Guía y consejos para TFGs y TFMs

Emilio Serrano

Departamento de Inteligencia Artificial de la Universidad Politécnica de Madrid

Enero de 2019

[Licencia Creative Commons](http://creativecommons.org/licenses/by/4.0/)  
Guía y consejos para TFGs y TFMs por [Emilio Serrano](http://emilioserra.oeg-upm.net/) se distribuye bajo una [Licencia Creative Commons Atribución 4.0 Internacional](http://creativecommons.org/licenses/by/4.0/).

Contenido

[1 Calendario tentativo 4](#_Toc1580117)

[2 Conceptos básicos 5](#_Toc1580118)

[2.1 Procrastinando 5](#_Toc1580119)

[2.2 Fechas, formato y secciones obligatorias 6](#_Toc1580120)

[2.3 Motivación y contenido de las secciones de la tesis o TFG 7](#_Toc1580121)

[2.4 Trabajos relacionados y bibliografía 7](#_Toc1580122)

[2.5 Extensión final 8](#_Toc1580123)

[2.6 Citar vs plagiar 9](#_Toc1580124)

[3 Primeros pasos 9](#_Toc1580125)

[3.1 Fechas de entrega 9](#_Toc1580126)

[3.2 Resumen del TFG/TFM 9](#_Toc1580127)

[3.3 Resumen gráfico 9](#_Toc1580128)

[3.4 Secciones 9](#_Toc1580129)

[3.5 Resumen de secciones 9](#_Toc1580130)

[4 Aclaraciones adicionales 10](#_Toc1580131)

[4.1 Ideas erróneas sobre el TFG/TFM 10](#_Toc1580132)

[4.2 Ejemplos de TFGs y TFMs completos 10](#_Toc1580133)

[4.3 Administración de tareas 11](#_Toc1580134)

[4.4 Describiendo desarrollos software y planificación del proyecto 11](#_Toc1580135)

[4.5 Proyectos de machine learning 12](#_Toc1580136)

[4.6 Reuniones de supervisión y entregables 13](#_Toc1580137)

[4.7 Cuándo y cómo pedir ayuda 13](#_Toc1580138)

[4.8 Consejos para la defensa 14](#_Toc1580139)

[4.9 Liberar software con licencia 14](#_Toc1580140)

[4.10 Registro de la propiedad intelectual 15](#_Toc1580141)

[4.11 Procesadores de texto 16](#_Toc1580142)

[4.11.1 Word 16](#_Toc1580143)

[4.11.2 LaTeX 18](#_Toc1580144)

[5 Bibliografía 19](#_Toc1580145)

# Calendario tentativo

A continuación, se establece un calendario de mínimos que el estudiante debe comprometerse a cumplir para poder someter su TFG a evaluación de un tribunal.

* 1er mes. Se debe completar una primera versión de la memoria del TFG/TFM que incluya los siguientes puntos.
  + Sección de trabajos relacionados con al menos 10 trabajos científicos explicados en el contexto de la contribución esperada y bibliografía incluyéndolos en formato IEEE.
  + Resumen dejando clara la contribución esperada (resumen en texto y en gráfico).
  + Título de todas las secciones y subsecciones que contendrá el TFG/TFM.
  + Al menos dos reuniones con los directores.
* 2º mes. Se debe depurar y mejorar la versión anterior de la memoria en base a la revisión, además de:
  + Sección de introducción.
  + Incluir un párrafo de resumen en todas las secciones con la información que se espera tratar en cada una de ellas.
  + Iniciar el desarrollo o experimentación del TFG/TFM.
  + Al menos dos reuniones con los directores.
* 3er mes y sucesivos.
  + Al menos una reunión cada 15 días donde se valorará si los avances son adecuados.
* Requisitos para las reuniones. Podrán ser presenciales u online. Con al menos 24 horas de antelación a la reunión se deberá dejar en trello un fichero zip con la fecha de entrega en el nombre y conteniendo tres ficheros:
  + MEMORIA: Versión de memoria del TFG/TFM anotando los cambios principales respecto a la versión anterior.
  + REVISIÓN: Informe listando comentarios de revisión anterior y describiendo cómo se han abordado cada uno de ellos, así como comentarios que sigan abiertos de revisores anteriores.
  + DUDAS: Informe de dudas y puntos a tratar en la reunión acompañado de una descripción de pasos que se han dado resolverlas de manera autónoma (enlaces/referencias revisadas, palabras clave en búsquedas realizadas, etcétera).

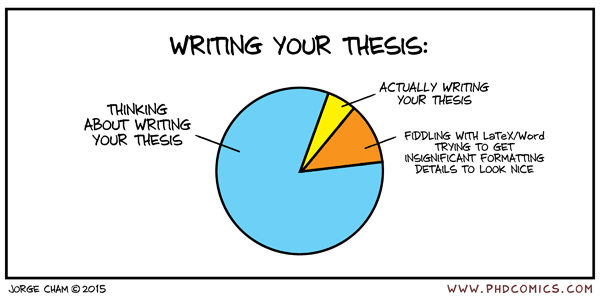
NOTA 1: el resto de la guía ofrece información y sugerencias para muchos de los puntos tratados en el calendario por lo que se aconseja revisarla en detalle.

NOTA 2: el calendario y los requisitos para las reuniones tienen una motivación muy clara: evitar la procrastinación. En un mes hay tiempo más que suficiente para completar los campos requeridos, tener claros los objetivos, y establecer una buena dinámica de reuniones y revisiones.

# Conceptos básicos

## Procrastinando

Es muy posible que te sorprendas buscando excusas para no trabajar en tu TFG o tesis, o incluso teniendo muchas molestias cuando piensas en el tema. El campo de la neurociencia ha descubierto que cuando se procrastina se activan las mismas áreas del cerebro que causan dolor físico, así que pensar en tu tesis puede (realmente) causarte dolor.



Existen muchos métodos y estudios sobre como vencer la procrastinación, siendo la [técnica pomodoro](https://es.wikipedia.org/wiki/T%C3%A9cnica_Pomodoro) una de la más populares. En ella, se plantea el trabajo como una serie de “pomodoros”, tramos de 25 minutos seguidos de 5 minutos de descanso. Aunque cualquier temporizador permite seguir el método, existen aplicaciones móviles como [Clockwork Tomato](https://play.google.com/store/apps/details?id=net.phlam.android.clockworktomato&hl=es)  para Android y extensiones de navegadores como [Marinera Asistente Pomodoro](https://chrome.google.com/webstore/detail/marinara-pomodoro%C2%AE-assist/lojgmehidjdhhbmpjfamhpkpodfcodef?hl=es) para Chrome que no sólo facilitan fijar las alarmas sino que también guardan un historial de pomodoros realizados por día.

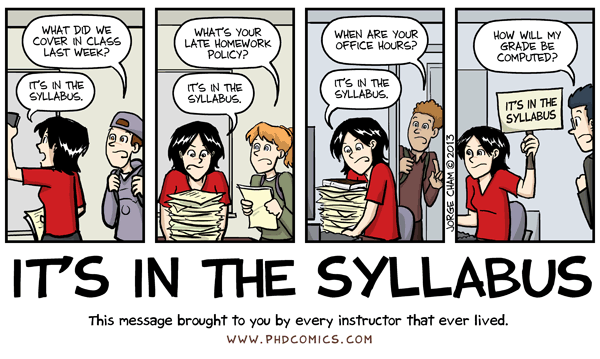
Hay varios motivos por lo que es una buena idea seguir este método cuando tienes una tarea particularmente incómoda y en la que procrastinas. En general, es bueno dividir el proyecto en una serie de tareas y centrarte en el tiempo en lugar de los resultados. Oblígate a tener contacto diario con tu tesis (digamos de al menos 25 minutos), independientemente de si consigues completar la sub-tarea que te hayas propuesto.

La buena noticia sobre la procrastinación, es que la sensación de malestar desaparece a los pocos minutos de enfrentarte al trabajo. Otra ventaja de asegurar un contacto diario con un problema es que tu subconsciente trabajará para buscar la solución.

Si te interesa la neurociencia del aprendizaje, este MOOC es muy didáctico: Aprendiendo a aprender <https://www.coursera.org/learn/learning-how-to-learn>

## Fechas, formato y secciones obligatorias

Es importante tener claros los aspectos burocráticos desde el primer día. Se tiende a pensar que son detalles a pulir una vez se tiene el contenido, pero pueden suponer la no presentación del TFG o TFM.



En el siguiente enlace están las recomendaciones de estructura y formato para TFG en la ETSIINF de la UPM:

* <http://www.fi.upm.es/?pagina=1475>

Para el Máster Universitario en Ingeniería Informática (MUII) la normativa es más compleja, pero es responsabilidad del estudiante conocerla. Particularmente si se está realizando la modalidad de tesis de máster adecuada. El siguiente enlace muestra dicha normativa:

* <http://www.fi.upm.es/?pagina=1957>

El siguiente enlace muestra las instrucciones de formato y secciones del MUII:

* <http://www.fi.upm.es/docs/estudios/muii/1957_Guia_elaboracion_TFMUII.pdf>

El siguiente enlace muestra las instrucciones de formato y secciones del MUIA:

* <http://www.dia.fi.upm.es/masteria/sites/default/files/PFM/Gu%C3%ADa%20de%20elaboraci%C3%B3n%20de%20TFM%20MUIA.pdf>

## Motivación y contenido de las secciones de la tesis o TFG

Si bien las secciones suelen venir dadas por el tipo de trabajo (ver sección 2.2), las siguientes guías ofrecen información muy interesante sobre qué se busca en el contenido de cada una de estas secciones:

* How to Organize your Thesis. Prof. John W. Chinneck.
  + <http://www.sce.carleton.ca/faculty/chinneck/thesis.html> ([traducción](http://www.sce.carleton.ca/faculty/chinneck/thesis/LASpanish.html))
* How to Write an Informatics Paper. Prof. Allan Bundy.
  + <https://sweb.inf.ed.ac.uk/bundy/how-tos/writingGuide.html>
* How to write a great research paper. Simon Peyton Jones Microsoft Research, Cambridge.
  + https://www.cis.upenn.edu/~sweirich/icfp-plmw15/slides/peyton-jones.pdf

Como dice en la primera guía, un TFG o tesis es siempre sobre “una contribución original al conocimiento”. Y en el área de la Ingeniería Informática, como dice en la segunda guía, esto suele significar que: (1) por primera vez se automatiza una tarea; (2) o, que se automatiza una tarea que ya ha sido tratada en otros trabajos, pero con una propuesta que hace esta automatización mejor en algún sentido.

Este espíritu debe estar presente en todas las secciones, desde la introducción a las conclusiones. Es muy típico que el estudiante tienda a escribir descripciones de herramientas o implementaciones en la memoria y presentación mientras que obvia su contribución al problema: aquello que es original y que se ha creado o descubierto fruto de su trabajo e investigación. “You may develop computer programs, prototypes, or other tools as a means of proving your points, but remember, the thesis is *not* about the tool, it is about the contribution to knowledge” (Chinneck , 1999).

## Trabajos relacionados y bibliografía

Los trabajos relacionados y la bibliografía son unas de las primeras secciones que repasará el tribunal y tiene una gran importancia en los trabajos académicos, hasta el punto de que pueden invalidarlos si no se consideran las referencias actuales y adecuadas o si no se cumple el formato especificado.

Para las secciones de estado del arte y trabajos relacionados, además de las guías anteriormente mencionadas, también se puede encontrar consejos muy interesantes bajo systematic review o revisión sistemática:

* Revisión sistemática general: <https://es.slideshare.net/grialusal/revisin-sistemtica-de-literatura-para-artculos?qid=ed997958-6324-47c2-b2a9-40fc2c2599f3&v=&b=&from_search=2>
* Revisión sistemática para ingeniería del software: <https://userpages.uni-koblenz.de/~laemmel/esecourse/slides/slr.pdf>
* Ejemplos de artículos de revisión sistemática:
  + [Verificación de sistemas multi-agente](https://link.springer.com/article/10.1007/s10489-017-1112-z).
  + [Evaluación de usabilidad en la web.](https://www.sciencedirect.com/science/article/abs/pii/S0950584911000607)
  + [Atributos de calidad para software](https://link.springer.com/article/10.1007%2Fs11219-011-9146-7).

Este tipo de revisiones son investigaciones en sí mismas, pero contienen información de interés sobre como buscar artículos, analizar y comparar artículos científicos.

Algunos buscadores de recursos científicos populares son:

* <https://scholar.google.es/>
* <http://www.sciencedirect.com/> (muchos recursos estarán disponibles sólo si se usa una IP de la Universidad, ya sea en laboratorios, con VPN, o con WIFIs como Eduroam o WifiUPM).
* <http://gen.lib.rus.ec/> y <http://sci-hub.cc/> son webs para el acceso libre a libros y artículos académicos, pero que pueden infringir los derechos de autor.

Es importante tener clara la contribución / pregunta de investigación, y analizar para cada trabajo encontrado: (1) como se ha tratado esa pregunta, (2) que cosas serán de utilidad en la tesis, y (3) que cosas se pueden mejorar (una vez más, la famosa contribución que aporta el TFG/TFM).

No hay reglas sobre el número de referencias a publicaciones científicas (no sitios webs de herramientas), pero se **recomiendan al menos 10 para TFG y 20 para TFM. También es importante ajustarse al formato IEEE o APA.** Si usas LaTex, el sistema bibTex permite introducir de manera rápida referencias en un fichero de extensión bib y luego mostrar la bibliografía en cualquier formato. Para Word o LibreOffice, es necesario usar el gestor de referencias y bibliografía automática que ofrecen ([enlace](https://support.office.com/es-es/article/Crear-una-bibliograf%C3%ADa-17686589-4824-4940-9c69-342c289fa2a5)). También existen gestores de referencias como Mendely o refWork que pueden simplificar la tarea de introducir referencias y mostrarlas en alguno de estos formatos.

Algunos consejos extra:

* No deberían aparecer grandes grupos de citas sin explicar/motivar su inclusión. “Algunos trabajos se basan en el uso de RNN ~~[12,13,14,15,16,17]~~”.
* No debería quedar una frase incompleta al eliminar la referencia. “~~En [15~~
* ~~] se utilizan RNN para…”~~ . Ni si quiera si se está usando el formato APA en lugar de IEEE.
* El método preferido para referenciar es nombrar a los autores: Serrano [1] utiliza… (autor único); Serrano and Botía [2] proponen (dos autores); Serrano et al. [3] explica (tres o más autores).

## Extensión final

Sobre la extensión del TFG o TFM, aunque no suele haber reglas, se **recomiendan al menos 40 páginas de cuerpo** (quitando índices, resumen, bibliografía, etc). Una vez más, la clave es tratar la contribución. Cuanto más se parezca el TFG o TFM a un pastiche de estado del arte, tutoriales de herramientas, y metodologías que no son aportación del autor; menos calidad tiene el trabajo.

## Citar vs plagiar

Citar un trabajo no permite cortar y pegar frases completas, todo debe ser reformulado con las propias palabras del autor del TFG/TFM. Un porcentaje alto de plagio en sistemas de revisión automática como turnitin pueden invalidar la tesis.

# Primeros pasos

## Fechas de entrega

Revisa las entregas y fechas del TFG o TFM, apúntalos en tu agenda, y **envía un correo a tus directores para que hagan lo mismo.**

## Resumen del TFG/TFM

Tras la primera reunión donde se ha explicado el problema, y una primera revisión de trabajos relacionados donde se ha identificado qué se va a hacer y dónde se va a contribuir; es momento de hacer el resumen del TFG/TFM. Este cambiará a la luz de revisiones más profundas y de experimentar, pero ya debe estar clara la contribución y cómo se espera conseguirla.

## Resumen gráfico

Una tarea de gran valor es construir algún tipo de figura gráfica de la contribución: una arquitectura, un diagrama de flujo de trabajo, una metodología, etcétera. Discutir sobre la figura en lugar de sobre el texto ahorrará mucho tiempo y hará la contribución mucho más clara. En UML existen diversas representaciones gráficas de la estructura y comportamiento de un sistema software que pueden inspirar esta representación gráfica de la contribución.

## Secciones

Las secciones obligatorias deberán extenderse con las subsecciones que idealmente contendrá el TFG/TFM para exponer la contribución. Incluyendo: portada con título, secciones, tabla de contenidos, índice de figuras, y bibliografía con cita de ejemplo (ver sección de editores de texto).

## Resumen de secciones

Otra tarea de gran utilidad es extender el esqueleto anterior con algunos puntos qué se tratarán en cada sección: “*…For each section and subsection, write a brief point-form description of the contents of that section. The entire outline might be 2 to 5 pages long. Now you and your thesis supervisor should carefully review this outline: is there unnecessary material (i.e. not directly related to the problem statement)? Then remove. Is there missing material? Then add. It is much less painful and more time-efficient to make such decisions early, during the outline phase, rather than after you've already done a lot of writing which has to be thrown away”*(Chinneck , 1999)***.***

# Aclaraciones adicionales

## Ideas erróneas sobre el TFG/TFM

1. *“En un TFG vale cualquier cosa, una web por ejemplo, y como se recomiendan notas superiores al 8, da igual lo que trabajes”.* Existen recomendaciones, pero son sólo eso, recomendaciones. En última instancia hay un tribunal libre de poner la nota que consideren oportuna. Personalmente he supervisado trabajos con notas de matrícula de honor, de cinco, y suspensos. Si sólo se aspira al aprobado, conviene decirlo a los directores al principio para reducir la dificultad del proyecto todo lo posible. Una contribución humilde pero bien presentada tiene más probabilidades de aprobar y obtener buena calificación que una contribución ambiciosa mal ejecutada.
2. *“Estoy muy ocupado con un trabajo / otras asignaturas; pero en semana santa /navidad me pongo al día”.* Es natural pensar que un TFG o TFM es algo que se pueda resolver a base de atracones porque es una estrategia que suele funcionar bien durante toda la carrera. Pero una tesis no es un sprint, sino una carrera de fondo. Si vas retrasando las entregas hay un alto riesgo de que no se permita presentar el TFG/TFM: por falta de calidad, por falta de tiempo de los directores para revisar la calidad de la tesis, por falta de credibilidad a la hora de justificar las horas y resultados requeridos, o por incumpliendo de la normativa si se requiere el seguimiento personalizado del tutor.
3. “*Me tienen que dar instrucciones para cada problema que me surja el TFG”.* Hay varios motivos por los que no se suele recibir instrucciones paso a paso en un TFG/TFM. La primera razón es que probablemente los directores no las saben. De saberlas, no estarías haciendo una contribución en el conocimiento y no sería una tesis. La segunda razón es que el aprendizaje autónomo y la creatividad del estudiante es fundamental en los TFGs/TFMs. Por lo tanto, debe ser capaz de tomar acciones para resolver los problemas y dudas que vayan surgiendo durante el desarrollo de la tesis.
4. *“Los directores no se quejan, así que todo está correcto”.* El nombre que va en grande en el TFG o TFM es el del autor, los directores sólo son guías. El autor, además de tener un grado importante de autonomía, debe ser autocrítico y buscar la mejora continua en su trabajo. Los TFGs, TFMs, y tesis doctorales dicen mucho de su autor porque: puede elegir la temática tratada, tiene una gran libertad sobre cómo se desarrollan estos trabajos, y sirven como colofón de una carrera.

## Ejemplos de TFGs y TFMs completos

Algunos TFGs y TFMs que he supervisado y que pueden servir de ejemplo son:

* Generación automática de música mediante tecnologías de aprendizaje profundo (<http://oa.upm.es/49682/>). TFG, calificación 10. El autor, Juan Carlos García Torrecilla, publicó una contribución en un congreso internacional de IA.
* Arquitectura multi-agente para sensorización móvil (<http://oa.upm.es/46768/>). TFM, calificación 9. El autor, Francisco Laport López, publicó una contribución en una revista científica internacional (Q1 del JCR).

## Administración de tareas

Para la gestión del proyecto se podrá usar Trello u otra alternativa, aunque los comentarios sobre la memoria serán la principal fuente de comunicación. Dropbox u otros servicios en la nube también son muy útiles para evitar envíos de ficheros pesados y llevar un historial de versiones.

*“Trello is an intuitive web-based project management application. Using the Kanban paradigmn, projects are represented by boards, which contain lists (corresponding to task lists). Lists contain cards (corresponding to tasks). Cards are supposed to progress from one list to the next (via drag-and-drop)”*

* <https://trello.com/>

Aquí una guía de Trello:

* <https://trello.com/guide/board-basics.html>

Esencialmente hay tableros (proyectos), que se componen de columnas (listas de tareas), y estas columnas tienen tarjetas (tareas). Las tarjetas deben fluir hacia las columnas de izquierda a derecha. Típicamente hay tres columnas: por hacer (todo), haciendo (doing) para seleccionar lo que se está haciendo en un momento dado, y hecho (done), para indicar que algo está hecho. Se pueden añadir otras columnas como “Revisando”, o distribuciones distintas de columnas como las de una metodología software (análisis, diseño, implementación, pruebas…).

## Describiendo desarrollos software y planificación del proyecto

El software, como la escritura de una novela, no es algo que se haga de forma lineal sino de manera iterativa e incremental. Avanzas, revisas, te das cuenta que hace falta volver atrás porque no introdujiste un personaje o la trama no era interesante, corriges, y vuelves al punto donde lo dejaste para avanzar. Por ello, para un TFG o TFM se recomienda seguir metodologías ágiles como Scrum, programación extrema (XP), o test driven development (TDD).

Sin embargo, aunque el software se desarrolle naturalmente como una serie de iteraciones, la escritura de estas puede ser confusa. Alternativas para conseguir un compromiso entre describir todas las iteraciones y presentar una metodología en cascada (que es irreal) pueden ser:

* Describir la metodología y a continuación presentar de manera “desenrollada” las grandes fases del desarrollo. Para ello, uno se puede apoyar en otra metodología no iterativa como el modelo en cascada (sub-apartados de: análisis, diseño, desarrollo, y pruebas) o el modelo en V. En la descripción se puede indicar la iteración en la que se realizó cada punto.
* Presentar al menos dos iteraciones que agrupen varias. Subapartados: análisis, iteración 1; diseño, iteración 1…; análisis iteración 2; diseño iteración 2... **Este esquema también es el recomendado para el diagrama de Gantt con la planificación del TFG o TFM.**

Los *casos de uso* e *historias de usuarios* pueden ser un buen método de aclarar y enfatizar las contribuciones del desarrollo realizado.

## Proyectos de machine learning

Machine learning (y subcampos como Deep Learning) es uno de los tópicos más populares de inteligencia artificial. Sin embargo, esta rama de la IA suele delegar en grandes conjuntos de datos que en el caso del aprendizaje supervisado deben ir etiquetados. Con ello, lo más complicado del machine learning no es el diseño y ajuste de paradigmas de aprendizaje, sino la recolección y etiquetado de datos.

Una de las mejores maneras de atacar un TFG o TFM en machine learning, que por su escasa duración no permite una recolección de datos, es utilizar datasets públicos para el diseño de un prototipo o prueba de concepto de la contribución que se pretende hacer. Algunas webs con datasets son:

* <https://medium.com/startup-grind/fueling-the-ai-gold-rush-7ae438505bc2>
* <https://machinelearningmastery.com/datasets-natural-language-processing/>
* <https://www.kaggle.com/datasets>

La evaluación de los modelos es algo que suele generar muchas dudas. En general, se realizan tres particiones de los datos: training, validation, testing. Por ejemplo, al 60, 20 y 20%. Los modelos entrenan con la partición de *training*, y deben aparecer métricas de calidad para estos datos: accuracy, precision, recall.

Pero no se usan estos resultados para decidir un paradigma de aprendizaje (árboles de decisión, redes neuronales…) ni valores de hiperparametros (Al usar métodos como la validación cruzada, el dataset de training coincide con el de validacíon.capas de la red neuronal, coeficiente de regularización…). Para eso, se evalúa el modelo entrenado con la partición de *validación*, dando nuevas métricas de calidad: accuracyV, precisionV, recallV. Estos valores permiten decidir sobre el mejor modelo. Al usar métodos como la validación cruzada, el dataset de training coincide con el de validacíon.

La tercera partición, *testing*, se usa para evaluar “el ganador” con datos nunca antes vistos y tener una medida de cómo se comportará en el mundo real. [Aquí](https://stats.stackexchange.com/questions/9357/why-only-three-partitions-training-validation-test) una discusión sobre el tema bastante clara. Existen muchas variantes perfectamente válidas, pero se deben referenciar para que se pueda revisar su veracidad y no se tome por un fallo o una mala comprensión de estos conceptos.

## Reuniones de supervisión y entregables

Se fijarán reuniones bimensuales (una cada dos semanas) o semanales de media hora para comprobar avances de la tesis o TFG. Es importante ser conciso y aprovechar al máximo este tiempo de supervisión.

El profesor Alan Bundy de la University of Edinburgh ha escrito una guía con consejos referente a estas reuniones y entregables: [http://homepages.inf.ed.ac.uk/bundy/how-tos/myStudent.html](http://homepages.inf.ed.ac.uk/bundy/how-tos/myStudent.html" \o "http://homepages.inf.ed.ac.uk/bundy/how-tos/myStudent.html)

Algunos puntos a atender son (en inglés):

* “Whenever possible please supply written material for discussion at least one day before the meeting. For instance, this might be a short progress report, a technical note or a draft dissertation chapter. Longer material, e.g. draft dissertation chapters, should be supplied several days in advance. Material should be supplied to all your supervisors.
* After a supervision meeting, email a summary of the main points, especially any actions, to all your supervisors. I will respond agreeing this is a correct record or noting any omissions or errors.
* Second or subsequent drafts should always have change bars and be accompanied by the annotated copies of the previous draft. This will enable your supervisors to focus their attention on the corrections and additions, thus making the most effective use of their supervisory time.
* **Be sure to address all comments made on a draft before submitting the next version for comment. “Addressing” a comment can include explaining why you decided not to implement the change suggested.**
* Run all written material through a spelling checker and grammar corrector before submitting it for comments. [Word provides both].
* Don’t try to hide problems from your supervisors. They are there to help you. They cannot do this if they are not aware of the problem. They will not be judgemental but will attempt to find solutions to any problems that are preventing good progress.”

## Cuándo y cómo pedir ayuda

(En inglés)

A major concern is how (and when) to get help in your studies. Dr. Roger D. Peng of the “Department of Biostatistics Johns Hopkins Bloomberg School of Public Health”, has published this great [document](https://github.com/jtleek/modules/blob/master/02_RProgramming/help/index.md) about how to get help in one of his R courses ([video](https://www.youtube.com/watch?v=ZFaWxxzouCY&amp;index=2&amp;list=PL7Tw2kQ2edvpNEGrU0cGKwmdDRKc5A6C4) and [slides](https://s3.amazonaws.com/accredible_card_attachments/attachments/54034/original/Week_1.2_-_Getting_Help.pdf) also available online). It contains excellent advice for getting help in general topics such as:

* **Try to find an answer by searching the Web (google is your friend).**
* Try to find an answer by inspection or experimentation.
* Asking questions via email is different from asking questions in person: People on the other side do not have the background information you have.
* If the answer is in the documentation, the answer will be “Read the documentation”
* It’s important to let other people know that you’ve done all of the previous things already.
* Do provide the minimum amount of information necessary (volume is not precision).
* Describe the goal, not the step.
* Be courteous (it never hurts).

It will go wrong if:

* Question was sent to the wrong mailing list.
  + Nota: Si se usa un gestor de proyectos tipo Trello, hay que asegurarse que se pide información en la tarjeta/tarea adecuada.
* Email subject was very vague.
* Question was very vague.
* Problem was not reproducible.
* No evidence of any effort made to solve the problem.
* RESULT: Recipe for disaster!

## Consejos para la defensa

* *Habla claramente y despacio*: la calidad no se mide por la cantidad de cosas que se dicen por segundo. No sirve de nada meter mucho contenido que no se puede asimilar.
* *Habla de tu trabajo/contribución*: evita centrarte en aspectos introductorios y de trabajos relacionados, especialmente si son temas que el tribunal probablemente ya conoce sobradamente. Si tu tesis se engloba en un proyecto o trabajo mayor, tiene que quedar claro donde empieza y termina tu trabajo respecto al de los demás.
* *No te enfrentes al tribunal:* el tribunal está para ayudar y hacer sugerencias de mejora. A nadie le gusta que critiquen su trabajo, pero es importante hacer un esfuerzo por entender de manera honesta sus sugerencias. Si lo estimas oportuno haz aclaraciones, pero nunca señalando errores en sus apreciaciones o entrando en debates. La defensa de una tesis es una parte importante del aprendizaje.
* *Se amable:* no proyectes nerviosismo.

## Liberar software con licencia

Típicamente se genera algún tipo de software en los TFGs o tesis. Se requiere subir el código fuente a [Github](https://github.com) (forja de software), incluyendo:

* Un nombre de proyecto representativo del software (evitar: TFG, Nombre Estudiante, etcétera).
* Un ejecutable si es posible.
* Un fichero [README.md](https://guides.github.com/features/mastering-markdown/) con código [markdown](https://guides.github.com/features/mastering-markdown/) describiendo el software, citando los autores (estudiantes y directores), y describiendo: descarga, instalación y ejecución.
* Una licencia GPL
  + LICENSE.txt con texto en <https://www.gnu.org/licenses/gpl-3.0.txt>
  + README.txt con el siguiente contenido modificando [NOMBRE PROYECTO] y autores.

[NOMBRE PROYECTO] has been developed by:

\* Nombre estudiante/s

\* Nombre director/es

The authors belong to the “Escuela Técnica Superior de Ingenieros Informáticos” of the “Universidad Politécnica de Madrid” http://www.etsiinf.upm.es/

This file is part of [NOMBRE PROYECTO].

[NOMBRE PROYECTO] is free software: you can redistribute it and/or modifyit under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or(at your option) any later version.

[NOMBRE PROYECTO]is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with [NOMBRE PROYECTO]. If not, see <http://www.gnu.org/licenses/>

## Registro de la propiedad intelectual

Para el registro de la propiedad intelectual es necesario proporcionar a los directores:

* DNI escaneado (pasaporte para estudiante extranjero).
* Memoria en Word con:
  + Portada con el título completo de la obra, autor/es y titular/es.
  + Una breve descripción del programa de ordenador.
  + El lenguaje de programación.
  + El entorno operativo.
  + Un listado de los ficheros que contiene.
  + Diagrama de flujo o UML.
* Siguientes tabla rellena en Word:

|  |  |
| --- | --- |
| NOMBRE Y APELLIDOS: | |
| DNI: | NACIONALIDAD: |
| GRUPO DE INVESTIGACIÓN: Sin grupo. | |
| DEPARTAMENTO: Sin departamento. | |
| CENTRO: Escuela Técnica Superior de Ingenieros Informáticos | |
| CARGO QUE OCUPA: Estudiante de RELLENAR TITULACIÓN | |
| DIRECCIÓN: | |
| TELÉFONO: | E-MAIL: |
| PORCENTAJE DE AUTORÍA: 70% | |

## Procesadores de texto

Se recomienda encarecidamente el uso de LaTeX. Con LaTeX perderás un día aprendiendo a definir el entorno y compilar; con Word perderás semanas peleando con el formato e introduciendo información bibliográfica. Además, los TFGs y TFMs en LaTeX tienen un aspecto más profesional que se suele reflejar en la calificación recibida.

### Word

Se puede partir de una plantilla de un estudiante que haya realizado la misma tesis o de este documento. Realizar la memoria en Word requiere conocer, al menos, las siguientes funcionalidades:

* **¡Muy importante!.** Gestor de referencias y bibliografía automática en Word. [Enlace](https://support.office.com/es-es/article/Crear-una-bibliograf%C3%ADa-17686589-4824-4940-9c69-342c289fa2a5). Existen gestores de referencias como Mendely o refWork que pueden simplificar la tarea de introducir referencias.
* Numeración de los títulos. [Enlace](https://support.office.com/es-es/article/Numerar-los-t%C3%ADtulos-ce24e028-4cb4-4d4a-bf25-fb2c61fc6585).
* Modificar y crear estilos. [Enlace](https://support.office.com/es-es/article/Aplicar-cambiar-crear-o-eliminar-un-estilo-1a2cead9-897f-48a7-9122-7849d3b5030a).
  + Al menos un estilo normal con el formato requerido (tamaño de letra, interlineado, alineación), y otro para código fuente (y pseudocódigo o salida).
  + Estos estilos permiten cambios rápidos en el formato si se estima oportuno como añadir bordes o cambar tipo de letra.

Aquí un ejemplo de estilo código fuente.

Cuadro 1. Ejemplo de código fuente.

* Borrar formato (particularmente al cortar/pegar texto con formato antes de aplicarle un estilo “normal”).
* Añadir número de página con páginas totales. [Enlace](https://support.office.com/es-es/article/Agregar-la-numeraci%C3%B3n-de-p%C3%A1gina-X-de-Y-a-un-documento-1f273c7c-661c-4444-97b5-c7ec07270f16).
* Añadir saltos de página entre secciones (ctrl+enter).
* Activar el control de cambios. [Enlace](https://support.office.com/es-es/article/Realizar-un-seguimiento-de-los-cambios-en-Word-197ba630-0f5f-4a8e-9a77-3712475e806a).
* Insertar títulos en figuras, tablas, y cuadros. [Enlace](https://support.office.com/es-es/article/Agregar-o-eliminar-t%C3%ADtulos-o-aplicarles-formato-en-Word-82fa82a4-f0f3-438f-a422-34bb5cef9c81).
* Referenciar a títulos anteriores con referencias cruzadas actualizables. [Enlace](https://support.office.com/es-es/article/Crear-o-actualizar-una-referencia-cruzada-aa35c606-34e8-4c64-b6eb-c6321d190645).
* Realizar índices automáticos de secciones, figuras y tablas. [Enlace](https://support.office.com/es-es/article/Crear-una-tabla-de-ilustraciones-c5ea59c5-487c-4fb2-bd48-e34dd57f0ec1) (la opción para figuras, tablas y otros es la misma modificando etiqueta de título, ver Figura 1).

Algunos ejemplos:

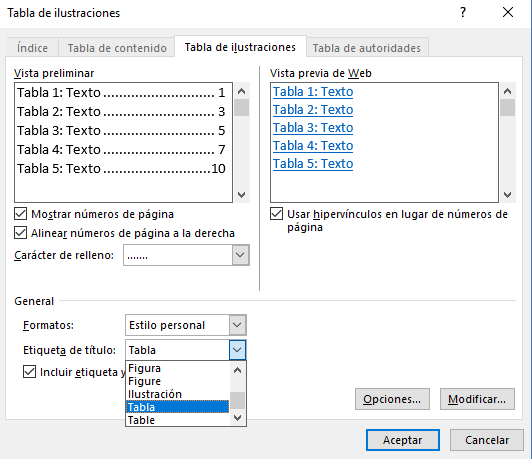


Figura 1. Insertar índice de tablas o figuras.

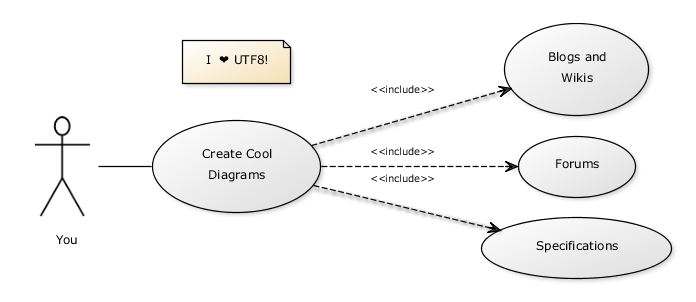


Figura 2. Ejemplo de figura.

La Figura 2 muestra un ejemplo de figura titulada, y el comienzo de esta frase una referencia cruzada a esta.

A continuación un índice de figuras:

[Figura 1. Insertar índice de tablas o figuras. 8](#_Toc474327916)

[Figura 2. Ejemplo de figura. 9](#_Toc474327917)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabla 1. Ejemplo de tabla.

Índice de tablas:

[Tabla 1. Ejemplo de tabla. 16](#_Toc516930003)

### LaTeX

LaTeX está naturalmente pensado para TFGs, TFMs, y tesis doctorales. La ventaja es que el formato queda totalmente desacoplado del contenido; y, que insertar citas bibliográficas y formulas es mucho más fácil que con MS Word. La desventaja es que hay que dedicar algo de tiempo a aprenderlo y depurar fallos de compilación. Por otro lado, el uso de MS Word también requiere cierto aprendizaje para una tesis como se vio.

Los primeros pasos en LaTeX incluyen instalar una distribución LaTeX y un editor LaTeX. El siguiente enlace muestra la instalación de MiKTeX y Texmaker en Windows, así como la generación de un primer documento:

* <https://mecatronicauaslp.wordpress.com/2013/07/25/latex-instalacion-miktex-texmaker-windows/>

También hay editores en la nube que evitan la necesidad de instalar software y que pueden facilitar la revisión compartiendo documentos. También ofrecen funciones avanzadas como autocompletar o seleccionar una línea en el pdf y que lleve a punto del código latex:

* <https://es.sharelatex.com/>
* <https://www.overleaf.com/>

Pasados estos primeros pasos, se puede partir de una plantilla y hacer modificaciones en esta. En el mejor caso, se puede pedir una plantilla usada por un estudiante anterior para modificar sobre esta. Si no, se puede partir plantillas simples[[1]](#footnote-1) e ir adaptándola al formato requerido. Aquí algunas:

* <https://es.sharelatex.com/templates/thesis>

A parte de la escritura ágil de fórmulas y la inclusión de bibliografía desde bibtex; Latex tiene una gran ventaja para la escritura de tesis: el texto comentado. Una buena práctica para organizar un texto científico es añadir un pequeño comentario al comienzo de cada párrafo indicando qué idea o que aporta ese párrafo. En revisiones se podrá seguir el hilo argumental siguiendo estos comentarios como si se tratase de código fuente y ahorrando mucho tiempo. También se podrá rescatar material que se haya decidido omitir por motivos de presentación si se quiere ampliar el texto más adelante. Word tiene comentarios también, pero tienden a desaparecer en la versión final.

# Bibliografía

Chinneck , J. W. (1999, Septiembre). *How to Organize your Thesis*. Retrieved from http://www.sce.carleton.ca/faculty/chinneck/thesis.html (último acceso 15/02/2017)

1. Se recomienda evitar plantillas muy complejas pues suelen derivar en problemas de mantenimiento al incluir grandes secciones de código no revisadas por el autor. [↑](#footnote-ref-1)