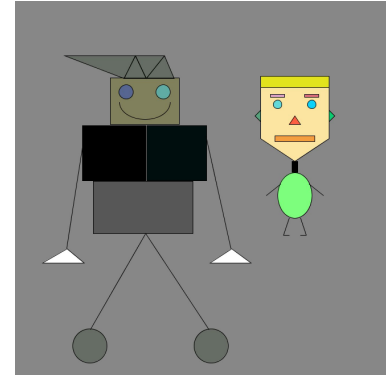
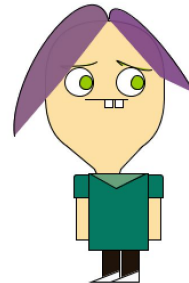
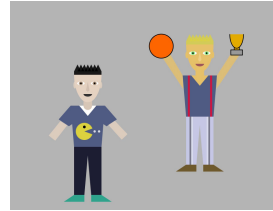
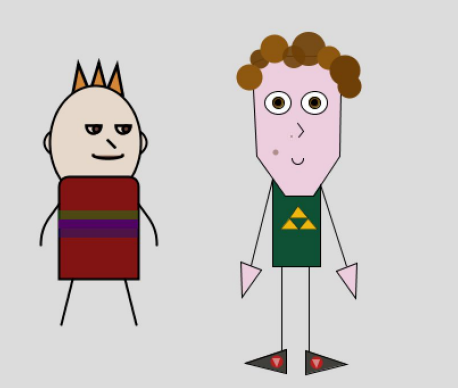
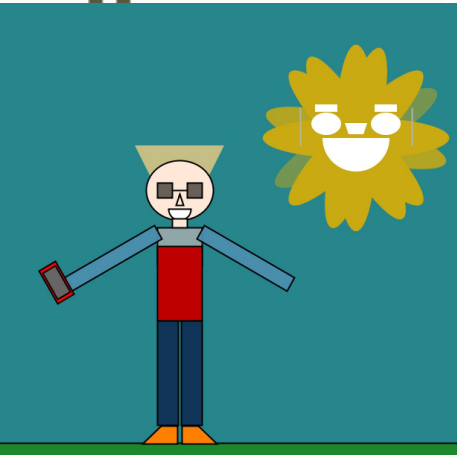
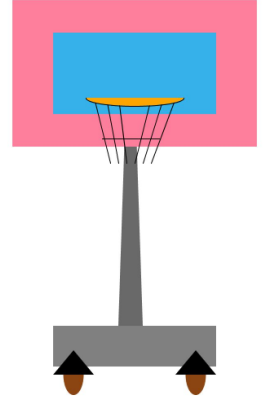
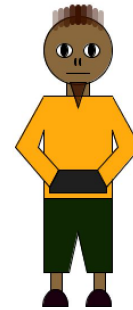
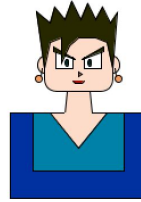
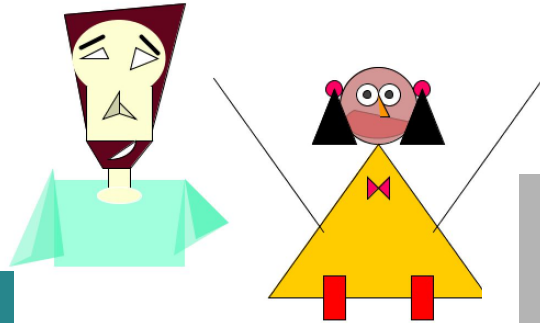




PPEM 2020

Imágenes

Bienvenidos alumnos 2020





Links interesantes

<https://www.youtube.com/watch?v=kV8v2GKC8WA>

<https://www.youtube.com/watch?v=I-EIVlHvHRM>

<https://www.openprocessing.org/sketch/486307>

<https://christianmiolclair.com/blackberrywinter>

<https://www.behance.net/gallery/14442795/Phantogram>

<https://www.openprocessing.org/sketch/422167>

<https://www.openprocessing.org/sketch/392202>

<https://es.gizmodo.com/este-fantastico-paisaje-esta-dibujado-usando-solo-codig-1842307725/amp>

https://www.youtube.com/watch?v=lvymqDif_9g

<https://www.youtube.com/watch?v=t1wBwyS94xY>

<https://www.youtube.com/watch?v=vBoPZg9ru1s>

<https://www.youtube.com/watch?v=Qjtj-nlPJaE>

<https://vimeo.com/150181307>

<https://vimeo.com/2845582>



Repaso

PGraphics

PShape para visualizar formato svg

Imágen como array de píxeles

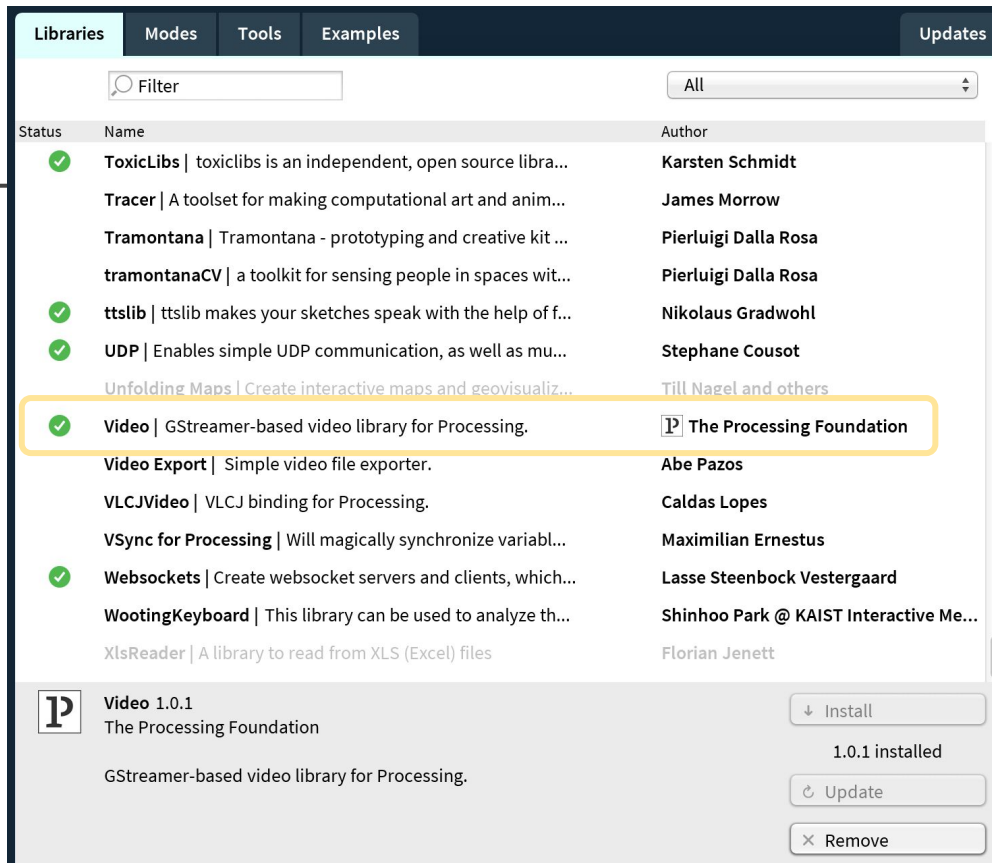
Funciones para acceder y manipular imágenes

Imágenes en movimiento (video)


Instalar la librería de video

Sketch -> import library ->
add library -> Video

Permite reproducir y grabar
videos.



The screenshot shows the 'Libraries' panel in the Processing IDE. The 'Libraries' tab is active, and a search filter is set to 'All'. A list of libraries is displayed, with the 'Video' library highlighted by a yellow box. The 'Video' library is described as 'GStreamer-based video library for Processing.' and is attributed to 'The Processing Foundation'. Below the list, the details for the 'Video 1.0.1' library are shown, including the version number, the author 'The Processing Foundation', and the description 'GStreamer-based video library for Processing.'. Action buttons for 'Install', 'Update', and 'Remove' are visible on the right side of the details panel.

Status	Name	Author
✓	ToxicLibs toxiclibs is an independent, open source libra...	Karsten Schmidt
	Tracer A toolset for making computational art and anim...	James Morrow
	Tramontana Tramontana - prototyping and creative kit ...	Pierluigi Dalla Rosa
	tramontanaCV a toolkit for sensing people in spaces wit...	Pierluigi Dalla Rosa
✓	ttslib ttslib makes your sketches speak with the help of f...	Nikolaus Gradwohl
✓	UDP Enables simple UDP communication, as well as mu...	Stephane Cousot
	Unfolding Maps Create interactive maps and geovisualiz...	Till Nagel and others
✓	Video GStreamer-based video library for Processing.	 The Processing Foundation
	Video Export Simple video file exporter.	Abe Pazos
	VLCJVideo VLCJ binding for Processing.	Caldas Lopes
	VSync for Processing Will magically synchronize variabl...	Maximilian Ernestus
✓	Websockets Create websocket servers and clients, which...	Lasse Steenbock Vestergaard
	WootingKeyboard This library can be used to analyze th...	Shinhoo Park @ KAIST Interactive Me...
	XlsReader A library to read from XLS (Excel) files	Florian Jenett

Video 1.0.1
The Processing Foundation
GStreamer-based video library for Processing.

↓ Install
1.0.1 installed
↻ Update
× Remove



Movie - formatos soportados

<https://github.com/processing/processing-video/wiki/Video-1.x-notes>

Sección: **Media Formats**



¿Error?

Error

"IllegalArgumentException: No such Gstreamer factory: v4l2src"

en mi Linux se solucionó instalando

"gstreamer0.10-plugins-good"



Movie



```
frameRate() // fijar framerate deseado (frames por segundo), NO me funcionó con .mov, ni con .mp4
speed()     // fijar velocidad
duration()  // preguntar por la duración
time()      // segundos que pasaron desde que arrancó el video
jump()      // posición en segundos hacia cuál debería saltar la reproducción (y se sigue reproduciendo el video)
available() // "true" si hay frames para leer
play()      // para reproducir el video una vez y parar en el último frame
loop()      // para looppear
noLoop()    // para detener el looping
pause()     // pausar
stop()      // detener la reproducción
read()      // leer el frame actual
```


Reproducir video

```
import processing.video.*;
Movie movie;

void setup() {
  size(640, 360);
  background(0);
  movie = new Movie(this, "visuals.mp4");
  movie.loop();
}
```

```
void movieEvent(Movie m) {
  m.read();
}
```

```
void draw() {
  image(movie, 0, 0, width, height);
}
```

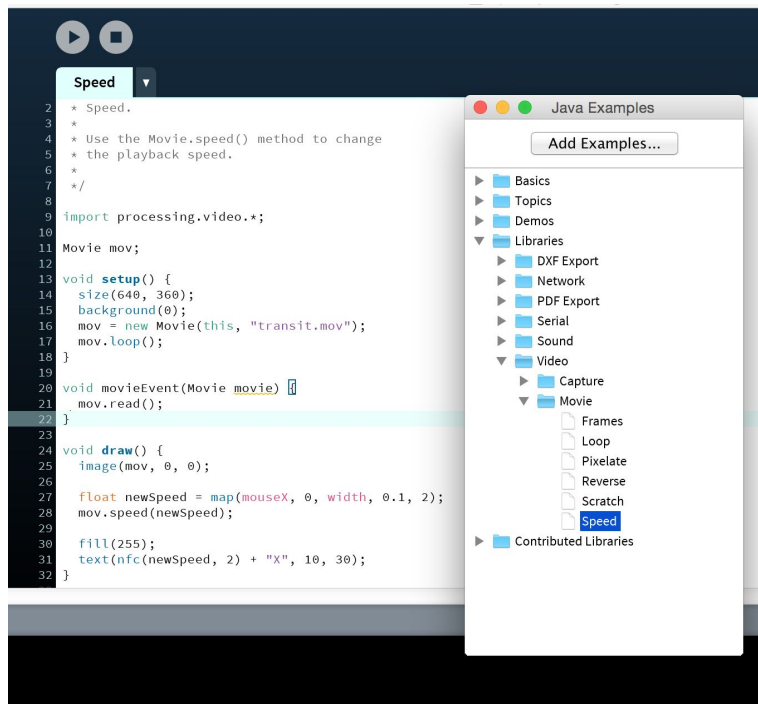
```
import processing.video.*;
Movie movie;

void setup() {
  size(640, 360);
  background(0);
  movie = new Movie(this, "visuals.mp4");
  movie.loop();
}
```

```
void draw() {
  if (movie.available() == true) {
    movie.read();
  }
}
```

```
image(movie, 0, 0, width, height);
}
```

Control de velocidad de la reproducción



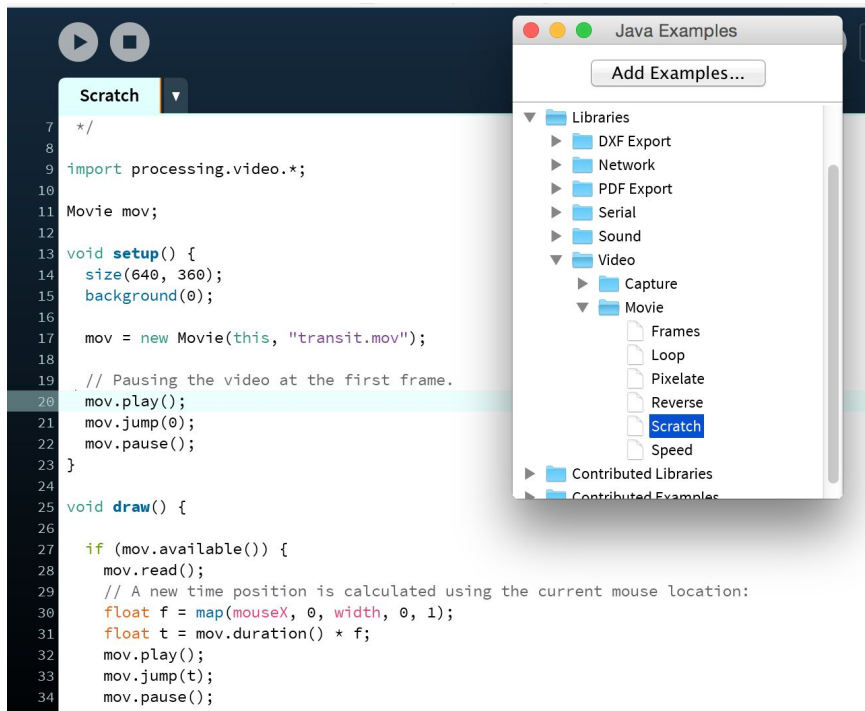
The image shows a code editor window with a dark theme. The code is in Java and implements a video player with speed control. The code includes comments, an import statement, a Movie object, and methods for setup, movieEvent, and draw. The draw method uses a slider to map mouse movement to a speed value between 0.1 and 2.0. A library browser window titled "Java Examples" is open on the right, showing a tree view of examples. The "Speed" example is selected under the "Video" > "Movie" category.

```
2  * Speed.  
3  *  
4  * Use the Movie.speed() method to change  
5  * the playback speed.  
6  *  
7  */  
8  
9  import processing.video.*;  
10  
11  Movie mov;  
12  
13  void setup() {  
14    size(640, 360);  
15    background(0);  
16    mov = new Movie(this, "transit.mov");  
17    mov.loop();  
18  }  
19  
20  void movieEvent(Movie movie) {  
21    mov.read();  
22  }  
23  
24  void draw() {  
25    image(mov, 0, 0);  
26  
27    float newSpeed = map(mouseX, 0, width, 0.1, 2);  
28    mov.speed(newSpeed);  
29  
30    fill(255);  
31    text(nfc(newSpeed, 2) + "x", 10, 30);  
32  }
```

Java Examples
Add Examples...

- Basics
- Topics
- Demos
- Libraries
 - DXF Export
 - Network
 - PDF Export
 - Serial
 - Sound
 - Video
 - Capture
 - Movie
 - Frames
 - Loop
 - Pixelate
 - Reverse
 - Scratch
 - Speed
- Contributed Libraries

Saltar a una posición en la línea de tiempo



The image shows a code editor window with a dark theme. The code is written in Java and uses the Processing library for video playback. A library panel on the right side of the editor is open, showing a tree view of examples. The 'Scratch' example is highlighted in blue. The code in the editor is as follows:

```
7  */
8
9  import processing.video.*;
10
11 Movie mov;
12
13 void setup() {
14     size(640, 360);
15     background(0);
16
17     mov = new Movie(this, "transit.mov");
18
19     // Pausing the video at the first frame.
20     mov.play();
21     mov.jump(0);
22     mov.pause();
23 }
24
25 void draw() {
26
27     if (mov.available()) {
28         mov.read();
29         // A new time position is calculated using the current mouse location:
30         float f = map(mouseX, 0, width, 0, 1);
31         float t = mov.duration() * f;
32         mov.play();
33         mov.jump(t);
34         mov.pause();
```

Vídeo pixelado

```
import processing.video.*;
```

```
int blockSize = 15;
```

```
Movie mov;
```

```
void setup() {
```

```
  size(640, 360);
```

```
  noStroke();
```

```
  mov = new Movie(this, "visuals.mp4");
```

```
  mov.loop();
```

```
}
```

```
void draw() {
```

```
  if (mov.available() == true) {
```

```
    mov.read();
```

```
    background(0);
```

```
    for (int i = 0; i < width; i+=blockSize){
```

```
      for (int j = 0; j < height; j+=blockSize){
```

```
        color c = mov.get(i, j);
```

```
        fill(c);
```

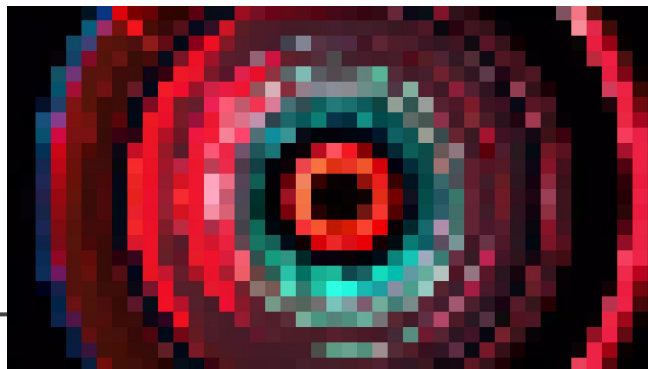
```
        rect(i, j, blockSize, blockSize);
```

```
      }
```

```
    }
```

```
  }
```

```
}
```



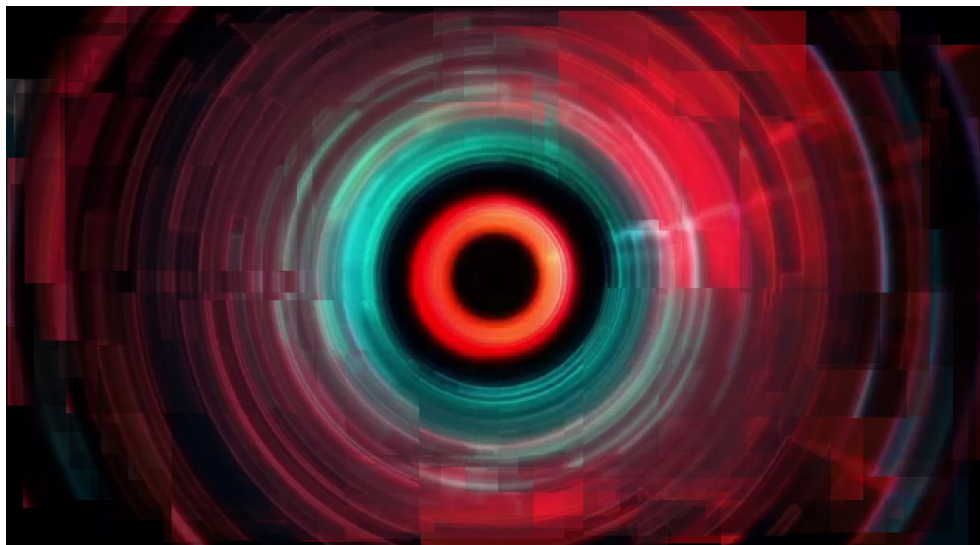
Recortar parte del video con get

```
import processing.video.*;
Movie mov;
void setup() {
  size(640, 360);
  mov = new Movie(this, "visuals.mp4");
  mov.loop();
}
void draw() {
  if (mov.available() == true) {
    mov.read();
    image(mov.get(240,135,160,90), 0, 0, width, height);
  }
}
```



Recortar partes de cada frame con get

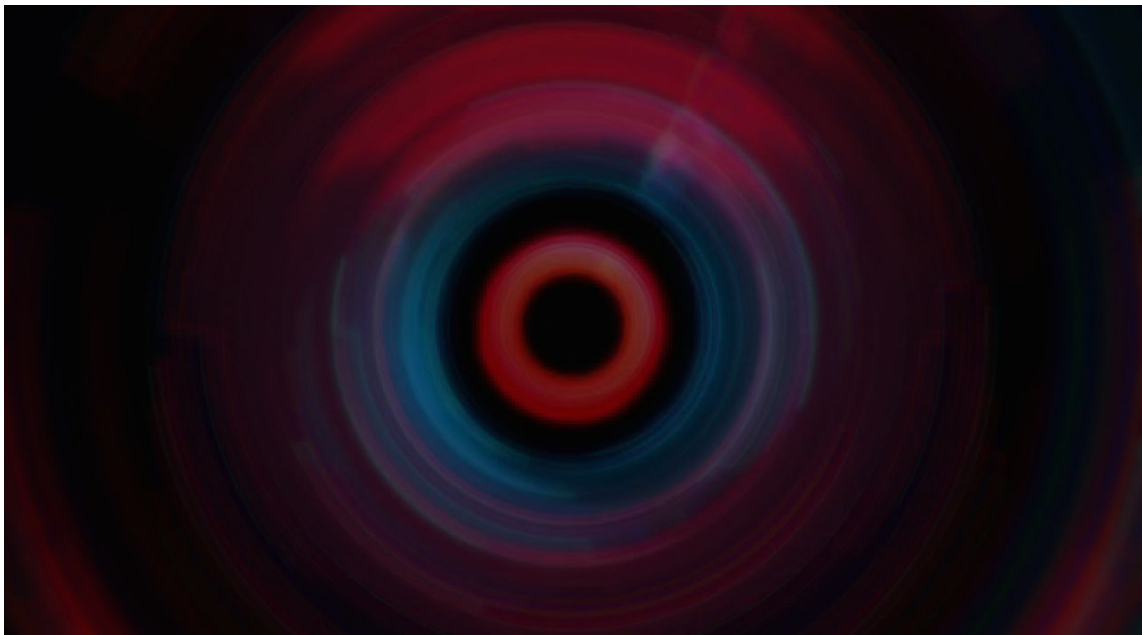
```
import processing.video.*;
Movie mov;
int partSize = 160;
void setup() {
  size(640, 360);
  mov = new Movie(this, "visuals.mp4");
  mov.loop();
  background(0);
}
void draw() {
  if (mov.available() == true) {
    mov.read();
    int randomX = ceil(random(mov.width - partSize));
    int randomY = ceil(random(mov.height - partSize));
    tint(255,100);
    image(mov.get(randomX,randomY,partSize,partSize), randomX, randomY, partSize, partSize);
  }
}
```





Tint

```
import processing.video.*;
Movie mov;
void setup() {
  size(640, 360);
  colorMode(HSB);
  mov = new Movie(this, "visuals.mp4");
  mov.loop();
}
void draw() {
  if (mov.available() == true) {
    mov.read();
    tint(random(255), 255, 255, 40);
    image(mov, 0, 0, width, height);
  }
}
```



Filtros

```
import processing.video.*;
```

```
Movie mov;
```

```
void setup() {
```

```
  size(640, 360);
```

```
  mov = new Movie(this, "visuals.mp4");
```

```
  mov.loop();
```

```
}
```

```
void draw() {
```

```
  if (mov.available() == true) {
```

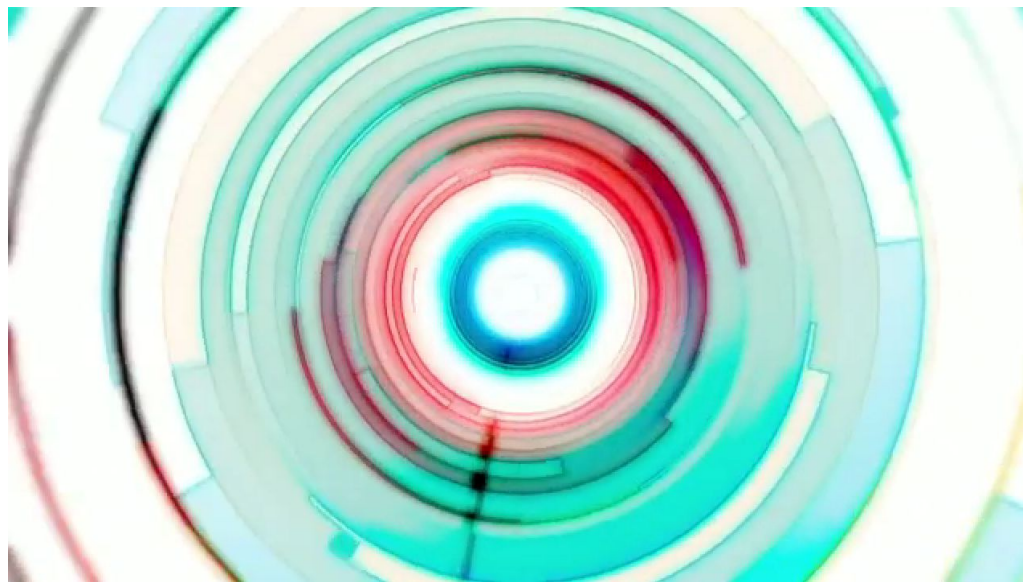
```
    mov.read();
```

```
    mov.filter(INVERT);
```

```
    image(mov, 0, 0, width, height);
```

```
  }
```

```
}
```





Captura en vivo

`Capture (parent)`

`Capture (parent, requestConfig)`

`Capture (parent, requestWidth, requestHeight)`

`Capture (parent, requestWidth, requestHeight, frameRate)`

`Capture (parent, requestWidth, requestHeight, cameraName)`

`Capture (parent, requestWidth, requestHeight, cameraName, frameRate)`



Cámaras disponibles

```
String[] cameras = Capture.list();
```

```
Camaras disponibles:
```

```
name=FaceTime HD Camera,size=1280x720,fps=30  
name=FaceTime HD Camera,size=1280x720,fps=15  
name=FaceTime HD Camera,size=1280x720,fps=1  
name=FaceTime HD Camera,size=640x360,fps=30  
name=FaceTime HD Camera,size=640x360,fps=15  
name=FaceTime HD Camera,size=640x360,fps=1  
name=FaceTime HD Camera,size=320x180,fps=30  
name=FaceTime HD Camera,size=320x180,fps=15  
name=FaceTime HD Camera,size=320x180,fps=1  
name=FaceTime HD Camera,size=160x90,fps=30  
name=FaceTime HD Camera,size=160x90,fps=15  
name=FaceTime HD Camera,size=160x90,fps=1  
name=FaceTime HD Camera,size=80x45,fps=30  
name=FaceTime HD Camera,size=80x45,fps=15  
name=FaceTime HD Camera,size=80x45,fps=1
```



Captura en vivo

`list()`

`read()`

`available()`

`start()`

`stop()`



Acceso a la cámara

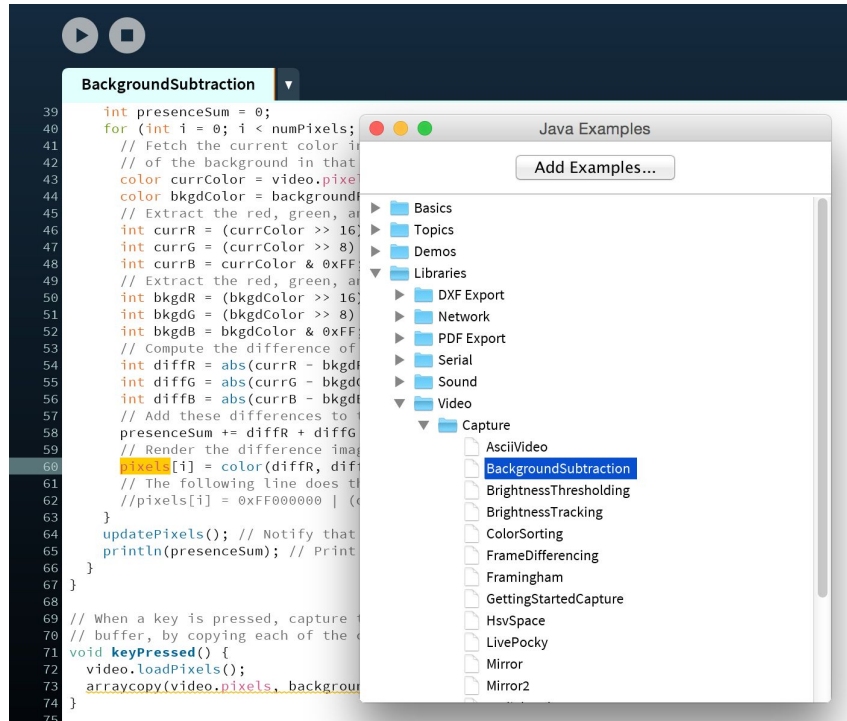
```
import processing.video.*;
Capture cam;
```

```
void setup() {
  size(640, 480);
  String[] cameras = Capture.list();
```

```
  if (cameras.length == 0) {
    println("No hay cámaras disponibles.");
    exit();
  } else {
    println("Cámaras disponibles:");
    for (int i = 0; i < cameras.length; i++) {
      println(cameras[i]);
    }
    cam = new Capture(this, cameras[0]);
    cam.start();
  }
}
```

```
void draw() {
  if (cam.available() == true) {
    cam.read();
  }
  image(cam, 0, 0);
  //set(0, 0, cam); // más rápido si no usamos resize, transformaciones o tint
}
```

Sustracción del fondo



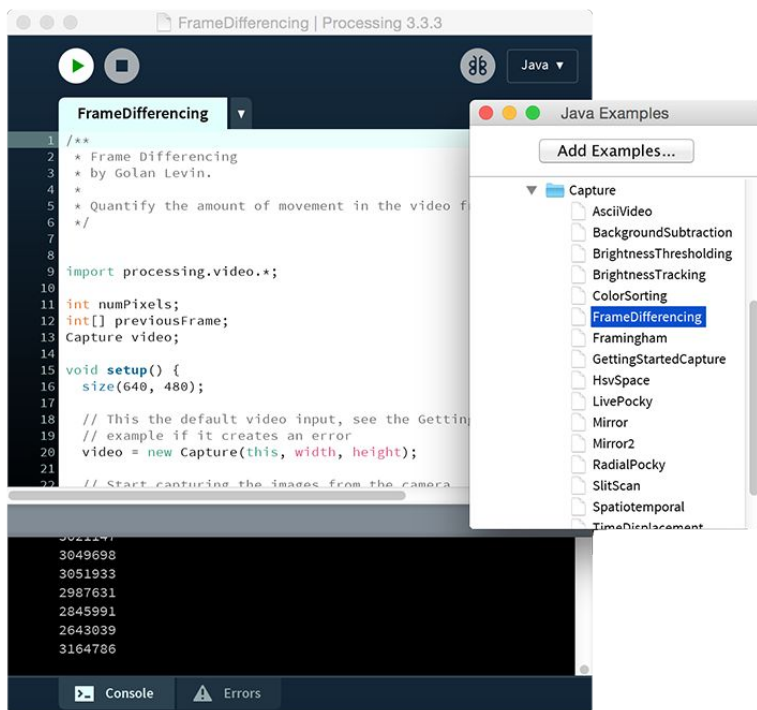
The image shows a Java IDE with a code editor and a file explorer window. The code editor displays the following Java code for background subtraction:

```
39 int presenceSum = 0;
40 for (int i = 0; i < numPixels; i++)
41     // Fetch the current color i
42     // of the background in that
43     color currColor = video.pixels[i];
44     color bkgdColor = background.pixels[i];
45     // Extract the red, green, and blue
46     int currR = (currColor >> 16) & 0xFF;
47     int currG = (currColor >> 8) & 0xFF;
48     int currB = currColor & 0xFF;
49     // Extract the red, green, and blue
50     int bkgdR = (bkgdColor >> 16) & 0xFF;
51     int bkgdG = (bkgdColor >> 8) & 0xFF;
52     int bkgdB = bkgdColor & 0xFF;
53     // Compute the difference of
54     int diffR = abs(currR - bkgdR);
55     int diffG = abs(currG - bkgdG);
56     int diffB = abs(currB - bkgdB);
57     // Add these differences to
58     presenceSum += diffR + diffG + diffB;
59     // Render the difference image
60     pixels[i] = color(diffR, diffG, diffB);
61     // The following line does the
62     //pixels[i] = 0xFF000000 | (currColor & 0xFF000000);
63 }
64 updatePixels(); // Notify that the image has changed
65 println(presenceSum); // Print the number of pixels
66 }
67 }
68
69 // When a key is pressed, capture the
70 // buffer, by copying each of the
71 void keyPressed() {
72     video.loadPixels();
73     arraycopy(video.pixels, background.pixels,
74     }
75 }
```

The file explorer window, titled "Java Examples", shows a tree view of the project structure. The "Video" folder is expanded, and the "Capture" sub-folder is also expanded. The "BackgroundSubtraction" file is highlighted in blue.

- Basics
- Topics
- Demos
- Libraries
 - DXF Export
 - Network
 - PDF Export
 - Serial
 - Sound
 - Video
 - Capture
 - AsciiVideo
 - BackgroundSubtraction
 - BrightnessThresholding
 - BrightnessTracking
 - ColorSorting
 - FrameDifferencing
 - Framingham
 - GettingStartedCapture
 - HsvSpace
 - LivePocky
 - Mirror
 - Mirror2

Sustracción del fondo constante

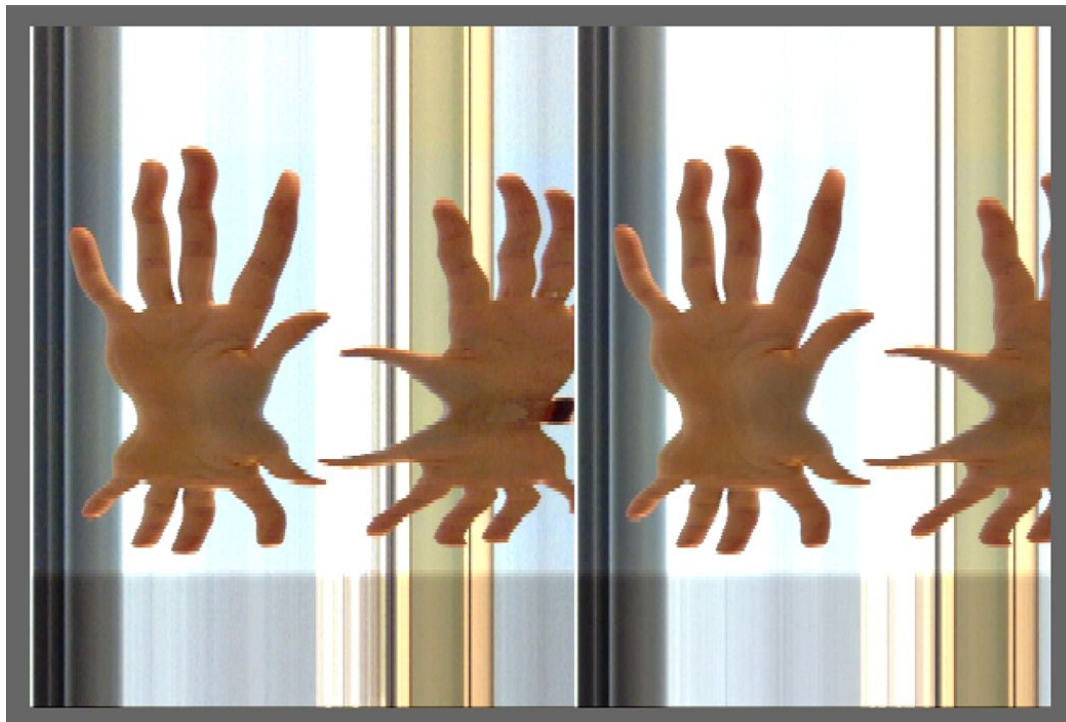




Y otros ejemplos muy interesantes

LivePocky

TimeDisplacement



Resumiendo

