

CREANDO APPS

aprende a programar
aplicaciones móviles

PARTE 3

CÁMARA

La cámara es algo propio de los dispositivos móviles que no tenemos en desktop. Tener esta característica en nuestro dispositivo móvil nos permite desarrollar aplicaciones que conectan con el mundo que nos rodea.

3.1 Cámara /

TOMAR UNA FOTO

Partimos de la plantilla de la aplicación de gestos, que tiene preparado un esqueleto de la aplicación en HTML / CSS, y las librerías básicas cargadas: zepto, fastclick y hammer.

Vamos a añadir el plugin de la cámara para cordova¹.

De esta forma, podemos acceder al método `getPicture` que nos permite hacer una foto. Tenemos que pasarle 3 parámetros:

- Una función que se ejecuta si tomamos la fotografía con éxito
- Una función que se ejecuta si hay un error
- Un diccionario con opciones

1

```
var app = {  
  ...  
  tomarFoto: function() {  
    var opciones = { ... };  
    navigator.camera.getPicture(  
      app.fotoCargada,  
      app.errorAlCargarFoto,  
      opciones  
    );  
  },  
  fotoTomada: function(imageURI){ ... },  
  errorAlTomarFoto: function(message){ ... }  
};
```

camera.js

(1) Plugin de Cordova para la cámara
<http://cordova.apache.org/docs/en/latest/reference/cordova-plugin-camera/index.html#camera>

3.1 Cámara / TOMAR UNA FOTO

A la función de éxito se le pasa la foto (los datos en bruto o una dirección al sistema de ficheros) para poder pintarlo en pantalla. En el cuerpo, accedemos al elemento imagen de nuestra web (en el archivo.html) y le asociamos como fuente (src) la foto.

```
fotoTomada: function(imageURI){
    var image = document.querySelector('#foto');
    image.src = imageURI;
},
```

A la función de error se le pasa un diccionario con información del error que ha sucedido. También se considera error, por ejemplo, si el usuario elige hacer una fotografía pero luego cancela la opción.

```
errorAlTomarFoto: function(message) {
    console.log('Fallo al tomar foto o toma
cancelada: ' + message);
}
```

El diccionario de opciones nos ayuda a describir algunas características sobre la captura de la foto. Las más relevantes para nosotros son:

- Indicamos la **calidad** de la foto 0 - 100: más calidad, mayor tamaño
- Podemos definir **dimensiones** (ancho y alto) de la foto, ya que al tamaño original es demasiado grande para manejarla en nuestro programa
- El **destino** de la foto sirve para indicar si queremos que a nuestra función de éxito se le pasen los datos en bruto de la foto, o un enlace a la foto en el disco
- Hay un parámetro booleano que nos ayuda a corregir la **orientación** de la foto porque depende de cómo esté montado el hardware en nuestro dispositivo
- Otro parámetro booleano nos sirve para indicar si queremos que la foto tomada se guarde en la **galería** del dispositivo

3.1 Cámara / TOMAR UNA FOTO

Creamos un objeto imagen en nuestra página sin fuente.

Accedemos al elemento imagen y enlazamos el contenido de la foto desde la función de éxito para que se pinte en la app.

Opciones para tomar una fotografía

<http://cordova.apache.org/docs/en/latest/reference/cordova-plugin-camera/index.html#cameracameraoptions--object>

3.2 Cámara / USANDO EL CANVAS

Las imágenes de una pantalla (de un móvil, computador, televisión) se dibujan usando pequeños puntos llamados píxeles. Un computador o terminal móvil maneja la información de la imagen como la información de cada píxel de la pantalla. Una forma habitual es manejar la información de color de cada píxel usando la escala RGB (del inglés Red-Green-Blue), representada con un valor de 0 a 255 (puede parecer un valor un poco raro, pero es muy normal a sabiendas de cómo manejan información los computadores). Con una combinación de valores de rojo, verde y azul podemos representar cualquier color con mucha precisión. Además de estos 3 valores, para cada píxel almacenamos un cuarto valor que indica la transparencia, de forma que 0 es totalmente transparente y 1 es el color sin ninguna transparencia.

Para procesar la imagen que hemos tomado con la cámara, podemos usar un elemento especial: el canvas. Es un lienzo donde podemos pintar cualquier cosa, y se utiliza por ejemplo para desarrollar videojuegos. Pero, en este caso, lo vamos a usar para pintar la imagen que hemos tomado con la cámara, y poder acceder a la información de cada píxel de la imagen.

Sobre el (contexto 2D del) canvas dibujamos la imagen usando el método `drawImage`.

- Canvas** del inglés lienzo, usado para referirnos a un componente web que nos permite pintar sobre él como un lienzo
- Pixel** hace referencia a un punto de la pantalla en un dispositivo que tiene un color y que junto con otros muchos puntos pinta en pantalla
- RGBA (Red-Green-Blue-Alpha)** sistema para definir un color basado en un % de rojo, % de verde y % de azul, además de una componente alpha que indica la transparencia
- Context** del inglés contexto, usado para referirnos al contexto sobre el que manejamos el canvas, que puede ser 2D (2 dimensiones)

3.3 Cámara / FILTROS

Ahora vamos a añadir los botones necesarios en el HTML/CSS para realizar las funciones: un botón por cada filtro. Añadimos los escuchadores.

Una vez dibujado recuperamos la información de cada píxel de la imagen usando sobre el canvas con `getImageData`.

Ahora vamos a realizar el procesamiento de la imagen. Para ello, hemos creado una librería que permite modificar la información de los píxeles de la imagen para convertirla en:

- Blanco y negro
- Sepia
- Negativo

Después, volveremos a poner la imagen modificada sobre el canvas con `putImageData`.

Filtro procesamiento que hacemos sobre una fotografía y que cambia sus componentes de color, por ejemplo, pasar una fotografía a blanco y negro

Image imagen

Data datos

3.4 Cámara /

CARGAR UNA FOTO DE LA GALERÍA

Modificamos nuestra maqueta para añadir un botón que permita cargar una foto de la galería.

Volvemos a llamar a `getPicture` pero con una opción especial en el diccionario de opciones para que como fuente de la imagen use la galería (`PHOTOLIBRARY`) en lugar de la cámara.

Los filtros siguen funcionando :)

(3) Repositorio de la app <https://github.com/h4ckademy/mooc-mobile-camera>

(4) Plugin de Cordova para la cámara <http://cordova.apache.org/docs/en/latest/reference/cordova-plugin-camera/index.html#camera>

(5) Opciones para tomar una fotografía <http://cordova.apache.org/docs/en/latest/reference/cordova-plugin-camera/index.html#cameracameraoptions--object>

Picture del inglés fotografía

Height del inglés altura

Width del inglés ancho

Header cabecera

Footer pie de página

Photo Library galería de fotos



Desarrollado por Telefónica Educación Digital. Todos los derechos reservados.