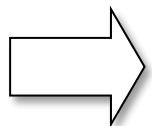
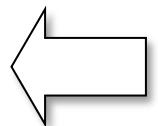


FORMACIÓN PROFESIONAL DUAL



**INFORME MENSUAL
DE FORMACIÓN PRÁCTICA
EN EMPRESA**



CÓDIGO N°

FORMACIÓN PROFESIONAL DUAL

CFP/UCP/ESCUELA: ETI _____

ESTUDIANTE: BRIONES GIL JHON EFRAIN _____

ID: 1522182 _____ BLOQUE: 25628_____

CARRERA: INGENIERIA DE SOFTWARE CON INTELIGENCIA ARTIFICIAL _____

INSTRUCTOR: KILLER PILCO SOTOMAYOR _____

SEMESTRE: QUINTO _____ DEL: 18 DE FEBRERO AL: 27 DE MAYO ____

INSTRUCCIONES PARA EL USO DEL INFORME MENSUAL DE FORMACIÓN PRÁCTICA EN EMPRESA

1. PRESENTACIÓN.

El Informe mensual de Formación Práctica en Empresa es un documento de control, en el cual el estudiante, registra diariamente, durante el mes, las tareas y operaciones que ejecuta en su formación práctica Empresa.

2. INSTRUCCIONES PARA EL USO DEL INFORME MENSUAL DE FORMACIÓN PRÁCTICA.

- 2.1 En el cuadro de rotaciones, el estudiante, registrará el nombre de las áreas o secciones por las cuales rota durante su formación práctica en la Empresa, precisando la fecha de inicio y término.
- 2.2 Con base al PEA publicado en SINFO, el estudiante selecciona el PEA del semestre que está cursando y transcribe el PEA en el informe de práctica.
El estudiante registrará y controlará su avance, marcando en la columna que corresponda.
- 2.3 Si el PEA tiene menos operaciones (151) de las indicadas en el presente formato, puede eliminar alguna página.
- 2.4 En el REGISTRO SEMANAL DE TRABAJOS REALIZADOS, el estudiante anotará diariamente los trabajos que ejecuta en la empresa, indicando el tiempo correspondiente. El día de asistencia a SENATI para las sesiones de tecnología, registrará los contenidos que desarrolla en clase. Al término de cada semana totalizará las horas.
De las tareas realizadas durante el mes, el estudiante **seleccionará la tarea más significativa** y realizará una descripción del proceso de ejecución con esquemas y dibujos correspondientes que aclaren dicho proceso.
Una de las características de la comunicación técnica es que debe contener información relevante y fácil de entender.
- 2.5 Mensualmente, el estudiante **presentará en físico** el informe de la tarea más significativa al Monitor, quien revisará, anotará las observaciones, las recomendaciones que considere y validará con su firma el respectivo informe.
Se recomienda que el monitor solicite al estudiante que **explique o fundamente el informe** que ha elaborado.
- 2.6 El informe validado por el monitor será presentado al instructor correspondiente (mediante carga en el LMS Blackboard), quien revisará y calificará el Informe mensual de Formación Práctica en Empresa haciendo las observaciones y recomendaciones que considere convenientes, en los aspectos relacionados a la elaboración de un Informe Técnico.

CUADRO DE ROTACIONES

PLAN ESPECÍFICO DE APRENDIZAJE (PEA)
CONTROL DE AVANCE

Llenar según avance

Nº	OPERACIONES/TAREAS	OPERACIONES EJECUTADAS*				OPERACIONES POR EJECUTAR	OPERACIONES PARA SEMINARIO
		1	2	3	4		
	DISEÑO Y DESARROLLO DE SOLUCIONES IOT						
01	Revisa los fundamentos de IoT y plataformas de hardware.	X					
02	Programa hardware con Arduino.	X					
03	Usa sensores, actuadores y crea programas de adquisición de datos.	X					
04	Revisa la comunicación y conectividad IoT.	X					
05	Investigación y planificación.	X					
06	Analizar referencias y tendencias de diseño de chatbox.	X					
07	Definir los requisitos y funcionalidades clave.	X					
08	Seleccionar herramientas de diseño.	X					
09	Creación del Wireframe	X					
10	Esbozar un diseño preliminar con disposición básica.	X					
11	Definir elementos clave: área de chat, botón de envío, iconos, etc.	X					
12	Ajustar el flujo de usuario para facilitar la interacción.	X					
13	Diseño del Mockup	X					
14	Aplicación de estilos visuales (colores, tipografía, iconografía).	X					
15	Configuración de elementos gráficos y disposición final.	X					
16	Creación de versiones adaptadas a distintos tamaños de pantalla.	X					
17	Revisión y optimización	X					
18	Recopilación de feedback de colegas o supervisores.	X					
19	Ajustes según comentarios para mejorar UX/UI.	X					
20	Corrección de posibles inconsistencias visuales.	X					
21	Incluir animaciones o transiciones	X					
22	Definir patrones de comportamiento del chat	X					
23	Presentar el mockup explicando el proceso de diseño.	X					
24	Justificar las decisiones estéticas y funcionales con ejemplos.	X					
25	Explicar cómo el diseño mejora la interacción usuario-sistema.	X					

*Número de repeticiones realizadas.

REGISTRO SEMANAL DE TRABAJOS EFECTUADOS

DÍA	TRABAJOS EFECTUADOS	HORAS
LUNES D/M/A	Este día Tuvimos una reunión presentando los casos que debíamos resolver en la empresa.	8 horas
MARTES D/M/A	Este día tuvimos una introducción al curso de diseño y desarrollo de soluciones IOT y lo que haremos durante las semanas.	9 horas
MIÉRCOLES D/M/A	Este día elegimos el caso que queríamos desarrollar, en este caso yo elegí replicar un mockup de una página de chatbox	8 horas
JUEVES D/M/A	Este día empiezo tratando de ver como repicar el mockup, hice la estructura de la página y los espacios donde irá cada cosa.	8 horas
VIERNES D/M/A	Este día hubo revisión de avances en este caso presente la estructura de mi pagina y los colores principales que irán.	8 horas
SÁBADO D/M/A		
TOTAL		

DÍA	TRABAJOS EFECTUADOS	HORAS
LUNES D/M/A	Este día tuvimos una reunión para corregir algunos puntos y despegar dudas.	8 horas
MARTES D/M/A	Este día aprendimos sobre los componentes que se usan para crear sistemas con circuitos y realizamos algunos ejercicios.	9 horas
MIÉRCOLES D/M/A	Este día codifiqué las imágenes de la página que nos dejaron en la empresa porque las quise realizar con código.	8 horas
JUEVES D/M/A	Este día agregue las imágenes al mockup y trate de arreglar algunas imperfecciones en la estructura.	8 horas
VIERNES D/M/A	Este día agregue el Font de las letras y los textos, también agregue botones y definí sus estilos.	8 horas
SÁBADO D/M/A		
TOTAL		

DÍA	TRABAJOS EFECTUADOS	HORAS
LUNES D/M/A	Este día Hubo una reunión para despejar dudas y hacer aclaraciones sobre el proyecto.	8 horas
MARTES D/M/A	Este día vimos las distintas funcionalidades que se le puede dar a los circuitos y hicimos una tienda inteligente en grupo.	9 horas
MIÉRCOLES D/M/A	Este día termine de definir los estilos de la página en los botones recién agregados y texto que tenía problemas.	8 horas
JUEVES D/M/A	Este día subí mi proyecto a GitHub y luego traté de agregar algunos comits y también poner los que había realizado antes.	8 horas
VIERNES D/M/A	Este día tuvimos una reunión para revisión del proyecto, teníamos que presentarlo ya casi terminado.	8 horas
SÁBADO D/M/A		
TOTAL		

DÍA	TRABAJOS EFECTUADOS	HORAS
LUNES D/M/A	Este día tuvimos reunión para despejar dudas y también nos dijeron que teníamos que hacer que la página se traduzca.	8 horas
MARTES D/M/A	Este día presentamos exposiciones de circuitos con mejoras hechas por nosotros y también realizamos ejercicios en Wokwi.	9 horas
MIÉRCOLES D/M/A	Este día me dedique a investigar cómo podía agregar un botón para traducir la pagina y acabar el trabajo.	8 horas
JUEVES D/M/A	Este día implemente el botón y le di los estilos parecidos a la página y agregue algunas transiciones a la página.	8 horas
VIERNES D/M/A	Este día hubo reunión para presentar el proyecto terminado con el traductor implementado y animaciones.	8 horas
SÁBADO D/M/A	-	
TOTAL		

INFORME MENSUAL

Tarea más significativa del mes: ¿Porqué eligió esta tarea y qué operaciones del PEA cumplió con su ejecución?

La tarea que elegí es el proyecto que realizamos en grupo en el que teníamos que automatizar algún espacio con circuitos que facilitara el uso y la estancia en el espacio.

Las operaciones del PEA que cumple son:

Revisa los fundamentos de IoT y plataformas de hardware.

Programa hardware con Arduino.

Usa sensores, actuadores y crea programas de adquisición de datos.

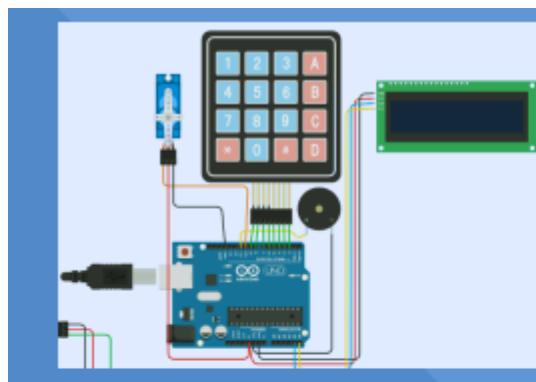
Revisa la comunicación y conectividad IoT.

Descripción del proceso:(Secuencia lógica de la ejecución de la tarea: operaciones, pasos, sub pasos)

1. Buscamos el problema a resolver que en este caso es la automatización de una tienda para tratar de optimizar procesos.
2. Buscar la solución que en este caso será realizar circuitos en Tinkercad para la tienda.
3. Vemos los circuitos que podrían ir y hacemos la guía de cada uno.



4. Lista de circuitos que realizamos:



Inclusión de Librerías

Funcion:

Se incluyen las librerías necesarias para manejar el servo motor, el teclado matricial y la pantalla LCD.

Configuración de la Pantalla LCD

Funcion:

Se inicializa la pantalla LCD con dirección 0x20 y formato de 16x2.

```
const int columnas = 4;
const int filas = 4;
char keys[filas][columnas] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
byte pinesFilas[4] = {9, 8, 7, 6};
byte pinesColumnas[4] = {5, 4, 3, 2};

Keypad teclado = Keypad(makeKeymap(keys),
    pinesFilas, pinesColumnas, filas, columnas);
```

Configuración del Servo Motor

Funcion:

Inicia el servo motor en el pin 10 y lo posiciona en 0° para mantenerlo bloqueado.

```
if (indice == 6) {
    if (strcmp(textoIngresado, pass) == 0) {
        Serial.println(" Acceso permitido");

        // Mostrar "Acceso permitido" en el LCD
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Acceso permitido");

        servo_10.write(90);
        delay(1000);
        servo_10.write(0);
    } else {
        Serial.println(" Acceso denegado");

        // Mostrar "Acceso denegado" en el LCD
        lcd.clear();
```

Objetivo:

El usuario debe ingresar un código en el teclado matricial, y si es correcto, el servo motor se activa para abrir una cerradura, mientras que la pantalla LCD muestra información sobre el estado del sistema. Además, el buzzer va a emitir sonidos de confirmación o alerta en caso de código incorrecto.

Componentes:

Arduino Uno: Microcontrolador que gestiona todas las señales y controla los demás componentes.

Teclado matricial 4x4: Permite la entrada de datos por parte del usuario.

Pantalla LCD 16x2: Muestra información sobre la interacción con el sistema.

Buzzer: Genera alertas sonoras dependiendo de las acciones realizadas.

Servo motor: Controlado por el Arduino, se activa según la entrada del teclado.

```
#include <Servo.h>
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x20, 16, 2);
```

Configuración del Teclado Matricial

Funcion:

Define el teclado matricial de 4x4 y los pines a los que está conectado.

```
Servo servo_10;

void setup() {
    Serial.begin(9600);
    pinMode(11, OUTPUT);
    servo_10.attach(10, 500, 2500);
    servo_10.write(0);
```

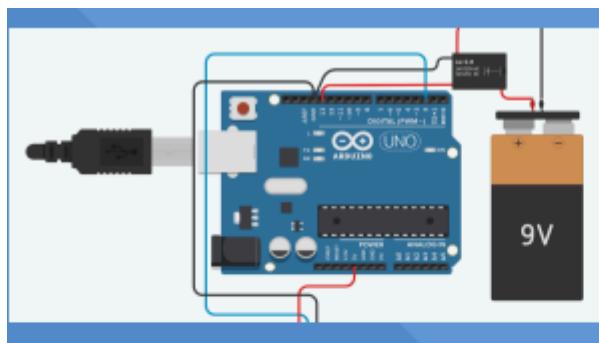
Validación de la Contraseña y Accionamiento del Servo

Funcion:

Si el usuario ingresa 6 caracteres, los compara con la contraseña.

Si es correcta, muestra "Acceso permitido" en la LCD y mueve el servo a 90° (posible apertura de puerta).

Si es incorrecta, muestra "Acceso denegado" y suena una alerta con el buzzer.



Componentes:

Arduino UNO: Microcontrolador que controla y gestiona la lógica del circuito.

Relé LU-5-R: Dispositivo electromecánico que permite controlar el encendido y apagado de un circuito de mayor potencia con una señal de baja potencia proveniente del Arduino.

Batería de 9V: Fuente de energía independiente para alimentar el circuito.

1. Definición de pines

Funcion:

relePin (pin 13): Controla el relé.
interruptorPin (pin 2): Recibe la señal del interruptor.
estadointerruptor: Almacena el estado del interruptor (presionado o no).

```
int relePin = 13;
int interruptorPin = 2;
int estadoInterruptor;
```

```
void setup() {
    pinMode(relePin, OUTPUT);
    pinMode(interruptorPin, INPUT);
}
```

Lectura del estado del interruptor

Funcion:

Lee si el interruptor está presionado (HIGH) o liberado (LOW).
Permite tomar decisiones en función de su estado.

```
if (indice == 6) {
    if (strcmp(textoIngresado, pass) == 0) {
        Serial.println(" Acceso permitido");
        // Mostrar "Acceso permitido" en el LCD
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Acceso permitido");

        servo_10.write(90);
        delay(1000);
        servo_10.write(0);
    } else {
        Serial.println(" Acceso denegado");
        // Mostrar "Acceso denegado" en el LCD
        lcd.clear();
    }
}
```

Objetivo:

Control de encendido/apagado de un dispositivo externo mediante Arduino.
El Arduino activa el relé, permitiendo que la batería de 9V alimente otro circuito.
Este tipo de configuración es común en sistemas automáticos de control de energía, como activación de motores, luces o cerraduras eléctricas.

Configuración de pines en el setup()

Funcion:

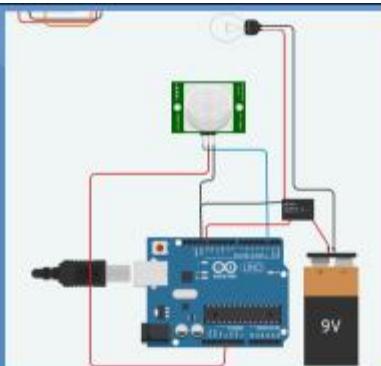
- Configura el relé como salida (para activarlo o desactivarlo).
- Configura el interruptor como entrada (para leer su estado).

```
void loop() {
    estadoInterruptor = digitalRead
    (interruptorPin);
```

Control del relé según el estado del interruptor

Funcion:

Si el interruptor está presionado (HIGH), activa el relé.
Si no está presionado (LOW), lo desactiva.



Componentes:

Arduino Uno: Recibe la señal del sensor PIR y activa el relé para encender/apagar la lámpara.

Sensor PIR (Infrarrojo Pasivo) Detecta el movimiento de personas u objetos con calor. Envía una señal al Arduino cuando detecta presencia.

Relé LU-5-R: Interruptor electrónico que permite activar o desactivar un circuito de mayor voltaje (en este caso, la lámpara de 9V). Recibe la señal del Arduino para cerrar o abrir el circuito de la lámpara.

Batería de 9V: Fuente de alimentación para la lámpara.

Lámpara: Se enciende cuando el sensor PIR detecta movimiento y el relé cierra el circuito.

Definición de pines y variables

Funcion:

Define los pines y variables necesarias para leer los sensores y controlar el relé.

```
int releNO = 13;
int inputPir = 2;
int val = 0;
int resuldoSensorLDR;
int sensorLDR = A0;
```

```
void setup()
{
    pinMode(releNO, OUTPUT);
    pinMode(inputPir, INPUT);
    pinMode(sensorLDR, INPUT);
    Serial.begin(9600);
}
```

Lectura de sensores y control de iluminación

Funcion:

Se leen los sensores PIR (movimiento) y LDR (nivel de luz) para determinar si se activa la luz.

```
if(resuldoSensorLDR<600)
{
    if(val == HIGH)
    {
        digitalWrite(releNO, HIGH);
        delay(2000);
    }
    else{
        digitalWrite(releNO, LOW);
        delay(300);
        }
        else{ digitalWrite (releNO, LOW);
        Serial.println(resuldoSensorLDR);
        delay(500);
        }
    }
```

Objetivo:

Sistema de iluminación automática con detección de movimiento.

Cuando el sensor PIR detecta movimiento, el Arduino activa el relé y enciende la lámpara.

Cuando no hay movimiento, la lámpara se apaga automáticamente.

Configuración en el setup()

Funcion:

Configura los pines como entrada/salida y habilita la comunicación serial.

```
void loop()
{
    val = digitalRead(inputPir);
    resuldoSensorLDR = analogRead(sensorLDR);
```

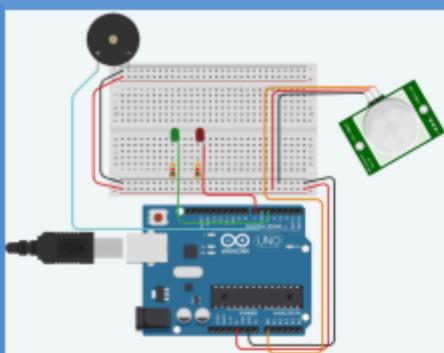
Condición de encendido/apagado de la luz

Funcion:

Si el ambiente está oscuro (LDR < 600) y hay movimiento, se enciende la luz.

Si no hay movimiento, la luz se apaga.

Si hay suficiente luz natural, la luz permanece apagada aunque haya movimiento.



Componentes:

Arduino UNO: Es una placa de microcontrolador de código abierto basada en el microchip ATmega328P.

Protoboard: Es una placa con orificios interconectados eléctricamente que permite crear prototipos de circuitos electrónicos sin necesidad de soldar.

Sensor PIR: Es un sensor electrónico que mide la luz infrarroja (IR) irradiada por los objetos en su campo de visión.

LEDs: Son componentes semiconductores que emiten luz cuando una corriente eléctrica pasa a través de ellos en la dirección correcta.

Resistencias: Son componentes electrónicos que limitan el flujo de corriente en un circuito.

Buzzer: Es un dispositivo electromecánico o piezoelectrónico que produce un sonido (un zumbido) cuando se le aplica una tensión eléctrica.

Configuración de Pines:

Funcion:

Define los componentes de hardware que se utilizarán y cómo interactuará el Arduino con ellos. Se declaran las variables para los pines de los LEDs (indicadores visuales), la bocina (alerta sonora) y el sensor PIR (detección de movimiento). Dentro de la función setup(), se establece la función de cada pin: los LEDs y la bocina como OUTPUT

```
int codigo;
int estado;
int led1= 5;
int led2= 7;
int bocinal= 6;
int sensor= A0;
int con= 1234;

void setup()
{
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(sensor, INPUT);
  pinMode(bocinal, OUTPUT);
  Serial.begin(9600);
}
```

```
if (estado == false){
  Serial.println("Valida tu acceso con el codigo");
  while (Serial.available() == 0);
  codigo = Serial.parseInt();
  if(codigo==con){
    Serial.println("El Sistema de Alarma esta Activado");
    estado=true;
  }
}
if(estado==true){
  codigo=Serial.parseInt();
  if(codigo==con){
    Serial.println("Alarma Desactivado porque no hay movimiento"
    estado=false;
  }
}
```

Detección de Movimiento y encendido de alarma

Funcion:

El sistema reacciona ante la detección de movimiento cuando la alarma está activa. Por lo que al detectar movimiento la alarma sonara hasta que esta se apague con su codigo.

Lógica de Activación

Funcion:

la activación y desactivación de la alarma mediante un código. El sistema solicita un código a través del monitor serial. Si el código ingresado (codigo) coincide con el código predefinido (con), la alarma se activa (estado = true). Y tambien, se permite una desactivación si no hay actividad lo que encendera el led verde

```
if (estado==true){
  if(digitalRead(sensor)==HIGH && estado==true){
    Serial.println("Se Detecto Movimiento");

  while(estado==true){
    tone(bocinal,2500);
    digitalWrite(led1, LOW);
    digitalWrite(led2, HIGH);
  }
}
```

```
if(con==codigo){
  Serial.println("La Alarma se Desactivo Correctamente");
  noTone(bocinal);
  estado=false;
  digitalWrite(led1,LOW);
}
```

Desactivación de la Alarma por Código durante Detección

El sistema continúa sonando la bocina y mostrando el LED rojo por lo que se espera continuamente la entrada de un código a través del monitor. Si el código ingresado coincide con el código de desactivación, se imprimen mensajes de confirmación, se detiene el sonido de la bocina, y se apaga el LED rojo.

Objetivo:

El objetivo principal de este circuito es detectar movimiento utilizando el sensor PIR y proporcionar una respuesta a esta detección a través de indicadores visuales (LEDs) y una alerta sonora (buzzer).

Máquinas, equipos, herramientas y materiales (Listar lo utilizado especificando características, medidas, etc)

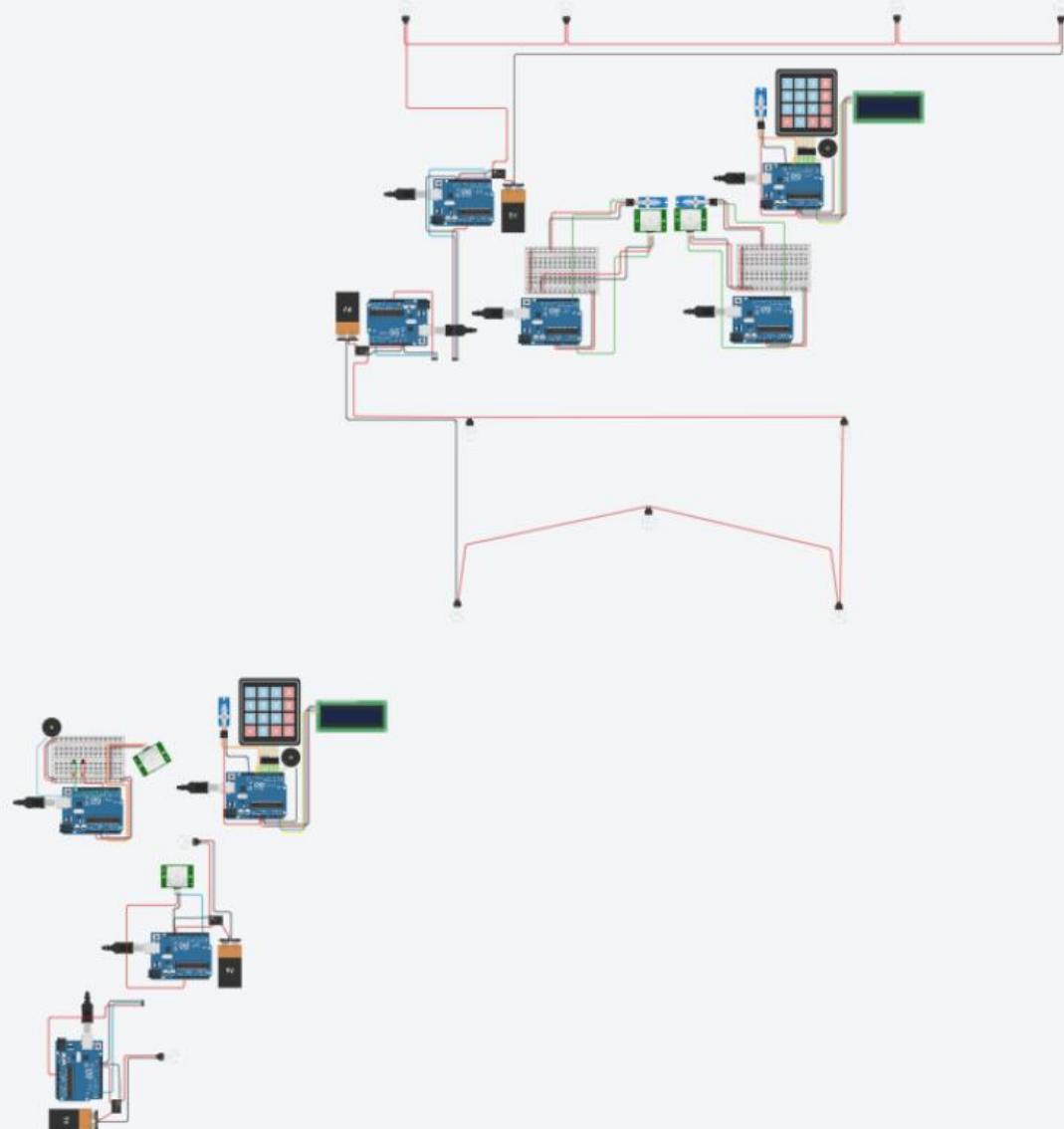
- Pc Windows 11 – Ryzen 5
- Tinkercad
- Wokwi

Seguridad e higiene industrial/ambiental (ATS, Charla de cinco minutos: SST/SGA)**Resultados de la ejecución de la tarea/Recomendaciones (¿Se logró el objetivo que motivó la ejecución de la tarea? Qué recomendaciones sugiere para garantizar la operatividad del bien o servicio realizado**

Si, con este proyecto aprendimos a realizar circuitos que automatizaran procesos en especial en una casa, tienda o empresa. Yo creo que podríamos poner a prueba los circuitos hechos capas en una tienda real para ver cómo funcionan los circuitos en la vida real.

HACER ESQUEMA, DIBUJO O DIAGRAMA

IMAGEN DEL CIRCUITO DE LA TIENDA



EVALUACIÓN DEL INFORME DE TRABAJO MENSUAL

NOTA

OBSERVACIONES Y RECOMENDACIONES DEL MONITOR DE EMPRESA:

FIRMA DEL ESTUDIANTE

FIRMA DEL MONITOR DE EMPRESA

JHON BRIONES GIL



**PROPIEDAD INTELECTUAL DE SENATI. PROHIBIDA SU
REPRODUCCIÓN Y VENTA SIN LA AUTORIZACIÓN
CORRESPONDIENTE**