//https://valentin.dasdeck.com/processing/

float w = width/2;

float h = height/2;

float n = 12;

int r\_elipse = 400;

float raio = width\*r\_elipse/1000\*0.9;

float ang = 2\*PI / 60;

float ang\_hora = 2\*PI / 12;

// numero de minutos ou segundos para dividir o círculo

float angulo = 360/60;

// numero de horas para dividir o círculo

float angulo\_hora = 360/12;

int raioHour =70;

void setup() {

size(500, 500);

stroke(0);

smooth();

w = width/2;

h = height/2;

size(500, 500);

n = 12;

r\_elipse = 400;

raio = width\*r\_elipse/1000\*0.9;

ang = 2\*PI / 60;

ang\_hora = 2\*PI / 12;

// numero de minutos ou segundos para dividir o círculo

float angulo = 360/60;

// numero de horas para dividir o círculo

float angulo\_hora = 360/12;

int raioHour =70;

}

void draw() {

// limpar e não ficar rastro

background(255);

//deslocar o centro

// translate(w, h);

// os 2 primeiros sao o centro e o 2 ultimos sao as medidas cria elipse (circulo)

fill(100,100,200,200);

ellipse(w,h,r\_elipse,r\_elipse);

noStroke();

int hora = hour();

hora=hora%12;

int m = minute();

int s = second();

float minute\_angle, hour\_angle, second\_angle;

//os dois primeiros são os pontos referentes a base, perto do centro do relogio, os outros dois sao referente as coordenadas do ponto perto da borda.

//minuto

fill(125,0,0,50);

triangle(w-8, h-8,w+8,h+8, w + raio\* cos(radians(m\*angulo- 15\*angulo)), h +raio\* sin(radians(m\*angulo - 15\* angulo)));

fill(125,0,0,200);

minute\_angle = (m\*TWO\_PI/60)-(HALF\_PI);

triangle(w-1, h-1,w+1,h+1, w + raio\* cos(minute\_angle), h +raio\* sin(minute\_angle));

//segundo

fill(150,0,200,70);

triangle(w-5, h-5,w+5,h+5, w + raio\* cos(radians(s\*angulo- 15\*angulo)), h +raio\* sin(radians(s\*angulo - 15\* angulo)));

fill(150,0,200,220);

second\_angle = (s\*TWO\_PI/60)-(HALF\_PI);

triangle(w-1, h-1,w+1,h+1, w + raio\* cos(second\_angle), h +raio\* sin(second\_angle));

//hora

fill(150,150,200,70);

triangle(w-25, h-25,w+25,h+25, w + raio\* cos(radians(hora\*angulo\_hora- 15\*angulo\_hora)), h +raio\* sin(radians(hora\*angulo\_hora - 15\* angulo\_hora)));

fill(0,250,200,255);

//numero de vezes que vai dividir

hour\_angle = (hora\*TWO\_PI/12)-(HALF\_PI);

//triangle(w-1, h-1,w+1,h+1, w + raio\* cos(hour\_angle), h +raio\* sin(hour\_angle));

line(w,h, w + 200\* cos(hour\_angle)-30, h +200\* sin(hour\_angle)-30);

// desenha as marcações de segundo

for (float f = 0; f < 2\*PI; f += ang) {

float Vx = h + cos(f) \* (raio+10);

float Vy = w + sin(f) \* (raio+10);

float Vxf = h + cos(f) \* (raio+20);

float Vyf = w + sin(f) \* (raio+20);

stroke(255);

strokeWeight(5);

line(Vx, Vy, Vxf, Vyf);

}

// desenha as marcações de hora

for (float f = 0; f < 2\*PI; f += ang\_hora) {

float Vx = h + cos(f) \* (raio+1);

float Vy = w + sin(f) \* (raio+1);

float Vxf = h + cos(f) \* (raio+15);

float Vyf = w + sin(f) \* (raio+15);

stroke(0);

strokeWeight(7);

line(Vx, Vy, Vxf, Vyf);

}

//remover borda

noStroke();

}