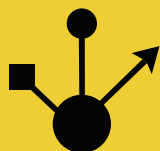




Escuela  
Politécnica  
Superior

# Título del Trabajo Fin de Grado/Máster



Grado en Ingeniería Multimedia

## Trabajo Fin de Grado

Autor:

Alexei Jilinskiy

Tutor:

Fidel Áznar Gregori

Junio 2020



Universitat d'Alacant  
Universidad de Alicante



# Título del Trabajo Fin de Grado/Máster

---

Subtítulo del proyecto

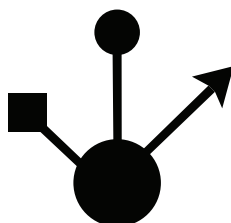
## **Autor**

Alexei Jilinskiy

## **Tutor**

Fidel Áznar Gregori

*Departamento de Ciencia de la Computación e Inteligencia Artificial*



Grado en Ingeniería Multimedia



Escuela  
Politécnica  
Superior



Universitat d'Alacant  
Universidad de Alicante

ALICANTE, Junio 2020



# Preámbulo

Poner aquí un texto breve que debe incluir entre otras:

“las razones que han llevado a la realización del estudio, el tema, la finalidad y el alcance y también los agradecimientos por las ayudas, por ejemplo apoyo económico (becas y subvenciones) y las consultas y discusiones con los tutores y colegas de trabajo. (?)”



# Agradecimientos

Dedico este trabajo a todas aquellas personas que han hecho de mi vida una más curiosa y dicha.





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



*Si consigo ver más lejos  
es porque he conseguido auparme  
a hombros de gigantes*

Isaac Newton.



# Índice general

<b>1. Introducción (Con ejemplos de contenido)</b>	<b>1</b>
1.0.1. Self-Organization and Emergence . . . . .	1
1.1. ¡Importante!, leer primero . . . . .	1
1.2. Estructura de un TFG . . . . .	1
1.3. Apartados dentro de los capítulos . . . . .	3
1.4. Esto es una sección . . . . .	3
1.4.1. Esto es una subsección . . . . .	3
1.4.1.1. Esto es una subsubsección . . . . .	3
1.4.1.1.1. Esto es un paragraph . . . . .	3
1.5. Citar bibliografía . . . . .	3
1.6. Notas a pie de página . . . . .	4
1.7. Estilos de texto . . . . .	4
1.8. Acrónimos . . . . .	5
1.9. Tareas por hacer . . . . .	5
<b>2. State of the Art</b>	<b>7</b>
2.1. Swarm Intelligence . . . . .	7
2.1.1. Definition . . . . .	7
2.1.2. Types of Swarm Intelligence Models . . . . .	10
2.1.3. Swarm Robotics . . . . .	10
2.2. Agent simulation . . . . .	10
2.3. Fokker-Planck . . . . .	10
2.4. Parallel Programming and Simulation . . . . .	10
<b>3. Objetivos (Con ejemplos de tablas)</b>	<b>11</b>
3.1. Tablas . . . . .	11
3.2. Otros diseños de tablas . . . . .	13
<b>4. Methodology</b>	<b>15</b>
4.1. Materials . . . . .	15
4.1.1. Software . . . . .	15
4.1.2. Hardware . . . . .	15
4.2. Interpretation of the Fokker-Planck equation . . . . .	15
4.3. Inserción de figuras . . . . .	15
<b>5. Desarrollo (Con ejemplos de código)</b>	<b>19</b>
5.1. Inserción de código . . . . .	19
5.2. Usos y personalización . . . . .	22
5.3. Importar archivos fuente . . . . .	23

<b>6. Resultados (Con ejemplos de gráficos)</b>	<b>25</b>
6.1. Diagramas . . . . .	25
6.2. Gráficas . . . . .	26
6.2.1. Línea . . . . .	26
6.2.2. Barras . . . . .	27
6.2.3. Polar . . . . .	28
6.3. Importados de MATLAB . . . . .	29
6.4. Ejemplo avanzado . . . . .	30
<b>7. Conclusiones (Con ejemplos de matemáticas)</b>	<b>33</b>
7.1. Matemáticas . . . . .	33
<b>A. Anexo I</b>	<b>37</b>
<b>B. Páginas horizontales</b>	<b>39</b>
<b>C. Importar PDF</b>	<b>43</b>

---

# Índice de figuras

4.1. Ejemplo de subfiguras . . . . .	16
4.2. Ejemplo de subfiguras vertical . . . . .	17
6.1. Diagrama realizado en latex con Tikz. . . . .	26
6.2. Gráfica sencilla. . . . .	27
6.3. OP/S003 . . . . .	27
6.4. Gráfica barras. . . . .	28
6.5. Directividad normalizada del altavoz (0 dBV en el eje). . . . .	29
6.6. Ejemplo de gráfica obtenida con matlab2tikz. . . . .	30
6.7. Amplitud de la aceleración en el modo número 8. . . . .	30
6.8. Señal realizada con Tikz, sin imágenes. . . . .	31





# Índice de tablas

3.1. Ejemplo de tabla. . . . .	11
3.2. Parámetros optativos de los entornos flotantes . . . . .	12
3.3. Parámetros de los altavoces elegidos de la marca Beyma®. . . . .	13
3.4. Ejemplo 2 . . . . .	14
4.1. Esta es una tabla con múltiples imágenes. Útil cuando se deben mostrar varias juntas. . . . .	16



# Índice de Códigos

5.1. ejemplo código C . . . . .	19
5.2. ejemplo código C en color . . . . .	20
5.3. ejemplo código PHP . . . . .	20
5.4. ejemplo código PHP . . . . .	20
5.5. ejemplo código Matlab en color . . . . .	21
5.6. ejemplo código Matlab en blanco y negro . . . . .	21
5.7. ejemplo código Python en color . . . . .	22
5.8. ejemplo código Python en blanco y negro . . . . .	22
5.9. Ejemplo de título abajo . . . . .	23
5.10. Archivo C++ importado . . . . .	24
5.11. Archivo Py importado . . . . .	24
5.12. Archivo Matlab importado . . . . .	24
6.1. Ejemplo de llamada a matlab2tikz . . . . .	29



# 1. Introducción (Con ejemplos de contenido)

## 1.0.1. Self-Organization and Emergence

Explicar aquí el origen de la swarm intelligence, los trabajos de heiko y tal que son pilar para esta investigación. Cómo no hay modelos matemáticos pero que diferentes estudios a lo largo del tiempo han intentado estudiar la matemática del caos y aún nada.

Antes de comenzar la lectura de este documento debo agradecer el trabajo realizado por Pedro Pernías Peco en su plantilla de “tfg” que se puede ver en <https://github.com/lcg51/tfg>. Gracias a esa plantilla me he lanzado a crear mi versión. Algunos contenidos aquí mostrados han sido extraídos de la plantilla de Pedro.

Esta plantilla se ha diseñado de 0 y por ello no utiliza la misma estructura que la plantilla de Pedro. Pero la estructura de contenido para un TFG/TFM es la misma y a continuación se muestran las diferentes partes que debe tener un TFG/TFM redactado por Pedro.

## 1.1. ¡Importante!, leer primero

Este texto está escrito pensando en orientar a los alumnos que usarán  $\text{\LaTeX}$  para escribir su Trabajo Final de Grado (TFG) y Trabajo Final de Máster (TFM).

Contiene información útil para aquellos que no tengan experiencia previa en  $\text{\LaTeX}$  así como algunos datos acerca de cómo escribir mejor su TFG. A continuación, se ofrece una copia de la información que hay en el libro de estilo para la realización de los TFG de la EPS de la Universidad de Alicante.

En los capítulos siguientes encontrarás ejemplos de muchas de las cosas que se pueden realizar con  $\text{\LaTeX}$ . Con un poco de paciencia, estudia cómo se hacen estas cosas y luego aplícalas en tus documentos.

## 1.2. Estructura de un TFG

En caso de que el TFG/TFM tenga como finalidad la elaboración de un proyecto o un informe científico o técnico, deberá ajustarse a lo dispuesto en las normas UNE 157001:2002 y UNE 50135:1996 respectivamente.

Si el TFG/TFM tiene por finalidad la elaboración de un trabajo monográfico, el documento presentado deberá constar de las siguientes partes, teniendo como base la norma UNE 50136:1997.

**Preámbulo:** se describirán brevemente la motivación que ha originado la realización del TFG/TFM, así como una breve descripción de los objetivos generales que se quieren alcanzar con el trabajo presentado.

**Agradecimientos:** se podrán añadir las hojas necesarias para realizar los agradecimientos, a veces obligatorios, a las entidades y organismos colaboradores.

**Dedicatoria:** se podrá añadir una única hoja con dedicatorias, su alineación será derecha.

**Citas:** (frases célebres) se podrá añadir una única hoja con citas, su alineación será derecha.

**Índices:** cada índice debe comenzar en una nueva página, se incluirán los índices que se estimen necesarios (conforme UNE 50111:1989), en este orden:

**Índice de contenidos:** (obligatorio siempre) se incluirá un índice de las secciones de las que se componga el documento, la numeración de las divisiones y subdivisiones utilizarán cifras arábigas (según UNE 50132:1994) y harán mención a la página del documento donde se ubiquen.

**Índice de figuras:** si el documento incluye figuras se podrá incluir también un índice con su relación, indicando la página donde se ubiquen.

**Índice de tablas:** en caso de existir en el texto, ídem que el anterior.

**Índice de abreviaturas, siglas, símbolos, etc.:** en caso de ser necesarios se podrán incluir cada uno de ellos.

**Cuerpo del documento:** en el contenido del documento se da flexibilidad para su organización y se puede estructurar en las secciones que se considere. En todo caso obligatoriamente se deberá, al menos, incluir los siguientes contenidos:

**Introducción:** donde se hará énfasis a la importancia de la temática, su vigencia y actualidad; se planteará el problema a investigar, así como el propósito o finalidad de la investigación.

**Marco teórico o Estado del arte:** se hará mención a los elementos conceptuales que sirven de base para la investigación, estudios previos relacionados con el problema planteado, etc.

**Objetivos:** se establecerán el objetivo general y los específicos.

**Metodología:** se indicarán el tipo o tipos de investigación, las técnicas y los procedimientos que serán utilizados para llevarla a cabo; se identificarán la población y el tamaño de la muestra así como las técnicas e instrumentos de recolección de datos.

**Resultados:** incluirá los resultados de la investigación o trabajo, así como el análisis y la discusión de los mismos.

**Conclusiones:** obligatoriamente se incluirá una sección de conclusiones donde se realizará un resumen de los objetivos conseguidos así como de los resultados obtenidos si proceden.

**Bibliografía y referencias:** se incluirá también la relación de obras y materiales consultados y empleados en la elaboración de la memoria del TFG/TFM. La bibliografía y las referencias serán indexadas en orden alfabético (sistema nombre y fecha) o se numerará correlativamente según aparezca (sistema numérico). Se empleará la familia 1 como tipo de letra. Podrá utilizarse cualquier sistema bibliográfico normalizado predominante en la rama de conocimiento, estableciéndose como prioritarios el sistema ISO 690, sistema

---

American Psychological Association (APA) o Harvard (no necesariamente en ese orden de preferencia). En esta plantilla Latex se propone usar el estilo APA indicándolo en la línea correspondiente como

```
\bibliographystyle{apacite}
```

**Anexos:** se podrán incluir los anexos que se consideren oportunos.

## 1.3. Apartados dentro de los capítulos

En L<sup>A</sup>T<sub>E</sub>X existen diferentes niveles de títulos para realizar secciones, subsecciones, etc. En esta web puedes ver más información al respecto [https://en.wikibooks.org/wiki/LaTeX/Document\\_Structure](https://en.wikibooks.org/wiki/LaTeX/Document_Structure)

Para ello se utilizan los siguientes comandos;

```
\section{Esto es una sección}
Y este el contenido de la sección.
\subsection{Esto es una subsección}
Y este el contenido de la subsección.
\subsubsection{Esto es una subsubsección}
Y este el contenido de la subsubsección.
\paragraph{Esto es un paragraph}
Y este el contenido del paragraph. Que siempre se inicia en la misma línea que el título del mismo.
```

Y se genera lo siguiente:

### 1.4. Esto es una sección

Y este el contenido de la sección.

#### 1.4.1. Esto es una subsección

Y este el contenido de la subsección.

##### 1.4.1.1. Esto es una subsubsección

Y este el contenido de la subsubsección.

**1.4.1.1.1. Esto es un paragraph** Y este el contenido del paragraph. Que siempre se inicia en la misma línea que el título del mismo.

## 1.5. Citar bibliografía

Para citar la bibliografía tal como se define en el sistema APA (en esta web se indica como debe aparecer en el texto la cita: <http://guides.libraries.psu.edu/apasquickguide/intext>) se debe realizar con alguno de los comandos mostrados a continuación:

---

Esto es una cita estándar: `\citet{Shaw1996}`, que también puedes mostrar con paréntesis así: `\citep{Shaw1996}`. También se puede realizar una cita indicando a qué parte te refieres `\citep[ver][Cap. 2]{Shaw1996}` o `\citep[Cap. 2]{Shaw1996}` o `\citep[ver]{}{Shaw1996}`.

También puedes mostrar todos los autores cuando hay más de 2 autores añadiendo un asterisco después del comando como: `\citet*{Akyildiz2005}`, sin el asterisco quedaría así: `\citet{Akyildiz2005}`.

O puedes citar dos o más fuentes al mismo tiempo: `\citep{Barkan1995,Leighton2012}`

Y  $\text{\LaTeX}$  genera lo siguiente:

Esto es una cita estándar: ?, que también puedes mostrar con paréntesis así: (?). También se puede realizar una cita indicando a qué parte te refieres (ver ?, Cap. 2) o (?, Cap. 2) o (ver ?).

También puedes mostrar todos los autores cuando hay más de 2 autores añadiendo un asterisco después del comando como: \*, sin el asterisco quedaría así: ?.

O puedes citar dos o más fuentes al mismo tiempo: (??)

## 1.6. Notas a pie de página

Para introducir notas a pie de página se debe escribir lo siguiente:

La plantilla necesita el motor XeLaTeX `\footnote{Para más información sobre XeLaTeX visita \url{https://es.sharelatex.com/learn/XeLaTeX}}` (el más recomendable actualmente), por lo que si el programa que utilizas compila la plantilla con el motor pdfLaTeX `\footnote{También puedes buscar más información en internet}` (el más habitual pero menos potente) debes cambiarlo por `\footnote{XeLaTeX en las opciones del programa. Si no sabes como hacerlo busca en el manual del programa o en google.}`

$\text{\LaTeX}$  genera lo siguiente (observa las notas a pie de página):

La plantilla necesita el motor XeLaTeX<sup>1</sup> (el más recomendable actualmente), por lo que si el programa que utilizas compila la plantilla con el motor pdfLaTeX<sup>2</sup> (el más habitual pero menos potente) debes cambiarlo por XeLaTeX en las opciones del programa. Si no sabes como hacerlo busca en el manual del programa o en google.

## 1.7. Estilos de texto

A continuación se muestran ejemplos de distintos estilos de texto:

- `\textit{Cursiva}` → *Cursiva*
- `\emph{Cursiva 2}` → *Cursiva 2*
- `\textbf{Negrita}` → **Negrita**
- `\texttt{Monoespacio}` → Monoespacio

<sup>1</sup>Para más información sobre XeLaTeX visita <https://es.sharelatex.com/learn/XeLaTeX>

<sup>2</sup>También puedes buscar más información en internet



- `\textsc{Mayúsculas capitales}` → MAYÚSCULAS CAPITALES
- `\uppercase{Todo mayúsculas}` → TODO MAYÚSCULAS

## 1.8. Acrónimos

Ahora vamos a ver cómo se ponen los acrónimos.

La norma dice que la primera vez que aparece un acrónimo debe ponerse su fórmula completa, es decir lo que significa, al lado del acrónimo. Después de ello, podemos usar sólo el acrónimo salvo cuando consideremos que debemos volver a usar la fórmula completa por alguna razón de legibilidad.

¿Cómo llevar la cuenta de cuándo es la primera vez que ponemos el acrónimo? si hacemos cambios en el doc es fácil que perdamos esa información así que lo mejor es que sea el propio L<sup>A</sup>T<sub>E</sub>X el que lleve esa cuenta. Para ello tenemos que hacer dos cosas:

**Primero:** creamos la entrada del acrónimo en el fichero `acronimos.tex`. Revisa los comentarios de su cabecera para saber cómo crear esa entrada. Básicamente lo que hacemos allí es poner la “fórmula corta” y la “fórmula larga” del acrónimo es decir, el propio acrónimo y su significado

**Segundo:** escribimos en el texto el acrónimo SIEMPRE diciendo que es un acrónimo y el tipo de fórmula que queremos usar. Por ejemplo, si siempre que queremos hacer referencia al IEEE escribimos

```
\gls{ieeee}
```

se consigue que la primera vez que aparezca el acrónimo ponga las fórmulas larga y corta y en las siguientes ocasiones sólo aparecerá la corta.

Aquí va un ejemplo:

Si escribimos:

El `\gls{ieeee}` es una institución muy importante en el mundo de la ingeniería. El `\gls{ieeee}` lleva marcando normas y protocolos desde hace mucho tiempo. Pero el `\gls{ieeee}` no está solo en esta tarea. Además del `\gls{ieeee}` hay muchas otras instituciones para ello.

Obtendremos:

El Institute of Electrical and Electronics Engineers (IEEE) es una institución muy importante en el mundo de la ingeniería. El IEEE lleva marcando normas y protocolos desde hace mucho tiempo. Pero el IEEE no está solo en esta tarea. Además del IEEE hay muchas otras instituciones para ello.

## 1.9. Tareas por hacer

En esta plantilla se ha incluido un paquete para incluir notas/comentarios en el texto para recordar partes que hay que revisar o terminar de desarrollar. El uso es sencillo, el manual para conocer todos los comandos se encuentra en <http://osl.ugr.es/CTAN/macros/latex/>

`contrib/todonotes/todonotes.pdf`, a continuación se muestran algunos ejemplos:

Para incluir un comentario sobre el texto:

Recomiendo utilizar programas LaTeX que permitan trabajar con sistema de archivos para poder editar el `\todo` conjunto de capítulos en la misma ventana. Este tipo de función lo tienen programas como `\todo` TexStudio, es multiplataforma. `\todo`{Incluir más ejemplos de programas}

LaTeX genera lo siguiente:

Recomiendo utilizar programas LaTeX que permitan trabajar con sistema de archivos para poder editar el conjunto de capítulos en la misma ventana. Este tipo de función lo tienen programas como TexStudio, es multiplataforma.

Incluir más ejemplos de programas

Para incluir un comentario sobre el texto pero dentro del texto:

Recomiendo utilizar programas LaTeX que permitan trabajar con sistema de archivos para poder editar el `\todo` conjunto de capítulos en la misma ventana. Este tipo de función lo tienen programas como `\todo` TexStudio, es multiplataforma. `\todo`[inline]{Incluir más ejemplos de programas}

LaTeX genera lo siguiente:

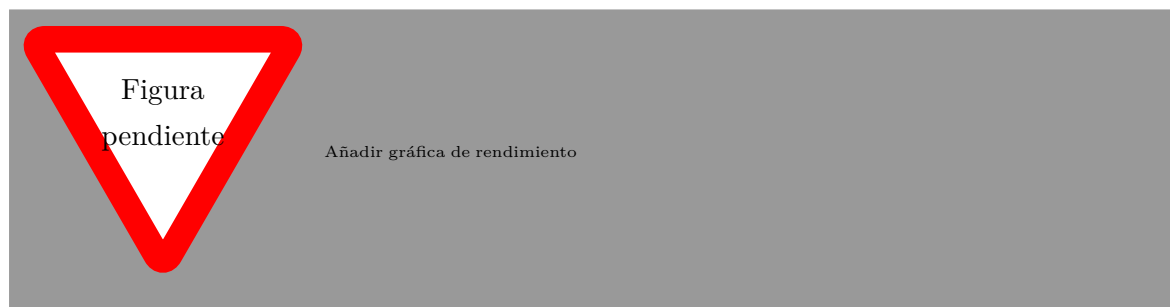
Recomiendo utilizar programas LaTeX que permitan trabajar con sistema de archivos para poder editar el conjunto de capítulos en la misma ventana. Este tipo de función lo tienen programas como TexStudio, es multiplataforma.

Incluir más ejemplos de programas

También se puede dejar indicado donde falta una imagen o figura, para incluirla más adelante del siguiente modo:

`\missingfigure`{Añadir gráfica de rendimiento}

LaTeX genera lo siguiente:



## 2. State of the Art

### 2.1. Swarm Intelligence

#### 2.1.1. Definition

Swarm Intelligence is fairly recent area of research that grabbed the attention of scientists of several different fields back in the 1980s. Gerardo Beni and Jing Wang [BIBLIOGRAPHY - C][Beni, G., Wang J., “Self Organizing Sensory Systems,” in “Highly Redundant Sensing in Robotic Systems”, Tou, J.T., Balchen, J. G., eds., Proc. of NATO Advanced Workshop on Highly Redundant Sensing in Robotic Systems, Il Ciocco, Italy (June 1988), Springer-Verlag, Berlin (1990) 251–262] coined the term back in 1989 when they presented a paper on Cellular Automata. Cellular Automata was (and is) a fascinating field of study related to self-replicating robots that assemble into complex shapes like biologic cells do.

Its core field of study covers analysing the collective intelligence that arises from swarms, i.e. a large number of homogeneous and simple individuals that interact with their environment and between themselves in a decentralised and self-organized manner with rules provided by said environment. The capability of swarms to evolve and adapt to solve complex tasks still baffles scientists and researchers across different areas such as Biology, Sociology, Economics, Robotics, etc... [BIBLIOGRAPHY - A]

The strength of the swarm arises under the presence of several of its individuals in a medium. Whereas an isolated instance of the swarm might not be simple, the complexity of its behaviour pales in comparison to that of the whole collective when it interacts with its environment. Through cooperation or competition, the swarm is capable of autonomously navigate through it and achieve solutions to several different challenges.

Due to its decentralized nature, positive properties arise from defining a swarm intelligence system to solve a difficult task when compared to building a centralized global solution.

**Cost effective:** On the one hand, the construction of a single complex robot to solve a complex task requires a high cost in design of the hardware and software, a large cost in manufacturing its custom designed components and the specialized maintenance that the machine would require. Also, an individual agent supposes a higher risk for the task since any failure in the software or broken piece of hardware might be costly to analyse and fix. In the other hand, a swarm intelligence system is more cost-effective to produce. Agents and their control system are simpler in comparison to its complex individual counterpart. The production of the agents can be parallelized, maintenance gets simpler and cheaper and even the complete failure of one agent of the swarm is easier to replace.

**Robust:** If an agent of a swarm were to be removed from it by any circumstance, the behaviour of the swarm wouldn't change and its effectivity in solving the problem would be barely

reduced. The decentralized nature of the swarm allows it to lose some of its members and still be able to perform the task it is due, which is really important in situations where failure is catastrophic.

**Less control:** Swarm intelligence is autonomous by nature. Each of the agents of the swarm are equipped with a set of simple rules to interact with the environment and with one-another. These same rules can adapt given the specific scenario they are facing to change its behaviour and be able to perform multiple tasks with little programming. There is no need for a central of control although it can be simply added for data-gathering.

**Parallelizable:** Given that swarms are simpler to build than a set of complex robots for the same tasks, their production can be done in parallel thus saving time and energy on it. This parallel nature is also expressed in the field when performing a task, since different sub-groups of the same swarm can form and work on several different scenarios that are apart from each other without any extra programming needed. Also, the task can be done continuously since agents that need to recharge their energy supply or repaired do little effect to the overall work of the swarm and can be substituted by other agents.

Clear examples surface when watching the behaviour of social animals. You can see swarm intelligence when watching the collision avoidance of birds when flocking, the architectural ability of bees to build their beehives or the food command chains of ants and their behaviour while dealing with their dead relatives [Bibliography: <https://www.nationalgeographic.com/news/2014/7/14070>] as a prime example of the power of self-organisation of simple individuals. Individuals of said aggrupations cannot exhibit the properties that emerge from the social collective by their own. You cannot see one bird flock, nor one bee building an entire beehive and less even an ant organizing an entire nest. Each of these individuals is equipped with a series of sensors that allows them to communicate and perceive their peers and the medium they inhabit. Adding that to a simple set of rules such as "See bird, avoid bird" or "Smell 4 warrior ants, become gatherer ant" [BIBLIOGRAPHY HERE] gives the swarm the ability to act in these complex ways.

Common behaviours in Nature are often the focus of studies in Swarm Intelligence. The study of behaviours of several social animals and their interactions led to the creation of techniques that work perfectly with stochastic problems, that is, problems that deal with random processes. These algorithms are called metaheuristics, and they are starting to emerge as an alternative to classical solutions to problems with uncertainty, stochastic or dynamic parts [Bibliography: Bianchi, L., Dorigo, M., Gambardella, L.M. et al. A survey on metaheuristics for stochastic combinatorial optimization. *Nat Comput* 8, 239–287 (2009). <https://doi.org/10.1007/s11047-008-9098-4>]. Some of these metaheuristics are briefly explained in the next list to highlight the importance of Swarm Intelligence [Bibliography: Saka, M. P., Doğan, E., Aydogdu, I. (2013). Analysis of Swarm Intelligence-Based Algorithms for Constrained Optimization. *Swarm Intelligence and Bio-Inspired Computation*, 25–48. doi:10.1016/b978-0-12-405163-8.00002-8 // Link: <https://sci-hub.tw/https://www.sciencedirect.com/science/a>]

**Ant Colony Optimization (ACO):** It is based in the ants colony's system to achieve the shortest path to a source of food by using pheromones. Pheromones are fleeting chemical substances that lingers in the air. Ants perceive the amount of pheromones through

their antennas as they walk by the different trails left by other ants, and when faced with a split path of pheromones, they always walk to the one that has the highest concentration. The shortest paths will always have a higher concentration in the end by having more ants passing through it, so the shortest paths are the ones that remain in the end. [BIBLIOGRAPHY: Dorigo, M., Maniezzo, V., Coloni, A., 1991. Distributed optimization by ant colony. Proceedings of ECAL91-First European Conference on Artificial Life. Elsevier, Paris, France, pp. 134-142. /// SIMPLIFIED: (Dorigo et al., 1991).]

**Particle Swarm Optimizer (PSO):** By imitating the intricate movement of birds when flocking or certain fishes when schooling, the algorithm solves optimization problems by exploration and information sharing. You initialize a certain number of agents with random positions and velocities. You assign a fitness function that represent how proximate the current solution is to the goal. Then, the agent decides where its next step going to be by making a weighted sum for its velocity that takes into account the previous velocity they had, a velocity that directs them to the best position they've found themselves and also a velocity that directs them to the best position the agents local to them found the best solution so far. Repeat the cycle for each agent until the agents find a solution and choose the agent that represents the best solution. [BIBLIOGRAPHY: Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. IEEE International Conference on Neural Networks, vol. 4, IEEE Press, Perth, Australia, pp. 1942-1948. /// SIMPLIFIED: (Kennedy and Eberhart, 1995)]

**Firefly Algorithm (FFA):** Fireflights use their lights to attract and find potential partners. An idealized version of this behaviour leads to a Constrained Optimization algorithm where the attractiveness of and distance between fireflights determines the objective of other fireflights. The brighter they are, the more attractive they seem. The farther they are set apart, the less attractive they seem. The algorithm sets that the fireflights will go to a fireflight that is brighter than them or random if none are found. [BIBLIOGRAPHY: Yang, X.S., 2009. Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (Eds.), Stochastic Algorithms: Foundations and Applications. SpringerVerlag, Berlin (SAGA 2009, Lecture Notes in Computer Science, 5792, 169-178). // SIMPLIFIED: (Yang, 2009).]

**Bat Algorithm (BA):** Bats emit loud pulses of sound that reverberate against the walls and objects present in their 3D environment. They use this echolocation to hunt and navigate without colliding inside the cave they're in. By measuring the difference in time of the echo between each ear, between emitting and listening it and the variation of loudness, it maps its surroundings and its able to locate the position and even the speeds of small insects. Converting this properties into an algorithm allows a number of agents to explore and hunt any prey in a closed environment. [BIBLIOGRAPHY: Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm. In: Gonzalez, J.R., et al., (Eds.), Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), vol. 284. Springer, Berlin, pp. 65-74. (Studies in Computational Intelligence). /// SIMPLIFIED: (Yang, 2010)]

**Hunting Search Algorithm (HS):** [BIBLIOGRAPHY: Oftadeh, R., Mahjoob, M.J., Shariat-

panahi, M., 2011. A novel metaheuristic optimization algorithm inspired by group hunting of animals: hunting search. *Comput. Math. Appl.* 60, 2087-2098. /// SIMPLIFIED: (Oftadeh et al., 2011).]

### **2.1.2. Types of Swarm Intelligence Models**

[HABLAR DE LAS HORMIGAS; EL ESTUDIO DE OLER COSAS] [HABLAR DE LA TROPHALLAXIS] [HABLAR DE ESTRATEGIAS DE COMPETITIVAD Y ESO]

### **2.1.3. Swarm Robotics**

## **2.2. Agent simulation**

Explicar aquí los trabajos de simulación en general de inteligencia artificial, como de estos algunos simulan hormigas de langdon, agentes cooperando ,etc. Hablar de las competiciones de DARPA y tal.

## **2.3. Fokker-Planck**

Hablar de Fokker-planck en su actualidad, como nació las mates de esto etc, lo de Einstein, ecuaciones langevin, su uso para el macro-micro, etc.

## **2.4. Parallel Programming and Simulation**

---

## 3. Objetivos (Con ejemplos de tablas)

### 3.1. Tablas

Ahora veremos otra estructura más: las tablas.

Aquí va una tabla<sup>1</sup> 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<sup>A</sup>T<sub>E</sub>X usa un sistema de parámetros para “decorar” las tablas. Puedes consultar estos parámetros en la tabla 3.2 de la página 12. La tabla se ubicará donde, a juicio de L<sup>A</sup>T<sub>E</sub>X, 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.

---

<sup>1</sup>En <http://www.tablesgenerator.com/> se puede encontrar un generador On-Line de tablas para L<sup>A</sup>T<sub>E</sub>X

Parámetro	Significado
<b>h</b>	Situa el elemento flotante <i>preferentemente</i> (es decir, si es posible) en la situación exacta donde se incluye este
<b>t</b>	Sitúa el elemento en la parte de arriba de la página
<b>b</b>	Sitúa el elemento en la parte de abajo de la página
<b>p</b>	Sitúa el elemento en una página aparte dedicada sólo a elementos flotantes; en el caso del formato <b>article</b> , 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\ \ \leftarrow
\rightarrow HORARIO: TARDE\end{tabular}}}} \ \ \hline
\textbf{dB(A)} & \textbf{Población expuesta tarde} & \textbf{\%} & \textbf{\scriptsize\leftarrow
\rightarrow 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<sup>A</sup>T<sub>E</sub>X 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
<b>TOTAL</b>	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
$L_e$ ( $\mu H$ )	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 <sup>3</sup> )	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
$\eta$	1.7%	2.4%
$S_d$ (cm <sup>2</sup> )	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		$\geq 4$	2 or 3	$\geq 4$	2 or 3	$\geq 4$	2 or 3	$\geq 4$	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. Methodology

### 4.1. Materials

Here in this section we'll detail the materials needed for the —. A

#### 4.1.1. Software

#### 4.1.2. Hardware

### 4.2. Interpretation of the Fokker-Planck equation

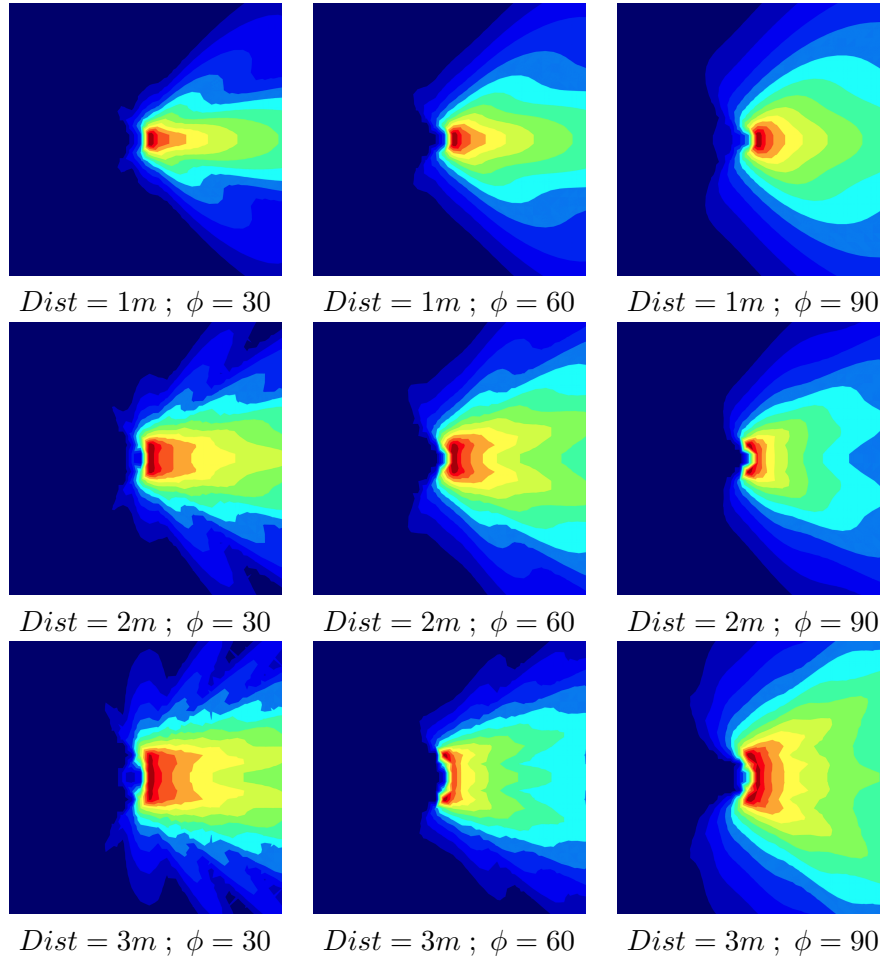
### 4.3. Inserción de figuras

Las figuras son un caso un poco especial ya que  $\text{\LaTeX}$  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 4.1 que está en la página 16, justo aquí debajo, pero  $\text{\LaTeX}$  puede que la ubique en otro lugar. (observa el código  $\text{\LaTeX}$  de este párrafo para observar como se realizan las referencias. Estos detalles también se aplican a tablas y otros objetos).

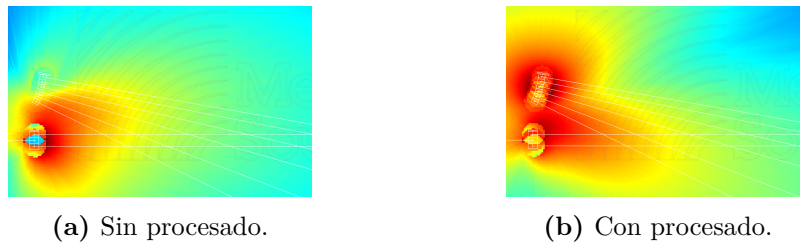
Existe también la posibilidad de realizarlo sin tablas, con subfiguras:

```
\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 linea en blanco la siguiente subfigura ira en una nueva linea
  \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}
```

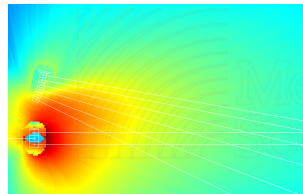
Si eliminas la línea ‘\caption’ de las subfiguras, tendrás las imágenes sin la información individual, aunque sí con la principal. Y obviamente, si eliminas el de la figura no se mostrará ninguna información.



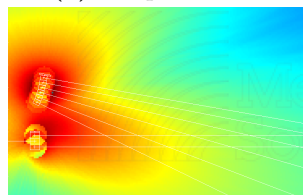
**Tabla 4.1:** Esta es una tabla con múltiples imágenes. Útil cuando se deben mostrar varias juntas.



**Figura 4.1:** Ejemplo de subfiguras



(a) Sin procesado.



(b) Con procesado.

**Figura 4.2:** Ejemplo de subfiguras vertical



## 5. Desarrollo (Con ejemplos de código)

1) Cómo hice la división de los planos en el grid <https://forum.unity.com/threads/why-is-1-unity-plane-made-of-10x10-planes.102792/>

### 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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 5.1:

```
\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 5.2 se indican algunos detalles.

Otro ejemplo, ahora para mostrar código PHP, sería escribir en tu fichero  $\text{\LaTeX}$  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<sup>A</sup>T<sub>E</sub>X 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  $\text{\LaTeX}$ .

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.

```

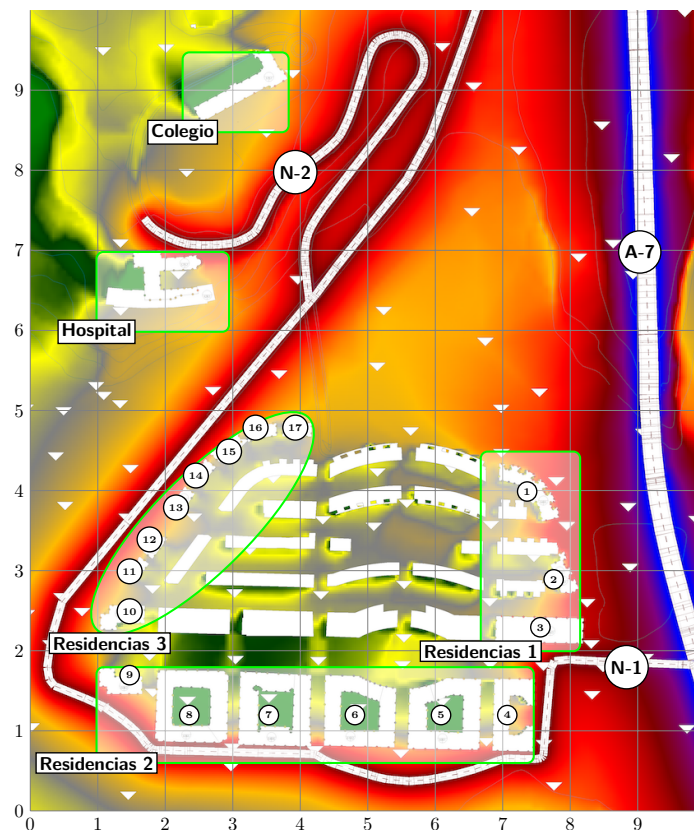
## 6. Resultados (Con ejemplos de gráficos)

### 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/>.

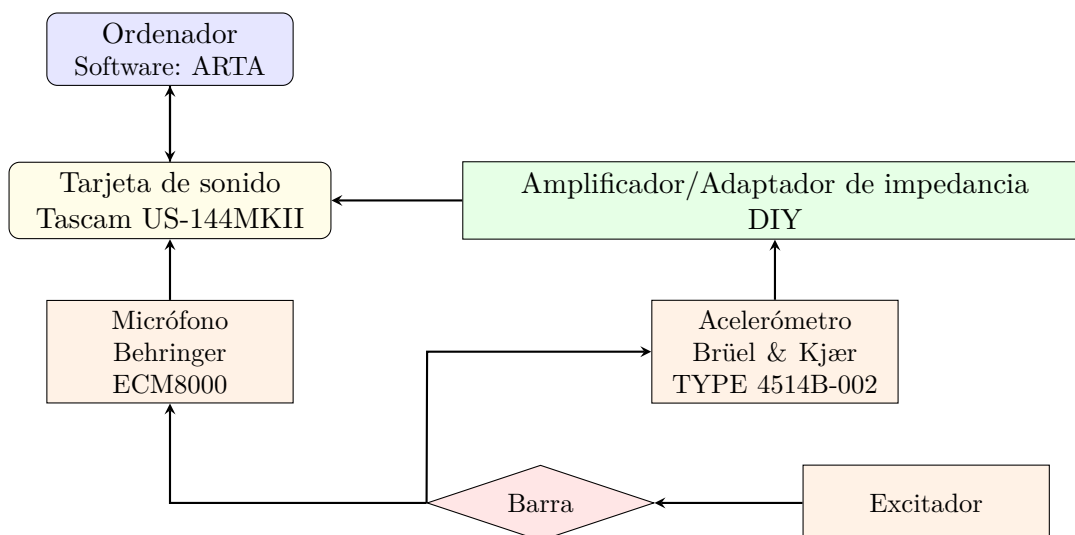


Figura 6.1: Diagrama realizado en latex con Tikz.

## 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 Tikz. También funciones para Matlab que genera las gráficas que muestra habitualmente pero en código para Tikz.

### 6.2.1. Línea

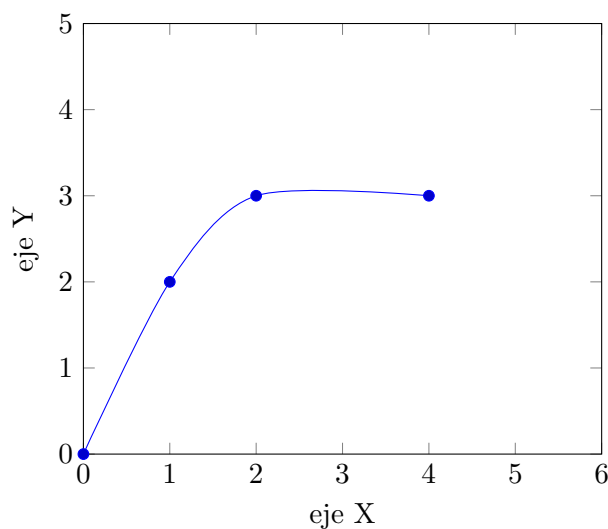
La forma más simple, aunque no sencilla cuando abarca muchos datos es la siguiente:

```

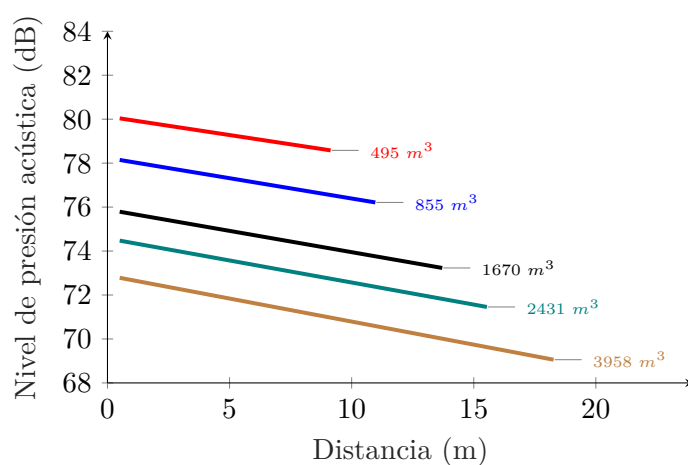
\begin{figure}[ht]
\centering
\begin{tikzpicture}
\begin{axis}
[ymin=0,ymax=5, % Límites del eje y
xmin=0,xmax=6, % Límites del eje x
ylabel= eje Y, % Nombre del eje y
xlabel= eje X] % Nombre del eje x
\addplot+[smooth] coordinates % Une los puntos curva suavizada
{(0,0) (1,2) (2,3 (4,3))}; % Puntos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

**Figura 6.3:** OP/S003

### 6.2.2. Barras

Otro ejemplo es la gráfica de barras:

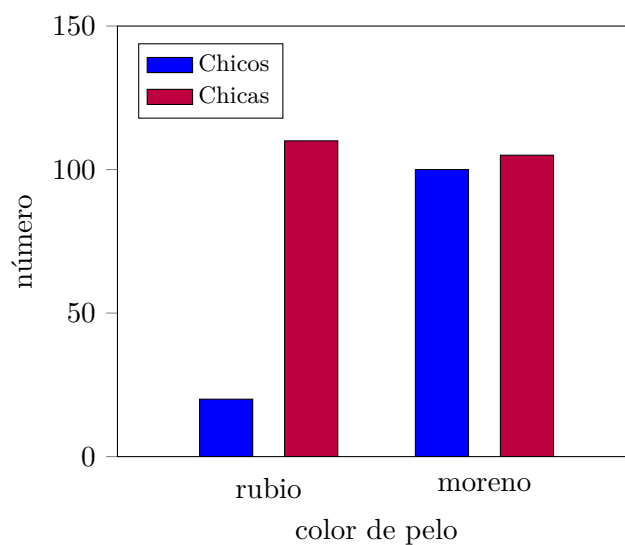
```
\begin{figure}[ht]
\centering
\begin{tikzpicture}
\begin{axis}[
ybar=12pt,
```

```

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`):



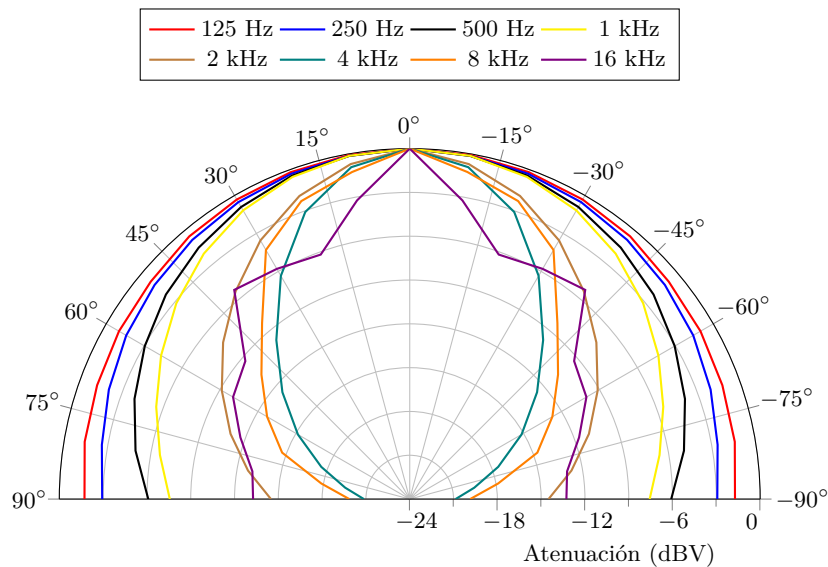


Figura 6.5: Directividad normalizada del altavoz (0 dBV en el eje).

### 6.3. Importados de MATLAB

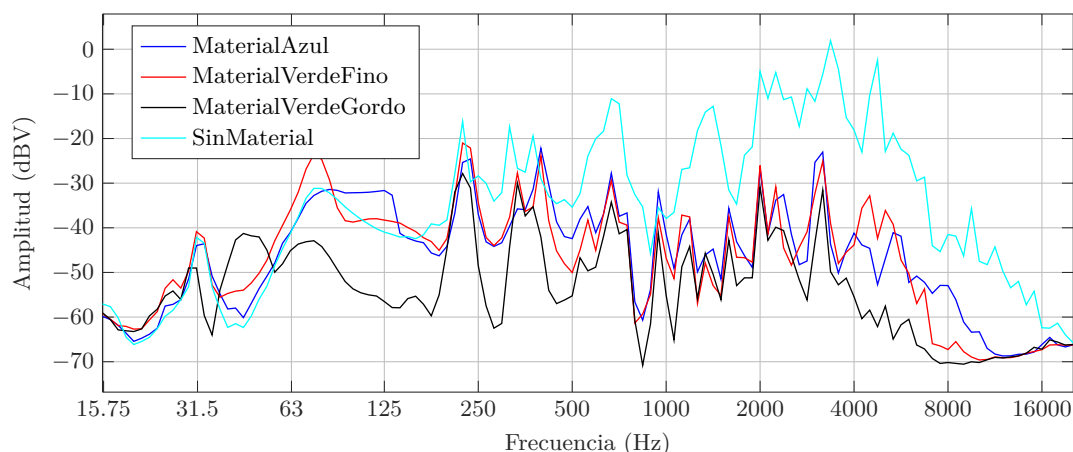
Gracias a la herramienta *matlab2tikz* (<https://es.mathworks.com/matlabcentral/fileexchange/22022-matlab2tikz-matlab2tikz>) se pueden exportar las gráficas de cualquier tipo de Matlab a latex. Después de incluir los archivos de *matlab2tikz* se debe escribir una llamada después de crear la figura tal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

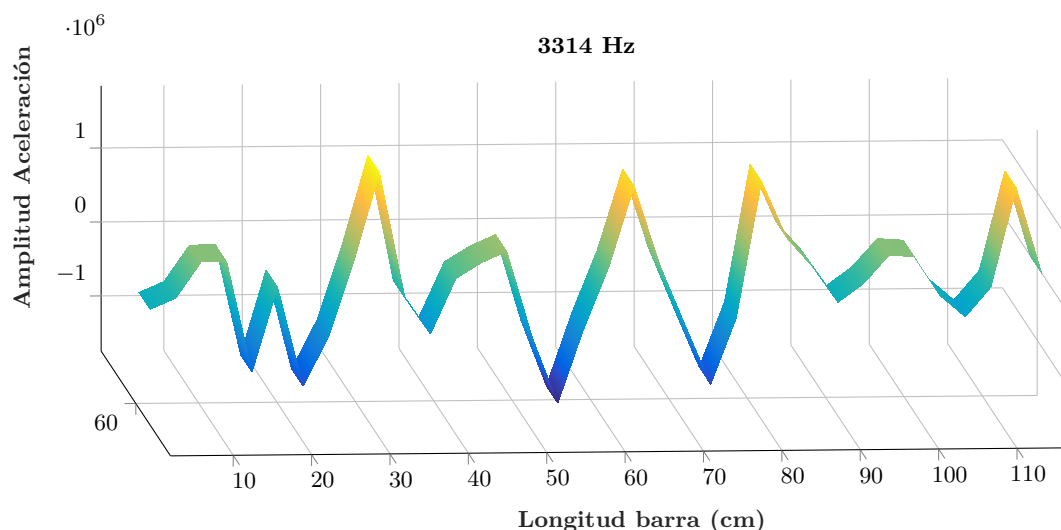
Y para utilizar el archivo generado por la función en este documento:

```
\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`:

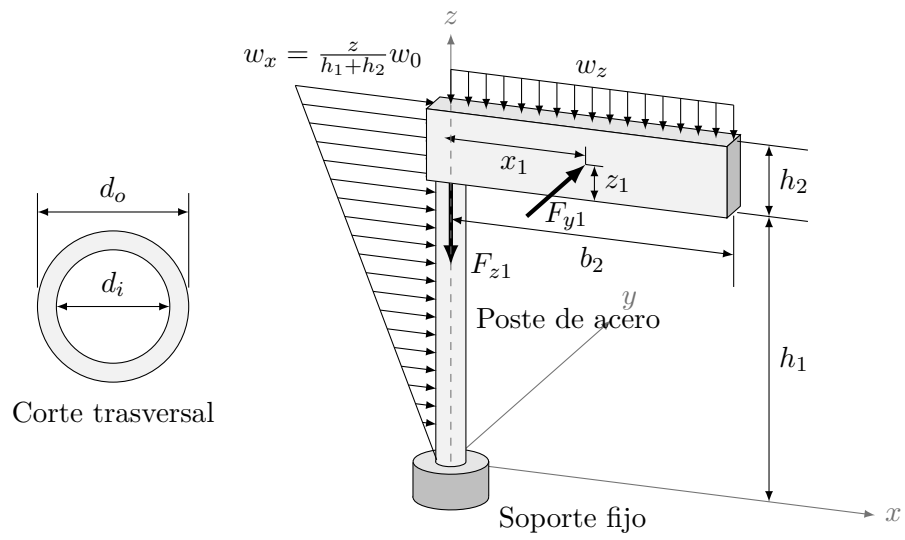


**Figura 6.7:** Amplitud de la aceleración en el modo número 8.

## 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.



## 7. Conclusiones (Con ejemplos de matemáticas)

### 7.1. Matemáticas

En  $\text{\LaTeX}$  se pueden mostrar ecuaciones de varias formas, cada una de ellas para un fin concreto.

Antes de ver algunas de estas formas hay que conocer cómo se escriben fórmulas matemáticas en  $\text{\LaTeX}$ . Una fuente de información completa es la siguiente: <https://en.wikibooks.org/wiki/LaTeX/Mathematics>. También existen herramientas online que permiten realizar ecuaciones mediante interfaz gráfica como <http://www.hostmath.com/>, <https://www.mathcha.io/editor> o <https://www.latex4technics.com/>

---

Para mostrar una ecuación numerada se debe utilizar:

```
\begin{equation}
\nabla\times{\mathbf H}=\left[\frac{1}{r}\frac{\partial}{\partial r}(rH_\theta)-\frac{1}{r}\frac{\partial H_r}{\partial\theta}\right]\hat{\mathbf z}
\label{ecuacion}
\end{equation}
```

$$\nabla \times \mathbf{H} = \left[ \frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} \right] \hat{\mathbf{z}} \quad (7.1)$$

---

Si es necesario agrupar varias ecuaciones en un mismo índice se puede escribir del siguiente modo:

```
\begin{subequations}
\begin{eqnarray}
{\mathbf E}&=&E_z(r,\theta)\hat{\mathbf z}\label{ecu1} \\
{\mathbf H}&=&H_r(r,\theta)\hat{\mathbf r}+H_\theta(r,\theta)\hat{\bm \theta}\label{ecu2}
\end{eqnarray}
\end{subequations}
% Se incluye '&' entre la igualdad para centrar las ecuaciones desde el '='.
```

$$\mathbf{E} = E_z(r, \theta) \hat{\mathbf{z}} \quad (7.2a)$$

$$\mathbf{H} = H_r(r, \theta) \hat{\mathbf{r}} + H_\theta(r, \theta) \hat{\boldsymbol{\theta}} \quad (7.2b)$$

---

Otras dos formas que son las habituales en muchos lugares para incluir ecuaciones son:

Ejemplo de fórmula en línea con el texto `\int_{a}^{b} f(x)dx = F(b) - F(a)`, esta ecuación quedará dentro  $\leftrightarrow$  del texto.

Esta otra, al utilizar dos `'$'`, se generará en una línea nueva `\int_{a}^{b} f(x)dx = F(b) - F(a)`

Ejemplo de fórmula en línea con el texto  $\int_a^b f(x)dx = F(b) - F(a)$ , esta ecuación quedará dentro del texto.

Esta otra, al utilizar dos `'$'`, se generará en una línea nueva

$$\int_a^b f(x)dx = F(b) - F(a)$$

También se puede añadir información adicional a una ecuación con la función *condiciones* creada para esta plantilla:

```
\begin{equation}
\underset{z=z_0}{\mathrm{Res}}\{f(z)\}=\frac{1}{(m-1)!}\lim_{z\rightarrow z_0}\left[\frac{d^{m-1}}{dz^{m-1}}\left[(z-z_0)^m f(z)\right]\right]
\end{equation}

\begin{condiciones}[donde:]
% Excepto 'Descripción y valor' el resto no es necesario el símbolo $para texto matemático.
% Item & Relación & Descripción o valor
m & \rightarrow & Es la multiplicidad del polo $z_0$ \\
z_0 & \rightarrow & Es la parte que se iguala a 0 con el polo. \\
f(z) & \rightarrow & Es la función contenida en la integral.
\end{condiciones}
```

$$\mathrm{Res}(f(z)) = \frac{1}{(m-1)!} \lim_{z \rightarrow z_0} \left[ \frac{d^{m-1}}{dz^{m-1}} [(z - z_0)^m f(z)] \right] \quad (7.3)$$

donde:  $m \rightarrow$  Es la multiplicidad del polo  $z_0$

$z_0 \rightarrow$  Es la parte que se iguala a 0 con el polo.

$f(z) \rightarrow$  Es la función contenida en la integral.

Si lo que deseas es una ecuación alineada a la izquierda o derecha puedes hacerlo con lo siguiente (el `'&'` simple es utilizado para alinear las ecuaciones desde ese punto, los iguales):

```
% Alineado a la izquierda al incluir al final el doble '&&'
\begin{flalign}
y_{h_1} &= \begin{bmatrix} 6\cos(\sqrt{6}x) \\ -\sqrt{6}\sin(\sqrt{6}x) \end{bmatrix} e^x && \\
y_{h_2} &= \begin{bmatrix} 6\sin(\sqrt{6}x) \\ \sqrt{6}\cos(\sqrt{6}x) \end{bmatrix} e^x && \\
\end{flalign}

% Alineado a la derecha al incluir al inicio el doble '&&'
\begin{flalign}
&& y_{h_1} = \begin{bmatrix} 6\cos(\sqrt{6}x) \\ -\sqrt{6}\sin(\sqrt{6}x) \end{bmatrix} e^x \\
&& y_{h_2} = \begin{bmatrix} 6\sin(\sqrt{6}x) \\ \sqrt{6}\cos(\sqrt{6}x) \end{bmatrix} e^x \\
\end{flalign}
```

$$y_{h_1} = \begin{bmatrix} 6 \cos(\sqrt{6}x) \\ -\sqrt{6} \sin(\sqrt{6}x) \end{bmatrix} e^x \quad (7.4)$$

$$y_{h_2} = \begin{bmatrix} 6 \sin(\sqrt{6}x) \\ \sqrt{6} \cos(\sqrt{6}x) \end{bmatrix} e^x \quad (7.5)$$

$$y_{h_1} = \begin{bmatrix} 6 \cos(\sqrt{6}x) \\ -\sqrt{6} \sin(\sqrt{6}x) \end{bmatrix} e^x \quad (7.6)$$

$$y_{h_2} = \begin{bmatrix} 6 \sin(\sqrt{6}x) \\ \sqrt{6} \cos(\sqrt{6}x) \end{bmatrix} e^x \quad (7.7)$$

Tanto con la función utilizada en (7.1,7.3), como en (7.2a,7.2b) y en las anteriores, si se les incluye un '\*' después de 'equation', 'subequation' o 'flalign', se elimina la numeración de las ecuaciones pero manteniendo el resto de características.





## **A. Anexo I**

Aquí vendría el anexo I



## B. Páginas horizontales

Aquí se muestra cómo incluir páginas en horizontal.  
Esta página está en vertical

Esta página está en horizontal

---

Esta página también está en horizontal

---

Esta página está de nuevo en vertical

---

## C. Importar PDF

A continuación se muestra una página importada de un PDF externo. Observar los comentarios en el código de este anexo para más información. También puedes leer el manual con todas las opciones en <http://osl.ugr.es/CTAN/macros/latex/contrib/pdfpages/pdfpages.pdf>.

Alicante 15 DE MARZO DE 2007

Expediente número

Referencia del petionario **AYUNTAMIENTO DE ALICANTE**  
Departamento de Medio Ambiente  
C/San Nicolás, nº 2, 4º  
03001 ALICANTE  
Contacto: Juan Luís Beresaluze

**DOCUMENTO DE SÍNTESIS**

***ELABORACIÓN DEL MAPA ACÚSTICO MUNICIPAL DE LA CIUDAD DE  
ALICANTE***

Fecha de realización del estudio: MAYO 2005 – MARZO 2007