

Lo primero que debes hacer es enfocar el TFC como la **documentación (y creación) de un PROYECTO INFORMÁTICO basado en la construcción de una SOLUCIÓN INFORMÁTICA en base a un PROBLEMA o NECESIDAD DE UN CLIENTE.**

Si el cliente no existe (al ser tú mismo) o no lo tienes claro, deberás crear un cliente ficticio o referirte a él mediante construcciones impersonales como "se pide", "se solicita", "se requiere" o "el proyecto informático requiere" o "la solución informática requiere"...

Con esto, dada una necesidad detectada y documentada, esta se ANALIZA (1), se DISEÑA (2) una solución, se IMPLEMENTA (3) en forma de proyecto informático (en este caso software), se PRUEBA (4), se DOCUMENTA (5), se IMPLANTA/DESPLIEGA (6) y se EXPLOTA/MANTIENE (7).

Si nos centramos en las dos primeras etapas (1 y 2), estas son las tareas que deberías priorizar.

Son bastante subjetivas y dependen mucho de los autores que consultes, por lo que esto es solo una propuesta/recomendación abierta a tu aportación personal.

- (1) Fase de Análisis (de requisitos)
 - ¿En qué consiste? En extraer qué necesita el cliente
 - ❑ Es importante evitar en este punto especificar la(s) tecnología(s) concreta(s) en la(s) que se va a implementar la solución (hardware/software).
 - ❑ *Por ejemplo, si el cliente QUIERE una app para iPhone especificaremos algo como que "sus potenciales usuarios usarán iPhone" o que "solo iPhone ofrece ciertas funcionalidades/garantías que el cliente NECESITA".*
 - ❑ Evitaremos, por tanto, mencionar un sistema operativo, un SGBD, un lenguaje o una plataforma de hardware concreta en este punto.
 - Documentación de esta fase
 - ❑ Diferenciar entre requisitos funcionales (qué necesita) y no funcionales (cómo lo necesita).
 - Requisitos funcionales
 - Se suele documentar como un listado de funcionalidades, acompañado por uno o varios diagramas.
 - El diagrama de casos de uso (UML) sea quizás la forma más usada, pero cualquier otra alternativa es bienvenida.
 - Requisitos no funcionales
 - Se suele documentar como un listado de exigencias relativas a software/hardware concretos (debidamente justificados), tiempos de ejecución, plazos de entrega y costes máximos y mínimos (si los hay).
 - Este apartado queda abierto a prácticamente cualquier cosa que nos "exija" el cliente, desde tener "x" personas dedicadas a su proyecto hasta tenerlo terminado en 1 mes.

- (2) Fase de Diseño (de la solución)
 - ¿En qué consiste? En crear una solución a partir de un problema o necesidad.
 - Revisada y estudiada al detalle toda la documentación del punto anterior (fase de análisis) se enumeran, a modo de conclusión, las especificaciones del sistema (sw y hw) y los tiempos.
 - Incluimos (aquí sí) y JUSTIFICAMOS, el lenguaje o lenguajes que usaremos, el/los SGBD, y todas las elecciones necesarias de SW y HW.
 - También incluimos en esta fase la metodología (ágil/clásica/híbrida) y la planificación del proyecto (en fases/tiempos), de la manera más detallada posible.
 - Documentación de esta fase
 - Especificaciones del sistema
 - Se suele documentar como un listado de especificaciones sw y hw, acompañado por uno o varios diagramas.
 - La(s) base(s) de datos suelen documentarse en forma de diagrama(s) Entidad-Relación (clásicos o en UML).
 - El código fuente (si decide trabajar con orientación a objetos) se documenta en forma de diagrama de clases (UML).
 - Metodología / Equipo humano
 - Qué metodología y equipo humano requerimos y por qué.
 - Planificación
 - Qué planificación temporal proponemos y por qué.
 - Presupuesto (como no se trata de un proyecto real, no incluiremos este punto)

Con esas dos fases bien claras, pasaríamos a la implementación.