

Sistema inteligente de iluminación de escaleras para el hogar

Ing. Pedro Santiago Escribá

Carrera de Especialización en Sistemas Embebidos

Director: Esp. Ing. Matías Nicolás Brignone (UBA)

Jurados:

Jurado 1 (pertenencia)

Jurado 2 (pertenencia)

Jurado 3 (pertenencia)

Ciudad de Córdoba, Mayo de 2026

Resumen

La presente monografía describe el trabajo realizado para el desarrollo de un sistema de iluminación de escaleras para el hogar. Detalla sus generalidades, los pasos efectuados para la ejecución del software y las pruebas de calidad del mismo. También busca que el lector pueda interiorizarse en los conceptos aplicados así como en el funcionamiento general de lo desarrollado.

Por otro lado, es objetivo de esta memoria dejar por escrito los resultados obtenidos a partir del trabajo efectuado, con la intención de que estos sean de utilidad para quienes decidan continuar con su desarrollo.

Agradecimientos

A mi familia, amigos y compañeros.

Índice general

Resumen	I
1. Introducción general	1
1.1. Introducción a los sistemas de iluminación	1
1.1.1. Iluminando una escalera moderna	1
1.1.2. Eficiencia energética	1
1.2. Motivación	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	11
2.1.5. Ecuaciones	12
3. Diseño e implementación	13
3.1. Análisis del software	13
4. Ensayos y resultados	15
4.1. Pruebas funcionales del hardware	15
5. Conclusiones	17
5.1. Conclusiones generales	17
5.2. Próximos pasos	17

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.	10
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
--	----

Capítulo 1

Introducción general

En este capítulo se introduce conceptos básicos de los sistemas de iluminación modernos y se presentan los objetivos y alcances de este proyecto.

1.1. Introducción a los sistemas de iluminación

A partir de la llegada de los LEDs (*Light Emitter Diode*) de alta potencia al mercado de la electrónica, los sistemas de iluminación del hogar (y en general) experimentaron una verdadera revolución comercial y tecnológica. Los bajos costos y la posibilidad de control de los mismos generaron un sin fin innovador que sigue vigente hoy en día. En una casa tipo es común observar lámparas LED de colores brillantes capaces de ser controladas con un teléfono celular produciendo un efecto visual poco pensado algunas décadas en el pasado o, también, su capacidad de ser encendidas o apagadas con un simple *click* programado". Sin embargo es aún un desafío para los ingenieros poder integrar estos sistemas de iluminación a la arquitectura edilicia y que los mismos sean eficientes, vistosos y de bajo costo.

1.1.1. Iluminando una escalera moderna

Uno de los mayores problemas de la iluminación de los hogares modernos es lograr un efecto lumínico vistoso sobre espacios asimétricos, con un número elevado de bordes irregulares donde las sombras juegan un desafío. Un claro ejemplo son las escaleras, donde los propios escalones proyectan sombras sobre sus predecesores o antecesores. La ingeniería en iluminación se encarga de calcular para estos casos la cantidad de lúmenes (es decir, la cantidad de luz producida por una fuente) que debe tener un cierto volumen del espacio considerando sus deformaciones y en sintonía con normas específicas. El Código Residencial Internacional recomienda 100 lúmenes por pie cuadrado (alrededor de 100 centímetros cuadrados) en sectores de interior con escaleras y cuya temperatura de color ronde entre 2700 y 3000 Kelvin.

Sin embargo, hay otro aspecto que estos códigos no contemplan del todo: la eficiencia energética. Este factor es fundamental a la hora de efectuar una obra lumínica en un hogar, ya que definirá la estrategia constructiva de la escalera.

1.1.2. Eficiencia energética

Tanto el acceso a la energía eléctrica como la cantidad de watts que se consuman en un sistema de iluminación van a generar consideraciones importantes a la hora de pensar un sistema lumínico para una escalera. Si se dispone de pocos

accesos a la energía, la oportunidad de agregar puntos de luz al espacio estará completamente limitado, afectando la capacidad de cubrir el mismo con los niveles de lúmenes adecuados. Además, deben considerarse el consumo total sobre dicho acceso energético y que la infraestructura lo soporte.

A pesar de estos inconvenientes, los avances tecnológicos han generado soluciones a partir de la invención de las tiras LED, cuyo consumo energético es alrededor de un 90 % menor a un sistema de iluminación tradicional. A grandes rasgos, estas tiras logran iluminar 100 lúmenes por vatio de energía, mientras que una bombilla incandescente ofrecen 20 lúmenes por vatio. Por otro lado, su desarrollo longitudinal permite transportar la electricidad desde un único punto de acceso energético hasta el fin del espacio a iluminar sin el adicional de cableado.

1.2. Motivación

Si bien existen múltiples sistemas de iluminación para el hogar, las escaleras son aún un territorio poco explorado y con gran versatilidad a diseños. Además, los avances de la electrónica juegan un rol fundamental para que estos diseños sean innovadores, con capacidad de control sobre el qué y el cómo se desea iluminar por parte del usuario.

La amplia gama de sensores y actuadores de bajo costo, así como los sistemas de comunicación sin cableado, son la motivación para generar un nuevo módulo lumínico transformable que responda a las necesidades de un usuario particular.

1.3. Objetivos y alcances

Este trabajo de desarrollo pretende integrar los avances e innovaciones tecnológicas en la iluminación del hogar agregando valor a bajo costo y control por parte del usuario. Es así que, integrando los conocimientos adquiridos a lo largo de la Carrera de Especialidad en Sistemas Embebidos de la Universidad de Buenos Aires, se desarrolla el Firmware en lenguaje de programación C de un sistema embebido basado en tiras LED y controlado mediante sensores de proximidad que permita iluminar una escalera de una cantidad fija de escalones. También es parte de este proyecto la incorporación de un control mediante tecnología Bluetooth de la tonalidad de la luz, brillo y color, así como su encendido o apagado de acuerdo a la circulación del usuario por la escalera.

Por otro lado, no es parte de este trabajo el desarrollo de un Hardware integral, sino que se produce un prototipo funcional utilizando componentes de fabricantes comerciales. Tampoco es parte de este proyecto el desarrollo del código de una interfaz de usuario ni brindar soporte a la misma.

1.4. Estado del arte

Antes 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 L^AT_EX, 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. 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. 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. 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.

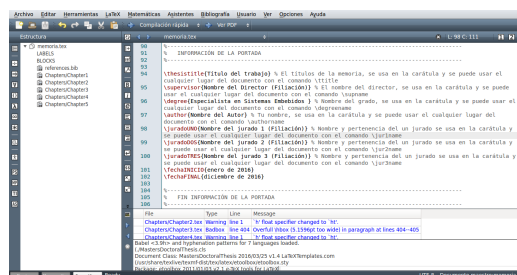


FIGURA 1.1. Entorno de trabajo de texMaker.

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.

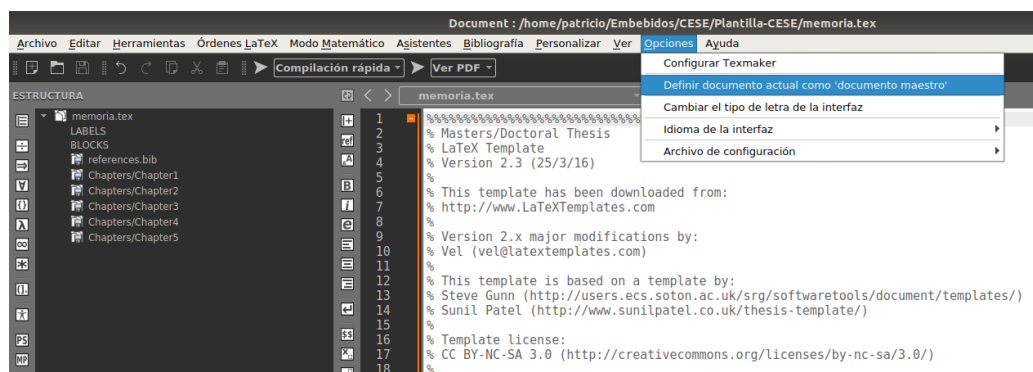


FIGURA 1.2. Definir memoria.tex como documento maestro.

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 bibliográficas se debe ejecutar primero PDFLaTeX, después BibTeX y finalmente PDFLaTeX dos veces por idénticos motivos.

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}    <- Definicion 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.

Las 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.

Finalmente y como se mencionó en la subsección 1.4.2, para actualizar las referencias bibliográficas tanto en la sección bibliografía como las citas en el cuerpo del documento, se deben ejecutar las herramientas de compilación PDFLaTeX, BibTeX, PDFLaTeX, PDFLaTeX, en ese orden. Este procedimiento debería resolver cualquier mensaje "Citation xxxxx on page x undefined".

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".



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

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

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

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 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

¹Imagen tomada de <https://goo.gl/images/i7C70w>

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
Amphiprion 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
Amphiprion 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.

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

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} +
\sigma^2 \left[ d\theta^2 +
\sin^2 \theta d\phi^2 \right] \right)
\end{equation}
```


Capítulo 3

Diseño e implementación

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.

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, se muestra el fragmento de código [3.1](#):

```
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

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.

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

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.

5.1. Conclusiones generales

La idea de esta sección es resaltar cuáles son los principales aportes del trabajo realizado y cómo se podría continuar. Debe ser especialmente breve y concisa. Es buena idea usar un listado para enumerar los logros obtenidos.

En esta sección no se deben incluir ni tablas ni gráficos.

Algunas preguntas que pueden servir para completar este capítulo:

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se pudo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

5.2. Próximos pasos

Acá se indica cómo se podría continuar el trabajo más adelante.