

Scrip.js Acciona GPS

```
// Base de datos SIMULADA (en producción usa Firebase/Backend)
const mobileData = {
  "Movil1": { coords: null, lastUpdate: null },
  "Movil2": { coords: null, lastUpdate: null }
};

// Configuración del mapa
const SCL_COORDS = [-33.3931, -70.7858];
const ZOOM_INICIAL = 14;
let map;
const markers = {}; // Almacena los marcadores

// Inicialización
document.addEventListener('DOMContentLoaded', function() {
  if (isLoginPage()) setupLogin();
  else initMap();
});

// Configurar login
function setupLogin() {
  document.getElementById('loginForm').addEventListener('submit', async function(e) {
    e.preventDefault();
    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;

    if (await authenticate(username, password)) {
      localStorage.setItem('currentUser', username);
      mobileData[username].status = 'online';
      window.location.href = 'index.html';
    } else {
      alert('Credenciales incorrectas');
    }
  });
}

// Autenticación simulada
```

```

async function authenticate(username, password) {
  // Verifica credenciales
  const validCredentials = {
    "Movil1": "MovilAcciona2025",
    "Movil2": "MovilAcciona2025",
    "Admin": "Acciona2025" // Usuario extra para visualización
  };
  return validCredentials[username] === password;
}

```

```

// Inicializar mapa
function initMap() {
  map = L.map('map').setView(SCL_COORDS, ZOOM_INICIAL);

  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '&copy; OpenStreetMap'
  }).addTo(map);

  // Marcador inicial para SCL
  L.marker(SCL_COORDS).addTo(map)
    .bindPopup('Aeropuerto SCL')
    .openPopup();

  // Actualizar posiciones cada 5 segundos
  setInterval(updateMobilePositions, 5000);
}

```

```

// Actualizar posiciones (simula recepción de datos GPS)
function updateMobilePositions() {
  // En un sistema real, aquí harías una petición a tu backend
  Object.keys(mobileData).forEach(mobile => {
    if (mobileData[mobile].coords) {
      updateOrCreateMarker(mobile, mobileData[mobile].coords);
    }
  });
}



```

```

// Crear/actualizar marcador

```

```

function updateOrCreateMarker(mobile, coords) {
  if (!markers[mobile]) {
    markers[mobile] = L.marker(coords, {
      icon: L.divIcon({
        className: `mobile-icon ${mobile}`,
        html: mobile === 'Movil1' ? '' : '',
        iconSize: [30, 30]
      })
    }).addTo(map)
    .bindPopup(`${mobile}`);
  } else {
    markers[mobile].setLatLng(coords);
  }
}

// Simulador de datos GPS (para prueba)
function simulateGPS(mobile) {
  // Coordenadas aleatorias cerca de SCL (solo para demo)
  const offset = () => (Math.random() * 0.01 - 0.005);
  mobileData[mobile].coords = [
    SCL_COORDS[0] + offset(),
    SCL_COORDS[1] + offset()
  ];
  mobileData[mobile].lastUpdate = new Date();

  // En sistema real, esto vendría del GPS del dispositivo
  updateOrCreateMarker(mobile, mobileData[mobile].coords);
}

// Iniciar seguimiento (llamar desde cada dispositivo)
function startTracking(mobile) {
  // En dispositivos reales:
  if (navigator.geolocation) {
    navigator.geolocation.watchPosition(
      position => {
        const coords = [position.coords.latitude, position.coords.longitude];
        mobileData[mobile] = {
          coords: coords,

```

```
        lastUpdate: new Date()
    };
    // Enviar datos al backend en sistema real
},
error => console.error(`Error GPS ${mobile}:`, error),
{ enableHighAccuracy: true }
);
} else {
    console.log(`${mobile}: GPS no disponible`);
    // Modo simulación para pruebas
    setInterval(() => simulateGPS(mobile), 10000);
}
}

// Helper para páginas
function isLoginPage() {
    return window.location.pathname.includes('login.html');
}
```