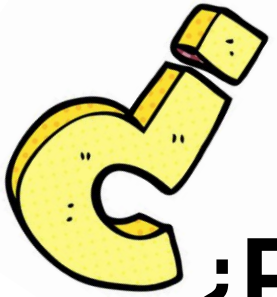
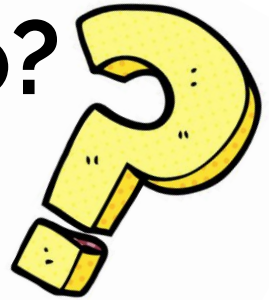
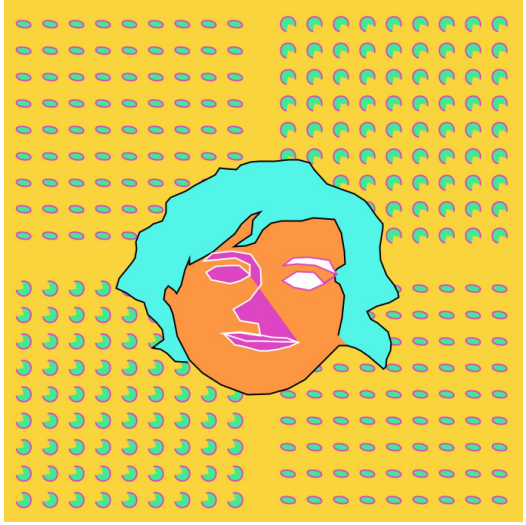
PPEM 2022

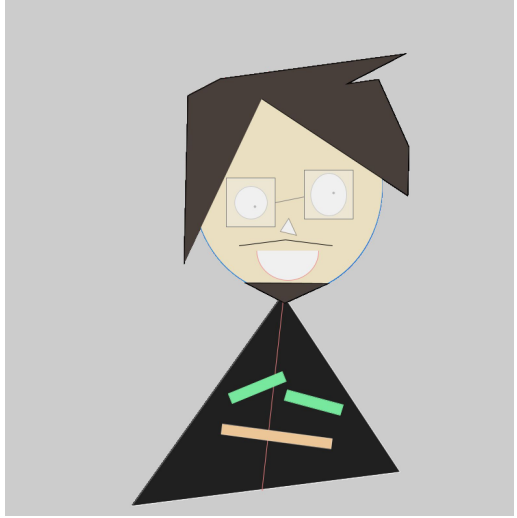
Texto

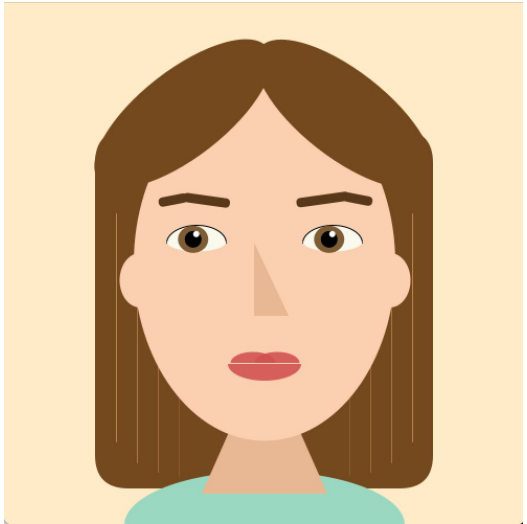


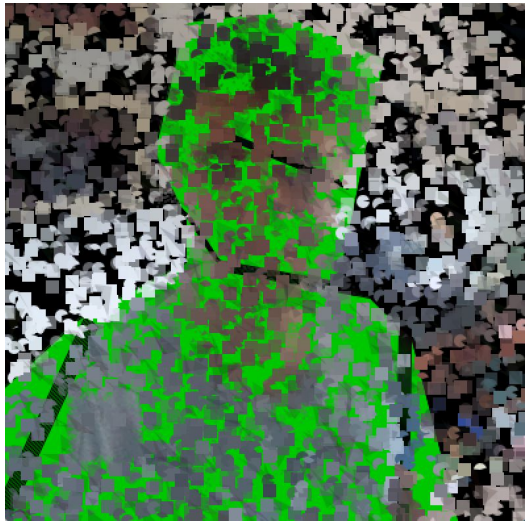
¿Porque nadie me dijo que la semana que viene es turismo?

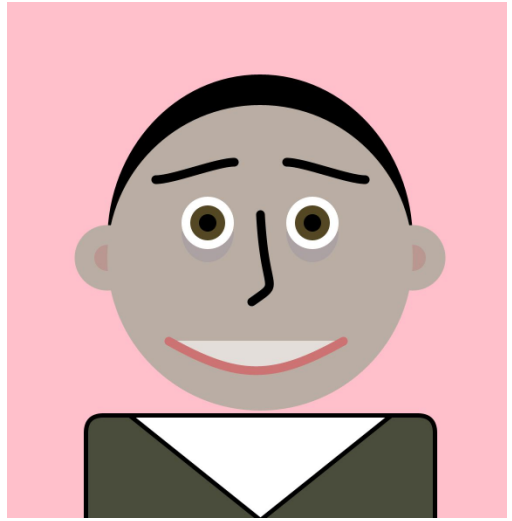












Links interesantes

https://vimeo.com/23316783?embedded=true&source=video_title&owner=2388809

https://www.youtube.com/watch?v=ON_zXFrix5ghttps://www.youtube.com/watch?v=ON_zXFrix5g

https://colab.research.google.com/github/justinjohn0306/VQGAN-CLIP/blob/main/VQGAN%2BCLIP_%28z%2Bquantize_method_with_augmentations%2C_user_friendly_interface%29.ipynb#scrollTo=af151e0b-8b57-4f8c-92dd-947f122db8c2

https://www.shadertoy.com/view/WsSBzh

https://www.youtube.com/watch?v=VIr1tvQb-wM https://www.youtube.com/shorts/IJjdJBtGbEI
https://www.kirellbenzi.com/art
https://timrodenbroeker.de/rasterize3d/

https://www.youtube.com/watch?v=iV-hah6xs2A
https://www.youtube.com/channel/UCJVYSZHAO2cxKlMivfV0xkQ/featured

Repaso

Micrófono

Cargar archivo de audio

Generar sonido

Analizar el sonido

Filtros

Efectos de audio

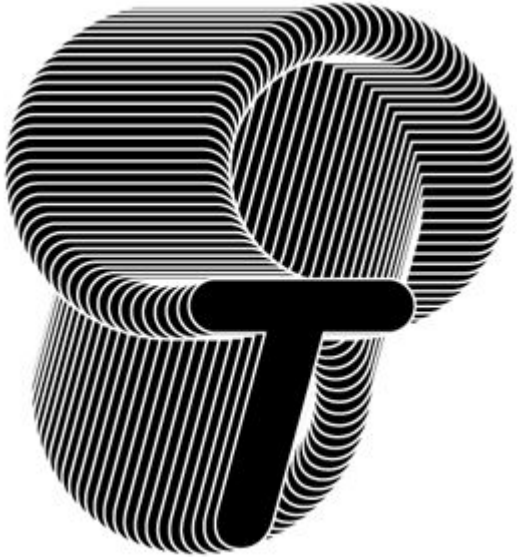
Texto

Mostrar el texto

Usar, cargar y exportar fuentes

Modificadores

Ejemplos



Mostrar el texto

Mostrar el texto

text(c, x, y) // mostrar un char

text(str, x, y) // mostrar un String

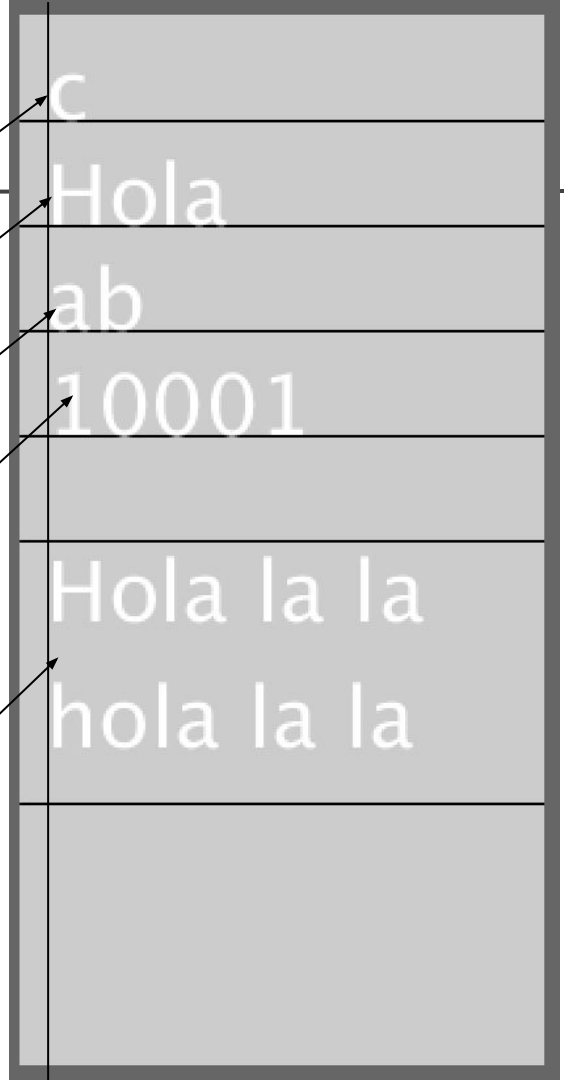
text(char[], start, stop, x, y) // mostrar parte de array de chars

text(num, x, y) // mostrar un número

text(str, x1, y1, x2, y2) // “encuadrar” un string

Ejemplo

```
size(200,400);
textSize(32);
char[] abcd = {'a','b','c','d'};
int i = 40;
int shift = 40;
line(0,i,width,i);
text('c', 10, i); // un char solo
i+=shift;
line(0,i,width,i);
text("Hola", 10, i); //un string
i+=shift;
line(0,i,width,i);
text(abcd, 0, 2, 10, i); // array de chars
i+=shift;
line(0,i,width,i);
text(10001, 10, i); // un numero
i+=shift;
line(0,i,width,i);
line(0,i+100,width,i+100);
text("Hola la la hola la la", 10, i, 200, i+100); // un string
```



rectMode para posicionar el texto

```
size(400,400);

textSize(32);

String la ="Hola la la hola la la la la hola la";

rectMode(RADIUS);

rect(160,160,100,100);

fill(0);

text(la, 160, 160, 100, 100);
```

Color del texto

rectMode - CORNER, CORNERS, RADIUS, CENTER

Usar, cargar y exportar fuentes

PFont

Qué fuentes puedo usar?

```
println( PFont.list());
```

Definir una fuente y mostrarla

```
createFont(name, size)
createFont(name, size, smooth)
createFont(name, size, smooth, charset)
```

```
PFont f;
f = createFont("Serif",64);
textFont(f);
size(300,300);
text("hola",100,height/2);
```

Para definir

Para usar

Para mostrar

Letras como vectores

Cargar una fuente de la computadora

```
PFont font;
font = loadFont("Dialog-48.vlw"); // de "data" folder
textFont(font);
size(300,300);
text("hola!", 10, 50);
```

Letras como texturas/imágenes

Exportar fuente

Para asegurar que al compartir el proyecto con otros estará disponible la fuente deseada:

Tools -> Create Font...

Setear el tamaño

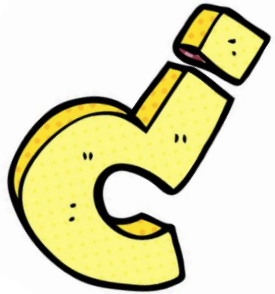
createFont(name, **size**)

createFont(name, **size**, smooth)

createFont(name, **size**, smooth, charset)

textFont(PFont, **size**)

textSize(**size**)



¿Dudas?

Modificadores

Modificadores

Ya vimos: textFont(), textSize(), rectMode(), fill()

Vamos a ver: textAlign(), textMode(), textLeading(), textWidth(), textAscent(), textDescent()

Alinear el texto

textAlign(alignX)

textAlign(alignX, alignY)

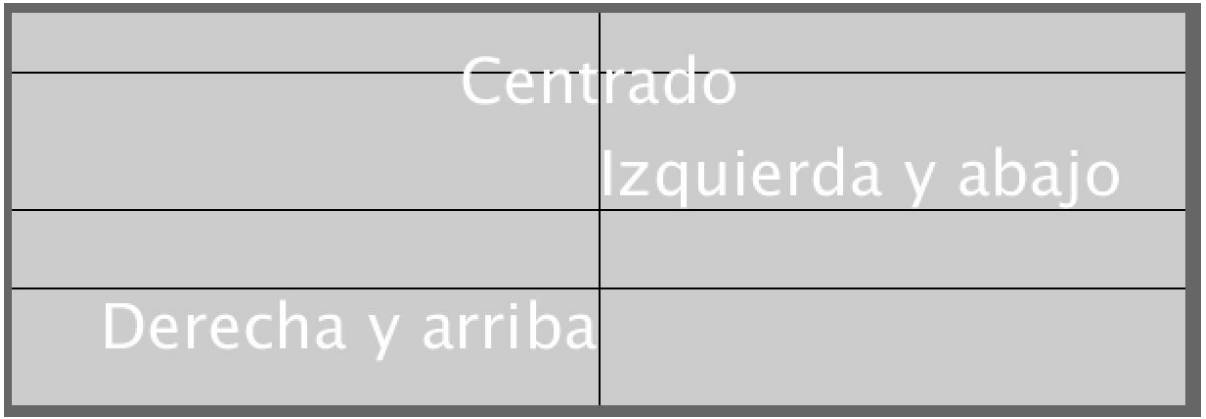
alignX: LEFT, CENTER, RIGHT

alignY: TOP, CENTER, BOTTOM, BASELINE

Por defecto: LEFT y BASELINE (con rect -> LEFT y TOP)

textAlign

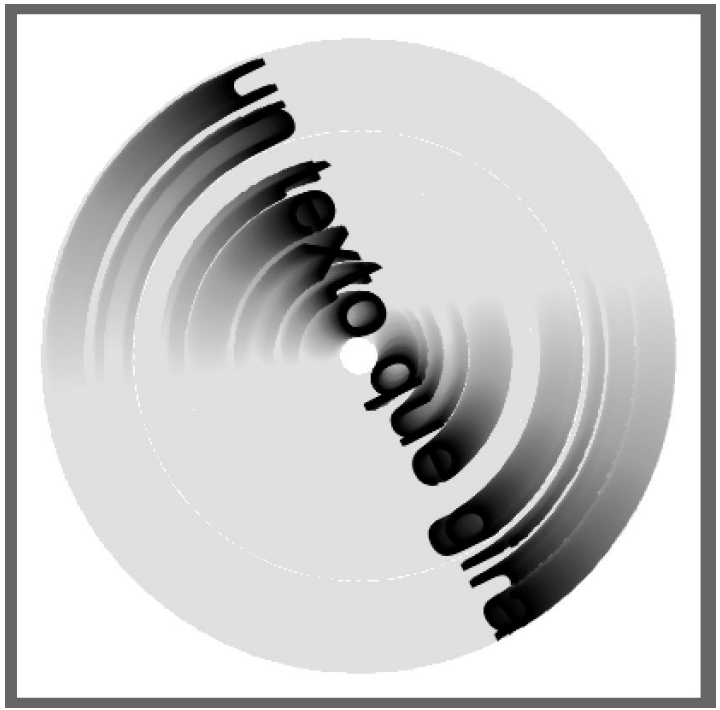
```
size(600,200);
textSize(32);
line(width/2,0,width/2,height);
line(0,30,width,30);
textAlign(CENTER, CENTER);
text("Centrado",width/2,30);

line(0,100,width,100);
textAlign(LEFT, BOTTOM);
text("Izquierda y abajo",width/2,100);

line(0,140,width,140);
textAlign(RIGHT, TOP);
text("Derecha y arriba",width/2,140);
```



¿Cómo podríamos hacer esto?

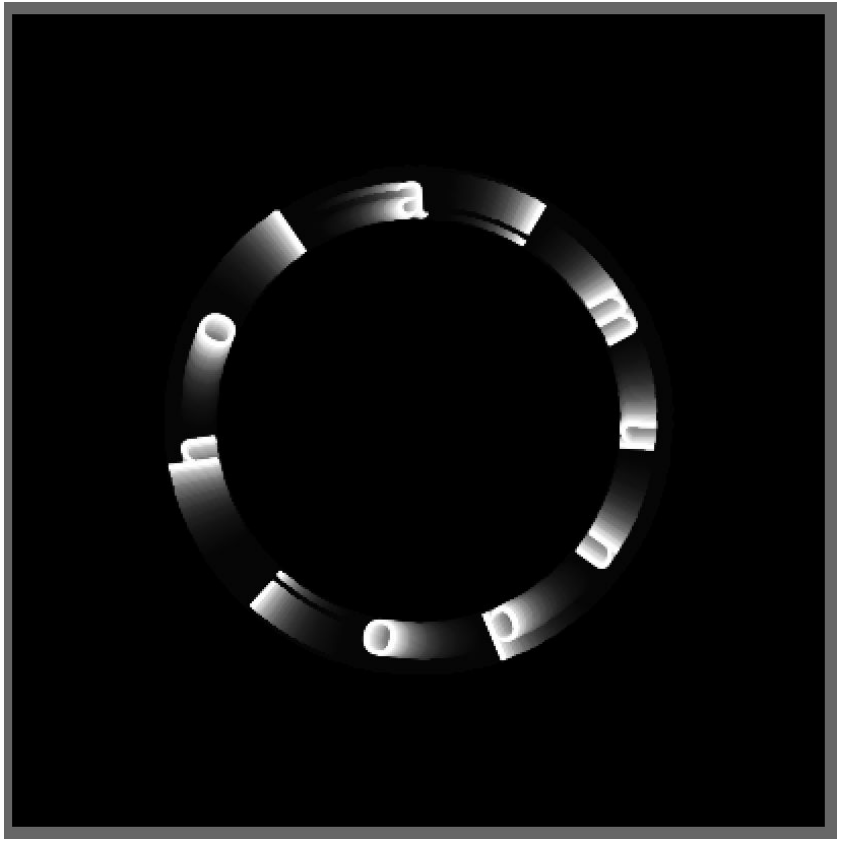
Un texto que gira

```
PFont f;
String message = "un texto que gira";
float theta;
void setup() {
  size(400, 400);
  f = createFont("Arial",50,true);
  textAlign(CENTER); // importante!
  textFont(f);
  background(255);
  noStroke();
}
void draw() {
  fill(255,4);
  rect(0,0,width,height);
  translate(width/2,height/2);  // me muevo al centro
  rotate(theta); // roto
  fill(0);
  text(message,0,0);  // escribo en 0,0 (el centro del texto estará ahí)
  theta += 0.01;
}
```

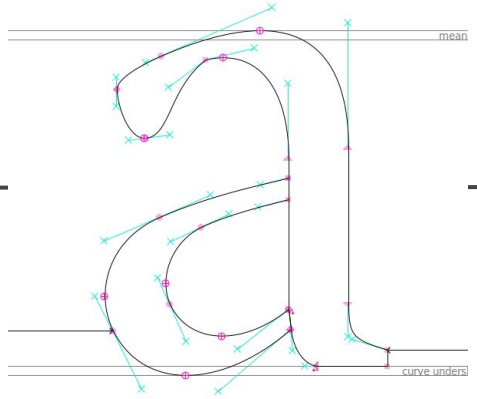


Un texto que gira distinto

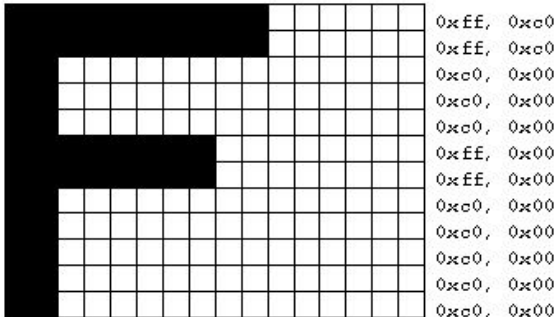
```
String message = "hola!mundo!";
float rotationDegrees;
float theta;
void setup(){
  size(400,400);
  textSize(32);
  rotationDegrees = 360/message.length();
  theta = 0;
  background(0);
  textAlign(CENTER);
}
void draw(){
  fill(0,20);
  rect(0,0,width,height);
  fill(255); // para escribir en blanco
  translate(width/2,height/2); // nos movemos al centro
  rotate(theta); // para que haya un giro constante
  for(int i =0;i<message.length();i++){
    pushMatrix();
      rotate(radians(rotationDegrees*i)); // para separar cada letra
      translate(100,0); // nos alejamos del centro
      rotate(radians(90)); // para rotar la letra para que “se apoye” sobre el círculo
      text(message.charAt(i),0,0); // dibujo la letra en 0,0
    popMatrix();
  }
  theta+=0.01;
}
```



textMode



MODEL (por defecto) o SHAPE



MODEL usa texturas para renderizar las fuentes

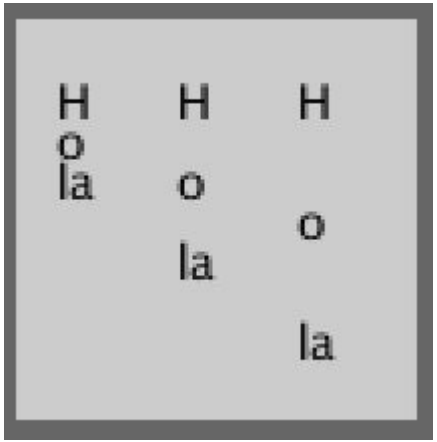
SHAPE usa contornos del dibujo vectorial (solo para PDF o P3D renderizador, para grabar texto con más precisión)

textLeading

Setea espacio entre líneas (en pixeles).

Ojo! textSize() cambia no solo tamaño de la fuente pero también el espacio entre líneas.

```
String lines = "H\no\nla";
textSize(12);
fill(0);  // las letras van a ser negras
textLeading(10);  // espacio entre lineas 10
text(lines, 10, 25);
textLeading(20);  // espacio entre lineas 20
text(lines, 40, 25);
textLeading(30);  // espacio entre lineas 30
text(lines, 70, 25);
```



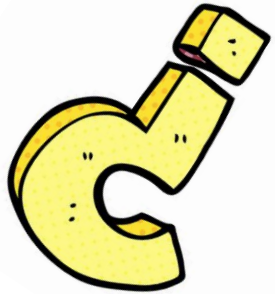
textWidth

Calcula el ancho en pixeles de un texto considerando la fuente y el tamaño seteado.

```
String s = "Hola ";
void setup(){
  size(600,300);
  textSize(32);
}
void draw(){
  background(0);
  float sw = textWidth(s);
  line(sw, 0, sw, height);
  text(s,0,65);
  rect(0,90,sw,60);
}
void keyPressed() {
  s+=key;
}
```

textAscent y textDescent


```
size(200,200);
float base = height/2; // dónde escribimos
float scalar = 0.8; // diferente para cada fuente
textSize(120);

line(0, base, width, base);
line(0, base+textDescent()* scalar, width, base+textDescent()* scalar);
line(0, base-textAscent() * scalar, width, base-textAscent() * scalar);
text("dp", 0, base);
```



¿Dudas?

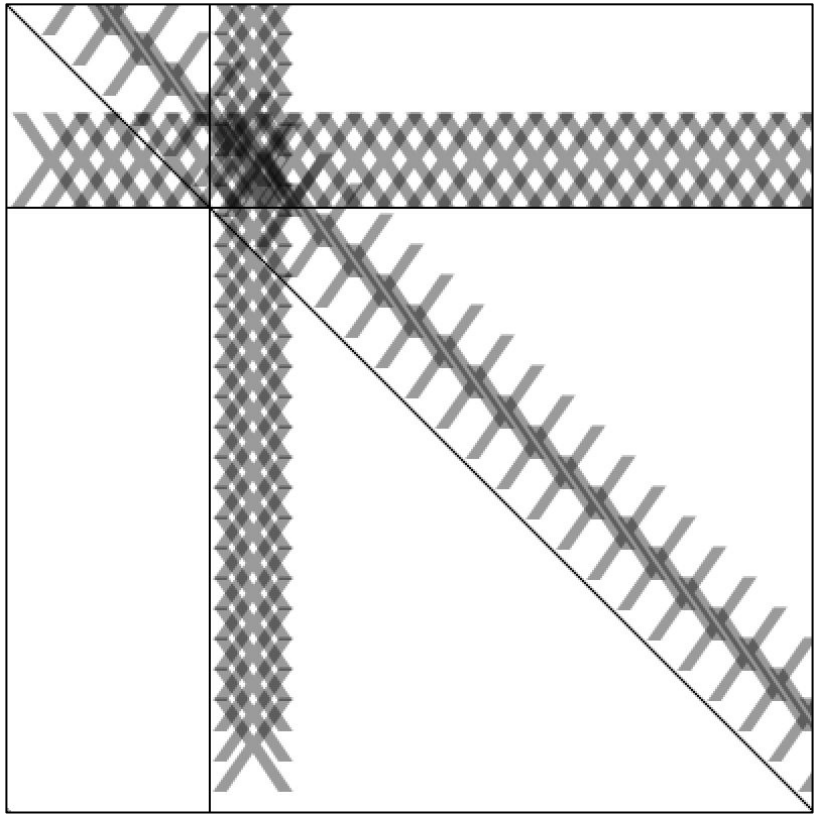
Ejemplos

Texto como elemento decorativo

```
PFont f = createFont("Verdana",64,true);
textFont(f);
size(400,200);
background(255);
fill(0,100);
textAlign(CENTER,CENTER);
text("TEXTO",width/2,100);
text("TEXTO",width/2+15,100);
```

Repetición

```
PFont f = createFont("Verdana",64,true);
textFont(f);
size(400,400);
background(255);
fill(0,100);
line(0,100,width,100);
line(100,0,100,height);
line(0,0,width,height);
for(int i=0;i<width;i+=15){
  text("X",i,100);
  text("X",100,i);
  text("X",i,i);
}
```

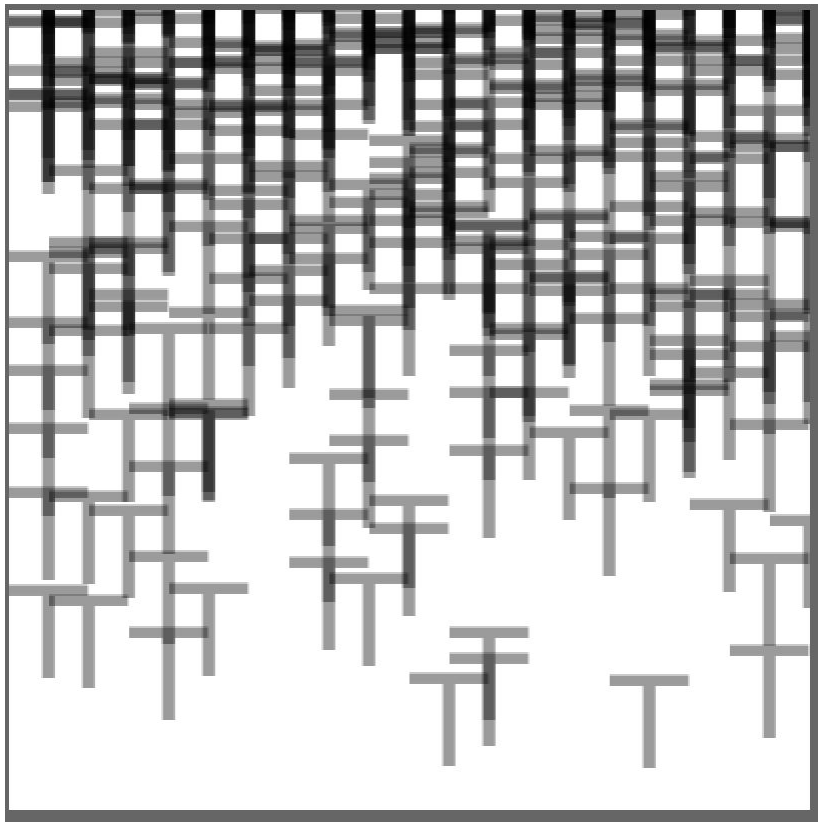


Tamaño aleatorio

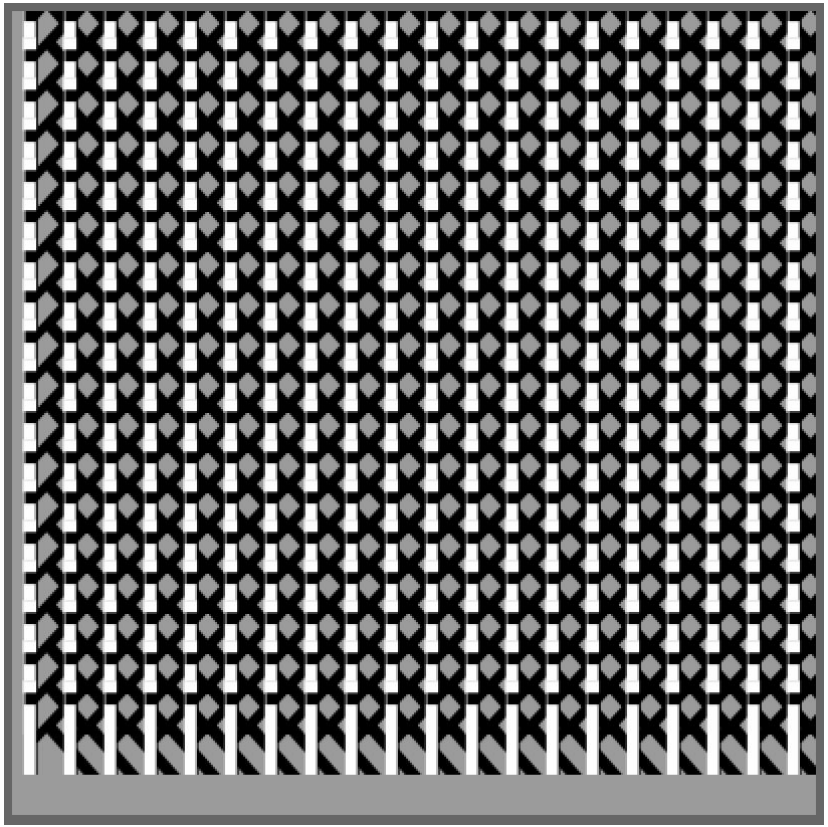
```
PFont f = createFont("Verdana",64,true);
String letters= "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
textAlign(LEFT,CENTER);
float shift=width/(letters.length()+1);
textFont(f);
size(400,400);
background(255);
fill(0);
for(int i=0;i<letters.length();i++){
  textSize(random(140));
  text(letters.charAt(i),shift*i,shift*i);
}
```

Texturas de texto

```
PFont f = createFont("Verdana",64,true);
textFont(f);
size(400,400);
background(255);
fill(0,100);
for(int i=0;i<width;i+=20){
  for(int j=0;j<height;j+=20){
    text("T",i,random(j));
  }
}
```

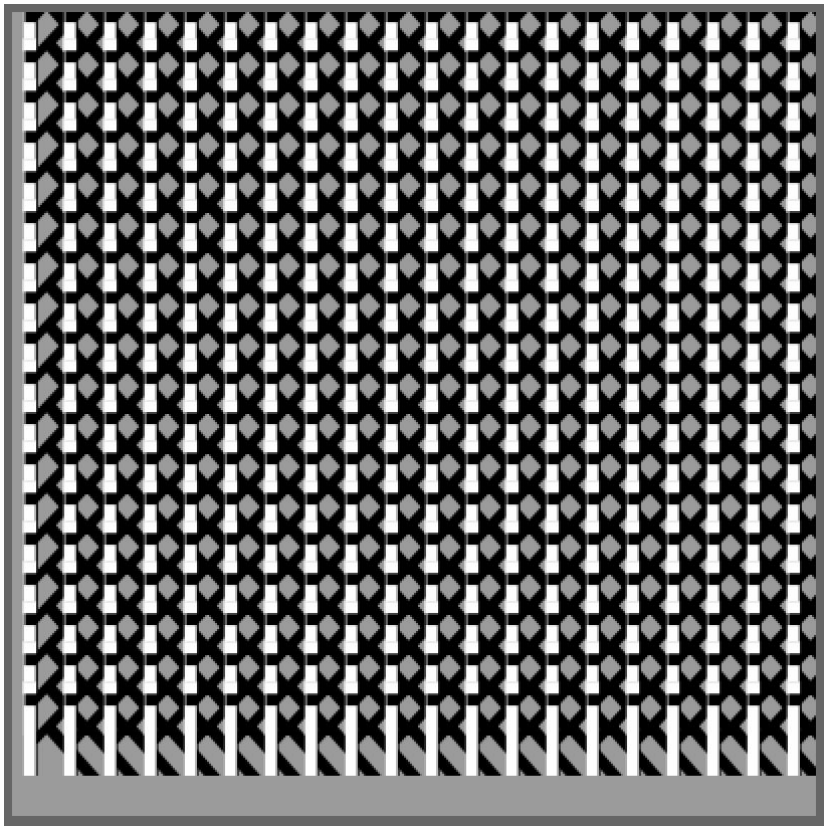


¿Qué letras son?



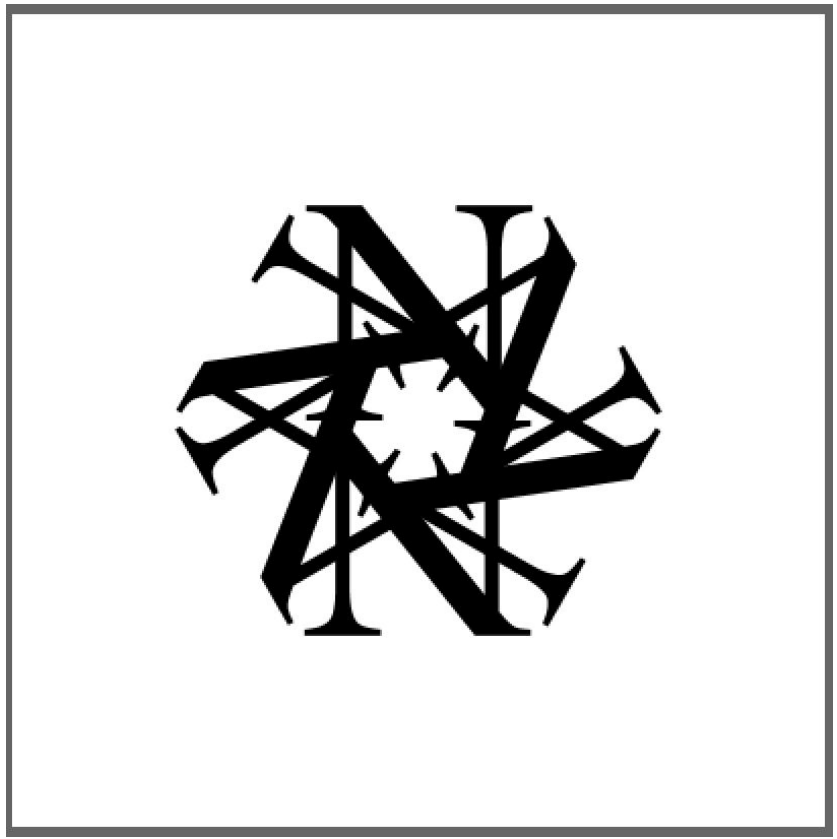
Texturas de texto

```
PFont f = createFont("Verdana",64,true);
textFont(f);
size(400,400);
background(155);
fill(0);
for(int i=0;i<width;i+=20){
  for(int j=0;j<height;j+=20){
    fill(0);
    text("K",i,j);
    fill(255);
    text("i",i,j);
  }
}
```



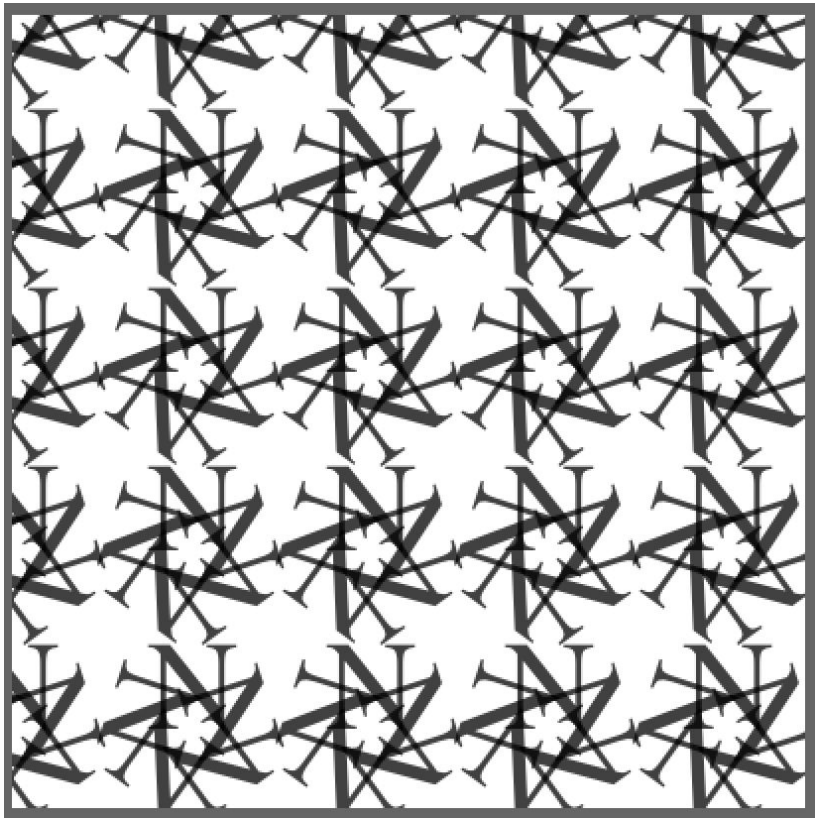
Rotación y repetición

```
PFont f = createFont("Serif",160,true);
int repeatNr =6;
int rotationDegrees = 360/repeatNr;
textFont(f);
textAlign(CENTER);
size(400,400);
background(255);
fill(0);
translate(width/2,height/2);
for(int i=0;i<repeatNr;i++){
  rotate(radians(rotationDegrees));
  text("N",0,0);
}
```



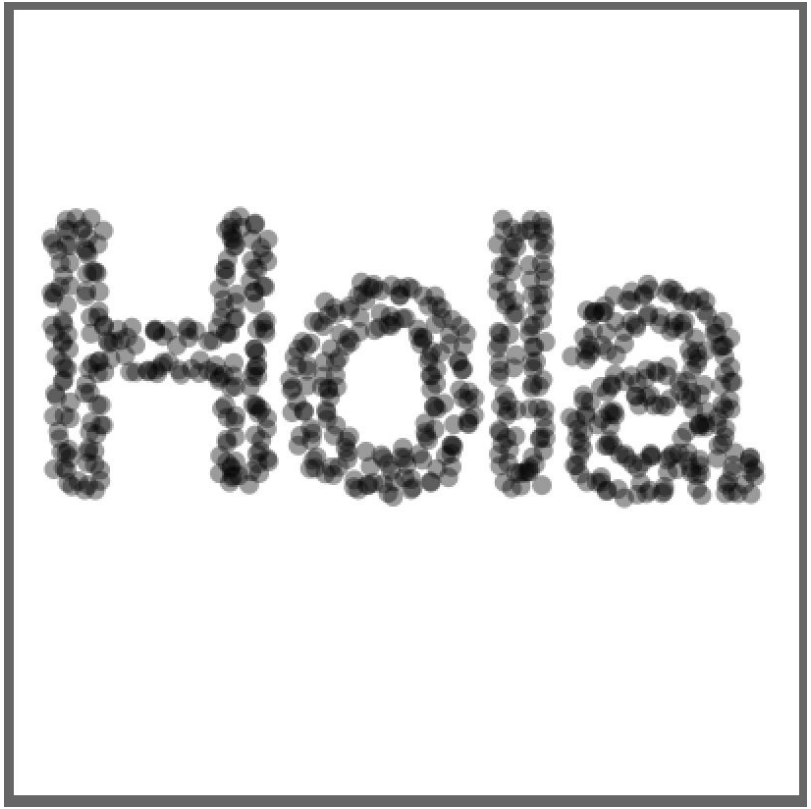
Texturas

```
PFont f = createFont("Serif",64,true);
int repeatNr = 5;
int rotationDegrees = 360/repeatNr;
textFont(f);
size(400,400);
background(255);
fill(0,190);
textAlign(CENTER);
for(int i=0;i<width;i+=90){
 for(int j=0;j<height;j+=90){
  for(int r=0;r<repeatNr;r++){
   pushMatrix();
   translate(i,j);
   rotate(radians(rotationDegrees)*r);
   text("N",0,0);
   popMatrix();
  }
 }
}
```



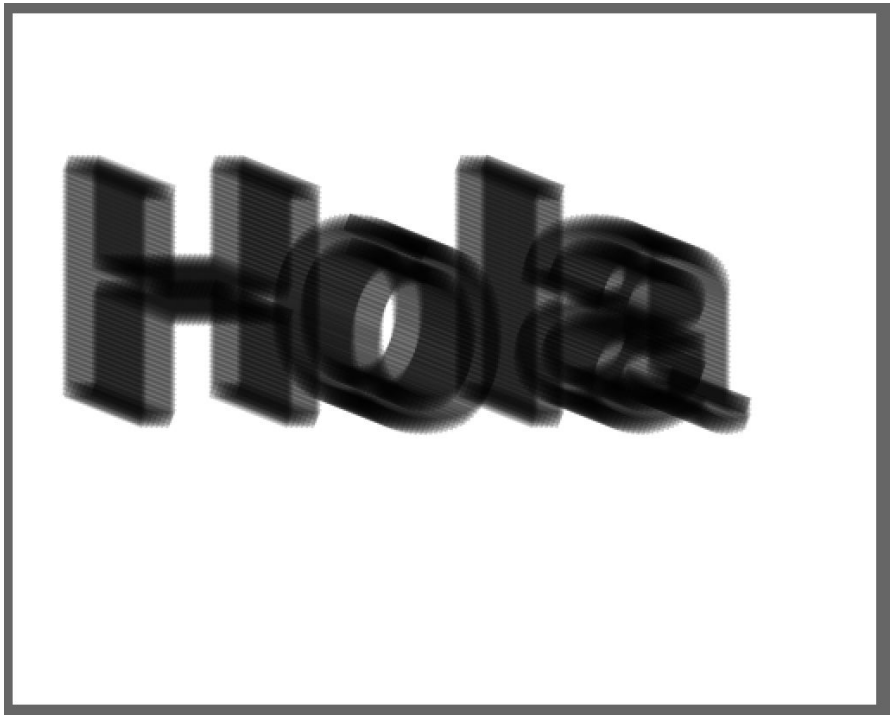
Geomerative y contorno de las letras

```
import geomerative.*;
String txt = "Hola";
RPoint[] pnts;
void setup(){
  size(400,400);
  fill(0,100);
  noStroke();
  RG.init(this);
  RFont font = new RFont("FreeSans.ttf", 180, RFont.LEFT);
  RCommand.setSegmentLength (3);
  RGroup grp = font.toGroup(txt);
  pnts = grp.getPoints();
  frameRate(10);
}
void draw(){
  background(255);
  translate(10,240);
  for (int i = 0; i < pnts.length; i++ ) {
    ellipse(pnts[i].x+random(-5,5), pnts[i].y+random(-5,5),10,10);
  }
}
```



Contorno de las letras variable

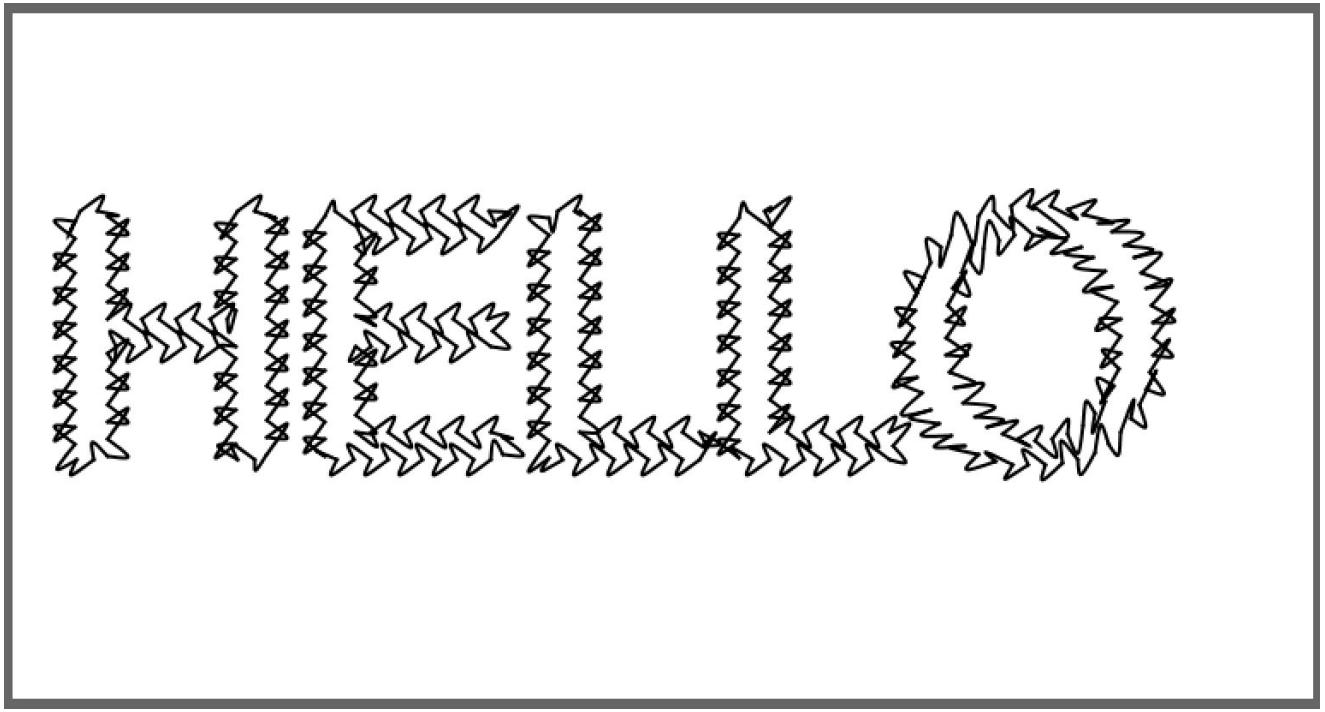
```
import geomerative.*;
RPoint[] pnts;
float rectSize;
void setup(){
 size(500,400);
 fill(0,60);
 noStroke();
 RG.init(this);
 RFont font = new RFont("FreeSans.ttf", 180, RFont.LEFT);
 RCommand.setSegmentLength (3);
 RGroup grp = font.toGroup("Hola");
 pnts = grp.getPoints();
}
void draw(){
 background(255);
 translate(60,240);
 rectSize = map(mouseY,0,height,2,50);
 for (int i = 0; i < pnts.length; i++ ) {
  pushMatrix();
   translate(pnts[i].x, pnts[i].y);
   rotate(map(mouseX,0,width,0,TWO_PI));
   rect(0,0,rectSize,10);
  popMatrix();
 }
}
```



Generative Design

http://www.generative-gestaltung.de/1/P_3_2_2_01

http://www.generative-gestaltung.de/1/code



Resumiendo

Sabemos escribir en la posición deseada.

Sabemos usar y cargar fuentes.

Sabemos generar bitmaps de fuentes para poder compartir tranquilamente los proyectos.