

# Guía útil para la realización del TFG

## DAM y DAW

### ¿Qué hago?

Para muchos éste es el paso más complicado. ¿Por dónde empezamos?

Lo primero que debes tener claro es qué parte de lo que se ha hecho durante los dos cursos te interesa más. ¿Una página web? ¿Una aplicación móvil? ¿Un videojuego? En un TFG se puede hacer cualquier tipo de producto que nos interese, lo importante no es el tipo de producto, sino la calidad de este y el tiempo invertido en ello.

Si no tienes una idea original, siempre puedes hacer algo “parecido” a lo que se ha hecho en clase, pero de un tamaño muy superior. Por ejemplo, si en clase se ha hecho una app para acceder a la api de películas y puntuarlas, puedes hacer lo mismo con otra api y además incluir creación de usuarios, conectividad con Facebook o instagram, un foro, meterle geolocalización, contenido multimedia... lo importante no es la temática, sino todo lo demás.

Tu tutor de TFG no te va a tirar atrás tu idea por ser una temática que no le guste (dentro de la lógica, no se van a aceptar temáticas ilegales o que atenten contra la dignidad de las personas, evidentemente), simplemente te va a orientar para que tu tfg tenga un tamaño adecuado, y en caso de que tu idea sea demasiado pequeña o grande, te dará ideas de por donde puedes recortar o ampliar.

### Tamaño del TFG

Debemos tener en cuenta que el módulo del proyecto fin de ciclo tiene 40 horas asignadas, esto significa que el TFG debe tener un tamaño adecuado para dedicar aproximadamente 40 horas. Si nos pasamos y llegamos a 60 no pasa nada, pero en ningún caso podremos quedarnos por debajo.

Esto no quiere decir que si lo que tienes en mente sean muchas más horas debas dejarlo aparcado sí o sí. Simplemente debes seleccionar aquellas partes del proyecto que vas a codificar y dejar para más adelante si te interesa el resto. Por ejemplo, si tengo en mente una aplicación que tiene control de usuarios, 15 tablas, aplicación web y móvil, geolocalización, autenticación con Facebook y otras 5 cosas más y sin ello tu aplicación no tendría sentido siempre se puede presentar el proyecto de la aplicación completa en la documentación pero seleccionar las partes factibles que se pueden hacer en esas 40 horas para presentar una aplicación que funcione y en la que se pueda ver la futura funcionalidad completa. Puede que la pantalla de autenticación en FB no funcione o que en vez de hacer todas las tablas solo utilices 5 y el diseño del resto queden hechas para más adelante si te interesa.

### Ideas “alternativas”

Yo tengo una idea alternativa, algo que nunca hemos visto en clase, o que he utilizado durante las prácticas o simplemente me interesa y que obviamente está relacionado con el ciclo, ¿Puedo hacerlo? En principio sí. Deberás contarle el proyecto a tu tutor y una vez te de luz verde adelante. Ni siquiera hace falta que el resultado de tu TFG sea una app o una web, puedes

realizar un trabajo de investigación, plantear un framework, trabajar con criptomonedas... No es lo habitual, pero se puede hacer.

## La memoria

Todas las cosas más formales se os especificarán en otro documento, pero lo que suelen tener todos los TFG es lo que indicamos a continuación. No quiere decir que todos lo deban tener, quiere decir que lo más normal en un proyecto cuyo resultado sea una aplicación o una web es lo siguiente.

### 1. Introducción:

Son unas 10 o 15 líneas en la que se cuenta de qué va el proyecto. Y deberá estar en inglés y en castellano. Podéis contactar con vuestro/a profesor/a de inglés para que os lo corrija (no para que os lo haga, obviamente)

### 2. Planteamiento del proyecto:

#### a. Resumen.

Comentar a lo largo de una o dos páginas en qué va a consistir el proyecto, qué tipo de resultado se va a obtener de él, etc.

#### b. Análisis.

¿Qué problema quiero solucionar? ¿Qué necesidad del mercado pretendo cubrir con mi proyecto? ¿Qué partes creo que debería tener?

Crearemos un listado de requisitos que debe cubrir mi proyecto.

#### c. Tecnologías utilizadas.

En base al análisis anterior, podemos pensar qué tecnologías vamos a necesitar para cubrir esas necesidades. (Si vamos a utilizar una bbdd relacional, si vamos a utilizar algún framework, qué lenguaje de programación utilizaremos etc)

#### d. Arquitectura de la solución propuesta.

Diagrama de componentes donde se pueda ver la relación entre los diferentes componentes del sistema (servidor de base de datos, servidor web/aplicaciones, app movil...). Diagrama de casos de uso y diagramas de flujo (secuencia, actividades, estados) del funcionamiento de los casos de uso principales o que requieran algo más de detalle. Cada diagrama debe llevar una breve explicación, cada componente debe tener un nombre con el que lo identificaremos a lo largo del proyecto.

#### e. Diseño de cada componente.

En este apartado incluiremos el diseño de cada componente, por ejemplo, de la base de datos, el modelo de datos más el diagrama entidad relación; de cada aplicación habrá que entregar una breve explicación de la arquitectura pensada y de la tecnología usada, más un diagrama de clases mínimo. Dependiendo del proyecto podrá ser necesario algún diagrama más, tu tutor te orientará.

#### f. Plan de pruebas

En este apartado incluiremos un listado de pruebas funcionales a realizar y el resultado esperado.

### 3. Código característico:

Como ya se ha dicho antes, una característica muy deseable del TFG es que se utilice algo que no se haya visto en clase (geolocalización, uso de wearables, frameworks etc). Se deberá explicar aquí ese código “curioso” o “interesante” que puede hacer que tu proyecto se diferencie de otros. No consiste en poner 10 páginas de código en el que no se entienda nada, sino de explicar de un modo didáctico lo que se ha hecho. Tampoco tiene por qué ser una tecnología novedosa, puede ser algo interesante hecho con java, Python etc...

4. **Manual de usuario:** con pantallazos explicaremos cómo usar las diferentes funcionalidades de la aplicación.
5. **Manual de instalación:** prerequisites del sistema operativo, versiones de los frameworks, apis, base de datos, subida al app store, hosting, ... lo que sea necesario para ponerla en funcionamiento.
6. **Presupuesto aproximado** (costes de licencias, hosting, ...), en caso de coste cero indicar los supuestos.

### Anexos deseables

Enlace a repositorio GitHub (u otro control de versiones) con el código, incluyendo scripts sql de creación de la base de datos.

El código debe estar comentado con Javadoc y generada la documentación automática en formato html.

El código debería tener alguna clase con tests unitarios. Incluirlos junto con su resultado.

Resultados de las pruebas funcionales.