Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Organización Computacional

Sección B

Ing. Otto Rene Escobar Leiva

Auxiliar: Carlos Rangel

## **PROYECTO 1**

Plotter Serial

## **Integrantes**

Grupo 3

|  |  |
| --- | --- |
| Nombre | Carné |
| Oswaldo Antonio Choc Cuteres | 201901844 |
| Jencer Hamilton Hernández Alonzo | 202002141 |
| Cristian Raúl Vega Rodríguez | 202010942 |
| Javier Andrés Monjes Solórzano | 202100081 |
| Angel Isaías Mendoza Martínez | 202180003 |
| Estephanie Alejandra Ruíz Pérez | 202201318 |
| Edin Rafael Santizo Barrera | 202202072 |
| Juan Pascual Itzep Coguox | 202202161 |
| Rocío Samaí López Vásquez | 201709035 |
| Diego Andrés Dubón Samayoa | 202202429 |
| Juan José Almengor Tizol | 202212209 |

Guatemala, 26 de abril 2024

## **Índice**

○ Introducción 3

○ Descripción del Problema 4

○ Lógica del Sistema 5

○ Diagramas del Diseño del Circuito 6

○ Código comentado 9

○ Equipo Utilizado 15

○ Diagramas con Explicación 16

○ Manual de Usuario 18

○ Manual Técnico 25

○ Presupuesto 38

○ Conclusiones 39

○ Recomendaciones 40

○ Anexos 41

## **Introducción**

Un plotter es un periférico de computadora que permite dibujar o representar diagramas y gráficos. Existen plotters monocromáticos y de cuatro, ocho o doce colores. El plotter funciona mediante el movimiento de plumas sobre el papel. Cuando la máquina debe realizar un trazo complejo, hace el dibujo muy lentamente debido al movimiento mecánico de las plumas. Esta lógica de funcionamiento hace que los plotters no sean adecuados para pintar superficies, ya que deben pasar las plumas en numerosas ocasiones. En cambio son útiles para la delineación. Las plumas se encuentran dentro de un tambor. El plotter dispone de dos motores paso a paso, que se mueven por el eje X (a lo ancho del papel) y por el eje Y (con movimiento vertical de las plumas o generando el movimiento del papel). Los motores stepper o paso a paso son motores los cuales están hechos para producir movimientos más precisos que los motores DC, estos existen de 2 tipos los cuales son: unipolares y bipolares, a los cuales se les tiene que ingresar una secuencia determinada para moverlos.

Por medio de la lógica del funcionamiento del plotter, se desarrollará una impresora innovadora y un software especializado que permita realizar impresiones de alta calidad con diseños personalizados, satisfaciendo así las necesidades de la universidad y proporcionando una solución tecnológica avanzada para la comunidad educativa.

## **Descripción del Problema**

La Escuela de Ciencias y Sistemas está organizando una demostración de proyectos de innovación en la cual nosotros como alumnos del curso de Organización Computacional, para dicho fin como equipo desarrollamos un proyecto, un nuevo tipo de impresora nada tradicional, dicha impresora será controlada por un software especial diseñado por su nosotros y la cual será controlada desde un PC por medio del puerto (“Serial/Paralelo”).

Lo que se busca como finalidad es que se tenga un sitio web con el juego de totito el cual contendrá 4 figuras y 4 colores. El Tipo de impresora a desarrollar será “Cuadri-Color” la cual será descrita a continuación:

**Requerimientos**

Se elaboraron varios circuitos combinacionales y secuenciales, los cuales serán capaces de manipular un sistema de ejes “X” y “Y” para el prototipado de un “Plotter” el cual replicará en una Hoja de Papel Bond mediante un tipo de lápiz lo modelado en una aplicación de escritorio.

## **Lógica del Sistema**

**Aplicación**

Se desarrolló una aplicación que cuenta con una interfaz gráfica y un lienzo en el cual se puedan realizar los dibujos utilizando el mouse del pc a modo de PixelArt en una matriz de 3 x 3, la aplicación cuenta con las opciones generales como “Archivo” (“abrir”, “nuevo” y “guardar”), “Ayuda”, “Analizar Entrada” e “Imprimir Matriz”, la extensión del archivo debe ser “.orga”. La aplicación también cuenta con una serie de figuras predefinidas.

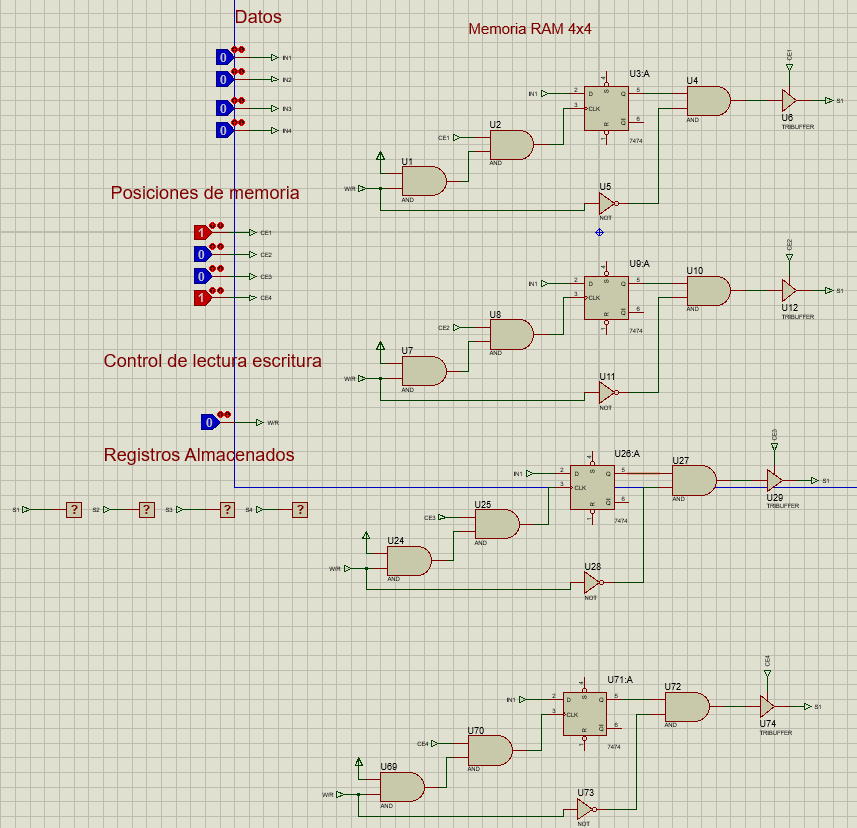
La conexión que se implementó por medio del puerto serial que se utilizó con la interfaz de envío y recepción de datos en forma serial RS232, este se comunica con el fronted mediante el pyserial quien enviará la información a los registros elaborados con flip-flops. La transmisión de datos se detallará con el envío desde la PC hacia el controlador de los motores del Plotter.

La conexión TX y RX; conecta el pin TX (transmisión) del módulo RS232 al pin RX1 (pin 19) del Arduino Mega y el RX (recepción) del módulo al pin TX1 (pin 18) del Arduino Mega. Esto permite la comunicación bidireccional. Se conectaron los pines de alimentación del módulo RS232 a las salidas correspondientes del Arduino Mega, generalmente 5V y GND. Y para la conexión de la led, esta se conecta al pin 13 y GND del Arduino, con una resistencia adecuada en serie para limitar la corriente (utilizada entre 220 y 470 ohmios).

La impresora que se realizó tiene un cabezal de impresión que se desplaza por el área de impresión, esta es capaz de almacenar 3 coordenadas a pintar por medio de la matriz de flip-flop simulada por la memoria RAM, la cual será programada por la PC para luego empezar a imprimir por medio de 2 lectores de CD, también para los controladores del disco se usó un módulo puente H for stepper motor, 2ª, 25w.

## **Diagramas del Diseño del Circuito**

**Memoria de Acceso Aleatoria.**



Diagrama, Esquemático

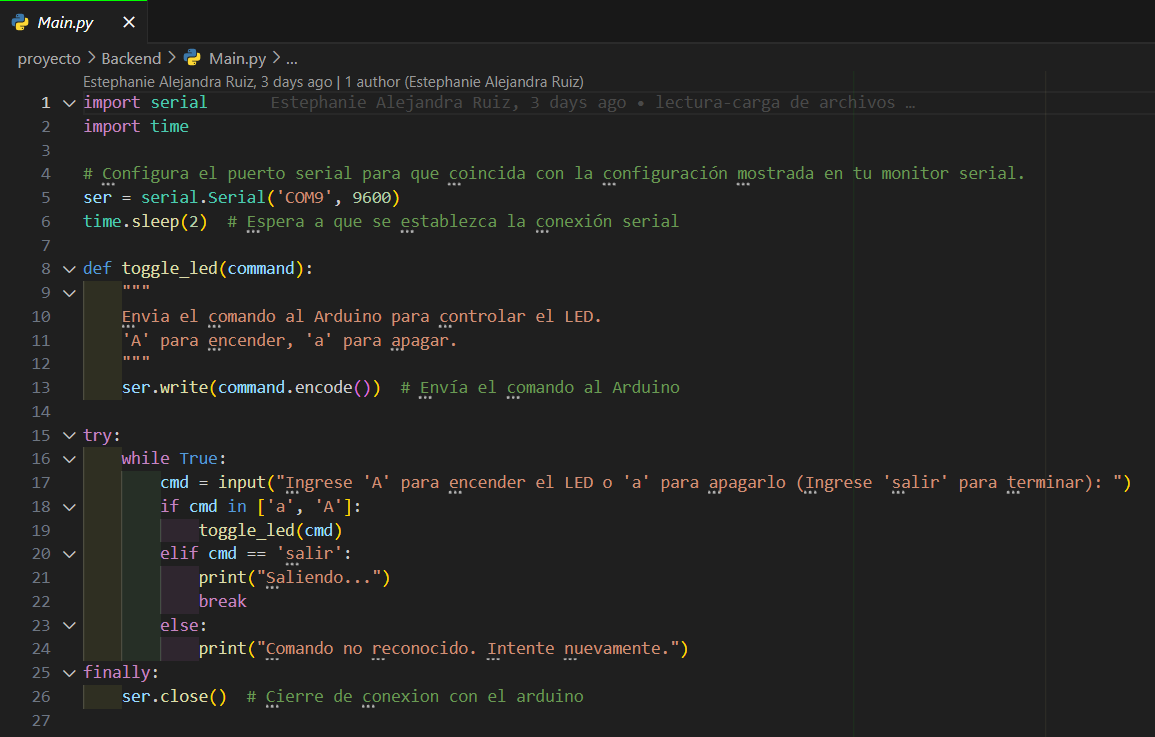
Descripción generada automáticamente

Diagrama, Esquemático

Descripción generada automáticamente

## **Código comentado**

**BACKEND**



Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**FRONTEND**

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

## **Equipo Utilizado**

* Arduino Mega
* Cable USB a serial
* Modulo RS232
* Led
* Resistencia
* Protoboard
* Compuertas 7404 y 7408
* Flip Flop D 74174
* 74LS126 TTL Buffers de Bus
* Modulo Sensor de Color TCS230 TCS3200
* MD – L298N uente uente H for stepper motor, 2A, 25W
* Servomotor
* Material reciclado para la impresora

## **Diagramas con Explicación**

Diagrama

Descripción generada automáticamente

## **Manual de Usuario**

**Plotter Serial**

Primero realizaremos las debidas conexiones de la memoria RAM a la PC que se utilizará y conexión con la impresora que se hizo.



Ingresaremos a nuestra página donde se encuentra la interfaz del juego, con cada una de las figuras que se pueden utilizar dentro la aplicación.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Nuestro menú principal contiene las opciones de “abrir” la cual nos servirá para poder escoger el archivo que la aplicación va a procesar para empezar a ver la colocación de cada figura en la matriz.

Texto

Descripción generada automáticamente

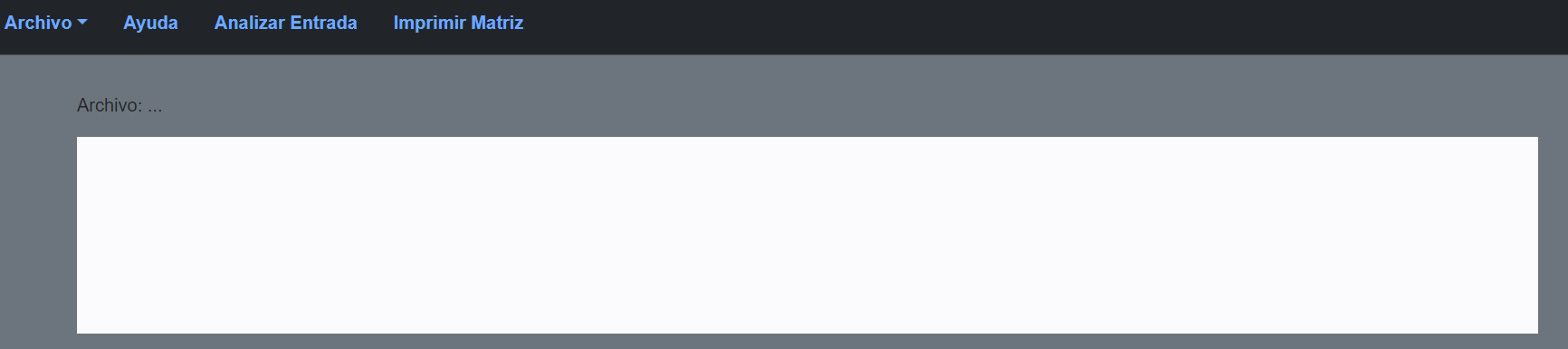
La opción “nuevo” lograra borrar el archivo anterior y poder escoger uno nuevo para iniciar un nuevo juego.

La opción “guardar” creará un archivo para guardar todo el proceso del juego que se realizó.

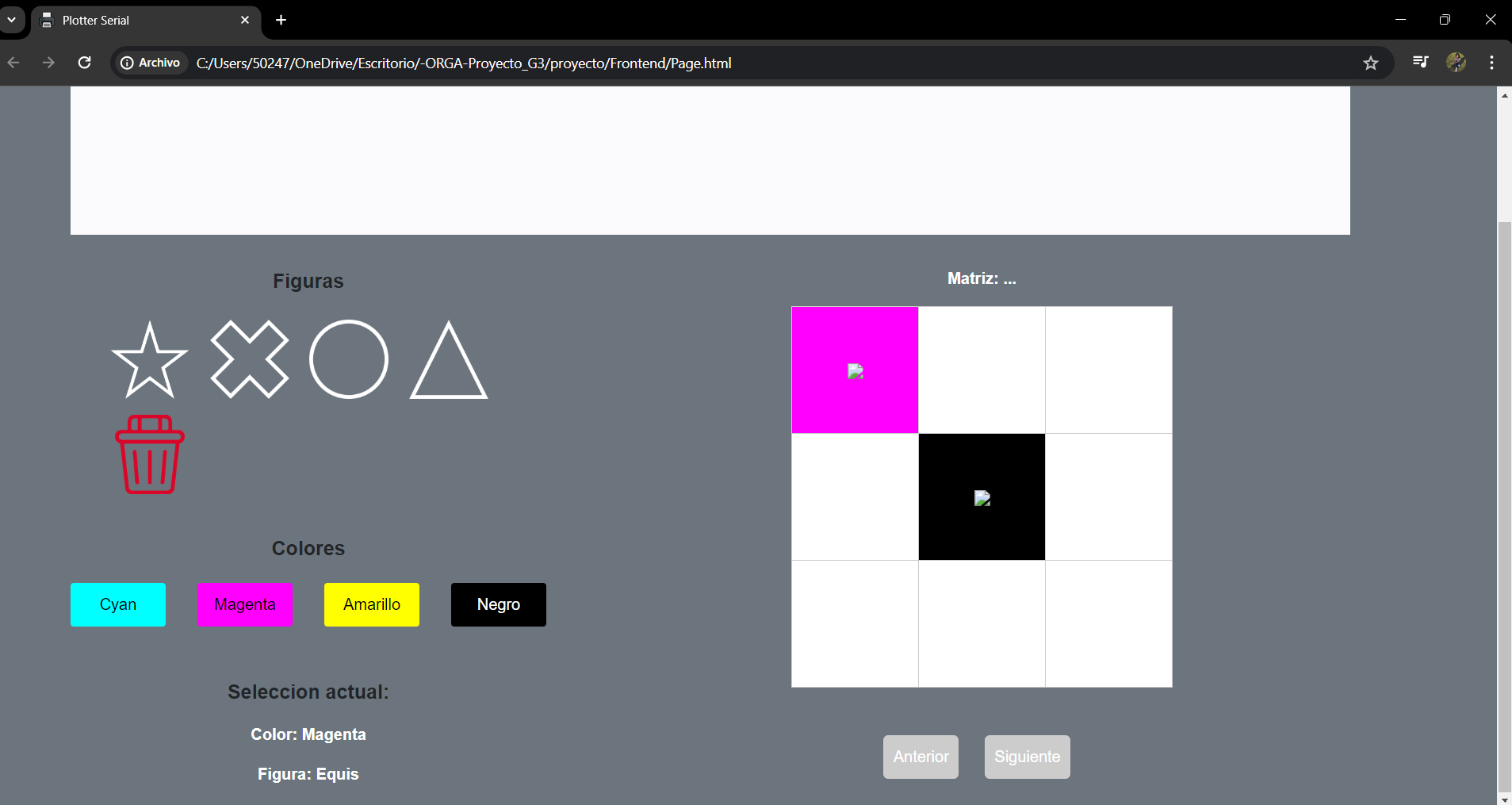
Interfaz de usuario gráfica

Descripción generada automáticamente

Acá se podrá visualizar y editar en vivo el archivo de entrada, esta funcionalidad permite no solo ver el contenido actual del archivo en cuestión, sino también modificarlo directamente en el mismo entorno.



Podrás realizar tu propio archivo, primero tendrás que escoger un color y luego la figura que deseas colocar dentro de la matriz, la extensión del archivo debe ser “.orga”. Y si alguna de las figuras no es la que se deseaba colocar, ésta se podrá eliminar y realizar el mismo procedimiento anterior para colocar la nueva figura.



Interfaz de usuario gráfica, Aplicación, Sitio web, Teams

Descripción generada automáticamente

En nuestra opción “Ayuda” se podrá visualizar los manuales necesarios para poder utilizar mejor las opciones de la aplicación, y sea más fácil para el usuario.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

“Analizar Entrada” podrá verificar cada coordenada que contiene el archivo que se cargó a la aplicación y ésta poder ingresar cada figura en la matriz.

Luego al terminar de agregar todas las figuras necesarias a la matriz, este archivo se guardará y se podrá presionar la opción de “imprimir matriz”, esta opción realizará la conexión con la impresora que se realizó y empezará a imprimir la matriz terminada que se creó en la aplicación.

El área de impresión será una hoja de papel bond (120 gramos) tamaño carta, la cual en cada esquina tendrá 1 indicador, dichos indicadores serán utilizados por sensores de color, el cual permitirá la impresión una vez que los 4 sensores detecten que el área de impresión está correctamente alineada. Se debe indicar por medio de un LED de color “Azul” si todos los sensores poseen una alineación correcta y uno “Amarillo” si más de alguno tienen una alineación incorrecta.

Gráfico, Gráfico de dispersión, Gráfico de burbujas

Descripción generada automáticamente

Imagen que contiene persona, mujer, tabla, sostener

Descripción generada automáticamente

Imagen que contiene Gráfico de dispersión

Descripción generada automáticamente

## **Manual Técnico**

**INTRODUCCIÓN**

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas pueda realizar la instalación del aplicativo creado para la poder utilizar la aplicación del Plotter Serial. Es importante tener en cuenta que en el presente manual se menciona las especificaciones mínimas de hardware y software para la correcta instalación del aplicativo.

**OBJETIVOS**

❏ Mejorar el uso de los distintos tipos de Flip Flops y contadores.

❏ Implementar una transmisión serial a través de los puertos de una PC.

❏ Desarrollar una estructura mecánica funcional.

❏ Aplicar e Implementar conocimientos de registros.

❏ Implementar software para el control de puertos.

❏ Aplicar e Implementar conocimientos de memorias.

❏ Aprender conocimientos de lógica secuencial y simplificación de estados.

❏ Aprender el funcionamiento y uso de motores paso a paso (Stepper).

❏ Aprender el uso de distintos tipos de sensores.

❏ Integrar circuitos eléctricos y producir movimientos mecánicos.

❏ Aplicar todos los conocimientos aprendidos en el curso

❏ Aplicar los conocimientos adquiridos de Arduino.

❏ Aprender la aplicación de una memoria RAM.

**REQUERIMIENTOS TÉCNICOS**

REQUERIMIENTOS MÍNIMOS DE HARDWARE

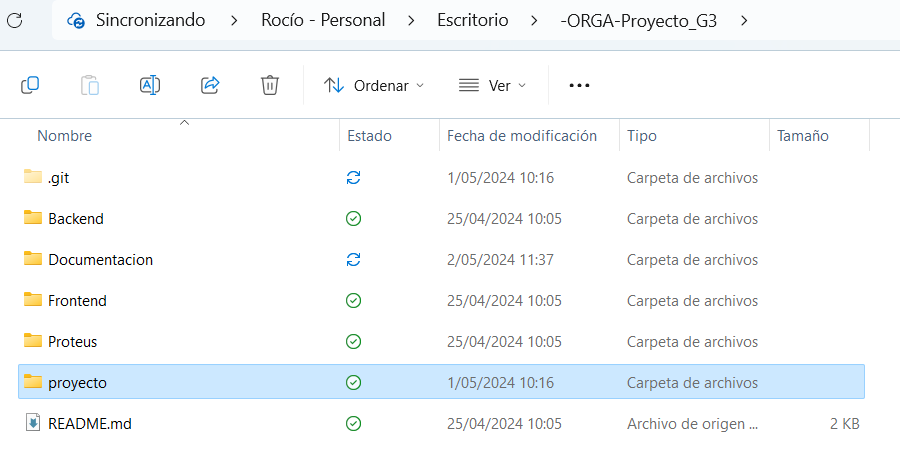
* Procesador
* Memoria RAM. Componentes: 2 sensor color tcs230, 2 motor stepper, 1 servomotor, 74-126, 74-08, 74-04 y 74-174.
* Disco Duro

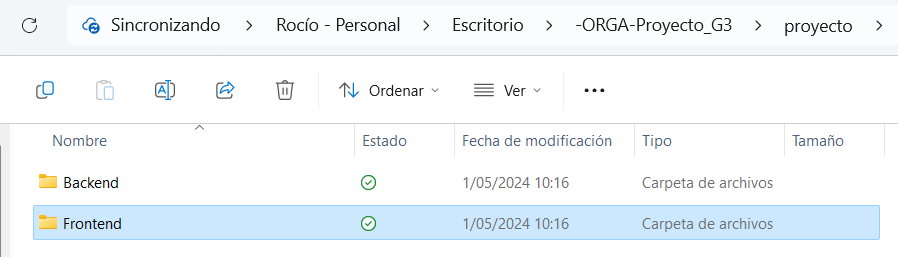
REQUERIMIENTOS MÍNIMOS DE SOFTWARE

* Privilegios de administrador
* Sistema Operativo

**INTALACIÓN**

Se ingresará en la carpeta de proyecto y luego en la de fronted y se podrá abrir la página llamada page.html para poder empezar a utilizar la aplicación.





Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Importar pyserial y CORS para la aplicación.

#import pyserial

app = Flask(\_\_name\_\_)

CORS(app)  # Habilitar CORS para la aplicación Flask

* Rutas para ingresar archivos.

@app.route('/', defaults={'path': ''})

@app.route('/<path:path>')

def serve\_frontend(path):

    if path == "":

        return send\_from\_directory('../Frontend', 'Page.html')

    elif os.path.exists(os.path.join('../Frontend', path)):

        return send\_from\_directory('../Frontend', path)

    else:

        return send\_from\_directory('../Frontend', 'Page.html')

* Analizar archivo.

@app.route('/plotterserial/analizar', methods=['POST'])

def analizar():

    datos = request.json

    codigo\_recibido = datos.get('texto', '')

    if codigo\_recibido == '':

        return jsonify({'error': 'No se recibió código'})

    parse = Gm.parse(codigo\_recibido)

    if not parse:

        return jsonify({

            'mensaje': 'Error en el análisis',

            'impresiones': []

        })

    sentencias = []

    for sent in parse:

        sentencias.append(json.dumps(sent.to\_dict()))

    try:

        return jsonify({

            'mensaje': 'Análisis terminado',

            'impresiones': sentencias

        })

    except Exception as e:

        print(e)

        return jsonify({

            'mensaje': 'Error en el análisis',

            'impresiones': []

        })

* Graficar en la matriz

@app.route('/plotterserial/graficar', methods=['POST'])

def graficar():

    datos = request.json

    matriz\_recibido = datos.get('matriz', '')

    if not matriz\_recibido:

        return jsonify({'error': 'No se recibió matriz'})

    for item in matriz\_recibido:

        fila = item['pos']['fila']

        columna = item['pos']['columna']

        color = item['color']

        figura = item['figura']

        print(f"Fila: {fila}, Columna: {columna}, Color: {color}, Figura: {figura}")

        print("")

        # TODO: Implementar logica impresora con pyserial

    return jsonify({'mensaje': 'Graficación terminada'})

* Configuaración de puerto serial y conexión de LED.

ser = serial.Serial('COM9', 9600)

time.sleep(2)  # Espera a que se establezca la conexión serial

def toggle\_led(command):

    """

    Envia el comando al Arduino para controlar el LED.

    'A' para encender, 'a' para apagar.

    """

    ser.write(command.encode())  # Envía el comando al Arduino

try:

    while True:

        cmd = input("Ingrese 'A' para encender el LED o 'a' para apagarlo (Ingrese 'salir' para terminar): ")

        if cmd in ['a', 'A']:

            toggle\_led(cmd)

        elif cmd == 'salir':

            print("Saliendo...")

            break

        else:

            print("Comando no reconocido. Intente nuevamente.")

finally:

    ser.close()  # Cierre de conexion con el arduino

* Combinaciones de los colores para las figuras.

class Figura:

    def \_\_init\_\_(self, forma, color, vacio):

        self.forma = forma

        self.color = color

        self.vacio = vacio

    def binario(self):

        if self.forma == 'estrella':

            if self.color == 'cyan':

                return '00001b'

            if self.color == 'magenta':

                return '00011b'

            if self.color == 'yellow':

                return '00101b'

            if self.color == 'black':

                return '00111b'

        if self.forma == "equis":

            if self.color == 'cyan':

                return '01001b'

            if self.color == 'magenta':

                return '01011b'

            if self.color == 'yellow':

                return '01101b'

            if self.color == 'black':

                return '01111b'

        if self.forma == "circulo":

            if self.color == 'cyan':

                return '10001b'

            if self.color == 'magenta':

                return '10011b'

            if self.color == 'yellow':

                return '10101b'

            if self.color == 'black':

                return '10111b'

        if self.forma == "triangulo":

            if self.color == 'cyan':

                return '11001b'

            if self.color == 'magenta':

                return '11011b'

            if self.color == 'yellow':

                return '11101b'

            if self.color == 'black':

                return '11111b'

        if self.vacio:

            return '00000b'

* Creación de la matriz y funciones que son las que hacen a que las figuras se puedan cambiar, o cambiarles el color y la posición,

let color = "cyan" // Color de la celda seleccionada

let celda = "celda00" // Celda seleccionada por defecto

let figura = "circulo" // Imagen de la celda seleccionada

// Función que cambia el color de la celda seleccionada

const celdas = document.getElementsByClassName("celda\_mat")

const lbl\_color\_selec = document.getElementById("color\_seleccionado")

const lbl\_figura\_selec = document.getElementById("figura\_seleccionado")

// Cuando se hace click en una celda, se llama a la función que la cambia y se pasa su id

for (var i = 0; i < celdas.length; i++) {

    celdas[i].addEventListener("click", function() {

        celda = this.id

        cambiar\_color()

        cambiar\_imagen()

    });

}

// Función que cambia el color de la celda seleccionada

function cambiar\_color() {

    document.getElementById(celda).style.backgroundColor = color

}

// Función que cambia la imagen de la celda seleccionada

function cambiar\_imagen() {

    document.getElementById(celda).innerHTML = "";

    if (figura == "") {

        return

    }

    var img = document.createElement("img")

    img.src = "/images/" + figura + ".png"

    document.getElementById(celda).appendChild(img)

}

// Funciones que cambian el color de la celda seleccionada

document.getElementById("col\_cyan").addEventListener('click', function(e) {

    e.preventDefault()

    color = "cyan"

    lbl\_color\_selec.innerHTML = "Color: Cyan"

})

document.getElementById("col\_yellow").addEventListener('click', function(e) {

    e.preventDefault()

    color = "yellow"

    lbl\_color\_selec.innerHTML = "Color: Amarillo"

})

document.getElementById("col\_magenta").addEventListener('click', function(e) {

    e.preventDefault()

    color = "magenta"

    lbl\_color\_selec.innerHTML = "Color: Magenta"

})

document.getElementById("col\_black").addEventListener('click', function(e) {

    e.preventDefault()

    color = "black"

    lbl\_color\_selec.innerHTML = "Color: Negro"

})

// Funciones que cambian la imagen de la celda seleccionada

document.getElementById("fig\_circulo").addEventListener('click', function(e) {

    e.preventDefault()

    figura = "circulo"

    lbl\_figura\_selec.innerHTML = "Figura: Circulo"

})

document.getElementById("fig\_triangulo").addEventListener('click', function(e) {

    e.preventDefault()

    figura = "triangulo"

    lbl\_figura\_selec.innerHTML = "Figura: Triángulo"

})

document.getElementById("fig\_equis").addEventListener('click', function(e) {

    e.preventDefault()

    figura = "equis"

    lbl\_figura\_selec.innerHTML = "Figura: Equis"

})

document.getElementById("fig\_estrella").addEventListener('click', function(e) {

    e.preventDefault()

    figura = "estrella"

    lbl\_figura\_selec.innerHTML = "Figura: Estrella"

})

document.getElementById("fig\_eliminar").addEventListener('click', function(e) {

    e.preventDefault()

    figura = ""

    color = "white"

    lbl\_color\_selec.innerHTML = "Color: ..."

    lbl\_figura\_selec.innerHTML = "Figura: Eliminar"

})

// MANEJO APIS

const btn\_impimir = document.getElementById('navbar\_item\_imprimir')

btn\_impimir.addEventListener('click', async function(e) {

    e.preventDefault()

    console.log("Imprimiendo...")

    //obtener informacion de todas las celdas de la matriz

    let matriz = []

    for (var i = 0; i < celdas.length; i++) {

        let celda = celdas[i]

        let pos = { "fila": celda.id[5], "columna": celda.id[6] }

        let color = celda.style.backgroundColor

        let figura = celda.innerHTML.includes("img") ? celda.innerHTML.split("/")[3].split(".")[0] : ""

        // verificar si la celda tiene una imagen

        if (figura != "") {

            matriz.push({ pos, color, figura })

        }

    }

    // Verificar si la matriz esta vacia

    if (matriz.length == 0) {

        alert("Matriz vacía")

        return

    }

    console.log(matriz)

    try {

        // Llamada a la api

        const response = await fetch('http://localhost:4000/plotterserial/graficar', {

            method: 'POST',

            headers: {

                'Content-Type': 'application/json'

            },

            body: JSON.stringify({ 'matriz': matriz })

        })

        const data = await response.json()

        alert(data.mensaje)

    } catch (error) {

        console.log(error)

    }

})

## **Presupuesto**

|  |  |  |
| --- | --- | --- |
| Facturas | Monto Total | Monto Individual (c/u) |
| Factura 1 | Q224.00 | Q20.36 |
| Factura 2 | Q35.00 | Q3.18 |
| Factura 3 | Q68.00 | Q6.18 |
| Sin factura | Q35.00 | Q3.18 |
| Factura 4 | Q6.00 | Q0.55 |
| Factura 6 | Q43.00 | Q3.91 |
| Factura 7 | Q65.00 | Q5.91 |
| TOTAL | **Q476.00** |  |

## **Aporte**

|  |  |
| --- | --- |
| Nombre | Aporte |
| Oswaldo Antonio Choc Cuteres | Maqueta |
| Jencer Hamilton Hernández Alonzo | RAM |
| Cristian Raúl Vega Rodríguez | Código Back y Arduino |
| Javier Andrés Monjes Solórzano | Maqueta |
| Angel Isaías Mendoza Martínez | RAM |
| Estephanie Alejandra Ruíz Pérez | Código Front y Back |
| Edin Rafael Santizo Barrera | Código Front y Back |
| Juan Pascual Itzep Coguox | RAM |
| Rocío Samaí López Vásquez | Documentación |
| Diego Andrés Dubón Samayoa | RAM |
| Juan José Almengor Tizol | Código Front y Back |

## **Conclusiones**

* Por medio de la aplicación y práctica de los Flip-Flops y contadores, nos permitió comprender mejor su funcionamiento y aplicaciones en circuitos secuenciales, lo que facilitó el diseño del sistema para que fuera más eficiente.
* Al principio se encontraron varios obstáculos para la conexión serial pero por medio de varias pruebas se logró obtener la transmisión serial tanto con el dominio del hardware como con el software necesario.
* La capacidad de diseñar y construir estructuras mecánicas funcionales demuestra una compresión integral de los principios de la ingeniería y su aplicación práctica en este proyecto.
* El uso de registros y memorias es fundamental en el almacenamiento y procesamiento de datos en sistemas digitales, lo que permite la creación de sistemas más complejos con una solución vial.

## **Recomendaciones**

* Seguir explorando y experimentando con diferentes Flip-Flops y contadores, ya que esto puede llegar a profundizar la compresión y descubrir nuevas aplicaciones en circuitos secuenciales.
* Considerar cuidadosamente las herramientas y tecnologías que se utilizaran para el desarrollo del software y el control de la impresora. Asegurarse de elegir las adecuadas para el tipo de aplicación que se desarrollo y permitir el cumplimiento de los requisitos establecidos.
* Realizar pruebas exhaustivas en todas las etapas del desarrollo para garantizar que el software y el hardware funcionen correctamente y cumplas los requisitos. Corregir problemas que surjan durante las pruebas y realizar pruebas de integración para asegurarse de que cada uno de los componentes funcionen correctamente.

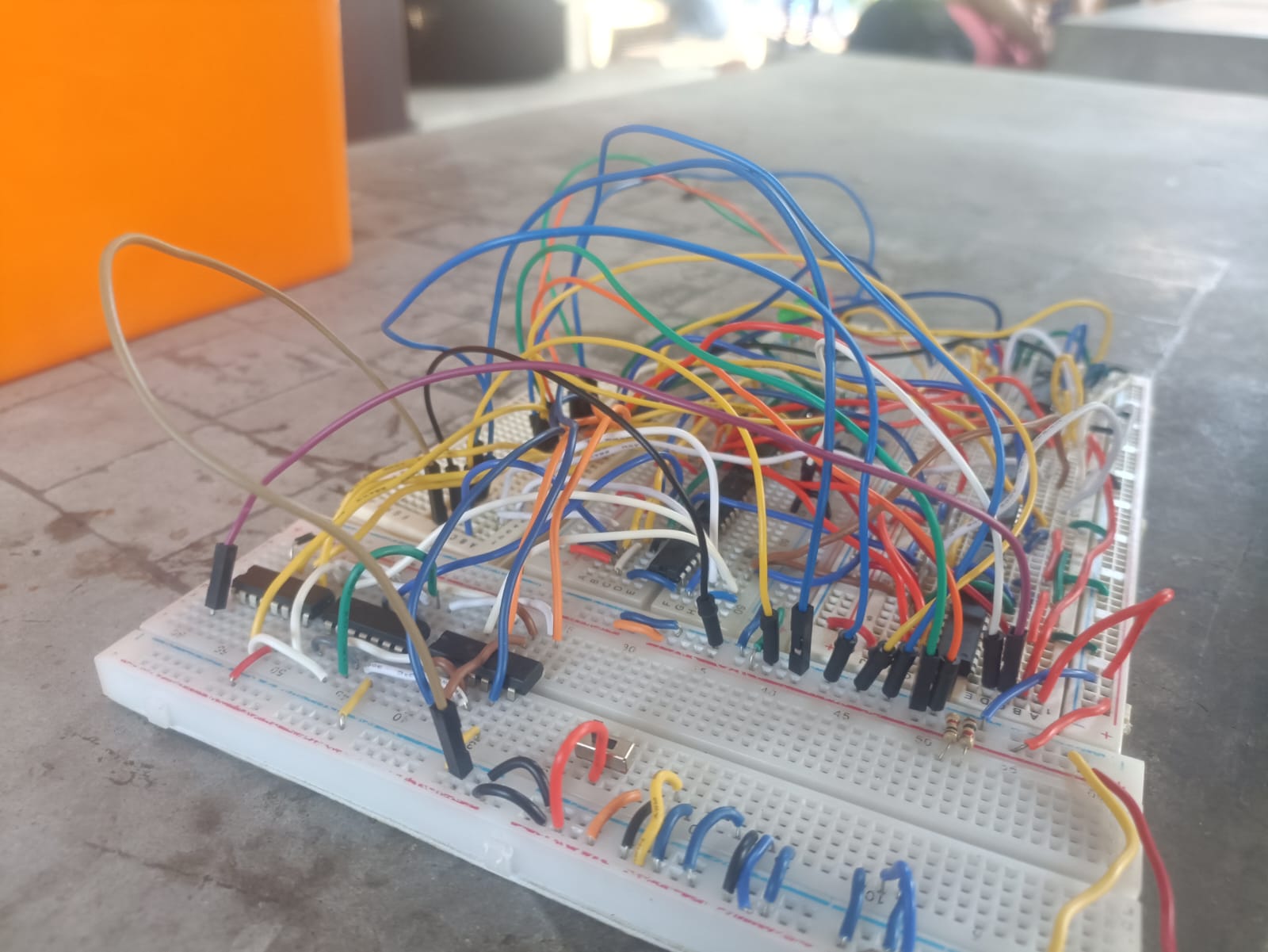
## **Anexos**

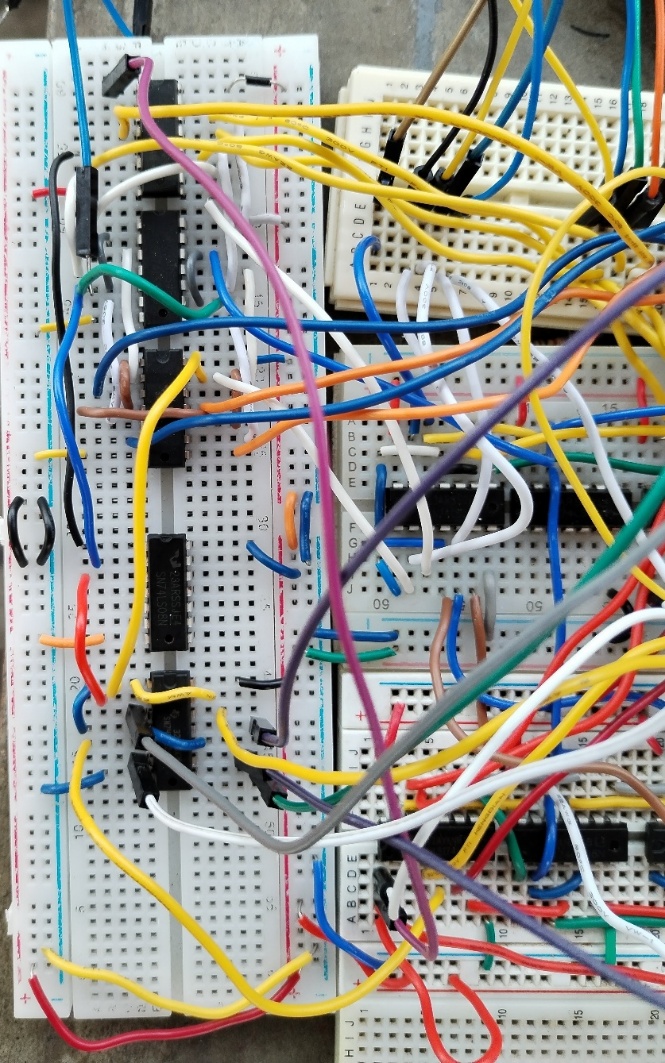
**Interfaz**

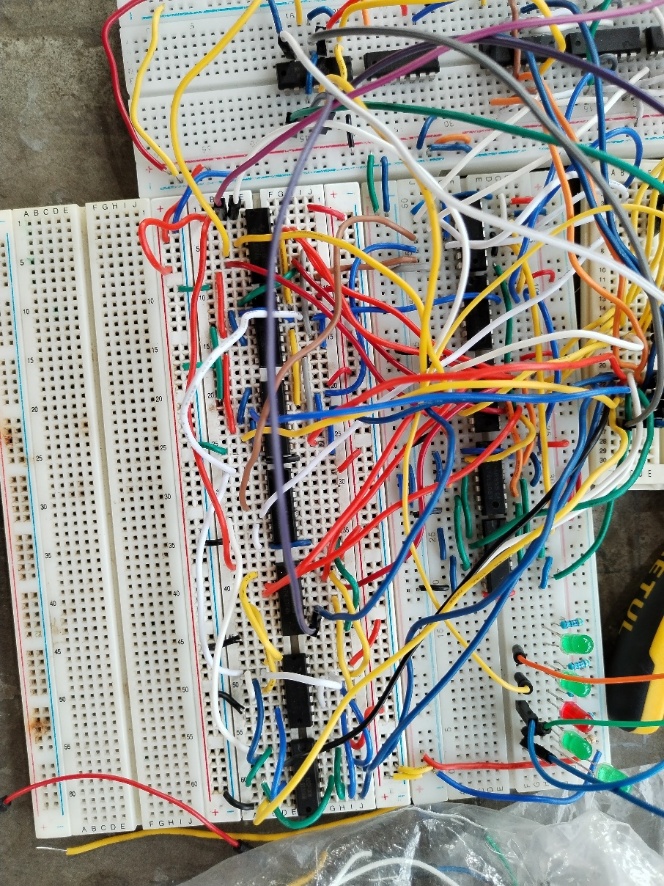
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**RAM**







**Impresora**

Pantalla de un video juego

Descripción generada automáticamente con confianza bajaImagen que contiene persona, mujer, tabla, sostener

Descripción generada automáticamente

**Maqueta final de la Impresora**

Imagen que contiene computadora, computer, escritorio, tabla

Descripción generada automáticamente