

Tesis de la Licenciatura en Ciencias de la computación

Clase y plantilla en L^AT_EX para realizar una tesis de la Licenciatura en Ciencias de la Computación

2016





UNIVERSIDAD DE SONORA
División de Ciencias Exactas y Naturales
Departamento de Matemáticas

Clase y plantilla en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ para realizar una tesis de la Licenciatura en Ciencias de la Computación

Tesis que para obtener el título de
Licenciado en Ciencias de la Computación

presenta

Julio Waissman Vilanova

Director de tesis: Zutano Funáñez Mengano.
Hermosillo, Sonora, 15 de febrero de 2016.

Resumen

En este apartado va un breve resumen del trabajo de tesis, de preferencia en media página en español y media página en inglés. Si no fuera el caso, pues una página a lo más en español. Es muy importante notar que el resumen no es una introducción. En el resumen no debe incluirse referencias bibliográficas ni listas. Mucha gente solo lee el resumen de un trabajo, así que es muy importante hacerlo muy cuidadosamente y que invite al que lo lee a seguir con el resto de la tesis. Esto es lo que se hace normalmente al último, cuando el resto de la tesis está acabada.

A la Virgencita de Guadalupe y el Santo Enmascarado de Plata

Agradecimientos

Esta sección se realiza al último, y una vez que se haya liberado la tesis para su impresión. Los agradecimientos son completamente libres y pueden ser tan sentimentales, graciosos o profundos como se quiera. A diferencia de la dedicatoria, que debe caber en una sola línea, aquí se le puede agradecer hasta al perro si se les da la gana. Los agradecimientos no los revisa el director de tesis y por lo tanto no puede exigir que se incluyan agradecimientos a una u otra persona en particular.

Igualmente, es importante resaltar que los agradecimientos y la dedicatoria son completamente accesorios y si no se tiene la intención de agradecer a nadie (o no se quiere dedicar el trabajo a alguien), pues simplemente se eliminan y no hay ningún problema.

Índice general

Introducción	1
Descripción del trabajo	1
Aportaciones principales	1
Organización del trabajo	2
1. Un primer capítulo	3
1.1. Introducción	3
1.2. ¿Como se organiza una tesis?	3
1.2.1. Numero de capítulos	3
1.2.2. ¿Y la cantidad de páginas de una tesis?	4
1.2.3. ¿Y como organizar el trabajo?	4
1.3. Escribir un trabajo en L ^A T _E X	5
1.3.1. Inserción de figuras	5
1.3.2. Inserción de cuadros	7
1.3.3. Escribiendo matemáticas	7
1.4. Conclusiones	8
2. Otro capítulo de ejemplo	9
2.1. Introducción	9
2.2. Que poner	9
2.2.1. Numero de capítulos	9
2.2.2. Otras cosas	9
2.2.3. Mas cosas pa poner	10

2.3. Conclusiones	10
3. Citas y algoritmos	11
3.1. Introducción	11
3.2. Citas textuales y referencias	11
3.2.1. A figuras y cuadros	11
3.2.2. A la bibliografía	11
3.2.3. Múltiples referencias	12
3.3. Códigos	12
3.3.1. Seudocódigos	12
3.3.2. Códigos fuente	12
Conclusiones	15
A. Información adicional	17
A.1. ¿Porque poner apéndices?	17
A.2. Otros casos	17
Bibliografía	25

Introducción

Descripción del trabajo

En la introducción es muy importante dejar claro cual era la problemática a tratar, la justificación de dicho trabajo, los objetivos (general y específicos) así como las metas. También es importante acotar los alcances del trabajo. Esta información ya se tiene del proyecto de tesis que se presentó antes de la realización de la tesis, aunque en muchos casos es necesario adaptarla a lo que se hizo realmente.

En un trabajo de tesis no es extraño empezar con un objetivo inicial y conforme se desarrolla el trabajo derivar a otro problema o acotar el problema original. Si este es el caso, es conveniente dejar claro en la introducción de la tesis los objetivos, metas, justificación y descripción *del trabajo que se está presentando*.

Este capítulo no es muy largo y se escribe más fácilmente después de escribir los capítulos del cuerpo de la tesis. Las secciones siguientes (aportaciones principales y organización del trabajo) suelen ser muy cortas (un párrafo o dos).

Aportaciones principales

Es una buena idea dejar en claro cuales son las aportaciones del trabajo desde el principio. En este apartado se puede especificar cuales son los trabajos, programas, publicaciones, prototipos o revisiones realizadas. Así, se facilita la tarea del comité revisor y facilita la lectura para quienes estén interesados en leer el trabajo.

Recuerda que es una tesis de licenciatura, y el hecho de estudiar un artículo (o una técnica no vista durante la carrera), desmenuzarlo, entenderlo, explicarlo y/o programarlo

es suficiente. En una tesis de licenciatura no se espera que se realice una aportación original.

Organización del trabajo

Facilita la lectura del trabajo describir que es lo que se presenta en el capítulo uno, capítulo dos, etc. así como cuáles son los apéndices que se agregan al trabajo de tesis.

Un primer capítulo

1.1. Introducción

Es una buena costumbre comenzar cada capítulo con una sección de introducción donde se describa lo que se va a mostrar en este capítulo. Es más fácil escribir esta sección después de haber escrito el capítulo completo. En este capítulo se introducen las ideas generales para escribir un trabajo de tesis y su organización, pensando en un trabajo de la Licenciatura en Ciencias de la Computación de la Universidad de Sonora.

1.2. ¿Como se organiza una tesis?

1.2.1. Numero de capítulos

¿Cuántos capítulos son necesarios para una tesis? Esto varía mucho dependiendo de la forma en que está organizado el trabajo, en común acuerdo con el director de tesis. Normalmente es mejor poner uno o dos capítulos que describan el estado del arte, las técnicas o las bases teóricas del trabajo a realizar, y que se utilizarán en el resto de la tesis. En estos capítulos no se presentan ideas nuevas ni el desarrollo que más costó trabajo, solo las bases necesarias para entender el trabajo realizado. Se puede considerar una especie de revisión bibliográfica y suelen tener la mayor parte de las referencias bibliográficas de todo el trabajo escrito.

En los capítulos subsecuentes, lo normal es ser muy específico y desarrollar con detalle el trabajo realizado, tanto si es un desarrollo teórico, un desarrollo tecnológico, como si es

simulación o experimentación real. Estos capítulos suelen ser relativamente cortos y con muchas imágenes, gráficas y tablas en el caso de trabajos aplicados.

Es común que una tesis de licenciatura tenga 3 capítulos, sin contar el capítulo de introducción y las conclusiones generales. Es importante resaltar que esto depende mucho de la naturaleza del trabajo desarrollado y debe ser establecido de antemano en común acuerdo entre el director de tesis y el estudiante. Igualmente, cabe resaltar que es normal, durante el desarrollo de la tesis (y más específicamente durante su redacción) que se considere agregar y/o quitar capítulos sobre el proyecto original.

1.2.2. ¿Y la cantidad de páginas de una tesis?

¿Cuántas cuartillas debe tener una tesis? Es una pregunta muy normal que se hacen todo mundo cuando comienza la fase de redacción. Las páginas correctas son *las que sean necesarias para explicar el trabajo desarrollado*. Este número cambia mucho dependiendo de la disciplina. Por ejemplo, los trabajos en computación teóricas suelen ser muy cortos (unas 30 páginas para una tesis de licenciatura), mientras los trabajos enfocados a ingeniería de *software* tienden a ser muy extensos (de 80 a 100 páginas).

Es recomendable hacer un trabajo breve, normalmente entre 30 y 50 páginas. Sin embargo, no existe una limitante, si el trabajo, por su naturaleza, requiere que sea explicado en más espacio, o naturalmente se tiende a escribir mucho. Se puede extender el trabajo de tesis *hasta 100 páginas como máximo*. Es importante tomar en cuenta que en muchas Universidades del mundo, se tiende a limitar los trabajos de tesis de doctorado a 150 páginas como máximo con todo y apéndices. Un trabajo muy largo no significa que sea mejor.

1.2.3. ¿Y como organizar el trabajo?

Para organizar el trabajo, L^AT_EX contempla el uso de `\section`, `\subsection`, `\subsubsection` y `\paragraph`. Si bien en este capítulo se hace uso de la mayoría de estos bloques con fines ilustrativos, en general es mejor tratar de mantener la organización del trabajo hasta `\subsection`. Considere el uso de listas, enumeraciones y descripciones si requiere escribir muchos párrafos cortos con títulos en `\subsubsection` o `\paragraph`. Igualmente, evite el uso excesivo de listas en un trabajo, ya que esto solo denota una pobre capacidad para expresar ideas en párrafos de texto.

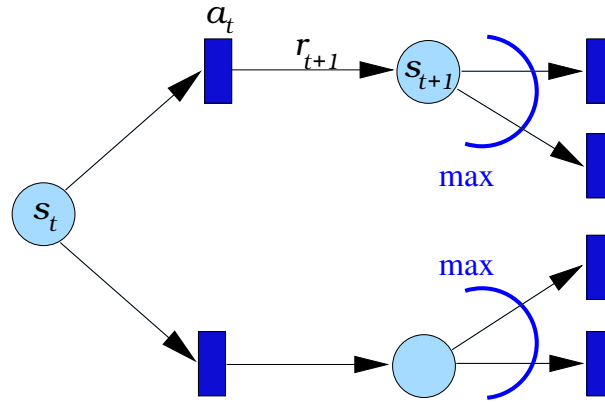


Figura 1.1: Ejemplo de inserción de una imagen en PDF

1.3. Escribir un trabajo en L^AT_EX

Para realizar un trabajo científico en el área de Ciencias de la Computación, especialmente en las ramas de computación teórica, computo científico e inteligencia artificial, es normal utilizar L^AT_EX. Se asume que es normal que un estudiante que está preparando una tesis de licenciatura en Ciencias de la Computación tenga experiencia previa en escribir trabajos más pequeños en L^AT_EX.

Un buen comienzo para escribir un trabajo en L^AT_EX, es buscar en otros trabajos un ejemplo de algo similar a lo que queremos hacer, y luego adaptarlo a las necesidades particulares. Por esta razón, en esta sección se incluyen ejemplos para la inserción de figuras, cuadros, listas y descripciones. Utilice estas plantillas, pero tenga cuidado en su uso. Un problema muy común es el copiar la plantilla para la inserción de figuras o cuadros, sin cambiar el comando `\label`, por lo que a la hora de compilar, se hace referencia en todo el documento a una sola imagen.

1.3.1. Inserción de figuras

Las imágenes soportadas en la clase (y compiladas con `pdflatex`) son PNG, JPG, GIF y PDF de preferencia. Evitar utilizar imágenes de tipo EPS. La figura 1.1 es un ejemplo de inserción de un gráfico en PDF. La figura 1.2 y la figura 1.3 son ejemplo para insertar imágenes en PNG y JPG respectivamente. La leyenda de las figuras *siempre* se encuentra por abajo de la figura o imagen.

La colocación de una figura se puede controlar con los comandos `h,t,b,p` (de *he-*

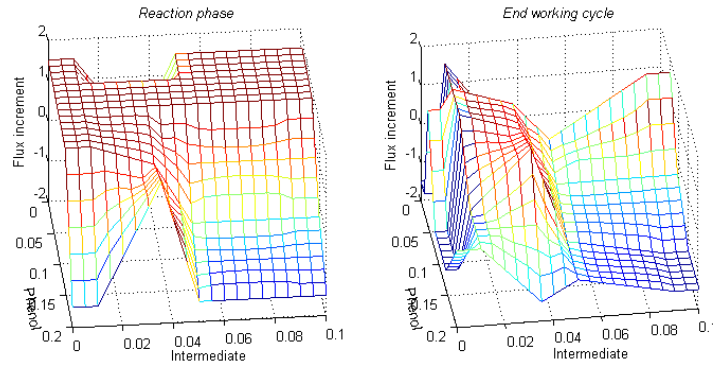


Figura 1.2: La leyenda de la figura siempre va abajo de ésta

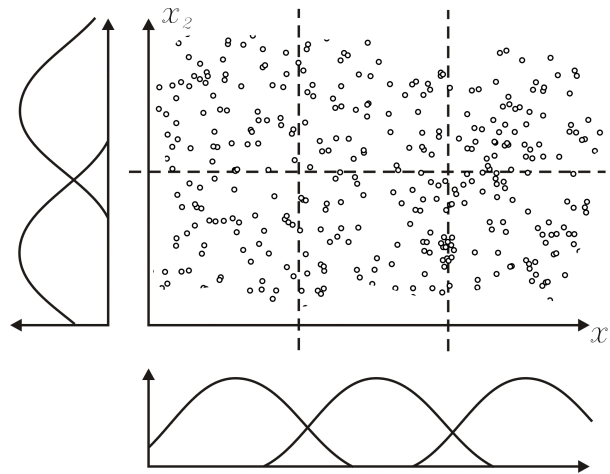


Figura 1.3: Procure que la leyenda explique claramente el título de la figura, aunque sea necesario que sea una leyenda larga que ocupe varias líneas

re,top,bottom,page) utilizando el orden como prioridad. Si por alguna extraña razón se desea colocar una figura en un lugar específico del texto, se coloca con el comando `H` (*here* pero mandatorio). En general (y esto es tanto para las tablas como para las figuras) no debe mucho preocupar si las figuras se colocan en páginas diferentes que en las que se hace referencia de ellas.

Editorialmente esto es válido y recomendable. L^AT_EX lo que intenta es colocar las figuras siempre arriba de una página y evitar que el porcentaje de texto en una página sea menor al 30 % de su contenido. En la clase de tesis `tesis1cc` se amplió el margen para poder colocar figuras mas grandes, o varias figuras en una página.

Es muy importante que se haga referencia de todas las figuras en el texto y se discuta sobre ellas, así sea sólo una línea de texto. Si no es posible hacer referencia de una figura

Cuadro 1.1: Tabla de muestra para la tesis

Texto	Formulas	Numero
Cosa	$\sqrt{x^2 + b^2}$	3.1416
Pasto	$3.2 + e^{-t}$	4578

en el texto, significa que no es necesario que se encuentre en el documento.

Otra práctica común y que suele ayudar mucho, sobre todo cuando la tesis se encuentra bastante avanzada o cuando se piden cambios por parte de los revisores, es el utilizar uno o varios directorios en los cuales se guarden las figuras. Esto es especialmente importante en trabajos experimentales o de desarrollo tecnológico que típicamente llevan varias capturas de pantallas o muchas gráficas. En este ejemplo, las figuras se encuentran agrupadas en un subdirectorio `figuras` (haciendo gala de falta de imaginación).

1.3.2. Inserción de cuadros

Por abuso del idioma, tendemos a decirle en español tablas a lo que de manera correcta deberíamos nombrar como *cuadros*. Esto se debe a que en inglés se les conoce como *tables*. La leyenda de los cuadros *siempre* se encuentra arriba del cuadro (al revés que en las figuras). El cuadro 1.1 es un ejemplo típico de cuadro. Es importante siempre citar y explicar un cuadro en el texto, de otra manera significa que no es una información necesaria.

Todas las referencias (a figuras, cuadros, secciones, etc.) se deben hacer utilizando los comandos `label` (para marcar un entorno) y `ref` (para citarlo en el texto). En documentos muy grandes esto es muy importante, ya que si se elimina o se agrega un cuadro o figura, no es necesario reescribir todo el texto y revisar que se haga referencia correcta de ellos en el texto.

1.3.3. Escribiendo matemáticas

Todos los estudiantes que realizan una tesis y requieren de formalismo matemático, en general escogen \LaTeX como entorno de edición preferencialmente, ya que la calidad tipográfica, como la facilidad para incluir notación matemática en el texto, hacen que, en términos generales, sea mucho más fácil escribir una tesis en \LaTeX que en un entorno tipo WYSIWYG como *World*. En esta sección se asume que el lector se encuentra familiarizado

con el uso de \LaTeX y el uso de notación matemática, por lo que solamente se realizarán pequeñas recomendaciones.

1.4. Conclusiones

En general, no es nada mala idea poner una última sección con conclusiones de lo que se mostró en el capítulo. Para los revisores y/o lectores del trabajo, les facilita mucho realizar una primer hojeada al documento.

Otro capítulo de ejemplo

2.1. Introducción

En todo trabajo es normal comenzar cada capítulo con una sección de introducción donde se describa lo que se va a mostrar en este capítulo. Como en la tesis, es más fácil hacer esta sección al final del capítulo.

2.2. Que poner

2.2.1. Numero de capítulos

Los capítulos que sean necesarios. Normalmente es mejor poner capítulos que describan el estado del arte, y las bases de lo que se usó en el desarrollo de la tesis. En estos capítulos, no se presentan ideas nuevas o desarrollos propios, solo lo que ya estaba. Es una especie de revisión bibliográfica.

En los capítulos subsecuentes, lo normal es ser muy específico y desarrollar con detalle el trabajo realizado, tanto si es desarrollo tecnológico como si es en simulación o en experimentación real. Estos capítulos suelen ser relativamente cortos y con muchas imágenes, gráficas y tablas.

2.2.2. Otras cosas

Para poner cosas es bueno basarse en formatos ya preestablecidos. La notación de las tablas va antes de la tabla, mientras que la de las figuras va abajo. Siempre que se muestra

Cuadro 2.1: Tabla de muestra para la tesis

Texto	Formulas	Numero
Cosa	$\sqrt{x^2 + b^2}$	3.1416
Pasto	$3.2 + e^{-t}$	4578

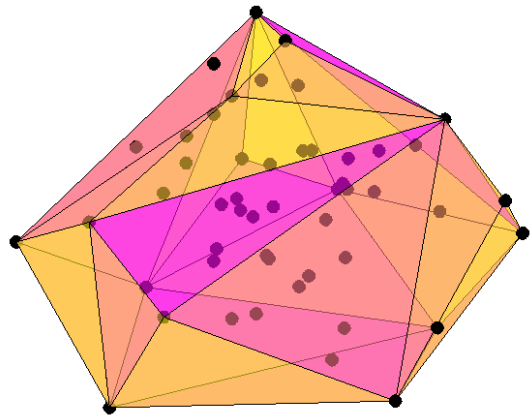


Figura 2.1: Ejemplo de una figura pa la tesis

una figura o una tabla hay que comentarla en el texto tal como la tabla (2.1)

2.2.3. Mas cosas pa poner

Pues en este caso solo se agrega una figura. Es importante ver que para que el documento se mantenga ordenado se ponen las figuras en un subdirectorio figuras y se llaman desde ahı.

y bueno pues con esto creo que vemos en general como hacerle en grandes lineas.

2.3. Conclusiones

En algunos casos no es nada mala idea poner una ultima seccion con conclusiones de lo que se mostro en el capıtulo.

Citas y algoritmos

3.1. Introducción

En este capítulo se abordará la manera agregar citas de la bibliografía; referencias a figuras y cuadros; algoritmos.

3.2. Citas textuales y referencias

En el documento se puede hacer referencia tanto a *labels* (etiquetas) o entradas en la bibliografía.

3.2.1. A figuras y cuadros

En el caso de las referencias a figuras o cuadros se utiliza el comando `\cref{label}`, en donde `label` es el nombre del identificador de la figura o tabla. Por ejemplo, en el cuadro 2.1 podemos observar que se ven unas fórmulas matemáticas chistosas.

3.2.2. A la bibliografía

En el caso de las referencias a entradas de la bibliografía se utiliza el comando `\cite{label}`, en donde `label` es el nombre de la entrada de la referencia bibliográfica. Por ejemplo, el

artículo de Sachdeva [?] presenta una novedosa y arcana técnica para la regularización de los *chakras*.

3.2.3. Múltiples referencias

En ocasiones es deseable referenciar a dos o mas etiquetas o entradas de la bibliografía, en dicho caso se pueden enlistar los argumentos de los comandos `\cref` y `\cite` separados por comas. Por ejemplo, dos trabajos de tesis que fueron inspiración para la elaboración de las figuras 1.1 y 1.2 fueron [?, ?].

3.3. Códigos

3.3.1. Seudocódigos

Para escribir algoritmos en pseudocódigo con palabras clave en español podemos utilizar el entorno `algorithm` y `algorithmic`, el primero nos permite etiquetar y titular el algoritmo, mientras que el segundo es el entorno en donde se escribe el pseudocódigo. Por ejemplo, el algoritmo algoritmo 3.1 describe la operación de potenciación.

Podemos tomar el ejemplo del algoritmo *quicksort* del Cormen, el cual tiene un tiempo de ejecución en el peor de los casos de $\Theta(n^2)$ y en el caso promedio $\Theta(n \lg n)$ sobre un arreglo de entrada de n números y escribirlo de manera similar a como aparece en el libro algoritmo 3.2.

3.3.2. Códigos fuente

Si el código que deseamos mostrar está escrito en algún lenguaje de programación en particular, utilizamos los entornos `listing` y `minted`. Por ejemplo, el listado 1 es un ejemplo de código en el lenguaje *Ruby*.

Algoritmo 3.1 Calcula $y = x^n$

Entrada: $n \geq 0 \vee x \neq 0$ **Salida:** $y = x^n$

```

 $y \leftarrow 1$ 
si  $n < 0$  entonces
     $X \leftarrow 1/x$ 
     $N \leftarrow -n$ 
si no
     $X \leftarrow x$ 
     $N \leftarrow n$ 
fin si
mientras  $N \neq 0$  hacer
    si  $N$  es par entonces
         $X \leftarrow X \times X$ 
         $N \leftarrow N/2$ 
    si no  $\triangleright N$  es impar
         $y \leftarrow y \times X$ 
         $N \leftarrow N - 1$ 
    fin si
fin mientras

```

Algoritmo 3.2 Ordenamiento *quicksort*

QUICKSORT (A, p, r)

```

1: si  $p < r$  entonces
2:    $q \leftarrow \text{PARTITION}(A, p, r)$ 
3:   QUICKSORT ( $A, p, q - 1$ )
4:   QUICKSORT ( $A, q + 1, r$ )
5: fin si

```

PARTITION (A, p, r)

```

1:  $x \leftarrow A[r]$ 
2:  $i \leftarrow p - 1$ 
3: para  $j \leftarrow p$  hasta  $r - 1$  hacer
4:   si  $A[j] \leq x$  entonces
5:      $i \leftarrow i + 1$ 
6:     intercambia  $A[i]$  con  $A[j]$ 
7:   fin si
8: fin para
9: intercambia  $A[i + 1]$  con  $A[r]$ 
10: regresar  $i + 1$ 

```

```
lazy_integers = (1..Float::INFINITY).lazy
lazy_integers.collect { |x| x ** 2 }.
  select { |x| x.even? }.
  reject { |x| x < 1000 }.
  first(5)
```

Código 1: Ejemplo de código en Ruby

El programa descrito en el listado 2 es todo un clásico en el mundo de la programación.

```
#include <stdio.h>

main()
{
    printf("hello, world\n");
}
```

Código 2: Clásico *hola mundo* en el lenguaje *C*.

Conclusiones

En las conclusiones no se separan por secciones, pero deben llevar tres elementos básicos:

1. Un breve resumen del trabajo realizado
2. Un análisis critico de las ventajas y desventajas del trabajo realizado, procurando mostrar, al realizar una critica razonada de lo realizado, que se tiene un dominio del tema.
3. Trabajos futuros derivados de lo realizado en esta tesis. Muchos de estos trabajos van a ser la solucion propuesta a los puntos débiles del trabajo realizado

Las conclusiones pueden llevar más de una página, pero no es normal encontrar conclusiones con una extensión mayor a las 3 páginas. En este capitulo es muy importante ser muy claro y conciso. Típicamente se realiza al final del trabajo y antes dehacer la introducción y el resumen.

Información adicional

A.1. ¿Porque poner apéndices?

Hay cosas en una tesis que costaron mucho trabajo, que son interesantes de mostrar para simplificar el trabajo de otras personas que continúen con el trabajo realizado, pero que son áridos de leer o no aportan mucho en el desarrollo del tema principal. En ese caso los apéndices es la solución ideal para mostrarlos.

Este es el caso de los programas (código fuente) o bases matemáticas de alguna técnica, en donde la aplicación es lo importante en el trabajo de tesis. Por favor, incluye código fuente solo si es estrictamente necesario, o si por la naturaleza del trabajo no se puede explicar con pseudocódigo. Es una tesis en Ciencias de la Computación y se asume que todo el que la lee sabe programar. Un caso particular donde el código es importante es cuando el trabajo se trata del uso de una plataforma o una técnica muy específica (como podría ser *Jess*, o el uso de librerías para *MPI*).

A.2. Otros casos

Otros casos importantes es donde es necesario poner a disposición la información utilizada de otra fuente, tal como las bases de datos o las bibliotecas de funciones realizadas por otras personas. Es común en algunas tesis de agregar un último apéndice con los artículos que se desarrollaron durante la tesis.

Índice de figuras

1.1. Ejemplo de inserción de una imagen en PDF	5
1.2. La leyenda de la figura siempre va abajo de ésta	6
1.3. Procure que la leyenda explique claramente el título de la figura, aunque sea necesario que sea una leyenda larga que ocupe varias líneas	6
2.1. Ejemplo de una figura pa la tesis	10

Índice de cuadros

1.1. Tabla de muestra para la tesis	7
2.1. Tabla de muestra para la tesis	10

Índice de Algoritmos

3.1. Calcula $y = x^n$	13
3.2. Ordenamiento <i>quicksort</i>	13

Índice de códigos

1.	Ejemplo de código en Ruby	12
2.	Clásico <i>hola mundo</i> en el lenguaje <i>C</i>	14