

Guía definitiva: cómo crear una app de cámara para iPhone con HTML, JavaScript y Capacitor (desde cero)

11:11



Ninguna foto capturada todavía

HACER FOTO

Aprender a usar los sensores de hardware del iPhone utilizando únicamente tecnologías web (HTML, CSS y JavaScript) es increíblemente sencillo gracias a **Ionic Capacitor**.

En este tutorial aprenderás, **paso a paso y sin omitir ningún detalle**, cómo crear una aplicación desde la terminal, configurar los estrictos permisos de privacidad de Apple y programar un botón táctil que active la cámara real del iPhone para mostrar la foto en pantalla.

Requisitos previos (Preparando la Mac)

Antes de empezar, necesitas dos herramientas clave instaladas en tu Mac:

1. **Node.js**: El motor que ejecutará los comandos. Puedes descargarlo e instalarlo desde su web oficial o usando Homebrew en la terminal con `brew install node`.
2. **Xcode**: El entorno oficial de Apple para compilar Apps de iOS. Descárgalo gratis desde la App Store de tu Mac.

Paso 1: Crear el directorio y la estructura del proyecto

Abre la aplicación **Terminal** en tu Mac y ejecuta los siguientes comandos uno por uno para crear la base de tu proyecto:

```
# 1. Crear una carpeta vacía para el proyecto
mkdir appcamara
```

```
# 2. Entrar a la carpeta que acabamos de crear
cd appcamara
```

```
# 3. Inicializar el proyecto Node.js (crea el archivo
package.json)
```

```
npm init -y
```

```
# 4. Instalar las dependencias principales de Capacitor  
npm install @capacitor/core @capacitor/cli
```

Paso 2: Inicializar la configuración de Capacitor

Ahora configuraremos la identidad de tu aplicación. Ejecuta el siguiente comando en la terminal:

```
npx cap init "MiCamaraApp" "com.tuusuario.camara" --web-dir=www
```

- "MiCamaraApp": Es el nombre que verá el usuario debajo del icono en su iPhone.
- "com.tuusuario.camara": Es el identificador único de la App (puedes cambiar «tuusuario» por tu nombre).
- --web-dir=www: Le dice a Capacitor que tu código web estará dentro de una carpeta llamada www.

Paso 3: Instalar el Plugin Nativo de Cámara

Para que JavaScript pueda saltar el navegador e invocar el lente de la cámara de Apple, necesitamos su plugin oficial. Ejecuta en la terminal:

```
npm install @capacitor/camera
```

Paso 4: Crear la interfaz Web (HTML5, CSS3 y JavaScript)

Seguimos en la terminal para crear la estructura de carpetas donde irá el diseño de la aplicación:

1. Crear la carpeta 'www' que le prometimos a Capacitor
mkdir www

Ahora, abre tu editor de código favorito (como VS Code) o utiliza un editor de texto, crea un archivo llamado **index.html** dentro de esa carpeta www y pega el siguiente código completo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Cámara Nativa iPhone</title>
  <style>
    body {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      min-height: 100vh;
      margin: 0;
      background-color: #000000; /* Fondo negro estilo
cámara profesional */
      color: #ffffff;
    }
    #container {
      text-align: center;
      width: 100%;
    }
    #photo-preview {
      width: 85%;
      max-width: 380px;
      height: 380px;
      border: 2px dashed #333333;
      margin-bottom: 30px;
      border-radius: 24px;
      object-fit: cover;
      display: none; /* Oculto hasta que se tome la foto
*/
    }
    #placeholder {
```

```

        height: 380px;
        display: flex;
        align-items: center;
        justify-content: center;
        color: #8e8e93;
        font-size: 16px;
    }
    .btn-capture {
        background-color: #ffffff;
        color: #000000;
        border: none;
        padding: 18px 45px;
        font-size: 18px;
        font-weight: bold;
        border-radius: 50px;
        cursor: pointer;
        box-shadow: 0 4px 20px rgba(255,255,255,0.25);
        transition: transform 0.1s ease;
    }
    .btn-capture:active {
        transform: scale(0.95);
        background-color: #e5e5ea;
    }
</style>
</head>
<body>

    <div id="container">
        <img id="photo-preview" src="" alt="Vista previa">
        <div id="placeholder">
            <p>Ninguna foto capturada todavía</p>
        </div>

        <br>

        <button class="btn-capture"
onclick="abrirCamara()">HACER FOTO</button>
    </div>

    <script>
        async function abrirCamara() {
            try {

```

```

        // Accedemos de forma global al plugin
inyectado por Capacitor
        if (window.Capacitor &&
window.Capacitor.Plugins && window.Capacitor.Plugins.Camera) {
            const Camera =
window.Capacitor.Plugins.Camera;

            // Configuración de la captura nativa
            const image = await Camera.getPhoto({
                quality: 90,           // Calidad de
la imagen (1-100)
                allowEditing: true,    // Permite al
usuario recortar/editar la foto tras tomarla
                resultType: 'uri',     // Devuelve una
ruta local optimizada
                source: 'CAMERA'       // Fuerza a
abrir la cámara física (puedes usar 'PHOTOS' para galería)
            });

            // Renderizar la foto en nuestro HTML
            const imageElement =
document.getElementById('photo-preview');
            const placeholderElement =
document.getElementById('placeholder');

            imageElement.src = image.webPath; // Ruta
compatible con el visor web
            imageElement.style.display = 'inline-
block';
            placeholderElement.style.display = 'none';

        } else {
            alert("Capacitor no se ha detectado.
Ejecuta la App en tu iPhone real.");
        }
    } catch (error) {
        console.log("El usuario cerró la cámara o hubo
un error:", error);
    }
}
</script>

```



```
</body>  
</html>
```

Paso 5: Añadir la plataforma iOS y sincronizar

De vuelta en la terminal, vincularemos nuestro proyecto web con el entorno de desarrollo nativo de Apple:

```
# 1. Instalar el soporte de Capacitor para iOS  
npm install @capacitor/ios
```

```
# 2. Agregar el proyecto nativo de iOS  
npx cap add ios
```

```
# 3. Sincronizar nuestro código HTML e instalar los plugins  
dentro de la App de iOS  
npx cap sync ios
```

Paso 6: Configurar Permisos de Privacidad en Xcode (¡Obligatorio!)

Apple requiere estrictamente que expliques al usuario **para qué** vas a usar la cámara. Si omites este paso, la App se cerrará sola al pulsar el botón.

1. Abre tu proyecto en Xcode ejecutando en la terminal: `npx cap open ios`
2. En la barra lateral izquierda de Xcode, haz clic en la carpeta azul raíz llamada **App**.
3. En la lista de carpetas internas, busca y haz clic sobre el archivo llamado **Info** (o **Info.plist**).
4. Posa el ratón sobre cualquier fila existente, haz clic en el icono de **+** (Add Row) que aparece y añade las siguientes tres llaves con sus respectivos textos de explicación en la columna *Value*:

Key	Type	Value
Privacy – Camera Usage Description	String	Necesitamos acceder a la cámara para tomar fotos en la app.
Privacy – Photo Library Additions Usage Description	String	Necesitamos guardar las fotos tomadas en tu carrete.
Privacy – Photo Library Usage Description	String	Necesitamos acceso para poder mostrar las imágenes guardadas.

Paso 7: El truco de conexión para evitar el «Error 908»

Para que la transferencia de la App a tu teléfono no se bloquee ni falle:

1. Desconecta el cable USB de tu iPhone.
2. En tu iPhone, abre el Centro de Control y **activa el Modo Avión** (asegúrate de que el Wi-Fi y el Bluetooth queden 100% apagados).
3. Vuelve a conectar tu iPhone a la Mac usando el cable USB.

Paso 8: Compilar y desplegar en el iPhone

Regresa a la pantalla de **Xcode**:

1. **Firmar la aplicación:** En el panel central, ve a la pestaña **Signing & Capabilities**. En el apartado *Team*, despliega el menú y selecciona tu **Apple ID** (esto permite que la app corra en tu teléfono de forma gratuita).

2. **Seleccionar tu iPhone:** En la barra superior de Xcode (junto al botón de Play), haz clic en el selector de dispositivos y **elige tu iPhone real** en lugar de un simulador.
3. **Limpieza e inicio:** En el menú superior de Xcode ve a **Product -> Clean Build Folder** (para vaciar cachés antiguas) y finalmente haz clic en el botón **Play** (el triángulo en la esquina superior izquierda).

Paso 9: Autorizar el perfil de desarrollador en iOS

Cuando Xcode termine, verás el icono de la App en tu pantalla de inicio del iPhone, pero al pulsarla saltará un aviso de *«Desarrollador no confiable»*.

Para solucionarlo de forma permanente:

1. En tu iPhone ve a **Ajustes > General**.
2. Entra en **Administración de dispositivos y VPN**.
3. Selecciona tu cuenta de Apple ID y pulsa en **«Confiar en [Tu Correo]»**.

¡Resultado Final!

¡Eso es todo! Abre la aplicación en tu iPhone, pulsa el botón **HACER FOTO**, acepta el permiso de privacidad de la cámara y verás cómo se despliega la interfaz nativa de iOS. Al capturar la foto, se incrustará limpiamente en tu plantilla web. ¡Felicidades!