



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

DISEÑO DE UNA APLICACIÓN WEB PARA LA SIMULACIÓN DEL RUIDO EMITIDO EN EL AEROPUERTO DE MÁLAGA

Alumno: Iván Bautista Moreno

Tutor: Prof. D. José Alberto Maroto Centeno
Depto.: Departamento de Física

Junio, 2014

Índice general

I	Memoria del documento	1
1.	Resumen	2
2.	Introducción	3
3.	Objetivos	5
4.	Materiales y métodos	6
4.1.	Lenguajes de desarrollo	6
4.2.	Lenguajes de marcas	8
4.3.	Gestores de bases de datos	8
4.4.	Servidores web	9
5.	Descripción del trabajo	11
5.1.	Arquitectura del trabajo	11
5.2.	Introducción sobre la teoría del ruido	13
5.3.	Desarrollo de la aplicación	16
5.3.1.	Maquetado visual de la aplicación	16
5.3.2.	Lenguaje para el control de operaciones de la aplicación	19
6.	Conclusiones	26
7.	Glosario	27
A.	Manual de instalación	29
B.	Manual de usuario	32
C.	Código fuente	35
C.1.	Código del fichero <code>index.html</code>	35
C.2.	Código del fichero <code>app.html</code>	40

II	Pliego de condiciones	52
III	Estudio económico	54

Índice de figuras

5.1. Estructurado de la aplicación	12
5.2. Arquitectura C/S	12
5.3. Mapa aeropuerto de Málaga	14
5.4. Gráfica de representación de los valores reales L_p	15
A.1. Opción para habilitar el contenido activo de un archivo	30
A.2. Opción para habilitar el contenido activo de un CD	31
B.1. Ventana principal de la aplicación	32
B.2. Ventana flotante	33
B.3. Ventana del simulador	33
B.4. Inicio de una simulación	34

Índice de tablas

5.1. Tabla de resultados	14
B.1. Inicio de aplicación	32
C.1. Costes generales	55
C.2. Costes de mano de obra	55
C.3. Costes totales	55

I

Memoria del documento

1

Resumen

Este trabajo fin de grado recoge la información necesaria para el desarrollo, instalación y uso de una aplicación de interés didáctico desarrollada para las prácticas de laboratorio de la asignatura de Física I, impartida en la Escuela Politécnica Superior de Linares (Universidad de Jaén).

Esta aplicación consiste en un simulador del nivel de ruido generado por un avión en su etapa de despegue y medido en distintos zonas puntos al aeropuerto de Málaga.

2

Introducción

Esta aplicación web está orientada para servir de complemento en la docencia práctica de la asignatura Física I para los grados de Ingeniería Civil, Ingeniería de Tecnologías Mineras e Ingeniería de Recursos Energéticos impartidos en la Escuela Politécnica Superior de Linares (Universidad de Jaén). De manera más concreta podemos decir que está pensada para configurar una sugestiva práctica de laboratorio, mediante la simulación del ruido generado en el aeropuerto de Málaga durante el despegue de un avión. Dado que está dirigida a alumnos de primer curso universitario se ha cuidado al máximo la interfaz de la misma, que resultará cómoda e intuitiva. Para el futuro graduado en ingeniería será una experiencia muy atractiva la evaluación del nivel de ruido provocado por el despegue de un avión en diez puntos situados a diferentes distancias, de forma que muchos de ellos están ubicados en núcleos urbanos colindantes. Para hacer más real la simulación y a fin de adiestrar al estudiante, los valores en dB(A)^1 generados en cada experiencia para una posición dada, no son siempre los mismos, sino que oscilan en torno a un valor central (que es el valor esperado). De esta forma, el alumno tendrá la oportunidad de aplicar la teoría de errores y de tratamiento de datos experimentales, es decir, uno de los objetivos principales que se plantea, en su conjunto, el laboratorio de Física en cualquiera de sus cursos y especialidades. En cada experiencia se reproduce el sonido de un avión al despegar y se pueden seguir sus evoluciones en la pista del aeropuerto, lo que hace la simulación más real o, al menos, más atractiva.

Dado que el sistema sobre el que se mide tiene existencia real (el aeropuerto de Málaga) y que el mismo, como se puede comprobar fácilmente, está afectado por problemas reales de ruido debido a la cercanía de núcleos habitados (como, por ejemplo, el barrio de Churriana), esta aplicación web debe servir para evidenciar que la Física sirve para abordar problemas reales y de gran relevancia, a la vez que ilustrará que el conocimiento de las leyes y aplicaciones de la Física son importantes para un graduado en Ingeniería.

Finalmente, es necesario precisar que se simula el ruido generado durante el despegue de un avión no de forma continua, sino en un punto dado (justo al abandonar el contacto con la pista) que es en torno al cual se genera un ruido más intenso. La medida del nivel de ruido generado de forma continua durante la carrera del despegue es un problema mucho más complejo, que se escapa a los objetivos didácticos planteados en la asignatura Física I y que, por tanto, ha sido obviado. En cualquier forma su predicción teórica no es un problema sencillo ya que el nivel de potencia sonora emitida varía continuamente según una regla no fácilmente cuantificable. Por el contrario y en aras a simplificar el problema se ha asumido que en el momento de dejar tierra el avión emite un nivel de potencia sonora de 140 dB(A).

En cuanto a la memoria del trabajo fin de grado, está dividida en los siguientes apartados:

1. Resumen.
2. Introducción.
3. Objetivos. Se describen los objetivos del trabajo fin de grado de manera escueta.
4. Materiales y métodos. Se comentan distintas tecnologías utilizadas para la simulación de contenido multimedia en aplicaciones web.
5. Descripción del trabajo. Aquí hablamos con todo detalle del desarrollo de la aplicación así como las dificultades que han aparecido y como han sido solventados.
6. Conclusiones. Se realiza una enumeración de los resultados del trabajo, comparando con los objetivos.
7. Glosario. Definición de alguna terminología técnica y algunos acrónimos usados en esta memoria.
8. Anexo:
 - a) Manual de instalación
 - b) Manual de usuario
 - c) Código fuente

3

Objetivos

Este trabajo fin de grado tiene como principal objetivo proporcionar a los alumnos de primer curso universitario una herramienta para poder entender y trabajar con la ecuación que relaciona la variación del nivel de presión sonora² con la distancia al foco emisor. Las distintas soluciones han sido condicionadas por un interés didáctico, ya que los resultados obtenidos van a ser incorporados a una práctica de laboratorio en el marco de la asignatura Física I para los grados de ingeniería civil, ingeniería de tecnologías mineras e ingeniería de recursos energéticos, impartidas en la Escuela Politécnica Superior de Linares. Por lo tanto, se ha marcado el objetivo de crear una aplicación que cumpla los siguientes requisitos:

- Análisis de la ecuación que expresa la variación del nivel de presión sonora (en decibelios³) con la distancia y teniendo en cuenta la absorción del aire.
- Análisis del plano del aeropuerto de Málaga y de los núcleos urbanos colindantes para la elección de los puntos óptimos de medida del nivel de presión sonora.
- Elección del instante de despegue del avión como óptimo para la medida del nivel de presión sonora.
- Diseño de una aplicación web con una interfaz minimalista que resulte cómoda e intuitiva para los alumnos de primer curso universitario.

4

Materiales y métodos

Cuando hablamos de aplicaciones web siempre nos viene a la cabeza páginas web, pero una aplicación web no es una simple página a la que accedemos desde un navegador. Una aplicación web está formada por un conjunto de herramientas software que trabajan entre sí para ofrecer algún tipo de servicio a los usuarios que acceden a ella. Este software debe ser desarrollado en lenguajes que un navegador sea capaz de soportar. La estructura más habitual de una aplicación web es una división entre el almacenamiento de datos, un controlador encargado de toda la lógica entre el usuario y los datos y una interfaz gráfica que es el elemento final que el usuario ve y sobre el que se guía para realizar las distintas tareas para la que ha sido diseñada la aplicación. El estudio se va a dividir en las siguientes partes:

1. Lenguajes de desarrollo.
2. Gestores de bases de datos.
3. Servidores web.

4.1. Lenguajes de desarrollo

En la actualidad existen una gran cantidad de lenguajes de programación para el desarrollo de aplicaciones web pero los más exitosos son los que llevan mucho tiempo en el mercado debido a un mayor soporte y una mayor divulgación. Entre ellos podemos citar los siguientes:

- JavaScript. Es un lenguaje de programación interpretado derivado del estándar ECMAScript⁴. Se define como orientado a objetos⁵, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), aunque existe una forma del lado del servidor (Server-side) algo más compleja. JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje Java. Sin embargo, Java y JavaScript son

totalmente diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web que se provee al lenguaje JavaScript de una implementación del DOM⁶ para interactuar con los elementos de esta.

Existen gran multitud de librerías en javascript que nos sirven de ayuda a la hora de desarrollar, pues nos evitan reescribir funciones comunes que pueden necesitar un desarrollador una gran cantidad de veces. Una de las librerías que se han hecho imprescindibles a la hora del desarrollo web es jQuery, una librería en JavaScript rápida y concisa que simplifica el uso de elementos HTML y manejo de eventos.

- **JAVA.** Java es un lenguaje de programación basado en clases y orientado a objetos de propósito general. En la actualidad es un lenguaje muy extendido y usado en el ámbito tanto de Internet como en la informática. Desarrollada por la compañía Sun Microsystems, la cual fue adquirida posteriormente por Oracle, con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras. Java se divide en tres plataformas, cada una orientada a cubrir un entorno de aplicación específico:
 - Java ME (Java Platform, Micro Edition) o J2ME está orientada a entornos de limitados recursos, como teléfonos móviles, PDAs, etc.
 - Java SE (Java Platform, Standard Edition) o J2SE está pensado para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario medio en un PC de escritorio.
 - Java EE (Java Platform, Enterprise Edition) o J2EE está orientada a entornos distribuidos empresariales o de Internet.
- **ASP (Active Server Pages):** Es el lenguaje de programación propio de Microsoft para la creación de páginas dinámicas. Utiliza un lenguaje de scripts que se colocan en la misma página web junto con el código HTML. El lenguaje de scripts usado es Visual Basic Script, que deriva del conocido Visual Basic, aunque también se pueden usar otros lenguajes, como JScript, derivado del conocido Javascript.
- **PHP.** Es un lenguaje del lado de servidor con gran potencia y simplicidad que lo caracterizan, además cuenta con un soporte generalizado en la mayoría de los servidores de hosting. PHP nos permite combinarlo con HTML y realizar determinadas acciones de una forma fácil y eficaz. Se puede decir que con PHP escribimos scripts dentro del código HTML.

Por otra parte, y es aquí donde reside su mayor interés con respecto a los lenguajes pensados para los CGI⁷, PHP ofrece una inmensa cantidad de funciones para

la explotación de bases de datos de una manera sencilla y sin la necesidad de importar paquetes externos a su API⁸.

4.2. Lenguajes de marcas

- **HTML.** Es un lenguaje basado en marcas que permite escribir texto de forma estructurada con etiquetas que marcan el inicio y el fin de cada elemento del documento. El documento no sólo se compone de texto también puede estar formado por elementos multimedia como audio, vídeo e imágenes. Los archivos deben tener la extensión html o htm, para que puedan ser visualizados en los navegadores, éstos se encargan de interpretar el código HTML y mostrar a los usuarios las páginas web resultantes del código generado.
- **XML.** Es un metalenguaje extensible que permite definir la gramática de lenguajes específicos para el almacenamiento de información de manera estructurada y entendible. Los archivos XML suelen ocupar poco tamaño por ello se suelen usar para el intercambio información, pero también pueden generar una página web transformando estas etiquetas
- **CSS.** No es un lenguaje de marcas, pero van unidas ya que este lenguaje es usado para definir la presentación de un documento escrito en HTML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación, al igual que ocurre con otros tipo de lenguajes como son el ejemplo de XML y XLS.

4.3. Gestores de bases de datos

Para la implementación de una página web muchas veces es necesaria la inclusión de una base datos para albergar cualquier tipo de información que podrá ser utilizada de forma dinámica. A continuación se mostrarán diversos sistemas de gestión de bases de datos existentes:

- **Microsoft SQL server:** Es un sistema para la gestión de bases de datos desarrollado por Microsoft. Su gestión está basada en gestión de políticas y entre sus características se encuentra el soporte de transacciones, de procedimientos almacenados además de escalabilidad, estabilidad y seguridad. Requiere como sistema operativo Windows. Puede trabajar conjuntamente con Microsoft Access para una realización de aplicaciones más complejas.
- **Microsoft Access:** Forma parte de la suite ofimática Office. Es un sistema de datos relacionales desarrollado por Microsoft para un entorno personal o en pequeñas

organizaciones. Las bases de datos están almacenadas como fichero que pueden ser fácilmente gestionadas por una interfaz gráfica simple. Además, estas bases de datos pueden ser consultadas por otros programas desde lenguajes como Visual Basic. Es necesario tener instalado el sistema operativo Microsoft Windows.

- MySQL: Este sistema de gestión de datos no contiene una interfaz gráfica pero puede ser administrada gráficamente con herramientas como phpmyadmin o mysql workbench. Es muy configurable y entre sus características destacan un gran conjunto de sentencias SQL que permiten un manejo de datos muy sencillo. Puede ser controlado desde multitud de lenguajes de programación gracias a los conectores existentes como JDBC, por ejemplo, para la conexión con Java.

4.4. Servidores web

Un servidor web es un software para ofrecer páginas web como respuesta a la petición de los clientes mediante el protocolo HTTP. Esto significa la entrega de los documentos HTML y cualquier contenido adicional que se puede incluir en un documento, como imágenes, hojas de estilo y scripts. El término también se emplea para referirse al ordenador que ofrece servicios y recursos a otros ordenadores que forman parte de una red. Entre los más importantes del mercado se encuentran:

- Apache: Es uno de los servidores más populares. Se desarrolla dentro del proyecto httpd de la Apache Software Foundation. Es de código abierto y muy seguro ya que la mayoría de las vulnerabilidades de seguridad descubiertas han sido resueltas de manera que tan sólo pueden ser aprovechadas por usuarios locales. La arquitectura de Apache se basa en módulos por lo que es muy configurable aunque no dispone de una interfaz gráfica.
- Tomcat: Es un servidor de aplicaciones utilizado para trabajar con Java en entornos webs, basado en las implementaciones de servlets y JSPs. También se puede usar como servidor pero con un peor rendimiento.
- Internet Information Services (IIS): Es un servidor web de Microsoft que ofrece una gran fiabilidad y un conjunto de servicios para su sistema operativo como FTP, SMTP y HTTP/HTTPS. Basado en módulos muy configurables que le dan capacidad para procesar distintos tipos de páginas pero especialmente las desarrolladas en ASP.
- Cherokee: Es un servidor web gratuito con licencia GPL y multiplataforma creado por el madrileño Álvaro López Ortega y actualmente lo mantiene Octality, de la que también es fundador. Está escrito en C y diseñado para ser mucho más liviano

que los mencionados anteriormente. Soporta PHP, SSL, TLS, CGI, Autenticación, etc. Todo el mundo puede contribuir en este proyecto y quién logra implementar alguna utilidad tiene recompensa económica que sale de patrocinadores como DynDNS.

- Algunos servicios web, como Google Drive, pueden ser utilizados como hosting⁹ para almacenar aplicaciones como la que estamos presentando en este proyecto que no requieren de procesamiento ni compilación de código ya que el encargado de realizar esto será el navegador.

5

Descripción del trabajo

En este capítulo analizamos en detalle el desarrollo de este trabajo fin de grado. La descripción está estructurada en los siguientes puntos:

1. Arquitectura del trabajo, que da una visión general de los elementos principales que intervienen en la aplicación.
2. Introducción sobre la teoría del ruido, donde se da un conocimiento previo para hacer entendible el posterior desarrollo del trabajo fin de grado.
3. Desarrollo de la aplicación, donde se explica la metodología del desarrollo y su lógica en los siguientes puntos:
 - a) Maquetado visual de la aplicación
 - b) Lenguaje para el control de operaciones de la aplicación

5.1. Arquitectura del trabajo

En este primer punto queremos dar una visión global de la estructura del trabajo y su funcionamiento. Lo primero que se describe es cómo esta aplicación hace uso de los distintos recursos (podemos ver los tipos de ficheros que utilizamos en la imagen 5.1); para ello esta aplicación está formada por dos archivos principales, los cuales se encuentran en el directorio raíz, y junto con ellos hay unas carpetas donde se encuentran los archivos secundarios y que se utilizan como apoyo a la lógica de la aplicación.

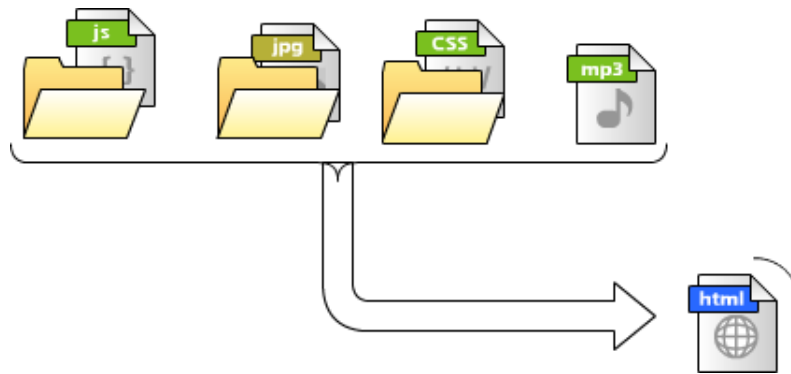


Figura 5.1: Estructurado de la aplicación

El fichero que inicia la aplicación tiene el nombre de `index.html`. Al iniciar este fichero se presenta una interfaz simple con un fondo del aeropuerto de Málaga sacado a través de la funcionalidad de una herramienta 3D para caracterizar nuestra aplicación. Este fichero simplemente tiene una pequeña información del nombre del proyecto con un botón para abrir una nueva ventana personalizada de una manera especial donde se carga el simulador. Aparte, también tiene una pequeña información del autor del trabajo y su profesor de apoyo.

Al ser una aplicación web, su funcionamiento puede ser muy variado aunque la arquitectura de funcionamiento más común es la establecida como cliente-servidor, donde la aplicación es almacenada en un equipo central llamado servidor en el cual existe un software especial que es capaz de interpretar el código y devolver una respuesta. En este tipo de arquitectura los clientes se conectan para descargarse los archivos necesarios para hacer funcionar la aplicación.

Sin embargo, nuestra aplicación **no es necesario** que se almacene en **un servidor** capaz de interpretar el código y devolver una respuesta al cliente. Esta aplicación es capaz de ejecutarse de forma local, ya que todo el código es capaz de ser cargado e interpretado por un simple navegador. **Tampoco necesitamos el uso de una base de datos**, ya que datos necesarios para las simulaciones están definidos en variables interiores del código. Con estas funcionalidades creamos una aplicación fácil de administrar, ya que **no es necesario tener conocimientos sobre bases de datos ni servidores**.

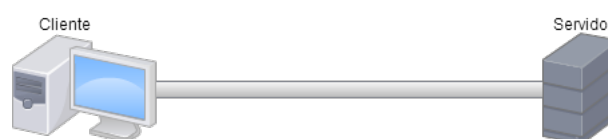


Figura 5.2: Arquitectura C/S

5.2. Introducción sobre la teoría del ruido

Lo primero que debemos responder es; ¿qué es el ruido?. Según la RAE, el ruido es un sonido inarticulado que por lo general es desagradable para el oído humano. En nuestro ambiente existen muchos ruidos provocados por distintas fuentes y con distintos niveles de potencia sonora. Para medir el ruido generado por estas fuentes existen instrumentos como los sonómetros que dan una medida aproximada del nivel de presión sonora generado en un punto.

El **umbral de audición** representa la cantidad mínima de sonido o de vibraciones por segundo requeridas para que el sonido lo pueda percibir el oído humano.

El nivel de presión acústica se mide en decibelios (dB) y se debe precisar que los ruidos especialmente molestos son los que corresponden a los tonos altos. Existe un acuerdo en que el ruido es dañino a partir de los 70-75 dB, que es el ruido que puede existir en una gran ciudad con mucho tráfico; el ruido causa dolor para el oído humano a partir de los 120 dB, que puede ser el ruido provocado a 1 metro de distancia en el despegue de un avión a reacción. Para hacernos una idea del daño que puede causar el ruido al oído humano podemos decir que el sistema auditivo requiere alrededor de 16 horas de reposo para reponerse tras 2 horas de exposición a 100 dB.

La ecuación de trabajo que relaciona la variación del nivel de presión sonora L_P con la distancia r a un foco sonoro que emite con un nivel de potencia sonora L_W viene dada por:

$$L_P = L_W - 20 \cdot \text{Log}_{10} r - 11 - \gamma \cdot r \quad (5.1)$$

Donde:

- L_P = Nivel de presión sonora, expresada en decibelios (dB).
- L_W = Nivel de potencia sonora, también en decibelios. Hemos tomado el valor de 140 dB, nivel aceptado para el ruido emitido por un avión durante el despegue.
- r = Es la distancia que separa el foco emisor que emite L_W al punto considerado, donde se mide un nivel de potencia L_P . Expresado en metros.
- γ = Coeficiente de atenuación del aire. Depende de la frecuencia del sonido considerado y de otros factores como la temperatura ambiente y la humedad del aire. Se expresa en dB/m. El valor tomado es 0.0025 dB/m, coeficiente de atenuación propio de un sonido de baja frecuencia.

La expresión 5.1 se interpreta físicamente en el sentido de que el nivel de potencia sonora emitido por una fuente se atenúa debido a la dispersión geométrica (distancia) y a la absorción del sonido por parte del aire. La figura 5.3 muestra el aeropuerto de Málaga, donde aparecen los 10 puntos seleccionados en los que se obtendrán las medidas del ruido generado por el avión en la posición donde se haya la bandera a cuadros.

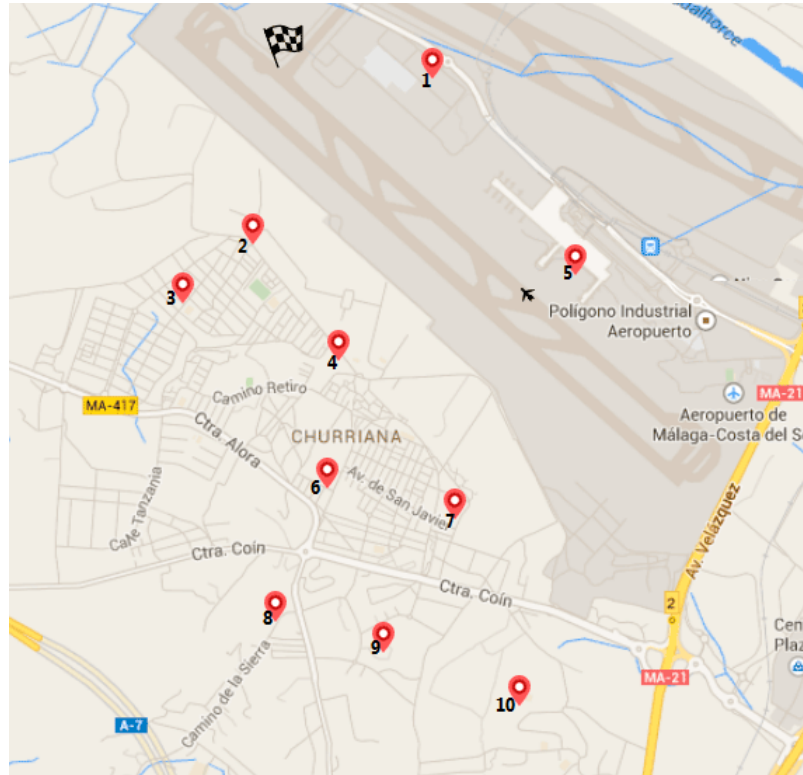


Figura 5.3: Mapa aeropuerto de Málaga

En la tabla 5.1 podemos ver los valores reales calculados mediante la ecuación 5.1.

Punto	Distancia en centímetros	Distancia en metros	Valor real de L_p
1	5	714,5	70,13371
2	6	857,4	68,19283
3	8	1143,2	64,97956
4	10	1429	62,32686
5	12	1714,8	60,02873
6	14	2000,6	57,97529
7	16	2286,4	56,10096
8	18	2572,2	54,36341
9	20	2858	52,73376
10	22	3143,8	51,1914

Tabla 5.1: Tabla de resultados

A continuación, realizamos una representación que visualiza la tendencia entre la presión sonora y la distancia.

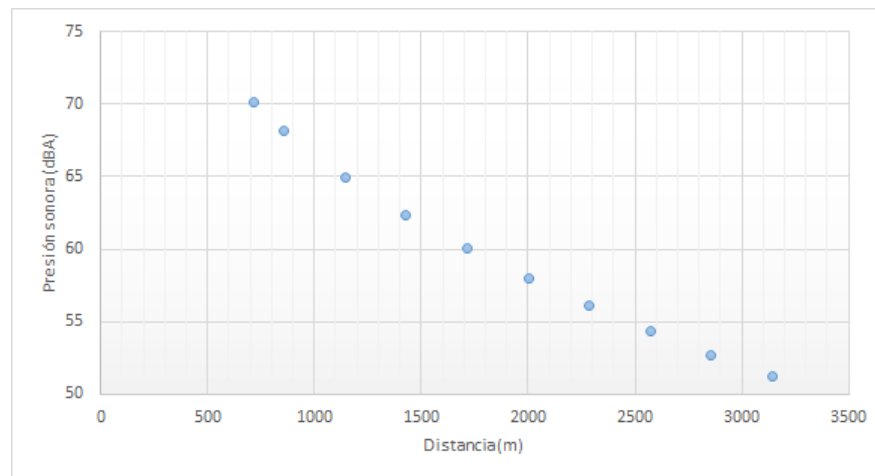


Figura 5.4: Gráfica de representación de los valores reales L_p

5.3. Desarrollo de la aplicación

En este punto hablamos del desarrollo de todas las clases necesarias para el funcionamiento y diseño de la aplicación web así como de los distintos lenguajes utilizados en su desarrollo, aunque hoy en día existen muchos lenguajes para el desarrollo de aplicaciones, estos últimos años han aumentado exponencialmente las aplicaciones desarrolladas con la nueva versión de html, junto con la nueva versión de CSS y el cada día más avanzado javascript. Es tal el crecimiento de aplicaciones con estos lenguajes que poco a poco grandes compañías van migrando sus servicios a esta tecnología, como por ejemplo, el portal de vídeo YouTube del gigante Google. En nuestro trabajo vamos a utilizar estas tres tecnologías para alcanzar los objetivos marcados.

5.3.1. Maquetado visual de la aplicación

Como ya hemos comentado en el punto 5.1, cuando hablamos de la arquitectura, la aplicación está dividida en dos interfaces sencillas de cara a el usuario para un uso sencillo, intuitivo y de fácil comprensión. Ambos ficheros (*index* & *app*) están escritos en el lenguaje de marcas llamado HTML 5 que nos permite ofrecer los recursos que necesitamos para nuestro trabajo de una manera sencilla.

Empezamos hablando por la estructura del fichero principal (*index.html*), el cuál está definido mediante la siguiente etiqueta que nos proporciona un tamaño de la imagen de fijo y siempre adaptado al tamaño de la ventana.

```
<body style="position: static; margin-top: 20%">
```

Luego tenemos definida una etiqueta para mostrar el título del trabajo fin de grado

```
<h2>Aplicación para la simulación del ruido emitido en el  
aeropuerto de Málaga</h2>
```

Aparte, en este fichero encontramos un botón central que se define mediante la siguiente etiqueta, en el cuál podemos ver que tiene asociada una función asociada llamada *app()*, definida en javascript y que, como veremos más adelante, enlaza a la función que nos permite crear una ventana totalmente personalizada con un estilo definido para el simulador. Por ejemplo, sin barras de herramientas ni direcciones, con tamaño de ventana fijo.

```
<button class="myButton" onclick="app();">Lanzar aplicaci&  
oacute;n</button>
```

El fichero del simulador (`app.html`) está dividido: por una parte tiene todos los marcadores necesarios para mostrar la información que el alumno requiere para la realización de las prácticas mediante la definición de las siguientes etiquetas.

```
<div id="par" style="padding-left:5%">
  <div class="auto-style4" style="width: 310px;">
    <div id="jqRadialGauge" style="width: 260px; height: 80px
      ;"></div>
    <div id="jqSegmentedDisplay" style="width: 260px; height: 70
      px;"></div>
    <div id="jqDistanceDisplay" style="width: 260px; height: 70px
      ;"></div>
  </div>
```

Y por otra parte encontramos el mapa sobre el cuál están situados los puntos que se deben pinchar para iniciar las distintas simulaciones. En esta parte del maquetado hemos utilizado una etiqueta que ya estaba implementada en la antigua definición de HTML. Esta etiqueta se trata de `<map>`, que nos permite crear dentro de una imagen formas geométricas situadas en unas coordenadas concretas y con un forma determinada con la singularidad de que al pinchar sobre estos puntos podremos comenzar la ejecución de las simulaciones.

Para nuestro uso concreto, hemos implementado pequeños rectángulos que bordearán los iconos rojos que nos permiten distinguir los puntos desde donde se mide el ruido durante el despegue del avión.

```
<div id="map" style="height: 37px;padding-top:230px;">
  <p style="position: relative; left: 40%; top: -400px; width:
    572px; height: 553px">
    
    <map name="airportmap">
      <area href="javascript:f(0);" alt="Punto 1" shape="rect"
        coords="295,30,310,55" />
      <area href="javascript:f(1);" alt="Punto 2" shape="rect"
        coords="165,151,182,175" />
      <area href="javascript:f(2);" alt="Punto 3" shape="rect"
        coords="115,191,132,215" />
      <area href="javascript:f(3);" alt="Punto 4" shape="rect"
        coords="394,171,412,195" />
      <area href="javascript:f(4);" alt="Punto 5" shape="rect"
        coords="227,233,242,255" />
      <area href="javascript:f(5);" alt="Punto 6" shape="rect"
```

```

        coords="218,325,235,345" />
<area href="javascript:f(6);" alt="Punto 7" shape="rect"
    coords="310,344,328,364" />
<area href="javascript:f(7);" alt="Punto 8" shape="rect"
    coords="181,418,195,445" />
<area href="javascript:f(8);" alt="Punto 9" shape="rect"
    coords="259,442,274,464" />
<area href="javascript:f(9);" alt="Punto 10" shape="rect"
    coords="354,479,375,499" />
</map>
</p>
</div>

```

Otra de las implementaciones es el efecto del sonido que se maneja desde javascript pero que está definido en el código HTML con una etiqueta que ha sido revolucionaria en la nueva versión de HTML5. Esta nueva etiqueta se define como `<audio>`, y proporciona una gran mejora para el control de estos archivos multimedia en las distintas aplicaciones web.

```

<audio id="audio" >
    <source src="av.mp3" type="audio/mpeg">
    Este navegador no es compatible. Avisa a tu administrador de
        sistemas para instalar un navegador de última generación.
</audio>

```

Aparte de todos los elementos que necesitamos para el funcionamiento de la aplicación, se ha definido un apartado donde hay dos enlaces que nos llevan a información complementaria para el uso de la herramienta y realización de la práctica. Estos dos apartados son *Instrucciones de uso & Aspectos teóricos del ruido*. Se han definido de la siguiente manera en etiquetas html con un formato muy concreto, mediante una clase CSS llamada *footer*, que permite mantener los enlaces situados en una posición fija, junto al borde inferior izquierdo:

```

<div class="footer">
    <p id="uses" class="links"><a href="#" style="color: rgba
            (0, 50, 250, 0.74);text-decoration: none;"> Instrucciones
            de uso</a></p>
    <p id="theory" class="links"><a href="#" style="color:
            rgba(0, 50, 250, 0.74);text-decoration: none;"> Aspectos
            teóricos del ruido emitido</a></p>
</div>

```

5.3.2. Lenguaje para el control de operaciones de la aplicación

En este apartado se hace referencia al principal lenguaje utilizado para el desarrollo de la aplicación web y sus motivos. **Javascript** es el lenguaje por antonomasia de estos últimos años y cada día vemos que su futuro está más que garantizado debido a las funcionalidades que aporta, sobretodo por ser un lenguaje muy flexible y que permite llevar una gran cantidad de tareas de una forma sencilla. Por otro lado, **no necesita** de un servidor que lo **interprete** como ocurre con otros lenguajes de programación como java. Esto último es el principal motivo de la elección de este lenguaje de cara al desarrollo de la lógica de esta aplicación. Este lenguaje se divide en dos tipos:

1. El código script que se ejecuta en el cliente (**client-side**); se utiliza para desarrollar la aplicación que incluye los objetos y eventos necesarios para trabajar con formatos, inputs, textareas, documents, etc...
2. El código script que se ejecuta en el servidor (**server-side**); normalmente sustituido por php debido a su complejidad.

Como nuestro código simplemente se ejecuta en los navegadores, podemos decir que el tipo de código script utilizado es del tipo *client-side*.

Como comentábamos en el apartado anterior, en el fichero principal hemos hablado de una función que nos permite crear una ventana personalizada. Las instrucciones aquí definidas nos permiten trabajar con el *DOM Window* haciendo uso de su método *open()*, el cuál nos facilita lanzar una nueva ventana con las propiedades especificadas más adelante.

```
function lanzarApp(){
    var opciones="toolbar=no,location=no,directories=no,status=no
        ,menubar=no,scrollbars=no,resizable=NO,width=1024,height
        =650,top=10,left=140";
    window.open('app.html',' ', opciones);
}
```

Las propiedades definidas en la variable *opciones*, nos indican lo siguiente:

- toolbar = no → Oculta la barra de herramientas.
- location = no → Oculta el campo de localización.
- directories = no → Oculta los botones de dirección.
- status = no → Oculta la barra de estado.
- menubar = no → Oculta las barra de menús.
- scrollbars = no → Oculta las barras de desplazamiento.

-
- `resizable = no` → Establece que la nueva ventana no podrá ser cambiada de tamaño (con el ratón).
 - `width = 1024 px`. Nos indica el ancho de la ventana en pixels.
 - `height = 650 px`. Nos indica el alto de la ventana en pixels.
 - `left = 140 px`. Nos indica la distancia en pixels desde el lado izquierdo de la pantalla a la que se debe colocar la ventana.
 - `top = 10 px`. Nos indica la distancia en pixels desde el lado superior de la pantalla a la que se debe colocar la ventana.

Para el código del fichero encargado de la lógica del simulador primero hacemos referencia a las librerías utilizadas, como por ejemplo jQuery, que ya hemos definido en el punto 4.1. La inclusión de esta librería nos permite utilizar un plugin¹⁰ que ha sido modificado y que se llama *jqGauges*. Este plugin nos ha facilitado la definición de unos marcadores útiles para mostrar los resultados de las simulaciones de una manera clara y muy vistosa. Para su configuración, las líneas a añadir son las siguientes:

```
$_id).jqSegmentedDisplay({
    background: '#DCDCDC',
    border: {
        padding: 10,
        lineWidth: 4,
        strokeStyle: '#76786A'
    },
    digits: 9,
    segmentMode: 'fourteenSegment',
    text: 'texto_a_mostrar_inicialmente',
    textForeground: '#333333',
    textForegroundUnlit: '#DCDCDC'
});
```

Mediante este código hemos definido los marcadores digitales y que hemos definido como parámetro los colores que deben llevar elementos tales como el fondo del marcador, borde, texto mostrado, etc.

Para modificar el valor del texto, según avanza la simulación, se realiza una tarea recursiva programada cada 1000 ms (1s) que modifica el texto según los valores que va tomando el parámetro *result*. Este parámetro se identifica con la medida de la respectiva operación.

```
/*
```

```

* updateDisplay() - Es la función para representación del
  marcador digital.
*/
function updateDisplay(result) {
    $(_id).jqSegmentedDisplay('option', 'text', "0"+ result
        +" dbA");
}

```

Para realizar la tarea recursiva, se hace a través del siguiente método que trae consigo javascript y que se define como:

```

var nombre_task = SetInterval('Función a llamar',1000);

```

Aquí hemos hablado de la tarea recursiva que nos facilita la modificación del valor del marcador digital de la medida que va tomando el sonómetro. A continuación, hacemos referencia a la parte del código que nos permite mostrar el nivel de presión sonora en el marcador analógico.

```

$(_id).jqRadialGauge({
    background: '#DCDCDC',
    border: {
        lineWidth: 6,
        strokeStyle: '#76786A',
        padding: 16
    },
    shadows: {
        enabled: true
    },
    anchor: {
        visible: true,
        fillStyle: anchorGradient,
        radius: 0.10
    },
    tooltips: {
        disabled: false,
        highlighting: true
    },
    animation: { /// Tiempo en avanzar la aguja hasta su valor
        correcto
        duration: 1
    },
    annotations: [ //Este es la parte de código que nos permite
        { //mostrar texto en el marcador
            text: 'Ruido(dBA)',

```

```

        font: '10px sans-serif',
        horizontalOffset: 0.5,
        verticalOffset: 0.80
    }
],
scales: [ // Aquí defino los valores que el marcador
    tendrá
    {
        minimum: 0,
        maximum: 100,
        startAngle: 180,
        endAngle: 360,
        majorTickMarks: {
            length: 12,
            lineWidth: 2,
            interval: 10,
            offset: 0.84
        },
        minorTickMarks: {
            visible: true,
            length: 8,
            lineWidth: 2,
            interval: 2,
            offset: 0.84
        },
        labels: { // Nos permite establecer los
            valores
            orientation: 'horizontal', // que se mostrarán
            interval: 10, // con unas determinadas
            características
            offset: 1.00
        },
        needles: [
            { // Tipo de aguja a usar
                value: 0,
                type: 'pointer',
                outerOffset: 0.8,
                mediumOffset: 0.7,
                width: 10,
                fillStyle: '#252E32'
            }
        ]
    }
]

```

```

],
  ranges: [    /// Nos permite programar los colores
    {    /// que se mostrarán según los valores
      outerOffset: 0.82,
      innerStartOffset: 0.76,
      innerEndOffset: 0.68,
      startValue: 40,
      endValue: 70,
      fillStyle: gradient1
    },
    {
      outerOffset: 0.82,
      innerStartOffset: 0.68,
      innerEndOffset: 0.60,
      startValue: 70,
      endValue: 100,
      fillStyle: gradient2
    }
  ]
}
]
});

```

La función que establece el valor según el resultado es la siguiente:

```

/*
 * updateLights() - Es la función para representar en el
 *   marcador analógico.
 */
function updateLights(result){
  var scales = $('#jqRadialGauge').jqLinearGauge('option',
    'scales');
  scales[0].needles[0].value = result;
  $('#jqRadialGauge').jqLinearGauge('update');
}

```

A continuación mostramos la definición de la ecuación que vincula el nivel de presión sonora definido en la ecuación 5.1. Como podemos apreciar, existe una función denominada $Lp()$. Esta función nos devuelve el cálculo del nivel de presión sonora en función de la distancia. Además, se puede ver que hay una variante en la función que se presenta como la suma del resultado de otra función que nos devuelve siempre un valor aproximado al valor teórico, pero siempre con un margen de error. De esta forma generamos una medida más cercana a la realidad. Esta variación X está comprendida

en el intervalo $[-0.5, 0.5]$.

Además podemos observar que existe una función llamada *log10()*, ya que en javascript el único método definido para el logaritmo es el logaritmo neperiano, y mediante las reglas de los logaritmos hemos diseñado una función que nos devuelve el valor para un logaritmo en base 10.

```
/*
 * Lp - Es la ecuación que relaciona la variación del nivel de
       presión sonora con la distancia a un foco sonoro.
 * parámetros : r - Es la distancia entre el foco emisor y el
       punto donde medimos.
 */
function Lp(r){
    var result=(140-20*log10(r)-11-0.0025*r)+getRandomArbitrary
        (-0.5,0.5);
    var n=result.toFixed(1);
    return n;
}
/*
 * log10 - Realiza la conversión del logaritmo neperiano Ln a
       logaritmo en base 10
 * parámetros : x - Número del cual queremos el logaritmo
 */
function log10(x){
    return Math.log(x)/Math.log(10);
}
/*
 * getRandomArbitrary - Devuelve un número aleatorio entre un
       mínimo y un máximo establecido
 * parámetros : min - N° mínimo del limite
 *               max - N° máximo del limite
 */
function getRandomArbitrary(min, max) {
    return Math.random() * (max - min) + min;
}
```

Por otro lado, hay que tener en cuenta que para hacer más sugestiva la simulación, durante la realización de la misma se genera el sonido del despegue que acompaña al movimiento del avión. Para hacer la simulación más real, hemos correlacionado el volumen de este sonido de despegue con el punto utilizado para la simulación, cuya

distancia a la fuente es variable. Para lograr este objetivo se le asigna al punto más cercano a la fuente (punto 1) el volumen sonoro más alto (1). Para realizar el cálculo en el resto de los puntos, hemos asignado un dígito que caracteriza el volumen sonoro proporcional a la distancia, y que está comprendido en el intervalo $[0,1)$.

```
/*
 * MusicOn - Obtiene el objeto audio y empieza a reproducirlo.
 */
function musicOn(d){
    $("#audio")[0].volume = 714.5/d;
    $('#audio')[0].play();
}
```

Para crear el dinamismo de la imagen mediante el despegue del avión, se pensó en primer lugar su realización a través de la definición de la etiqueta canvas que incorpora HTML5. Sin embargo, la forma de trabajo de esta etiqueta no era la más recomendable, ya que su fundamento de trabajo se basa en un bucle de imágenes que da dinamismo dentro de un lienzo definido, y que tiene el problema de crear una gran cantidad de imágenes. Por ejemplo, si simuláramos 1 vez se generarían 40 imágenes; si simuláramos 100 veces la cantidad de imágenes creadas serían de 4000, lo que implicaría el consumo de excesivos recursos y memoria. Por este hecho, y porque canvas aún no es compatible en muchos navegadores, se optó por realizar la siguiente función, la cuál llama recursivamente a imágenes generadas y que van modificándose conforme avanza el tiempo de simulación:

```
function img_canvas() {
    j=j+1;
    $("#airport").attr("src","images/canvas/"+j+".png");
};
```

6

Conclusiones

1. Se ha diseñado un aplicación web que proporciona a los alumnos de primer curso universitario una herramienta útil para entender y trabajar con la ecuación que relaciona la variación del nivel de presión sonora con la distancia al foco emisor.
2. Esta aplicación web toma la forma de una simulación del ruido generado por un avión durante su fase de despegue en el aeropuerto de Málaga, y será utilizada a partir del curso 2014/2015 como una práctica de laboratorio en el marco de la asignatura Física I para los grados de ingeniería civil, ingeniería de tecnologías mineras e ingeniería de recursos energéticos, impartidas en la Escuela Politécnica Superior de Linares (Universidad de Jaén).

7

Glosario

Definición de términos

¹Decibelio ponderado dB(A): Es la unidad del nivel de potencia o intensidad sonora medido con un filtro previo que elimina los sonidos de frecuencias bajas y muy altas, y deja sólo los que mejor son percibidas por el oído humano.

²Nivel de presión sonora, Lp: Es una medida de la cantidad de energía asociada al ruido. (Comentario extraído del Real Decreto 286/2006: Guía técnica para la evaluación y prevención de los riesgos relacionados con la exposición de los trabajadores al ruido.)

³Decibelio (dB): Es una unidad logarítmica que expresa la relación entre dos magnitudes: la magnitud que se estudia y una magnitud de referencia.

⁴ECMAScript: Es un lenguaje de scripting basado en objetos. Es aceptado como estándar en ISO 16262.

⁵Orientado a objetos: Es un paradigma de programación que usa los objetos en sus interacciones, donde los objetos se intercambian mensajes entre sí, cada objeto encapsula sus propios datos y tiene unas acciones específicas diferentes a los demás objetos.

⁶DOM: Acrónimo de Data Object Model. Es la estructura que el navegador genera al cargar un documento.

⁷CGI: Es una interfaz que permite transferir datos entre el cliente y un programa que se ejecuta en un servidor web.

⁸API: Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

⁹Hosting: Servicio para el alojamiento de aplicaciones webs.

¹⁰Plugin: Componente que se añade a una librería para dar nuevas funciones y características.

Anexo A

Manual de instalación

Este manual nos proporciona los pasos a seguir para aplicar la configuración óptima que nos permite tener una fantástica interacción con la aplicación. Esta aplicación está diseñada de una manera especial que permite ejecutarse directamente en un navegador configurado con este manual, por tanto, no requiere de instalación de software adicional, pero los requerimientos descritos en este apartado son **aconsejables de configurar**.

Los requisitos necesarios son:

- Utilizar como **navegador predeterminado Internet Explorer 9 o superior**.
La aplicación ha sido diseñada para este navegador específico, aunque también puede ser utilizada en otros navegadores, aunque no es recomendable.
- Utilizar una versión del sistema operativo **Windows actualizado**, condicionada por el punto anterior, debido que en Windows XP no soporta Internet Explorer 9 o superior.
- Según desde donde se ejecute la aplicación (desde un CD o directamente una copia en el disco duro), debemos habilitar la casilla de Internet Explorer para permitir el código activo en estas ubicaciones sin pedir confirmación. Para ello, a través de *Menú herramientas (Desde el navegador) → Opciones de Internet → Opciones avanzadas* encontramos al final del diálogo las opciones que se muestra a continuación:

- Desde una copia en el ordenador, debemos activar la casilla de ***Permitir que el contenido activo se ejecute en los archivos de mi equipo.***

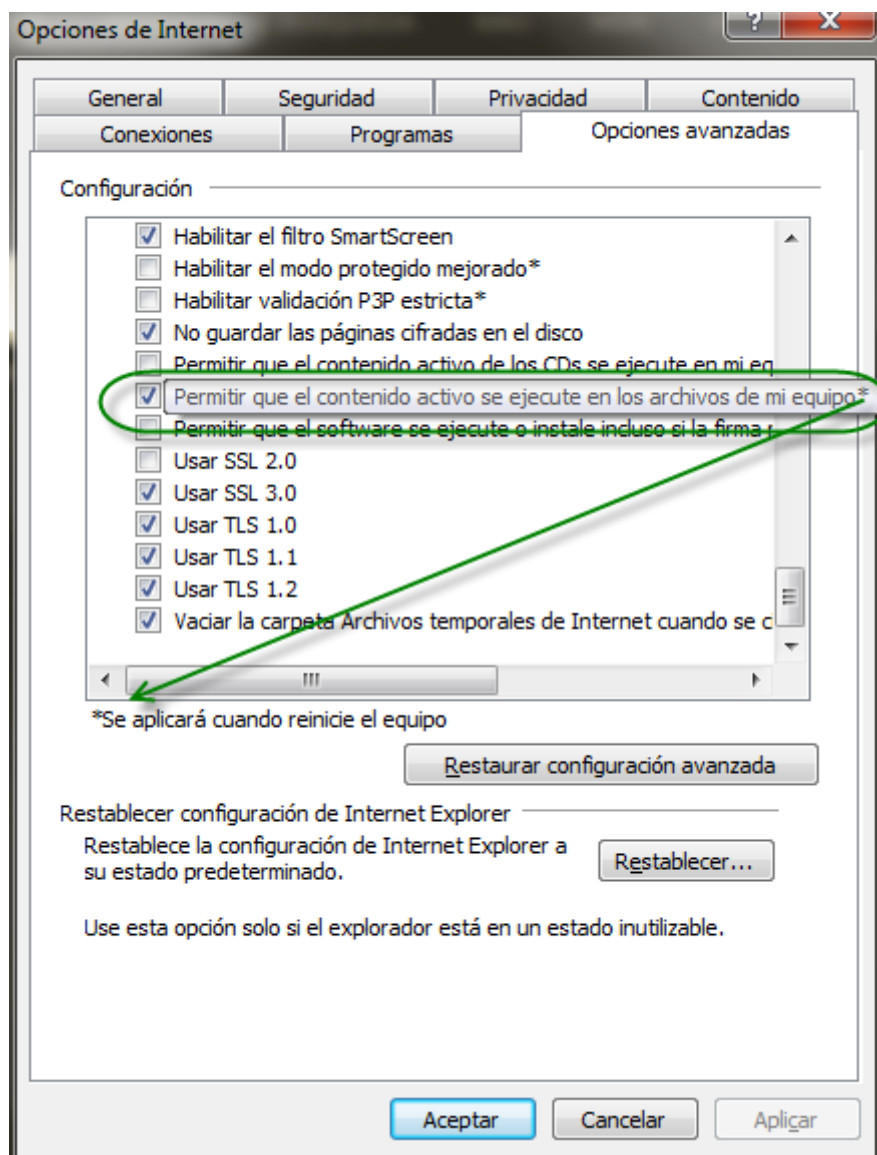


Figura A.1: Opción para habilitar el contenido activo de un archivo

- Desde un CD directamente, debemos activar la casilla de *Permitir que el contenido activo de los CDs se ejecute en mi equipo*.

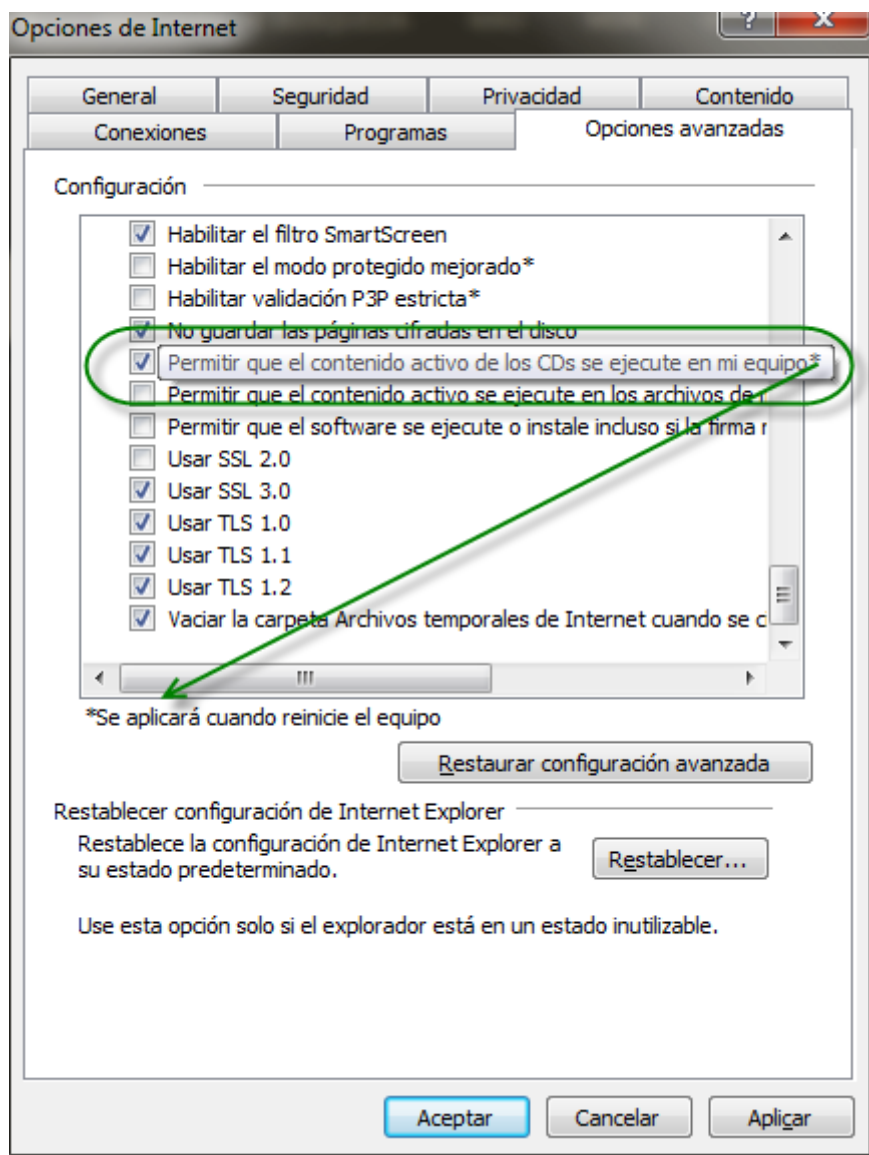


Figura A.2: Opción para habilitar el contenido activo de un CD

¡Recordamos que esta aplicación se puede ejecutar desde otro navegador (Google Chrome o Firefox) pero recordamos que fue diseñada para su uso concreto en **Internet Explorer 9 o superior**!

Anexo B

Manual de usuario

Este manual detalla la forma de utilización de la aplicación. Simplemente ejecutando el archivo `index.html` se abre el navegador y accedemos a la pantalla principal de la aplicación (figura B.1). Para darle más carácter de aplicación se ha desarrollado un pequeño ejecutable para abrir este fichero, que nos permite mostrar en una ventana el contenido del fichero como si se tratase de un navegador, pero con la peculiaridad de no tener barra de dirección, barra de menús, etc. Para utilizar esta aplicación debemos iniciar el ejecutable llamado `Simulador.exe`

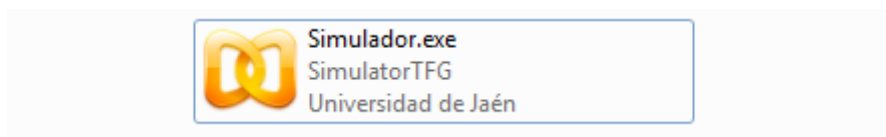


Tabla B.1: Inicio de aplicación

Una vez inicie la interfaz de usuario se muestra la pantalla principal de la aplicación.



Figura B.1: Ventana principal de la aplicación

Se trata de una pantalla inicial al simulador donde podemos apreciar en su aspecto los escudos de la universidad y de la escuela de Linares, el nombre de la aplicación y un botón central que al pulsar que abre una ventana flotante (figura B.2).

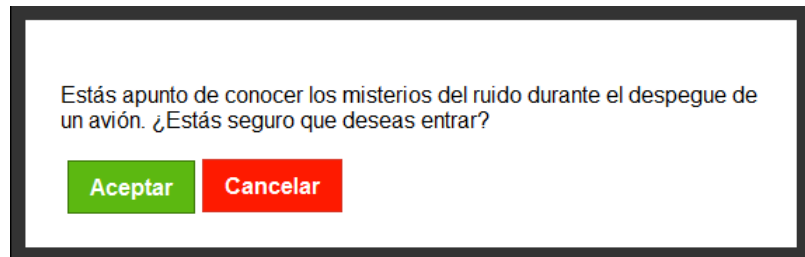


Figura B.2: Ventana flotante

Cuando pulsamos en *Aceptar* se abrirá la ventana personalizada que contiene el simulador.

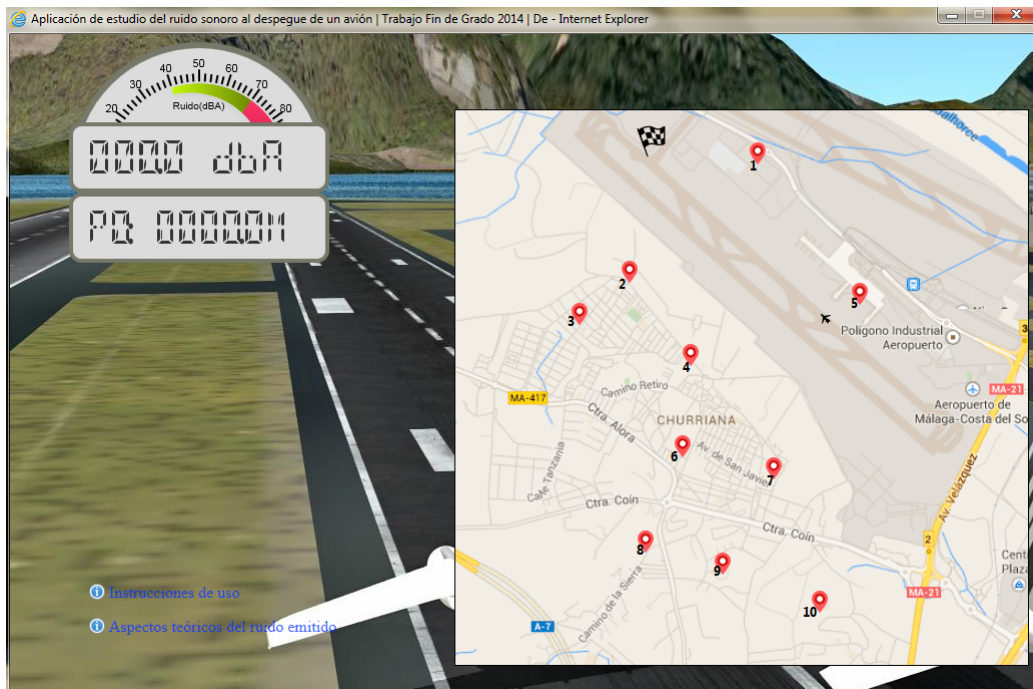


Figura B.3: Ventana del simulador

Una vez tenemos esta pantalla, simplemente para empezar a simular y obtener las medidas, sólo tenemos que pinchar sobre los puntos rojos marcados en el mapa. En los marcadores de la izquierda, empezando desde arriba podemos visualizar los valores correspondientes a cada simulación siguiendo el orden siguiente:

1. Información de la presión sonora (en dB(A)) en el marcador de franjas de colores, que nos permitirá ver si el ruido será molesto o no.
2. Medida cuantitativa del valor exacto de la presión sonora simulada (en dB(A)).
3. Da información sobre el punto simulado y la distancia (en metros) que hay entre este punto y el foco emisor (bandera a cuadros).

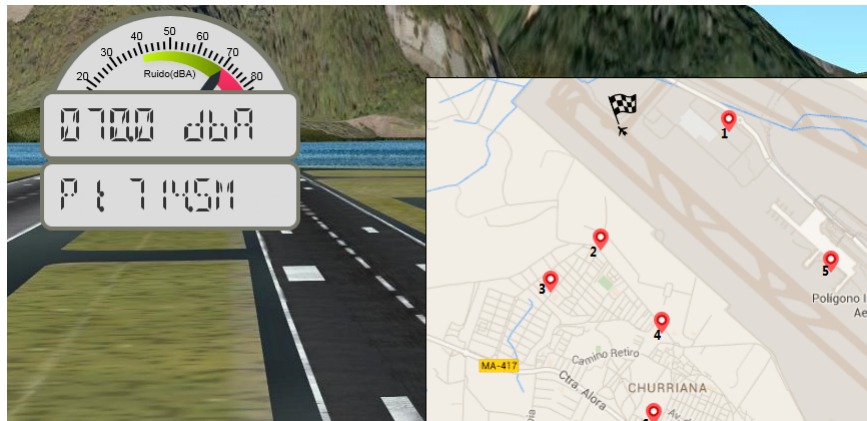


Figura B.4: Inicio de una simulación

Anexo C

Código fuente

C.1. Código del fichero index.html

```
<!DOCTYPE html>
<html>
<meta content="es" http-equiv="Content-Language">
<meta content="Iván Bautista Moreno" http-equiv="author">
<meta content="text/html; charset=utf-8" http-equiv="Content-
  Type">
<link rel="shortcut icon" href="images/1398100892_130429.ico"
  />
<title>Inicio de aplicación | Trabajo Fin de Grado 2014 |
  Desarrollo de simulación sonora</title>
<link rel="stylesheet" type="text/css" href="styles/alertify.
  default.css" />
<link rel="stylesheet" type="text/css" href="styles/alertify.
  core.css" />
<script src="lib/jquery-1.7.2.min.js" type="text/javascript"></
  script>
<script src="lib/jquery.jqGauges.min.js" type="text/javascript
  "></script>
<script src="lib/alertify.js" type="text/javascript"></script>
<script language='JavaScript'>
$(document).ready(function(){
  $("#author").click(function(){
    alertify.alert('<h3>Autor de la aplicación</h3><hr/>Esta
      aplicación web es el fruto de un trabajo fin de grado
      realizado por el alumno Iván Bautista, siendo el tutor
      José Alberto Maroto Centeno.<p>Año 2014</p>');
  });
});
```



```

});
function app(){
    alertify.confirm("Estás apunto de conocer los misterios del
        ruido durante el despegue de un avión. ¿Estás seguro que
        deseas entrar?", function (e) {
        if (e) {
            lanzarApp();
        } else {
            alertify.error('Has cancelado el inicio de aplicación')
        }
        });
}
function lanzarApp(){
    var opciones="toolbar=no,location=no,directories=no,status=no
        ,menubar=no,scrollbars=no,resizable=NO,width=1024,height
        =650,top=10,left=140";
    window.open('app.html', '', opciones);
}
</script>
<style type="text/css">
    body{
        background-image: url(images/init.png);
        background-repeat: no-repeat;
        background-position: center center;
        background-attachment: fixed;
        background-size: cover;
    }
    .myButton {
        -moz-box-shadow:inset 0px 1px 0px 0px #54a3f7;
        -webkit-box-shadow:inset 0px 1px 0px 0px #54a3f7;
        box-shadow:inset 0px 1px 0px 0px #54a3f7;
        background:-webkit-gradient(linear, left top, left bottom,
            color-stop(0.05, #007dc1), color-stop(1, #0061a7));
        background:-moz-linear-gradient(top, #007dc1 5%, #0061a7
            100%);
        background:-webkit-linear-gradient(top, #007dc1 5%, #0061a7
            100%);
        background:-o-linear-gradient(top, #007dc1 5%, #0061a7
            100%);
        background:-ms-linear-gradient(top, #007dc1 5%, #0061a7
            100%);
    }

```

```

background:linear-gradient(to bottom, #007dc1 5%, #0061a7
    100%);
filter:progid:DXImageTransform.Microsoft.gradient(
    startColorstr='#007dc1', endColorstr='#0061a7',
    GradientType=0);
background-color:#007dc1;
-moz-border-radius:3px;
-webkit-border-radius:3px;
border-radius:3px;
border:1px solid #124d77;
display:inline-block;
cursor:pointer;
color:#ffffff;
font-family:arial;
font-size:13px;
padding:6px 24px;
text-decoration:none;
text-shadow:0px 1px 0px #154682;
}
.myButton:hover {
    background:-webkit-gradient(linear, left top, left bottom,
        color-stop(0.05, #0061a7), color-stop(1, #007dc1));
    background:-moz-linear-gradient(top, #0061a7 5%, #007dc1
        100%);
    background:-webkit-linear-gradient(top, #0061a7 5%, #007dc1
        100%);
    background:-o-linear-gradient(top, #0061a7 5%, #007dc1
        100%);
    background:-ms-linear-gradient(top, #0061a7 5%, #007dc1
        100%);
    background:linear-gradient(to bottom, #0061a7 5%, #007dc1
        100%);
    filter:progid:DXImageTransform.Microsoft.gradient(
        startColorstr='#0061a7', endColorstr='#007dc1',
        GradientType=0);
    background-color:#0061a7;
}
.myButton:active {
    position:relative;
    top:1px;
}

```

```
.auto-style1 {
    text-align: center;
    color:#FFFFFF;
}

.auto-style2 {
    text-decoration: underline;
}

.auto-style4 {
    font-size: x-large;
}

.footer{
    position: absolute;
    bottom: 0;
    text-align:center;
    -moz-box-shadow:inset 0px 0px 15px 3px #23395e;
    -webkit-box-shadow:inset 0px 0px 15px 3px #23395e;
    box-shadow:inset 0px 0px 15px 3px #23395e;
    background:-webkit-gradient(linear, left top, left bottom,
        color-stop(0.05, #2e466e), color-stop(1, #415989));
    background:-moz-linear-gradient(top, #2e466e 5%, #415989
        100%);
    background:-webkit-linear-gradient(top, #2e466e 5%, #415989
        100%);
    background:-o-linear-gradient(top, #2e466e 5%, #415989
        100%);
    background:-ms-linear-gradient(top, #2e466e 5%, #415989
        100%);
    background:linear-gradient(to bottom, #2e466e 5%, #415989
        100%);
    filter:progid:DXImageTransform.Microsoft.gradient(
        startColorstr='#2e466e', endColorstr='#415989',
        GradientType=0);
    background-color:#2e466e;
    -moz-border-radius:17px;
    -webkit-border-radius:17px;
    border-radius:17px;
    border:1px solid #1f2f47;
    display:inline-block;
```

```

        cursor:pointer;
        color:#ffffff;
        font-family:arial;
        font-size:15px;
        padding:0px 10px;
        text-decoration:none;
        text-shadow:0px 1px 0px #263666;
    }
    .footer:hover {
        background:-webkit-gradient(linear, left top, left bottom,
            color-stop(0.05, #415989), color-stop(1, #2e466e));
        background:-moz-linear-gradient(top, #415989 5%, #2e466e
            100%);
        background:-webkit-linear-gradient(top, #415989 5%, #2e466e
            100%);
        background:-o-linear-gradient(top, #415989 5%, #2e466e
            100%);
        background:-ms-linear-gradient(top, #415989 5%, #2e466e
            100%);
        background:linear-gradient(to bottom, #415989 5%, #2e466e
            100%);
        filter:progid:DXImageTransform.Microsoft.gradient(
            startColorstr='#415989', endColorstr='#2e466e',
            GradientType=0);
        background-color:#415989;
    }
</style>
<body style="position: static; margin-top: 20%">
<div class="auto-style1">
    <h2>Aplicación para la simulación del ruido emitido en el
    aeropuerto de Málaga</h2>
    <button class="myButton" onclick="app();">Lanzar aplicaci&
        oacute;n</button>
</div>
<div class="footer">
<p id="author" style="">Autor de la aplicación</p>
</div>
<script src="lib/locked.js" type="text/javascript"></script>
</body>
<html>

```

C.2. Código del fichero app.html

```
<!DOCTYPE html>
<html>
<head>
<meta content="es" http-equiv="Content-Language">
<meta content="Iván Bautista Moreno" http-equiv="author">
<meta content="text/html; charset=utf-8" http-equiv="Content-
  Type">
<link rel="shortcut icon" href="images/1398100892_130429.ico"
  />
<link rel="stylesheet" type="text/css" href="styles/jquery.
  jqGauges.css" />
<link rel="stylesheet" type="text/css" href="styles/alertify.
  default.css" />
<link rel="stylesheet" type="text/css" href="styles/alertify.
  core.css" />
<script src="lib/jquery-1.7.2.min.js" type="text/javascript"></
  script>
<script src="lib/jquery.jqGauges.min.js" type="text/javascript
  "></script>
<script src="lib/alertify.js" type="text/javascript"></script>
<!--[if IE]><script lang="javascript" type="text/javascript"
  src="lib/excanvas.js"></script><![endif]-->
<title>Aplicación de estudio del ruido sonoro al despegue de un
  avión | Trabajo Fin de Grado 2014 | Desarrollo de simulación
  sonoro</title>
<style type="text/css">
.auto-style1 {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  text-align: center;
  color: lightgray;
}
.auto-style3 {
  vertical-align: baseline;
  border-style: solid;
  border-width: 1px;
  margin: 1px;
  text-decoration: underline;
}
.auto-style4 {
```

```

    text-align: left;
}
.auto-style5 {
    text-decoration: underline;
}
.links{
    padding-left: 20px;
}
</style>
<meta content="Página para el trabajo fin de grado de estudio
    de ruido con finalidad temática para la asignatura de física
    ." name="description">
<script lang="javascript" type="text/javascript">
    /*=====
    * Variables globales
    *=====
    */
    var result, temp = 0, aux=1, j=0;
    var executing = false;
    var objDspl, objImg;
    var distance = 500;
    /*=====
    * Funciones inicio
    *=====
    */
    $(document).ready(function () {
        //Notificaciones del texto
        $('#uses').click(function(){
            alertify.alert("<h2>¿Cómo usar la aplicación?</
            h2><hr/><p style='text-align:justify'>Esta
            aplicación te permite realizar una simulación
            concreta del despegue de un avión y el ruido
            que produce. Para empezar a utilizarla,
            sigue los siguientes pasos: <ol style='text-
            align:justify'><li>Haz click en uno de los
            puntos marcados y espera a que termine la
            simulación.</li><br><li >Una vez ha terminado
            , en los marcadores digitales de la izquierda
            podrás ver el valor cuantitativo del nivel
            de presión sonora ejercida en ese punto y la
            distancia hasta el punto de despegue.</li><br

```

```

        ><li>El marcador analógico mostrará si este
        ruido será molesto para el oído humano,
        mediante una barra de color.</li></ol></p>");
    });
    $('#theory').click(function(){
        alertify.alert("<h2>Teoría del ruido</h2><hr/><
        p style='text-align:justify'>La ecuación de
        trabajo que relaciona la variación del nivel
        de presión sonora  $L_P$  con la
        distancia  $r$  a un foco sonoro que emite
        con un nivel de potencia sonora  $L_W$ 
        > viene dada por la siguiente ecuación: <br><
        p style='text-align:center'> $L_P = L_W - 20 \cdot \log_{10} r - 11 - \alpha$ ;
        </p><p style='text-align:left'> $L_P$ : Nivel de presión sonora.<br/>
 $L_W$ : Nivel potencia sonora.<br/>
 $r$ : Distancia entre puntos.<br/>
 $\alpha$ : Coeficiente de atenuación del aire. (0.0025dB/m)</p>");
    });
    //Nivel de ruido (dBA)
    $('#jqSegmentedDisplay').jqSegmentedDisplay({
        background: '#DCDCDC',
        border: {
            padding: 10,
            lineWidth: 4,
            strokeStyle: '#76786A'
        },
        digits: 9,
        segmentMode: 'fourteenSegment',
        text: '000.0 dbA',
        textForeground: '#333333',
        textForegroundUnlit: '#DCDCDC'
    });
    $('#jqDistanceDisplay').jqSegmentedDisplay({
        background: '#DCDCDC',
        border: {
            padding: 10,
            lineWidth: 4,
            strokeStyle: '#76786A'
        },

```

```

        digits: 10,
        segmentMode: 'fourteenSegment',
        text: 'P0: 0000.0M',
        textForeground: '#333333',
        textForegroundUnlit: '#DCDCDC'
    });
    var scaleBackgroundGradient = {
        type: 'linearGradient',
        x0: 0,
        y0: 1,
        x1: 0,
        y1: 0,
        colorStops: [{ offset: 0, color: '#A4D020' },
                     { offset: 0.993, color: '#BFFE01' }
                    ]
    };

    var gradient1 = {
        type: 'linearGradient',
        x0: 0,
        y0: 0.5,
        x1: 1,
        y1: 0.5,
        colorStops: [{ offset: 0, color: '#C5F80B' },
                     { offset: 1, color: '#6B8901' }
                    ]
    };

    var gradient2 = {
        type: 'linearGradient',
        x0: 0.5,
        y0: 0,
        x1: 0.5,
        y1: 1,
        colorStops: [{ offset: 0, color: '#FF3366' },
                     { offset: 1, color: '#B2183E' }
                    ]
    };

    var anchorGradient = {
        type: 'radialGradient',
        x0: 0.35,
        y0: 0.35,
        r0: 0.0,

```



```

        x1: 0.35,
        y1: 0.35,
        r1: 1,
        colorStops: [{ offset: 0, color: '#4F6169' },
                      { offset: 1, color: '#252E32' }]
    };

    $('#jqRadialGauge').jqRadialGauge({
        background: '#DCDCDC',
        border: {
            lineWidth: 6,
            strokeStyle: '#76786A',
            padding: 16
        },
        shadows: {
            enabled: true
        },
        anchor: {
            visible: true,
            fillStyle: anchorGradient,
            radius: 0.10
        },
        tooltips: {
            disabled: false,
            highlighting: true
        },
        animation: {
            duration: 1
        },
        annotations: [
            {
                text: 'Ruido(dBA)',
                font: '10px sans-serif',
                horizontalOffset: 0.5,
                verticalOffset: 0.80
            }
        ],
        scales: [
            {
                minimum: 0,
                maximum: 100,

```

```

    startAngle: 180,
    endAngle: 360,
    majorTickMarks: {
      length: 12,
      lineWidth: 2,
      interval: 10,
      offset: 0.84
    },
    minorTickMarks: {
      visible: true,
      length: 8,
      lineWidth: 2,
      interval: 2,
      offset: 0.84
    },
    labels: {
      orientation: 'horizontal',
      interval: 10,
      offset: 1.00
    },
    needles: [
      {
        value: 0,
        type: 'pointer',
        outerOffset: 0.8,
        mediumOffset: 0.7,
        width: 10,
        fillStyle: '#252E32',
      }
    ],
    ranges: [
      {
        outerOffset: 0.82,
        innerStartOffset:
          0.76,
        innerEndOffset:
          0.68,
        startValue: 40,
        endValue: 70,
        fillStyle:

```

```

                                gradient1
                                },
                                {
                                    outerOffset: 0.82,
                                    innerStartOffset:
                                        0.68,
                                    innerEndOffset:
                                        0.60,
                                    startValue: 70,
                                    endValue: 100,
                                    fillStyle:
                                        gradient2
                                }
                            ]
                        }
                    ]
                });

            });

/*=====
* Funciones auxiliares
*=====
*/
function f(p){
    if (executing==false){
        calcularLp(p);
        updateLights(0);
        statusOn();
        objImg=setInterval('img_canvas()',750);
        objDsp=setInterval('fakeDs()',1000);
    }else{
        alertify.error("Simulación en ejecución..");
    }
}

/*
* caluclarLp() - Es la función que calculará según el t el
  valor correspondiente al punto pulsado en la imagen.
*/
function calcularLp(p){
    var r=[714.5,857.4,1143.2,1429,1714.8,
        2000.6,2286.4,2572.2,2858,3143.8];

```

```

musicOn(r[p]);
result=Lp(r[p]);
    updateDistance(p,r[p]);
//Retrasamos el cálculo del valor hasta que no se ha
    parseado todo el ruido
window.setTimeout(function(){
    clearInterval(objImg);
    updateDisplay(result);
        updateLights(result);
    },31000);
window.setTimeout(function(){
    statusOff();
    temp=0;result=0;j=0;
    },41000);
}
/*
* fakeDs - Establece los valores aleatorios de dB(A).
*/
function fakeDs(){
    temp=temp+1;
    updateDisplay(dbRandom());
}
/*
* MusicOn - Obtiene el objeto audio y empieza a
    reproducirlo.
*/
function musicOn(d){
    $("#audio")[0].volume =714.5/d;
    $('#audio')[0].play();
}
/*
* StatusXXX() - Cambia el estado de la variable global
*/
function statusOn(){
    executing=true;
}
function statusOff(){
    executing=false;
}
/*
* updateDisplay() - Es la función para representación del

```

```

    marcador digital.
*/
function updateDisplay(result) {
    $('#jqSegmentedDisplay').jqSegmentedDisplay('option
        ', 'text', "0"+ result +" dbA");

    (temp<30)? console.log() :clearInterval(objDsp);
}
/*
*   updateDistance() - Mostrará en cada simulación la
    distancia "real" entre el punto rojo y la bandera
*/
function updateDistance(p,distance) {
    $('#jqDistanceDisplay').jqSegmentedDisplay('option
        ', 'text', "P"+(p+1)+" : "+ distance +"m");
}
/*
* updateLights() - Es la función para representar en el
    marcador analógico.
*/
function updateLights(result){
    var scales = $('#jqRadialGauge').jqLinearGauge('
        option', 'scales');
    scales[0].needles[0].value = result;
    $('#jqRadialGauge').jqLinearGauge('update');
}
/*
* Lp - Es la ecuación que relaciona la variación del
    nivel de presión sonora con la distancia a un foco
    sonoro.
* parámetros : r - Es la distancia entre el foco emisor
    y el punto donde medimos.
*/
function Lp(r){
    var result=(140-20*log10(r)-11-0.0025*r)+
        getRandomArbitrary(-0.5,0.5);
    var n=result.toFixed(1);
    return n;
}
/*
* log10 - Realiza la conversión del logaritmo neperiano Ln

```

```

        a logaritmo en base 10
    * parámetros : x - Número del cual queremos el logaritmo
    */
    function log10(x){
        return Math.log(x)/Math.log(10);
    }
    /*
    * getRandomArbitrary - Devuelve un número aleatorio entre
        un mínimo y un máximo establecido
    * parámetros : min - N° mínimo del limite
    *               max - N° máximo del limite
    */
    function getRandomArbitrary(min, max) {
        return Math.random() * (max - min) + min;
    }
    /*
    * dbRandom - Calcula los dB(A) aleatorios.
    */
    var lastDb;
    function dbRandom(){
        var n=temp+getRandomArbitrary(0,temp);
        var controlDb=n.toFixed(1);
        if (controlDb>result){
            return lastDb;
        }else{
            lastDb=controlDb;
            return controlDb;
        }
    }
    /*
    * img_canvas - La función que realiza un canvas manual y
        dinámico.
    */
    function img_canvas() {
        j=j+1;
        $("#airport").attr("src","images/canvas/"+j+".png");
    };
</script>
</head>

<body style="background:url(images/background.jpg);">

```

```

<div id="par" style="padding-left:5%">
  <div class="auto-style4" style="width: 310px;">
    <div id="jqRadialGauge" style="width: 260px; height: 80px
      ;"></div>
    <div id="jqSegmentedDisplay" style="width: 260px; height: 70
      px;"></div>
    <div id="jqDistanceDisplay" style="width: 260px; height: 70px
      ;"></div>
  </div>
<div id="map" style="height: 37px;padding-top:230px;">
  <p style="position: relative; left: 40%; top: -400px; width:
    572px; height: 553px">
    
    <map name="airportmap">
      <area href="javascript:f(0);" alt="Punto 1" shape="rect"
        coords="295,30,310,55" />
      <area href="javascript:f(1);" alt="Punto 2" shape="rect"
        coords="165,151,182,175" />
      <area href="javascript:f(2);" alt="Punto 3" shape="rect"
        coords="115,191,132,215" />
      <area href="javascript:f(3);" alt="Punto 4" shape="rect"
        coords="227,233,242,255" />
      <area href="javascript:f(4);" alt="Punto 5" shape="rect"
        coords="394,171,412,195" />
      <area href="javascript:f(5);" alt="Punto 6" shape="rect"
        coords="218,325,235,345" />
      <area href="javascript:f(6);" alt="Punto 7" shape="rect"
        coords="310,344,328,364" />
      <area href="javascript:f(7);" alt="Punto 8" shape="rect"
        coords="181,418,195,445" />
      <area href="javascript:f(8);" alt="Punto 9" shape="rect"
        coords="259,442,274,464" />
      <area href="javascript:f(9);" alt="Punto 10" shape="rect"
        coords="354,479,375,499" />
    </map>
  </p>
</div>
<audio id="audio" >
  <source src="av.mp3" type="audio/mpeg">
  Este navegador no es compatible.

```

```
Avisa a tu administrador de sistemas para instalar un
navegador de última generación.
</audio><br><br>
<div class="footer">
  <p id="uses" class="links"><a href="#" style="color: rgba
    (0, 50, 250, 0.74);text-decoration: none;"> Instrucciones
    de uso</a></p>
  <p id="theory" class="links"><a href="#" style="color:
    rgba(0, 50, 250, 0.74);text-decoration: none;"> Aspectos
    teóricos del ruido emitido</a></p>
</div>
<script src="lib/locked.js" type="text/javascript"></script>
</body>
</html>
```


II

Pliego de condiciones

En este capítulo se van a detallar las características hardware y software necesarias para el funcionamiento de este Trabajo Fin de Grado.

Hardware

Este trabajo fin de grado necesita un ordenador con las siguientes características hardware:

- Procesador Intel® H61 Express o superior.
- 2GB Memoria RAM DDR3 1600MHz.
- Intel® Celeron® G1610T con gráficos Intel HD (2,3 GHz, 2 MB de caché, 2 núcleos).
- Salida de audio para altavoces.

Software

Este trabajo fin de grado necesita los siguientes sistemas operativos y aplicaciones:

- Sistema operativo Windows 7 (o superior), en cualquiera de sus versiones.
- Navegador Internet Explorer 10 o superior (**recomendado**), aunque también es compatible con una versión superior del navegador Google Chrome Versión 23.0.1271.97 m o Mozilla Firefox v.27 o superior.
- Configuración de las opciones de Internet para habilitar el contenido activo en un CD o un archivo del disco duro. (*Véase manual de instalación*)
- Resolución de pantalla 1024x768 o superior.

III

Estudio económico

El presente documento informa sobre los costes económicos referidos a la elaboración de este trabajo fin de grado.

En los costes hardware y software, vamos asumir la compra de un equipo actualizado y con el software compatible que cumpla con las condiciones recogidas en el pliego de condiciones.

Concepto	Uds.	Coste/unidad	Coste
ORDENADOR HP CELERON G1610T 2GB/ 500GB/ DVD±RW/ FREEDOS	1	285,14 €	285,14 €
SUBTOTAL			285,14 €

(a) Costes del hardware

Concepto	Uds.	Coste /unidad	Coste
S.O Windows 7 Home Premium 32 bits	1	86,67 €	86,67 €
Service Pack 1 W7 x86	1	0 €	0 €
Update IE10 Windows 6.1 x86	1	0 €	0 €
SUBTOTAL			86,67 €

(b) Costes del software

Tabla C.1: Costes generales

Los costes de mano de obra asociados a este trabajo serán:

Concepto	Horas	Coste/hora	Coste
Diseño de aplicación web	90	15,00 €	1.350 €
Desarrollo aplicación web	270	36,00 €	9.720 €
Desarrollo de la documentación	120	0 €	0 €
SUBTOTAL			11.070 €

Tabla C.2: Costes de mano de obra

Por tanto, una vez desglosados todos los costes, los costes totales de este trabajo fin de grado son:

Concepto	Coste
Costes hardware	285,14 €
Costes software	86,67 €
Costes mano de obra	11.070 €
IVA 21 %	2.402,78 €
TOTAL	13.844,59 €

Tabla C.3: Costes totales