

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

**TÍTULO DEL PROYECTO**

AUTOR/A: NOMBRE DEL ALUMNO/A

Puerto Real, mes y año

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

# TÍTULO DEL PROYECTO

DIRECTOR/A: NOMBRE DEL DIRECTOR/A  
AUTOR/A: NOMBRE DEL ALUMNO/A

Puerto Real, febrero de 2023



# Declaración personal de autoría

---

[Nombre y apellidos del estudiante] con DNI [Nº de DNI –incluyendo letra- del estudiante], estudiante del [Título de Grado o Máster para el que se presenta el TFG/M] en la Escuela Superior de Ingeniería de la Universidad de Cádiz, como autor de este documento académico titulado [Título del TFG/M] y presentado como Trabajo Final de [Grado o Máster, según corresponda]

## DECLARO QUE

Es un trabajo original, que no copio ni utilizo parte de obra alguna sin mencionar de forma clara y precisa su origen tanto en el cuerpo del texto como en su bibliografía y que no empleo datos de terceros sin la debida autorización, de acuerdo con la legislación vigente. Asimismo, declaro que soy plenamente consciente de que no respetar esta obligación podrá implicar la aplicación de sanciones académicas, sin perjuicio de otras actuaciones que pudieran iniciarse. En Puerto Real, a [FECHA]

Fdo: [Nombre y apellidos del estudiante]



# Agradecimientos

---

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



# Resumen

---

Introduzca aquí un resumen no superior a 500 palabras, que servirá de descripción pública del trabajo realizado. Debe tener en cuenta que este documento es una plantilla genérica y que deberá adaptarse a la tipología específica del proyecto. Por ello, no todas las secciones o capítulos serán de obligatorio cumplimiento. Por otra parte, debe recordar que todas las figuras, las tablas y los listados de código que aparezcan en el trabajo deberán estar convenientemente referenciados en el texto.

**Palabras clave:** Palabra clave 1, palabra clave 2, palabra clave 3...





# Índice general

---

Índice de figuras	x
Índice de tablas	xi
<b>I Prolegómeno</b>	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Motivación . . . . .	3
1.2. Alcance y Objetivos . . . . .	3
1.3. Glosario de términos . . . . .	3
1.4. Organización del documento . . . . .	3
<b>2. Antecedentes</b>	<b>5</b>
2.1. Contexto . . . . .	5
2.2. Estado de la técnica . . . . .	5
<b>3. Plan de gestión de proyecto</b>	<b>7</b>
3.1. Metodología de desarrollo . . . . .	7
3.2. Tecnologías . . . . .	7
3.3. Planificación del proyecto . . . . .	7
3.4. Organización . . . . .	7
3.5. Costes . . . . .	8
3.6. Riesgos . . . . .	8
3.7. Gestión de la configuración . . . . .	8
3.8. Aseguramiento de calidad . . . . .	8
<b>II Desarrollo</b>	<b>9</b>
<b>4. Requisitos del Sistema</b>	<b>11</b>
4.1. Objetivos de Negocio . . . . .	11
4.2. Objetivos del Sistema . . . . .	11
4.3. Requisitos funcionales . . . . .	11
4.4. Requisitos no funcionales . . . . .	11
4.5. Requisitos de información . . . . .	11
4.6. Reglas de negocio . . . . .	12
4.7. Análisis GAP . . . . .	12
<b>5. Análisis del Sistema</b>	<b>15</b>
5.1. Modelo Conceptual . . . . .	15
5.2. Modelo de Casos de Uso . . . . .	15
5.3. Modelo de Interfaz de Usuario . . . . .	15

5.4. Modelo de Comportamiento . . . . .	15
<b>6. Diseño del Sistema</b>	<b>17</b>
6.1. Arquitectura del Sistema . . . . .	17
6.1.1. Diseño de alto nivel . . . . .	17
6.1.2. Diseño detallado . . . . .	17
6.2. Parametrización del software base . . . . .	18
6.3. Diseño Físico de Datos . . . . .	18
6.4. Diseño de la Interfaz de Usuario . . . . .	18
<b>7. Construcción del Sistema</b>	<b>21</b>
7.1. Entorno de Construcción . . . . .	21
7.2. Código Fuente . . . . .	21
7.3. Scripts de Base de datos . . . . .	21
<b>8. Pruebas del Sistema</b>	<b>23</b>
8.1. Estrategia . . . . .	23
8.2. Pruebas Unitarias . . . . .	23
8.3. Pruebas de Integración . . . . .	23
8.4. Pruebas de Sistema . . . . .	23
8.4.1. Pruebas Funcionales . . . . .	24
8.4.2. Pruebas No Funcionales . . . . .	24
8.5. Pruebas de Aceptación . . . . .	24
<b>9. Despliegue del Sistema</b>	<b>27</b>
9.1. Arquitectura Física . . . . .	27
9.2. Instrucciones de despliegue . . . . .	27
9.2.1. Requisitos previos . . . . .	27
9.2.2. Inventario de componentes . . . . .	27
9.2.3. Procedimientos de instalación . . . . .	27
9.2.4. Pruebas de implantación . . . . .	27
9.3. Instrucciones para la operación del sistema y mantenimiento del nivel de servicio . . . . .	28
<b>III Epílogo</b>	<b>29</b>
<b>10. Conclusiones</b>	<b>31</b>
10.1. Objetivos alcanzados . . . . .	31
10.2. Lecciones aprendidas . . . . .	31
10.3. Trabajo futuro . . . . .	31
<b>Información sobre Licencia</b>	<b>33</b>
<b>Bibliografía</b>	<b>35</b>

<b>IV</b>	<b>Anexos</b>	<b>37</b>
<b>A.</b>	<b>Manual del desarrollador</b>	<b>39</b>
A.1.	Introducción . . . . .	39
A.2.	Preparación del entorno de trabajo . . . . .	39
A.3.	Consideraciones generales sobre el desarrollo . . . . .	39
A.4.	Instrucciones para construcción . . . . .	39
<b>B.</b>	<b>Manual de usuario</b>	<b>41</b>
B.1.	Introducción . . . . .	41
B.2.	Instalación . . . . .	41
B.3.	Uso del sistema . . . . .	41
<b>C.</b>	<b>Comandos útiles de latex</b>	<b>43</b>
C.1.	Tipos de tablas . . . . .	43
C.2.	Una imagen . . . . .	43
C.3.	Dos imágenes . . . . .	44
C.4.	Para referenciar archivos del documento . . . . .	44
C.5.	Referenciar citas bibliográficas . . . . .	45
C.6.	Listas . . . . .	45
C.7.	URL . . . . .	45
C.8.	Notas pie de página . . . . .	46
C.9.	Estilos . . . . .	46

# Índice de figuras

---

C.1. TFG . . . . .	44
C.2. Figuras de dispositivos de RA . . . . .	44

# Índice de tablas

---

C.1. Dispositivos de Gestión Integral . . . . .	43
C.2. Dispositivos de Gestión Integral . . . . .	43



Parte I  
Prolegómeno



La primera parte de la memoria del Trabajo Fin de Grado (TFG) debe contener una introducción y una planificación del proyecto. La introducción es un capítulo que, a modo de resumen, debe contener una breve descripción del contexto de la disciplina en la que el proyecto tiene aplicación y la motivación para su desarrollo, así como del alcance previsto. En el segundo capítulo deberán describirse los antecedentes del proyecto, esto es, su contexto y la situación actual. Además, se desarrollará el estado de la técnica en relación con la propuesta de proyecto. El tercer capítulo debe incluir el plan de gestión del proyecto. La planificación deberá ajustarse a las prácticas de ingeniería en general, y de la ingeniería del software en particular. Deberá tener en cuenta los plazos, los entregables (documentos y software), los recursos (humanos y de equipamiento inventariable) y el método de ingeniería de software a emplear.

# 1. Introducción

---

A continuación, se describe la motivación del presente proyecto y su alcance. También se incluye un glosario de términos y la organización del resto de la presente documentación.

## 1.1. Motivación

Qué motivación nos ha llevado a su desarrollo. Contexto y ámbito en el que se desarrolla el proyecto.

## 1.2. Alcance y Objetivos

Esta sección debe describir a qué elementos organizativos de la organización Cliente afecta el desarrollo del nuevo sistema. También debe describir los principales objetivos que se esperan alcanzar cuando el sistema a desarrollar esté en producción.

## 1.3. Glosario de términos

Esta sección debe contener una lista ordenada alfabéticamente de los principales términos, acrónimos y abreviaturas específicos del dominio del problema, especialmente de los que se considere que su significado deba ser aclarado. Cada término, acrónimo o abreviatura deberá acompañarse de su definición.

## 1.4. Organización del documento

Descripción de los contenidos de la presente memoria, así como del software entregado en soporte informático.



## 2. Antecedentes

---

En este capítulo se detalla la situación actual que origina el desarrollo o la mejora de un sistema informático. Asimismo, se describen las alternativas tecnológicas existentes.

### 2.1. Contexto

Los trabajos de desarrollo de software pretenden dar una solución a la problemática existente en una determinada organización o de un conjunto de personas. Por ello es preciso describir la situación actual, incluyendo información sobre el entorno tecnológico existente (si lo hubiera). En tal caso, es preciso analizar las fortalezas y debilidades del mismo.

Adicionalmente, puede incluirse información sobre los modelos de procesos de negocio actuales existentes en la organización. Un proceso de negocio no es más que es el conjunto de actividades o tareas relacionadas entre sí y estructuradas en una secuencia específica para producir un servicio o producto (cumple un determinado objetivo de negocio) para un cliente en particular. Pueden emplearse notaciones como BPMN o diagramas de actividades UML.

### 2.2. Estado de la técnica

Antes de comenzar con la planificación del proyecto, es preciso realizar un estudio de las invenciones existentes que permitan solventar la problemática descrita en la sección anterior. Para ello se utilizarán buscadores web generales, como Google; buscadores especializados en trabajos científicos, como Google Scholar; y repositorios de aplicaciones, como Apple App Store.

Este estudio permitirá, por un lado, descubrir el grado de novedad de la nueva solución informática a desarrollar y, por otro lado, descubrir soluciones ya existentes (habitualmente de software libre) que puedan servir de plataforma base para el nuevo desarrollo.



## **3. Plan de gestión de proyecto**

---

En esta sección se describen todos los aspectos relativos a la gestión del proyecto: metodología, organización, costes, planificación, riesgos y aseguramiento de la calidad.

### **3.1. Metodología de desarrollo**

Definición del proceso de desarrollo, ciclo de vida y metodología empleada durante la elaboración del proyecto. Las fases y/o iteraciones que proponga el método empleado deberán quedar recogidas en la planificación que se detalle más adelante.

### **3.2. Tecnologías**

Deben resumirse (usando las referencias adecuadas) y justificarse las tecnologías/software que se van a aplicar en el proyecto.

### **3.3. Planificación del proyecto**

Estimación temporal y definición del calendario básico (hitos principales e iteraciones). Desarrollo de la planificación detallada, utilizando un diagrama de Gantt. Los diagramas de Gantt que se vean correctamente (girados y divididos si hace falta).

Se debe incluir una comparación cuantitativa del tiempo y el esfuerzo realmente invertido frente al estimado y planificado. Estos datos pueden recogerse del sistema de gestión de tareas empleado para el seguimiento del proyecto.

### **3.4. Organización**

Relación de las personas (roles) involucradas en el proyecto y de cómo se estructuran las relaciones entre las mismas para ejecutar el proyecto. Relación de los recursos inventariables utilizados en el proyecto: equipamiento informático (hardware y software), herramientas empleadas, etc.

### **3.5. Costes**

Estudio y presupuesto de los costes de los recursos (humanos y materiales) descritos anteriormente, necesarios para el proyecto. Para el cálculo de costes de personal puede consultarse alguna fuente disponible en Internet ajustada a la realidad del mercado laboral en España. El cálculo del coste del personal del proyecto puede hacerse en personas-mes, y luego hacer la correspondencia al coste monetario.

### **3.6. Riesgos**

Enumeración de los riesgos del proyecto, indicando su posible impacto (efecto que la ocurrencia del citado riesgo tendría en el desarrollo del proyecto) y la probabilidad de ocurrencia. Una vez los riesgos son identificados y priorizados, hay que definir los planes necesarios para reducir los efectos del riesgo una vez se haya materializado o disminuir que este ocurra.

### **3.7. Gestión de la configuración**

Descripción de las estrategias a seguir para el control de las versiones del código fuente, así como de la gestión de las *releases*.

### **3.8. Aseguramiento de calidad**

En esta sección se incluirán las actividades y tareas relacionadas con el aseguramiento de calidad a realizar durante el desarrollo del software. Se incluirán los estándares, prácticas y normas aplicables durante el desarrollo del software.

También, deberán recogerse los diferentes tipos de revisiones, verificaciones y validaciones que se van a llevar a cabo, los criterios para la aceptación o rechazo de cada producto y los procedimientos para implementar acciones correctoras o preventivas.

# Parte II

## Desarrollo



En esta parte se debe describir el desarrollo del proyecto siguiendo la metodología empleada. Sus capítulos no deben ser una descripción exhaustiva de todos los documentos, diagramas, código fuente y, en general, entregables generados, sino más bien una explicación resumida del desarrollo, estructurada según las etapas principales del proceso de ingeniería. Deben seleccionarse aquellos diagramas, fragmentos de código y secciones de los entregables que sean más significativos para dicha explicación. La totalidad de los entregables resultado del proyecto se ubicarán en los anexos y/o en el material digital que acompañe al proyecto.

## 4. Requisitos del Sistema

---

En este capítulo se presentan los objetivos y el catálogo de requisitos del nuevo sistema informático. Opcionalmente, se incluye el análisis de la brecha entre los requisitos planteados y la solución base seleccionada.

### 4.1. Objetivos de Negocio

Esta sección debe contener los objetivos de negocio que se esperan alcanzar cuando el sistema software a desarrollar esté en producción. Adicionalmente, se pueden incluir los modelos de procesos, que normalmente son los modelos de procesos de negocio actuales con ciertas mejoras.

### 4.2. Objetivos del Sistema

Esta sección debe contener la especificación de los objetivos o requisitos generales del sistema.

### 4.3. Requisitos funcionales

Descripción completa de la funcionalidad que ofrece el sistema.

### 4.4. Requisitos no funcionales

Descripción de otros requisitos (relacionados con la calidad del software) que el sistema deberá satisfacer: portabilidad, seguridad, mantenibilidad, accesibilidad, usabilidad, fiabilidad, eficiencia, etc. Puede usarse de referencia lo indicado en la norma ISO/IEC 25010.

### 4.5. Requisitos de información

En esta sección se describen los requisitos de gestión de información (datos) que el sistema debe gestionar.

## **4.6. Reglas de negocio**

En el desarrollo del sistema, hay que tener en cuenta las denominadas reglas de negocio, es decir, el conjunto de restricciones, normas o políticas de la organización que deben ser respetadas por el sistema, las cuales suelen ser cambiantes.

## **4.7. Análisis GAP**

Esta sección es opcional y es de aplicación cuando en la revisión de la técnica se hayan localizado soluciones informáticas que permitan servir de base para el nuevo sistema software. En esta sección se debe identificar y medir las diferencias entre lo que proporcionan dichas soluciones y los requisitos definidos para el proyecto. El resultado de este análisis permitirá identificar cuáles de éstos requisitos ya están solventados total o parcialmente por los distintos sistemas y poder elegir uno de ellos.





# 5. Análisis del Sistema

---

Este capítulo cubre el análisis del sistema de información a desarrollar, haciendo uso del lenguaje de modelado UML.

## 5.1. Modelo Conceptual

A partir de los requisitos de información, se desarrollará un diagrama conceptual de clases UML, identificando las clases, atributos, relaciones, restricciones adicionales y reglas de derivación necesarias.

## 5.2. Modelo de Casos de Uso

A partir de los requisitos funcionales descritos anteriormente, se emplearán los casos de uso como mecanismo para representar las interacciones entre los actores y el sistema bajo estudio. Para cada caso de uso deberá indicarse los actores implicados, las precondiciones y postcondiciones, los pasos que conforman el escenario principal y el conjunto de posibles escenarios alternativos. Los actores pueden ser roles de personas físicas, sistemas externos o incluso el tiempo (para eventos temporales).

## 5.3. Modelo de Interfaz de Usuario

En esta sección se deberá incluir un prototipo de baja fidelidad o mockup de la interfaz de usuario del sistema. Además, es preciso elaborar un diagrama de navegación, reflejando la secuencia de pantallas a las que tienen acceso los diferentes roles de usuario y la conexión entre éstas.

## 5.4. Modelo de Comportamiento

En esta sección opcional, se incluirá el modelo de comportamiento desarrollado a partir de los casos de uso definidos. Para ello, se realizarán los diagramas de secuencia del sistema, donde se identificarán las operaciones o servicios del sistema. Luego, se detallará el contrato de las operaciones identificadas.



# 6. Diseño del Sistema

---

En este capítulo se recoge el diseño de la arquitectura del sistema informático, la parametrización del software base (opcional), el diseño físico de datos y el diseño detallado de la interfaz de usuario.

## 6.1. Arquitectura del Sistema

En esta sección se define la arquitectura general del sistema, especificando la infraestructura tecnológica necesaria para dar soporte al software y la estructura de los componentes que lo forman. Nos centraremos, en este caso, en la arquitectura lógica o de desarrollo del software, dejando para un capítulo posterior la arquitectura física o de despliegue.

### 6.1.1. Diseño de alto nivel

Para describir la arquitectura se puede emplear el modelo C4 [Brown \(2022\)](#), el cual permite representar la arquitectura del software mediante varios diagramas a distintos niveles de abstracción, a saber, el *diagrama de contexto* (reflejando las interacciones entre el sistema, sus usuarios y los sistemas externos), el *diagrama de contenedores* (reflejando el conjunto de aplicaciones, servicios y almacenes de datos que componen el sistema) y los *diagramas de componentes* (representando los componentes o módulos que conforman las aplicaciones o servicios a desarrollar).

Existen diferentes patrones o estilos arquitectónicos. Por ejemplo, en los sistemas web de información es común la utilización del patrón Layers (Capas), con el cual estructuramos el sistema en un número apropiado de capas, de forma que todos los componentes de una misma capa trabajan en el mismo nivel de abstracción y los servicios proporcionados por la capa superior utilizan internamente los servicios proporcionados por la capa inmediatamente inferior.

### 6.1.2. Diseño detallado

En el nivel de abstracción más bajo de los propuestos en el modelo C4, nos encontramos con los diagramas de clases UML que representan fielmente la estructura de código fuente. Adicionalmente, se puede definir la interacción existente entre las clases de objetos que permitan responder a eventos externos, mediante diagramas de secuencia o de comunicación.



## **6.2. Parametrización del software base**

En esta sección (opcional), se detallan las modificaciones a realizar sobre el software base, que son requeridas para la correcta construcción del sistema. En esta sección incluiremos las actuaciones necesarias sobre la interfaz de administración del sistema, sobre el código fuente o sobre el modelo de datos.

## **6.3. Diseño Físico de Datos**

En esta sección se define la estructura física de datos que utilizará el sistema, a partir del modelo de conceptual de clases, de manera que teniendo presente los requisitos establecidos para el sistema de información y las particularidades del entorno tecnológico, se consiga un acceso eficiente de los datos. La estructura física se compone de tablas, índices, procedimientos almacenados, secuencias y otros elementos dependientes del SGBD a utilizar.

## **6.4. Diseño de la Interfaz de Usuario**

En esta sección se detallarán los principios a tener en cuenta a la hora de implementar la interfaz de usuario del sistema, incluyendo el conjunto de estilos (colores, tipografías, iconografía, etc.) y el conjunto de componentes visuales que se utilizarán. Se incluirá, además, un prototipo de alta fidelidad con el diseño de algunas de las pantallas del sistema.





# 7. Construcción del Sistema

---

Este capítulo trata sobre todos los aspectos relacionados con la implementación del sistema en código, haciendo uso de un determinado entorno tecnológico.

## 7.1. Entorno de Construcción

En esta sección se debe indicar el marco tecnológico utilizado para la construcción del sistema: entorno de desarrollo (IDE), lenguaje de programación, herramientas de ayuda a la construcción y despliegue, control de versiones, repositorio de componentes, integración continua, etc.

## 7.2. Código Fuente

Organización del código fuente, describiendo la utilidad de los diferentes ficheros y su distribución en paquetes o directorios. Asimismo, se incluirá algún extracto significativo de código fuente que sea de interés para ilustrar algún algoritmo o funcionalidad específica del sistema.

## 7.3. Scripts de Base de datos

Organización del código fuente, describiendo la utilidad de los diferentes ficheros y su distribución en paquetes o directorios. Asimismo, se incluirá el script de algún disparador o un procedimiento almacenado, que sea de interés para ilustrar algún aspecto concreto de la gestión de la base de datos.



# 8. Pruebas del Sistema

---

En este capítulo se presenta el plan de pruebas del sistema de información, incluyendo los diferentes tipos de pruebas que se han llevado a cabo, ya sean manuales (mediante listas de comprobación) o automatizadas mediante algún software específico de pruebas.

## 8.1. Estrategia

En esta sección se debe incluir el alcance de las pruebas, hasta donde se pretende llegar con ellas, si se registrarán todas o sólo aquellas de un cierto tipo y cómo se interpretarán y evaluarán los resultados. También, se incluirá el procedimiento a seguir para las pruebas de regresión, esto es, la repetición de ciertas pruebas para comprobar que nuevos cambios que se vayan introduciendo no originen errores en el software ya probado. Describir en esta sección cuáles serán los perfiles y participantes necesarios para la ejecución de cada uno de los niveles de prueba. Incluir en este apartado los requisitos de los entornos hardware/software donde se ejecutarán las pruebas.

## 8.2. Pruebas Unitarias

Las pruebas unitarias tienen por objetivo localizar errores en cada nuevo artefacto software desarrollado, antes que se produzca la integración con el resto de artefactos del sistema.

## 8.3. Pruebas de Integración

Este tipo de pruebas tienen por objetivo localizar errores en módulos o subsistemas completos, analizando la interacción entre varios artefactos software.

## 8.4. Pruebas de Sistema

En esta actividad se realizan las pruebas de sistema de modo que se asegure que el sistema cumple con todos los requisitos establecidos: funcionales, de almacenamiento, reglas de negocio y no funcionales. Se suelen desarrollar en un entorno específico para pruebas.

### **8.4.1. Pruebas Funcionales**

Con estas pruebas se analiza el buen funcionamiento de la implementación de los flujos normales y alternativos de los distintos casos de uso del sistema.

### **8.4.2. Pruebas No Funcionales**

Estas pruebas pretenden comprobar el funcionamiento del sistema, con respecto a los requisitos no funcionales identificados: eficiencia, seguridad, etc.

## **8.5. Pruebas de Aceptación**

El objetivo de estas pruebas es demostrar que el producto está listo para el paso a producción. Suelen ser las mismas pruebas que se realizaron anteriormente pero en el entorno de producción. En estas pruebas, es importante la participación del cliente final.







# 9. Despliegue del Sistema

---

Este capítulo recoge la arquitectura física planteada para el sistema, las instrucciones para su despliegue y las instrucciones para la operación y mantenimiento del nivel de servicio.

## 9.1. Arquitectura Física

En este apartado, describimos los principales elementos hardware que forman la arquitectura física de nuestro sistema. Se debe incluir un modelo de despliegue en el cual se describe cómo los elementos software son instalados y ejecutados en los elementos hardware (físicos o virtuales). También se incluyen las especificaciones y los requisitos del hardware (servidores, etc.), así como de los elementos software (sistemas operativos, servicios, aplicaciones, etc.) necesarios.

## 9.2. Instrucciones de despliegue

A continuación, se describen las instrucciones de instalación del sistema sobre la infraestructura física descrita anteriormente.

### 9.2.1. Requisitos previos

Requisitos hardware y software para la correcta instalación del sistema.

### 9.2.2. Inventario de componentes

Lista de los componentes hardware y software que se incluyen en la versión del producto.

### 9.2.3. Procedimientos de instalación

Procedimientos de instalación y configuración de cada componente hardware y software (base y desarrollado) para asegurar la correcta instalación y explotación del sistema, así como aquellos procedimientos necesarios de migración/carga de datos.

### 9.2.4. Pruebas de implantación

Descripción de las pruebas a realizar después de la instalación del sistema.

### **9.3. Instrucciones para la operación del sistema y mantenimiento del nivel de servicio**

Procedimientos necesarios para asegurar el correcto funcionamiento, rendimiento, disponibilidad y seguridad del sistema: back-ups, chequeo de logs, etc. También, es preciso indicar claramente aquellas actuaciones precisas necesarias para el mantenimiento preventivo del sistema y así prevenir posibles fallos en el mismo.

# Parte III

## Epílogo

En esta parte se incluyen las conclusiones, la información relativa a la licencia del software y la bibliografía.

# 10. Conclusiones

---

En este último capítulo se detallan las lecciones aprendidas tras el desarrollo del presente proyecto y se identifican las posibles oportunidades de mejora sobre el software desarrollado.

## 10.1. Objetivos alcanzados

Este apartado debe resumir los objetivos generales y específicos alcanzados, relacionándolos con todo lo descrito en el capítulo de introducción.

## 10.2. Lecciones aprendidas

A continuación, se detallan las buenas prácticas adquiridas, tanto tecnológicas como procedimentales, así como cualquier otro aspecto de interés. Resumir cuantitativamente el tiempo y esfuerzo dedicados al proyecto a lo largo de su desarrollo, en lugar de escribir un sencillo 'he trabajado mucho en este proyecto'.

## 10.3. Trabajo futuro

En esta sección, se presentan las diversas áreas u oportunidades de mejora detectadas durante el desarrollo del proyecto y que podrán ser abarcadas en futuras versiones del software.

Los elementos aquí descritos deben estar en relación con lo relatado en el apartado de objetivos y alcance del proyecto descritos en la introducción.



# Información sobre Licencia

---

## Información sobre Licencia

Incluir aquí la información relativa a la licencia seleccionada para la documentación y software del presente proyecto.





# Bibliografía

---

Brown, S. (2022). The c4 model for visualising software architecture.

Hevner, A. R., March, S. T., Park, J., y Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75–105.

Mota, J. M., Ruiz-Rube, I., Dodero, J. M., y Arnedillo-Sánchez, I. (2018). Augmented reality mobile app development for all. *Computers & Electrical Engineering*, 65:250–260.



# Parte IV

## Anexos

En esta última parte quedarán recogidas los manuales necesarios para el manejo de la aplicación resultado del desarrollo. Si se ha realizado algún tipo de evaluación de la solución proporcionada, más allá de las pruebas del sistema, también deberá venir recogida en un capítulo separado dentro de esta parte. Pueden consultarse diversos tipos de evaluaciones sobre sistemas de información en [Hevner et al. \(2004\)](#): casos de estudio, análisis estático, análisis dinámico, simulación, experimento controlado, etc.

# A. Manual del desarrollador

---

A continuación se recogen las instrucciones necesarias para evolucionar el software. Este manual está dirigido a los desarrolladores que pretenden extender o modificar el código fuente, con el fin de incorporar nuevas funcionalidades o modificar las ya existentes. A lo largo de este capítulo se deberán hacer referencias explícitas a aquellos epígrafes de los capítulos de Diseño, Construcción, Pruebas y Despliegue del Sistema que resulten de interés.

## A.1. Introducción

Resumen de los principales objetivos, ámbito, características y alcance del software desarrollado.

## A.2. Preparación del entorno de trabajo

Descripción de los requisitos (hardware y software) previos. Datos de interés relativos al control de versiones del software. Detalles sobre la instalación en local del entorno de desarrollo y, si fuesen necesarios, de otros componentes como bases de datos, servidores de aplicaciones, etc.

## A.3. Consideraciones generales sobre el desarrollo

Aspectos importantes a tener en cuenta a la hora de modificar y extender el código fuente, guías de estilo, etc. Asimismo, se detallarán las directrices que sean de aplicación a la hora de realizar pruebas sobre las nuevas mejoras introducidas.

## A.4. Instrucciones para construcción

Secuencia de pasos requeridos para llevar a cabo la compilación del código fuente y así poder construir y depurar el software sobre una máquina de desarrollo.



# **B. Manual de usuario**

---

Las instrucciones de uso del software se detallan a continuación. Este manual se dirige al usuario final del software objetivo de este proyecto.

## **B.1. Introducción**

Resumen de los principales objetivos, ámbito, características y alcance del software desarrollado.

## **B.2. Instalación**

Se detallarán los pasos necesarios para la obtención e instalación del software, así como los requisitos previos de hardware y software.

## **B.3. Uso del sistema**

Describir todos los aspectos necesarios para una utilización efectiva y eficiente de las características del sistema por parte de los usuarios.





# C. Comandos útiles de latex

---

Veremos en cada sección ejemplos para la utilización de Tablas o imágenes entre otras características de L<sup>A</sup>T<sub>E</sub>X.

## C.1. Tipos de tablas

En esta sección mostramos algunos de los tipos de tablas más usuales en los TFG/TFM para que sirvan de ejemplo. Las tablas siguen la estructura de la normativa APA con la referencia y el título en la parte superior.

Tabla C.1

*Dispositivos de Gestión Integral*

Gestión Integral			
Fabricante	Modelo	Clase	Motor
BMW	Serie 3	Berlina	Diésel
Peugeot	508	Berlina	Gasolina
Chrysler	Voyager	Monovolumen	Gasolina
Land Rover	Defender	Todoterreno	Gasolina

Segundo ejemplo de tabla, en este caso con líneas en todos los bordes, que no cumple la normativa APA, pero puede ser útil para presentar grandes cantidades de datos.

Tabla C.2

*Dispositivos de Gestión Integral*

Gestión Integral			
<b>Fabricante</b>	<b>Modelo</b>	<b>Clase</b>	<b>Motor</b>
BMW	Serie 3	Berlina	Diésel
Peugeot	508	Berlina	Gasolina
Chrysler	Voyager	Monovolumen	Gasolina
Land Rover	Defender	Todoterreno	Gasolina

## C.2. Una imagen

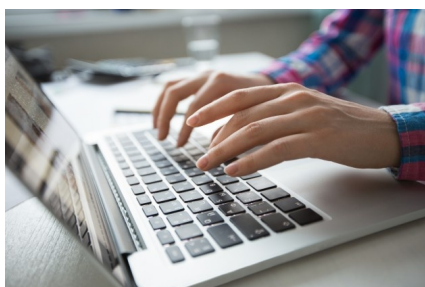
Para añadir una imagen a Overleaf, hay que subirla desde "Add Files" en la carpeta "figuras". Luego, utilizamos el comando de "figure".



Figura C.1: TFG

### C.3. Dos imágenes

En este caso debemos trabajar con "subfigure". Hay que trabajar en este caso con dos tamaños, el primero es el que va a ocupar el espacio donde se va a poner la imagen, y el segundo es el tamaño de la imagen en ese espacio.



(a) Principio



(b) Final

Figura C.2: Figuras de dispositivos de RA

### C.4. Para referenciar archivos del documento

Aquí tenemos la referencia a cada imagen del grupo de dos imágenes, referencia a la imagen de la izquierda [C.2a](#) y esta a la de la derecha [C.2b](#), pero también podemos referenciar la imagen completa [C.2](#). También podemos hacer referencia a una [Tabla C.1](#)

## C.5. Referenciar citas bibliográficas

Aquí vamos a detallar los pasos necesarios para crear una bibliografía y citar dichas entradas. Primero cuando tengamos el documento que queremos referenciar deberemos su referencia en formato *Bibtex* y pegarlo en el archivo de "bibliography.bib". Una vez ya la tenemos copiada, para citar una referencia bibliográfica escribiremos `\cite{}` y aquí escribiremos la cita correspondiente que queremos citar}, donde solamente el año de la cita estará entre paréntesis o también podremos escribir `\citep{}`, donde no solamente estará el año entre paréntesis sino toda la cita. Este es un ejemplo usando *cite* [Mota et al. \(2018\)](#) y usando *citep* ([Mota et al. 2018](#)).

## C.6. Listas

A continuación vamos a explicar como poder añadir una lista, ya sean numeradas o no numeradas. Primero antes que nada, señalar que las listas enumeradas son las que siguen una enumeración, y son creadas dentro del entorno *enumerate*. sin embargo, las no numeradas consisten en *ítems* que a su izquierda tienen un símbolo no numérico, y son creadas con el entorno *itemize*

Lista numerada. Así empieza una lista numerada por ejemplo la compra:

1. Pan
2. Huevos
3. Leche
4. Café

Lista no numerada. Así empieza una lista no enumerada de por ejemplo, mis aficiones:

- Hacer deporte
- escuchar música
- ir al cine
- dar un paseo

## C.7. URL

Puedes enlazar una web externa mediante el comando url: <https://www.uca.es/>. También se puede vincular un enlace a un texto mediante el comando href: [Universidad de Cádiz](#).

## C.8. Notas pie de página

En este capítulo vamos a definir las palabras que consideremos que tengan alguna dificultad en cuanto a su significado. En este caso, son la UCA<sup>1</sup>, la asignatura IDA<sup>2</sup> y la palabra estudiar<sup>3</sup>

## C.9. Estilos

Se pueden aplicar estilos al texto como **negritas**, *cursiva* y subrayado. También se **pueden aplicar colores**, y combinar estilos. Se recomienda no utilizar negritas, solo para hacer énfasis, pero no demasiado y cursiva solo para señalar términos en inglés.

---

<sup>1</sup>UCA - Universidad de Cádiz

<sup>2</sup>IDA - Información y Documentación Administrativa

<sup>3</sup>Estudiar - Ejercitar el entendimiento para alcanzar o comprender algo