

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Laboratorio de Organización Computacional
Sección “C”

Proyecto 1 Plotter Serial

No. Carné	Nombre
201800665	Samuel Alejandro Pajoc Raymundo
201901441	Kevin Manuel Véliz Gálvez
201907769	Byron Enrique Rumpich Sal
202103250	Oscar David Padilla Vasquez
202105212	Aaron Abdam Saravia Martinez
201801608	Aaron Emanuel Trujillo Ibarra
202207595	Pamela Abigail Ratzam Xajil
201900081	José Luis Saloj Julajuj
202101927	Miguel Adrian Tubac Agustin
201807028	Balam Jose Tomas Aju
202200031	Edison Mauricio García Rodríguez
202200061	Jonatan Samuel Rojas Maeda

Índice

- ✓ **Introducción**
- ✓ **Descripción del Problema**
- ✓ **Lógica del Sistema**
- ✓ **Funciones Booleanas y Mapas de Karnaugh**
- ✓ **Diagramas de Estado**
- ✓ **Diagramas del Diseño del Circuito**
- ✓ **Código comentado**
- ✓ **Equipo Utilizado**
- ✓ **Diagramas con Explicación**
- ✓ **Manual de Usuario**
- ✓ **Manual Técnico**
- ✓ **Presupuesto**
- ✓ **Conclusiones**
- ✓ **Recomendaciones**
- ✓ **Enlace del Vídeo**

Introducción

Los plotters son dispositivos informáticos especializados que se utilizan para crear representaciones gráficas precisas y detalladas. A diferencia de las impresoras convencionales, los plotters no imprimen mediante tinta o tóner, sino que utilizan plumas o marcadores para trazar líneas continuas en papel u otros medios. Los plotters funcionan mediante el movimiento controlado de plumas sobre el papel. Cuando se requiere un trazo complejo, la máquina realiza el dibujo lentamente debido al movimiento mecánico preciso de las plumas.

La Escuela de Ciencias y Sistemas ha coordinado una exhibición de proyectos de innovación, donde como estudiantes del curso de Organización Computacional participaremos mostrando nuestras habilidades y creatividad. En esta ocasión, se nos ha encomendado la tarea de desarrollar un proyecto que fomente la innovación tecnológica en el campo de la impresión.

Nuestro equipo se ha sumergido en la creación de una nueva generación de impresoras, dándole un giro a lo tradicional. Durante este proyecto se trabajó en una impresora “Cuadri-Color” que promete cambiar la forma en que interactuamos con la impresión. Controlada por un software especializado, esta impresora ofrecerá una experiencia única al usuario, permitiendo la impresión de diseños complejos en papel Bond mediante un sistema de ejes “X” e “Y”.

Para complementar esta innovadora impresora, también se desarrolló una aplicación de escritorio que revolucionará la forma en que diseñamos gráficos. Con una interfaz intuitiva y características avanzadas como la edición y el guardado, esta aplicación permitirá a los usuarios crear dibujos de estilo PixelArt en una matriz de 3x3 utilizando el mouse de su PC.

Descripción del Problema

El problema descrito implica el desarrollo de un proyecto de innovación que consiste en la creación de una impresora no convencional controlada por un software especial diseñado por el equipo. Esta impresora, denominada "Cuadri-Color", tiene como objetivo imprimir figuras y colores específicos replicados desde una aplicación de escritorio. Aquí hay algunos aspectos clave del problema:

1. Desarrollo del Hardware:

- Se requiere la elaboración de circuitos combinacionales y secuenciales para controlar un sistema de ejes "X" y "Y" que manipulará un "Plotter". Este Plotter imprimirá en papel bond utilizando un lápiz, reproduciendo lo modelado en la aplicación de escritorio.

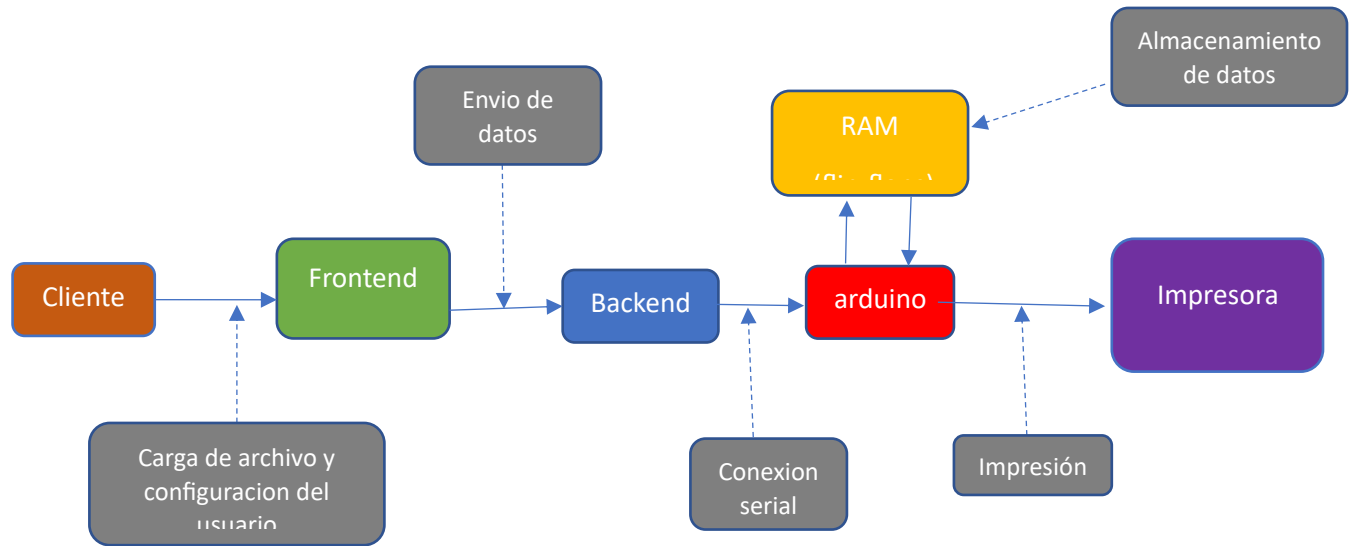
2. Desarrollo de la Aplicación:

- La aplicación de escritorio debe contar con una interfaz gráfica que permita realizar dibujos tipo PixelArt en una matriz de 3x3 utilizando el mouse del PC.
- Debe incluir opciones como "Abrir, Editar, Guardar, Guardar Como e Imprimir", con la extensión de archivo ". orga".
- La aplicación debe tener figuras predefinidas y opciones para ver documentación acerca de su funcionamiento.

3. Interfaz PC y Transmisión de Datos:

- Se requiere una conexión serial entre la PC y el controlador de motores del Plotter, que puede ser implementada a través del puerto paralelo LPT1 o del puerto serial.
- Si se utiliza el puerto paralelo, se debe simular la transmisión serial utilizando solo 2 pines.
- Los datos enviados hacia el circuito de registros del Plotter deben cumplir con la simulación serial, pero la elección específica de los datos enviados queda a discreción del estudiante

Lógica del Sistema



Funciones Booleanas y Mapas de Karnaugh

Símbolo y tabla de verdad del flip flop tipo D

D Flip-flop

Symbol

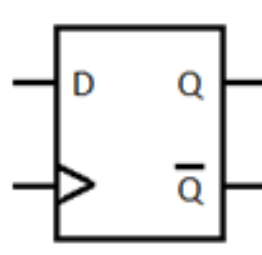


Table of truth:

clk	D	Q	\overline{Q}
0	0	Q	\overline{Q}
0	1	Q	\overline{Q}
1	0	0	1
1	1	1	0

Mapa de Karnaugh y función booleana del flip flop tipo D

Q_n

D	0	1
0	0	X
1	1	X

$$J = D$$

Diagramas de Estado

Diagramas del flip flop tipo D

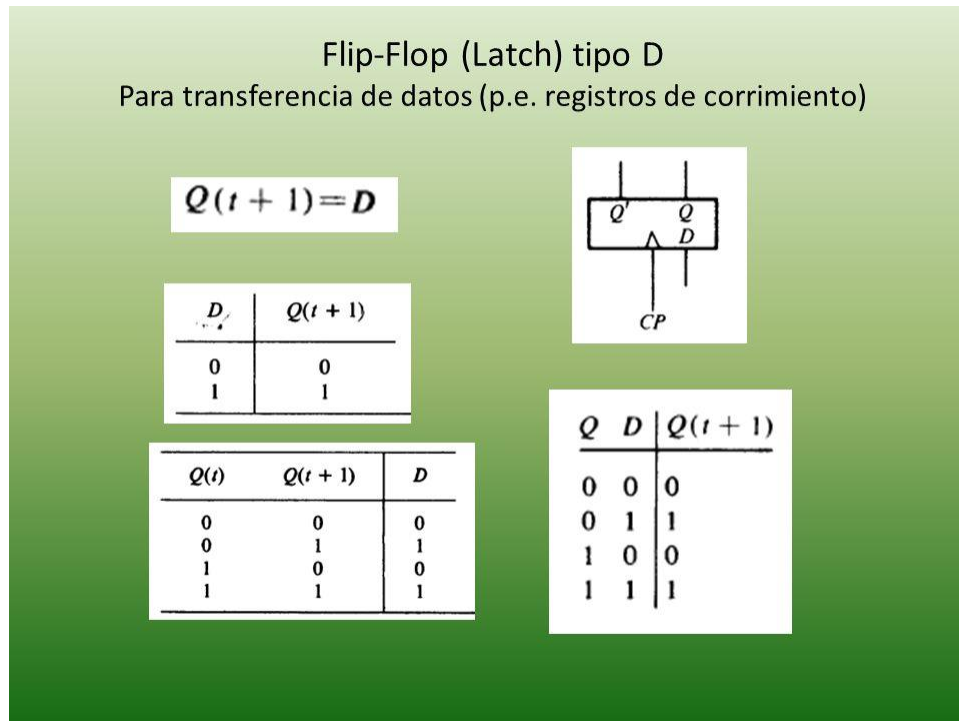


Tabla de verdad y de transiciones del flip flop tipo D

Tabla de verdad

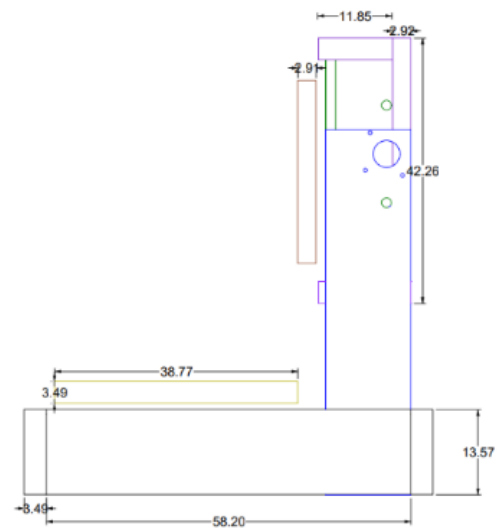
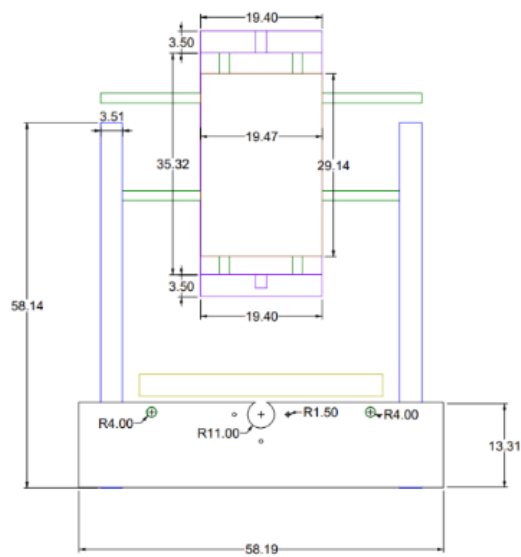
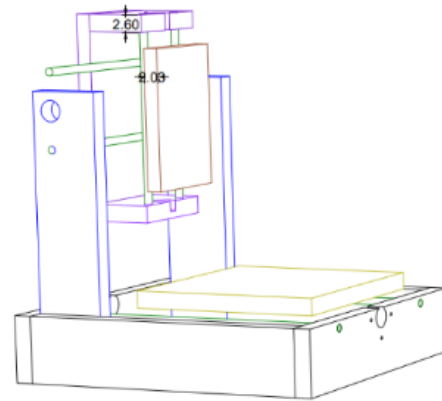
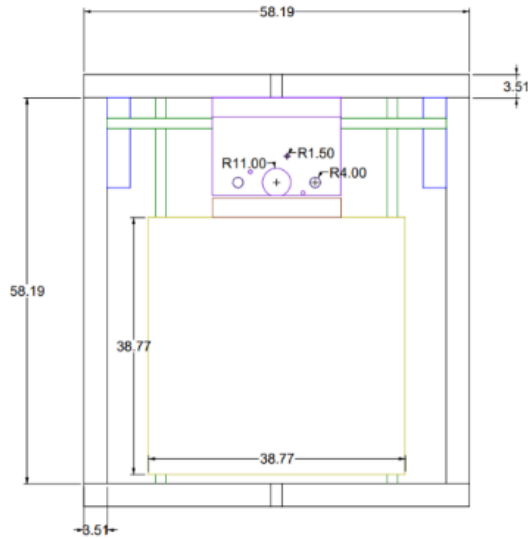
D	Q
0	0
1	1

Tabla de transiciones

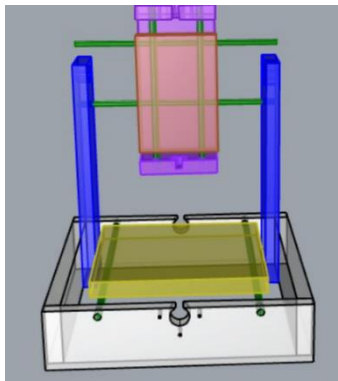
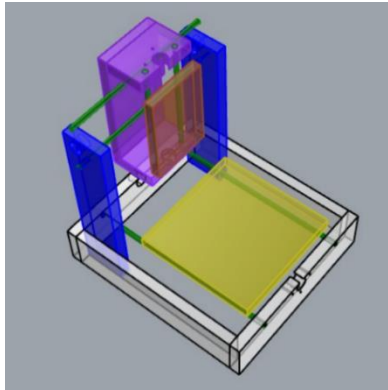
D	Q _{previa}	Q
0	0	0
1	0	1
0	1	0
1	1	1

Diagramas del Diseño

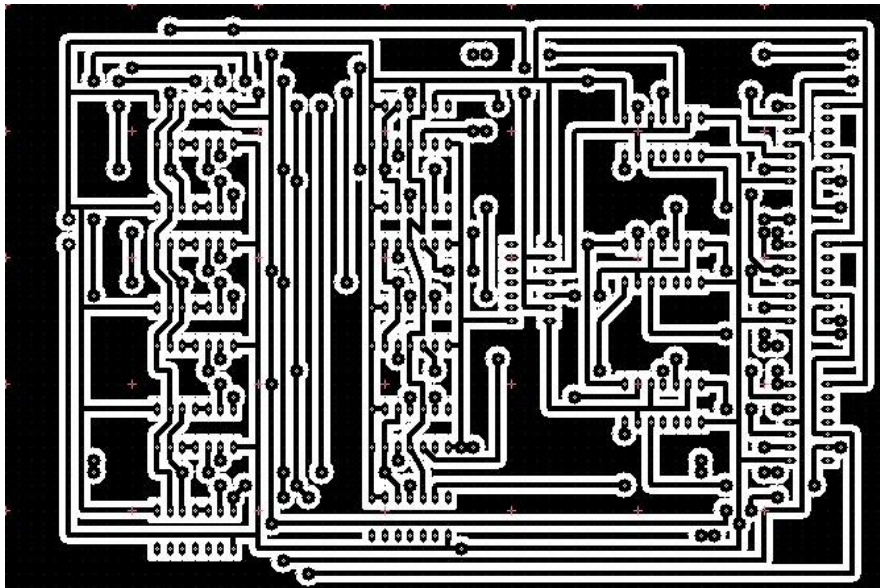
- Plotter



- 3D Plotter



- Placa

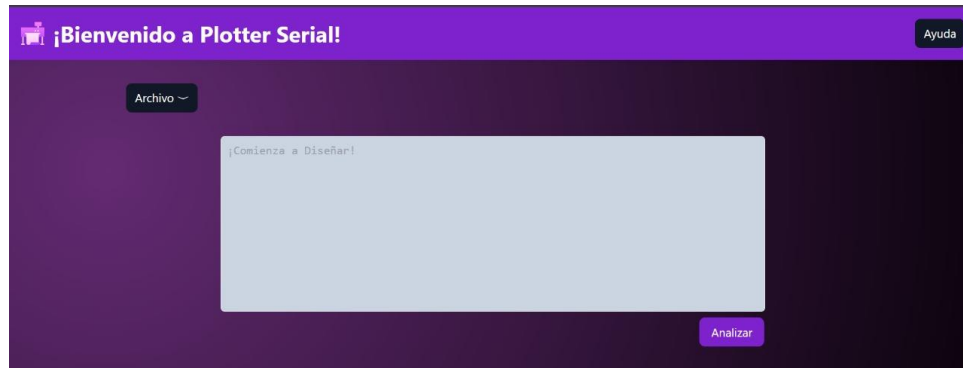


Equipo Utilizado

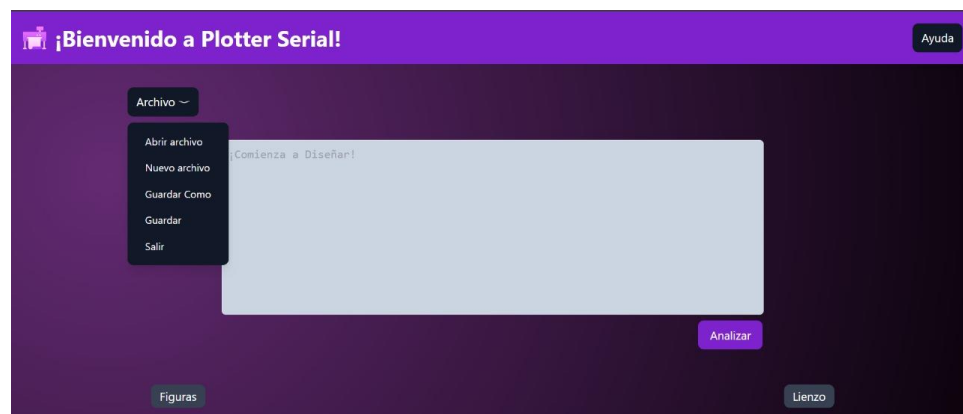
Nombre del Equipo	Cantidad
Motores Stepper	4
Sensores de color	4
Tablas de madera	-----
Cojinete lineal 8mm	12
Acople flexible	3
Eje Acerado	5
Tornillo M3	12
Tornillo M4	24
Varilla roscada 500mmx8mm	2
Varilla roscada 200mmx8mm	1
Cojinete 608zz	3
Flip Flop D	15
Compuertas AND, OR etc.	20
Resistencias	20
Cable y Jumpers	-----
Dremel	1
Placa	1
Cautin	1

Manual de Usuario

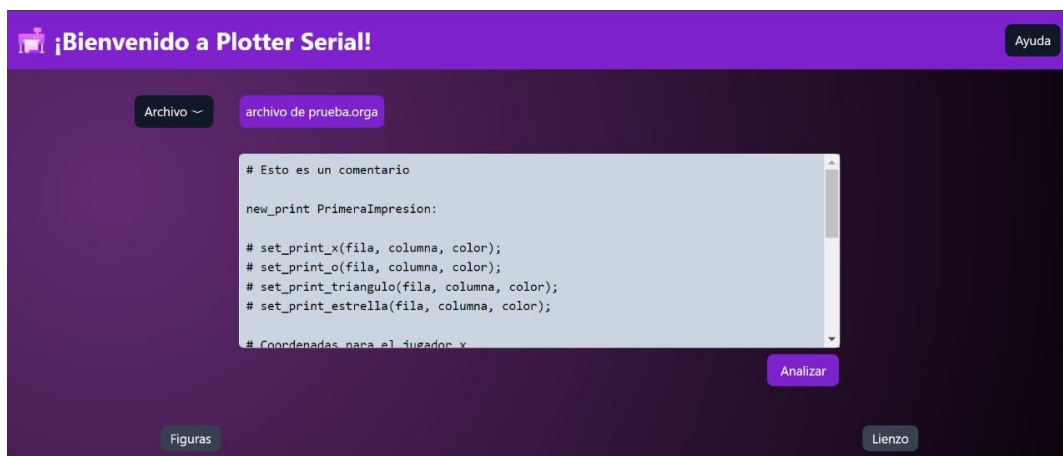
El usuario visualiza una página web



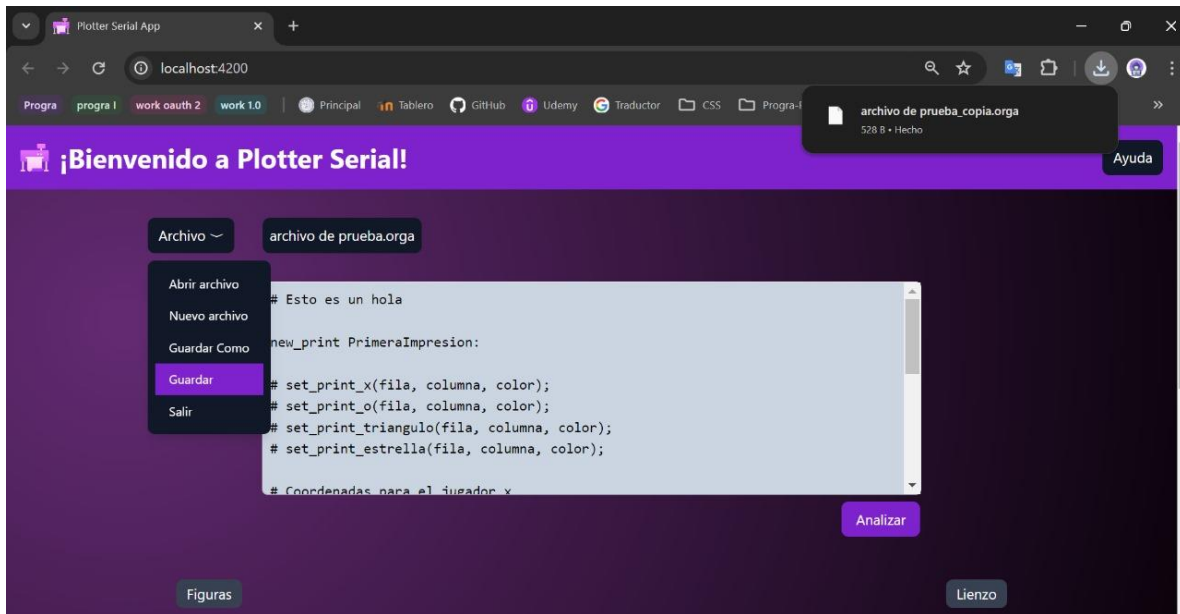
Selecciona el archivo de entrada a ejecutar



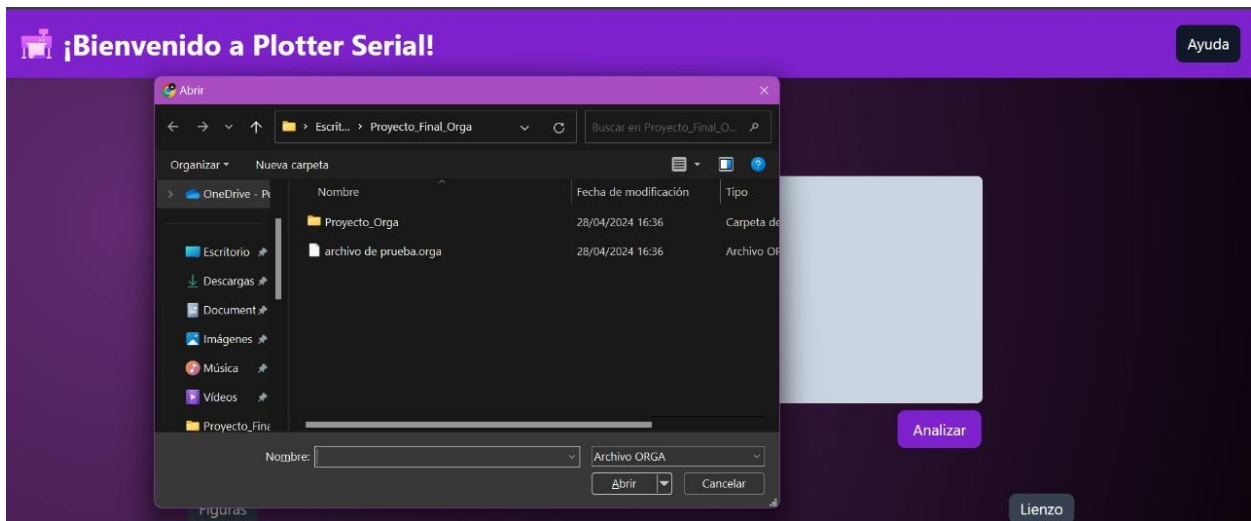
Se carga y se visualiza los datos



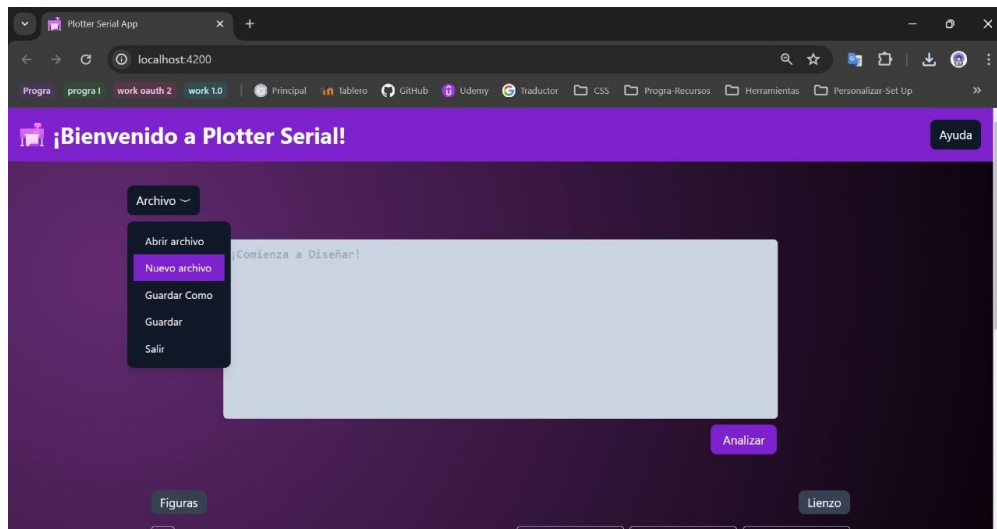
El usuario puede guardar y crear una copia del archivo de entrada actual



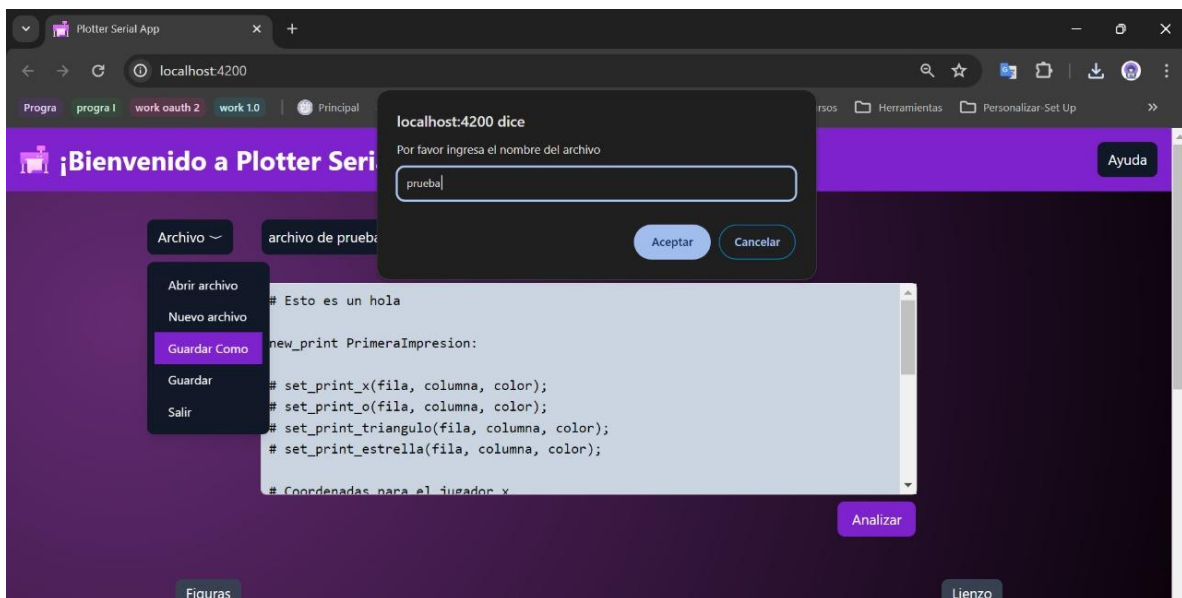
Se abre el archivo. orga de nuestro escritorio



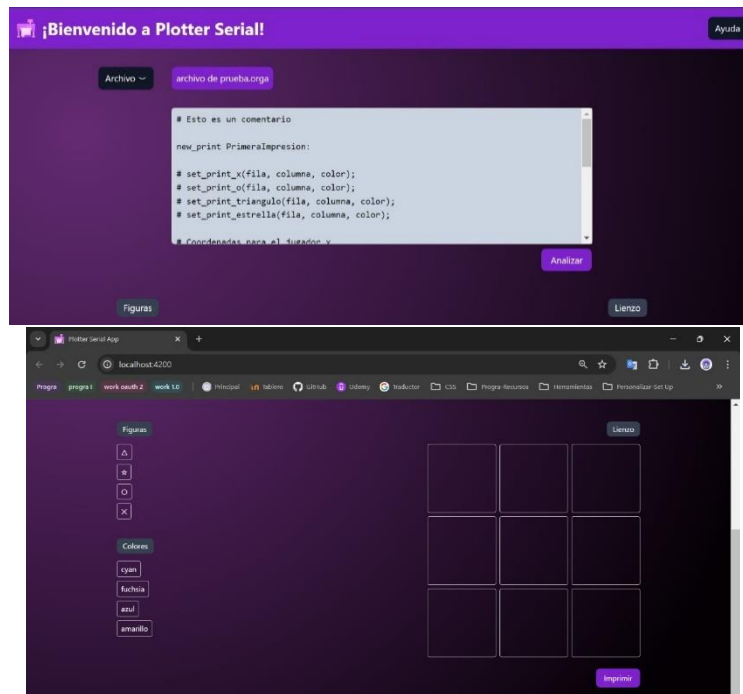
Se selecciona el nuevo archivo, y se inicializa el área de texto.



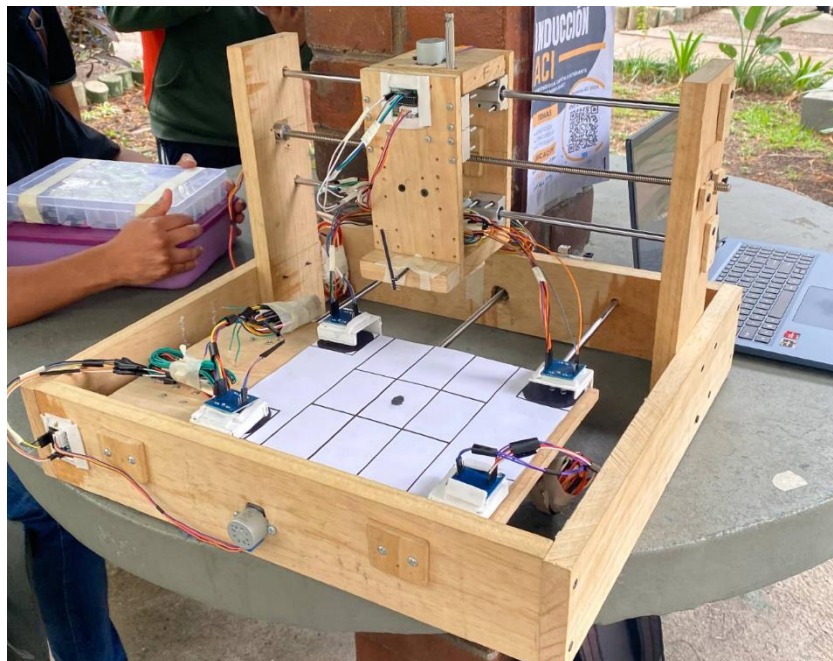
Se selecciona guardar como, se agrega una opción para cambiar nombre y descargar él .orga



Después de darle analizar se muestra el resultado de la impresión especificando que forma tiene y el color.



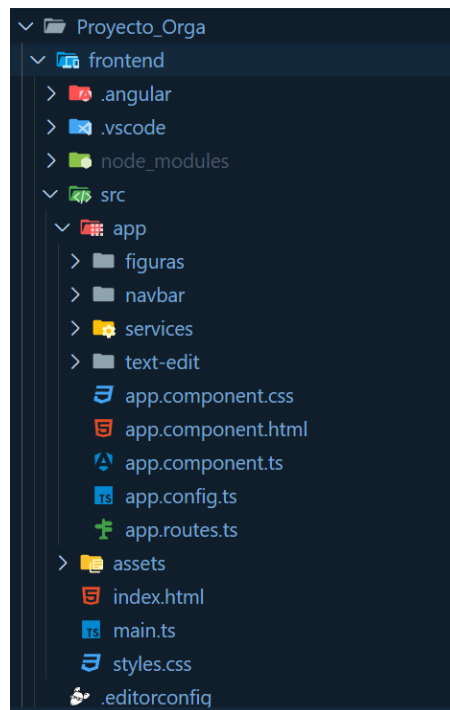
Se imprime en nuestro plotter



Manual Técnico

Para el desarrollo del software Plotter Serial se utilizaron tecnologías como; angular para el desarrollo del frontend y Django, para el desarrollo del backend. El flujo del software consiste en; el usuario puede interactuar con el frontend cargando un archivo de texto plano con extensión. orga, posterior a ello, se analiza enviando esta información al backend, luego se envía estos datos previamente analizados a la matriz para que muestre el valor en cada celda, luego en al presionar el botón “Imprimir” se envían estos datos a la impresora para poder imprimir valga la redundancia imprimir los datos de figuras y colores agregadas a la matriz. A continuación, se muestran porciones de código implementadas en el frontend (angular) y el backend (Django).

Frontend (angular)



Navbar

```
navbar.component.html X
Projecto_Orga > frontend > src > app > navbar > navbar.component.html > nav.bg-purple-700.flex.justify-between.text-white.p-4.items-center > div > a.flex.items-center
Go to component
1 <nav class="bg-purple-700 flex justify-between text-white p-4 items-center">
2   <div>
3     <a href="#" class="flex items-center space-x-3 rtl:space-x-reverse">
4       
5       <span class="self-center text-3xl font-bold whitespace-nowrap dark:text-white">¡Bienvenido a Plotter Serial!</span>
6     </a>
7   </div>
8   <div>
9     <button class="md:dark:bg-gray-900 px-3 py-2 rounded-lg ml-2" (click)="ayuda()">Ayuda</button>
10  </div>
11 </nav>

navbar.component.ts X
Projecto_Orga > frontend > src > app > navbar > navbar.component.ts > NavbarComponent > ayuda
1 import { Component } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 @Component({
5   selector: 'app-navbar',
6   standalone: true,
7   imports: [CommonModule],
8   templateUrl: './navbar.component.html',
9   styles: ``
10 })
11
12 export class NavbarComponent {
13
14   ayuda() {
15     window.open('assets/pdf/DOCUMENTACION.pdf', '_blank');
16   }
17 }
```

Text-edit

```
text-edit.component.html 1 X
Projecto_Orga > frontend > src > app > text-edit > text-edit.component.html > div.flex.justify-end.mx-10.w-3/4.my-12.mt-0
Go to component
1 <div class="relative ml-40 my-8">
2   <input type="file" #fileInput (change)="fileChange($event)" accept=".orga" style="display: none">
3   <button (mouseover)="toggleDropdown()" class="md:dark:bg-gray-900 px-3 py-2 rounded-lg">Archivo </button>
4   <div (mouseleave)="hideDropdown()" *ngIf="showDropdown" class="absolute md:dark:bg-gray-900 text-white mt-2 rounded-lg shadow-lg p-2"
5     <a class="block px-4 py-2 text-sm hover:bg-purple-700 hover.cursor-pointer" (click)="openFile(); $event.preventDefault()">Abrir
6     <a class="block px-4 py-2 text-sm hover:bg-purple-700 hover.cursor-pointer" (click)="newFile(); $event.preventDefault()">Nuevo
7     <a class="block px-4 py-2 text-sm hover:bg-purple-700 hover.cursor-pointer" (click)="saveasFile(); $event.preventDefault()">Guardar
8     <a class="block px-4 py-2 text-sm hover:bg-purple-700 hover.cursor-pointer" (click)="saveFile(); $event.preventDefault()">Guardar
9     <a class="block px-4 py-2 text-sm hover:bg-purple-700 hover.cursor-pointer" (click)="exit(); $event.preventDefault()">Salir</a>
10  </div>
11  <p *ngIf="fileName" class="md:dark:bg-gray-900 text-white hover:bg-purple-700 hover.cursor-pointer rounded-lg ml-8 inline-block p-2"
12  </p>
13 </div>
14 <div class="flex justify-center items-center h-auto w-3/4 mx-auto my-2">
15   <textarea id="fileContent" class="flex items-center h-60 w-3/4 border p-2 rounded-md resize-none overflow-auto bg-slate-300 text-b
16 </div>
17 <div class="flex justify-end mx-10 w-3/4 my-12 mt-0">
18   <button class="md:bg-purple-700 md:dark:bg-gray-900 hover:dark:bg-gray-900 hover.cursor-pointer px-4 py-2 rounded-lg">Analizar</button>
19 </div>
```



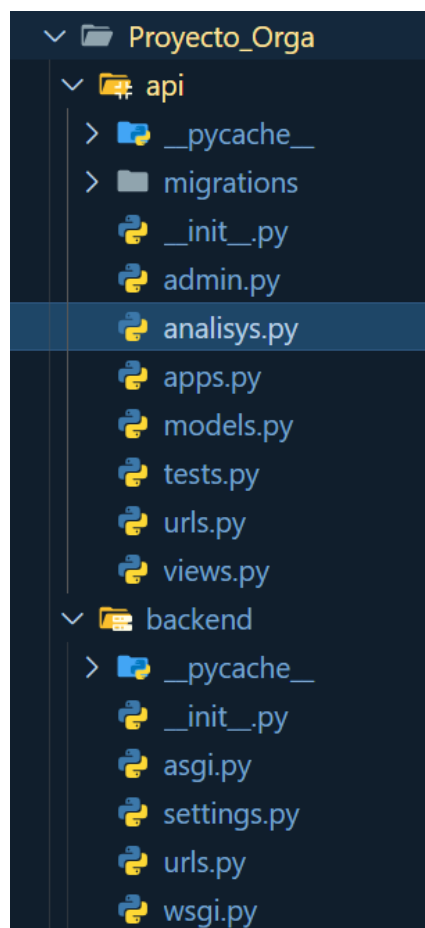
```
text-edit.component.ts X
Proyecto_Orga > frontend > src > app > text-edit > text-edit.component.ts > TextEditComponent
1 import { Component, ElementRef, ViewChild } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { FormsModule, ReactiveFormsModule } from '@angular/forms';
4 import { AnalizarService } from '../services/analizar.service';
5
6 @Component({
7   selector: 'app-text-edit',
8   standalone: true,
9   imports: [CommonModule, FormsModule, ReactiveFormsModule],
10  templateUrl: './text-edit.component.html',
11 })
12 export class TextEditComponent {
13   fileContent: string = '';
14   fileName: string = '';
15   showDropdown = false;
16   canSave: boolean = false;
17   @ViewChild('fileInput', { static: false }) fileInputRef!: ElementRef<HTMLInputElement>;
18   //constructor(private analizarService: AnalizarService) {}
19
20   toggleDropdown() {
21     this.showDropdown = !this.showDropdown;
22   }
23
24   hideDropdown() {
25     this.showDropdown = false;
26   }
27 }
```

Figuras

```
figuras.component.html X
Proyecto_Orga > frontend > src > app > figuras > figuras.component.html > div.flex.justify-between.px-8.mb-10
Go to component
1 <div class="flex justify-between px-8 mb-10">
2   <div class="flex flex-col items-start px-40">
3     <div class="flex flex-col items-baseline">
4       <label class="bg-gray-700 block rounded-lg mb-4 px-3 py-1">Figuras</label>
5       <span class="px-2 py-1 border rounded mb-2">△</span>
6       <span class="px-2 py-1 border rounded mb-2">☆</span>
7       <span class="px-2 py-1 border rounded mb-2">○</span>
8       <span class="px-2 py-1 border rounded mb-2">×</span>
9     </div>
10    <div class="flex flex-col items-baseline py-8">
11      <label class="bg-gray-700 block rounded-lg mb-4 px-3 py-1">Colores</label>
12      <span class="px-2 py-1 border rounded mb-2">cyan</span>
13      <span class="px-2 py-1 border rounded mb-2">fuchsia</span>
14      <span class="px-2 py-1 border rounded mb-2">azul</span>
15      <span class="px-2 py-1 border rounded mb-2">amarillo</span>
16    </div>
17  </div>
```

```
figuras.component.ts ×
Proyecto_Orga > frontend > src > app > figuras > figuras.component.ts > ...
1 import { Component } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 @Component({
5   selector: 'app-figuras',
6   standalone: true,
7   imports: [CommonModule],
8   templateUrl: './figuras.component.html',
9 })
10 export class FigurasComponent {
11   matriz = Array(3).fill(Array(3).fill(''));
12 }
```

Backend (django)



analysys.py 1 x

Proyecto_Orga > api > analysys.py > ...

```
24 def get_data(name):
25
26     array = [['-' for _ in range(3)] for _ in range(3)]
27
28     with open(name, 'r') as archivo:
29         for line in archivo:
30             if not line.strip().startswith('#'):
31                 instruction = line.replace(' ', '')
32                 instruction = instruction.replace('; ', '')
33                 instruction = instruction.replace('\n', '')
34                 instruction = instruction.replace('\\', '')
35                 instruction = instruction.split('(')
36                 if instruction[0].lower() == 'set_print_x':
37                     data = instruction[1].split(',')
38                     if len(data) == 3:
39                         row = int(data[0])
40                         column = int(data[1])
41                         color = data[2].strip().lower()
42
43
44                     if 1<= row <= 3 and 1<= column <= 3 and color in ['cyan', 'negro', 'magenta', 'amarillo']:
45                         if array[row-1][column-1] == '-':
46
47                             set_print_x(row, column, color)
48                             array[row-1][column-1] = ['x',color,1]
49                         else:
50                             print('posicion ya ocupada',data[0], data[1])
51                             break
```

Código Comentado (Arduino)

Código Arduino

Declaración de Variables para pines de motores y sensores.

```

1 #include <Stepper.h>
2
3 int S0_sensor1 = 59; // S0_sensor1 a pin 59
4 int S1_sensor1 = 60; // S1_sensor1 a pin 60
5 int S2_sensor1 = 61; // S2_sensor1 a pin 61
6 int S3_sensor1 = 62; // S3_sensor1 a pin 62
7 int OUT_sensor1 = 63; // OUT_sensor1 a pin 63
8
9 int S0_sensor2 = 2; // S0_sensor2 a pin 2
10 int S1_sensor2 = 3; // S1_sensor2 a pin 3
11 int S2_sensor2 = 4; // S2_sensor2 a pin 4
12 int S3_sensor2 = 5; // S3_sensor2 a pin 5
13 int OUT_sensor2 = 6; // OUT_sensor2 a pin 6
14
15 int S0_sensor3 = 54; // S0_sensor3 a pin 54
16 int S1_sensor3 = 55; // S1_sensor3 a pin 55
17 int S2_sensor3 = 56; // S2_sensor3 a pin 56
18 int S3_sensor3 = 57; // S3_sensor3 a pin 57
19 int OUT_sensor3 = 58; // OUT_sensor3 a pin 58
20
21 int S0_sensor4 = 45; // S0_sensor4 a pin 45
22 int S1_sensor4 = 43; // S1_sensor4 a pin 43
23 int S2_sensor4 = 41; // S2_sensor4 a pin 41
24 int S3_sensor4 = 39; // S3_sensor4 a pin 39
25 int OUT_sensor4 = 37; // OUT_sensor4 a pin 37
26
27 int Bit1_PrimerNumero = 25;
28 int Bit2_PrimerNumero = 23;
29 int Bit1_SegundoNumero = 22;
30 int Bit2_SegundoNumero = 27;

```

Declaración de los pasos por motor e instrucciones y pines.

```

30 int giro_segundomotor = 27;
31
32 int pasosporvuelta = 1024;
33 Stepper motorBase (pasosporvuelta, 8,10,9,11);
34 Stepper motorSuperior (pasosporvuelta, 47,51,49,53);
35 Stepper motorSubeyBaja (pasosporvuelta, 29,33,31,35);
36 Stepper motorSellos (pasosporvuelta, 64,66,65,67);
37 String Instrucciones[] = {"0","0","0","triangulo","magenta","1","0","cyan","1","0","0","0","0","0","negro","1","0","0","0","0","0","x","amarillo","1","x","cyan","1"};
38 // triangulo = 0 pasos
39 // o = 512 pasos
40 // cuadrado = 1024 pasos
41 // x = 1536 pasos
42 boolean Imprimir = true;
43 int UltimaRotacion = 0;
44 int Posicion = 0;
45

```

```
void setup() {

    motorBase.setSpeed(25);
    motorSuperior.setSpeed(25);
    motorSubeyBaja.setSpeed(25);
    motorSellos.setSpeed(25);

    // configuracion RAM

    pinMode(Bit1_PrimerNumero, OUTPUT);
    pinMode(Bit2_PrimerNumero, OUTPUT);
    pinMode(Bit1_SegundoNumero, OUTPUT);
    pinMode(Bit2_SegundoNumero, OUTPUT);

    // configuracion sensor de color 1
    pinMode(S0_sensor1, OUTPUT); // pin 59 como salida
    pinMode(S1_sensor1, OUTPUT); // pin 60 como salida
    pinMode(S2_sensor1, OUTPUT); // pin 61 como salida
    pinMode(S3_sensor1, OUTPUT); // pin 62 como salida
    pinMode(OUT_sensor1, INPUT); // pin 63 como salida
    digitalWrite(S0_sensor1,HIGH); // establece frecuencia de salida
    digitalWrite(S1_sensor1,LOW); // del modulo al 20 por ciento
```

Funciones de instrucciones para el mapeo del Plotter.

```
void loop() {
    // ciclo donde realizara la impresion en cada posicion
    Posicion = 0;
    for(int i = 0; i < 27; i++){
        EnviarRAM(Posicion);
        Posicion++;
    }
    Posicion = 0;
    for(int i = 0; i < 27; i++){
        if(instrucciones[i+2] == "1"){
            UltimaRotacion = SeleccionarForma(UltimaRotacion, Instrucciones[i]);
            RotosarSello(instrucciones[i+1]);
            MoverBase(Posicion);
            MoverSuperior(Posicion);
            ColocarSello();
            RegresarBaseCentro(Posicion);
            Posicion++;
        }
    }
}

void ColocarSello(){
    for(int i = 1; i <= 10; i++){
        if(i <= 5){
            motorSubeyBaja.step(pasosporvuelta);
        }
        else if(i == 6){
            delay(3000);
            motorSubeyBaja.step(-pasosporvuelta);
        }
        else{
            motorSubeyBaja.step(-pasosporvuelta);
        }
    }
}

void MoverBase(int Act){
    if(Act == 0 || Act == 3 || Act == 6){
        //se mueve a la posicion 1,1 || 2,1 || 3,1
        DarMedioPaso(19, -1, 1);
        MoverSuperior(Act);
    }
    else if(Act == 1 || Act == 4 || Act == 7){
        //no se mueve, las posiciones 1,2 || 2,2 || 3,2 son las centrales
        MoverSuperior(Act);
    }
    else if(Act == 2 || Act == 5 || Act == 8){
        //se mueve a la posicion 1,3 || 2,3 || 3,3
        DarMedioPaso(19, 1, 1);
        MoverSuperior(Act);
    }
}

void MoverSuperior(int Act){
    if(Act <= 2){
        //se mueve a la posicion 1,1 || 2,1 || 3,1
        DarMedioPaso(11, -1, 2);
    }
    else if(Act >= 3 || Act <= 5){
        //no se mueve, las posiciones 1,2 || 2,2 || 3,2 son las centrales
    }
    else if(Act >= 6){
        //se mueve a la posicion 1,3 || 2,3 || 3,3
        DarMedioPaso(11, 1, 2);
    }
}
```

Funciones del movimiento en nuestra Hoja (Dirección y color)

```
165 void SeleccionarForma(String color){
166     if(color == "rojo"){
167         DarMedioPaso(42, -1, 2);
168         ColocarSello();
169         // movimiento hacia el color rojo
170     }
171     else if(color == "verde"){
172         DarMedioPaso(42, -1, 2);
173         ColocarSello();
174         // movimiento hacia el color verde
175     }
176     else if(color == "amarillo"){
177         DarMedioPaso(55, -1, 2);
178         ColocarSello();
179         // movimiento hacia el color amarillo
180     }
181     else if(color == "lila"){
182         DarMedioPaso(55, -1, 2);
183         ColocarSello();
184         // movimiento hacia el color lila
185     }
186 }
187
188 int SeleccionarForma(int UltimaRotacion, String sello){
189     if(UltimaRotacion == 0){
190         if(sello == "triangulo"){
191             // se mueve 1824 pasos || 180 grados
192             motorSello.step(pasosporvuelta);
193             return 1824;
194         }
195         else if(sello == "o"){
196             // se mueve 512 pasos sentido antihorario || -90 grados
197             motorSello.step(-(pasosporvuelta/2));
198             return 1536;
199         }
200         else if(sello == "x"){
201             // se mueve 512 pasos sentido horario || 90 grados
202             motorSello.step(pasosporvuelta/2);
203             return 512;
204         }
205         else if(sello == "cuadrado"){
206             // no se mueve || 0 grados
207             return 0;
208         }
209     }
210     else if(UltimaRotacion == 512){
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

Código del envío de datos de la memoria RAM.

```
void EnviarRAM(int Act){
  if(Act == 0){
    // enviar a la posicion 1,1
    digitalWrite(Bit1_PrimerNumero, HIGH);
    digitalWrite(Bit1_SegundoNumero, HIGH);
    delay(100);
    digitalWrite(Bit1_PrimerNumero, LOW);
    digitalWrite(Bit1_SegundoNumero, LOW);
  }else if(Act == 1){
    // enviar a la posicion 1,2
    digitalWrite(Bit1_PrimerNumero, HIGH);
    digitalWrite(Bit2_SegundoNumero, HIGH);
    delay(100);
    digitalWrite(Bit1_PrimerNumero, LOW);
    digitalWrite(Bit2_SegundoNumero, LOW);
  }else if(Act == 2){
    // enviar a la posicion 1,3
    digitalWrite(Bit1_PrimerNumero, HIGH);
    digitalWrite(Bit1_SegundoNumero, HIGH);
    digitalWrite(Bit2_SegundoNumero, HIGH);
    delay(100);
    digitalWrite(Bit1_PrimerNumero, LOW);
    digitalWrite(Bit1_SegundoNumero, LOW);
    digitalWrite(Bit2_SegundoNumero, LOW);
  }else if(Act == 3){
    // enviar a la posicion 2,1
    digitalWrite(Bit2_PrimerNumero, HIGH);
    digitalWrite(Bit1_SegundoNumero, HIGH);
    delay(100);
    digitalWrite(Bit2_PrimerNumero, LOW);
    digitalWrite(Bit1_SegundoNumero, LOW);
  }else if(Act == 4){
    // enviar a la posicion 2,2
    digitalWrite(Bit2_PrimerNumero, HIGH);
    digitalWrite(Bit2_SegundoNumero, HIGH);
    delay(100);
    digitalWrite(Bit2_PrimerNumero, LOW);
    digitalWrite(Bit2_SegundoNumero, LOW);
  }else if(Act == 5){
    // enviar a la posicion 2,3
    digitalWrite(Bit2_PrimerNumero, HIGH);
    digitalWrite(Bit1_SegundoNumero, HIGH);
    digitalWrite(Bit2_SegundoNumero, HIGH);
  }
```
















Funciones para los sensores del color.

```
694 digitalWrite(S2_sensor1,HIGH); // establece fotodiodos
695 digitalWrite(S3_sensor1,HIGH); // con filtro verde
696 int verde1 = pulseIn(OUT_sensor1, LOW); // obtiene duracion @ pulso @ salida del sensor
697 delay(100); // demora @ 100 msag
698
699 digitalWrite(S2_sensor2,HIGH); // establece fotodiodos
700 digitalWrite(S3_sensor2,HIGH); // con filtro verde
701 int verde2 = pulseIn(OUT_sensor2, LOW); // obtiene duracion @ pulso @ salida del sensor
702 delay(100); // demora @ 100 msag
703
704 digitalWrite(S2_sensor1,HIGH); // establece fotodiodos
705 digitalWrite(S3_sensor1,HIGH); // con filtro verde
706 int verdes = pulseIn(OUT_sensor1, LOW); // obtiene duracion @ pulso @ salida del sensor
707 delay(100); // demora @ 100 msag
708
709 digitalWrite(S2_sensor4,HIGH); // establece fotodiodos
710 digitalWrite(S3_sensor4,HIGH); // con filtro verde
711 int verdes = pulseIn(OUT_sensor4, LOW); // obtiene duracion @ pulso @ salida del sensor
712 delay(100); // demora @ 100 msag
713
714 digitalWrite(S2_sensor1,LOW); // establece fotodiodos
715 digitalWrite(S3_sensor1,HIGH); // con filtro azul
716 int azul1 = pulseIn(OUT_sensor1, LOW); // obtiene duracion @ pulso @ salida del sensor
717 delay(100); // demora @ 100 msag
718
719 digitalWrite(S2_sensor2,LOW); // establece fotodiodos
720 digitalWrite(S3_sensor2,HIGH); // con filtro azul
721 int azul2 = pulseIn(OUT_sensor2, LOW); // obtiene duracion @ pulso @ salida del sensor
722 delay(100); // demora @ 100 msag
723
724 digitalWrite(S2_sensor3,LOW); // establece fotodiodos
725 digitalWrite(S3_sensor3,HIGH); // con filtro azul
726 int azul3 = pulseIn(OUT_sensor3, LOW); // obtiene duracion @ pulso @ salida del sensor
727 delay(100); // demora @ 100 msag
728
729 digitalWrite(S2_sensor4,LOW); // establece fotodiodos
730 digitalWrite(S3_sensor4,HIGH); // con filtro azul
731 int azul4 = pulseIn(OUT_sensor4, LOW); // obtiene duracion @ pulso @ salida del sensor
732 delay(100); // demora @ 100 msag
733
734 if((rojo1 > 100 && verde1 > 100 && azul1 > 100){
735   if((rojo2 > 100 && verde2 > 100 && azul2 > 100){
736     if((rojo3 > 100 && verde3 > 100 && azul3 > 100){
737       if((rojo4 > 100 && verde4 > 100 && azul4 > 100){
738         return true;
739       }
740     }
741   }
742 }
```

Presupuesto

Engranajes, varillas y tornillos	Q.1055.80
Madera	Q.260.00
Placa (RAM)	Q.120.00
Extras (Cable, Jumpers, Integrados)	Q.200.00
TOTAL	Q.1635.80

Facturas

Nombre del Producto	Modelo	Cantidad	Precio	Total	
Cojinete lineal de 8mm con base SCRUI	SCRUI	12	Q29.00	Q348.00	 
Acople flexible para eje de motor 8 x 5 mm	CNCAC-8X5FLEX	3	Q23.00	Q69.00	 
Eje acorado de 500mm x 8mm	CNC-EJE500x8	5	Q68.00	Q340.00	 
Tornillo M3	TORM3	12	Q0.50	Q6.00	 
Tornillo M4	TORM4	24	Q0.45	Q10.80	 
Varilla roscada con tuerca trapezoidal T8 de 500mm x 8mm	T8-VAR500	2	Q90.00	Q180.00	 
Varilla roscada con tuerca trapezoidal T8 de 200mm x 8mm	T8-VAR200	1	Q49.00	Q49.00	 
Cojinete 608ZZ para impresora 3D	K-608ZZ	3	Q8.00	Q24.00	 
Sub-Total				Q1,026.80	
Entrega en 2 a 3 horas				Q29.00	
Total				Q1,055.80	

#1. Bien
 Cantidad: 1
 P.Unit con IVA: 68.00
 Descuento: .00
 Total: 68.00
 CAB-MH-SERUS853
 Cable Manhattan 205153 Convierte Serial
 1 A USB
 Garantia: 1 Mes(es) S/N: 76662320515
 3

TOTALES:
 Descuento: .00
 Total: 68.00

*Sujeto a Pagos Trimestrales ISR
 *Agente de Retencion del IVA

Datos del Certificador:
 GUATEFACTURAS, SOCIEDAD ANONIMA.
 NIT: 5440773-4

Conclusiones

1. Innovación en la Educación:

- La organización de demostraciones de proyectos de innovación en instituciones educativas promueve el desarrollo de habilidades creativas y técnicas entre los estudiantes, preparándolos para enfrentar desafíos del mundo real en sus futuras carreras.

2. Integración de Tecnología y Diseño:

- La creación de una impresora no tradicional que utiliza hardware y software personalizados demuestra la convergencia entre la tecnología y el diseño. Este tipo de proyectos fomenta la exploración de nuevas formas de interactuar con la información y la expresión creativa.

3. Aprendizaje Interdisciplinario:

- El desarrollo de la impresora Cuadri-Color involucra conocimientos de ingeniería, programación y diseño de interfaces de usuario. Este enfoque interdisciplinario refleja la naturaleza colaborativa de la resolución de problemas en el mundo real, donde diversas habilidades y perspectivas se combinan para lograr resultados innovadores.

4. Uso de Tecnologías Emergentes:

- La utilización de tecnologías como la conexión serial y el control de motores para el Plotter refleja el interés por explorar y aprovechar nuevas herramientas y técnicas en el campo de la informática y la ingeniería. Esto puede inspirar a los estudiantes a mantenerse actualizados sobre las tendencias tecnológicas y a explorar áreas emergentes de estudio.

5. Enfoque Práctico en la Educación:

- La combinación de teoría y práctica en la elaboración de este proyecto proporciona a los estudiantes una experiencia práctica invaluable que complementa su aprendizaje en el aula. Este tipo de enfoque educativo puede mejorar la retención del conocimiento y preparar a los estudiantes para enfrentar desafíos del mundo real en sus futuras carreras.

Recomendaciones

1. Planificación Detallada:

- Antes de comenzar con el desarrollo del proyecto, es crucial realizar una planificación detallada que abarque todos los aspectos del hardware y software requeridos. Esto incluye la definición de los requisitos, la asignación de tareas, los plazos de entrega y la estimación de recursos necesarios.

2. División de Tareas:

- Dado que el proyecto implica múltiples componentes, como el desarrollo del hardware, el diseño de la interfaz de usuario y la implementación de la comunicación serial, es recomendable dividir las tareas entre los miembros del equipo de manera eficiente. Esto puede aumentar la productividad y garantizar que cada aspecto del proyecto reciba la atención necesaria.

3. Uso de Herramientas y Tecnologías Familiares:

- Aprovecha las herramientas y tecnologías con las que estés familiarizado para agilizar el proceso de desarrollo. Esto puede incluir el uso de entornos de desarrollo integrados (IDE) para la programación, bibliotecas de código abierto para funciones específicas y software de diseño asistido por ordenador (CAD) para el diseño de circuitos.

4. Pruebas Continuas:

- Implementa un enfoque de desarrollo iterativo que incluya pruebas continuas a lo largo del proceso. Esto te permitirá identificar y solucionar problemas a medida que surjan, en lugar de esperar hasta el final del proyecto para abordarlos. Las pruebas de integración entre el hardware y el software son especialmente importantes para garantizar la compatibilidad y el rendimiento adecuado del sistema en su conjunto.

5. Documentación Completa:

- Documenta meticulosamente cada etapa del proceso de desarrollo, incluidos los diseños de circuitos, el código fuente, los procedimientos de prueba y cualquier decisión de diseño importante.

Enlace del video

<https://drive.google.com/file/d/1EHFhHF1YVwRkbPlst3HCMol0u4SMXxRZ/view?usp=sharing>