



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

TRABAJO TEÓRICO

María Victoria Alcázar Clemente
Alejandro Paniagua Rodríguez
Jesús Calzado González
Enrique Rubio Gonzalo
Clara Sacedón Ortega
Diego Alba Ruiz

Asignatura: Ingeniería del Software II

Grupo de Titulación (21/22): BC.02

Titulación: Grado en Ingeniería Informática

Fecha: 26-11-2021

ÍNDICE

FICHA DEL TRABAJO.....	2
ÁNÁLISIS DE REQUISITOS.....	3
CASOS DE USO	4
PROCESO UNIFICADO DE DESARROLLO	5
<i>PLANIFICACIÓN SECUENCIAL</i>	<i>6</i>
<i>AGENDA</i>	<i>7</i>
<i>PLANIFICACIÓN PARALELA</i>	<i>8</i>
<i>AGENDA</i>	<i>9</i>
CONTROL DE CALIDAD.....	10
Adecuación funcional	10
Eficiencia de rendimiento	10
Seguridad.....	10
Capacidad de uso.....	10
GESTIÓN DE LA CONFIGURACIÓN.....	11

FICHA DEL TRABAJO

Código:		Fecha:	26-11-2021
Título:	TRABAJO TEÓRICO		

Equipo N°: BC.02		
Apellidos y Nombre	Firma	Puntos
MARÍA VICTORIA ALCÁZAR CLEMENTE	20616826-V	16.67
ALEJANDRO PANIAGUA RODRÍGUEZ	05739109-B	16.67
JESÚS CALZADO GONZÁLEZ	05739349-K	16.67
ENRIQUE RUBIO GONZALO	05720535-K	16.67
CLARA SACEDÓN ORTEGA	02598786-Q	16.67
DIEGO ALBA RUIZ	06286705-T	16.67

ANÁLISIS DE REQUISITOS

Los requisitos del trabajo son los siguientes:

- 1) Crear una nueva rama de seguros
- 2) Dar de baja una nueva rama de seguros
- 3) Consultar el listado de ramas
- 4) Añadir un nuevo producto a una de las ramas
- 5) Modificar un producto existente
- 6) Dar de baja un producto, haciendo que deje de esta vigente.
- 7) Consultar el listado de productos por rama
- 8) Crear una nueva póliza para un tipo de producto
- 9) Modificar una póliza para un tipo de seguro
- 10) Dar de baja una póliza
- 11) Consultar todas las pólizas por producto en un rango de fechas determinado
- 12) Consultar el número de producto que tiene contratado un cliente.
- 13) Gestionar políticas de descuentos en función de los productos contratados.
- 14) Gestionar los pagos de las pólizas
- 15) Gestionar los impagos de las pólizas.

Hemos definido cuatro niveles prioridades a trabajar de menor a mayor (1-más importante, 4-menos importante):

Req#	Prioridad
RF.1	2
RF.2	2
RF.3	3
RF.4	2
FR.5	3
FR.6	2
FR.7	3
FR.8	2
FR.9	3
FR.10	2
FR.11	3
FR.12	3
FR.13	4
FR.14	1
FR.15	1

NIVEL DE PRIORIDAD 1: Todo lo relacionado con movimientos de dinero porque es uno de los temas que más le suelen preocupar al cliente.

NIVEL DE PRIORIDAD 2: Funciones esenciales del proyecto como añadir y eliminar, pero no tan importante como un buen sistema de pago porque un error en dicho sistema puede causar importantes pérdidas.

NIVEL DE PRIORIDAD 3: Funciones similares a las anteriores, pero que, según nuestro criterio, deben estar en este nivel porque podríamos prescindir de ellas, pasan a un segundo plano respecto a añadir y eliminar.

NIVEL DE PRIORIDAD 4: Resto de requisitos a implementar.

CASOS DE USO

Tras realizar varias lecturas del problema y analizar los requisitos funcionales hemos establecido que la relación entre requisitos funcionales y casos de uso sea 1:1, es decir, cada caso de uso representa un requisito funcional.



Se nos ha informado de que existen tres tipos de actor en el sistema, y que cada uno realiza tareas independientes y que no hay ningún actor que realice diversas tareas que puedan realizar otros actores, por eso no existe relación de generalización o especialización entre los actores.

En cuanto a los casos de uso, aunque pueda parecer que existe relación entre ellos, no están contenidos o forman parte entre ellos, por eso tampoco declaramos ninguna extensión o inclusión.

PROCESO UNIFICADO DE DESARROLLO

Se supone que la jornada laboral es de 40 horas semanales (8h/diarias * 5 días a la semana) y todas las personas rinden al 100%. También utilizamos una aproximación de un caso de uso por requisito (1:1).

Además, suponemos que todo el personal tiene conocimientos básicos a cerca de los requisitos. Por eso, para esa tarea hemos seleccionado como encargado al tester porque con él obtendremos un presupuesto más reducido, también estará disponible personal cualificado en esas áreas para los siguientes trabajos.

Req#	CDU	Prioridad	Requisitos	Análisis	Diseño	Implementación	Testing
RF.1	CDU.1.	2	4	7	25	55	9
RF.2	CDU.2.	2	3	6	14	40	7
RF.3	CDU.3.	3	2	5	13	25	8
RF.4	CDU.4.	2	3	6	15	40	7
RF.5	CDU.5.	3	2	4	10	20	5
RF.6	CDU.6.	2	3	6	14	40	7
RF.7	CDU.7.	3	2	5	13	25	8
RF.8	CDU.1.	2	4	7	25	55	9
RF.9	CDU.9.	3	2	4	10	20	5
RF.10	CDU.10.	2	3	6	14	40	7
RF.11	CDU.11.	3	2	5	13	25	8
RF.12	CDU.12.	3	2	5	13	25	8
RF.13	CDU.13.	4	5	10	25	60	5
RF.14	CDU.14.	1	5	10	25	70	10
RF.15	CDU.15.	1	5	10	25	70	10

Tabla 1: Casos de uso, prioridad y esfuerzo estimado por cada tarea

PLANIFICACIÓN SECUENCIAL

FASE	INICIO	ELABORACIÓN
ITERACIÓN	It ₀	It ₁
COSTE	500	C1=4800
DURACIÓN	5	D1=15
AGENDA	1 (01NOV - 05NOV)	A1 (8NOV-26NOV)

CONSTRUCCIÓN			
It ₂	It ₃	It ₄	It ₅
C2=4800	C3=4040	C4=2780	C5=2840
D2=15	D3=13	D4=9	D5=9
A2 (29NOV-17DIC)	A3 (20DIC - 05ENE)	A4 (6ENE - 18ENE)	A5 (19ENE - 31ENE)

It ₆	It ₇	It ₈	It ₉
C6=2780	C7=4040	C8=2780	C9=2160
D6=9	D7=13	D8=9	D9=7
A6 (1FEB - 11FEB)	A7 (14FEB - 02MAR)	A8 (03MAR - 15MAR)	A9 (16MAR - 24MAR)

It ₁₀	It ₁₁	It ₁₂	It ₁₃
C10=1700	C11=2180	C12=1700	C13=2180
D10=6	D11=7	D12=6	D13=7
A10 (25MAR - 01ABR)	A11 (04ABR - 12ABR)	A12 (13ABR - 20ABR)	A13 (21ABR - 29ABR)

		Transición	Total
It ₁₄	It ₁₅	It _f	
C14=2180	C15=4400	2500	48360 euros
D14=7	D15=14	15	166 JORNADAS
A14 (02MAY - 10MAY)	A15 (11MAY - 30MAY)	3 (31MAY - 20JUN)	34 SEMANAS

Tabla 2: Tabla con planificación secuencial

	It1				It2				It3				It4				It5				It6				It7				It8				It9				It10				It11				It12				It13				It14				It15			
RRHH	R	A	D	I	T	R	A	D	I	T	R	A	D	I	T	R	A	D	I	T	R	A	D	I	T	R	A	D	I	T	R	A	D	I	T	R	A	D	I	T	R	A	D	I	T	R	A	D	I	T										
R1	5									9						3								7						2			8							8																				
R2			10								3						7																																											
R3				5										7										4						8					2																									
R4					10						3																																																	
R5						5									7																																													
R6	10						7					6					7								3						5				5																									
R7				10								6														6																																		
R8		15											14														14																																	
R9					15																																																							
R10								25																																																				
R11			70																																																									
R12						70																																																						
R13									25																																																			
R14											40																																																	
R15													40																																															
R16															40																																													
R17																		55																																										
R18																																																												
R19																																																												
R20																																																												
TOTAL HORAS	110					110								71						70																																								
TOTAL COSTE	4800					4800								4040						2840																																								

Tabla 3: Tabla con iteraciones

AGENDA

Suponemos que empezamos 1 de noviembre de 2021. (COLOR POR ITERACIÓN)

MES	SEMANA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
11	1	1	2	3	4	5
	2	8	9	10	11	12
	3	15	16	17	18	19
	4	22	23	24	25	26
12	5	29	30	1	2	3
	6	6	7	8	9	10
	7	13	14	15	16	17
	8	20	21	22	23	24
01	9	27	28	29	30	31
	10	3	4	5	6	7
	11	10	11	12	13	14
	12	17	18	19	20	21
02	13	24	25	26	27	28
	14	31	1	2	3	4
	15	7	8	9	10	11
	16	14	15	16	17	18
03	17	21	22	23	24	25
	18	28	1	2	3	4
	19	7	8	9	10	11
	20	14	15	16	17	18
04	21	21	22	23	24	25
	22	28	29	30	31	1
	23	4	5	6	7	8
	24	11	12	13	14	15
05	25	18	19	20	21	22
	26	25	26	27	28	29
	27	2	3	4	5	6
	28	9	10	11	12	13
06	29	16	17	18	19	20
	30	23	24	25	26	27
	31	30	31	1	2	3
	32	6	7	8	9	10
	33	13	14	15	16	17
	34	20	21	22	23	24

¡NO TENEMOS EN CUENTA LOS FESTIVOS NACIONALES! Si los tuviésemos, se retrasaría y acabaría el 27 de junio de 2022, en vez del 20 de junio de 2022.

Esta planificación (secuencial) no es la más adecuada porque hay muchos huecos y, por lo tanto, hay personas que en determinados momentos se quedan paradas.

PLANIFICACIÓN PARALELA

FASE	INICIO	ELABORACIÓN
ITERACIÓN	It_0	It_1
COSTE	500	C1=4800
DURACIÓN	5	D1=3
AGENDA	1 (01NOV-05NOV)	A1 (08NOV-10NOV)

CONSTRUCCIÓN			
It ₂	It ₃	It ₄	It ₅
C2=4800	C3=4040	C4=2780	C5=2840
D2=3	D3=3	D4=2	D5=2
A2 (11NOV-15NOV)	A3 (16NOV- 18NOV)	A4 (19 NOV - 22NOV)	A5 (23NOV – 24NOV)

lt ₆	lt ₇	lt ₈	lt ₉
C6=2780	C7=4040	C8=2780	C9=2160
D6=2	D7=3	D8=2	D9=2
A6 (25NOV – 26NOV)	A7 (29NOV – 01DIC)	A8 (02DIC – 03DIC)	A9 (06DIC – 07DIC)

It ₁₀	It ₁₁	It ₁₂	It ₁₃
C10=1700	C11=2180	C12=1700	C13=2180
D10=2	D11=2	D12=2	D13=2
A10 (08DIC – 09DIC)	A11 (10DIC – 13DIC)	A12 (14DIC – 15DIC)	A13 (16DIC – 17DIC)

		Transición	Total
It ₁₄	It ₁₅	It _f	
C14=2180	C15=4400	2500	48360 euros
D14=2	D15=3	15	166 JORNADAS
A14 (20DIC – 21DIC)	A15 (22DIC – 24DIC)	3 (27DIC – 14ENE)	34 SEMANAS

Tabla 2: Tabla con planificación secuencial

[illegible]

Tabla 3: Tabla con iteraciones

AGENDA

Suponemos que empezamos 1 de noviembre de 2021. (COLOR POR ITERACIÓN)

MES	SEMANA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
11	1	1	2	3	4	5
	2	8	9	10	11	12
	3	15	16	17	18	19
	4	22	23	24	25	26
12	5	29	30	1	2	3
	6	6	7	8	9	10
	7	13	14	15	16	17
	8	20	21	22	23	24
01	9	27	28	29	30	31
	10	3	4	5	6	7
	11	10	11	12	13	14
	12	17	18	19	20	21

¡NO TENEMOS EN CUENTA LOS FESTIVOS NACIONALES! Si los tuviésemos, se retrasaría y acabaría el 20 de enero de 2022, en vez del 14 de enero de 2022.

Esta planificación tiene el mismo presupuesto que la secuencial porque se trabajan las mismas horas, pero se reparten entre varios trabajadores. Sin embargo, la agenda se reduce considerablemente, en la preparación secuencial se emplean 34 semanas y en la paralela tan solo 11.

La planificación paralela, aunque es mejor que la anterior, hay desperdicio porque unos trabajan más que otros.

CONTROL DE CALIDAD

Cuando hablamos de la calidad del software, tenemos que ver con la norma ISO 25000. De una manera más concreta estaremos utilizando de las divisiones, ISO 250Xn que habla sobre el modelo de calidad de los productos de software.

En este planteamiento podemos encontrar varias características de calidad y ahora hablaremos de cuáles de esas características están relacionadas con el software.

Adecuación funcional

En esta característica se incluyen:

- ***Integridad funcional***: Podemos alcanzar esta calidad asegurando que se cumplan todos los requisitos funcionales.
- ***Corrección funcional***: Para asegurar esto tenemos que demostrar que cada característica que ofrece el sistema se realiza correctamente.

Eficiencia de rendimiento

En esta característica se incluyen:

- ***Comportamiento temporal***: Como tenemos que acceder a una enorme base de datos entonces debemos asegurarnos de que una simple búsqueda de una de las propiedades o incluso un conjunto de ellas, sea rápida. Una forma sencilla de hacerlo es incluir una optimización del código durante la implementación.
- ***Uso de recursos***: Como tendremos que almacenar la información, podemos utilizar una base de datos externa para almacenar toda la información.

Seguridad

En esta característica se incluyen otras 5 más:

- ***Confidencialidad***: Podemos tener esta característica de seguridad implementando algún tipo de método de autenticación para los empleados.
- ***Integridad***: Debemos comprobar qué está introduciendo el usuario en el sistema, evitando algunos problemas de entrada, que no haya fallos de seguridad en el código, como que le falte introducir el usuario y contraseña.

Capacidad de uso

En esta característica se incluyen otras 6 más:

- ***Capacidad a utilizar***: Podemos asegurar que nuestro sistema será muy utilizable diseñando muy buenas interfaces de usuario.

GESTIÓN DE LA CONFIGURACIÓN

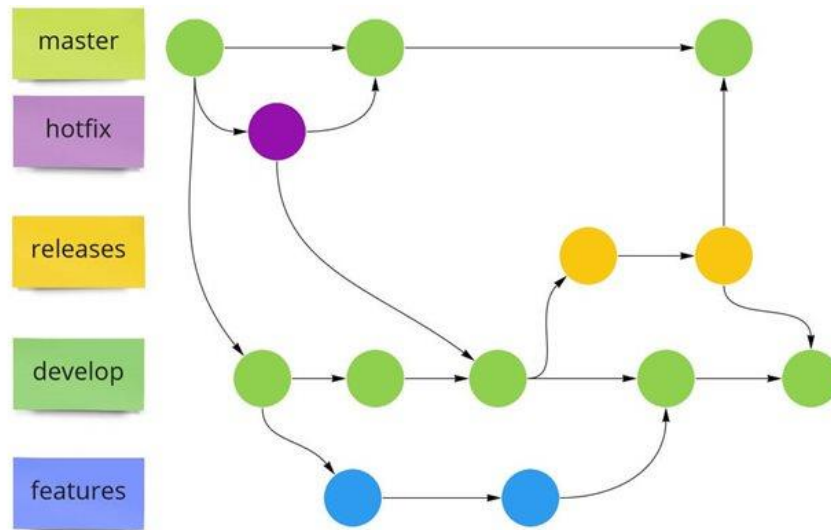
Para la **gestión de cambios**, denominaremos a alguien del equipo para encargarse del papel de CBB siendo así el que haga el análisis de los costes y beneficios de los cambios propuestos, evalúe el riesgo de no realizar los cambios o de realizarlos y propagarlos. Teniendo en cuenta además si son viables o no.

En la **gestión de versiones** hemos seleccionado el sistema de control de versiones **GIT** por su versatilidad, ya que nos permite de una forma fácil y eficaz hacer un seguimiento de las diferentes versiones de los componentes de software o "Baselines", y asegurar que los cambios de nuevas versiones no interfieran o creen conflictos unos con otros.

Para esto último mencionado, los conflictos entre el trabajo de varios programadores, usaremos el proceso de ramificación denominado **GitFlow**.

Dentro de este proceso de ramificación tendremos las siguientes 5 ramas:

- **Master.** La más importante ya que se encarga de la producción principal. En caso de detectar un fallo urgente lo arreglaremos enviándolo a la rama Hotfix, y posteriormente volverá a la rama de producción. Además, desde esta rama haremos el despliegue final del software.
- **Hotfix.** Rama encargada para arreglar errores o fallos urgentes encontrados en la rama Master. Una vez arreglado el fallo, enviaremos la nueva versión tanto a la rama Master como a la de desarrollo para no arrastrar fallos detectados.
- **Releases.** Es nuestra rama de preproducción. Una vez tenemos una versión del producto con las características suficientes para su producción, usaremos esta rama para probarlo y corregir defectos detectados. Los defectos corregidos, igual que en la rama de hotfix, se enviarán a la rama de desarrollo para no arrastrar esos defectos.
- **Develop.** Rama donde se implementan las nuevas funcionalidades y se integran los fallos corregidos en Hotfixes para finalmente formar una versión para preproducción.
- **Features.** Rama para desarrollar nuevas funcionalidades o características para el sistema. Una vez hechas se envían a la rama de desarrollo o Develop.



Dentro de todo el conjunto de versiones, para tener un mayor control usaremos la semántica de versiones **M.m.p** o también denominada **Major.Minor.Patch**. Usaremos la sección *Major* para incluir cambios nuevos o que rompen con la compatibilidad anterior del sistema. La sección *Minor* la usaremos para añadir cambios que no rompen con la compatibilidad del sistema. Por último, la sección de *Patch* será usada para la corrección de fallos o Bugs.

Para la **construcción del sistema** usaremos la herramienta **MAVEN**. Nos permitirá crear un sistema ejecutable, compilando y vinculando todos los componentes obtenidos de las versiones. En concreto realizaremos una generación de scripts de construcción automática, además de su integración; creación del sistema ejecutable, Generación de Informes, Generación de documentación del código...

En la **liberación del sistema** realizaremos una liberación del Software masivo, y por defecto una release Mayor siendo esta una nueva versión del producto. En caso de encontrar fallos que se resolverían en la rama Hotfix tendremos una release menor para corregir el sistema con un parche.