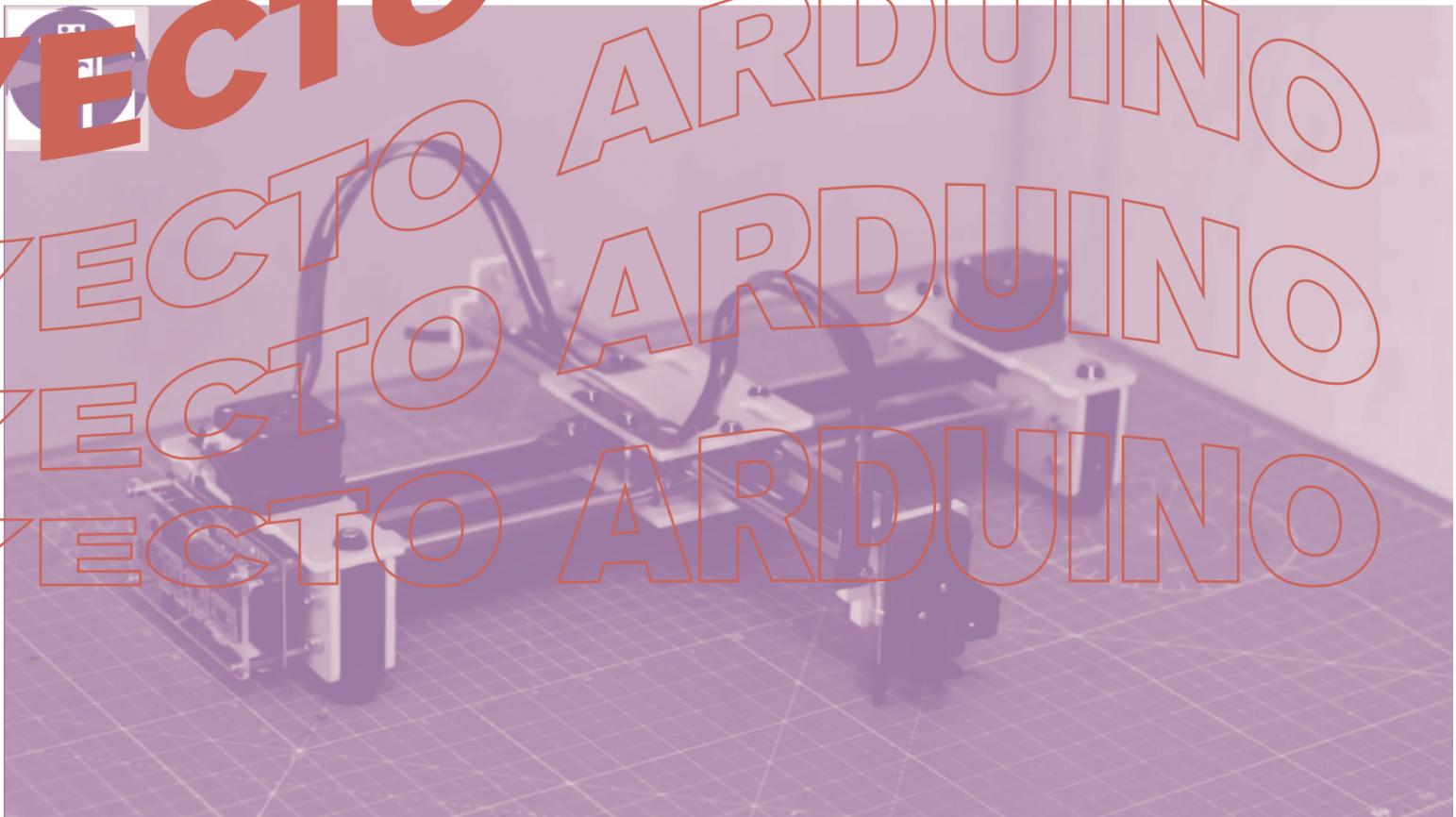


PROYECTO ARDUINO



INTRODUCCIÓN

INTRODUCCIÓN

INTRODUCCIÓN

El proyecto consiste en desarrollar un cartel generativo con parte del trabajo conjunto de clase, compuesto con el retrato de cada uno.

El trabajo se desarrollará mediante Procesing y arduino.

¿Qué es Processing?

Processing es un dialecto de Java que fue diseñado para el desarrollo del arte gráfico, para las animaciones y aplicaciones gráficas de todo tipo. Desarrollado por artistas y para artistas.

Es un software basado en Java y por lo tanto, multiplataforma. Desarrollado a partir del 2001 por el más que conocido departamento del MIT, el Media Lab. Los creadores, Casey Reas y Ben Fry, son discípulos del profesor John Maeda creador del método DBN (Design By Numbers) que, en los años 90, pretendía introducir a la programación a diseñadores,, artistas y no programadores de una forma sencilla. Precisamente Processing está inspirado en DBN que en la actualidad ya no está activo.

Máquina de dibujo Arduino

Es un robot capaz de dibujar sobre un papel. Para ello emplearemos el sistema de automatización CNC (control numérico computarizado). Este sistema permite almacenar un conjunto de instrucciones que el robot ejecutará secuencialmente para completar el dibujo.

Usando el software adecuado, podemos generar esta secuencia de instrucciones a partir de una imagen o pdf

FASE ANÁLITICA

FASE ANÁLITICA

FASE ANÁLITICA

FASE ANÁLITICA

OBJETIVOS

- Entender el sistema de coordenadas en Processing.
- Dibujar utilizando las formas básicas en Procesing.
- Desarrollar un proyecto utilizando las funciones principales.

CONTENIDO

- Trabajo colaborativo
- Documentación en el repositorio
- Principios de programación gráfica y generativa
- Desarrollo de una gráfica aplicada al cartel.

PROCESO DE TRABAJO

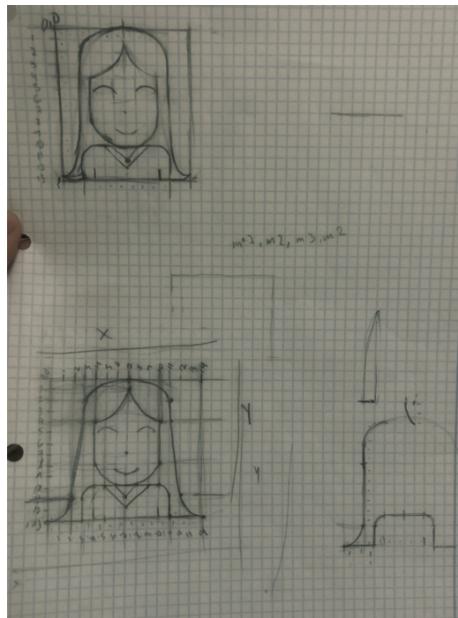
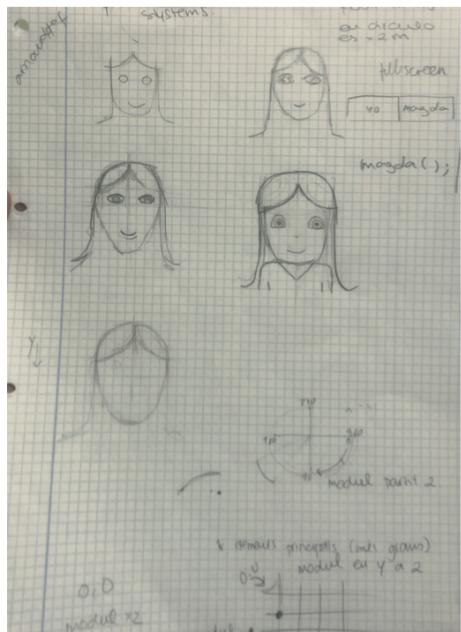
PROCESO DE TRABAJO

PROCESO DE TRABAJO

PROCESO DE TRABAJO

Desarrollaremos nuestro autorretrato con formas geométricas en papel cuadriculado para numerar las zonas y poder transportarlo a processing.

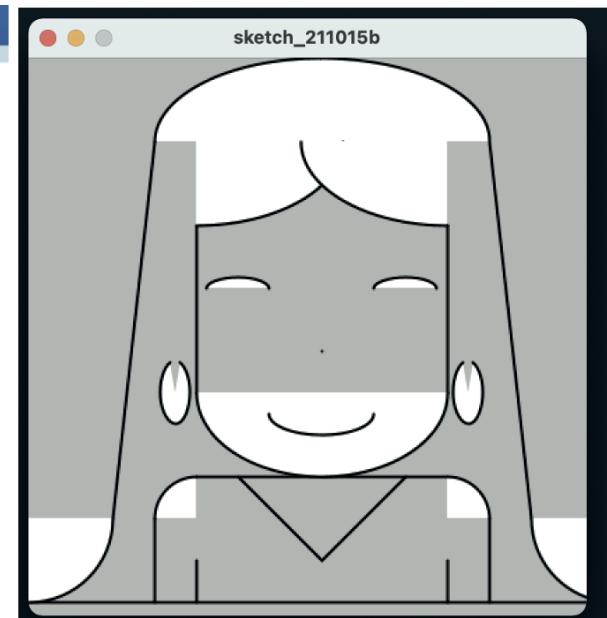
Una vez desarrollado lo pasaremos al lenguaje (coordenadas) correspondiente en Processing para realizar nuestro retrato.



```

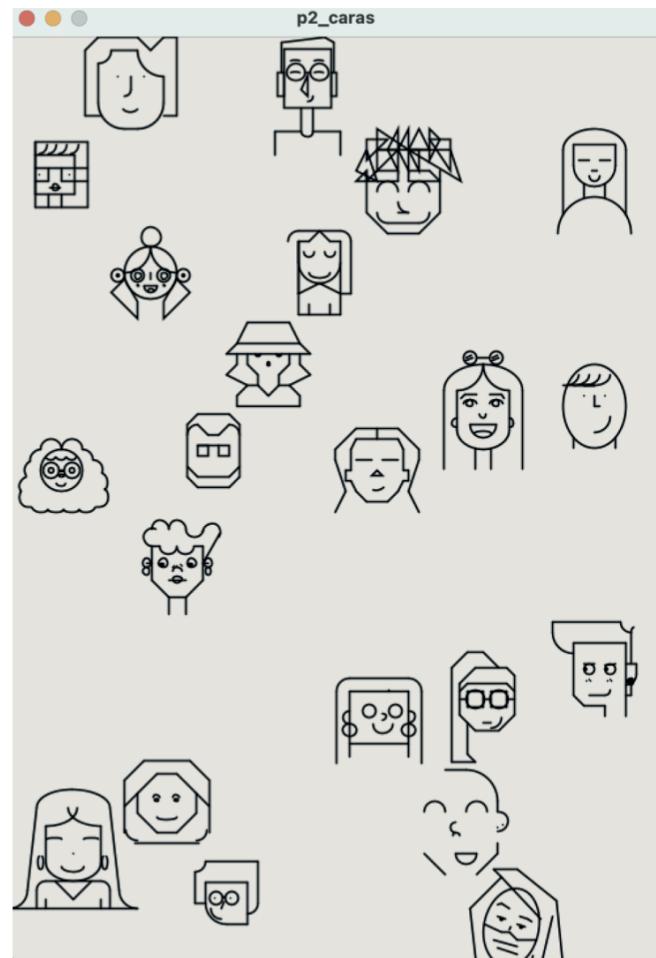
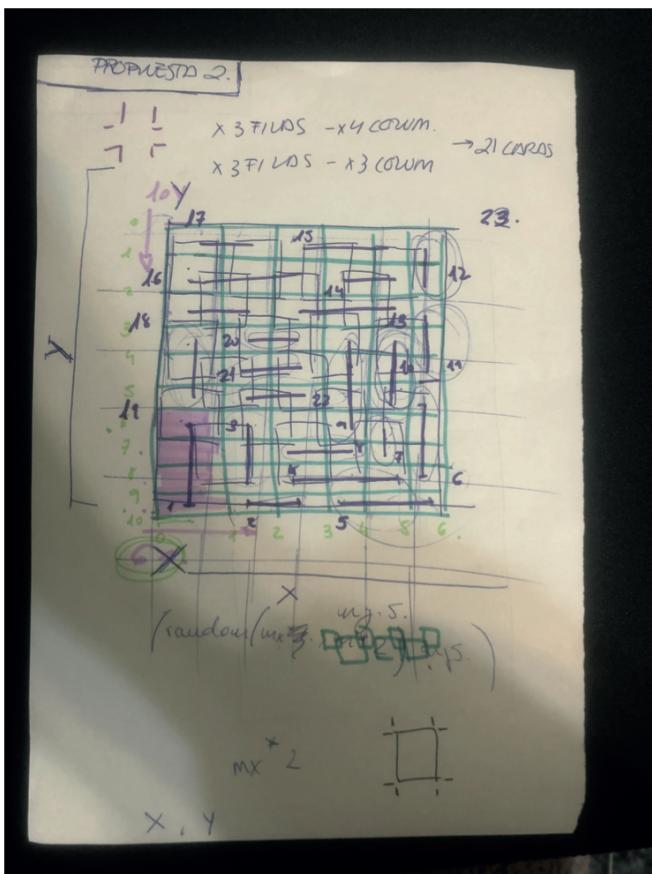
sketch_211015b
1 float m=30;
2
3 void setup () {
4   size (400, 400);
5   strokeWeight (2);
6 }
7
8 void draw() {
9   arc(m*7,m*2,m*8,m*4,radians(180),radians(270));
10  arc(m*7,m*2,m*8,m*4,radians(270),radians(360));
11  noLoop();
12  line(m*3,m*2,m*2,m*11);
13  line(m*11,m*2,m*12,m*11);
14  line(m*4,m*4,m*4,m*8);
15  line(m*10,m*4,m*10,m*8);
16  line(m*3,m*11,m*3,m*13);
17  line(m*11,m*11,m*11,m*13);
18  line(m*4,m*10,m*10,m*10);
19  line(m*0,m*13,m*14,m*13);
20  line(m*5,m*10,m*7,m*12);
21  line(m*9,m*10,m*7,m*12);
22  line(m*4,m*13,m*4,m*12);
23  line(m*10,m*13,m*10,m*12);
24  arc(m*4,m*2,m*7,m*4,radians(0),radians(90));
25  arc(m*10,m*2,m*7,m*4,radians(90),radians(180));
26  arc(m*0,m*11,m*4,m*4,radians(0),radians(90));
27  arc(m*14,m*11,m*4,m*4,radians(90),radians(180));
28  arc(m*4,m*11,m*2,m*2,radians(180),radians(270));
29  arc(m*10,m*11,m*2,m*2,radians(270),radians(360));
30  arc(m*7,m*8.5,m*2.5,m*1,radians(0),radians(180));
31  arc(m*5,m*5.5,m*1.5,m*0.5,radians(180),radians(360));
32  arc(m*9,m*5.5,m*1.5,m*0.5,radians(180),radians(360));
33  point(m*7,m*7);
34
35  arc(m*3.5,m*8,m*0.7,m*1.5,radians(-70),radians(250));
36  arc(m*10.5,m*8,m*0.7,m*1.5,radians(-70),radians(250));
37
38

```



Pondremos en común nuestro retrato y generamos una cuadrícula en la que se encuentren todas las caras.

La cuadrícula se creará en papel y como hemos realizado anteriormente lo transformaremos en coordenadas añadiendo las caras de los compañeros.



The screenshot shows the Processing IDE interface. The top menu bar includes 'File', 'Edit', 'Sketch', 'Debug', 'Tools', and 'Help'. A toolbar with icons for selection, zoom, and orientation is visible. The main area has tabs for 'p2_caras' (selected), 'MariaSaavedra', 'Paula', 'jorge', 'laura', and 'lidia'. The code editor contains Java pseudocode for a Processing sketch. The preview window on the right shows a black background with white text and shapes representing the sketch's output.

```
p2_caras    MariaSaavedra    Paula    jorge    laura    lidia

1 import processing.pdf.*;
2 PGraphicsPDF pdf;
3 float mx=40;
4
5 void setup(){
6     size(500,707);
7     frameRate(2);
8     pdf = (PGraphicsPDF)beginRecord(PDF, "retrats.pdf");
9     strokeWeight(1.5);
10 }
11 void draw(){
12     background(240);
13     float mx=width/6;
14     float my=height/10;
15     noFill();
16     ...
17 //1
18     m7;
19     marina(0, random (my*6, my*9));
20
21     ...
22     m8;
23     magda(random(mx,mx*2),my*9);
24     ...
25     m5;
26     sara(mx, random (my*6, my*8));
27     ...
28     m8;
29     lide(random(mx*2,mx*4),my*6);
30     ...
31     m9;
32     sylvia(random(mx*3,mx*5),my*9);
33     ...
34     m8;
35     cris(mx*5, random (my*5,my*8));
36     ...
37     m9;
38     ...
39     irene_estelles(mx*4, random(my*6, my*7));
40
41 lidie
```

Una vez exportado en pdf, utilizaremos Illustrator para dividir en tres documentos las caras, las cuales dividiremos en grupos de tres, por colores para realizar 3 impresiones distintas, que se realizarán en un mismo papel, de forma que queden superpuestas.

Finalmente, se imprimirán con la máquina de arduino, que leerá las coordenadas y generará las caras. Durante la impresión se cambiará 3 veces de rotulador (tinta) , ajustando la presión manualmente y sin mover la hoja.

