

Guía para el desarrollo de TFG

Grado en Ingeniería Multimedia



Trabajo Fin de Grado

Autor:

José Vicente Berná Martínez

Tutor/es:

José Vicente Berná Martínez

mayo 2019



**Guía para el Desarrollo de los Trabajos Fin de Grado y Fin de Master.**

Este documento ha sido confeccionado por José Vicente Berná Martínez con el fin de facilitar a los alumnos de grado y máster el desarrollo de sus trabajos finales. Contiene además de los estilos, una descripción de los posibles apartados que pueden ser incluidos en el trabajo. Los alumnos pueden utilizarlo como plantilla, sustituyendo la explicación de cada apartado por sus propios contenidos, conservando formatos y estilos. El alumno puede personalizar el documento.

**Versión del documento**

2019.05.10

**Licencia**

Se permite la reproducción, distribución y comunicación pública de la obra, incluso con fines comerciales siempre y cuando reconozca y cite la obra de la forma especificada por el autor o el licenciante.

https://licensebuttons.net/l/by/4.0/88x31.png

# Resumen

Incluir en la memoria un resumen de máximo 500 palabras. Este resumen será el mismo que se enviará junto a la documentación en la entrega del TFG. Este resumen normalmente se creará una vez finalizado el trabajo y con la visión del trabajo completo.

También puede ser traducido a las tres lenguas oficiales de la Universidad de Alicante: español, inglés y valenciano.

# Motivación, justificación y objetivo general

Esta sección es una breve introducción al proyecto que se quiere realizar. La idea es presentar la motivación, el por qué, del alumno para desarrollar el tema escogido. Puede ser una motivación personal, puede ser por intereses social, porque al final escogió el tema de entre los que quedaban disponibles, el que sea. La idea es dejar claro que el alumno está preparado para abordarlo, que es un tema que interesa, para él o para una parte de la comunidad. Se pueden abordar respuestas a estas preguntas como:

* ¿Cómo saltó la idea?
* ¿Es interesante el tema?
* ¿Es en principio viable?
* ¿Servirá para algo?
* ¿Me motiva el tema?
* ¿Es parte de mis atribuciones?
* ¿Es original o es una mejora de algo que ya existe?
* ¿Cuándo lo termine demostrará mis habilidades como ingeniero?

Este apartado es un tanto personal y describe al final cual es la pretensión a grandes rasgos del proyecto. En este momento del documento todavía no se tiene claro que es lo que se va a desarrollar en el proyecto, solo la intención de lo que se quiere hacer, pero todavía no se ha estudiado el problema en profundidad y no se puede tener una visión clara del resultado.

La motivación de este documento es proporcionar a los alumnos una plantilla base donde tengan recogidos todos los posibles contenidos a desarrollar con una explicación de cada uno de ellos junto además una plantilla bien formateada. En este documento iré explicando las distintas características para crear formatos correctos y así facilitar la escritura de grandes documentos.

Este apartado puede desarrollar de 1 a 3 páginas.

# Agradecimientos

Esta sección se utiliza para escribir unos agradecimientos breves a todas aquellas personas, organizaciones, empresas, etc., que han contribuido a que el alumno haya alcanzado el final de su grado o que han colaborado en su trabajo: padres, familiares, pareja, tutor, amigos... Es voluntaria y completamente personal.

Este apartado suele ser 1 página.

# Citas

Incluir una o varias citas que demuestren el carácter de tu trabajo, de tu personalidad y actitud en la vida.

*Cuando tenía 17 años, leí una cita que decía algo como: “Si vives cada día como si fuera el último, algún día sin duda estarás en lo cierto”. Causó una honda impresión en mí, y desde entonces, por los últimos 33 años, me he mirado en el espejo cada mañana y me pregunto: “Si hoy fuera el último día de mi vida, ¿querría hacer lo que voy a hacer hoy?” Y si la respuesta es no por demasiados días seguidos, sé que necesito cambiar algo.*

*Steve Jobs*

*Si algo se vuelve demasiado complicado, se atasca o no te convence: reinicia.*

*José Vicente Berná*

Índice de contenidos

[Resumen 3](#_Toc528232667)

[Motivación, justificación y objetivo general 4](#_Toc528232668)

[Agradecimientos 5](#_Toc528232669)

[Citas 6](#_Toc528232670)

[Índice de figuras 9](#_Toc528232671)

[Índice de tablas 10](#_Toc528232672)

[1. Introducción 11](#_Toc528232673)

[2. Estudio de viabilidad 13](#_Toc528232674)

[2.1. Análisis DAFO 13](#_Toc528232675)

[2.2. Lean Canvan 14](#_Toc528232676)

[2.3. Análisis de riesgos 15](#_Toc528232677)

[3. Planificación 16](#_Toc528232678)

[4. Estado del arte. 18](#_Toc528232679)

[4.1. Antecedentes. 19](#_Toc528232680)

[5. Objetivos 20](#_Toc528232681)

[6. Metodología 21](#_Toc528232682)

[7. Análisis y especificación 23](#_Toc528232683)

[8. Diseño 25](#_Toc528232684)

[8.1. Diseño de la persistencia 25](#_Toc528232685)

[8.2. Diseño arquitectura conceptual 26](#_Toc528232686)

[8.3. Diseño API Rest (si hay) 27](#_Toc528232687)

[8.4. Diseño arquitectura tecnológica Front/Back-end 28](#_Toc528232688)

[8.5. Diseño Interacción o Experiencia de Usurio 29](#_Toc528232689)

[8.6. Diseño Interfaces 31](#_Toc528232690)

[8.7. Guías de estilos 35](#_Toc528232691)

[8.8. Diseño de pruebas y validación 35](#_Toc528232692)

[9. Implementación 37](#_Toc528232693)

[10. Pruebas y validación 38](#_Toc528232694)

[11. Resultados 39](#_Toc528232695)

[12. Conclusiones y trabajo futuro 40](#_Toc528232696)

[Referencias 42](#_Toc528232697)

[Apéndice I 44](#_Toc528232698)

# Índice de figuras

[Figura 1. Opción Inicio>Numeración para crear listas numeradas 11](#_Toc495787305)

[Figura 2. Esquema de un análisis DAFO. 13](#_Toc495787306)

[Figura 3. Cuadro para el análisis Lean Canvan 14](#_Toc495787307)

[Figura 4. Ventana para seleccionar la referencia que se desea insertar. 15](#_Toc495787308)

[Figura 5. Ejemplo de diseño de BD 26](#_Toc495787309)

[Figura 6. Diseño arquitectura conceptual panel IoT. 27](#_Toc495787310)

[Figura 7. Especificación de la Open API de Smart University 28](#_Toc495787311)

[Figura 8. Stack tecnológico del proyecto Smart University 29](#_Toc495787312)

[Figura 9. Ejemplo de la diferencia entre usable, accesible y experiencia de usuario. 30](#_Toc495787313)

[Figura 10. Ejemplo de wireframe con Balsamiq. 32](#_Toc495787314)

[Figura 11. Mockup desarrollado con Justinmind 33](#_Toc495787315)

[Figura 12. Diseño desarrollado con Sketch. 34](#_Toc495787316)

[Figura 13. Ejemplo de mapa de interacciones. 34](#_Toc495787317)

[Figura 14. Paleta de colores Unviersidad de Cádiz 35](#_Toc495787318)

[Figura 15. Ejemplo de ejecución de varias invocaciones a servicio. 36](#_Toc495787319)

[Figura 16. Primer boceto de la aplicación. 44](#_Toc495787320)

# Índice de tablas

[Tabla 1. Planificación temporal TFG 14](#_Toc495786392)

# Introducción

Este es el primer capítulo de tu trabajo y lo primero que hay que hacer es explicar el contexto donde se ubicará tu trabajo. Un tribunal completamente ajeno a tu trabajo debe comprender el contexto de tu proyecto, la introducción debe explicar el marco informal sobre el que se desarrolla, el mundillo sobre el que versa tu proyecto, quienes están interesados en él, que solución puede implicar, que impacto puede tener lo que tu estas proponiendo.

Por ejemplo en este caso, uno de los grandes problemas de los alumnos es que no identifican los apartados que debe contener su trabajo fin de grado, a pesar de que lo han estudiado a lo largo de la carrera aunque sea en asignaturas diferentes. Otro de los problemas es que acaban gastando mucho tiempo en el formateo correcto del documento porque no suelen invertir tiempo en conocer las herramientas con las que trabajan (Microsoft Word, Open Office…). El uso correcto de los estilos facilita mucho la redacción de los documentos, y más cuando son de gran tamaño.

Todos los títulos de primer nivel deben ir en el estilo “Titulo 1”. Además a partir del capítulo introducción deben ir numerados utilizando la opción “Inicio>Numeración” como muestra la Figura 1.

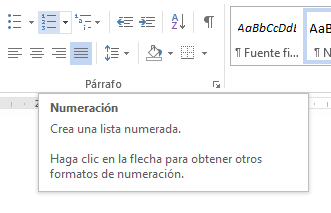


Figura 1. Opción Inicio>Numeración para crear listas numeradas

(Fuente propia)

Los títulos de segundo nivel deben utilizar el estilo “Titulo 2” y además también numerarse pero utilizando un segundo nivel, esta opción está disponible también dentro de “Numeración” para crear listas numeradas.

Si es necesario se puede utilizar hasta un tercer nivel dentro del texto aunque en este caso hay que considerar si es mejor dividir el capítulo en varios. Hasta un nivel 3 es aceptable, más de 3 niveles de numeración que originen capítulos del estilo “3.5.4.6.5. Titulo capítulo” no suele ser recomendable y lo que indica es que ese capítulo posiblemente debería haberse desglosado en varios capítulos más.

Cada apartado de nivel 1 debe comenzar en una nueva página, por lo que la final de la sección anterior introducir un “Insertar>Salto de página”.

La introducción puede desarrollar 2 o más páginas, dependerá de la complejidad.

# Estudio de viabilidad

En esta sección se puede hacer un estudio de viabilidad. Antes de arrancar con el resto del proyecto se puede analizar un poco si el proyecto en sí mismo, sus pretensiones u objetivos son viables, son pertinentes, son necesarios. También es muy adecuado indicar los riesgos y los planes de contingencia.

Para hacer este estudio de viabilidad se pueden utilizar herramientas como las siguientes, aunque el alumno puede seleccionar la que más le guste o crea conveniente, incluso valorando opciones.

El tamaño total de esta sección dependerá del contenido.

## Análisis DAFO

El DAFO (de las iniciales de Debilidades, Amenazas, Fortalezas y Oportunidades) es una metodología de estudio de la situación de un proyecto, analizando sus características internas (Debilidades y Fortalezas) y su situación externa (Amenazas y Oportunidades) en una matriz cuadrada como muestra la Figura 2.



Figura 2. Esquema de un análisis DAFO.

(Fuente http://egesoftware.blogspot.com.es)

Las figuras siempre deben contener un pie de figura explicando su contenido y además deben estar siempre referenciadas desde el texto tal y como ocurre en el párrafo anterior. No se deben incluir imágenes, figuras, fotos u otros por el simple hecho de rellenar y hacer “bonito”. Todas las figuras deben indicar la fuente de la que han sido extraídas o bien si son propias.

## Lean Canvan

Lean Canvan es otra de las herramientas que pueden ser idóneas para analizar un producto que está siendo creando y que tiene carácter innovador, ya sea porque es una solución nueva a un problema ya existente o bien porque es una mejora sobre otras soluciones que ya existen. La idea de Lean Canvan es analizar el proyecto desde diferentes perspectivas de interés como muestra la Figura 3, no solo desde el desarrollo o los costes.

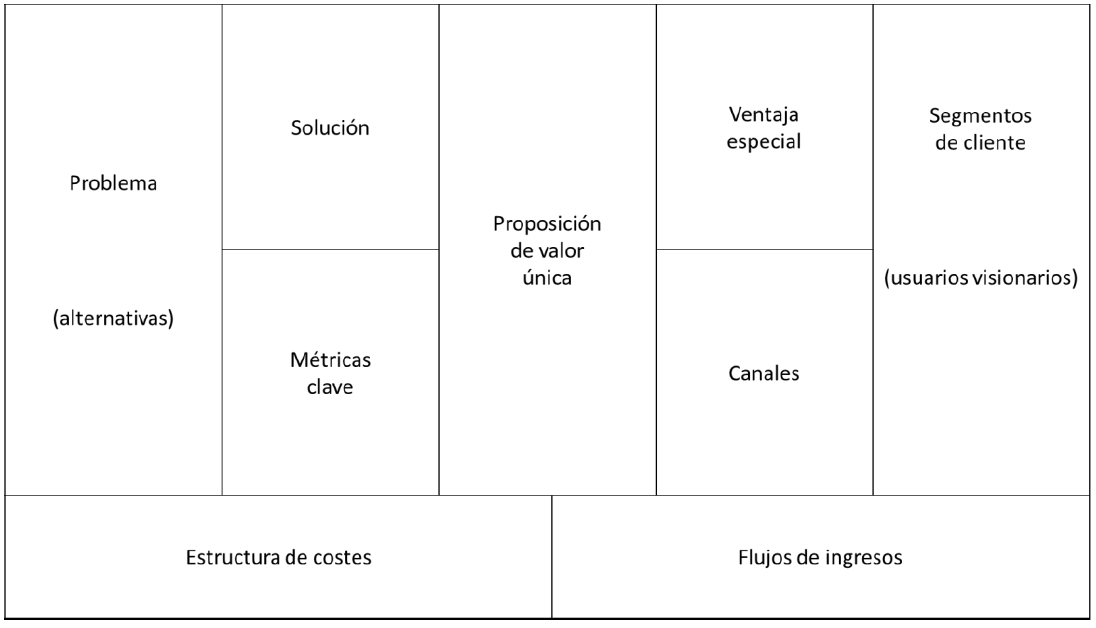


Figura 3. Cuadro para el análisis Lean Canvan

(Fuente propia)

Para mantener adecuadamente el formato insertar los pie de figura desde la opción “Referencias>Insertar título”. Después del título que se inserta, como por ejemplo el anterior de “Figura 3….”, insertar otra línea donde índice la fuente de la figura (es decir, el creador). Para hacer referencias en el texto, utilizar la opción “Referencias>Referencia cruzada”, aparecerá una ventana como muestra la Figura 4, entonces seleccionar “solo rótulo y número” e insertar.

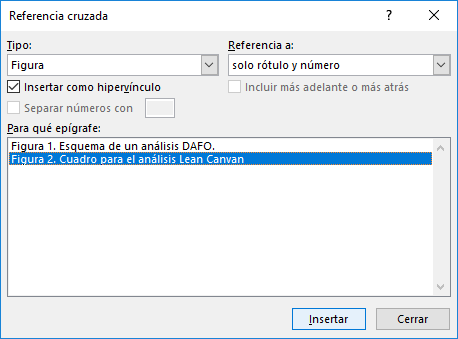


Figura 4. Ventana para seleccionar la referencia que se desea insertar.

(Fuente propia)

De esta manera si más adelante insertamos una nueva imagen, la numeración se mantendrá correlativa y los índices funcionaran adecuadamente.

## Análisis de riesgos

A lo largo de un proyecto que es de carácter anual (12 créditos equivalen a unas 300 horas), es adecuado que antes de iniciar nada el alumno pueda tener en consideración riesgos como son el desconocimiento de tecnologías, posibilidad de enfermar levemente durante el curso, sufrir variaciones de carga de trabajo debido a asignaturas, etc. La propuesta de planes de contingencia puede facilitar el éxito del trabajo fin de grado.

# Planificación

Una buena planificación desde el principio puede ayudar también en la consecución del proyecto. Además permitirá una vez terminado analizar junto a los resultados la adecuación de la planificación al desarrollo completo del proyecto. Esto dará experiencia a los alumnos ya que al planificar se realiza una estimación de costes temporales en función de unas capacidades o habilidades supuestas. Después, al analizar los resultados podrás comparar si sub-estimaste tus capacidades o no. La planificación temporal además te ayudará a seguir unos pasos en el tiempo, no retrasarte y tampoco invertir demasiado tiempo en aquello que no debe.

A continuación se proporciona la Tabla 1 como ejemplo de una planificación para un proyecto de desarrollo de TFG convencional. Dependerá de los apartados de tu proyecto y de tus propias habilidades pero esta distribución temporal permitiría el desarrollo del proyecto en plazos para su entrega en la convocatoria de junio. Se hace una agrupación de varios capítulos, el alumno debería hacer el esfuerzo de deducir cuanto tiempo debería emplear en cada uno de ellos y hacer una planificación detallada.

Tabla 1. Planificación temporal TFG

|  |  |  |
| --- | --- | --- |
| **Contenidos** | **Tiempo total** | **Fecha límite fin** |
| Motivación, justificación, objetivo general, Introducción  Estado del arte | 1 mes | 15 noviembre |
| Objetivos  Metodología  Análisis y especificación  Presupuesto, estimaciones, planificación | 1 mes | 20 diciembre |
| Diseño | 2 meses | 25 febrero |
| Implementación | 1 mes | 15 abril |
| Pruebas y validación  Resultados  Conclusiones y trabajo futuro  Referencias, bibliografía y apéndices  Agradecimientos, citas, índices | 1 mes | 1 junio |

Este ejemplo de planificación es para tener una versión de proyecto legada la fecha de 1 de junio y así poder entregar adecuadamente para esa convocatoria. Cada fecha límite marca el plazo en el que debería ser entregada una versión de la memoria con esos capítulos completados al tutor para que este pueda corregirla y proporcionar feedback al alumno.

Las tablas también han de contener un título de la forma “Tabla X…” y han de estar referenciadas en el texto. El formato de la tabla debe conservarse constante a lo largo de todo el documento y deben ser claras y visibles al imprimirse en blanco y negro.

# Estado del arte.

El estado del arte es una de las secciones más importantes del trabajo y debe ser desarrollado antes si quiera de diseñar ningún tipo de solución, ya que permitirá al alumno conocer más información tanto sobre el problema como de las posibles soluciones actuales o en proceso.

La idea del estado del arte es recoger toda aquella información que pueda resultar de interés para el desarrollo del proyecto. Puede por tanto contener varios apartados dentro del propio estado del arte. Si la investigación es muy extensa puede ser recomendable dividir la sección estado del arte en varias secciones como “Estudio de la problemática”, “Antecedentes”, “Tecnologías para el desarrollo”. Todo depende de lo extensa y profunda que sea este apartado.

La sección de estado del arte es un estudio sobre la problemática concreta que aborda el proyecto. Antes de proponer una solución a un problema es necesario conocer en profundidad tanto el problema como su contexto, los actores que intervienen, los recursos disponibles, el impacto de dicho problema. Por ejemplo, si tenemos un TFG que trata sobre cómo hacer un buen marketing digital de una plataforma web concreta, sería bueno estudiar los fundamentos del marketing digital y para ello podríamos recurrir a diversos libros como el de José Sixto García [1], analizando cuales pueden ser dichos factores a favor. También se podrían buscar ejemplos de otros trabajos relacionados con el tema y analizar cuáles fueron las propuestas y los resultados [2]. Esta sección debería estar plagada de referencias a textos, libros, artículos y otros materiales de relevancia. Hay que tener mucho cuidado con referenciar posts o artículos online de sitios que no tengan reputación. Se pueden utilizar materiales online pero las fuentes tienen que ser de calidad, es decir, que el autor tiene que tener experiencia demostrada en el tema, ser un medio de comunicación fiable y su contenido ser coherente y creíble. Las fuentes pueden estar en cualquiera de las lenguas oficiales de la Universidad de Alicante (Español, Inglés o Valenciano). Todas las referencias deben estar recogidas en el apartado final de referencias y bibliografía. Para insertarlas en el texto siempre se utilizará “Referencias>Referencia cruzada” para insertarla.

El apartado “Estado del Arte” sirve para recoger todo el *know who* del tema. Pensad que puede hablarse de marketing, pero también podríamos hablar sobre la problemática de detectar si un balón entra o no en una portería mediante el procesamiento de imagen, sobre el desarrollo de actividades de ocio digital por parte de personas con movilidad reducida o cualquier otro tema. Por ello hay que recoger todo el conocimiento que se pueda para ser experto en la materia y así poder después hace una propuesta sensata y útil.

Muchos de los grandes errores que los alumnos comenten en su propuesta de un TFG/TFM se derivan precisamente de tomar decisiones sin conocer suficientemente el problema al que se enfrentan, su contexto, sus necesidades, limitaciones físicas o incluso aspectos legales. Todo ello debe recogerse en esta sección para asegurarnos de que el trabajo será realista.

También es importante recoger propuestas de solución alternativas, describirlas e incluso compararlas, extrayendo sus principales características. Esto puede marcar tendencias del sector que en nuestra propuesta tendremos que implementar. Por ejemplo, es muy común en cualquier aplicación que realiza fotos tener acceso directo a funcionalidades de compartir, esto implica que aunque inicialmente no esté dentro de nuestro problema igual debemos considerar incluir funcionalidades como estas para respetar la norma de facto del sector.

## Antecedentes.

Si el estado del arte trata sobre la problemática del trabajo, los antecedentes son una sección que recoge los trabajos y experiencia previos del alumno con la problemática. Por ejemplo, si el alumno ha realizado unas prácticas en un curso anterior o en una asignatura a partir de las cuales ya se ha avanzado en la solución o tiene una versión previa de la solución que servirá como punto de partida, este es el apartado donde se explicará. Se debe dejar muy claramente explicado cual es el trabajo previo realizado para que el tribunal tenga clara después cual ha sido la aportación del alumno en su trabajo respecto a lo ya existente.

Este apartado de estado del arte y su contenido puede extenderse todas las páginas que sea necesario, no se recomienda ni mínimo ni máximo pues dependerá de la profundidad con la que se trate este capítulo.

# Objetivos

Definir cuál es el objetivo principal del proyecto, u objetivos principales, y los sub-objetivos de cada uno de estos. Los objetivos definen cuales son las metas de nuestro proyecto. No son funcionalidades. A partir de estos objetivos se dispondrán después recursos, diseños y tomas de decisión. La consecución de los objetivos supondrá la superación de las dificultades o problemas inicialmente planteados.

Los objetivos definen la pretensión del proyecto (no la funcionalidad), y es interesante que cumplan una serie de características para asegurarnos de que nos aportan valor. Una estratégica interesante es hacer que cumplan el principio SMART propuesto por Doran [4], según la cual cada objetivo debe ser:

* Specific (Específico): debemos especificar y concretar al máximo cada uno de nuestros objetivos, con el mayor nivel de detalle posible. Si no sabes si un objetivo tiene suficiente detalle, hazte esta pregunta: con los datos que aportas, ¿podría saber otra persona ajena al proyecto qué vas a hacer y cómo?
* Measurable (Medible): en ocasiones resulta difícil pensar en un objetivo que podamos medir, pero tiene que quedar muy claro que nuestras metas deben serlo. Solo de esa manera podremos cuantificar si alcanzamos los resultados esperados en función de los indicadores que establezcamos para cada uno de ellos. Recuerda siempre que tus objetivos sean cuantitativos; si no se puede medir, será difícil saber si has alcanzado el éxito.
* Attainable (Alcanzable): está claro que nuestro objetivo ha de ser alcanzable en la situación en la que nos encontremos. Fijar objetivos que son imposibles de conseguir no tiene ningún sentido, y además puede mermar nuestra moral. Piensa en tus objetivos como en retos ambiciosos, pero posibles. Además, siempre debe existir la posibilidad de que puedan ser reajustados si nuestro entorno sufre cambios. Intenta evitar objetivos “globales”, es decir, de los que están destinados a alcanzar “toda la población mundial”.
* Realistic (Realista): el objetivo debe ser realista y estar orientado a resultados objetivos y concretos. Como ya hemos mencionado, que estén dentro de nuestras posibilidades, por lo que hemos de tener en cuenta factores ajenos como nuestro entorno o los recursos de los que disponemos. Aquí los análisis DAFO o de riesgos anteriores pueden ser interesantes,
* Timely (Tiempo): todo objetivo tiene que tener un horizonte temporal. Por ejemplo: si quieres conseguir 10.000 visitas a tu sitio web... quieres lograrlo ¿en un mes? ¿en un año? Hay mucha diferencia, por eso hemos de definir muy bien el horizonte de tiempo para cada uno de nuestros objetivos. Si es un objetivo a largo plazo, es recomendable hacer un esquema o línea temporal donde podamos marcar los distintos hitos o etapas que nos llevarán a conseguir nuestra meta final.

No olvides que si uno de tus objetivos está ligado a la monetización de tu solución (ya sean soluciones de pago, freemium, licencia, venta del producto, etc), debe estar contemplado como un objetivo. Si no es un objetivo después no deberían aparecer funcionalidades como por ejemplo el pago a través de una pasarela de la licencia.

Este capítulo se extiende habitualmente 1 o 2 páginas. Pueden ser más.

# Metodología

Describir los procedimientos mediante los cuales se va a a desarrollar el trabajo, esta sección describe “el cómo” se va a trabajar. Dependiendo del tipo de proyecto se podrán utilizar unas metodologías u otras o incluso combinación de varias. Metodologías ágiles tipo Scrum o Lean pueden ayudar a asimilar cambios y dinámicas en el transcurso del trabajo mientras que otras más tradicionales como la Cascada puede ayudar a eliminar la incertidumbre desde el principio generando mayor confianza en el proyecto. El alumno debe describir cómo será su procedimiento. Utilizar una metodología ágil no quiere decir no generar documentación ni que no haya fases de análisis, especificación o diseño. Todas estas fases en mayor o menor medida se dan a lo largo de un proyecto, la metodología solo indica cuándo se desarrollarán y la profundidad y completitud de estas fases. Por ejemplo en una metodología Scrum se describen historias a partir de las cuales de definen funcionalidades que son diseñadas y planificadas en mediante sprints. En una metodología en cascada todo el análisis, especificación y diseño se realizan en bloque y por completo al inicio del proyecto para luego pasar a desarrollar la solución. Si se trata de una investigación donde se ha de generar un nuevo algoritmo se puede utilizar el método hipotético-deductivo para describir soluciones y luego demostrarlas, o si se basa por ejemplo en la imitación de un organismo celular se podría utilizar el método inductivo.

La metodología o metodologías a emplear las define en alumno, según le sea más cómodo. Se trata de describir cómo va a trabar para que no sea una improvisación continua y se ordene la secuencia de pasos que va a realizar.

También es una buena práctica definir e indicar que herramienta o herramientas serán utilizadas para gestionar el proyecto, contabilizar las tareas realizadas y llevar un control de aquellas pendientes. Estas herramientas deberán permitir desarrollar el trabajo según la metodología escogida. Trello, Microsoft Project, JIRA, Toggle, son solo algunas de estas herramientas, justifica además tu elección.

Este apartado puede extenderse 1 o 2 página, dependiendo de la justificación o desarrollo que se haga incluso más.

# Análisis y especificación

En esta sección nos centramos en el problema concreto que queremos solución y cuál va a ser el alcance la solución a desarrollar. Dicho alcance depende de muchos factores como por ejemplo los usuarios que lo utilizarán, la población a la que está orientada, la forma en la se distribuirá en función de sus necesidades, etc. En general y para no dejarnos elementos sin definir es bueno utilizar alguna herramienta de análisis y especificación, como las estudiadas durante la carrera de IEEE 830. Seguir esta plantilla facilita una definición sistemática de aquellos aspectos que son necesarios tener en cuenta a la hora de diseñar nuestra solución software. Ahora sí que se definirán aspectos relativos a requerimientos de rendimiento, de capacidad, funcionales, de interfaz con sistemas de terceros, limitaciones. Puede utilizarse como guía el artículo [5].

Como resultado principal de esta sección se ha obtener una lista de requerimientos funcionales y no funcionales deseados, un conjunto de tablas. Estos requerimientos deben estar identificados con un identificador único de forma que después, en la fase de diseño, se pueda indicar claramente como están siendo resueltos. Los requerimientos funcionales pueden ser identificados con los IDs RF1, RF2, etc., mientras que los no funcionales pueden ser identificados con RNF1, RNF2, etc. No pueden incluirse requerimientos que no estén debidamente justificados y relacionados con el problema, de igual forma que luego el diseño posterior responderá exclusivamente a las necesidades de requerimientos establecidos.

Algunos requerimientos deberían estar de serie en cualquier sistema como son:

* El control de errores: aportar mecanismos que permitan informar tanto al usuario como a los administradores de los errores que se hayan producido, a cualquier nivel, ya sean de acceso a BD, de acceso a disco, de acceso a algún servicio de tercero, errores de login, etc.
* Auditoría: deberían aportarse mecanismos suficientes para proveer de mecanismos de auditoría a los administradores y así poder observar comportamientos sospechosos de los usuarios, de agentes externos, mal funcionamiento del propio sistema. Estos mecanismos son históricos y logs a diferentes niveles como sistemas y aplicación.
* Alineación con la legalidad: dependiendo de la información a almacenar es necesario cumplir con los requerimientos legales marcados por la legislación a la que se verá sometida. Se necesario detectar estos requerimientos y dejarlos especificados claramente ya que pueden condicionar nuestra infraestructura, desarrollo, funcionalidades, etc.
* Requerimientos no funcionales: los relacionados con la mantenibilidad del sistema como son su flexibilidad, su capacidad, su modularidad. Definirla en función de las necesidades reales. Si un software está destinado a ser utilizado por los miembros de una PYME no tiene los mismos que si es un servicio destinado a todos los jóvenes españoles de entre 14 y 25 años, y por tanto estos aspectos no funcionales tampoco son lo mismo.

Esta sección contendrá todas las páginas que sea necesaria para dejar bien especificado los requerimientos del proyecto.

# Diseño

El capítulo de diseño es con diferencia el más importante del TFG. Podría decirse que es el corazón de dicho trabajo. En este punto se ha de diseñar la solución de forma que dé respuesta a todos y cada una de los requerimientos funcionales y no funcionales anteriormente establecidos. Por tanto en esta sección se hará constante referencia a los identificadores de los requerimientos que se están abordando. Hay que tener en cuenta que un requerimiento puede necesitar de diferentes elementos en un software para ser resuelto. Por ejemplo, si uno de los requerimientos es el control de acceso posiblemente requerirá de dotar de identificación mediante usuario/contraseña al sistema, y por tanto necesitaremos un sistema de gestión de usuarios y permisos de usuario compuesto por una interfaz gráfica, posiblemente algún tipo de API, web servirse u otro acceso a datos, algún sistema de persistencia (base de datos, motor de base de datos, etc.), control de errores, cifrado de las comunicaciones a nivel de infraestructura y cifrado de datos a nivel de BD, etc.

El diseño de soluciones es muy complejo y en él intervienen aspectos a distintos niveles o vistas de un proyecto. Una buena práctica es agrupar el diseño en apartados en función de la matera o el aspecto que se está definiendo, concentrando así todo lo relativo a ese contexto en un único apartado. Una aplicación será la suma en realidad de todos esos diseños.

## Diseño de la persistencia

La persistencia es el aspecto relativo a todo lo concerniente al almacenamiento de datos del nuestra propuesta. Hay que entender que esto debería incluir los siguientes aspectos:

* Almacenamiento de datos: analizar y detectar todos los campos y datos a almacenar, sus tipos, sus tamaños, sus relaciones. Diseñar las colecciones o tablas en las que deban ser almacenados. Definir el modelo de base de datos para ellos: SQL, noSQL, documental, estática, dinámica, orientadas a objetos, etc. Leer el artículo sobre “Bases de Datos” [6] para tener en mente que hay muchas características que definen las bases de datos y que es necesario definir una correctamente en función de los datos que manejamos. Además indicar la necesidad de cifrado o no de los datos y el tipo, según la legislación o según los requerimientos.
* Almacenamiento de datos de administración o metadatos: definición de los datos que serán almacenados referentes a la administración del sistema como los logs, la trazabilidad de acciones, los errores, la auditoría. Definir igual que antes estructuras, modelos de bases de datos, motores, definición de archivos y sus estructuras, etc.
* Seguridad e integridad: definir los aspectos relativos a las políticas de seguridad para acceso al servicio de base de datos, la infraestructura donde será instalada, cómo se establecerán los mecanismos de copias de seguridad o incluso de detección de errores en la integridad, si se proveerá de herramientas que generen alertas o avisos sobre estas situaciones, etc.

Todos estos aspectos pueden ser desarrollados aunque cómo mínimo debería obtenerse un modelo de diseño de la base de datos donde se refleje la información que se maneja, algo como lo mostrado en la Figura 5.

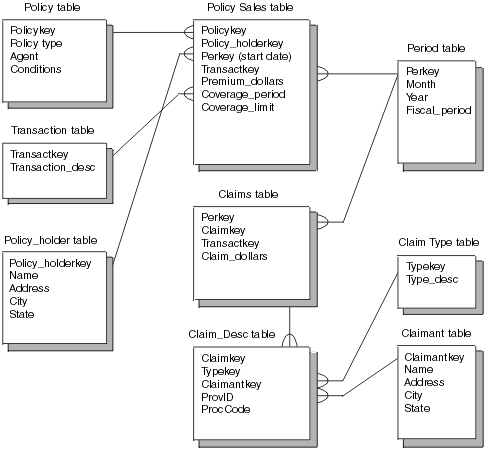


Figura 5. Ejemplo de diseño de BD

(Fuente https://www.ibm.com)

## Diseño arquitectura conceptual

El diseño de la arquitectura conceptual permite definir a grandes rasgos los módulos y bloques funcionales que contendrá la solución. Es una descripción de cómo se plantea desarrollar la solución y condicionará las tecnologías que posteriormente se utilizaran. Este diseño puede hacerse en varias fases donde cada uno de ellos profundiza en un bloque. Podemos observar un ejemplo en los siguientes diagramas extraídos del TFG de Alejandro Torres [Figura 6]

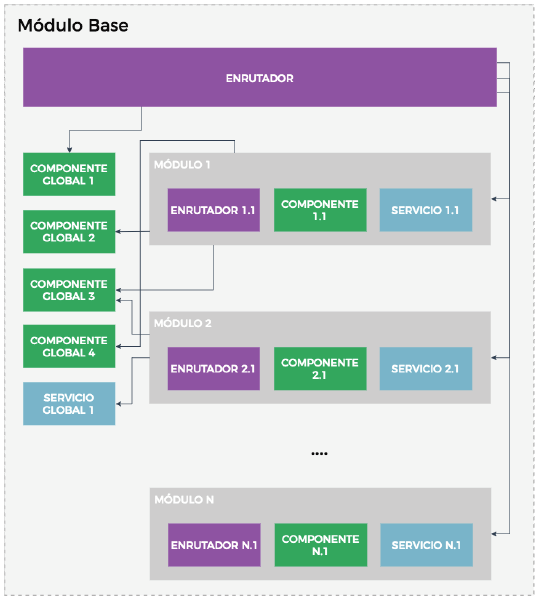
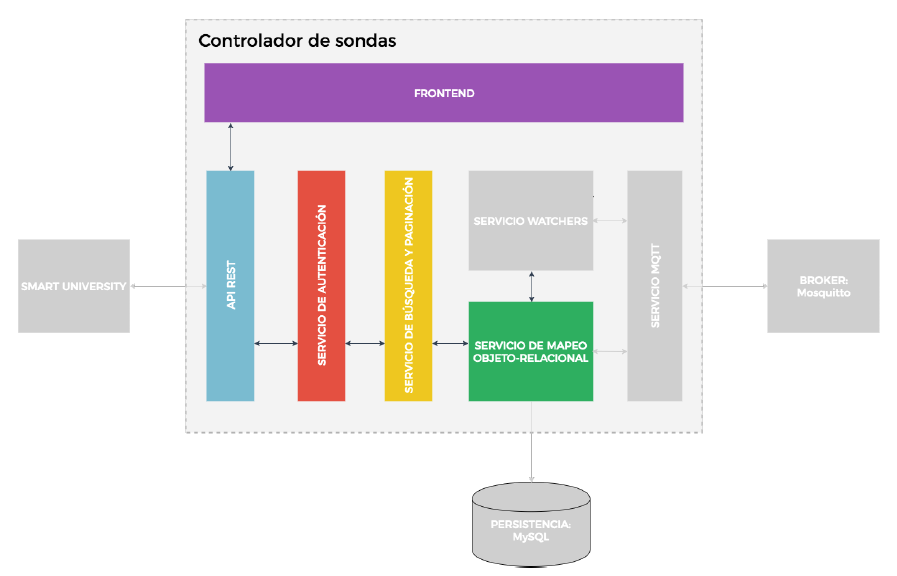


Figura 6. Diseño arquitectura conceptual panel IoT.

(Fuente TFG Alejandro Torres Mateu)

Estos diagramas ayudaran en el desarrollo a estructurar archivos, directorios, agrupar funcionalidades y distribuir el código.

## Diseño API Rest (si hay)

Dependiendo del tipo de solución a abordar y siguiendo esquemas conceptuales actuales muy extendidos, es posible que el producto se base en una API Rest. Es necesario diseñar estar API al margen de qué tecnologías serán utilizadas, identificando las entidades que la formarán, las operaciones a realizar, por lo menos las básicas o llamadas CRUD y las rutas. Muchas veces no comprender que es una API Rest y cómo utiliza los métodos HTTP para hacer su trabajo genera una mala definición de esta API, que aunque funcional, conceptualmente es errónea. Es recomendable algún material de referencia como puede ser el libro de Enrique Amodeo [8], con él será muy fácil diseñar buenas API Rest que luego será muy fácil implementar. Esta guía además explica aspectos que pueden comprometer nuestro funcionamiento como la concurrencia o la utilización de recursos multimedia.

Cómo mínimo deberías obtener tras este apartado un listado con el routing de nuestra API Rest, la definición de los parámetros y los datos/formatos que devuelve nuestra API. Algo parecido a lo que por ejemplo nos ofrece la OPEN API Doc de Smart University [9].

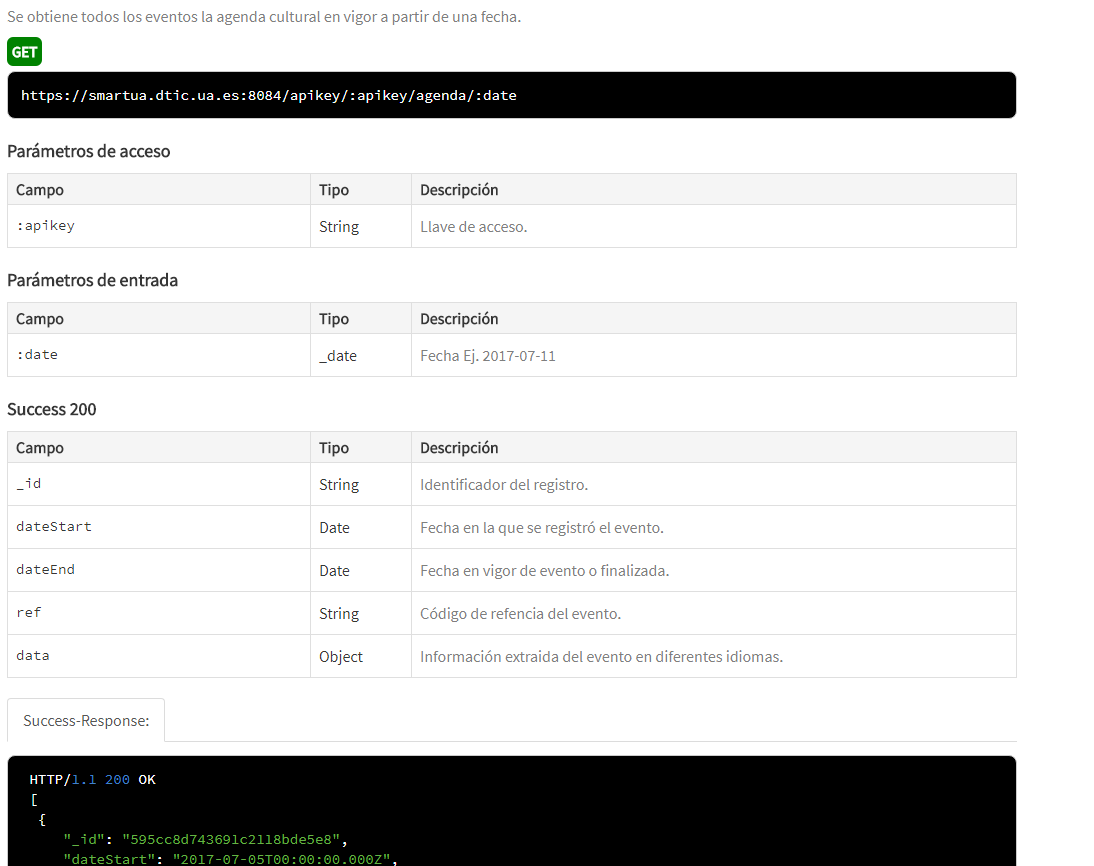


Figura 7. Especificación de la Open API de Smart University

(Fuente Smart University www.smart.ua.es)

## Diseño arquitectura tecnológica Front/Back-end

Este apartado ya incluye una propuesta de las tecnologías que serán utilizadas para el desarrollo del proyecto. Estas tecnologías también deberán escogerse en función de los requerimientos tanto funcionales como no funcionales. Si la solución deberá estar destinada por ejemplo a conformarse como un servicio web, las tecnologías serán web, si debe ser una app móvil, las tecnologías estarán relacionadas con esto. La definición de tecnologías puede llegar incluso a sistemas como: servicios de identificación de terceros, herramientas de copias de seguridad, Sistemas Operativos, sistemas de lanzamiento de tareas en segundo plano, etc. Aquí se especificará e incluso se discutirán alternativas para presentar las decididas finalmente. La Figura 8 muestra por ejemplo el stack tecnológico seleccionado para implementar la arquitectura del proyecto Smart University en función de su arquitectura conceptual de 3 niveles. El alumno decidirá el suyo en función de su propio diseño conceptual.

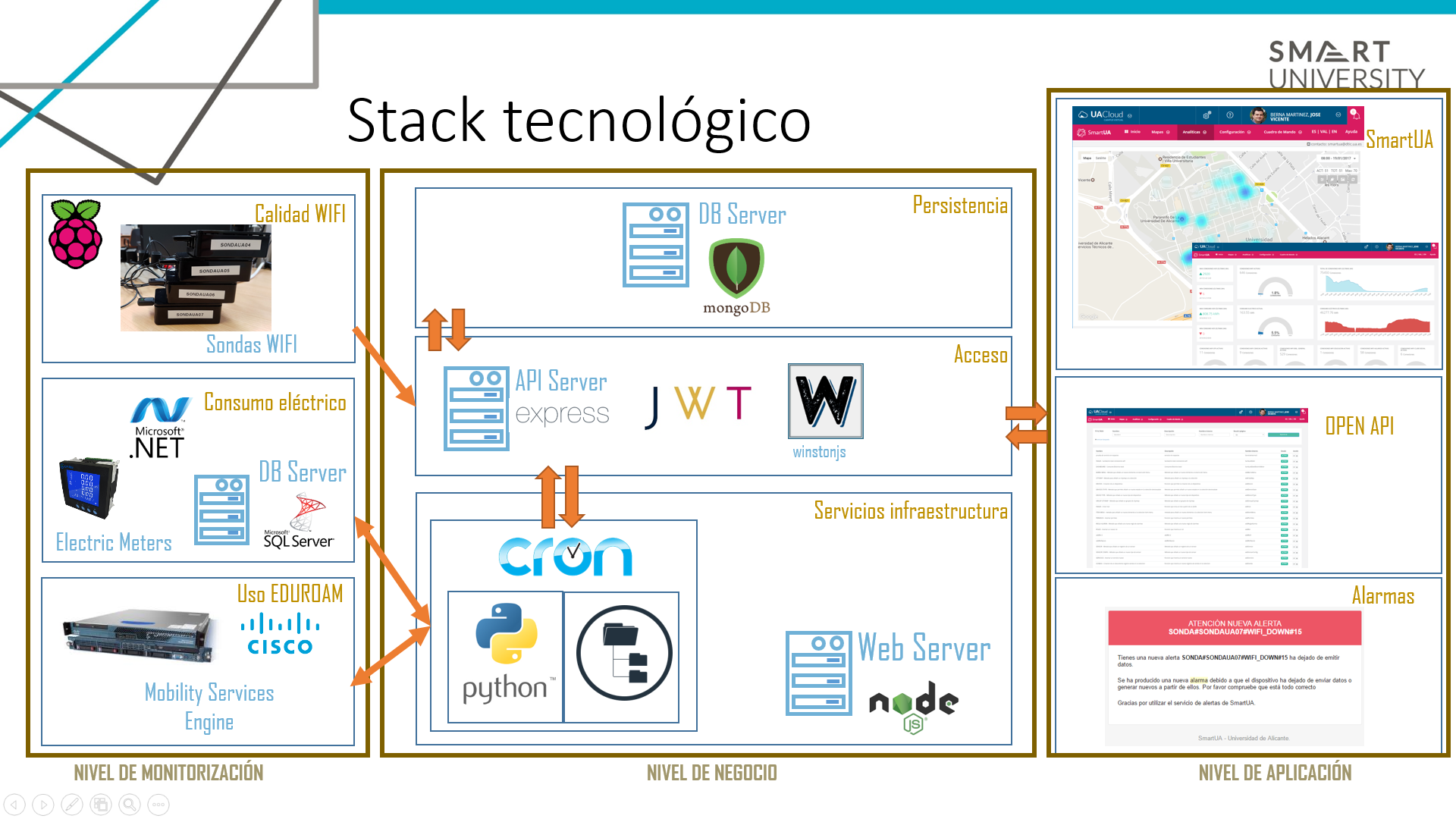


Figura 8. Stack tecnológico del proyecto Smart University

(Fuente Smart University www.smart.ua.es)

## Diseño Interacción o Experiencia de Usuario

La experiencia de usuario es un componente muy importante de las actuales aplicaciones y suele ser además un factor de fracaso o éxito. La experiencia de usuario es un concepto todavía controvertido y polémico que muchas veces se asume desde los tradicionales de accesibilidad y usabilidad pero que en realidad actualmente es mucho más que estos conceptos. Sirva el siguiente ejemplo de la Figura 9 como definición clarificadora de la diferencia entre los conceptos de accesibilidad, usabilidad y experiencia de usuario. La función de una máquina de resonancia magnética es la generación de una imagen que proporciona información sobre la composición y estructura del cuerpo humano sin invadirlo literalmente. Si bien la funcionalidad está clara, para el paciente su experiencia de usuario se resumen en que ha de colocar en el interior de la máquina y verse sometido durante un periodo de tiempo variable a un ruido estridente que lo puede poner nervioso, estresar e incluso inducir al página. La evolución de esta máquina en el tiempo ha mejorado en cuestión de la calidad del resultado que se obtiene de la prueba en su funcionamiento prácticamente es el mismo. Como podemos ver en la Figura 9-1, las versiones básicas de la máquina pensaban poco en el paciente y solo se centraban en la funcionalidad. El paciente debía subirse a una camilla alta, utilizando escalones o adaptadores, recostarse en la camilla y sufrir el proceso. Además el profesional que realizaba la prueba no podía estar con él para no interferir durante el proceso electromagético lo cual le dejaba solo en la habitación y con el peligro de que si durante la prueba se movía, tenían que repetirla por completo. Uno de los pasos fue mejorar la usabilidad de la máquina como vemos en la Figura 9-2, diseñando habitaciones donde el profesional pudiese controlar en todo momento al paciente y a la vez el paciente de alguna forma se pueda ver acompañado y guiado en el proceso. Pero aun así la experiencia de usuario mejora poco, aunque sí la usabilidad de la máquina por parte de los profesionales. Sin embargo en la Figura 9-3 ya vemos que se introduce un elemento destino exclusivamente al paciente, la utilización de camillas elevadoras de forma que el paciente ahora puede hacerse la prueba más cómodamente, esto mejora el acceso (accesibilidad) a la realización de la prueba (pero no su funcionalidad), mejora algo la experiencia de usuario pero en realidad es funcionalidad mejorada para el usuario. Por último en la Figura 9-4 vemos una modificación que no tiene nada que ver absolutamente con el aspecto funcional, ni del usuario ni del paciente, pero que está completamente destinado a la experiencia de usuario. La habitación y la propia máquina han sido decoradas con distintos motivos de forma que un niño (pacientes habituales también) no tendrá miedo e incluso puede disfrutar de la experiencia (quien sabe si tal vez le queden ganas incluso de repetir).



Figura 9. Ejemplo de la diferencia entre usable, accesible y experiencia de usuario.

(Fuente propia)

La consideración de aspectos destinados a la UX normalmente no interfieren en la usabilidad ni accesibilidad (otras veces sí), pero sí que interfieren en la huella que generan en el usuario y por tanto en su predisposición a volver a nuestro servicio.

Hay muchas técnicas para estudiarla. Uno de los primeros aspectos a comprender es precisamente que es lo que se espera que un usuario haga en nuestro servicio, entender cómo esperamos que interaccione con él y para ello nos puede ayudar los User Journey Map, aquí este artículo [10] os puede ayudar a entender para qué sirven.

También puede ser muy útil saber más sobre UX, sus aspectos claves y ciertas cosas que se pueden tener en cuenta en general y que te ayudaran a que tus usuarios quieran utilizar más tus aplicaciones. En el libro de Yusef Hassan [11] podrás encontrar muchas claves. La idea es tenerlas claras para cuando diseñes tus interfaces.

## Diseño Interfaces

El diseño de interfaces es la parte donde los RF y RNF se materializan, es decir, es la parte de la ingeniería donde se desarrollan los mecanismos físicos mediante los cuales los distintos usuarios podrán interaccionar con nuestro sistema. Pueden ser interfaces web, aplicaciones de escritorio, aplicaciones móviles o incluso líneas de comandos. Este diseño debería implementar los requerimientos funcionales y no funcionales que no han sido satisfechos por otras parte del diseño.

Es importante tener en cuenta las distintas prácticas de usabilidad, accesibilidad y UX tanto estudiadas como vistas en apartados anteriores. La definición de las interfaces permite la toma de decisiones en etapas de diseño y no en desarrollo, lo cual agilizará esta última fase. El diseño de interfaces puede hacerse con distintos grados de fidelidad lo cual facilitará más o menos el desarrollo. Hay que tener en cuento que cuanto mayor exactitud tenga el diseño de interfaces ahora, menos decisiones y desarrollos hay que hacer después. Podemos hablar de tres niveles de fidelidad.

* Wireframe

Es la primera representación de nuestra propuesta de solución, normalmente utiliza cajas y símbolos entendibles y está orientado a reflejar de una forma poco refinada las funcionalidades del sistema y dónde encajan más o menos los elementos que lo forman. La Figura 10 es un ejemplo. Los wireframes son indispensables en las primeras fases del diseño. Aquí se incluyen ya los aspectos de usabilidad y accesibilidad que están más relacionados con las funcionalidades. Tiene como ventaja que es muy rápido su desarrollo ya que no se tienen en cuenta por menores de diseño y está muy orientado a expresar de qué forma se implementen los RF y RNF. El inconveniente es que dista mucho de lo que será el producto final, al cliente le puede inducir a error y hay muchas interacciones que pueden no percibirse y quedar olvidadas. Una herramienta es Balsamiq [12].

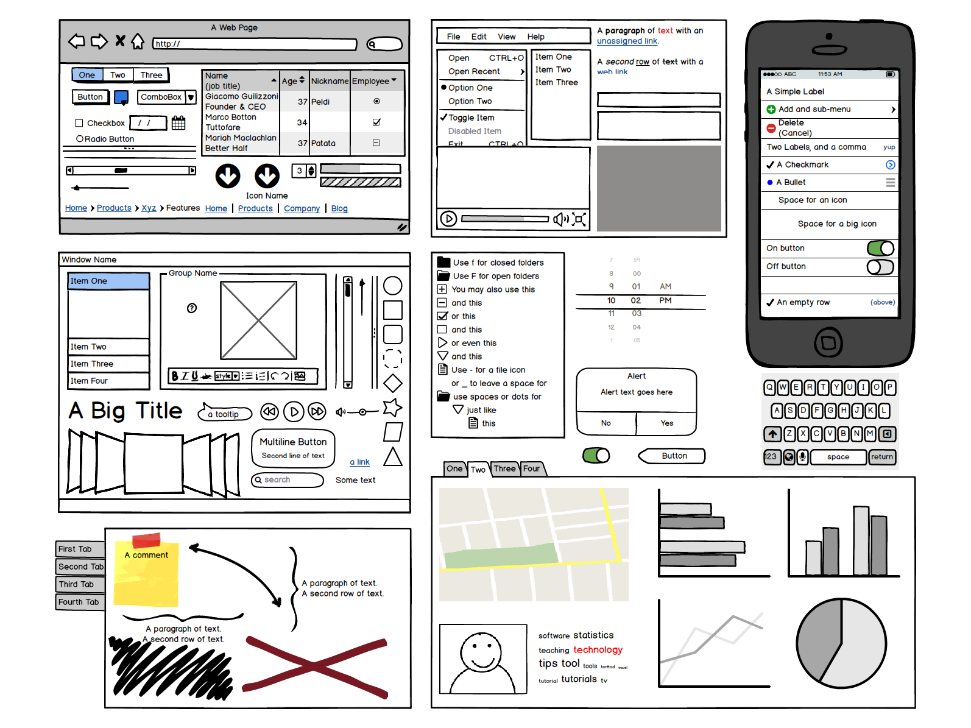


Figura 10. Ejemplo de wireframe con Balsamiq.

(Fuente Balsamiq www.balsamiq.com)

* Mockup

Un mockup es una representación visual y estática de un diseño como muestra la Figura 11. Es una vista exacta de cómo será la interfaz que percibirá el usuario y por tanto indica no solo la posición de os elementos sino su estado final una vez implementados. Además en este diseño ya se contemplan temas relacionados con la UX ya que colores, tipos y fuentes, tamaños, fondos, aspecto general, percepción, etc., ya están definidos. Tiene como principal ventaja que ya es muy cercano al desarrollo y por tanto muchos elementos pueden ser utilizados para agilizarlo, además de que muchas decisiones como la disposición de los elementos en un formulario, el orden de aparición de ciertos pasos, etc., ya están tomadas. El inconveniente es el tiempo necesario para su desarrollo ya que incluye aspectos puramente de diseño que ralentizan por tanto su creación. Se pueden utilizar herramientas cómo Canva [13] o Justinmind [14].

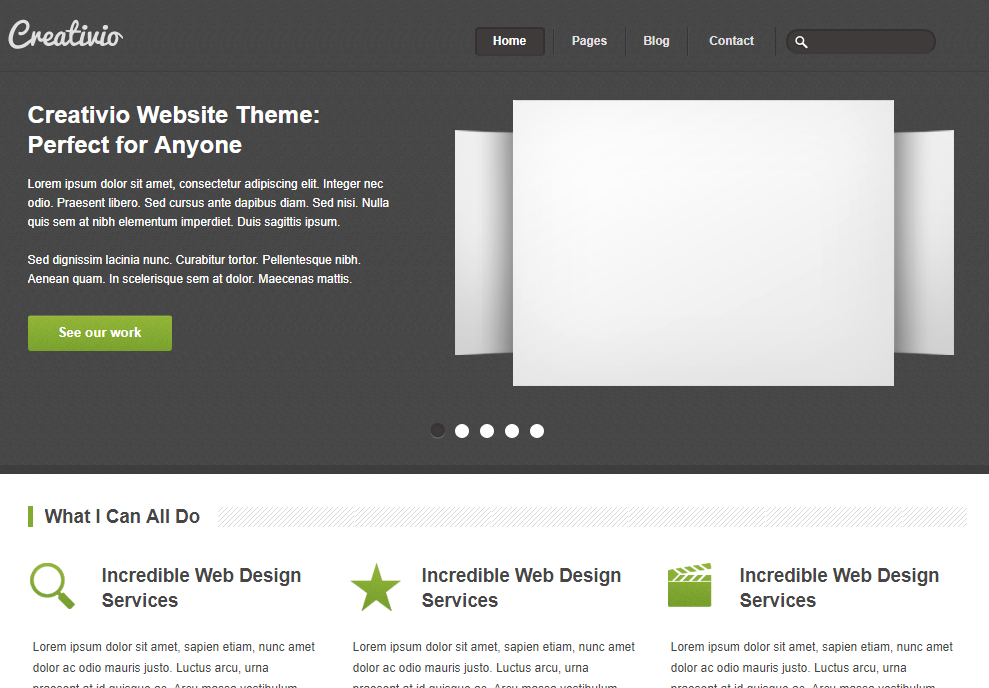


Figura 11. Mockup desarrollado con Justinmind

(Fuente Justinmind www.justinmind.com)

* Prototipo

Un prototipo es también una representación de la versión final de un diseño que, además, simula la interactividad. Un prototipo permite entender mejor la interacción del usuario con la herramienta y es un buen principio para mostrar a un cliente ya que le permitirá a él interactuar con dicho producto sin que se haya sido necesario implementar código. La ventaja es evidente, tener una simulación del producto final que permitirá medir no solo el UX sino también validar la interacción como muestra la Figura 12. La gran desventaja es que requiere de invertir mucho más tiempo en diseño y puede ser incluso prohibitivo en plazos. Es aconsejable solo en los casos en los que se requiere de una aprobación del cliente. Herramientas como Justinmind [14] o Sketch [15].

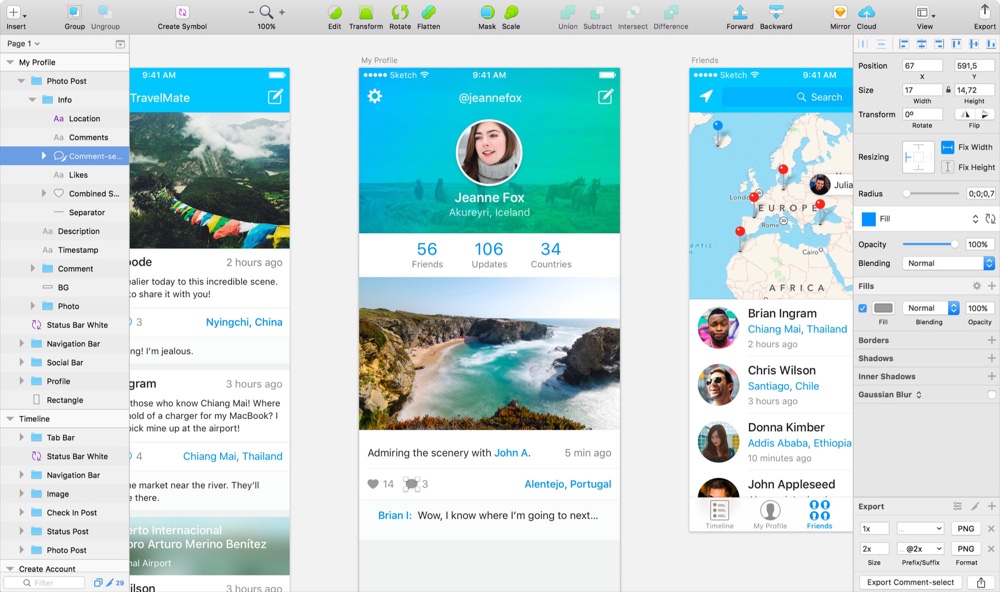


Figura 12. Diseño desarrollado con Sketch.

(Fuente Sketch www.sketchapp.com)

Sea cual sea el grado de diseño se puede acompañar además de los mapas de interacción, es decir, de un diagrama en el que se muestra el flujo del usuario entre las distintas interfaces. Este mapa ayuda a visualizar que al usuario no se le somete a excesivos pasos intermedios para alcanzar funcionalidades y que todos los puntos de retorno son correctos. La Figura 13 muestra un mapa en un diseño de wireframe Balsamiq.

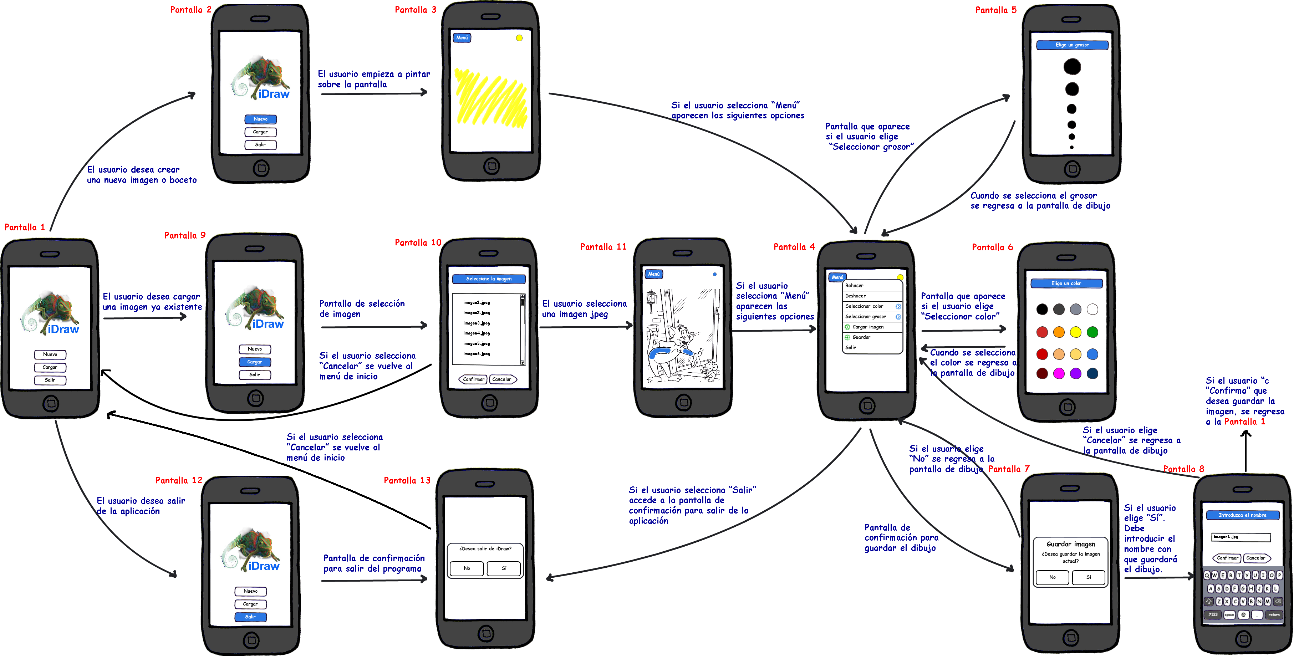


Figura 13. Ejemplo de mapa de interacciones.

(Fuente blog wordpress https://josemigueldih.wordpress.com)

Las herramientas como Sketch o Justinmind ya producen estos mapas de forma automática cuando se diseñan, pero la contrapartida es que hay que añadir la interacción en el diseño.

## Guías de estilos

Si tu diseño es de alta fidelidad la guía de estilos estará implícita en el propio diseño. Aun así es bueno que la aísles y la describas. La guía de estilos define fuentes, tipos, tamaños, colores y ubicaciones en general. No es que exista una única forma de realizar una guía de estilos pero puede ser muy útil buscar una de una gran corporación y pensar en detallar los mismos elementos. Por ejemplo, te puede servir de la de la Universidad de Cádiz [16] donde se definen desde la ubicación de los elementos, las paletas de colores (Figura 14) y tipografías. Si tienes todos los elementos definidos e incluso plantillas, el desarrollo será muchísimo más rápido y tu aplicación final mucho más coherente. Si esto ya lo has hecho en tu diseño de interfaces, asegúrate de que es coherente y constante.

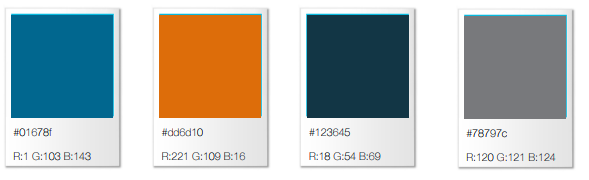


Figura 14. Paleta de colores Unviersidad de Cádiz

(Fuente Guía de Estilos Universidad de Cádiz)

## Diseño de pruebas y validación

También es interesante que definas otros elementos relacionados con la comprobación de tu sistema a diversos niveles. Tipos de pruebas y validaciones hay decenas y podría escribirse un TFG solo con el diseño de estas, pero dado que en el caso del desarrollo de un TFG el objetivo principal no es el diseño de pruebas, debemos definir al menos algo que nos permita concluir que hemos alcanzado los objetivos.

Por un lado es muy recomendable, ya que vas a implementar posiblemente un API o servicios de acceso a datos que definas y automatices alguna prueba que te permita comprobar que tras un cambio estos servicios siguen funcionando. Esto puede ser tan complejo o simple como quieras pero por ejemplo, en el caso de una API puedes utilizar herramientas como Postman [17]. Cada vez que desarrolles una llamada nueva de tu API puedes definirte una llamada que genere una respuesta correcta y una que genere una respuesta incorrecta, con Postman puedes crear baterías de pruebas que vas incrementado junto a tu desarrollo. Una vez que realizas una modificación puedes fácilmente ejecutar una secuencia de llamadas ya analizar que todas han devuelto el resultado esperado (.

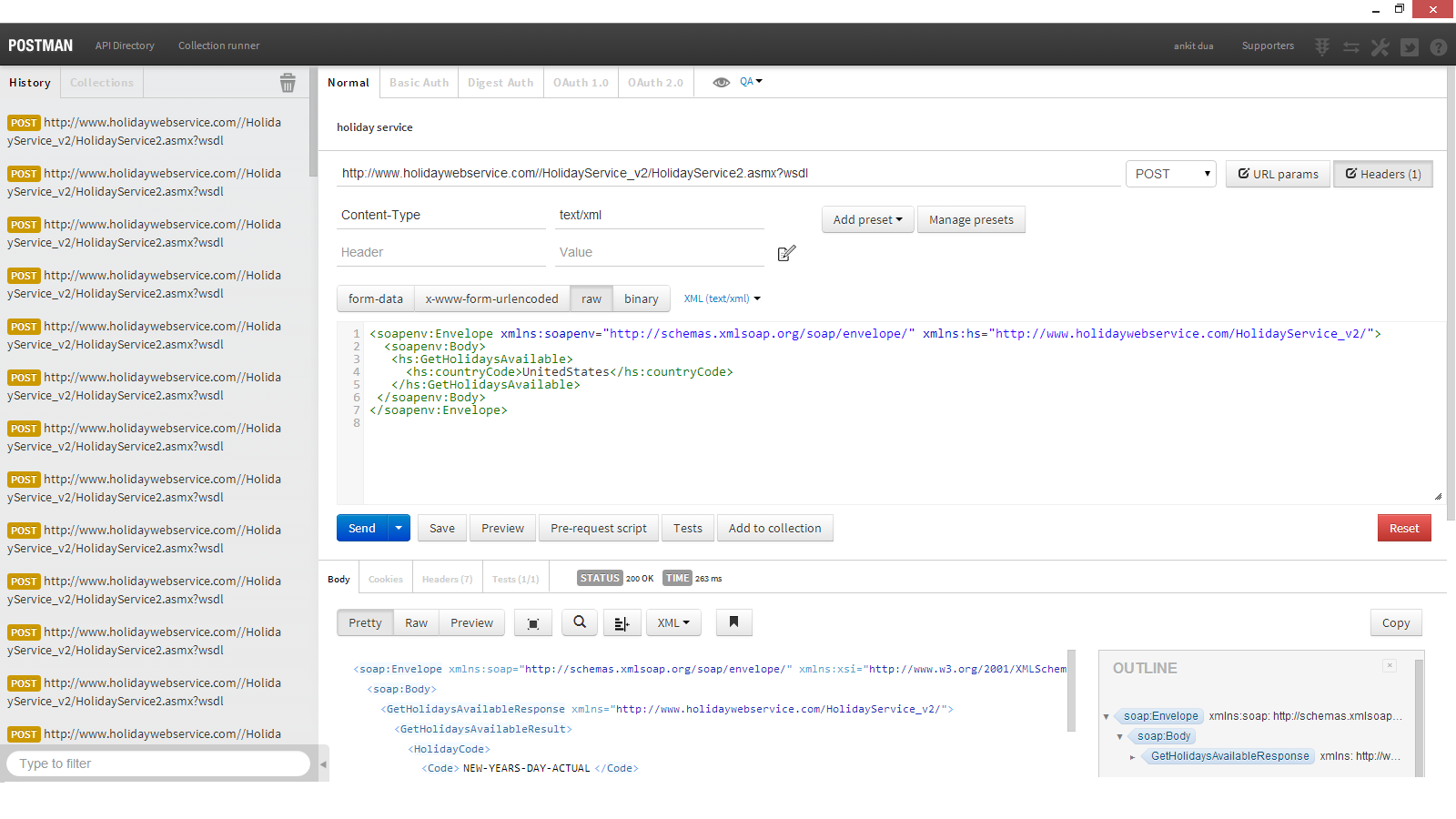


Figura 15. Ejemplo de ejecución de varias invocaciones a servicio.

(Fuente Postman http://blog.getpostman.com)

También es adecuado definir indicadores y los valores esperados para estos indicadores. Por ejemplo si tienes un RNF sobre la capacidad de tu servicio y lo has diseñado pensando en que el número de conexiones va a estar en torno a 100 instancias por segundo, puedes definir un indicador *NumConextiones*, y un rango: 0-50 bajo, 50-80 medio, 80-100 bueno, 100< muy bueno. Cuando hagas pruebas de estrés de conexiones en función del valor obtenido sabrás si has alcanzado con éxito tu RNF.

Otra buena práctica es crearse una check list con los RF y RNF y asegurarse de que han sido implementados según su definición ya que muchas veces y debido a que las funcionalidades están dispersas a lo largo de diseños de distinto tipo, pueden perderse o quedar definidos a medias.

# Implementación

En este apartado se documenta el proceso de desarrollo, no se escribe el código. Es muy importante tener en cuenta que debe adecuarse a la metodología propuesta y exhibir evidencias de dicha metodología. Si se indicó por ejemplo una metodología iterativa, deberían indicarla las iteraciones realizadas y las funcionalidades que se incluyeron en cada una. Si la metodología es Scrum podrían mostrarse algunos sprints.

También se puede documentar aspectos como la preparación de los entornos en caso de que se hayan preparado entornos de desarrollo, integración, preproducción y producción, herramientas para el desarrollo y soporte (como por ejemplo gestión de versiones con Git).

Se indica la estructura del proyecto, la distribución de archivos, archivos de configuración a nivel de sistemas (por ejemplo si es necesario modificar la configuración de nuestro servidor web Apache) y de aplicación (si nuestro aplicación por ejemplo tiene un idioma por defecto y se configura en alguna variable).

Se analizan los problemas ocurridos durante el desarrollo, las decisiones tomadas, los ajustes necesarios y su justificación (por ejemplo descarte de funcionalidades por razones temporales).

Si se han desarrollado algún algoritmo o solución muy a destacar se puede mostrar el código comentado pero se debe ser muy crítico con esto y es preferible mostrar pseudocódigo o algoritmos a código puro. Se puede utilizar la sección de Anexos añadida al final para mostrar código, documentos de seguimiento, documentos manuales que han sido utilizados u otros elementos de interés.

Exhibir el producto final, sus interfaces destacables para comprobar que el resultado es similar al diseñado.

# Pruebas y validación

Una vez la herramienta ha sido finalizada deben incluirse las pruebas y validaciones diseñadas en el apartado anterior. La idea es demostrar que no solo se ha generado la herramienta esperada sino que es fiable y robusta. Validar finalmente que la resuelve el problema analizado y propuesto de una forma aceptable. Si es posible incluso recurriendo a pruebas de campo con usuarios.

Si se han definido indicadores es un buen momento para demostrar que las métricas son adecuadas.

Si hay problemas en las pruebas se deben documentar y analizar a que son debidas proponiendo las posibles soluciones, estas pueden ser propuestas como trabajo futuro.

# Resultados

Analizar el producto final y mostrar los resultados. Si está en producción o ha sido probada por usuarios tester o reales indicar las valoraciones, el número de usuarios, si tiene contenidos, si está generando monetización o los indicadores de tráfico.

Realizar demostración. Preparar un pequeño guion que puede dejarse aquí documentado también para hacer la demostración. La demostración se realiza en el tiempo de Demo que tiene el alumno exclusivamente para ello, pero en esta sección se pueden introducir los contenidos que serán mostrados.

Realizar también un análisis interno sobre los costes temporales planificados y los reales, analizar las diferencias entre ellos y justificarlas. Analizar también otros ajustes como los funcionales, si se deben a un exceso de funcionalidades, a otros problemas como posibles errores que producen retrasos o problemas con el hardware.

También se puede hacer un recorrido por algunas asignaturas de la carrera enlazando los contenidos vistos con los desarrollados en el trabajo. De esta forma se pone en valor la puesta en práctica de las competencias desarrolladas durante la carrera. Es una buena reflexión que puede ayudar a hacer un análisis final sobre aquellas asignaturas más significativas para nuestro proyecto.

# Conclusiones y trabajo futuro

Apartado de carácter obligatorio que todo trabajo debe contener. En este apartado se recogerán las conclusiones extraídas de nuestro trabajo. Cosas como recalcar el alcance de los objetivos, si se han logrado por completo o no, sobre los resultados obtenidos que repercusiones pueden tener en el entorno o comunidad destino, sobre la dificultad global del proyecto y tus propias impresiones personales.

Además todo trabajo siempre está inacabado y por ello debemos finalizar las conclusiones con unas líneas de trabajo futuro indicando por ejemplo mejoras que deseamos realizar en posteriores versiones, una línea de trabajo para su puesta en producción, funcionalidades que no han podido ser implementadas e incluso si hay un plan de comercialización a la vista.

Sirva como ejemplo las propias conclusiones de la generación de este documento.

Tras varios años de experiencia donde se aprecia que los estudiantes tienen un gran problema en la estructuración de su TFG, tanto a nivel de contenidos como en planificación, y también en proporcionar un formato adecuado se ha decidido solucionar el problema generando una guía que sirva tanto de plantilla y modelo de documento como de ejemplos y descripción de los posibles contenidos. Esta guía no es exhaustiva ni universal, pues está orientada a proyectos de tipo desarrollo software, de ahí sus secciones tan orientadas a este proceso. Pero sin embargo puede servir como base a cualquier proyecto y TFG/TFM.

Se desprende de esta guía que el trabajo del ingeniero es precisamente el arte y técnica de aplicar los conocimientos científicos a la invención, diseño, perfeccionamiento y manejo de nuevos procedimientos en la industria y otros campos de aplicación científicos. En resumen, la creación e innovación. Por ello ha de poner su empeño no tanto en el desarrollo de la solución final sino en el proceso que sigue a lo largo de todo el proyecto y que le lleva hacia el producto final.

Este es el resultado de varios años de trabajo en la dirección de TFG/TFM y después de haber elaborado diversos recursos, talleres y seminarios, y creo que será un buen principio donde los alumnos podrán superar fácilmente la primera dificultad, el del documento vacío y ponerse a trabajar siguiendo indicaciones claras y una hoja de ruta en forma de plantilla.

Como trabajo futuro inmediato queda por supuesto evaluar formal o informalmente el resultado de esta guía, recogiendo impresiones de mis propios alumnos de TFG/TFM junto con sus tasas de éxito, la calidad de los trabajos tras la evaluación del tribunal e incluso el impacto en otros TFG/TFM.

Como trabajo a largo plazo queda la mejora continua de dicha guía a partir de los comentarios de alumnos, profesores, compañeros y tribunales junto con los propios resultados de la evaluación.

# Referencias

La sección de referencias debe incluir las obras y materiales consultados y empleados en la elaboración de la memoria. La bibliografía puede indexarse en una lista numerada siguiendo el orden de aparición con lo que serán referenciadas utilizando en el número entre corchetes, o en orden alfabético con lo que serán referenciadas utilizando nombre y fecha. Para más detalles se puede consultar la Guía de Estilos de la EPS [3]. La lista siguiente compone la lista de referencias utilizadas en las explicaciones de la guía. El número de referencias totales de un TFG no está fijado pero es un indicativo de la investigación previa realizada.

1. Sixto Garcia, J. *Fundamentos de marketing digital*. Ed. Comunicacion Social Ediciones Y Publicaciones. 2016.
2. Moreno Molina, M. *Cómo triunfar en las redes sociales: Consejos prácticos y técnicas para conseguir todo lo que te propongas en Internet y sacarle más partido a tus redes ... Facebook, Twitter, Instagram, LinkedIn*. Ed. Gestión 2000, mayo 2015.
3. Libro de Estilos para la redacción de trabajos TFG/TFM de la EPS. Disponible en: <https://maktub.eps.ua.es/servicios/gestorContenidos/contenidos/normativaEPS/Pdf/9910.pdf>
4. Doran, G. T. *There's a S.M.A.R.T. Way to Write Management's Goals and Objectives*. Management Review, Vol. 70 (11), pp. 35-36. 1981
5. *Especificación de Requisitos según el estándar de IEEE 830*. Disponible en: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
6. *Bases de Datos*. Wikipedia. Disponible en: <https://es.wikipedia.org/wiki/Base_de_datos>
7. Torres Mateu, A. *Panel frontal para el controlador IoT de Smart University*. TFG Ingeniería Multimedia 2016-2017. Disponible en: <http://hdl.handle.net/10045/69787>
8. Amodeo, E. *Principios de diseño de APIs REST*. 2013.
9. Documentación de la API de Smart University de la Universidad de Alicante. Disponible en: <https://smartua.dtic.ua.es:80/public/apidoc/>
10. *El Mapa de Experiencia del Cliente o Customer Journey Map*. Disponible en: <http://innokabi.com/claves-para-emocionar-a-tu-cliente-customer-journey-map/>
11. Hassan Montero, Y. Experiencia de Usuario: Principios y Métodos. 2015. Disponible en: <http://yusef.es/Experiencia_de_Usuario.pdf>
12. Balsamiq. Herramienta para crear wireframes. Disponible en: <https://balsamiq.com/>
13. Canva. Herramienta para crear mockups. Disponible en <https://www.canva.com>
14. Justinmind. Herramienta para crear mockups y diseños interactivos. Disponible en <https://www.justinmind.com/>
15. Sketch. Herramiento para crear mockups y diseños interactivos para Mac. <https://www.sketchapp.com/>
16. Guía de Estilo Web de la Universidad de Cádiz. Disponible en: <http://www.uca.es/recursos/doc/Unidades/Gab_Com_Mark/465200059_19420109123.pdf>
17. Postman. Herramienta para análisis de API REST. En forma de plugin para Chrome y aplicación. Disponible en: <https://www.getpostman.com/>

# Apéndice I

Se pueden añadir tantos apéndices como sean necesarios. Los apéndices recogen todo aquello que no es recomendable que esté en mitad del texto como por ejemplo código sobre el que se habla específicamente pero que es demasiado largo como para insertarlo en mitad de la memoria, documentos escaneados de originales y que han servido para extraer el análisis de datos, bocetos realizados a mano y que luego fueron digitalizados en las herramientas de diseño. Todo esto si se quiere dejar constancia puede ser insertado en estos apéndices.

Hay que tener en cuenta que todo apéndice tiene que estar referenciado siempre en la memoria, es decir, que en algún momento en el texto tiene que hacerse mención al apéndice y su contenido.

Además el apéndice debe contener no solo imágenes sino también el texto suficiente para explicar su contenido pero sin repetir texto de la memoria. Por ejemplo supongamos que se ha hecho un primer wireframe a papel que después ha sido digitalizado. En la sección de diseño correspondiente indicaríamos que:

*“… en el Apéndice I hemos dejado el primero boceto manual que aquí está digitalizado y que incluye los elementos a, b, c…”*

Mientras que en el Apendice I indicaríamos lo siguiente, junto a la imagen bocetada:

*“A continuación se muestran los primeros bocetos hechos a mano alzada y que dieron paso a los posteriores diseño de la aplicación. Se incluyen aquí,* Figura 16*, para que pueda apreciarse como las primeras ideas son perfeccionadas posteriormente en los procesos digitales”.*



Figura 16. Primer boceto de la aplicación.

(Fuente SmartBlog http://www.smartblog.es)