```
//Expresión Digital II
//s06c01IsidoraGarin
//Composición gráfica compleja que integre 1 for loop, 2 if, 2 interacciones


//Variables de posición, velocidad y dirección ejes x e y cuadrado
float posx, posy, velx, vely, dirx, diry;
//Variables stroke, cambio dirección rectángulos y random líneas fondo
float sw, m, n, g;
//Variables para colores
color bla, neg, ver, tur, bla2, neg2, ver2, tur2, ver3, rver, razul, rrojo;
//Variables para color de distintos elementos
color colorCuadrado, colorStroke;
//Variables de tamaño del cuadrado
float sizeC;
//Boolean cuadrados
boolean t, l;
//Variables para líneas
float y = 100;
float x = 100;


void setup() {
  size(1280, 800);

  //Inicialización de posición, velocidad y dirección eje x e y
  posx = width/3;
  posy = width/3;
  velx = 3;
  vely = 3;
  dirx = 1;
  diry = 1;

  //Inicialización de strokeWeight y dirección rectángulos
  sw = 10;
  m = 4;
  n = 20;
  g = 5;

  //Colores
  bla = color(255, 255, 255); //blanco
  neg = color(0, 0, 0); //negro
  tur = color(27, 204, 167); //turquesa
  ver = color(128, 255, 130); //verde
  bla2 = color(255, 255, 255, 125); //blanco transparencia
  neg2 = color(0, 0, 0, 125); //negro transparencia
  tur2 = color(27, 204, 167, 125); //turquesa transparencia
  ver2 = color(128, 255, 130, 125); //verde transparencia
```

```
  ver3 = color(128, 255, 130, 0); //verde totalmente transparente
  rver = color(0, random(0, 255), 0, 125); //random verde transparencia
  razul = color(0, 0, random(0, 255), 125); //random azul transparencia
  rrojo = color(random(0, 255), 0, 0, 125); //random rojo transparencia


  //Asignación de color a colorCuadrado y colorStroke
  colorCuadrado = bla;
  colorStroke = tur;

  //Asignación de tamaño de cuadrado a sizeC
  sizeC = 60;

  //Inicialización para valores de t y l
  t = true;
  l = true;

  //Inicialización variables x e y
  y = height * 0.5;
  x = width * 0.5;
}


void draw() {
  background(bla);

  //Cuadrado que rebota por el canvas
  fill(colorCuadrado);
  strokeWeight(sw);
  stroke(colorStroke);
  rect(posx, posy, sizeC, sizeC);

  //Velocidad
  posx = posx + velx * dirx;
  posy = posy + vely * diry;

  //Si posx mayor al ancho, cambia dirección, fill turquesa, stroke verde de
  if (posx > width - (80 + sw)) {
    dirx *= -1;
    background(bla);
    colorStroke = ver;
    strokeWeight(sw);
    colorCuadrado = tur;
    if (m == 4) {
      m = 20;
      n = 4;
    } else if (m == 20) {
      m = 4;
```

```
    n = 20;
  }
}

//Si posx menor a 0, cambia dirección, fill verde, stroke blanco de 4
if (posx < 0) {
  dirx *= -1;
  background (bla);
  colorStroke = neg;
  strokeWeight (sw);
  colorCuadrado = ver;
  if (m == 4) {
    m = 20;
    n = 4;
  } else if (m == 20) {
    m = 4;
    n = 20;
  }
}

//Si posy mayor al ancho, cambio dirección, fill blanco, stroke turquesa de
if (posy > height - (80 + sw)) {
  diry *= -1;
  background (bla);
  colorStroke = tur;
  strokeWeight (sw);
  colorCuadrado = bla;
  if (m == 4) {
    m = 20;
    n = 4;
  } else if (m == 20) {
    m = 4;
    n = 20;
  }
}

//Si posy menor a 0, cambia dirección, fill blanco, stroke turquesa de 4
if (posy < 0) {
  diry *= -1;
  background (bla);
  colorStroke = tur;
  strokeWeight (sw);
  colorCuadrado = bla;
  if (m == 4) {
    m = 20;
    n = 4;
  } else if (m == 20) {
    m = 4;
```

```
    n = 20;
  }
}

//Círculos chicos transparencia
if (t == true) {
  fill(random(0, 255), random(0, 255), random(0, 255), 125);
  noStroke();
  ellipse(mouseX, mouseY, mouseX/4, mouseX/4);
}
if (t == false) {
  fill(ver3);
  noStroke();
  ellipse(mouseX, mouseY, mouseX/4, mouseX/4);
}

//Líneas perpendiculares con estela
stroke(neg);
line(0, y, width, y);
line(x, 0, x, height);

y = y - 1;
if (y < 0) {
  y = height;
    }

x = x - 1;
if (x < 0) {
  x = width;
}

//Patrón de rectángulos que cambian de dirección
for (int i = 0; i <= width; i += 25) {
  for (int j = 0; j <= height; j += 25) {
    noStroke();
    fill(0, random(0, 255), 0, 200);
    rect(i, j, m, n);
  }
}

//Líneas fondo control teclado
if (l == true) {
  for(int i=0; i <= height*2; i+=10){
    noStroke();
    fill(razul);
    rect(i, i, g, i);
    loop();
  }
```

```
  }
  if (l == false) {
    for(int i=0; i <= height*2; i+=10){
      noStroke();
      fill(razul);
      loop();
    }
  }
}


void mousePressed() {
  //Verdadero o falso para los círculos
  t = !t;

  //Cambio de tamaño del cuadrado que rebota por el canvas con click derecho
  if (mouseButton == LEFT) {
    sizeC = 100;
  } else if (mouseButton == RIGHT) {
    sizeC = 30;
  } else {
    sizeC = 60;
  }
}


void keyPressed() {
  //Verdadero o falso para mostrar o esconder líneas de fondo
  if (key == ' ') {
     l = !l;
  }

  //Tecla g para el grosor de las líneas de fondo
  if (key == 'g'){
   g = random(1, 20);
  }
}
```