

Creación de una «caja negra» GPS en segundo plano para iOS usando Capacitor 8

10:27



CAJA NEGRA GPS

Persistencia Local cada 2 Minutos



INICIAR MONITORIZACIÓN (2 MIN)



ABRIR FICHERO DE POSICIONES



BORRAR FICHERO HISTÓRICO



ÚLTIMA CAPTURA NATIVA:

Latitud: --

Longitud: --

Estado: Inactivo

Consola lista. Esperando acción...



CONTENIDO DEL FICHERO EN DISCO:

El archivo está cerrado o vacío.

PASO 1: Creación del Proyecto e Instalación Limpia (Terminal)

Abre una **nueva ventana de la Terminal** en tu Mac, copia este bloque completo, pégalo y pulsa Enter:

```
mkdir rastreador-gps-definitivo
cd rastreador-gps-definitivo
npm init -y
npm install @capacitor/core@latest @capacitor/cli@latest
@capacitor/ios@latest --legacy-peer-deps
npm install @capacitor-community/background-geolocation --
legacy-peer-deps
npm install @capacitor/preferences@latest --legacy-peer-deps
npx cap init "BotonPanico" "com.jesusninoc.prueba" --web-
dir=www
```

PASO 2: Configuración del Motor Nativo (capacitor.config.json)

Abre el archivo **capacitor.config.json** en la raíz de tu proyecto, borra todo lo que tenga y pega esta configuración nativa limpia:

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

PASO 3: Creación de la Interfaz y Filtros Anti-Saturación (www/index.html)

1. Crea la carpeta web desde tu terminal: `Bashmkdir www`
2. Crea el archivo **index.html** dentro de la carpeta `www/` y pégale este código fuente completo y verificado (con el reloj guardián corregido de forma limpia):

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Caja Negra GPS - Pánico</title>
  <script src="capacitor.js"></script>
  <style>
    body {
      background-color: #0f172a;
      color: #ffffff;
      margin: 0;
      padding: 20px;
      display: flex;
      flex-direction: column;
      align-items: center;
      min-height: 100vh;
      box-sizing: border-box;
    }
    h1 {
      font-size: 22px;
      color: #f43f5e;
      text-transform: uppercase;
      letter-spacing: 1px;
      margin-bottom: 5px;
      text-align: center;
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <h1>Caja Negra GPS</h1>
</body>
</html>
```

```
.subtitle {
  font-size: 13px;
  color: #94a3b8;
  margin-bottom: 25px;
  text-align: center;
}

.container {
  width: 100%;
  max-width: 400px;
  display: flex;
  flex-direction: column;
  gap: 12px;
}

.btn {
  border: none;
  border-radius: 12px;
  padding: 16px;
  font-size: 15px;
  font-weight: bold;
  color: white;
  cursor: pointer;
}

.btn-start { background-color: #6d28d9; }
.btn-read { background-color: #0284c7; }
.btn-delete { background-color: #b91c1c; }

.gps-display {
  background-color: #1e293b;
  border: 1px solid #334155;
  border-radius: 14px;
  padding: 15px;
  font-size: 13px;
  color: #38bdf8;
}

.file-viewer {
  background-color: #020617;
  border: 1px solid #1e293b;
  border-radius: 14px;
  padding: 15px;
  font-size: 12px;
  color: #10b981;
  max-height: 250px;
}
```

```

        overflow-y: auto;
        white-space: pre-wrap;
        word-break: break-all;
    }
    .status {
        text-align: center;
        font-size: 13px;
        color: #f59e0b;
        margin-top: 5px;
    }
</style>
</head>
<body>

    <h1>☐ CAJA NEGRA GPS</h1>
    <div class="subtitle">Persistencia Local Controlada (2
Minutos)</div>

    <div class="container">
        <button class="btn btn-start"
onclick="iniciarRastreoSegundoPlano()">
            ☐ INICIAR MONITORIZACIÓN (2 MIN)
        </button>

        <button class="btn btn-read"
onclick="abrirYLeerFicheroGps()">
            ☐ ABRIR FICHERO DE POSICIONES
        </button>

        <button class="btn btn-delete"
onclick="borrarFicheroGps()">
            ☐☐ BORRAR FICHERO HISTÓRICO
        </button>

        <div class="gps-display">
            <strong>☐☐ ÚLTIMA CAPTURA
FILTRADA:</strong><br><br>
            Latitud: <span id="gps-lat">--</span><br>
            Longitud: <span id="gps-lon">--</span><br>
            Estado: <span id="gps-status" style="color:
#64748b;">Inactivo</span>

```

```

    </div>
    <div id="status-log" class="status">Consola lista.
Esperando acción...</div>

    <strong>[] CONTENIDO DEL FICHERO EN DISCO:</strong>
    <div id="file-content" class="file-viewer">El archivo
está cerrado o vacío.</div>
</div>

<script>
const { Preferences } = window.Capacitor.Plugins;
const INTERVALO_MINIMO = 120000; // 2 minutos exactos en
milisegundos

    async function iniciarRastreoSegundoPlano() {
        const log = document.getElementById('status-log');
        if (typeof window.Capacitor === 'undefined' ||
!window.Capacitor.Plugins.BackgroundGeolocation) {
            log.textContent = "[] Error: Puente nativo no
disponible.";
            return;
        }

                const BackgroundGeolocation =
window.Capacitor.Plugins.BackgroundGeolocation;

        try {
            log.textContent = "Solicitando permisos a iOS...";

            await BackgroundGeolocation.addWatcher(
                {
                    backgroundTitle: "Seguimiento de
Emergencia",
                    backgroundMessage: "Guardando tu ruta en
el almacenamiento de forma optimizada.",
                    requestPermissions: true,
                    stale: false,
                    distanceFilter: 10,                // Margen de
metros
                    interval: 120000,
                    fastestInterval: 120000,

```

```

    },
    async function callback(location, error) {
        if (error) {
            console.error(error);
            return;
        }

        if (location) {
            const ahoraMilisegundos = Date.now();

            try {
                // 1. LEER EL TIEMPO GUARDADO EN
                EL DISCO DURO (No se borra aunque iOS congele la app)
                const { value: tiempoGuardado } =
                await Preferences.get({ key: 'ultimo_tiempo_disco' });
                const ultimoTiempo =
                tiempoGuardado ? parseInt(tiempoGuardado) : 0;

                // COMPROBACIÓN MATEMÁTICA REAL
                CONTRA EL DISCO

                if (ultimoTiempo > 0 &&
                (ahoraMilisegundos - ultimoTiempo) < INTERVALO_MINIMO) {
                    console.log("❌ Omitido por
                persistencia: No han pasado los 2 minutos.");
                    return;
                }

                // 2. FIJAR EL NUEVO TIEMPO
                INMEDIATAMENTE EN EL DISCO DURO
                await Preferences.set({
                    key: 'ultimo_tiempo_disco',
                    value:
                ahoraMilisegundos.toString()
                });

                // Formatear tiempos para el
                reporte válido

                const ahora = new Date();
                const fechaFormateada =
                ahora.toLocaleDateString();

                const horaFormateada =

```

```

ahora.toLocaleTimeString();

        document.getElementById('gps-
lat').textContent = location.latitude;
        document.getElementById('gps-
lon').textContent = location.longitude;
        document.getElementById('gps-
status').textContent = "ACTIVO (FILTRADO POR DISCO)";
        log.textContent = "☐ Guardado
verificado: " + horaFormateada;

        // 3. AÑADIR LA POSICIÓN AL
HISTORIAL
        const { value: archivoPosiciones }
= await Preferences.get({ key: 'archivo_posiciones' });
        let registros = archivoPosiciones
? JSON.parse(archivoPosiciones) : [];

        const nuevoPunto = {
            fecha: fechaFormateada,
            hora: horaFormateada,
            latitud: location.latitude,
            longitud: location.longitude,
            precision:
location.accuracy.toFixed(1) + " metros"
        };

        registros.push(nuevoPunto);

        await Preferences.set({
            key: 'archivo_posiciones',
            value:
JSON.stringify(registros)
        });

        console.log("☐ Escrito con éxito
en la caja negra:", nuevoPunto);

    } catch (errStorage) {
        console.error("Error en
operaciones de lectura/escritura:", errStorage);
    }

```

```

        }
    }
    );

    log.textContent = "❑ Filtro por hardware de disco activo.";

    } catch (err) {
        console.error(err);
        log.textContent = "❑ Error en el puente GPS.";
    }
}

async function abrirYLeerFicheroGps() {
    const viewer = document.getElementById('file-content');
    const log = document.getElementById('status-log');
    try {
        const { value } = await Preferences.get({ key: 'archivo_posiciones' });
        if (!value || JSON.parse(value).length === 0) {
            viewer.textContent = "El archivo está vacío. No hay posiciones registradas todavía.";
            log.textContent = "❑ Archivo leído (Vacío).";
            return;
        }

        const datos = JSON.parse(value);
        let textoFormateado = === REPORTE OPTIMIZADO (Total: ${datos.length} registros) ===\n;
        datos.forEach((pos, index) => {
            textoFormateado += [`${index + 1}] F: ${pos.fecha} H: ${pos.hora} -> Lat: ${pos.latitud}, Lon: ${pos.longitud} (Precisión: ${pos.precision})\n`;
        });

        viewer.textContent = textoFormateado;
        log.textContent = "❑ Archivo abierto correctamente.";
    } catch (err) {

```

```

        viewer.textContent = "❌ Error al abrir el
archivo.";
    }
}

async function borrarFicheroGps() {
    const viewer = document.getElementById('file-
content');
    const log = document.getElementById('status-log');
    if (confirm("¿Estás seguro de que deseas destruir el
archivo histórico de posiciones GPS?")) {
        try {
            await Preferences.remove({ key:
'archivo_posiciones' });
            await Preferences.remove({ key:
'ultimo_tiempo_disco' }); // Borramos también la marca de
tiempo
            viewer.textContent = "El archivo está cerrado
o vacío.";
            log.textContent = "✅ Archivo destruido por
completo.";
        } catch (err) {
            log.textContent = "❌ No se pudo borrar el
archivo.";
        }
    }
}
</script>
</body>
</html>

```

PASO 4: Sincronización y Script de Parche Automático (si no funciona, se abre el fichero y se cambia el valor por 8.4.0)

Ejecuta estos tres comandos en orden dentro de tu terminal. El segundo inyectará la compatibilidad nativa con Capacitor 8

directamente sobre el plugin sin que tengas que usar el Finder de tu Mac:

```
npx cap add ios
npx cap sync ios
sed -i '' 's/\.upToNextMajor(from:
"7.0.0")/\.upToNextMajor(from:
"8.4.0")/g'
node_modules/@capacitor-community/background-
geolocation/Package.swift
```

PASO 5: Apertura y Sincronización Interna de Xcode

Lanza el software nativo desde tu consola:

```
npx cap open ios
```

En cuanto **Xcode** cargue por completo en la pantalla y te muestre la advertencia del error rojo en la esquina superior, ve a la barra de menús de arriba de la pantalla de tu Mac y pulsa exactamente: **File -> Packages -> Resolve Package Versions**

(Xcode compilará las dependencias en segundo plano y el error se tornará en verde de inmediato).

PASO 6: Configuración de Seguridad de Apple (Panel Gráfico)

A. Modos de Fondo (Background Modes)

En el lateral izquierdo de Xcode, haz un clic sobre el icono raíz azul superior llamado **App**.

En el panel central que se despliega, entra en la pestaña **Signing & Capabilities**.

Haz clic en el botón pequeño **+ Capability** (situado justo bajo

la pestaña).

Escribe **Background Modes** en el buscador flotante y dale doble clic.

Selecciona únicamente estas dos casillas de verificación:

- **Location updates**
- **Background fetch**

B. Llaves de Privacidad de Ubicación Nativa

Haz clic en la pestaña **Info** (al lado de *Signing & Capabilities*).

Desplaza el ratón sobre cualquier fila de la lista, haz clic en el icono del + que aparece a la derecha y añade estas 3 llaves:

- Privacy – Location When In Use Usage Description → *Uso del GPS para alertas de pánico.*
- Privacy – Location Always and When In Use Usage Description → *Monitorización de la caja negra GPS cada 2 minutos.*
- Privacy – Location Always Usage Description → *Se requiere el acceso permanente a la localización.*

PASO 7: Despliegue en tu iPhone real

1. Conecta tu iPhone al Mac mediante cable de datos.
2. Selecciónalo en el menú desplegable de dispositivos nativos en la barra de arriba a la izquierda en Xcode.
3. Realiza una limpieza del código residual con las teclas:

Cmd + Shift + K.

4. Pulsa el botón de **Play** (el triángulo).

⚠️ Configuración de Permisos Crítica (En tu iPhone):

1. En cuanto se abra la App, haz clic en el botón morado **INICIAR MONITORIZACIÓN (2 MIN)**.
2. iOS te pedirá los permisos; pulsa sobre «**Permitir al usar la app**».
3. Cierra la visualización de la app, abre los **Ajustes** del iPhone → **Privacidad y seguridad** → **Localización** → **BotonPanico**.
4. Modifica de forma manual el permiso marcando la opción «**Siempre**» y confirma que la casilla verde de «**Ubicación exacta**» está encendida.

¡Listo! Bloquea el móvil o ponlo en segundo plano. Al pulsar **ABRIR FICHERO DE POSICIONES**, verás la telemetría grabada de forma matemática, controlada y limpia cada 2 minutos exactos en disco.