



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

**CARRERA DE ESPECIALIZACIÓN EN
SISTEMAS EMBEBIDOS**

MEMORIA DEL TRABAJO FINAL

**Sistema de Notificación de Alarma de
Incendio**

Autor:

Ing.Daniel Marquez

Director:

Mg.Ing. Wilmer Sanz (UC)

Jurados:

Dr.Ing.Adrián Stacul (UTN)

Esp. Ing. Diego Fernández (FIUBA, DEBMEDIA)

Esp. Ing. Santiago Salamandri (FIUBA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires,
entre marzo de 2020 y diciembre de 2020.*

Resumen

La presente memoria describe la implementación de un sistema de monitoreo de alarmas de incendio, con el objetivo de satisfacer la necesidad detectada por la firma Isolse SRL; brindar a sus clientes la posibilidad de conocer de forma remota el estado de su instalación de detección de incendio.

El sistema se elabora utilizando conocimientos de programación de microcontroladores e ingeniería de software, para el desarrollo de un sistema modularizado y respaldado con control de versiones. Se basa en el desarrollo de aplicaciones en sistemas operativos Linux, protocolos de comunicación, diseño de circuitos impresos, manejo de bases de datos y micro-servidor web.

Agradecimientos

Esta sección es para agradecimientos personales y es totalmente **OPCIONAL**.

Índice general

| | |
|---|-----------|
| Resumen | I |
| 1. Introducción general | 1 |
| 1.1. Aprendiendo \LaTeX | 1 |
| 1.1.1. Una introducción (no tan corta) a \LaTeX | 1 |
| Una subsubsección | 1 |
| 1.1.2. Guía matemática rápida para \LaTeX | 1 |
| 1.2. Utilizando esta plantilla | 2 |
| 1.3. Qué incluye esta plantilla | 2 |
| 1.3.1. Carpetas | 2 |
| 1.3.2. Archivos | 3 |
| 1.4. Entorno de trabajo | 4 |
| 1.4.1. Paquetes adicionales | 4 |
| 1.4.2. Configurando TexMaker | 5 |
| 1.5. Personalizando la plantilla, el archivo memoria.tex | 6 |
| 1.6. El código del archivo memoria.tex explicado | 6 |
| 1.7. Bibliografía | 7 |
| 2. Introducción específica | 9 |
| 2.1. Estilo y convenciones | 9 |
| 2.1.1. Uso de mayúscula inicial para los título de secciones | 9 |
| 2.1.2. Este es el título de una subsección | 9 |
| 2.1.3. Figuras | 10 |
| 2.1.4. Tablas | 12 |
| 2.1.5. Ecuaciones | 12 |
| 3. Diseño e implementación | 15 |
| 3.1. Análisis del software | 15 |
| 4. Ensayos y Resultados | 17 |
| 4.1. Pruebas funcionales del hardware | 17 |
| 5. Conclusiones | 19 |
| 5.1. Conclusiones generales | 19 |
| 5.2. Próximos pasos | 20 |

Índice de figuras

| | |
|---|----|
| 1.1. Entorno de trabajo de texMaker. | 5 |
| 1.2. Definir memoria.tex como documento maestro. | 6 |
| 2.1. Ilustración del cuadrado azul que se eligió para el diseño del logo. | 11 |
| 2.2. Imagen tomada de la página oficial del procesador ¹ | 11 |
| 2.3. ¿Por qué de pronto aparece esta figura? | 11 |
| 2.4. Tres gráficos simples | 11 |

Índice de Tablas

| | |
|------------------------------|----|
| 2.1. caption corto | 12 |
|------------------------------|----|

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción general

1.1. Aprendiendo L^AT_EX

L^AT_EX no es WYSIWYG (What You See is What You Get), a diferencia de los procesadores de texto como Microsoft Word o Pages de Apple o incluso LibreOffice en el mundo open-source. En lugar de ello, un documento escrito para L^AT_EX es en realidad un archivo de texto simple o llano que *no contiene formato*. Nosotros le decimos a L^AT_EX cómo deseamos que se aplique el formato en el documento final escribiendo comandos simples entre el texto, por ejemplo, si quiero usar texto en itálicas para dar énfasis, escribo `\it{texto}` y pongo el texto que quiero en itálicas entre medio de las llaves. Esto significa que L^AT_EX es un lenguaje del tipo «mark-up», muy parecido a HTML.

1.1.1. Una introducción (no tan corta) a L^AT_EX

Si sos nuevo en L^AT_EX, hay un muy buen libro electrónico - disponible gratuitamente en Internet como un archivo PDF - llamado, «A (not so short) Introduction to L^AT_EX». El título del libro es generalmente acortado a simplemente *lshort*. Puede descargar la versión más reciente en inglés (ya que se actualiza de vez en cuando) desde aquí: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

Se puede encontrar la versión en español en la lista en esta página: <http://www.ctan.org/tex-archive/info/lshort/>

Una subsubsección

Acá tiene un ejemplo de una “subsubsección” que es el cuarto nivel de ordenamiento del texto, después de capítulo, sección y subsección. Como se puede ver, las subsubsecciones no van numeradas en el cuerpo del documento ni en el índice. El formato está definido por la plantilla y no debe ser modificado.

1.1.2. Guía matemática rápida para L^AT_EX

Si estás escribiendo un documento con mucho contenido matemático, entonces es posible que desees leer el documento de la AMS (American Mathematical Society) llamado, «A Short Math Guide for L^AT_EX». Se puede encontrar en línea en el siguiente link: <http://www.ams.org/tex/amslatex.html> en la sección «Additional Documentation» hacia la parte inferior de la página.

1.2. Utilizando esta plantilla

Si estás familiarizado con \LaTeX , entonces podés explorar la estructura de directorios de esta plantilla y proceder a personalizarla agregando tu información en el bloque *INFORMACIÓN DE LA PORTADA* en el archivo `memoria.tex`.

Se puede continuar luego modificando el resto de los archivos siguiendo los lineamientos que se describen en la sección 1.5 en la página 6.

Debés asegurarte de leer el capítulo 2 acerca de las convenciones utilizadas para las Memoria de los Trabajos Finales de la Carrera de Especialización en Sistemas Embebidos.

Si sos nuevo en \LaTeX , se recomienda que continúes leyendo el documento ya que contiene información básica para aprovechar el potencial de esta herramienta.

1.3. Qué incluye esta plantilla

1.3.1. Carpetas

Esta plantilla se distribuye como un único archivo .zip que se puede descomprimir en varios archivos y carpetas. Asimismo, se puede consultar el repositorio git para obtener la última versión de los archivos, <https://github.com/patriciobos/Plantilla-CESE.git>. Los nombres de las carpetas son, o pretender ser, auto-explicativos.

Appendices – Esta es la carpeta donde se deben poner los apéndices. Cada apéndice debe ir en su propio archivo `.tex`. Se incluye un ejemplo y una plantilla en la carpeta.

Chapters – Esta es la carpeta donde se deben poner los capítulos de la memoria. Cada capítulo debe ir en su propio archivo `.tex` por separado. Se ofrece por defecto, la siguiente estructura de capítulos y se recomienda su utilización dentro de lo posible:

- Capítulo 1: Introducción general
- Capítulo 2: Introducción específica
- Capítulo 3: Diseño e implementación
- Capítulo 4: Ensayos y resultados
- Capítulo 5: Conclusiones

Esta estructura de capítulos es la que se recomienda para las memorias de la especialización.

Figures – Esta carpeta contiene todas las figuras de la memoria. Estas son las versiones finales de las imágenes que van a ser incluidas en la memoria. Pueden ser imágenes en formato *raster*¹ como `.png`, `.jpg` o en formato vectoriales² como

¹https://en.wikipedia.org/wiki/Raster_graphics

²https://en.wikipedia.org/wiki/Vector_graphics

.pdf, .ps. Se debe notar que utilizar imágenes vectoriales disminuye notablemente el peso del documento final y acelera el tiempo de compilación por lo que es recomendable su utilización siempre que sea posible.

1.3.2. Archivos

También están incluidos varios archivos, la mayoría de ellos son de texto plano y se puede ver su contenido en un editor de texto. Después de la compilación inicial, se verá que más archivos auxiliares son creados por LaTeX o BibTeX, pero son de uso interno y no es necesario hacer nada en particular con ellos. Toda la información necesaria para compilar el documento se encuentra en los archivos **.tex**, **.bib**, **.cls** y en las imágenes de la carpeta Figures.

referencias.bib - este es un archivo importante que contiene toda la información de referencias bibliográficas que se utilizarán para las citas en la memoria en conjunto con BibTeX. Usted puede escribir las entradas bibliográficas en forma manual, aunque existen también programas de gestión de referencias que facilitan la creación y gestión de las referencias y permiten exportarlas en formato BibTeX. También hay disponibles sitios web como books.google.com que permiten obtener toda la información necesaria para una cita en formato BibTeX. Ver sección 1.7

MastersDoctoralThesis.cls – este es un archivo importante. Es el archivos con la clase que le informa a LaTeX cómo debe dar formato a la memoria. El usuario de la plantilla no debería necesitar modificar nada de este archivo.

memoria.pdf – esta es su memoria con una tipografía bellamente compuesta (en formato de archivo PDF) creado por LaTeX. Se distribuye con la plantilla y después de compilar por primera vez sin hacer ningún cambio se debería obtener una versión idéntica a este documento.

memoria.tex – este es un archivo importante. Este es el archivo que tiene que compilar LaTeX para producir la memoria como un archivo PDF. Contiene un marco de trabajo y estructuras que le indican a LaTeX cómo diagramar la memoria. Está altamente comentado para que se pueda entender qué es lo que realiza cada línea de código y por qué está incluida en ese lugar. En este archivo se debe completar la información personalizada de las primeras sección según se indica en la sección 1.5.

Archivos que *no* forman parte de la distribución de la plantilla pero que son generados por LaTeX como archivos auxiliares necesarios para la producción de la memoria.pdf son:

memoria.aux – este es un archivo auxiliar generado por LaTeX, si se borra LaTeX simplemente lo regenera cuando se compila el archivo principal **memoria.tex**.

memoria.bbl – este es un archivo auxiliar generado por BibTeX, si se borra BibTeX simplemente lo regenera cuando se compila el archivo principal **memoria.tex**. Mientras que el archivo **.bib** contiene todas las referencias que hay, este archivo **.bbl** contine sólo las referencias que han sido citadas y se utiliza para la construcción de la bibiografía.

memoria.blg – este es un archivo auxiliar generado por BibTeX, si se borra BibTeX simplemente lo regenera cuando se compila el archivo principal **memoria.tex**.

memoria.lof – este es un archivo auxiliar generado por \LaTeX , si se borra \LaTeX simplemente lo regenera cuando se compila el archivo principal **memoria.tex**. Le indica a \LaTeX cómo construir la sección *Lista de Figuras*.

memoria.log – este es un archivo auxiliar generado por \LaTeX , si se borra \LaTeX simplemente lo regenera cuando se compila el archivo principal **memoria.tex**. Contiene mensajes de \LaTeX . Si se reciben errores o advertencias durante la compilación, se guardan en este archivo **.log**.

memoria.lot – este es un archivo auxiliar generado por \LaTeX , si se borra \LaTeX simplemente lo regenera cuando se compila el archivo principal **memoria.tex**. Le indica a \LaTeX cómo construir la sección *Lista de Tablas*.

memoria.out – este es un archivo auxiliar generado por \LaTeX , si se borra \LaTeX simplemente lo regenera cuando se compila el archivo principal **memoria.tex**.

De esta larga lista de archivos, sólo aquellos con la extensión **.bib**, **.cls** y **.tex** son importantes. Los otros archivos auxiliares pueden ser ignorados o borrados ya que \LaTeX y BibTeX los regenerarán durante la compilación.

1.4. Entorno de trabajo

Ante de comenzar a editar la plantilla debemos tener un editor \LaTeX instalado en nuestra computadora. En forma análoga a lo que sucede en lenguaje C, que se puede crear y editar código con casi cualquier editor, existen ciertos entornos de trabajo que nos pueden simplificar mucho la tarea. En este sentido, se recomienda, sobre todo para los principiantes en \LaTeX la utilización de TexMaker, un programa gratuito y multi-plataforma que está disponible tanto para windows como para sistemas GNU/Linux.

La versión más reciente de TexMaker es la 4.5 y se puede descargar del siguiente link: <http://www.xmlmath.net/texmaker/download.html>. Se puede consultar el manual de usuario en el siguiente link: <http://www.xmlmath.net/texmaker/doc.html>.

1.4.1. Paquetes adicionales

Si bien durante el proceso de instalación de TexMaker, o cualquier otro editor que se haya elegido, se instalarán en el sistema los paquetes básicos necesarios para trabajar con \LaTeX , la plantilla de los trabajos de Especialización y Maestría requieren de paquete adicionales.

Se indican a continuación los comandos que se deben introducir en la consola de Ubuntu (ctrl + alt + t) para instalarlos:

```
$ sudo apt install texlive-lang-spanish texlive-science
$ sudo apt install texlive-bibtex-extra biber
$ sudo apt install texlive texlive-fonts-recommended
$ sudo apt install texlive-latex-extra
```

1.4.2. Configurando TexMaker

Una vez instalado el programa y los paquetes adicionales se debe abrir el archivo `memoria.tex` con el editor para ver una pantalla similar a la que se puede apreciar en la figura 1.1.

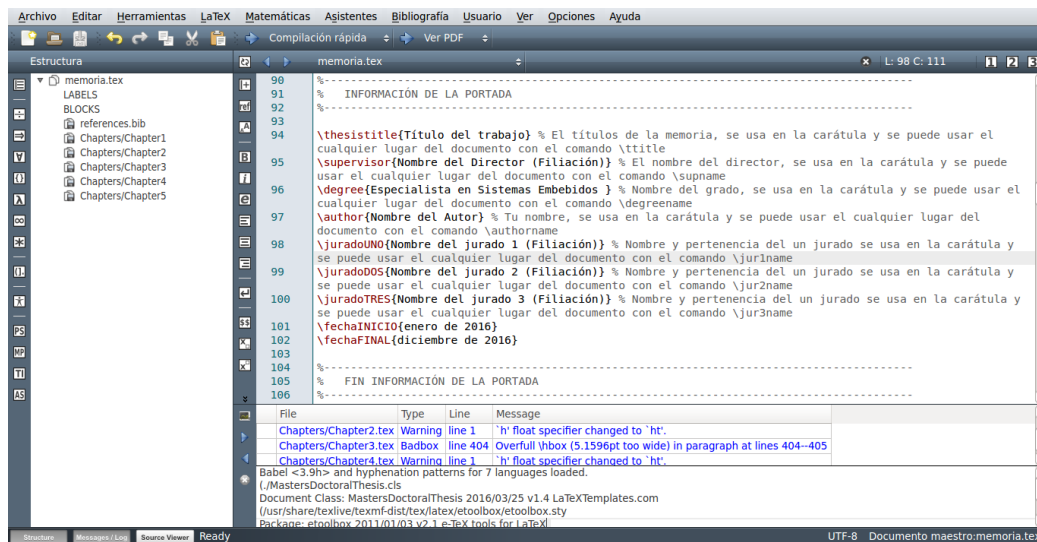


FIGURA 1.1. Entorno de trabajo de texMaker.

Notar que existe una vista llamada Estructura a la izquierda de la interfaz que nos permite abrir desde dentro del programa los archivos individuales de los capítulos. A la derecha se encuentra una vista con el archivo propiamente dicho para su edición. Hacia la parte inferior se encuentra una vista del log con información de los resultados de la compilación. En esta última vista pueden aparecer advertencias o *warning*, que normalmente pueden ser ignorados, y los errores que se indican en color rojo y deben resolverse para que se genere el PDF de salida.

Recordar que el archivo que se debe compilar con PDFLaTeX es `memoria.tex`, si se tratara de compilar alguno de los capítulos saldría un error. Para salvar la molestia de tener que cambiar de archivo para compilar cada vez que se realice una modificación en un capítulo, se puede definir el archivo `memoria.tex` como “documento maestro” yendo al menú opciones -> “definir documento actual como documento maestro”, lo que permite compilar con PDFLaTeX `memoria.tex` directamente desde cualquier archivo que se esté modificando. Se muestra esta opción en la figura 1.2.

En el menú herramientas se encuentran las opciones de compilación. Para producir un archivo PDF a partir de un archivo `.tex` se debe ejecutar PDFLaTeX (el shortcut es F6). Para incorporar nueva bibliografía se debe utilizar la opción BibTeX del mismo menú herramientas (el shortcut es F11).

Notar que para actualizar las tablas de contenidos se debe ejecutar PDFLaTeX dos veces. Esto se debe a que es necesario actualizar algunos archivos auxiliares antes de obtener el resultado final. En forma similar, para actualizar las referencias se debe ejecutar primero PDFLaTeX, después BibTeX y finalmente PDFLaTeX dos veces por idénticos motivos.

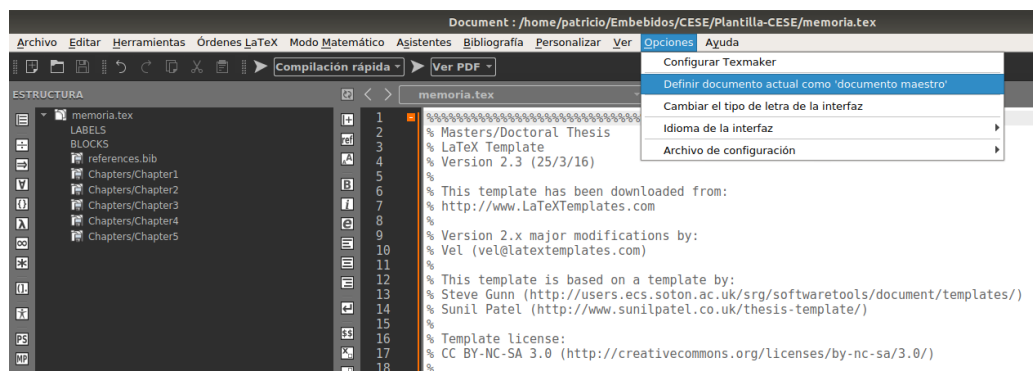


FIGURA 1.2. Definir memoria.tex como documento maestro.

1.5. Personalizando la plantilla, el archivo `memoria.tex`

Para personalizar la plantilla se debe incorporar la información propia en los distintos archivos `.tex`.

Primero abrir `memoria.tex` con TexMaker (o el editor de su preferencia). Se debe ubicar dentro del archivo el bloque de código titulado *INFORMACIÓN DE LA PORTADA* donde se deben incorporar los primeros datos personales con los que se construirá automáticamente la portada.

1.6. El código del archivo `memoria.tex` explicado

El archivo `memoria.tex` contiene la estructura del documento y es el archivo de mayor jerarquía de la memoria. Podría ser equiparable a la función `main()` de un programa en C, o mejor dicho al archivo fuente `.c` donde se encuentra definida la función `main()`.

La estructura básica de cualquier documento de \LaTeX comienza con la definición de clase del documento, es seguida por un preámbulo donde se pueden agregar funcionalidades con el uso de paquetes (equiparables a bibliotecas de C), y finalmente, termina con el cuerpo del documento, donde irá el contenido de la memoria.

```
\documentclass{article}  <- Definición de clase
\usepackage{listings}    <- Preambulo

\begin{document}         <- Comienzo del contenido propio
    Hello world!
\end{document}
```

El archivo `memoria.tex` se encuentra densamente comentado para explicar qué páginas, secciones y elementos de formato está creando el código \LaTeX en cada línea. El código está dividido en bloques con nombres en mayúsculas para que resulte evidente qué es lo que hace esa porción de código en particular. Inicialmente puede parecer que hay mucho código \LaTeX , pero es principalmente código para dar formato a la memoria por lo que no requiere intervención del usuario de la plantilla. Sí se deben personalizar con su información los bloques indicados como:

- Informacion de la memoria
- Resumen
- Agradecimientos
- Dedicatoria

El índice de contenidos, las listas de figura de tablas se generan en forma automática y no requieren intervención ni edición manual por parte del usuario de la plantilla.

En la parte final del documento se encuentran los capítulos y los apéndices. Por defecto se incluyen los 5 capítulos propuestos que se encuentran en la carpeta */Chapters*. Cada capítulo se debe escribir en un archivo *.tex* separado y se debe poner en la carpeta *Chapters* con el nombre **Chapter1**, **Chapter2**, etc... El código para incluir capítulos desde archivos externos se muestra a continuación.

```
\include{Chapters/Chapter1}
\include{Chapters/Chapter2}
\include{Chapters/Chapter3}
\include{Chapters/Chapter4}
\include{Chapters/Chapter5}
```

Los apéndices también deben escribirse en archivos *.tex* separados, que se deben ubicar dentro de la carpeta *Appendices*. Los apéndices vienen comentados por defecto con el caracter % y para incluirlos simplemente se debe eliminar dicho caracter.

Finalmente, se encuentra el código para incluir la bibliografía en el documento final. Este código tampoco debe modificarse. La metodología para trabajar las referencias bibliográficas se desarrolla en la sección 1.7.

1.7. Bibliografía

Las opciones de formato de la bibliografía se controlan a través del paquete de latex *biblatex* que se incluye en la memoria en el archivo *memoria.tex*. Estas opciones determinan cómo se generan las citas bibliográficas en el cuerpo del documento y cómo se genera la bibliografía al final de la memoria.

En el preámbulo se puede encontrar el código que incluye el paquete *biblatex*, que no requiere ninguna modificación del usuario de la plantilla, y que contiene las siguientes opciones:

```
\usepackage[backend=bibtex,
             natbib=true,
             style=numeric,
             sorting=none]
{biblatex}
```

En el archivo **reference.bib** se encuentran las referencias bibliográficas que se pueden citar en el documento. Para incorporar una nueva cita al documento lo primero es agregarla en este archivo con todos los campos necesario. Todas las entradas bibliográficas comienzan con @ y una palabra que define el formato de la entrada. Para cada formato existen campos obligatorios que deben completarse.

No importa el orden en que las entradas estén definidas en el archivo .bib. Tampoco es importante el orden en que estén definidos los campos de una entrada bibliográfica. A continuación se muestran algunos ejemplos:

```
@ARTICLE{ARTICLE:1,
  AUTHOR="John Doe",
  TITLE="Title",
  JOURNAL="Journal",
  YEAR="2017",
}

@BOOK{BOOK:1,
  AUTHOR="John Doe",
  TITLE="The Book without Title",
  PUBLISHER="Dummy Publisher",
  YEAR="2100",
}

@INBOOK{BOOK:2,
  AUTHOR="John Doe",
  TITLE="The Book without Title",
  PUBLISHER="Dummy Publisher",
  YEAR="2100",
  PAGES="100-200",
}

@MISC{WEBSITE:1,
  HOWPUBLISHED = "\url{http://example.com}",
  AUTHOR = "Intel",
  TITLE = "Example Website",
  MONTH = "12",
  YEAR = "1988",
  URLDATE = {2012-11-26}
}
```

Se debe notar que los nombres *ARTICLE:1*, *BOOK:1*, *BOOK:2* y *WEBSITE:1* son nombres de fantasía que le sirve al autor del documento para identificar la entrada. En este sentido, se podrían reemplazar por cualquier otro nombre. Tampoco es necesario poner : seguido de un número, en los ejemplos sólo se incluye como un posible estilo para identificar las entradas.

La entradas se citan en el documento con el comando:

```
\citep{nombre_de_la_entrada}
```

Y cuando se usan, se muestran así: **[ARTICLE:1]**, **[BOOK:1]**, **[BOOK:2]**, **[WEBSITE:1]**. Notar cómo se conforma la sección Bibliografía al final del documento.

Capítulo 2

Introducción específica

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

2.1. Estilo y convenciones

2.1.1. Uso de mayúscula inicial para los título de secciones

Si en el texto se hace alusión a diferentes partes del trabajo referirse a ellas como capítulo, sección o subsección según corresponda. Por ejemplo: “En el capítulo 1 se explica tal cosa”, o “En la sección 2.1 se presenta lo que sea”, o “En la subsección 2.1.2 se discute otra cosa”.

Cuando se quiere poner una lista tabulada, se hace así:

- Este es el primer elemento de la lista.
- Este es el segundo elemento de la lista.

Notar el uso de las mayúsculas y el punto al final de cada elemento.

Si se desea poner una lista numerada el formato es este:

1. Este es el primer elemento de la lista.
2. Este es el segundo elemento de la lista.

Notar el uso de las mayúsculas y el punto al final de cada elemento.

2.1.2. Este es el título de una subsección

Se recomienda no utilizar **texto en negritas** en ningún párrafo, ni tampoco texto subrayado. En cambio sí se debe utilizar *texto en itálicas* para palabras en un idioma extranjero, al menos la primera vez que aparecen en el texto. En el caso de palabras que estamos inventando se deben utilizar “comillas”, así como también para citas textuales. Por ejemplo, un *digital filter* es una especie de “selector” que permite separar ciertos componentes armónicos en particular.

La escritura debe ser impersonal. Por ejemplo, no utilizar “el diseño del firmware lo hice de acuerdo con tal principio”, sino “el firmware fue diseñado utilizando tal principio”.

El trabajo es algo que al momento de escribir la memoria se supone que ya está concluido, entonces todo lo que se refiera a hacer el trabajo se narra en tiempo pasado, porque es algo que ya ocurrió. Por ejemplo, "se diseñó el firmware empleando la técnica de test driven development".

En cambio, la memoria es algo que está vivo cada vez que el lector la lee. Por eso transcurre siempre en tiempo presente, como por ejemplo:

“En el presente capítulo se da una visión global sobre las distintas pruebas realizadas y los resultados obtenidos. Se explica el modo en que fueron llevados a cabo los test unitarios y las pruebas del sistema”.

Se recomienda no utilizar una sección de glosario sino colocar la descripción de las abreviaturas como parte del mismo cuerpo del texto. Por ejemplo, RTOS (*Real Time Operating System*, Sistema Operativo de Tiempo Real) o en caso de considerarlo apropiado mediante notas a pie de página.

Si se desea indicar alguna página web utilizar el siguiente formato de referencias bibliográficas, dónde las referencias se detallan en la sección de bibliografía de la memoria, utilizando el formato establecido por IEEE en [IEEE:citation]. Por ejemplo, “el presente trabajo se basa en la plataforma EDU-CIAA-NXP [CIAA], la cual...”.

2.1.3. Figuras

Al insertar figuras en la memoria se deben considerar determinadas pautas. Para empezar, usar siempre tipografía claramente legible. Luego, tener claro que **es incorrecto** escribir por ejemplo esto: “El diseño elegido es un cuadrado, como se ve en la siguiente figura:”



La forma correcta de utilizar una figura es con referencias cruzadas, por ejemplo: “Se eligió utilizar un cuadrado azul para el logo, como puede observarse en la figura 2.1”.

El texto de las figuras debe estar siempre en español, excepto que se decida reproducir una figura original tomada de alguna referencia. En ese caso la referencia de la cual se tomó la figura debe ser indicada en el epígrafe de la figura e incluida como una nota al pie, como se ilustra en la figura 2.2.

La figura y el epígrafe deben conformar una unidad cuyo significado principal pueda ser comprendido por el lector sin necesidad de leer el cuerpo central de la

¹<https://goo.gl/images/i7C70w>



FIGURA 2.1. Ilustración del cuadrado azul que se eligió para el diseño del logo.



FIGURA 2.2. Imagen tomada de la página oficial del procesador¹.

memoria. Para eso es necesario que el epígrafe sea todo lo detallado que corresponda y si en la figura se utilizan abreviaturas entonces aclarar su significado en el epígrafe o en la misma figura.



FIGURA 2.3. ¿Por qué de pronto aparece esta figura?

Nunca colocar una figura en el documento antes de hacer la primera referencia a ella, como se ilustra con la figura 2.3, porque sino el lector no comprenderá por qué de pronto aparece la figura en el documento, lo que distraerá su atención.

Otra posibilidad es utilizar el entorno *subfigure* para incluir más de una figura, como se puede ver en la figura 2.4. Notar que se pueden referenciar también las figuras internas individualmente de esta manera: 2.4a, 2.4b y 2.4c.



(A) Un caption.



(B) Otro.



(C) Y otro más.

FIGURA 2.4. Tres gráficos simples

El código para generar las imágenes se encuentra disponible para su reutilización en el archivo **Chapter2.tex**.

2.1.4. Tablas

Para las tablas utilizar el mismo formato que para las figuras, sólo que el epígrafe se debe colocar arriba de la tabla, como se ilustra en la tabla 2.1. Observar que sólo algunas filas van con líneas visibles y notar el uso de las negritas para los encabezados. La referencia se logra utilizando el comando `\ref{<label>}` donde label debe estar definida dentro del entorno de la tabla.

```
\begin{table}[h]
\centering
\caption[caption corto]{caption largo más descriptivo}
\begin{tabular}{l c c}
\toprule
\textbf{Especie} & \textbf{Tamaño} & \textbf{Valor}\\
\midrule
Amhiprion Ocellaris & 10 cm & \$ 6.000 \\
Hepatus Blue Tang & 15 cm & \$ 7.000 \\
Zebrasoma Xanthurus & 12 cm & \$ 6.800 \\
\bottomrule
\hline
\end{tabular}
\label{tab:peces}
\end{table}
```

TABLA 2.1. caption largo más descriptivo

| Especie | Tamaño | Valor |
|---------------------|--------|----------|
| Amhiprion Ocellaris | 10 cm | \$ 6.000 |
| Hepatus Blue Tang | 15 cm | \$ 7.000 |
| Zebrasoma Xanthurus | 12 cm | \$ 6.800 |

En cada capítulo se debe reiniciar el número de conteo de las figuras y las tablas, por ejemplo, figura 2.1 o tabla 2.1, pero no se debe reiniciar el conteo en cada sección. Por suerte la plantilla se encarga de esto por nosotros.

2.1.5. Ecuaciones

Al insertar ecuaciones en la memoria dentro de un entorno *equation*, éstas se numeran en forma automática y se pueden referir al igual que como se hace con las figuras y tablas, por ejemplo ver la ecuación 2.1.

$$ds^2 = c^2 dt^2 \left(\frac{d\sigma^2}{1 - k\sigma^2} + \sigma^2 [d\theta^2 + \sin^2 \theta d\phi^2] \right) \quad (2.1)$$

Es importante tener presente que si bien las ecuaciones pueden ser referidas por su número, también es correcto utilizar los dos puntos, como por ejemplo “la expresión matemática que describe este comportamiento es la siguiente:”

$$\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r}) \Psi = -i\hbar \frac{\partial \Psi}{\partial t} \quad (2.2)$$

Para generar la ecuación 2.1 se utilizó el siguiente código:

```
\begin{equation}
\label{eq:metric}
ds^2 = c^2 dt^2 \left( \frac{d\sigma^2}{1-k\sigma^2} + \right.
\left. \sigma^2 \left[ d\theta^2 + \sin^2\theta d\phi^2 \right] \right)
\end{equation}
```

Y para la ecuación 2.2:

```
\begin{equation}
\label{eq:schrodinger}
\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r}) \Psi =
-i\hbar \frac{\partial \Psi}{\partial t}
\end{equation}
```


Capítulo 3

Diseño e implementación

3.1. Análisis del software

La idea de esta sección es resaltar los problemas encontrados, los criterios utilizados y la justificación de las decisiones que se hayan tomado.

Se puede agregar código o pseudocódigo dentro de un entorno `lstlisting` con el siguiente código:

```
\begin{lstlisting}[caption= "un epígrafe descriptivo"]
las líneas de código irían aquí...
\end{lstlisting}
```

A modo de ejemplo:

```
1 #define MAX_SENSOR_NUMBER 3
2 #define MAX_ALARM_NUMBER 6
3 #define MAX_ACTUATOR_NUMBER 6
4
5 uint32_t sensorValue[MAX_SENSOR_NUMBER];
6 FunctionalState alarmControl[MAX_ALARM_NUMBER]; //ENABLE or DISABLE
7 state_t alarmState[MAX_ALARM_NUMBER]; //ON or OFF
8 state_t actuatorState[MAX_ACTUATOR_NUMBER]; //ON or OFF
9
10 void vControl() {
11
12     initGlobalVariables();
13
14     period = 500 ms;
15
16     while(1) {
17
18         ticks = xTaskGetTickCount();
19
20         updateSensors();
21
22         updateAlarms();
23
24         controlActuators();
25
26         vTaskDelayUntil(&ticks, period);
27     }
28 }
```

CÓDIGO 3.1. Pseudocódigo del lazo principal de control.

Capítulo 4

Ensayos y Resultados

4.1. Pruebas funcionales del hardware

La idea de esta sección es explicar cómo se hicieron los ensayos, qué resultados se obtuvieron y analizarlos.

Capítulo 5

Conclusiones

5.1. Conclusiones generales

El trabajo finaliza con el desarrollo de un sistema de monitoreo remoto, compatible con sistemas de detección que funcionen con contactos secos programables, además cuenta con la capacidad de poder establecer comunicación inalámbrica con diferentes dispositivos, en una estructura de red de nodos repetidores. Cada nodo ejecuta un protocolo de recuperación ante pérdidas de comunicación, que les permite restaurarse de forma automática ante diferentes eventos de falla, evitando en la gran mayoría de ocasiones el traslado de personal técnico al lugar de la instalación.

Este sistema fue puesto a prueba en obra para validar su correcto funcionamiento, se utilizaron dispositivos con acceso web y se comprobó que el usuario pudiese visualizar el estado actual de su sistema de seguridad de forma remota.

Se resaltan las prácticas que fueron clave para cada etapa del proyecto:

- Planificación: aunque la planificación no se ejecutó según lo planteado originalmente, se considera una herramienta indispensable, ya que permite monitorear el progreso de las diferentes actividades que integran el proyecto, así como identificar retrasos y/o tareas que requieren mayor dedicación, lo cual permite tomar las acciones necesarias en el tiempo adecuado para cumplir con los objetivos planteados.
- Diseño: utilizar un patrón de diseño, combinado con una estrategia de desarrollo de software modularizado, permitió generar una estructura de código enfocado principalmente en escalabilidad. Este diseño evolucionó en una metodología de desarrollo, fundamentado en la elaboración de pequeños bloques funcionales probados con elementos de testing. Lo cual facilitó la inclusión de los conocimientos adquiridos a lo largo de la carrera de forma organizada.
- Software: a partir de las experiencias previas con sistemas operativos, fue posible realizar dos dispositivos funcionales, basados en sistemas operativos diferentes. El dispositivo primario aprovecha la gestión de un sistema operativo de propósito general para establecer una interfaz web bidireccional, asimismo el dispositivo secundario utiliza un sistema operativo cooperativo, que brinda compatibilidad con diferentes microcontroladores y a la vez escalabilidad al proyecto.

- Hardware: un aporte invaluable para el trabajo fueron los conocimientos adquiridos en el curso de diseño de circuitos impresos, donde se realizaron cambios necesarios al diseño original, mejorando la robustez, manufactura, facilitando el futuro mantenimiento y generando documentación adecuada para el proyecto.

5.2. Próximos pasos

El sistema actual cumple con los requerimientos planteados inicialmente. Sin embargo, durante su desarrollo surgieron puntos de mejora y trabajos a desarrollar, entre los cuales se resaltan los siguientes:

- Aprovechar la conectividad WiFi del dispositivo secundario, para establecer una vía alterna para la carga del estado del sistema en el servidor web, en caso de fallas en el dispositivo primario.
- Desarrollo de una interfaz web personalizada, que complemente los avances realizados en la plataforma Node-RED.
- Implementación de un servidor que regule el acceso de los usuarios, que además permita visualizar la información desde cualquier punto con conectividad a internet.
- Elaboración de aplicaciones Android e IOS con la capacidad de generar notificaciones, ante el arribo de nuevos eventos.

El sistema actual se basa en el monitoreo de contactos secos programables, con el objetivo de mantener compatibilidad con diferentes marcas comerciales. A la vez existen segmentos del mercado con requerimientos más exigentes. Algunos posibles proyectos son:

- Hacer uso del puerto de conexión de impresoras de algunas marcas comerciales de centrales de alarma de incendio, para obtener un mayor detalle de los eventos que ocurren en la instalación.
- Diseñar una interfaz gráfica que se adecue a cada obra, facilitando al usuario la documentación de la instalación, tal como información de los equipos instalados, instrucciones ante diferentes eventos, planos con el estado de real de los dispositivos, datos históricos, próximos mantenimientos, etc.

Un aporte significativo que se espera poder alcanzar con esta serie de proyectos, consiste en establecer un sistema con la robustez y confiabilidad suficiente, para instaurar un vínculo con personal capacitado. Lo cual puede favorecer a generar una respuesta temprana ante situaciones de incendio, disminuyendo los daños ocasionados y salvaguardando de manera eficaz la vida de las personas en situaciones de emergencia.