



PROYECTO
FIN DE GRADO



Este documento expone el trabajo realizado para la implementación de un sistema software capaz de cumplir las normativas acústicas exigidas a todos los locales de ocio que reproduzcan música en el establecimiento, de forma que se garantice el bienestar tanto de los clientes y trabajadores del local como de los residentes cercanos al mismo.

A lo largo de este documento se presentan las distintas fases de desarrollo de software: análisis de requisitos, planificación, diseño, implementación, prueba y evaluación del software, y por último, despliegue.

A estas fases hay que añadir una etapa preliminar de Ingeniería Inversa, ya que se dispone de un limitador de sonido funcional, el cual se estudiará a fondo y supondrá el punto de partida del proyecto y una guía a seguir en un campo completamente desconocido.



Alejandro Ruiz Becerra es el estudiante a cargo del diseño e implementación del proyecto, y con este trabajo finaliza el Grado en Ingeniería Informática con mención en Ingeniería del Software.



Andrés María Roldán Aranda es el profesor ingeniero a cargo del presente proyecto, así como el tutor del alumno. Actualmente es profesor del departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

Limitador de sonido
para locales de música

Alejandro Ruiz Becerra

INGENIERÍA
INFORMÁTICA

2020 - 2021

UNIVERSIDAD DE GRANADA

Grado en
Ingeniería Informática

**Limitador de sonido
para locales de música**

Alejandro Ruiz Becerra
2020-2021

Tutor: Andrés María Roldán Aranda



GRADO EN
INGENIERÍA INFORMÁTICA
TRABAJO DE FIN DE GRADO

Limitador de sonido
para locales de música

CURSO ACADÉMICO: 2020 - 2021

Alejandro Ruiz Becerra



GRADO EN INGENIERÍA INFORMÁTICA

Limitador de sonido para locales de música

REALIZADO POR:

Alejandro Ruiz Becerra

DIRIGIDO POR:

Andrés María Roldán Aranda

DEPARTAMENTO:

Electrónica y Tecnología de Computadores

D. Andrés María Roldán Aranda, Profesor del Departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada, como director del Trabajo Fin de Grado de D. Alejandro Ruiz Becerra,

Informa:

Que el presente trabajo, titulado:

Limitador de sonido para locales de música

ha sido realizado y redactado por el mencionado alumno bajo mi dirección, y con esta fecha autorizo a su presentación.

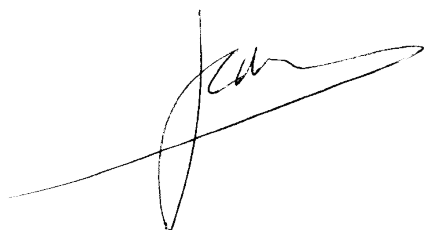
Granada, a 21 de julio de 2021

A handwritten signature in black ink, appearing to read 'Andrés Roldán', with a long, sweeping horizontal stroke extending to the right.

Fdo. Andrés María Roldán Aranda

Los abajo firmantes autorizan a que la presente copia de Trabajo Fin de Grado se ubique en la Biblioteca del Centro y/o departamento para ser libremente consultada por las personas que lo deseen.

Granada, a 21 de julio de 2021



Fdo. Alejandro Ruiz Becerra



Fdo. Andrés María Roldán Aranda

Limitador de sonido para locales de música

Alejandro Ruiz Becerra

PALABRAS CLAVE:

GranaSAT, Acústica y audio, Ingeniería Acústica, Ingeniería Inversa, Control de ruidos, Ecualización, Electrónica.

RESUMEN:

El objetivo del presente proyecto es diseñar e implementar el software necesario para la construcción de un limitador de sonido para locales de ocio, de forma que se cumplan las especificaciones legales y ordenanzas exigidas por las instituciones en éste ámbito, siendo beneficiara del presente trabajo la empresa **Heimdal Sound Control**.

El proyecto puede dividirse en tres grandes bloques: ingeniería inversa, diseño e implementación. Durante la primera fase se estudian y analizan limitadores de sonido de la competencia, ya presentes en el mercado; para luego diseñar el sistema en base a los requisitos extraídos del proceso de ingeniería inversa, y finalmente desarrollar y probar el software del producto.

Este Trabajo de Fin de Grado se sitúa en el ámbito de un proyecto mayor, ambicioso y de largo recorrido, y se apoya en el trabajo realizado por otros alumnos pertenecientes a diversas competencias. Por tanto, el presente trabajo no debe verse como un todo, sino como un gran engranaje dentro de una máquina mayor, el cuál permite que el conjunto de componentes interaccionen entre ellos.

La complejidad y el ámbito multidisciplinar de este Trabajo de Fin de Grado permite cubrir, no sólo algunas de las diferentes especialidades del Grado en Ingeniería Informática, sino también adquirir conocimientos y habilidades transversales o específicos de otros campos de la Ingeniería, como la **Electrónica** y la **Acústica**.

El resultado de todo lo expuesto culmina con un equipo real de limitación de sonido completo y funcional, que cumple con los requisitos definidos en las etapas iniciales del proyecto, y con el cual se cierra la etapa universitaria de Grado.

Sound limiter for music venues

Alejandro Ruiz Becerra

KEYWORDS:

GranaSAT, Acoustics and audio, Acoustical Engineering, Reverse Engineering, Noise control, Equalization, Electronics.

ABSTRACT:

The main objective of this project is to design and implement the required software to build a sound limiter for music venues, so that the legal specifications and ordinances demanded by the institutions are enforced, being benefited of the current work the **Heimdal Sound Control** company.

The project can be divided in three blocks: reverse engineering, design and implementation. During the first phase, a functional sound limiter out of the market will be analyzed and investigated. As a result, our system's design will be based on the functional and non-functional requirements out of the reverse engineering process. Finally, the system's software will be implemented and tested accordingly.

This Bachelor's Thesis is under the scope of a bigger, ambitious and long-run project; and is based on the work of other students from different competences. Therefore, the current work should not be seen as a whole, but as a big gear inside a bigger machine, which allows the set of components interact.

The complexity and multi-disciplinary scope of this Bachelor's Thesis does not only allow to cover some of the different competences from the Degree in Computer Engineering, but also allows to acquire knowledge and skills from other Engineering areas, as **Electronics** and **Acoustics**.

The defined engineering process produces a complete and functional real-time sound limiter, which complies with the system's requirements defined on the initial steps of the project, and with which the University stage of Bachelor's Degree concludes.

'First do it, then do it right, then do it better'

Addy Osmani - Google Developer

1969 — 2019
50th anniversary of the first human landing on the moon



Science-fiction yesterday, fact today, obsolete tomorrow
Otto O. Binder

Acknowledgments:

This work has been possible thanks to a very small amount of people, but to whom I owe a great acknowledgment; although the one I can show you in these lines will undoubtedly be insufficient, serve this brief space as such.

The first of them must be for my family, my parents, José and Amparo, and my brother Francisco Javier. Ever since I can remember, their unconditional support in every aspect has been of primary importance for every of my undertakings to come to fruition. The support provided during this project has been only another proof of this.

To Luis and Alberto, for the technical support, the laughs, endless coffees and warnings to stop doing what you are doing. To Pablo, for the countless hours of advice, for the vision where mine got clouded, for the important things in life, for the wine and the debate near a bar.

This Master's Thesis would not have been born without the persistence of my tutor, Andrés Roldán Aranda, in the face of my initial reticence. His ambition has got me closer to numerous projects throughout all these years, and this work unbeatably closes this unforgettable stage.

And of course, to Natalia, for being with me during the course of this long stage, and keeping supporting me at the worst moments, for the understanding and the patience, thank you very much.

Everyone of the mentioned is part of this Thesis. To all of them, and to those who could not see it, thank you.

Agradecimientos:

Este trabajo ha sido posible gracias a un número muy reducido de personas, pero a las cuales debo un gran agradecimiento; aunque el que pueda mostrarle en estas líneas será sin duda insuficiente, sirva este breve espacio como tal.

El primero de ellos ha de ir destinado a mi familia, mis padres, José y Amparo, y mi hermano Francisco Javier. Desde siempre, su apoyo incondicional en todos los aspectos ha resultado de importancia capital para que toda empresa acometida llegara a buen puerto. El apoyo mostrado durante el periodo de desarrollo de este proyecto no ha sido sino una muestra más del mismo.

A Luis y Alberto por el apoyo técnico, las risas, los cafés interminables y los avisos para dejar lo que estás haciendo. A Pablo, por las incontables horas de asesoramiento y visión allí donde la mía se nublaba, por las cosas importantes de la vida, por el vino y el debate cerca de una barra.

Este Trabajo Fin de Máster no habría nacido sin la persistencia de mi tutor, Andrés Roldán Aranda, frente a mi reticencia inicial. Su ambición me ha acercado a numerosos proyectos a lo largo de todos estos años, y este trabajo cierra de manera inmejorable esta inolvidable etapa.

Y por supuesto, a Natalia, por acompañarme durante el transcurso de esta larga etapa, y seguir aún sirviendo de apoyo en los peores momentos, por la comprensión y la paciencia, muchas gracias.

Cada uno de los mencionados son partícipes de este Trabajo. A todos ellos, y a los que no pudieron verlo, muchas gracias.

Tabla de contenidos

Autorización Lectura	v
Autorización Depósito Biblioteca	vi
Resumen	vii
Dedicatoria	x
Agradecimientos	xii
Tabla de contenidos	xv
Lista de figuras	xix
List of Videos	xxi
Lista de tablas	xxiii
1 Introducción	3
1.1 Motivación	4
1.2 Objetivos del proyecto	5
1.3 Contexto	6
1.4 Estructura del proyecto	7
2 Especificación de requisitos	11

2.1	Requisitos funcionales	11
2.2	Requisitos no funcionales	12
3	Ingeniería Inversa	15
3.1	Instalación del limitador	15
3.1.1	Resumen	19
3.2	Extracción de credenciales	20
3.2.1	Credenciales del limitador	20
3.2.2	Credenciales del sistema operativo	22
3.3	Especificaciones del sistema	24
3.4	Análisis del rendimiento	25
3.5	Interfaz web	27
3.5.1	Imágenes	28
3.5.2	Análisis técnico	28
3.5.3	Comunicación IU-Limitador mediante ficheros	31
3.5.4	Comunicación IU-Limitador mediante procesos	34
3.6	Estructura del proyecto	35
3.7	Esquema general	37
4	Análisis del sistema	39
5	Diseño del sistema	41
6	Implementación del sistema	43
7	Validación y test	45
8	Conclusiones y trabajo futuro	47
A	Presupuesto	49
A.1	Recursos físicos	49

A.2 Recursos humanos	49
A.3 Software	49
A.4 Presupuesto final	49
B Normativa acústica de Granada	51

Lista de figuras

1.1	Logotipo de GranaSAT	3
3.1	Esquema conceptual del montaje del LM7	16
3.2	Parte trasera. Conexiones del LM7	16
3.3	Configuración IP requerida en el PC adicional para conectarse al limitador	18
3.4	Interfaz web del LM7	18
3.5	Estructura de directorios de un sistema GNU/Linux [®] Fuente : [?]	20
3.6	Estructura de las claves en el fichero /etc/shadow	22
3.7	Contenido del fichero /etc/shadow del LIMITER MODULE 7 (LM7)	23
3.8	Imagen de una placa base ALIX-1B [?]	25
3.9	27
3.10	27

Índice de vídeos

Lista de tablas

3.1	Especificaciones hardware del LM7	25
3.2	Cientes SSH utilizados en el proyecto.	26
3.3	Especificaciones del servidor web del LM7	27

Capítulo 1

Introducción

El presente proyecto se desarrolla en el marco del Trabajo de Fin de Grado con el cual se finaliza el Grado Universitario en Ingeniería Informática, con especialidad en Ingeniería del Software, cursado por el alumno Alejandro Ruiz Becerra en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada. El objetivo principal del presente trabajo es demostrar los conocimientos y habilidades adquiridos por el alumno durante la realización de dicho Grado. Para ello, se ha realizado el presente proyecto, el cual propone el diseño e implementación de un software que permita controlar los niveles acústicos en locales con equipos de música.

Este Trabajo de Fin de Grado se realiza en colaboración con el proyecto académico [GranaSAT](#), un proyecto multidisciplinario, dirigido por el Profesor Andrés María Roldán Aranda, que reúne a personas de diferentes campos que quieren adquirir conocimientos relativos a la Ingeniería Aeroespacial y a la Ingeniería Electrónica.



Figura 1.1 – Logotipo de [GranaSAT](#)

El laboratorio de [GranaSAT](#) así como el equipamiento y materiales necesarios para la

1

realización de este proyecto se encuentran en la Avenida de Madrid, frente al Escuela Internacional de Posgrado de la Universidad de Granada y el Antiguo Hospital Clínico de Granada (España). Este proyecto ha sido desarrollado tanto presencialmente en el laboratorio como de forma remota, debido a la situación actual de pandemia causada por el virus COVID-19.

1.1 Motivación

Todos los establecimientos comerciales que dispongan de equipos de reproducción musical o audiovisual pueden, especialmente de noche, afectar no solo al descanso de los vecinos, sino también a la salud auditiva de los clientes y a los trabajadores del mismo. Es por ello que surge la necesidad de controlar la emisión de ruidos desde estos establecimientos tanto hacia a las viviendas o oficinas adyacentes, como a la calle, así como mantener unos niveles acústicos adecuados en el interior del local. Esta necesidad se hace requisito para este tipo de establecimientos debido a la Ley del Ruido del año 2003.

La Ley del Ruido (Ley 37/2003) del 17 de noviembre de 2003 dio lugar al comienzo de la lucha legislada contra la contaminación acústica, mediante la cual se dota de un esquema básico a nivel estatal para que, en los niveles autonómico y local puedan elaborarse promulgaciones de esta ley. Por tanto, queda en manos de las comunidades autónomas, y en última instancia, de los ayuntamientos, el definir las medidas, normas e infracciones a aplicar en esta materia.

El Ayuntamiento de Granada aplica su propia normativa en este ámbito mediante la *Ordenanza Municipal de Protección del Medio Ambiente Acústico en Granada* del año 2007. Es de especial interés, dado el proyecto que nos ocupa, los artículos 36 y 37.

1.1.1 Artículo 36. Condiciones acústicas particulares en actividades y edificaciones donde se generan niveles elevados de ruido

En este artículo se definen una serie de condiciones para los establecimientos de espectáculos públicos, actividades recreativas y comerciales que generen elevados niveles de ruido.

A modo de resumen, este artículo:

- Clasifica las edificaciones donde se generan un nivel de ruido superior a 70 **dBA** en 3 tipos.
- Exige la aplicación de un aislamiento acústico al local.
- Prohíbe niveles de de presión sonora superiores a 90 **dBA** en zonas destinadas al

público, salvo que se dé advierta correctamente en los accesos a dichas zonas.

Si bien, estos tipos de establecimientos tienen la obligación de aislar acústicamente el local mediante la aplicación de diferentes soluciones arquitectónicas, esto no es suficiente para garantizar que no se sobrepasan los límites de emisión e inmisión de ruido definidos en la normativa, la cual se puede consultar en el Anexo B. Esto mismo queda reflejado en el artículo 37 de la normativa que puede verse a continuación.

1.1.2 Artículo 37. Instalación de equipos limitadores controladores acústicos

En aquellos locales descritos en el artículo 36 de la presente Ordenanza, donde se disponga de equipo de reproducción musical o audiovisual en los que los niveles de emisión sonora pudieran de alguna forma ser manipulados directa o indirectamente, se instalará un equipo limitador-controlador que permita asegurar, de forma permanente, que bajo ninguna circunstancia las emisiones del equipo musical superen los límites admisibles de nivel sonoro tanto en el interior del propio local (artículo 20 bis) como en las edificaciones adyacentes, así como que cumplen los niveles de emisión al exterior exigidos en esta Ordenanza.

Por tanto, resulta evidente observar que existe una necesidad por parte de los establecimientos con equipos de reproducción de música de controlar sus niveles de contaminación acústica de forma que se cumpla la normativa, para evitar molestias a vecinos, y denuncias y sanciones a los propietarios del local.

1.2 Objetivos del proyecto

El objetivo primario de este proyecto es el siguiente:

- Presentar un software capaz de registrar los niveles acústicos de emisión y recepción, y que actúe sobre dichos niveles en base a la normativa aplicada.

Este objetivo general define de forma concreta qué es lo que se pretende hacer en el proyecto. Asimismo, existen otros objetivos más específicos:

1. Conocer y analizar las necesidades reales del sistema de monitorización de ruido ambiental requerido.
2. Extraer los requisitos funcionales y no funcionales del sistema, a partir de conversaciones con el cliente y del proceso de ingeniería inversa.

1

3. Realizar una análisis de las tecnologías utilizadas en el software presente en el limitador del laboratorio, y estudiar si pueden solucionar las necesidades de cada uno de los requisitos.
4. Diseñar e implementar un prototipo del producto software requerido.
5. Realizar los tests de validación y pruebas necesarios para verificar el correcto funcionamiento del prototipo.
6. Acercar al alumno a un entorno real de Ingeniería.
7. Poner de manifiesto los conocimientos y habilidades adquiridas en el Grado de Ingeniería Informática.
8. Superar la signatura de Fin de Grado, y por ende, el Grado en Ingeniería Informática.

1.3 Contexto

Hasta ahora, se han establecido cuales son las necesidades y las motivaciones del proyecto, así como la idea general y los objetivos del mismo, pero nada se sabe aún de su punto de partida ni de su contexto, ¿se va a partir de cero? ¿existe una arquitectura hardware específica para este propósito o es parte del proyecto definirla y construirla? ¿hay más personas involucradas en el proyecto? En las siguientes páginas resolveremos estas cuestiones y más, y se proporcionará un enfoque más detallado y completo del la envergadura del trabajo que en este documento se presenta.

Para comenzar, en el laboratorio de [GranaSAT](#) se dispone de un limitador de sonido operativo, pero bastante antiguo, el cual es necesario estudiar para poder entender se funcionamiento y, por ende, conocer cuales son las tareas concretas que debe realizar un limitador de sonido para efectuar sus funciones. Este limitador será objeto que un minucioso proceso de ingeniería inversa, del cual extraeremos una gran cantidad de conocimiento, en forma de requisitos, diagramas y estrategias de acción para resolver las problemáticas a las que nos enfrentamos en el proyecto. Se dispone así mismo del todo el código fuente de este limitador, lo cual facilitará bastante nuestro trabajo.

Se dedicará un capítulo completo a este limitador, dónde se detallará el proceso de ingeniería inversa al que se le ha sometido, se explicará cómo funciona este limitador y se expondrán todos los conocimientos que se han podido extraer del él. Este limitador, por tanto, supondrá el punto de partida y la principal guía del presente proyecto, siendo así su piedra angular, ya que se volverá é constantemente para comparar y validar el trabajo realizado.

;; INSERTAR IMAGEN DEL LIMITADOR !!

También en laboratorio de [GranaSAT](#) se dispone de un limitador en fase de prototipo, sobre el cual correrá el sistema de limitación y control de sonido que se va a desarrollar. Esta arquitectura supone la inclusión de ciertas restricciones y consideraciones a la hora de diseñar e implementar el sistema, ya que se dispone de un hardware especializado y cuyo centro de cómputo es una Raspberry Pi.

;;INSERTAR IMAGEN DEL LIMITADOR NUEVO !!

Luis Peinado Córdoba, es un estudiante del Máster en Ingeniería de Telecomunicaciones y el responsable de esta arquitectura. El trabajo realizado en este prototipo supone su Trabajo de Fin de Máster.

;;INSERTAR IMAGEN DE LUIS !!

Por otro lado, es necesario poder configurar el sistema cuando se realice la instalación del mismo en el establecimiento. Los datos configurables en el equipo supondrán unos requisitos de datos y especificarán en el capítulo 2, pero a modo de resumen, es necesario configurar una serie de variables que modificarán el funcionamiento del sistema, como el aislamiento aplicado al local y la normativa del ayuntamiento o comunidad autónoma en el que se encuentra el local.

;;INSERTAR IMAGEN DEL APP CONF !!

Dani Ruiz Medina, estudiante del Grado en Ingeniería Informática, como proyecto para su Trabajo de Fin de Grado va a diseñar e implementar una aplicación para configurar nuestro sistema.

;;INSERTAR IMAGEN DE DANI !!

Nuestro sistema, por tanto, deberá comunicarse tanto con el hardware especializado como con la aplicación de configuración. Para ello deberán identificarse y definirse unas interfaces de comunicación para que los sistemas cooperen, lo cual se traduce en nuevos requisitos en nuestro proyecto. Se profundizará en estas interfaces en el capítulo de diseño, 5.

;; EN GENERAL, EXTENDER !!

1.4 Estructura del proyecto

El proyecto se divide en 8 capítulos y varios anexos que describen cada una de las partes del proceso de desarrollo del producto propuesto.

Los capítulos que componen el presente documento son:

- El presente capítulo, numerado como 1, pretende ser una introducción al proyecto,

1

tanto en su faceta de Trabajo de Fin de Grado como en su faceta de colaboración con GranaSAT. En este capítulo también puede encontrarse la planificación temporal del proyecto en forma de diagrama de Gantt.

- El capítulo 2 es un breve resumen de la idea global del producto, extraída de conversaciones con el cliente. A partir de esta información se generan una definición de necesidades principales y secundarias para el producto, en forma de requisitos del cliente.
- El capítulo 3 se procede a documentar el proceso de ingeniería inversa al que se ha sometido a dos versiones de un mismo limitador disponible en el laboratorio de GranaSAT. La idea principal de este proceso de ingeniería inversa es realizar un reconocimiento de las tareas técnica necesarias para completar los objetivos del proyecto.
- A continuación, en el capítulo 4, se utilizará el conocimiento obtenido mediante proceso del ingeniería inversa realizado en el capítulo anterior para presentar un primer análisis en profundidad del trabajo necesario a realizar para cumplir los objetivos del proyecto, así como un desglose de dicho trabajo en forma de requisitos del sistema. En este capítulo también se detallarán los requerimientos técnicos del producto. Una vez concretados estos requerimientos se analizarán los pros y los contras de diferentes tecnologías o enfoques que solucionan la problemática de cada uno de los subsistemas que componen el proyecto. De este análisis en profundidad se extrae una estrategia de trabajo, compuesta por un conjunto de soluciones técnicas viables para el producto y una planificación temporal para el desarrollo del proyecto que se puede observar en el diagrama de la figura figura X.Y.
- Tras realizar el análisis y siguiendo la metodología propuesta en la figura X.Y, se presenta en el capítulo 5 el diseño del producto, es decir, qué se va a hacer exactamente y cómo. Para ello se han tenido en cuenta las cuestiones relativas al análisis del capítulo anterior y se han seleccionado las soluciones y tecnologías óptimas para diseñar el sistema.
- En el capítulo 6 se presentan los detalles sobre la implementación del sistema y la metodología seguida durante este proceso. Los procesos de ingeniería inversa, análisis y diseño tendrán un gran valor en este punto, y serían cruciales durante el proceso de implementación.
- En el capítulo 7 se describen una serie de pruebas de validación y test de forma que se verifique que el producto generado a partir del diseño propuesto en el capítulo 5 cumple con las especificación de requisitos descrita en el 2, y que a demás, lo realiza de forma correcta.

- Por último, el capítulo 8 contiene un conjunto de conclusiones a modo de resumen, donde se repasan los objetivos del proyecto y cómo se han abordado cada uno de ellos. También se incluye una lista de mejoras del sistema como trabajo futuro y continuación del proyecto.
- En el anexo A se detalla el presupuesto y los costes asociados a este proyecto.

1

Capítulo 2

Especificación de requisitos

Una vez introducido el tema de estudio y los objetivos del presente proyecto, se va a dar paso en este segundo capítulo a definir cuáles son los requisitos del sistema, clasificándolos en requisitos funcionales y no funcionales.

Estos requisitos tienen su origen en las conversaciones que se han establecido con el cliente en primera instancia y como resultado del conocimiento extraído del proceso de ingeniería inversa.

2.1 Requisitos funcionales

Un controlador-limitador de sonido tiene como objetivo medir los valores de presión acústica que se emiten en el local, y evitar que se sobrepasen los límites establecidos en la normativa del núcleo de población en el que se encuentra el establecimiento.

- El sistema debe ser capaz de calibrar sus sensores.
- El sistema debe poder emitir ruido rosa.
- El sistema debe comprobar que no ha sido manipulado. Para ello se verificará en el inicio y/o fin de cada sesión que las calibraciones de sus sensores y señales de audio corresponden a los valores actuales de emisión y recepción.
- El sistema debe poder conservar un registro de sus lecturas por un período al menos 2 meses, con una periodicidad de un minuto.
- El sistema debe proporcionar un mecanismo de comunicación hacia el exterior, de forma que se pueda configurar y obtener métricas.
- El sistema debe poder leer los valores de emisión en el local.

- El sistema debe actuar en caso de que los valores de emisión en el local sobrepasen los valores límite definidos en la normativa vigente, limitando el nivel de presión acústica.

2.2 Requisitos no funcionales

Como requisitos no funcionales tenemos:

- El sistema debe poder ejecutarse en el hardware provisto (módulo de computación Raspberry Pi con sistema operativo DietPi y arquitectura [ARM](#)).
- El sistema debe ser robusto y tolerante a fallos.
- El producto debe ser lo más ligero posible en términos de consumo de recursos de computación y espacio en disco, ya que estos recursos son especialmente limitados en la arquitectura objetivo.
- El sistema debe usar [JSON](#) como formato estándar de intercambio de datos.
- Los valores de presión acústica deben darse en decibelios ponderados (dbA).

o

2

2

Capítulo 3

Ingeniería Inversa

En este capítulo se procede a documentar el proceso de ingeniería inversa al que se ha sometido a limitador que puede verse en la imagen imagen.

Tal y como se comentó en la sección 1.3, el punto de partida del proyecto es el estudio y análisis de un limitador funcional y operativo que se encuentra en el laboratorio de GranaSAT. Es importante recordar que se dispone del código fuente de este limitador, así como su manual de usuario, el cual nos será de gran ayuda para poder instalarlo correctamente y conocer las capacidades y características generales del producto. De aquí en adelante, se hará referencia al este limitador como LM7, ya que ese es el nombre técnico del producto.

!! IMAGEN DE LIMITADOR

3.1 Instalación del limitador

El primer paso a realizar para comenzar el proceso de ingeniería inversa es instalar el equipo. Para ello, se recurre al manual de usuario del equipo, y se siguen las instrucciones de instalación. En la figura 3.1 se puede observar el esquema de montaje del limitador en un entorno objetivo, es decir, en un establecimiento. En nuestro caso, nuestra entrada no se corresponde a una mesa de mezclas, sino a un ordenador, mediante el cual podremos enviar audio al limitador y comprobar su respuesta.

[Como empezar]

Montaje del limitador
Esquema de montaje

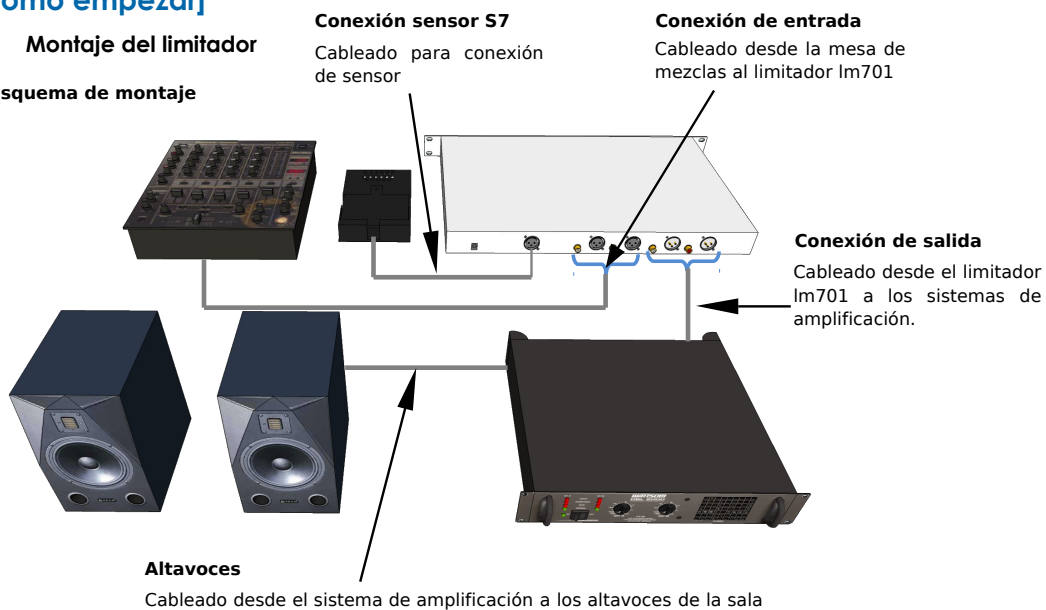


Figura 3.1 – Esquema conceptual del montaje del LM7

En la figura superior, podemos observar de izquierda a derecha y de arriba a bajo los siguientes elementos: mesa de mezclas (entrada), micrófono del limitador, limitador de sonido LM7, altavoces, amplificador (salida).

[Descripción e imágenes]

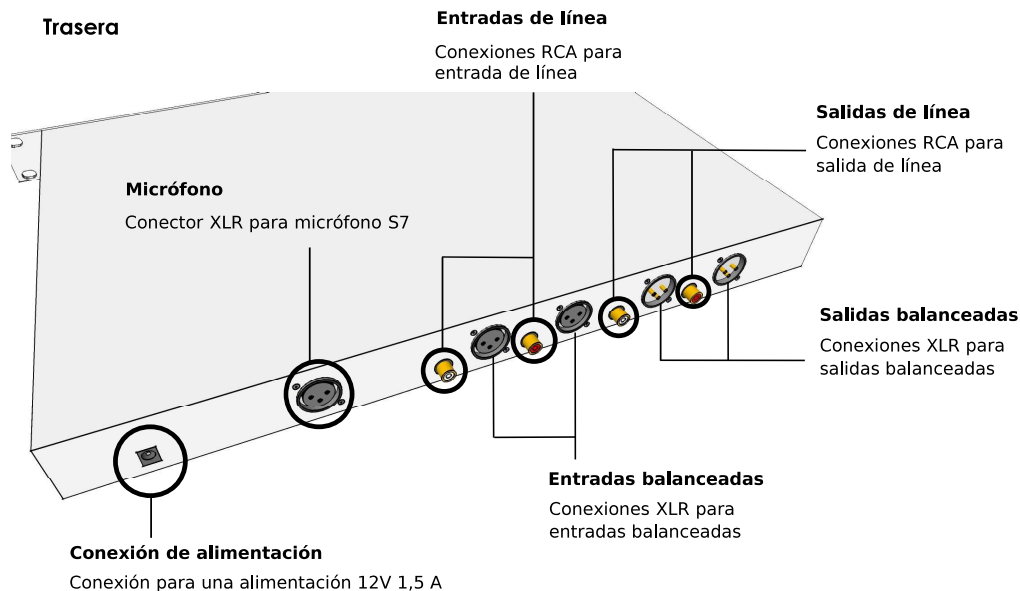


Figura 3.2 – Parte trasera. Conexiones del LM7

El sensor S7 (micrófono) es un componente esencial del limitador y forma parte del mismo. Gracias a él, el limitador puede medir la presión acústica en cada momento y actuar en consecuencia. Asimismo, y como se verá más adelante en este documento, será un elemento esencial en la calibración del limitador. El resto del elementos que se muestran en la figura 3.1 son elementos externos al limitador, y no alteran el funcionamiento del sistema en ninguna forma.

En la figura 3.2 se pueden observar las principales conexiones del LM7, las cuales constan de:

- Toma de alimentación eléctrica.
- Sensor S7 (micrófono) con conector XLR.
- Entradas balanceadas para audio con conectores XLR.
- Salidas balanceadas para audio con conectores XLR.
- Entradas y salidas no balanceadas con conectores RCA.
- Conector RJ-45 para conexión directa en área local (parte frontal del limitador, no visible en las figuras).

Conectamos el micrófono al limitador, así como la entrada y la salida de audio. La salida de audio se conecta a un amplificador de sonido disponible en el laboratorio, y como salida a este sistema de amplificación se conecta un dodecaedro. De forma que podamos tener interacción con el equipo, abrimos el limitador retirando la carcasa metálica exterior, para así poder conectar un teclado y una pantalla a la placa base del equipo. De esta forma descubrimos por primera vez las entrañas del limitador, y nos encontramos con una tecnología bastante desfasada y un ensamble prácticamente casero. Tal y como podemos ver en la imagen ??, el Hardware del limitador se compone de una placa base de tipo industrial, cuyas características de detallan en la tabla 3.1, un circuito integrado para las entradas y salidas de audio del limitador vistas en la figura 3.2 y una pequeña caja negra, la cual contiene un relé, y cuya funcionalidad se verá más adelante. Además de esto podemos observar la existencia de una tarjeta de sonido USB conectada a un puerto USB de la placa base y cuyas salidas van hacia la caja negra que podemos ver en la imagen. Como dispositivo de almacenamiento interno tenemos un CompactFlash, con 1GB de capacidad. En fases posteriores del actual proceso de ingeniería inversa extraeremos esta tarjeta de almacenamiento para acceder a sus archivos, ya que será necesario descubrir o modificar las credenciales de acceso tanto al equipo como al sistema de limitación, pues todas ellas son, por ahora, desconocidas.

!! IMAGEN DEL LIMITADOR ABIERTO

Tras revisar todas las conexiones, conectamos el equipo a la corriente eléctrica, y podemos por primera vez ver el equipo en funcionamiento. Mientras arranca el sistema, vemos en la pantalla que el sistema operativo instalado es Debian, una distribución de [GNU/Linux](#)® que se caracteriza por ser minimal. Esta versión en concreto no contiene interfaz gráfica.

Una vez completado el arranque, la pantalla se llena de caracteres que desaparecen rápidamente dando lugar a otros nuevos. De entre lo que es posible leer, se deduce que estos caracteres son datos relativos al limitador (su estado en cada momento, conteniendo lecturas de micrófono y líneas, atenuación aplicada, etc) y que los procesos del limitador vuelcan sus salidas por pantalla a la salida estándar, saturando la terminal principal y dejándola inutilizable. Por tanto, se accede a otra terminal (TTY) mediante la cual podamos trabajar. Esta nueva terminal nos pide usuario y contraseña, las cuales desconocemos.

Continuando con el manual de usuario se descubre que el equipo tiene un servidor web en el cual hay desplegada una interfaz web del limitador, mediante la cual podemos ver su estado, modificar sus configuraciones y obtener informes. Por ello, el próximo y último paso para completar la instalación del limitador es conectarlo a una red interna vía Ethernet, así como conectar y configurar un ordenador adicional mediante el cual podamos acceder, no solo a dicha interfaz web, sino al equipo en sí mediante [SSH](#).

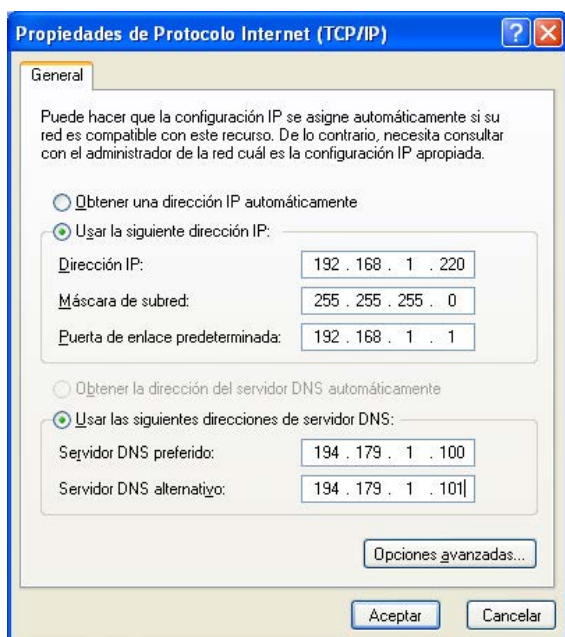


Figura 3.3 – Configuración IP requerida en el PC adicional para conectarse al limitador

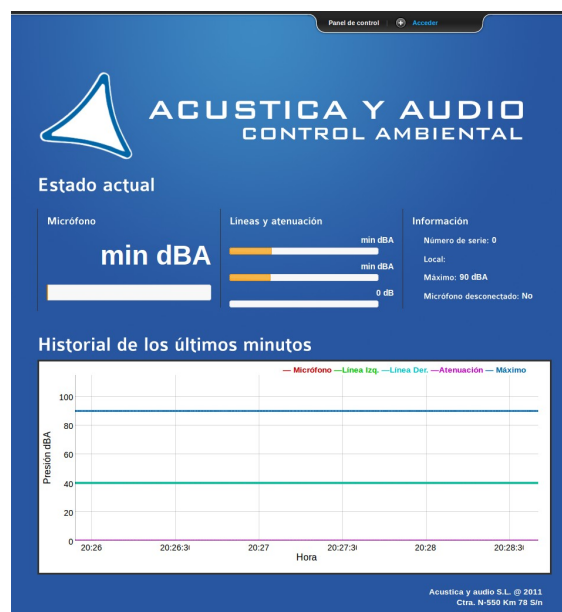


Figura 3.4 – Interfaz web del LM7

Una vez aplicada la configuración IP que se muestra en la imagen 3.3 en el equipo

auxiliar, probamos a acceder a la interfaz web de limitador que se encuentra desplegada en la dirección IP <http://192.168.1.223> y vemos por primera vez la aplicación web vista en la imagen 3.4. Al entrar en la interfaz web del limitador puede verse la ventana de estado, que informa sobre el estado actual del limitador. Esta interfaz arroja información básica, como:

- La presión actual en [dBA](#) tanto de las líneas como del sensor.
- Atenuación aplicada por el limitador en ese instante.
- El número de serie del limitador.
- El local de instalación
- Si el sensor (micrófono) se encuentra conectado o no.
- Un gráfico de los últimos 5 minutos de actuación del limitador.

En la zona superior derecha, se puede acceder al panel de control y a la sección de obtención de informes. Mediante este panel de control podremos también acceder y modificar la configuración del limitador, aunque para ellos será necesario proveer una clave de acceso válida. En el manual de usuario, se indica que existe un usuario *consultor*, cuya contraseña es a su vez *consultor*. Se trata de un usuario sin privilegios mediante el cual podremos consultar la configuración actual del limitador. Podemos a su vez cerrar sesión mediante el botón “Cerrar sesión”. Para poder acceder más allá necesitamos averiguar las claves de acceso al limitador.

Para acabar con el proceso de instalación, se comprueba si es posible conectarse mediante [SSH](#) al limitador. Tal y como se esperaba hay conectividad entre los equipos pero necesitamos proporcionar un usuario y una contraseña para acceder al sistema operativo.

Llegados a este punto, el limitador se encuentra instalado y funcionando, pero nuestro control sobre el mismo es completamente nulo ya que no disponemos de las credenciales necesarias para acceder al sistema ni al limitador. Por tanto, procederemos a extraer su dispositivo de almacenamiento, una [CompactFlash](#), para investigar desde otro ordenador su contenido y poder encontrar dichas credenciales, dando lugar así al proceso que da nombre a este capítulo, comenzaremos con el proceso de Ingeniería Inversa.

3.1.1 Resumen

Como resumen a esta sección, se han realizado las siguientes acciones:

- Como consecuencia de dichas acciones, tenemos los dos equipos conectados en red, por lo que se puede acceder al limitador desde el equipo auxiliar mediante [SSH](#), aunque desconoce el usuario y contraseña del sistema.

3.2 Extracción de credenciales

```

graph TD
    Root["/ \"root\""]
    Root --- Bin["/bin  
\"essential user command binaries\""]
    Root --- Etc["/etc  
\"configuration files for the system\""]
    Root --- Sbin["/sbin  
\"essential system binaries\""]
    Root --- Usr["/usr  
\"read-only user application support data & binaries\""]
    Root --- Var["/var  
\"variable data files\""]
    Root --- Dev["/dev  
\"device files incl. /dev/null\""]

    Bin --- BinList["bash  
cat  
chmod  
cp  
date  
echo  
grep  
gunzip  
gzip  
hostname  
kill  
less  
ln  
ls  
mkdir  
more  
mount  
mv  
nano  
open  
ping  
ps  
pwd  
rm  
sh  
su  
tar  
touch"]
    Etc --- EtcList["crontab  
cups  
fonts  
fstab  
host.conf  
hostname  
hosts  
hosts.allow  
hosts.deny  
init  
init.d  
issue  
machine-id  
mtab  
mttools.conf  
nanorc  
networks  
passwd  
profile  
protocols  
resolv.conf  
rpc  
securetty  
services  
shells  
timezone"]
    Sbin --- SbinList["fdisk  
fsck  
getty  
halt  
ifconfig  
init  
mkfs  
mkswap  
reboot  
route"]
    Usr --- UsrList["/usr/bin  
\"most user commands\"  
  
/usr/include  
\"standard include files for 'C' code\"  
  
/usr/lib  
\"obj, bin, lib files for coding & packages\"  
  
/usr/local  
\"local software\"  
/usr/local/bin  
/usr/local/lib  
/usr/local/man  
/usr/local/sbin  
/usr/local/share  
  
/usr/share  
\"static data sharable across all architectures\"  
/usr/share/man  
\"manual pages\""]
    Var --- VarList["/var/cache  
\"application cache data\"  
  
/var/lib  
\"data modified as programmes run\"  
  
/var/lock  
\"lock files to track resources in use\"  
  
/var/log  
\"log files\"  
  
/var/opt  
\"variable data for installed packages\"  
  
/var/spool  
\"tasks waiting to be processed\"  
/var/spool/cron  
/var/spool/cups  
/var/spool/mail  
  
/var/tmp  
\"temporary files saved between reboots\""]
    Dev --- DevList["/home  
\"user home directories\"  
  
/lib  
\"libraries & kernel modules\"  
  
/mnt  
\"mount files for temporary filesystems\"  
  
/opt  
\"optional software applications\"  
  
/proc  
\"process & kernel information files\"  
  
/root  
\"home dir. for the root user\""]
  
```

3.2.1 Credenciales del limitador

Alejandro Ruiz Becerra

operativo que corre sobre la máquina, sino las de los usuarios que tienen acceso a la configuración del limitador, es decir, son los usuarios y contraseñas necesarios para acceder a la aplicación web del limitador (imagen 3.4), así como los permisos de estos usuarios sobre la configuración del limitador.

El directorio **slr/** será de gran importancia en el ámbito del proyecto, ya que en este directorio se almacenarán la gran mayoría de ficheros relacionados con el limitador: datos de configuración, información de usuarios, ficheros de sonido, e incluso ficheros que funcionarán a modo de variables globales del limitador. Conforme se vaya avanzado en el documento se irá describiendo y explicando cada uno de estos ficheros.

El nombre del directorio se deduce que es un acrónimo de *SoundLimiterRecords*.

Para nuestra sorpresa, se descubre que los datos no sólo están almacenados en un fichero de texto plano, sino que tampoco se encuentran cifrados. Cada una de las líneas del fichero define un usuario con los siguientes campos:

- DNI.
- Nombre.
- Contraseña.
- Gestor de usuarios (si puede crear, modificar o eliminar usuarios).
- Gestor de configuración (si puede modificar la configuración del limitador).
- Fecha del alta del usuario en el sistema.

En el listado 3.1 pueden verse claros ejemplos del patrón definido justo sobre estas líneas. Adicionalmente, existen en el fichero otras líneas que no siguen este patrón y definen otros patrones nuevos. Ejemplos de estas líneas son las que encontramos en la línea 4 y 14 del listado 3.1. Estas líneas definen el modo de acceso al limitador y la eliminación de un usuario en el sistema, respectivamente, así como la marca de tiempo de la acción que dio origen a la inserción de esas líneas en el fichero.

Para confirmar el descubrimiento de las credenciales del limitador se comprueban algunos de los usuarios y claves encontradas en la interfaz web del limitador, con resultado satisfactorio. Se consigue por tanto el acceso a la configuración del limitador y tenemos ahora el control sobre él, aunque seguimos sin disponer de control sobre el sistema operativo sobre el que corre.

```

1 dni=*lm7-passwordUser&name=Password user&password=\t****&userManager=1&configManager=1&time
  =2012/05/11-11:48:36
2 dni=*lm7-remoteUser&name=Remote system user&password=\t****&userManager=0&configManager=0&time
  =2012/05/11-11:48:36
3 dni=consultor&name=Consultor&password=consultor&userManager=0&configManager=0&time=2012/05/11-11:48:36
4 key=authMethod&value=onlyPassword&time=2012/08/30-05:35:16
5 dni=E19578186&name=NOISEOFF&password=COCHA012015&userManager=1&configManager=1&time=2015/06/25-23:07:38
6 dni=E19578186&name=NOISEOFF&password=COCHA012015&userManager=1&configManager=1&time=2015/06/25-23:08:13
7 dni=E19578186&name=NOISEOFF&password=01COCHA2015&userManager=1&configManager=1&time=2015/06/26-14:55:51
8 key=authMethod&value=onlyPassword&time=2015/06/26-14:57:58
9 dni=E19578186&name=NOISEOFF&password=COCHA0130115&userManager=1&configManager=1&time=2016/09/20-19:14:48
10 dni=E19578186&name=NOISEOFF&password=COCHA0130115&userManager=1&configManager=1&time=2016/09/20-19:16:21
11 key=authMethod&value=onlyPassword&time=2016/09/20-19:16:50
12 dni=E19578186&name=NOISEOFF&password=cochao130115&userManager=1&configManager=1&time=2017/01/02-12:08:33
13 dni=44289989Q&name=NOISEOFF&password=cochao130115&userManager=1&configManager=1&time=2017/01/02-12:10:05
14 dni=E19578186&deleted=1
15 key=authMethod&value=onlyPassword&time=2017/01/02-12:12:01
16 dni=44289989Q&name=NOISEOFF&password=cochao130115&userManager=1&configManager=1&time=2017/01/05-09:50:40
17 key=authMethod&value=onlyPassword&time=2017/01/05-09:52:03

```

Listado 3.1 – Contenido del fichero *users.auth*

3

3.2.2 Credenciales del sistema operativo

Para obtener el control sobre el sistema operativo es necesario disponer de un usuario y contraseña válidos de forma que podamos acceder a él mientras el equipo está en funcionamiento.

Del sistema operativo sabemos que es [Debian](#), una distribución [GNU/Linux](#)®, por tanto, se investiga dónde almacena este sistema sus usuarios. Son datos estáticos así que deben encontrarse almacenados en alguna parte dentro del sistema de archivos contenidos en el disco, el cual, recordamos, tenemos conectado a otro ordenador mediante un adaptador [USB](#).

A través de la consulta de foros ([cyberciti.biz](#)) y manuales ([Debian.org](#)) en línea, descubrimos que los ficheros que necesitamos son */etc/passwd* y */etc/shadow*. Aunque ambos contienen información crítica sobre los usuarios y sus permisos, existen pequeñas diferencias. En resumen, mientras que */etc/passwd* almacena información mayormente relativa al usuario, */etc/shadow* contiene las claves de usuario (encriptadas) e información relacionada a ellas, no al usuario.

Las contraseñas encriptadas y otra información relacionada, como la información de caducidad de la contraseña, se almacenan en el fichero */etc/shadow*. Este fichero contiene una entrada por línea por cada uno de los usuarios listados en el fichero */etc/passwd*. Cada una de estas entradas están compuestas por una serie de campos separados por dos puntos (:), y generalmente, sigue este patrón.

```

vivek:$1$fnfffc$pGteyHdicpGOffX4ow#5:13064:0:99999:7::
  ↓           ↓           ↓   ↓   ↓   ↓
  1           2           3   4   5   6

```

Figura 3.6 – Estructura de las claves en el fichero *letcl/shadow*

1. Nombre de usuario.
2. Contraseña encriptada (**hash**). Sigue el formato **\$id\$salt\$hash**. El campo **\$id** representa el algoritmo usado:
 - (a) **\$1\$** es MD5.
 - (b) **\$2a\$** es Blowfish.
 - (c) **\$2y\$** es Blowfish.
 - (d) **\$5\$** es SHA-256.
 - (e) **\$6\$** es SHA-512.
3. Días desde que se cambió la contraseña, contando desde el 1 de enero de 1970.
4. Días tras los cuales la contraseña debe ser cambiada.
5. Días de antelación a la expiración de la contraseña en los que se avisa al usuario.
6. Días tras los cuales se desactiva una cuenta cuya contraseña está caducada.

Una contraseña **hash** no es más que una cadena de caracteres única, la cual es resultado de la ejecución de un algoritmo de encriptación, dada otra cadena de caracteres como entrada. Esta contraseña **hash** es la que se almacena en el sistema, y no la contraseña original. Durante el proceso de inicio de sesión, se comprueba el **hash** de la contraseña insertada por el usuario y la contraseña **hash** almacenada, verificando así la integridad de la misma.

Abrimos el fichero **/etc/shadow** y encontramos la información que necesitamos. La imagen 3.7 muestra el contenido de este fichero.

```

root:$6$C58tCb.868a0C3M0YxNvNz/AYUaekzYrI3Dft0w8gZrvA8/hj3Ex/N2k733oZ4xSHm68KwYFRXIT7Pd32ChCgckXUZaK6/:15222:0:99999:7:::
daemon:*:15222:0:99999:7:::
bin:*:15222:0:99999:7:::
sys:*:15222:0:99999:7:::
sync:*:15222:0:99999:7:::
games:*:15222:0:99999:7:::
man:*:15222:0:99999:7:::
lp:*:15222:0:99999:7:::
mail:*:15222:0:99999:7:::
news:*:15222:0:99999:7:::
uucp:*:15222:0:99999:7:::
proxy:*:15222:0:99999:7:::
www-data:*:15222:0:99999:7:::
backup:*:15222:0:99999:7:::
list:*:15222:0:99999:7:::
lirc:*:15222:0:99999:7:::
gnats:*:15222:0:99999:7:::
nobody:*:15222:0:99999:7:::
libuid:*:15222:0:99999:7:::
sshd:*:15222:0:99999:7:::
lmuser:$6$2p0K0e5fYzL3Vo1BVoq0.nsTv5AQV4V0Tjh9KLzH0g0Svs9l.Eq05tePhXKcQlYdGYsvUZytSQZMB.Mq0lWCAeDb31:15222:0:99999:7:::
telnetd:*:15225:0:99999:7:::
ntp:*:15304:0:99999:7:::
messagebus:*:16358:0:99999:7:::

```

Figura 3.7 – Contenido del fichero **/etc/shadow** del LM7

Los usuarios que nos interesan son **root** y **lmuser**. Ambas contraseñas se han cifrado usando el algoritmo SHA-512, ya que su primer campo es \$6\$. En un primer intento se modifica el fichero y se eliminan los campos que contienen la contraseña *Hash*, con la intención de forzar el inicio de sesión sin la necesidad de contraseña (contraseña vacía). Sin embargo, este enfoque no produce el resultado esperado y se busca otra solución. Como conocemos tanto el formato de las entradas de este fichero como el algoritmo de cifrado utilizado, el segundo enfoque será sustituir estas claves *Hash* con otras nuevas, generadas por nosotros. Generamos una nueva contraseña usando la utilidad [OpenSSL](#) [®].

```
1 | $~ openssl passwd -6 <cadena>
```

Listado 3.2 – Generación de Hash usando cifrado SHA-512

3

La ejecución del comando [3.2](#) nos genera una clave Hash única para la cadena que le proporcionemos como entrada. La opción `-6` indica que debe usarse el algoritmo de cifrado SHA-512. Para más información sobre este comando y sus opciones puede consultarse la documentación oficial [?].

Una vez cambiadas las contraseñas por las que hemos generado, re-instalamos el [CompactFlash](#) en la placa base del limitador y se procede a encenderlo para comprobar que podemos acceder al sistema con las nuevas credenciales. Tras finalizar el arranque se verifica que el inicio de sesión con ambos usuarios es satisfactorio, y que por tanto, se tiene control sobre el sistema operativo. La conexión al equipo mediante [SSH](#) también funciona correctamente usando las nuevas contraseñas. Anteriormente, se ha comprobado que la configuración [SSH](#) del sistema permite la conexión mediante el usuario *root*. Para ello se comprueba que en el fichero de configuración en la ruta `/etc/sshd_conf` contiene la directiva *PermitRootLogin* y que su valor es YES. Tras comprobar el fichero, se verifica que la directiva existe y está configurada correctamente. Este no es el valor por defecto, por lo tanto ha tenido que ser activado con anterioridad. Además, se añade la siguiente directiva para garantizar el acceso a los usuario *root* y *lmuser*:

```
1 | AllowUsers root lmuser
```

Listado 3.3 – Directiva del fichero

3.3 Especificaciones del sistema

El primer paso para comenzar a investigar el equipo es descubrir ante que tipo de hardware nos enfrentamos. En secciones anteriores (imagen [3.1](#)) se mostró el hardware del limitador bajo estudio, y se comentó brevemente sus características. En esta sección, se va a realizar un análisis más profundo sobre las capacidades y características

hardware del sistema:

Nos encontramos antes un equipo en el que se ha montado una placa base de tipo industrial, en la que vienen integrados todos los recursos hardware necesarios para correr un sistema operativo. En la siguiente tabla pueden verse los componentes básicos con los que cuenta el sistema:

Placa base	ALIX-1E
Procesador	AMD Geode LX
Memoria	128/256MB SDRAM
Almacenamiento	CompactFlash 1GB
Interfaces	4xUSB, 1xVGA, 1xLPT, 2xCOM
Conectividad	Ethernet
Dimensiones	17x17cm

Tabla 3.1 – Especificaciones hardware del LM7

Como componentes que no forman parte de la placa base, tenemos una tarjeta de sonido USB conectada a uno de los puertos y un circuito integrado para las entradas y salidas de audio, conectado a la placa base mediante su puerto serie.

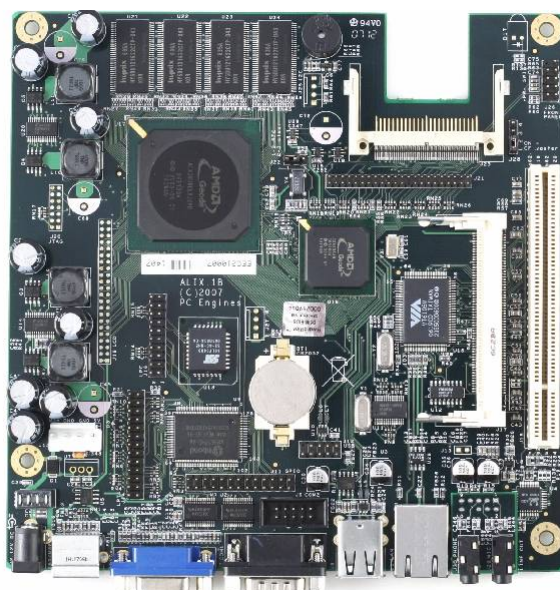


Figura 3.8 – Imagen de una placa base ALIX-1B [?]

3.4 Análisis del rendimiento

Tal y como se comentó en la primera sección de este capítulo, el ordenador auxiliar que se ha instalado en el laboratorio, y en red como el LM7, tiene Windows® 10 como

sistema operativo. Para facilitar el acceso remoto mediante [SSH](#) al equipo limitador, se hace uso de clientes [SSH](#). En la tabla 3.2 se lista el cliente usado tanto en Windows 10 como en Linux.

Sistema Operativo	Cliente SSH
Linux	Snowflake SSH
Windows	MobaXterm

Tabla 3.2 – Clientes [SSH](#) utilizados en el proyecto.

Estos clientes aportan ciertas ventajas, como por ejemplo:

- Permiten guardar los datos de acceso a equipos remotos. Una conexión [SSH](#) a otro equipo se resume en hacer click sobre un botón.
- Proporcionan una interfaz gráfica.
- Disponen de un navegador de archivos.
- Monitorizan el sistema remoto, lo que nos permite analizar el uso de recursos (disco, memoria, [CPU](#) y red) en tiempo real.

Tras configurar el cliente [SSH](#), se lanza una conexión remota al [LM7](#). Los datos de monitoreo muestran una alta tasa de utilización del [CPU](#), en torno al 85-100% (ver imágenes 3.9 y 3.10). Indudablemente, el hecho de que el [CPU](#) disponga de un único núcleo es significativo, sin embargo, se procede a identificar los procesos en funcionamiento que pertenecen al limitador. Para ello, se recurre al código fuente del limitador, en concreto al fichero [Makefile](#), el cual se encarga de compilar los distintos ejecutables (a los que llamaremos **módulos**) y de moverlos al directorio raíz del sistema: `/bin`. Al colocar los programas en esta carpeta, se pueden lanzar de forma global, es decir, sin especificar su ruta.

El fichero [Makefile](#) nos proporciona una lista de nombres de procesos a buscar. Listamos los procesos activos con la orden `ps`. El comando completo así como el resultado devuelto pueden observarse en las capturas de pantalla 3.9 y 3.10. Las opciones pasadas al invocar al comando `ps` permite mostrar no solo los procesos en activo, sino también las relaciones jerárquicas que existen entre ellos. La mayoría de los procesos del limitador son fácilmente detectables incluso en el supuesto de que se conocieran sus nombres con anterioridad. Los procesos relativos al limitador se encuentran remarcados en rojo, y se puede apreciar el detalle de que todos ellos tienen como proceso padre a `init`, con `PID` 1. Podemos ver esta información en la primera columna de la tabla, `PPID`. Esto significa que los procesos del limitador son lanzados automáticamente al arrancar el sistema.

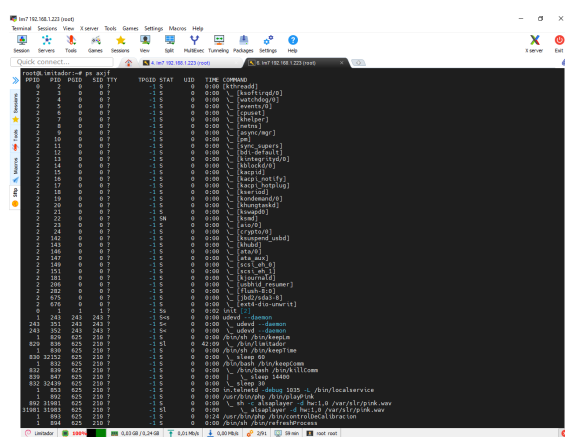


Figura 3.9

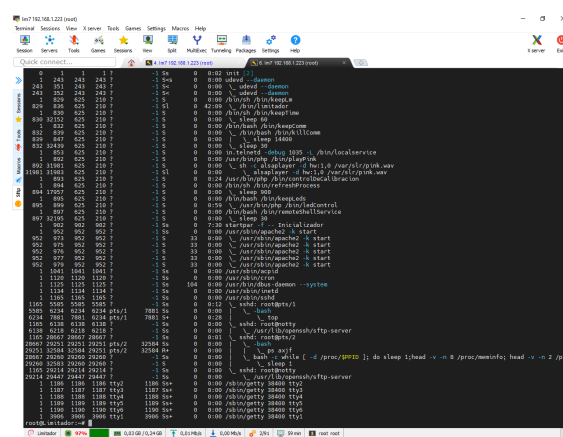


Figura 3.10

El proceso **init** o **systemd** es un gestor de servicios para sistemas operativos Linux. Es el primer proceso en arrancar (con PID 1) y el último en acabar durante el apagado. Su función es levantar los procesos a nivel de usuario.

!! EXPLICAR INIT.D

Los procesos que componen en limitador se verán con profundidad en la sección ??, pero antes, se presentará y estudiará la interfaz web del limitador en su totalidad (ya se presentó la sección principal con la imagen 3.4), mediante la que podemos configurarlo y visualizar las lecturas de los sensores y la atenuación aplicada en cada momento.

3.5 Interfaz web

La interfaz web del limitador se encuentra desplegada en un servidor **Apache**, y está programada usando **PHP**, **CSS**, **HTML** y **Javascript**. En la tabla 3.3 puede consultarse un análisis más detallado de las tecnologías usadas en la interfaz web.

Tecnología	Versión
Sistema Operativo	Debian 4.3.5
Servidor Web	Apache 2.2.16
PHP	5.5.3

Tabla 3.3 – Especificaciones del servidor web del LM7

El estudio del la interfaz representa un aspecto clave en el proceso de ingeniería inversa en el que nos encontramos. La interfaz web arroja una gran cantidad de información mediante la cual somos capaces no sólo de comprender qué hace el sistema, qué datos almacena y como los representa y exporta, sino que también nos

permite generar un conjunto de requisitos funcionales, no funcionales y de datos que nuestro proyecto deberá satisfacer, y como mínimo, debe hacerlo tan bien (o mal) como el sistema estudiado (requisitos de calidad). Por otra parte, será una guía y un recurso al cual recurriremos una y otra vez a lo largo de este proyecto. En definitiva, la interfaz web nos permite, de un vistazo, comprender el sistema y sus capacidades.

3.5.1 Imágenes

En esta subsección se va a presentar la interfaz web en imágenes de forma que el lector pueda ver la interfaz web por sí mismo, aunque de forma estática. Algunas de las imágenes han sido extraídas del manual de usuario original del limitador, mientras que otras han sido directamente tomadas de la interfaz web del limitador durante su funcionamiento.

!!! meter y explicar imágenes de la interfaz

3.5.2 Análisis técnico

En esta subsección profundizamos un poco más en la interfaz web recurriendo a su código fuente. Bajamos por tanto un escalón importante a nivel de abstracción. Por razones obvias, no se va a incluir la totalidad del código de la interfaz web del limitador en esta subsección, sino que se va a realizar un análisis técnico del mismo, destacando las cualidades o deficiencias más importantes que se han detectado y presentando bloques de código que den soporte al análisis y ayuden a comprenderlo.

Como se ha comentado en la introducción a esta sección, el código de la interfaz web se compone de código PHP, HTML, CSS y JS. Parte del código CSS y JS corresponde a librerías estáticas, descargas y usadas por el proyecto, como por ejemplo [jQuery](#), y su plugin [SlideJS](#), [dygraphs](#) o [ExCanvas](#). Estas librerías se usan de forma extensiva para construir toda la parte reactiva y asíncrona de la interfaz, como las acciones lanzadas mediante la pulsación de botones, el envío de datos al servidor y la actualización de datos en la interfaz, sin necesidad de recargar la página; para ello, se hace uso de AJAX. Las dos últimas librerías se usan concretamente para la generación de gráficos, en este caso, la gráfica principal que muestra las lecturas de los sensores y la atenuación aplicada por el limitador en cada momento.

En el esquema [3.6](#) podemos observar la estructura de la carpeta `www` dentro del código fuente del limitador, la cual contiene el código de la interfaz web. Por simplicidad, se han omitido algunos de los ficheros no esenciales, ya que el listado sería demasiado extenso y la intención es mostrar la estructura y organización del proyecto que materializa la aplicación web, así como describir brevemente el propósito y responsabilidad de cada fichero. Para generar esta representación del directorio y su

contenido se ha utilizado la herramienta [GIO](#), disponible en distribuciones de Linux.

```

1  # gio tree .../lms.7/lms/www
2
3  file:///lms.7/lms/www
4  |-- access.php                # Gestión de usuarios, sesiones y permisos
5  |-- authPart-ExtendedNormative.php
6  |-- calibrationControl.lib.php
7  |-- config.xml
8  |-- configFail.inc
9  |-- configuration.php         # Definición de variables globales
10 |-- css/
11 |-- fonts/
12 |-- functions/               # Scripts AJAX
13 |   |-- calibrate.php        # Actualizador de calibración
14 |   |-- network.php          # Controlador de configuración de red
15 |   |-- pink.php             # Control de la emisión de ruido rosa
16 |   |-- restart.php
17 |   |-- soundControl.php
18 |   |-- status.php           # Consulta el estado del limitador
19 |   |-- updateCnfg.php        # Actualizador de configuración
20 |   |-- updateDataCu.php      # Actualizador de datos del local
21 |   |-- updateExtendedNormative.php # Actualizador de la normativa
22 |   |-- updatePwd.php         # Actualizador de contraseña
23 |   |-- updateTime.php        # Actualizador de fecha y hora
24 |   `-- validateUserCode.php
25 |-- help.png
26 |-- images/
27 |-- index.php                # Controlador principal
28 |-- jquery.css
29 |-- js/                      # Librerías JS (jQuery, dygraph, slide, excanvas..)
30 |-- leftEqualizer.php        # Template de ecualizador izquierdo (HTML, CSS, JS, PHP)
31 |-- login.php                # Script de login (PHP)
32 |-- logoPagina.png
33 |-- logs.php                 # Logger
34 |-- manual.php
35 |-- manual3.pdf              # Manual de usuario de los limitadores serie 3
36 |-- manual7.pdf              # Manual de usuario de los limitadores serie 7
37 |-- new/
38 |-- old/                     # Versiones antiguas de la aplicación web
39 |   |-- www/
40 |-- reportOutputTest.html
41 |-- reports/                 # Generación y exportación de informes
42 |   |-- CreateReport.php
43 |   |-- PhpDataGetter.php
44 |   |-- data.php
45 |   |-- data.xml.php
46 |   |-- engine.php
47 |   |-- engine_render.php
48 |   |-- engine_render2.php
49 |   |-- engine_render3.php
50 |   |-- getReport.php
51 |   |-- images/
52 |   |-- index.php
53 |   |-- logoReports.png
54 |   |-- reportContent.php
55 |   |-- reportError.html
56 |   |-- reportError.php
57 |   |-- reportHasNoScreenableOutput.html
58 |   |-- reportHelp.html
59 |   |-- reportTemp -> /tmp/templateTempDirectory/
60 |   |-- saveCreatedReport.php
61 |   |-- testEngine.php
62 |   |-- testSaved.php
63 |   `-- tmp -> /tmp
64 |-- rightEqualizer.php       # Template de ecualizador derecho (HTML, CSS, JS, PHP)
65 |-- scripts.js               # Código JS global (la mayoría está embebido en los .php)
66 |-- simple/                  # Versión simplificada de la web
67 |-- smallInfoPanel.php
68 |-- statusPart.php
69 |-- style-Blue.css
70 |-- style.css
71 |-- tabAuth.php              # Template del panel superior si se está autenticado
72 |-- tabAuthRegister.php      # Template del panel superior para el registro
73 |-- tabNormal.php            # Template del panel superior si NO se está autenticado
74 |-- templateMng.php
75 |-- tests/
76 |-- tractis/                 # DNI-e (no usado)
77 `-- users.php                # Gestión de usuarios (deprecated en favor de access.php)

```

Listado 3.4 – Estructura de directorios y ficheros de la interfaz web

La página web se genera de forma dinámica mediante PHP, en función de:

- Tipo de usuario (roles y permisos).
- Si el usuario ha iniciado sesión o no.
- Tipo de sistema (registrador o limitador).
- Los datos (actualización asíncrona).

La generación de código dinámico se basa en la renderización condicional de bloques de código fuente, así como la inclusión (a veces también condicional) de otros ficheros con extensión PHP. Es importante destacar que en este aspecto **no se hace uso de ninguna herramienta** que ayude a gestionar la generación dinámica del código, como puede ser un **gestor de plantillas** como Twig. Esto, junto con el **uso de código en varios lenguajes** (PHP, JS, HTML y CSS) de manera simultánea en el mismo fichero e incluso **en el mismo bloque de código** hace que el código de la aplicación web sea **altamente complejo y difícil** de leer, entender y sobretodo, **de mantener**. Podemos observar un ejemplo claro de esto en el siguiente extracto de código, perteneciente al fichero *tabAuth.php*:

```

1  ...
2  <div id="chTime" title="Cambio de hora">
3  <p style="font-size:bigger; font-weight:bold;">Cambio de hora del sistema</p>
4
5  <p>Hora actual <span id="chTime_actualTime"></span></p>
6  <p>
7  <form id="chTime_form">
8  Nueva hora <input type="text" id="chTime_new" name="chTime_new" style="width:120px" value="" <?php if (!
   $configurator){ echo( 'disabled="disabled"'); }?> </input>
9  </form>
10 </p>
11 <br/>
12  Recuerde: El sistema actualizar&aacute; autom&aacute;ticamente la hora al estar
   conectado a internet.
13 <table width=100%>
14 <tr>
15 <td>
16 <input style="float:right" type="button" value="Cerrar" id="" class="bt_register" onclick="javascript:$('#
   chTime').dialog('close');" />
17 </td>
18 <td align=right>
19 <?php if ($configurator){ ?>
20 <input style="float:right" type="button" value="Ok" id="" class="bt_register" onclick="javascript:
   ChTime_UpdateClock();" />
21 <?php } ?>
22 </td>
23 </tr>
24 </table>
25 </div>
26 <script>
27 $('#chTime_new').datetimepicker({
28   dateFormat: 'yy/mm/dd'
29 });
30 </script>
31
32 <script>
33 function ChTime_UpdateClock() {
34   $.ajax({
35     data: $('#chTime_form').serialize(),
36     type: "POST",
37     dataType: "json",
38     url: "functions/updateTime",
39     success: function(data) {
40       alert("Hora actualizada");

```

```

41 |         }
42 |     });
43 | };
44 | </script>
45 | ...

```

Listado 3.5 – Código HTML, CSS, JS y PHP usado de forma conjunta

El código 3.5 ha sido ligeramente indentado y reestructurado para mejorar su comprensión, sin embargo, al problema que supone desarrollar código como el que acabamos de ver, hay que sumar una indentación prácticamente inexistente, la ausencia casi total de comentarios y la prolongación de líneas de código por encima de los 100 caracteres de longitud.

Tan solo los ficheros *tabNormal.php*, *tabAuth.php* y *tabAuthRegister.php* suman un total de 3000 líneas de código, por lo que la complejidad del código crece rápidamente en un proyecto de estas características cuando se siguen malas prácticas como las que se acaban de ver. Además de las claras deficiencias del código en sí, la interfaz web está **fuertemente acoplada** al sistema operativo y al software del limitador, ya que la comunicación IU-Limitador se realiza exclusivamente mediante ficheros o mediante la invocación directa de procesos / servicios del limitador que se ejecutan sobre el sistema operativo. Estos dos procedimientos de comunicación se verán en detalle en la siguiente sección.

En definitiva, la complejidad del código debido a malas prácticas y las complejidades técnicas que supone el uso de PHP 5 a la fecha de la redacción de este documento provocan que no sea factible la reutilización de la aplicación web, y por tanto, queda exclusivamente como objeto de estudio y recurso de soporte a nuestro proyecto.

A fecha de 10 de agosto de 2021, la última versión estable de PHP es la versión 8.0.8, con fecha de lanzamiento el 1 de julio de 2021. La versión 5.5.3 de PHP usada en la aplicación web del limitador LM7 fue lanzada el 22 de agosto de 2013, es decir, hace 8 años.

3.5.3 Comunicación IU-Limitador mediante ficheros

Los ficheros son usados de forma muy extensa por el limitador en su conjunto, y en general, interpretan el papel tanto de almacenamiento persistente de datos como vía de comunicación entre procesos, y entre estos y la interfaz web. Estos últimos serán los estudiados en esta sección.

Es tanta la importancia de los ficheros que **el limitador no hace ningún uso de bases de datos**, sino que almacena todo lo que requiere almacenar en ficheros de texto o directamente en formato binario. El control (escritura y lectura) de estos ficheros requiere el uso de procesos diseñados e implementados exclusivamente para este propósito, y que

se verán en detalle en la sección: ??.

Los ficheros que utiliza la interfaz web son los siguientes:

- `/var/slr/users.auth`
 - Almacén de cuentas de usuario, contraseñas y permisos.
 - Accedido por `access.php`.
- `/var/slr/logs.serial`
 - Registros (logs) sobre las acciones que realizan los usuarios.
 - Accedido por `logs.php`
- `/var/slr/sessions.serial`
 - Registros (logs) sobre las sesiones de calibración.
 - Se indica el comienzo y el final de la cada [sesión](#)¹, así como el valor de calibración.
 - También registra la hora de arranque del sistema.
 - Almacenado en formato **base64**.
 - Accedido por `logs.php`.
- `/var/slr/network.config`
 - Almacena la configuración de red del equipo para su lectura.
 - Accedido en varias partes de la UI.
- `/var/slr/network.script`
 - Almacena la configuración de red del equipo para su actualización.
 - Corresponde a un script Bash que hace uso del comando `ifconfig` y `route`.
 - Accedido en varias partes de la UI.
- `/tmp/inSession`
 - Fichero vacío para indicar si el equipo se encuentra en una [sesión](#), a modo de variable de estado.
 - Los procesos que necesitan conocer esta información detecta el archivo y actúan en consecuencia [`fopen()`].

¹Una sesión corresponde al intervalo de tiempo durante el cual el limitador se encuentra funcionando de forma activa debido a que los valores de recepción están por encima de un determinado umbral

- Se elimina al finalizar la sesión.
- Accedido por `calibrationControl.php`.
- `/tmp/conf.tmp`
 - Fichero temporal para almacenar la configuración recientemente cambiada por el usuario desde la UI.
 - Inmediatamente después de la actualización de este archivo por parte de la UI, se invoca a un proceso del limitador para que recoja esta nueva información y aplique la nueva configuración.
 - Accedido por varias partes de la UI.
- `/tmp/pink`
 - Fichero vacío que indica si se debe emitir ruido rosa, a modo de variable de estado.
 - Los procesos que necesitan conocer esta información detectan el archivo y actúan en consecuencia (emitir ruido rosa) [`fopen()`].
 - Accedido por `calibrationControl.php`.
- `/tmp/externalAttenuation`
 - Utilizado para indicar si se debe aplicar un nuevo valor de atenuación.
 - Almacena el valor de atenuación indicado por el usuario en la UI.
 - El proceso principal del limitador detecta este fichero, lee el nuevo valor de atenuación contenido en él, y lo aplica.
 - Accedido por `calibrationControl.php`.

Todos los ficheros son de relativa importancia ya que desempeñan un papel en el limitador, pero de entre todos ellos destaca notablemente por su importancia el fichero `/tmp/conf.tmp`. Este fichero es la principal vía de comunicación entre la interfaz web y el limitador, y a través de él fluyen todas las configuraciones definidas por el instalador o el usuario. En primer lugar, se configura el sistema desde la interfaz web y mediante botones, se aplica la nueva configuración. El proceso de configuración es transparente para el usuario de la interfaz y tiene efecto inmediato, pero lo que en realidad ocurre es que esta nueva configuración ha sido volcada al fichero `/tmp/conf.tmp` para que sea procesada por el proceso del limitador encargado de ello. Por supuesto, los contenido del fichero debe estar en un formato que ambas partes conozcan de forma que la comunicación pueda llevarse a cabo: la **interfaz**.

!!! EXPLICAR FORMATO DEL FICHERO `tmp/conf`

3.5.4 Comunicación IU-Limitador mediante procesos

En esta sección se va describir una lista de procesos pertenecientes al software del limitador los cuales son contactados de forma directa por la interfaz web. De nuevo, los procesos del limitador se listarán y explicarán en su totalidad en la sección ??.

Los ficheros que utiliza la interfaz web son los siguientes:

- getConfig
 - Devuelve los datos del local y la configuración del limitador en formato XML o JSON.
- getStatus
 - Devuelve los datos del local e información sobre el estado del limitador:
 - * Tipo de sistema (registrador o limitador).
 - * Lecturas.
 - * Atenuación actual.
- limInfo
 - Registros (logs) sobre las sesiones de calibración.
 - Se indica el comienzo y el final de la cada [sesión²](#), así como el valor de calibración.
 - También registra la hora de arranque del sistema.
 - Almacenado en formato **base64**.
 - Accedido por `logs.php`.
- utilslr
 - Almacena la configuración de red del equipo para su lectura.
 - Accedido en varias partes de la UI.
- localservice
 - Almacena la configuración de red del equipo para su actualización.
 - Corresponde a un script Bash que hace uso del comando `ifconfig` y `route`.
 - Accedido en varias partes de la UI.

²Una sesión corresponde al intervalo de tiempo durante el cual el limitador se encuentra funcionando de forma activa debido a que los valores de recepción están por encima de un determinado umbral

3.6 Estructura del proyecto

De la misma forma que se hizo en con la interfaz web, en este apartado se va a presentar la estructura de directorios del proyecto y sus ficheros más importantes. De nuevo, algunos de ellos se han eliminado del listado por ser ficheros no útiles para el proyecto, normalmente restos de versiones anteriores. Asimismo, los ficheros que contiene el código de la interfaz web se ha omitido ya que ha sido estudiado en el listado 3.6.

```

1  # gio tree .../lms.7/lms/
2
3  file:///lms.7/lms
4  |-- Makefile
5  |-- comun
6  |   |-- Calibracion.cpp
7  |   |-- Calibracion.h
8  |   |-- Config.h
9  |   |-- Configuracion.cpp
10 |   |-- Configuracion.h
11 |   |-- Configurador.cpp
12 |   |-- Configurador.h
13 |   |-- EstadoDeModificadores.cpp
14 |   |-- EstadoDeModificadores.h
15 |   |-- EstadoDelLimitador.cpp
16 |   |-- EstadoDelLimitador.h
17 |   |-- Funciones.cpp
18 |   |-- Funciones.h
19 |   |-- Makefile
20 |   |-- Serial.cpp
21 |   |-- TestConfig.cpp
22 |   |-- aes.cpp
23 |   |-- aes.h
24 |   |-- compacc
25 |   |-- compact.cc
26 |   |-- gSerial
27 |   |-- getSerial.cpp
28 |   |-- serial.h
29 |-- comv2
30 |   |-- Funciones.h
31 |   |-- Main.cpp
32 |   |-- Makefile
33 |   |-- Test.cpp
34 |   |-- TestRegistrador.cpp
35 |   |-- WhiteRabbit.cpp
36 |   |-- WhiteRabbit.h
37 |   |-- base64.c
38 |   |-- cliente.c
39 |   |-- encrypter.cpp
40 |   |-- funciones_Conexion.cpp
41 |   |-- funciones_envioDeDatos.cpp
42 |   |-- funciones_formato.cpp
43 |   |-- funciones_hora.cpp
44 |   |-- globales.h
45 |   |-- slrComm
46 |-- configuracion
47 |-- connectNetwork
48 |-- construye
49 |-- defines.txt
50 |-- instalador
51 |   |-- MAKEDEV
52 |   |-- fdiskCommands
53 |   |-- lm70if.localCommands
54 |   |-- sdbInstall
55 |   |-- sdcInstall
56 |-- limitador
57 |   |-- Atenuador.cpp
58 |   |-- AtenuadorMk163.cpp
59 |   |-- AtenuadorPga.cpp
60 |   |-- AtenuadorPga.h
61 |   |-- Calibracion.cpp
62 |   |-- Calibracion.h
63 |   |-- CalibrationTest.cpp
64 |   |-- Estado.cpp
65 |   |-- Estado.h
66 |   |-- Lector.cpp
67 |   |-- LectorAlternativo.cpp

```

```
68 | | | | -- Lectura2C.cpp
69 | | | | -- Limitador.cpp
70 | | | | -- Makefile
71 | | | | -- PruebaAtenuador
72 | | | | -- PuertoParalelo.cpp
73 | | | | -- Registrador.cc
74 | | | | -- Registro.cpp
75 | | | | -- Registro.h
76 | | | | -- ServidorDeEstado.cpp
77 | | | | -- ServidorDeEstado.h
78 | | | | -- Sonometro.cpp
79 | | | | -- Sonometros.cpp
80 | | | | -- SwitchPink.cpp
81 | | | | -- TP.cpp
82 | | | | -- TestAtenuador.cpp
83 | | | | -- TestPga.cpp
84 | | | | -- audioOrder
85 | | | | -- filterTest.cpp
86 | | | | -- filtroA
87 | | | | `-- filtroA.cpp
88 | | | | -- iir.cpp
89 | | | | -- limitador
90 | | | | -- mixers
91 | | | | -- octaveBandFilter.c
92 | | | | -- qiir
93 | | | | | | -- compila.sh
94 | | | | | | -- datosSOS_fbanks_scaled.h
95 | | | | | | -- in_int.dat
96 | | | | | | -- out_int.dat
97 | | | | | | -- pruebaiirq
98 | | | | | | -- pruebaiirq.c
99 | | | | | | -- qiir.c
100 | | | | | | -- qiir.h
101 | | | | | | -- tQiir.c
102 | | | | | | `-- tqiir
103 | | | | -- sonometro
104 | | | | -- sonometros
105 | | | | -- testAt
106 | | | | -- testDeCalibracion.cpp
107 | | | | `-- testPga
108 |-- preparatoria
109 |-- reports
110 | | | | -- Makefile
111 | | | | -- cabeceraInformes.svg
112 | | | | -- dataGet.cpp
113 | | | | -- dygraph-combined.js
114 | | | | -- excanvas.min.js
115 | | | | -- gRegList.php
116 | | | | -- gReport.php
117 | | | | -- getConfig.cpp
118 | | | | -- getData.cpp
119 | | | | -- getEvents
120 | | | | -- getInputs.cpp
121 | | | | -- getStatus.cpp
122 | | | | `-- testPerformance.php
123 |-- scripts
124 | | | | -- connectNetwork
125 | | | | -- continuousPink
126 | | | | -- controlDeCalibracion
127 | | | | -- keepComm
128 | | | | -- keepLeds
129 | | | | -- keepLm
130 | | | | -- keepScreen
131 | | | | -- keepTime
132 | | | | -- killComm
133 | | | | -- ledControl
134 | | | | -- playPink
135 | | | | -- refreshProcess
136 | | | | `-- remoteShellService
137 |-- setAsNew
138 |-- tests
139 | | | | -- estadisticos.cpp
140 | | | | -- estadisticos.php
141 | | | | -- normativaExtendida.cpp
142 | | | | -- tActivo
143 | | | | -- tActivo.cpp
144 | | | | -- tReg.cpp
145 | | | | -- tcAl
146 | | | | `-- testDeCal.cpp
147 |-- utiles
148 | | | | -- AutoEqualizador.cpp
149 | | | | -- BoanergesConfigurator.cpp
150 | | | | -- Inicializador.cpp
151 | | | | -- LimInfo.cpp
152 | | | | -- Linet.cpp
```

```

153 | | | | -- Makefile
154 | | | | -- ServidorSerie .cpp
155 | | | | -- autoEq
156 | | | | -- controlDeCalibracion
157 | | | | -- gSerial
158 | | | | -- gen
159 | | | | -- gen .cpp
160 | | | | -- getSerial .cpp
161 | | | | -- helper .cpp
162 | | | | -- playPink
163 | | | | -- serial .h
164 | | | | -- utilslr .cpp
165 | | | | -- x .cpp
166 | | | | -- www

```

Listado 3.6 – Estructura de directorios y ficheros de la interfaz web

El proyecto se encuentra segmentado en carpetas, en lo que se asume que es un intento de organizar el código por categorías en base a funcionalidades comunes. Cada uno de los directorios, además del conjunto de ficheros que compone el código fuente, suele contener un fichero Makefile mediante el cual compilar el código fuente y generar ejecutables de forma automatizada, por lo que cada carpeta, por lo general, proporciona uno o varios ejecutables. Aunque en esta versión el concepto no termina de tomar significado, de aquí en adelante nombraremos a cada uno de estos directorios **módulo**, ya que a priori, parece que la idea de los programadores originales era el generar distintos módulos que trabajasen de forma conjunta, y de ahí la separación en distintas carpetas.

Esta segregación sin embargo, lejos de mejorar la situación, la empeora considerablemente, añadiendo una complejidad innecesaria debido a las siguientes razones:

1. Aunque a primera vista parece que los ejecutables y su código queda perfectamente englobado en su directorio, la realidad es que prácticamente todos ellos necesitan de código pertenecientes a otros módulos, por lo que los módulos acaban siendo muy interdependientes y las constantes inclusiones a código que se encuentra en otras carpetas acaba complicando y ensuciando el código.
2. Se ha decidido crear un Makefile para cada módulo en lugar de un solo Makefile global que se encargue de gestionar la compilación y el enlazado del código, por lo que en lugar de gestionar un solo Makefile hay que gestionar tantos Makefiles como módulos haya más el Makefile general, el cual se encarga de invocar a cada uno de los Makefiles de los módulos.

3.7 Procesos del limitador LM7

??

3.8 Esquema general

Capítulo 4

Análisis del sistema

4

Capítulo 5

Diseño del sistema

5

Capítulo 6

Implementación del sistema

6

Capítulo 7

Validación y test

Capítulo 8

Conclusiones y trabajo futuro

Apéndice A

Presupuesto

A.1 Recursos físicos

A.2 Recursos humanos

A.3 Software

A.4 Presupuesto final

1

Apéndice B

Normativa acústica de Granada

ANEXO I.

TABLA Nº 1. NIVELES LIMITE DE INMISION DE RUIDO EN EL INTERIOR DE LAS EDIFICACIONES.

NIVEL ACUSTICO DE EVALUACION. NAE

		Niveles Límites (dBA)	
		Día (7-23)	Noche (23-7)
ZONIFICACION	TIPO DE LOCAL		
	Equipamiento		
	Sanitario y bienestar social	30	25
	Cultural y religioso	30	30
Servicios Terciarios	Educativo	40	30
	Para el ocio	40	40
	Hospedaje	40	30
	Oficinas	45	35
	Comercio	55	45
		Día (7-23)	Noche (23-7)
ZONIFICACION	TIPO DE LOCAL		
Residencial	Piezas habitables, excepto cocinas y cuartos de baño	35	30
	Pasillos, aseos y cocinas	40	35
	Zonas de acceso común	50	40

TABLA Nº 2. NIVELES LIMITE DE EMISION DE RUIDO EN EL EXTERIOR DE LAS EDIFICACIONES.

NIVEL DE EMISION EXTERIOR. NEE

		NIVELES LIMITES (dBA)	
		Día (7-23)	Noche (23-7)
SITUACION ACTIVIDAD			
Zona de equipamiento sanitario.		60	50
Zona con residencia, servicios terciarios, no comerciales o equipamientos no sanitarios.		65	55
Zona con actividades comerciales.		70	60
Zona con actividad industrial o servicio urbano excepto servicios de administración.		75	70

TABLA Nº 3. NIVELES LIMITE DE RUIDO AMBIENTAL EN FACHADAS DE EDIFICACIONES

		Niveles Límite (dBA)	
		Día (7-23)	Noche (23-7)
Area de Sensibilidad Acústica			
Tipo I (Area de Silencio)		L _{Aeqn} 55	L _{Aeqn} 40
Tipo II (Area Levemente Ruidosa)		55	45
Tipo III (Area Toleradamente Ruidosa)		65	55
Tipo IV (Area Ruidosa)		70	60
Tipo V (Area Especialmente Ruidosa)		75	65

TABLA Nº 4. CURVAS BASE LIMITE DE INMISION DE VIBRACIONES EN EL INTERIOR DE LAS EDIFICACIONES ESTANDARES LIMITADORES PARA LA TRANSMISION DE VIBRACIONES

Uso del recinto afectado	Período	Curva Base
SANITARIO	Diurno	1
	Nocturno	1
RESIDENCIAL	Diurno	2
	Nocturno	1,4
OFICINAS	Diurno	4
	Nocturno	4
ALMACEN Y COMERCIAL	Diurno	8
	Nocturno	8

GRAFICO Nº 1. CURVAS BASES DE NIVELES DE INMISION DE VIBRACIONES

		Aceleración (m/s2)			
Frecuencia, Hz	K1	K 1.4	K2	K4	K8
1	0,003600	0,005040	0,007200	0,014400	0,028800
1,25	0,003600	0,005040	0,007200	0,014400	0,028800
1,6	0,003600	0,005040	0,007200	0,014400	0,028800
2	0,003600	0,005040	0,007200	0,014400	0,028800
2,5	0,003720	0,005208	0,007440	0,014880	0,029760
3,15	0,003870	0,005418	0,007740	0,015480	0,030960
4	0,004070	0,005698	0,008140	0,016280	0,032560
5	0,004300	0,006020	0,008600	0,017200	0,034400
6,3	0,004600	0,006440	0,009200	0,018400	0,036800
8	0,005000	0,007000	0,010000	0,020000	0,040000
10	0,006300	0,008820	0,012600	0,025200	0,050400
12,5	0,007800	0,010920	0,015800	0,031200	0,062400
16	0,010000	0,014000	0,020000	0,040000	0,080000
20	0,012500	0,017500	0,025000	0,050000	0,100000
25	0,015600	0,021840	0,031200	0,062400	0,124800
31,5	0,019700	0,027580	0,039400	0,078800	0,157600
40	0,025000	0,035000	0,050000	0,100000	0,200000
50	0,031300	0,043820	0,062600	0,125200	0,250400
63	0,039400	0,055160	0,078800	0,157600	0,315200
80	0,050000	0,070000	0,100000	0,200000	0,400000

ANEXO II

TABLA I. LIMITES MAXIMOS DE EMISION SONORA POR MOTOCICLETAS Y CICLOMOTORES

Los límites máximos de nivel sonoro para ciclomotores y vehículos automóviles de cilindrada no superior a 50 c.c., serán:

De dos ruedas: 80 dBA.

De tres ruedas: 82 dBA.

Los límites para las motocicletas serán los siguientes:

Categoría de motocicletas

Cilindrada	Valores expresados en dB(A)
≤ 80 c.c.	78
≤ 125 c.c.	80
≤ 350 c.c.	83
= 500 c.c.	85
> 500 c.c.	86

TABLA II. LIMITES MAXIMOS DE EMISION SONORA PARA OTROS VEHICULOS

Categorías de vehículos. Valores expresados en dB(A)

- Vehículos destinados al transporte de personas con capacidad para 8 plazas sentadas como máximo, además del asiento del conductor: 80

- Vehículos destinados al transporte de personas con capacidad para más de 8 plazas sentadas, además del asiento del conductor, y cuyo peso máximo no sobrepase las 3,5 toneladas: 81

- Vehículos destinados al transporte de personas con capacidad para más de 8 sentadas, además del asiento del conductor, y cuyo peso máximo exceda las 3,5 toneladas: 82

- Vehículos destinados al transporte de personas con capacidad para más de 8 plazas sentadas, además del asiento del conductor, cuyo motor tenga una potencia igual o superior a 147 kW (ECE): 85

- Vehículos destinados al transporte de mercancías, que tengan un peso máximo que no exceda de 12 toneladas: 86