
**Diseño y Optimización de un Sistema Empotrado de Bajo
Consumo para la Detección de Anomalías en Tiempo Real
mediante TinyML**

**Design and Optimization of a Low-Power Embedded System for
Real-Time Anomaly Detection using TinyML**



**Trabajo de Fin de Máster
Curso 2025–2026**

Autor

Ana Isabel García Fernández

Director

José Luis Risco Martín

Máster en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Diseño y Optimización de un Sistema
Empotrado de Bajo Consumo para la
Detección de Anomalías en Tiempo Real
mediante TinyML

Design and Optimization of a Low-Power
Embedded System for Real-Time Anomaly
Detection using TinyML

Trabajo de Fin de Máster en Ingeniería Informática
Departamento de Arquitectura de Computadores y Automática

Autor

Ana Isabel García Fernández

Director

José Luis Risco Martín

Convocatoria: Junio 2026

Calificación: 10 (Matrícula de Honor)

Máster en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

21 de julio de 2026

Autorización de difusión

El abajo firmante, matriculado en el Máster en Ingeniería Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo de Fin de Máster: “Diseño y Optimización de un Sistema Empotrado de Bajo Consumo para la Detección de Anomalías en Tiempo Real mediante TinyML”, realizado durante el curso académico 2025–2026 bajo la dirección de José Luis Risco Martín en el Departamento de Arquitectura de Computadores y Automática, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Ana Isabel García Fernández

27 de julio de 2025

Dedicatoria

*A mis padres por haber creído en mí desde el
comienzo de mi etapa universitaria.*

Agradecimientos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Resumen

Diseño y Optimización de un Sistema Empotrado de Bajo Consumo para la Detección de Anomalías en Tiempo Real mediante TinyML

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Palabras clave

Palabra Clave 1, Palabra Clave 2, Palabra Clave 3, Palabra Clave 4, Palabra Clave 5, Palabra Clave 6, Palabra Clave 7.

Abstract

Design and Optimization of a Low-Power Embedded System for Real-Time Anomaly Detection using TinyML

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Keywords

Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5, Keyword 6, Keyword 7.

Índice general

| | |
|---|-----------|
| 1. Introducción | 1 |
| 1.1. Citas y Referencias Bibliográficas | 1 |
| 1.1.1. Añadir una nueva fuente | 1 |
| 1.1.2. Citar en el texto | 2 |
| 1.2. Uso del Glosario y Acrónimos | 2 |
| 1.2.1. Definir un nuevo término | 2 |
| 1.2.2. Citar términos en el texto | 2 |
| 1.3. Inclusión de Figuras | 3 |
| 1.4. Inclusión de Tablas | 4 |
| 1.4.1. Tablas que ocupan varias páginas | 5 |
| 1.5. Inclusión de Listados de Código | 6 |
| 1.6. Uso de Ecuaciones | 7 |
| 1.6.1. Ecuaciones en línea | 7 |
| 1.6.2. Ecuaciones numeradas y centradas | 7 |
| 1.6.3. Sistemas de ecuaciones | 8 |
| 2. Resto de capítulos | 9 |
| 3. Conclusiones y trabajo futuro | 11 |
| 4. Introduction | 13 |
| 5. Conclusion and future work | 15 |
| Bibliografía | 17 |
| Acrónimos | 19 |

Índice de tablas

| | |
|---|---|
| 1.1. Ejemplo de tabla formateada con el paquete booktabs. | 5 |
|---|---|

Índice de figuras

| | |
|---|---|
| 1.1. Diagrama de bloques del sistema empotrado para detección de anomalías. | 4 |
|---|---|

Índice de listados de código

| | |
|---|---|
| 1.1. Ejemplo de un listado de código en Python. | 7 |
|---|---|

Capítulo 1

Introducción

Este capítulo sirve como un manual de referencia para utilizar las funcionalidades de L^AT_EX incorporadas en esta plantilla. El objetivo es facilitar la redacción del documento, asegurando un formato consistente y profesional para las citas bibliográficas, figuras, tablas, ecuaciones, etc. A continuación, se detalla el procedimiento para cada uno de estos elementos.

1.1. Citas y Referencias Bibliográficas

La gestión de la bibliografía en este documento está automatizada mediante BibTeX. Todas las fuentes (artículos, libros, páginas web, etc.) deben ser añadidas al archivo `biblio.bib` que se encuentra en el directorio principal del proyecto.

1.1.1. Añadir una nueva fuente

Para añadir una nueva referencia, abre el archivo `biblio.bib` y añade una entrada con el formato BibTeX correspondiente. Cada entrada tiene un tipo (p.ej., `@article`, `@book`, `@misc`), una clave de cita única y varios campos con la información de la fuente.

Por ejemplo, para añadir una referencia a la API de simulación xDEVS, podrías añadir la siguiente entrada:

```
@article{RiscoMartin2023,  
  title={x{DEVS}: {A} toolkit for interoperable modeling and simulation of  
        formal discrete event systems},  
  author={Risco-Mart{\`i}n, Jos{\'e} L and Mittal, Saurabh and Henares, Kevin  
        and Cardenas, Rom{\'a}n and Arroba, Patricia},  
  journal={Software: Practice and Experience},  
  volume={53},  
  number={3},  
  pages={748--789},  
  year={2023},  
  publisher={Wiley Online Library}  
}
```

La clave de cita en este caso es `RiscoMartin2023`, y es la que usaremos para referenciar esta fuente en el texto.

1.1.2. Citar en el texto

Una vez que la fuente está en el archivo `.bib`, puedes citarla en cualquier parte del documento usando el comando `\cite{}`, introduciendo la clave de la cita dentro de las llaves. Por ejemplo:

El formalismo Especificación de Sistemas de Eventos Discretos, del inglés *Discrete Event System specification* (DEVS) permite modelar sistemas complejos mediante la descomposición en modelos atómicos y acoplados. Para implementar el modelo DEVS, se utilizó la Interfaz de Programación de Aplicaciones, del inglés *Application Programming Interface* (API) de simulación xDEVS [1].

Si necesitas citar varias fuentes en el mismo punto, simplemente sepáralas por comas dentro del mismo comando `\cite{}`. Por ejemplo:

```
...varios autores han tratado este tema \cite{Zeigler2018, RiscoMartin2023}.
```

Que quedaría: varios autores han tratado este tema [2, 1].

La bibliografía se generará automáticamente al final del documento, en la sección “Bibliografía”, incluyendo todas las fuentes que hayas citado en el texto.

1.2. Uso del Glosario y Acrónimos

Para asegurar la claridad y consistencia del documento, la plantilla facilita el uso de un glosario y una lista de acrónimos gestionados por el paquete `glossaries`. Esto permite definir un término una sola vez y referenciarlo a lo largo del texto de forma automática.

1.2.1. Definir un nuevo término

Todos los términos, ya sean acrónimos o entradas de glosario, deben definirse en el fichero `glosario.tex`, que se carga automáticamente en el documento principal.

Acrónimos

Para añadir un nuevo acrónimo, utiliza el comando `\newacronym`. Su estructura es:

```
\newacronym{etiqueta}{acrónimo}{nombre completo}
```

La `etiqueta` es una clave única que usarás para referenciarlo en el texto.

Por ejemplo, para definir el acrónimo TFM:

```
\newacronym{tfm}{TFM}{Trabajo de Fin de Máster}
```

También se puede definir un glosario, pero no es lo habitual en este tipo de trabajos. Por ello, solo usaremos la opción de crear y usar acrónimos.

1.2.2. Citar términos en el texto

Una vez definidos los términos en `glosario.tex`, puedes usarlos en el texto con los siguientes comandos. El paquete se encargará automáticamente de expandir el acrónimo la primera vez que se use y de utilizar solo la forma corta en las siguientes apariciones.

- `\gls{etiqueta}`: Es el comando principal. La primera vez que se usa, muestra el nombre completo seguido del acrónimo entre paréntesis. Por ejemplo, la primera vez que escribas `\gls{tfm}`, aparecerá “Trabajo de Fin de Máster (TFM)”. Las siguientes veces, solo aparecerá “TFM”.

- `\glspl{etiqueta}`: Se usa para la forma plural del término. Por ejemplo, `\glspl{hems}` mostrará “Home Energy Management Systems (HEMSs)” la primera vez, y “HEMSs” después. En español hay que tener mucho cuidado con esto porque los acrónimos no tienen el mismo tratamiento que en inglés (el plural no se forma igual y además no suelen ir anticipadas por artículo) [?].
- `\Gls{etiqueta}` y `\Glspl{etiqueta}`: Son las versiones con mayúscula inicial, ideales para usar al principio de una frase.

Al final del documento, la lista de acrónimos y el glosario se generarán automáticamente, incluyendo solo los términos que hayas utilizado en el texto.

1.3. Inclusión de Figuras

Las figuras son un componente esencial para ilustrar conceptos, arquitecturas o resultados. Para una correcta organización del proyecto, se recomienda guardar todos los archivos de imagen en una subcarpeta dedicada, como por ejemplo la carpeta `fig/`, que ya se encuentra creada en esta plantilla.

Para insertar una figura en el documento, se utiliza el entorno `figure`. Este entorno gestiona la colocación de la imagen como un elemento flotante, aunque se puede forzar su posición con el especificador `[H]` (del paquete `float`) si se desea que aparezca exactamente en ese punto del texto¹.

El siguiente código muestra cómo añadir una figura con su leyenda y prepararla para ser referenciada:

```
\begin{figure}
  \centering
  \includegraphics[width=0.8\textwidth]{fig/mi_figura.png}
  \caption{Este es el pie de foto que describe la figura.}
  \label{fig:mi_figura}
\end{figure}
```

Los componentes clave de este bloque son:

- `\centering`: Centra la figura horizontalmente en la página.
- `\includegraphics[width=...]{...}`: Es el comando que inserta la imagen. La opción `width` permite escalar la imagen a un ancho determinado (en el ejemplo, al 80 % del ancho de texto). La ruta del archivo apunta a la imagen que se desea incluir.
- `\caption{...}`: Añade una leyenda descriptiva y numerada automáticamente debajo de la figura.
- `\label{...}`: Asigna un identificador único a la figura. Es fundamental para poder referenciarla desde otras partes del documento. Se recomienda usar un prefijo como `fig:` para evitar confusiones con etiquetas de tablas, ecuaciones, etc.

Una vez que la figura ha sido etiquetada con `\label`, puedes hacer referencia a ella en el texto de forma automática utilizando el comando `\autoref{}`. Esta es la forma recomendada, ya que genera automáticamente el tipo de elemento y su número (p.ej., “Figura 1.1”).

Por ejemplo, para citar la figura anterior, escribirías:

... la arquitectura del sistema se muestra en la `\autoref{fig:mi_figura}`.

¹El uso de la opción `H` se desaconseja totalmente.

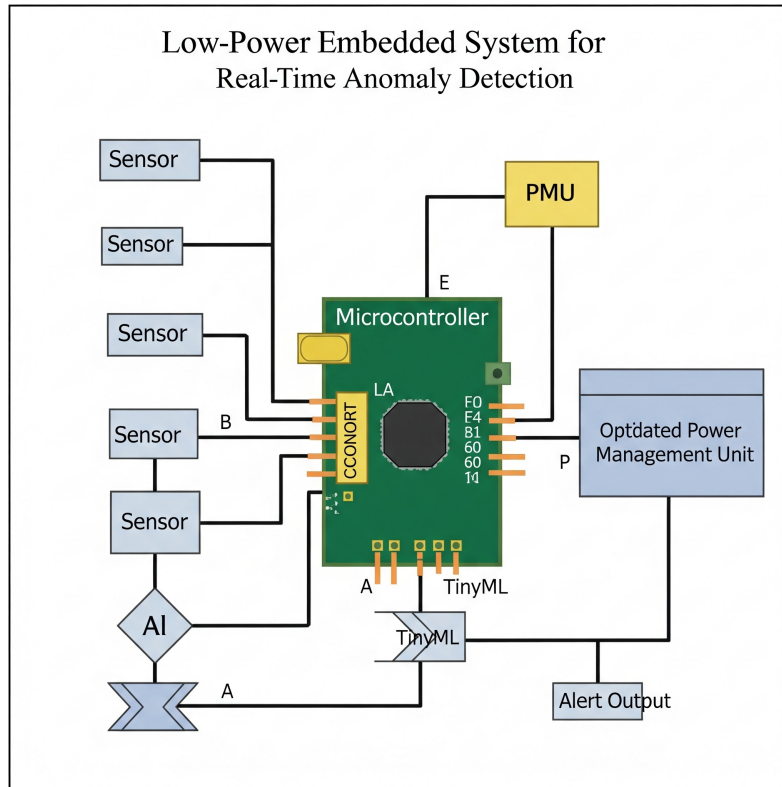


Figura 1.1: Diagrama de bloques del sistema empujado para detección de anomalías.

Al compilar el documento, LaTeX reemplazará `\autoref{fig:mi_figura}` por “Figura 1.1” (o el número que corresponda), creando además un hipervínculo a la misma gracias al paquete `hyperref`. Por ejemplo:

En la [Figura 1.1](#) se ilustra el diagrama de bloques del sistema empujado que se ha diseñado en este Trabajo Fin de Máster (TFM).

1.4. Inclusión de Tablas

Las tablas son una herramienta esencial para presentar datos de forma estructurada, clara y comparativa. En \LaTeX , las tablas se construyen generalmente utilizando el entorno `table`, que actúa como un contenedor flotante, junto con el entorno `tabular`, que es el encargado de organizar el contenido en filas y columnas.

Para garantizar una apariencia profesional y mejorar la legibilidad, esta plantilla incluye el paquete `booktabs`. Se recomienda encarecidamente utilizar sus comandos (`\toprule`, `\midrule` y `\bottomrule`) para las líneas horizontales, ya que crean un espaciado más adecuado que el tradicional `\hline`. Como norma general, es preferible evitar las líneas verticales para un diseño más limpio.

El siguiente código muestra cómo definir una tabla, asignarle un título y prepararla para ser referenciada:

```
\begin{table}[H]
  \centering
  \caption{Ejemplo de tabla formateada con el paquete booktabs.}
  \label{tab:ejemplo_consumo}
  \begin{tabular}{lcc}
```

Tabla 1.1: Ejemplo de tabla formateada con el paquete booktabs.

| Modo | Consumo (mA) | Potencia (mW) |
|-------------|--------------|---------------|
| Activo | 50 | 165 |
| Con WiFi | 250 | 825 |
| Light Sleep | 1,2 | 3,96 |

```

\toprule
\textbf{Componente} & \textbf{Consumo (W)} & \textbf{Horas de uso} \\
\midrule
Lavadora & 2200 & 1.5 \\
Lavavajillas & 1800 & 1.0 \\
Cargador VE & 7400 & 3.0 \\
\bottomrule
\end{tabular}
\end{table}

```

Los componentes clave son:

- `\begin{table}[H]`: Inicia el entorno flotante. Al igual que con las figuras, el especificador `[H]` del paquete `float` fuerza su aparición en ese punto exacto del texto, aunque se desaconseja su uso para permitir que `LATEX` optimice la maquetación.
- `\centering`: Centra la tabla horizontalmente.
- `\caption{...}`: Añade un título descriptivo y numerado que aparecerá encima de la tabla.
- `\label{tab:ejemplo_consumo}`: Asigna un identificador único para poder referenciarla. Se recomienda usar el prefijo `tab:`.
- `\begin{tabular}{lcc}`: Define la estructura de la tabla, en este caso con tres columnas: la primera alineada a la izquierda (`l`) y las otras dos centradas (`c`).
- `\toprule`, `\midrule`, `\bottomrule`: Comandos de `booktabs` para las líneas superior, media e inferior, respectivamente.

Para hacer referencia a la tabla en el texto, se utiliza el comando `\autoref{}`, que genera automáticamente la palabra "Tabla" seguida de su número y crea un hipervínculo.

En la `\autoref{tab:ejemplo_consumo}` se detallan los consumos de la `\gls{cpu}` del microcontrolador, según el modo de funcionamiento

Vemos el ejemplo en uso a continuación: En la [Tabla 1.1](#) se detallan los consumos de la Unidad Central de Proceso, del inglés *Central Processing Unit* (CPU) del microcontrolador, según el modo de funcionamiento.

1.4.1. Tablas que ocupan varias páginas

Si una tabla es demasiado extensa para caber en una sola página, el entorno `table` no es adecuado. Para estos casos, la plantilla carga el paquete `longtable`, que permite que las tablas se dividan y continúen en las páginas siguientes de forma automática, repitiendo la cabecera si así se especifica.

1.5. Inclusión de Listados de Código

Para presentar algoritmos o fragmentos de código fuente, la plantilla integra el paquete `listings`, que proporciona un entorno específico para formatear el código de manera clara y legible. Se ha preconfigurado un estilo visual consistente para asegurar que todos los listados de código mantengan una apariencia profesional y homogénea a lo largo del documento.

Para insertar un bloque de código, se utiliza el entorno `lstlisting`. Este entorno numera automáticamente los listados y les aplica el formato definido en la plantilla (resaltado de sintaxis, color de fondo, etc.).

El siguiente ejemplo muestra cómo incluir un listado de código, con su leyenda y etiqueta para poder ser referenciado posteriormente:

```
\begin{lstlisting}[language=Python, caption={Ejemplo de un listado de código en Python.}, la
# Este es un comentario
def factorial(n):
    """
    Calcula el factorial de un número entero no negativo.
    """
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

# Llamada a la función
resultado = factorial(5)
print(f"El factorial de 5 es {resultado}")
\end{lstlisting}
```

Los componentes clave son:

- `\begin{lstlisting}[...]`: Inicia el entorno para el listado de código. Dentro de los corchetes se pueden especificar opciones importantes:
 - `language=...`: Define el lenguaje de programación (p. ej., Python, C++, Java, SQL, XML). Esto es crucial para que el paquete resalte la sintaxis correctamente.
 - `caption={...}`: Añade una leyenda descriptiva y numerada debajo del listado de código.
 - `label={...}`: Asigna un identificador único. Se recomienda usar el prefijo `lst:` para evitar confusiones con otras etiquetas.
- `\end{lstlisting}`: Finaliza el entorno del listado de código.

Una vez que el listado ha sido etiquetado con `\label`, puedes hacer referencia a él en el texto utilizando el comando `\autoref{}`. De forma similar a las figuras y tablas, este comando generará automáticamente el tipo de elemento y su número (p. ej., "Listado 1.1"), creando además un hipervínculo.

Por ejemplo, para citar el listado anterior, escribirías:

En el `\autoref{lst:ejemplo_python}` se muestra una función para calcular el factorial.

Que viene a quedar como sigue: En el [Listado 1.1](#) se muestra una función para calcular el factorial.

```
# Este es un comentario
def factorial(n):
    """
    Calcula el factorial de un número entero no negativo.
    """
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

# Llamada a la función
resultado = factorial(5)
print(f"El factorial de 5 es {resultado}")
```

Listado 1.1: Ejemplo de un listado de código en Python.

1.6. Uso de Ecuaciones

Las ecuaciones son cruciales para expresar formalmente conceptos matemáticos, modelos y algoritmos. LaTeX ofrece varias formas de incluir ecuaciones, ya sea en línea con el texto o como elementos separados y numerados.

1.6.1. Ecuaciones en línea

Para incluir ecuaciones pequeñas directamente dentro de una frase, se utiliza el modo matemático en línea, delimitado por signos de dólar (\$). Por ejemplo, para expresar la famosa ecuación de Einstein, escribiríamos $E=mc^2$, que se verá como $E = mc^2$. Este método es útil para fórmulas cortas que no necesitan una numeración o un espacio dedicado.

1.6.2. Ecuaciones numeradas y centradas

Cuando una ecuación es más compleja, necesita destacarse del texto, o se desea referenciarla posteriormente, se utiliza el entorno `equation`. Este entorno centra la ecuación automáticamente y le asigna un número secuencial.

El siguiente ejemplo muestra cómo definir una ecuación:

```
\begin{equation}\label{eq:parabola}
    y = ax^2 + bx + c
\end{equation}
```

Los componentes clave son:

- `\begin{equation}`: Inicia el entorno de ecuación.
- `\label{eq:parabola}`: Asigna un identificador único a la ecuación. Se recomienda usar el prefijo `eq:` para evitar confusiones con otras etiquetas.

Para hacer referencia a esta ecuación en el texto, se utiliza el comando `\eqref{}`. Este comando genera automáticamente el número de la ecuación entre paréntesis (p.ej., “(1.1)”) y crea un hipervínculo. Por ejemplo:

En (1.1) se muestra la ecuación general de una parábola.

$$y = ax^2 + bx + c \tag{1.1}$$

1.6.3. Sistemas de ecuaciones

Para sistemas de ecuaciones, el entorno `eqnarray` es una opción que permite alinear múltiples ecuaciones. Sin embargo, se recomienda el uso del paquete `amsmath` y sus entornos como `align` o `gather` para una mayor flexibilidad y un mejor espaciado.

Aquí se muestra un ejemplo básico con `eqnarray` (aunque se sugiere `amsmath` para casos más avanzados):

```
\begin{eqnarray}\label{eq:sistema_ecuaciones}
  2x + 3y &=& 7 \\\
  x - y &=& 1
\end{eqnarray}
```

Este código produce el siguiente sistema de ecuaciones:

$$2x + 3y = 7 \tag{1.2}$$

$$x - y = 1 \tag{1.3}$$

Se puede hacer referencia a este sistema de la misma manera que una ecuación individual, por ejemplo, utilizando la ecuación (1.2).

Capítulo 2

Resto de capítulos

Aquí irían el resto de capítulos. Generalmente *Estado del arte*, *Metodología*, y *Experimentos y resultados*, por poner un ejemplo.

Capítulo 3

Conclusiones y trabajo futuro

Aquí van las conclusiones y trabajo futuro.

Capítulo 4

Introduction

Traduce aquí tu capítulo de introducción.

Capítulo 5

Conclusion and future work

Traduce aquí tu capítulo de conclusiones y trabajo futuro.

Bibliografía

- [1] J. L. Risco-Martín, S. Mittal, K. Henares, R. Cardenas, and P. Arroba, “xDEVs: A toolkit for interoperable modeling and simulation of formal discrete event systems,” *Software: Practice and Experience*, vol. 53, no. 3, pp. 748–789, 2023.
- [2] B. P. Zeigler, A. Muzy, and E. Kofman, *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic Press, 2018.

Acrónimos

API Interfaz de Programación de Aplicaciones, del inglés *Application Programming Interface*. [2](#)

CPU Unidad Central de Proceso, del inglés *Central Processing Unit*. [5](#)

DEVS Especificación de Sistemas de Eventos Discretos, del inglés *Discrete Event System specification*. [2](#)

TFM Trabajo Fin de Máster. [4](#)