

Cómo se hace un TFG de Ingeniería Informática

Guillermo Hernández

2023-12-06

Versión en línea: <https://www.dih5.es/guia-tfg-gii.html>.

Este artículo pretende servir de guía para la realización de un Trabajo de Fin Grado (TFG) para el Grado en Ingeniería Informática (GII). Es responsabilidad del alumno asegurarse de que la información proporcionada aquí se corresponda con la actualmente vigente (y si lo deja de estar, agradecería un aviso), así como valorar con su tutor la adecuación de los planteamientos que aquí se recogen.

Lo primero que hay que tener presente es la normativa específica del título. Es mucha información, pero hay que leerla al menos una vez. Después de eso se puede trabajar mirando hacia ejemplos de trabajos buenos, y revisando de vez en cuando la normativa para ver qué están reflejando o qué carencias pueden tener. La información oficial de los TFG a nivel de Facultad actualizada está [aquí](#). Más en particular, en Informática tenemos información más específica en [la página de Diaweb sobre ello](#). Dentro de lo que se enlaza allí, son especialmente importantes la [guía del Departamento](#), así como la [normativa del Colegio de Ingenieros](#) que a partir del curso 2023/24 se puede utilizar como modelo *alternativo* para la memoria. Por último, se pueden ver algunos ejemplos del Grado con buena nota en el [repositorio Gredos](#).

Habitualmente los TFG del GII se centran en un desarrollo *software* que sirva para **demostrar los conocimientos adquiridos durante la carrera**, que, complementados con otros adquiridos en el desarrollo del mismo, dan lugar a un producto *software* que debe cumplir con los requisitos de la propuesta, y ser acorde a la dimensión del TFG en cuanto a asignatura (12 ECTS).

1 Estructura de los anexos

El objetivo general de la memoria es recoger los aspectos más importantes del desarrollo *software*, pero los artefactos completos del proceso de ingeniería se entregan en **anexos**, que conviene tener presentes desde el comienzo del desarrollo. En el esquema clásico (guía del Departamento) son los siguientes:

- **Anexo 1: Plan del proyecto software**, incluyendo planificación temporal - Gantt, metodología elegida.
- **Anexo 2: Especificación y análisis de requisitos del software**: Tablas de requisitos no funcionales, requisitos funcionales, requisitos de información..., así como el análisis de requisitos (habitualmente en casos de uso o alternativas, según metodología).
- **Anexo 3: Especificación de diseño**. Secciones típicas dentro de este, que pueden tener o no sentido dependiendo del proyecto, son:
 - Diseño del sistema: Diagramas de clases y modelo usado (Modelo-vista-controlador (MVC), Modelo-vista-presentador (MVP), Modelo-vista-modelo de vista (MVVM), ...)
 - Diseño procedimental: “Algoritmos” o procedimientos propios diseñados.
 - Diseño de interfaz: interfaces, menús...
 - Diseño de datos: Descripción de los datos del sistema (base(s) de datos(s), diagrama entidad-relación...)
- **Anexo 4: Documentación técnica de programación**. Lo ideal es integrarla en el propio desarrollo, en el formato propio de los lenguajes, y generarla al final con alguna herramienta [[javadoc](#) (Java), [sphinx](#) (python), [doxygen](#) (multilenguaje)...]. Esto es, este anexo debería, en buena medida, generarse, no escribirse.
- **Anexo 5: Manual de usuario** (incluyendo capturas, explicando el flujo de trabajo de la aplicación...).

Una buena estrategia para empezar a trabajar puede ser ir volcando todo lo producido durante el desarrollo software sobre los anexos e ir tomando nota de las cosas más importantes, que tendrán su propia sección también en la memoria.

2 Estructura de la memoria

Respecto a la memoria en sí, para la estructura se debe mirar sobre todo a ejemplos existentes en Gredos y a la normativa que se escoja seguir (la “clásica” de Diaweb o la del Colegio de Ingenieros). En el esquema clásico, la **estructura** es la siguiente:

1. **Introducción y antecedentes**: Contar en un par de páginas de qué va el proyecto y hablar de otros proyectos parecidos para motivar la importancia del desarrollo.
2. **Objetivos del proyecto**: Concretos y de forma esquematizada. Deben ser coherentes con lo que aparece en la propuesta de TFG.
3. **Conceptos teóricos**: Explicar en subsecciones independientes los conceptos más relevantes que aparecen en el proyecto. Intentar ir relacionándolo con por qué son importantes para el proyecto, evitar que sea demasiado generalista.
4. **Metodología, técnicas y herramientas**:

1. Metodología: Hablar de la metodología software que se ha seguido (cuál se escogió y por qué, descomposición en iteraciones, planificación...)
2. Técnicas: Técnicas concretas como pueden ser patrones de diseño especialmente relevantes.
3. Herramientas: Software utilizado.
5. **Aspectos relevantes del desarrollo:** Aquí es donde se recogerían los artefactos más importantes de la metodología de desarrollo, a partir de los anexos, así como otros aspectos que sea relevante contar. Una versión breve del manual de usuario (anexo 5) también puede tener cabida, pero orientado a mostrar lo más importante del desarrollo.
6. **Resultados:** Un resumen de lo que se ha logrado, que debería ser coherente con lo que aparece en la introducción y en la propuesta del proyecto. Se pueden incluir también propuestas de trabajo futuro.
7. **Bibliografía:** Se debe ir recopilando desde el principio, a ser posible utilizando la funcionalidad del procesador de textos que se emplee (para Word, [aquí](#)).