

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE ELECTRÓNICA



Tesis Doctoral

**Plantilla unificada para la generación de memorias de
PFCs/TFGs/TFMs y tesis doctorales**

Javier Macías-Guarasa

2014

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE ELECTRÓNICA



Plantilla unificada para la generación de memorias de
PFCs/TFGs/TFMs y tesis doctorales

Autor

Javier Macías-Guarasa

Director

Roberto Barra Chicote

2014

Tesis Doctoral

A nuestros alumnos pasados, presentes y futuros...

“Empieza haciendo lo necesario, luego haz lo posible y de pronto empezarás a hacer lo imposible.”

Francisco de Asís

Agradecimientos

A todos los que la presente vieron y entendieron.

Inicio de las Leyes Orgánicas. Juan Carlos I

Este trabajo es el fruto de muchas horas de trabajo, tanto de los autores últimos de los ficheros de la distribución como de todos los que en mayor o menor medida han participado en él a lo largo de su proceso de gestación.

Mención especial merece Manuel Ocaña, el autor de la primera versión de las plantillas de proyectos fin de carrera y tesis doctorales usadas en el Departamento de Electrónica de la Universidad de Alcalá, con contribuciones de Jesús Nuevo, Pedro Revenga, Fernando Herránz y Noelia Hernández.

En la versión actual, la mayor parte de las definiciones de estilos de partida proceden de la tesis doctoral de Roberto Barra-Chicote, con lo que gracias muy especiales para él.

También damos las gracias a Manuel Villaverde, David Casillas y Jesús Pablo Martínez, que nos han proporcionado secciones completas y ejemplos puntuales de sus proyectos fin de carrera.

Finalmente, hay incontables contribuyentes a esta plantilla, la mayoría encontrados gracias a la magia del buscador de Google. Hemos intentado referenciar los más importantes en los fuentes de la plantilla, aunque seguro que hemos omitido alguno. Desde aquí les damos las gracias a todos ellos por compartir su saber con el mundo.

Resumen

Este documento ha sido generado con una plantilla para memorias de trabajos fin de carrera, fin de máster, fin de grado y tesis doctorales. Está especialmente pensado para su uso en la Universidad de Alcalá, pero debería ser fácilmente extensible y adaptable a otros casos de uso. En su contenido se incluyen las instrucciones generales para usarlo, así como algunos ejemplos de elementos que pueden ser de utilidad. Si tenéis problemas, sugerencias o comentarios sobre el mismo, dirigidlas por favor a Javier Macías-Guarasa <macias@depeca.uah.es>.

Palabras clave: Plantillas de trabajos fin de carrera/máster/grado y tesis doctorales, L^AT_EX, soporte de español e inglés, generación automática.

Abstract

This document has been generated with a template for Bsc and Msc Thesis (trabajos fin de carrera, fin de máster, fin de grado) and PhD. Thesis, specially thought for its use in Universidad de Alcalá, although it should be easily extended and adapted for other use cases. In its content we include general instructions of use, and some example of elements than can be useful. If you have problemas, suggestions or comments on the template, please forward them to Javier Macías-Guarasa <macias@depeca.uah.es>.

Keywords: Bsc., Msc. and PhD. Thesis template, L^AT_EX, English/Spanish support, automatic generation.

Índice general

Resumen	ix
Abstract	xi
Índice general	xiii
Índice de figuras	xv
Índice de tablas	xvii
Índice de listados de código fuente	xix
Índice de algoritmos	xxi
Lista de acrónimos	xxiii
Lista de símbolos	xxv
1 Introducción	1
1.1 Presentación	1
1.2 Uso de la plantilla	1
1.2.1 Prerrequisitos	1
1.2.2 Compilación	2
1.2.3 Estructura del documento generado por la plantilla	3
1.2.4 Definiciones específicas del tipo de documento	4
1.2.5 Plantilla de anteproyecto	6
1.2.6 Papeleo adicional para la defensa de los TFGs	6
1.2.7 Problemas conocidos	7
1.3 Ejemplos de elementos de utilidad	7
1.3.1 Uso de comandos definidos	7
1.3.2 Uso de “frases célebres”	7
1.3.3 Inclusión de diagramas	7
1.3.4 Definición y uso de acrónimos	8

1.3.5	Definición y uso de símbolos	9
1.4	Motivación y objetivos	9
1.5	Organización de la memoria	9
2	Estudio teórico	11
2.1	Introducción	11
2.2	Estado del Arte	11
2.3	Técnicas utilizadas	12
2.4	Conclusiones	12
3	Desarrollo	13
3.1	Introducción	13
3.2	Desarrollo del sistema de experimentación	13
3.3	Planteamiento matemático	13
3.4	Conclusiones	13
4	Resultados	15
4.1	Introducción	15
4.2	Entorno experimental	15
4.2.1	Bases de datos utilizadas	15
4.2.2	Métricas de calidad	15
4.2.3	Estrategia y metodología de experimentación	15
4.3	Resultados experimentales	15
4.4	Conclusiones	20
5	Conclusiones y líneas futuras	23
5.1	Conclusiones	23
5.2	Líneas futuras	23
	Bibliografía	25
A	Manual de usuario	27
A.1	Introducción	27
A.2	Manual	27
A.3	Ejemplos de inclusión de fragmentos de código fuente	27
A.4	Ejemplos de inclusión de algoritmos	29
B	Herramientas y recursos	31

Índice de figuras

1.1	Clasificación de los objetos para la gramática.	8
2.1	Departamento de Electrónica.	12
2.2	Departamento de Electrónica en el lateral.	12
4.1	Optimal frames number in the training data set.	17
4.2	Idiap Smart Meeting Room for AV16.3 recordings. (a) Room layout showing the centered table, and the microphones arranged in two circular arrays. (b) Sample of recorded video frame showing the arrays area.	18
4.3	Comparison between the steered power response generated by the model (solid line) and that calculated using simulated waveforms in the AV16.3 environment (stems). Results for the speaker in given angles and the array steered from -90° to $+90^\circ$ are shown.	19

Índice de tablas

4.1	Comparativa.	16
4.2	Resultados de la correlación cruzada.	16
4.3	Resultados TEST CLEAR 2006.	21

Índice de listados de código fuente

A.1	Ejemplo de código fuente con un <code>lstinputlisting</code> dentro de un <code>codefloat</code>	28
A.2	Ejemplo de código fuente con estilo <code>Cnice</code> , de nuevo con un <code>lstinputlisting</code> dentro de un <code>codefloat</code>	28
A.3	Ejemplo de código fuente con estilo <code>Cnice</code> , modificado para que no aparezca la numeración.	29
A.4	Ejemplo con colores usando el estilo <code>Ccolor</code>	29

Índice de algoritmos

A.1	How to write algorithms	30
A.2	IntervalRestriction	30

Lista de acrónimos

ANN	Artificial Neural Network.
DBN	Dynamic Bayesian Network.
EIR	Emotion Identification Rate.
EMODB	Berlin Database of Emotional Speech.
ES	Emotional Strength.
ETTS	Emotional Text To Speech.
SIR	Speaker Identification Rate.
SOC	System on a Chip.
SQ	Speech Quality.
STRAIGHT	Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum.
TD-PSOLA	Time Domain Pitch Synchronous OverLap Add.

Lista de símbolos

\AA non-SI unit of length.

$\perp\!\!\!\perp$ conditional independence.

Ω unit of electrical resistance.

$x(t)$ Audio signal.

$x_i(t)$ Audio signal captured at microphone i .

Capítulo 1

Introducción

Desocupado lector, sin juramento me podrás creer que quisiera que este libro [...] fuera el más hermoso, el más gallardo y más discreto que pudiera imaginarse¹.

Miguel de Cervantes, Don Quijote de la Mancha

1.1 Presentación

Esta plantilla² pretende proporcionar un conjunto de estilos consistentes y unificados para cubrir las necesidades de generación de memorias L^AT_EX para cada uno de los TFCs/TFMs/TFGs y tesis doctorales que se generen en la Escuela Politécnica Superior de la Universidad de Alcalá³.

Para utilizar la plantilla se han generado algunos capítulos genéricos en los que se han incluido secciones “tutoriales”, en la que se explican algunas de sus características y se muestran ejemplos de elementos típicos que pueden ser de utilidad (pero sin el objetivo de que esto sea una guía de L^AT_EX).

Igualmente se proporciona un modelo simplificado de un anteproyecto (para el caso de los TFCs/TFMs/TFGs), así como parte de la documentación que hay que presentar para la defensa de los TFGs de la Universidad de Alcalá.

1.2 Uso de la plantilla

1.2.1 Prerrequisitos

Para usar la plantilla tal y como está definida, hace falta disponer de una serie de paquetes de estilos L^AT_EX (ficheros `.sty`), todos ellos definidos en el fichero `config/preamble.tex`.

No vamos a hacer un listado de todos lo necesarios (sería demasiado largo⁴), pero en la mayoría de distribuciones GNU/Linux serán fáciles de conseguir en caso de que la compilación genere un error

¹Tomado de ejemplos del proyecto T_EX_S.

²Asegúrate de compilar de nuevo el documento (como cuenta la sección 1.2.2), para verificar que todo funciona y por si ha habido algún cambio en los fuentes que no está reflejado en los pdf de ejemplo precompilados.

³También se incluye la definición para las tesis de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid. La extensión a los TFGs de la misma es sencilla, aunque no se ha realizado por el momento.

⁴Los que suelen no estar instalados en un ubuntu estándar son `texlive-publishers`, `texlive-science`

de fichero no encontrado. Si os sucede, buscadlos en alguno de los paquetes `texlive-*`. En caso de no encontrarlos, una búsqueda en google (que con casi total seguridad referenciará a alguna página en CTAN) os dará el enlace a la descarga correspondiente. A partir de ahí, su inclusión en directorios locales será suficiente (como por ejemplo hemos tenido que hacer con el paquete `background`, incluido en la distribución en el directorio `sty/background`). Lo más cómodo es hacer una instalación del `texlive-full`.

Igualmente será necesario tener instaladas una serie de utilidades y aplicaciones:

- `make`, si se quiere utilizar la facilidad del `Makefile` suministrado. Está disponible en todas las distribuciones GNU/Linux.
- `rubber`, si se quiere utilizar la prestación de compilación de código \LaTeX incluida en el `Makefile`. Debería estar disponible en cualquier distribución GNU/Linux, pero si no es así, puedes optar por descargarla, o bien usar la alternativa de `latexmk` (para el que también se incluyen targets específicos en el `Makefile` suministrado).
- `dia`, si se quiere utilizar el ejemplo proporcionado de generación de esquemas con dicha herramienta.
- `epspdf`, si se quiere utilizar la facilidad de la conversión automática de ficheros `eps` a `pdf` (también se usa en la conversión de ficheros `.dia`).
- `makeglossaries`, si se quiere utilizar la prestación de manejo de listas de acrónimos y variables. Suele estar en el paquete `texlive-latex-extra` en distribuciones basadas en Debian (como `ubuntu` y derivados).

1.2.2 Compilación

Para facilitar la generación del documento se incluye un `Makefile` relativamente sencillo. No somos expertos ni en `make` ni en \LaTeX , con lo que seguro que no es el mejor de los `Makefiles` del mundo, pero creemos que hace su función.

El `Makefile` tiene las siguientes características y prestaciones:

- Hace uso de la herramienta `rubber`. En caso de no disponer de ella, puede usarse `latexmk` (hay targets específicos de ejemplo para ese caso), pero esta última herramienta no ha sido probada intensivamente. Si no se dispone de ninguna de ellas, habría que generar la estructura típica de compilación: (al estilo `pdflatex + bibtex + pdflatex + pdflatex`).
- Soporta la generación automática de ficheros `pdf` a partir de los `dia` y `svg` (se han elegido estos a modo de ejemplo, pero se puede adaptar fácilmente a otras necesidades).
- Genera la información de listas de acrónimos y símbolos con la herramienta `makeglossaries`. Es imprescindible tenerla instalada si se desea usar esa capacidad.
- Soporta los targets:
 - `all` (que es la opción predeterminada si se ejecuta `make` sin más argumentos), que genera el fichero `book.pdf` correspondiente, usando `rubber` y `makeglossaries`. No nos hemos planteado la generación de ficheros en otros formatos.
 - `all_latexmk`, que genera el fichero `book.pdf` correspondiente, usando `latexmk` y `makeglossaries`. Si ésta es tu opción, sustituye el target `all` por éste, para facilitarte la compilación.

- `anteproyecto`, que genera el fichero `anteproyecto.pdf` correspondiente, usando `rubber`. No nos hemos planteado la generación de ficheros en otros formatos.
- `anteproyecto_latexmk`, que genera el fichero `anteproyecto.pdf` correspondiente, usando `latexmk`. Si ésta es tu opción, sustituye el target `anteproyecto` por éste, para facilitarte la compilación.
- `tar`, que genera un fichero `tgz` que contiene todo lo necesario para la distribución de la plantilla.
- `clean`, que borra todos los ficheros temporales usando `rubber` (para simplificar los targets, la limpieza se hace tanto para los temporales de generación del `book.pdf` como los del `anteproyecto.pdf`).
- `clean_latexmk`, que borra todos los ficheros temporales usando `latexmk`. Si ésta es tu opción, sustituye el target `clean` por éste, para facilitarte la compilación (para simplificar los targets, la limpieza se hace tanto para los temporales de generación del `book.pdf` como los del `anteproyecto.pdf`).

1.2.3 Estructura del documento generado por la plantilla

La plantilla definida presenta la siguiente estructura:

- Portada y página de información sobre el trabajo, que será dependiente del tipo de trabajo y la titulación. Se genera automáticamente a partir de la información definida en la sección [1.2.4](#)
- Dedicatoria, con un ejemplo incluido en el fichero `dedication/dedicatoria.tex`.
- Agradecimientos, con un ejemplo incluido en el fichero `dedication/agradecimientos.tex`.
- Resumen en español, con un ejemplo incluido en el fichero `abstract/resumen.tex`.
- Resumen en inglés, con un ejemplo incluido en el fichero `abstract/abstract.tex`.
- Resumen extendido en español (opcional en algunos tipos de documento), con un ejemplo incluido en el fichero `abstract/resumen-extendido.tex`.
- Índice de contenidos, índice de figuras e índice de tablas.
- Índices adicionales, de los que se incluye un ejemplo de listado específico en el fichero `cover/extralistings.tex`, que incluye el listado de fragmentos de código fuente definidos en el `appendix/manual.tex`.
- Listado de acrónimos utilizados, que se definen en el fichero `acronyms/acronymsgl.tex`, y del que incluimos más información en la sección [1.3.4](#).
- Listado de símbolos utilizados, que se definen en el fichero `symbols/symbolsgl.tex`, y del que incluimos más información en la sección [1.3.5](#).
- Capítulos del documento, del que hay varios ejemplos que siguen la estructura típica (introducción, estudio teórico, desarrollo, resultados y conclusiones).
- Pliego de condiciones y presupuesto, opcionales (se incluyen un par de ejemplos del trabajo fin de carrera de Jesús Martínez en los ficheros `pliego/pliego-ejemplo.tex` y `presupuesto/presupuesto-ejemplo.tex`).

- Bibliografía, de la que se puede cambiar el estilo utilizado y los ficheros `.bib` en el fichero `biblio/bibliography.tex`
- Apéndices, de los que ahora se incluyen dos ejemplos en los ficheros `appendix/manual.tex` (que sirve de pretexto para mostrar cómo se insertan fragmentos de código fuente), y `appendix/herramientas.tex`.
- Contraportada, sólo para el caso de los TFGs en UAH.

Por supuesto, modificad la estructura para que encaje en las directrices que tengáis al respecto de cómo documentar vuestro trabajo.

1.2.4 Definiciones específicas del tipo de documento

Para comenzar a usar la plantilla es fundamental revisar el fichero `book.tex` en el que se incluyen todos los detalles genéricos de la estructura usada en el documento, con comentarios que esperamos que os ayuden a entenderlo. Si sois de los impacientes, basta con que comencéis por la parte en la que se incluyen los distintos capítulos (buscad la parte de los `\input{chapters/*.tex}`).

Uno de los ficheros de configuración más importantes es el `config/myconfig.tex` en el que se incluyen elementos para determinar la configuración específica de tu documento. El primero de ellos (para facilitar la generación de documentos en español o inglés), es el que define el idioma que vas a utilizar. Para seleccionarlo, basta con asignar `spanish` o `english` a la variable `\mybooklanguage`. A partir de ella, el sistema generará las cabeceras y títulos adecuados a cada una.

Igualmente, tendréis que definir las siguientes variables sobre el tipo de trabajo y el autor:

- Acrónimo de la titulación correspondiente al trabajo (variable `\mydegree`): Que se seleccionará entre los definidos (por ahora⁵ son IT, IE, ITTSE, ITTST, ITI, GIEC, GIEAI, GIST, GITT, GIT, GIC, GII, GSI, MUSEA, PHDUAH y PHDUPM) y que automáticamente configura portadas⁶, entre otras cosas.
- Título del documento (variable `\mybooktitle`).
- Nombre del autor del trabajo (variable `\mybookauthor`).
- DNI del autor del trabajo, usado en el papeleo de los TFGs (variable `\mybookDNI`).
- Departamento en el que se realiza el trabajo (variable `\mybookdepartment`, en español, y `\mybookdepartmentEnglish`, en inglés).
- Grupo de investigación en el que se realiza el trabajo (variable `\mybookresearchgroup`).
- Centro en el que se realiza el trabajo (variable `\mybookschool`, que debería ser la Escuela Politécnica Superior, pero se incluye por generalidad).
- Universidad en el que se realiza el trabajo (variable `\mybooksuniversidad`, que debería ser la de Alcalá, pero se incluye por generalidad).
- Titulación del autor (usada en las tesis de UPM, (variable `\mybookauthordegree`)).

⁵Este documento se generó a finales de 2013.

⁶Un comentario sobre el color de las bandas en la portada de los TFGs: de acuerdo con la normativa, dicho color debe ser PANTONE 160c. Yo he intentado utilizar dicho color, pero el aspecto con el que finalmente aparece no es ni de lejos similar al del modelo que proporciona la EPS, con lo que he optado por utilizar el que se ve realmente en dicho modelo. Si quieres cambiarlo, busca “PANTONE” en `config/preamble.tex`.

- Email del autor (variable `\mybookemail`).
- Nombre del primer (o único, en su caso) director del trabajo (variable `\mybookNameFirstAdvisor`).
- Nombre del segundo (en su caso) director del trabajo (variable `\mybookNameSecondAdvisor`).
- Nombre del presidente del tribunal (variable `\mybookpresident`).
- Nombre del primer vocal del tribunal (variable `\mybookfirstvocal`).
- Nombre del segundo vocal del tribunal (variable `\mybooksecondvocal`).
- Nombre del secretario del tribunal, en su caso (variable `\mybooksecretary`).
- Año del trabajo (variable `\mybookyear`).
- Fecha del anteproyecto, en su caso (variable `\mybookanteproyectodate`), en el caso de que se usen la plantilla del anteproyecto.
- Fecha de la defensa del trabajo, en su caso (variable `\mydefensedate`).
- Palabras clave en inglés (variable `\mybookkeywords`).
- Palabras clave en español (variable `\mybookpalabrasclave`).

Y también variables para el caso de los trabajos que necesitan papeleo adicional (publicación en abierto, autorizaciones, etc.).

- Nombre del Secretario del Departamento (que firmará parte del papeleo) (variable `\mybookdepartmentsecretary`).
- Fecha que aparecerá en la firma del papeleo (variable `\mybookdateforpaperwork`).
- DNI del alumno (variable `\mybookDNIOpenPublishing`).
- Figura del autor del trabajo, que es normalmente “alumno” (variable `\mybookProfFigureOpenPublishing`).
- DNIs del/de los tutores, para los permisos de publicación en abierto (variables `\mybookDNIFirstAdvisor` y `\mybookDNISecundAdvisor`, en su caso)

Se ha comenzado a trabajar en la versión alfa del soporte para *research reports*, y en ese caso se usa la variable `\mybookresearchreportID`.

Parte de esa información se utilizará para rellenar la metainformación incluida en el fichero pdf que se genera.

También se definen los colores que se usarán en los hiperenlaces del documento. En concreto⁷:

- Color de los enlaces en el índice de contenidos (variable `\mytoclinkcolor`).
- Color de los enlaces en el índice de figuras (variable `\myloflinkcolor`).
- Color de los enlaces en el índice de tablas (variable `\mylotlinkcolor`).

⁷Os rogamos encarecidamente que cambiéis los colores definidos actualmente, que se han usado para verificar que todo funciona correctamente.

- Color de los enlaces en otros índices (variable `\myothertoclinkcolor`), de los que ahora se incluye un ejemplo en el fichero `cover/extralistings.tex`.
- Color de los enlaces (`\ref`) en el documento (variable `\mylinkcolor`).
- Color de los enlaces a URLs (variable `\myurlcolor`).
- Color de los enlaces a referencias bibliográficas (variable `\mycitecolor`).

Basta con que defináis las variables correspondientes y la plantilla generará automáticamente las portadas adecuadas a la normativa y usará las definiciones específicas que hayas seleccionado.

Por si os hace falta, en `config/postamble.tex` se definen algunas variables relacionadas con el tipo de trabajo. Por ejemplo las variables `\mydegreefull` (igual a “Estudios de Doctorado” en esta compilación), `\mybookworktype` (igual a “PHDUAH” en esta compilación) y `\mybookworktypefull` (igual a “Tesis Doctoral” en esta compilación). Otro ejemplo sería el autor de contacto: Javier Macías-Guarasa <macias@depeca.uah.es>.

Importante para las tesis de UAH: si necesitáis incluir ficheros pdf (los de autorización e informes de los tutores, por ejemplo), esta plantilla lo permite: mirad los `\includepdf` en el `book.tex`.

1.2.5 Plantilla de anteproyecto

Para el caso de los TFM/TFGs/TFCs, se incluye una plantilla para realizar el anteproyecto.

La plantilla se encuentra en el fichero `anteproyecto.tex` y el Makefile genera automáticamente el `anteproyecto.pdf`, haciendo un `make anteproyecto`.

1.2.6 Papeleo adicional para la defensa de los TFGs

Para el caso de los TFGs y TFM (al menos los del MUSEA), se incluyen en el directorio papeleo algunos de los documentos que hay que generar en el momento de la defensa. En concreto:

- La autorización del tutor para la publicación en abierto, en el fichero `autorizacionTutorPublicarRepositorio.tex` para TFGs.
- La autorización del autor para la publicación en abierto, en el fichero `autorizacionAutorPublicarRepositorio.tex`
- El visto bueno del tutor para la defensa del TFG `vistoBuenoTutorTFG.tex`
- La autorización del autor y tutor/tutores para la publicación en abierto, en el fichero `autorizacionPublicarAbiertoTFM-MUSEA.tex`.
- El visto bueno del tutor para la defensa del TFM `vistoBuenoTutorTFM-MUSEA.tex`

En el directorio se incluye un Makefile que genera los pdfs correspondientes a esos documentos. El sistema automáticamente adapta los singulares/plurales necesarios para el caso de que haya uno o varios directores/tutores.

1.2.7 Problemas conocidos

Resumimos a continuación los problemas con los que nos hemos ido encontrando tras el uso más generalizado de esta plantilla, y, en su caso, la solución propuesta/adaptada:

- Al menos en la versión 12.04 de ubuntu, se producía un fallo de compilación por un problema del *option clash* del paquete `xcolor`. La solución fue incluir las opciones `[RGB,rgb]` de dicho paquete en el `documentclass`, y eliminar la inclusión de `xcolor` (que ya lo incluye, al menos, `listings`). No es muy bonito como solución, pero funcionaba. Eso dio lugar a otro problema, que hacía que todas las página aparecieran como en color cuando se llevaba a la imprenta (con el consiguiente incremento de precio). Para intentar solucionarlo, he vuelto a eliminar las opciones `[RGB,rgb]` del `documentclass`, y se las paso con un `PassOptionsToPackage`, pero está pendiente de verificación. Please, confirmadme que funciona (tanto lo del color como la compilación en un ubuntu 12.04 (o posterior)).
- También hemos observado en la versión 12.04 de ubuntu que `evince` no es capaz de visualizar la primera página de los TFGs (generada con `tikz`), y que `xpdf` genera un core cuando intenta abrir el fichero compilado. La solución es usar `qpdfview` o `acroread`.

Si das con más problemas, escribid por favor a Javier Macías-Guarasa <macias@depeca.uah.es>contándonoslos y trataremos de solucionarlo (y si tenéis la solución, contádnosla también).

1.3 Ejemplos de elementos de utilidad

1.3.1 Uso de comandos definidos

A modo de ejemplo, hemos definido el comando `\texten{}` en `config/myconfig.tex` para usarlo, por ejemplo, para marcar palabras escritas en inglés (aka *English*). Sigue el ejemplo para definir aquellos que utilices con frecuencia.

Si quieres escribir el símbolo `backslash` puedes usar el comando `\backlash{}: \`.

Lo mismo aplica para el símbolo `tilde`, para lo que puedes usar el comando `\textasciitilde{}: ~`.

1.3.2 Uso de “frases célebres”

Respecto a la frase célebre del inicio de los capítulos: todas las que hemos usado y las definiciones que las generen están sacadas del excelente trabajo de Marco Antonio Gomez-Martín y Pedro Pablo Gomez-Martín en el proyecto `TEXIS`, una plantilla para la creación de tesis y otros documentos y disponible en [\[1\]](#).

1.3.3 Inclusión de diagramas

Para incluir gráficos, la compilación que utilizamos permite usar ficheros `png`, `jpg` y `pdf`, en el comando `\includegraphics`. Si queréis ahorraros incluir el path a cada fichero, podéis definir todos aquellos en los que haya ficheros gráficos en el `\graphicspath` del `book.tex`.

En la figura 1.1 se muestra un ejemplo de gráfico generado automáticamente a partir de un fichero `.dia`⁸: `diagrams/Esquema_objetos.dia` (podéis generalizar su generación en el `Makefile`).

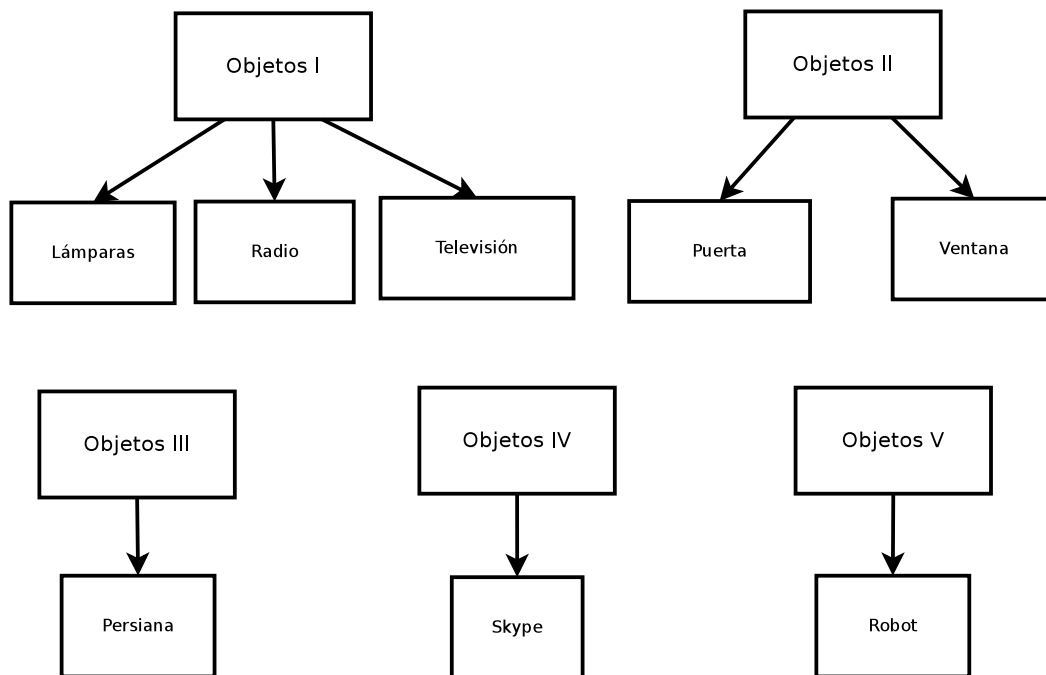


Figura 1.1: Clasificación de los objetos para la gramática.

1.3.4 Definición y uso de acrónimos

El uso del paquete `glossaries` permite definir los acrónimos y el sistema automáticamente gestiona su inclusión completa la primera vez que se usa. Los acrónimos de ejemplo están en el fichero `acronyms/acronymsgl.tex`.

Así, si nos referimos a *Emotional Text To Speech (ETTS)* o bien a *Berlin Database of Emotional Speech (EMODB)*, veremos como aparecen expandidas la primera vez. A partir de ahí, sólo se usará el acrónimo como puede verse al volver a hablar de **ETTS** y **EMODB**.

Tiene también soporte para resetear todos los acrónimos como si no estuvieran usados. Vuelvo a incluir el párrafo anterior tras un reset:

El uso del paquete `acronym` permite definir los acrónimos y el sistema automáticamente gestiona su inclusión completa la primera vez que se usa. Así, si nos referimos a *Emotional Text To Speech (ETTS)* o bien a *Berlin Database of Emotional Speech (EMODB)*, veremos como aparecen expandidas la primera vez. A partir de ahí, sólo se usará el acrónimo como puede verse al volver a hablar de **ETTS** y **EMODB**.

Y permite también forzar que se vuelva a citar completo aunque ya se haya utilizado (con el acrónimo entre paréntesis), como puede verse en *Emotional Text To Speech* (equivalente a *Emotional Text To Speech* que vale para cualquier glosario), y también a usar forzosamente el acrónimo. Primero reseteamos de nuevo.

Y ahora forzamos el acrónimo: **EMODB** (equivalente a **EMODB** que vale para cualquier glosario). También podemos forzar a que lo ponga todo, con *Berlin Database of Emotional Speech (EMODB)*.

Podemos seguir definiendo entradas de acrónimos, referirnos a *Dynamic Bayesian Network (DBN)* por primera vez, y las siguientes aparecerá como **DBN**. Pongo ahora el resto de acrónimos *Speech Quality*

⁸Tomadas de los proyectos fin de carrera de David Casillas y Manuel Villaverde.

(*SQ*), *Emotion Identification Rate (EIR)*, *Speaker Identification Rate (SIR)* y *Emotional Strength (ES)*. Finalmente los repito para que se vea el efecto: *SQ*, *EIR*, *SIR* y *ES*.

Y gestiona bien los plurales, ponemos el plural como *Systems on a Chip (SOCs)* la primera vez, y luego la segunda como *SOCs*. Y podemos volver al singular con *SOC*.

1.3.5 Definición y uso de símbolos

Los símbolos definidos están incluidos en el fichero `symbols/symbolsgl.tex` y en esta sección mostramos algunos ejemplos.

El \AA se usa en biología estructural, mientras que el Ω se usa en electrónica. También podemos poner $x(t)$.

$$x(t) \tag{1.1}$$

Y finalmente la que nos falta: $x_i(t)$, también dentro de fórmulas (otra cosa es que sea conveniente o útil):

$$x_i(t) = \sqrt{i} \tag{1.2}$$

Acabamos con un par de acrónimos: *Time Domain Pitch Synchronous OverLap Add (TD-PSOLA)* y *Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum (STRAIGHT)*.

Recientemente nos han pedido información sobre cómo introducir el comando de independencia incondicional y también funciona (la definición está en el `config/myconfig.tex`, y se usa tal cual en el `symbols/symbolsgl.tex` (revisadlos para ver cómo se implementa): \perp .

1.4 Motivación y objetivos

La motivación de este proyecto...

Los objetivos principales de este trabajo son (ejemplo utilizando "enumerate"):

1. Primer objetivo...
2. Segundo objetivo...
 - (a) Objetivo 2.1...
 - (b) Objetivo 2.2...
3. Tercer objetivo...

1.5 Organización de la memoria

Esta memoria se organiza en cinco grandes capítulos. El primero ...

Capítulo 2

Estudio teórico

Y así, del mucho leer y del poco dormir, se le secó el cerebro de manera que vino a perder el juicio¹.

Miguel de Cervantes Saavedra

2.1 Introducción

En este capítulo se cuenta tal y tal.

El capítulo se estructura en n apartados...

2.2 Estado del Arte

En el estado del arte se enumeran los trabajos más relevantes de otros grupos de investigación. A continuación se muestra un ejemplo del uso de viñetas que nos proporciona `itemize`:

- En el trabajo
- En el siguiente trabajo.....

O citas en un párrafo real: Sin embargo, hay entornos acústicos donde las tasas de error conseguidas son todavía demasiado altas. En concreto, las aplicaciones en las que la captura de la señal de habla se hace usando micrófonos alejados del locutor (típicamente para distancias superiores a un metro) muestran una fuerte sensibilidad a los problemas de reverberación, ruido aditivo y baja relación señal a ruido ([2],[3]). En estos entornos, se ha propuesto el uso de arrays de micrófonos como un método para mejorar la calidad del habla capturada [4][5].

Existen múltiples formas de insertar figuras en Latex. A continuación, se muestra un ejemplo del uso de `figure`. Como se puede ver en la Figura 2.1 también se pueden poner referencias a las figuras por medio de `ref` y la etiqueta `label` de la figura en particular.

Y ahora un ejemplo en el que ponemos el `caption` en el lateral:

¹Tomado de ejemplos del proyecto T_EX_S.



Figura 2.1: Departamento de Electrónica.

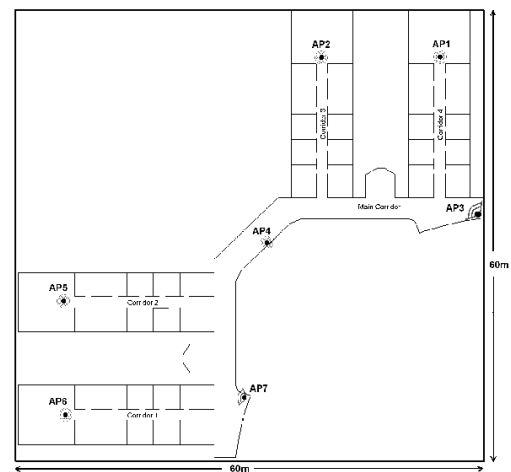


Figura 2.2: Departamento de Electrónica en el lateral.

2.3 Técnicas utilizadas

Blah, blah, blah...

2.4 Conclusiones

Blah, blah, blah...

Capítulo 3

Desarrollo

*A fuerza de construir bien, se llega a buen arquitecto*¹.

Aristóteles

3.1 Introducción

En este capítulo se incluirá la descripción del desarrollo del trabajo.

El capítulo se estructura en n apartados:...

3.2 Desarrollo del sistema de experimentación

Blah, blah, blah. . .

3.3 Planteamiento matemático

También resulta útil poder introducir ecuaciones que se encuentran tanto en línea con el texto (como por ejemplo $\sigma = 0,75$), como en un párrafo aparte (como en la ecuación [3.1](#)). Al igual que ocurre con las figuras, también se pueden referenciar las ecuaciones.

$$p[q_t = \sigma_t | q_{t-1} = \sigma_{t-1}] \tag{3.1}$$

3.4 Conclusiones

Blah, blah, blah. . .

¹Tomado de ejemplos del proyecto T_EX_S.

Capítulo 4

Resultados

*Rem tene, verba sequentur (Si dominas el tema, las palabras vendrán solas)*¹.

Catón el Viejo

4.1 Introducción

En este capítulo se introducirán los resultados más relevantes del trabajo.

La estructura del capítulo es...

4.2 Entorno experimental

Blah, blah, blah.

4.2.1 Bases de datos utilizadas

Blah, blah, blah.

4.2.2 Métricas de calidad

Blah, blah, blah.

4.2.3 Estrategia y metodología de experimentación

Blah, blah, blah.

4.3 Resultados experimentales

A continuación, se muestra un ejemplo de tabla simple (ver tabla [4.1](#)).

¹Tomado de ejemplos del proyecto T_EX_S.

Tabla 4.1: Comparativa.

Method	Training Time	Man-Work (%)
Propagation model	< 30 sec	5
Manual	9 h 30 min	24
Automatic	2 h	10 8

Cuando las tablas ocupan más de un página se debe utilizar un tipo especial de tablas denominado longtable. A continuación, se muestra un ejemplo del mismo (ver tabla 4.2).

Tabla 4.2: Resultados de la correlación cruzada.

Posición Real	Posición estimada	Coef. Correlación	Acierto/Fallo
2P0	2P0	0,004954	A
2P1	2P4	0,005752	F
2P2	2P2	0,005461	A
2P3	2P0	0,004634	F
2P5	2P4	0,005991	F
2P6	2P16	0,004410	F
2P7	3P9	0,008038	F
2P8	3P9	0,003753	F
2P9	2P7	0,004908	F
2P10	2P10	0,007273	A
2P14	2P16	0,006485	F
2P15	2P15	0,004932	A
2P16	2P16	0,006237	A
2P17	2P15	0,005110	F
2P18	3P18	0,006235	F
2P19	3P18	0,004827	F
2P20	2P20	0,006877	A
2P22	3P18	0,003048	F
2P24	2P24	0,006833	A
2P25	2P25	0,004875	A
2P26	2P31	0,005511	F
2P27	2P28	0,004590	F
2P30	2P31	0,005576	F
2P31	2P31	0,007213	A
2P32	2P35	0,003340	F
2P34	2P34	0,004128	A
2P36	2P35	0,003329	F
2P37	2P37	0,003468	A
2P39	2P38	0,002577	F
2P40	2P43	0,004303	F
2P41	2P41	0,001573	A
Continúa en la página siguiente			

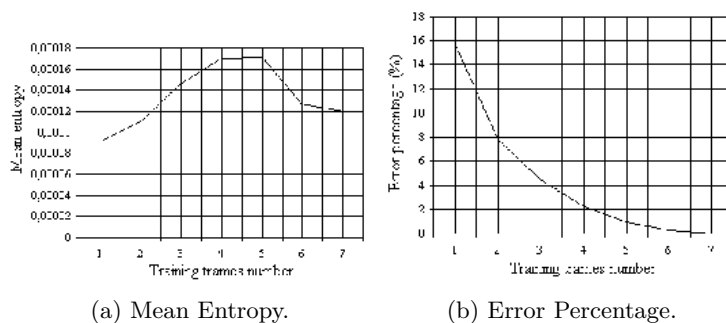


Figura 4.1: Optimal frames number in the training data set.

Tabla 4.2 – continúa en la página anterior

Posición Real	Posición estimada	Coef. Correlación	Acierto/Fallo
2P42	2P41	0,000846	F
2P44	2P44	0,002732	A
2P45	23P45	0,001958	F
2P47	2P34	0,002869	F
2P48	2P43	0,004569	F
2P49	3P51	0,001374	F
2P50	2P34	0,002274	F
2P51	2P63	0,003931	F
2P52	2P55	0,003537	F
2P53	3P56	0,003126	F
2P54	2P67	0,005560	F
2P56	2P55	0,002817	F
2P57	2P67	0,006168	F
2P58	2P58	0,005278	A
2P60	3P66	0,004966	F
2P61	3P61	0,004748	A
2P64	2P67	0,005342	F
2P66	2P4	0,004172	F
2P67	2P67	0,005706	A
3P0	3P0	0,003674	A
3P61	2P61	0,003263	F
3P64	2P67	0,003484	F
3P65	2P67	0,002975	F
3P66	2P58	0,005029	F
3P67	3P67	0,003714	A

En algunas ocasiones, también resulta útil emplear el entorno `subfigure` para añadir múltiples imágenes dentro de la misma figura. A continuación, se muestra un ejemplo del uso en la figura 4.1. También se pueden referenciar las sub-figuras de forma individual, por ejemplo la sub-figura 4.1b (usando un método de cita), o bien la sub-figura 4.1.b (usando otro alternativo).

La figura 4.2 muestra otro ejemplo con referencias a las subfigures en el caption principal.

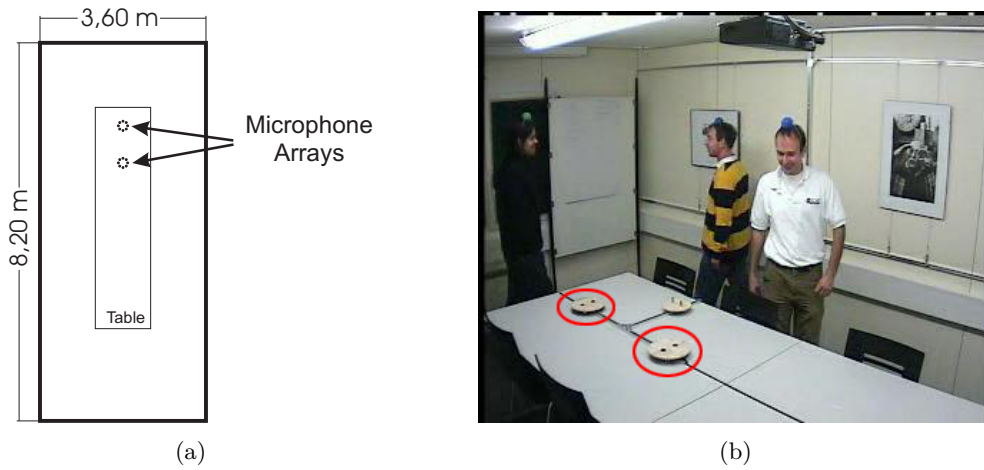


Figura 4.2: Idiap Smart Meeting Room for AV16.3 recordings. (a) Room layout showing the centered table, and the microphones arranged in two circular arrays. (b) Sample of recorded video frame showing the arrays area.

Y en las figuras 4.3e y mostramos un ejemplo un poco más complejo.

Incluso podemos poner una tabla “apaisada”, como en la 4.3, donde se muestra un resumen de los resultados obtenidos en una serie de experimentos de localización de locutores.

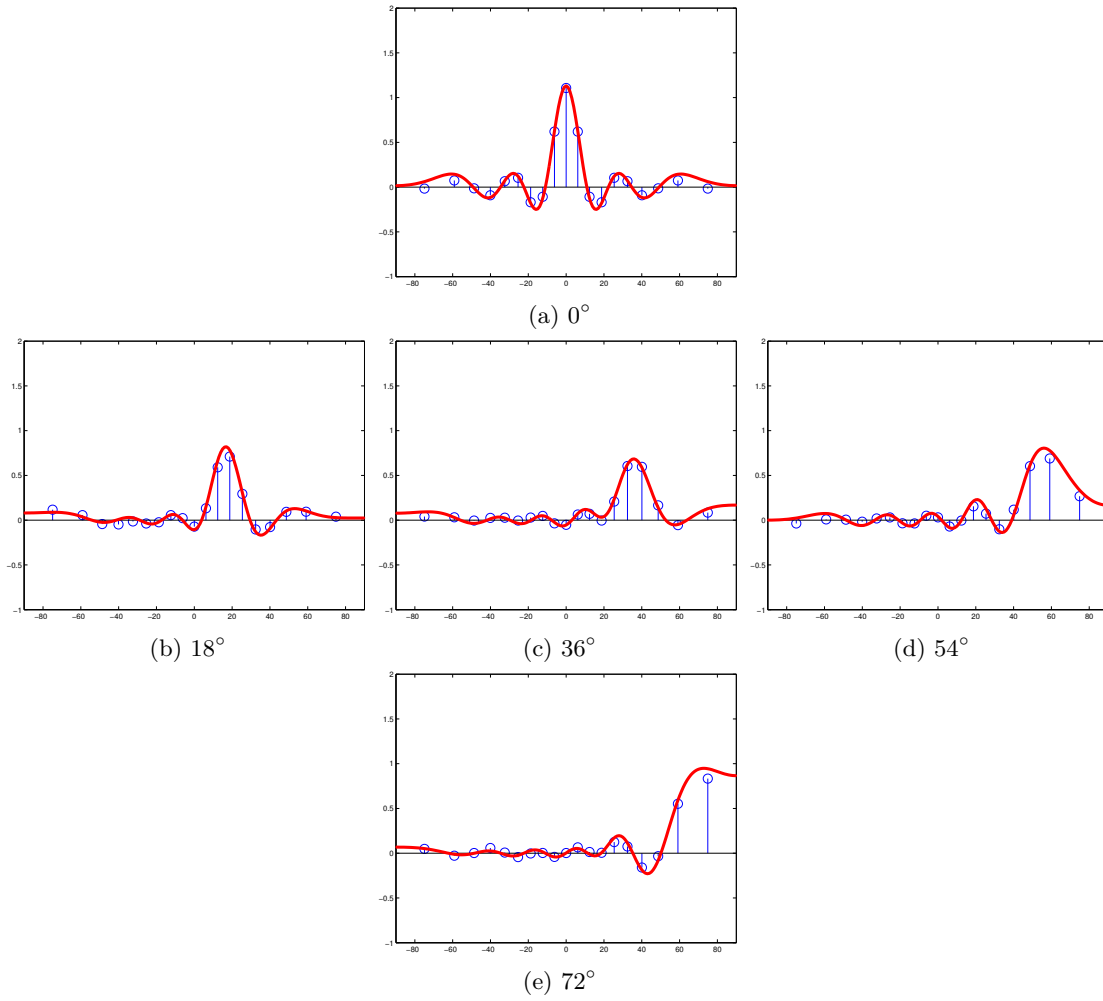


Figura 4.3: Comparison between the steered power response generated by the model (solid line) and that calculated using simulated waveforms in the AV16.3 environment (stems). Results for the speaker in given angles and the array steered from -90° to $+90^\circ$ are shown.

4.4 Conclusiones

Blah, blah, blah.

	UKA	ITC	AIT	UPC	IBM
P _{cor}	57,0 ± 1,4 %	84,0 ± 3,3 %	47,0 ± 3,1 %	20,0 ± 2,5 %	67,0 ± 2,9 %
Bias fine (x:y:z) [mm]	20 : -42 : -75	45 : 27 : -41	-27 : -77 : -40	-59 : 112 : 52	91 : -69 : -38
Bias fine+gross (x,y,z) [mm]	735 : -93 : -258	67 : 439 : -134	17 : -402 : -118	-141 : 255 : 39	474 : -141 : -14
AEE fine [mm] = MOTP	210	130	266	344	228
Fine+gross [mm]	1201	632	1006	1188	884
Loc. frames	5035	22	995	977	1023
Ref. duration (s)	6287,0	596,0	1143,0	1180,0	1194,0

Tabla 4.3: Resultados TEST CLEAR 2006.

Capítulo 5

Conclusiones y líneas futuras

En este apartado se resumen las conclusiones obtenidas y se proponen futuras líneas de investigación que se deriven del trabajo.

La estructura del capítulo es...

5.1 Conclusiones

Para añadir una referencia a un autor, se puede utilizar el paquete `cite`. En el trabajo [6], se muestra un trabajo...

Y podemos usar de nuevo algún acrónimo, como por ejemplo [TD-PSOLA](#), o uno ya referenciado como *Artificial Neural Network (ANN)*.

5.2 Líneas futuras

Pues eso.

Bibliografía

- [1] “Tesis: Una plantilla de latex para tesis y otros documentos,” <http://gaia.fdi.ucm.es/projects/tesis/> [Último acceso 1/diciembre/2013].
- [2] D. Gelbart and N. Morgan, “Double the trouble: Handling noise and reverberation in far-field automatic speech recognition,” in *International Conference on Spoken Language Processing (ICSLP)*, 2002.
- [3] S. Kochkin and T. Wickstrom, “Headsets, far field and handheld microphones: Their impact on continuous speech recognition,” EMKAY, a division of Knowles Electronics, Tech. Rep., 2002.
- [4] M. L. Seltzer, “Microphone array processing for robust speech recognition,” Ph.D. dissertation, Carnegie Mellon University, 2003.
- [5] W. Herbordt, *Sound capture for human/machine interfaces - Practical aspects of microphone array signal processing*. Springer, Heidelberg, Germany, March 2005.
- [6] L. Armani, M. Matassoni, M. Omologo, and P. Svaizer, “Use of a csp-based voice activity detector for distant-talking asr,” in *European Conference on Speech Communication and Technology*, 2003, pp. 501–504.
- [7] “Información sobre gnu/linux en wikipedia,” <http://es.wikipedia.org/wiki/GNU/Linux> [Último acceso 1/noviembre/2013].
- [8] “Página de la aplicación emacs,” <http://savannah.gnu.org/projects/emacs/> [Último acceso 1/noviembre/2013].
- [9] “Página de la aplicación kdevelop,” <http://www.kdevelop.org> [Último acceso 1/noviembre/2013].
- [10] L. Lamport, *LaTeX: A Document Preparation System, 2nd edition*. Addison Wesley Professional, 1994.
- [11] “Página de la aplicación octave,” <http://www.octave.org> [Último acceso 1/noviembre/2013].
- [12] “Página de la aplicación cvs,” <http://savannah.nongnu.org/projects/cvs/> [Último acceso 1/noviembre/2013].
- [13] “Página de la aplicación gcc,” <http://savannah.gnu.org/projects/gcc/> [Último acceso 1/noviembre/2013].
- [14] “Página de la aplicación make,” <http://savannah.gnu.org/projects/make/> [Último acceso 1/noviembre/2013].

Apéndice A

Manual de usuario

A.1 Introducción

Blah, blah, blah...

A.2 Manual

Pues eso.

A.3 Ejemplos de inclusión de fragmentos de código fuente

Para la inclusión de código fuente se utiliza el paquete `listings`, para el que se han definido algunos estilos de ejemplo que pueden verse en el fichero `config/preamble.tex` y que se usan a continuación.

Así se inserta código fuente, usando el estilo `CppExample` que hemos definido en el `preamble`, escribiendo el código directamente :

```
#include <stdio.h>

// Esto es una función de prueba
void funcionPrueba(int argumento)
{
    int prueba = 1;

    printf("Esto_es_una_prueba_[ %d][ %d]\n", argumento , prueba );
}
```

O bien insertando directamente código de un fichero externo, como en el ejemplo [A.1](#), usando `\lstinputlisting` y cambiando el estilo a `Cbluebox` (además de usar el entorno `codefloat` para evitar `pagebreaks`, etc.).

Listado A.1: Ejemplo de código fuente con un `lstinputlisting` dentro de un `codefloat`

```
#include <stdio.h>

// Esto es una función de prueba
void funcionPrueba(int argumento)
{
    int prueba = 1;

    printf("Esto es una prueba [%d][%d]\n", argumento, prueba);
}
```

O por ejemplo en matlab, definiendo settings en lugar de usar estilos definidos:

```
%
% add_simple.m - Simple matlab script to run with condor
%
a = 9;
b = 10;

c = a+b;

fprintf(1, 'La suma de %d y %d es igual a %d\n', a, b, c);
```

O incluso como en el listado A.2, usando un layout más refinado (con los settings de <http://www.rafalinux.com/?p=599> en un `lststyle Cnice`).

Listado A.2: Ejemplo de código fuente con estilo Cnice, de nuevo con un `lstinputlisting` dentro de un `codefloat`

```
1  #include <stdio.h>
2
3  #define LOOP_TIMES 5
4
5  int main(int argc, char* argv[])
6  {
7      int i;
8
9      for (i = 1; i < LOOP_TIMES; i++)
10         puts("Hola mundo!");
11 }
```

Y podemos reutilizar estilos cambiando algún parámetro, como podemos ver en el listado A.3, en el que hemos vuelto a usar el estilo Cnice eliminando la numeración.

Listado A.3: Ejemplo de código fuente con estilo Cnice, modificado para que no aparezca la numeración.

```
#include <stdio.h>

#define LOOP_TIMES 5

int main(int argc, char* argv[])
{
    int i;

    for (i = 1; i < LOOP_TIMES; i++)
        puts("Hola mundo!");
}
```

Ahora compila usando gcc:

```
$ gcc -o hello hello.c
```

Y también podemos poner ejemplos de código *coloreado*, como se muestra en el [A.4](#).

Listado A.4: Ejemplo con colores usando el estilo Ccolor

```
#include <stdio.h>

#define LOOP_TIMES 5

int main(int argc, char* argv[])
{
    int i;

    for (i = 1; i < LOOP_TIMES; i++)
        puts("Hola mundo!");
}
```

Finalmente aquí tenéis un ejemplo de código shell, usando el estilo BashInputStyle:

```
#!/bin/sh

HOSTS_ALL="gc000 gc001 gc002 gc003 gc004 gc005 gc006 gc007"

for h in $HOSTS_ALL
do
    echo "Running [$*] in $h..."
    echo -n " "
    ssh root@$h $*
done
```

A.4 Ejemplos de inclusión de algoritmos

En la versión actual (abril de 2014), empezamos a usar el paquete `algorithm2e` para incluir algoritmos, y hay ajustes específicos y dependientes de este paquete tanto en `config/preamble.tex` como en `cover/extralistings.tex` (editadlos según vuestras necesidades).

Hay otras opciones disponibles (por ejemplo las descritas en <http://en.wikibooks.org/wiki/LaTeX/Algorithm>), y podemos abordarlas, pero por el momento nos quedamos con `algorithm2e`.

Incluimos dos ejemplos directamente del manual: uno sencillo en el algoritmo A.1, y otro un poco más complicado en el algoritmo A.2.

Data: this text

Result: how to write algorithm with L^AT_EX2e

initialization;

```
while not at end of this document do
  read current;
  if understand then
    go to next section;
    current section becomes this one;
  else
    go back to the beginning of current section;
```

Algoritmo A.1: How to write algorithms

Data: $G = (X, U)$ such that G^{tc} is an order.

Result: $G' = (X, V)$ with $V \subseteq U$ such that G'^{tc} is an interval order.

begin

```
 $V \leftarrow U$ 
 $S \leftarrow \emptyset$ 
for  $x \in X$  do
   $NbSuccInS(x) \leftarrow 0$ 
   $NbPredInMin(x) \leftarrow 0$ 
   $NbPredNotInMin(x) \leftarrow |ImPred(x)|$ 
for  $x \in X$  do
  if  $NbPredInMin(x) = 0$  and  $NbPredNotInMin(x) = 0$  then
     $AppendToMin(x)$ 
while  $S \neq \emptyset$  do
  remove  $x$  from the list of  $T$  of maximal index
  while  $|S \cap ImSucc(x)| \neq |S|$  do
    for  $y \in S - ImSucc(x)$  do
      { remove from  $V$  all the arcs  $zy : \}$ 
      for  $z \in ImPred(y) \cap Min$  do
        remove the arc  $zy$  from  $V$ 
         $NbSuccInS(z) \leftarrow NbSuccInS(z) - 1$ 
        move  $z$  in  $T$  to the list preceding its present list
        {i.e. If  $z \in T[k]$ , move  $z$  from  $T[k]$  to  $T[k - 1]$ }
         $NbPredInMin(y) \leftarrow 0$ 
         $NbPredNotInMin(y) \leftarrow 0$ 
         $S \leftarrow S - \{y\}$ 
         $AppendToMin(y)$ 
       $RemoveFromMin(x)$ 
```

end

Algoritmo A.2: IntervalRestriction

Apéndice B

Herramientas y recursos

Las herramientas necesarias para la elaboración del proyecto han sido:

- PC compatible
- Sistema operativo GNU/Linux [7]
- Entorno de desarrollo Emacs [8]
- Entorno de desarrollo KDevelop [9]
- Procesador de textos L^AT_EX[10]
- Lenguaje de procesamiento matemático Octave [11]
- Control de versiones CVS [12]
- Compilador C/C++ gcc [13]
- Gestor de compilaciones make [14]

