

Crear un panel de pánico con GPS real en iPhone

20:30



PANEL DE PÁNICO

Acciones rápidas de asistencia activa



REPRODUCIR ALARMA



TRANSMITIR POSICIÓN GPS



LLAMAR EMERGENCIAS (091)



ENVIAR ALERTA WHATSAPP

El mayor desafío al crear aplicaciones híbridas con **Ionic Capacitor** no es el código web, sino la integración nativa y la gestión estricta de la privacidad de Apple (Sandbox). Muchos tutoriales fallan porque la App se bloquea en el iPhone al intentar acceder al GPS, o porque el puente interno de Capacitor no se detecta debido a una mala configuración del esquema de red en iOS.

En este megapost vamos a construir un **Cuadro de Mando de Pánico Avanzado**. Aprenderás a integrar geolocalización real por satélite, configurar la alarma acústica por software, enlazar con llamadas de emergencia y WhatsApp, y lo más importante: **aprenderás a solucionar los errores persistentes de Sandbox y del módulo de Capacitor en iOS**.

Requisitos previos

Para poder compilar este proyecto, necesitas un Mac con el siguiente software instalado y actualizado:

1. **Node.js**: El motor para instalar módulos.
2. **Xcode**: La herramienta oficial de Apple para compilar apps.

Parte 1: Creación y Estructura del Proyecto

Vamos a levantar los cimientos desde la **Terminal** de tu Mac. Asegúrate de seguir el orden exacto.

```
# 1. Crear una carpeta limpia para el megaproyecto
mkdir megapopanico
```

```
# 2. Entrar a la carpeta
cd megapopanico
```

```
# 3. Inicializar el entorno Node.js (genera el archivo
package.json)
npm init -y
```

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

Parte 2: Configuración Crítica del Servidor de Capacitor

Definimos la identidad de la aplicación. Ejecuta este comando en la terminal:

```
npx cap init "BotonPanico" "com.jesusninoc.prueba" --web-dir=www
```

El Paso Clave para que funcione en el iPhone:

Ahora, abre el archivo recién creado llamado **capacitor.config.json** en tu editor de código favorito (VS Code, Sublime Text, o incluso TextEdit) y añade el objeto "server" para forzar el esquema seguro de Capacitor en iOS.

Edita el archivo para que quede **EXACTAMENTE** así:

```
{
  "appId": "com.jesusninoc.prueba",
  "appName": "BotonPanico",
  "webDir": "www",
  "server": {
    "iosScheme": "capacitor",
    "hostname": "localhost"
  }
}
```

Nota: Esta configuración es vital para solucionar el error «módulo capacitor no detectado» en el iPhone físico.

Parte 3: Instalación del Plugin de Geolocalización

Para poder invocar el chip GPS real desde JavaScript, necesitamos su plugin oficial. Vuelve a la terminal y ejecuta:

```
npm install @capacitor/geolocation
```

Parte 4: Crear la interfaz del Panel de Pánico (HTML5 Completo)

Generamos la carpeta web contenedora:

```
mkdir www
```

Crea el archivo **index.html** dentro de la carpeta `www/` y pega este código unificado que ya incluye el script puente (`capacitor.js`), la alarma por software, las APIs de comunicación y el control robusto de geolocalización con permisos integrados:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Panel de Emergencia GPS</title>
  <script src="capacitor.js"></script>

  <style>
    body {
      background-color: #111827;
      color: #ffffff;
      margin: 0;
      padding: 20px;
      display: flex;
      flex-direction: column;
      align-items: center;
```

```
        justify-content: center;
        min-height: 100vh;
        box-sizing: border-box;
    }
    h1 {
        font-size: 26px;
        color: #ef4444;
        text-transform: uppercase;
        letter-spacing: 1px;
        margin-bottom: 5px;
        text-align: center;
    }
    .subtitle {
        font-size: 14px;
        color: #9ca3af;
        margin-bottom: 25px;
        text-align: center;
    }
    .container {
        width: 100%;
        max-width: 400px;
        display: flex;
        flex-direction: column;
        gap: 15px;
    }
    .btn {
        border: none;
        border-radius: 16px;
        padding: 20px;
        font-size: 16px;
        font-weight: bold;
        color: white;
        cursor: pointer;
        display: flex;
        align-items: center;
        justify-content: center;
        gap: 12px;
        box-shadow: 0 4px 15px rgba(0,0,0,0.3);
        transition: transform 0.1s ease;
    }
    .btn:active {
```

```

        transform: scale(0.95);
    }
    .btn-alarm { background-color: #dc2626; box-shadow: 0
4px 15px rgba(220, 38, 38, 0.3); }
    .btn-gps { background-color: #7c3aed; box-shadow: 0
4px 15px rgba(124, 58, 237, 0.3); }
    .btn-call { background-color: #2563eb; box-shadow: 0
4px 15px rgba(37, 99, 235, 0.3); }
    .btn-msg { background-color: #16a34a; box-shadow: 0
4px 15px rgba(22, 163, 74, 0.3); }
    .gps-display {
        background-color: #1f2937;
        border: 1px solid #374151;
        border-radius: 14px;
        padding: 15px;
        font-size: 14px;
        color: #38bdf8;
        display: none;
    }
    .status {
        text-align: center;
        font-size: 13px;
        color: #f59e0b;
        margin-top: 10px;
    }
</style>
</head>
<body>

    <h1>⚠️ PANEL DE PÁNICO</h1>
    <div class="subtitle">Acciones rápidas de asistencia
activa</div>

    <div class="container">
        <button class="btn btn-alarm"
onclick="alternarAlarma()">
            ☐ <span id="txt-alarma">REPRODUCIR ALARMA</span>
        </button>

        <button class="btn btn-gps" onclick="obtenerGPS()">
            ☐ TRANSMITIR POSICIÓN GPS
    </div>

```

```
</button>
```

```
<div id="gps-box" class="gps-display">
    <strong>☐☐ DATOS DE TELEMETRÍA
GPS:</strong><br><br>
    Latitud: <span id="gps-lat">--</span><br>
    Longitud: <span id="gps-lon">--</span><br>
    Precisión: ±<span id="gps-acc">--</span> metros
</div>
```

```

    <button class="btn btn-call"
onclick="iniciarLlamada()">
    ☐ LLAMAR EMERGENCIAS (091)
</button>
```

```

    <button class="btn btn-msg"
onclick="enviarMensajeEmergencia()">
    ☐ ENVIAR ALERTA WHATSAPP
</button>
<div id="status-log" class="status"></div>
</div>
```

```
<script>
    // --- FUNCIÓN ALARMA SONORA POR SOFTWARE ---
    let audioCtx = null, oscillator = null, gainNode =
null, alarmaActiva = false;

    function alternarAlarma() {
        const txt = document.getElementById('txt-alarma');
        const log = document.getElementById('status-log');
        if (!alarmaActiva) {
            if (!audioCtx) audioCtx = new
(window.AudioContext || window.webkitAudioContext)();
            oscillator = audioCtx.createOscillator();
            gainNode = audioCtx.createGain();
            oscillator.type = 'sawtooth';
            oscillator.frequency.setValueAtTime(880,
audioCtx.currentTime);
            gainNode.gain.setValueAtTime(0.5,
audioCtx.currentTime);
            oscillator.connect(gainNode);
```

```

        gainNode.connect(audioCtx.destination);
        oscillator.loop = true;
        oscillator.start();
        alarmaActiva = true;
        txt.textContent = "DETENER ALARMA";
        log.textContent = "ALERTA ACÚSTICA ACTIVADA";
    } else {
        if (oscillator) { oscillator.stop();
oscillator.disconnect(); }
        alarmaActiva = false;
        txt.textContent = "REPRODUCIR ALARMA";
        log.textContent = "Alarma silenciada.";
    }
}

// --- FUNCIÓN: OBTENER GPS REAL (CON CORRECCIÓN DE
PERMISOS NATIVOS NATIVOS) ---
// Soluciona el error: unable to make sandbox
extension: [1: Operation not permitted]
async function obtenerGPS() {
    const log = document.getElementById('status-log');
    log.textContent = "Verificando puente nativo...";
    // Validar si Capacitor ya inicializó el puente
interno
    if (typeof window.Capacitor === 'undefined' ||
!window.Capacitor.Plugins.Geolocation) {
        log.textContent = "❌ Error: Inicializando
sistema nativo. Reintente en un segundo.";
        return;
    }

    try {
        const Geolocation =
window.Capacitor.Plugins.Geolocation;
        // 1. Forzar a iOS a pedir el permiso nativo
mediante código (requestPermissions)
        const permissionStatus = await
Geolocation.requestPermissions();
        if (permissionStatus.location === 'granted' ||
permissionStatus.location === 'whenInUse') {
            log.textContent = "Buscando satélites

```

```

GPS...";

        // 2. Ahora que el Sandbox está abierto,
        pedimos la posición por hardware
        const coordinates = await
Geolocation.getCurrentPosition({
            enableHighAccuracy: true,
            timeout: 10000
        });

        // Renderizar los datos
        document.getElementById('gps-
lat').textContent = coordinates.coords.latitude;
        document.getElementById('gps-
lon').textContent = coordinates.coords.longitude;
        document.getElementById('gps-
acc').textContent = Math.round(coordinates.coords.accuracy);
        document.getElementById('gps-
box').style.display = 'block';
        log.textContent = "✅ Ubicación fijada
correctamente.";
    } else {
        log.textContent = "❌ Permiso GPS denegado
por el usuario.";
    }
} catch (error) {
    console.error("Error completo de
geolocalización:", error);
    log.textContent = "❌ Error de Sandbox: Acceso
al chip denegado.";
}

}

// --- FUNCIÓN LLAMADA ---
function iniciarLlamada() {
    window.location.href = tel:091;
}

// --- FUNCIÓN WHATSAPP ---
function enviarMensajeEmergencia() {
    const numeroContacto = "346000000000"; // Cambiar
por el número de tu contacto de emergencia

```

```
        const textoMensaje = encodeURIComponent("⚠️  
¡Emergencia! He activado mi botón de pánico. Por favor  
ayúdame.");  
        window.location.href =  
https://wa.me/${numeroContacto}?text=${textoMensaje};  
    }  
</script>  
</body>  
</html>
```

Parte 5: Sincronizar y Reconstruir la plataforma iOS

Vuelve a la **Terminal** para transferir el HTML y enlazar los módulos nativos. Seguiremos un orden estricto de limpieza para evitar errores de caché en Xcode:

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

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

```
# 3. Transferir el HTML y generar los scripts puente (como  
capacitor.js)  
npx cap sync ios
```

Parte 6: Configurar Permisos de Privacidad GPS en Xcode

Apple exige estrictamente que documentes por qué usas la geolocalización. Si no lo haces en Xcode, la App se cerrará sola.

1. Abre tu Xcode ejecutando en la terminal: `Bash npx cap open ios`
2. En la barra lateral izquierda, haz clic en la carpeta azul raíz llamada **App**.

3. En la lista de carpetas, haz clic sobre el archivo llamado **Info** (o Info.plist).
4. Pasa el ratón sobre cualquier fila libre, haz clic en el icono de + (Añadir fila) e introduce la siguiente llave con tu explicación obligatoria:

Key	Type	Value
Privacy – Location When In Use Usage Description	String	Necesitamos acceder al GPS para rastrear tu posición exacta en caso de pánico.

Parte 7: El truco para evitar el Error de Sesión XPC y Despliegue Final

1. Desconecta tu iPhone del cable USB de la Mac.
2. Abre el Centro de Control y **activa el Modo Avión** (cerciórate de que Wi-Fi y Bluetooth queden completamente inhabilitados de forma manual).
3. Vuelve a conectar tu iPhone a la Mac con el cable USB.
4. Regresa a **Xcode**:
5. En la barra superior, cambia el simulador genérico y **selecciona tu iPhone físico**.
6. **LIMPIEZA DE CACHE (VITAL)**: Ve al menú superior, haz clic en **Product -> Clean Build Folder** (o presiona Cmd + Shift + K). Esto soluciona los persistentes errores de xpc_user_sessions que has estado viendo.
7. Presiona el botón **Play**.

Parte 8: Otorgar confianza en iOS y

Prueba final

Al ser la primera vez que instalas el proyecto bajo tu Apple ID:

1. Ve a tu iPhone en **Ajustes > General**.
2. Entra a **Administración de dispositivos y VPN**.
3. Pulsa sobre tu cuenta de desarrollador y haz clic en **«Confiar»**.

¡Felicidades! Abre la aplicación, el aviso de error desaparecerá, pulsa en **TRANSMITIR POSICIÓN GPS**, acepta el permiso de privacidad que desplegará iOS y verás tus coordenadas geoespaciales exactas en tiempo real. ¡Todo resuelto perfectamente!