



## TRABAJO FIN DE MÁSTER

Desarrollo de una app para la captura de ruido ambiental geolocalizado

  
aplicacions i gestió 1ª edició

Organizado por: **UAB**  
Universitat Autònoma  
de Barcelona

  
**ICGC**  
Institut  
Cartogràfic i Geològic  
de Catalunya

Institución colaboradora: **ISGlobal** Instituto de  
Salud Global  
Barcelona

Autor: Eric Grau Parera  
Tutora Empresa: Antònia Valentín

Tutor Máster: Aitor Alsina

Curso 2017 - 2018

## Contenido

<b>RESUMEN</b> .....	1
<b>RESUM</b> .....	2
<b>ABSTRACT</b> .....	3
<b>1. INTRODUCCIÓN</b> .....	4
1.1. ANTECEDENTES.....	4
1.2. MARCO INSTITUCIONAL.....	6
1.3. CONTEXTO.....	7
1.4. PLAN DE TRABAJO Y FASES DEL PROYECTO.....	7
<b>2. OBJETIVOS</b> .....	9
<b>3. ANÁLISIS DE LOS REQUERIMIENTOS</b> .....	10
3.1. ESPECIFICACIONES .....	10
3.1.1. Propósito .....	10
3.1.2. Alcance.....	11
3.1.3. Definiciones.....	11
3.1.4. Capacidades y condiciones .....	13
3.1.5. Suposiciones y dependencias .....	14
3.2. REQUERIMIENTOS FUNCIONALES .....	14
3.3. REQUERIMIENTOS NO FUNCIONALES .....	15
3.4. CARACTERÍSTICAS DE LOS USUARIOS .....	16
3.5. DIAGRAMA GENERAL DE LOS CASOS DE USO .....	16
3.6. RESTRICCIONES.....	17
3.7. SOLUCIONES EXISTENTES .....	17
3.7.1. Análisis de las aplicaciones y servicios.....	18
3.7.1.1. Aplicaciones nativas.....	18
3.7.1.2. Plugins .....	20
<b>4. DISEÑO FUNCIONAL</b> .....	21
4.1. CASOS DE USO.....	21
4.2. ARQUITECTURA DE LA SOLUCIÓN .....	25
<b>5. METODOLOGÍA E IMPLEMENTACIÓN</b> .....	27
5.1. DESCRIPCIÓN DEL ENTORNO DE DESARROLLO.....	27
5.2. LA BASE DE DATOS.....	28
5.3. APLICACIÓN PARA LA OBTENCIÓN DE DATOS .....	30
5.3.1. FrontEnd. Lado Cliente.....	30

5.3.2. Middleware y BackEnd. Lado Servidor .....	33
5.4. EL VISOR DE DATOS .....	33
<b>6. RESULTADOS.....</b>	<b>35</b>
<b>7. CONCLUSIONES.....</b>	<b>36</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>38</b>
<b>ANEXOS.....</b>	<b>40</b>

## Índice de figuras

Figura 1. Diagrama de planificación del proyecto.....	7
Figura 2. Diagrama General de los casos de uso .....	16
Figura 3. Caso de uso "permitir la geolocalización" .....	21
Figura 4. Caso de uso "reportar registros acústicos" .....	22
Figura 5. Caso de uso "interactuar con el mapa" .....	23
Figura 6. Caso de uso "Gestionar la base de datos" .....	24
Figura 7. Arquitectura de la solución propuesta. ....	25
Figura 8. Arquitectura de aplicación de Cordova.....	27
Figura 9. Sentencia SQL a ejecutar para la creación de la tabla "Datos".....	29
Figura 10. Mensaje devuelto tras sentencia SQL ejecutada correctamente .....	30
Figura 11. Muestra de la tabla de datos sin registros.....	30
Figura 12. Parte de código fuente referente a coordenadas y dB en form.html .....	31
Figura 13. Código PHP referente al POST entre formulario y BBDD .....	33
Figura 14. Pantalla de la app en carga .....	35
Figura 15. Pantalla del menú principal.....	35
Figura 16. Pantalla inicial del formulario al cargar.....	35
Figura 17. Formulario al presionar "Capt. Pos. y Decibelios" y pregunta percepción. .	35
Figura 18. Formulario tras apretar botón "Enviar Datos" .....	35
Figura 19. Mapa con la caja de información del punto clickado. ....	35
Figura 20. Icono aplicación "NoiseTube" .....	40
Figura 21. Icono aplicación "Sound meter - Sonómetro".....	40
Figura 22. Icono aplicación "Noise Pollution Monitor - City Noise" .....	40
Figura 23. Icono aplicación "Noise Capture" .....	41
Figura 24. Página de registro para obtención hosting gratuito.....	41
Figura 25. Página para hacer login en el hosting web gratuito .....	41
Figura 26. Menú principal Awardspace tras login.....	42
Figura 27. Página de creación de bases de datos MySQL .....	42
Figura 28. Visualización de datos referentes a BBDD creada .....	42
Figura 29. Sección BBDD para importar archivo SQL .....	43
Figura 30. Visualización de la tabla "datos" creada.....	43
Figura 31. Código script para obtención "UUID" .....	43
Figura 32. Código "UUID" en el formulario .....	43
Figura 33. Código "fecha" en el formulario .....	43
Figura 34. Código de la función JS para obtención de "fecha" .....	44
Figura 35. Código "hora" en el formulario .....	44
Figura 36. Código de la función JS para obtención de "hora".....	44
Figura 37. Código de la función JS para obtención de "geolocalización" .....	45
Figura 38. Página de GitHub para implantación de "DBmeter".....	45
Figura 39. Código JS para "registro de decibelios durante 5 segundos".....	46
Figura 40. Código para "calcular promedio de decibelios" .....	46
Figura 41. Código referente a la pregunta de "percepción del ruido" en el formulario .	46
Figura 42. Parte del script del formulario para envío de variables y valores .....	47
Figura 43. Script del formulario para realización POST y envío datos .....	47
Figura 44. Código del fichero Aws_db.php que permite conexión.....	48
Figura 45. Visualización de datos recibidos en MySQL desde formulario de la app....	48
Figura 46. Código para obtención de json y mapa Leaflet en la app.....	49
Figura 47. Código para la obtención de popup al clickar un punto en el mapa.....	49
Figura 48. Consulta lanzada en geojson.php para mapa.....	49

## Índice de manuales

MANUAL 1. INSTALACIÓN APACHE CORDOVA Y PRIMEROS PASOS .....	50
MANUAL 2. XAMPP PARA LA CREACIÓN DE UNA BBDD CON MySQL EN LOCALHOST ...	51
MANUAL 3. CREACIÓN BASE DE DATOS CON MYSQL EN LOCALHOST .....	53
MANUAL 4. PASOS REALIZADOS EN EL TEST DE CALIDAD DE LA APP .....	54
MANUAL 5. PASOS REALIZADOS EN LA INSTALACIÓN DE LA APP .....	55

## RESUMEN

### Objetivos

- Creación de un prototipo funcional de una aplicación para dispositivo móvil usando software de código abierto.
- Recogida y evaluación del ruido ambiental georreferenciado mediante uso del formulario.
- Inclusión en el formulario de la percepción del ruido por parte del usuario.
- Evaluación y elección de las soluciones existentes para llevar a cabo el proyecto.
- Conseguir una app que mantenga la privacidad del usuario en todo momento.

### Metodología

- Creación de una app móvil híbrida con Apache Cordova (HTML5, CSS3 y JS).
- Creación e implementación de una base de datos MySQL en un hospedaje de web gratuito para recepción de datos provenientes de la app.
- Diseño y desarrollo del FrontEnd con JQuery, JQuery Mobile entre otros.
- Configuración y programación del BackEnd con PHP.
- Implantación y configuración de un mapa para la visualización de los datos almacenados en la base de datos con el uso de Leaflet.

### Resultados

- Se ha realizado una app móvil híbrida que permite el registro de la contaminación acústica mediante el uso de un formulario.
- Se han implementado todos los plugins necesarios para cumplir con los objetivos (registro de decibelios, posicionamiento del usuario, UUID, fecha y hora).
- Ha sido creada una base de datos en un servidor web para la recolección de datos provenientes de la app.
- Se ha añadido una lista de opciones al formulario para que el usuario de la app exprese su percepción en base a la contaminación acústica.
- Se ha llevado a cabo el diseño e implementación de todos los niveles de desarrollo web: interfaz de usuario, FrontEnd, Middleware, Backend y Base de Datos.

### Conclusiones

- Se han llevado a cabo todos los objetivos propuestos (definición, diseño, implementación, test de calidad e instalación).
- Toda la información se registra correctamente.
- Obtención de una app híbrida realizada con código abierto y completamente funcional.
- Creación de una estructura de la app, un formulario y un mapa adecuados a los objetivos establecidos y cuidando la estética.
- Existencia de limitaciones: calibración y cancelación del ruido ambiental en los dispositivos móviles.
- Hay margen de mejora para futuras versiones.
- Resultado muy favorable: esta aplicación tiene un valor añadido porque permite contrastar el registro del ruido con la percepción del usuario.
- Se han puesto en práctica muchos conocimientos adquiridos durante el máster.

## RESUM

### Objectius

- Creació d'un prototip funcional d'una aplicació per a dispositiu mòbil fent servir programari de codi obert.
- Recollida i avaluació del soroll ambiental georeferenciat mitjançant ús del formulari.
- Inclusió en el formulari de la percepció del soroll per part de l'usuari.
- Avaluació i elecció de les solucions existents per dur a terme el projecte.
- Aconseguir una app que mantingui la privacitat de l'usuari en tot moment.

### Metodologia

- Creació d'una app mòbil híbrida amb Apache Cordova (HTML5, CSS3 i JS).
- Creació i implementació d'una base de dades MySQL en un allotjament web gratuït per a recepció de dades provinents de l'app.
- Disseny i desenvolupament del FrontEnd amb JQuery, JQuery Mobile entre d'altres.
- Configuració i programació del BackEnd amb PHP.
- Implantació i configuració d'un mapa per a la visualització de les dades emmagatzemades a la base de dades amb l'ús de Leaflet.

### Resultats

- S'ha realitzat una app mòbil híbrida que permet el registre de la contaminació acústica mitjançant l'ús d'un formulari.
- S'han implementat tots els connectors necessaris per complir amb els objectius (registre de decibels, posicionament de l'usuari, UUID, data i hora).
- Ha estat creada una base de dades en un servidor web per a la recollida de dades provinents de l'app.
- S'ha afegit una llista d'opcions al formulari perquè l'usuari de l'app expressi la seva percepció en relació a la contaminació acústica.
- S'ha dut a terme el disseny i implementació de tots els nivells de desenvolupament web: interfície d'usuari, FrontEnd, Middleware, Backend i Base de Dades.

### Conclusions

- S'han dut a terme tots els objectius proposats (definició, disseny, implementació, test de qualitat i instal·lació).
- Tota la informació es registra correctament.
- Obtenció d'una app híbrida realitzada amb codi obert i completament funcional.
- Creació d'una estructura de l'app, un formulari i un mapa adequats als objectius establerts i tenint cura de la estètica.
- Existència de limitacions: calibratge i cancel·lació del soroll ambiental en els dispositius mòbils.
- Hi ha marge de millora per a futures versions.
- Resultat molt favorable: aquesta aplicació té un valor afegit perquè permet contrastar el registre del soroll amb la percepció de l'usuari.
- S'han posat en pràctica molts coneixements adquirits durant el màster.

## ABSTRACT

### Objectives

- Creation of a functional prototype of an application for mobile device using open source software.
- Collection and evaluation of georeferenced environmental noise by using the form.
- Inclusion of user's perception in the form about noise.
- Evaluation and choice of existing solutions to carry out the project.
- Get an app that maintains the user's privacy at all times.

### Methodology

- Creation of a hybrid mobile app with Apache Cordova (HTML5, CSS3 and JS).
- Creation and implementation of a MySQL database in a free web hosting for reception of data from the app.
- Design and development of FrontEnd with JQuery, JQuery Mobile among others.
- Configuration and programming of BackEnd with PHP.
- Implementation and configuration of a map for the visualization of the data stored in the database with the use of Leaflet.

### Results

- A hybrid mobile application has been made that allows the recording of noise pollution through the use of a form.
- All necessary plugins have been implemented to achieve the goals (decibel registration, user positioning, UUID, date and time).
- A database has been created on a web server to collect data from the application.
- A list of options has been created for the users can express their own perception about noise pollution.
- The design and implementation of all levels of web development has been carried out: user interface, FrontEnd, Middleware, Backend and Database.

### Conclusions

- All the proposed objectives have been carried out (definition, design, implementation, quality test and installation).
- All information is recorded correctly.
- Obtaining a hybrid application made with open code and fully functional.
- Creation of a structure of the application, a form and a map adapted to the established objectives and care of aesthetics.
- Existence of limitations: calibration and cancellation of environmental noise in mobile devices.
- There is room for improvement for future versions.
- Very favourable result: this application has an added value because it allows contrasting the recording of noise with the perception of the user.
- Many knowledges acquired during the master's degree have been put into practice.



# 1. INTRODUCCIÓN

## 1.1. ANTECEDENTES

En el presente año 2018, se establece el primer convenio de colaboración entre el Departamento de Geografía de la Universidad Autónoma de Barcelona (UAB) y el Instituto de Salud Global (IS Global), y este es el primer proyecto que se genera fruto de ese acuerdo para la primera edición del Máster en Geoinformación.

En este sentido, los precedentes de que se dispone en el estudio de la relación del “ruido ambiental con la salud” que aquí se trabaja, es únicamente producto del trabajo realizado por IS Global, no de colaboraciones anteriores con el Dpto. de Geografía de la UAB, aunque cabe destacarlos por su importancia y rigor científico y para entender bien el tema central en el que se está trabajando.

En primer lugar, hay que poner de manifiesto que el estudio de este fenómeno en relación con la percepción del usuario no es nuevo, pues ya en el año 1993, un equipo de IC BEN, Community Response to Noise, se marcó como objetivo la creación de encuestas de alta calidad en cuestión de ruido y, en el año 2001 ya hubo un estudio clave para estudiar las diferencias existentes entre encuestas que pretendían medir la molestia por el ruido que generaba en la población (Fields, De Jong, y otros, 2001). A su vez, se realizó una estandarización de las preguntas y estudio para seleccionar diferentes palabras asociadas a una determinada escala de molestia.

Desde el año 2016, la entidad y especialmente la investigadora María Foraster se ha dedicado a “estudiar los efectos de las exposiciones relacionadas con el tráfico a lo largo de la vida, con un enfoque especial en los efectos del ruido del transporte sobre las enfermedades cardiometabólicas en adultos y en la salud de los niños” (ISGLOBAL.ORG).

A mediados del año 2016, se centran en estudiar la afectación que tiene el ruido del transporte a largo plazo y que puede contribuir en una menor actividad física, lo cual puede dar como resultado trastornos del sueño y afectación en la salud. A su vez, se pone de manifiesto el efecto que tiene sobre la salud las molestias de ruido, con la alteración del sueño y/o la percepción de la calidad ambiental

que pueden acabar conduciendo a enfermedades metabólicas (Foraster, y otros, 2016).

En Junio de 2017, se evaluaron los niveles de ruido del transporte y sus características temporales los cuales pueden afectar a la rigidez arterial y predecir el surgimiento de enfermedades cardiovasculares. Con ello se determinaba que la exposición a largo plazo al ruido ferroviario y especialmente de la carretera en un ambiente nocturno, podía afectar la rigidez arterial más que otras características del ruido (Foraster, y otros, 2017).

En Agosto de 2017, se realizó una investigación con más de 2000 participantes sobre contaminación atmosférica y enfermedades pulmonares y del corazón en adultos (SAPALDIA) mediante cuestionarios y datos de biomarcadores, lo que les permitió identificar casos de diabetes y casos con mala calidad de sueño. Pero lo realmente significativo, es ver que el ruido del transporte era incluso más relevante a la hora de desarrollar diabetes y que podía actuar a través de alteraciones del sueño (Ikenna, Foraster, y otros, 2017).

A finales de ese mismo año y ya desde fuera del ámbito de ISGlobal, se hizo una investigación para estudiar en qué medida causaba molestia el ruido ambiental procedente de aviones, carreteras y ferrocarriles, así como turbinas de viento y combinaciones de fuentes de ruido. Se quiso ver el grado de afectación que realmente tenían las personas que vivían cerca de estos focos. A su vez, se realizaron también encuestas acústicas y sociales, las cuales resultaban de mucho interés (Guski, Schreckenber y Schuemer, 2017).

Todo este atractivo por la contaminación acústica ha hecho que este fenómeno también haya tenido cabida con el surgimiento de los teléfonos inteligentes y el mercado de las apps. Buena prueba de estas aplicaciones son Noise Tube, Sound Meter, Noise Pollution Monitor y Noise Capture, los cuales son analizados en el apartado 3.7 de este documento.

El kit de la cuestión es que ninguna de ellas permite asociarlo con la percepción que tienen los usuarios, elemento clave y punto de partida de este estudio.

## 1.2. MARCO INSTITUCIONAL

Las prácticas de especialización de Desarrollo de Aplicaciones del Máster de Geoinformación se realizaron en ISGlobal Campus Mar durante los meses de Abril a Junio de 2018.

Se trata del Instituto de Salud Global de Barcelona, el cual surgió de una alianza entre la Obra Social “La Caixa” e instituciones académicas y gubernamentales que tenían como misión, afrontar los retos de la salud en un mundo globalizado.

ISGlobal tiene su origen en los ámbitos hospitalarios del Clínic y Parque de Salud Mar, así como académico (UB y UPF), haciendo que tengan a sus espaldas más de 30 años de experiencia y se haya consolidado como un nodo de excelencia basado en la investigación dentro de este campo. Apuestan por el conocimiento generado por la Ciencia a través de la formación, análisis y desarrollo global.

El verdadero objetivo de esta entidad, es corregir las desigualdades en el estado de salud de las diversas poblaciones del mundo.

El 30 de Junio de 2016, se integra a ISGlobal el Centro de Investigación en Epidemiología Ambiental (CREAL), una institución que ha dedicado diez años a la promoción y el desarrollo de la investigación epidemiológica avanzada sobre los factores ambientales que afectan a la salud humana.

De esta manera, el Instituto de Salud Global se consolida con su innovación en el conocimiento ligado a la investigación para afrontar retos de salud, especialmente de las enfermedades infecciosas, enfermedades crónicas no transmisibles y destacadamente en los determinantes ambientales.

Dicho de otro modo, las dos grandes líneas de investigación son las enfermedades infecciosas por un lado y, enfermedades no transmisibles y medio ambiente por el otro. Este último, sería de una u otra manera, el verdadero ámbito que impulsa y da sentido a nuestro objeto de estudio, aunque desde el punto de vista tecnológico: el desarrollo de una app para la captura de ruido ambiental geolocalizado (ISGLOBAL).

### 1.3. CONTEXTO

ISGlobal propone la creación de una app con información de contaminación acústica geolocalizada, registrando los decibelios y vinculándola a la percepción del usuario. Por ello, se facilitará el acceso y conocimiento de los mismos a los usuarios finales (ciudadanos y cualquier persona interesada) de forma totalmente gratuita y sin necesidad de registrarse.

Partiendo de dicha situación, las necesidades actuales pasan por crear una app que envíe los datos del usuario y su dispositivo móvil mediante un formulario a una base de datos creada (y alojada en un servidor web) para tal propósito. A su vez, se pretende que el usuario pueda consultar los datos mediante un mapa interactivo y a los técnicos de la entidad gestionarla mediante la BBDD.

### 1.4. PLAN DE TRABAJO Y FASES DEL PROYECTO

En el siguiente diagrama se muestra de forma resumida la planificación del proyecto, sus tareas y las fechas clave.

Tarea	Duración	Inicio	Fin	Abril	Mayo	Junio
F1. Análisis	14 días	23/04/2018	06/05/2018			
F2. Diseño	14 días	07/05/2018	20/05/2018			
F3. Implementación	14 días	21/05/2018	03/06/2018			
F4. Test	7 días	04/06/2018	10/06/2018			
F5. Instalación	5 días	11/06/2018	15/06/2018			

FIGURA 1. DIAGRAMA DE PLANIFICACIÓN DEL PROYECTO. FUENTE: ELABORACIÓN PROPIA

A continuación, se detallan las etapas de cada una de las fases:

#### **FASE 1. ANALISIS DE REQUERIMIENTOS**

Etapas 1. Reuniones.

Etapas 2. Redacción del documento técnico de análisis de requerimientos.

#### **FASE 2. DISEÑO**

Etapas 1. Diseño funcional y gráfico de la aplicación.

Etapas 2. Diseño arquitectura de la solución.

#### **FASE 3. IMPLEMENTACIÓN**

Etapas 1. Descripción del entorno de desarrollo.

Etapas 2. Descripción y configuración de la Base de Datos.

Etapas 3. Descripción de la metodología de obtención de los datos.

Etapas 4. Visualización de los datos almacenados en la Base de Datos.

#### **FASE 4. TEST DE CALIDAD**

Etapa 1. Test de velocidad.

Etapa 2. Comprobación funcionalidad y control de errores.

#### **FASE 5. INSTALACIÓN**

Etapa 1. Instalación del programa.

Etapa 2. Puesta en marcha.

## 2. OBJETIVOS

Con el presente proyecto se pretende crear un prototipo funcional de una aplicación que tiene como misión la recogida y evaluación del ruido ambiental a través de su georreferenciación, mediante el análisis, diseño e implementación del mismo en un dispositivo móvil Android. Estos datos de contaminación acústica, se deberán complementar dando al usuario la posibilidad de expresar su percepción del ruido.

Se espera realizar un proyecto fiel y útil donde se puedan poner en práctica los conocimientos adquiridos durante el máster y también aquellos que se vayan generando con el proceso de creación de esta aplicación y que, a su vez, resulte práctico, sencillo y funcional para quienes hagan uso de ella.

Para cumplir con este fin, se ha establecido un capítulo introductorio (punto 1), donde se ha puesto en antecedente y se ha contextualizado al usuario en los estudios de contaminación acústica. Con ello, hemos obtenido la información necesaria para establecer un plan de acción en base a las necesidades del proyecto que tenemos ante nosotros.

Más adelante (punto 3.7), se evaluarán las soluciones ya existentes (apps de Play Store y plugins) para determinar, en base a nuestros conocimientos, necesidades y tiempo, los siguientes pasos que se deben llevar a cabo para su correcto desarrollo, diseño, implementación, testeo e instalación.

## 3. ANÁLISIS DE LOS REQUERIMIENTOS

Esta etapa se realiza a partir de las reuniones mantenidas con Antonia Valentín, persona responsable de la sección de GIS en la empresa ISGLOBAL. A través de ellas se han fijado los objetivos a acometer mediante la generación de la aplicación.

Para ello, se redacta el Documento Técnico de Análisis de Requisitos (apartados 3.1 a 3.7) en el cual se muestran los resultados obtenidos mediante la temporalización creada para cada una de las fases de desarrollo (apartado 1.4), tal como se muestra a continuación.

### 3.1. ESPECIFICACIONES

#### 3.1.1. Propósito

El apartado 3.1. tiene como finalidad definir las especificaciones funcionales, no funcionales y del sistema, definir las características de los usuarios, dar una visión global de los casos de uso y las restricciones, así como dar a conocer las soluciones existentes para tal fin. Todo ello, con la idea de implementarlo en una aplicación móvil híbrida que deberá permitir a ISGlobal, administrar, gestionar y consultar la información referente a los usuarios de esta app y al público general interesado que haga uso de ella.

Esta aplicación trabajará bajo sistema operativo Android y permitirá mediante un formulario, almacenar en una base de datos los decibelios (dB) de un ruido captado por el micrófono del teléfono móvil durante 5 segundos con su correspondiente geolocalización. Estos datos irán asociados a la sensación del usuario la cual será indicada por el mismo en la app. Además, el usuario podrá ver los puntos de ruido geolocalizados por todos los usuarios en un mapa que se insertará en la aplicación con dicha finalidad.

Para que se puedan llevar a cabo todas estas funciones, los usuarios darán acceso al ID del dispositivo (UUID por sus siglas en inglés) y su ubicación (para dar a conocer la geolocalización de la contaminación acústica), a la vez que generarán datos referentes a la fecha, la hora, la captura de los decibelios, así como la percepción del usuario en base al ruido registrado. Toda la información generada, podrá ser consultada y gestionada por los técnicos de la empresa mediante una base de datos que será creada para dicho fin.

En cuanto al acceso a dicho programa, ISGlobal Noise (“IG-Noise” de aquí en adelante), se debe remarcar que este será restringido ya que se trata de una prueba piloto para la empresa, la cual en un futuro decidirá si dar acceso libre o no a futuros usuarios.

### 3.1.2. Alcance

Para llevar a cabo el diseño, desarrollo e implementación de una aplicación móvil para Sistema Operativo Android, se hará uso de Apache Cordova, un entorno de desarrollo *Open Source*.

La idea básica de IG-Noise es ser una aplicación que, permita realizar un registro de los ruidos captados por el usuario (en decibelios y geolocalizados) y ser ligado a sus sensaciones para finalmente ser transferida a una base de datos. Se centra principalmente en el área de Cataluña, aunque es fácilmente extensible a otras zonas ya que no se limitará su uso fuera de este espacio.

En este sentido, la información podrá ser visualizada y consultada por el usuario mediante el uso de un mapa, el cual permitirá el uso de zoom y escala, siempre adaptándose al terminal del usuario (siendo *responsive*).

A su vez, la aplicación tendrá flexibilidad de acceso a la información geoespacial, así como a las actualizaciones dinámicas que se vayan sucediendo de la misma.

### 3.1.3. Definiciones

A continuación, se explican algunos de los términos propios de este proyecto para mayor comprensión del lector.

- a. **Administrador del Sistema:** Persona encargada de ofrecer el soporte técnico y operativo a la aplicación.
- b. **Apache Cordova:** Framework para el desarrollo de aplicaciones móviles híbridas utilizando herramientas genéricas como HTML5, CSS y Javascript.
- c. **Aplicación:** Programa informático diseñado para facilitar al usuario la realización de determinados tipos de trabajo.
- d. **Base de datos:** Conjunto de información que pertenecen a un mismo ámbito o contexto almacenados sistemáticamente para su posterior uso.



- e. **Contaminación acústica:** Exceso de sonido que altera las condiciones normales del ambiente en una determinada zona, la cual puede causar grandes daños en la calidad de vida de las personas si no se controla bien o adecuadamente. Hace referencia al ruido (sonido excesivo y molesto), provocado por las actividades humanas (tráfico, industrias, locales de ocio, aviones, etc.) que produce efectos negativos sobre la salud auditiva, física y mental de los seres vivos.
- f. **CSS (Cascading Stylesheets):** Lenguaje de diseño gráfico utilizado para establecer el diseño visual de los documentos web escritos en HTML.
- g. **Decibelio:** Unidad de intensidad acústica equivalente a la décima parte de 1 belio y expresada como “dB”.
- h. **Framework (o entorno de desarrollo/trabajo):** Estructura conceptual y tecnológica de asistencia definida con artefactos o módulos concretos de software que puede servir para la organización y desarrollo de software y que, a su vez, puede incluir soporte de programas, bibliotecas y lenguaje interpretado, entre otros.
- i. **HTML (HyperText Markup Language):** Elemento de construcción más básico de una página web que se usa para crear y representar visualmente una página web. Determina el contenido, pero no su funcionalidad.
- j. **Javascript:** Lenguaje de programación orientado a objetos del lado del cliente que puede ser aplicado a un documento HTML para crear interactividad dinámica en los sitios web.
- k. **Jquery:** Framework optimizado para dispositivos táctiles (smartphones, tablets, ...) que permite trabajar con Javascript y CSS entre otros.  
es un Framework optimizado para dispositivos táctiles que está siendo desarrollado actualmente por el equipo de proyectos de jQuery
- l. **Leaflet:** Librería Javascript de código abierto que permite generar mapas interactivos optimizados para dispositivos móviles.
- m. **MySQL:** Sistema de Gestión de bases de datos relacional muy popular, especialmente para entornos de desarrollo web.
- n. **Open Source:** Software que se puede adquirir de manera gratuita y también libertad para modificar la fuente del programa sin restricciones de licencia.

- o. **PHP (Hypertext Preprocessor):** Lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
- p. **Sistema de Gestión de Base de Datos (SGBD):** Es un tipo de software específico que sirve de interfaz entre la base de datos, los usuarios y las aplicaciones que la utilizan.
- q. **Usuario:** Persona que utiliza la aplicación, en este caso, sin la necesidad de un proceso de autenticación.

#### 3.1.4. Capacidades y condiciones

Seguidamente se detallan las funciones y procesos que debe tener el presente proyecto.

- a. **Administración de la base de datos:** Los técnicos y administradores del sistema podrán gestionar la base de datos (agregar, modificar, eliminar o buscar información y registros) fácilmente.
- b. **Consultar información:** Proceso en el que un usuario procesa y busca información del sistema.
- c. **Consultar y visualizar registros de contaminación acústica:** Proceso que permite a los usuarios finales examinar los diferentes puntos en función de sus necesidades o interés mediante el uso de un mapa integrado en la aplicación.
- d. **Gestionar y administrar registro de sonidos:** Proceso que permite a los técnicos de la empresa administrar, crear y modificar los registros de sonidos geolocalizados (recibidos de los usuarios) en la base de datos, en caso necesario.
- e. **Publicar información:** Proceso que realizan los técnicos para actualizar, agregar o modificar información del sistema cada vez que lo estimen oportuno.
- f. **Realización de capturas de sonido:** Los usuarios darán acceso al micrófono de su dispositivo mediante la aplicación para reportar la media de decibelios registrado en un período limitado (en nuestro caso, 5 segundos) a través del formulario integrado.
- g. **Reportar registros acústicos:** Los usuarios podrán notificar ruidos o sonidos en cualquier punto geográfico a través del formulario integrado.

- h. **Permitir la geolocalización:** Los usuarios podrán ser localizados en el sistema si previamente han activado la “ubicación” en sus dispositivos móviles.
- i. **Exploración del mapa:** Los usuarios podrán consultar todos los datos que se ofrecerán en el sistema, así como hacer “zoom in” y “zoom out” cuando lo estimen oportuno.
- j. **Navegación:** Proceso por el cual una persona podrá utilizar el sistema.

### 3.1.5. Suposiciones y dependencias

Se asume que los requisitos descritos previamente satisfacen las necesidades del sistema y que, por tanto, no cambiarán durante el desarrollo de la aplicación.

Por lo tanto, el sistema será genérico para poderlo ejecutar en cualquier dispositivo móvil con sistema operativo Android independientemente de la versión que utilicen.

## 3.2. REQUERIMIENTOS FUNCIONALES

A continuación, se detallan cada uno de los requerimientos funcionales:

**RF1:** Mostrar la ubicación de cada uno de los puntos de contaminación acústica registrados por los usuarios en un mapa.

**RF2:** El usuario puede interactuar con en el mapa.

**RF3:** El usuario podrá reportar puntos de ruido mediante un formulario.

**RF4:** La app tendrá un formulario para notificar capturas de sonido y la percepción del usuario en base a ella.

**RF5:** El técnico puede crear, modificar o eliminar información alfanumérica y/o espacial.

**RF6:** El usuario deberá dar acceso a su micrófono para poder reportar registros de sonidos.

**RF7:** El usuario deberá dar acceso a su ubicación para su geolocalización.

**RF8:** El usuario deberá dar acceso al identificador único del móvil (UUID) si quiere hacer uso de esta aplicación.

**RF9:** Deberá tener la información al menos en idioma español.

**RF10:** La app debe proporcionar en todo momento el anonimato y la privacidad del usuario.

**RF11:** El usuario podrá realizar búsquedas en el mapa según su área de interés.

**RF12:** Al apretar el botón “capturar posición y decibelios” en el formulario, la aplicación accederá a la localización del usuario y al registro de la media de decibelios capturados durante 5 segundos en dicha ubicación.

**RF13:** Al pulsar el botón “enviar datos”, se enviará la información al servidor y este retornará un mensaje para notificar al usuario que se ha realizado correctamente. Posteriormente, se retornará al usuario al menú principal.

### 3.3. REQUERIMIENTOS NO FUNCIONALES

En cuanto a los requerimientos no funcionales, la aplicación:

- Será compatible con la mayoría de versiones y sistemas Android.
- Tiene que ser estable y controlar que no se produzcan errores en su uso.
- Deberá ser lo más ligera posible para ocupar el mínimo espacio en la memoria del teléfono.
- Deberá tener una interfaz gráfica amigable para el usuario.
- Pesará poco, motivo por el cual el dispositivo móvil deberá tener al menos 11 MB libres para poder proceder a su instalación, así como un procesador mínimo de 512 MB (aunque recomendable 1GB).
- Deberá cumplir con los estándares mínimos de seguridad y legales, en términos de confidencialidad, protección de datos, permisos, licencias, condiciones de uso e información.

### 3.4. CARACTERÍSTICAS DE LOS USUARIOS

El sistema estará compuesto por 2 tipos de usuarios que interactuarán y lo gestionarán: visitante y administrador del sistema. A continuación, se detalla brevemente el perfil específico de cada uno de ellos con el sistema:

- **Visitante:** Persona que interactuará con la aplicación y que deberá tener un conocimiento básico sobre navegación y uso de apps móviles.
- **Administrador del sistema:** Usuario técnico con gran conocimiento del sistema, encargado de gestionar el contenido geoespacial y alfanumérico de forma responsable.

### 3.5. DIAGRAMA GENERAL DE LOS CASOS DE USO

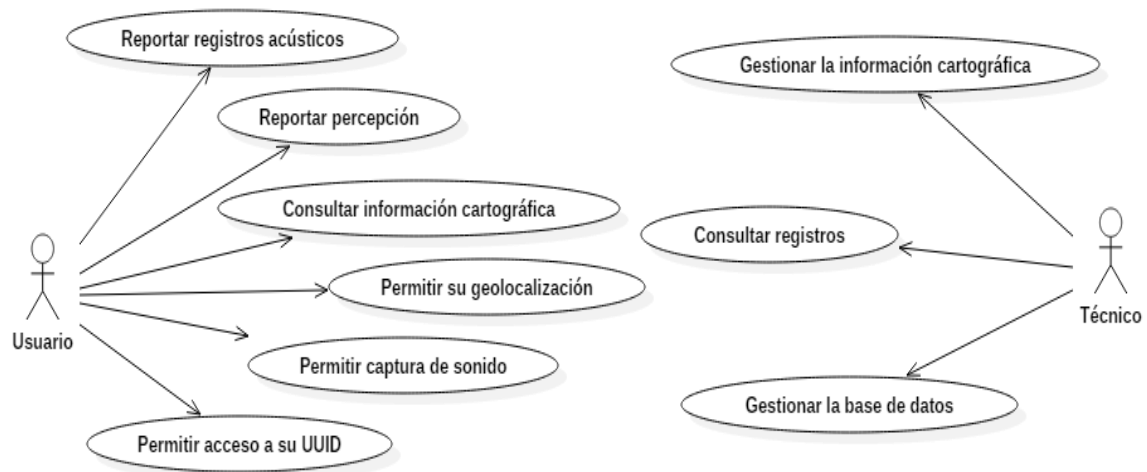


FIGURA 2. DIAGRAMA GENERAL DE LOS CASOS DE USO. FUENTE: ELABORACIÓN PROPIA

El diagrama anterior, muestra de forma generalizada las acciones posibles que se pueden llevar a cabo por parte de los agentes finales implicados en este proyecto: el usuario y el técnico.

Pretende dar una visión de conjunto para saber realmente donde están los límites del proyecto y acotar en caso de necesidad si este resultara demasiado ambicioso y extenso. Se puede ver de forma más detallada en el apartado 4.1.

### 3.6. RESTRICCIONES

A continuación, se presentan las principales restricciones:

- a. La aplicación se desarrollará mediante software *open source* y, por tanto, no se deberá pagar por su uso (será gratis). Podrá ser descargada por cualquier usuario a través de la página web de la entidad o del Play Store (en caso de implantación definitiva).
- b. Para utilizar esta aplicación será necesario que el usuario disponga de un Smartphone con sistema operativo Android.
- c. Todo el contenido que se ponga a disposición del usuario debe de estar en castellano. A posteriori, se implementarían también en catalán e inglés dado el carácter internacional de la institución ISGlobal.
- d. Para poder hacer pleno uso de la aplicación, el usuario debe disponer de una conexión a internet.
- e. Para garantizar la consistencia y correcta funcionalidad de la aplicación, esta será sometida a diferentes pruebas previamente a su lanzamiento.

### 3.7. SOLUCIONES EXISTENTES

En el presente apartado, realizamos un estudio analítico de las diferentes opciones existentes para iniciarnos en la evaluación del ruido ambiental.

Tratamos de buscar herramientas que den información a partir de un registro de audio el cual pueda ser incorporado fácilmente a una app con posterioridad.

La meta a alcanzar es la grabación de archivos de audio desde la app, pasar el archivo generado a una función, módulo o servicio para que se nos dé información acerca del mismo (los decibelios) y, una vez logrado, almacenarlo en una base de datos y geolocalizarlo en un mapa contenido en la aplicación.

Si no fuera posible utilizar ninguno de estos medios, se podría estudiar el uso de funciones Javascript para el análisis de sonido o, servicios web como última vía. Es decir, se trata de considerar todas las alternativas, incluso mediante servicios externos por si no hubiera mejor solución a nuestro alcance. Lo esencial es demostrar la funcionalidad de este prototipo.

### 3.7.1. Análisis de las aplicaciones y servicios

En este punto, se va a realizar un estudio de los medios existentes para captar y evaluar el ruido mediante una app. Esto nos permitirá incorporar dichas soluciones en una app propia que seguirá el mismo fin.

A su vez, se van a estudiar también las diferentes opciones que se tienen para gestionar los datos de la aplicación, así como los lenguajes de programación necesarios para desarrollarla. Este análisis servirá para determinar el diseño de la aplicación.

Entre las opciones existentes en el mercado y a nuestro alcance, se dispone por un lado de los códigos fuentes de aplicaciones nativas Android provenientes de sus APK y, por otro, de plugins que se pueden utilizar en nuestro código mediante HTML5, CSS y JS si nuestra finalidad es crear una app híbrida.

#### 3.7.1.1. Aplicaciones nativas

Entre las aplicaciones ya existentes en el Play Store, vemos que las que más nos pueden interesar por su adecuación a nuestro propósito serían NoiseTube, Sound Meter, Noise Pollution Monitor y Noise Capture (principalmente programadas en Java, lenguaje principal en Android Studio). Es importante saber, tal y como se muestra a continuación que, en ninguna de ellas se tiene en cuenta la **percepción del usuario**. Justamente, este es el **valor añadido que debe tener nuestra aplicación**.

Entre todas las aplicaciones del Play Store que han sido investigadas, se analizan las siguientes como posibles soluciones, gracias a la descarga y uso de las mismas, así como al estudio y estructura de su código interno con la ayuda del Software “Raccoon” para descargar las APK en nuestro ordenador y, “APK Easy Tool” para decompilar dichos códigos y estudiarlos.

**NoiseTube**.(figura 20). Dentro de esta app, tenemos el medidor de ruido Noise Measure, el cual nos recomienda una grabación mínima de 30 segundos para su correcto funcionamiento. Una vez registrado, este nos da su duración, así como las mediciones mínimas, máximas y promedio en decibelios.

Nos permite consultar nuestras grabaciones en el apartado “My tracks”, así como acceder a un mapa en el cual podemos estar geolocalizados si le damos permiso a la aplicación. Todos nuestros datos son almacenados por defecto en la memoria local del dispositivo móvil y, a su vez, pueden ser almacenados en su propia página web NoiseTube.net.

**Sound Meter – Sonómetro** (figura 21). Solo permite ver el ruido, así como el tiempo transcurrido, pero no registrarlo. También se puede ver un gráfico a tiempo real con la variación del sonido (pulsando M).

**Noise Pollution Monitor - City Noise** (figura 22). Nos muestra los sonidos a tiempo real en una especie de medidor con diferentes franjas según la cantidad de ruido existente. A través del apartado Noise Data, se nos permite visualizar y consultar nuestros propios datos (My Data) y a nivel de ciudades, pero en este último caso solo las pertenecientes a la India.

**Noise Capture** (figura 23). Te permite analizar el ruido a tiempo real, así como grabarlo. A su vez, se puede analizar mediante espectro, espectrograma y mapa. En este último caso, se nos da un mapa muy bien trabajado, con sonidos geolocalizados y una leyenda muy atractiva. Se adapta a lo que queremos conseguir. A su vez nos permite describirlos brevemente, así como ver los resultados, nuestro historial y realizar calibración en caso necesario lo cual resulta muy interesante.

De todas las opciones existentes, Noise Capture es la que al estudiar su código fuente (GITHUB – Noise Capture) más se asemeja a la idea de aplicación que teníamos inicialmente, aunque con ciertas variaciones.



### 3.7.1.2. Plugins

En el mercado actual, se dispone también de ciertos plugins en el caso de optar por la creación de una WebApp a través de *frameworks open source* para su desarrollo tales como Cordova o Phonegap.

A nivel de plugins oficiales, disponemos del **Plugin Media Capture** el cual nos muestra en su documentación (CORDOVA – Plugin Media Capture) que permite únicamente capturar audio, imágenes y video, pero no procesarlos, lo cual a priori no nos interesa tanto para cumplir nuestra meta.

Tras la búsqueda de otras opciones, encontramos un plugin llamado **cordova-plugin-dbmeter** (GITHUB – Plugin dbmeter), el cual nos permite ver a tiempo real el ruido en decibelios que registra nuestro dispositivo móvil, independientemente de lo fiable o no que pueda resultar con respecto a la opción de un sonómetro profesional.

Esta se nos presenta, como **la mejor de las opciones** para capturar el ruido durante un tiempo determinado, aunque deberemos modificar y añadir ligeramente el código de la función Javascript si deseamos crear registros de ello.

Si optamos por esta última opción, deberemos tener presente que necesitaremos utilizar el plugin de geolocalización (CORDOVA – Plugin Geolocation) el cual ya se implementó tanto en el proyecto de segundo semestre como en el módulo de Desarrollo de Geoapps.

## 4. DISEÑO FUNCIONAL

La funcionalidad y el diseño visual son dos aspectos que cobran mucho interés en el proceso de creación de las aplicaciones móviles. En este apartado, nos centraremos en ellos ya que nos servirán para establecer como deberán reaccionar los dispositivos ante determinadas acciones del usuario, actuando también en consecuencia para evitar (o minimizar) los errores que se puedan ir produciendo. Para cubrir estos aspectos, nos centraremos especialmente en los casos de uso (4.1.) y en la propuesta de una arquitectura de la solución (4.2.).

### 4.1. CASOS DE USO

Seguidamente se presentan los principales casos de uso de IG-Noise:

CASO DE USO 1: Permitir la geolocalización			
Versión	1.0.	Fecha	09/05/2018
Descripción	Este caso de uso describe como el usuario permite su geolocalización a la aplicación con el fin de situar el sonido registrado en una posición geográfica (en el formulario) y facilitar su posterior visualización en el mapa.		
Actores	Todos los usuarios		
Precondición	-		
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario da acceso a la aplicación para registrar su geolocalización (y la del sonido) en la base de datos mediante el uso del formulario.</li> <li>2. El usuario abre el mapa.</li> <li>3. El usuario visualiza todos los puntos geolocalizados.</li> </ol>		
Flujos alternativos	El usuario no permite su geolocalización. La app no sitúa al usuario haciendo que las funciones sean más limitadas y que no se pueda cumplimentar todos los campos del formulario, los cuales son obligatorios.		
Postcondición	Si el usuario pulsa otro punto del mapa, se muestra la información del punto seleccionado. Si en cambio, pulsa al mapa (no en un punto), el popup se cierra.		
Diagrama	<pre> graph LR     Usuario((Usuario)) --&gt; Geolocalización((Geolocalización))     Geolocalización --&gt; LaPermite((La permite))     Geolocalización --&gt; NoLaPermite((No la permite))     LaPermite --&gt; AppRegistra((La app registra los sonidos geolocalizados mediante el uso del formulario))     AppRegistra --&gt; UsuarioAbreMapa((El usuario abre el mapa y ve los puntos geolocalizados que previamente se han registrado en el formulario))     NoLaPermite --&gt; AppNoSitua((La app no sitúa el sonido))     AppNoSitua --&gt; FuncionesLimitadas((Funciones son más limitadas))     </pre>		

FIGURA 3. CASO DE USO "PERMITIR LA GEOLOCALIZACIÓN". FUENTE: ELABORACIÓN PROPIA

CASO DE USO 2: Reportar registros acústicos			
Versión	1.0.	Fecha	09/05/2018
Descripción	Este caso de uso describe como el usuario reporta un registro acústico en una posición geográfica del lugar en que se encuentra.		
Actores	Todos los usuarios		
Precondición	Que esté habilitada la ubicación del usuario y que la aplicación tenga acceso al UUID y micrófono del dispositivo móvil.		
Flujo principal	<ol style="list-style-type: none"> <li>1. Se precargan en el menú del formulario el UUID, la fecha y la hora.</li> <li>2. El usuario pulsa el botón "capturar posición y decibelios".</li> <li>3. Se rellena automáticamente las coordenadas geográficas del usuario (latitud y longitud) y, 5 segundos más tarde se incorpora también en su correspondiente casilla el promedio de decibelios del sonido captado.</li> <li>4. El usuario selecciona la respuesta que cree más conveniente.</li> <li>4. El usuario pulsa el botón "Enviar Datos".</li> <li>5. Se envía la información a la BBDD de la institución.</li> <li>5. Se notifica al usuario que los datos del formulario se han guardado correctamente.</li> <li>6. Se devuelve al usuario a la página inicial que contiene el menú principal de la aplicación.</li> </ol>		
Flujos alternativos	<ol style="list-style-type: none"> <li>1. El usuario no activa su ubicación. Cuando pulse el botón "capturar sonido y decibelios", no se completarán los campos "latitud y longitud" y como consecuencia de ello, el usuario recibirá en su pantalla el siguiente mensaje: "Quedan campos por completar" y los datos no se enviarán hasta que se hayan completado todos los campos.</li> <li>2. El usuario pulsa el botón de retroceder antes del envío del formulario. En este caso, el usuario vuelve a la página anterior y los datos del formulario no son enviados.</li> <li>3. El usuario no pulsa el botón Enviar Datos. No se envía la incidencia y se queda la pantalla de la aplicación en modo Stand by hasta que el usuario haga alguna acción.</li> </ol>		
Postcondición	Se devuelve al usuario a la pantalla principal de la aplicación una vez enviado el formulario.		
Diagrama	<pre> graph LR     Usuario((Usuario)) --&gt; A([Se abre el formulario y se carga UUID fecha y hora])     A --&gt; B([Pulsa botón Capturar Posición y Decibelios])     A --&gt; C([No pulsa botón Capturar Posición y Decibelios])     B --&gt; D([Se rellena lat. long. y dB])     C --&gt; E([No se rellena lat. long. y dB])     D --&gt; F([Selecciona respuesta])     E --&gt; F     F --&gt; G([Aprieta al botón enviar])     F --&gt; H([No aprieta al botón enviar])     G --&gt; I([Se envía la información])     H --&gt; J([No se envía la información])     I --&gt; K([Se notifica que se ha enviado correctamente])     </pre>		

FIGURA 4. CASO DE USO "REPORTAR REGISTROS ACÚSTICOS". FUENTE: ELAB. PROPIA

CASO DE USO 3: Interactuar con el mapa			
Versión	1.0.	Fecha	09/05/2018
Descripción	Este caso de uso describe como el usuario interactúa con el mapa de la aplicación.		
Actores	Todos los usuarios		
Precondición	Que haya los botones de Zoom y Pan activados.		
Flujo principal	<p>1. El usuario realiza un Zoom In (A), Zoom Out (B), Pan (C) o pulsa en un punto registrado (D)...</p> <p>2A. El usuario se aproxima a la zona de interés.</p> <p>2B. El usuario se aleja de la zona de interés.</p> <p>2C. El usuario se mueve a la zona de interés.</p> <p>2D. El usuario presiona a un punto registrado.</p>		
Flujos alternativos	El usuario no realiza ninguna acción en el mapa. No se produce ninguna alteración en el funcionamiento de la app.		
Postcondición	El mapa (A) se muestra a mayor escala, (B) a menor escala. (C) se mueve en función de lo que el usuario le haya demandado o (D) se abre un popup con la información referente a la fecha, las coordenadas, los decibelios y la percepción.		
Diagrama	<pre> graph LR     Usuario((Usuario)) --&gt; Zoom((Zoom))     Usuario --&gt; Pan((Pan))     Usuario --&gt; Punto((Pulsa un punto registrado))     Zoom --&gt; ZoomIn((Zoom in))     Zoom --&gt; ZoomOut((Zoom out))     ZoomIn --&gt; ResultZoomIn([Se aproxima a la zona de interés. El mapa se muestra a mayor escala])     ZoomOut --&gt; ResultZoomOut([Se aleja de la zona de interés. El mapa se muestra a menor escala])     Pan --&gt; ResultPan([Se mueve a la zona de interés])     Punto --&gt; ResultPunto([Se abre un popup que muestra la información almacenada])     </pre>		

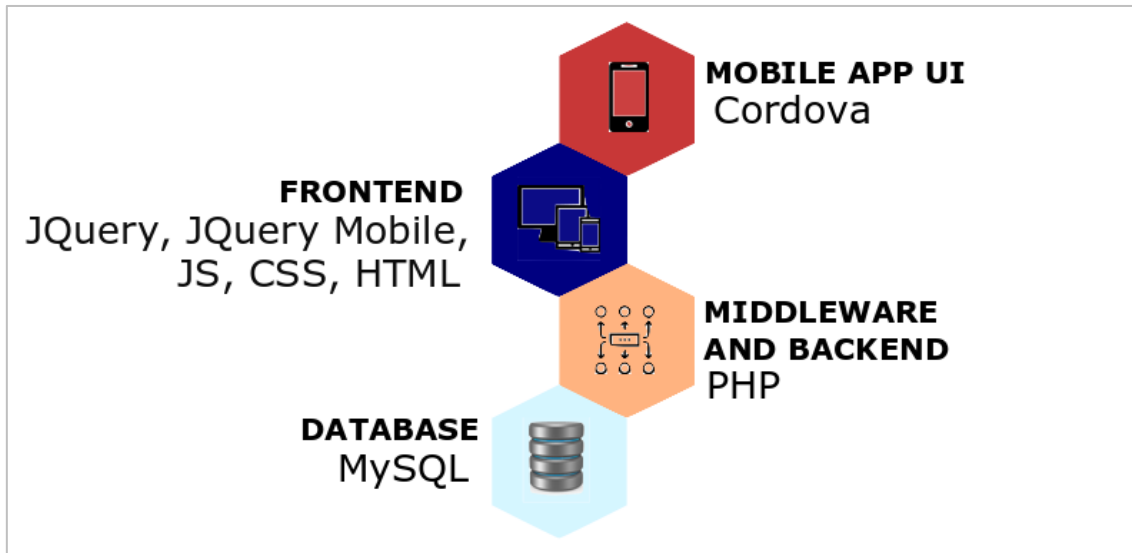
FIGURA 5. CASO DE USO “INTERACTUAR CON EL MAPA”. FUENTE: ELABORACIÓN PROPIA

CASO DE USO 4: Gestionar la base de datos			
Versión	1.0.	Fecha	15/03/2018
Descripción	Este caso de uso describe como el técnico gestiona las bases de datos alfanumérica.		
Actores	Técnicos		
Precondición	Que los técnicos tengan pleno acceso a las BBDD.		
Flujo principal	<p>1. El técnico pulsa cualquiera de las opciones disponibles: (A) Crear Registros, (B) Modificar Registros y (C) Eliminar Registros.</p> <p>2A. El técnico puede añadir cualquier registro nuevo.</p> <p>2B. El técnico puede cambiar o alterar cualquier registro en caso necesario.</p> <p>2C. El técnico puede suprimir cualquier registro.</p> <p>3. Una vez finalizada cualquiera de las acciones, el técnico pulsa el botón Aplicar.</p>		
Flujos alternativos	El técnico no pulsa el botón Aplicar. No se producen cambios ni alteraciones en la BBDD. En este caso, no aparecerá ningún mensaje ya que se entiende que la acción está inacabada y se puede retomar más tarde.		
Postcondición	Se regresa a la pantalla de gestión de bases de datos inicial al técnico.		
Diagrama	<pre> graph LR     Técnico((Técnico)) --&gt; GestiónBBDD((Gestión BBDD))     GestiónBBDD --&gt; CreaRegistros((Crea registros))     GestiónBBDD --&gt; ModificaRegistros((Modifica registros))     GestiónBBDD --&gt; EliminaRegistros((Elimina registros))         </pre>		

FIGURA 6. CASO DE USO "GESTIONAR LA BASE DE DATOS". FUENTE: ELABORACIÓN PROPIA

## 4.2. ARQUITECTURA DE LA SOLUCIÓN

Tras haber analizado las diferentes opciones existentes en el mercado, sabiendo el conocimiento adquirido durante el desarrollo del máster y tras haber estudiado las necesidades que tenemos para llevar a cabo con éxito nuestro proyecto, se ha determinado a criterio propio cual es la mejor solución tal como se muestra en la figura 7.



**FIGURA 7. ARQUITECTURA DE LA SOLUCIÓN PROPUESTA. FUENTE: ELABORACIÓN PROPIA**

La **interfaz de usuario de la app móvil** se realizará con Apache Cordova (ver 5.1), un popular entorno de desarrollo para aplicaciones móviles híbridas con código abierto el cual se puede utilizar fácilmente con un cierto conocimiento de programación web. Ello nos permitirá crear la conocida WebApp.

El **FrontEnd** (tecnologías utilizadas en el lado del cliente/navegador web), está centrado principalmente en tres lenguajes muy reconocidos en el mundo de la navegación web tales como HTML, CSS y Javascript. Con ello, se consigue trabajar la parte visual de la app y nos centramos en la denominada User Experience (Experiencia del Usuario) para que esta se encuentre dispuesta de forma ordenada y resulte atractiva, sencilla e intuitiva.

Dicho de otro modo, nos centramos en el diseño (visualmente atractivo), funcionalidad, creación de una navegación cómoda y procesamiento de ciertas solicitudes (principalmente Ajax y JSON). En este caso, se hará uso de las bibliotecas JQuery y JQuery Mobile ya que nos facilitará enormemente nuestra tarea.

El **Middleware** y, especialmente el **BackEnd** (que se encuentra en el lado del servidor) se trabaja con varios lenguajes como PHP, Python y Java entre otros, aunque nosotros nos centraremos en el primero de ellos.

Con el uso de PHP, interactuaremos con la base de datos mediante una serie de peticiones (o *request*) que haremos con la app, para así obtener un resultado concreto el cual resultaría visible en el denominado FrontEnd (parte visible de la app por parte del usuario). Para esta parte, utilizamos PHP que nos ayudará a crear contenido de forma dinámica e interactuar con la BBDD, siempre tratando de mantener nuestro servidor seguro.

Finalmente, tenemos la **Base de Datos** (Database en inglés) que “es un conjunto de información y de datos organizados de forma sistemática”. En los últimos tiempos ha pasado de un sistema analógico y trabajar únicamente en local (en una red concreta) a sistemas cada vez más virtuales (el caso que nos ocupa), ya que la base de datos se encuentra en la red o nube y resulta accesible desde cualquier lugar del mundo en caso de que así se quiera.

Las bases de datos resultan interesantes para la realización de consultas a través de la organización lógica de los datos cuando se realizan a través de servidores y sistemas de gestión de información.

En general, se utilizan los sistemas gestores de bases de datos (SGBD o DBMS, en inglés) que tienen como objetivo la organización y estructuración de la base de datos. En nuestro caso particular, se trata de bases de datos relacionales en las cuales se organizan los datos y la información en tablas bidimensionales (campo, registro y dato) y que generalmente utilizan el lenguaje de programación SQL (Structured Query Language). En este caso en particular, se trabajará con MySQL, uno de los SGBD más populares y que, aun no habiendo hecho uso de él con anterioridad, resulta muy sencillo de utilizar. Y mejor aún, dado el uso no intensivo que se va a hacer de él, se va a utilizar de forma totalmente gratuita.

## 5. METODOLOGÍA E IMPLEMENTACIÓN

En esta fase nos centramos en la implementación del sistema y la programación de la aplicación, ya que en etapas anteriores ya se ha realizado el análisis de los requerimientos específicos necesarios para llevar a cabo nuestro proyecto, el estudio de algunas soluciones existentes en el mercado y el diseño de la aplicación a desarrollar. Dicho de otro modo, se han ejecutado anteriormente las tareas necesarias para proceder con garantías a la realización del paso actual.

### 5.1. DESCRIPCIÓN DEL ENTORNO DE DESARROLLO

Para la creación e implementación de nuestro proyecto, se ha optado por el uso de **Apache Cordova** (ver figura 8), un entorno de desarrollo móvil de código abierto. Lo interesante de este IDE (*Integrated Development Environment*) es que permite el uso de tecnologías estándar web (HTML5, CSS3 y Javascript) para el desarrollo multiplataforma, lo que se traduce en el no uso de lenguajes de desarrollo nativo, un punto a tener muy en cuenta, especialmente entre aquellos que no hemos utilizado nunca este tipo de plataformas de desarrollo.

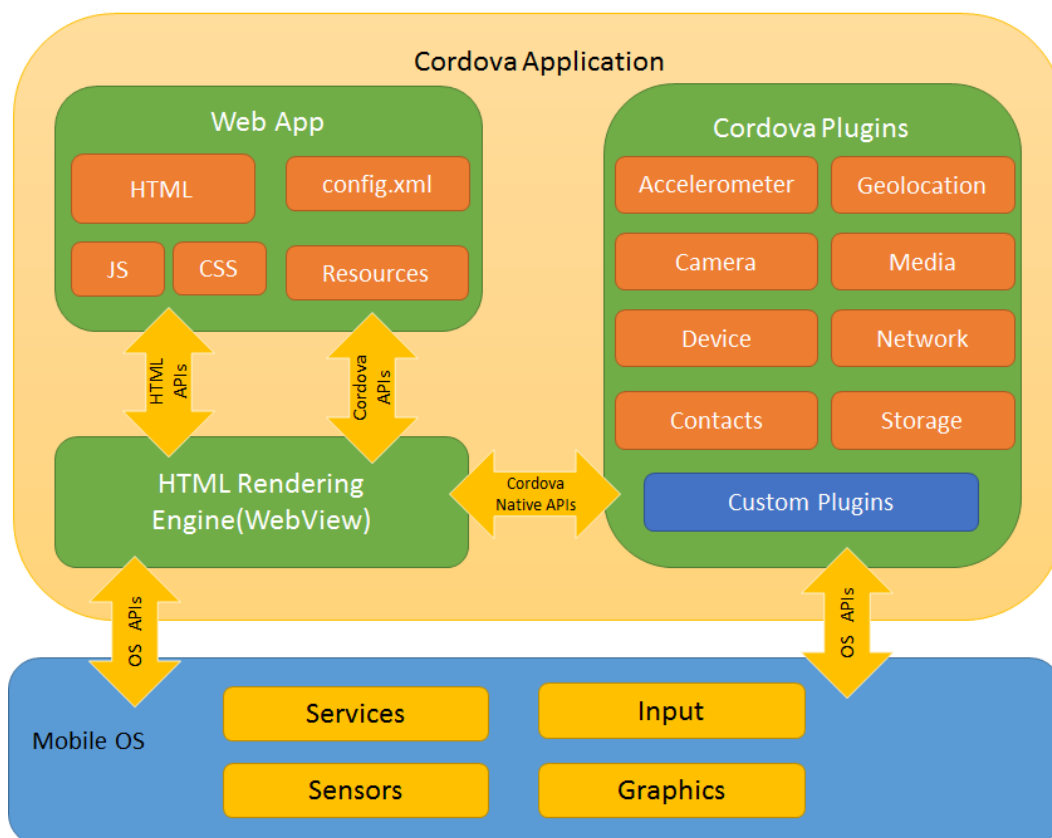


FIGURA 8. ARQUITECTURA DE APLICACIÓN DE CORDOVA. FUENTE: [HTTPS://CORDOVA.APACHE.ORG](https://cordova.apache.org)



En este tipo de aplicaciones se dependerá de enlaces estándares API para acceder por ejemplo al uso de la cámara o el micrófono de los dispositivos móviles.

Se puede decir que las principales ventajas de Apache Cordova son:

- El desarrollo de aplicaciones móviles multiplataforma a través de un solo programa sin necesidad de utilizar el lenguaje propio de cada plataforma en la que se desea implementar.
- Los conocimientos en desarrollo web son suficientes para implementar una aplicación web o WebApp que se pueda utilizar en dispositivos móviles.
- El desarrollador acabará implementando una app móvil híbrida ya que con el uso de este IDE mezclará componentes propios de una aplicación nativa que tenga acceso a las API de nivel de dispositivo, con otros propios de las webs.

En cuanto a los componentes básicos, se puede decir que este tiene principalmente el `config.xml` (archivo donde se proporciona información acerca de la aplicación y se especifican ciertos parámetros de la app), el `index.html` (propio del desarrollo de páginas web que aquí se utiliza como página inicial de la aplicación el cual es trabajado principalmente con CSS y JavaScript ) y, una interfaz de `plugins` (que permite invocar un código Javascript, el cual suele servir como vía de comunicación entre Cordova y algunos componentes nativos. En el manual 1 se explican algunos de los primeros pasos para hacer uso de Apache Cordova.

## 5.2. LA BASE DE DATOS

En este punto, se procede a crear la base de datos con un hospedaje de web gratuito (Awardspace), ya que previamente (manual 2 y 3) se ha hecho con éxito en localhost (ordenador o dispositivo utilizado mediante una IP determinada) mediante el uso de XAMPP.

Procedemos a explicar la creación y despliegue en servidor web ya que es el que nos ayuda a cumplir con uno de nuestros propósitos: que la BBDD sea accesible desde cualquier dispositivo independientemente del lugar en el que se encuentre.

Los pasos a seguir son los siguientes:

- 1) Registro en un hospedaje de web gratuito. En nuestro caso, awardspace.com (figura 24).
- 2) Accedemos con los datos que hemos registrado, usuario y contraseña (figura 25), al panel de control asociado a nuestra cuenta.
- 3) Una vez ubicado en el menú principal, nos dirigimos al módulo de Herramientas Avanzadas y seleccionamos Gestor de Base de Datos (figura 26).
- 4) En esta nueva pantalla, nos viene seleccionado “Crear Base de Datos MySQL” que es el que nos interesa, aunque podríamos optar por PostgreSQL (trabajado durante el primer cuatrimestre del Máster) si así lo quisiéramos. Cumplimentamos las casillas de Nombre de Base de Datos y Contraseña de la Base de Datos (figura 27) y seleccionamos “Crear Base de Datos”.
- 5) Con el paso anterior se nos habrá creado la BBDD. Ahora, simplemente hay que ir al apartado “Todas las Bases de Datos” donde nos aparecerá la que hemos creado (figura 28). En este caso, se llama “2740980\_ignoise” y debemos acceder a ella presionando “phpMyAdmin 4” dentro de la columna Administración.
- 6) Accedemos a la pestaña “Importar” en el menú de phpMyAdmin, donde acto seguido pulsaremos “Seleccionar Archivo” (figura 29) y cargaremos el SQL que hemos hecho para la creación de la tabla y su correspondiente configuración el cual tiene el siguiente aspecto:

```
1  
2 CREATE TABLE `datos` (  
3   `iddispositivo` varchar(155) COLLATE utf8_spanish_ci NOT NULL,  
4   `fecha` varchar(155) COLLATE utf8_spanish_ci NOT NULL,  
5   `hora` varchar(8) COLLATE utf8_spanish_ci NOT NULL,  
6   `latitud` float NOT NULL,  
7   `longitud` float NOT NULL,  
8   `decibelios` decimal(5,2) NOT NULL,  
9   `percepcion` varchar(155) COLLATE utf8_spanish_ci NOT NULL  
10  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;  
11
```

FIGURA 9. SENTENCIA SQL A EJECUTAR PARA LA CREACIÓN DE LA TABLA "DATOS".  
FUENTE: ELABORACIÓN PROPIA

Para que realmente se suba, deberemos pulsar Continuar y se nos indicará Importación ejecutada exitosamente.

```

✔ Importación ejecutada exitosamente. 1 consulta ejecutada. (Crear_Tabla_Datos_ignoise.sql)

✔ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.1768 segundos.)

CREATE TABLE `datos` (
  `iddispositivo` varchar(155) COLLATE utf8_spanish_ci NOT NULL,
  `fecha` varchar(155) COLLATE utf8_spanish_ci NOT NULL,
  `hora` varchar(8) COLLATE utf8_spanish_ci NOT NULL,
  `latitud` float NOT NULL,
  `longitud` float NOT NULL,
  `decibelios` decimal(5,2) NOT NULL,
  `percepcion` varchar(155) COLLATE utf8_spanish_ci NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
  
```

**FIGURA 10. MENSAJE DEVUELTO TRAS SENTENCIA SQL EJECUTADA CORRECTAMENTE. FUENTE: ELABORACIÓN PROPIA**

Para este sexto paso, se podría optar también por la creación manual de una tabla, especificando y configurando cada una de las columnas y sus propiedades, tal como se hizo en las pruebas realizadas en localhost.

- 7) Para ver el aspecto de nuestra tabla pulsaremos en la pestaña Estructura y seguidamente en Datos (figura 30), la tabla que hemos creado.
- 8) Observamos de este modo, que nuestra tabla se muestra correctamente.

iddispositivo	fecha	hora	latitud	longitud	decibelios	percepcion

**FIGURA 11. MUESTRA DE LA TABLA DE DATOS SIN REGISTROS. FUENTE: ELAB. PROPIA**

Debemos tener presente que en estos momentos se encuentra sin ningún registro.

Con todos los pasos que se han seguido anteriormente, podemos decir que la tabla **Datos** y nuestra base de datos **2740980\_ignoise** han sido creadas correctamente y se encuentran listas para su uso.

### 5.3. APLICACIÓN PARA LA OBTENCIÓN DE DATOS

Con objeto de explicar de forma clara la obtención de los datos, nos centraremos en explicar el FrontEnd (en nuestro caso, JQuery, JQuery Mobile, CSS, HTML y Javascript) y, el Middleware y BackEnd (centrado en el uso de PHP).

#### 5.3.1. FrontEnd. Lado Cliente

En este punto, tratamos **inicialmente** de obtener toda la información necesaria para rellenar los campos (archivo **form.html**) creados en nuestra BBDD: iddispositivo, fecha, hora, latitud, longitud, decibelios, percepción. Para entender mejor como se ha realizado, los separaremos por información obtenida al (1) iniciar formulario, (2) pulsar botón y (3) manual.

Dentro de la información que se obtiene al abrir el formulario, tenemos el iddispositivo, la fecha y la hora.

Nos centramos en obtener el **UUID** (o Identificador Único de Dispositivo) porque se quiere ante todo mantener la privacidad del usuario y se cree que no es necesario ningún registro de los usuarios para hacerles un seguimiento. Lo que se quiere en realidad es almacenar toda la información del ruido cruzada con la percepción para ver a medio y largo plazo si hay una relación directa (que está demostrado que sí) y que resultados concluyentes se pueden sacar al respecto. Para ello, se instala **cordova-plugin-device** (como se indica en su página web) y lo configuramos en el form.html (figuras 31-32), la página de formulario que hemos creado para este proyecto.

Para la obtención de **la fecha**, configuramos el input correspondiente en el HTML así como la ejecución de una función JS, obtenida de las muchísimas que hay en internet (figuras 33-34).

Para conseguir **la hora**, hacemos exactamente lo mismo que con la fecha, programar el HTML y el JS correspondiente (figuras 35-36).

Al pulsar un botón “onclick”, en nuestro formulario el correspondiente a “capturar posición y decibelios”, lo que hacemos es ejecutar dos funciones de manera simultánea:

```
<label for="latlon" style="text-align:left">Latitud y Longitud:</label>
<input name="latitud" id="text-latitud" value="" placeholder="Latitud según geoposicionamiento" type="text" />
<input name="longitud" id="text-longitud" value="" placeholder="Longitud según geoposicionamiento" type="text" />
<label for="decibelios" style="text-align:left">El promedio de decibelios es:</label>
<input name="decibelios" id="text-ruido" value="" placeholder="Aquí se almacenan los dB registrados" type="text" />
<a onclick="getCurrentPosition();getDBMeter();" class="ui-btn ui-corner-all ui-shadow ui-shadow-icon ui-btn-a ui-mini">Capturar posición y decibelios</a>
```

**FIGURA 12. PARTE DE CÓDIGO FUENTE REFERENTE A COORDENADAS Y DB EN FORM.HTML. FUENTE: ELABORACIÓN PROPIA**

Para lograr **las coordenadas del usuario** (latitud y longitud), hacemos uso de la función JS utilizada para la elaboración del proyecto de segundo semestre con el profesor Nacho Ferrero (figura 37).

Por otro lado, para conocer **los decibelios** (dB) se ha tenido que hacer uso del único plugin de Cordova existente (dbmeter) el cual ha sido creado por el usuario “akofman” (figura 38). El código ha sido modificado y adaptado para que haga un registro de dB durante cinco segundos en intervalos de un segundo (figura 39) y, tome el promedio de este (figura 40).

Para la última de las opciones, obtener información de forma manual, se ha optado por implementar Radio Buttons (de JQuery Mobile), es decir, proveer una lista de opciones donde solo una opción pueda ser seleccionada (figura 41). Para obtener la **percepción del usuario** acerca del ruido, se le plantea la siguiente pregunta: *¿Cómo describiría el sonido que usted escucha en este lugar?* El usuario deberá seleccionar obligatoriamente una de las siguientes opciones: *Muy bueno, Bueno, Ni Bueno ni Malo, Malo o Muy Malo*.

Tras haber conseguido configurar las diferentes funciones asociadas a los campos del formulario, **nos centramos en cómo enviar las variables y la información asociada a ellas** para que después pueda ser procesada desde el lado del servidor mediante un archivo PHP que servirá como nexo entre la app y la base de datos.

Para conseguirlo, incluimos un script en el `form.html` que se ha ido trabajando en este apartado. Lo que hacemos es definir las variables y asociarlas al id del campo rellenado en el formulario e indicarle posteriormente que son cadenas de datos (dataString). Seguidamente se indica que ninguno de los campos puede estar vacío mediante el uso de "length > 0" (figura 42).

Y finalmente, lo que hacemos mediante una función Ajax es lanzar un POST a un archivo PHP que se encontrará en el lado del servidor (figura 43) para que inserte (INSERT) toda esa información en la BBDD. En caso satisfactorio, nos enviará una alerta de mensaje que nos dirá "¡Datos guardados con éxito!" y nos enviará a la página inicial de la app, `index.html`, donde hay un menú para seleccionar Mapa o Formulario. De no ser así, nos enviará "error" si no se han podido enviar los datos correctamente o "Completa el formulario" en caso de que haya algún campo vacío.

Con todo ello, ya habremos conseguido trabajar la obtención de los datos desde esta parte FrontEnd.

### 5.3.2. Middleware y BackEnd. Lado Servidor

En este punto procedemos a configurar el PHP en el lado del servidor. Estos archivos deben ser almacenados en nuestro servidor web. Desde form.html se llama a `AWS_insert.php` el cual tiene el aspecto siguiente:

```

1 <?php
2 header("Access-Control-Allow-Origin: *");
3 include "AWS_db.php";
4 if(isset($_POST['insert']))
5 {
6     $iddispositivo=$_POST['iddispositivo'];
7     $fecha=$_POST['fecha'];
8     $hora=$_POST['hora'];
9     $latitud=$_POST['latitud'];
10    $longitud=$_POST['longitud'];
11    $decibelios=$_POST['decibelios'];
12    $percepcion=$_POST['percepcion'];
13
14    $sql="INSERT INTO datos VALUES ('$iddispositivo',
15                                     '$fecha',
16                                     '$hora',
17                                     '$latitud',
18                                     '$longitud',
19                                     '$decibelios',
20                                     '$percepcion')";
21
22    $q=mysqli_query($con, $sql);
23
24    if($q)
25        echo "1";
26    else
27        echo "Ha habido algún error";
28    } else {
29        echo "No hay datos para POST";
30    }
31 }

```

**FIGURA 13. CÓDIGO PHP REFERENTE AL POST ENTRE FORMULARIO Y BBDD. FUENTE: ELABORACIÓN PROPIA**

Básicamente, llamamos a la base de datos, ejecutando `AWS_db.php` (figura 44) para luego generar una serie de posts que enviarán la información a la base de datos mediante una sentencia SQL donde pasamos todos los parámetros y le insertamos los valores (INSERT INTO datos VALUES ("x")).

Con todo ello, ya se habrán ejecutado todos los pasos necesarios para la obtención y envío de los datos desde el formulario a la base de datos creada. Se muestran los datos recibidos en MySQL tras ser enviados mediante el uso del formulario desde diferentes dispositivos móviles en la figura 45.

### 5.4. EL VISOR DE DATOS

En este apartado, nos centramos en la visualización de los datos almacenados en la BBDD, mediante la implantación de un mapa en nuestra app móvil. Utilizaremos en parte, los conocimientos adquiridos en el módulo "Programación de geoaplicaciones web". Para ello programamos `map.html`, es decir el archivo de nuestra app donde queremos visualizar el mapa.

Principalmente, haremos uso de Leaflet-OSM, en el cual establecemos la proyección EPSG:4326 (asociada al WGS84 con el que capturamos la latitud y longitud y nuestro formulario). También definimos en el "setView", las

coordenadas en las que se centra inicialmente el mapa (correspondiente a Barcelona), así como su nivel de zoom.

Más allá de lo aprendido en el máster, utilizamos un getJson (asociado a geojson.php), donde le indicamos que queremos obtener un json, el cual posteriormente se transformará en un punto y se añadirá a la capa (pointToLayer) y este será devuelto en el mapa con un determinado tamaño, contorno, relleno, opacidad y color (figura 46).

A su vez, queremos añadir un popup con la información asociada a cada uno de los puntos (almacenados en la BBDD) del mapa cada vez que se clique. Y le indicamos que esta ventana nos muestre los resultados en formato tabla (figura 47).

En el documento `geojson.php`, hacemos una consulta SQL (figura 48) en la que pedimos que nos seleccione ciertos campos que proceden de Datos ("SELECT fecha, latitud, longitud, decibelios, percepcion FROM datos") y le indicamos que la salida sea en formato json. Con ello, la aplicación ya resultaría funcional.

## 6. RESULTADOS

Tras haber finalizado la implementación, se ha procedido a la realización de un test de calidad (manual 4) y su instalación (manual 5) en un dispositivo móvil para posteriormente encontrarnos ante los resultados del proyecto en este apartado. En otras palabras, aquí vemos los resultados obtenidos mediante las diferentes interfaces de usuario de la aplicación que sirven para darse cuenta de la correcta ejecución del proyecto, tanto por el diseño como por su funcionalidad.



FIGURA 14. PANTALLA DE LA APP EN CARGA

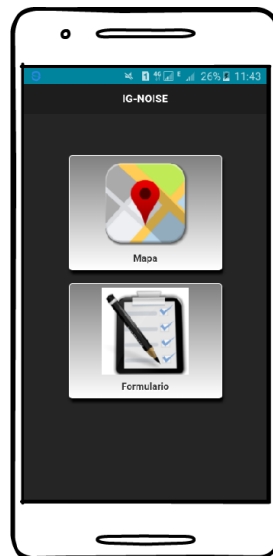


FIGURA 15. PANTALLA DEL MENÚ PRINCIPAL



FIGURA 16. PANTALLA INICIAL DEL FORMULARIO AL CARGAR



FIGURA 17. FORMULARIO AL PRESIONAR "CAPTURAR POSICIÓN Y DECIBELIOS" Y SER CARGADO. TAMBIÉN VISIBLE PREGUNTA Y RESPUESTA DE PERCEPCIÓN.

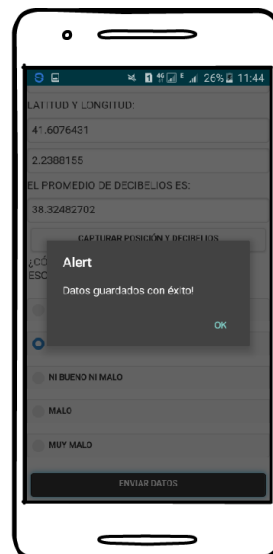


FIGURA 18. FORMULARIO TRAS APRETAR BOTÓN "ENVIAR DATOS".

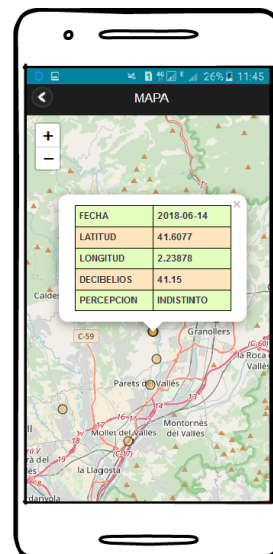


FIGURA 19. MAPA CON LA CAJA DE INFORMACIÓN DEL PUNTO CLICKADO.



## 7. CONCLUSIONES

El resultado final del proyecto nos permite ver que se han llevado a cabo con éxito la mayor parte de los objetivos propuestos inicialmente, tanto en términos de definición como en diseño, implementación, calidad e instalación del mismo.

Hay que resaltar el correcto registro de los datos mediante el acceso a la fecha y hora actual del dispositivo, la captura del promedio de dB, el acceso al ID del dispositivo (UUID) y a la posición del ruido, el cumplimiento de los requerimientos funcionales y no funcionales, así como el hecho de ser *responsive*, mantener la privacidad del usuario y permitir al usuario expresar su percepción en base a un determinado registro de contaminación acústica, hechos todos ellos muy significativos.

Del diseño cabe destacar, el hecho de haber implementado, a nuestro entender, unos iconos y estructuras de página atractivos a ojos del usuario final, haciendo que la navegación sea sencilla e intuitiva en todo momento y mediante el uso de software *Open Source*.

En términos de implantación y programación (ciclo de desarrollo) se constata también un notable éxito del mismo ya que la aplicación y sus diferentes funcionalidades cumplen perfectamente con su cometido.

Recalcar también los buenos resultados que nos ha dado la app, ya que tras las comprobaciones (funcionales y de velocidad) y correcciones de los errores que han ido surgiendo a lo largo del proyecto, se ha acabado implementando con totales garantías.

Todo ello, ha sido posible gracias al buen trabajo realizado mediante los análisis de requerimientos, su planificación y, sobre todo por el cumplimiento y ejecución de cada una de las fases establecidas. No se puede olvidar que el uso de los *frameworks* (Apache Cordova y JQuery Mobile), editores de código (Sublime Text), BBDD (MySQL), lenguajes de código abierto (PHP) y las librerías empleadas (CSS y JS) han sido primordiales para el buen hacer de este proyecto.

A pesar de cumplir con todos los objetivos marcados inicialmente, se presentan algunas limitaciones y propuestas de mejora.

Entre las limitaciones, nos encontramos con los problemas de la **calibración** y la **cancelación del ruido ambiental** que se genera en los micrófonos de nuestros dispositivos móviles, en comparación con los sonómetros profesionales y, con el cual se *“sacrifica precisión y capacidad de respuesta a favor de una solución integrada y barata”*. Estas diferencias son especialmente notables en linealidad de la respuesta en frecuencia, ya que *“al medir sonido con un Smartphone estaríamos filtrando las frecuencias más bajas y distorsionando el resultado”* y por “el ruido inherente” motivo por el cual *“no podemos medir niveles sonoros cercanos o inferiores al ruido inherente”* (Laguna, 2014).

Como propuestas de mejora para próximas versiones estarían: añadir una pregunta adicional en la que se pregunte al usuario si el **ruido** se ha **capturado** en **interior** (p.ej. algún edificio) o **exterior** (al aire libre). A su vez, resultaría interesante poder **compartir información** propia con nuestros amigos y **clasificar los puntos** del mapa por **color y cara**, en función de la calidad del ruido y la percepción del mismo por parte del usuario.

Se cree también que esta aplicación, podría resultar **interesante para personas que buscan piso** ya que podrían decidir dónde vivir a partir de variables relacionadas con el ruido y visibles en el mapa.

En términos generales, el resultado obtenido en sí mismo ha sido muy satisfactorio, ya que se ha dado lugar a una aplicación móvil para el registro de la contaminación ambiental al servicio de los usuarios que, aunque con limitaciones, da un **valor añadido** a las aplicaciones existentes de captura del ruido ya que esta **permite contrastar el ruido** en si mismo (cantidad de dB) **con la percepción del usuario**.

Como resultado, se han puesto en práctica muchos de los conocimientos adquiridos durante el máster y otros con el desarrollo de este proyecto, en tanto que se ha creado una aplicación y una base de datos a la vez que se ha cuidado también su estética y funcionalidad, mediante el uso de las herramientas adecuadas y los estándares establecidos inicialmente.

## REFERENCIAS BIBLIOGRÁFICAS

### Artículos de investigación

Eze, I.; Foraster M.; Schaffner E.; Vienneau D.; Héritier H.; Rudzik F.; Thiesse L.; Pieren R.; Imboden M.; von Eckardstein A.; Schindler C.; Brink M.; Cajochen C.; Wunderli JM.; Rösli M.; Probst-Hensch N.; *Long-term exposure to transportation noise and air pollution in relation to incident diabetes in the SAPALDIA study*. International Journal of Epidemiology. Volume 46, Issue 4, Ago 2017, Pages 1115-1125.

Fields, J. M.; de Jong, R. G.; Gjestland, T.; Flindell, I. H.; Job, R. F. S.; Kurra, S.; Lercher, P.; Vallet, M.; Yano, T.; Guski, R.; FelscherSuhr, U.; Schumer, R.; Community Response To Noise Team Of Icben (International Commission On The Biological Effects Of Noise) (2001). *Standardized general-purpose noise reaction questions for community noise surveys: research and a recommendation*. Journal of Sound and Vibration. Volume 242, Issue 4. Pages 641-679.

Foraster, M., Eze, I. C., Schaffner, E., Vienneau, D., Héritier, H., Endes, S., Probst-Hensch, N. (Jun 2017). *Exposure to source-specific transportation noise levels and temporal noise characteristics in association with arterial stiffness*. In 12th ICBEN congress on noise as a public health problem (p. (5 pp.)). Zurich, Switzerland.

Foraster, M.; Eze, I.; Vienneau, D.; Brink, M.; Cajochen, C.; Caviezel, S.; Héritier, H.; Schaffner, E.; Schindler, C.; Wanner, M.; Wunderli, J.M.; Rösli, M.; Probst-Hensch, N.; *Long-term transportation noise annoyance is associated with subsequent lower levels of physical activity*. Environment International. Volume 91, May 2016, Pages 341-349.

Guski, R.; Schreckenber, D.; Schuemer, R. *WHO Environmental Noise Guidelines for the European Region: A Systematic Review on Environmental Noise and Annoyance*. Int. J. Environ. Res. Public Health, Dic 2017, 14, 1539.

### Webs de interés

APK Easy Tool. Disponible en: <https://forum.xda-developers.com/android/software-hacking/tool-apk-easy-tool-v1-02-windows-gui-t3333960>

APACHE CORDOVA. Disponible en:

<https://cordova.apache.org/#getstarted>

CONTAMINACIÓN ACÚSTICA. Disponible en:

[www.contaminacionacustica.net](http://www.contaminacionacustica.net)

CORDOVA – Plugin Geolocation. Disponible en:

<https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-geolocation/>

CORDOVA - Plugin Media Capture. Disponible en:

<https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-media-capture/>

GITHUB - Noise Capture. Disponible en:

[https://github.com/lfsttar/NoiseCapture/blob/master/app/src/main/java/org/noise\\_planet/noisecapture/Storage.java](https://github.com/lfsttar/NoiseCapture/blob/master/app/src/main/java/org/noise_planet/noisecapture/Storage.java)

GITHUB – Plugin dBmeter. Disponible en:

<https://github.com/akofman/cordova-plugin-dbmeter>

HEALTHY HEARING. Disponible en:

<https://www.healthyhearing.com/report/47805-The-best-phone-apps-to-measure-noise-levels>

ISGLOBAL. Disponible en: [www.isglobal.org](http://www.isglobal.org)

MOISES LAGUNA. Disponible en: <http://moiseslaguna.com/un-smartphone-no-es-un-sonometro/>

NOISETUBE. Disponible en: [www.noisetube.net](http://www.noisetube.net)

NOISE CAPTURE. Disponible en: <http://noise-planet.org/noisecapture.html>

NOISE POLLUTION MONITOR. Disponible en: <https://apkpure.com/noise-pollution-monitor/com.miiint.citynoise>

RACCOON – The APK Downloader. Disponible en: <https://raccoon.onyxbits.de/>

SER PROGRAMADOR. Disponible en:

<https://serprogramador.es/que-es-frontend-y-backend-en-la-programacion-web/>

SOUND METER. Disponible en:

<https://play.google.com/store/apps/details?id=com.gamebasic.decibel&hl=es>

## ANEXOS



FIGURA 20. ICONO APLICACIÓN "NOISETUBE". FUENTE : PLAY STORE



FIGURA 21. ICONO APLICACIÓN "SOUND METER – SONÓMETRO". FUENTE : PLAY STORE



FIGURA 22. ICONO APLICACIÓN "NOISE POLLUTION MONITOR - CITY NOISE". FUENTE : PLAY STORE



FIGURA 23. ICONO APLICACIÓN "NOISE CAPTURE". FUENTE : PLAY STORE

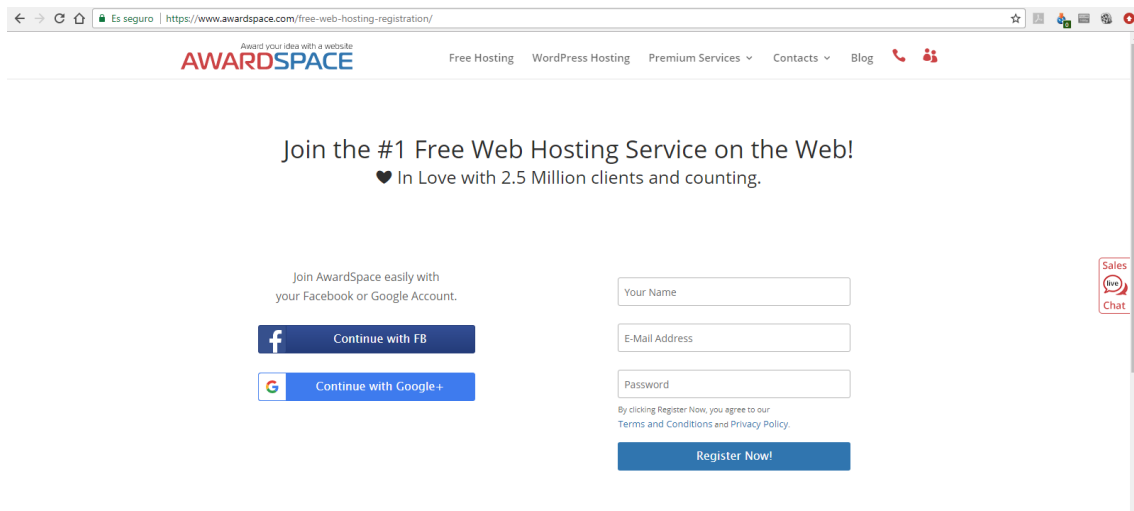


FIGURA 24. PÁGINA DE REGISTRO PARA OBTENCIÓN HOSTING GRATUITO.

FUENTE: [WWW.AWARDSPACE.COM](http://WWW.AWARDSPACE.COM)

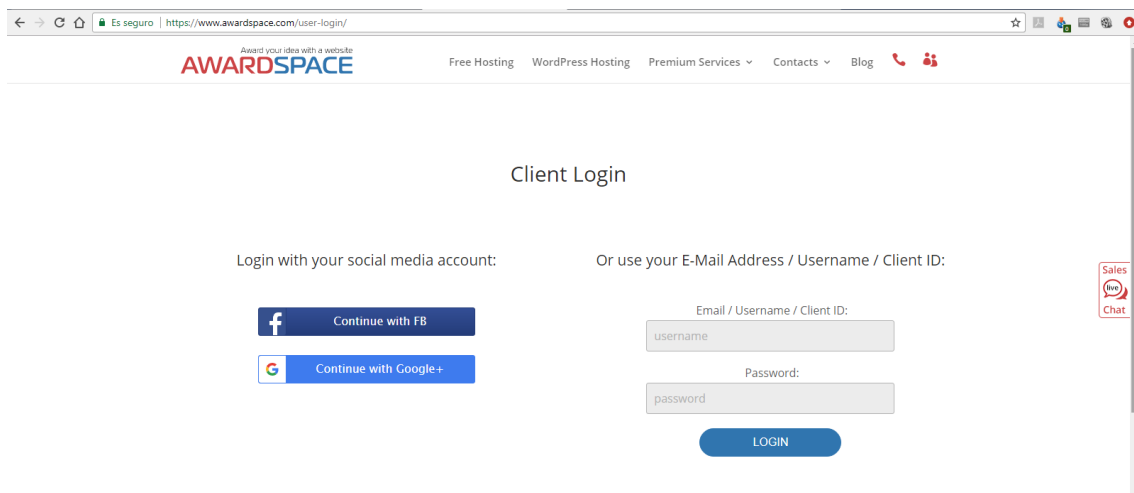


FIGURA 25. PÁGINA PARA HACER LOGIN EN EL HOSTING WEB GRATUITO.

FUENTE: [WWW.AWARDSPACE.COM](http://WWW.AWARDSPACE.COM)



FIGURA 26. MENÚ PRINCIPAL AWARDSPACE TRAS LOGIN. FUENTE: [WWW.AWARDSPACE.COM](http://WWW.AWARDSPACE.COM)

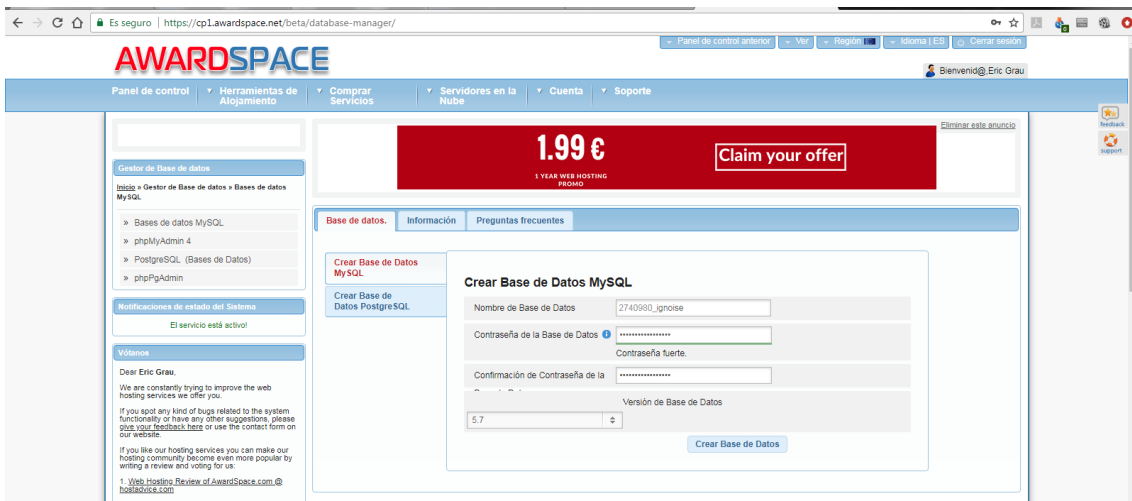


FIGURA 27. PÁGINA DE CREACIÓN DE BASES DE DATOS MYSQL.

FUENTE: [WWW.AWARDSPACE.COM](http://WWW.AWARDSPACE.COM)

Todas las Bases de Datos							
Bases de datos MySQL		PostgreSQL (Bases de Datos)					
Filtro de búsqueda: <input type="text" value="Type database to filter"/>							
Nombre	Usuario	Alojamiento	Puerto	Cuota	Administración	Tipo	Opciones
+ 2740980_ignoise	2740980_ignoise	fdb21.awardspace.net	3306	Disponible: {cuota} En uso: 112 KB	phpMyAdmin_4 <a href="#">Ver todas las herramientas</a>	MySQL	

FIGURA 28. VISUALIZACIÓN DE DATOS REFERENTES A BBDD CREADA. FUENTE: [WWW.AWARDSPACE.COM](http://WWW.AWARDSPACE.COM)

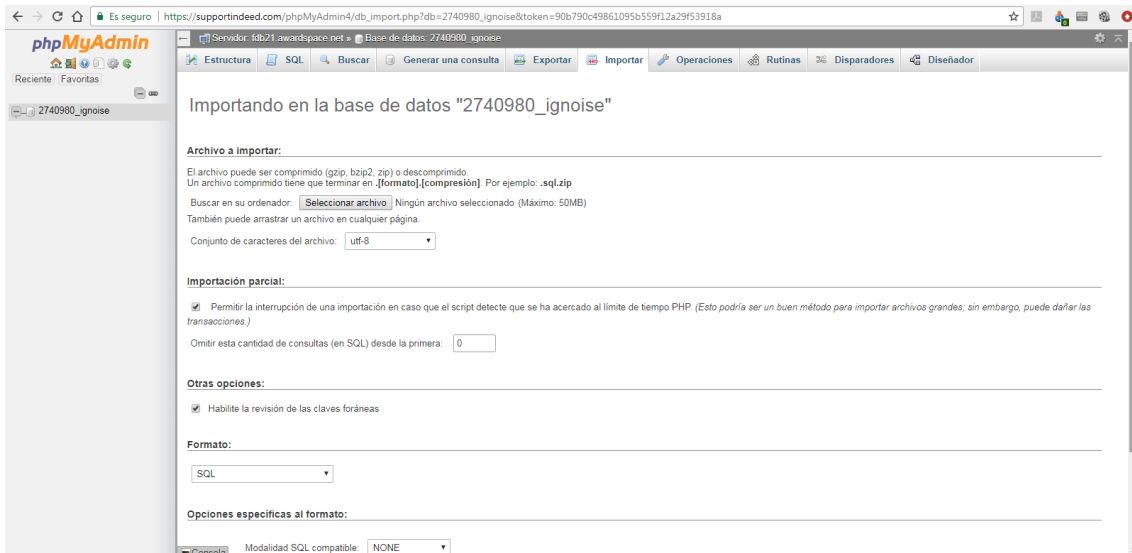


FIGURA 29. SECCIÓN BBDD PARA IMPORTAR ARCHIVO SQL.

FUENTE: [WWW.AWARDSPACE.COM](http://WWW.AWARDSPACE.COM)



FIGURA 30. VISUALIZACIÓN DE LA TABLA "DATOS" CREADA.

FUENTE: [WWW.AWARDSPACE.COM](http://WWW.AWARDSPACE.COM)

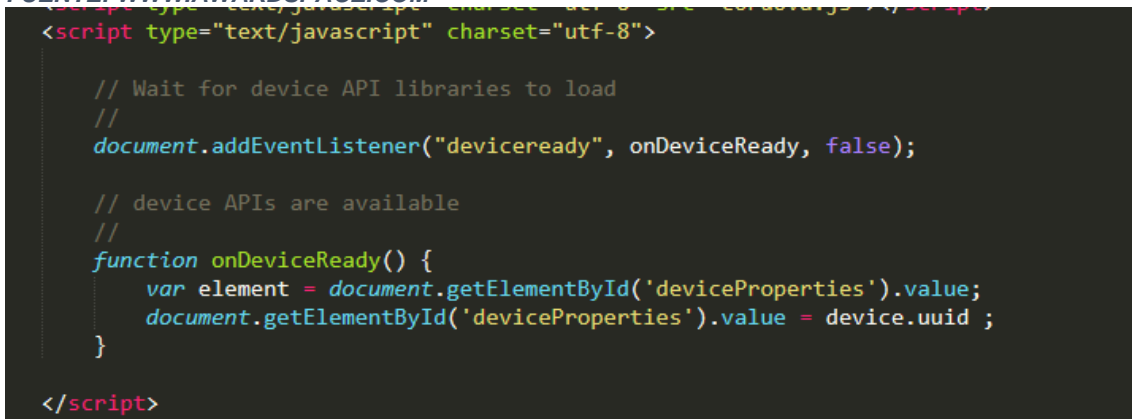


FIGURA 31. CÓDIGO SCRIPT PARA OBTENCIÓN "UUID". FUENTE: ELABORACIÓN PROPIA

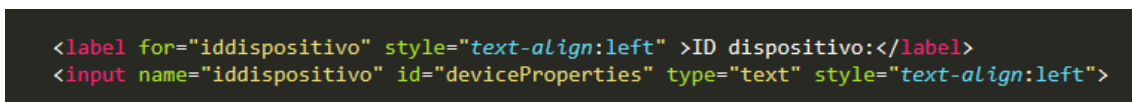


FIGURA 32. CÓDIGO "UUID" EN EL FORMULARIO. FUENTE: ELABORACIÓN PROPIA

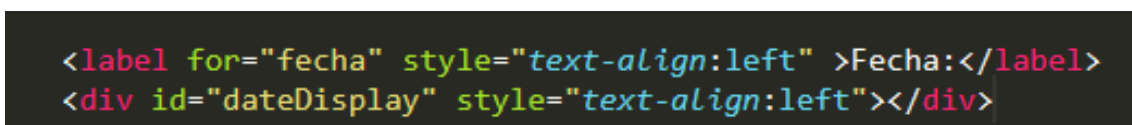


FIGURA 33. CÓDIGO "FECHA" EN EL FORMULARIO. FUENTE: ELABORACIÓN PROPIA



```

form.html app.js
1 // -----FECHA-----
2
3 let today = new Date().toISOString().substr(0, 10);
4 document.querySelector("#today").value = today;

```

FIGURA 34. CÓDIGO DE LA FUNCIÓN JS PARA OBTENCIÓN DE "FECHA". FUENTE: ELAB.PROPIA

```

<label for="hora" style="text-align:left">Hora:</label>
<input type="text" name="hora" id="hora" value="now"/>

```

FIGURA 35. CÓDIGO "HORA" EN EL FORMULARIO. FUENTE: ELABORACIÓN PROPIA

```

form.html app.js
// -----HORA-----
function startTime() {
    var today = new Date();
    var h = today.getHours();
    var m = today.getMinutes();
    var s = today.getSeconds();
    m = checkTime(m);
    s = checkTime(s);
    document.getElementById('hora').value =
    h + ":" + m + ":" + s;
}
function checkTime(i) {
    if (i < 10) {i = "0" + i}; // add zero in front of numbers < 10
    return i;
}

```

FIGURA 36. CÓDIGO DE LA FUNCIÓN JS PARA OBTENCIÓN DE "HORA". FUENTE: ELAB. PROPIA

```
// ----- GEOLOCALIZACIÓN -----
function getCurrentPosition() {
  var onSuccess = function(position) {
    console.log(
      'Latitude: ' + position.coords.latitude + '\n' +
      'Longitude: ' + position.coords.longitude + '\n' +
      'Altitude: ' + position.coords.altitude + '\n' +
      'Accuracy: ' + position.coords.accuracy + '\n' +
      'Altitude Accuracy: ' + position.coords.altitudeAccuracy + '\n' +
      'Heading: ' + position.coords.heading + '\n' +
      'Speed: ' + position.coords.speed + '\n' +
      'Timestamp: ' + position.timestamp + '\n'
    );
    $('#text-latitud').val(position.coords.latitude);
    $('#text-longitud').val(position.coords.longitude);
  };

  function onError(error) {
    alert('code: ' + error.code + '\n' + 'message: ' + error.message + '\n');
  }

  navigator.geolocation.getCurrentPosition(onSuccess, onError);
}
```

FIGURA 37. CÓDIGO DE LA FUNCIÓN JS PARA OBTENCIÓN DE "GEOLOCALIZACIÓN". FUENTE: ELABORACIÓN PROPIA

The screenshot shows the GitHub repository page for 'akofman / cordova-plugin-dbmeter'. At the top, there are navigation links for Features, Business, Explore, Marketplace, and Pricing, along with a search bar and 'Sign in' / 'Sign up' buttons. The repository name is 'akofman / cordova-plugin-dbmeter' with 4 watches, 22 stars, and 14 forks. Below this, there are tabs for Code, Issues (3), Pull requests (0), Projects (0), and Insights. A 'Join GitHub today' banner is present. The repository description is 'Cordova plugin to get decibels from the microphone'. It shows 54 commits, 1 branch, 3 releases, and 1 contributor. A commit history table is visible below.

Commit	Message	Time
akofman	removes swift dependency and reorg project structure	6 months ago
plugin	removes swift dependency and reorg project structure	6 months ago
editorconfig	add eslint config	2 years ago
eslintrc	add eslint config	2 years ago
gitignore	Fix demo app	8 months ago
.npmignore	add eslint config	2 years ago
.travis.yml	update configs and rm circleci because tests doesn't work on it	2 years ago
LICENSE	init project	2 years ago
README.md	removes swift dependency and reorg project structure	6 months ago
package.json	removes swift dependency and reorg project structure	6 months ago

FIGURA 38. PÁGINA DE GITHUB PARA IMPLANTACIÓN DE "DBMETER". FUENTE: GITHUB

```

    setTimeout(function()
    {
        stop();

        setTimeout(function()
        {
            calculaPromedio(mediciones);
        }, 1000);
    }, 5000);
}
catch(err) {
    alert(err);
}
}

```

FIGURA 39. CÓDIGO JS PARA “REGISTRO DE DECIBELIOS DURANTE 5 SEGUNDOS”.

FUENTE: ELABORACIÓN PROPIA

```

function calculaPromedio(mediciones)
{
    var sumatoria = 0;
    for (var i = 0; i < mediciones.length; i++)
    {
        sumatoria += mediciones[i];
    }
    document.getElementById('text-ruido').value = ( (sumatoria/mediciones.length) );
}

```

FIGURA 40. CÓDIGO PARA “CALCULAR PROMEDIO DE DECIBELIOS”.

FUENTE: ELABORACIÓN PROPIA

```

<fieldset id="percepcion" data-role="controlgroup" data-mini="true" style="text-align:left">
<legend>¿Cómo describiría el sonido que usted escucha en este lugar?</legend>
<input type="radio" name="percepcion" id="muybueno" value="muy bueno" />
<label for="muybueno">Muy Bueno</label>
<input type="radio" name="percepcion" id="bueno" value="bueno"/>
<label for="bueno">Bueno</label>
<input type="radio" name="percepcion" id="indistinto" value="indistinto" checked="checked"/>
<label for="indistinto">Ni bueno ni malo</label>
<input type="radio" name="percepcion" id="malo" value="malo"/>
<label for="malo">Malo</label>
<input type="radio" name="percepcion" id="muymalo" value="muy malo"/>
<label for="muymalo">Muy Malo</label>
</fieldset>

```

FIGURA 41. CÓDIGO REFERENTE A LA PREGUNTA DE “PERCEPCIÓN DEL RUIDO” EN EL FORMULARIO. FUENTE: ELABORACIÓN PROPIA

```
<script type="text/javascript">
$(document).ready(function() {
    $("#insert").click(function() {

        var iddispositivo = $("#deviceProperties").val();
        var fecha = $("#today").val();
        var hora = $("#hora").val();
        var latitud = $("#text-latitud").val();
        var longitud = $("#text-longitud").val();
        var decibelios = $("#text-ruido").val();
        var percepcion = $("#percepcion input:checked").val();

        var dataString =
            "iddispositivo=" + iddispositivo +
            "&fecha=" + fecha +
            "&hora=" + hora +
            "&latitud=" + latitud +
            "&longitud=" + longitud +
            "&decibelios=" + decibelios +
            "&percepcion=" + percepcion +
            "&insert=1";

        if
            ($.trim(iddispositivo).length > 0 &
            $.trim(fecha).length > 0 &
            $.trim(hora).length > 0 &
            $.trim(latitud).length > 0 &
            $.trim(longitud).length > 0 &
            $.trim(decibelios).length > 0 &
            $.trim(percepcion).length > 0
            ) {
```

FIGURA 42. PARTE DEL SCRIPT DEL FORMULARIO PARA ENVÍO DE VARIABLES Y VALORES. FUENTE: ELABORACIÓN PROPIA

```
        $.ajax({
            type: "POST",
            url: "http://xxx/AWS_insert.php",
            data: dataString,
            crossDomain: true,
            cache: false,
            beforeSend: function() {
                $("#insert").val('Conectando...');
            },
            success: function(data) {

                if (data == "1") {
                    alert("Datos guardados con éxito!");

                    window.location='index.html';
                    $("#insert").val('submit');
                } else {
                    alert("error");
                }
            }
        });

    } else {
        alert("Completa el formulario");
    }
    return false;
});
});
</script>
```

FIGURA 43. SCRIPT DEL FORMULARIO PARA REALIZACIÓN POST Y ENVÍO DATOS. FUENTE: ELABORACIÓN PROPIA

```
<?php
header("Access-Control-Allow-Origin: *");
$con=mysqli_connect("fdb21.awardspace.net","2740980_ignoise","xxx");
$db=mysqli_select_db($con,"2740980_ignoise");
?>
```

FIGURA 44. CÓDIGO DEL FICHERO AWS\_DB.PHP QUE PERMITE CONEXIÓN. FUENTE: ELABORACIÓN PROPIA

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición están desactivados.

Mostrando filas 0 - 19 (total de 20, La consulta tardó 0.0012 segundos.)

`SELECT * FROM `datos``

Número de filas: 25    Filtrar filas: Buscar en esta tabla

iddispositivo	fecha	hora	latitud	longitud	decibelios	percepcion
60eda0b1c7d2ff82	2018-06-19	11:43:13	41.6076	2.23882	38.32	bueno
60eda0b1c7d2ff82	2018-06-09	0:47:59	41.6077	2.2388	46.69	bueno
60eda0b1c7d2ff82	2018-06-09	0:49:57	41.6077	2.2388	46.57	bueno
60eda0b1c7d2ff82	2018-06-10	21:44:42	41.6077	2.2388	44.03	bueno
60eda0b1c7d2ff82	2018-06-10	21:52:20	41.6077	2.23882	37.00	muy bueno
60eda0b1c7d2ff82	2018-06-11	19:02:25	41.6168	2.29455	60.62	malo
60eda0b1c7d2ff82	2018-06-12	8:11:11	41.5894	2.24224	69.28	malo
60eda0b1c7d2ff82	2018-06-12	8:19:44	41.5718	2.23646	67.27	malo
60eda0b1c7d2ff82	2018-06-12	8:24:43	41.5332	2.21691	77.78	muy malo
60eda0b1c7d2ff82	2018-06-12	8:31:42	41.4819	2.18434	71.78	muy malo
60eda0b1c7d2ff82	2018-06-12	8:32:33	41.4819	2.18434	70.74	muy malo
60eda0b1c7d2ff82	2018-06-12	8:33:12	41.4819	2.18434	69.51	malo
60eda0b1c7d2ff82	2018-06-12	8:34:42	41.4405	2.18601	74.02	muy malo
60eda0b1c7d2ff82	2018-06-12	8:47:58	41.4208	2.18865	62.29	malo
60eda0b1c7d2ff82	2018-06-12	9:21:56	41.3847	2.19492	47.96	bueno
60eda0b1c7d2ff82	2018-06-12	11:23:04	41.3847	2.19492	51.33	indistinto
60eda0b1c7d2ff82	2018-06-14	19:14:42	41.6077	2.23878	41.15	indistinto
95bb36393604f036	2018-06-14	0:08:16	41.5556	2.15981	46.29	muy bueno
d7416890fda08bf3	2018-06-14	0:18:38	41.6884	2.47544	32.55	indistinto
60eda0b1c7d2ff82	2018-06-15	11:04:41	41.3847	2.19471	67.92	indistinto

Número de filas: 25    Filtrar filas: Buscar en esta tabla

FIGURA 45. VISUALIZACIÓN DE LOS DATOS RECIBIDOS EN MYSQL DESDE FORMULARIO DE LA APP. FUENTE: ELABORACIÓN PROPIA

```
<script type="text/javascript">
  var mapa;

  var crs4326;
  function init()
  {
    crs4326 = new L.Proj.CRS('EPSG:4326');
    mapa = L.map("map_canvas").setView([41.3836555, 2.1948108000000275], 12);

    var openStreetMap = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
    {
      maxZoom: 19,
      minZoom: 1,
      attribution: 'OSM',
      crs: crs4326
    });
    mapa.addLayer(openStreetMap);

    var jsonResponse;
    $.getJSON('http://xxx/geojson.php', function(json)
    {
      jsonResponse = json;
    }).done(function (data)
    {
      var featuresLayer = new L.Proj.geoJson(jsonResponse,
      {
        'pointToLayer' : function (feature, latlng)
        {
          return L.circleMarker(latlng,
          {
            color: '#1c2833',
            fillColor: '#f5b041',
            fillOpacity: 0.5,
            radius: 6,
            weight: 1
          });
        },
        'onEachFeature' : onEachFeature
      }).addTo(mapa);
    });
  }
}
```

FIGURA 46. CÓDIGO PARA OBTENCIÓN DE JSON Y MAPA LEAFLET EN LA APP.

FUENTE: ELABORACIÓN PROPIA.

```
function onEachFeature(feature, Layer)
{
  var popupContent = "<table class='table'>";
  if (feature.properties)
  {
    $.each(feature.properties,function (i, val)
    {
      popupContent += "<tr><th width='30%'>"+i+"</th><td width='70%'>"+val+"</td></tr>";
    });
  }
  popupContent += "</table>";
  layer.bindPopup(popupContent);
}
</script>
```

FIGURA 47. CÓDIGO PARA LA OBTENCIÓN DE POPUP AL CLICKAR UN PUNTO EN EL MAPA.  
FUENTE: ELABORACIÓN PROPIA.

```
if ($resultado = $mysqli->query("SELECT fecha,latitud,longitud,decibelios,percepcion FROM datos"))
```

FIGURA 48. CONSULTA LANZADA EN GEOJSON.PHP PARA MAPA. FUENTE: ELABORACIÓN PROPIA.

## **MANUAL 1. INSTALACIÓN APACHE CORDOVA Y PRIMEROS PASOS**

Autor: Eric Grau - Fecha de elaboración: Mayo 2018

Previamente, descargamos, instalamos y configuramos sus correspondientes variables de entorno, JDK de Java, SDK de Android Studio. A su vez, también instalamos NODE JS.

Comenzamos la instalación de Cordova. Para ello seguimos los siguientes pasos:

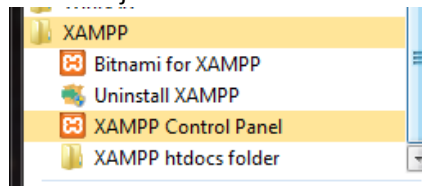
- 1- Accedemos a la ruta del proyecto desde la consola de Windows (Símbolo del sistema).
- 2- Ejecutamos `"npm install -g cordova"` para instalar Cordova.
- 3- Ejecutamos `"cordova create MyApp"` para crear el directorio del proyecto de la app con sus correspondientes archivos de configuración.
- 4- Ejecutamos `"cordova platform add android"` para añadir la plataforma que deseamos para nuestro proyecto. En este caso, Android.
- 5- Instalamos el plugin de DB Meter (`cordova plugin add cordova-plugin-dbmeter`) así como otros que vamos a necesitar (`geolocation, ...`)

**MANUAL 1. INSTALACIÓN APACHE CORDOVA Y PRIMEROS PASOS . FUENTE: ELABORACIÓN PROPIA.**

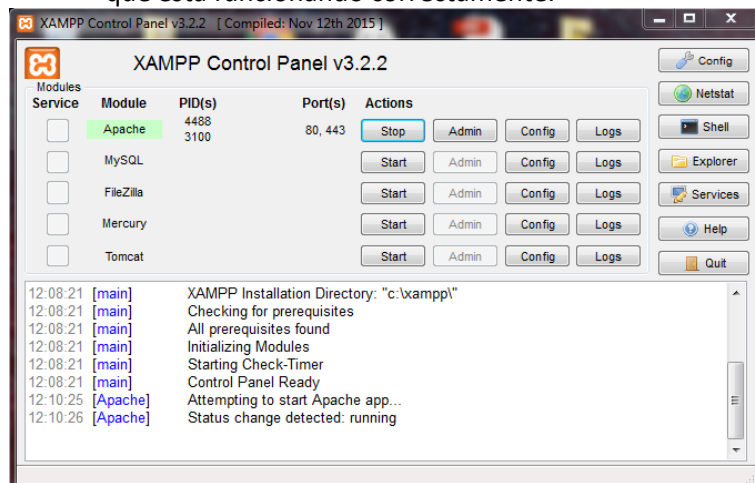
## MANUAL 2. XAMPP PARA LA CREACIÓN DE UNA BBDD CON MySQL EN LOCALHOST

Autor: Eric Grau - Fecha de elaboración: Mayo 2018

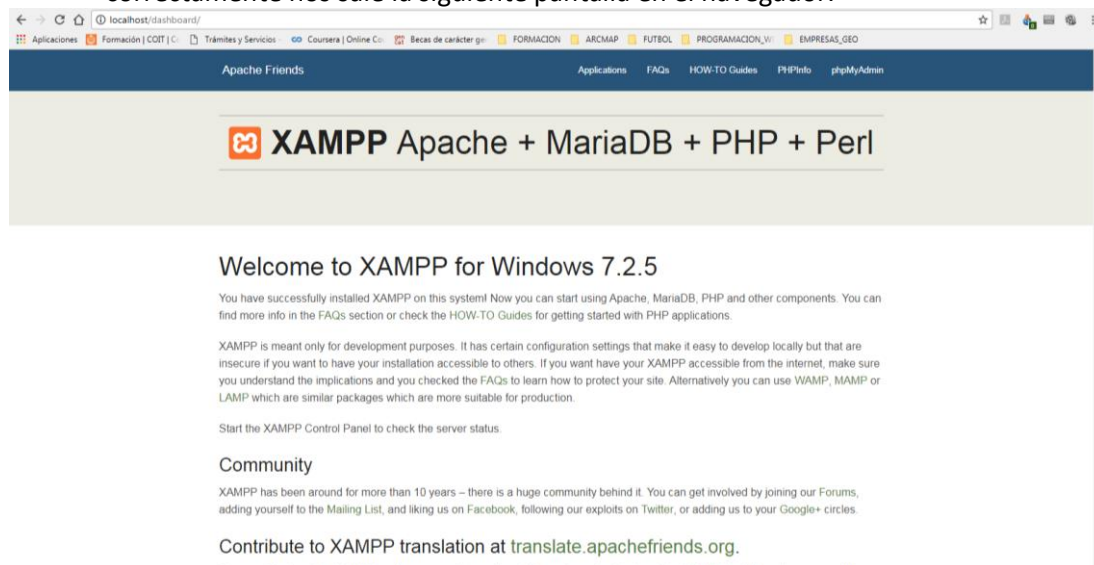
1. Instalamos XAMPP
2. Ejecutamos XAMPP Control Panel



3. Apretamos al primer botón Start asociado al Módulo "Apache". Si sale en verde es que está funcionando correctamente.



4. Ejecutamos en el navegador: localhost. Al haber realizado la instalación de XAMPP correctamente nos sale la siguiente pantalla en el navegador.



5. Vamos al directorio donde tenemos instalado XAMPP y creamos una nueva carpeta para el formulario. En nuestro caso la ruta es: C:\xampp\htdocs

**NOTA:** El campo "Name" del formulario creado en HTML, es la variable que se recupera y se utiliza en PHP.

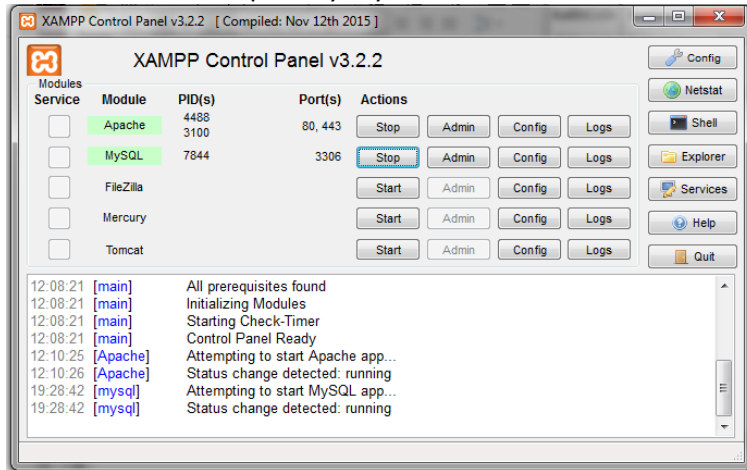
**MANUAL 2. XAMPP PARA LA CREACIÓN DE UNA BBDD CON MYSQL EN LOCALHOST. FUENTE: ELABORACIÓN PROPIA.**



## MANUAL 3. CREACIÓN BASE DE DATOS CON MYSQL EN LOCALHOST

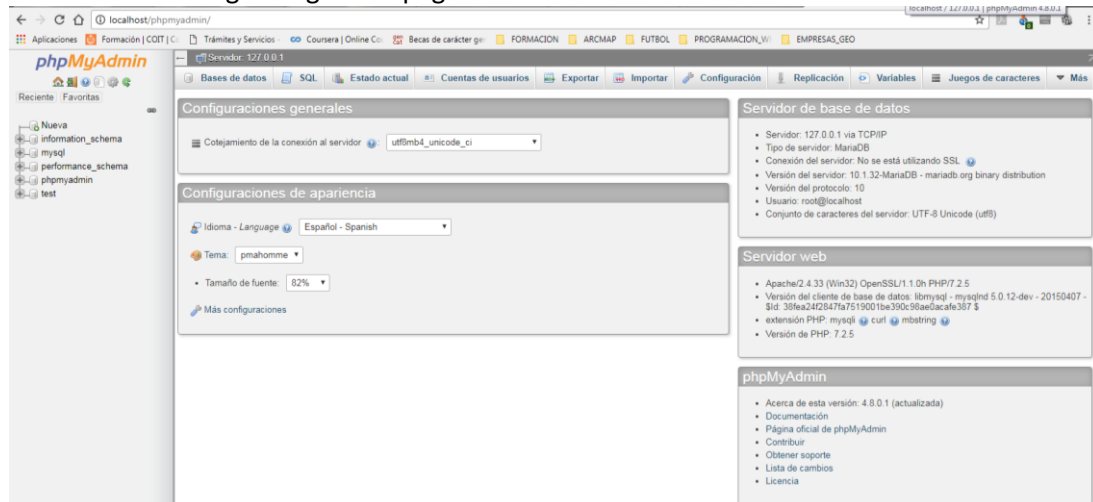
Autor: Eric Grau - Fecha de elaboración: Mayo 2018

1. Vamos a XAMPP Control Panel
2. Activamos Apache y MySQL



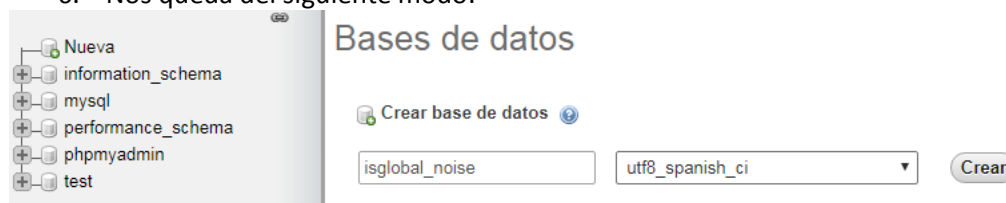
3. Entramos a: localhost/phpmyadmin

4. Se nos carga la siguiente página:



5. Apretamos a “Nueva” en el panel izquierdo para crear nuestra base de datos.

6. Nos queda del siguiente modo:

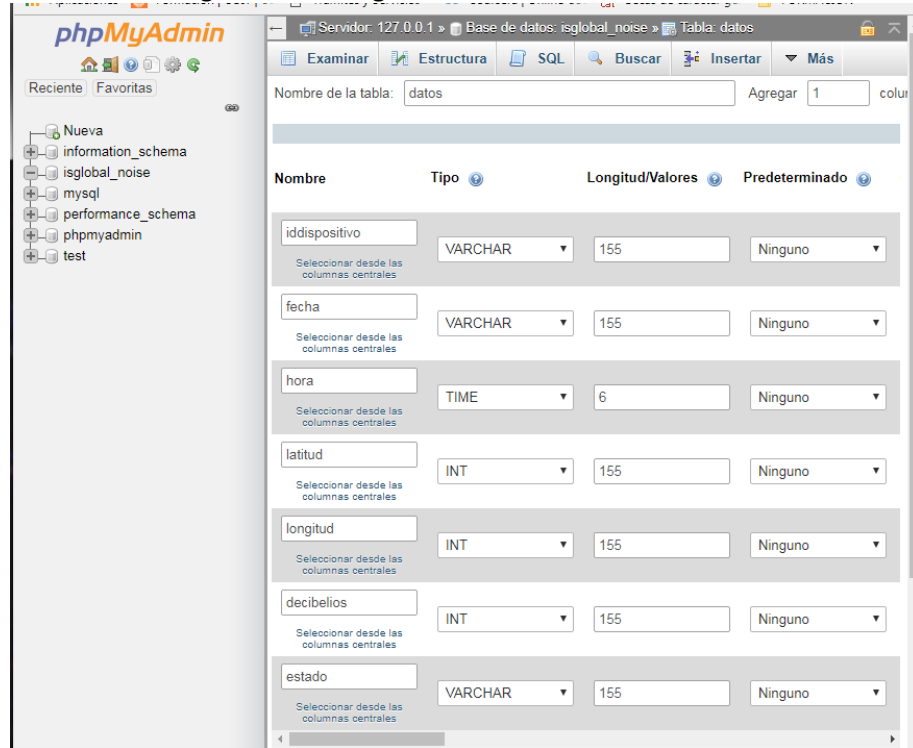


7. Pulsamos sobre la base de datos creada y que puede visualizarse en la parte izquierda. En nuestro caso, este se llama “isglobal\_noise”.

8. Ahora creamos una tabla la cual llamaremos “dato”s y le indicaremos el número de columnas(campos) que va a tener. Quedará del siguiente modo:



9. Lo configuramos del siguiente modo:



**NOTA:** También se podría haber creado dicha tabla y sus columnas habiendo realizado un archivo SQL y habiéndolo ejecutado.

10. Para finalizar, apretamos a Guardar y se nos creará.

**NOTA:** También podríamos haber creado la tabla “datos” mediante la ejecución de una sentencia SQL como la siguiente:

```
CREATE TABLE `isglobal_noise`.`datos` ( `iddispositivo` VARCHAR(155) NOT NULL , `fecha` VARCHAR(155) NOT NULL , `hora` TIME(6) NOT NULL , `latitud` INT(155) NOTNULL , `longitud` INT(155) NOT NULL , `decibelios` INT(155) NOT NULL , `estado` VARCHAR(155) NOT NULL ) ENGINE = InnoDB;
```

11. A partir de aquí, procedemos a crear nuestro código en PHP para almacenar datos en esta BBDD.

**IMPORTANTE:** Se presentan algunas diferencias entre la implementación final realizada en el host web y la realizada previamente en localhost. Básicamente, se ha querido comprobar su plena funcionalidad y correcto funcionamiento.

MANUAL 3. CREACIÓN BASE DE DATOS CON MYSQL EN LOCALHOST. FUENTE: ELABORACIÓN PROPIA

#### **MANUAL 4. PASOS REALIZADOS EN EL TEST DE CALIDAD DE LA APP**

Autor: Eric Grau - Fecha de elaboración: Mayo 2018

En este punto, nos centramos básicamente en la realización de pruebas de control de calidad para cerciorarnos que la velocidad de navegación es adecuada, a la vez que realizamos una comprobación de toda la funcionalidad y control de errores.

Básicamente, estudiamos si la facilidad de uso, funcionalidad, usabilidad, seguridad y rendimiento de nuestra aplicación es la esperada para así poder asegurar que estamos ante un producto que cumple con los estándares de calidad y rendimiento y que, por lo tanto, funciona correctamente en todos los dispositivos móviles bajo sistema operativo Android.

Vamos testeando cada una de las páginas, sus botones y funcionalidades y así observamos su comportamiento y evitamos fallos de la aplicación. Para ello, estudiamos inicialmente su comportamiento en un entorno de escritorio (navegador web Firefox), aunque posteriormente este acabará siendo testeado en nuestros propios dispositivos móviles.

Para ello, en el navegador abrimos con la tecla F12 la consola, la cual a medida que vamos navegando por las diferentes páginas nos facilita enormemente la detección de fallos y errores. En la mayoría de casos, se trata de problemas de código, motivo por el cual, si se detecta alguno de ellos, se corrige y se verifica de nuevo hasta que se constata su total resolución.

***MANUAL 4. PASOS REALIZADOS EN EL TEST DE CALIDAD DE LA APP.***

***FUENTE: ELABORACIÓN PROPIA***

## **MANUAL 5. PASOS REALIZADOS EN LA INSTALACIÓN DE LA APP**

Autor: Eric Grau - Fecha de elaboración: Mayo 2018

Procedemos a la instalación del programa en nuestro teléfono móvil.

Antes de poder dar este paso, tenemos que haber configurado Android Studio 3.1.1. correctamente. Para ello indicamos e instalamos la versión de SDK 19.1 aunque se podría haber optado por cualquier otra siempre y cuando no esté “obsoleta”. Corroboramos a su vez que las variables de entorno del SDK de Android (ANDROID\_HOME) y del JDK (Java) son correctas y sino las corregimos. Si todo ello se encuentra bien definido, ya podremos establecer la ruta de nuestro proyecto en la consola y crear el proyecto de Cordova propiamente dicho, así como integrar los diferentes plugins que podamos ir necesitando (geolocalización y dbmeter). En este caso, se ha decidido modificar también el icono de la app, así como incluir el *splash screen* (en config.xml), es decir, la pantalla de inicio que sale al ejecutar dicha aplicación con la idea de darle un toque más personal y relacionado con el logo de la entidad.

Una vez que lo tenemos todo a punto, en la consola de Windows (CMD) ejecutamos “cordova build android” para así compilar nuestra aplicación. Esto nos genera una APK, es decir una aplicación en formato Android la cual encontraremos en la ruta “platforms\android\app\build\outputs\apk\debug”. En este caso, se podría haber optado por una versión Release en vez de Debug pero las diferencias resultan prácticamente imperceptibles.

La descargamos y ejecutamos en nuestro dispositivo móvil a la vez que verificamos que todo funciona según lo esperado. De no ser así, se podría volver a modificar el código hasta que lo demos por bueno y solo haría falta repetir los pasos hechos con anterioridad.

Finalmente, acabamos obteniendo una aplicación plenamente funcional y con unos resultados más que satisfactorios (ver apartado 6. Resultados).

**MANUAL 5. PASOS REALIZADOS EN LA INSTALACIÓN DE LA APP.**

**FUENTE: ELABORACIÓN PROPIA**