

**ESCUELA SUPERIOR DE INFORMÁTICA - ESI (UCLM)**

**MEMORIA TRABAJO TEÓRICO 1**



**NUESTRA WIKI:** <https://github.com/ismapisma04/ISO25-ISO2-BC02/wiki>

**Asignatura:** Ingeniería del software II

**Grupo de titulación (25-26):** BC02



**Titulación:** Grado en Ingeniería Informática

**Fecha:** 29/10/2025

## ÍNDICE

0 - Análisis y supuestos.....	3
1- Planificación PUD.....	4
1.1 - Prioridades.....	5
1.2 - Tabla de esfuerzo.....	5
1.3 - Tabla de planificación.....	6
1.3.1 - Iteración 1 y 2.....	6
1.3.2 - Iteración 3 y 4.....	6
1.3.3 - Iteración 5 y 6.....	6
1.3.4 - Iteración 7 y 8.....	6
1.3.5 - Iteración 9 y 10.....	7
1.3.6 - Iteración 11 y 12.....	7
1.3.7 - Iteración 13 y 14.....	7
1.3.8 - Iteración 15 y 16.....	7
1.3.9 - Iteración 17 y 18.....	8
1.3.10 - Iteración 19 y 20.....	8
1.3.1 - Iteración 21 y 22.....	8
1.4 - Resultados finales.....	9
1.5 - Agenda.....	9
2 - Gestión de configuración.....	11
2.1 - Introducción.....	11
2.2 - Criterios para la identificación de los elementos de configuración.....	12
2.3 - Limitaciones y suposiciones.....	12
2.4 Responsabilidades y autoridades del plan.....	12
2.5 Organización del proyecto.....	12
2.6 Responsabilidades de la gestión de configuración.....	12
2.7 Políticas aplicables, directivas, y procedimientos.....	14
2.8 Actividades planificadas, agendas y recursos.....	14

2.9 Mantenimiento del plan de gestión de configuración.....	16
2.10 Control de versiones.....	16
3. Gestión de calidad.....	17
3.1 Fiabilidad.....	17
3.2 Portabilidad.....	17
3.3 Usabilidad.....	18

## 0. ANÁLISIS Y SUPUESTOS

Tras analizar lo que nos pide el problema, determinamos que se nos proporcionan los siguientes datos:

- Arquitectura: Cliente-Servidor
- Iteración inicial (fase de inicio) con coste fijo 1000€
- Ultima iteración N (tareas de cierre) con un coste fijo de 2000€
- Una semana de trabajo son 40h semanales
- Un requisito funcional se mapeará en un solo caso de uso (1:1)
- Un caso de uso se realizará en una y solo una iteración (1:1)

Además, completamos el análisis realizando nuestros propios supuestos para poder completar la planificación. Los supuestos son:

- La agenda se hará en días en lugar de semanas
- Duración de la iteración inicial de 6 días
- Duración de la iteración N de 10 días
- Salario de cada trabajador: 22€/h
- Los trabajadores no trabajaran en fin de semana ni festivo
- El proyecto inicia el 16/10/2025
- Para la arquitectura de cliente-servidor, usaremos la técnica de solapamiento en las iteraciones de elaboración y construcción, creando una tabla de esfuerzos solapada entre los dos
- Cada trabajador tendrá conocimientos de todas las áreas, pudiendo trabajar en cualquier sector

## 1.PLANIFICACIÓN PUD

Para comenzar la planificación, realizaremos una tabla en la que especificaremos la prioridad de cada uno de nuestros casos de uso:

CASO DE USO	DESCRIPCIÓN	PRIORIDAD
CDU 1	Consulta de material bibliográfico	1
CDU 2	Gestión de préstamos y devoluciones	1
CDU 3	Acceso a recursos electrónicos	2
CDU 4	Solicitud material bibliográfico	4
CDU 5	Solicitud del espacio físico o hardware específico	5
CDU 6	Unión a grupos de estudio y lectura	6
CDU 7	Gestión de bibliografía de asignaturas y proyectos de investigación	2
CDU 8	Gestión de grupos de estudio	5
CDU 9	Consulta de estadísticas de alumnos e investigadores del grupo	3
CDU 10	Acceso a recursos restringidos	3
CDU 11	Solicitud de estudios bibliométricos de I+D	5
CDU 12	Gestión de usuarios	1
CDU 13	Gestión de registros bibliográficos	1
CDU 14	Gestión de reservas	2
CDU 15	Gestión de inventario	3
CDU 16	Realización de informes	3
CDU 17	Configuración de parámetros del sistema	1
CDU 18	Gestión de roles	2
CDU 19	Auditoría de actividades	4
CDU 20	Integración de sistemas	2
CDU 21	Generación de estadísticas	3
CDU 22	Gestión de actualizaciones	5

### GRUPOS FUNCIONALES

En el CDU1 hemos unido las funcionalidades de Búsqueda y consulta de catálogo bibliográfico en general y la relevante por cada una de las asignaturas en las que el alumno está matriculado.

En el CUD2 hemos unido las funcionalidades de Reserva y renovación de préstamos de fondos bibliográficos y la de Gestión de historial de préstamos.

En el CDU5 hemos unido las funcionalidades de Petición de uso de espacios físicos y la de Petición de uso y préstamos de hardware.

En el CDU7 hemos unido las funcionalidades de Gestión de bibliografía recomendada para las asignaturas y la de Gestión de bibliografía y suscripciones para proyectos de investigación.

En el CUD8 hemos unido las funcionalidades de Acceso a grupos de estudio por asignaturas o por temas de interés y la de Acceso a clubs de lectura generales y especializada.

En el CDU14 hemos unido las funcionalidades de Gestión de préstamos y devoluciones y la de Control de reservas y renovaciones.

## 1.1 PRIORIDADES

En base a nuestra asignación, explicaremos la asignación y relevancia de prioridades a cada uno de nuestros casos de uso (del nivel 1 como más prioritario al 6 como menos prioritario):

*-Prioridad 1:* Son los casos de uso crítico, necesarios e imprescindibles para la primera versión del proyecto

*-Prioridad 2:* Casos de uso importantes y que deben seguir a los críticos

*-Prioridad 3:* Tienen importancia, pero se pueden realizar más adelante

*-Prioridad 4:* Casos de uso con una prioridad media con poco impacto en el funcionamiento

*-Prioridad 5:* Casos de uso poco relevantes que realizan funciones complementarias o de soporte

*-Prioridad 6:* Últimos casos de uso a realizar, que aportan mejoras o funciones sociales al proyecto

## 1.2. Tabla de esfuerzo

Tras realizar nuestros supuestos, hemos realizado la siguiente tabla de esfuerzo basándonos en la arquitectura cliente/servidor para cada uno de nuestros casos de uso, donde hemos decidido solapar al cliente y al servidor en una misma tabla, quedando así:

CDU	Prioridad	Requisitos	Análisis	Diseño	Implementación	Pruebas
CDU1	1	3	9	15	21	9
CDU2	1	6	9	15	21	9
CDU3	2	3	8	13	17	7

<b>CDU4</b>	4	2	5	8	11	4
<b>CDU5</b>	5	2	4	6	9	3
<b>CDU6</b>	6	3	3	5	7	3
<b>CDU7</b>	2	4	8	13	17	7
<b>CDU8</b>	5	4	4	6	9	3
<b>CDU9</b>	3	3	6	10	14	6
<b>CDU10</b>	3	2	6	10	14	6
<b>CDU11</b>	5	5	4	6	9	3
<b>CDU12</b>	1	5	9	15	21	9
<b>CDU13</b>	1	5	9	15	21	9
<b>CDU14</b>	2	5	8	13	17	7
<b>CDU15</b>	3	4	6	10	14	6
<b>CDU16</b>	3	6	9	10	14	6
<b>CDU17</b>	1	5	8	15	21	9
<b>CDU18</b>	2	4	5	13	17	7
<b>CDU19</b>	4	6	8	8	11	4
<b>CDU20</b>	2	6	6	13	17	7
<b>CDU21</b>	3	4	4	10	14	6
<b>CDU22</b>	5	5	4	6	9	3

### 1.3. Tabla de planificación

#### 1.3.1 Iteración 1 y 2

	ITERACION 1					ITERACION 2				
RRHH	R	A	D	I	P	R	A	D	I	P
P1	1	2	2	3	0	1	2	2	3	1
P2	1	1	2	3	0	1	2	2	3	1
P3	1	1	2	3	0	1	2	2	3	1
P4	1	1	2	3	0	1	1	3	4	2
P5	1	1	2	3	1	1	1	3	4	2
P6	0	1	2	2	1	1	1	3	4	2
TH	43					60				
TC	946					1320				
AGENDA	6					8				

#### 1.3.2 Iteración 3 y 4

ITERACION 3					ITERACION 4				
R	A	D	I	P	R	A	D	I	P
1	2	4	3	2	1	1	8	3	0
1	2	4	3	1	1	1	8	3	0
1	2	4	3	1	1	1	8	3	1
1	2	4	3	1	1	1	8	3	1
1	1	3	3	1	0	2	7	4	1
1	1	3	3	1	0	2	7	4	1
63					82				
1386					1804				
8					11				

### 1.3.3 Iteración 5 y 6

ITERACION 5					ITERACION 6				
R	A	D	I	P	R	A	D	I	P
1	1	2	2	2	1	1	2	3	1
1	1	2	3	2	1	1	2	3	1
1	1	2	3	2	1	1	2	3	0
2	1	2	3	2	0	1	2	3	0
2	1	2	3	1	0	2	2	3	0
2	0	3	3	1	0	2	3	2	0
54					43				
1188					946				
7					6				

### 1.3.4 Iteración 7 y 8

ITERACION 7					ITERACION 8				
R	A	D	I	P	R	A	D	I	P
1	1	2	3	1	2	2	1	1	0
1	1	2	3	1	1	1	2	1	0
1	1	2	3	1	1	1	1	1	1
1	1	2	2	2	1	1	1	0	1
0	1	1	2	2	1	0	1	1	1
0	2	2	2	2	0	0	1	1	1
46					22				
1012					484				
6					3				

### 1.3.5 Iteración 9 y 10

	ITERACION 9					ITERACION 10				
RRHH	R	A	D	I	P	R	A	D	I	P
P1	1	1	2	0	0	1	1	2	0	0
P2	1	1	2	2	1	1	1	2	0	3
P3	1	1	2	3	1	0	1	2	1	1
P4	0	1	2	3	0	0	1	2	3	1
P5	0	0	2	3	1	0	1	2	3	1
P6	0	0	0	3	3	0	1	2	3	1
TH	36					37				
TC	792					814				
AGENDA	5					5				

### 1.3.6 Iteración 11 y 12

ITERACION 11					ITERACION 12				
R	A	D	I	P	R	A	D	I	P
1	0	2	3	1	1	1	2	2	1
1	1	2	2	1	1	1	2	2	2
1	1	2	2	1	0	1	2	2	1
0	1	2	2	1	0	1	2	2	1
0	1	1	2	1	0	1	2	1	1
0	0	1	3	1	0	1	2	1	1
37					33				
814					726				
5					5				

### 1.3.7 Iteración 13 y 14

ITERACION 13					ITERACION 14				
R	A	D	I	P	R	A	D	I	P
1	1	1	2	2	1	1	2	2	1
1	1	1	2	1	1	1	2	2	1
1	1	1	2	1	1	1	2	3	1
1	1	1	3	1	1	1	2	2	1
1	2	1	2	1	0	1	2	2	1
1	1	1	2	1	0	1	2	2	1
39					41				
858					902				
5					6				

### 1.3.8 Iteración 15 y 16

ITERACION 15					ITERACION 16				
R	A	D	I	P	R	A	D	I	P
0	1	2	2	1	0	1	1	1	1
1	1	2	2	0	0	1	1	1	1
1	1	2	2	1	0	1	1	1	1
1	1	2	2	1	0	1	1	2	0
1	1	2	3	1	1	0	1	2	0
1	1	2	2	1	1	0	1	2	0
41					24				
902					528				
6					3				

### 1.3.9 Iteración 17 y 18

ITERACION 17						ITERACION 18				
RRHH	R	A	D	I	P	R	A	D	I	P
P1	1	1	1	1	0	1	1	1	1	1
P2	1	1	1	1	0	1	1	1	1	1
P3	1	1	1	1	1	1	1	1	1	1
P4	1	1	1	1	1	1	0	1	1	1
P5	1	1	1	1	1	1	0	2	1	1
P6	1	0	2	0	1	1	0	2	2	1
TH	27					24				
TC	594					528				
AGENDA	4					3				

### 1.3.10 Iteración 19 y 20



ITERACION 19					ITERACION 20				
R	A	D	I	P	R	A	D	I	P
1	1	1	2	0	1	1	2	0	1
1	1	1	2	1	1	1	2	1	1
1	1	1	1	1	1	0	2	1	1
1	1	2	1	1	0	0	1	1	0
0	0	2	1	1	0	1	1	1	1
0	1	2	1	0	0	1	1	1	1
30					26				
660					572				
4					4				

### 1.3.11 Iteración 21 y 22

ITERACION 21					ITERACION 22				
R	A	D	I	P	R	A	D	I	P
1	1	0	2	1	1	0	1	2	1
1	1	1	2	1	1	1	1	2	0
1	1	1	2	1	1	1	0	2	0
1	2	1	2	1	1	1	0	1	0
1	2	1	2	1	0	1	1	1	0
0	2	1	2	1	0	1	1	1	1
37					24				
814					528				
5					3				

### 1.4 Resultados finales

Tras realizar todos los cálculos, obtenemos los siguientes resultados de horas, costes y días, necesarios para gestionar nuestro proyecto:

FASE	INICIO	ELABORACION										CONSTRUCCIÓN												TRANSICIÓN
ITERACION	It0	It1	It2	It3	It4	It5	It6	It7	It8	It9	It10	It11	It12	It13	It14	It15	It16	It17	It18	It19	It20	It21	It22	It23
COSTE	880	946	1320	1386	1804	1188	946	1012	484	792	814	814	726	858	902	902	528	594	528	660	572	814	528	1760
HORAS	40	43	60	63	82	54	43	46	22	36	37	37	33	39	41	41	24	27	24	30	26	37	24	80
AGENDA	6	6	8	8	11	7	6	6	3	5	5	5	5	5	6	6	3	4	3	4	4	5	3	10

TH	TC
137	3014
154	3388
151	3322
148	3256
144	3168
141	3102
997	
22118	
134	

### 1.5 Agenda

Teniendo en cuenta el supuesto de que el día de inicio del proyecto será el 16 de octubre, obtenemos la siguiente agenda resultante:

OCTOBER 2025						
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	It1 16	It1 17	18	19
It1 20	It1 21	It1 22	It1 23	It2 24	25	26
It2 27	It2 28	29	30	31		

NOVEMBER 2025						
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
					1	2
It2 3	It2 4	It3 5	It3 6	It3 7	8	9
It3 10	It3 11	It3 12	It3 13	It3 14	15	16
It4 17	It4 18	It4 19	It4 20	It4 21	22	23
It4 24	It4 25	It4 26	It4 27	It4 28	29	30

DECEMBER 2025						
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
It4 1	It5 2	It5 3	It5 4	It5 5	6	7
Festivo 8	It5 9	It5 10	It5 11	It6 12	13	14
It6 15	It6 16	It6 17	It6 18	It6 19	20	21
It7 22	It7 23	It7 24	Festivo 25	It7 26	27	28
It7 29	It7 30	It8 31				

JANUARY 2026						
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
			Festivo 1	It8 2	3	4
It8 5	Festivo 6	It9 7	It9 8	It9 9	10	11
It9 12	It9 13	It10 14	It10 15	It10 16	17	18
It10 19	It10 20	It11 21	It11 22	It11 23	24	25
It11 26	It11 27	It12 28	It12 29	It12 30	31	

FEBRUARY 2026						
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
						1
It12 2	It12 3	It13 4	It13 5	It13 6	7	8
It13 9	It13 10	It14 11	It14 12	It14 13	14	15
It14 16	It14 17	It14 18	It15 19	It15 20	21	22
It15 23	It15 24	It15 25	It15 26	It16 27	28	

MARCH 2026						
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
						1
It16 2	It16 3	It17 4	It17 5	It17 6	7	8
It14 9	It18 10	It18 11	It18 12	It19 13	14	15
It19 16	It19 17	It19 18	It20 19	It20 20	21	22
It20 23	It20 24	It21 25	It21 26	It21 27	28	29
It21 30	It21 31					

APRIL 2026						
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
		It22 1	Festivo 2	Festivo 3	4	5
Festivo 6	It22 7	It22 8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

## 2 - GESTIÓN DE CONFIGURACIÓN

### 2.1 - INTRODUCCIÓN

- Propósitos: crear el plan de configuración de la gestión de la biblioteca universitaria, utilizando para ello una arquitectura cliente-servidor y realizando un control de versiones, controles de cambio y optimización del sistema.
- Alcance del plan: Este plan se aplica a todas las fases del ciclo de vida de nuestro proyecto, abarcando desde la toma de requisitos hasta la entrega y mantenimiento. En ella tendremos las configuraciones de software, documentación, versiones y bases de datos
- Relación con la organización y otros proyectos: Se usarán políticas y metodologías, así como estándares de buenas prácticas como el IEEE 828:2012 o el ISO/IEC

- Términos clave: Base line, branch, commit, pull, elemento de configuración, tag, merge, release, issue
- Referencias: Términos como línea base, auditorías, semantic versioning y estándares como IEE 828:2012

## **2.2 – CRITERIOS PARA LA IDENTIFICACIÓN DE LOS ELEMENTOS DE CONFIGURACIÓN**

Aplicamos el Configuration Management (CM) para proteger la integridad del producto. Tendremos documentos (requisitos, análisis, diseño...), bases de datos, pruebas, configuraciones del servidor (con scripts de despliegues) y entregables.

## **2.3 – LIMITACIONES Y SUPOSICIONES**

Tendremos un total de 6 trabajadores, todos con capacidades de realizar cualquier disciplina. Se tiene en cuenta una arquitectura cliente-servidor con solapamiento entre ambos. Se utilizarán herramientas gratuitas pues se cuenta con limitaciones económicas para la realización.

Se trabajará en una única rama para el desarrollo y la producción. El presupuesto para la iteración inicial sera de 1000€ y la iteración final de 2000€

## **2.4- RESPONSABILIDADES Y AUTORIDADES DEL PLAN**

Para realizar la aplicación del Plan de Gestión de Configuración vamos a requerir una serie de roles definidos, dando así consigo responsabilidades y niveles de autoridad.

Cada miembro conoce las funciones dentro del proceso, garantizando calidad, un control de cambios y trazabilidad.

Todo la Gestión de Configuración va a estar supervisado por el coordinador del equipo (mediante el Configuration Management Authority) encargado de supervisar el cumplimiento de políticas de control de versiones, resolver conflictos de control de cambios y validar informes de configuración.

Todo el equipo estará implicado sobre todo en seguir el plan acordado e implementar correctamente las politicas definidas previamente por la CMA e ir gestionando las versiones, las ramas, auditorías de configuración, etc.

Todo esto se hace mediante Configuration Manager (CM)

## **2.5 – ORGANIZACIÓN DEL PROYECTO**

El equipo trabaja con capacidades multidisciplinarias, de acuerdo con nuestros supuestos, siguiendo la planificación propuesta de trabajo teniendo en cuenta las horas y los costes necesarios. Se realizará además un control de versiones para la realización de las iteraciones.

Los roles coordinador quedan:

- Coordinador: Jesus Márquez Duque
- Responsables de coordinación: Ainhoa Sánchez Esteban y Carlos Salgado Rojas
- Desarrolladores: Manuel Porrero Sierra y David Ruíz Sánchez
- Revisor de calidad: Ismael Serrano Usero

## **2.6 – RESPONSABILIDADES DE LA GESTIÓN DE CONFIGURACIÓN**

Se deberá realizar un control de versiones mediante ramas (main, develop, feature, release y hotfix). Se usaran tags para marcar las versiones de los elementos de configuracion, auditorias para la revision antes de la liberación y backups para la realizacion de copias periodicas.

### **2.6.1 – CREACIÓN DE RAMAS**

Hemos hecho un conjunto de ramas feature para cada caso de uso que hemos definido, por lo que nos quedaria de esta manera:

#### **Estudiantes**

feature/server/CDU1-consulta-catalogo

feature/client/CDU1-ui-consulta

feature/server/CDU2-prestamos

feature/client/CDU2-ui-prestamos

feature/server/CDU3-recursos-electronicos

feature/client/CDU3-ui-recursos-electronicos

feature/server/CDU4-solicitud-adquisiciones

feature/client/CDU4-ui-solicitud-adquisiciones

feature/server/CDU5-reserva-espacios-hardware

feature/client/CDU5-ui-reserva-espacios-hardware

feature/server/CDU6-grupos-estudio

feature/client/CDU6-ui-grupos-estudio

#### **Docentes e Investigadores**

feature/server/CDU7-bibliografia-docencia-investigacion

feature/client/CDU7-ui-bibliografia

feature/server/CDU8-gestion-grupos-estudio

feature/client/CDU8-ui-gestion-grupos

feature/server/CDU9-estadisticas-uso

feature/client/CDU9-ui-estadisticas

feature/server/CDU10-acceso-recursos-restringidos

feature/client/CDU10-ui-recursos-restringidos

feature/server/CDU11-estudios-bibliometricos

feature/client/CDU11-ui-bibliometricos

### **Personal bibliotecario**

feature/server/CDU12-gestion-usuarios

feature/client/CDU12-ui-usuarios

feature/server/CDU13-registros-bibliograficos

feature/client/CDU13-ui-registros

feature/server/CDU14-gestion-reservas

feature/client/CDU14-ui-reservas

feature/server/CDU15-inventario-materiales

feature/client/CDU15-ui-inventario

feature/server/CDU16-informes-operativos

feature/client/CDU16-ui-informes

### **Administradores del sistema**

feature/server/CDU17-configuracion-sistema

feature/client/CDU17-ui-configuracion

feature/server/CDU18-gestion-roles-permisos

feature/client/CDU18-ui-roles

feature/server/CDU19-auditoria-actividades

feature/client/CDU19-ui-auditoria

feature/server/CDU20-integracion-sistemas-SSO

# (si hay consola admin)

feature/client/CDU20-ui-integracion

feature/server/CDU21-estadisticas-globales

feature/client/CDU21-ui-estadisticas-globales

feature/server/CDU22-gestion-actualizaciones

feature/client/CDU22-ui-actualizaciones

### **Cross-cutting / soporte (útiles para el proyecto)**

feature/cross/notificaciones-vencimientos

feature/cross/api-gateway-o-bff

feature/cross/observabilidad-logs-metrics

## **2.7 - POLITICAS APLICABLES, DIRECTIVAS Y PROCEDIMIENTOS.**

- Semantic versioning 2.0 (formato m.m.p):
  - MAJOR: Cambios que rompen la compatibilidad
  - MINOR: Cambios que no rompen la compatibilidad de las dependencias
  - PATCH: Arreglos que no cambian la interfaz

- Commits
- Revisiones de código
- Auditorias de cambios
- Liberaciones: Solo versiones válidas probadas

## 2.8 – ACTIVIDADES PLANIFICADAS, AGENDAS Y RECURSOS

Agendas del CM: El proyecto inicia con la *It0*, dedicada a la planificación e identificación de elementos de configuración, continuando con el control de versiones y auditorías en iteraciones posteriores hasta la liberación final.

Recursos del CM: Usaremos Git como repositorio, para el control de versiones, documentación y organización del proyecto, además de maven para las compilaciones y visual paradigm para el modelado UML

## 2.9 – MANTENIMIENTO DEL PLAN DEL CONFIGURATION MANAGEMENT (CM)

Tendremos un documento de configuración que habrá que revisar en cada iteración, registrando cambios en una petición de cambios evaluada por el CM, para poder adaptar el proyecto a necesidades cambiantes.

El plan de configuración permitirá la integración, mostrando las versiones liberadas y la trazabilidad de componentes.

Quedaran documentadas todas las modificaciones y reuniones en la wiki para garantizar el avance del desarrollo

## 2.10 – CONTROL DE VERSIONES

Realizaremos un informe de control de versiones para controlar la realización de todos los componentes de nuestro sistema, comprobando que quedan perfectamente identificados y realizados. Para su realización usaremos el esquema de versionado de Semantic Versioning.

### 2.10.1 - VERSIONADO

ITERACIONES	VERSIONES	DESCRIPCIÓN
it0 (Inicio)	0.1.0	Inicio del proyecto, especificación de requisitos y modelo del negocio.
it1-it5 (Elaboración)	0.2.0	Implementación de los casos de uso principales con prioridad 1 relacionados con la gestión del sistema. Análisis y diseño inicial.
it6-it10 (Elaboración)	0.3.0	Ampliación del modelo con casos de uso de prioridad 2. Refinamiento de la arquitectura y pruebas iniciales
it11-it15 (Construcción)	0.4.0	Desarrollo de la aplicación cliente-servidor. Integración de componentes y pruebas unitarias.
it16-it17(Construcción)	0.5.0	Optimización del código, mejora de la interfaz y preparación para pruebas de aceptación.
it18-it22(Construcción)	0.6.0	Finalización del sistema y validación de funcionalidades completas.
it23(Transición)	1.0.0	Pruebas finales, despliegue del sistema y entrega al cliente.

El proyecto utiliza un sistema de versionado incremental basado en el esquema **MAJOR.MINOR.PATCH**, donde solo se modifica el número **MINOR**, ya que cada bloque de iteraciones completa un conjunto significativo de funcionalidades. De esta manera, cada grupo de iteraciones representa una versión más avanzada y completa del sistema.

La versión **0.1.0** corresponde a *it0*, donde se realiza el inicio del proyecto, el análisis inicial y la especificación de requisitos. Aunque aún no existe implementación, esta fase marca la primera entrega formal del trabajo.



Las iteraciones **it1 a it5** forman la versión **0.2.0**, ya que en ellas se desarrollan e implementan los casos de uso de prioridad alta relacionados con la gestión principal del sistema. Este bloque representa un avance notable respecto a la fase inicial.

A continuación, las iteraciones **it6 a it10** generan la versión **0.3.0**. En este periodo se amplían las funcionalidades con los casos de uso de prioridad media, se refina la arquitectura y se consolidan aspectos fundamentales del diseño.

Las iteraciones **it11 a it15** corresponden a la versión **0.4.0**, porque aquí comienza la fase de construcción del sistema. Se desarrolla la arquitectura cliente-servidor, se integran componentes y se realizan pruebas unitarias, lo que supone un incremento significativo de funcionalidad.

En las iteraciones **it16 y it17**, que forman la versión **0.5.0**, se llevan a cabo mejoras del rendimiento, optimización del código y ajustes en la interfaz. Estas mejoras hacen que el sistema avance hacia un estado más estable y cercano al producto final.

Posteriormente, las iteraciones **it18 a it21** producen la versión **0.6.0**, ya que en este bloque se completan las funcionalidades restantes, se realizan validaciones y se prepara el sistema para su fase final.

Finalmente, las iteraciones **it22 y it23** conducen a la versión **1.0.0**, que representa la entrega definitiva del proyecto. En esta fase se ejecutan las pruebas finales, se realizan los últimos ajustes y se lleva a cabo el despliegue del sistema listo para su uso.

### 3 - GESTIÓN DE CALIDAD

Para gestionar la calidad de nuestro proyecto tenemos que elegir las características de calidad que se adecuarían a nuestro proyecto, teniendo en cuenta la planificación, pues puede afectar al tiempo y costes de nuestro proyecto. Entre las características de calidad disponibles, hemos elegido:

- **Fiabilidad:** Consiste en tener un proyecto con madurez, disponibilidad, tolerante a fallos y capacidad de recuperación
- **Portabilidad:** Consiste en tener un proyecto con adaptabilidad y capacidad de ser instalado y reemplazado
- **Usabilidad:** Consiste en tener un proyecto completo con el fin de ser usado por usuarios específicos para así conseguir una eficacia, eficiencia y satisfacción en un contexto de uso.

#### 3.1- FIABILIDAD

Para que sea fiable, el proyecto ha de:

- Tener mayor madurez, es decir, que el proyecto esté preparado para su uso. Para ello, nuestro sistema tiene que hacer correctamente las funciones tanto del cliente, como del servidor
- Tener disponibilidad, es decir, que el proyecto sea operativo siempre. Para ello, nuestro sistema puede tener varios servidores para que, si uno falla, otro pueda sustituirse en su lugar

- Tener tolerancia frente a fallos, es decir, que funcione correctamente, a pesar de tener fallos. Para ello, en nuestro proyecto se puede crear una captura de errores del sistema para que puedan ser arreglados
- Tener capacidad de recuperación, es decir, que pueda ser recuperado de forma sencilla cada vez que lo perdamos. Para ello, nuestro proyecto puede tener un control frente a pérdidas de datos o de partes del sistema, de tal forma que sean recuperados

Esta característica de calidad afecta a las siguientes fases de planificación de proyecto:

- Implementación, ya que hay que implementar el control de recuperación y las capturas de errores, lo que incrementa al menos 1 hora de tiempo en esta fase por caso de uso, y al menos 22 euros en el coste por cada caso de uso
- Pruebas, ya que hay que probar si el programa está operativo frente a fallos del servidor, y si el programa está preparado para ser usado. Esta fase incrementará 1 hora de tiempo como máximo, para cada caso de uso, e incrementará al menos 22 euros en cuanto al coste por cada uso

### **3.2- PORTABILIDAD**

Para que sea portable, el proyecto ha de:

- Adaptarse al cliente que use este sistema. Para ello, hay que probar en distintos dispositivos el correcto funcionamiento del sistema
- Tener capacidad de ser instalado, de tal forma que el proyecto pueda instalarse en cualquier sistema operativo. Para ello, se pueden realizar pruebas de instalación del sistema en cada uno de los sistemas operativos
- Tener capacidad de ser reemplazado, es decir, que el sistema pueda ser reemplazado cada vez que se mejore o se cambie en él. Para ello, habrá que implementar un sistema de cambios para que no haya errores al reemplazarse

Esta característica de calidad afecta a las siguientes fases de planificación de proyecto:

Implementación, ya que hay que implementar un sistema de cambios para que pueda ser reemplazado correctamente, lo que incrementa 1 hora de tiempo en esta fase como máximo, por caso de uso

- Pruebas, ya que hay que probar si el programa puede ser usado por los clientes, y probar si es capaz de ser instalado en cada sistema operativo. Esta fase incrementará 2 horas de tiempo como máximo por caso de uso, y 44 euros como máximo en el coste por caso de uso

### **3.3- USABILIDAD**

Un proyecto para que sea usable, el proyecto ha de ser:

- Debe ser reconocible, es decir que el usuario pueda identificar fácilmente para que sirve el sistema propuesto y como debe usarlo. Para que este sistema sea reconocible se debería usar una interfaz clara, coherente y sobre todo que sea entendible para que así el usuario saque las funciones principales sin problema alguno.

- Debe proteger frente a errores del usuario, es decir, que el sistema evite algunos errores comunes o permita corregirlos de una manera fácil y entendible. Para ello, el proyecto puede incluir mensajes de confirmación antes de acciones críticas como el salir del programa cuando estás editando cualquier cosa.
- Ser operable, que el usuario pueda utilizar el sistema sin dificultad y con eficiencia, respondiendo correctamente las acciones del usuario y garantizando una navegación fluida.

Esta característica afecta a algunas fases de planificación del proyecto y suelen ser las siguientes:

- Diseño, ya que hay que planificar la estructura visual y la disposición de los elementos en la interfaz, lo que incrementa un total 2 horas de trabajo por caso de uso. Todo esto tendrá un coste de 44€ por caso de uso.
- Implementación, ya que hay que desarrollar los elementos visuales, mensajes de ayuda y validaciones necesarias. Esto llevará una hora y tendrá un coste de 22€.
- Pruebas, ya que hay que comprobar que la interfaz es intuitiva, que las validaciones funcionan correctamente y que el sistema es accesible para todos los usuarios. Para realizar dichas pruebas contamos con 2 horas de trabajo para poder asegurarnos de que nuestro sistema estará bien preparado. Al realiza estas pruebas nos llevará a un coste de 44€.