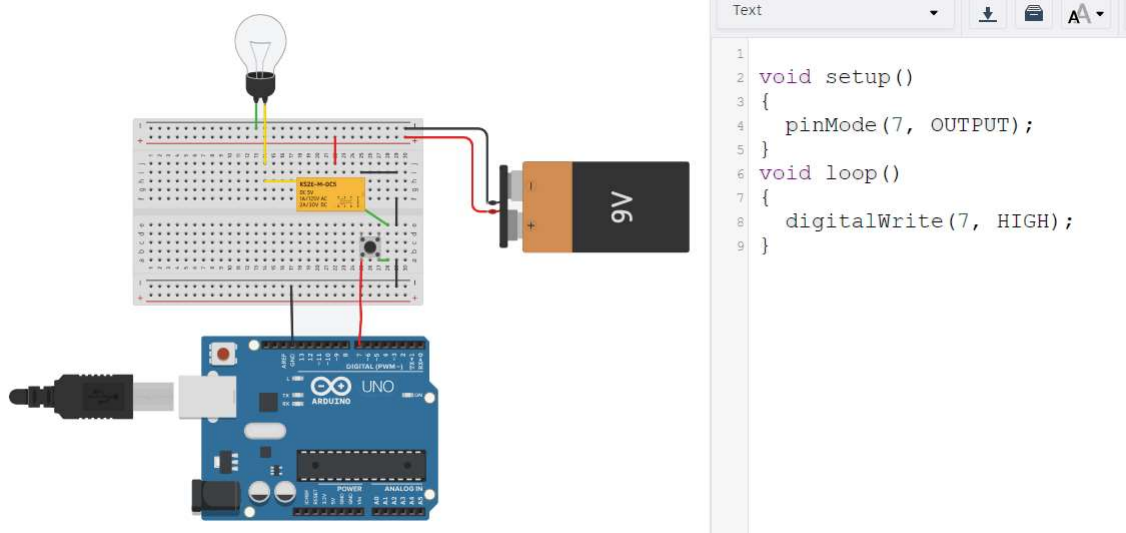


Veamos el conexionado sin el protoboard para entender mejor el circuitado:

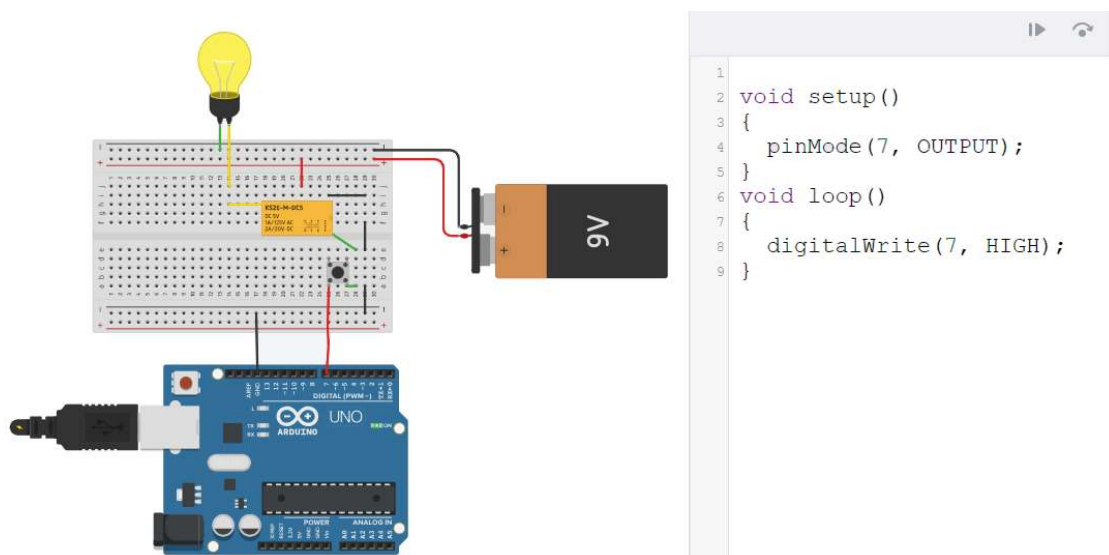


Ahora veamos como funcionaria el sistema con un Arduino:

Realizamos la conexión y la programación donde en los parámetros iniciales definimos el pin 7 como una salida la cual tendrá un valor alto.

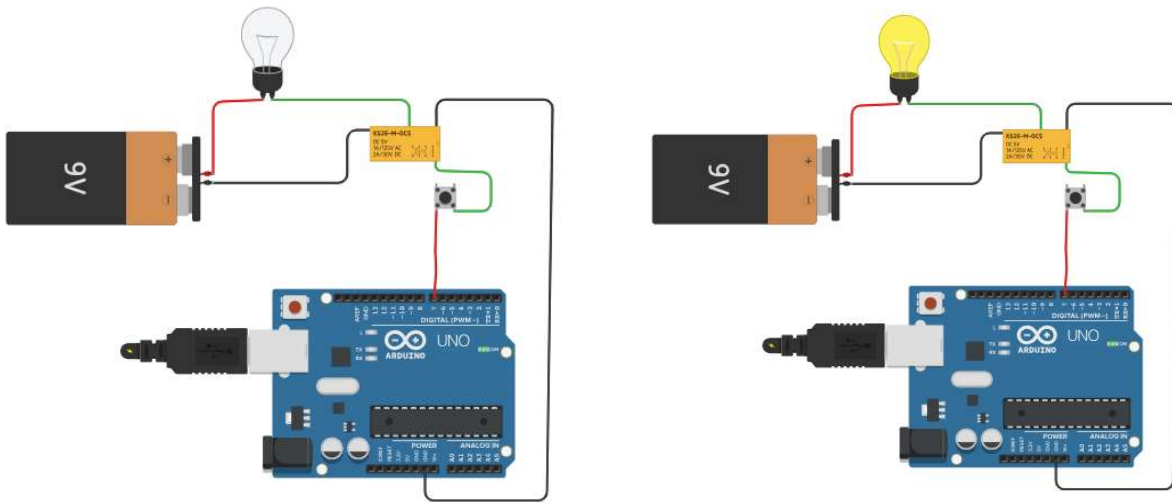


Al accionar el pulsador se generara un campo magnético con la corriente que circulara del pin 7 y se activara el relé conmutando y provocando el encendido de la lámpara.



Veamos el circuito sin el protoboard para comprenderlo el conexionado mejor:

En este conexionado se puede observar como la batería está conectada a la lámpara a través del normar abierto del relé, el cual se cierra cuando por la bobina circula una corriente eléctrica a través de la salida digital definida (7) en la programación.



1. Actividad Explicativa: Realización del circuito y la programación.

Situación Problema: “Sistema de Iluminación Automática con Detección de Movimiento”

A un grupo de estudiantes de una escuela técnica se les ha encomendado diseñar y desarrollar un sistema de iluminación automática para una maqueta de un hogar inteligente. El objetivo es crear un sistema que encienda automáticamente una lámpara cuando se detecte movimiento en una habitación, lo que brindará seguridad y comodidad.

Requisitos del Proyecto:

- **Circuito:** Diseñar un circuito que conste de un Arduino, un relé, una lámpara, una batería de 9 voltios y un sensor PIR (Detector de Movimiento por Infrarrojos).
- **Sensor PIR:** Utilizar el sensor PIR para detectar el movimiento en la habitación. El sensor debe ser capaz de captar los cambios en la radiación infrarroja causados por la presencia de una persona.
- **Programación:** Programar el Arduino de manera que, cuando el sensor PIR detecte movimiento, el relé se active y encienda la lámpara. Cuando no se detecte movimiento, el sistema debe apagar la lámpara.

Desafíos para los Estudiantes:

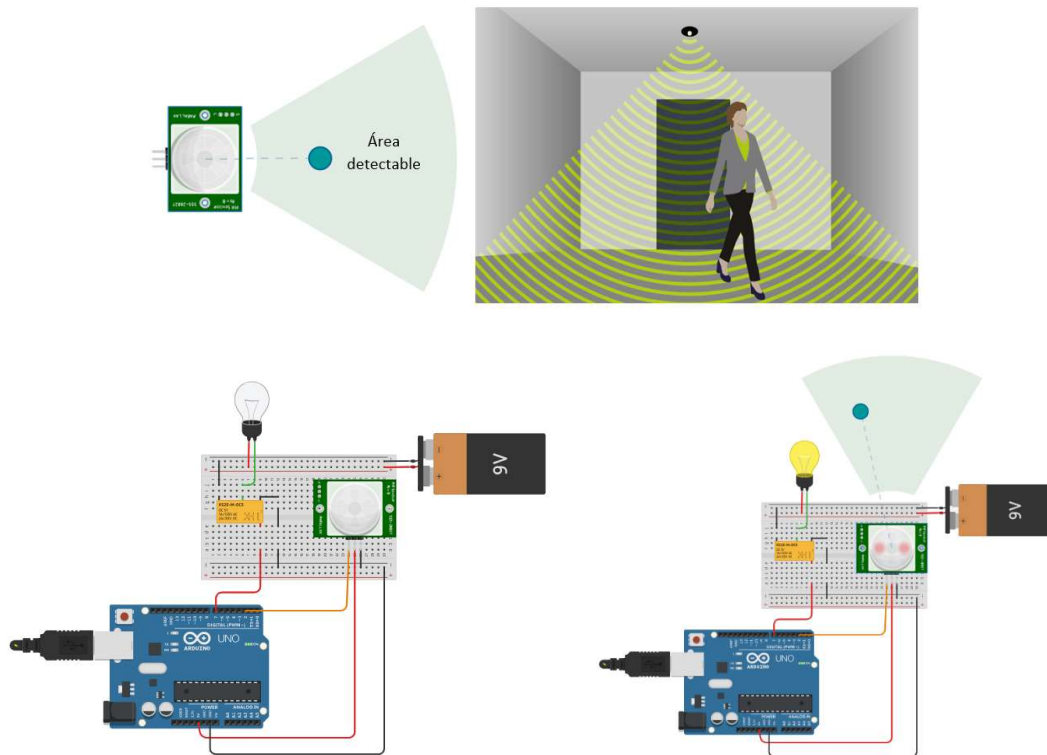
- **Conexiones de Componentes:** Aprender a conectar adecuadamente el sensor PIR, el relé, la lámpara y la batería al Arduino, asegurándose de que los componentes estén conectados correctamente y de acuerdo con sus especificaciones.
- **Programación del Arduino:** Comprender cómo leer la señal del sensor PIR en el Arduino y utilizar esta información para controlar el estado del relé y, por lo tanto, de la lámpara.
- **Optimización:** Ajustar el sistema para evitar falsos positivos (encendido de la lámpara sin movimiento real) y falsos negativos (fallo en detectar movimiento real).

Resultado Esperado:

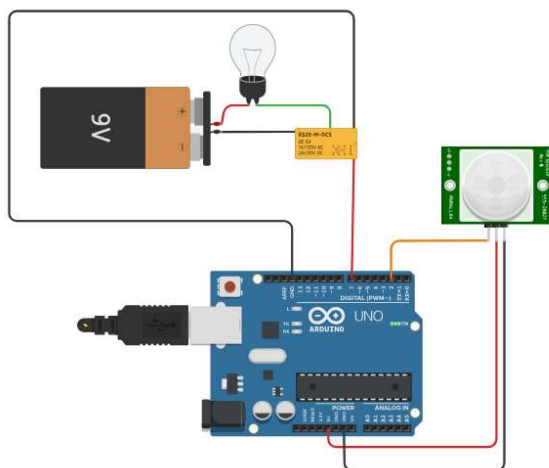
Al finalizar este proyecto, se espera que los estudiantes hayan diseñado y construido un sistema de iluminación automática que se active cuando se detecta movimiento en la habitación. La lámpara debe encenderse automáticamente en respuesta al movimiento y apagarse cuando no haya movimiento detectado. El sistema debe ser confiable y eficiente en términos de consumo de energía.

Cuestiones a tener en cuenta:

Los sensores infrarrojos pasivos (PIR) son dispositivos para la detección de movimiento. Se basan en la medición de la radiación infrarroja. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piro eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica.

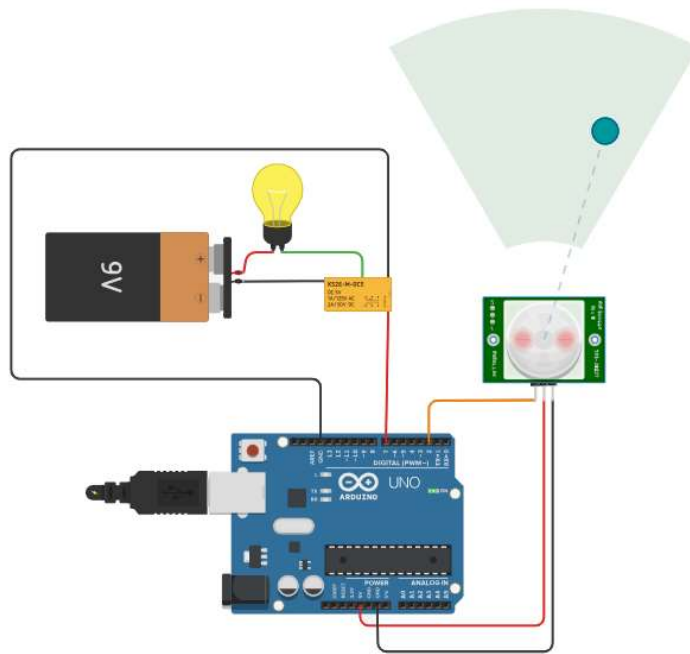


Veamos el circuito sin el protoboard para comprenderlo el conexionado mejor y analizar la programación:



```
1 void setup()//Escribo los parametros iniciales
2 {
3   pinMode(7, OUTPUT);//El pin 7 sera una salida
4   pinMode(2, INPUT);//El pin 2 sera una entrada
5 }
6 void loop()//Escribo el bucle del programa
7 {
8   //Si
9   if (digitalRead(2)==HIGH)//Lectura digital Alta
10  {
11    digitalWrite(7, HIGH);//Escritura digital Alta
12  }
13  //Si no
14  else
15  {
16    digitalWrite(7, LOW);//Escritura digital Baja
17  }
18 }
19 }
```

<https://www.tinkercad.com/things/eILBWwif02O>



Cuando el Sensor PIR detecta movimiento en su rango de visión, emite una señal eléctrica que se transmite a la entrada digital 2 del Arduino. Este evento de detección de movimiento es interpretado por el Arduino, que está programado para monitorear esta entrada.

Al detectar la señal del sensor PIR, el Arduino responde activando la salida digital 7. Esta salida digital, a su vez, está conectada a la bobina del relé configurado en modo normalmente abierto.

La bobina del relé se energiza debido a la activación de la salida digital 7 del Arduino. Esta energización provoca un cambio en la posición del contacto del relé: el contacto normalmente abierto, que inicialmente estaba separado, ahora se cierra.

La conexión del contacto normalmente abierto del relé se establece en el circuito que conecta la batería de 9 voltios y la lámpara. Cuando el relé cambia su estado y cierra el contacto, se completa el circuito eléctrico entre la batería y la lámpara.

Esta conexión cerrada permite que la corriente eléctrica fluya desde la batería hacia la lámpara. Como resultado, la lámpara se ilumina, proporcionando luz en respuesta al movimiento detectado por el sensor PIR.

En resumen, el sistema funciona en una secuencia coherente: el Sensor PIR detecta movimiento y envía una señal al Arduino, el Arduino activa la bobina del relé, el relé cambia la posición del contacto abierto a contacto cerrado y se cierra el circuito entre la batería y la lámpara. Como resultado, la lámpara se enciende en respuesta al movimiento detectado. Este proceso demuestra cómo la interacción entre el sensor, el Arduino y el relé permite controlar la iluminación basada en eventos de detección de movimiento.

2. Actividad Explicativa: Realización del circuito y la programación.

Situación Problema: “Optimización del Sistema de Iluminación Automática con Detección de Movimiento”

Continuando con el proyecto de diseño de un sistema de iluminación automática con detección de movimiento para un hogar inteligente, se ha identificado la necesidad de optimizar el código de programación para garantizar un funcionamiento más eficiente y mejorar la legibilidad del mismo.

Requisitos del Proyecto:

- **Circuito y Componentes:** El circuito consta de un Arduino, un relé, una lámpara, una batería de 9 voltios y un sensor PIR. Los componentes deben estar conectados de manera adecuada y funcional.
- **Sensor PIR:** El sensor PIR debe seguir detectando el movimiento en la habitación y transmitir la señal al Arduino.
- **Programación Optimizada:** Rediseñar la programación del Arduino utilizando constantes para representar los números de pin y otros valores importantes. Las constantes deben utilizarse en lugar de números directos para mejorar la legibilidad y permitir futuras modificaciones más sencillas.

- **Funcionalidad:** El sistema debe encender automáticamente la lámpara cuando se detecta movimiento y apagarla cuando no haya movimiento. La optimización del código no debe afectar negativamente la funcionalidad básica del sistema.

Desafíos para los Estudiantes:

- **Definición de Constantes:** Aprender a definir constantes con nombres significativos para representar los números de pin y otros valores en el código.
- **Adaptación del Código:** Modificar el código existente para reemplazar los números de pin y valores directos por las constantes definidas, manteniendo la lógica de funcionamiento intacta.
- **Pruebas y Verificación:** Realizar pruebas exhaustivas para asegurarse de que la optimización del código no haya alterado la funcionalidad del sistema y que la lámpara se encienda y apague correctamente en respuesta al movimiento.

Resultado Esperado:

Se espera que los estudiantes logren optimizar el código de programación utilizando constantes para representar los números de pin y otros valores importantes en lugar de números directos. El sistema de iluminación automática debe seguir funcionando de manera confiable y eficiente, con una mejora en la legibilidad y mantenibilidad del código.

Ahora modifiquemos la programación definiendo las constantes para los pines del sensor PIR y del relé.

Definir constantes en la programación es una práctica importante por varias razones, y en este caso específico de la programación para controlar un sistema con Arduino, hay algunas razones clave:

- A) **Legibilidad y Mantenibilidad:** Al asignar un nombre significativo a un valor constante, como un número de pin o un umbral, el código se vuelve más legible y comprensible. Esto es especialmente útil cuando vuelves a revisar o mantienes el código en el futuro, ya que facilita la identificación de los componentes y valores utilizados.
- B) **Flexibilidad y Cambios Futuros:** Si en el futuro necesitas cambiar un número de pin o un valor específico en tu programa, solo necesitas hacer el cambio en un solo lugar: la definición de la constante. Esto evita la necesidad de buscar en todo el código para reemplazar el valor en múltiples lugares y reduce el riesgo de cometer errores.
- C) **Evita Números Mágicos:** Los "números mágicos" son valores literales que aparecen directamente en el código sin una explicación clara. Usar números mágicos puede dificultar la comprensión del código y hacer que sea más propenso a errores. Al definir constantes, das un nombre descriptivo a esos valores, lo que hace que su propósito sea más evidente.
- D) **Facilita la Reutilización:** Si tienes que utilizar el mismo número de pin o valor en varios lugares de tu código o incluso en diferentes proyectos, definirlo como una constante facilita la reutilización. Puedes simplemente importar o copiar esa definición de constante en otros proyectos sin preocuparte por los detalles internos.
- E) **Cambios Globales:** Si en algún momento necesitas cambiar la configuración de los pines o valores, solo necesitas cambiar la definición de la constante en un solo lugar y todo el código se ajustará automáticamente.

En el caso específico de esta programación, definir las constantes "**pirPin**" y "**relayPin**" permite organizar y gestionar los números de pin de manera eficiente. Esto mejora la legibilidad del código, hace que sea más fácil ajustar los pines en caso de cambios y promueve buenas prácticas de programación que conducen a un código más manejable y menos propenso a errores.

Este código se encarga de controlar un sistema que utiliza un sensor de movimiento (PIR), un relé y una lámpara de 9 voltios conectada a una batería. La programación se realiza en un microcontrolador Arduino y se basa en la detección de movimiento para encender y apagar la lámpara a través del relé. A continuación, se describe la funcionalidad paso a paso.

```

1
2 // Definición de pines
3 const int pirPin = 2; // Pin del sensor PIR
4 const int relayPin = 7; // Pin de control del relé
5
6 void setup() {
7     pinMode(pirPin, INPUT); // Configura el pin del sensor PIR como entrada
8     pinMode(relayPin, OUTPUT); // Configura el pin de control del relé como salida
9     digitalWrite(relayPin, LOW); // Apaga inicialmente el relé (sin energía en la bobina)
10 }
11
12 void loop() {
13     // Lee el estado del sensor PIR
14     int motionDetected = digitalRead(pirPin);
15
16     // Si se detecta movimiento (sensor PIR HIGH)
17     if (motionDetected == HIGH) {
18         // Activa el relé (energiza la bobina) para encender la lámpara
19         digitalWrite(relayPin, HIGH);
20     } else {
21         // Apaga el relé (sin energía en la bobina) para apagar la lámpara
22         digitalWrite(relayPin, LOW);
23     }
24
25     // Espera un breve periodo para evitar lecturas repetidas muy rápidas
26     delay(100);
27 }

```

- 1) **Definición de Pines:** Se definen dos constantes para representar los números de los pines que se utilizarán en el Arduino. “**pirPin**” se establece en el pin 2 para conectarse al sensor PIR, y “**relayPin**” se establece en el pin 7 para controlar el relé.

```

// Definición de pines
const int pirPin = 2; // Pin del sensor PIR
const int relayPin = 7; // Pin de control del relé

```

- 2) **Configuración Inicial (setup):** En la función setup (), se configuran los pines definidos previamente. El “**pirPin**” se configura como una entrada para recibir la señal del sensor PIR, mientras que el “**relayPin**” se configura como una salida para controlar el relé. Además, se apaga inicialmente el relé al establecer el estado del “**relayPin**” en LOW, asegurando que no haya energía en la bobina del relé al inicio.

```

void setup() {
    pinMode(pirPin, INPUT); // Configura el pin del sensor PIR como entrada
    pinMode(relayPin, OUTPUT); // Configura el pin de control del relé como salida
    digitalWrite(relayPin, LOW); // Apaga inicialmente el relé (sin energía en la bobina)
}

```

- 3) **Bucle Principal (loop):** La función loop() se ejecuta continuamente después de la configuración inicial. Aquí ocurre la mayor parte de la lógica del programa.

Se lee el estado del sensor PIR utilizando “**digitalRead(pirPin)**”, y el resultado se almacena en la variable “**motionDetected**”.

Luego, se utiliza una estructura condicional “**if-else**” para determinar si se ha detectado movimiento. Si “**motionDetected**” es HIGH, significa que se detectó movimiento, por lo que se activa el relé utilizando “**digitalWrite(relayPin, HIGH)**”. Esto energiza la bobina del relé, cierra el contacto y enciende la lámpara.

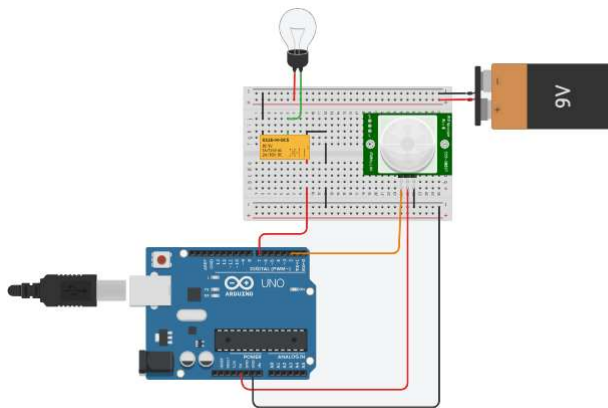
Si no se detecta movimiento (cuando “**motionDetected**” es LOW), se desactiva el relé utilizando **digitalWrite(relayPin, LOW)**. Esto corta la energía en la bobina del relé, abre el contacto y apaga la lámpara.

Después de tomar una decisión basada en la detección de movimiento, se agrega un retraso de 100 milisegundos utilizando **delay(100)**. Esto evita que se realicen lecturas repetidas muy rápidas y permite un tiempo de espera entre detección.

```
void loop() {
  // Lee el estado del sensor PIR
  int motionDetected = digitalRead(pirPin);

  // Si se detecta movimiento (sensor PIR HIGH)
  if (motionDetected == HIGH) {
    // Activa el relé (energiza la bobina) para encender la lámpara
    digitalWrite(relayPin, HIGH);
  } else {
    // Apaga el relé (sin energía en la bobina) para apagar la lámpara
    digitalWrite(relayPin, LOW);
  }

  // Espera un breve periodo para evitar lecturas repetidas muy rápidas
  delay(100);
}
```



```
1 // Definición de pines
2 const int pirPin = 2; // Pin del sensor PIR
3 const int relayPin = 7; // Pin de control del relé
4
5
6 void setup() {
7   pinMode(pirPin, INPUT); // Configura el pin del sensor PIR como entrada
8   pinMode(relayPin, OUTPUT); // Configura el pin de control del relé como salida
9   digitalWrite(relayPin, LOW); // Apaga inicialmente el relé (sin energía en la bobina)
10 }
11
12 void loop() {
13   // Lee el estado del sensor PIR
14   int motionDetected = digitalRead(pirPin);
15
16   // Si se detecta movimiento (sensor PIR HIGH)
17   if (motionDetected == HIGH) {
18     // Activa el relé (energiza la bobina) para encender la lámpara
19     digitalWrite(relayPin, HIGH);
20   } else {
21     // Apaga el relé (sin energía en la bobina) para apagar la lámpara
22     digitalWrite(relayPin, LOW);
23   }
24
25   // Espera un breve periodo para evitar lecturas repetidas muy rápidas
26   delay(100);
27 }
```

https://www.tinkercad.com/things/iYAu2DVxfRi?sharecode=w_ECN72EN4bH28foXynPf_Qbbkz4yfHFglLEwMKVs8

Situaciones Problemas

A continuación se plantearán una serie de situaciones problemas para poder trabajar con los estudiantes. A cada una de ellas corresponderá una solución alternativa, para la cual las y los estudiantes utilizarán las herramientas del pensamiento computacional y el método de Pólya para resolver problemas.

Pensamiento computacional



Las herramientas del pensamiento computacional son fundamentales para abordar problemas de manera efectiva en la informática y más allá. Ayudan a desglosar problemas complicados, simplificar conceptos, identificar soluciones comunes y diseñar procedimientos precisos para resolver tareas. Estas habilidades son aplicables tanto en la programación como en la resolución de problemas en la vida cotidiana.

Descomposición: La descomposición implica dividir un problema complejo en partes más pequeñas y manejables. En lugar de abordar el problema en su totalidad, se descompone en subproblemas más simples. Esto facilita la comprensión y el enfoque en cada componente por separado, lo que a su vez permite una resolución más efectiva del problema general.

Abstracción: La abstracción consiste en simplificar un problema o concepto al eliminar los detalles innecesarios y centrarse en los aspectos esenciales. Al abstraer, se destacan las características clave y se ignoran los detalles menos importantes. En programación, esto puede significar la creación de funciones, clases o módulos para representar ideas complejas de manera más simple y manejable.

Patrones: Los patrones son soluciones recurrentes a problemas comunes. En el pensamiento computacional, identificar patrones significa reconocer similitudes y regularidades en diferentes situaciones o conjuntos de datos. Al comprender los patrones, puedes crear soluciones más eficientes y reutilizables, ya que no tienes que reinventar la rueda cada vez que te enfrentas a un problema similar.

Algoritmo: Un algoritmo es un conjunto preciso y ordenado de instrucciones que se diseñan para resolver un problema o realizar una tarea específica. Los algoritmos son esenciales en la programación, ya que indican el camino exacto para lograr un resultado deseado. Pueden ser simples o complejos, pero todos siguen una estructura lógica y se componen de pasos discretos y definidos.



- 1) **Comprender el problema.** Reconocer que se pregunta, identificar lo que hay que resolver y las condiciones asociadas.
- 2) **Elaborar un plan.** Se trata de establecer la vinculación entre los datos presentes y el problema a resolver, determinar los recursos que se utilizarán, verificar la similitud con otros problemas previamente resueltos y también la posibilidad de utilizar teorías o modelos útiles, todo esto en función de buscar una manera de resolver el problema.
- 3) **Ejecutar el plan.** Desarrollar el resultado de la respuesta, a partir de ejecutar el plan, avanzando y verificando cada paso.
- 4) **Revisar y verificar.** Controlar que hace y que dice el resultado, con vistas a considerar la posibilidad de transferir la solución a otros problemas.

3. Problemática circuital y de programación. (Pensamiento Computacional - Método de Pólya)

Situación Problema: “Control de temperatura ambiental para un Invernadero Automatizado”

En la ciudad, un grupo de estudiantes apasionados por la tecnología están interesados en diseñar un sistema de control ambiental para un invernadero comunitario. El objetivo es mantener las condiciones óptimas de temperatura en el invernadero para el crecimiento saludable de las plantas. Sin embargo, debido a los cambios climáticos inesperados, es difícil mantener una temperatura constante. A veces, el invernadero se calienta demasiado, lo que puede dañar las plantas.

Tarea:

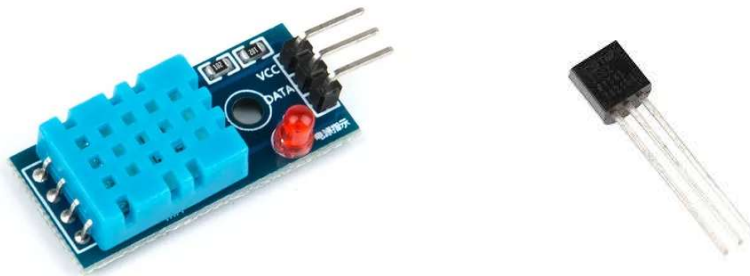
Diseñar un sistema de control ambiental utilizando un Arduino y sensores de temperatura. El sistema debe monitorear la temperatura dentro del invernadero y activar un motor de ventilación (controlado por un relé) para enfriar el invernadero cuando la temperatura supera los 28 grados Celsius. Además, el sistema debe enviar alertas visuales y sonoras cuando la temperatura alcanza niveles críticos.

Requerimientos:

Utilizar un Arduino para leer la temperatura con un sensor analógico y controlar un motor de ventilación con un relé.
Configurar umbrales de temperatura para encender y apagar el motor de ventilación.
Implementar señales visuales y sonoras (usando LED y zumbador) para alertar cuando la temperatura es crítica.
Diseñar una interfaz de usuario simple en el monitor serial para mostrar la temperatura actual y las acciones realizadas por el sistema.
Documentar el proceso de diseño, incluyendo el código, diagramas de conexión y explicaciones.

Para tener en cuenta:

Un sensor de temperatura es un componente electrónico diseñado para medir y proporcionar información sobre la temperatura del entorno en el que se encuentra. Estos sensores son utilizados comúnmente en proyectos electrónicos y sistemas de automatización para monitorear y controlar la temperatura. El sensor de temperatura permite a los dispositivos interactuar con su entorno al capturar datos térmicos y convertirlos en valores que pueden ser interpretados y utilizados por un microcontrolador como el Arduino.



Para usar un sensor de temperatura con Arduino, generalmente se requiere conectarlo correctamente al pin analógico del Arduino y luego utilizar el código adecuado para leer el valor analógico del sensor. Este valor se puede convertir en una temperatura en grados Celsius o Fahrenheit utilizando fórmulas y calibraciones específicas para cada tipo de sensor. Los sensores de temperatura se conectan a un pin analógico en Arduino porque muchos de estos sensores generan una señal de salida analógica que varía en función de la temperatura. En otras palabras, la salida de estos sensores no es simplemente "encendido" o "apagado" como en los pines digitales, sino que proporcionan un rango continuo de valores que representan la temperatura.

Los pines analógicos en Arduino son capaces de medir voltajes en un rango de 0 a 5 voltios y convertirlos en valores digitales en el rango de 0 a 1023, lo que significa que pueden interpretar niveles variables de voltaje. Esto es ideal para leer las salidas analógicas de los sensores, como los sensores de temperatura, que generan una señal proporcional al valor de la temperatura.

Cuando conectas un sensor de temperatura a un pin analógico, puedes usar la función **analogRead()** en Arduino para obtener el valor de voltaje analógico generado por el sensor. Luego, puedes usar cálculos y calibraciones específicas para convertir ese valor de voltaje en una temperatura en grados Celsius o Fahrenheit.

Aplicamos el pensamiento computacional para resolver el problema



En conjunto, estas herramientas y conceptos del pensamiento computacional permiten abordar la problemática de manera estructurada y eficiente, ayudando a los estudiantes a diseñar soluciones paso a paso, comprender las partes esenciales del problema y crear un código claro y funcional.

1. Descomposición: La descomposición implica dividir un problema en subproblemas más pequeños y manejables. En el contexto de control de temperatura en un invernadero, podríamos descomponerlo de la siguiente manera:

- **Lectura del Sensor:** Dividir en subproblemas la lectura del sensor de temperatura, la conversión del valor analógico a temperatura y la visualización de la temperatura en la consola.
- **Control del Relé:** Dividir en subproblemas el control del relé (encender o apagar) en función de la temperatura.
- **Alertas:** Dividir en subproblemas la generación de alertas visuales y sonoras cuando la temperatura es crítica.

2. Abstracción: La abstracción implica identificar los detalles esenciales de un problema y enfocarse en las partes relevantes. En este caso:

- Abstractar el funcionamiento interno del sensor de temperatura, centrándonos en cómo leer su valor y convertirlo en una temperatura.
- Abstractar el funcionamiento interno del relé, considerando solamente su capacidad para encender o apagar el motor.
- Abstractar las alertas, pensando en cómo activar visuales y sonoras sin preocuparse por los detalles internos de los componentes.

3. Patrones: Los patrones son soluciones reutilizables a problemas comunes. Algunos patrones que podrían aplicarse aquí incluyen:

- **Patrón de Control:** Utilizar un patrón de control condicional (if-else) para tomar decisiones basadas en la temperatura.
- **Patrón de Alerta:** Utilizar un patrón de activación y desactivación para generar alertas visuales y sonoras.

4. Algoritmo: Un algoritmo es una secuencia de pasos para resolver un problema. En este caso, el algoritmo general para controlar la temperatura podría ser:

- Leer el valor analógico del sensor de temperatura.
- Convertir el valor a temperatura en grados Celsius.
- Mostrar la temperatura en la consola.
- Si la temperatura es mayor o igual al umbral:
 - Encender el relé para activar el motor.
 - Hacer sonar el buzzer.
 - Encender el LED.
- Si la temperatura es menor al umbral:
 - Apagar el relé para detener el motor.
 - Detener el buzzer.
 - Apagar el LED.
 - Esperar un segundo antes de repetir el ciclo.

Aplicamos el método de Póyla teniendo en cuenta las herramientas del pensamiento computacional



Etapas del método Pólya de Resolución de Problemas

1. Comprender el Problema: Implica clarificar el objetivo central, considerar las limitaciones y recursos disponibles, analizar los datos proporcionados (como el funcionamiento del sensor de temperatura) y asegurarse de que se entiende plenamente lo que se pide hacer. Esta fase establece una base sólida para abordar el problema de manera efectiva y precisa.

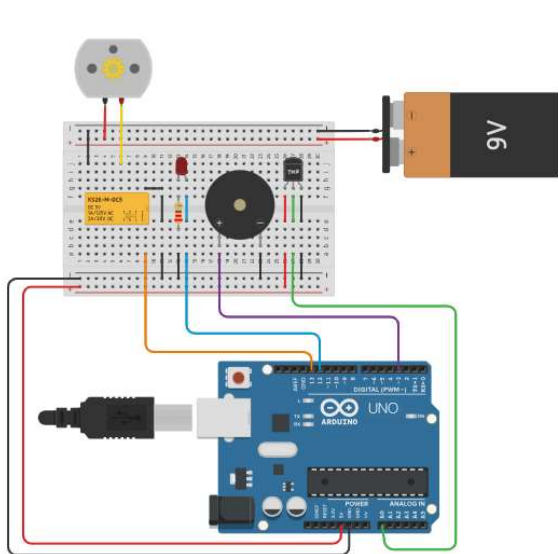
- **Identificar el objetivo:** Diseñar un sistema que controle la temperatura en un invernadero y active el motor, el buzzer y el LED cuando la temperatura supere el umbral de 28 grados Celsius.
- **Entender las limitaciones:** Utilizar un Arduino, un sensor de temperatura, un relé, un buzzer y un LED.
- **Analizar los datos:** El sensor de temperatura proporciona un valor analógico que debe convertirse en temperatura en grados Celsius.

2. Elaborar un Plan: Implica dividir el problema en pasos manejables, utilizar abstracciones para concentrarse en las partes esenciales y aplicar patrones de solución conocidos. En esta fase, se define un plan claro y estructurado para resolver el problema, considerando cómo abordar cada componente y paso necesario de manera eficiente y efectiva.

- **Descomposición:** Dividir el problema en tres partes principales: Lectura del sensor y conversión, control del relé y generación de alertas.
- **Abstracción:** Enfocarse en la funcionalidad esencial de cada parte, sin preocuparse por los detalles internos de los componentes.
- **Patrones:** Utilizar condicionales (if-else) para tomar decisiones basadas en la temperatura. Utilizar un patrón de activación y desactivación para las alertas.

3. Ejecutar el Plan: Implica poner en práctica el plan elaborado anteriormente. Aquí es donde se implementan las instrucciones y acciones definidas en el plan para resolver el problema. En esta fase, se lleva a cabo la programación, la conexión de componentes y cualquier acción necesaria para seguir el enfoque diseñado y lograr la solución deseada.

- Implementar el código según el algoritmo definido previamente.
- Conectar correctamente los componentes según la descripción de conexión.



```
1 const int analogPin = A0; // Pin analógico A0 donde está conectado el sensor de temperatura
2 const int RELAY = 13; // Pin digital 13 conectado al relé
3 const int BUZZER = 3; // Pin digital 3 conectado al Buzzer
4 const int LED = 12; // Pin digital 12 conectado al LED
5 float temperatureThreshold = 28.0; // Umbral de temperatura para encender el motor
6
7
8 void setup() {
9   pinMode(RELAY, OUTPUT); // Configurar el pin digital 13 como salida
10  pinMode(BUZZER, OUTPUT); // Configurar el pin digital 3 como salida
11  pinMode(LED, OUTPUT); // Configurar el pin digital 12 como salida
12  Serial.begin(9600); // Iniciar la comunicación serial
13 }
14
15 void loop() {
16   // Leer el valor analógico del sensor de temperatura
17   int sensorValue = analogRead(analogPin);
18
19   // Convertir el valor analógico a temperatura en grados Celsius
20   float voltage = sensorValue * (5.0 / 1023.0);
21   float temperature = (voltage - 0.5) * 100.0;
22
23   // Mostrar la temperatura en la consola serial
24   Serial.print("Temperatura: ");
25   Serial.print(temperature);
26   Serial.println(" °C");
27
28   // Controlar el motor a través del relé
29   // Si
30   if (temperature >= temperatureThreshold) {
31     digitalWrite(RELAY, HIGH); // Encender el motor a través del relé
32     tone(BUZZER, 500, 1000); // Reproducir el tono
33     digitalWrite(LED, HIGH);
34   } // Si No
35   else {
36     digitalWrite(RELAY, LOW); // Apagar el motor a través del relé
37     digitalWrite(LED, LOW); // Apagar el Led
38     noTone(BUZZER); // No reproducir el tono
39   }
40   delay(1000); // Esperar un segundo antes de realizar la próxima lectura
41 }
42
```

<https://www.tinkercad.com/things/kEcS5zhdSNS?sharecode=hifNI3FTxWpiWkH4yqMFRk6Zg8l4ronWhHH37R96gBg>

4. Revisar y Verificar:

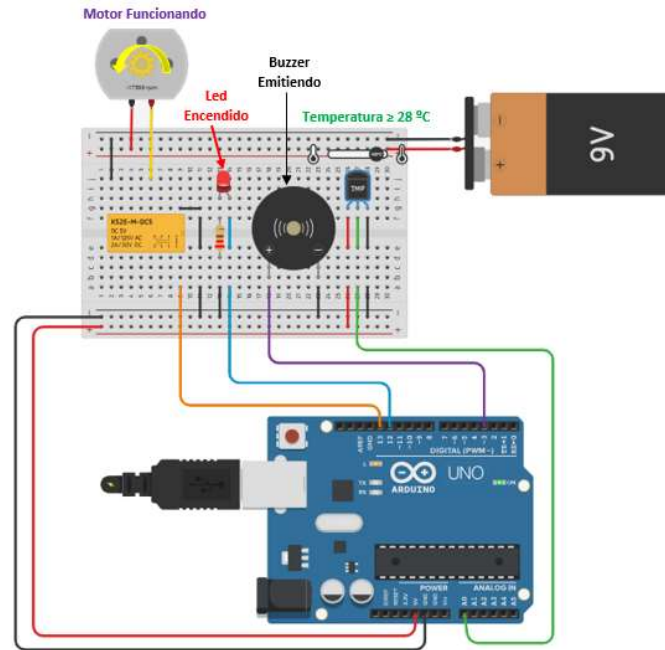
Verificar: Ejecutar el programa y observar el comportamiento. Comprobar si el motor, el buzzer y el LED se activan y desactivan correctamente según la temperatura.

Revisar: Analizar el código para asegurarse de que cumple con los requisitos y las condiciones.

Modificar: Realizar ajustes si es necesario, como cambiar los umbrales de temperatura o ajustar los tiempos de espera.

Evaluar el proceso de resolución: Considerar si se aplicaron correctamente las herramientas de pensamiento computacional, si el diseño fue eficiente y si el código es claro y comprensible.

Reflexionar sobre posibles mejoras: ¿Cómo se podría expandir el sistema? ¿Existen maneras de hacerlo más preciso o eficiente? ¿Se podrían agregar más sensores o funcionalidades?



```

1 const int analogPin = A0; // Pin analógico A0 donde está conectado el sensor de temperatura
2 const int RELAY = 13; // Pin digital 13 conectado al relé
3 const int BUZZER = 3; // Pin digital 3 conectado al Buzzer
4 const int LED = 12; // Pin digital 12 conectado al LED
5 float temperatureThreshold = 28.0; // Umbral de temperatura para encender el motor
6
7 void setup() {
8   pinMode(RELAY, OUTPUT); // Configurar el pin digital 13 como salida
9   pinMode(BUZZER, OUTPUT); // Configurar el pin digital 3 como salida
10  pinMode(LED, OUTPUT); // Configurar el pin digital 12 como salida
11  Serial.begin(9600); // Iniciar la comunicación serial
12 }
13
14 void loop() {
15   // Leer el valor analógico del sensor de temperatura
16   int sensorValue = analogRead(analogPin);
17
18   // Convertir el valor analógico a temperatura en grados Celsius
19   float voltage = sensorValue * (5.0 / 1023.0);
20   float temperature = (voltage - 0.5) * 100.0;
21
22   // Mostrar la temperatura en la consola serial
23   Serial.print("Temperatura: ");
24   Serial.print(temperature);
25   Serial.println(" °C");
26
27   // Controlar el motor a través del relé
28   // Si
29   if (temperature >= temperatureThreshold) {
30     digitalWrite(RELAY, HIGH); // Encender el motor a través del relé
31     tone(BUZZER, 500, 1000); // Reproducir el tono
32     digitalWrite(LED, HIGH);
33     // Si No
34   } else {
35     digitalWrite(RELAY, LOW); // Apagar el motor a través del relé
36     digitalWrite(LED, LOW); // Apagar el Led
37     noTone(BUZZER); // No reproducir el tono
38   }
39
40   delay(1000); // Esperar un segundo antes de realizar la próxima lectura
41 }
42

```

Descripción de la solución a la problemática:

Este circuito y su programación utilizan un sensor de temperatura para monitorear la temperatura ambiente. Si la temperatura supera el umbral establecido, activa un motor a través de un relé, hace sonar un buzzer y enciende un LED como señales de alerta. Si la temperatura es inferior al umbral, el motor se apaga, el buzzer se detiene y el LED se apaga. Esto podría ser utilizado, por ejemplo, para controlar las condiciones en un invernadero y alertar a los usuarios cuando la temperatura sea inadecuada para el crecimiento de las plantas.

Descripción de la Conexión:

- **Sensor de Temperatura:** El sensor de temperatura está conectado al pin analógico A0 del Arduino. Este sensor proporciona una señal analógica que varía según la temperatura.
- **Relé:** El relé se conecta al pin digital 13 del Arduino. Un relé es un interruptor controlado eléctricamente que puede encender o apagar dispositivos de mayor potencia, como un motor.
- **Buzzer:** El buzzer se conecta al pin digital 3 del Arduino. Un buzzer es un dispositivo que puede generar sonidos audibles cuando se activa.
- **LED:** El LED se conecta al pin digital 12 del Arduino. Un LED es un dispositivo que puede emitir luz cuando se aplica voltaje.

Descripción de la Programación:

- **Configuración Inicial (setup):**
 - Los pines del relé, buzzer y LED se configuran como salidas digitales utilizando las funciones pinMode.
 - Se inicia la comunicación serial con una velocidad de 9600 baudios para permitir la comunicación con el monitor serial.
- **Bucle Principal (loop):**
 - Se lee el valor analógico del sensor de temperatura conectado al pin A0. Este valor se convierte en una tensión y luego en una temperatura en grados Celsius.
 - La temperatura se muestra en el monitor serial.
 - Se comprueba si la temperatura es mayor o igual al umbral de 28 grados Celsius. Si es así:
 - El relé se activa para encender el motor (se asume que el motor está conectado al relé).
 - El buzzer emite un tono de 500 Hz durante 1000 ms.
 - El LED se enciende.
 - Si la temperatura es menor que el umbral:
 - El relé se desactiva para apagar el motor.
 - El buzzer se apaga (se detiene el tono).
 - El LED se apaga.
 - Después de cada ciclo, hay un retraso de 1000 ms (1 segundo) antes de la próxima lectura de temperatura.

4. Problemática circuital y de programación. (Pensamiento Computacional - Método de Pólya)

Situación Problema: Control de Temperatura en un Invernadero con visualización en display LCD

En la escuela, los estudiantes están encargados de diseñar un sistema automatizado de control de temperatura para un invernadero. Además, se les ha pedido que implementen un sistema de visualización utilizando un display LCD para mostrar información sobre la temperatura en el invernadero, así como también se requiere ajustar el contraste del display para una mejor legibilidad.

Requisitos y Desafíos:

Sensor de Temperatura: Los estudiantes deben seleccionar y conectar un sensor de temperatura al Arduino para medir la temperatura dentro del invernadero en tiempo real.

Control de Ventilador: Se requiere utilizar un ventilador para regular la temperatura dentro del invernadero. El ventilador se activará cuando la temperatura supere un umbral específico (por ejemplo, 28 grados Celsius) y se apagará cuando la temperatura baje por debajo de ese umbral.

Display LCD con Contraste Ajustable: Los estudiantes deben implementar un display LCD para mostrar información sobre la temperatura en el invernadero. Además, se espera que consideren la legibilidad del display y realicen ajustes para mejorarla si es necesario.

Interfaz de Usuario Intuitiva: Los usuarios deben poder entender y operar el sistema fácilmente. Los estudiantes pueden considerar agregar botones u otras formas de interacción para ajustar los parámetros del sistema.

Eficiencia Energética: Asegurar que el sistema de control de temperatura funcione de manera eficiente, activando el ventilador solo cuando sea necesario.

Documentación y Presentación: Los estudiantes deben documentar el diseño, las conexiones y el código utilizado en su proyecto. Luego, deberán presentar el proyecto a sus compañeros, explicando cómo el sistema controla la temperatura y cómo se logra una visualización adecuada en el display LCD.

Objetivos de Aprendizaje:

- Aplicar conocimientos de electrónica y programación para resolver un problema real.
- Entender cómo los sistemas automatizados pueden mejorar procesos y eficiencia.
- Aprender a controlar componentes electrónicos (ventilador) en función de condiciones ambientales (temperatura).
- Explorar la implementación de una interfaz de usuario simple y legible en un display LCD.
- Mejorar habilidades de documentación y presentación.
- Esta situación problema permitirá a los estudiantes enfrentar desafíos técnicos mientras abordan el control de temperatura y la presentación de datos en un invernadero, mejorando así su comprensión de la electrónica y la programación.

Para tener en cuenta:

Un display LCD (Liquid Crystal Display) para Arduino es un dispositivo visual que utiliza tecnología de cristales líquidos para mostrar información en forma de texto y gráficos. Estos displays son una forma eficaz de interactuar con proyectos electrónicos y microcontroladores como Arduino, ya que permiten mostrar datos de manera legible y fácil de entender. A continuación, se explican sus características y usos:

Características de un Display LCD para Arduino:

- **Pantalla:** Los displays LCD suelen tener una pantalla rectangular que puede mostrar caracteres alfanuméricos, números y símbolos.
- **Tecnología de Cristales Líquidos:** Utilizan cristales líquidos que responden a corrientes eléctricas y, según cómo se aplican estas corrientes, permiten o bloquean la luz, creando caracteres y gráficos en la pantalla.
- **Controlador:** Vienen con un controlador integrado que simplifica la interfaz con el microcontrolador, como Arduino. Esto facilita la programación y la visualización de datos.
- **Conexión:** Se conectan a través de varios pines del microcontrolador, generalmente utilizando una conexión paralela o a través de protocolos como I2C o SPI.

- **Tamaño y Resolución:** Vienen en diferentes tamaños y resoluciones, como 16x2 (16 caracteres en 2 líneas) o 20x4 (20 caracteres en 4 líneas).
- **Retroiluminación:** Algunos displays LCD tienen retroiluminación LED incorporada para mejorar la visibilidad en condiciones de poca luz.

Usos del Display LCD para Arduino:

- **Interfaz de Usuario:** Se utilizan para mostrar información relevante en proyectos electrónicos, como lecturas de sensores, estados de dispositivos, mensajes de usuario, etc.
- **Monitoreo:** Son útiles para monitorear variables en tiempo real, como temperatura, humedad, voltaje, corriente, entre otros.
- **Proyectos Educativos:** Los displays LCD son populares en proyectos educativos para enseñar programación y electrónica, ya que ofrecen una forma visual y tangible de aprender conceptos.
- **Visualización de Datos:** Permiten visualizar datos de manera más legible que a través de la consola serial, lo que facilita la interpretación de los resultados.

Aplicamos el pensamiento computacional para resolver el problema



En conjunto, estas herramientas del pensamiento computacional permiten a los estudiantes abordar problemas complejos de manera estructurada, lo que les ayuda a desarrollar habilidades de resolución de problemas y razonamiento lógico en contextos tecnológicos.

Descomposición: Dividimos el problema en partes más pequeñas y manejables. Por ejemplo, podemos dividirlo en la lectura del sensor de temperatura, el control del ventilador, la interfaz del display LCD y la eficiencia energética.

- **Lectura de Temperatura:**
 - Configurar el sensor de temperatura y el pin de lectura.
 - Leer el valor analógico del sensor.
 - Convertir el valor en grados Celsius.
- **Control del Ventilador:**
 - Definir el umbral de temperatura deseado.
 - Comparar la temperatura actual con el umbral.
 - Si la temperatura es mayor que el umbral, activar el ventilador.
 - Si la temperatura es menor que el umbral, apagar el ventilador.
- **Interfaz de Usuario:**
 - Configurar el display LCD y los pines de conexión.

- Mostrar el mensaje inicial en el display.
- Mostrar la temperatura en tiempo real en el display.
- Mostrar si el ventilador está encendido o apagado en el display.

- **Eficiencia Energética:**

- Definir un intervalo de tiempo para las lecturas del sensor.
- Activar el ventilador solo cuando sea necesario (temperatura > umbral) y desactivarlo cuando no sea necesario (temperatura ≤ umbral).

Abstracción: Identifica los conceptos clave y las entidades involucradas en el problema. Por ejemplo, puedes abstraer el sensor de temperatura como una fuente de datos, el ventilador como un actuador que responde a ciertas condiciones, el display LCD como una "interfaz visual" y el potenciómetro como un "regulador" que ajusta el contraste.

- **Sensor de Temperatura:** Ver el sensor de temperatura como una "fuente de datos" que proporciona valores analógicos relacionados con la temperatura del invernadero. Esto nos permite centrarnos en cómo obtener y procesar los datos de temperatura sin entrar en detalles específicos del sensor.
- **Control del Ventilador:** Considerar el ventilador como un "actuador" que puede estar "encendido" o "apagado". Su estado depende de la "comparación" entre la temperatura actual y el umbral de temperatura. Esta abstracción nos permite enfocarnos en cómo tomar decisiones basadas en la temperatura para controlar el ventilador, sin preocuparnos por la implementación específica del ventilador en sí.
- **Interfaz de Usuario:** Ver el display LCD como una "interfaz visual" que muestra información relevante, como la temperatura y el estado del ventilador. Esto nos ayuda a centrarnos en la presentación y actualización de la información para el usuario, en lugar de preocuparnos por los detalles internos del display.
- **Ajuste de Contraste (opcional):** Considerar el potenciómetro como un "regulador" que ajusta el contraste del display LCD para mejorar la legibilidad. Esta abstracción nos permite centrarnos en cómo el potenciómetro cambia el contraste sin necesidad de entrar en detalles técnicos sobre cómo funciona el potenciómetro en sí.

Patrones: Utiliza la estructura básica de lectura de datos del sensor y toma de decisiones para activar el ventilador cuando la temperatura supera el umbral.

- **Lectura de Temperatura:** Este patrón está presente en muchos proyectos que involucran sensores analógicos y permite una lectura consistente y precisa.
 - **Utilizar el patrón de "Lectura de Sensor Analógico":**
 - Configurar el pin de lectura del sensor.
 - Leer el valor analógico del sensor.
 - Convertir el valor a temperatura en grados Celsius.
- **Control del Ventilador:** Este patrón proporciona una estructura lógica para tomar decisiones basadas en condiciones específicas.
 - **Utilizar el patrón de "Toma de Decisiones":**
 - Definir un umbral de temperatura deseado.
 - Comparar la temperatura actual con el umbral.
 - Si la temperatura es mayor que el umbral, encender el ventilador.
 - Si la temperatura es menor o igual que el umbral, apagar el ventilador.
- **Interfaz de Usuario:** Este patrón se utiliza para comunicar información de manera efectiva a los usuarios a través de una interfaz visual.

- **Utilizar el patrón de "Interfaz de Usuario Visual":**
 - Configurar el display LCD y los pines de conexión.
 - Mostrar información en el display, como la temperatura y el estado del ventilador.

Algoritmo: Aplicar algoritmos aporta una estructura lógica y detallada a la implementación de soluciones. Cada algoritmo representa una serie de pasos que se pueden seguir para realizar una tarea específica en el contexto del problema. Los estudiantes pueden escribir, revisar y ajustar los algoritmos para garantizar que el sistema funcione de manera eficiente y efectiva. Crea algoritmos claros y ordenados para cada parte:

- **Algoritmo para leer el sensor de temperatura y convertir los valores en grados Celsius:**
 - Configurar el sensor de temperatura y el pin de lectura.
 - Leer el valor analógico del sensor.
 - Convertir el valor a temperatura en grados Celsius.
 - Devolver la temperatura.
- **Algoritmo para controlar el ventilador en función de la temperatura:**
 - Definir el umbral de temperatura deseado (por ejemplo, 28 grados Celsius).
 - Leer la temperatura actual utilizando el algoritmo de lectura.
 - Si la temperatura es mayor que el umbral: Encender el ventilador (activar el actuador).
 - Si la temperatura es menor o igual al umbral: Apagar el ventilador (desactivar el actuador).
- **Interfaz de Usuario:**
 - Configurar el display LCD y los pines de conexión.
 - Mostrar el mensaje inicial en el display.
 - En bucle:
 - Leer la temperatura actual utilizando el algoritmo de lectura.
 - Mostrar la temperatura en tiempo real en el display.
- **Algoritmo para mostrar información en el display LCD.**

Aplicamos el método de Pólya teniendo en cuenta las herramientas del pensamiento computacional



Aplicando el método de Pólya junto con las herramientas del pensamiento computacional, los estudiantes pueden abordar el problema de manera estructurada, desarrollando una solución completa y efectiva. Cada etapa del método y cada herramienta juegan un papel importante para garantizar una resolución exitosa del problema.

1. Comprender el Problema:

- **Utiliza la herramienta de Descomposición para dividir el problema en partes más manejables:**
 - Lectura de temperatura.
 - Control del ventilador.
 - Interfaz de usuario.

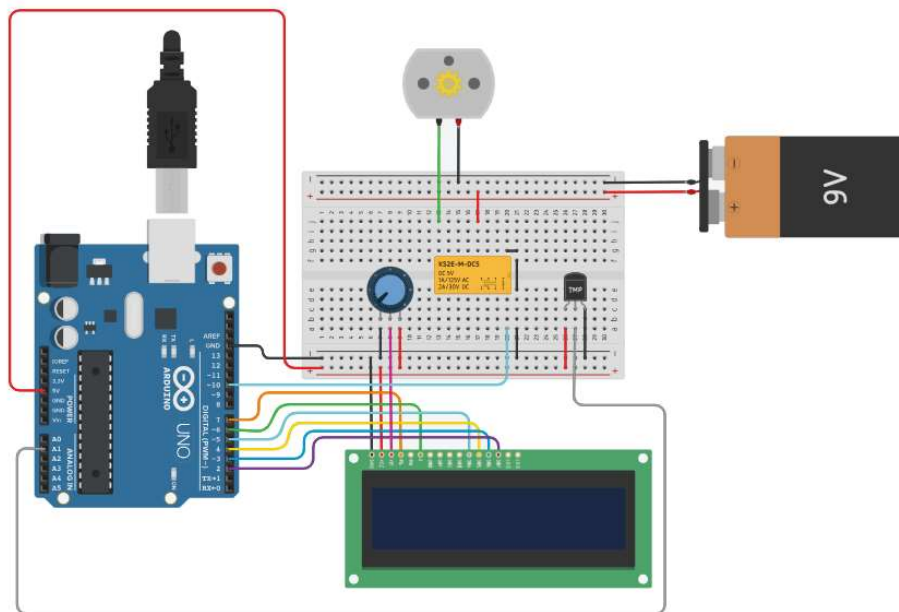
- Eficiencia energética.
- Documentación y presentación.
- **Abstrae los conceptos clave utilizando la herramienta de Abstracción:**
 - Sensor de temperatura como fuente de datos.
 - Ventilador como actuador controlado por umbral.
 - Display LCD como interfaz visual.
- **Identifica patrones comunes en el problema utilizando la herramienta de Patrones:**
 - Toma de decisiones basada en condiciones.
 - Lectura de sensores y actuadores.

2. Elabora un Plan:

- **Utiliza la herramienta de Algoritmo para planificar cada paso específico para resolver cada subproblema:**
 - Diseña algoritmo para lectura de temperatura.
 - Diseña algoritmo para control del ventilador.
 - Diseña algoritmo para la interfaz de usuario.
 - Diseña algoritmo para ajuste de contraste (opcional).
 - Diseña algoritmo para eficiencia energética.
 - Diseña algoritmo para documentación y presentación.

3. Ejecutar el Plan:

- **Implementa los algoritmos paso a paso siguiendo la planificación realizada en la fase anterior:**
 - Programa la lectura del sensor de temperatura.
 - Implementa el control del ventilador en función de la lectura de temperatura.
 - Crea la interfaz de usuario en el display LCD.
 - Integra el ajuste de contraste si es necesario.
 - Asegura la eficiencia energética al controlar el ventilador.
 - Prepara la documentación y la presentación para comunicar la solución.



<https://www.tinkercad.com/things/1rBkwWKS4Od>

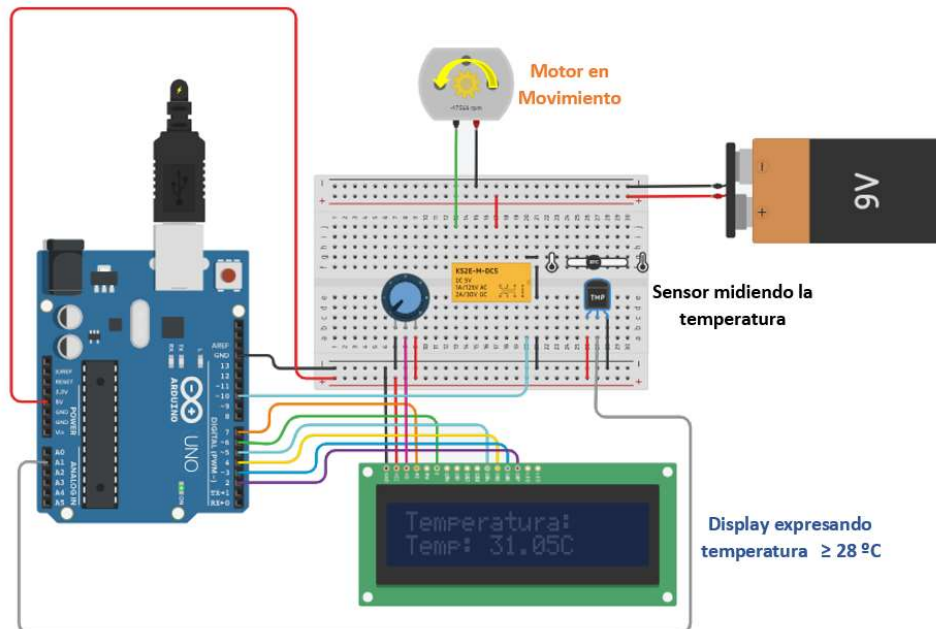
```

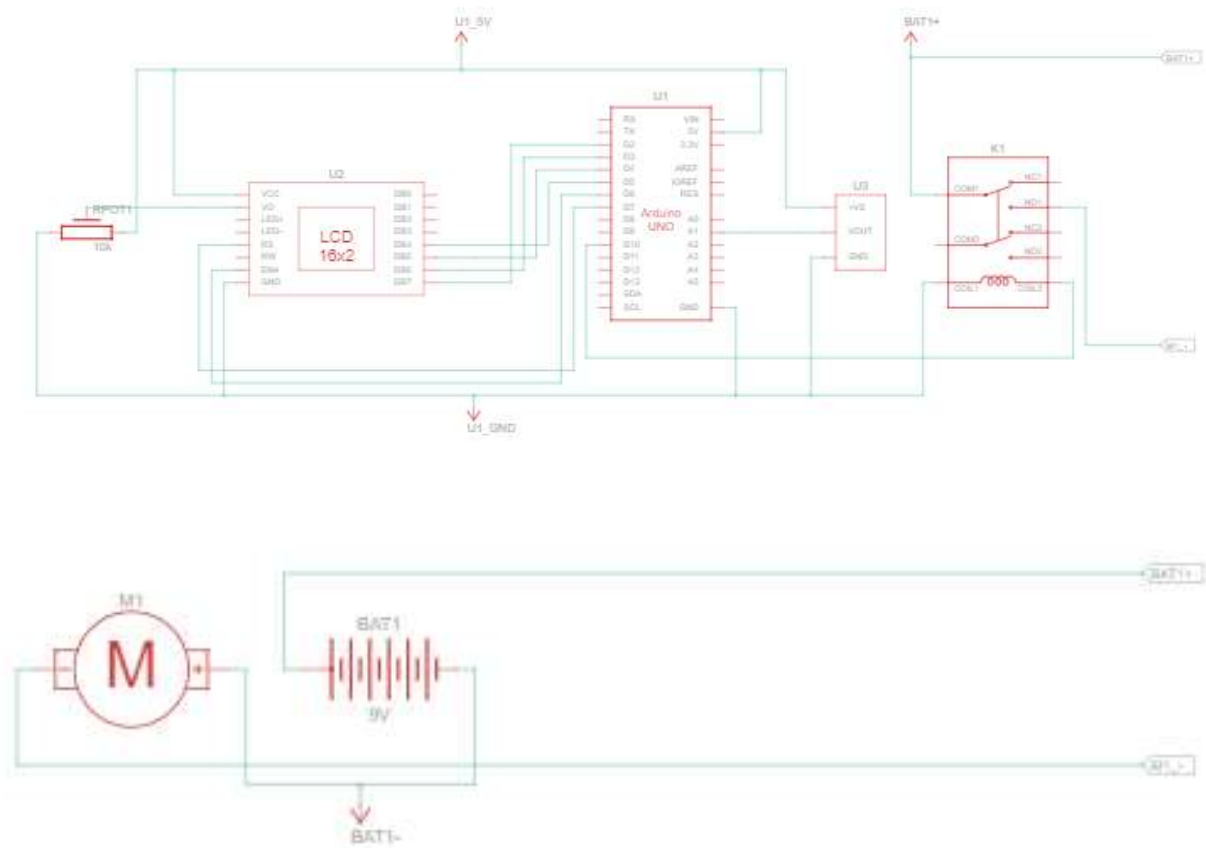
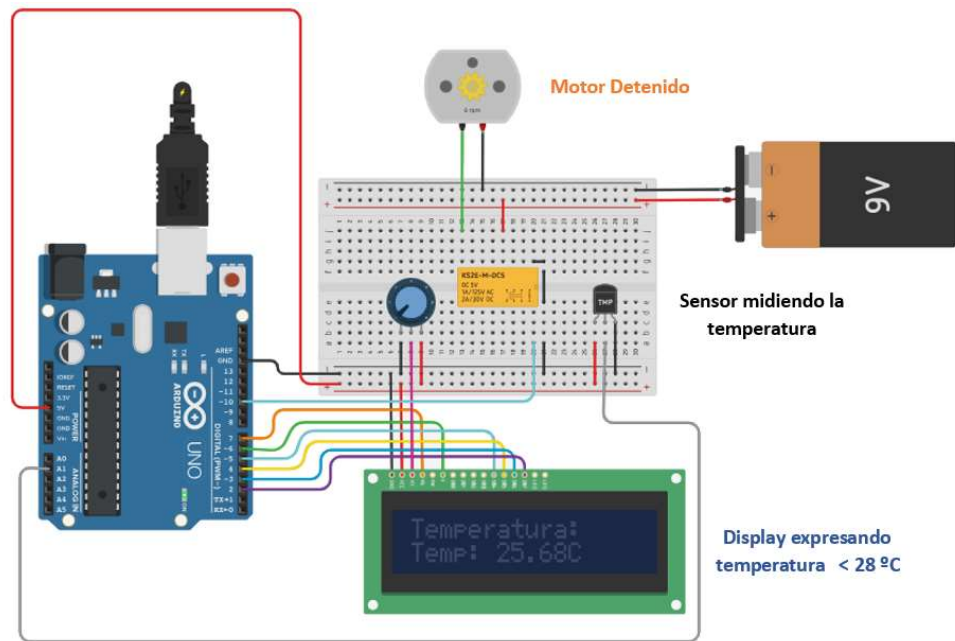
1 #include <LiquidCrystal.h>
2
3 LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // Configura tu LCD según las conexiones
4
5 const int tempSensorPin = A1; // Pin analógico del sensor de temperatura
6 const int relayPin = 10; // Pin digital para el control del relé
7 const float umbralTemperatura = 28.0; // Umbral de temperatura en grados Celsius
8
9 void setup() {
10     pinMode(relayPin, OUTPUT); // Configura el pin del relé como salida
11
12     lcd.begin(16, 2); // Inicializa el LCD
13     lcd.print("Temperatura:");
14 }
15
16 void loop() {
17     int sensorValue = analogRead(tempSensorPin); // Lee el valor del sensor
18     float temperature = (sensorValue * 5.0 / 1024.0 - 0.5) * 100; // Calcula la temperatura en grados Celsius
19
20     lcd.setCursor(0, 1); // Establece la posición del cursor en la segunda fila
21     lcd.print("Temp: ");
22     lcd.print(temperature);
23     lcd.print("C");
24
25     // Si la temperatura supera el umbral, activa el motor (enciende el relé)
26     if (temperature >= umbralTemperatura) {
27         digitalWrite(relayPin, HIGH); // Enciende el relé (enciende el motor)
28     } else {
29         digitalWrite(relayPin, LOW); // Apaga el relé (apaga el motor)
30     }
31
32     delay(1000); // Pausa de 1 segundo antes de la siguiente lectura
33 }
34
35

```

4. Revisar y Verificar:

- Utilizamos la herramienta de Abstracción para revisar el trabajo y asegurándonos de que todos los conceptos clave estén presentes y funcionando.
- Utilizamos la herramienta de Patrones para verificar que los patrones de toma de decisiones y lectura de sensores estén implementados correctamente.
- Utilizamos la herramienta de Algoritmo para revisar cada algoritmo y verificar que los pasos sean lógicos y consistentes.
- Probamos el sistema con diferentes temperaturas para verificar su funcionamiento, ajustando si es necesario.





5. Problemática circuital y de programación. (Pensamiento Computacional - Método de Pólya)

Situación Problema: Sistema de Riego Automático Inteligente para Plantas de Interior

Descripción: En una clase de taller, los estudiantes están aprendiendo sobre la automatización de sistemas articulando con ciencias naturales la importancia del riego adecuado para el crecimiento de las plantas. Como proyecto, se les pide diseñar un sistema de riego automático inteligente para plantas de interior utilizando un Arduino en Tinkercad.

Desafío: Los estudiantes deben crear un sistema que monitoree constantemente la humedad del suelo de una maceta y actúe en consecuencia para mantener el nivel de humedad óptimo para el crecimiento de la planta. Deben utilizar un sensor de humedad en tierra conectado a un pin analógico del Arduino para medir la humedad del suelo.

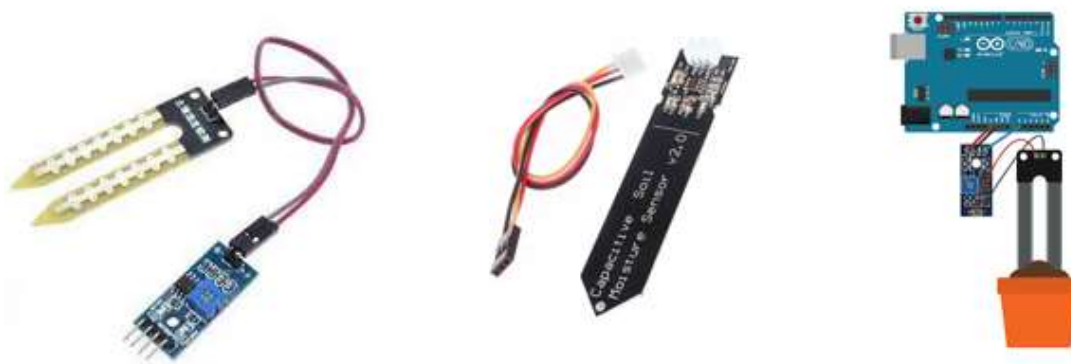
Objetivos: Diseñar y conectar correctamente el sensor de humedad en tierra al pin analógico y el relé al pin digital. Programar el Arduino para leer la humedad del suelo y controlar el motor de la bomba de agua a través del relé según los umbrales de humedad especificados.

Comprender y aplicar los conceptos de programación, electrónica y automatización en un contexto práctico.

Fomentar la resolución de problemas, la creatividad y el pensamiento crítico para crear un sistema funcional y eficiente.

Para tener en cuenta:

Un sensor de humedad en tierra es un dispositivo que se utiliza para medir la humedad del suelo, lo que resulta útil en aplicaciones de riego automático, monitoreo de jardinería, agricultura y otros proyectos relacionados con el cuidado de las plantas. Estos sensores funcionan midiendo la conductividad eléctrica del suelo, la cual está relacionada con su humedad.



Vemos una descripción básica de cómo funciona un sensor de humedad en tierra para Arduino:

Principio de medición: La mayoría de los sensores de humedad en tierra utilizan el principio de medición resistiva. Esto significa que miden la resistencia eléctrica del suelo, la cual varía en función de la cantidad de humedad presente.

Configuración física: El sensor consta de dos o más electrodos metálicos que se insertan en el suelo. Estos electrodos son generalmente de acero inoxidable y están separados por una distancia corta. Cuando el suelo está húmedo, actúa como un conductor y permite que la corriente fluya entre los electrodos. Si el suelo está seco, la resistencia eléctrica aumenta y la corriente disminuye.

Conversión a señal usable: El sensor de humedad está conectado a un circuito de interfaz que convierte la variación en la resistencia en una señal eléctrica adecuada para ser leída por un microcontrolador, como Arduino. Esto generalmente se logra mediante un divisor de voltaje o un circuito amplificador. La señal resultante suele ser un voltaje proporcional a la humedad del suelo.

Interfaz con Arduino: El voltaje de salida del sensor se conecta a uno de los pines analógicos del Arduino. Luego, el Arduino realiza una conversión analógico-digital (ADC) para obtener un valor digital que representa la lectura de humedad del suelo.

Calibración y lectura: Dado que la relación entre la resistencia del suelo y la humedad no es lineal ni consistente entre diferentes tipos de suelos, los sensores de humedad suelen requerir calibración. Esto implica determinar los valores de referencia para suelo seco y suelo saturado, y luego calcular la humedad relativa en función de las lecturas del sensor.

Programación: En el Arduino, debes escribir el código para leer el valor analógico del sensor y luego aplicar la fórmula de calibración para convertirlo en un valor de humedad relativa. Dependiendo de los valores que obtengas, puedes tomar decisiones sobre riego o mostrar información en una pantalla, entre otras acciones.

Aplicamos el pensamiento computacional para resolver el problema



Mediante la descomposición, abstracción, patrones y un algoritmo claro, resolvemos el problema de control de riego utilizando un sensor de humedad en tierra en Arduino. Estas herramientas nos permiten concentrarnos en aspectos clave y desarrollar un programa organizado y funcional.

Descomposición: Dividir el problema en partes más pequeñas y manejables es esencial para abordar tareas complejas. En el contexto del sistema de riego automatizado:

- **Identificar Componentes:** Descompongamos el sistema en componentes clave: el sensor de humedad, el Arduino y el relé que controlara la bomba de agua.
- **Conexiones y Comunicación:** Dividamos el problema en subproblemas más pequeños, como conectar el sensor al Arduino, escribir el código para leer el sensor y controlar el relé, y diseñar la interacción entre los componentes.

Abstracción: Simplificar el problema centrándonos en los detalles más relevantes y dejando de lado la información innecesaria:

- **Centrarnos en la Lógica:** En lugar de preocuparnos por detalles físicos específicos, abstractamente nos enfocamos en cómo los componentes interactúan y cómo la humedad del suelo afecta el riego.
- **Ignorar Detalles Técnicos Menores:** Enfocarnos en cómo la programación y los datos del sensor se conectan en lugar de profundizar en los detalles internos de cada componente.

Patrones: Identificar patrones recurrentes en el problema que nos permitan aplicar soluciones similares:

- **Condicionales:** Reconocer que necesitamos una estructura condicional (if-else) para activar o desactivar el riego según la humedad del suelo.
- **Bucles:** Si es necesario, podemos utilizar bucles para realizar lecturas periódicas del sensor y tomar decisiones de riego en intervalos regulares.

Algoritmo: Desarrollar una serie de pasos lógicos y secuenciales para resolver el problema:

- **Planificación del Código:** Crear un algoritmo que describa cómo el programa debe interactuar con el sensor, tomar decisiones basadas en la humedad y controlar el relé.
- **Estructura del Código:** Organizar el algoritmo en funciones y bloques lógicos coherentes, como la lectura del sensor, la toma de decisiones y el control del relé.

Aplicamos el método de Pólya teniendo en cuenta las herramientas del pensamiento computacional



Al seguir este método de Pólya y utilizar las herramientas del pensamiento computacional, podemos resolver el problema de control de riego de manera sistemática y eficiente.

1. Comprender el Problema: En esta etapa, se trata de comprender completamente el problema y lo que se nos pide. Aplicando las herramientas del pensamiento computacional:

- **Descomposición:** Al abordar un problema complejo, es más fácil comprenderlo si lo descomponemos en partes más simples que podemos analizar individualmente. En la etapa de comprensión del problema, utilizamos la descomposición para desglosar el problema general en componentes, subproblemas o aspectos específicos que podemos abordar por separado.
- **Abstracción:** Algunas veces, los detalles específicos pueden ser abrumadores o pormenorizados, y no es necesario entender todos los aspectos en profundidad en la etapa inicial. En la etapa de comprensión del problema, utilizamos la abstracción para captar la esencia del problema, centrarnos en los aspectos clave y comprender cómo se relacionan entre sí.

2. Elaborar un Plan: En esta etapa, se desarrolla un plan para resolver el problema. Utilizando las herramientas del pensamiento computacional:

- **Patrones:** En la etapa de "Elaborar un Plan", identificamos patrones que se han utilizado en situaciones similares y los aplicamos al problema actual. Esto nos permite aprovechar soluciones probadas y evitar reinventar la rueda. En el contexto del riego automatizado, podemos reconocer el patrón de uso de estructuras condicionales (if-else) para activar o desactivar el riego según la humedad.
- **Algoritmo:** En la etapa de "Elaborar un Plan", creamos un algoritmo que detalla cómo interactuarán los componentes, qué decisiones se tomarán y cómo se ejecutará el sistema en función de las condiciones. El algoritmo nos brinda una guía clara y detallada para implementar la solución.

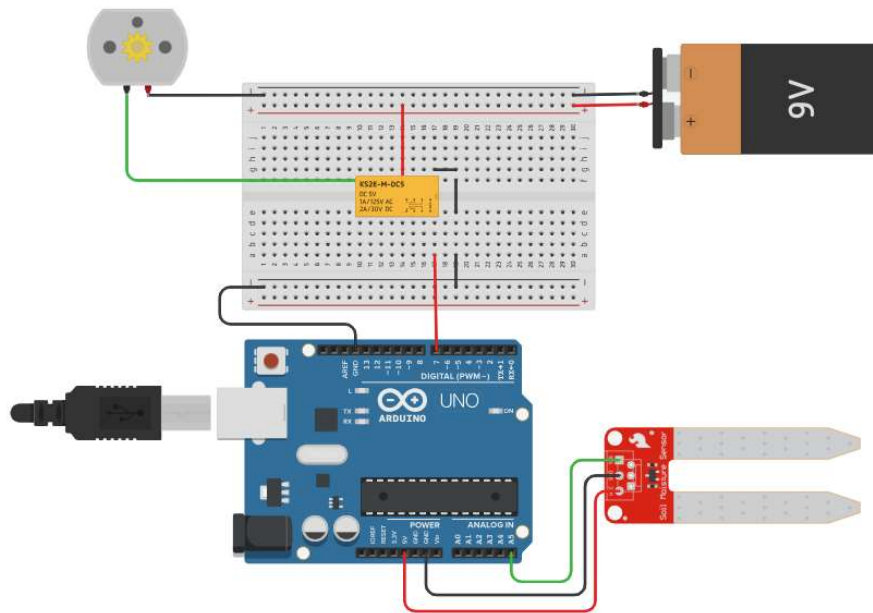
3. Ejecutar el Plan: Llevamos a cabo el plan desarrollado en la etapa anterior aplicando las herramientas del pensamiento computacional. Al ejecutar un plan, estamos implementando físicamente lo que hemos planificado. La descomposición nos ayuda a manejar partes específicas, los patrones nos guían con enfoques efectivos y el algoritmo asegura que sigamos una secuencia lógica y ordenada. Juntos, estos elementos nos permiten llevar a cabo la implementación con estructura y eficiencia, lo que resulta en una solución completa y funcional.

- **Descomposición:** Implementamos las diversas partes o componentes que identificamos durante la descomposición. Cada componente se implementa por separado y se integra en el sistema más grande. Al descomponer, estamos manejando cada parte de manera más manejable y controlada, lo que facilita la implementación y solución de problemas si surge algún inconveniente.
- **Patrones:** Aplicamos los patrones que hemos identificado previamente. Estos patrones son enfoques probados que sabemos que funciona bien para resolver ciertos tipos de problemas. Al utilizar patrones, podemos aprovechar las soluciones efectivas que otros han desarrollado previamente, lo que ahorra tiempo y evita posibles errores.
- **Algoritmo:** El algoritmo es la guía paso a paso que hemos creado para resolver el problema. Durante la ejecución, seguimos este algoritmo de manera secuencial, realizando cada paso en el orden especificado. El algoritmo actúa

como un conjunto de instrucciones claras y coherentes que aseguran que todos los componentes y acciones se coordinen correctamente.

4. Revisar y Verificar la Solución: En esta etapa, revisamos y mejoramos nuestra solución. Utilizamos las herramientas del pensamiento computacional de la siguiente manera:

- **Descomposición:** Revisamos cada componente y subproblema para asegurarnos de que funcionen correctamente.
- **Abstracción:** Verificamos que la lógica general de nuestro sistema sea sólida y que los detalles técnicos sean coherentes.
- **Patrones:** Comprobamos si hemos aplicado correctamente las estructuras condicionales y bucles.
- **Algoritmo:** Revisamos el flujo general del algoritmo para asegurarnos de que funcione según lo planeado.



https://www.tinkercad.com/things/e7s13wjxzVe?sharecode=Grcm_Tqvfcsv5jjNEE1EHEjKxuiR3EotCL4qsEdh4

```

1  const int sensorPin = A5;    // Pin analógico para el sensor de humedad
2  const int relayPin = 7;      // Pin digital para el relé
3
4  void setup() {
5      pinMode(relayPin, OUTPUT); // Configura el pin del relé como salida
6      pinMode(sensorPin, INPUT); // Configura el pin del relé como entrada
7      Serial.begin(9600);       // Inicializa la comunicación serial para la depuración
8  }
9
10 void loop() {
11     int sensorValue = analogRead(sensorPin); // Lee el valor analógico del sensor
12     Serial.print("Lectura de humedad: ");
13     Serial.println(sensorValue);             // Muestra la lectura en el monitor serial
14
15     if (sensorValue >= 800) { // Si la humedad es baja (Valor mayor o igual a 800) (suelo seco)
16         digitalWrite(relayPin, HIGH); // Activa el relé
17         Serial.println("Activando el relé");
18     } else { // Suelo húmedo
19         digitalWrite(relayPin, LOW); // Desactiva el relé
20         Serial.println("Desactivando el relé");
21     }
22
23     delay(1000); // Espera un segundo antes de la siguiente lectura
24 }
25

```

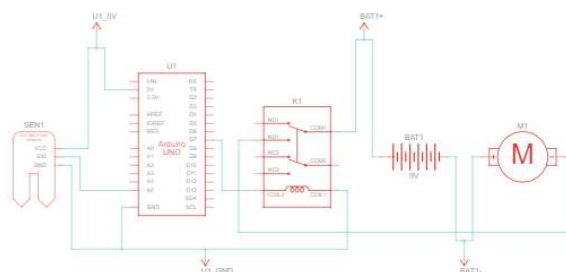
Veamos el código:

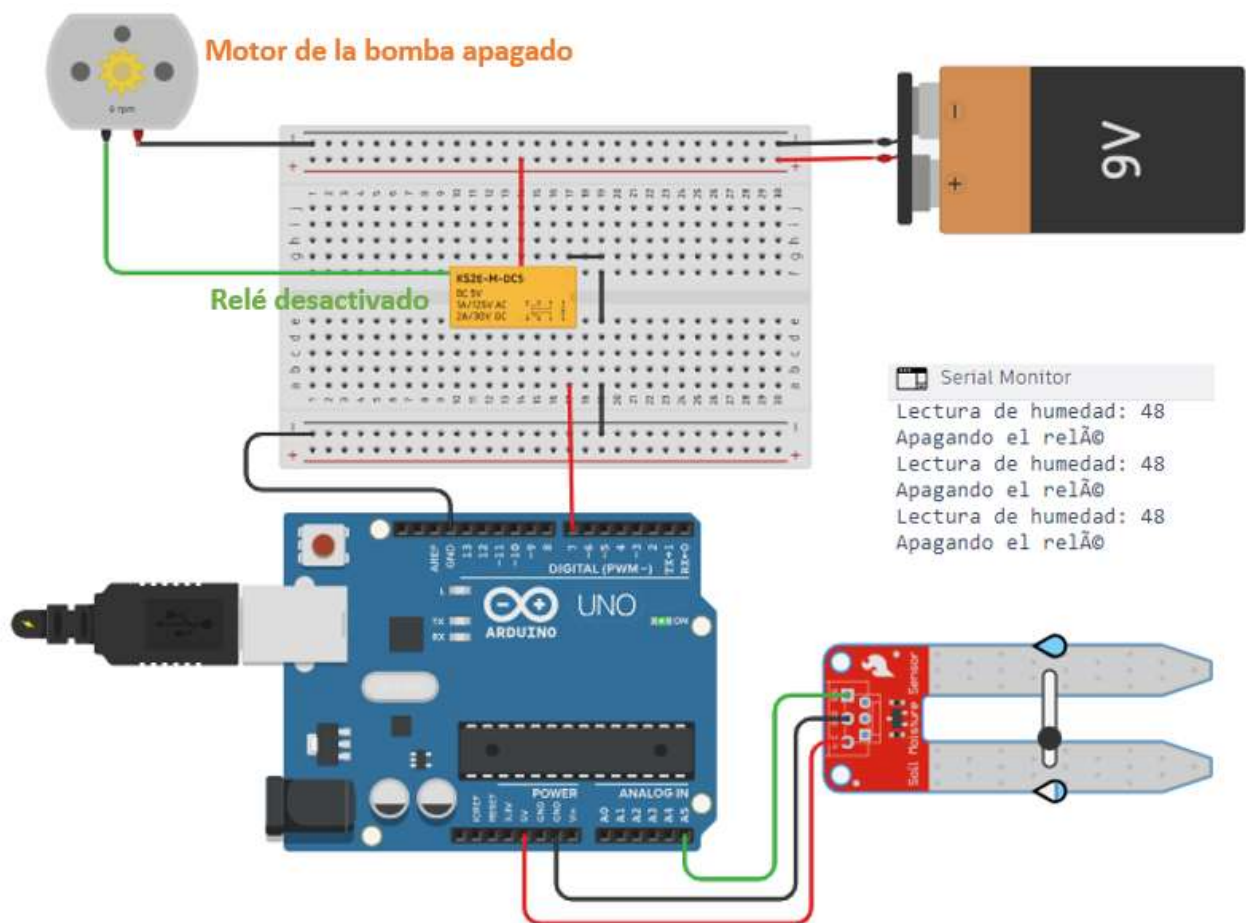
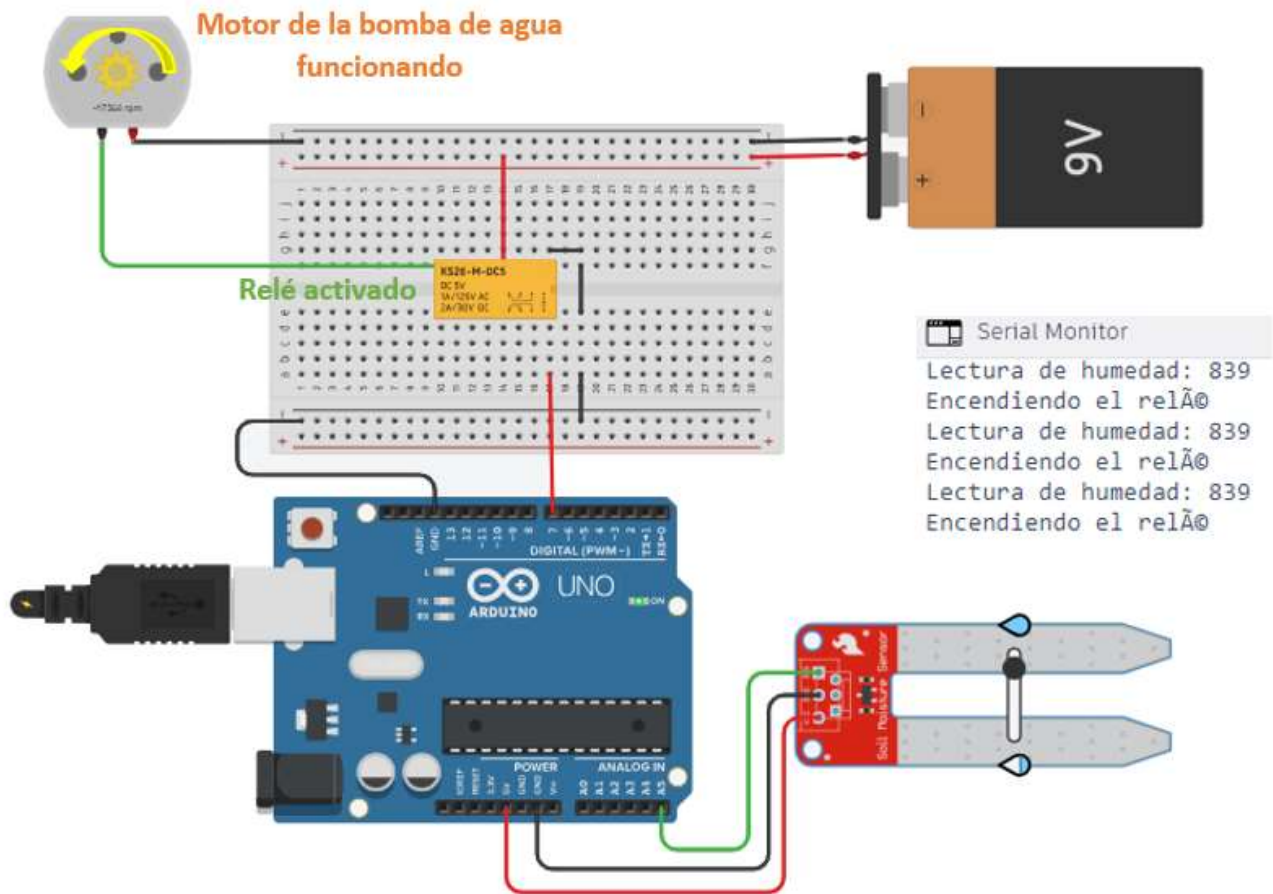
- **Definimos las constantes:** Esa parte del código define dos constantes importantes que se utilizarán en el programa para identificar los pines en el Arduino que están conectados al sensor de humedad y al relé. Estas líneas de código asignan nombres legibles a los pines utilizados para el sensor de humedad y el relé. Esto hace que el código sea más claro y fácil de entender, ya que podemos usar estos nombres en lugar de tener que recordar constantemente los números específicos de los pines. Es una buena práctica utilizar nombres descriptivos para los pines en lugar de números mágicos en el código.
- **función setup():** Esta parte del código se ejecuta una vez al principio cuando el microcontrolador se inicia. En esta parte del código, estamos configurando el pin del relé como una salida para que pueda controlar la activación y desactivación del relé. Además, estamos inicializando la comunicación serial a una velocidad de 9600 baudios para permitir la depuración y la impresión de información útil para el monitoreo del programa.
- **función loop():** que se ejecuta repetidamente después de que setup() se haya ejecutado al menos una vez. Este bloque de código lee el valor del sensor de humedad en el suelo, imprime la lectura en el monitor serial y luego decide si activar o desactivar el relé según el valor leído. Después de cada lectura y acción, hay un retardo de 1 segundo antes de realizar la siguiente lectura. Esto crea un ciclo continuo donde el sistema monitorea y controla el riego según la humedad del suelo.

El sistema de riego automatizado diseñado funcionaría de la siguiente manera:

El sistema de riego automatizado utiliza el sensor de humedad en el suelo para determinar si es necesario activar el riego. Cuando la humedad del suelo cae por debajo de un umbral específico (800 en este caso), el relé se activa, lo que a su vez activa el sistema de riego para suministrar agua a las plantas. Este proceso se repite en bucle, manteniendo el nivel de humedad adecuado en el suelo y garantizando el cuidado eficiente de las plantas.

- 1) **Lectura del Sensor:** En cada ciclo de ejecución, el sistema leería el valor de humedad del suelo utilizando un sensor de humedad conectado al pin analógico A5. El sensor proporciona una lectura analógica proporcional a la humedad del suelo, donde valores más bajos indican suelo húmedo y valores más altos indican suelo seco.
- 2) **Interacción con el Relé:** Una vez que se obtiene la lectura del sensor, el sistema verifica si el valor es mayor o igual a 800. Si el valor es alto (indicando suelo seco), el relé conectado al pin digital 7 se activa, permitiendo que fluya la corriente eléctrica hacia el sistema de riego.
- 3) **Control del Riego:** Al activarse el relé, se inicia el sistema de riego, suministrando agua a las plantas en función de la lectura del sensor. El relé se enciende durante un tiempo determinado, asegurando que las plantas reciban suficiente agua.
- 4) **Monitorización y Depuración:** Mientras el sistema funciona, el monitor serial se utiliza para mostrar información relevante. El valor leído del sensor se imprime en la pantalla, junto con mensajes que indican si el relé se enciende o apaga. Esto permite monitorear el comportamiento del sistema y asegurarse de que esté funcionando como se espera.
- 5) **Ciclo Continuo:** Después de cada ciclo de riego, el sistema espera un segundo utilizando la función delay(1000) antes de realizar la siguiente lectura del sensor. Este ciclo se repite de manera continua, asegurando que el sistema controle el riego en función de la humedad del suelo en intervalos regulares.





6. Problemática circuital y de programación. (Pensamiento Computacional - Método de Pólya)

Situación Problema: Control Inteligente de Iluminación para Habitaciones Eficientes

Descripción: En el marco del taller del primer ciclo de una escuela técnica, los estudiantes están explorando el emocionante mundo de la electrónica y la programación. Este proyecto desafía a los estudiantes a crear un sistema inteligente de iluminación para habitaciones, combinando la tecnología y la eficiencia energética. El objetivo es utilizar un Arduino para diseñar un sistema que controle automáticamente la iluminación en función de la presencia de personas en la habitación.

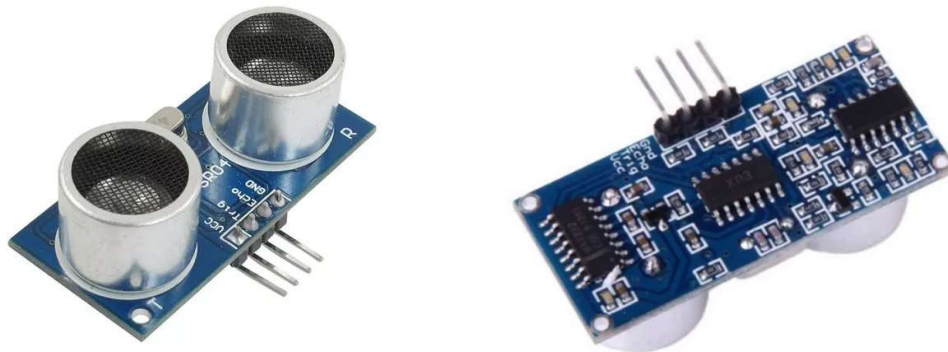
Desafío: Los estudiantes deben desarrollar un sistema que pueda detectar la presencia de alguien en la habitación y ajustar la iluminación en consecuencia. Deben emplear un sensor ultrasónico HC-SR04 conectado a un pin del Arduino para medir la distancia de los objetos. Además, deben utilizar un relé conectado a otro pin para encender y apagar una lámpara alimentada por una batería de 9 voltios.

Objetivos:

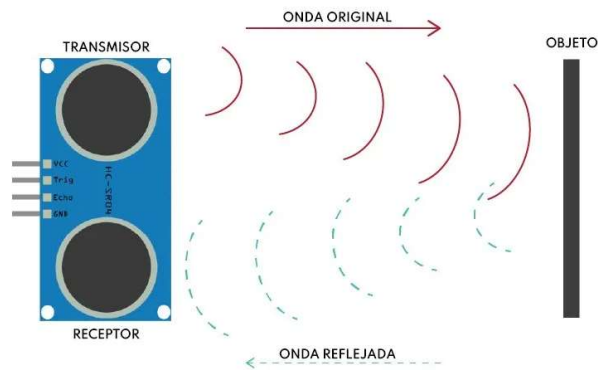
- Diseñar y conectar correctamente el sensor ultrasónico HC-SR04 al Arduino.
- Conectar el relé al pin adecuado para controlar la lámpara.
- Programar el Arduino para medir la distancia con el sensor ultrasónico y encender la lámpara si alguien se acerca a menos de 180 cm.
- Implementar un temporizador para apagar la lámpara si no hay movimiento detectado durante un período de 30 segundos.
- Aplicar los conocimientos en electrónica y programación para resolver un problema práctico y útil.
- Estimular la creatividad y el pensamiento crítico al diseñar una solución eficiente y automatizada.

Para tener en cuenta:

El sensor ultrasónico HC-SR04 es un dispositivo que se utiliza para medir distancias utilizando ondas ultrasónicas. Funciona emitiendo un pulso ultrasónico en el aire y luego calculando el tiempo que tarda en recibir el eco de ese pulso al rebotar en un objeto. Basándose en el tiempo que tarda el eco en regresar, el sensor puede estimar la distancia entre el sensor y el objeto.



Funcionamiento: El HC-SR04 consta de dos componentes principales: un transmisor ultrasónico y un receptor ultrasónico. El transmisor emite un pulso de ultrasonido, que viaja a la velocidad del sonido en el aire. Cuando este pulso encuentra un objeto, se refleja y regresa al sensor. El receptor del sensor detecta este eco ultrasónico y mide el tiempo que tarda en regresar.



Cálculo de la Distancia:

La velocidad del sonido en el aire es constante, por lo que utilizando la ecuación de distancia = velocidad × tiempo, el sensor puede calcular la distancia al objeto dividiendo el tiempo total entre dos (ya que el pulso viaja hacia el objeto y luego regresa). Esto proporciona una medida precisa de la distancia entre el sensor y el objeto.

Aplicaciones:

El sensor ultrasónico HC-SR04 tiene diversas aplicaciones en la electrónica y la automatización, como:

- **Medición de Distancias:** Puede utilizarse para medir la distancia a objetos cercanos o lejanos en proyectos como sistemas de detección de obstáculos, robots autónomos o sistemas de estacionamiento automático.
- **Sistemas de Alarma:** Puede detectar la presencia de objetos o personas en ciertas áreas, lo que lo hace adecuado para sistemas de seguridad y alarmas.
- **Navegación:** En proyectos de robótica, el sensor ultrasónico puede utilizarse para permitir que los robots eviten obstáculos y se desplacen de manera segura.
- **Automatización Residencial:** Se puede implementar en sistemas de iluminación o climatización que se adapten automáticamente a la presencia de personas en una habitación.
- **Control de Acceso:** Puede usarse para detectar personas o vehículos y activar dispositivos de control de acceso como puertas automáticas.

Conexiones:

VCC (Voltaje): Este pin se conecta a la fuente de alimentación (5V) del Arduino para proporcionar energía al sensor.

GND (Tierra): Este pin se conecta al GND (tierra) del Arduino para completar el circuito eléctrico.

Trig (Trigger): Este es el pin de entrada del sensor. Debes enviar un pulso corto (10 microsegundos) a este pin para iniciar la medición de distancia.

Echo: Este es el pin de salida del sensor. Proporciona un pulso de duración proporcional al tiempo que le lleva a la señal ultrasónica rebotar desde un objeto y regresar al sensor.

Aplicamos el pensamiento computacional para resolver el problema



Vamos a aplicar las herramientas del pensamiento computacional (descomposición, abstracción, patrones y algoritmo) para abordar la situación problema del "Control Inteligente de Iluminación para Habitaciones Eficientes":

Descomposición: Divide el problema en componentes más pequeños y manejables. En este caso:

- **Configuración del hardware:** Conectar el sensor ultrasónico y el relé al Arduino.
- **Lectura del sensor:** Programar el Arduino para medir la distancia con el sensor ultrasónico.
- **Control de la lámpara:** Programar el encendido y apagado de la lámpara mediante el relé.
- **Temporización:** Implementar un temporizador para apagar la lámpara después de cierto tiempo sin movimiento.

Abstracción: Identifica patrones comunes y simplifica detalles innecesarios. Por ejemplo:

- Enfócate en la lógica general del sistema y no en los detalles internos de los componentes.
- Trata la interacción del sensor y el relé como "entradas y salidas" sin preocuparte por cómo funcionan internamente.

Patrones: Busca soluciones o enfoques similares que hayan sido exitosos en problemas relacionados:

- **Establecer umbrales:** Usar un umbral de distancia (180 cm) para encender la lámpara.
- **Temporización:** Utilizar un temporizador para apagar la lámpara si no hay movimiento.

Algoritmo: Diseña una secuencia de pasos que resuelva el problema. Aquí tienes un ejemplo de algoritmo para controlar la iluminación:

- **Configuración:**
 - Conectar el sensor ultrasónico al Arduino.
 - Conectar el relé al Arduino.
 - Lectura del sensor:
 - Inicializar los pines.
 - Enviar un pulso desde el pin Trig del sensor.
 - Medir el tiempo de eco en el pin Echo.
 - Calcular la distancia basada en el tiempo de eco.
- **Control de la lámpara:**
 - Si la distancia es menor a 180 cm, encender el relé (y por ende, la lámpara).
 - Si no, apagar el relé (y la lámpara).
- **Temporización:**
 - Iniciar un temporizador cuando se enciende la lámpara.
 - Si el temporizador alcanza 30 segundos sin detección de movimiento, apagar el relé (y la lámpara).

Aplicamos el método de Pólya teniendo en cuenta las herramientas del pensamiento computacional



Vamos a aplicar el método de resolución de problemas de Pólya en conjunto con las herramientas del pensamiento computacional para abordar la situación problema del "Control Inteligente de Iluminación para Habitaciones Eficientes":

1. Comprender el Problema: La etapa de "Comprender el Problema" es el primer paso esencial en el proceso de resolución de problemas, según el método de Pólya. Implica tomar el tiempo necesario para entender completamente la naturaleza y los requisitos del problema que estás tratando de resolver. Cuando aplicamos las herramientas del pensamiento computacional en esta etapa estamos creando una base sólida y estructurada para avanzar hacia la siguiente etapa de "Elaborar un Plan". Esto te permitirá abordar el problema de manera sistemática y creativa, utilizando tus habilidades de pensamiento computacional para diseñar una solución efectiva.

- **Descomposición:** Identificamos los componentes y las acciones clave que se deben tomar para resolver el problema, como la detección de distancia y el control de la lámpara.
- **Abstracción:** Nos enfocamos en los resultados deseados y cómo los componentes interactúan, sin preocuparnos por los detalles técnicos internos.
- **Patrones:** Reconocemos que el problema involucra establecer umbrales y utilizar temporizadores.
- **Algoritmo:** Diseñamos un algoritmo de alto nivel que describa cómo los componentes interactuarán para lograr el control inteligente de la iluminación.

2. Elaborar un Plan: Durante esta etapa, se definen los pasos intermedios necesarios para llegar a la solución deseada. Esto implica diseñar un algoritmo o conjunto de acciones específicas que te llevarán desde el punto de inicio hasta el objetivo final. Se consideran herramientas disponibles, se aplican patrones previamente identificados y se anticipan desafíos potenciales. Al aplicar las herramientas del pensamiento computacional en esta etapa, estaremos diseñando una estrategia detallada y organizada para resolver el problema de manera efectiva y estructurada. Este plan nos guiará hacia la implementación exitosa de la solución.

- **Descomposición:** Dividimos la solución en pasos claros y manejables, como inicialización, medición de distancia, evaluación de la presencia, control de la lámpara y temporización.
- **Abstracción:** Nos enfocamos en los pasos y la lógica general de cómo los componentes interactúan.
- **Patrones:** Aplicamos el patrón de establecer umbrales para determinar la presencia basada en la distancia medida.
- Utilizamos el patrón de temporización para apagar la lámpara después de un período sin movimiento.
- **Algoritmo:** Diseñamos una serie de pasos secuenciales y detallados que guiarán la ejecución de la solución al problema identificado. Estos pasos definen cómo se utilizarán los componentes, cómo se llevarán a cabo las acciones y cómo se logrará el objetivo final. El algoritmo se crea con la intención de proporcionar una estructura clara y ordenada para la implementación de la solución:
 - **Inicialización:**
 - Configura los pines del sensor ultrasónico y el relé en el Arduino.
 - Configura el temporizador para el apagado automático.
 - **Medición de Distancia:**
 - Enviamos un pulso ultrasónico y medimos el tiempo de eco.
 - Calculamos la distancia utilizando la fórmula de conversión.
 - **Evaluación de la Presencia:**
 - Si la distancia es menor al umbral (180 cm), alguien está presente.
 - **Control de la Lámpara:**
 - Si hay presencia, activa el relé para encender la lámpara.
 - Si no hay presencia, apaga el relé para apagar la lámpara.
 - **Temporización:**
 - Si la lámpara está encendida debido a la presencia, inicia el temporizador.
 - Si el temporizador alcanza 30 segundos sin presencia, apaga la lámpara.

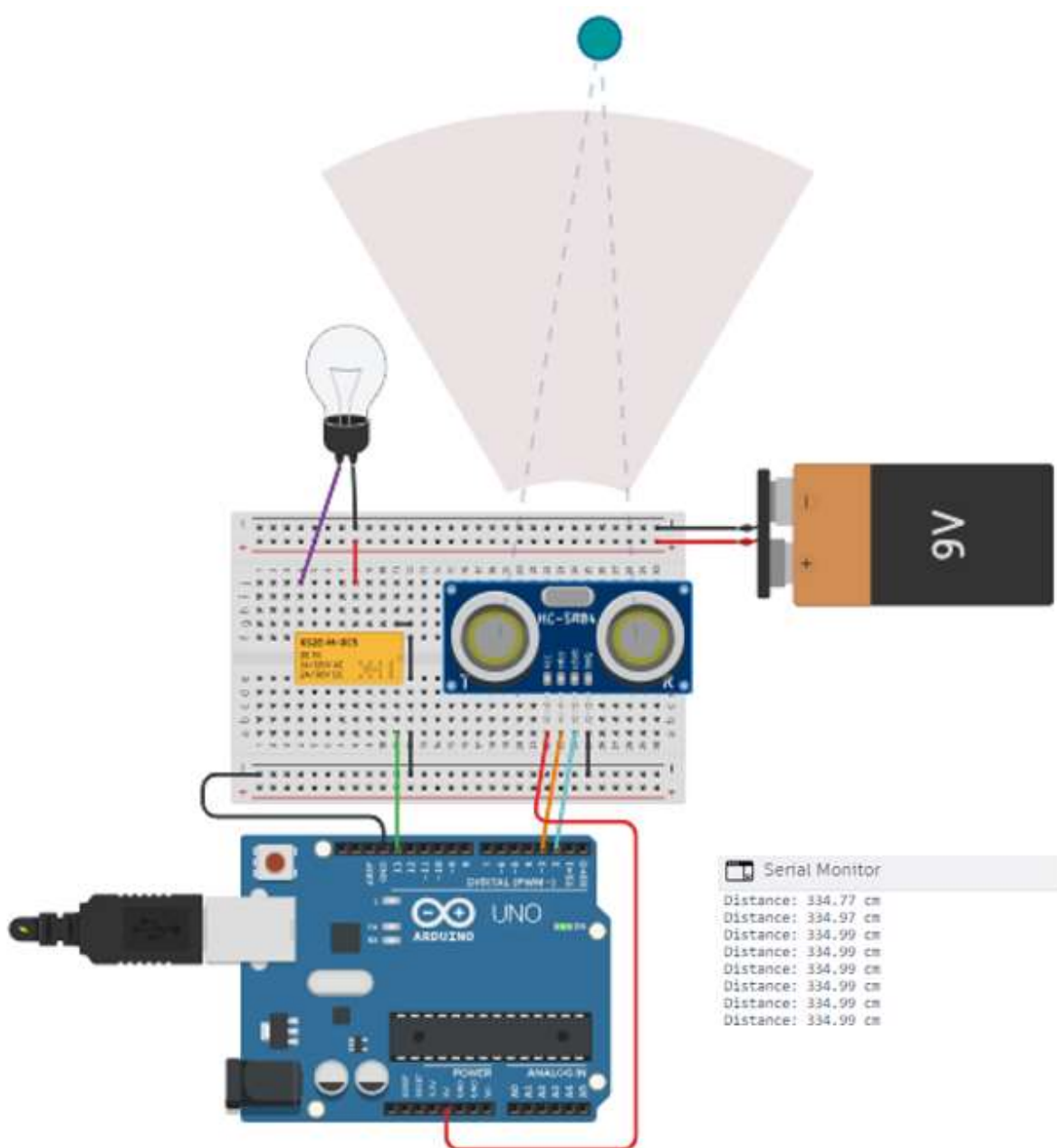
3. Ejecutar el Plan:

- Configura físicamente el hardware siguiendo el plan previamente diseñado.
- Codifica el algoritmo en el entorno de programación del Arduino.
- Asegúrate de que los pines y las conexiones sean correctos.

4. Revisar y Verificar la Solución:

- Realiza pruebas exhaustivas con diferentes escenarios: presencia, no presencia, distancia variable, etc.
- Observa si la lámpara se enciende y apaga correctamente según las condiciones.
- Verifica si la temporización funciona como se esperaba, apagando la lámpara después de 30 segundos sin movimiento.
- Asegúrate de que no haya errores en el código y que todos los componentes funcionen correctamente.

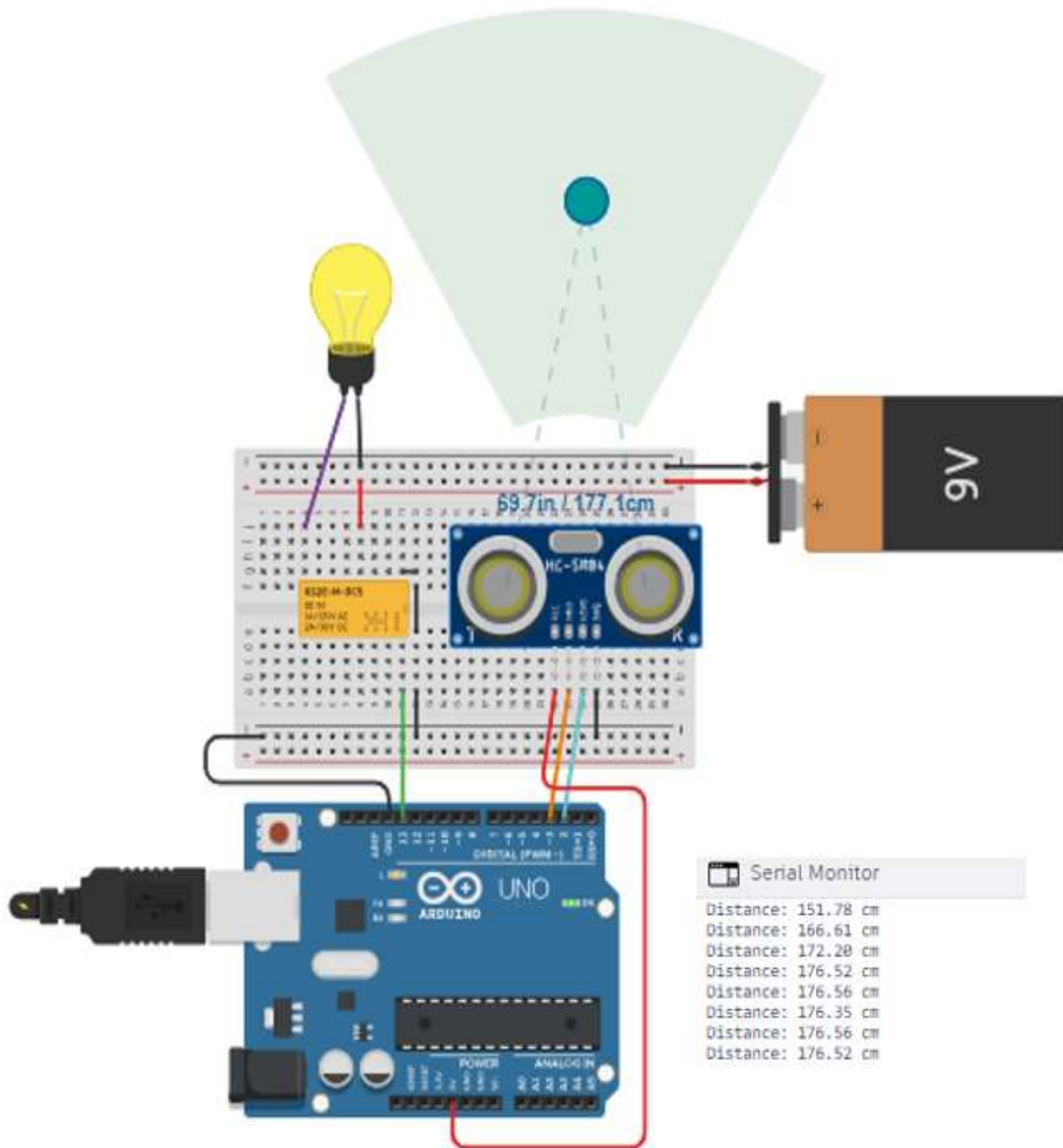
Cuando el sensor ultrasónico no detecta un objeto porque está fuera del rango por encima de 180 cm, el sistema interpretará esto como ausencia de presencia y apagará la lámpara.



Cuando el sensor ultrasónico detecta un objeto dentro del umbral de los 180 cm, el sistema interpretará esto como la presencia de alguien en la habitación. Como resultado, la lámpara se encenderá automáticamente para proporcionar iluminación.

El monitor serial mostrará la distancia medida en centímetros con el mensaje "Distance:", seguido del valor de la distancia en centímetros.

La información en el monitor serial te permitirá verificar el funcionamiento del sistema, la detección de presencia y la activación/desactivación de la lámpara según las condiciones establecidas en el programa.



En el siguiente código, podemos observar como se realizan mediciones de distancia utilizando un sensor ultrasónico y se controla una lámpara mediante un relé. La distancia medida se muestra en el monitor serial, y la lámpara se enciende si alguien está a menos de 180 cm de distancia.

Este programa demuestra la aplicación de conceptos de electrónica y programación para crear un sistema de iluminación eficiente y automatizada. El uso del sensor ultrasónico y el relé permite una respuesta inteligente a la presencia en la habitación, contribuyendo a la eficiencia energética y la comodidad.

```
const int trigPin = 3;      // Pin para enviar el pulso ultrasónico
const int echoPin = 2;     // Pin para recibir el eco ultrasónico
const int relePin = 13;    // Pin para controlar el relé que enciende la lámpara

void setup() {

  Serial.begin(9600); // Inicia la comunicación serial a una velocidad de 9600 baudios
  pinMode(trigPin, OUTPUT); // Configura el pin trigPin como salida para enviar el pulso ultrasónico
  pinMode(echoPin, INPUT); // Configura el pin echoPin como entrada para recibir el eco ultrasónico
  pinMode(relePin, OUTPUT); // Configura el pin relePin como salida para controlar el relé
}

void loop() {
  // Prepara el sensor ultrasónico para medir la distancia
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Mide el tiempo que tarda el eco ultrasónico en regresar
  long duration = pulseIn(echoPin, HIGH);

  // Calcula la distancia en centímetros usando la velocidad del sonido
  float distance_cm = (duration / 2.0) * 0.0343;

  // Muestra la distancia medida en el puerto serial
  Serial.print("Distance: ");
  Serial.print(distance_cm);
  Serial.println(" cm");

  // Comprueba si la distancia es menor que 180 cm
  if (distance_cm < 180) {
    // Enciende el relé para activar la lámpara
    digitalWrite(relePin, HIGH);
  } else {
    // Apaga el relé para apagar la lámpara
    digitalWrite(relePin, LOW);
  }

  // Espera 1 segundo antes de repetir el ciclo
  delay(1000);
}
```

https://www.tinkercad.com/things/ece9gU6zdVW?sharecode=AJQMTSeaDs5IbSIXQ7gcH4v0UV_EWBqAdyc35ZgKpb0

Funcionamiento del sistema:

El sistema utiliza el sensor ultrasónico para medir la distancia, lo que se traduce en la detección de presencia en la habitación. Dependiendo de si alguien está cerca o no, el relé controla la lámpara para encenderla o apagarla. Además, un temporizador asegura que la lámpara se apague si no se detecta movimiento durante un período de tiempo específico. De esta manera, el sistema logra un control eficiente y automatizado de la iluminación en función de la presencia en la habitación.

1. Configuración Inicial:

- Se definen constantes para los pines utilizados en el Arduino: **trigPin**, **echoPin** y **relePin**.
- En la función **setup()**, se inicia la comunicación serial a **9600 baudios**, configurando los pines como entradas o salidas necesarios.

2. Loop Principal:

- En el bucle **loop()**, el sistema se ejecuta repetidamente para monitorear la presencia en la habitación y controlar la lámpara en consecuencia.

3. Medición de Distancia:

- Se inicia una medición de distancia con el sensor ultrasónico: se envía un pulso ultrasónico breve desde el pin **trigPin** y se mide el tiempo que tarda en regresar el eco al pin **echoPin**.

4. Cálculo de Distancia:

- Utilizando la velocidad del sonido, se calcula la distancia en centímetros a partir del tiempo de eco medido. El resultado se almacena en la variable **distance_cm**.

5. Visualización de la Distancia:

- La distancia medida se muestra en la consola serial mediante los comandos **Serial.print()** y **Serial.println()**. Esto permite verificar la detección y medición.

6. Control de la Lámpara:

- Se verifica si la distancia medida (representando la presencia) es menor que 180 cm.
- Si alguien está dentro de ese rango, el relé conectado al **reléPin** se activa mediante **digitalWrite()** para encender la lámpara.

7. Temporización:

- Si la lámpara se enciende debido a la presencia, se inicia un temporizador que cuenta 30 segundos mediante el uso de la función **delay()**.
- Si no se detecta movimiento y el temporizador llega a 30 segundos, el relé se desactiva para apagar la lámpara.

8. Repetición del Ciclo:

- Después de completar un ciclo, se espera un segundo (**delay(1000)**) antes de que el **bucle loop()** vuelva a ejecutarse, permitiendo que el sistema monitoree constantemente la presencia y ajuste la iluminación según sea necesario.

Situaciones Problemáticas Tecnológicas. “Problemática Abierta” Metodología Proyectual

Proyecto Tecnológico

Desde el punto de vista metodológico, existen una gran cantidad de herramientas procedimentales que se utilizan a la hora de trabajar en el taller del ciclo básico de nuestras escuelas técnicas; pero, la que encierra mayor nivel de integración operacional, es la que denominamos “Proyecto Tecnológico”, a partir del cual es posible resolver problemas de manera eficiente y ordenada, desde un encuadre sistémico. Gran parte de la riqueza formativa de un proyecto tecnológico se halla en los modos y formas en el cómo los alumnos y alumnas llevan adelante las diferentes actividades necesarias para su concreción.

Para poder entender el concepto de Proyecto Tecnológico, solo nos basta con desglosar el término y desarrollar los conceptos de las dos palabras que lo forman:

Proyecto: Plan de actividades planificadas y ejecutadas que se relacionan entre sí para cumplir un objetivo que se ha fijado previamente.

Todo proyecto debe integrar una serie de elementos que lo configuran. Entre esos elementos cabe destacar:

- **Tiempo:** El periodo que va a durar el proyecto.
- **Coste:** El coste de los recursos necesarios para sacar el proyecto.
- **Organización y planificación:** Las actividades a desarrollar, así como los plazos, para lanzar el proyecto.
- **Comunicación:** Cómo se va a desarrollar la comunicación, así como los flujos de información.
- **Asignación de tareas:** Reparto de las tareas y funciones.
- **Resultados:** La medición de los objetivos que esperamos obtener del proyecto.

Tecnológico: Es un término que se vincula a la palabra tecnología. Tecnología es un término de uso cotidiano que puede tener muchas acepciones y siempre puede abarcar más. Si recopilamos todos los usos habituales de este término, nos encontraríamos con un largo listado en el que sería posible identificar algunos rasgos coincidentes:

- La tecnología da respuesta a necesidades y a demandas sociales; resuelve problemas.
- La tecnología se plantea como una actividad creativa.
- La tecnología aparece frecuentemente vinculada con artefactos.
- La tecnología se asocia con la técnica y con la ciencia.
- La tecnología se despliega en ciclos de innovación.
- La tecnología abarca conocimientos complejos.

En relación a estos rasgos coincidentes podemos caracterizar a la tecnología como una actividad social, la cual brinda respuestas creativas a demandas o necesidades, a través de la asociación de la ciencia, la técnica, los conocimientos complejos y ciclos de innovación.

El Método de la Tecnología

La tecnología adopta un esquema de trabajo basado en el método proyectual, el cual posee diferentes etapas o fases para su concreción.

- Percepción del problema
- Búsqueda de alternativas de solución
- Selección de la solución adecuada
- Diseño de la solución
- Construcción de modelos
- Evaluación y perfeccionamiento

Por fines educativos y teniendo en cuenta las características de las y los estudiantes del primer ciclo, esta metodología proyectual, tomara otra forma. La fase o etapa denominada “Percepción del problema”, tomara una rol más orientativo definido por los contenidos, las expectativas de logro y las capacidades que las y los jóvenes deben desarrollar. Esta fase se redefinirá en dos fases simultáneas “Diseño de la situación Problemática” y “Planteo de la problemática”.

- Diseño de la situación Problemática.
 - Planteo de la Problemática
 - Búsqueda de alternativas de solución
 - Selección de la solución adecuada
 - Diseño de la solución
 - Construcción de modelos
 - Evaluación y perfeccionamiento
- } Percepción del problema

El proyecto tecnológico será el punto de convergencia de los distintos saberes que las y los estudiantes desarrollen durante su trayectoria por el taller del primer ciclo. Las etapas o fases que componen la metodología proyectual nos garantizan esa convergencia a través de las acciones que requieren su desarrollo: Saberes relacionados con la búsqueda, selección, clasificación y producción de la información tecnológica, el diseño, la aplicación de los sistemas, la organización del trabajo, los procesos de fabricación, el uso correcto de las máquinas y herramientas mancomunadas a estos procesos, las normas de seguridad e higiene, incluso el análisis crítico del desarrollo tecnológico actual y sus implicancias en nuestra sociedad y en el mundo del trabajo.

Esta convergencia promueve la creatividad, la construcción, la revalorización del error en el proceso de aprendizaje.

El proyecto dará como resultado un prototipo diseñado y fabricado por las y los estudiantes, el cual será un primer modelo que represente o simule el producto y les permita, verificar el diseño y confirmar que cuenta con las características específicas planteadas a las y los jóvenes para resolver el problema.

A fines orientativos los invitamos a recorrer una breve descripción de la metodología proyectual orientada a proyectos de primer ciclo de nuestras escuelas técnicas y agrotécnicas:

- **El diseño de la problemática**

Esta es la primera instancia donde podemos pensar la articulación de los saberes desarrollados en el taller, e incluso con aquellos espacios de la formación general y científico tecnológico (Cuantos más espacios formativos trabajen en el diseño y especificación de la situación problema, más espacios formativos serán los que contribuyan en su resolución por parte de los y las jóvenes) y pondrá en valor estos espacios. Pensar la problemática no es tarea fácil ya que es parte de una construcción colectiva, y como toda construcción en este sentido se debe realizar a través de consensos y resultantes, si nos limitamos a la asociación de los módulos, áreas o espacios que conforman el taller del primer ciclo, los y las docentes deberán diseñar una situación problema que no se desvíe de la transmisión de saberes en base a los contenidos y expectativas de logro y con el desarrollo de capacidades específicas básicas y si la problemática pretende abarcar los espacios de la formación general y específica, la misma debe estar asociada a los NAP (Núcleos de Aprendizaje Prioritarios) de cada materia involucrada. Es sabido que en la enseñanza de la educación técnica, y más aún en el taller existe un consenso generalizado en relación a la idea de los estudiantes como creadores e investigadores, de que el aprendizaje tecnológico es más efectivo a medida que los y las estudiantes participan activamente en procesos de investigación y prototipado, orientados por el o la docente, pero centrados en el tratamiento de situaciones problemáticas abiertas, nuevas, que las y los estudiantes consideren útiles y que no sean resolubles mediante la aplicación de rutinas preestablecidas, que generan el interés de los y las estudiantes, y que integren la teoría con la práctica en un proceso único de construcción del conocimiento.

Por eso, es de especial interés pensar que tipos de problemas se deben diseñar y plantear, ya que no es lo mismo formular un problema que propicie en los estudiantes un proceso de investigación o reflexión, esquivando una respuesta memorística y única y que cuestione sus concepciones al respecto, que plantear un problema que propicie una respuesta mecánica, aleatoria y con un mínimo nivel de movilización o cuestionamiento de sus concepciones.

Lo ideal sería que los y las estudiantes a medida que vayan adentrándose en la especialidad vayan desarrollando la capacidad de poder percibir los problemas vinculados a su perfil profesional, pero esa construcción debe ser paulatina y en el caso de los y las estudiantes del primer ciclo es muy importante el diseño de un enunciado que contextualice la situación problema de su entorno, para que ellos puedan apropiarse del problema significativamente.

La acción comienza pensando una situación problema para los estudiantes. Diseñar una situación problema tecnológica para estudiantes del primer debe ser un proceso colaborativo entre el grupo de docentes. Esto asegura que la actividad sea completa, efectiva y beneficiosa para los estudiantes, permitiéndoles desarrollar habilidades técnicas, pensamiento crítico y creatividad mientras abordan un problema del mundo real.

Aquí te proporciono una secuencia de pasos que podrían seguir para comenzar la acción de diseñar la situación problema:

- **Reunión Inicial:** El grupo de docentes de taller involucrados se reúne para discutir y planificar el diseño de la situación problema. En esta reunión, se establece el objetivo general del ejercicio, que es desarrollar una situación problemática enriquecedora y desafiante para los estudiantes.
- **Definir Objetivos de Aprendizaje:** Juntos, determinan los objetivos de aprendizaje que quieren lograr con esta situación problema. Estos objetivos deben estar alineados con el currículo y las capacidades que se espera que los estudiantes desarrollen en esta etapa de su formación.
- **Seleccionar un Tema y Contexto:** El grupo elige un tema tecnológico relevante y apropiado para el nivel de las y los estudiantes. Los docentes crean una descripción detallada de la situación problema. Esto podría incluir el contexto en el que se encuentra el problema, los recursos disponibles, los objetivos que deben cumplirse y posiblemente algunas restricciones o limitaciones.

- **Planteo de la Problemática**

Una vez diseñada la problemática, las o los docentes deben plantearse a las y los estudiantes de la forma más abierta posible. La descripción del problema será realizada de modo tal que pueda ser considerada desde diferentes perspectivas. Esto se refiere a que los y las estudiantes tengan una percepción global en relación al problema como etapa previa a la búsqueda de una solución.

- **Búsqueda de alternativas de solución**

En esta etapa se requiere utilizar y confrontar los aspectos de la situación abordados en la etapa anterior, con la búsqueda de información ampliatoria, de modo tal que sea posible generar distintas alternativas de solución.

Si bien los docentes deben hacer un andamiaje durante todo el proceso. En esta etapa será de suma importancia sus estrategias de mediación para lograr que los jóvenes puedan consensuar en una propuesta de solución lo más colectiva posible y que este dentro de su alcance.

Es muy probable que la construcción colectiva de una solución los y las estudiantes de primer ciclo no logren ponerse de acuerdo fácilmente, ya que no tienen plenamente desarrollada, la capacidad de trabajar con otros para un fin compartido. Los y las docentes deberán plantear acuerdos y metas para que alumnos y alumnas puedan trabajar coordinadamente a fin de lograr un objetivo compartido, realizando un esfuerzo de concertación para llegar a metas comunes, formas de trabajo y mecanismos para regular sus propios comportamientos, construyendo de este modo un ambiente de trabajo inclusivo, activo y armónico

Si bien son los y las estudiantes los que realizan las investigaciones que los acerquen a la solución deseada, el o la docente debe orientarlos a encontrar fuentes fidedignas y pertinentes que les permitan satisfacer su necesidad de información, permitiendo que desarrollen sus propios criterios de validación.

Los estudiantes deben entender que trabajar en equipo no es estar reunidos en un espacio, en un mismo momento; sino que es compartir ideales, formas de trabajo e intereses, contando con un propósito común en el que cada estudiante realice su aporte.

Los docentes además de tener claridad sobre el concepto de trabajo en equipo, lo deberán proyectar a sus estudiantes de manera sencilla y clara, apoyándose en el concepto de que el hombre es un ser social y por tal motivo el aprendizaje depende en gran parte del prójimo, porque a través de él, logra la comunicación, el intercambio de ideas, y la construcción de conocimiento.

Esta etapa del método proyectual implica la necesidad, que los estudiantes adquieran saberes relacionados con la búsqueda y análisis de información. Hoy en día, uno de los espacios más comunes para la búsqueda de información es internet ya que entre otras cosas es un gran reservorio de información. Y para poder acceder a ese cúmulo de información es necesario, además del acceso a equipos y a Internet, que los jóvenes adquieran ciertas estrategias que permitan:

- Encontrar la información que están buscando, y no otra.
- Determinar si la información que encontraron es pertinente y confiable.

Estas dos estrategias podrían parecer sencillas, pero no lo son. Sobre todo para los adolescentes, quienes a través de las diferentes experiencias que transitan en la escuela secundaria están construyendo su sentido crítico sobre la información y las ideas. Por tal razón la búsqueda de información y sus técnicas, estrategias y metodologías deben desarrollarse desde los primeros años de su formación.

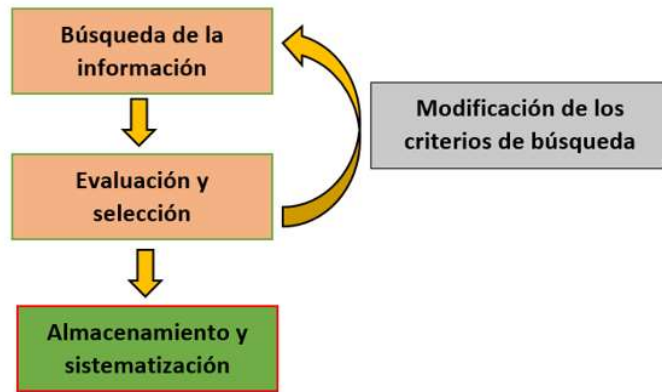
La cantidad y variedad de información disponible en Internet determina la necesidad de contar con ciertas herramientas para obtener información que resulte significativa, es decir, útil, relevante y confiable. Para ello es necesario que al iniciar un proceso de búsqueda alumnos y alumnas, consideren los siguientes aspectos:

- Conocimiento de los recursos involucrados: es decir, características de la red, programas de navegación, de administración de archivos y otras utilidades.
- Conocimiento de los sitios de búsqueda y sus respectivas estrategias de búsqueda.

Los procesos de búsqueda de información son complejos y cíclicos, e implican una serie de actividades tales como:

- a) Búsqueda, evaluación y selección de la información.
- b) Almacenamiento de resultados parciales.
- c) Comparación y análisis de la información obtenida.
- d) Modificación de los criterios de búsqueda: ampliar, especificar o redefinir los criterios.

Si bien estas acciones se realizan en paralelo, en la siguiente imagen se procurará realizar una secuencia de pasos que permita analizar cada uno de sus componentes y definir una secuencia posible para su enseñanza.



- **Selección de la solución adecuada**

En esta instancia los estudiantes tendrán que seleccionar la solución que mejor cumpla con la problemática planteada por el o los docentes.

En esta etapa predomina la ubicuidad. Estamos hablando de la búsqueda de la solución que pueden apropiarse. La solución debe ser posible de realizar con los recursos que tenemos o que podemos llegar a tener.

En la selección de la solución, los docentes deberán realizar formulaciones a través de diálogos, realizando preguntas al grupo en relación a las características del prototipo que van a fabricar para dar respuesta al problema planteado.

- **La etapa de diseño**

El diseño es la parte creativa del proyecto tecnológico, porque es aquí donde, a partir de la información que lo/as jóvenes han consultado y del conocimiento de la situación en la realidad, realizaran su propia respuesta al problema. Involucra aprendizajes personales y colectivos y requiere de métodos tales como: croquis, bocetos y planos; cálculos y estimaciones (de costos, de magnitudes físicas, etc.); la elaboración de planes de acción, etc.

- **Organización del trabajo**

Una vez que los y las estudiantes han optado por una alternativa y realizado el diseño, antes de comenzar a trabajar en la fabricación, es conveniente que dediquen un tiempo a pensar en cómo organizar y sistematizar las tareas que les permitirán llevar adelante la solución seleccionada. Esto quiere decir, planificar el modo que se llevara a cabo, detallando todas las tareas a realizar, su secuencia, el tiempo estimado, preverán los recursos necesarios para la ejecución del proyecto y la distribución y asignación de estas tareas entre los integrantes del equipo de trabajo.

- **Construcción de modelos**

Durante esta etapa los estudiantes deberán seleccionar y utilizar los materiales, herramientas, sistemas, máquinas, instrumentos y distintos procedimientos, planificando su aplicación eficiente. Explicar a terceros cómo se desarrolla el trabajo. Aplicar técnicas manuales y técnicas digitales de fabricación. Seleccionar caminos alternativos cuando aparezcan dificultades y solicitar ayuda cuando la necesiten.

- **La etapa de evaluación y perfeccionamiento**

En esta etapa, se revisará todo el proceso que llevo a la fabricación del prototipo, se compararán el resultado obtenido con los objetivos iniciales.

Probarán su funcionamiento e incluso podrán sugerir cambios y mejoras en cada una de las fases anteriores. Analizarán las consecuencias deseadas y las no deseadas. Entre el diseño que anticipa cómo será el producto y el producto terminado suele haber diferencias, lo que podrá deberse tanto a errores en la concreción del diseño como a modificaciones que se han detectado como necesarias durante el proceso de construcción. Esta etapa será útil, para sugerir cambios y mejoras en cada una de las fases anteriores y analizar las consecuencias deseadas y las no deseadas.

7. Situación Problema Tecnológica. “Problemática Abierta” (Metodología Proyectual)

Título de la problemática: Invernadero automatizado

Descripción: La Cooperativa Agropecuaria "VerdeSur", situada en una pintoresca ciudad del sur de la provincia de Buenos Aires, se enfrenta a un reto fundamental para mejorar sus prácticas agrícolas y maximizar la producción de cultivos: la necesidad de optimizar el ambiente en sus invernaderos. Con el objetivo de modernizar sus técnicas y reducir su impacto ambiental, la cooperativa busca implementar un sistema innovador que permita controlar la temperatura y la humedad del suelo de manera automatizada y en tiempo real. Esto no solo garantizará un entorno propicio para el crecimiento de los cultivos, sino que también posibilitará una gestión eficiente de los recursos hídricos y energéticos. El prototipo debe ser controlado por un Arduino y debe contar con las siguientes características:

Medición y Visualización de Temperatura: Uno de los aspectos cruciales para el desarrollo saludable de los cultivos es la regulación de la temperatura dentro de los invernaderos. Para abordar este desafío, el equipo de estudiantes de la escuela técnica se enfocará en desarrollar un sistema de medición y visualización de temperatura preciso y automatizado. Se utilizarán sensores de temperatura ubicados dentro del invernadero para recolectar datos en tiempo real. Estos sensores enviarán señales eléctricas proporcionales a la temperatura, lo que permitirá calcular la temperatura en grados Celsius. La información recopilada se visualizará en un display LCD instalado en un lugar conveniente, brindando a los agricultores una instantánea clara y rápida de la temperatura actual.

Medición y Visualización de la Humedad en Tierra: La humedad del suelo es un factor esencial que impacta directamente en la salud y el rendimiento de los cultivos. El equipo de estudiantes también abordará este aspecto mediante la implementación de un sistema de medición y visualización de la humedad en el suelo. Para lograrlo, se emplearán sensores de resistencia de humedad que detectarán los niveles de humedad en la tierra circundante. Estos sensores medirán la resistencia eléctrica del suelo, que varía en función de la humedad presente. Se establecerán umbrales específicos para determinar si el suelo está seco o húmedo según los valores de resistencia medidos. Los datos de humedad se mostrarán junto a los datos de temperatura en el display LCD, proporcionando a los agricultores una visión completa de las condiciones ambientales en el invernadero.

Pensamiento computacional aplicado a la programación del prototipo:



Descomposición: Dividimos el proyecto en tareas más pequeñas y manejables. Por ejemplo: lectura de sensores, toma de decisiones, control de actuadores, visualización en el LCD, etc.

- Funciones para leer sensores (temperatura y humedad en el suelo).
- Función para tomar decisiones sobre riego y ventilación.
- Funciones para controlar los actuadores (bomba y ventilador).
- Función para mostrar datos en el display LCD.

Abstracción: Creamos funciones que realicen tareas específicas. Por ejemplo, funciones para leer temperatura, leer humedad, tomar decisiones sobre riego y ventilación, controlar actuadores, etc. En lugar de preocuparnos por cómo

funcionan internamente los sensores y los actuadores, nos concentramos en cómo utilizarlos y en cómo afectan el sistema general.

- Utilizamos funciones para ocultar los detalles de cómo se leen los sensores y se controlan los actuadores.
- Nos enfocamos en la funcionalidad general de cada función, no en su implementación interna.

Patrones: Definimos patrones claros para activar/desactivar el riego y el ventilador basados en las lecturas de temperatura y humedad en el suelo. Implementamos un patrón que lee los valores analógicos de los sensores y los convierte en lecturas significativas.

- Implementamos el patrón de control de umbral para decidir cuándo regar el suelo y cuándo activar el ventilador.
- Seguimos el patrón de lectura de sensores analógicos para obtener lecturas significativas.

Algoritmo: Diseñamos un algoritmo que lee los valores de los sensores de temperatura y humedad en el suelo. Convertimos estos valores en lecturas legibles y comprensibles para el sistema. Definimos un algoritmo que toma decisiones basadas en las lecturas de los sensores. Esto incluye verificar si es necesario regar el suelo o activar el ventilador.

- Algoritmo de lectura de sensores: Lee los valores analógicos de los sensores, realiza cálculos y convierte esos valores en lecturas legibles.
- Algoritmo de toma de decisiones: Basado en las lecturas de los sensores, decide si es necesario regar o ventilar y controla los actuadores en consecuencia.

Diseño y fabricación de un prototipo de Invernadero automatizado para la Cooperativa Agropecuaria "VerdeSur"

- **Diseño de la problemática**

En el caso de este ejemplo, la dará respuesta a una iniciativa de automatización de un invernadero automatizado para la Cooperativa Agropecuaria "VerdeSur", ubicada en una ciudad del sur de la provincia de Buenos Aires, la cual está en busca de soluciones innovadoras para mejorar sus prácticas agrícolas.



La problemática debe estar basada en la simulación de una situación real, y su enunciado tendrá la función de contextualizar el problema o la demanda y vincular a las y los jóvenes con el medio donde se desarrollaran como futuros profesionales. Si bien tendrá un carácter lúdico desde el rol profesional, será una forma para la construcción y desarrollo de capacidades profesionales básicas desde los primeros años de su formación.¹

La etapa de diseño de la problemática es esencial para la formulación de una situación desafiante que inspire a los estudiantes a abordar con entusiasmo los conceptos clave y desarrollar habilidades prácticas en el contexto de la automatización de un invernadero.

¹ Ver Capacidades Básicas y Profesionales Básicas: ANEXO II Capacidades básicas, capacidades profesionales básicas y capacidades profesionales específicas (Página 50 a 58): <http://www.inet.edu.ar/wp-content/uploads/2015/11/Evaluacion-de-Capacidades-Profesionales.pdf>.

Paso 1: Reunión Inicial. Los docentes del taller se reúnen para colaborar en la planificación y el diseño de una problemática significativa. Definen el objetivo principal: diseñar un prototipo de invernadero automatizado que incorpore tecnologías de automatización y control para mejorar los métodos tradicionales de producción agropecuaria.

Paso 2: Objetivos de Aprendizaje. En equipo, determinan los objetivos educativos que se desean lograr con esta problemática. Los objetivos deben reflejar la adquisición de habilidades técnicas, la aplicación práctica de conocimientos, el desarrollo de capacidades, la promoción de la colaboración y la innovación entre los estudiantes.

Paso 3: Selección del Contexto y Tema. Eligen el contexto de un "Invernadero automatizado" y unifican los elementos clave: medición de temperatura y medición de la humedad en tierra. La problemática se basa en la necesidad de integrar tecnologías para mejorar la eficiencia en la producción.

Paso 4: Enunciado de la Problemática. "Como estudiantes del primer ciclo del taller de la escuela técnica, se les encomienda la emocionante tarea de diseñar y construir un prototipo maqueterizado de un invernadero que dé respuesta a la necesidad de una cooperativa agropecuaria que está en busca de soluciones innovadoras para mejorar sus prácticas agrícolas. El sistema estará orientado a controlar la temperatura y la humedad del suelo en los cultivos, y a tomar decisiones en tiempo real para mejorar el riego y la ventilación de los invernaderos. Los estudiantes se sumergirán en la combinación de electrónica, programación y soluciones prácticas para la agricultura moderna.

Beneficios Esperados:

- **Adquisición de Habilidades Técnicas:** Los estudiantes obtendrán habilidades prácticas en áreas clave de la tecnología, como la electrónica y la programación. Aprenderán a conectar componentes, programar microcontroladores y diseñar sistemas funcionales, lo que les proporcionará una base sólida para futuras exploraciones tecnológicas.
- **Aplicación Práctica del Conocimiento:** Los estudiantes verán cómo los conceptos aprendidos en clase pueden aplicarse directamente a situaciones del mundo real. Esto les ayudará a comprender mejor la relevancia de su educación y cómo pueden utilizar sus habilidades para abordar desafíos concretos.
- **Comprensión de la Agricultura Moderna:** Al abordar los desafíos específicos de una cooperativa agropecuaria, los estudiantes obtendrán una comprensión más profunda de la agricultura moderna y la importancia de la tecnología en la mejora de los métodos tradicionales de producción.
- **Innovación y Resolución de Problemas:** Los estudiantes se enfrentarán a desafíos técnicos y deberán idear soluciones creativas para diseñar y construir el sistema de monitoreo. Esto fomentará su capacidad de resolución de problemas y su pensamiento innovador.
- **Colaboración en Equipo:** Durante el taller, los estudiantes trabajarán en equipos para ensamblar y programar el prototipo. Aprenderán a colaborar, comunicarse y aprovechar las fortalezas individuales para lograr objetivos comunes, reflejando situaciones del mundo laboral real.
- **Conciencia Ambiental:** Al crear un sistema que optimice el riego y la ventilación en los invernaderos, los estudiantes estarán contribuyendo a un uso más eficiente de los recursos naturales. Esto promoverá su conciencia sobre la importancia de la sostenibilidad en la producción agropecuaria.
- **Presentación de Resultados:** La oportunidad de presentar sus prototipos y resultados ante la comunidad y sus compañeros de clase fomentará la habilidad de comunicación de los estudiantes. Aprenderán a transmitir sus ideas de manera clara y persuasiva.

- **Empoderamiento y Logro Personal:** Al ver su prototipo funcionando y contribuyendo a una solución real, los estudiantes experimentarán un sentido de logro y empoderamiento que les dará confianza en su capacidad para enfrentar desafíos técnicos y tecnológicos.

- **Planteo de la Problemática de la cooperativa "VerdeSur"**



Contexto: La Cooperativa Agropecuaria "VerdeSur", ubicada en una ciudad del sur de la provincia de Buenos Aires, está en busca de soluciones innovadoras para mejorar sus prácticas agrícolas. Como estudiantes del primer ciclo una escuela técnica, tienen la oportunidad de contribuir al desarrollo de un invernadero que permita a "VerdeSur" optimizar sus cultivos y recursos. El sistema estará orientado a controlar la temperatura y la humedad del suelo en los cultivos, y a tomar decisiones en tiempo real para mejorar el riego y la ventilación de los invernaderos. Los estudiantes se sumergirán en la combinación de electrónica, programación y soluciones prácticas para la agricultura moderna.

- **Desafío:** Se les presenta el desafío de diseñar, crear y fabricar un prototipo de invernadero inteligente que controle de manera automatizada la temperatura y la humedad. El objetivo es maximizar la producción de cultivos y minimizar el impacto ambiental, mediante el uso de tecnologías que puedan ser implementadas por la cooperativa.
- **Objetivos:** Trabajando en equipos, deben lograr lo siguiente:
- **Planificación de la Fabricación:** Utiliza la metodología proyectual para planificar la fabricación del prototipo, definiendo los pasos y los recursos necesarios.
- **Diseño del Invernadero:** Planificar cómo será la estructura del invernadero, teniendo en cuenta su tamaño y disposición. Considerar elementos como la ubicación de los cultivos, la circulación del aire y la entrada de luz solar.
- **Selección de Sensores y Actuadores:** Investigar y seleccionar sensores de temperatura y humedad adecuados para medir las condiciones del invernadero. Elegir motores de bomba y ventilador como actuadores para ajustar las condiciones ambientales.
- **Fabricación del Prototipo:** Utilizar materiales y herramientas disponibles para construir un prototipo funcional del invernadero. Esto implica la creación de la estructura, la instalación de sensores, actuadores y otros componentes necesarios.
- **Programación y Control:** Utilizar plataformas como Arduino para programar el control de los actuadores en función de las mediciones de los sensores. Programar la visualización de datos en un display LCD u otro medio.

- **Documentación:** Mantener un registro detallado de las decisiones de diseño, el código de programación, los resultados de pruebas y los posibles desafíos enfrentados durante la fabricación del prototipo.

- **Búsqueda de alternativas de solución**

En esta fase es muy importante el acompañamiento del o la docente sin sofocar el desarrollo la creatividad por parte de los y las jóvenes con la finalidad de generar soluciones funcionales y realizables, entre otros rasgos. Esto significa que el o la docente acompañara el proceso para que los estudiantes puedan recopilar la información pertinente, que les permita contrastar las características de los modelos de invernadero investigados para poder seleccionar la solución más adecuada que dé respuestas a la cooperativa agropecuaria "VerdeSur".



- **Selección de la solución adecuada**

Los estudiantes se enfrentan al desafío de elegir la mejor solución que aborde la problemática planteada por los docentes. Esta etapa es crucial para definir la dirección del proyecto y asegurarse de que la solución sea factible y adecuada para el contexto en el que se va a implementar. Las y los jóvenes se comprometen con un proceso de toma de decisiones en el que deben considerar múltiples aspectos para seleccionar la solución más adecuada y factible. Los docentes juegan un papel clave en guiar las discusiones y cuestionar las opciones para garantizar que la solución elegida aborde eficazmente la problemática del invernadero.



- **La etapa de diseño**

La etapa de diseño es un proceso creativo donde los estudiantes toman los conceptos y requisitos establecidos y los transforman en un diseño concreto y realizable. Utilizan herramientas como bocetos, cálculos, planos y diagramas para visualizar y planificar su solución antes de pasar a la etapa de implementación.

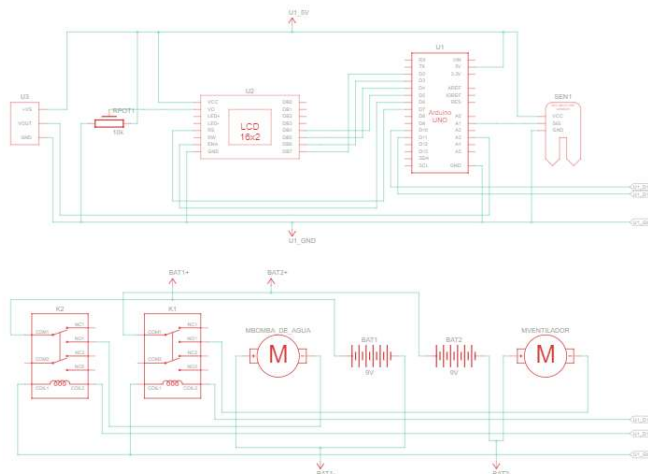
Se centrarán en crear una solución concreta y detallada para abordar la problemática planteada. Aquí están algunas acciones que los estudiantes deberán realizar durante esta etapa:

Análisis de Requisitos y Restricciones: Los estudiantes deben comprender completamente los requisitos del invernadero y las restricciones del proyecto. Esto incluye las necesidades de temperatura, humedad y riego, así como cualquier limitación de presupuesto, recursos y tiempo.

Generación de Ideas y Conceptos: Los estudiantes pueden comenzar a generar ideas y conceptos para su solución. Pueden crear bocetos, croquis y diagramas de cómo se verá su diseño final.

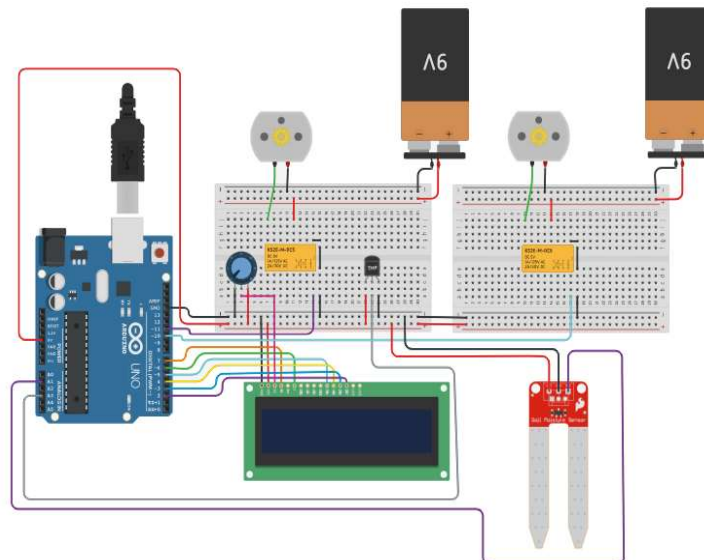
Selección de Componentes: Basándose en la etapa de selección de la solución, los estudiantes deben elegir los componentes específicos que utilizarán en su diseño. Esto incluye sensores, microcontroladores, relés, bombas, ventiladores, display LCD y otros.

Diseño del Circuito Electrónico: Los estudiantes deben crear un diseño detallado del circuito electrónico que conectará los componentes entre sí.



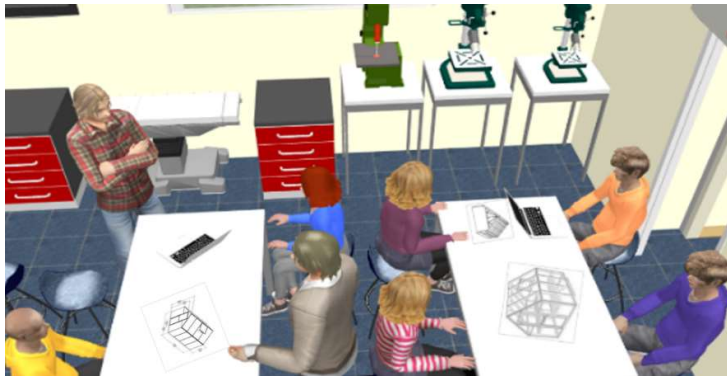
Diseño de la Interfaz de Usuario: Si su diseño incluye un display LCD, los estudiantes deben definir cómo se mostrará la información. Pueden diseñar la interfaz gráfica para mostrar la temperatura, humedad y otras indicaciones.

Planificación del Montaje Físico: Los estudiantes deben planificar cómo organizarán físicamente los componentes en el invernadero. Esto implica considerar la ubicación de sensores, relés y otros dispositivos, así como cómo se conectará todo.



Cálculos y Estimaciones: Si es necesario, los estudiantes pueden realizar cálculos y estimaciones para asegurarse de que su diseño funcione correctamente. Esto puede incluir cálculos de voltaje, corriente, resistencia y cualquier otro parámetro relevante.

Elaboración de Planos y Diagramas: Los estudiantes pueden crear planos y diagramas detallados de su diseño. Estos pueden incluir planos de circuitos, diagramas de conexión y diagramas de flujo de datos.



Plan de Acción: Los estudiantes deben elaborar un plan detallado de cómo llevarán a cabo la implementación de su diseño. Esto puede incluir pasos específicos, plazos y responsabilidades.

Revisión y Evaluación: Los estudiantes deben revisar y evaluar su diseño para asegurarse de que cumpla con los requisitos y objetivos. Pueden realizar simulaciones en línea o pruebas de concepto para validar su diseño antes de la implementación.

Iteración y Mejora: Si es necesario, los estudiantes pueden realizar iteraciones en su diseño para mejorar su funcionalidad, eficiencia o estética.

- **Organización del trabajo**

La Organización del Trabajo implica una planificación minuciosa para asegurar que el proceso de implementación se desarrolle de manera eficiente y efectiva. Los estudiantes deben considerar todos los aspectos, desde las tareas y los plazos hasta los recursos y la asignación de responsabilidades, para garantizar el éxito de su proyecto de invernadero.

Para nuestro modelo de invernadero las y los estudiantes sistematizarán las tareas, algunos realizarán técnicas operativas, otros se dedicarán al ensamblado, otros a la programación del sistema. También podrán realizar cada tarea en conjunto, todo dependerá de lo que consideren para agilizar la fabricación del invernadero de la forma más colectiva posible.

Los estudiantes deben desglosar el proceso en tareas más pequeñas y manejables. Esto podría incluir tareas como fabricar la estructura, montar el circuito, programar el microcontrolador, conectar los sensores y relés, crear la interfaz de usuario en el display, realizar pruebas, etc.

Las tareas deben organizarse en una secuencia lógica. Algunas tareas pueden depender de otras, por lo que es importante determinar el orden en que deben realizarse para evitar retrasos.

Al estar trabajando en equipo, los estudiantes pueden asignar responsabilidades a cada miembro del equipo para tareas específicas. Cada persona deberá saber qué tarea le corresponde.



- **Construcción de modelos**

Durante el desarrollo de nuestro prototipo de invernadero las y los jóvenes deberán realizar distintos procedimientos de fabricación junto con técnicas asociadas a estos. Si el prototipo requiere una estructura metálica deberán medir, trazar y cortar los perfiles metálicos de acuerdo con las características del diseño previamente realizado. Realizar las uniones que el prototipo requiera, aplicando correctamente las técnicas operativas vinculadas a estos procedimientos de fabricación, respetando las normas de seguridad e higiene del taller.

Integrar distintas tareas a la etapa de Construcción de Modelos garantizará un proceso fluido y seguro, así como un resultado final de alta calidad en el prototipo del invernadero automatizado:

Preparación de Materiales: Identificar y organizar los materiales necesarios para la construcción, como perfiles metálicos, plásticos, cables, sensores, actuadores, etc. Verificar la calidad y cantidad de los materiales antes de comenzar la construcción.

Mediciones y Trazado: Utilizar instrumentos de medición (cintas métricas, escuadras, niveles, calibres) para tomar medidas precisas de los materiales y componentes. Marcar los puntos de corte, perforación y ensamblaje según las especificaciones del diseño.

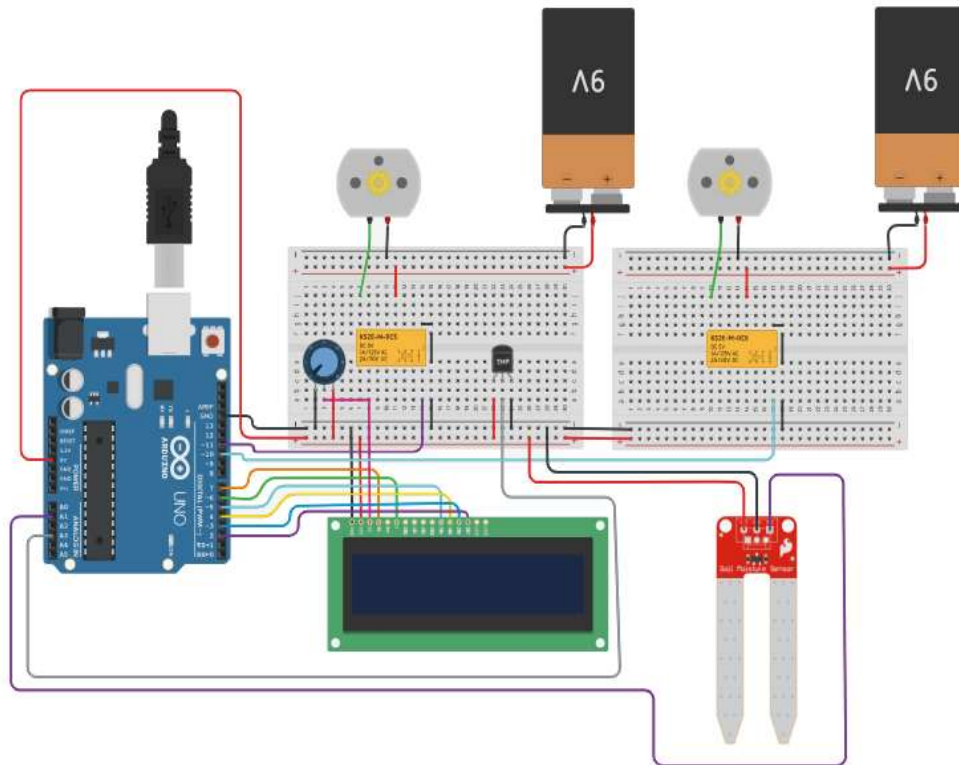


Técnicas de corte, unión y manipulación de materiales: Utilizar herramientas adecuadas para cortar los materiales, como sierras, cortadoras láser, impresoras 3D o herramientas de corte manual. Realizar operaciones de doblado, curvado o conformado en los materiales según sea necesario y aplicarán técnicas operativas adecuadas para unir y conectar los diferentes elementos. Esto puede incluir técnicas de soldadura, atornillado, pegado y ensamblaje mecánico, según las necesidades del diseño.



Ensamblaje y Montaje: Realizar el ensamblaje de las partes del prototipo siguiendo las indicaciones del diseño. Aplicar técnicas de montaje que aseguren la estabilidad y la resistencia del prototipo.





```

1 #include <LiquidCrystal.h>
2
3 LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // Configura tu LCD según las conexiones
4
5 const int tempSensorPin = A3; // Pin analógico del sensor de temperatura
6 const int humSensorPin = A1; // Pin analógico del sensor de humedad en tierra
7 const int relayVentiladorPin = 10; // Pin digital para el control del relé del ventilador
8 const int relayBombaPin = 11; // Pin digital para el control del relé de la bomba
9 const float umbralTemperatura = 25.0; // Umbral de temperatura en grados Celsius
10 const int umbralResistenciaSeco = 700; // Umbral de resistencia para suelo seco
11 const int umbralResistenciaHumedo = 300; // Umbral de resistencia para suelo húmedo
12
13 void setup() {
14     pinMode(relayVentiladorPin, OUTPUT); // Configura el pin del relé del ventilador como salida
15     pinMode(relayBombaPin, OUTPUT); // Configura el pin del relé de la bomba como salida
16
17     lcd.begin(16, 2); // Inicializa el LCD
18     lcd.setCursor(0, 0);
19     lcd.print("Temp: "); // Imprime el encabezado para la temperatura
20     lcd.setCursor(0, 1);
21     lcd.print("Humedad: "); // Imprime el encabezado para la humedad
22 }
23
24 void loop() {
25     int sensorValue = analogRead(tempSensorPin); // Lee el valor del sensor de temperatura
26     float temperature = (sensorValue * 5.0 / 1024.0 - 0.5) * 100; // Calcula la temperatura en grados Celsius
27
28     lcd.setCursor(6, 0); // Establece la posición del cursor para la temperatura
29     lcd.print(temperature, 1); // Imprime la temperatura en el LCD con un decimal
30
31     int resistenciaSuelo = analogRead(humSensorPin); // Lee el valor del sensor de resistencia de humedad en tierra
32
33     lcd.setCursor(10, 1); // Establece la posición del cursor para la humedad
34     if (resistenciaSuelo < umbralResistenciaSeco) {
35         lcd.print("Sec");
36         digitalWrite(relayBombaPin, HIGH); // Enciende el relé de la bomba para regar
37     } else if (resistenciaSuelo > umbralResistenciaHumedo) {
38         lcd.print("Hum");
39         digitalWrite(relayBombaPin, LOW); // Apaga el relé de la bomba
40     }
41
42     // Si la temperatura supera el umbral, activa el ventilador (enciende el relé del ventilador)
43     if (temperature > umbralTemperatura) {
44         digitalWrite(relayVentiladorPin, HIGH); // Enciende el relé del ventilador
45     } else {
46         digitalWrite(relayVentiladorPin, LOW); // Apaga el relé del ventilador
47     }
48
49     delay(1000); // Espera un segundo antes de realizar las siguientes lecturas
50 }
51

```

Normas de Seguridad e Higiene: Respetarán rigurosamente las normas de seguridad e higiene establecidas en el taller. Utilizarán equipo de protección personal adecuado, trabajarán en un entorno limpio y ordenado, y seguirán prácticas seguras en todas las etapas de la construcción.

- **La etapa de evaluación y perfeccionamiento**

En esta etapa, se revisará todo el proceso que llevó a la fabricación del prototipo, se comparará el resultado obtenido con los objetivos iniciales.

Probarán su funcionamiento e incluso podrán sugerir cambios y mejoras en cada una de las fases anteriores. Analizarán las consecuencias deseadas y las no deseadas. Entre el diseño que anticipa cómo será el producto y el producto terminado suele haber diferencias, lo que podrá deberse tanto a errores en la concreción del diseño como a modificaciones que se han detectado como necesarias durante el proceso de construcción. Esta etapa será útil, para sugerir cambios y mejoras en cada una de las fases anteriores y analizar las consecuencias deseadas y las no deseadas.

En el caso de nuestro ejemplo de invernadero las y los estudiantes evaluarán si el prototipo diseñado y fabricado cumple las expectativas generadas en base al problema o demanda planteada por los y las docentes. En esta instancia evaluarán también todo el proceso, sugiriendo cambios o mejoras en el prototipo. También podrán pensar que otras problemáticas se pueden resolver con prototipos similares al realizado.



Las etapas o fases descritas en esta metodología proyectual no suponen una prescripción rígida para abordar los problemas; constituyen una orientación para el trabajo. En ciertas ocasiones es necesario realizar el recorrido en distintos sentidos, reiteradamente, hasta lograr un ajuste de las diferentes etapas, ya que no se trabaja con un conocimiento acabado y definido, sino que se mejora, paulatinamente la apreciación sobre un problema, a partir de la adquisición de datos, conocimientos, ensayos, etc.

Si bien los proyectos de los y las estudiantes desarrollen deben tener un importante grado de autogestión, el o la docente debe establecer una disciplina de trabajo para evitar posibles dispersiones o incluso la decepción que llevaría al abandono. Debe enseñar formas de administrar y gestionar proyectos, como así también impulsar, desde el inicio, la documentación de las actividades en forma de bitácoras. Esto permitirá reconstruir lo trabajado en función de mejorar sus aprendizajes y compartir sus experiencias con otros grupos o personas.

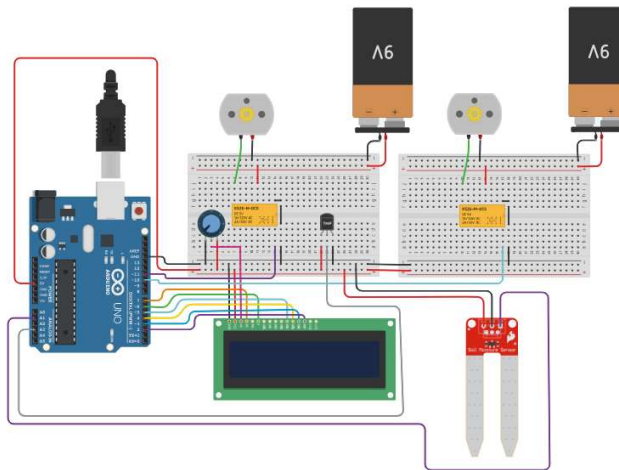
Explicación de la programación y el circuito asociado a esta problemática:

Este código y circuito permiten controlar y monitorear las condiciones ambientales en un invernadero. El sistema ajusta automáticamente el riego y la ventilación en función de la humedad en el suelo y la temperatura. La información relevante se muestra en el display LCD para que los usuarios puedan tener una visión en tiempo real de las condiciones dentro del invernadero.

Circuito Asociado: El circuito consta de un microcontrolador Arduino conectado a los sensores de temperatura y humedad en tierra (resistencia del suelo) en los pines A3 y A1, respectivamente.

Se utilizan dos relés conectados a los pines digitales 10 y 11 para controlar el ventilador y la bomba de agua cuyo circuito será alimentado por dos baterías de 9 volt respectivamente.

Se utiliza un display LCD para mostrar la temperatura en la parte superior y la condición de humedad en la parte inferior del display. Un potenciómetro para regular el contraste del display. Y se utilizaran dos protoboard y cables de distintos colores para realizar el conexionado.



Ver en Tinkercad:

https://www.tinkercad.com/things/8ABC2fhDdPF?sharecode=JVT99Y5Me982w8egKy8B2tQx5i_vhgaCp_R3YCYbR-c

Funcionamiento de la Programación:

Bloque de definición o bloque de declaración: En este bloque, se definen todas las variables globales y constantes que se utilizarán en el programa. En el caso de nuestro código, es donde se realizan las configuraciones y asignaciones iniciales para los pines y los valores de umbral.

En este bloque, se incluye la librería **LiquidCrystal.h** y se definen las constantes y pines que se utilizarán en el programa. Estas definiciones son globales y se pueden acceder desde cualquier parte del código, lo que facilita su uso y mantenimiento.

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // Configura tu LCD según las conexiones

const int tempSensorPin = A3; // Pin analógico del sensor de temperatura
const int humSensorPin = A1; // Pin analógico del sensor de humedad en tierra
const int relayVentiladorPin = 10; // Pin digital para el control del relé del ventilador
const int relayBombaPin = 11; // Pin digital para el control del relé de la bomba
const float umbralTemperatura = 25.0; // Umbral de temperatura en grados Celsius
const int umbralResistenciaSeco = 700; // Umbral de resistencia para suelo seco
const int umbralResistenciaHumedo = 300; // Umbral de resistencia para suelo húmedo
```

Configuración Inicial (setup): Se configuran los pines de los relés como salidas. Se inicializa el display LCD y se posiciona el cursor en las posiciones adecuadas para la temperatura y la humedad.

```
void setup() {
  pinMode(relayVentiladorPin, OUTPUT); // Configura el pin del relé del ventilador como salida
  pinMode(relayBombaPin, OUTPUT); // Configura el pin del relé de la bomba como salida

  lcd.begin(16, 2); // Inicializa el LCD
  lcd.setCursor(0, 0);
  lcd.print("Temp: "); // Imprime el encabezado para la temperatura
  lcd.setCursor(0, 1);
  lcd.print("Humedad: "); // Imprime el encabezado para la humedad
}
```

Bucle Principal (loop): Este código demuestra cómo se utilizan los valores de los sensores y las constantes definidas previamente para tomar decisiones y controlar actuadores en función de las condiciones del invernadero. Se ejecuta repetidamente en un ciclo infinito mientras el microcontrolador esté encendido.

```
void loop() {
  int sensorValue = analogRead(tempSensorPin); // Lee el valor del sensor de temperatura
  float temperature = (sensorValue * 5.0 / 1024.0 - 0.5) * 100; // Calcula la temperatura en grados Celsius

  lcd.setCursor(6, 0); // Establece la posición del cursor para la temperatura
  lcd.print(temperature, 1); // Imprime la temperatura en el LCD con un decimal

  int resistenciaSuelo = analogRead(humSensorPin); // Lee el valor del sensor de resistencia de humedad en tierra

  lcd.setCursor(10, 1); // Establece la posición del cursor para la humedad
  if (resistenciaSuelo < umbralResistenciaSeco) {
    lcd.print("Sec");
    digitalWrite(relayBombaPin, HIGH); // Enciende el relé de la bomba para regar
  } else if (resistenciaSuelo > umbralResistenciaHumedo) {
    lcd.print("Hum");
    digitalWrite(relayBombaPin, LOW); // Apaga el relé de la bomba
  }

  // Si la temperatura supera el umbral, activa el ventilador (enciende el relé del ventilador)
  if (temperature > umbralTemperatura) {
    digitalWrite(relayVentiladorPin, HIGH); // Enciende el relé del ventilador
  } else {
    digitalWrite(relayVentiladorPin, LOW); // Apaga el relé del ventilador
  }

  delay(1000); // Espera un segundo antes de realizar las siguientes lecturas
}
```

Cuestiones a tener en cuenta:

Esta línea de código toma el valor analógico del sensor de temperatura, realiza una serie de cálculos para convertirlo en una medida relativa de temperatura en grados Celsius y almacena ese valor en la variable "temperature". Sin embargo, este cálculo puede requerir calibración adicional según las especificaciones del sensor utilizado y las características del entorno.

```
float temperature = (sensorValue * 5.0 / 1024.0 - 0.5) * 100;
```

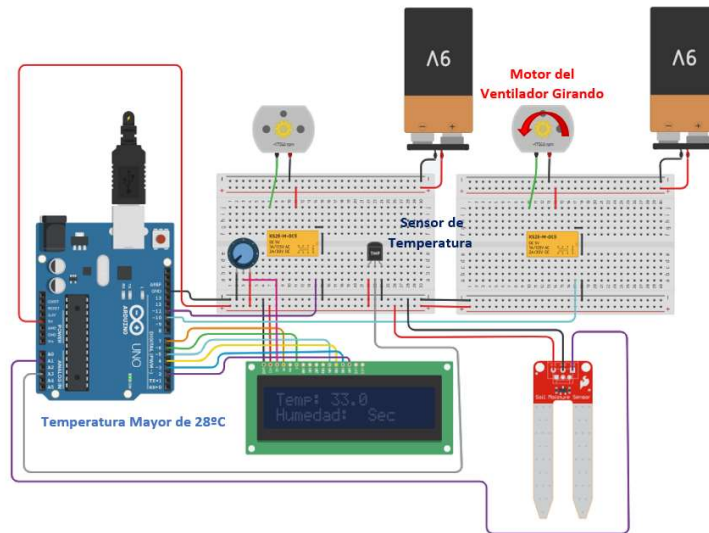
Esta línea de programación realiza un cálculo para convertir el valor analógico leído por un sensor de temperatura en una temperatura en grados Celsius. Vamos a desglosar cada parte de la expresión:

1. **sensorValue:** Es el valor analógico leído por el sensor de temperatura. Los valores analógicos son números que representan niveles de voltaje en una escala de 0 a 1023 en Arduino (10 bits de resolución).
2. **(sensorValue * 5.0 / 1024.0 - 0.5):** Esta parte de la expresión convierte el valor analógico en una medida de voltaje. El sensor lee voltajes proporcionales a la temperatura. Multiplicar por 5.0/1024.0 convierte el valor analógico en voltaje (5.0 es el voltaje de referencia de Arduino y 1024 es el rango de valores analógicos). Restar 0.5 es una compensación para ajustar la medición.
3. **(sensorValue * 5.0 / 1024.0 - 0.5) * 100:** Luego, esta parte multiplica el valor de voltaje obtenido por 100. El resultado de esta operación será una medida relativa de temperatura en Celsius, pero aún no estará en la escala correcta.
4. **float temperature:** Aquí se declara una variable llamada "temperature" de tipo flotante (número con decimales) para almacenar el resultado final de la conversión.

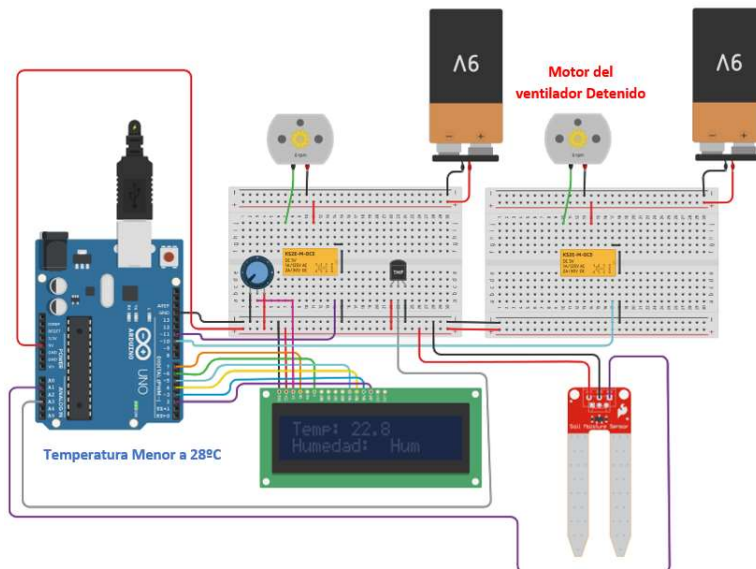
Lectura de Temperatura: Se lee la lectura analógica del sensor de temperatura en el pin A3 y se calcula la temperatura en grados Celsius. La temperatura se muestra en el display LCD en la parte superior.



Control del Ventilador: Se verifica si la temperatura supera el umbral predefinido (umbralTemperatura). Si la temperatura es mayor que el umbral, se acciona el relé encendiendo el motor del ventilador para enfriar el invernadero.



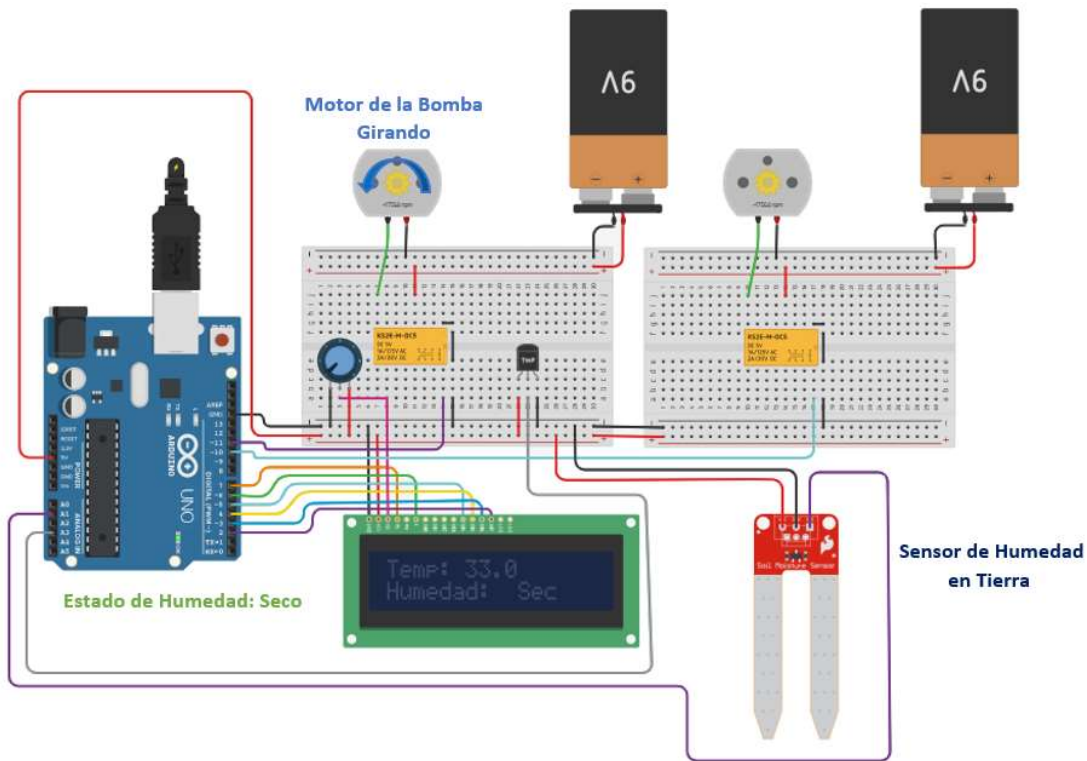
Si la temperatura es menor o igual al umbral, se desactiva el relé que comanda el motor y se apaga el ventilador.



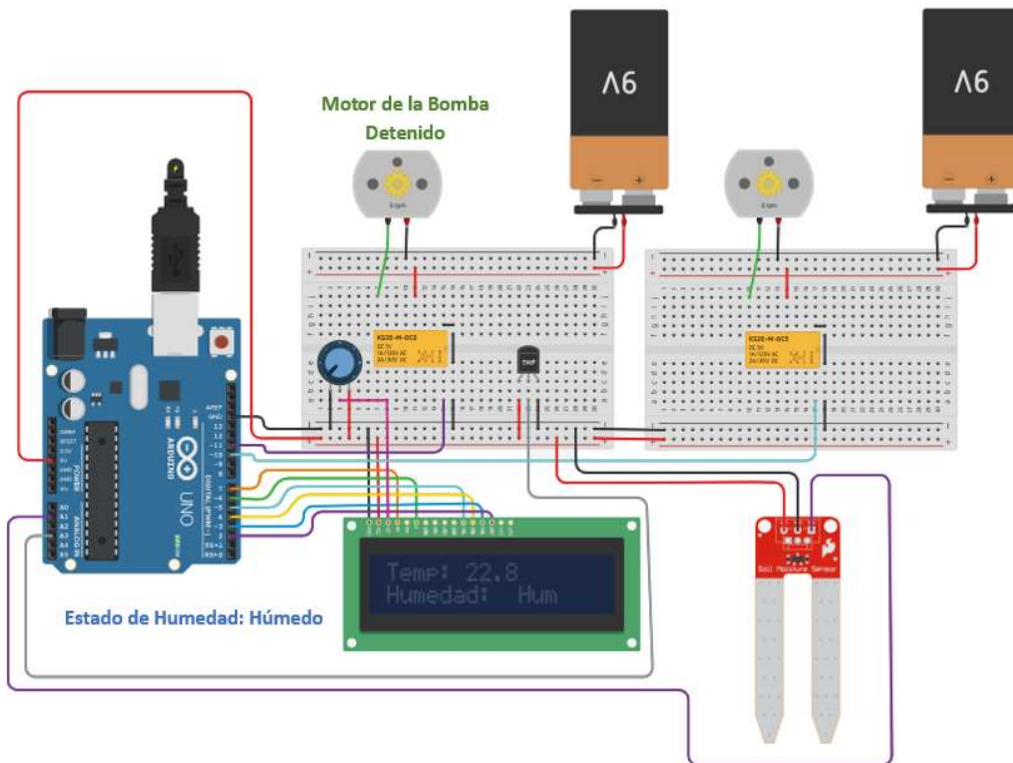
Sensor de Humedad en Tierra: Se lee la lectura analógica del sensor de resistencia de humedad en tierra en el pin A1. Se compara la lectura de resistencia con los umbrales definidos (umbralResistenciaSeco y umbralResistenciaHumedo) para determinar si el suelo está seco o húmedo.



Control e la Bomba de Agua: Si la resistencia es menor que el umbral de suelo seco, se muestra "Sec" en el display LCD y se enciende el relé que activa el motor de la bomba para iniciar el riego de las plantas.



Si la resistencia es mayor que el umbral de suelo húmedo, se muestra "Hum" en el display LCD y se desactiva el relé que comanda el motor de la bomba de riego.



Ver en Tinkercad:

https://www.tinkercad.com/things/8ABC2fhDdPF?sharecode=JVT99Y5Me982w8egKy8B2tQx5i_vhgaCp_R3YCYbR-c

Espera y Repetición: Se espera un segundo mediante delay(1000) antes de realizar las siguientes lecturas y tomar decisiones nuevamente.

8. Situación Problema Tecnológica. “Problemática Abierta” (Metodología Proyectual)

Título de la Problemática: Casa Inteligente con Automatización y Control

Descripción: En la actualidad, la tecnología se encuentra en constante evolución y se integra cada vez más en nuestras vidas diarias. Una de las áreas en las que la tecnología ha tenido un gran impacto es la automatización del hogar. Como alumnos y alumnas del primer ciclo del taller de la escuela técnica, se les ha asignado la tarea de diseñar y construir un prototipo maquetizado de una casa inteligente que incorpore diversas funcionalidades para mejorar la comodidad y eficiencia de sus habitantes. El prototipo debe ser controlado por un Arduino y debe contar con las siguientes características:

Medición y Visualización de Temperatura: El prototipo debe medir la temperatura ambiente y mostrarla en un display. El usuario debe ser capaz de conocer la temperatura actual de la habitación en todo momento.

Detección de Presencia y Control de Iluminación: Se debe integrar un sensor de presencia que, al detectar la presencia de una persona en la habitación, active un relé que encienda una lámpara de 9 voltios. Cuando no haya presencia, la lámpara debe apagarse automáticamente.

Detección de aproximación de la puerta de entrada: Utilizando un sensor ultrasónico, se debe diseñar un sistema en el cual cuando un objeto se encuentre a una distancia menor a 200 cm del sensor, el prototipo debe activar un relé que encienda una lámpara alimentada por una fuente de 9 voltios. Cuando el sensor ya no detecte objetos dentro de ese umbral, la lámpara deberá apagarse.

Pensamiento computacional aplicado a la programación del prototipo:



Podemos utilizar las herramientas del pensamiento computacional para abordar la resolución del problema de la Casa Inteligente con Automatización y Control de manera más estructurada y efectiva. A continuación, te muestro cómo aplicar cada una de estas herramientas:

Descomposición: Dividimos el problema general en problemas más pequeños y manejables. Por ejemplo, podemos dividir el proyecto en las funcionalidades individuales: medición de temperatura, detección de presencia, control de luces, etc. Nos enfocamos en resolver cada problema pequeño por separado, lo que facilitará el desarrollo y la depuración de cada parte del programa.

Abstracción: Identificamos los aspectos clave del problema y aislamos los detalles menos relevantes. Por ejemplo, en lugar de preocuparnos por los detalles internos de cómo se mide la temperatura, nos concentramos en el proceso general de medición y cómo se utiliza esa información.

Abstraemos componentes complejos en funciones o módulos reutilizables. Por ejemplo, podemos crear funciones para controlar la luz y reutilizarlas para diferentes luces en el programa.

Patrones: Identificamos patrones recurrentes en el problema. Por ejemplo, si debemos controlar varias luces de manera similar, buscamos patrones en el código y creamos una función genérica para manejar estas acciones.

Utilizamos estructuras de control y bucles para repetir acciones similares, como la lectura de sensores y el control de actuadores.

Algoritmo: Diseñamos algoritmos claros y secuenciales para cada funcionalidad. Por ejemplo, creamos un algoritmo que describa cómo se realizará la medición de temperatura: activar el sensor, leer la entrada analógica, convertir el valor, mostrar en el LCD, etc. Dividimos los algoritmos en pasos discretos y secuenciales para lograr los objetivos específicos de cada funcionalidad.

Metodología Proyectual recomendada

- **Diseño de la Problemática**

La etapa de diseño de la problemática es esencial para la formulación de una situación desafiante que inspire a los estudiantes a abordar con entusiasmo los conceptos clave y desarrollar habilidades prácticas en el contexto de la automatización de una casa inteligente.



Paso 1: Reunión Inicial. Los docentes del taller se reúnen para colaborar en la planificación y el diseño de una problemática significativa. Definen el objetivo principal: diseñar un prototipo de casa inteligente que incorpore tecnologías de automatización y control para mejorar la vida de sus habitantes.

Paso 2: Objetivos de Aprendizaje. En equipo, determinan los objetivos educativos que se desean lograr con esta problemática. Los objetivos deben reflejar la adquisición de habilidades técnicas, la aplicación práctica de conocimientos, el desarrollo de capacidades, la promoción de la colaboración y la innovación entre los estudiantes.

Paso 3: Selección del Contexto y Tema. Eligen el contexto de una "Casa Inteligente" y unifican los elementos clave: medición de temperatura, detección de presencia y control de iluminación, y detección de aproximación a la puerta de entrada. La problemática se basa en la necesidad de integrar tecnologías para mejorar la eficiencia y comodidad en el hogar.

Paso 4: Enunciado de la Problemática. "Como estudiantes del primer ciclo del taller de la escuela técnica, se les encomienda la emocionante tarea de diseñar y construir un prototipo maquetizado de una Casa Inteligente. Esta casa del futuro debe incorporar tecnologías de automatización y control basadas en Arduino para mejorar la calidad de vida de sus habitantes. El reto consiste en desarrollar un sistema que permita medir y visualizar la temperatura ambiente, detectar la presencia de personas para controlar la iluminación y utilizar un sensor ultrasónico para activar luces cuando alguien se acerque a la puerta de entrada. La Casa Inteligente debe ser un ejemplo tangible de cómo la tecnología puede aumentar la comodidad y la eficiencia en el entorno doméstico."

Paso 5: Beneficios Esperados

- **Adquisición de Habilidades Técnicas:** Los estudiantes desarrollarán competencias en electrónica, programación y diseño de sistemas automatizados.
- **Aplicación Práctica del Conocimiento:** Aplicarán conceptos teóricos en la creación de un prototipo funcional, viendo la relevancia de su educación en la vida real.
- **Innovación y Resolución de Problemas:** Afrontarán desafíos técnicos que requerirán soluciones creativas y pensamiento innovador.
- **Colaboración en Equipo:** Trabajarán en grupos para idear y crear la Casa Inteligente, mejorando sus habilidades de colaboración y comunicación.
- **Conciencia Ambiental:** Al desarrollar soluciones eficientes, considerarán la importancia de la sostenibilidad y el consumo responsable de recursos.
- **Presentación de Resultados:** Tendrán la oportunidad de presentar sus prototipos ante sus compañeros y docentes, desarrollando habilidades de comunicación efectiva.
- **Empoderamiento y Logro Personal:** Ver su prototipo funcionando generará un sentido de logro y confianza en sus habilidades técnicas y creativas.

- **Planteo de la problemática**

Una vez que la problemática ha sido diseñada, el paso siguiente es presentarla a los estudiantes de manera abierta y desafiante. La descripción del problema debe ser formulada de manera que permita a los estudiantes explorar diversas perspectivas y enfoques. Esto establece una visión general de la situación problemática y brinda una base para la búsqueda de soluciones. La fase de planteo de la problemática se enmarca en los siguientes parámetros:



Contexto: Los estudiantes serán sumergidos en el contexto de una sociedad en la que la automatización y la tecnología se están convirtiendo en parte integral de la vida cotidiana. La problemática se enmarca en la tarea de diseñar y construir un prototipo funcional de una “Casa Inteligente” que mejore la comodidad y la eficiencia de sus habitantes. Se enfocarán en aspectos como la medición de temperatura, la detección de presencia y el control de iluminación, entre otros.

Desafío: El desafío consiste en que los estudiantes apliquen sus habilidades técnicas y creativas para crear una solución innovadora y funcional para la automatización de una casa. Deben considerar los requisitos de medición, detección y control, y encontrar formas de integrar diferentes tecnologías para lograr una experiencia inteligente y eficiente para los habitantes.

Objetivos:

Trabajando en equipos, los estudiantes tienen los siguientes objetivos:

- **Planificación de la Fabricación:** Diseñar una hoja de ruta que incluya los pasos clave para la fabricación del prototipo de la Casa Inteligente. Esto implica definir los componentes necesarios y establecer un cronograma de trabajo.
- **Diseño de la casa:** Crear un diseño detallado de la Casa Inteligente que incluya la disposición de sensores, actuadores, pantallas y otros componentes. Deben considerar la usabilidad y la estética en el diseño.
- **Selección de Sensores y Actuadores:** Identificar los sensores y actuadores adecuados para las funciones requeridas, como medir la temperatura, detectar presencia y controlar la iluminación. Evaluarán las características técnicas y la compatibilidad con el sistema.
- **Fabricación del Prototipo:** Construir físicamente el prototipo de la Casa Inteligente siguiendo el diseño establecido. Esto puede involucrar el ensamblaje de componentes electrónicos, la creación de estructuras físicas y la integración de tecnologías.
- **Programación y Control:** Desarrollar el software necesario para que los componentes de la Casa Inteligente funcionen juntos de manera coherente. Programarán microcontroladores y otros dispositivos para lograr la automatización y el control requeridos.
- **Documentación:** Mantener un registro detallado de los pasos tomados, decisiones de diseño, códigos de programación y cualquier problema o solución encontrados durante el proceso. Esto permitirá una comprensión completa del trabajo realizado.

- **Búsqueda de alternativas de solución**

En esta fase es muy importante el acompañamiento del o la docente sin sofocar el desarrollo la creatividad por parte de los y las jóvenes con la finalidad de generar soluciones funcionales y realizables, entre otros rasgos. Esto significa que el o la docente acompañara el proceso para que los estudiantes puedan recopilar la información pertinente, que les permita contrastar las características de los modelos de casas inteligentes investigados para poder seleccionar la solución más adecuada que dé respuestas a la problemática planteada.



- **Selección de la solución adecuada**

Los estudiantes se enfrentan al desafío de elegir la mejor solución que aborde la problemática planteada por los docentes. Esta etapa es crucial para definir la dirección del proyecto y asegurarse de que la solución sea factible y adecuada para el contexto en el que se va a implementar. Las y los jóvenes se comprometen con un proceso de toma de decisiones en el que deben considerar múltiples aspectos para seleccionar la solución más adecuada y factible. Los docentes juegan un papel clave en guiar las discusiones y cuestionar las opciones para garantizar que la solución elegida aborde eficazmente la problemática de la casa inteligente.

- **Etapa de Diseño**

La etapa de diseño es esencial para transformar los requisitos y conceptos iniciales en un plan concreto y realizable para la Casa Inteligente. Durante esta fase, los estudiantes utilizarán diversas herramientas y métodos para visualizar y planificar su solución de manera detallada, allanando el camino para la implementación exitosa. Aquí se describen las acciones clave que los estudiantes llevarán a cabo en esta etapa:

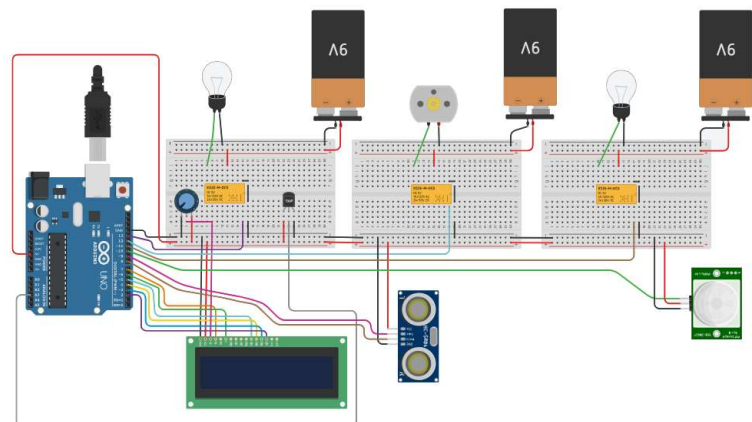
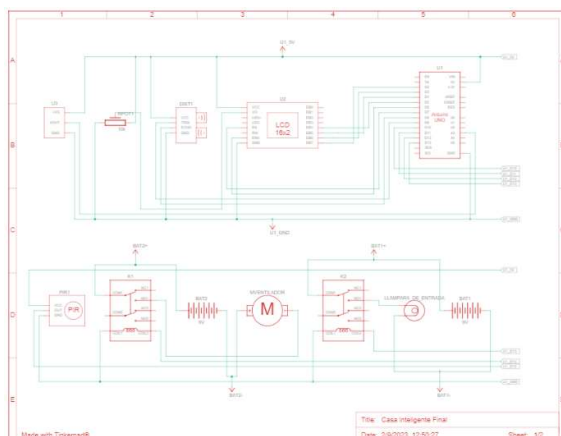


Análisis de Requisitos y Restricciones: Los estudiantes revisarán detenidamente los requisitos establecidos en la problemática y las restricciones técnicas y de recursos. Esto les ayudará a comprender claramente lo que se espera y las limitaciones en las que deben operar.

Generación de Ideas y Conceptos: Los estudiantes generarán ideas y conceptos creativos para abordar los desafíos planteados en la problemática. Explorarán diferentes enfoques y soluciones innovadoras que puedan integrar en el diseño final.

Selección de Componentes: Identificarán los sensores, actuadores, microcontroladores y otros componentes necesarios para implementar las funcionalidades de la Casa Inteligente. Evaluarán las especificaciones técnicas y la compatibilidad entre los componentes.

Diseño del Circuito Electrónico: Elaborarán un diseño detallado del circuito electrónico que conectará los diferentes componentes. Esto incluye la disposición de los componentes en la placa de circuito impreso y las conexiones eléctricas necesarias.



Diseño de la Interfaz de Usuario: Diseñarán la interfaz de usuario a través de la cual los habitantes interactuarán con la Casa Inteligente. Esto puede incluir pantallas, botones y otros elementos de control.

Planificación del Montaje Físico: Los estudiantes planificarán la disposición física de los componentes dentro de la maqueta de la casa. Definirán la ubicación de sensores, actuadores y dispositivos de control.

Cálculos y Estimaciones: Realizarán cálculos técnicos relevantes, como cálculos de consumo de energía, tiempos de respuesta de sensores y otros parámetros necesarios para garantizar un funcionamiento adecuado del sistema.

Elaboración de Planos y Diagramas: Crearán planos y diagramas detallados que muestren la disposición de los componentes, las conexiones eléctricas y otros aspectos importantes del diseño.

Plan de Acción: Elaborarán un plan paso a paso para la implementación del diseño. Esto incluirá la secuencia de montaje, la programación necesaria y las pruebas que se realizarán.

Revisión y Evaluación: Los estudiantes revisarán su diseño en busca de posibles problemas, conflictos o mejoras. Evaluarán si el diseño cumple con los requisitos y objetivos establecidos.

Iteración y Mejora: Si es necesario, realizarán iteraciones en el diseño para optimizarlo y abordar cualquier desafío que surja durante la revisión. Esto garantizará la calidad y eficacia del diseño final.

- **Organización del trabajo**

La Organización del Trabajo implica una planificación minuciosa para asegurar que el proceso de implementación se desarrolle de manera eficiente y efectiva. Los estudiantes deben considerar todos los aspectos, desde las tareas y los plazos hasta los recursos y la asignación de responsabilidades, para garantizar el éxito de su proyecto de casa inteligente. Para nuestro modelo de casa las y los estudiantes sistematizaran las tareas, algunos realizaran técnicas operativas, otros se dedicaran al ensamblado, otros a la programación del sistema. También podrán realizar cada tarea en conjunto, todo dependerá de lo que consideren para agilizar la fabricación del prototipo de la forma más colectiva posible.

Los estudiantes deben desglosar el proceso en tareas más pequeñas y manejables. Esto podría incluir tareas como fabricar la estructura, montar el circuito, programar el microcontrolador, conectar los sensores y relés, crear la interfaz de usuario en el display, realizar pruebas, etc.

Las tareas deben organizarse en una secuencia lógica. Algunas tareas pueden depender de otras, por lo que es importante determinar el orden en que deben realizarse para evitar retrasos.

Al estar trabajando en equipo, los estudiantes pueden asignar responsabilidades a cada miembro del equipo para tareas específicas. Cada persona deberá saber qué tarea le corresponde.

- **Construcción de Modelos**

La etapa de construcción de modelos es crucial para dar vida al diseño conceptual de la Casa Inteligente y transformarlo en un prototipo físico funcional. Durante esta fase, los estudiantes llevarán a cabo una serie de procedimientos de fabricación y técnicas operativas para construir la maqueta de la casa. La seguridad, la precisión y la atención al detalle serán fundamentales en este proceso. A continuación, se describen las acciones clave que los estudiantes realizarán en esta etapa:

Preparación de materiales y herramientas: Los estudiantes se asegurarán de tener todos los materiales, herramientas y componentes necesarios según el diseño previamente elaborado. Esto incluye los componentes electrónicos, materiales estructurales y cualquier otro elemento requerido.

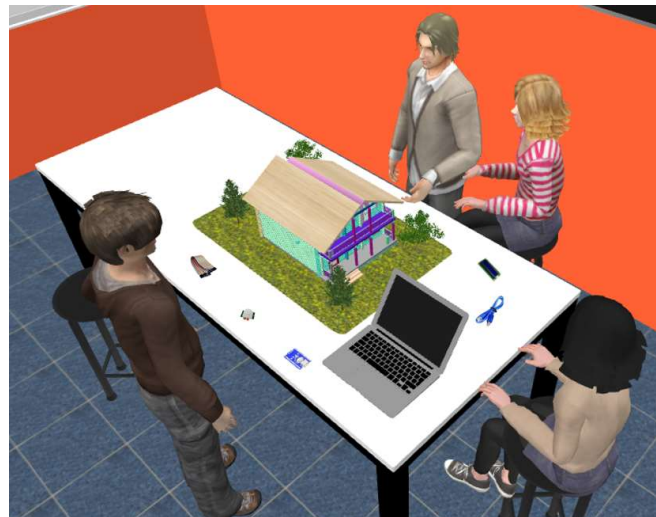


Mediciones y Trazado: Realizarán mediciones precisas de los materiales y componentes que serán utilizados en la construcción. Luego, trazarán marcas y líneas de corte en los materiales de acuerdo con las dimensiones y especificaciones del diseño.

Técnicas de corte, unión y manipulación de materiales: Utilizar herramientas adecuadas para cortar los materiales, como sierras, cortadoras láser, impresoras 3D o herramientas de corte manual. Realizar operaciones de doblado, curvado o conformado en los materiales según sea necesario y aplicarán técnicas operativas adecuadas para unir y conectar los diferentes elementos. Esto puede incluir técnicas de soldadura, atornillado, pegado y ensamblaje mecánico, según las necesidades del diseño.



Ensamblaje y Montaje: Realizarán el ensamblaje de los componentes y materiales de acuerdo con el diseño. Esto puede incluir la unión de piezas estructurales, la fijación de sensores y actuadores, y la creación de conexiones eléctricas.



Normas de Seguridad e Higiene: Respetarán rigurosamente las normas de seguridad e higiene establecidas en el taller. Utilizarán equipo de protección personal adecuado, trabajarán en un entorno limpio y ordenado, y seguirán prácticas seguras en todas las etapas de la construcción.

- **La etapa de evaluación y perfeccionamiento**

En esta etapa, se revisará todo el proceso que llevó a la fabricación del prototipo, se comparará el resultado obtenido con los objetivos iniciales.

Probarán su funcionamiento e incluso podrán sugerir cambios y mejoras en cada una de las fases anteriores. Analizarán las consecuencias deseadas y las no deseadas. Entre el diseño que anticipa cómo será el producto y el producto terminado suele haber diferencias, lo que podrá deberse tanto a errores en la concreción del diseño como a modificaciones que se han detectado como necesarias durante el proceso de construcción. Esta etapa será útil, para sugerir cambios y mejoras en cada una de las fases anteriores y analizar las consecuencias deseadas y las no deseadas.

En el caso de nuestro ejemplo de casa inteligente las y los estudiantes evaluarán si el prototipo diseñado y fabricado cumple las expectativas generadas en base al problema o demanda planteada por los y las docentes. En esta instancia evaluarán también todo el proceso, sugiriendo cambios o mejoras en el prototipo. También podrán pensar que otras problemáticas se pueden resolver con prototipos similares al realizado.



Las etapas o fases descritas en esta metodología proyectual no suponen una prescripción rígida para abordar los problemas; constituyen una orientación para el trabajo. En ciertas ocasiones es necesario realizar el recorrido en distintos sentidos, reiteradamente, hasta lograr un ajuste de las diferentes etapas, ya que no se trabaja con un conocimiento acabado y definido, sino que se mejora, paulatinamente la apreciación sobre un problema, a partir de la adquisición de datos, conocimientos, ensayos, etc.

Si bien los proyectos de los y las estudiantes desarrollen deben tener un importante grado de autogestión, el o la docente debe establecer una disciplina de trabajo para evitar posibles dispersiones o incluso la decepción que llevaría al abandono. Debe enseñar formas de administrar y gestionar proyectos, como así también impulsar, desde el inicio, la documentación de las actividades en forma de bitácoras. Esto permitirá reconstruir lo trabajado en función de mejorar sus aprendizajes y compartir sus experiencias con otros grupos o personas.

Explicación de la programación y el circuito asociado a esta problemática:

La Casa Inteligente diseñada y construida por los estudiantes del taller de la escuela técnica ofrece una serie de funcionalidades para mejorar la comodidad y eficiencia de sus habitantes. Utiliza una combinación de sensores y actuadores para mejorar la comodidad y eficiencia en el hogar. La programación adecuada permite que los dispositivos

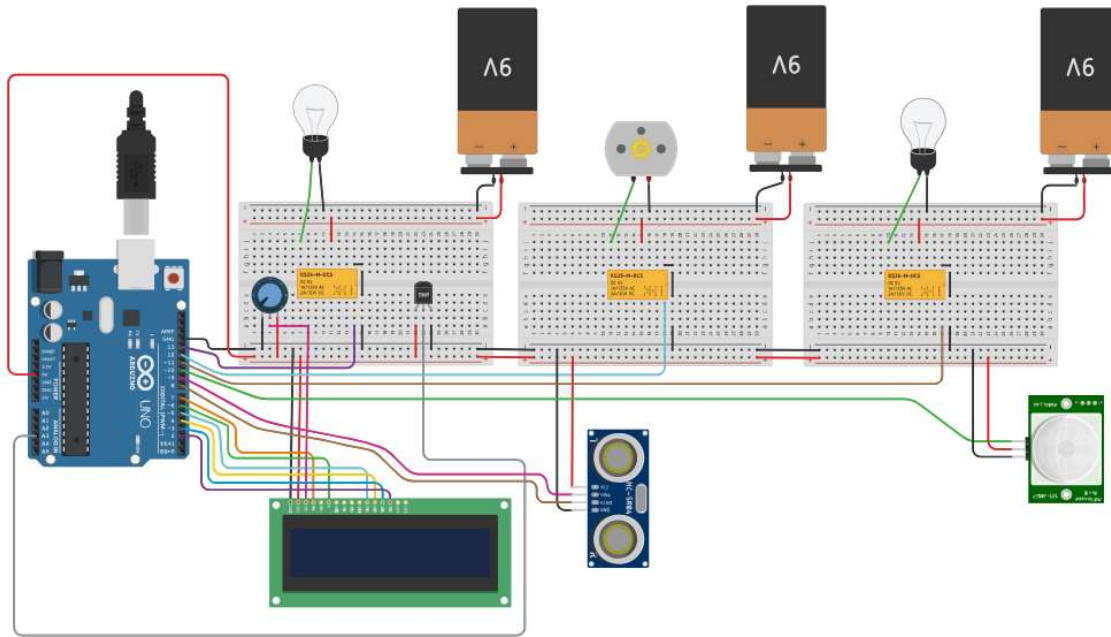
respondan de manera automática a los cambios en el entorno, brindando a los habitantes una experiencia más conveniente y sostenible. Aquí se presenta un resumen del funcionamiento de las diferentes características del prototipo:

- **Medición y Visualización de Temperatura:**
 - El sensor de temperatura instalado en la habitación mide constantemente la temperatura ambiente.
 - El Arduino convierte la lectura del sensor en una temperatura legible.
 - La temperatura se muestra en un display LCD colocado en un lugar visible de la habitación.
 - Los habitantes pueden conocer la temperatura actual y ajustar su entorno según sus preferencias.
- **Detección de Presencia y Control de Iluminación:**
 - El sensor de presencia detecta movimientos en la habitación.
 - Si alguien entra a la habitación, el sensor envía una señal al Arduino.
 - El Arduino activa un relé que enciende una lámpara de 9 voltios.
 - Cuando no hay movimiento detectado, el sensor envía una señal al Arduino para apagar automáticamente la lámpara.
- **Detección de Aproximación de la Puerta de Entrada:**
 - El sensor ultrasónico emite una señal ultrasónica y mide el tiempo que tarda en rebotar en un objeto cercano.
 - El Arduino calcula la distancia al objeto utilizando el tiempo medido.
 - Si un objeto se encuentra a una distancia menor a 200 cm del sensor (como alguien acercándose a la puerta), el Arduino activa un relé.
 - El relé enciende una lámpara de la puerta durante un período corto, iluminando el área alrededor de la entrada.
- **Interacción y Beneficios:**
 - Los habitantes pueden monitorear la temperatura en tiempo real y ajustar la calefacción o refrigeración según sea necesario.
 - La iluminación se enciende automáticamente cuando alguien entra a una habitación y se apaga cuando no hay presencia, ahorrando energía.
 - La lámpara de la puerta se enciende cuando alguien se acerca, proporcionando iluminación adicional y seguridad en la entrada.

Circuito Asociado:

El circuito asociado a la Casa Inteligente con automatización y control involucra la interconexión de varios componentes electrónicos, sensores y actuadores con una placa Arduino. Aquí se proporciona una descripción general del circuito y cómo se conectan los diferentes elementos:

- **Medición y Visualización de Temperatura:** Se conecta el pin de salida del sensor de temperatura (por ejemplo, LM35) al pin analógico A3 del Arduino. Se conecta el pin de alimentación y el pin de tierra del sensor de temperatura a los correspondientes pines del Arduino y a una fuente de alimentación. Se conecta el display LCD a los pines 7, 6, 5, 4, 3 y 2 del Arduino. Se conectan los pines de alimentación y tierra del display LCD a la fuente de alimentación.
- **Detección de Presencia y Control de Iluminación:** Se conecta el pin de salida del sensor de presencia (PIR) al pin digital 10 del Arduino. Se conecta un relé para la lámpara de 9 voltios al pin digital 11 del Arduino. Se conecta la lámpara de 9 voltios al relé y a una fuente de alimentación.
- **Detección de Aproximación de la Puerta de Entrada:** Se conecta el pin de salida del sensor ultrasónico (por ejemplo, HC-SR04) al pin digital 9 (TRIG) del Arduino para emitir el pulso ultrasónico. Se conecta el pin de entrada del sensor ultrasónico al pin digital 8 (ECHO) del Arduino para recibir el eco. Se conecta un relé para la lámpara de la puerta al pin digital 13 del Arduino. Se conecta la lámpara de la puerta al relé y a una fuente de alimentación.
- **General:** Se conectan los pines de alimentación y tierra de todos los componentes a una fuente de alimentación común, asegurando la correcta polaridad. Se conecta el Arduino a una fuente de alimentación adecuada (USB o batería) y se asegura de que los pines de alimentación y tierra estén conectados correctamente. Los pines de los componentes y del Arduino se conectan utilizando cables jumper.



Bloque de definición o bloque de declaración: Es una parte fundamental de la programación, ya que en él se establecen las variables, constantes, funciones y configuraciones iniciales que serán utilizadas en el programa. En el contexto de la Casa Inteligente con Automatización y Control, aquí te presento un ejemplo de cómo podría ser el bloque de definición:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

const int tempSensorPin = A3; // Pin analógico para el sensor de temperatura
const int relayVentiladorPin = 12; // Pin para controlar el relé del ventilador
const int sensorPir = 10; // Pin para el sensor PIR de movimiento
const int relayLuzHabitacion = 11; // Pin para controlar el relé de la luz de la habitación
const int relayLuzPuerta = 13; // Pin para controlar el relé de la luz de la puerta
const int TRIG = 9; // Pin para el trigger del sensor ultrasónico
const int ECHO = 8; // Pin para la recepción del eco del sensor ultrasónico
const float umbralTemperatura = 25.0; // Umbral de temperatura para encender el ventilador
```

Configuración Inicial (setup): La configuración inicial, es una parte clave del programa en Arduino. Aquí es donde se realizan todas las configuraciones iniciales, como la inicialización de pines, la configuración de los modos de funcionamiento y cualquier otra preparación necesaria antes de que comience el bucle principal del programa. En el contexto de la Casa Inteligente con Automatización y Control, aquí tienes un ejemplo de cómo podría ser la configuración inicial:

```
void setup() {
  lcd.begin(16, 2); // Inicializar la pantalla LCD de 16x2
  lcd.print("Casa Inteligente"); // Mostrar mensaje en la pantalla
  delay(2000); // Esperar 2 segundos
  lcd.clear(); // Limpiar la pantalla LCD

  pinMode(relayVentiladorPin, OUTPUT); // Configurar el pin del relé del ventilador como salida
  pinMode(sensorPir, INPUT); // Configurar el pin del sensor PIR como entrada
  pinMode(relayLuzHabitacion, OUTPUT); // Configurar el pin del relé de la luz de la habitación como salida
  pinMode(relayLuzPuerta, OUTPUT); // Configurar el pin del relé de la luz de la puerta como salida
  pinMode(TRIG, OUTPUT); // Configurar el pin del trigger del sensor ultrasónico como salida
  pinMode(ECHO, INPUT); // Configurar el pin de recepción del eco del sensor ultrasónico como entrada

  digitalWrite(relayVentiladorPin, LOW); // Apagar el ventilador al inicio
  digitalWrite(relayLuzHabitacion, LOW); // Apagar la luz de la habitación al inicio
  digitalWrite(relayLuzPuerta, LOW); // Apagar la luz de la puerta al inicio
}
```

Bucle Principal (loop): El bucle principal, es donde ocurre la ejecución continua del programa en Arduino. Aquí es donde se monitorean los sensores, se toman decisiones basadas en las condiciones y se controlan los actuadores. En el contexto de la Casa Inteligente con Automatización y Control, aquí tienes un ejemplo de cómo podría ser el bucle principal:

```
void loop() {
    float temperatura = analogRead(tempSensorPin) * 0.48875; // Leer el valor analógico del sensor de temperatura y convertirlo

    lcd.setCursor(0, 0); // Establecer la posición en la pantalla LCD
    lcd.print("Temp: "); // Mostrar etiqueta en la pantalla
    lcd.print(temperatura); // Mostrar el valor de temperatura en la pantalla
    lcd.print(" C"); // Mostrar la unidad Celsius en la pantalla

    controlarVentilador(temperatura); // Llamar a la función para controlar el ventilador
    controlarLuzHabitacion(); // Llamar a la función para controlar la luz de la habitación
    controlarLuzPuerta(); // Llamar a la función para controlar la luz de la puerta
}

void controlarVentilador(float temp) {
    if (temp > umbralTemperatura) {
        digitalWrite(relayVentiladorPin, HIGH); // Encender el ventilador si la temperatura es mayor al umbral
    } else {
        digitalWrite(relayVentiladorPin, LOW); // Apagar el ventilador si la temperatura es menor o igual al umbral
    }
}

void controlarLuzHabitacion() {
    int pirState = digitalRead(sensorPir); // Leer el estado del sensor PIR

    if (pirState == HIGH) {
        digitalWrite(relayLuzHabitacion, HIGH); // Encender la luz de la habitación si se detecta movimiento
    } else {
        digitalWrite(relayLuzHabitacion, LOW); // Apagar la luz de la habitación si no hay movimiento
    }
}

void controlarLuzPuerta() {
    digitalWrite(TRIG, LOW); // Apagar el trigger del sensor ultrasónico
    delayMicroseconds(2);
    digitalWrite(TRIG, HIGH); // Enviar un pulso de trigger
    delayMicroseconds(10);
    digitalWrite(TRIG, LOW);

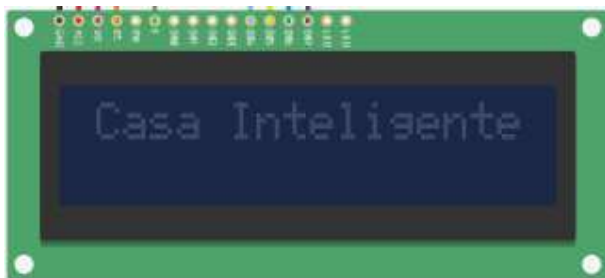
    long duration = pulseIn(ECHO, HIGH); // Medir la duración del eco
    float distance = duration * 0.034 / 2.0; // Calcular la distancia basada en el tiempo del eco

    if (distance < 200) {
        digitalWrite(relayLuzPuerta, HIGH); // Encender la luz de la puerta si la distancia es menor a 200 cm
        delay(1000); // Mantener la luz encendida por un segundo
        digitalWrite(relayLuzPuerta, LOW); // Apagar la luz de la puerta
    }
}
```

En este ejemplo, el bucle principal realiza las siguientes acciones:

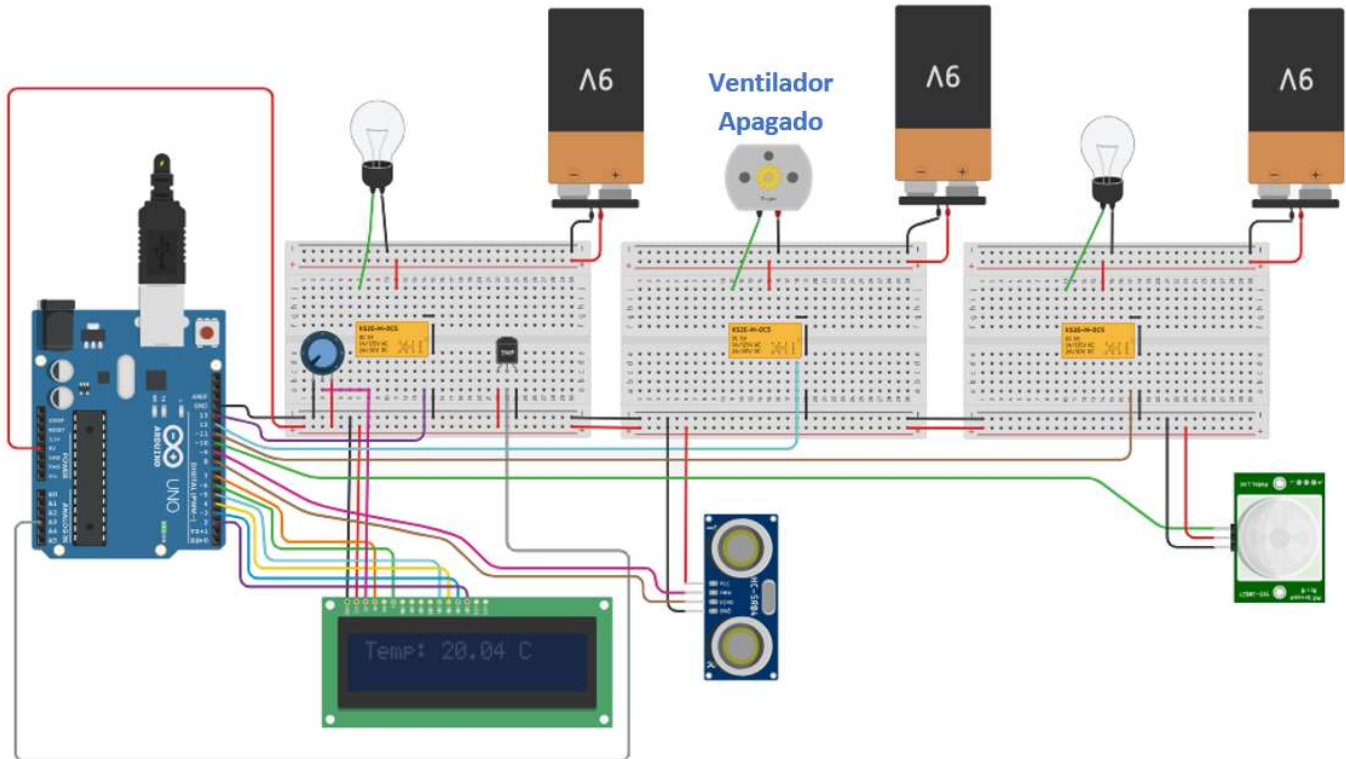
Medición de Temperatura: Lee el valor analógico del sensor de temperatura conectado al pin A3 y lo convierte en una temperatura en grados Celsius utilizando una fórmula de conversión.

Utiliza el display LCD para mostrar la temperatura en la primera línea del display.

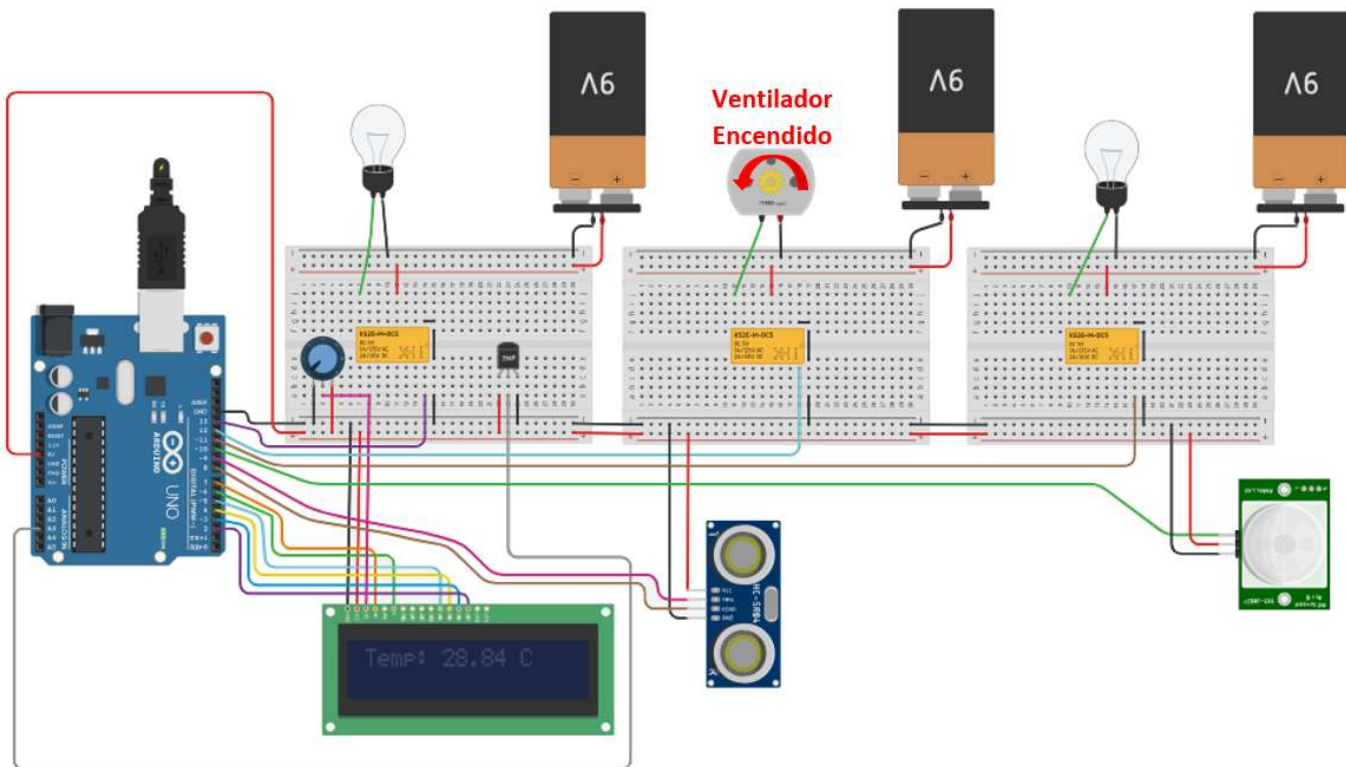


Control del Ventilador: Llama a la función controlarVentilador() pasándole como argumento la temperatura medida. Esta función toma la decisión de encender o apagar el ventilador en base a la temperatura y el umbral predefinido.

Cuando la temperatura ambiente no supera el umbral definido en el “Bloque de definición” 25°C. El ventilador permanecerá apagado.

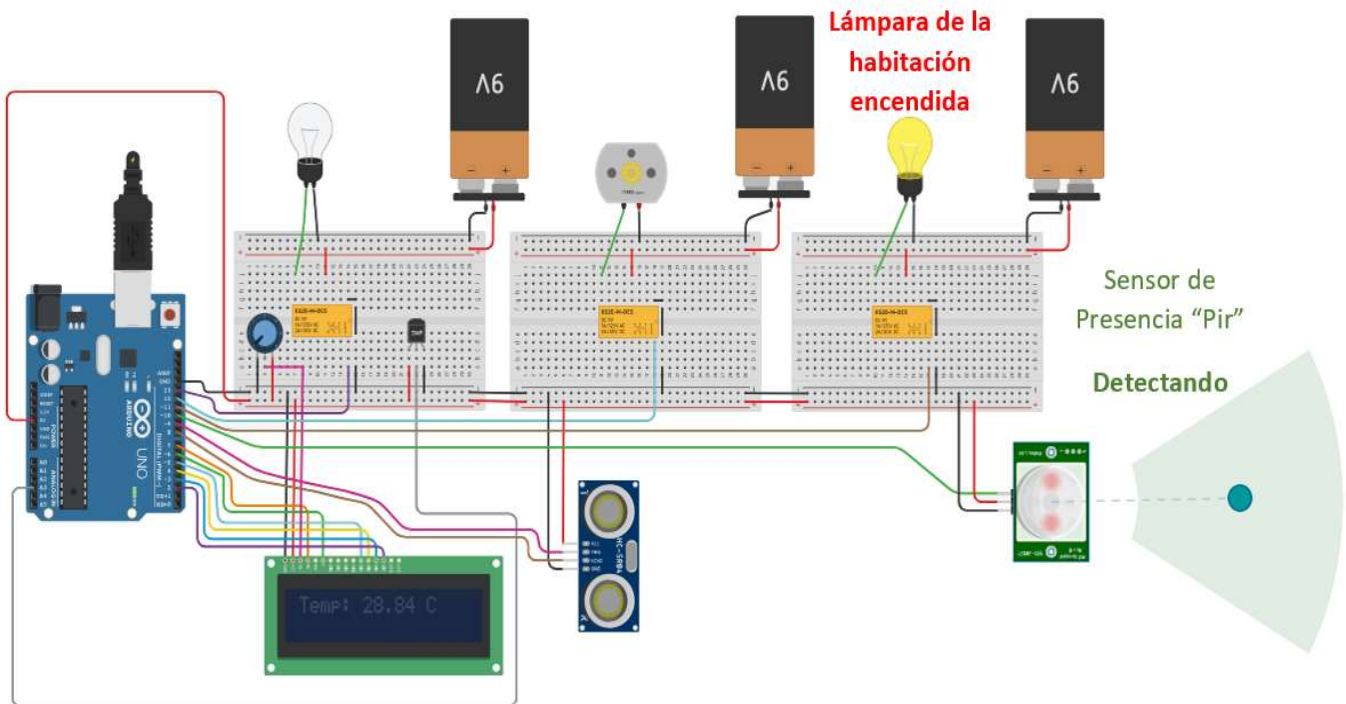


Cuando la temperatura ambiente supera el umbral definido en el “Bloque de definición” 25°C. El ventilador se encenderá.

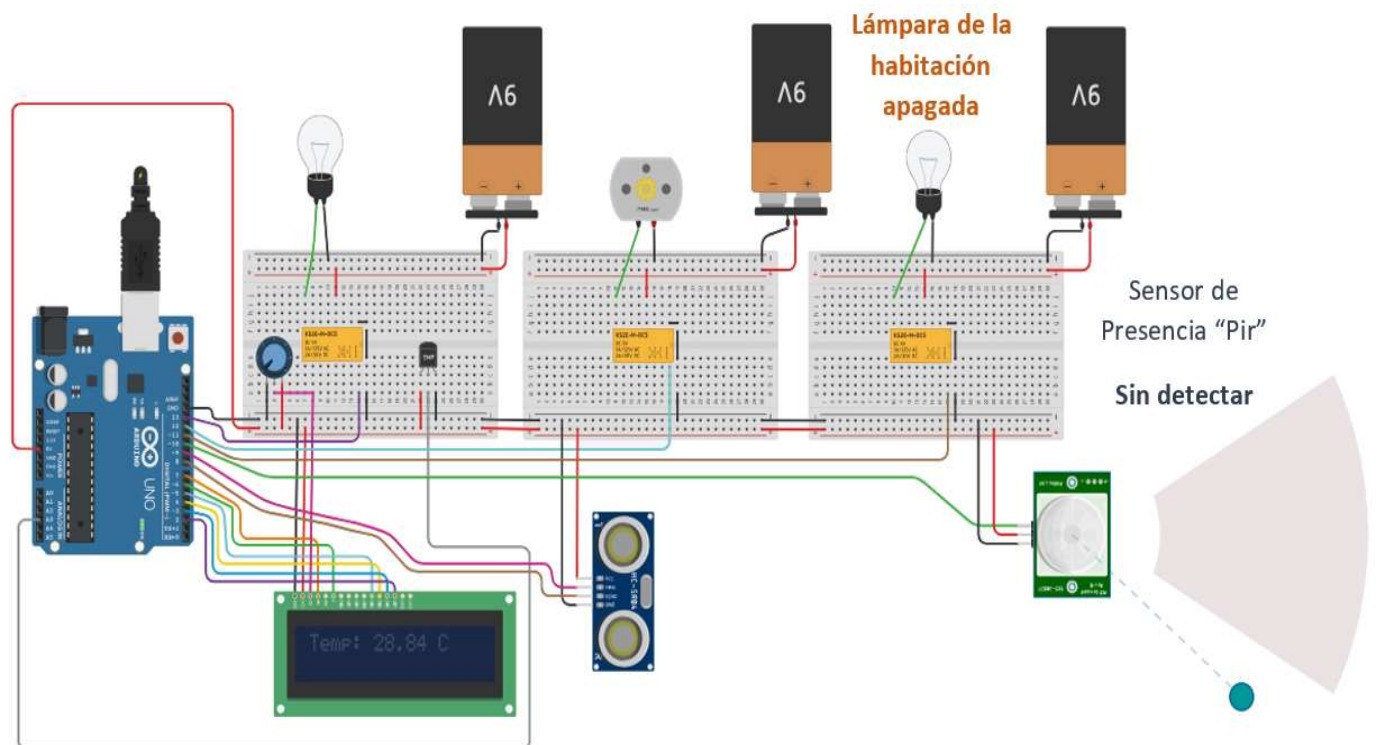


Control de la Luz de Habitación: Llama a la función `controlarLuzHabitacion()` para verificar si hay presencia y controlar la luz de habitación en consecuencia.

Cuando el “sensorPir” detecte presencia en la habitación la lámpara se encenderá.

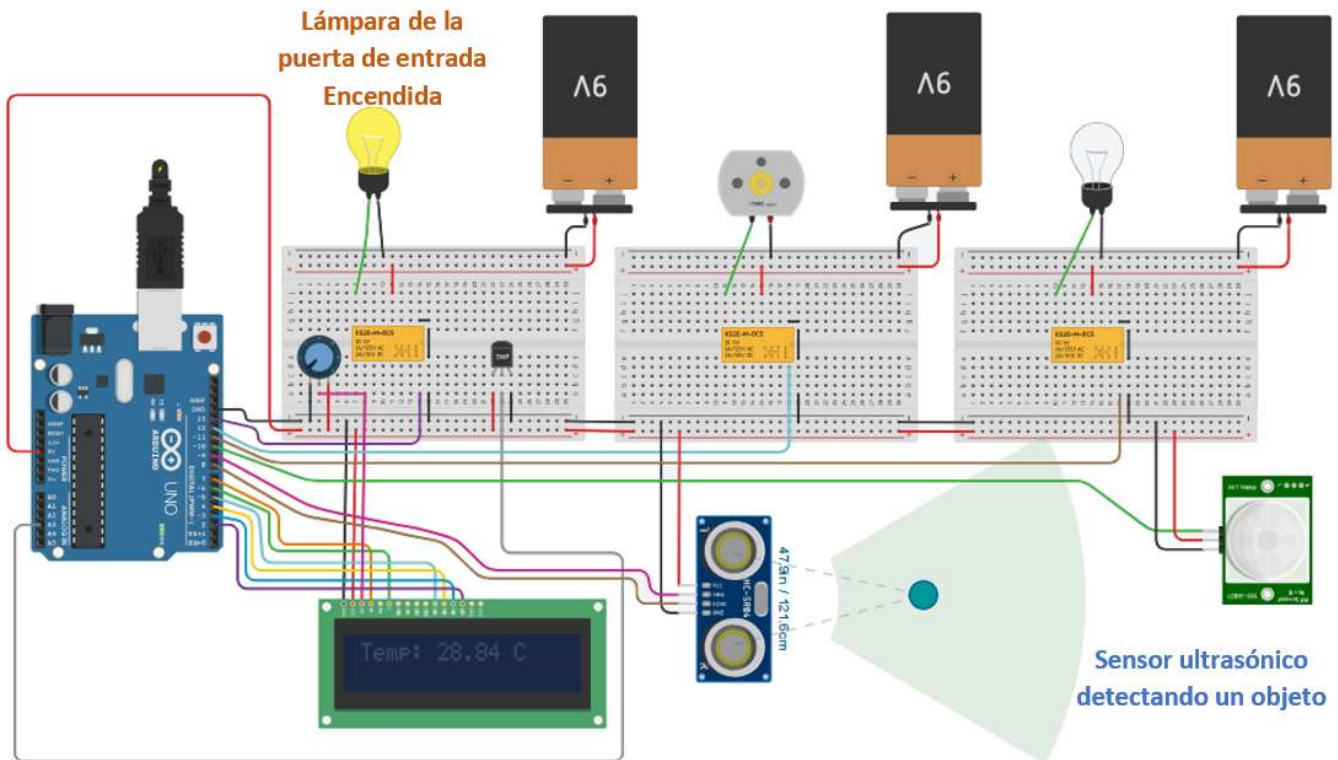


Cuando el “sensorPir” no detecte presencia en la habitación la lámpara permanecerá apagada.

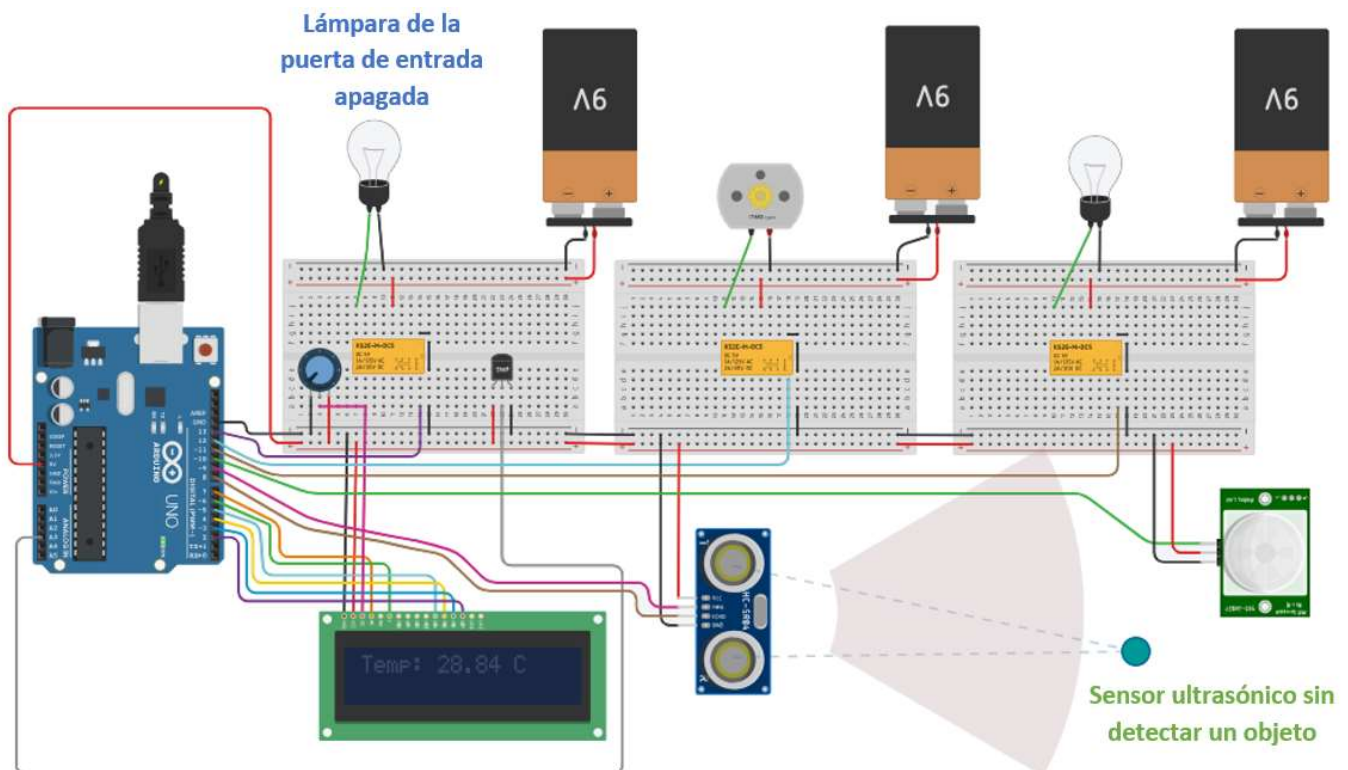


Control de la Luz de Puerta: Llama a la función `controlarLuzPuerta()` para detectar la distancia con el sensor ultrasónico y controlar la luz de puerta en consecuencia.

Cuando el sensor ultrasónico detecta un objeto a una distancia menor que el umbral definido la lámpara de la puerta de entrada se encenderá.



Cuando el sensor ultrasónico no detecta ningún objeto a una distancia menor que el umbral definido la lámpara de la puerta de entrada permanecerá apagada.



<https://www.tinkercad.com/things/2wEt9BH38VN?sharecode=JkaciaAmfAFcmhIJRaB6gliVYCrZtqbD3U4A3XNc80Q>

Estas acciones se repiten continuamente en el bucle principal, asegurando que las funcionalidades de la Casa Inteligente se monitoreen y controlen en tiempo real según las condiciones detectadas por los sensores.

El proyecto tecnológico y la evaluación de capacidades profesionales básicas

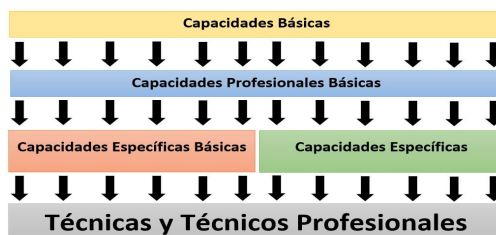
Uno de los rasgos, que hace a la especificidad de las ofertas formativas de ETP es que se estructura tomando como referencia los Perfiles Profesionales. El logro del Perfil Profesional propio de cada especialidad imprime rasgos particulares a las instituciones de ETP en lo que se refiere al diseño y organización curricular, a los manejos de tiempos, espacios y recursos relativos a la vida institucional en la ETP.

Reconociendo el carácter particular y polisémico de los conceptos de “capacidades” y “competencias”, así como su vinculación con enfoques y abordajes conceptuales muy diversos, resulta pertinente considerarlos tal como son referidos en la Ley 26.058. Allí, entre los propósitos de la ETP se plantea el de: “Formar técnicos medios y superiores en áreas ocupacionales específicas cuya complejidad requiera la disposición de competencias profesionales que se desarrollan a través de procesos sistemáticos y prolongados de formación para generar en las personas capacidades profesionales que son la base de esas competencias”

Entonces podemos decir que uno de los propósitos de la educación técnica profesional es el desarrollo de capacidades profesionales. Cuya construcción y desarrollo por parte de los y las estudiantes se realiza durante los seis o siete años de formación según la jurisdicción. Por tratarse de puntos de llegada complejos, es necesario considerar que la adquisición de capacidades se alcanza en distintas instancias y dimensiones del proceso de formación. Por ejemplo, es posible distinguir:

- Capacidades que se alcanzan en un espacio curricular determinado.
- Capacidades que se alcanzan en un conjunto de espacios curriculares afines
- Capacidades que se alcanzan en cada ciclo de la Educación Técnica.
- Capacidades que se alcanzan como resultado del proceso integral de la ETP.

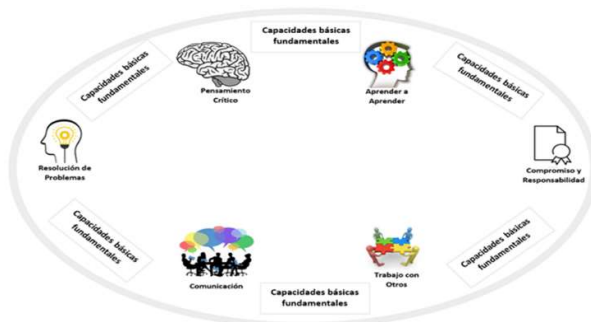
Así como es importante considerar que esas capacidades profesionales se desarrollan a partir de la construcción de capacidades básicas, capacidades profesionales básicas, y capacidades profesionales específicas, que hacen al perfil profesional.



Capacidades Básicas

Son aquellas capacidades que están en la base del desempeño de la ciudadanía responsable y crítica. Son el núcleo y soporte de las demás y se comparten con la educación secundaria orientada.

Las seis capacidades básicas fundamentales a desarrollar por los y las estudiantes durante su trayecto escolar obligatorio son: Resolución de problemas, pensamiento crítico, aprender a aprender, trabajo con otros, comunicación, y compromiso y responsabilidad. Y en nuestra modalidad, su desarrollo se asocia con la construcción y desarrollo de las capacidades profesionales ya sean básicas o específicas ya que están estrechamente relacionadas entre sí.

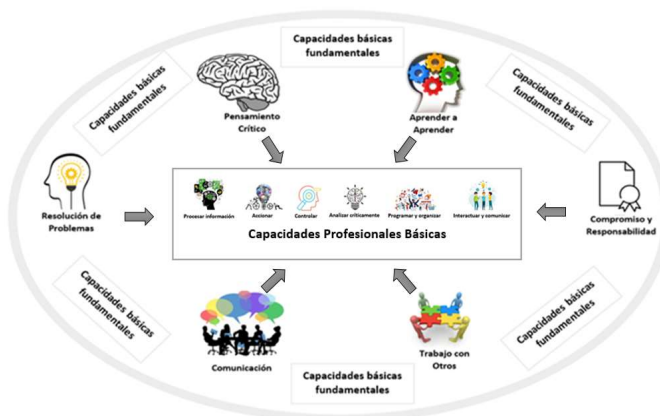


Las capacidades profesionales básicas

Las capacidades profesionales básicas son generales y comunes a todo técnico en tanto se articulan a todas las especialidades y resultan de un primer nivel de desagregación y especificación de las capacidades básicas. La construcción de estas capacidades está sustentada en los saberes y habilidades que son propias para cualquier técnica o técnico con independencia de la especialización y se desarrollan desde el inicio de la trayectoria formativa. Estas capacidades constituyen una suerte de bisagra entre la especificidad de la modalidad técnico-profesional y cierta desespecialización en relación al campo técnico específico. Las seis capacidades profesionales básicas a desarrollar por los estudiantes son: Interactuar y comunicar - Programar y organizar - Analizar críticamente - Procesar información - Controlar – Accionar. Para su construcción se deben considerar:

- **Interactuar y comunicar:** Se refiere a la capacidad de interacción y comunicación presente en toda relación humana y actividad social y la necesidad de establecerla considerando el respeto y rescate de la cultura y los saberes de las distintas personas y ámbitos donde se desarrolla su vinculación social y actividad profesional.
- **Programar y organizar:** Se refiere a la capacidad de formular y desarrollar proyectos significativos y viables en función de objetivos y de los recursos disponibles, analizando condiciones de rentabilidad y sustentabilidad.
- **Analizar críticamente:** Se refiere a la lectura de los contextos sociales en los que actúa más allá de lo observable, con capacidad para identificar causas y formular hipótesis consistentes con las situaciones dadas.
- **Procesar información:** Se refiere a la capacidad de generar información de distintas características a partir de distintas fuentes y a la obtención de datos necesarios para el relevamiento de situaciones para usos específicos.
- **Resolver problemas:** Se refiere a la capacidad de articular saberes de distinto tipo en situaciones concretas para enfrentar los problemas de manera realista y objetiva; planificar en forma sistemática métodos básicos para llegar a soluciones satisfactorias, con creatividad y originalidad en el uso de tecnologías estándares.
- **Controlar:** Se refiere a la capacidad de detectar en tiempo y forma errores, seleccionar los mecanismos de control entre los disponibles en su ámbito de desempeño, identificar las discrepancias respecto de lo esperado y anticipar y prevenir las consecuencias del error.
- **Accionar:** Se refiere al actuar, ejercer una acción, obrar, trabajar, ejecutar, producir un resultado, hacer funcionar, maniobrar, el hacer algo, el producir un efecto en situaciones de trabajo en donde la persona ejerce un control de lo que está haciendo a partir de un conocimiento previo, es decir, “sabe” cuáles serán los efectos de su “operar”.

En términos generales estas consideraciones coinciden con el campo científico tecnológico y dan cuenta de las reglas identitarias de la modalidad. Tomemos, por ejemplo, la capacidad para resolver problemas. Si la consideramos en clave de capacidad profesional básica, si bien habrá un cierto anclaje respecto de la especialidad, habrá también cierta transversalidad respecto a los modos de pensar y resolver los problemas en la perspectiva de la ETP. Al considerarla como parte del campo científico tecnológico esa resolución estará sostenida en un conjunto de saberes más o menos inespecíficos respecto de la modalidad. En tanto estas capacidades se articulan con las capacidades profesionales específicas se avanza y profundiza respecto de las particularidades que asumen en cada especialidad los posibles problemas y los modos y procedimientos considerados correctos para su resolución. La especificación de las capacidades profesionales básicas delimita un recorte de saberes y habilidades que conjugan conocimiento y prácticas de carácter tradicional en convergencia con saberes y prácticas de mayor grado de innovación como ser en la actualidad los saberes digitales y los saberes ligados a lo proyectual como modo de resolución de problemas en el campo tecnológico.



La evaluación de capacidades profesionales básicas en el proyecto tecnológico.

Dentro del contexto de un proyecto tecnológico, donde se busca desarrollar un prototipo de las características que hemos desarrollado, las capacidades profesionales básicas mencionadas anteriormente (Interactuar y comunicar, Programar y organizar, Analizar críticamente, Procesar información, Resolver problemas, Controlar, Accionar) son fundamentales para evaluar el desempeño de los estudiantes a lo largo de las diferentes etapas del proyecto. A continuación, se destacan algunas de las capacidades que se pueden evaluar en cada etapa:

- **Interactuar y comunicar:** La capacidad de "Interactuar y comunicar" es esencial en el contexto de un proyecto tecnológico, como los desarrollados. Esta capacidad implica la habilidad de los estudiantes para interactuar efectivamente con otras personas, ya sea en equipos de trabajo, con docentes o con posibles usuarios del proyecto. También abarca la capacidad de comunicar ideas, información y resultados de manera clara y efectiva. Aquí hay algunas consideraciones específicas relacionadas con esta capacidad en el contexto del proyecto:
 - **Colaboración en Equipos:** Evaluar cómo los estudiantes trabajan en equipo. ¿Pueden colaborar de manera efectiva con sus compañeros para planificar, diseñar, construir y programar el invernadero o la casa inteligente? Observar si pueden comunicar sus ideas y escuchar las de otros miembros del equipo. La comunicación efectiva es clave para la colaboración exitosa.
 - **Comunicación con los docentes:** Evaluar la capacidad de los estudiantes para comunicarse con sus docentes. ¿Pueden plantear preguntas, buscar orientación y expresar sus necesidades de manera clara? Analizar cómo responden a la retroalimentación y las sugerencias proporcionadas por los docentes.
 - **Presentación de Proyectos:** Evaluar la habilidad de los estudiantes para comunicar los resultados de su proyecto. ¿Pueden presentar de manera clara y persuasiva el propósito, diseño, funcionamiento y ventajas de su invernadero o su casa inteligente? Observar cómo manejan las preguntas y respuestas durante la presentación.
 - **Documentación del Proyecto:** Evaluar la capacidad de los estudiantes para documentar adecuadamente su trabajo. ¿Pueden redactar informes, manuales o documentación técnica que explique el proyecto de manera comprensible? Analizar la organización y claridad de la documentación que producen.
 - **Interacción con Usuarios:** Si es posible, evaluar cómo los estudiantes interactúan con posibles usuarios o clientes de su invernadero inteligente. ¿Pueden entender sus necesidades y explicar cómo su prototipo aborda esas necesidades? Observar si pueden responder preguntas y preocupaciones de los usuarios de manera efectiva.
- **Programar y organizar:** La capacidad de "Programar y organizar" es esencial en el contexto de un proyecto tecnológico como el desarrollo de un prototipo de invernadero inteligente. Esta capacidad implica la habilidad de los estudiantes para planificar y estructurar de manera efectiva el trabajo en el proyecto, desde la definición de los objetivos y recursos hasta la implementación de tareas y el seguimiento del progreso. Aquí hay algunas consideraciones específicas relacionadas con esta capacidad en el contexto del proyecto:
 - **Planificación del Proyecto:** Evaluar la capacidad de los estudiantes para diseñar una planificación sólida para el proyecto. ¿Pueden definir claramente los objetivos del proyecto, los entregables esperados y los plazos de tiempo? Analizar la estructura de su plan, incluyendo la secuencia de tareas y la asignación de responsabilidades.
 - **Organización de Recursos:** Evaluar cómo los estudiantes identifican y organizan los recursos necesarios para el proyecto. Esto puede incluir materiales, herramientas, componentes electrónicos y software. Observar si pueden estimar de manera realista los costos y tiempos asociados a la adquisición de recursos.
 - **Gestión del Tiempo:** Observar cómo los estudiantes manejan su tiempo durante el proyecto. ¿Pueden cumplir con los plazos establecidos en la planificación? Evaluar su capacidad para ajustar y reorganizar el plan si surgen desafíos o retrasos inesperados.
 - **Diseño de Proyectos Significativos:** Evaluar la calidad y relevancia de los proyectos que los estudiantes diseñan y proponen. ¿Pueden formular proyectos que aborden desafíos reales y que sean viables? Analizar cómo consideran factores como la rentabilidad y la sustentabilidad en la planificación de proyectos.

- **Evaluación de Riesgos y Contingencias:** Observar si los estudiantes son capaces de identificar posibles riesgos en el desarrollo del proyecto y si pueden diseñar estrategias de contingencia para mitigar esos riesgos. Evaluar cómo manejan situaciones inesperadas o problemas que puedan surgir durante la implementación del proyecto.
- **Documentación de la Planificación:** Analizar cómo los estudiantes documentan su planificación, ya sea en forma de diagramas de Gantt, listas de tareas, cronogramas u otros formatos. Evaluar si la documentación es clara y coherente, y si se actualiza a medida que avanza el proyecto.
- **Analizar críticamente:** La capacidad de "Analizar críticamente" desempeña un papel esencial en un proyecto tecnológico, ya sea el desarrollo de un prototipo de invernadero, una casa inteligente o cualquier otro prototipo. Esta capacidad implica la habilidad de los estudiantes para evaluar, cuestionar y comprender de manera profunda los aspectos técnicos, conceptuales y contextuales del proyecto. Aquí hay algunas consideraciones específicas relacionadas con esta capacidad en el contexto del proyecto:
 - **Análisis de Requerimientos:** Evaluar cómo los estudiantes analizan críticamente los requisitos del proyecto. ¿Pueden comprender las necesidades específicas de la cooperativa agrícola y traducirlas en especificaciones técnicas? Observar si pueden identificar posibles desafíos técnicos y limitaciones desde el principio.
 - **Evaluación de Soluciones Técnicas:** Analizar cómo los estudiantes evalúan diferentes enfoques técnicos para abordar los problemas del invernadero o la casa inteligente. ¿Pueden identificar y comparar soluciones alternativas? Observar si tienen la capacidad de seleccionar la solución más adecuada teniendo en cuenta factores como costo, eficiencia y viabilidad.
 - **Comprender el Contexto:** Evaluar la capacidad de los estudiantes para comprender críticamente el contexto en el que se implementará el prototipo del invernadero o casa inteligente. ¿Pueden considerar factores ambientales, económicos y sociales relevantes? Observar si pueden identificar posibles impactos y limitaciones relacionados con el contexto.
 - **Resolución de Problemas Complejos:** Analizar cómo los estudiantes abordan problemas técnicos complejos durante la construcción y programación del prototipo. ¿Pueden descomponer estos problemas en partes más manejables? Evaluar su capacidad para formular hipótesis y buscar soluciones efectivas.
 - **Adaptación a Cambios:** Observar si los estudiantes pueden analizar críticamente las situaciones inesperadas o los cambios en el proyecto y ajustar sus estrategias y enfoques en consecuencia. Evaluar su capacidad para tomar decisiones basadas en análisis críticos en tiempo real.
 - **Evaluación de Resultados:** Evaluar cómo los estudiantes analizan críticamente los resultados obtenidos a medida que avanzan en el proyecto. ¿Pueden identificar discrepancias entre los resultados esperados y los reales? Observar si pueden proponer mejoras o ajustes basados en su análisis crítico de los resultados.
- **Procesar información:** La capacidad de "Procesar información" es esencial para tomar decisiones informadas y respaldar el diseño, construcción y programación del invernadero o la casa inteligente. La evaluación de esta capacidad se centra en cómo los estudiantes gestionan y utilizan la información a lo largo del proyecto, desde la fase de investigación inicial hasta la implementación y seguimiento continuo. Implica la habilidad de los estudiantes para recopilar, organizar, interpretar y utilizar la información de manera efectiva a lo largo de todas las etapas del proyecto. Aquí hay algunas consideraciones específicas relacionadas con esta capacidad en este contexto:
 - **Recopilación de Datos y Fuentes de Información:** Evaluar la capacidad de los estudiantes para identificar fuentes de información relevantes para el proyecto, como manuales técnicos, investigaciones previas, especificaciones de componentes, etc. Observar cómo recopilan datos necesarios, como lecturas de sensores, mediciones de temperatura, humedad en el suelo, etc.
 - **Organización de Información:** Analizar cómo los estudiantes organizan la información recopilada de manera que sea fácilmente accesible y utilizable. Esto puede incluir la creación de bases de datos, hojas de cálculo u otros sistemas de organización. Evaluar la estructura y claridad de la información almacenada.
 - **Interpretación de Datos:** Evaluar la capacidad de los estudiantes para interpretar los datos recopilados y extraer conclusiones relevantes. ¿Pueden identificar patrones, tendencias o anomalías en los datos? Observar si pueden relacionar la información con los objetivos del proyecto y las necesidades.

- **Uso de Herramientas de Procesamiento de Datos:** Evaluar si los estudiantes utilizan herramientas informáticas o software específico para procesar datos, como Excel o software de análisis de datos. Observar su habilidad para realizar cálculos, gráficos y representaciones visuales de datos.
 - **Comunicación de Resultados:** Evaluar cómo los estudiantes utilizan la información procesada para respaldar sus decisiones y comunicar los resultados del proyecto. Observar si pueden explicar de manera efectiva cómo los datos influyen en el diseño y funcionamiento de los prototipos desarrollados.
 - **Adaptación a Nuevos Datos:** Observar si los estudiantes pueden adaptar su enfoque y estrategia en función de nueva información o datos que surgen durante el proyecto. Evaluar su capacidad para tomar decisiones basadas en datos actualizados.
 - **Gestión de Datos a Largo Plazo:** En proyectos que involucren monitoreo a largo plazo, evaluar cómo los estudiantes gestionan y procesan continuamente los datos recopilados durante un período prolongado. Observar si pueden mantener la calidad de los datos a lo largo del tiempo.
- **Resolver problemas:** La capacidad de "Resolver problemas" es esencial para superar los desafíos técnicos y conceptuales que surgen en el desarrollo de un proyecto tecnológico. La evaluación de esta capacidad se centra en la eficacia de los estudiantes para identificar, analizar y solucionar problemas en todas las etapas del proyecto, así como en su capacidad para aprender de estas experiencias y mejorar continuamente su habilidad para resolver problemas. Aquí hay algunas consideraciones específicas relacionadas con esta capacidad en este contexto:
 - **Identificación de Problemas:** Evaluar la capacidad de los estudiantes para identificar problemas técnicos y conceptuales que surgen durante el desarrollo del proyecto. Observar si pueden definir claramente los desafíos y comprender su impacto en el proyecto.
 - **Descomposición de Problemas:** Evaluar cómo los estudiantes descomponen problemas complejos en partes más manejables y comprensibles. Observar su habilidad para analizar la causa raíz de un problema y definir los pasos necesarios para abordarlo.
 - **Generación de Soluciones:** Analizar cómo los estudiantes generan múltiples soluciones posibles para un problema dado. Evaluar su creatividad y capacidad para considerar enfoques alternativos.
 - **Evaluación de Soluciones:** Evaluar cómo los estudiantes evalúan críticamente las soluciones propuestas en términos de viabilidad, eficacia y costo. Observar si pueden tomar decisiones informadas sobre cuál es la mejor solución.
 - **Implementación de Soluciones:** Evaluar la capacidad de los estudiantes para implementar las soluciones de manera efectiva en el proyecto, ya sea a nivel de hardware o software. Observar si pueden llevar a cabo los pasos necesarios para resolver el problema.
 - **Adaptación a Problemas Emergentes:** Observar cómo los estudiantes manejan problemas que pueden surgir durante la implementación del proyecto y cómo se adaptan para abordarlos. Evaluar su capacidad para tomar decisiones rápidas y efectivas en situaciones de cambio.
 - **Documentación de Problemas y Soluciones:** Evaluar cómo los estudiantes documentan los problemas encontrados y las soluciones implementadas. Observar si pueden comunicar claramente cómo se resolvieron los problemas y cómo esto afecta al proyecto en su conjunto.
 - **Aprendizaje de Problemas Anteriores:** Evaluar si los estudiantes pueden aplicar lecciones aprendidas de problemas anteriores a situaciones similares que puedan surgir en el proyecto. Observar si mejoran con el tiempo en la resolución de problemas.
 - **Controlar:** La capacidad de "Controlar" en un proyecto tecnológico abarca la detección de errores en tiempo y forma, la selección de mecanismos de control adecuados, la identificación de discrepancias respecto a lo esperado y la anticipación y prevención de las consecuencias del error. Esta capacidad es crítica para garantizar que el proyecto avance de manera efectiva y que se logren los objetivos previstos. La evaluación de esta capacidad se basa en la capacidad de los estudiantes para gestionar activamente el control en todas las etapas del proyecto. A continuación, podrás tener una descripción detallada de esta capacidad en el contexto de un proyecto tecnológico, manteniendo el enfoque en la detección de errores, selección de mecanismos de control, identificación de discrepancias y prevención de consecuencias del error:

- **Detectar Errores en Tiempo y Forma:** Los estudiantes deben ser ágiles en la detección de cualquier anomalía o error en el sistema del invernadero inteligente. Esto implica estar alerta a lecturas incorrectas de sensores, mal funcionamiento de actuadores o cualquier otro problema que pueda surgir. Evalúa si los estudiantes son capaces de monitorear activamente el funcionamiento del sistema o dispositivo en tiempo real. ¿Pueden identificar rápidamente cualquier anomalía o error en las lecturas de sensores o en el comportamiento de los actuadores?, ¿Son capaces de identificar desviaciones significativas en las mediciones o en el comportamiento del sistema?
 - **Selección de Mecanismos de Control Apropriados:** Cuando se detecta un error o una discrepancia, los estudiantes deben tener la capacidad de evaluar y seleccionar los mecanismos de control adecuados disponibles en su ámbito de desempeño. Esto podría implicar ajustar parámetros, cambiar la configuración de actuadores o tomar medidas correctivas. Evalúa si los estudiantes pueden seleccionar y aplicar los mecanismos de control adecuados para abordar problemas o desviaciones. ¿Eligen las estrategias de corrección más apropiadas según la naturaleza del error? ¿Pueden implementar cambios en la programación o ajustar los componentes del sistema para resolver problemas?
 - **Identificación de Discrepancias Respecto a lo Esperado:** Una parte fundamental de "Controlar" es la habilidad de comparar el rendimiento actual del invernadero inteligente con lo que se espera o desea. Los estudiantes deben ser capaces de identificar discrepancias y desviaciones en las lecturas de sensores, condiciones ambientales y resultados esperados.
 - **Anticipar y Prevenir Consecuencias del Error:** No basta con corregir errores; los estudiantes deben ser proactivos en la anticipación y prevención de las posibles consecuencias negativas de esos errores. Por ejemplo, si detectan un aumento de temperatura, deben tomar medidas para prevenir daños a los cultivos antes de que ocurran. Evalúa si pueden anticipar posibles problemas y tomar medidas preventivas para evitar errores futuros. ¿Implementan sistemas de alarma o medidas de seguridad proactivas?
 - **Documentación y Registro Preciso:** Es esencial mantener un registro detallado de todas las operaciones de control realizadas. Esto incluye ajustes, correcciones, acciones preventivas y respuestas a errores. La documentación proporciona una referencia invaluable para tomar decisiones informadas. Evalúa si mantienen registros detallados de las acciones de control tomadas, los errores identificados y las soluciones implementadas. ¿Documentan de manera adecuada el proceso de control?
 - **Adaptación Continua:** En un entorno de proyecto dinámico, los estudiantes deben estar preparados para adaptar el control a medida que surgen nuevos desafíos o situaciones cambiantes. La capacidad de tomar decisiones rápidas y efectivas es esencial. Evalúa su capacidad para adaptarse a situaciones cambiantes y ajustar los mecanismos de control según sea necesario.
- **Accionar:** La capacidad de "Accionar" en el contexto de un proyecto tecnológico se refiere a la habilidad de los estudiantes para llevar a cabo acciones con un propósito específico, ejercer control sobre los sistemas y procesos involucrados, y producir resultados con base en un conocimiento previo. Esta capacidad es fundamental en la educación técnica y profesional, ya que implica poner en práctica lo aprendido y aplicar el conocimiento de manera efectiva en situaciones de trabajo. La evaluación de esta capacidad se basa en la observación directa de las acciones y decisiones de los estudiantes en el contexto del proyecto. Se busca medir su habilidad para llevar a cabo tareas técnicas, aplicar conocimientos previos y producir resultados efectivos en situaciones de trabajo. La evaluación contribuye al desarrollo de las capacidades profesionales de los estudiantes, ya que les permite aplicar y fortalecer su conocimiento en un entorno práctico y tecnológico. A continuación, se explora cómo se puede evaluar la capacidad de "Accionar" en el contexto de un proyecto tecnológico y cómo esta evaluación contribuye al desarrollo de las capacidades profesionales de los estudiantes:
 - **Implementación Técnica:** Evaluar la habilidad de los estudiantes para implementar físicamente los componentes y sistemas necesarios para el proyecto. ¿Pueden montar y conectar sensores, actuadores y otros dispositivos de manera efectiva?
 - **Programación y Control:** Evaluar su capacidad para programar y controlar los dispositivos y sistemas utilizados en el proyecto. ¿Pueden escribir código efectivo para automatizar procesos y tomar decisiones en tiempo real?

- **Producción de Resultados:** Evaluar si los estudiantes pueden producir resultados medibles y cuantificables a partir de su trabajo en el proyecto. ¿Logran optimizar el rendimiento del invernadero inteligente y maximizar la producción de cultivos?
- **Ejercicio del Conocimiento Previo:** Evaluar cómo aplican el conocimiento previo adquirido en sus estudios técnicos y científicos para abordar situaciones específicas en el proyecto. ¿Utilizan principios de electrónica, programación y tecnología para resolver problemas?
- **Manejo de Herramientas y Equipos:** Evaluar su capacidad para manejar herramientas y equipos de manera segura y efectiva. ¿Operan las herramientas y equipos necesarios con destreza y precaución?
- **Resolución de Problemas en Tiempo Real:** Evaluar si pueden tomar decisiones y realizar acciones correctivas en tiempo real cuando surgen problemas o desafíos inesperados durante el proyecto.
- **Documentación y Registro:** Evaluar si mantienen registros precisos de las acciones tomadas, los cambios realizados y los resultados obtenidos. ¿Documentan de manera adecuada su proceso de trabajo?
- **Adaptación y Mejora Continua:** Evaluar si son capaces de adaptarse a situaciones cambiantes y buscar constantemente formas de mejorar el funcionamiento del proyecto.
- **Trabajo en Equipo:** Evaluar su capacidad para colaborar con otros miembros del equipo en la implementación y control de sistemas complejos.
- **Gestión de Recursos:** Evaluar si utilizan eficientemente los recursos disponibles, como materiales, tiempo y herramientas, para lograr los objetivos del proyecto.