

Guía definitiva: cómo usar HTML, JavaScript y Xcode para ver el porcentaje de batería de tu iPhone

Si quieres aprender a transformar tus conocimientos de desarrollo web (HTML, CSS y JavaScript) en aplicaciones móviles reales para iPhone que interactúen con el hardware del dispositivo, **Ionic Capacitor** es la herramienta moderna más eficiente.

En esta guía aprenderás paso a paso y desde el absoluto cero cómo configurar tu entorno, saltarte las restricciones de privacidad de Apple y **leer el porcentaje de batería real de tu iPhone** en tiempo real mediante un contenedor nativo.

El problema con el JavaScript clásico en iOS

Si intentas usar la API estándar de JavaScript `navigator.getBattery()` en un iPhone, tu aplicación fallará con un error de tipo `TypeError: navigator.getBattery is not a function`.

¿Por qué pasa esto? Apple eliminó por completo el soporte de esta API en su motor WebKit (Safari/WKWebView) por motivos de privacidad, evitando que webs maliciosas rastrearán a los usuarios. Para solucionarlo en una App real, necesitamos usar un **Plugin Nativo de Capacitor** que le pida el porcentaje de batería directamente al sistema operativo iOS y se lo devuelva de forma segura a nuestro código web.

Paso 1: Instalar las herramientas necesarias en tu Mac

Antes de tocar una sola línea de código, necesitas asegurarte de tener el motor de desarrollo listo en tu sistema:

- Descarga e instala **Node.js** desde su página oficial (o mediante Homebrew ejecutando `brew install node`).
- Abre la App Store en tu Mac, busca **Xcode** (el entorno oficial de Apple) e instálalo.

Paso 2: Crear el proyecto desde la Terminal

Abre la aplicación Terminal en tu Mac y ejecuta los siguientes comandos en orden para crear la estructura de tu proyecto:

1. **Crear una carpeta vacía e ingresar a ella:**`Bashmkdir mibateria && cd mibateria`
2. **Inicializar el entorno de Node.js:**`Bashnpm init -y`
3. **Instalar el núcleo de Capacitor:**`Bashnpm install @capacitor/core @capacitor/cli`
4. **Crear la estructura de configuración inicial:**`Bashnpx cap init "EstadoBateria" "com.ejemplo.bateria" --web-dir=www`
(Nota: Aquí le estamos indicando a Capacitor que busque nuestros archivos web dentro de una carpeta llamada www).

Paso 3: Instalar el Plugin de Hardware

Para poder hablar con el sistema operativo iOS y extraer los datos de la batería, instalamos el plugin oficial de

dispositivos de Capacitor:

```
npm install @capacitor/device
```

Paso 4: Crear el código HTML5 y JavaScript

Seguimos en la terminal para crear la estructura web de nuestra aplicación:

1. **Crear la carpeta contenedora www:** `Bashmkdir www`
2. **Crear el archivo index.html:** Abre tu editor de código favorito (como VS Code) y crea el archivo index.html dentro de la carpeta www. Copia y pega este código exacto:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Estado de la Batería</title>
  <style>
    body {
      display: flex;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
      background-color: #f2f2f7;
    }
    .card {
      text-align: center;
      padding: 30px;
      border-radius: 20px;
      background-color: white;
      box-shadow: 0 10px 25px rgba(0,0,0,0.08);
      width: 80%;
```

```

        max-width: 350px;
    }
    h1 {
        color: #1c1c1c;
        font-size: 24px;
        margin-bottom: 20px;
    }
    .battery-level {
        font-size: 64px;
        font-weight: bold;
        color: #34c759;
        margin: 10px 0;
    }
    .status {
        font-size: 16px;
        color: #8e8e93;
        margin-top: 10px;
    }
</style>
</head>
<body>

    <div class="card">
        <h1>Batería del iPhone</h1>
        <div id="percentage" class="battery-level">--%</div>
        <div id="charging-status" class="status">Detectando
estado...</div>
    </div>

    <script>
        async function updateBatteryInfo() {
            try {
                // Verificamos si el puente de Capacitor está
inyectado globalmente por iOS
                if (window.Capacitor &&
window.Capacitor.Plugins && window.Capacitor.Plugins.Device) {
                    const Device =
window.Capacitor.Plugins.Device;
                    const info = await
Device.getBatteryInfo();
                    // iOS devuelve un float (ej: 0.84), lo

```

```

convertimos a entero de porcentaje (84%)
        const level = Math.round(info.batteryLevel
* 100);
document.getElementById('percentage').textContent = level +
'%';

        // Cambiar el color del texto
dinámicamente al estilo iOS
        const percentageElement =
document.getElementById('percentage');
        if (level <= 20) {
            percentageElement.style.color =
'#ff3b30'; // Rojo (Batería baja)
        } else if (level <= 50) {
            percentageElement.style.color =
'#ff9500'; // Naranja (Medio)
        } else {
            percentageElement.style.color =
'#34c759'; // Verde (Óptimo)
        }

        // Determinar el estado de carga actual
        const chargingStatus = info.isCharging ?
"⚡ Cargando" : "🔋 Descargando";
        document.getElementById('charging-
status').textContent = chargingStatus;
        } else {
            document.getElementById('charging-
status').textContent = "Esperando puente nativo...";
        }
    } catch (error) {
        document.getElementById('charging-
status').textContent = "Error de hardware";
        console.error("Error al obtener datos:",
error);
    }
}

// Ejecutar de forma segura cuando la interfaz esté
lista
window.addEventListener('DOMContentLoaded', () => {

```

```
        // Un pequeño delay inicial para asegurar que el
puente nativo se haya inyectado
        setTimeout(updateBatteryInfo, 500);
        // Actualizar automáticamente cada 5 segundos
        setInterval(updateBatteryInfo, 5000);
    });
</script>

</body>
</html>
```

Paso 5: Añadir la plataforma iOS y Sincronizar

Con nuestro código web y plugins listos, es hora de preparar el entorno de Apple. Ejecuta estos comandos en tu terminal:

```
npm install @capacitor/ios
npx cap add ios
npx cap sync ios
```

Paso 6: Evitar el Error 908 (Truco de conexión)

Xcode suele trabarse al procesar los símbolos de sistema de dispositivos iOS modernos si la conexión de red interfiere. Para evitar que la compilación falle con el molesto código de error 908, haz esto antes de abrir el entorno:

1. Desconecta el cable USB de tu iPhone.
2. En tu iPhone, abre el Centro de Control y activa el **Modo Avión**. Asegúrate manualmente de que el Wi-Fi y el Bluetooth estén completamente apagados.
3. Vuelve a conectar tu iPhone a la Mac mediante el cable USB.

Ahora sí, abre tu proyecto en Xcode ejecutando:

npx cap open ios

Paso 7: Compilar y ejecutar desde Xcode

Una vez que Xcode se cargue por completo, realiza estos tres pasos clave para transferir la App:

1. **Configurar tu firma de desarrollador:** Haz clic en la raíz del proyecto (la carpeta azul llamada App arriba a la izquierda). Ve a la pestaña **Signing & Capabilities** en el panel central. En la sección *Team*, despliega el menú y selecciona tu Apple ID.
2. **Seleccionar tu dispositivo:** En la barra de herramientas superior (al lado del botón de *Play*), haz clic en el selector de dispositivos y elige tu **iPhone real** conectado por USB.
3. **Limpiar y lanzar:** Ve al menú superior, selecciona **Product -> Clean Build Folder** (para evitar cachés viejos) y luego haz clic en el botón **Play** (el triángulo equilátero).

Paso 8: Autorizar la App en el iPhone (Solo la primera vez)

Cuando Xcode termine, verás el icono en tu pantalla de inicio, pero al pulsarla te indicará que es un «*Desarrollador no confiable*». Es una medida de protección de iOS.

Para solucionarlo:

1. En tu iPhone, ve a **Ajustes -> General**.
2. Desplázate hacia abajo y entra en **Administración de dispositivos y VPN**.
3. Selecciona tu correo de Apple ID y pulsa en «**Confiar en**

[Tu Correo]».

***iListo!** Abre la aplicación en tu iPhone. Ahora verás en tiempo real el porcentaje exacto de tu batería y el estado de carga actual corriendo de forma nativa gracias al poder de Capacitor y JavaScript. ¡A programar!*