



CARRERA DE ESPECIALIZACIÓN EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

Emulador de microprocesador Leon3 para desarrollo de software satelital y simuladores

Autor:

Iriarte Fernandez, Nicolás Ezequiel

Director:

Nicolás Eduardo Horro (INVAP)

Jurados:

Nombre del jurado 1 (pertenencia)

Nombre del jurado 2 (pertenencia)

Nombre del jurado 3 (pertenencia)

*Este trabajo fue realizado en la ciudad de Didcot, Oxfordshire, Reino Unido,
entre marzo de 2020 y diciembre de 2020.*

Resumen

Agradecimientos

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

Índice general

Resumen	I
1. Introducción general	1
1.1. Desarrollo de simuladores espaciales y emuladores	1
1.2. Estado del arte	3
1.3. Objetivos y alcance	4
2. Introducción específica	5
2.1. Elementos componentes del sistema	5
2.2. Principio de funcionamiento	5
2.3. Requerimientos	5
3. Diseño e implementación	7
3.1. Consideraciones y decisiones técnicas	7
3.2. Diagrama de bloques	7
3.3. Arquitectura del software	7
3.4. Módulos componentes del software	7
3.5. Desarrollo del software	7
4. Ensayos y resultados	9
4.1. Ambiente de pruebas	9
4.2. Pruebas unitarias	9
4.3. Pruebas funcionales	9
4.4. Pruebas de integración	9
4.5. Caso de uso	9
5. Conclusiones	11
5.1. Conclusiones generales	11
5.2. Próximos pasos	11
6. Introducción general	15
6.1. Aprendiendo \LaTeX	15
6.1.1. Una introducción (no tan corta) a \LaTeX	15
Una subsubsección	15
6.1.2. Guía matemática rápida para \LaTeX	15
6.2. Utilizando esta plantilla	16
6.3. Qué incluye esta plantilla	16
6.3.1. Carpetas	16
6.3.2. Archivos	17
6.4. Entorno de trabajo	18
6.4.1. Paquetes adicionales	18
6.4.2. Configurando TexMaker	18
6.5. Personalizando la plantilla, el archivo <code>memoria.tex</code>	20

6.6. El código del archivo memoria.tex explicado	20
6.7. Bibliografía	21
7. Introducción específica	23
7.1. Estilo y convenciones	23
7.1.1. Uso de mayúscula inicial para los título de secciones	23
7.1.2. Este es el título de una subsección	23
7.1.3. Figuras	24
7.1.4. Tablas	25
7.1.5. Ecuaciones	26
8. Diseño e implementación	29
8.1. Análisis del software	29
9. Ensayos y resultados	31
9.1. Pruebas funcionales del hardware	31
10. Conclusiones	33
10.1. Conclusiones generales	33
10.2. Próximos pasos	33

Índice de figuras

1.1. Relación entre simuladores, emuladores y software de vuelo.	2
1.2. Tres gráficos simples	2
1.3. Logo de QEMU, tomado de su página web.	3
6.1. Entorno de trabajo de texMaker.	19
6.2. Definir memoria.tex como documento maestro.	19
7.1. Ilustración del cuadrado azul que se eligió para el diseño del logo.	24
7.2. Imagen tomada de la página oficial del procesador ¹	25
7.3. ¿Por qué de pronto aparece esta figura?	25
7.4. Tres gráficos simples	25

Índice de tablas

1.1. Tipos de emuladores	4
7.1. caption corto	26

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción general

En este capítulo se presenta una introducción general al trabajo final, en el cual se describen los antecedentes, la motivación, los objetivos y alcance del trabajo. Además, se presenta el estado del arte en emuladores de sistemas espaciales.

1.1. Desarrollo de simuladores espaciales y emuladores

Para productos de ámbito espacial, como lo son los satélites, muchas veces es difícil, y en ocasiones imposible, generar escenarios realistas para pruebas de los elementos que los componen. Ya sea por no poder generar las mismas condiciones ambientales, o porque la naturaleza de la maniobra que se busca probar implicaría un daño a los equipos bajo revisión.

Otro desafío que se presenta es la escasez de hardware disponible para realizar dichas pruebas, ya que los componentes espaciales no se suelen tener en grandes cantidades, y suelen ser caros. Haciendo que el acceso a dichos recursos sea limitado.

En este contexto, es común replicar los elementos de interés de manera programada, cumpliendo con cierto grado de representatividad. De manera tal que se comporten de la manera más similar posible a su contraparte física. Estos elementos, íntegramente desarrollados en software, se los denominan emulados o simulados. Uno de los componentes que se suele tener mayor interés en simular es el procesador de la computadora a bordo.

El término Computadora a Bordo (OBC, por las siglas en inglés de On Board Computer) suele referirse a la unidad en la que se ejecuta el Software A Bordo (OBSW, por sus siglas en inglés de On Board Software) y su rol principal es la orquestación del resto de subsistemas en el satélite. Esto incluye recolectar información de los periféricos conectados, analizarla y tomar las decisiones y acciones apropiadas cuando sea requerido.

En la Figura 1.1 se muestra la relación entre simuladores, emuladores y software de vuelo. Los simuladores satelitales son más abarcativos, incluyen todos los subsistemas simulados del satélite, entre ellos suele encontrarse el emulador de microprocesador. El emulador, por otro lado, se enfocan en reproducir el comportamiento del microprocesador específico a la misión. Por último, el software de vuelo es el software que se ejecuta en la OBC del satélite, y es el software que se busca probar en los emuladores de microprocesadores.

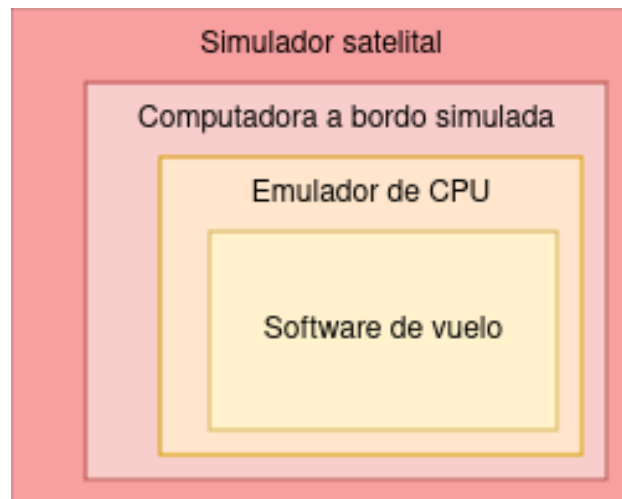
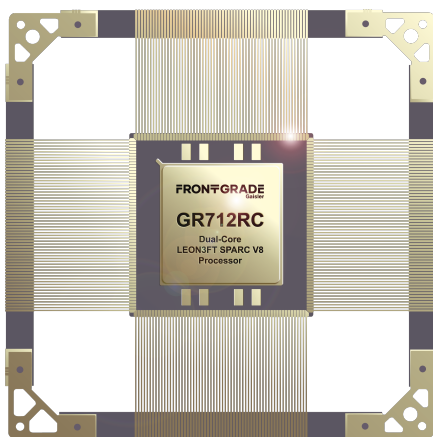


FIGURA 1.1. Relación entre simuladores, emuladores y software de vuelo.

En el caso de los satélites, la OBC suele estar compuesta por un microprocesador, memoria y periféricos, los últimos suelen variar dependiendo de la misión. El microprocesador es el encargado de ejecutar el OBSW, y es el componente que se busca emular en este trabajo. En la Figura 1.2 se muestran dos microprocesadores de uso espacial, el GR712RC (1.2a) y el UT700 (1.2b). Ambos utilizan la arquitectura SPARC V8, que es la arquitectura que se desarrolló en este trabajo.



(A) Procesador GR712RC.



(B) Procesador UT700.

FIGURA 1.2. Tres gráficos simples

Cabe aclarar que el desarrollo de un emulador de microprocesador no es trivial, ya que se debe replicar el comportamiento del microprocesador físico ciclo a ciclo, incluyendo sus instrucciones, registros, y comunicación con los periféricos. Además, se debe tener en cuenta que el microprocesador real puede tener características específicas que no se encuentran en otros microprocesadores, lo que puede dificultar la tarea de emulación.

Dado que el presente desarrollo se realiza en el marco de una carrera de grado, no se buscó replicar todas las funcionalidades del microprocesador, sino que un subconjunto de las mismas. En particular, el desarrollo se enfocó en la emulación de las instrucciones básicas del microprocesador GR712RC, omitiendo múltiples instrucciones y periféricos.

1.2. Estado del arte

En la actualidad, existen dos grandes grupos de emuladores de microprocesadores: los emuladores de microprocesadores de propósito general y los de propósito específico.

Los emuladores de propósito general son aquellos que buscan emular múltiples microprocesadores de manera genérica, es decir, sin tener en cuenta las características específicas de cada microprocesador. Estos emuladores suelen ser muy complejos y suelen tener un rendimiento inferior a los emuladores de propósito específico, ya que su implementación debe ser lo más abarcativa posible. En particular, los emuladores de propósito general suelen ser utilizados en el desarrollo de en el cual el rendimiento no es un factor crítico.

Un ejemplo de este tipo de emuladores es QEMU, un emulador de código abierto que permite emular múltiples arquitecturas de microprocesadores, incluyendo x86, ARM, SPARC y otros. QEMU es un emulador muy popular y es utilizado en múltiples proyectos de software libre y de código abierto. Teniendo una comunidad activa de desarrolladores y usuarios. Haciendo que sea una opción muy atractiva para su uso.



FIGURA 1.3. Logo de QEMU, tomado de su página web.

Por otro lado, los emuladores de propósito específico son aquellos que buscan emular un microprocesador específico. Estos emuladores suelen ser más simples que los emuladores de propósito general, ya que no buscan generalidad, sino que

buscan emular un microprocesador específico de la manera más precisa y eficiente posible. Pudiendo utilizar optimizaciones específicas que podrían romper con la compatibilidad con otros microprocesadores.

Una clara ventaja de este tipo de emuladores, además de su rendimiento, es que suelen ser más fáciles de utilizar y de integrar con otros sistemas, porque están diseñados desde un inicio para ser embebidos en otros sistemas. Lo cual es de suma importancia en el desarrollo de sistemas espaciales, ya que gran parte del desarrollo suele hacerse dentro de la misma organización.

TABLA 1.1. Comparación entre emuladores de propósito general y de propósito específico.

Emuladores específicos	Emuladores genéricos
Usualmente privativos y pagos	Solidas opciones gratis disponibles
Alto rendimiento	No enfocado en el rendimiento
Alta integrabilidad	Baja integrabilidad
Soporte limitado	Comunidades activas y código abierto

1.3. Objetivos y alcance

Capítulo 2

Introducción específica

2.1. Elementos componentes del sistema

2.2. Principio de funcionamiento

2.3. Requerimientos

Capítulo 3

Diseño e implementación

- 3.1. Consideraciones y decisiones técnicas**
- 3.2. Diagrama de bloques**
- 3.3. Arquitectura del software**
- 3.4. Modulos componentes del software**
- 3.5. Desarrollo del software**

Capítulo 4

Ensayos y resultados

- 4.1. Ambiente de pruebas
- 4.2. Pruebas unitarias
- 4.3. Pruebas funcionales
- 4.4. Pruebas de integración
- 4.5. Caso de uso

Capítulo 5

Conclusiones

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.

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.

*Esta página ha sido intencionalmente dejada en blanco y no forma parte de la memoria.
Es unicamente para mantener los capítulos de ejemplo.*

*Esta página ha sido intencionalmente dejada en blanco y no forma parte de la memoria.
Es unicamente para mantener los capítulos de ejemplo.*

Capítulo 6

Introducción general

6.1. Aprendiendo \LaTeX

\LaTeX 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 \LaTeX es en realidad un archivo de texto simple o llano que *no contiene formato*. Nosotros le decimos a \LaTeX 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 \LaTeX es un lenguaje del tipo «mark-up», muy parecido a HTML.

6.1.1. Una introducción (no tan corta) a \LaTeX

Si sos nuevo en \LaTeX , hay un muy buen libro electrónico - disponible gratuitamente en Internet como un archivo PDF - llamado, «A (not so short) Introduction to \LaTeX ». 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.

6.1.2. Guía matemática rápida para \LaTeX

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 \LaTeX ». 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.

6.2. 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 6.5 en la página 20.

Debés asegurarte de leer el capítulo 7 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.

6.3. Qué incluye esta plantilla

6.3.1. 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/patricibos/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 `.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.

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

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

6.3.2. 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 6.7

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 6.5.

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.

6.4. 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>.

6.4.1. 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
```

6.4.2. 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 6.1. 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 6.1. 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 6.1. 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 6.1.

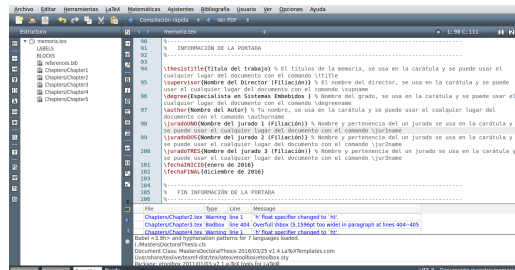


FIGURA 6.1. Entorno de trabajo de texMaker.

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 6.2.

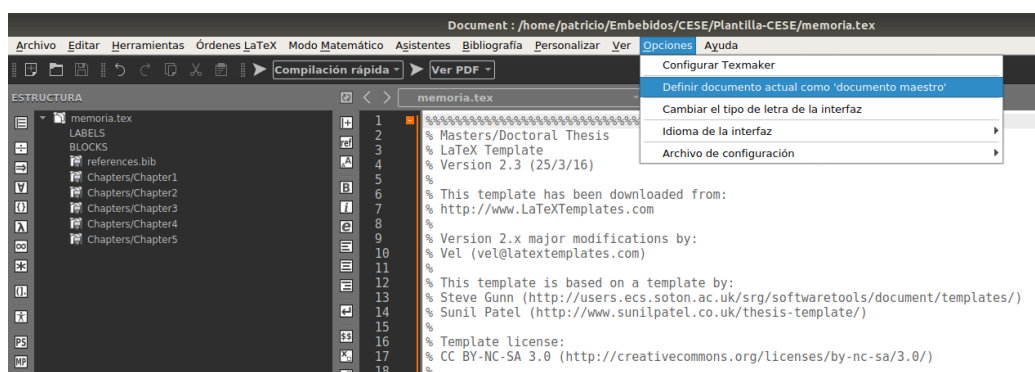


FIGURA 6.2. Definir `memoria.tex` como documento maestro.

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.

6.5. 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.

6.6. 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 6.7.

6.7. 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",
  URLLDATE = {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 6.4.2, 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 `Çitation xxxxx on page x undefined`.

Capítulo 7

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.

7.1. Estilo y convenciones

7.1.1. 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 6 se explica tal cosa”, o “En la sección 7.1 se presenta lo que sea”, o “En la subsección 7.1.2 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.

7.1.2. 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...".

7.1.3. 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 7.1".



FIGURA 7.1. 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 7.2. 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 7.2.

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 7.3. ¿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 7.3, 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 7.4. Notar que se pueden referenciar también las figuras internas individualmente de esta manera: 7.4a, 7.4b y 7.4c.



(A) Un caption.



(B) Otro.



(C) Y otro más.

FIGURA 7.4. Tres gráficos simples

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

7.1.4. 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 7.1. 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 7.1. 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.

7.1.5. 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 7.1.

$$ds^2 = c^2 dt^2 \left(\frac{d\sigma^2}{1 - k\sigma^2} + \sigma^2 [d\theta^2 + \sin^2 \theta d\phi^2] \right) \quad (7.1)$$

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:”

$$\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r}) \Psi = -i\hbar \frac{\partial \Psi}{\partial t} \quad (7.2)$$

Para generar la ecuación 7.1 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}

```

Y para la ecuación 7.2:

```

\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 8

Diseño e implementación

8.1. 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 8.1. Pseudocódigo del lazo principal de control.

Capítulo 9

Ensayos y resultados

9.1. Pruebas funcionales del hardware

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

Capítulo 10

Conclusiones

10.1. 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.

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?

10.2. Próximos pasos

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