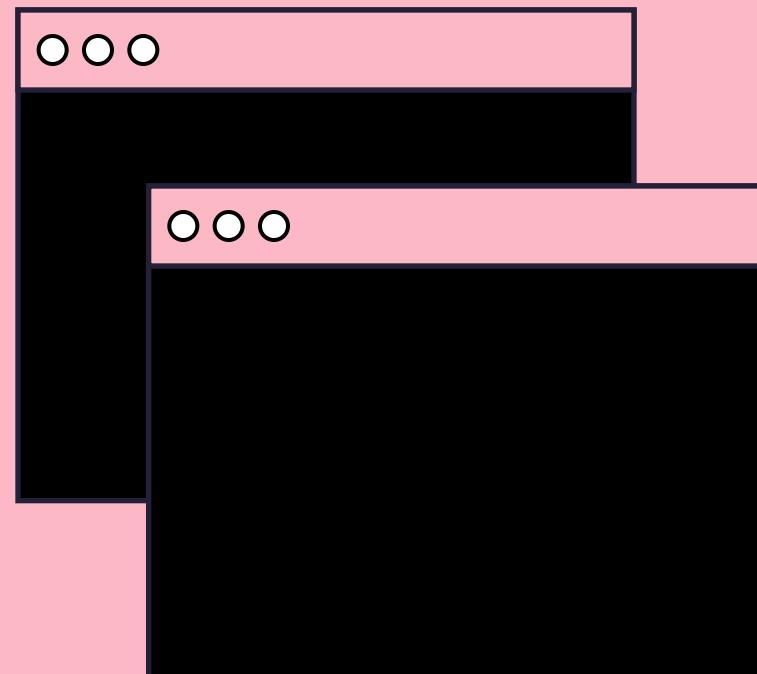
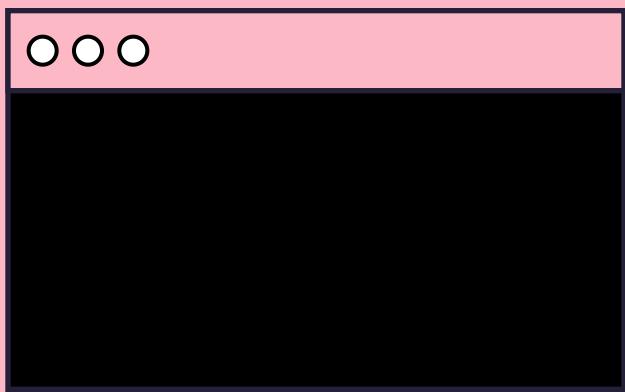


# Interfaces invisibles

VENTANAS



# Hola a todxs



Ejercicios

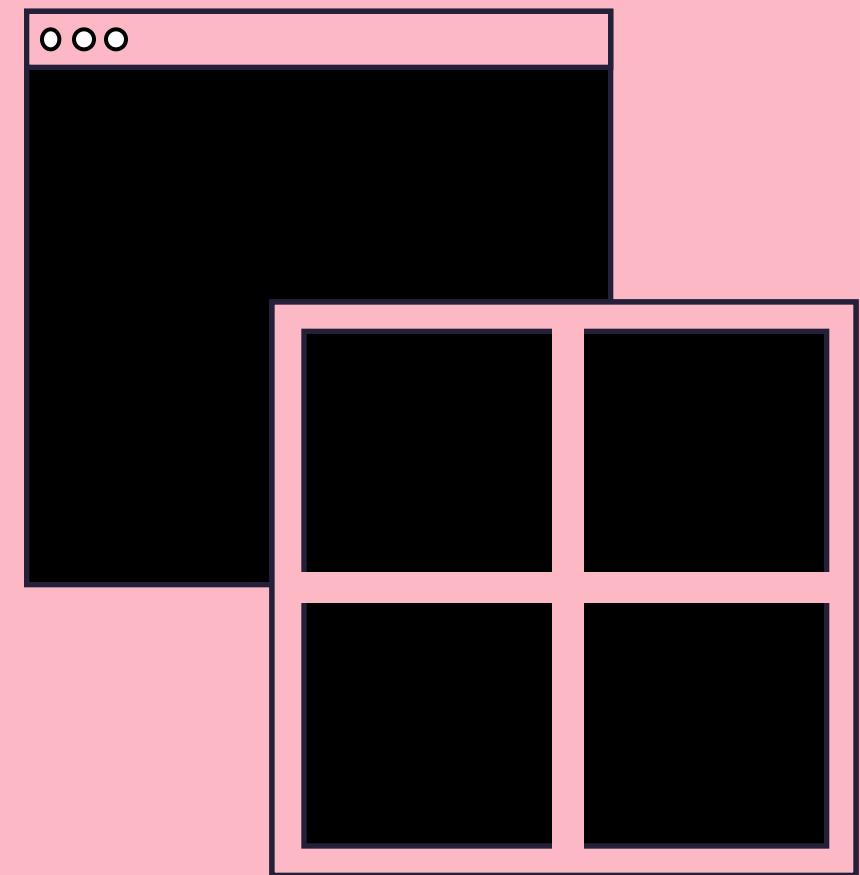


Laurajunco.live

Lauracjunko@gmail.com

# ¿Qué es una ventana?

???



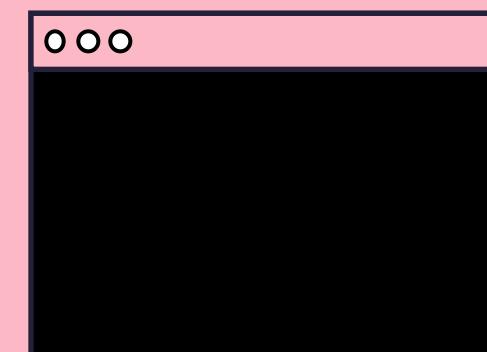
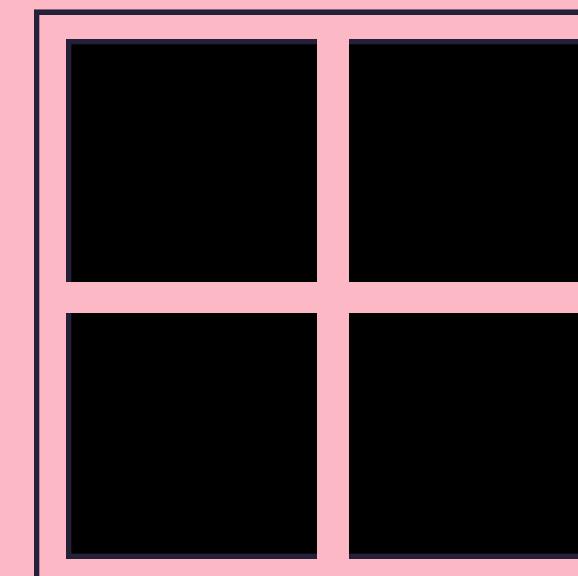
???



???

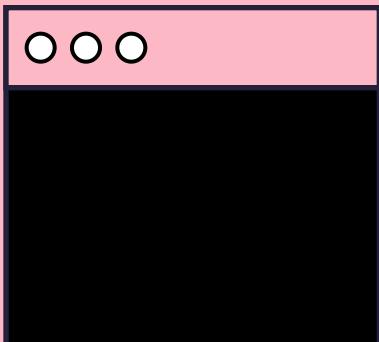


???



# ¿Qué es una ventana?

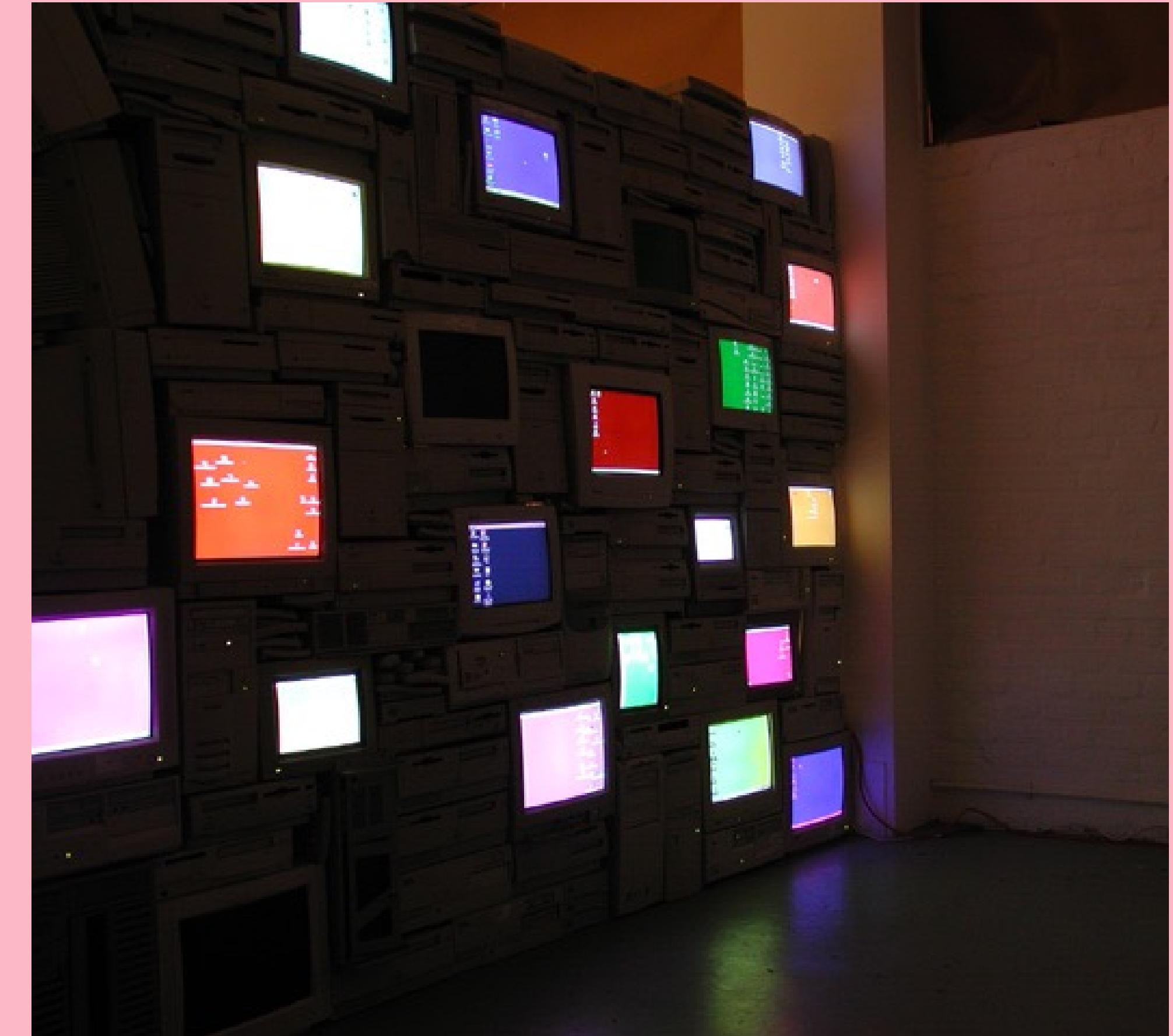
- Parte de la interfaz gráfica de usuario
- Elemento metáforico
- Contenedor de información
- Entrada y salida de datos
- Pasan desapercibidas



# Contenedores invisibles



Firewall (2005) Sandy Smith

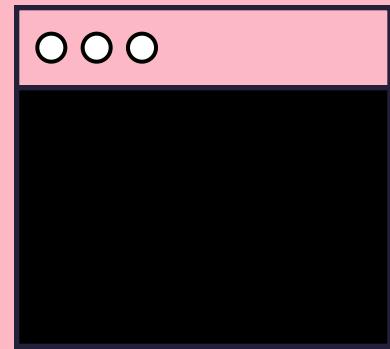


# El escritorio

¿Cómo nace la metáfora de la ventana?

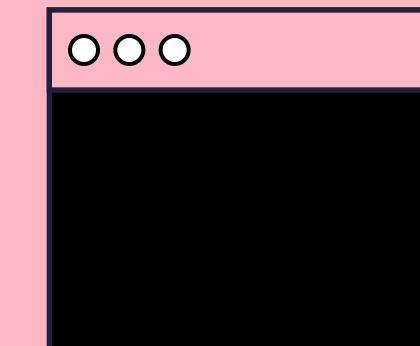
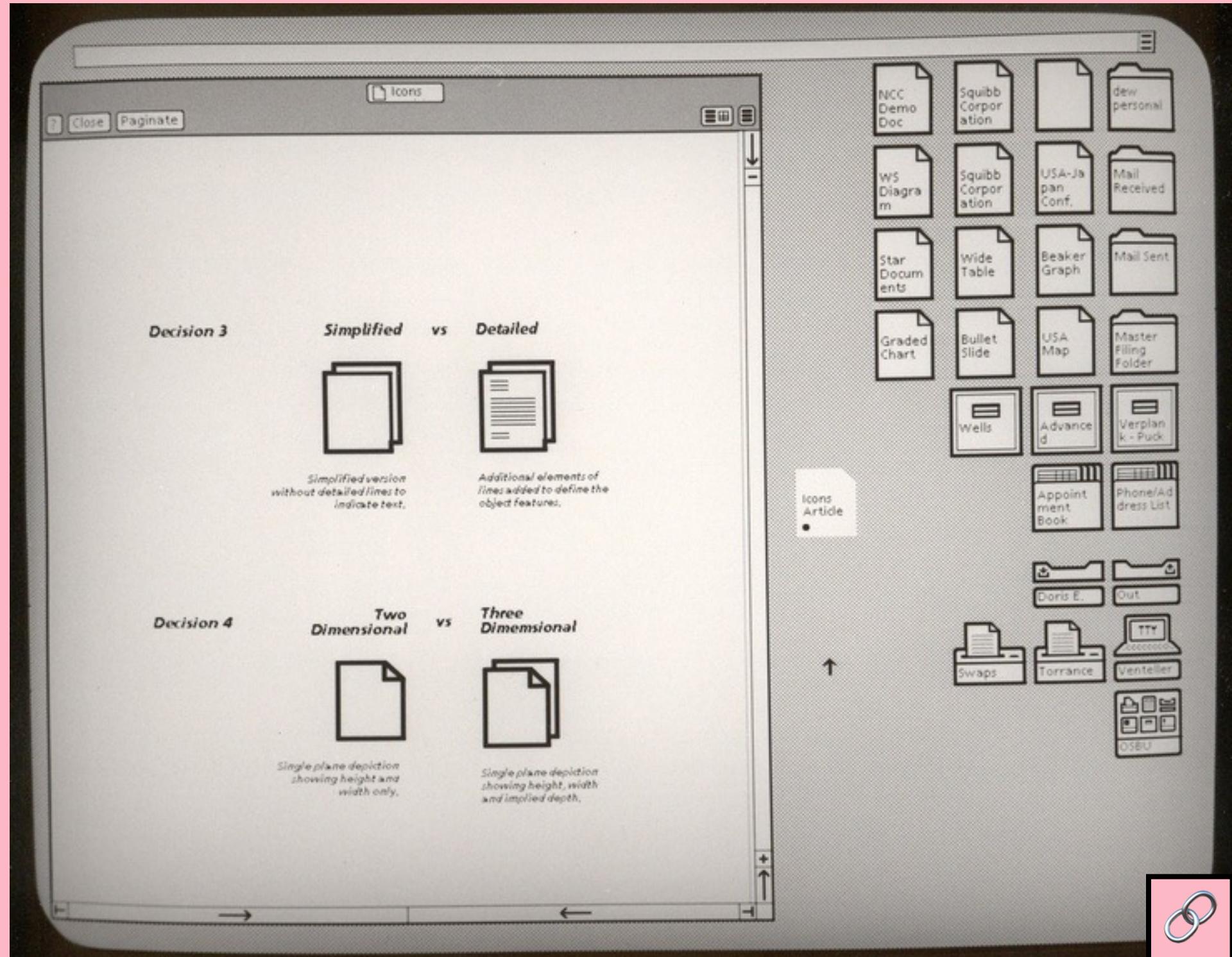


Ordenador personal IBM 5160 y monitor IBM 5151



# El escritorio

¿Cómo nace la metáfora de la ventana?



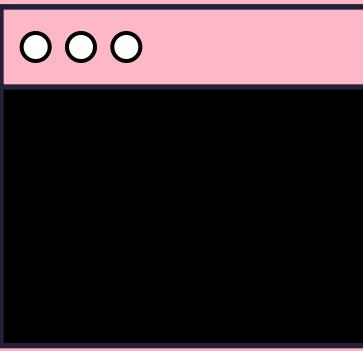
Interfaz gráfica de usuario Xerox Star 8010

# El escritorio

¿Cómo nace la metáfora de la ventana?



Comercial de apple Macintosh



# ¿Y las ventanas?



- ???
- Ventana de una oficina
- Portal
- Delimitador de procesos

# Elementos plásticos

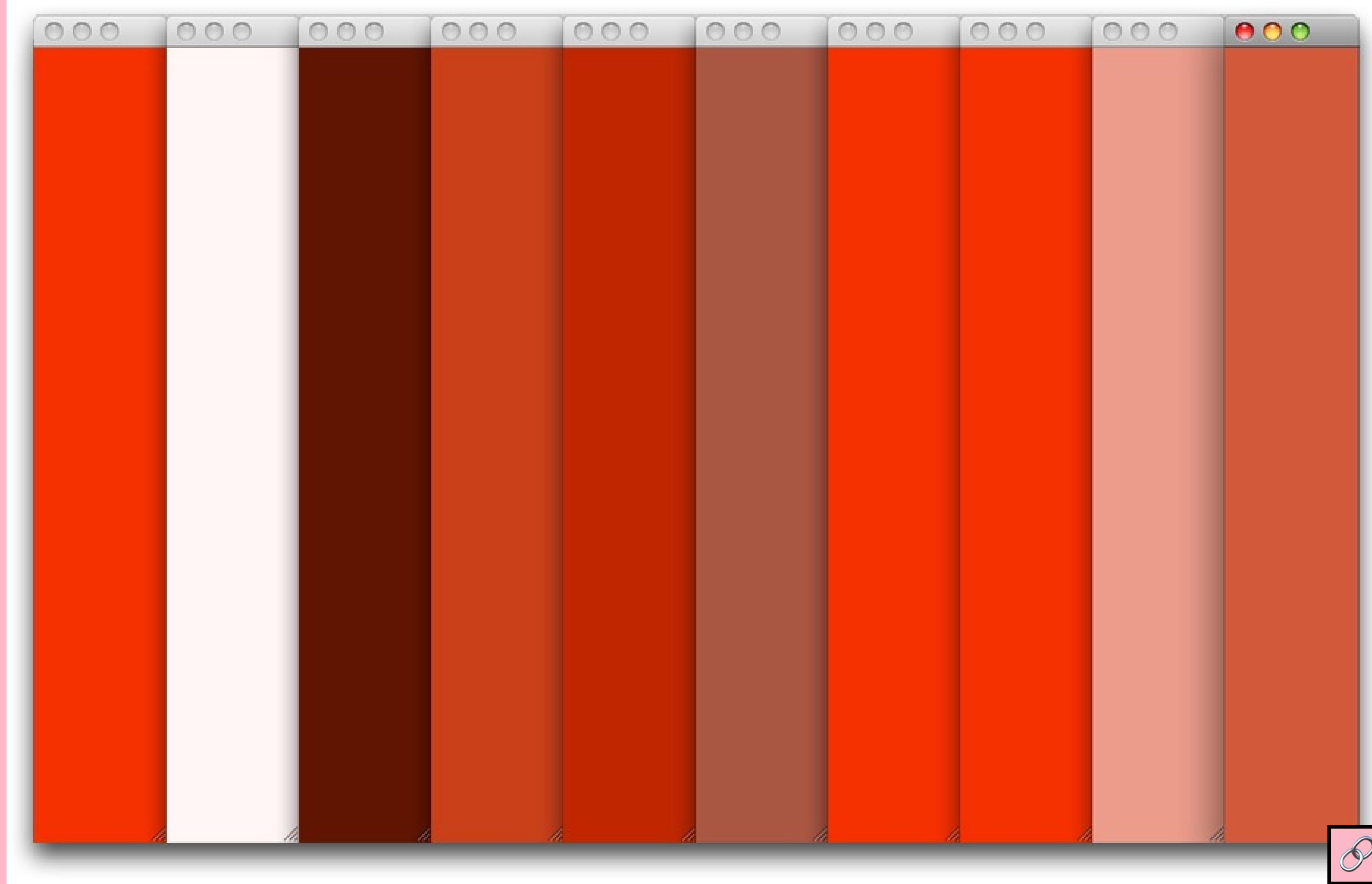


Roman Härak, CC BY-SA 2.0



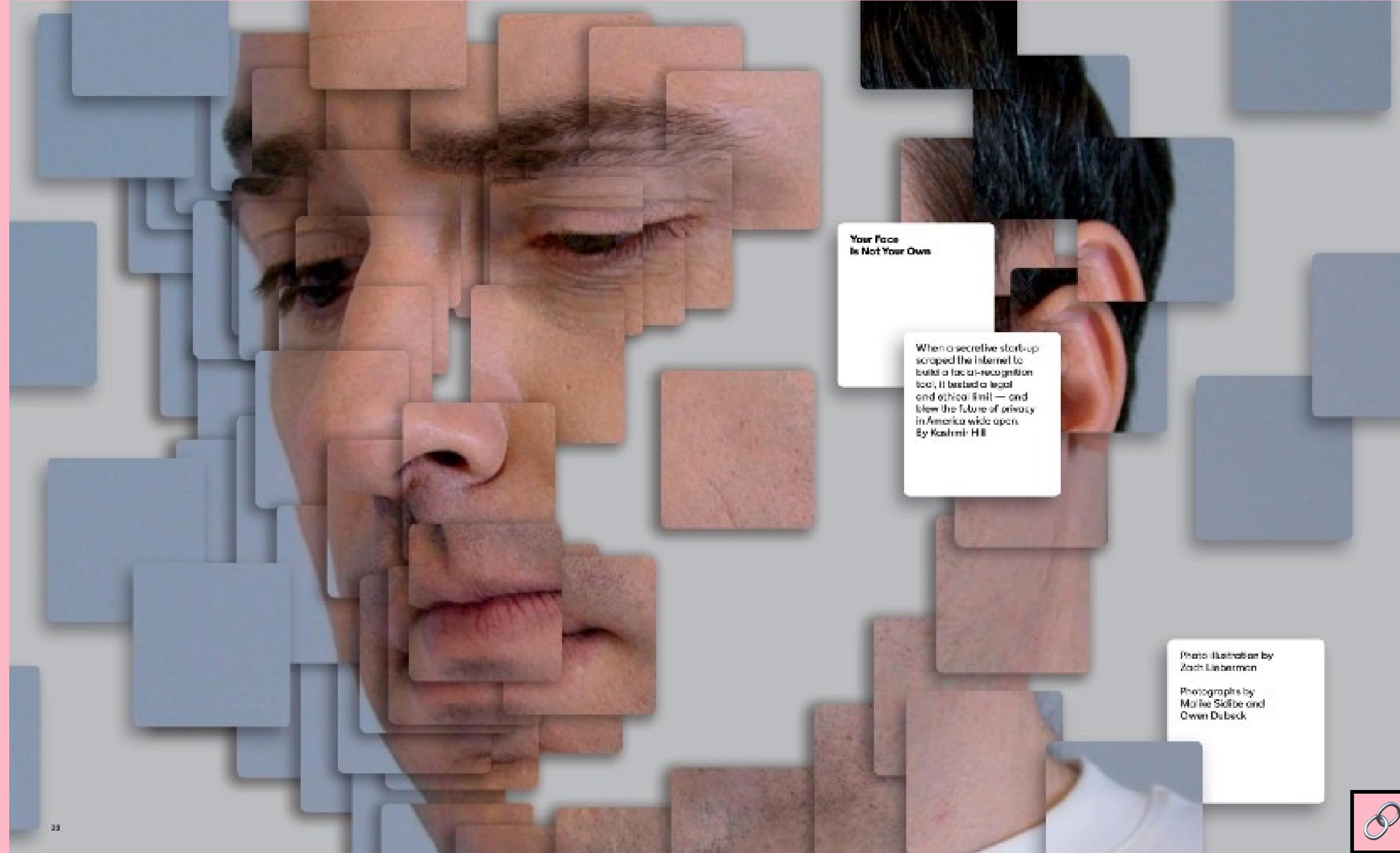
Karl432, CC BY-SA 3.0

# Elementos plásticos



Michæl Demers: “Color Field Paintings”

# Clementos plásticos



Zach Lieberman portada para NYT magazine

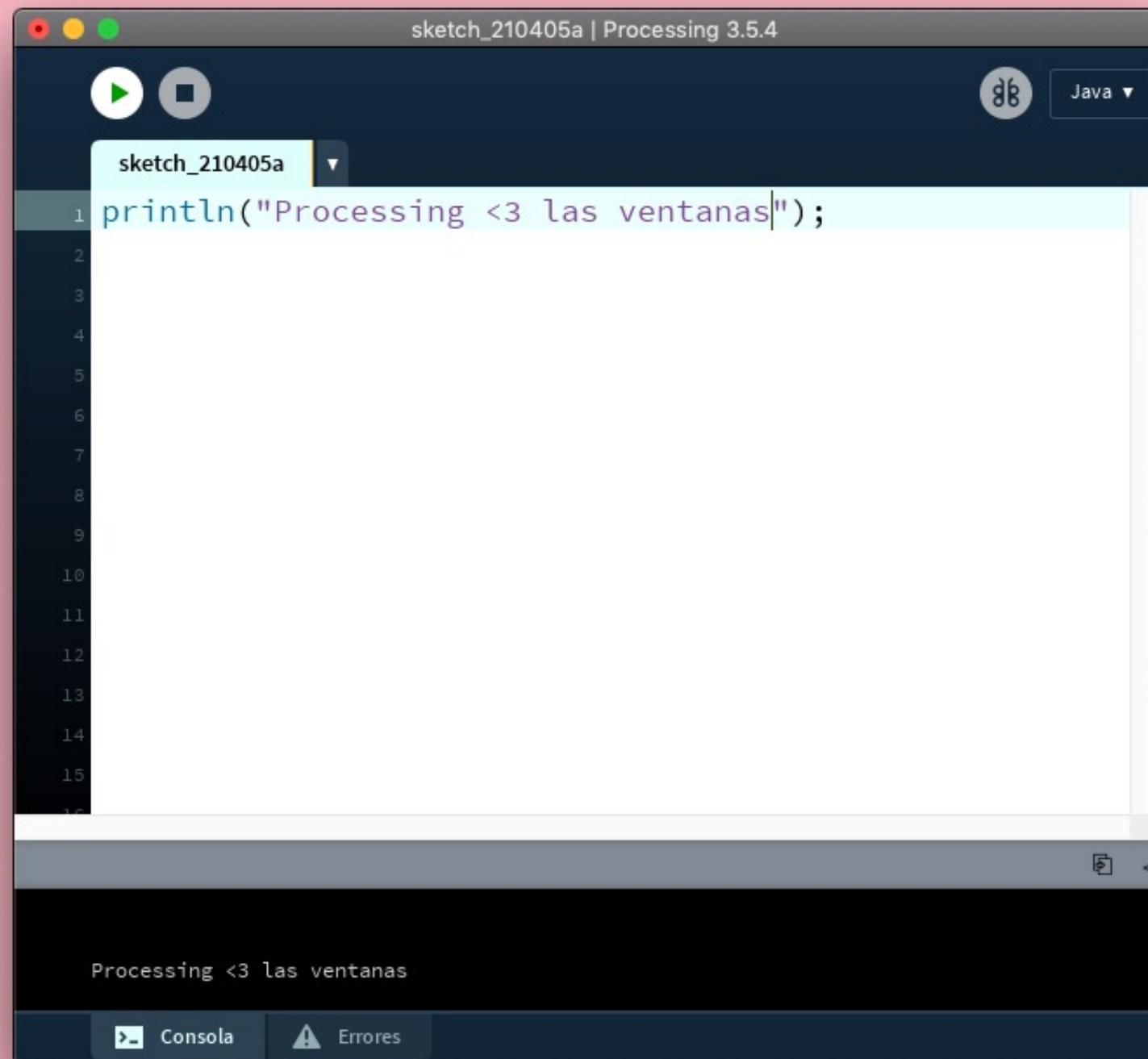


# Elementos plásticos



# Windowed Nyan Cat - Raven Kwok

# ¿Por qué Processing?



- **Aplicaciones en nuestras máquinas**
- **Java! (Write once, run everywhere)**
- **Es fácil**

# Superficies

Objetos con características (posición, tamaño, color)

Tamaño = size(128, 128);

Color = background(255, 255, 0);

Posición = ???

# Superficies

**Objetos que tienen características (posición, tamaño, color)**

**Tamaño** = surface.setSize(128, 128);

**Color** = background(255, 255, 0);

**Posición** = surface.setLocation(500, 500);

# Superficies

Objetos que tienen características (posición, tamaño, color)

Tamaño

= surface.setSize(128, 128);

Color

= background(255, 255, 0);

Posición

= surface.setLocation(500, 500);

Ancho de la pantalla

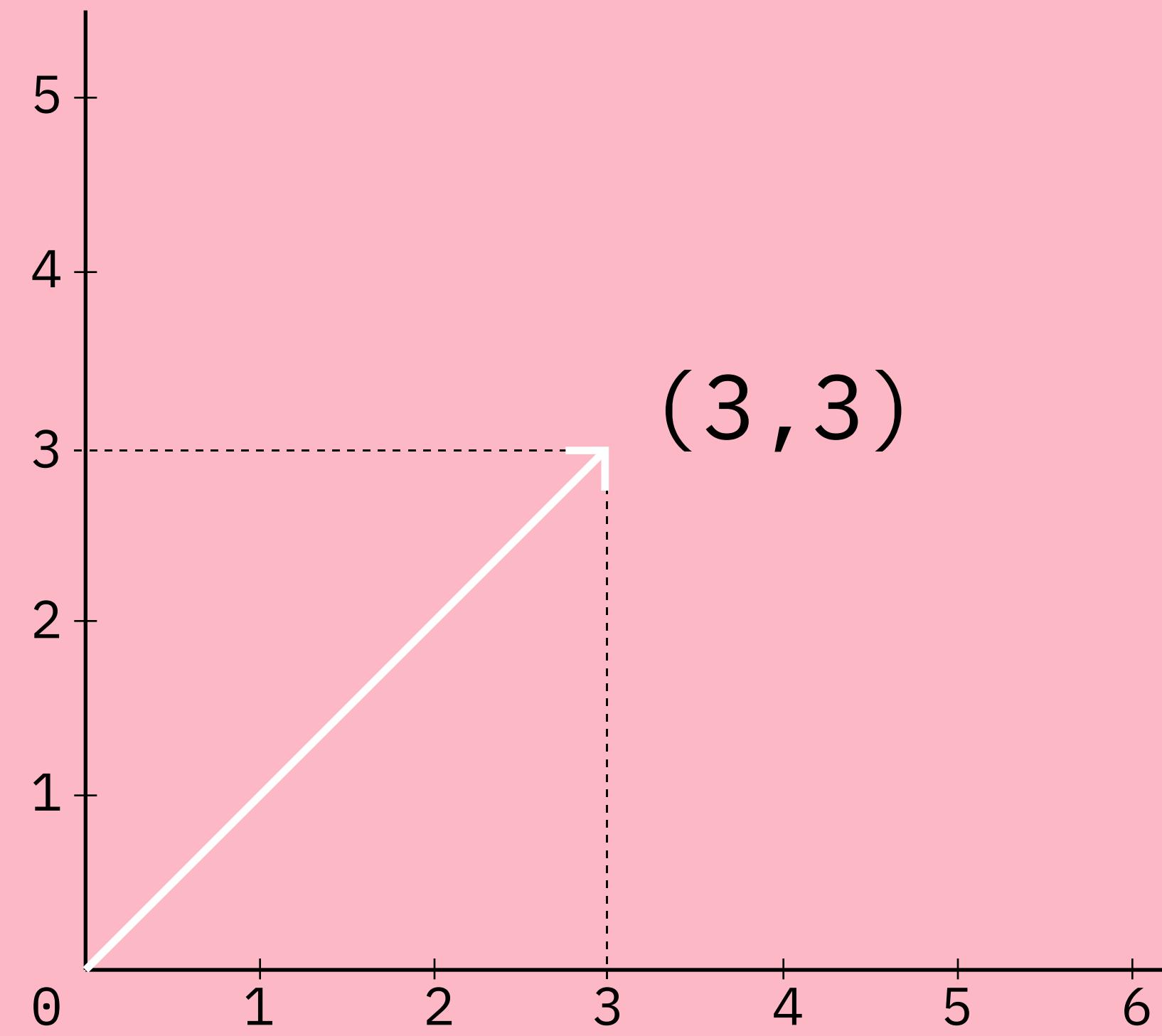
= displayWidth

Alto de la pantalla

= displayHeight

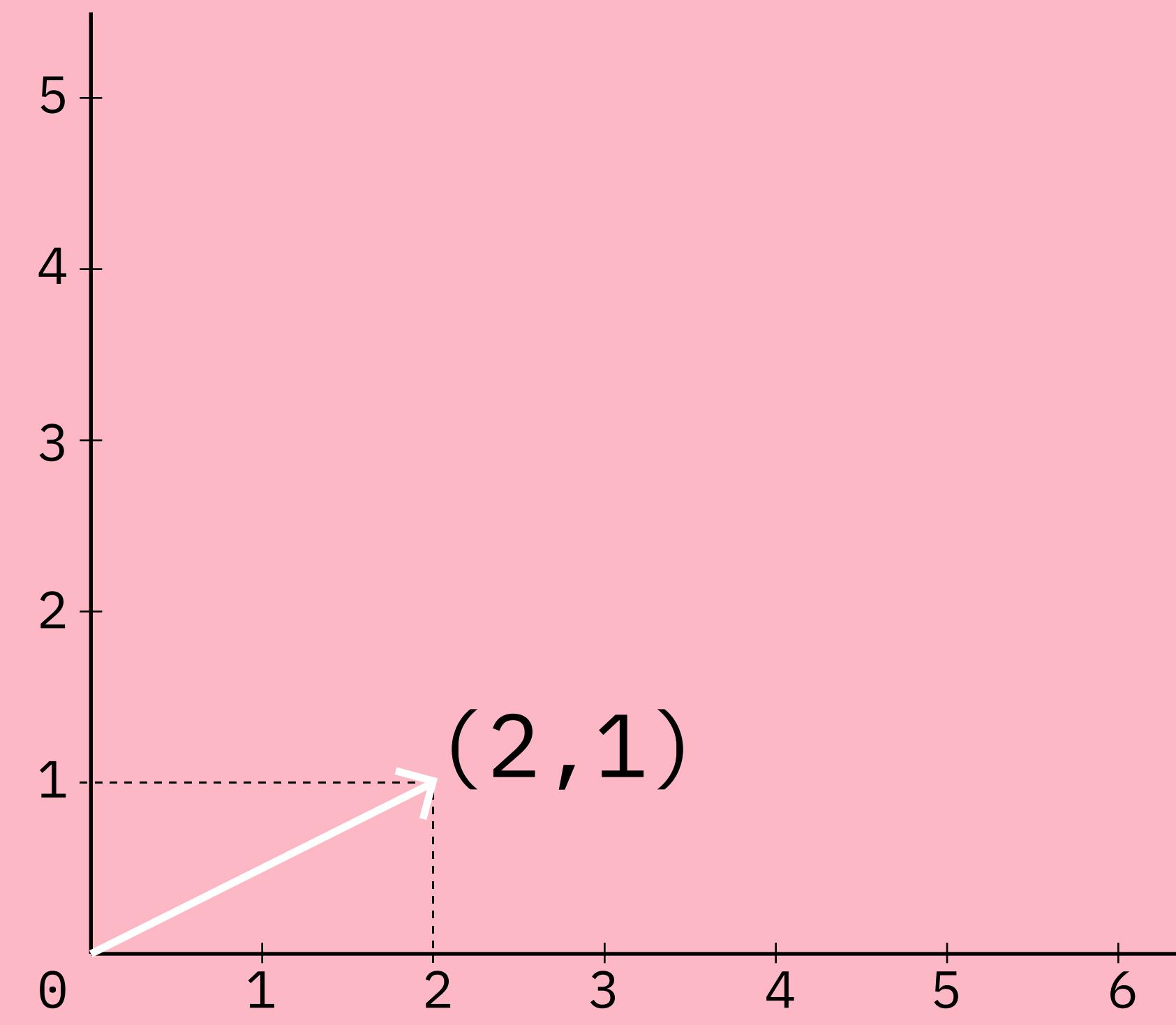
# Vectores

Objetos que tienen características (posición, tamaño, color)



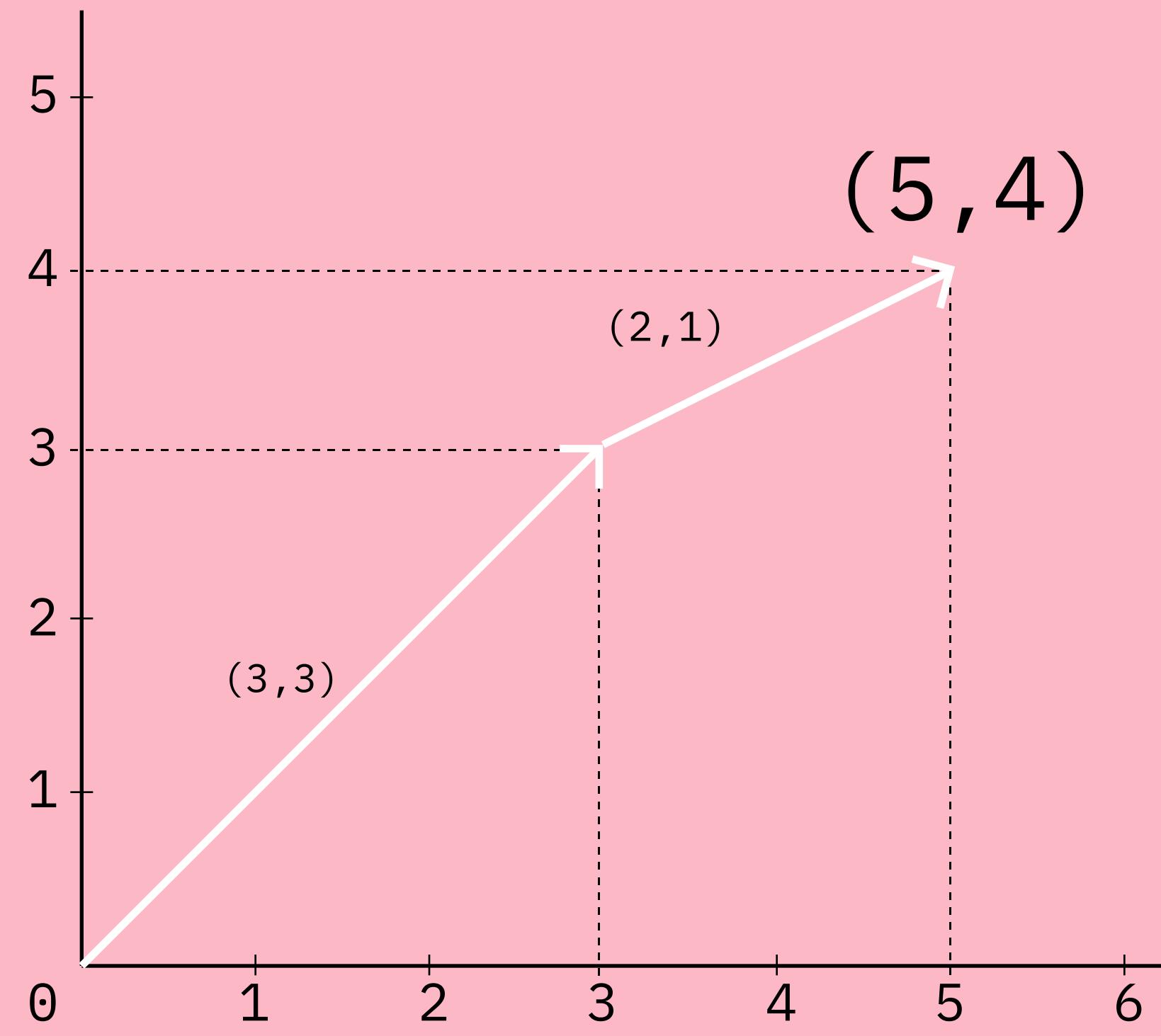
```
ubicacion = new PVector(3,3);
```

# Vectores



```
velocidad = new PVector(2,1);
```

# Vectores



```
ubicacion = new PVector(3,3);  
velocidad = new PVector(2,1);
```

```
ubicacion.add(velocidad);
```

# Posición del mouse

```
import java.awt.MouseInfo;  
import java.awt.Point;  
import java.awt.PointerInfo;
```

```
PointerInfo info = MouseInfo.getPointerInfo();  
Point p = info.getLocation();
```

# Easing o suavizado

Se calcula la diferencia entre la posición de la ventana y el cursor. Si la distancia es superior a 1 pixel, la ventana se desplaza una parte de la distancia (0,01) desde su posición actual hacia el cursor.

```
float easing = 0.01;  
float mX = p.x;  
float dx = mX - x;  
x += dx * easing;
```

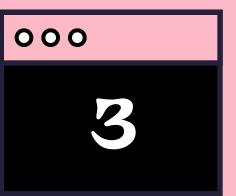
# Frames

- Contenedor de una ventana
- Crear varios y definir su comportamiento
- Leer y manipular la interacción con el usuario

# Frames

```
import java.awt.Frame;  
import java.awt.Color;  
  
public class Ventana extends Frame {  
    Color c;  
  
    Ventana() {  
        c = new Color(0,255,0);  
        setVisible(true);  
        setBounds(100, 100, 100, 100);  
        setBackground(c);  
    }  
}
```

```
Ventana ventana  
  
void setup () {  
    ventana = new Ventana();  
}
```



# Matriz

(0,0)	(1,0)	(2,0)	(3,0)
(0,1)	(1,1)	(2,1)	(3,1)
(0,2)	(1,2)	(2,2)	(3,2)
(0,3)	(1,3)	(2,3)	(3,3)

```
Ventana[][] ventanas;  
  
void setup () {  
    ventanas = new Ventana[4][4];  
}
```

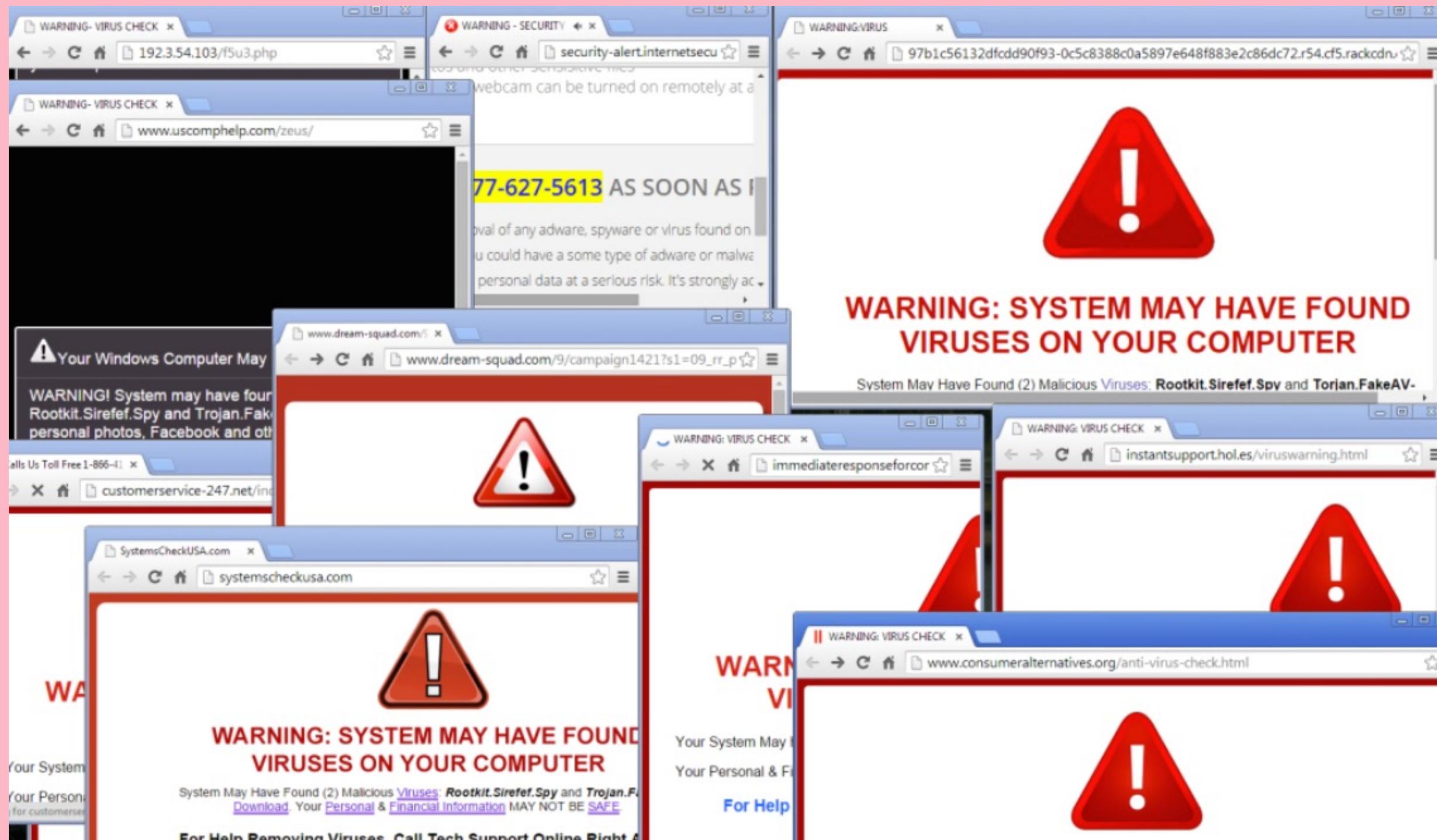
# Matriz

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

```
for(int i = 0; i < numRows; i++) {  
    for(int j = 0; j < numCols; j++) {  
        Vx = j*Vw;  
        Vy = I*Vh;  
        ventanas[j][i] = new Ventana(Vx,Vy,Vw,Vh);  
    }  
}
```

# Spam!

¿Cómo leer la interacción de un usuario con una ventana?



# Spam!

**¿Cómo leer la interacción de un usuario con una ventana?**

```
import java.awt.event.WindowEvent;
import java.awt.event.WindowAdapter;

this.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent we) {
        //acciones
    }
});
```

# Papplets

¿Cómo crear múltiples aplicaciones de processing?

- Todas las aplicaciones de Processing son Papplets
- Definir el comportamiento de la ventana y su contenido

# Papplets

```
public class Ventana extends PApplet {  
  
    public Ventana() {  
    }  
  
    public void settings() {  
        size(128, 128);  
    }  
  
    public void setup() {  
        background(0);  
    }  
}
```

```
Ventana v;  
  
void setup() {  
    String[] args = {"ventana"};  
    v = new Ventana();  
    PApplet.runSketch(args, v);  
}
```

**¡Gracias!**