

Tutoriales
Diseño en
Medios
Interactivos

05

Gestos e interacción (móviles)

- a.*** Configuración
- b.*** Rotación
- c.*** Toques
- d.*** Movimiento
- e.*** Tiempo de toque
- f.*** Aceleración

Los **dispositivos móviles** generan un gran campo de posibilidad en términos de interactividad con una persona. A las interacciones comunes que se pueden tener con una pantalla se agrega toda una dimensión de **interacción física** con el dispositivo en sí (tocar, girar, mover) .

Este tutorial explica 5 ejemplos sencillos con tipos de interacción en dispositivos móviles, con el fin de ser un punto de partida para explorar las posibilidades de **interacción física y digital** que ofrecen estos dispositivos.

El código de todos los ejemplos se puede ver y descargar en:
[**github.com/disenMediosInteractivos/tutoriales/tree/master/05_gestos_mobile/c%C3%B3digo**](https://github.com/disenMediosInteractivos/tutoriales/tree/master/05_gestos_mobile/c%C3%B3digo)

¿Cómo garantizar que un programa o sketch funcione
bien en un dispositivo móvil?

a.
Configuración

1. Generar un canvas de pantalla completa

```
function setup() {  
  //crea un canvas del tamaño de la ventana  
  createCanvas(windowWidth, windowHeight);  
}
```

Esto permite que sin importar el tamaño del dispositivo en el que se abra el sketch, este va a ser de **pantalla completa**.

2. Optimizar sketch para móvil.

Por lo general, dispositivos móviles tienen una **densidad de pixeles** distinta a la de pantallas de escritorio. Por esto, se debe tomar el tamaño de la pantalla del dispositivo como base para renderizar el programa.

Para esto, se agrega una simple línea de código al archivo **index.html** del sketch:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Esta línea de código hace que el navegador del dispositivo muestre una página web de manera que ocupe la **mayor cantidad de espacio en la pantalla**.

¿Cómo utilizar la rotación de un dispositivo móvil
como elemento de interacción?

Se puede ver el código de este ejemplo en:

alpha.editor.p5js.org/laurajunco/sketches/rkbRR5DyG

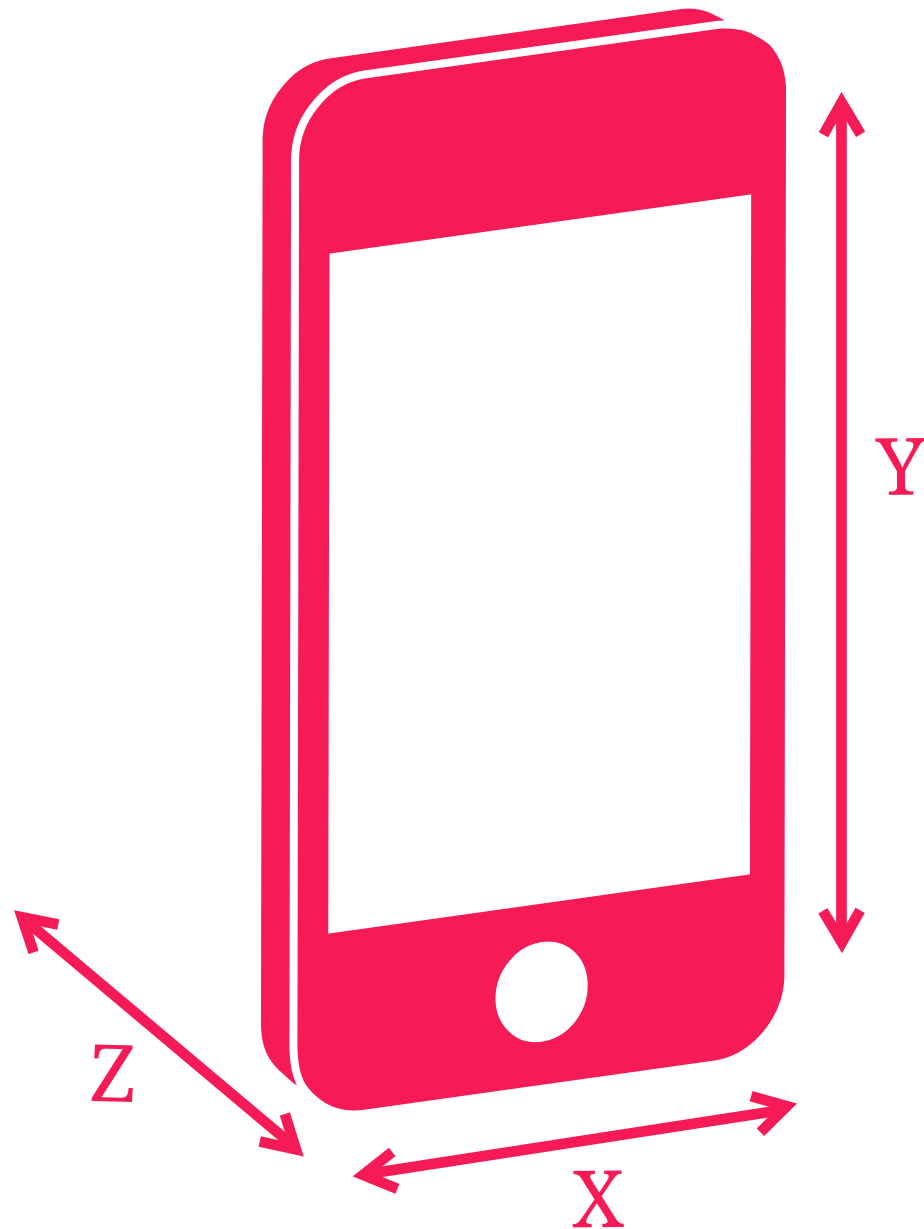
Se puede probar en:

alpha.editor.p5js.org/full/rkbRR5DyG

b.

Rotación

¿Como se detecta la rotación?



La mayoría de dispositivos móviles actuales, tienen un elemento llamado **acelerómetro**. Más en: youtube.com/watch?v=KZVgKu6v808

Un acelerómetro es un dispositivo electrónico capaz de medir la **rotación y aceleración** de la pantalla en 3 ejes: X, Y y Z.

En p5.js se pueden encontrar estos datos como 3 **variables**:

RotationX: toma valores entre -90 y 90

RotationY: toma valores entre -90 y 90

RotationZ: toma valores entre 0 y 359

No en todos los dispositivos móviles se puede acceder a los datos de rotación. Para **comprobar** si es posible en su dispositivo abra este ejemplo: p5js.org/examples/mobile-acceleration-color.html (Si la pantalla cambia de color al agitar el dispositivo significa que funciona)

Ya que la rotación del dispositivo es guardada en variables de p5.js es muy fácil acceder a ella y utilizarla como cualquier otro valor. En este caso, la inclinación del dispositivo se utilizará para mover un objeto en la pantalla.

1. Imprimir rotación en el canvas

```
//imprime los valores de rotacion en X y en Y
fill(255);
noStroke();
text("X: " + floor(rotationY), 20, 40);
text("Y: " + floor(rotationX), 20, 20);
```

En este caso solo se utilizará la rotación en X y en Y del dispositivo.

2. Utilizar esas variables para determinar la velocidad de algún objeto

```
//rotationX controla Y
this.vely = map(rotationX, -90, 90, -3, 3);

//rotationY controla X
this.velx = map(rotationY, -90, 50, -3, 3);
```

Con la función **map()** se traducen los grados de rotación del dispositivos a valores entre -3 y 3 para la velocidad del objeto.

¿Cómo aprovechar la cantidad de toques simultáneos
que permite un dispositivo móvil para generar nuevas
formas de interacción?

Se puede ver el código de este ejemplo en:

alpha.editor.p5js.org/laurajunco/sketches/Hki-lpvkM

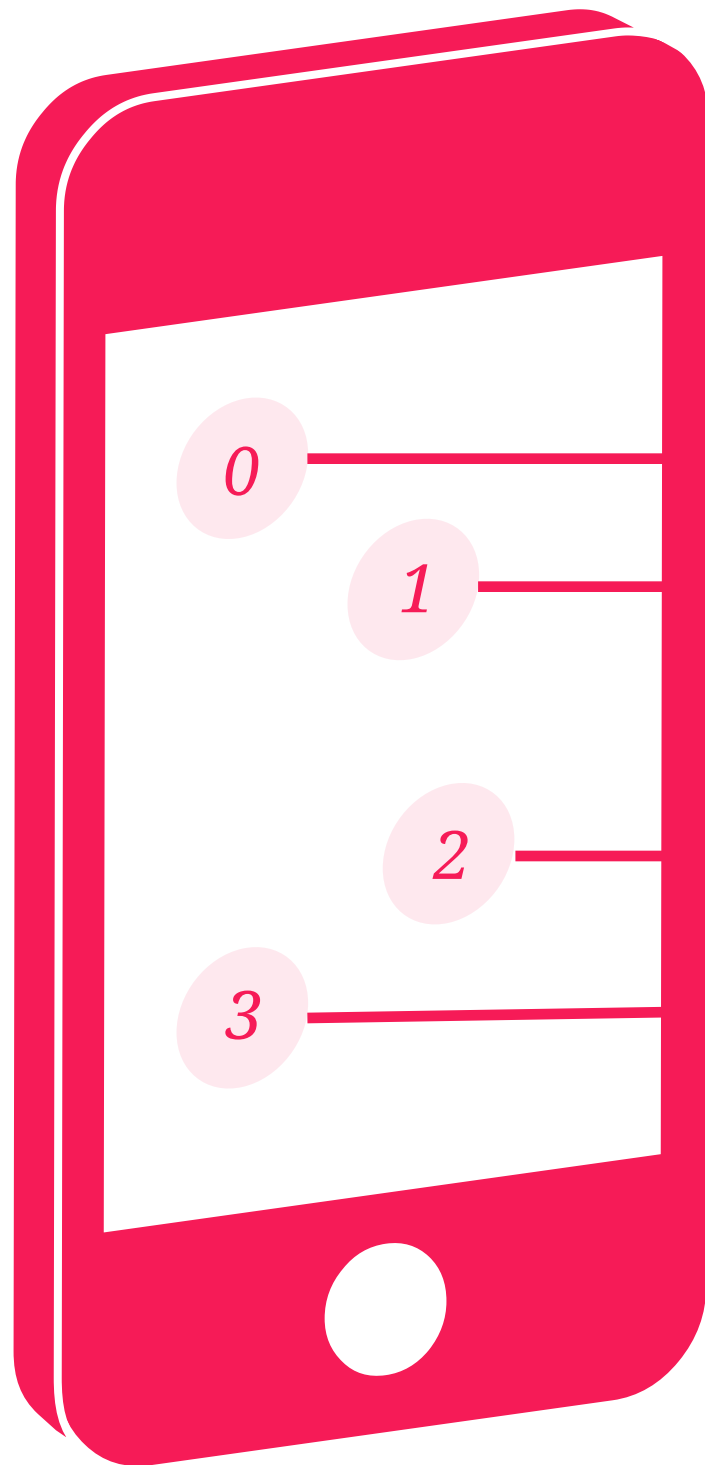
Se puede probar en:

alpha.editor.p5js.org/full/Hki-lpvkM

C.

Toques

Touches



Dentro de p5.js existe una variable llamada **touches[]**. p5js.org/reference/#/p5/touches

Touches[] es una lista que guarda todos los puntos de contacto con la pantalla y sus coordenadas.

En este caso de ejemplo la variable touches[] tiene 4 elementos:

touches[0]

touches[1]

touches[2]

touches[3]

La forma de acceder a las coordenadas X y Y de un toque es: **touches[i].x** y **touches[i].y**

Para este ejemplo se realizará una herramienta de dibujo de figuras a partir de todos los puntos de contacto que haya con la pantalla en un momento.

1. Crear una figura con todos los toques de la pantalla

```
//inicia una figura
beginShape();

//recorre todos los puntos en los que se esta tocando la pantalla
for (var i = 0; i < touches.length; i++) {

    //en las coordenadas de cada punto crea un vertice
    vertex(touches[i].x, touches[i].y);
}

//cierra la figura
endShape(CLOSE);
```

Después de iniciar una figura, se recorren todos los puntos de contacto con la pantalla y se genera un vértice en las coordenadas de cada uno de ellos. Finalmente se cierra la figura.

2. Revisar si se oprimió algún botón

```
//recorre todos los puntos en lso que se esta tocando la  
pantalla y revisa su distancia con respecto al boton
```

```
for (var j = 0; j < touches.length; j++) {
```

Recorre el arreglo

```
if (dist(touches[j].x, touches[j].y, botx, boty) < 20) {
```

Revisa la distancia de cada
toque con respecto a un botón

```
    //si el boton es presionado borra la pantalla  
    background(0);
```

```
}
```

```
}
```

¿Cómo saber si un toque ha sido movido y como
aprovechar esta acción?

Se puede ver el código de este ejemplo en:

alpha.editor.p5js.org/laurajunco/sketches/BJ2ziavyz

Se puede probar en:

alpha.editor.p5js.org/full/BJ2ziavyz

d.

Movimiento

En este ejemplo se pueden eliminar bolas con el movimiento de un toque en la pantalla

1. Reconocer que un toque ha sido movido

```
//esta funcion se activa cuando el dispositivo  
reconoce que se movio un toque en la pantalla  
  
function touchMoved() {  
  
}
```

Cada vez que un toque de la pantalla se mueve, el código dentro de esta función empieza a correr.

2. Verificar la distancia del toque con un objeto

```
//recorre toda la lista de bolas  
for (var i = 0; i < numBolas; i++) {  
  
//revisa si se estaba tocando alguna bola y en tal caso la mata  
    if (dist(touches[0].x, touches[0].y, bolas[i].x, bolas[i].y) < 15) {  
        bolas[i].matar();  
    }  
}
```

Se usa **touches[0]** porque se asume que solo hay un toque a la vez

¿Cómo saber el tiempo que se mantiene presionada la pantalla y usar esto como elemento de interacción?

Se puede ver el código de este ejemplo en:

alpha.editor.p5js.org/laurajunco/sketches/S1TmXKDyG

Se puede probar en:

alpha.editor.p5js.org/full/S1TmXKDyG

e.

Tiempo

1. Saber si se empezó a tocar la pantalla

```
//esta funcion se activa cuando la pantalla es tocada:  
cambia contar a verdadero  
function touchStarted()  
{  
  
}
```

2. Crear variables para contar el tiempo de presión y para saber si se está contando o no.

```
//contador de tiempo presionado  
var contador = 0;  
  
//variable para saber si esta contando o no  
var contar = false;
```


3. Empezar a contar si se toca la pantalla

```
function touchStarted()  
{  
  contar = true;  
}
```

4. Aumentar el contador

```
//contar: la cuenta aumenta si contar es true y si el numero de  
//frame es multiplo de 10  
if (frameCount % 10 != 0 && contar) {  
  
  //aumenta el valor del contador  
  contador++;  
}
```

La cuenta sólo aumenta cada 10 frames y si la variable contar es verdadera.

5. Crear un nuevo objeto al dejar de oprimir la pantalla.

```
function touchEnded()  
{  
    // se crea una nueva bola que recibe por parametro  
    // las velocidades en x y y del mouse  
  
    bolas[numBolas] = new bola(contador);  
  
    //aumenta el numero de bolas de la lista  
    numBolas++;  
  
    //contar vuelve a ser falso y se reinicia el contador  
    contar = false;  
    contador = 0;  
}
```

La función **touchEnded ()** se activa cuando algún toque ha dejado de oprimir la pantalla.

¿Cómo saber la velocidad con la que se esta moviendo
un dispositivo móvil?

Se puede ver el código de este ejemplo en:

alpha.editor.p5js.org/laurajunco/sketches/rJD8F0vy

Se puede probar en:

alpha.editor.p5js.org/full/rJD8F0vyz



Aceleración

Para este ejemplo se aumentará la aceleración de un objeto de acuerdo a la velocidad con la que se agite un dispositivo móvil.

1. Imprimir aceleración en el canvas

```
//imprime los valores de aceleracion en X y en Y  
text("AccX: " + accelerationX 30, 30);  
text("AccY: " + accelerationY 30, 50);
```

Al igual que la rotación, los valores de aceleración en X, Y y Z son guardados en **variables** de p5.js

2. Crear variables de aceleración en X y Y para un objeto

```
//variables de aceleracion  
this.accx = 0;  
this.accy = 0;
```

```
//la velocidad se asigna por: velocidad actual + aceleracion  
this.velx = this.velx + this.accx;  
this.vely = this.vely + this.accy;
```

La aceleración se define como el **cambio** en la velocidad de un cuerpo. De esta manera, debe ser sumada a la variable de velocidad de un objeto.

3. Limitar los valores de velocidad y aceleración

```
//se limitan las variables de velocidad hasta un valor maximo de 30
this.velx = constrain(this.velx, 0, 30);
this.vely = constrain(this.vely, 0, 30);

//se limitan las variables de velocidad hasta un valor maximo de 2
this.accx = constrain(this.accx, 0, 2);
this.accy = constrain(this.accy, 0, 2);
```

Se definen límites en la velocidad y la aceleración del objeto para tener mayor control sobre el movimiento.

4. Crear una función para aumentar la aceleración del objeto

```
this.acelerar = function() {
  //se toman los valores en X y en Y de aceleracion el
  dispositivo y se le suman a la aceleracion de la bola

  this.accx += abs(accelerationX);
  this.accy += abs(accelerationY);
}
```

Se suma la aceleración del dispositivo a la aceleración del objeto

Se utiliza **abs()** para que la aceleración siempre sea un valor positivo.

5. Si el dispositivo es movido se llama a la función acelerar del objeto

```
//cuando el dispositivo detecta movimiento llama a la funcion acelerar de la bola  
  
function deviceMoved() {  
    b.acelerar();  
}
```