

ESCUELA SUPERIOR DE INGENIERÍA

GRADO EN INGENIERÍA INFORMÁTICA

**REVISIÓN SISTEMÁTICA DE LA LITERATURA**

AUTOR(A): Ángel Rafael González Toro

11 de mayo de 2017



ESCUELA SUPERIOR DE INGENIERÍA

GRADO EN INGENIERÍA INFORMÁTICA

**REVISIÓN SISTEMÁTICA DE LA LITERATURA**

AUTOR(A): ÁNGEL RAFAEL GONZÁLEZ TORO  
DIRECTOR(A): IVAN RUÍZ RUBE

Cádiz, 11 de mayo de 2017



## ***Agradecimientos***

*Introduzca aquí, si lo desea, los agradecimientos.*

## 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:** Lista de palabras clave que reflejen el contenido del trabajo en aras de facilitar su búsqueda en sistemas bibliográficos.

# Índice general

<b>I</b>	<b>Prolegómeno fENOMENO</b>	<b>1</b>
<b>1.</b>	<b>Introducción</b>	<b>5</b>
1.1.	Motivación . . . . .	5
1.2.	Alcance . . . . .	5
1.3.	Glosario de Términos . . . . .	5
1.4.	Organización del documento . . . . .	5
<b>2.</b>	<b>Planificación</b>	<b>7</b>
2.1.	Metodología de desarrollo . . . . .	7
2.2.	Planificación del proyecto . . . . .	7
2.3.	Organización . . . . .	7
2.4.	Costes . . . . .	7
2.5.	Riesgos . . . . .	8
2.6.	Aseguramiento de calidad . . . . .	8
<b>II</b>	<b>Desarrollo</b>	<b>9</b>
<b>3.</b>	<b>Requisitos del Sistema</b>	<b>13</b>
3.1.	Situación actual . . . . .	13
3.1.1.	Procesos de Negocio . . . . .	13
3.1.2.	Entorno Tecnológico . . . . .	13
3.1.3.	Fortalezas y Debilidades . . . . .	13
3.2.	Necesidades de Negocio . . . . .	13
3.2.1.	Objetivos de Negocio . . . . .	14
3.2.2.	Procesos de Negocio . . . . .	14
3.3.	Objetivos del Sistema . . . . .	14
3.4.	Catálogo de Requisitos . . . . .	14
3.4.1.	Requisitos funcionales . . . . .	14
3.4.2.	Requisitos no funcionales . . . . .	14
3.4.3.	Reglas de negocio . . . . .	14
3.4.4.	Requisitos de información . . . . .	14
3.5.	Alternativas de Solución . . . . .	14
3.6.	Solución Propuesta . . . . .	15

<b>4. Análisis del Sistema</b>	<b>17</b>
4.1. Modelo Conceptual . . . . .	17
4.2. Modelo de Casos de Uso . . . . .	17
4.2.1. Actores . . . . .	17
4.3. Modelo de Comportamiento . . . . .	17
4.4. Modelo de Interfaz de Usuario . . . . .	17
<b>5. Diseño del Sistema</b>	<b>19</b>
5.1. Arquitectura del Sistema . . . . .	19
5.1.1. Arquitectura Física . . . . .	19
5.1.2. Arquitectura Lógica . . . . .	19
5.2. Parametrización del software base . . . . .	20
5.3. Diseño Físico de Datos . . . . .	20
5.4. Diseño detallado de Componentes . . . . .	20
5.5. Diseño detallado de la Interfaz de Usuario . . . . .	20
<b>6. Construcción del Sistema</b>	<b>21</b>
6.1. Entorno de Construcción . . . . .	21
6.2. Código Fuente . . . . .	21
6.3. Scripts de Base de datos . . . . .	21
<b>7. Pruebas del Sistema</b>	<b>23</b>
7.1. Estrategia . . . . .	23
7.2. Entorno de Pruebas . . . . .	23
7.3. Roles . . . . .	23
7.4. Niveles de Pruebas . . . . .	23
7.4.1. Pruebas Unitarias . . . . .	23
7.4.2. Pruebas de Integración . . . . .	24
7.4.3. Pruebas de Sistema . . . . .	24
7.4.4. Pruebas de Aceptación . . . . .	24
<b>III Epílogo</b>	<b>25</b>
<b>8. Manual de implantación y explotación</b>	<b>29</b>
8.1. Introducción . . . . .	29
8.2. Requisitos previos . . . . .	29
8.3. Inventario de componentes . . . . .	29
8.4. Procedimientos de instalación . . . . .	29
8.5. Pruebas de implantación . . . . .	29
8.6. Procedimientos de operación y nivel de servicio . . . . .	29
<b>9. Manual de usuario</b>	<b>31</b>
9.1. Introducción . . . . .	31
9.2. Instalación . . . . .	31
9.3. Uso del sistema . . . . .	31



<b>10. Manual del desarrollador</b>	<b>33</b>
10.1. Introducción . . . . .	33
10.2. Preparación del entorno de trabajo . . . . .	33
10.3. Consideraciones generales sobre el desarrollo . . . . .	33
10.4. Instrucciones para construcción y despliegue . . . . .	33
<b>11. Conclusiones</b>	<b>35</b>
11.1. Objetivos alcanzados . . . . .	35
11.2. Lecciones aprendidas . . . . .	35
11.3. Trabajo futuro . . . . .	35
<b>Bibliografía</b>	<b>37</b>
<b>Información sobre Licencia</b>	<b>39</b>



# Índice de figuras



# Índice de tablas



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.

El segundo capítulo debe incluir una planificación 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.



# Capítulo 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

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.



## Capítulo 2

# Planificación

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

### 2.1. 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.

### 2.2. 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.

### 2.3. 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.

### 2.4. 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 pueden consultarse las tablas salariales de la UCA para el personal técnico de apoyo contratado laboral [CCOO, 2010], o bien otras más ajustadas a la realidad. El cálculo del coste del personal del proyecto debe hacerse en personas-mes, y luego

hacer la correspondencia al coste monetario.

## **2.5. 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.

## **2.6. 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 en CD/DVD que acompañe al proyecto.



## Capítulo 3

# Requisitos del Sistema

En esta sección se detalla la situación actual de la organización y las necesidades de la misma, que originan el desarrollo o mejora de un sistema informático. Luego se presentan los objetivos y el catálogo de requisitos del nuevo sistema. Finalmente se describen las diferentes alternativas tecnológicas y el análisis de la brecha entre los requisitos planteados y la solución base seleccionada, si aplica.

### 3.1. Situación actual

Esta sección debe contener información sobre la situación actual de la organización para la que se va a desarrollar el sistema software.

#### 3.1.1. Procesos de Negocio

Esta sección debe contener información sobre los modelos de procesos de negocio actuales, que suelen ser la base de los modelos de procesos de negocio a implantar.

#### 3.1.2. Entorno Tecnológico

Esta sección debe contener información general sobre el entorno tecnológico en la organización del cliente antes del comienzo del desarrollo del sistema software, incluyendo hardware, redes, software, etc.

#### 3.1.3. Fortalezas y Debilidades

Esta sección debe contener información sobre los aspectos positivos y negativos del negocio actual de la organización para la que se va a desarrollar el sistema software.

### 3.2. Necesidades de Negocio

Esta sección debe contener información sobre los objetivos de negocio de clientes y usuarios, incluyendo los modelos de procesos de negocio a implantar.

### **3.2.1. Objetivos de Negocio**

Esta sección debe contener los objetivos de negocio que se esperan alcanzar cuando el sistema software a desarrollar esté en producción.

### **3.2.2. Procesos de Negocio**

Esta sección, debe contener los modelos de procesos de negocio a implantar, que normalmente son los modelos de procesos de negocio actuales con ciertas mejoras.

## **3.3. Objetivos del Sistema**

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

## **3.4. Catálogo de Requisitos**

Esta sección debe contener la descripción del conjunto de requisitos específicos del sistema a desarrollar para satisfacer las necesidades de negocio del cliente.

### **3.4.1. Requisitos funcionales**

Descripción completa de la funcionalidad que ofrece el sistema.

### **3.4.2. Requisitos no funcionales**

Descripción de otros requisitos (relacionados con la calidad del software) que el sistema deberá satisfacer: portabilidad, seguridad, estándares de obligado cumplimiento, accesibilidad, usabilidad, etc.

### **3.4.3. 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.

### **3.4.4. Requisitos de información**

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

## **3.5. Alternativas de Solución**

En esta sección, se debe ofrecer un estudio del arte de las diferentes alternativas tecnológicas que permitan satisfacer los requerimientos del sistema, para luego seleccionar (si procede) la herramienta o conjunto de herramientas que utilizaremos como base para el software a desarrollar.

### **3.6. Solución Propuesta**

Si se ha optado por utilizar un software de base, debemos identificar y medir las diferencias entre lo que proporciona este software 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 el sistema base y cuales tendremos que diseñar e implementar la propuesta de solución.



## Capítulo 4

# Análisis del Sistema

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

### 4.1. 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.

### 4.2. 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.

#### 4.2.1. Actores

En este apartado se describirán los diferentes roles que juegan los usuarios que interactúan con el sistema. Los actores pueden ser roles de personas físicas, sistemas externos o incluso el tiempo (eventos temporales).

### 4.3. Modelo de Comportamiento

A partir de los casos de uso anteriores, se crea el modelo de comportamiento. 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.

### 4.4. 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.





## Capítulo 5

# Diseño del Sistema

En esta sección se recoge la arquitectura general del sistema de información, la parametrización del software base (opcional), el diseño físico de datos, el diseño detallado de componentes software y el diseño detallado de la interfaz de usuario.

### 5.1. Arquitectura del Sistema

En esta sección se define la arquitectura general del sistema de información, especificando la infraestructura tecnológica necesaria para dar soporte al software y la estructura de los componentes que lo forman.

#### 5.1.1. Arquitectura Física

En este apartado, describimos los principales elementos hardware que forman la arquitectura física de nuestro sistema, recogiendo por un lado los componentes del entorno de producción y los componentes de cliente.

Se debe incluir un modelo de despliegue en el cual se describe cómo los elementos software son desplegados en los elementos hardware. También se incluyen las especificaciones y los requisitos del hardware (servidores, etc.), así como de los elementos software (sistemas operativos, servicios, aplicaciones, etc.) necesarios.

#### 5.1.2. Arquitectura Lógica

La arquitectura de diseño especifica la forma en que los artefactos software interactúan entre sí para lograr el comportamiento deseado en el sistema. En esta sección se muestra la comunicación entre el software base seleccionado, los componentes reutilizados y los componentes desarrollados para cumplir los requisitos de la aplicación. También, se recogen los servicios de sistemas externos con los que interactúa nuestro sistema. Se debe incluir un diagrama de componentes que muestre en un alto nivel de abstracción los artefactos que conforman el sistema.

Existen diferentes patrones o estilos arquitectónicos. 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. Habitualmente se tienen las siguientes capas:

**Capa de presentación (frontend)** Este grupo de artefactos software conforman la capa de presentación del sistema, incluyendo tanto los componentes de la vista como los elementos de control de la misma.

**Capa de negocio** Este grupo de artefactos software conforman la capa de negocio del sistema, incluyendo los elementos del modelo de dominio y los servicios (operaciones del sistema).

**Capa de persistencia** Este grupo de artefactos software conforman la capa de integración del sistema, incluyendo las clases de abstracción para el acceso a datos (BD o sistema de ficheros) o a sistemas heredados.

Es común que a la capa de negocio y de datos de los sistemas web, se denomine conjuntamente como backend o modelo de la aplicación.

Opcionalmente, podemos disponer de un conjunto de artefactos software que pueden ser usados por elementos de cualquiera de las capas del sistema y que fundamentalmente proporcionan servicios relacionados con requisitos no funcionales (calidad).

## 5.2. Parametrización del software base

En esta sección, 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.

## 5.3. 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.

## 5.4. Diseño detallado de Componentes

Para cada uno de los módulos funcionales del sistema debemos realizar un diagrama de secuencia, para definir la interacción existente entre las clases de objetos que permitan responder a eventos externos.

## 5.5. Diseño detallado de la Interfaz de Usuario

En esta sección se detallarán las interfaces entre el sistema y el usuario, incluyendo un prototipo de alta fidelidad con el diseño de la IU. Se definirá el comportamiento de las diferentes pantallas, indicando qué ocurre en los distintos componentes visuales de la interfaz cuando aparecen y qué acciones se disparan cuando el usuario trabaja con ellas.

## Capítulo 6

# 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.

### 6.1. 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.

### 6.2. 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.

### 6.3. 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.



## Capítulo 7

# 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.

### 7.1. 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.

### 7.2. Entorno de Pruebas

Incluir en este apartado los requisitos de los entornos hardware/software donde se ejecutarán las pruebas.

### 7.3. Roles

Describir en esa sección cuáles serán los perfiles y participantes necesarios para la ejecución de cada uno de los niveles de prueba.

### 7.4. Niveles de Pruebas

En esta sección se documentan los diferentes tipos de pruebas que se han llevado a cabo, ya sean manuales o automatizadas mediante algún software específico de pruebas.

#### 7.4.1. 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.

#### **7.4.2. Pruebas de Integración**

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

#### **7.4.3. 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.

##### **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.

##### **Pruebas No Funcionales**

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

#### **7.4.4. 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.

Parte III

Epílogo





En esta última parte quedarán recogidas las conclusiones y 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.



## Capítulo 8

# Manual de implantación y explotación

Las instrucciones de instalación y explotación del sistema se detallan a continuación. Este manual resulta de aplicación en aquellos casos en que se requiere realizar la instalación del software objetivo de este proyecto sobre algún entorno de servidor o cuando su instalación no sea trivial.

### 8.1. Introducción

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

### 8.2. Requisitos previos

Requisitos hardware y software para la correcta instalación del sistema.

### 8.3. Inventario de componentes

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

### 8.4. 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.

### 8.5. Pruebas de implantación

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

### 8.6. Procedimientos de operación y 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.



## Capítulo 9

# 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.

### 9.1. Introducción

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

### 9.2. Instalación

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

### 9.3. Uso del sistema

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



## Capítulo 10

# 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 y Pruebas del Sistema que resulten de interés.

### 10.1. Introducción

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

### 10.2. 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.

### 10.3. 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.

### 10.4. Instrucciones para construcción y despliegue

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.





# Capítulo 11

## 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.

### 11.1. Objetivos alcanzados

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

### 11.2. 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 que escribir un sencillo 'he trabajado mucho en este proyecto'.

### 11.3. 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.



# Bibliografía

- [CCOO, 2010] CCOO (2010). Tablas salariales 2010 IV Convenio Colectivo. <http://www.uca.es/sindicato/ccoo/documentos/tabla-salarial-pas-laboral-2010.pdf>.
- [Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75–105.



## Información sobre Licencia

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