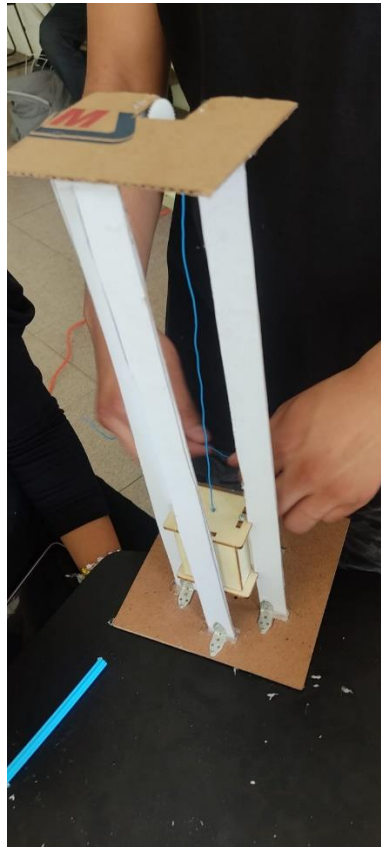




**TECNOLÓGICO
NACIONAL DE MÉXICO**



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



INSTITUTO TECNOLÓGICO DE MILPA ALTA II

TRABAJO – REPORTE PRACTICA ELEVADOR.

Alumna: Hernández Santos Yamilet

Profesor: Edgar Padilla Garces

Asignatura: Inteligencia Artificial

Unidad: 4

Carrera: Ingeniería en Sistemas Computacionales

INTRODUCCION

En la materia de inteligencia artificial, hemos decidido crear un proyecto para dar a conocer más sobre un ascensor con Arduino. concreta mente el proyecto que desarrollamos ha sido, un ascensor que permita subir y bajar entre 5 plantas predeterminadas, En primer lugar, el ascensor que se encuentra en una planta determinada, según el botón que pulsemos es capaz de subir a la planta seleccionada, habiendo previamente nosotros programado el tiempo de subida o bajada del motor, en función de la planta en la que se encuentra el motor, y a la planta a la que pensamos dirigirnos.

MATERIALES

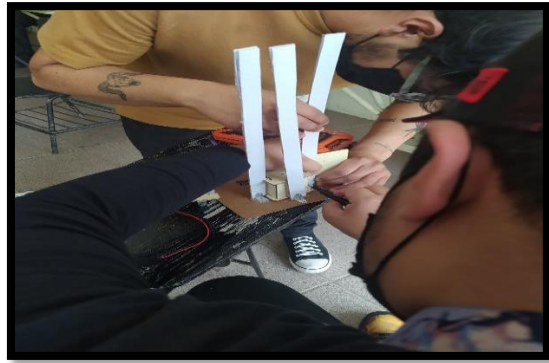
- Arduino Uno
- Puente H
- Pat Numérico
- Motor SDS
- Jumpers
- LSD 16 X 2
- Papel Cascarón
- Caja de Madera
- Tornillos
- Palitos de Bandera
- Hilo
- Soldadura y Cautín

DESARROLLO

Creación de la Maqueta del Elevador

- 1- Lo primero que vamos a hacer es la creación de la maqueta del elevador vamos a cortar el papel cascarón para crear la base del elevador y también vamos a fijarla en una base de cartón.



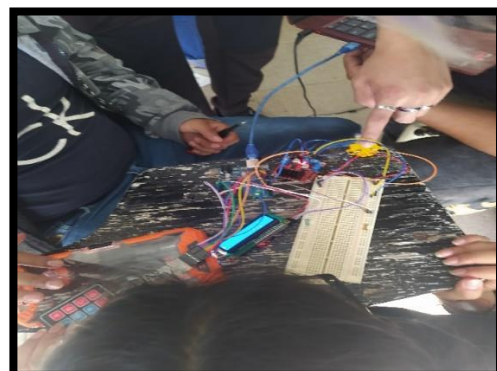
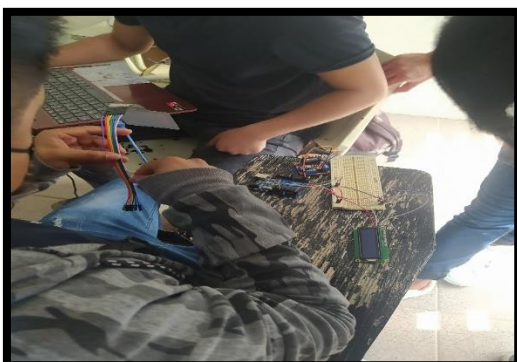


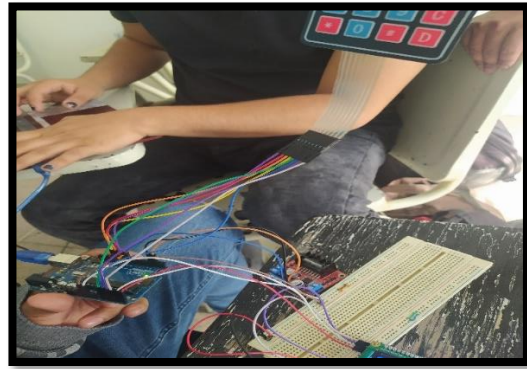
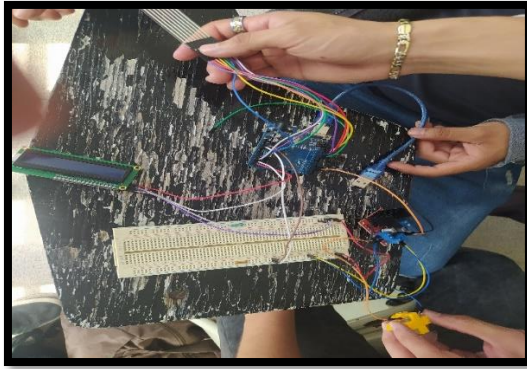
- 2- Después de fijar el soporte y colocar la parte de arriba vamos a ver como se ve la maqueta del elevador terminada colocaremos un hilo el cual ara la función para subir y bajar.



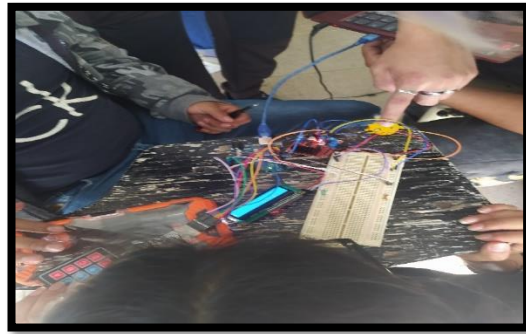
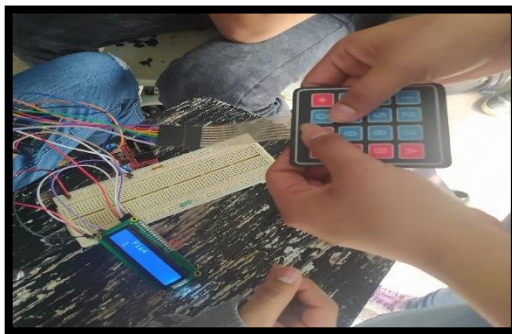
El Armado del Arduino

- 1- En esta parte vamos a conectar con un protovolt el Arduino uno el cual proporciona el control de las operaciones también vamos a conectar el teclado numérico a él Arduino uno con los jumpers. también vamos a conectar la LSD al Arduino uno

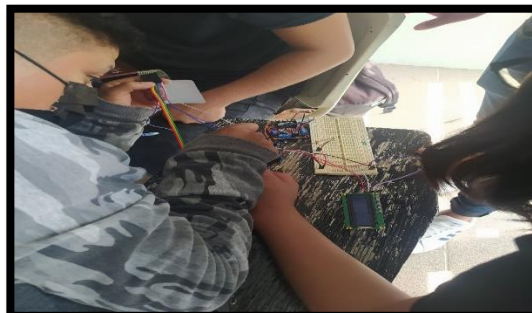
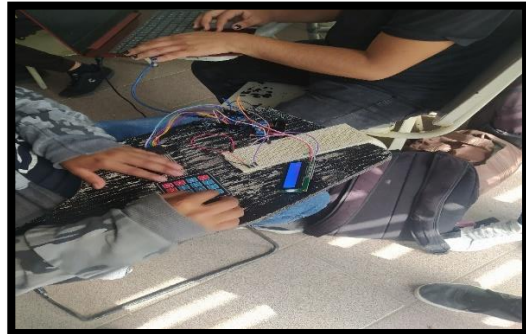
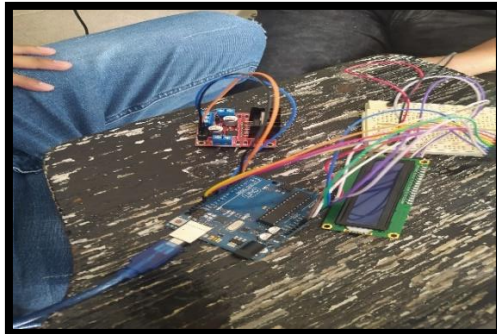




- 2- Se va a comprobar si la LSD y el teclado funcionan, luego vamos a conectar los pines al motor para ver el funcionamiento previamente



- 3- Por último, va a quedar así el armado de Arduino y vamos a cargar el código al Arduino uno.



Elaboración del Programa

Se va a crear el código en Arduino dentro del código vamos a definir los pines para el puente H para control de dirección serán dos y uno para la habilitación del motor luego se hará la configuración de la LSD también se va a configurar el teclado numérico número de filas y columnas se ajustan según la configuración deseada, vamos a crear las variables crearemos uno para el piso actual uno para objetivo y uno para indicar si el elevador está en movimiento también vamos a indicar el tiempo para subir y bajar un piso serán 3 segundos, vamos a configurar los pines para el puente H y motor como salida, vamos a inicializar la LSD.



```
Proyecto | Arduino IDE 2.1.0
File Edit Sketch Tools Help
Select Board

Proyecto.ino
1 #include <LiquidCrystal_I2C.h>
2 #include <Keypad.h>
3
4 // Definición de pines
5 const int motorPin1 = 2; // Pin 1 del puente H para control de dirección
6 const int motorPin2 = 3; // Pin 2 del puente H para control de dirección
7 const int motorEnablePin = 9; // Pin de habilitación del motor
8
9 // Configuración de la pantalla LCD
10 LiquidCrystal_I2C lcd(0x27,20,4);
11
12 // Configuración del teclado numérico
13
14 const byte numRows = 4; // Número de filas
15 const byte numCols = 4; // Número de columnas
16 char keys[numRows][numCols] = {
17   {'1', '2', '3', 'A'},
18   {'4', '5', '6', 'B'},
19   {'7', '8', '9', 'C'},
20   {'*', '0', '#', 'D'}
21 };
22
23 byte rowPins[numRows] = {10, 11, 12, 13}; // Pines de las filas (puedes ajustar los pines según tu configuración)
24 byte colPins[numCols] = {A0, A1, A2, A3}; // Pines de las columnas (puedes ajustar los pines según tu configuración)
25 Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, numRows, numCols);
26
27 // Variables
28 int currentFloor = 1; // Piso actual
29 int targetFloor = 1; // Piso objetivo
30 bool isMoving = false; // Indica si el elevador está en movimiento
31
32 // Tiempo requerido para subir o bajar un piso (ajustar según necesidad)
33 const unsigned long floorTravelTime = 3000; // 3 segundos
34
35 void setup() {
36   // Configurar pines del puente H y motor como salidas
37   pinMode(motorPin1, OUTPUT);
38   pinMode(motorPin2, OUTPUT);
39   pinMode(motorEnablePin, OUTPUT);
40   digitalWrite(motorPin1, LOW);
41   digitalWrite(motorPin2, LOW);
42   digitalWrite(motorEnablePin, LOW);
43
44   // Inicializar la pantalla LCD // initialize the lcd
45   lcd.init();
46   lcd.setCursor(3,0);
47   lcd.print("Piso: ");
```

También se va a inicializar la conexión con el monitor serial también se va a comprobar el funcionamiento de las teclas numéricas con un if también vamos a determinar el movimiento hacia arriba y hacia abajo con un if luego vamos a vamos a encender el motor utilizando un While y un digital White para apagar el motor también vamos a actualizar la LSD con el piso actual

```
Proyecto | Arduino IDE 2.1.0
File Edit Sketch Tools Help
Select Board

Proyecto.ino
48 lcd.setCursor(2,1);
49 lcd.print(currentFloor);
50
51 // Iniciar comunicación con el monitor serial
52 Serial.begin(9600);
53 }
54
55 void loop() {
56 // Comprobar tecla presionada en el teclado numérico
57 char key = keypad.getKey();
58 if (key != NO_KEY && !isMoving) {
59     if (key >= '1' && key <= '5') {
60         targetFloor = key - '0';
61         if (targetFloor != currentFloor) {
62             moveElevator();
63         }
64     }
65 }
66 // Resto del código...
67 }
68 void moveElevator() {
69     isMoving = true;
70     unsigned long startTime = millis();
71
72     // Determinar la dirección del movimiento
73     if (targetFloor > currentFloor) {
74         digitalWrite(motorPin1, HIGH); // Dirección hacia arriba
75         digitalWrite(motorPin2, LOW);
76     } else {
77         digitalWrite(motorPin1, LOW); // Dirección hacia abajo
78         digitalWrite(motorPin2, HIGH);
79     }
80
81     digitalWrite(motorEnablePin, HIGH); // Encender el motor
82
83     while (millis() - startTime <= floorTravelTime) {
84         // Resto del código de control del movimiento (por ejemplo, detección de pisos)
85     }
86
87     digitalWrite(motorEnablePin, LOW); // Apagar el motor
88     currentFloor = targetFloor;
89     isMoving = false;
90
91     // Actualizar la pantalla LCD con el piso actual
92     updateLCD();
93
94     // Mostrar el piso actual y objetivo en el monitor serial
```

Por último, vamos a mostrar el piso actual y objetivo en el monitor serial también vamos a crear la ubicación en la pantalla LSD para imprimir el piso actual y también vamos a crear una línea en el LSD para limpiar el espacio antes de imprimir el nuevo piso.

```
95 Serial.print("Piso actual: ");
96 Serial.println(currentFloor);
97 Serial.print("Piso objetivo: ");
98 Serial.println(targetFloor);
99 }
100
101 void updateLCD() {
102     lcd.setCursor(2, 1); // Ubicación en la pantalla LCD para imprimir el piso actual
103     lcd.print(" "); // Limpiar el espacio antes de imprimir el nuevo piso
104     lcd.setCursor(2, 1);
105     lcd.print(currentFloor);
106 }
```

