



Escuela
Politécnica
Superior

Aplicación multiplataforma para el Entrenamiento de Oído



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Sergiy Kazantsev

Tutor:

Miguel Ángel Teruel Martínez

Enero 2026



Universitat d'Alacant
Universidad de Alicante

Aplicación multiplataforma para el Entrenamiento de Oído

Desarrollo de un entrenador de oído multiplataforma para mejorar las
capacidades auditivas

Autor

Sergiy Kazantsev

Tutor

Miguel Ángel Teruel Martínez
Lenguajes y Sistemas Informáticos



Grado en Ingeniería Informática



Escuela
Politécnica
Superior



Universitat d'Alacant
Universidad de Alicante

ALICANTE, Enero 2026

Preámbulo

El desarrollo de este proyecto se llevó a cabo motivado tanto por la investigación de las tecnologías informáticas como por el interés personal sobre la música y armonía. Actualmente, los niños y adolescentes que cursan estudios musicales asisten a escuelas de música, donde se imparten los conceptos básicos de la estructura de las partituras, el lenguaje musical y el uso de los instrumentos.

Este trabajo tiene como objetivo facilitar el aprendizaje de música para los alumnos a través de una aplicación multiplataforma a la cual se podrá acceder en cualquier momento desde un dispositivo móvil o un navegador, principalmente durante el tiempo libre o a la hora de hacer los deberes musicales en casa. Para ello, se pretende diseñar una interfaz y experiencia de usuario *user-friendly*. Los alumnos podrán poner en práctica los conceptos teóricos aprendidos a través de la realización de tareas cortas basadas en el Aprendizaje Basado en Juegos. Cabe destacar que no es una aplicación para aprender la teoría musical, sino que ha sido creada pensando en el usuario que ya dispone de conocimientos previos, y sirve para el entrenamiento de sus capacidades auditivas y el refuerzo de las lecciones aprendidas en la escuela de música.

Tiene como finalidad abordar varios campos del aprendizaje de música, tales como el reconocimiento de las notas, de los intervalos, modos, tonalidades, acordes, etc. dividido en varios niveles accesibles una vez completado el nivel previo. Como el instrumento musical principal, se usará el piano, generando sus sonidos con MIDI y Soundfont.

El desarrollo de la aplicación se lleva a cabo haciendo uso de las Metodologías Ágiles, aplicando el desarrollo incremental basado en iteraciones. Además, se implementa la integración de gestión de usuarios. Dentro de la aplicación, cada usuario podrá registrarse e iniciar sesión con el fin de poder hacer un seguimiento de su evolución dentro de cada apartado de la aplicación (p. ej. de la sección del reconocimiento de intervalos).

Revisar el alcance una vez hecho el proyecto

Agradecimientos¹

Este trabajo no habría sido posible sin el apoyo y el estímulo de mi colega y amigo, Doctor Rudolf Fliesning, bajo cuya supervisión escogí este tema y comencé la tesis. Sr. Quentin Travers, mi consejero en las etapas finales del trabajo, también ha sido generosamente servicial, y me ha ayudado de numerosos modos, incluyendo el resumen del contenido de los documentos que no estaban disponibles para mi examen, y en particular por permitirme leer, en cuanto estuvieron disponibles, las copias de los recientes extractos de los diarios de campaña del Vigilante Rupert Giles y la actual Cazadora la señorita Buffy Summers, que se encontraron con William the Bloody en 1998, y por facilitarme el pleno acceso a los diarios de anteriores Vigilantes relevantes a la carrera de William the Bloody.

También me gustaría agradecerle al Consejo la concesión de Wyndham-Pryce como Compañero, el cual me ha apoyado durante mis dos años de investigación, y la concesión de dos subvenciones de viajes, una para estudiar documentos en los Archivos de Vigilantes sellados en Munich, y otra para la investigación en campaña en Praga. Me gustaría agradecer a Sr. Travers, otra vez, por facilitarme la acreditación de seguridad para el trabajo en los Archivos de Munich, y al Doctor Fliesning por su apoyo colegial y ayuda en ambos viajes de investigación.

No puedo terminar sin agradecer a mi familia, en cuyo estímulo constante y amor he confiado a lo largo de mis años en la Academia. Estoy agradecida también a los ejemplos de mis difuntos hermano, Desmond Chalmers, Vigilante en Entrenamiento, y padre, Albert Chalmers, Vigilante. Su coraje resuelto y convicción siempre me inspirarán, y espero seguir, a mi propio y pequeño modo, la noble misión por la que dieron sus vidas.

Es a ellos a quien dedico este trabajo.

Cambiar

¹Por si alguien tiene curiosidad, este “simpático” agradecimiento está tomado de la “Tesis de Lydia Chalmers” basada en el universo del programa de televisión Buffy, la Cazadora de Vampiros.<http://www.buffy-cavampiros.com/Spiketesis/tesis.inicio.htm>

*A mi esposa Marganit, y a mis hijos Ella Rose y Daniel Adams,
sin los cuales habría podido acabar este libro dos años antes ²*

Cambiar

²Dedicatoria de Joseph J. Roman en "An Introduction to Algebraic Topology"

*Toda la sabiduría humana
está contenida en estas dos palabras:
Esperar y tener esperanza.*

El Conde de Montecristo,
Alexandre Dumas.

Índice general

1	Introducción	1
1.1	El proceso de enseñanza musical	1
1.2	Entrenamiento del oído	2
1.2.1	Notas musicales	2
1.2.2	Instrumento utilizado	3
1.2.3	Octavas	3
1.2.3.1	Intervalos	4
1.2.4	Melodía	4
1.2.4.1	Escalas	5
1.2.4.1.1	Modos	5
1.2.5	Armonía	5
1.2.5.1	Tonalidades	5
1.2.5.2	Acordes	5
1.2.6	Ritmo	5
1.3	Problemas de enseñanza tradicional	5
2	Marco Teórico (Con ejemplos de listas)	6
2.1	Listas	6
2.2	Listas de definición	7
3	Objetivos (Con ejemplos de tablas)	9
3.1	Tablas	9
3.2	Otros diseños de tablas	11
	Bibliografía	13

Índice de figuras

1.1	Nombres de las notas musicales	2
1.2	Las octavas del piano (Díaz Caballero y cols., 2013)	3
1.3	Los intervalos de semitonos (<i>Referencia : semitonos cromáticos y diatónicos,</i> s.f.)	4

Índice de tablas

1.1	Los intervalos en música.	4
3.1	Ejemplo de tabla.	9
3.2	Parámetros optativos de los entornos flotantes	10
3.3	Parámetros de los altavoces elegidos de la marca Beyma®.	11
3.4	Ejemplo 2	12

Índice de Códigos

1 Introducción

Las aplicaciones de dispositivos móviles se han convertido en una parte importante y esencial en nuestras vidas cotidianas, alcanzando casi 96% de la población en países desarrollados entre la población joven (Atske, 2025). Este fenómeno ha provocado un auge tanto de las tecnologías nativas del móvil, como las de multiplataforma, alcanzando así una mayor cuota del mercado.

Este trabajo tiene como objetivo estudiar el framework multiplataforma Flutter y el proceso de desarrollo de una aplicación compatible con los navegadores Web y el sistema operativo Android.

1.1 El proceso de enseñanza musical

La música, en su manifestación, es una creación artística. Sin embargo, hoy en día, la enseñanza musical, aparte del arte, incluye el estudio teórico, la memorización y repetición de patrones. En España, es una asignatura de estudios primarios y secundarios obligatorios, aunque con el contenido impartido limitado a la lectura de partituras y el uso de instrumentos como la flauta o el ukulele.

Por otro lado, en España también existen docenas de conservatorios y escuelas musicales que tienen un programa de enseñanza más amplio y regulado por el Gobierno de España y de la Comunidad Autónoma. Dichas instituciones de enseñanza musical ofrecen varios niveles de profesionalización, siendo los estudios superiores de música el nivel más alto.

Este Trabajo Final de Grado (TFG) está destinado a las personas estudiantes de música que poseen conocimientos teóricos elementales y profesionales sobre la lectura y reconocimiento de las partituras y del lenguaje musical, que se alcanza durante la realización de estudios por las personas de edades de 8 a 18 años. Según el Real Decreto 1577/2006 (2006), los alumnos deben tener, entre otras, las siguientes habilidades:

- Desarrollar el oído interno tanto en el análisis como en la realización de ejercicios escritos.
- Utilizar el «oído interno» como base de la afinación, de la audición armónica y de la interpretación musical.
- Identificar a través de la audición:
 - Notas
 - Intervalos
 - Acordes
 - Modos

1.2 Entrenamiento del oído

El entrenamiento auditivo es una actividad de desarrollo de la destreza que le permite al músico a reconocer distintas notas, intervalos y modos al escucharlos, sin tener acceso a ninguna herramienta ni partitura. El entrenamiento del oído tiene muchas similitudes con el uso del oído en la vida cotidiana: su propósito es diferenciar y entender auditivamente. En el caso de la música, son los sonidos de un instrumento; en la vida cotidiana serían las palabras e intonaciones de la voz.

Dentro de este trabajo, se tratará sobre los elementos auditivos puramente sonoros, sin considerar los aspectos artísticos y emocionales que transmite la música. Los elementos esenciales del sonido son:

- La altura: definida con un nombre (Do, Re; A, B)
- La intensidad
- La duración

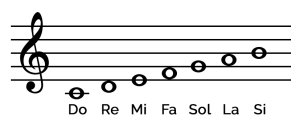
También se van a considerar las estructuras como:

- Estructuras melódicas: modos y escalas
- Estructuras armónicas: tonalidades, acordes
- Ritmos

A continuación se definen los conceptos esenciales del sonido y música.

1.2.1 Notas musicales

Las notas musicales son sonidos con una determinada frecuencia (Hz). La altura de una nota es directamente proporcional a su frecuencia. Cada nota tiene su propio nombre:



(a) Sistema solfeo



(b) Sistema anglo-sajón

Figura 1.1: Nombres de las notas musicales

En este trabajo, se utilizará el sistema solfeo para referirse a las notas. Cada nota musical también puede representarse en un ordenador a través del protocolo Musical Instrument Digital Interface (MIDI), tema que se abordará en los siguientes capítulos.

1.2.2 Instrumento utilizado

Para la mayoría de los estudiantes del conservatorio, el piano se convierte en el primer instrumento con el cual se acostumbran a lo largo de su formación musical. Incluso si el estudiante decide especializarse en otro instrumento, el piano resulta fundamental para impartir las asignaturas de teoría musical y el solfeo ¹.

Para el desarrollo de una aplicación de entrenamiento del oído, el piano es una opción adecuada, ya que es un instrumento que se puede virtualizar y representar las notas musicales con precisión y exactitud, incluso cuando el usuario no dispone de periféricos.

1.2.3 Octavas

En el piano hay 88 teclas, cada una de las cuales tiene una frecuencia determinada (desde 20 hz y hasta 20000 Hz). La distancia mínima en frecuencia/altura de cada tecla siempre es la misma, y se denomina "semitono". Gracias a los tonos y semitonos, se puede saber exactamente la distancia entre cada nota, por ejemplo: entre Do y Re la distancia es un tono, entre Do y Do# la distancia es un semitono (ver figura 1.2).

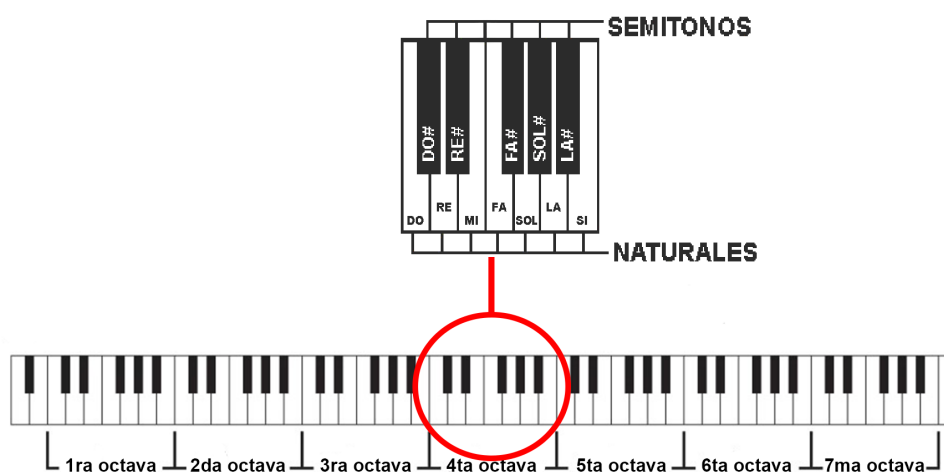


Figura 1.2: Las octavas del piano (Díaz Caballero y cols., 2013)

¹En este TFG, el solfeo se refiere al entrenamiento del oído interno y a la lectura de las partituras

1.2.3.1 Intervallos

Un intervalo es la diferencia entre dos sonidos aislados, y se puede referir tanto a los acordes (suenan simultáneamente) como a las melodías (sonidos sucesivos).

Los dos tipos de semitonos que existen según el intervalo son:

- **Semitono cromático:** Las dos notas del intervalo tienen el mismo nombre (Ver 1.3. Intervalo entre La-La \sharp).
- **Semitono diatónico:** Las dos notas del intervalo tienen nombres distintos (Ver 1.3. Intervalo entre La-Si \flat).

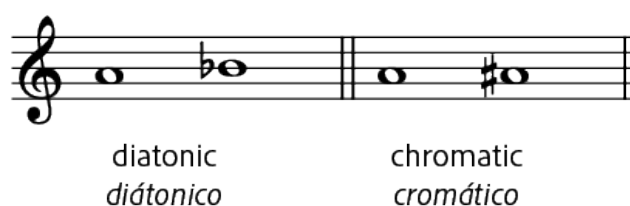


Figura 1.3: Los intervallos de semitonos (*Referencia : semitonos cromáticos y diatónicos, s.f.*)

Los intervallos de uno o más semitonos se clasifican en:

Nombre	Cantidad de semitonos
2ª Menor	1 Semitono
2ª Mayor	1 Tono
3ª Menor	1 Tono + 1 Semitono
3ª Mayor	2 Tonos
4ª Justa	2 Tonos + 1 Semitono
5ª Justa	3 Tonos + 1 Semitono
6ª Menor	4 Tonos
6ª Mayor	4 Tonos + 1 Semitono
7ª Menor	5 Tonos
7ª Mayor	5 Tonos + 1 Semitono
8ª Justa	6 Tonos

Tabla 1.1: Los intervallos en música.

1.2.4 Melodía

La melodía se define como una secuencia de notas o tonos con sus duraciones rítmicas determinadas. Dentro de una obra musical, la melodía se mantiene reconocible, y es el elemento que identifica la obra como tal. Un ejemplo claro de melodía sería la parte que interpreta el vocalista principal en una aria de Mozart, que se diferencia claramente del acompañamiento.

Los principales componentes de una melodía son la altura de la nota y su duración. Además, la melodía se construye sobre un modo específico, lo que le proporciona un carácter tonal y emocional.

1.2.4.1 Escalas

Se define una escala musical como un conjunto de notas consecutivas ascendentes o descendentes clasificadas por su altura, normalmente teniendo el tamaño de una octava. Las escalas sirven para construir obras musicales al igual que un pintor elige la paleta para su cuadro: mientras que el compositor selecciona un conjunto de notas, en el pintor elige un grupo de colores.

Cada denominación de cada escala se compone de dos elementos:

- La tónica: es la nota inicial que da el nombre a la escala. También puede ser un sostenido (#) o un bemol (b).
- El modo: define el carácter de la escala u obra.

1.2.4.1.1 Modos En el sistema musical occidental, las notas dentro de cada escala se organizan según el modo de la escala.

En música existen dos calidades emocionales: Mayor y Menor. La calidad mayor proporciona una apariencia "feliz" y "alegre" de la obra, mientras que el modo menor la hace "triste" y "melancólica".

1.2.5 Armonía

1.2.5.1 Tonalidades

1.2.5.2 Acordes

1.2.6 Ritmo

1.3 Problemas de enseñanza tradicional

Actualmente, la forma más eficiente del aprendizaje de la música, y la preferida por los tutores, es la asistencia a los conservatorios y escuelas de música de forma extracurricular. En consecuencia, el niño es obligado a asistir tanto a su escuela de primaria/secundaria como, durante su tiempo libre, a la escuela de música. Además, es obligado a la realización de las tareas académicas de ambas durante su tiempo libre, lo cual podría generar falta de motivación, resentimiento y el síndrome de agotamiento (*burnout*). Por otro lado, los alumnos también informan de la falta de reconocimiento y la falta de apoyo (Orzel, 2010).

Otro problema relevante está relacionado con los estudiantes de música procedentes de familias de bajos recursos económicos. Según Busby (2019), los niños procedentes de familias pobres tienen 3 veces más probabilidades de no participar en actividades extracurriculares, como los estudios musicales. En contraste, la mayoría de los padres de familias de bajos ingresos reconocen el efecto positivo que la música y la formación musical tienen sobre sus hijos (Ho y cols., 2020).

2 Marco Teórico (Con ejemplos de listas)

2.1 Listas

Hacer una lista es simple en \LaTeX . Para ello has de crear un entorno (así se llama) `itemize` con

```
\begin{itemize}
...
\end{itemize}
```

Y dentro de esa estructura, añadir cada elemento de la lista precedido de

```
\item primer ítem de lista
\item segundo ítem de lista
...
\item ultimo ítem de lista
```

Es importante que revises este texto tal como aparece en la plantilla y relaciones el aspecto que tiene el PDF final con cómo está escrito el documento \LaTeX .

Aquí va una lista con subtérminos:

```
\begin{itemize}
\item Ingeniería Informática.
\item Ingeniería Sonido e Imagen en Telecomunicación.
\item Ingeniería Multimedia.
    \subitem Mención: Creación y ocio digital.
    \subitem Mención: Gestión de Contenidos.
\end{itemize}
```

El resultado es el siguiente:

- Ingeniería Informática.
- Ingeniería Sonido e Imagen en Telecomunicación.
- Ingeniería Multimedia.
 - Mención: Creación y ocio digital.
 - Mención: Gestión de Contenidos.

Aquí va una lista con subtérminos pero numerada:

```
\begin{enumerate}
\item Ingeniería Informática.
\item Ingeniería Sonido e Imagen en Telecomunicación.
```

```

\item Ingeniería Multimedia.
\begin{enumerate}
  \item Mención: Creación y ocio digital.
  \item Mención: Gestión de Contenidos.
\end{enumerate}
\end{enumerate}

```

El resultado es el siguiente:

1. Ingeniería Informática.
2. Ingeniería Sonido e Imagen en Telecomunicación.
3. Ingeniería Multimedia.
 - a) Mención: Creación y ocio digital.
 - b) Mención: Gestión de Contenidos.

2.2 Listas de definición

Puedes realizar una lista de conceptos con su definición del siguiente modo:

```

\begin{description} % Inicio de la lista
\item[MAPP XT:] Programa desarrollado por \textit{Meyer Sound} para el diseño y ajuste de sistemas ↵
  ↵ formados por altavoces de su marca.
\begin{description} % Realiza una lista dentro de la lista
\item[Ventajas:]~
  El programa permite realizar múltiples ajustes tal como se podría realizar en la realidad con un ↵
  ↵ procesador real.

  Permite analizar la fase recibida en cualquier punto y compararla con otras mediciones.

  Dispone de varios tipos de filtros, inversiones de fase, etc.
\item[Inconvenientes:]~
  No existe una lista global de los altavoces ubicados en el plano, por lo tanto solo se pueden editar ↵
  ↵ seleccionándolos sobre el plano.

  Sólo permite diseñar en 2 dimensiones, principalmente sobre la vista lateral ya que los array de ↵
  ↵ altavoces no permite voltearlos.
\end{description}
\end{description}

```

Y \LaTeX genera lo siguiente:

MAPP XT: Programa desarrollado por *Meyer Sound* para el diseño y ajuste de sistemas formados por altavoces de su marca.

Ventajas: El programa permite realizar múltiples ajustes tal como se podría realizar en la realidad con un procesador real.

Permite analizar la fase recibida en cualquier punto y compararla con otras mediciones.

Dispone de varios tipos de filtros, inversiones de fase, etc.

Inconvenientes: No existe una lista global de los altavoces ubicados en el plano, por lo tanto solo se pueden editar seleccionándolos sobre el plano.

Sólo permite diseñar en 2 dimensiones, principalmente sobre la vista lateral ya que los array de altavoces no permite voltearlos.

3 Objetivos (Con ejemplos de tablas)

3.1 Tablas

Ahora veremos otra estructura más: las tablas.

Aquí va una tabla¹ para que se vea cómo insertar una tabla simple dentro del documento.

```
\begin{table}[h]
\centering
\begin{tabular}{llll}
& columna A & columna B & columna C \\
\hline
fila 1 & fila 1, columna A & fila 1, columna B & fila 1, columna C \\
fila 2 & fila 2, columna A & fila 2, columna B & fila 2, columna C \\
fila 3 & fila 3, columna A & fila 3, columna B & fila 3, columna C \\
\hline
\end{tabular}
\caption{Ejemplo de tabla.}
\label{tabladeejemplo}
\end{table}
```

	columna A	columna B	columna C
fila 1	fila 1, columna A	fila 1, columna B	fila 1, columna C
fila 2	fila 2, columna A	fila 2, columna B	fila 2, columna C
fila 3	fila 3, columna A	fila 3, columna B	fila 3, columna C

Tabla 3.1: Ejemplo de tabla.

L^AT_EX usa un sistema de parámetros para “decorar” las tablas. Puedes consultar estos parámetros en la tabla 3.2 de la página 10. La tabla se ubicará donde, a juicio de L^AT_EX, menos moleste por lo que puede no aparecer necesariamente donde se ha insertado en el texto original.

Existe la posibilidad de forzar que las tablas, figuras u otros objetos aparezcan en la zona del texto que se desea aunque en ocasiones puede dejar grandes espacios en blanco. El comando a utilizar es:

```
\FloatBarrier
```

Que introducido justo después de una tabla, figura, etc (después del comando `\end{...}`) fuerza la aparición en el texto, empujando el contenido.

¹En <http://www.tablesgenerator.com/> se puede encontrar un generador On-Line de tablas para L^AT_EX

Parámetro	Significado
h	Situa el elemento flotante <i>preferentemente</i> (es decir, si es posible) en la situación exacta donde se incluye este
t	Sitúa el elemento en la parte de arriba de la página
b	Sitúa el elemento en la parte de abajo de la página
p	Sitúa el elemento en una página aparte dedicada sólo a elementos flotantes; en el caso del formato article , ésta se sitúa al final del documento, mientras que para el book es colocada al final de cada capítulo

Tabla 3.2: Parámetros optativos de los entornos flotantes

También es posible elegir el ancho de cada columna y la orientación del texto en cada una. Por ejemplo:

```
\begin{table}[ht]
\centering
\begin{tabular}{|C{2cm}|C{2cm}|C{2cm}|C{2cm}|} % 4 columnas de 2cm — texto centrado y con bordes
\hline
\multicolumn{4}{|c|}{\textbf{\begin{tabular}{c}{@{}c@{}}FUENTE: TRÁFICO RODADO\end{tabular}}} \leftrightarrow
\leftrightarrow HORARIO: TARDE\end{tabular}} \\\hline
\textbf{dB(A)} & \textbf{Población expuesta tarde} & \textbf{\%} & \textbf{\scriptsize CENTENAS} \\\hline
\textbf{>70} & 0 & 0,000 & 0 \\\hline
\textbf{65 - 70} & 348,9 & 9,792 & 3 \\\hline
\textbf{60 - 65} & 1594,7 & 44,757 & 16 \\\hline
\textbf{55 - 60} & 322,1 & 9,040 & 3 \\\hline
\textbf{50 - 55} & 0 & 0,000 & 0 \\\hline
\textbf{>50} & 1297,3 & 36,410 & 13 \\\hline
\textbf{TOTAL} & 3563 & 100 & 35 \\\hline
\end{tabular}
\label{my-label}
\end{table}
```

L^AT_EX genera esto:

FUENTE: TRÁFICO RODADO HORARIO: TARDE			
dB(A)	Población expuesta tarde	%	CENTENAS
>70	0	0,000	0
65 - 70	348,9	9,792	3
60 - 65	1594,7	44,757	16
55 - 60	322,1	9,040	3
50 - 55	0	0,000	0
>50	1297,3	36,410	13
TOTAL	3563	100	35

Donde C{2cm} indica que la columna tiene el texto centrado y un ancho de 2 cm. Tambien

se puede utilizar $L\{\}$ o $R\{\}$ para poner el texto a la izquierda o derecha y definir un ancho concreto.

Páginas como <https://www.tablesgenerator.com/> ayudan a realizar tablas fácilmente, es lo más recomendado, ahorra mucho tiempo de trabajo y luego si falta algún detalle se puede retocar en el documento.

El formato estándar de las columnas es c , l o r , así lo genera la web mencionada antes, pero una vez generada puedes cambiar ese formato por el definido anteriormente para ajustar el ancho de las columnas, o mantenerlo así si el resultado ya es el deseado.

Para conocer más sobre las tablas puedes leer manuales como este: <https://latexlive.files.wordpress.com/2009/04/tablas.pdf> que contiene muchos ejemplos y explicaciones.

3.2 Otros diseños de tablas

Modelo	15LEX1600Nd	15P1000Fe V2
f_s (Hz)	41	45
R_e (ohm)	5.5	5.2
L_e (μH)	1600	1500
B_l (N/A)	25.7	27.4
M_{MS} (g)	175	157
C_{MS} ($\mu m/N$)	84	78
R_{MS} (kg/s)	6.8	7.6
d (cm)	33.5	33
V_{as} (dm ³)	91	80.7
Q_{TS}	0.36	0.30
Q_{MS}	6.6	5.9
Q_{ES}	0.38	0.31
Sens (dB @ 2.83V/1m)	96	98
η	1.7%	2.4%
S_d (cm ²)	880	855

Tabla 3.3: Parámetros de los altavoces elegidos de la marca Beyma®.

		140PU				50PU			
		Phase II		Phase I		Phase II		Phase I	
# BJet		≥ 4	2 or 3	≥ 4	2 or 3	≥ 4	2 or 3	≥ 4	2 or 3
# Bkg		123	76	12	7	84	35	7	3
Asimov	NM1	13	6	9	3	15	9	11	4
	NM2	6	2	4	1	7	3	5	1
	NM3	3	1	2	0	4	1	2	0
	STC	6	3	4	1	7	5	5	2

Tabla 3.4: Ejemplo 2

4 Metodología (Con ejemplos de figuras)

4.1 Inserción de figuras

Las figuras son un caso un poco especial ya que \LaTeX busca el mejor lugar para ponerlas, no siendo necesariamente el lugar donde está la referencia. Por ello es importante añadirle un “caption” y un “label” para poder hacer referencia a ellas en el párrafo correspondiente. Nosotros ponemos la referencia a la figura ?? que está en la página ??, justo aquí debajo, pero \LaTeX puede que la ubique en otro lugar. (observa el código \LaTeX de este párrafo para observar como se realizan las referencias. Estos detalles también se aplican a tablas y otros objetos).

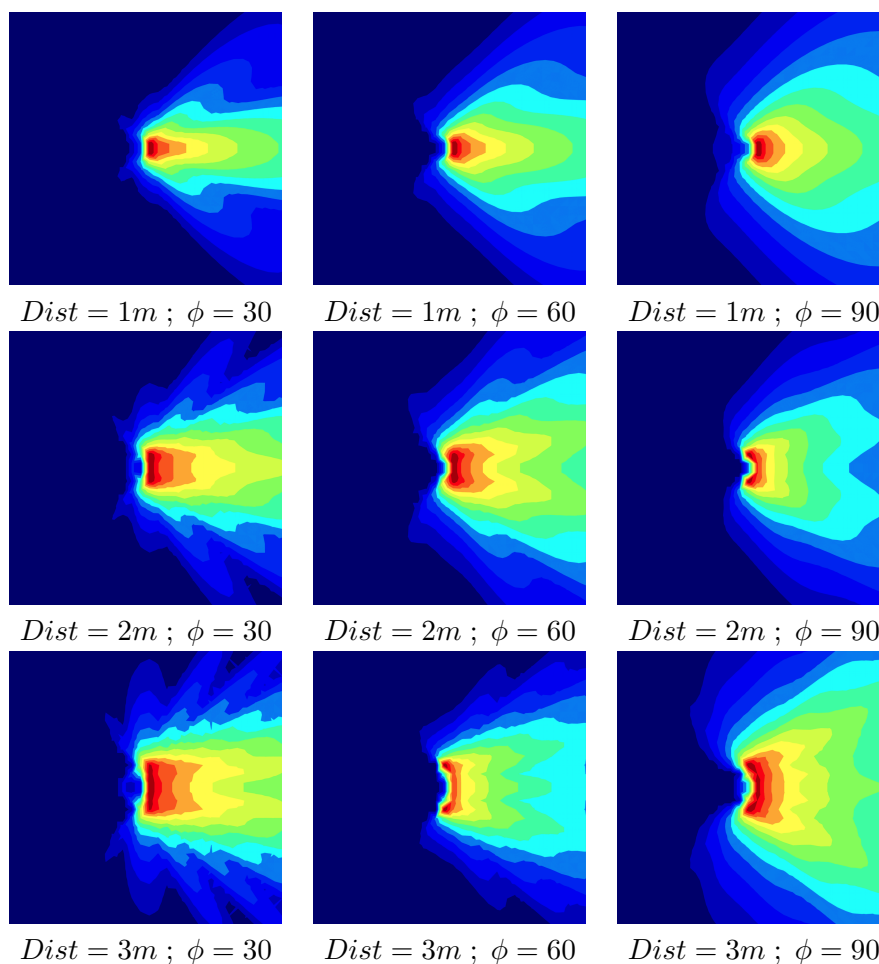
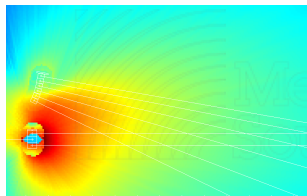


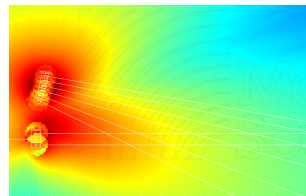
Tabla 4.1: Esta es una tabla con múltiples imágenes. Útil cuando se deben mostrar varias juntas.

Existe también la posibilidad de realizarlo sin tablas, con subfiguras:

```
\begin{figure}[h]
  \centering
  \begin{subfigure}[b]{0.4\textwidth} % Espacio horizontal ocupado por la subfigura
    \centering
    \includegraphics[width=4cm]{archivos/subs-sin} % Tamaño de la imagen
    \caption{Sin procesado.}
    \label{fig:gull}
  \end{subfigure}
  ~ % Añadir el espacio deseado, si se deja la línea en blanco la siguiente subfigura irá en una nueva línea
  \begin{subfigure}[b]{0.4\textwidth} % Espacio horizontal ocupado por la subfigura
    \centering
    \includegraphics[width=4cm]{archivos/subs-con} % Tamaño de la imagen
    \caption{Con procesado.}
    \label{fig:tiger}
  \end{subfigure}
  \caption{Ejemplo de subfiguras} \label{sistemass}
\end{figure}
```

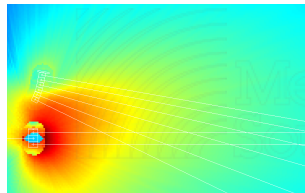


(a) Sin procesado.

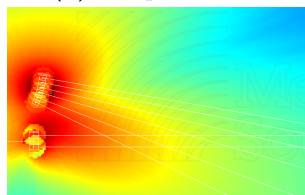


(b) Con procesado.

Figura 4.1: Ejemplo de subfiguras



(a) Sin procesado.



(b) Con procesado.

Figura 4.2: Ejemplo de subfiguras vertical

Si eliminas la línea '`\caption`' de las subfiguras, tendrás las imágenes sin la información individual, aunque sí con la principal. Y obviamente, si eliminas el de la figura no se mostrará ninguna información.

5 Desarrollo (Con ejemplos de código)

5.1 Inserción de código

A veces tendrás que insertar algún pedazo de código fuente para explicar algo relacionado con él. No sustituyas explicaciones con códigos enormes. Si pones algo de código en tu TFG que sea para demostrar algo o explicar alguna solución.

L^AT_EX te ayuda a escribir código de manera que su presentación tenga las marcas y tabulaciones propias de este tipo de texto. Para ello, debes poner el código que escribas DENTRO de un entorno que se llama “listings”. La plantilla ya tiene una serie de instrucciones para incluir el paquete “listings” y añadirle algunos modificadores por lo que no tienes que incluirlo tú. Simplemente, mete tu código en el entorno “lstlisting” y ya está. Puedes indicar el lenguaje en el que está escrito el código y así L^AT_EX lo mostrará mejor.

En el archivo *estiloscodigoprogramacion.tex* están definidos algunos lenguajes para mostrarlos con un diseño concreto, se pueden modificar para cambiar el coloreado del código, qué términos se ponen en negrita, etc. Si se quiere profundizar más en la función “listings” se puede consultar su manual en <http://osl.ugr.es/CTAN/macros/latex/contrib/listings/listings.pdf>, aunque hay mucha información en foros y blog’s que es más fácil de comprender.

Veamos un ejemplo en la figura ??:

```
\begin{lstlisting}[style=C, caption={ejemplo código C},label=C_code]
#include <stdio.h>
int main(int argc, char* argv[]) {
    puts("Hola mundo!");
}
\end{lstlisting}
```

El resultado será:

Código 5.1: ejemplo código C

```
1 #include <stdio.h>
2 // Comentario
3 int main(int argc, char* argv[]) {
4     puts("Hola mundo!");
5 }
```

Si lo quieres en color, está definido el estilo C-color en el archivo *estiloscodigoprogramacion.tex*, con algunos parámetros para mejorar la visualización:

```
\begin{lstlisting}[style=C-color, caption={ejemplo código C en color},label=C_code-color]
#include <stdio.h>
// Comentario
int main(int argc, char* argv[]) {
```

```
puts("Hola mundo!");
}
\end{lstlisting}
```

Código 5.2: ejemplo código C en color

```
1  #include <stdio.h>
2  // Comentario
3  int main(int argc, char* argv[]) {
4  puts("Hola mundo!");
5  }
```

Por supuesto, puedes mejorar esta presentación utilizando más modificadores. En la sección ?? se indican algunos detalles.

Otro ejemplo, ahora para mostrar código PHP, sería escribir en tu fichero \LaTeX lo siguiente:

```
\begin{lstlisting}[style=PHP, caption={ejemplo código PHP},label=PHP_code]
/*
Ejemplo de código en PHP para escribir tu primer programa en este lenguaje
Copia este código en tu ordenador y ejecútalo
*/
<html>
  <head>
    <title>Prueba de PHP</title>
  </head>
  <body>
    <?php echo '<p>Hola Mundo</p>'; ?> //esto lo escribe TODO el mundo
  </body>
</html>
\end{lstlisting}
```

y el resultado es el siguiente:

Código 5.3: ejemplo código PHP

```
100 /*
101 Ejemplo de código en PHP para escribir tu primer programa en este lenguaje. Copia este código en tu ↵
    ↵ ordenador y ejecútalo
102 */
103 <html>
104 <head>
105   <title>Prueba de PHP</title>
106 </head>
107 <body>
108   <?php echo '<p>Hola Mundo</p>'; ?> //esto lo escribe TODO el mundo
109 </body>
110 </html>
```

O también en color:

Código 5.4: ejemplo código PHP

```
1 /*
2 Ejemplo de código en PHP para escribir tu primer programa en este lenguaje. Copia este código en tu ↵
    ↵ ordenador y ejecútalo
3 */
4 <html>
5 <head>
6   <title>Prueba de PHP</title>
```

```

7 </head>
8 <body>
9   <?php echo '<p>Hola Mundo</p>'; ?> //esto lo escribe TODO el mundo
10 </body>
11 </html>

```

Observa cómo L^AT_EX ha puesto los comentarios en gris y ajustado el código para que se muestre más claro.

A continuación se muestran otros ejemplos:

Código 5.5: ejemplo código Matlab en color

```

1 %% Code sections are highlighted.
2 % System command are supported...
3 !touch testFile.txt
4 A = [1, 2, 3;... %... as is line continuation.
5     4, 5, 6];
6 fid = fopen('testFile.text', 'w');
7 for k=1:10
8     fprintf(fid, '%6.2f \n', k)
9 end
10 x=1; %% this is just a comment, not the start of a section
11 % Context-sensitive keywords get highlighted correctly...
12 p = properties(person); %(here, properties is a function)
13 x = linspace(0,1,101);
14 y = x(end:-1:1);
15 % ... even in nonsensical code.
16 ]end()()(((end while { end )end })))end (end
17 %{
18     block comments are supported
19 %} even
20 runaway block comments are

```

Código 5.6: ejemplo código Matlab en blanco y negro

```

1 %% Code sections are highlighted.
2 % System command are supported...
3 !touch testFile.txt
4 A = [1, 2, 3;... %... as is line continuation.
5     4, 5, 6];
6 fid = fopen('testFile.text', 'w');
7 for k=1:10
8     fprintf(fid, '%6.2f \n', k)
9 end
10 x=1; %% this is just a comment, not the start of a section
11 % Context-sensitive keywords get highlighted correctly...
12 p = properties(person); %(here, properties is a function)
13 x = linspace(0,1,101);
14 y = x(end:-1:1);
15 % ... even in nonsensical code.
16 ]end()()(((end while { end )end })))end (end
17 %{
18     block comments are supported
19 %} even
20 runaway block comments are

```

Código 5.7: ejemplo código Python en color

```

1 class Example (object):
2     def __init__ (self, account, password):
3         """e.g. account = 'bob@example.com/test'
4             password = 'bigbob'
5         """
6
7         reg = telepathy.client.ManagerRegistry()
8         reg.LoadManagers()
9
10        # get the gabble Connection Manager
11        self.cm = cm = reg.GetManager('gabble')
12
13        # get the parameters required to make a Jabber connection
14        # begin ex.basics.dbus.language-bindings.python.methods.call
15        cm[CONNECTION_MANAGER].RequestConnection('jabber',
16        {
17            'account': account,
18            'password': password,
19        },
20        reply_handler = self.request_connection_cb,
21        error_handler = self.error_cb)
22        # end ex.basics.dbus.language-bindings.python.methods.call

```

Código 5.8: ejemplo código Python en blanco y negro

```

1 class Example (object):
2     def __init__ (self, account, password):
3         """e.g. account = 'bob@example.com/test'
4             password = 'bigbob'
5         """
6
7         reg = telepathy.client.ManagerRegistry()
8         reg.LoadManagers()
9
10        # get the gabble Connection Manager
11        self.cm = cm = reg.GetManager('gabble')
12
13        # get the parameters required to make a Jabber connection
14        # begin ex.basics.dbus.language-bindings.python.methods.call
15        cm[CONNECTION_MANAGER].RequestConnection('jabber',
16        {
17            'account': account,
18            'password': password,
19        },
20        reply_handler = self.request_connection_cb,
21        error_handler = self.error_cb)
22        # end ex.basics.dbus.language-bindings.python.methods.call

```

5.2 Usos y personalización

El texto que acompaña al código puedes incluirlo o no, también puedes decidir si el texto va numerado o no. A continuación se muestra como:

```

% Con esta línea el código no tendrá título
\begin{lstlisting}[style=Python]
micodigo
\end{lstlisting}

```



```
1 micodigo
```

```
% Con esta línea el código tendrá el título abajo
\begin{lstlisting}[style=Python, caption={Ejemplo de título abajo},captionpos=b]
micodigo
\end{lstlisting}
```

```
1 micodigo
```

Código 5.9: Ejemplo de título abajo

```
% Con esta línea el código tendrá título no numerado
\begin{lstlisting}[style=Python, title={Ejemplo de título no numerado}]
micodigo
\end{lstlisting}
```

Ejemplo de título no numerado

```
1 micodigo
```

```
% Con esta línea el código no tendrá las líneas numeradas
\begin{lstlisting}[style=Python,numbers=none, title={Ejemplo de código sin número de líneas}]
micodigo
sin
número
de
líneas
\end{lstlisting}
```

Ejemplo de código sin número de líneas

```
micodigo
sin
número
de
líneas
```

5.3 Importar archivos fuente

Existe la posibilidad de importar un archivo de código en lugar de copiar su contenido y pegarlo en \LaTeX .

Para realizarlo debes escribir:

```
\lstinputlisting[style=C++-color,caption={Archivo C++ importado}]{archivos/ejemplos/holamundo.cpp}
```

Y se importará con el formato establecido entre los '[']':

Código 5.10: Archivo C++ importado

```

1 #include <stdio.h>
2 int main()
3 {
4     // printf() displays the string inside quotation
5     printf("Hello, World!");
6     return 0;
7 }

```

A continuación se muestran otros ejemplos

\lstinputlisting[style=Python-color,caption={Archivo Py importado},label=importado_py]{archivos/↩
↩ ejemplos/holamundo.py}

Código 5.11: Archivo Py importado

```

1 #-----↩
2 # helloworld.py
3 #-----↩
4
5 import stdio
6
7 # Write 'Hello, World' to standard output.
8 stdio.writeln('Hello, World')
9
10 #-----↩
11
12 # python helloworld.py
13 # Hello, World

```

\lstinputlisting[style=Matlab-color,caption={Archivo Matlab importado},label=importado_m]{archivos/↩
↩ ejemplos/holamundo.m}

Código 5.12: Archivo Matlab importado

```

1 function y = hello_world %#codegen
2
3 y = 'Hello World!';
4
5 end
6 % Copyright 2010 The MathWorks, Inc.

```

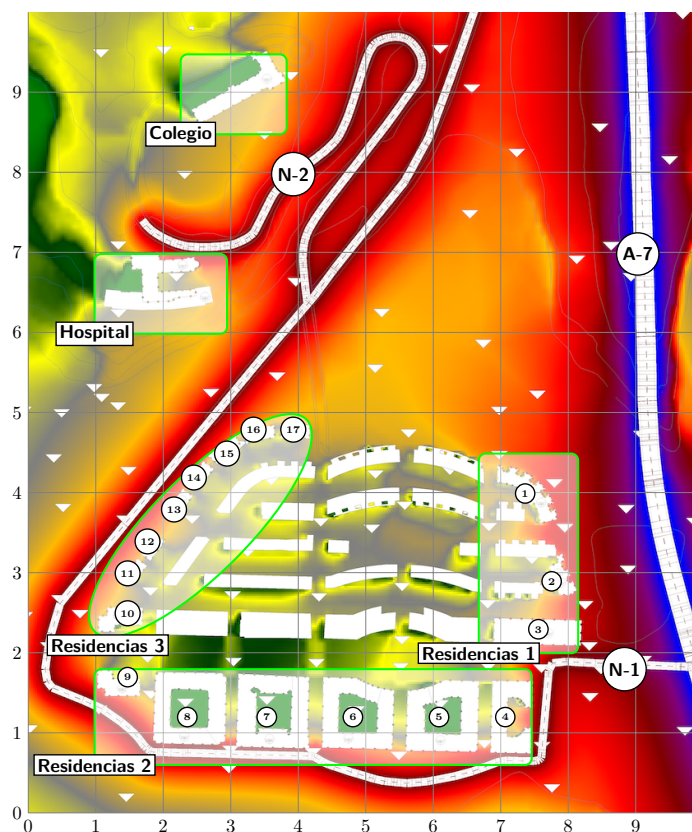
6 Resultados (Con ejemplos de gráficos)

6.1 Diagramas

Gracias al paquete *Tikz* se pueden incluir multitud de medios gráficos, diagramas, capas sobre imágenes, etc. Existen múltiples formas de realizarlo, para ello es recomendable consultar la guía de iniciación disponible aquí: <http://cremeronline.com/LaTeX/minimaltikz.pdf> y también el manual completo disponible aquí: <http://osl.ugr.es/CTAN/graphics/pgf/base/doc/pgfmanual.pdf>.

A continuación se muestran algunos ejemplos. Revisa el archivo .tex para ver cómo se utilizan.

Imagen a la que se le ha añadido cuadros y texto desde latex:



En muchas ocasiones es necesario realizar un diagrama de bloques, más abajo se muestra

un ejemplo de ello. En la red hay multitud de ejemplos que pueden ser fácilmente modificables para un fin concreto, como por ejemplo en esta web: <http://www.texample.net/tikz/examples/tag/block-diagrams/>.

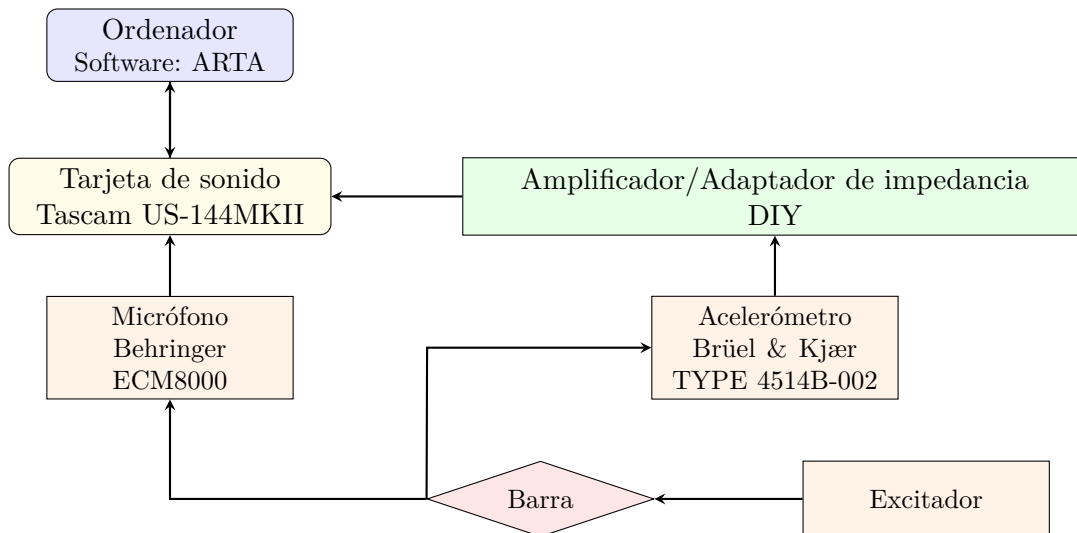


Figura 6.1: Diagrama realizado en latex con Tikz.

6.2 Gráficas

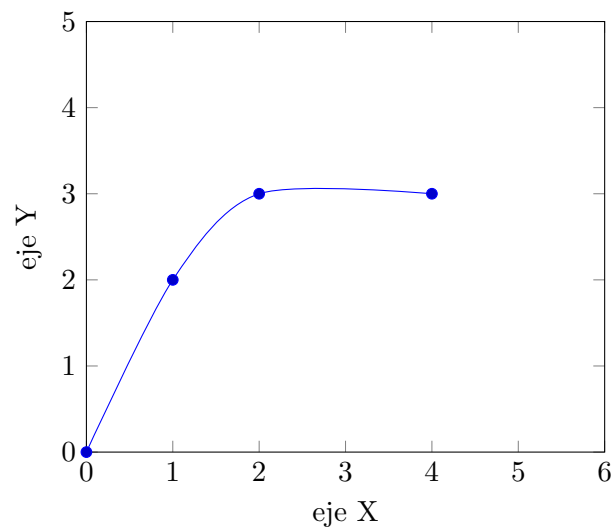
Existen múltiples formas de generar gráficas para latex. Hay disponibles herramientas como GeoGebra que dispone de la utilidad para exportar los gráficos en formato Tkiz. También funciones para Matlab que genera las gráficas que muestra habitualmente pero en código para Tkiz.

6.2.1 Línea

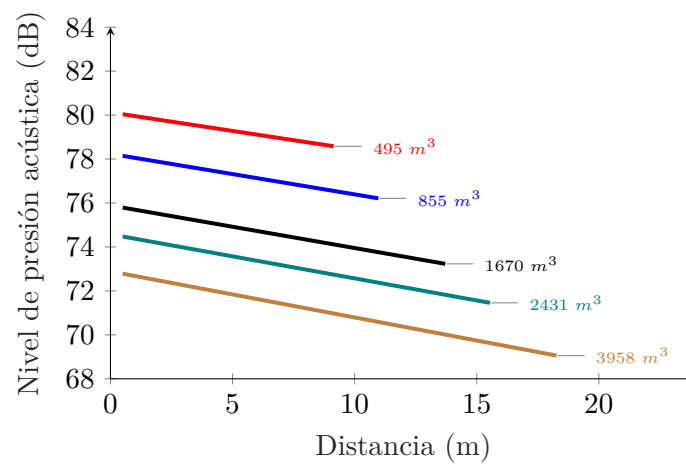
La forma más simple, aunque no sencilla cuando abarca muchos datos es la siguiente:

```
\begin{figure}[ht]
\centering
\begin{tikzpicture}
\begin{axis}
[ymin=0,ymax=5, % Límites del eje y
xmin=0,xmax=6, % Límites del eje x
ylabel= eje Y, % Nombre del eje y
xlabel= eje X] % Nombre del eje x
\addplot+[smooth] coordinates % Une los puntos curva suavizada
{(0,0) (1,2) (2,3 (4,3))}; % Puntos de la gráfica
\end{axis}
\end{tikzpicture}
\caption{Gráfica sencilla.}
\end{figure}
```

El resultado es el siguiente:

**Figura 6.2:** Gráfica sencilla.

Otro ejemplo, en este caso las líneas están calculadas directamente en LaTeX y después cada una tiene una anotación (el código se encuentra en el archivo `archivos/ejemplos/perjudicialesopticaentro.tex`).

**Figura 6.3:** OP/S003

6.2.2 Barras

Otro ejemplo es la gráfica de barras:

```
\begin{figure}[ht]
\centering
\begin{tikzpicture}
\begin{axis}[
ybar=12pt,
ymin=0,ymax=150,
xtick=data,
```

```

enlarge x limits={abs=2cm},
symbolic x coords={rubio, moreno},
bar width = 20pt,
ylabel= número,
xlabel= color de pelo,
ytick align=outside,
ytick pos=left,
major x tick style = transparent,
legend style={at={(0.04,0.96)},anchor=north west, font=\footnotesize, legend cell align=left,},
]
\addplot[ybar,fill=blue, area legend] coordinates {
(rubio,20)
(moreno,100)};
\addplot[ybar,fill=purple, area legend] coordinates {
(rubio,110)
(moreno,105)};
\legend{Chicos, Chicas}
\end{axis}
\end{tikzpicture}
\caption{Gráfica barras.}
\end{figure}

```

El resultado es el siguiente:

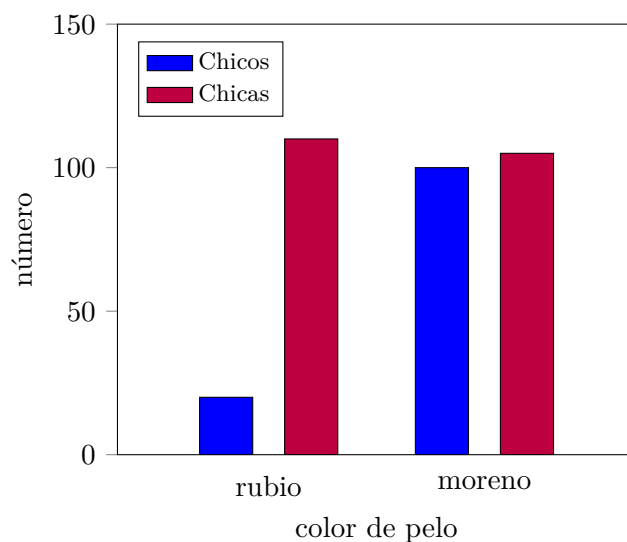


Figura 6.4: Gráfica barras.

6.2.3 Polar

Un ejemplo de gráfica polar semicircular (ver archivo `archivos/ejemplos/polarnorm.tex`):

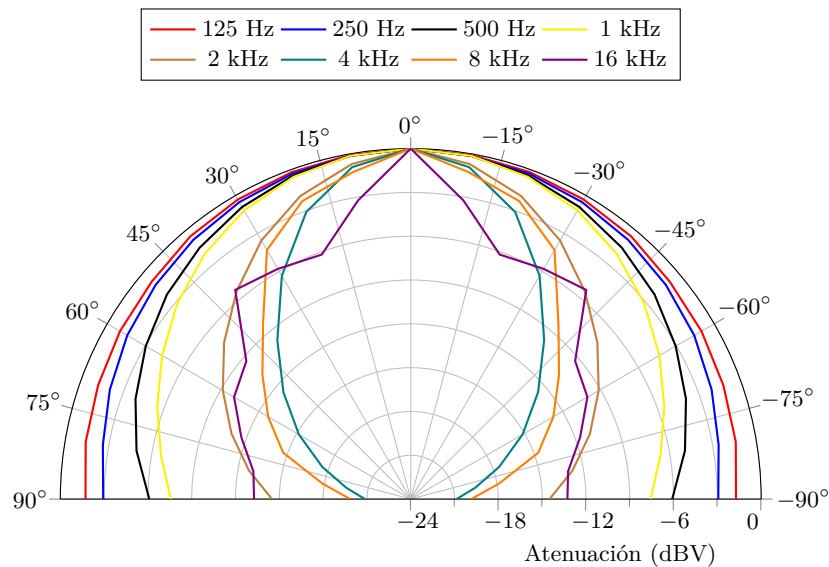


Figura 6.5: Directividad normalizada del altavoz (0 dBV en el eje).

6.3 Importados de MATLAB

Gracias a la herramienta *matlab2tikz* (<https://es.mathworks.com/matlabcentral/fileexchange/22022-matlab2tikz-matlab2tikz>) se pueden exportar las gráficas de cualquier tipo de Matlab a latex. Después de incluir los archivos de *matlab2tikz* se debe escribir una llamada después de crear la figura tal que:

Código 6.1: Ejemplo de llamada a *matlab2tikz*

```
1 fig = plot(x,y);
2 matlab2tikz('figurehandle',fig,'NombreArchivo.tex','height','5cm','width','13.5cm','strict',true,'↔
  ↳ showHiddenStrings',true,'showInfo',false)
```

Y para utilizar el archivo generado por la función en este documento:

```
\begin{figure}[ht]
\centering
{\scalefont{0.8}\input{archivos/ejemplos/ParedFina} }
\caption{Ejemplo de gráfica obtenida con matlab2tikz.}
\end{figure}
```

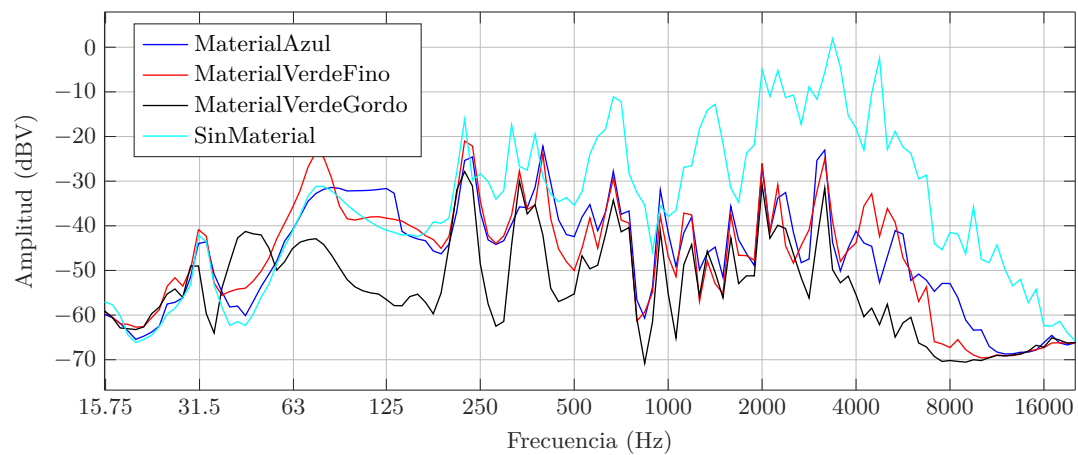


Figura 6.6: Ejemplo de gráfica obtenida con `matlab2tikz`.

Ejemplo de una gráfica 3D generada en Matlab y exportada por `matlab2tikz`:

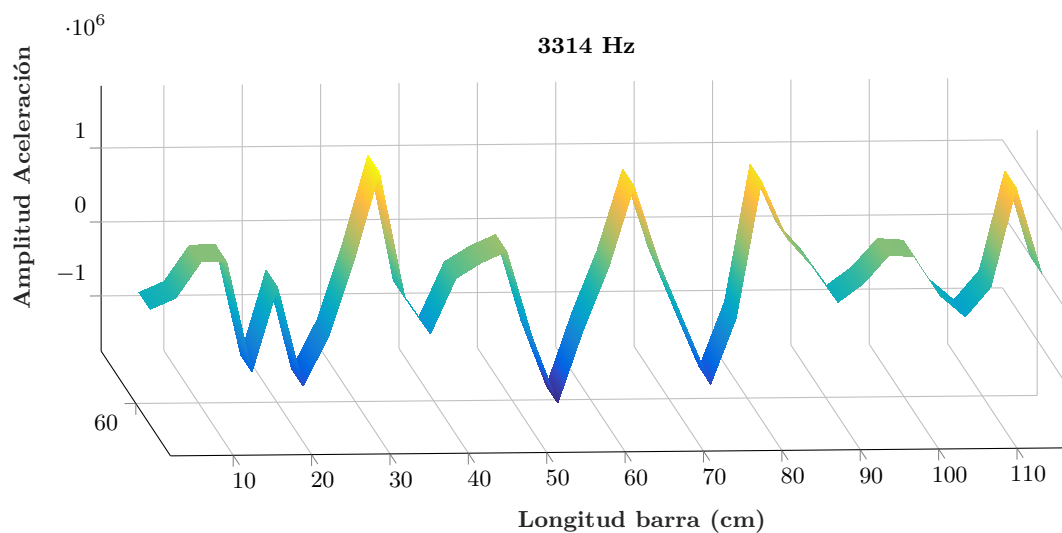


Figura 6.7: Amplitud de la aceleración en el modo número 8.

6.4 Ejemplo avanzado

El potencial del paquete *Tikz* es muy alto, se pueden realizar muchísimas cosas. En la red se facilitan muchos ejemplos para poder ver el funcionamiento y aprender. Existen hilos donde la gente publica sus mejores diseños de *Tikz* como en <https://tex.stackexchange.com/questions/158668/nice-scientific-pictures-show-off> o páginas donde facilitan muchas plantillas como <http://www.texample.net/tikz/examples/all/>.

Un ejemplo de lo que se puede llegar a conseguir es el siguiente:

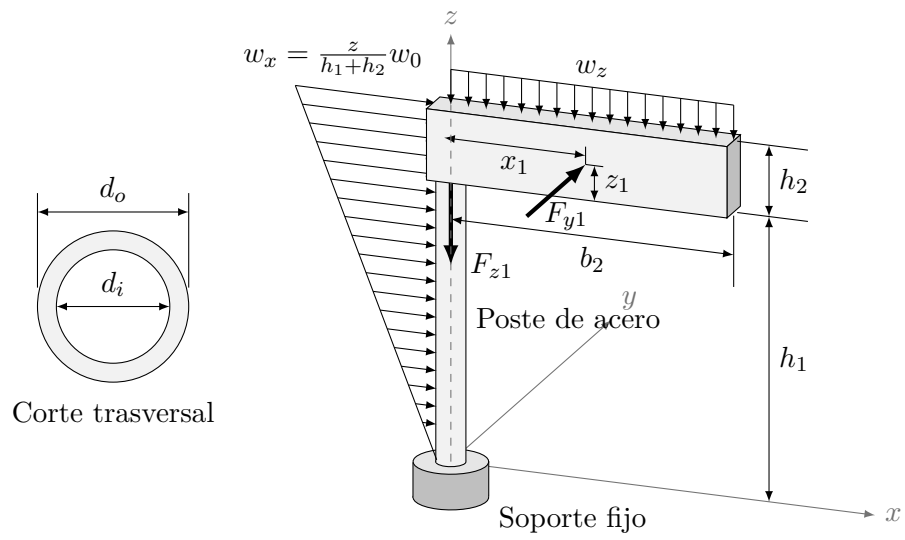


Figura 6.8: Señal realizada con Tikz, sin imágenes.

Bibliografía

- AENOR. (1997). *norma une 50136:1997*. Descargado de http://docubib.uc3m.es/CURSOS/Documentos_cientificos/Normas%20y%20directrices/UNE_50136=ISO%207144.pdf
- Akyildiz, I. F., Pompili, D., y Melodia, T. (2005). Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3), 257–279.
- Alexander, G. E., y Crutcher, M. D. (1990). Functional architecture of basal ganglia circuits: neural substrates of parallel processing. *Trends in neurosciences*, 13(7), 266–271.
- Anderson, J. R. (2013). *The architecture of cognition*. Psychology Press.
- Atske, S. (2025, 12). *Teens and Internet, Device Access Fact sheet*. Descargado de <https://www.pewresearch.org/internet/fact-sheet/teens-and-internet-device-access-fact-sheet/>
- Auld, B. A. (1973). *Acoustic fields and waves in solids*. .
- Baddeley, A. D. (1966). Short-term memory for word sequences as a function of acoustic, semantic and formal similarity. *The Quarterly Journal of Experimental Psychology*, 18(4), 362–365.
- Barkan, A., Merlino, R. L., y D’angelo, N. (1995). Laboratory observation of the dust-acoustic wave mode. *Physics of Plasmas*, 2(10), 3563–3565.
- Bass, L. (2007). *Software architecture in practice*. Pearson Education India.
- Biot, M. A. (1962). Mechanics of deformation and acoustic propagation in porous media. *Journal of applied physics*, 33(4), 1482–1498.
- Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., y Weiss, W. (1998). *An architecture for differentiated services* (Inf. Téc.).
- BOE. (2012, marzo). *Resolución de 7 de marzo de 2012, de la universidad de alicante, por la que se publica el plan de estudios de graduado en ingeniería multimedia*. BOE, 22 marzo de 2012. Descargado de <http://www.boe.es/boe/dias/2012/03/22/pdfs/BOE-A-2012-4008.pdf>
- Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing*, 27(2), 113–120.
- Burdic, W. S. (1991). *Underwater acoustic system analysis*. Prentice Hall.

- Busby, E. (2019, 7). *Poorest children three times more likely to miss out on extra-curricular activities, study finds* / *The Independent*. Descargado de <https://www.independent.co.uk/news/education/education-news/poverty-children-school-extracurricular-music-sport-social-mobility-family-a9010936.html>
- Buschmann, F., Henney, K., y Schimdt, D. (2007). *Pattern-oriented software architecture: on patterns and pattern language* (Vol. 5). John Wiley & Sons.
- Cerjan, C., Kosloff, D., Kosloff, R., y Reshef, M. (1985). A nonreflecting boundary condition for discrete acoustic and elastic wave equations. *Geophysics*, 50(4), 705–708.
- Colton, D., y Kress, R. (2012). *Inverse acoustic and electromagnetic scattering theory* (Vol. 93). Springer Science & Business Media.
- Conrad, R. (1964). Acoustic confusions in immediate memory. *British journal of Psychology*, 55(1), 75–84.
- Cox, T. J., D'antonio, P., y Schroeder, M. (2005). Acoustic absorbers and diffusers, theory, design and application. *The Journal of the Acoustical Society of America*, 117(3), 988–988.
- Díaz Caballero, J., Canino Ramos, C., y Morejon, G. (2013, 02). Heuristics of the platonic polyhedra for the high restrictions reality research. *Revista Cubana de Ingeniería*, 4. doi: 10.1234/rci.v4i1.135
- Eisenstein, D. J., Zehavi, I., Hogg, D. W., Scoccimarro, R., Blanton, M. R., Nichol, R. C., ... others (2005). Detection of the baryon acoustic peak in the large-scale correlation function of sdss luminous red galaxies. *The Astrophysical Journal*, 633(2), 560.
- Fahlman, S. E., y Lebiere, C. (1990). The cascade-correlation learning architecture. En *Advances in neural information processing systems* (pp. 524–532).
- Fant, G. (1971). *Acoustic theory of speech production: with calculations based on x-ray studies of russian articulations* (Vol. 2). Walter de Gruyter.
- Fodor, J. A., y Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1), 3–71.
- Foschini, G. J. (1996). Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. *Bell labs technical journal*, 1(2), 41–59.
- Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison-Wesley Longman Publishing Co., Inc.
- Griffin, D. R. (1958). Listening in the dark: the acoustic orientation of bats and men.
- Heinz, M., Carsten, y Hoffmann, J. (2014, March). *The listings package, march 2014*. <http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf>. Descargado 12/12/2014, de <http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf>
-

- Heinzelman, W. B., Chandrakasan, A. P., y Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, 1(4), 660–670.
- Hennessy, J. L., y Patterson, D. A. (2011). *Computer architecture: a quantitative approach*. Elsevier.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., ... others (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6), 82–97.
- Ho, L. L. K., Li, W. H. C., Cheung, A. T., Xia, W., Ho, K. Y., y Chung, J. O. K. (2020, 9). Low-income parents' perceptions of the importance of a musical training programme for their children: a qualitative study. *BMC Public Health*, 20(1), 1454. Descargado de <https://pmc.ncbi.nlm.nih.gov/articles/PMC7519511/> doi: 10.1186/s12889-020-09568-7
- Hubel, D. H., y Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1), 106–154.
- Hubel, D. H., y Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1), 215–243.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. En *Proceedings of the 22nd acm international conference on multimedia* (pp. 675–678).
- Kemp, D. T. (1978). Stimulated acoustic emissions from within the human auditory system. *The Journal of the Acoustical Society of America*, 64(5), 1386–1391.
- Kushwaha, M. S., Halevi, P., Dobrzynski, L., y Djafari-Rouhani, B. (1993). Acoustic band structure of periodic elastic composites. *Physical review letters*, 71(13), 2022.
- Leighton, T. (2012). *The acoustic bubble*. Academic press.
- McCarthy, B. (2012). *Sound systems: design and optimization: modern techniques and tools for sound system design and alignment*. CRC Press.
- Mitola, J. (2000). Cognitive radio—an integrated agent architecture for software defined radio.
- Muthupillai, R., Lomas, D., Rossman, P., Greenleaf, J. F., Manduca, A., y Ehman, R. L. (1995). Magnetic resonance elastography by direct visualization of propagating acoustic strain waves. *science*, 269(5232), 1854–1857.
- Orzel, H. J. (2010). *Undergraduate music student stress and burnout*. Descargado de https://scholarworks.sjsu.edu/etd_theses/3887?utm_source=scholarworks.sjsu.edu%2Fetd_theses%2F3887&utm_medium=PDF&utm_campaign=PDFCoverPages
- Rao, N., Shukla, P., y Yu, M. Y. (1990). Dust-acoustic waves in dusty plasmas. *Planetary and space science*, 38(4), 543–546.

- Real Decreto 1577/2006. (2006). *Real decreto 1577/2006, de 22 de diciembre, por el que se fijan los aspectos básicos del currículo de las enseñanzas profesionales de música*. Descargado de <https://www.boe.es/buscar/pdf/2007/BOE-A-2007-1221-consolidado.pdf>
- Referencia : semitonos cromáticos y diatónicos. (s.f.). Descargado de <https://www.teoria.com/es/referencia/c/crom-diat.php>
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., y Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. En *Proceedings of the 1994 acm conference on computer supported cooperative work* (pp. 175–186).
- Rosenbaum, J. (1988). *Bulk acoustic wave theory and devices*. Artech House on Demand.
- Shaw, M., y Garlan, D. (1996). *Software architecture: perspectives on an emerging discipline* (Vol. 1). Prentice Hall Englewood Cliffs.
- Simon, H. A. (1991). The architecture of complexity. En *Facets of systems science* (pp. 457–476). Springer.
- Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research policy*, 24(3), 419–440.
- Villa, D. (2008, 4). *Latex: Listados de código cómodos y resultones con listings*. <http://crysol.org/es/node/909>. Descargado 12/12/2014, de <http://crysol.org/es/node/909>
- Wolniansky, P. W., Foschini, G. J., Golden, G., y Valenzuela, R. A. (1998). V-blast: An architecture for realizing very high data rates over the rich-scattering wireless channel. En *Signals, systems, and electronics, 1998. issse 98. 1998 ursi international symposium on* (pp. 295–300).
-