



Universidad **C**omplutense de **M**adrid

Facultad De Informática

Ingeniería del Software ***Plan de Proyecto***

Todo a la Caja B Team

Todo a la Caja B Team

O

nMarket

Proyecto: OnMarket

Plan de Proyecto

(Formato IEEE Std. 830-1998)

Miembros del equipo:

Daniel, Reyes Tubón	Miriam, Cabana Ramírez
Valentín, Campillo Martínez	Jagoba, Montes Larrabaster
Alejandro, Nieto Vila	Alejandro, Pedrosa García
Alejandro, Povedano Atienza	Alejandro, Sevilla Romero
Gonzalo, Pablos Sánchez	David, García Fernández
Enrique, Fuertes Franco	

Control de cambios

Número de versión	Fecha	Autores	Descripción
1.0	13/12/2016	Todo el equipo	Primer desarrollo del plan
2.0	16/02/2017	Valentin Campillo	Actualización de informes de gestión(6.2)
3.0	20/02/2017	Enrique Fuertes	Corrección planificación y MS project
3.1	21/02/2017	Alejandro Sevilla Alejandro Nieto	Replanteamiento de los riesgos y su planificación
3.2	21/02/2017	Jagoba Montes	Finalización del replanteamiento de los riesgos y su planificación
3.3	22/02/2017	Daniel Reyes	Se rehace entero apartado 2 Estimación Del Proyecto
3.4	22/02/2017	Daniel Reyes	Replanteamiento de 7.2 Control de Gestión de Cambios
3.5	8/04/2017	Alejandro Pedrosa Alejandro Sevilla	Reducción de riesgos tratados, replanteamiento de la planificación y correcciones

Índice

1. Introducción	7
1.1. Propósito del plan	7
1.2. Ámbito del proyecto y objetivos	7
1.2.1. Declaración del ámbito	7
1.2.2. Funciones principales	7
1.2.3. Aspectos de rendimiento	13
1.2.4. Restricciones y técnicas de gestión	14
1.3. Modelo de proceso	14
2. Estimaciones del proyecto	15
2.1. Datos históricos	15
2.2. Técnicas de estimación	16
2.3. Estimaciones de esfuerzo, coste y duración	18
3. Estrategia de gestión del riesgo	20
3.1. Análisis del riesgo	20
3.2. Estudio de los riesgos	22
3.3. Plan de gestión del riesgo	24
4. Planificación temporal	28
4.1. Estructura de descomposición del trabajo o Planificación temporal	28
4.2. Gráfico Gantt	29

4.3. Red de tareas	32
4.4. Tabla de uso de recursos	32
5. Recursos del proyecto	33
5.1. Personal	33
5.2. Hardware y software	33
5.3. Lista de recursos	34
6. Organización del personal	34
6.1. Estructura de equipo	34
6.2. Informes de gestión	36
7. Mecanismos de seguimiento y control	36
7.1. Garantía de calidad y control	36
7.2. Gestión y control de cambios	37
8. Apéndices	40

1. Introducción

1.1 Propósito del plan

El propósito del plan de proyecto es proporcionar toda la información relativa a la gestión del proyecto, al personal de desarrollo y al cliente. En este documento comunicamos el ámbito y los objetivos de la aplicación e informamos del modelo de proceso que vamos a seguir. Analizamos los riesgos que amenazan al proyecto y proporcionamos técnicas para su control y gestión. Detallamos la planificación temporal, concretando las tareas que se van a llevar a cabo, en qué fechas y por qué integrantes de grupo. Comunicamos los recursos disponibles para este proyecto (de personal, hardware y software), explicando además cómo se estructurará y organizará el equipo de trabajo. Finalmente, describimos cómo se garantizará la calidad del producto final y de qué manera se gestionarán los cambios lo largo del proyecto.

1.2 Ámbito del proyecto y objetivos

1.2.1. Declaración del ámbito

OnMarket es una aplicación que gestiona un supermercado online. Permite realizar compras como en supermercado real sin tener que moverse de casa. Además nos permite gestionar los clientes y los trabajadores de la aplicación.

La aplicación está pensada para realizar las gestiones pertinentes mediante una interfaz fácil de usar gracias a sus menús.

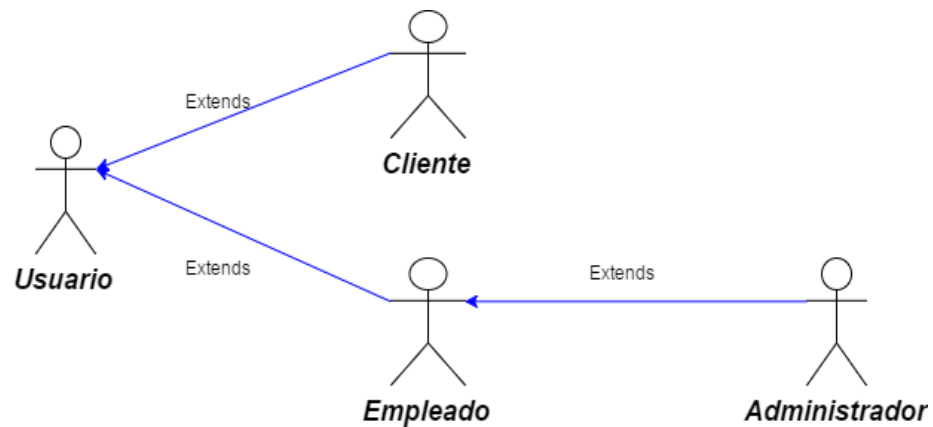
1.2.2. Funciones principales

En esta sección listamos las funciones principales de nuestra aplicación.

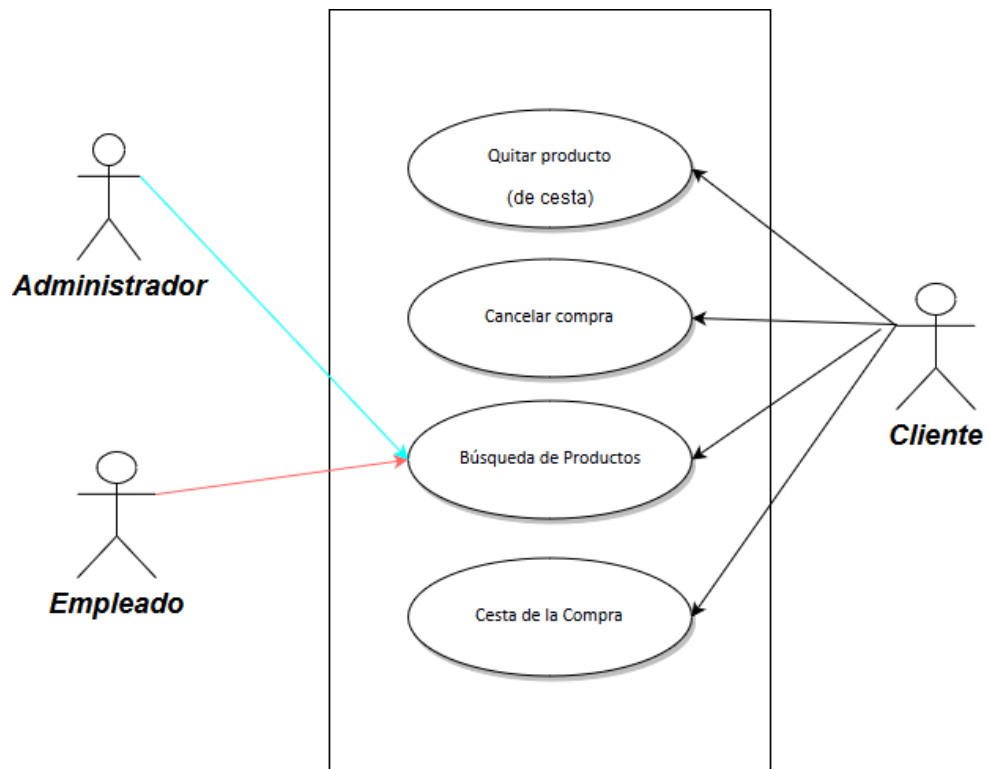
Para ello, vamos a usar diagramas de casos de uso para facilitar la comprensión al mencionar los actores y funciones específicamente:

Jerarquía de usuarios

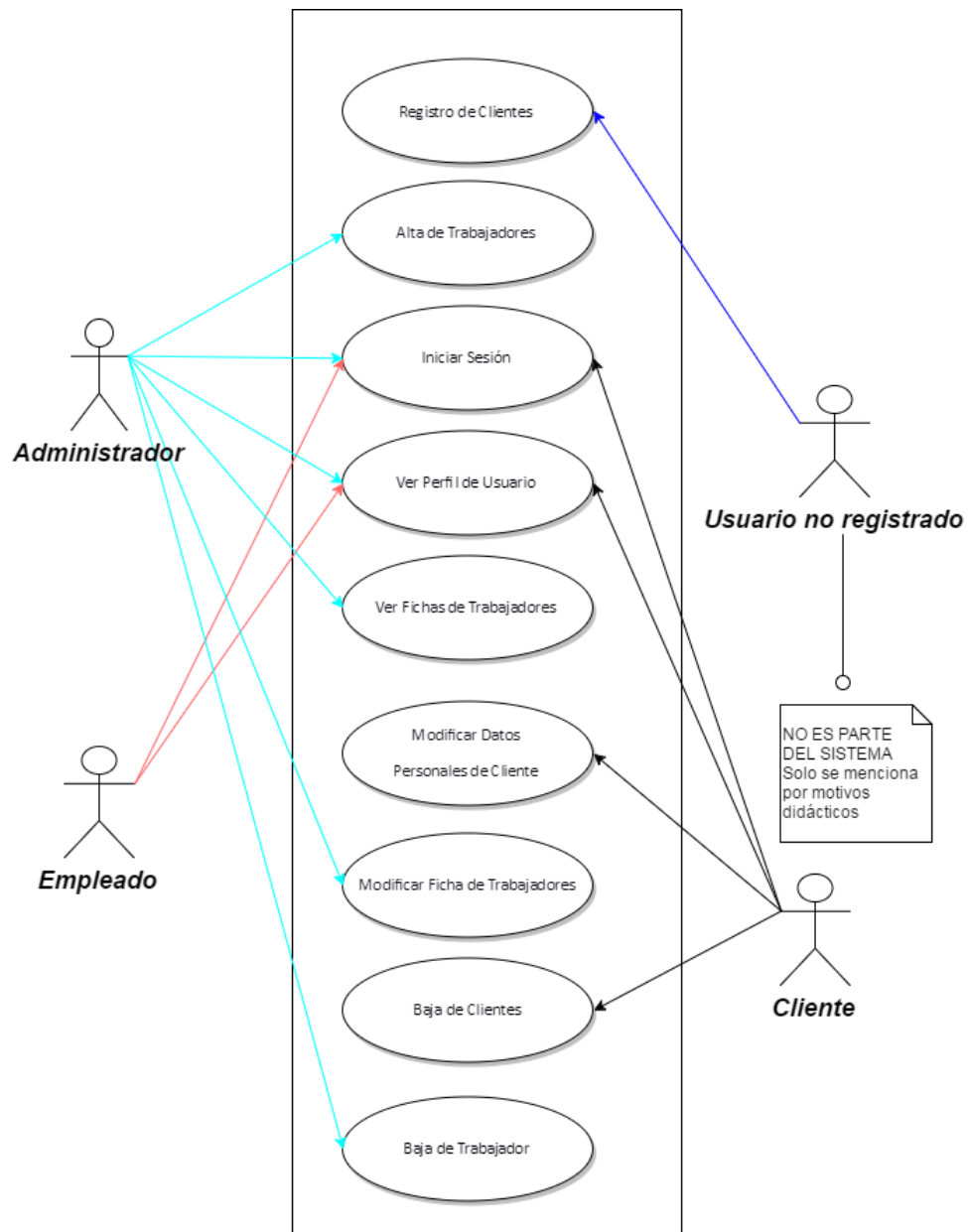
Para todos los diagramas usamos el siguiente esquema de jerarquía de usuarios, en adelante no se especifica gráficamente dicha jerarquía:



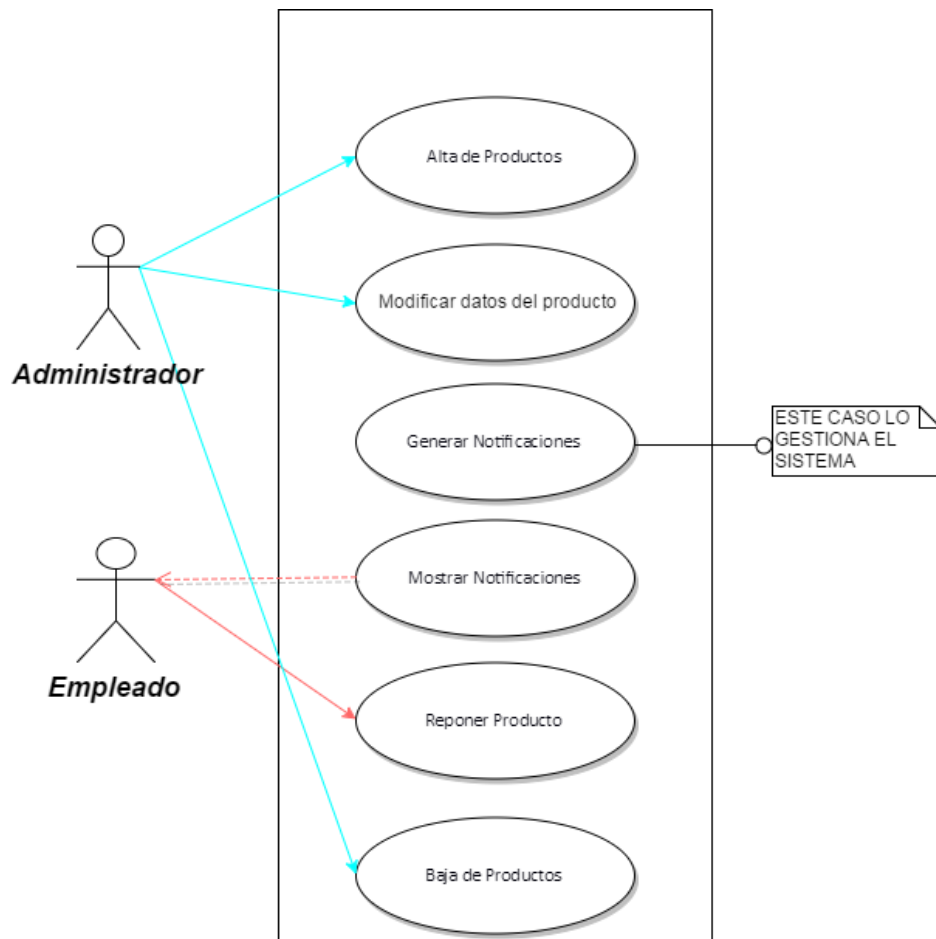
Módulo de uso de Catálogo



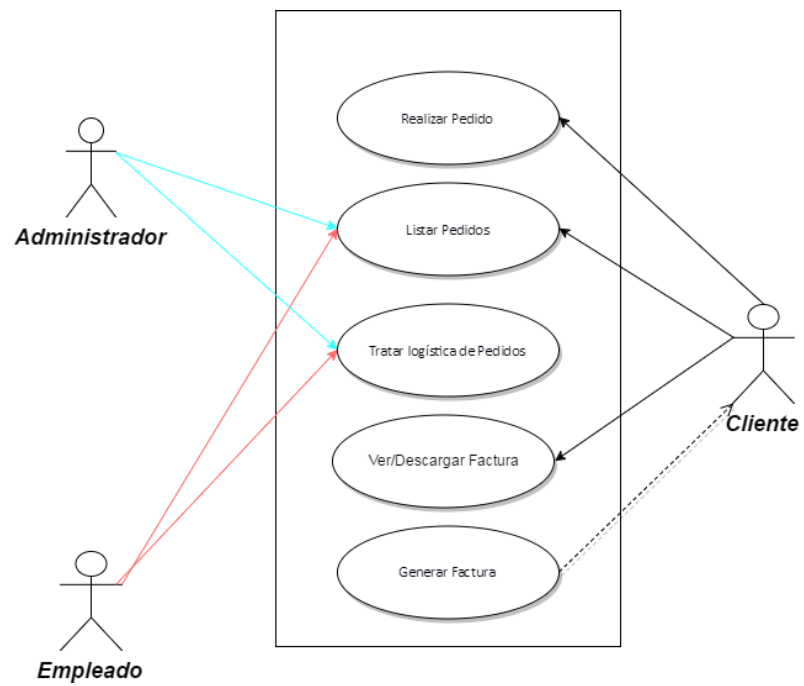
Módulo de Gestión de Usuarios



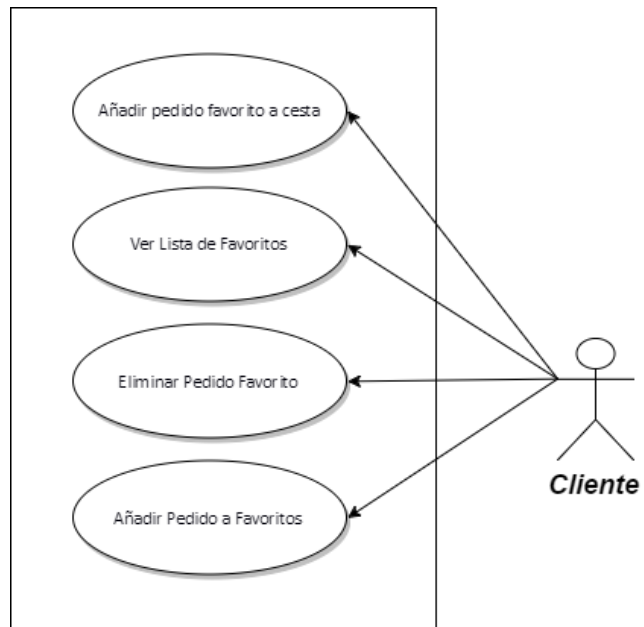
Módulo de Gestión de Productos



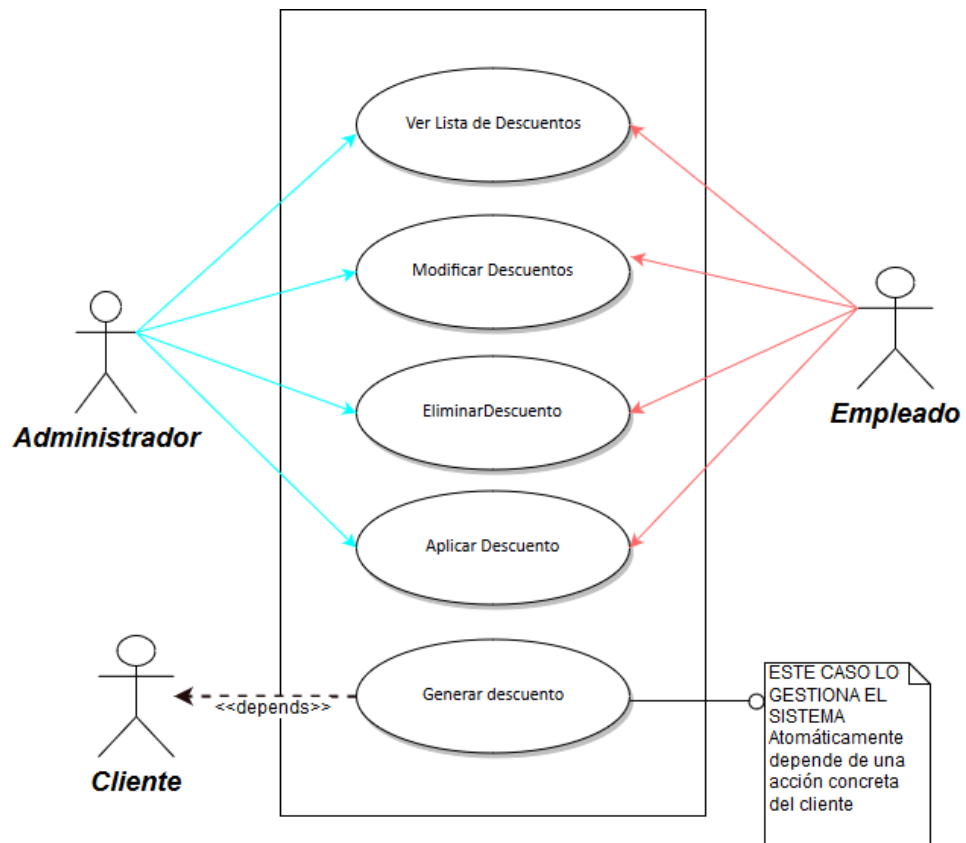
Módulo de Gestión de Pedidos



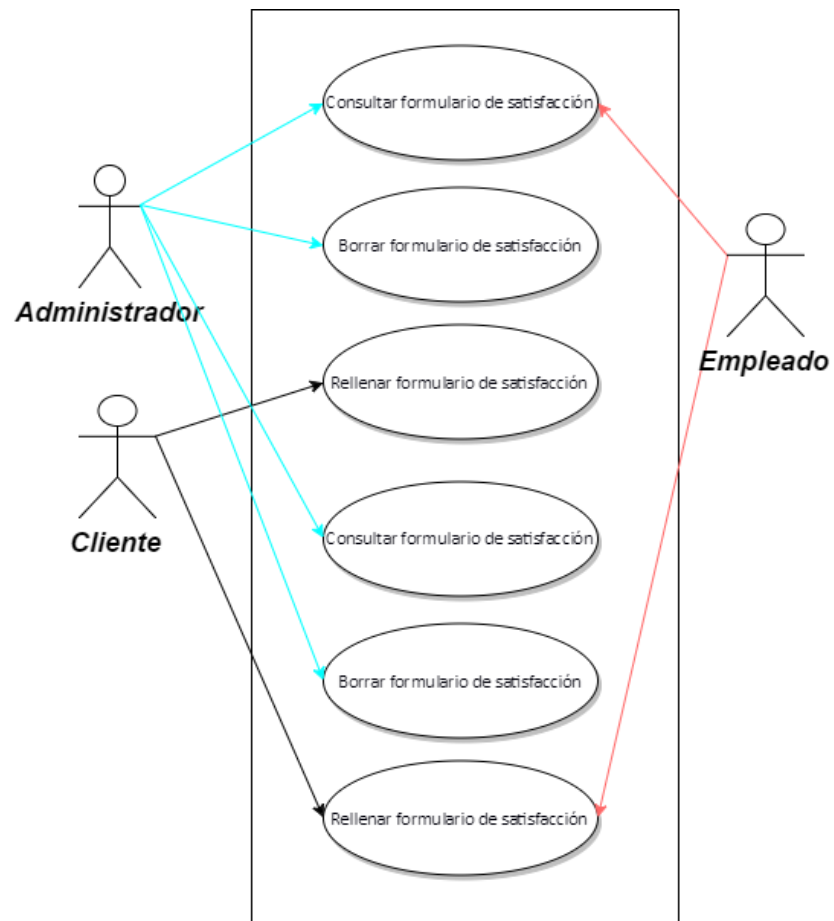
Módulo de Gestión de Favoritos



Módulo de Gestión de Descuentos



Postventa y Atención al Cliente



1.2.3. Aspectos de rendimiento

La aplicación debe soportar su ejecución en más de un terminal simultáneamente y su uso en paralelo por distintos tipos de usuarios. Dado que usamos una base de datos relacional con control de transacciones no se van a dar casos en los que varios clientes intenten acceder a un mismo producto si no hay existencia para los dos.

Después de cada una de las modificaciones realizadas por los usuarios de la aplicación, la base de datos se actualizará.

1.2.4. Restricciones y técnicas de gestión

Todos los componentes del proyecto están almacenados en una carpeta compartida por todo el personal del proyecto en Google Drive. La aplicación puede ejecutarse simultáneamente en varios terminales pudiendo usarse de forma paralela por distintos usuarios.

El proyecto se desarrolla hasta la entrega del producto final a lo largo de 7 meses en diversas entregas hasta la entrega de un plan final a finales de mayo de 2017.

La implementación del proyecto se realizará en Java y las BBDD serán almacenadas mediante MySql.

1.3. Modelo de proceso

El modelo de proceso que utilizaremos será el Proceso Unificado de Desarrollo. Es un modelo dirigido por casos de uso, iterativo e incremental.

Iterativo porque vamos a realizarlo a base de pasos de flujo de trabajo. Es incremental porque se comenzará por pequeños proyectos que se irán aumentando con el tiempo hasta conformar el resultado final. Cada iteración está compuesta por 4 fases: inicio, elaboración, construcción y transición. En cada iteración se tiene que llevar a cabo las tareas de identificar los requisitos de usuario, análisis, diseño, implementación y pruebas. Si una interacción cumple su objetivo se continúa con la siguiente interacción.

Para la implementación de los casos de uso usaremos el Lenguaje de Modelado Unificado (UML), ya que permite capturar los requisitos funcionales.

La principal ventaja de este modelo de proceso es que es un modelo racional. Permite un enfoque riguroso en el reparto de tareas y responsabilidades dentro del equipo de desarrollo. También utiliza tecnologías de componentes, lo que quiere decir, es que usa componentes software interconectados por medio de interfaces. Este modelo garantiza un producto software de alta calidad y que satisfaga todos los requerimientos del usuario final, en unas fechas de entrega y con un coste estimable de forma precisa.

El inconveniente que tiene este modelo es que está muy ligado al método, y esta metodología es bastante compleja. Es difícil aprender cómo funciona el proceso y cómo aplicarlo correctamente al proyecto, requiriendo de un equipo de desarrollo con experiencia en el uso de este modelo de proceso o que tenga en cuenta el esfuerzo adicional necesario para estudiarlo. Además, en proyectos de tecnología punta la reutilización de componentes no es posible y por tanto el tiempo que se ahorra gracias a la reutilización no se puede tener en cuenta.

2. Estimaciones del proyecto

2.1. Datos históricos

Para la creación del proyecto, no tenemos experiencia en la creación de proyectos. En referencia al material histórico, lo único de lo que disponemos son páginas similares a nuestro proyecto con el fin de conocer cuáles son sus alcances y cómo funcionan.

En consecuencia, nos hemos propuesto hallar la **productividad** del equipo.

Primero estableciendo que nos limitamos a lo que podemos hacer en **3 meses** (tiempo asignado al proyecto) con una perspectiva de trabajo de carga medio-alta desde el punto de vista de cada uno de los miembros.

Y segundo, en función de dos componentes:

- El tiempo que cada uno puede dedicar a escribir LDC:
- Las LDC que cada uno de los miembros puede aportar según su experiencia en 3 meses

	X=horas/semana	Y = X*4 =horas/mes	Z = LDC/hora	W= Y*Z= LDC/mes
Miembro 1	12	48	15	720
Miembro 2	10	40	15	600
Miembro 3	10	40	15	600
Miembro 4	9	36	17	612
Miembro 5	8	32	12	384
Miembro 6	8	32	15	480
Miembro 7	7	28	17	476

Miembro 8	9	36	12	432
Miembro 9	8	32	15	480
Miembro 10	8	32	12	384
Miembro 11	6	24	10	240
Total	95	380	155	5408

En conclusión, como equipo podemos aportar en 3 meses:

- Tiempo de trabajo = 380 horas/mes * 3 meses = 1140 horas = **1,58 meses**
- LDC = 5408 LDC/mes * 3 meses = **16224 LDC**

Luego:

- **Esfuerzo E** = 11 personas * 1,58 meses = **17,38 [PM]**
- **Productividad P** = 16224 [LDC] / 17,38 [PM] = **933,49 [LDC/PM]**

2.2. Técnicas de estimación

Para realizar la estimación, vamos a utilizar la técnica basada en la descomposición del problema, en este caso centrándonos en las LDC necesarias para implementar el Producto. Esta técnica se basa en descomponer el software en funciones para estimar su tamaño en función de las Líneas de Código (LCD) .

Lo que vamos hacer, es dividir el producto en módulos (según el SRS) y sobre cada módulo hacer un análisis estimado de las LDC que este va a tener con respecto de las siguientes funciones:

- **Interfaz de usuario (IU)**

Se estiman las LDC en función del número de interfaces de usuario que requiere y se aplica la fórmula $VE = (V_o + 4 V_m + V_p) / 6$ donde por cada interfaz:

V_o=100 LDC (basados en reutilización de código de elementos estáticos)

V_m= 160 LDC (basado en experiencia de miembros del equipo)

V_p= 180 LDC (basado en aparición de interfaces no contempladas)

→ **VE= 920/6= 153 LDC**

Luego,

Total LDC = N. de interfaces del módulo* VE

Total LDC = N. de interfaces del módulo* 153

- **Funciones de control de IU (FCIU)**

Se estiman las LDC en función del número de interfaces de usuario que requiere y el número de controles necesarios por cada una de ellas y se aplica la fórmula $VE = (V_o + 4 V_m + V_p) / 6$ donde por cada caso:

$V_o = 80$ LDC (basados en reutilización de código de elementos similares)

$V_m = 100$ LDC (empírico, basado en experiencias)

$V_p = 150$ LDC (basado en aparición de interfaces no contempladas)

→ **$VE = 920/6 = 105$ LDC**

N.controles(en c/interfaz) = $\text{óptimo} + 6 (\text{medio}) + \text{peor} / 6 = 3 + 4(5) + 7 / 6 = \sim 4$

Luego,

Total LDC = N. de interfaces del módulo*N.Controles*VE

Total LDC = N. de interfaces del módulo*4*105

- **Análisis y modelado de datos (AMD)**

Se estiman las LDC en función del número de entidades (clases) identificadas en el modelo del dominio y ponderadas a cada módulo y se aplica la fórmula $VE = (V_o + 4 V_m + V_p) / 6$ donde por cada entidad:

$V_o = 50$ LDC (basados en existencia de entidades cortas)

$V_m = 90$ LDC (basado en experiencia de miembros del equipo)

$V_p = 110$ LDC (basado en aparición de funciones de entidad no contempladas)

→ **$VE = 520/6 = 87$ LDC**

Luego,

N. Entidades=14 → N.Entidades por Módulo = $14/7 = 2$

Total LDC = N. Entidades por Módulo* VE

Total LDC = $2 * 87 = 174$ (para todos los módulos, ponderado)

- **Funciones de comunicación con datos persistentes (FCDP)**

Se estiman las LDC en función del número de interfaces de usuario que requiere y el número de funciones necesarias de este tipo aproximadas por cada una de ellas y se aplica la fórmula $VE = (V_o + 4 V_m + V_p) / 6$ donde por cada caso:

$V_o = 20$ LDC (basados en reutilización de código de elementos similares)

$V_m = 45$ LDC (empírico, basado en experiencias)

$V_p = 80$ LDC (basado en aparición de interfaces no contempladas)

→ **$VE = 280/6 = 47$ LDC**

N.Funciones (por c/interfaz)= $\text{óptimo}+6 \text{ (medio)}+\text{peor}/6= 3+4(5)+7/6 \approx 4$

Luego,

Total LDC = N. de interfaces del módulo*N.Funciones*VE

Total LDC = N. de interfaces del módulo*4*47

- **Módulos de diseño y gestión de la persistencia (MDGP)**

Se estiman las LDC en función del número necesario para la Creación de la estructura persistente más las necesarias para la inserción de datos de prueba por entidad y se aplica la fórmula $VE=(Vo+ 4 Vm + Vp)/6$ donde para cada caso:

Vo=100 LDC (basado en uso de herramientas y ayudas)

Vm= 200 LDC (basado en experiencia)

Vp= 250 LDC (casos no contemplados)

→ **VE= 600/6= 191 LDC**

Luego,

N. Entidades=14 → N.Entidades por Módulo = 14/7 = 2 (ponderado)

Total LDC = N. Entidades por Módulo* VE

Total LDC = 2* 191= 382 (para todos los módulos)

De acuerdo a las fórmulas y valores hallados anteriormente, podemos recoger una tabla donde se van acumulando las LDC de cada módulo desglosado por funciones:

	N. Interfaces de usuario (N)	IU (N*153)	FCIU (N*4*105)	AMD (174)	FCDP (N*4*47)	MDGP (382)	Total [LDC]
Módulo de Uso de Catálogo	3	459	1260	174	564	382	2839
Módulo de Gestión de Usuarios	7	1071	2940	174	1316	382	5883
Módulo de Gestión de Productos	4	612	1680	174	752	382	3600
Módulo de	5	765	2100	174	940	382	4361

Gestión de Pedidos							
Módulo de Gestión de Favoritos	2	306	840	174	376	382	2078
Módulo de Gestión de Descuentos	3	459	1260	174	564	382	2839
Módulo de Post Venta	5	765	2100	174	940	382	4361

Total: 25961 [LDC]

En conclusión, el tamaño del producto en líneas de Código, nos da **25961 [LDC]**. Sin embargo hay que recordar que esta aproximación, no es más que estimada. Y aunque tratamos de contar LDC basadas en experiencias reales de algunos de los miembros, puede que no sean muy reales.

Desde luego habría que tener en cuenta, posibles riesgos de aquellas partes o funciones de implementación no contempladas.

De momento, vamos a trabajar con esta estimación, y durante el proceso de adecuación de cosas a implementar, que queda pendiente con el profesor, estimaremos un posible margen de error en estos cálculos.

2.3. Estimaciones de esfuerzo, coste y duración

La estimación de los proyectos de software es una parte vital del proyecto, pues realiza estimaciones del esfuerzo, coste, tiempo y recursos que conducirán a crear un programa o una aplicación.

La estimación de los costes y del esfuerzo deberá ser realizada por el gestor del proyecto. Este gestor, deberá documentar e informar sobre su aproximación, presentando datos e información que verifiquen sus aproximaciones. Este proceso debe ser continuo, debido a que el proyecto progresa y se identifican las variables y datos que entran en juego. Toda esta información debe ser introducida y analizada en el modelo de estimación que será presentado posteriormente.

En *OnMarket*, podemos vamos a basarnos en los valores obtenidos en los apartados

anteriores **2.1 y 2.2** para realizar estos cálculos. Sin embargo, debemos de tener presente que estas variables se encuentran directamente relacionadas con varios parámetros como pueden ser la experiencia, el grado de estudios, así como la motivación de los desarrolladores. También debemos tener en cuenta el reparto no eficiente de tareas, que pueden repercutir de manera negativa al progreso del proyecto.

Recordando, lo obtenido anteriormente tenemos:

a) Lo que el equipo puede ofrecer:

- Tiempo de trabajo = 380 horas/mes * 3 meses = 1140 horas = **1,58 meses**
- LDC = 5408 LDC/mes * 3 meses = **16224 LDC**
- **Esfuerzo** E= 11 personas * 1,58 meses = **17,38 [PM]**
- **Productividad** P = 16224 [LDC] / 17,38 [PM] = **933,49 [LDC/PM]**

b) Tamaño estimado del producto en LDC: 25961 [LDC]

Podemos entonces, estimar para el proyecto :

- Esfuerzo Necesario= Tamaño del Producto [LDC]/Productividad que ofrece el equipo
→ **E=25961 [LDC] / 933.49 [LDC/PM] = 27,8 [PM]**

Claramente podemos observar, que obtenemos un esfuerzo, mayor al que el equipo puede ofrecer en el tiempo determinado (3 meses). Por tanto, para poder cumplir con el proyecto, se necesitaría aumentar el esfuerzo de equipo en un factor **de 1.6** lo que resulta de dividir (27,8/17,38), el esfuerzo requerido entre el esfuerzo ofrecido. O bien, aumentar el tiempo proporcionado para el trabajo.

- En cuanto a tiempo para poder realizar el proyecto, se necesitan completar las LDC que propone el tamaño del producto **25961** , sabiendo que el equipo puede ofrecer **5408 [LDC/mes]** podemos plantear, :
 $5408[\text{LDC/mes}] * X[\text{meses}] = 25961 [\text{LDC}]$

→ **X= 5,14 [meses], sería el tiempo que el equipo necesitaría.**

- Para terminar, en cuanto a coste monetario, dado que no conocemos mucho el tema y no tenemos experiencia, es difícil aventurarse a dar una cifras, aún así hemos creído oportuno, tomar un ejemplo visto en clase.
Suponiendo que se tiene una tarifa laboral de 7800 [€/PM] el coste viene dado por

$$\text{Coste} = \text{Esfuerzo} / \text{Tarifa} \rightarrow \text{Coste} = 27,8 \text{ [PM]} * 7800 \text{ [€/PM]} = 216\,840 \text{ €}$$

3. Estrategia de gestión del riesgo

3.1. Análisis del riesgo

Vamos a proceder a estudiar aquellos riesgos que perjudiquen la planificación de nuestro proyecto y las posibles situaciones que puedan darse a lo largo del periodo de tiempo en que se lleva a cabo. Para ello, hemos tomado una estrategia proactiva, con la cual vamos a identificar los diversos riesgos potenciales y evaluar la probabilidad y consecuencia de los riesgos.

A continuación, priorizaremos dichos riesgos, siguiendo el modelo de Boehm (*Top 10 Software Risk Items*). Los riesgos a identificar en nuestro proyecto son los siguientes:

No adquisición de las competencias técnicas necesarias para llevar a cabo el proyecto:

Deficiencias debidas a la inexperiencia o a la nula adaptación del personal al esquema de trabajo del proyecto.

Abandono/falta de compromiso de los miembros del equipo:

Se puede producir un abandono de un miembro provocado por una renuncia a la asignatura u otras causas similares. Además se podría producir una expulsión por parte de los subdirectores y director del equipo si así corresponde. Por otra parte, la falta de compromiso con el proyecto también es un factor a tener en cuenta ya que va a provocar el retraso y mal avance del trabajo.

Mala planificación:

Falta de exactitud a la hora de estimar el esfuerzo y el tiempo requerido para concluir la totalidad de las tareas, lo que implica una planificación errónea.

Desarrollo de las funciones y propiedades erróneas:

Desarrollo no adecuado de las funciones, lo que genera errores inesperados o comportamientos distintos a los esperados por el cliente.

Incumplimiento de la planificación temporal:

Debido a la aleatoriedad que presentan ciertas asignaturas, añadiendo entregables inesperados o que requieren más esfuerzo del estimado al principio del proyecto, ciertos trabajadores no cumplen con sus obligaciones para con el proyecto, lo que implica la adición de trabajo a otros integrantes, pudiendo suponer un retraso en la fecha de entrega del proyecto.

Cambios constantes en los requisitos:

Debido a las faltas de entendimiento sobre los requerimientos del cliente, por parte del equipo de desarrollo, se introducen, mejoran o suprimen requisitos ya acabados o en proceso de serlo, lo que supone mayor esfuerzo que el estimado al principio del proyecto; esto genera la necesidad de una nueva planificación y un retraso de la fecha de entrega.

Reparto ineficiente de las tareas:

El hecho de que unos trabajadores tengan una cantidad de trabajo mayor, hasta el límite de no poder llevarla a cabo, junto al hecho de que otros trabajadores no tengan trabajo suficiente o no estén activos la cantidad de tiempo estimada, es un claro indicio de la ineficiencia en el reparto de las tareas.

Falta de comunicación:

Uso indebido del control de cambios o modificaciones en el proyecto no notificadas. Pueden suponer desentendimientos y cambios simultáneos en el proyecto que lleven

irremediablemente a perder información, lo que supone pérdida de tiempo y esfuerzo, que retrasan la entrega del proyecto.

Errores en entendimiento o implementación de los requisitos:

Diferencias entre las expectativas del cliente y la implementación del requisito relacionado con estas.

3.2. Estudio de los riesgos

Estimamos la probabilidad de que los riesgos mencionados anteriormente se cumplan y las repercusiones que tendrían. La probabilidad nos va a dar una idea de cómo de posible es que el riesgo se convierta en un hecho mientras que la repercusión nos indica la gravedad de la situación en la que el riesgo se hace real.

Vamos a priorizar dichos riesgos según los niveles descritos en el SQAS-SEI, que son los siguientes, en orden creciente:

- Grado de probabilidad: *Improbable, Remoto, Ocasional, Probable, Frecuente*.
- Grado de repercusión: *Insignificante, Menor, Seria, Crítica, Catastrófica*.

No adquisición de las competencias técnicas necesarias para llevar a cabo el proyecto:

- Probabilidad: Probable. No todos los miembros conocían plataformas como el MS Project o GitHub y es difícil la coordinación entre todo el equipo al ser 11 integrantes.
- Repercusión: Seria.

Abandono/falta de compromiso de los miembros del equipo:

- Probabilidad: Ocasional.
- Repercusión: Crítica. Sería necesario establecer un nuevo reparto de tareas e incluso tener que aumentar el esfuerzo realizado por el personal.

Mala planificación:

- Probabilidad: Ocasional. Al existir personal sin experiencia en ningún otro proyecto, existe la posibilidad de que estimen de manera errónea el esfuerzo que pueden invertir.
- Repercusión: Crítica.

Con estos datos, establecemos el nivel de riesgo mediante la siguiente tabla:

Probabilidad Gravedad	Frecuente	Probable	Ocasional	Remota	Improbable
Catastrófica	IN	IN	IN	A	M
Crítica	IN	IN	A	M	B
Seria	A	A	M	B	T
Menor	M	M	B	T	T
Insignificante	M	B	T	T	T

Donde T sería: tolerable, B bajo, M medio, A alto, IN intolerable.

1. Tolerables: Ninguno
2. Bajos: Ninguno
3. Medios: Ninguno
4. Altos:
 - No adquisición de las competencias técnicas necesarias para llevar a cabo el proyecto
 - Abandono/falta de compromiso de los miembros del equipo
5. Intolerables:
 - Incumplimiento de la planificación temporal

Y obtenidos estos datos, preparamos un plan de actuación para aquellos riesgos que no sean aceptables.

3.3. Plan de gestión del riesgo

Al usar la tabla SQAS-SEI hemos considerado controlar únicamente el abandono/expulsión de los miembros del equipo, la no adquisición de las competencias técnicas necesarias para llevar a cabo el proyecto, la planificación y presupuestos poco realistas y el incumplimiento de la planificación temporal ya que son aquellos que más pueden darse en nuestro proyecto

(lo que no implica que los otros riesgos no sean importantes, con esto queremos decir que los otros no van a darse o no contemplamos que se den en nuestro proyecto) . Para controlar dichos riesgos aplicaremos medidas de reducción, supervisión y gestión. Para los riesgos de menor nivel de peligrosidad se propondrán alternativas para evitarlos, controlarlos, o bien asumirlos o transferirlos.

- **Abandono/falta de compromiso de los miembros del equipo:**
 - **Resolución:** Comprobamos que todos los miembros del equipo tengan un cierto interés y compromiso con el proyecto, para evitar el incumplimiento de tareas que puedan llevar a la expulsión del personal.
 - **Monitorización:** Se procederá a una revisión periódica de las tareas para realizar un seguimiento de quién las está cumpliendo en tiempo y quién no. Para ello se encargará dicha función a los subdirectores.
 - **Gestión:** Se producirá una redistribución de las tareas intentando no aumentar demasiado la carga de trabajo de cada miembro.
- **No adquisición de las competencias técnicas necesarias para llevar a cabo el proyecto**
 - **Resolución:** Los subdirectores y/o director de proyecto y de tareas intentarán asignar las actividades de manera que tengan constancia de que la persona que las va a llevar a cabo tenga las competencias necesarias para realizarla.
 - **Monitorización:** Comprobar que cada uno de los individuos que se encargan de un cometido concreto esté realizando de manera correcta su labor, de forma que tenga coherencia.
 - **Gestión:** Si se produce algún problema durante el transcurso de alguna tarea debido a la deficiencia del personal, se reasignará dicha tarea a otra persona, o se le instruirán las competencias necesarias, si no perjudica la planificación del proyecto.
- **Mala planificación:**
 - **Resolución:** Durante el desarrollo del proyecto se pueden ir estimando los costes de llevar a cabo cada componente del mismo y además se puede incrementar un poco más su valor con el fin de poder tener un “colchón” si se da el caso de que necesita más inversión de la estimada.

- **Monitorización:** Se comprobará que todo elemento que se necesite sea el adecuado para la labor que tiene que desempeñar.
- **Gestión:** Si se produce un incremento del coste del proyecto se investigará cuál es el motivo y se intentará reducir el coste de algún otro componente si es posible. También se le informará al cliente dicho aumento y cuál ha sido el motivo.

4. Planificación temporal

4.1. Estructura de descomposición del trabajo o Planificación temporal

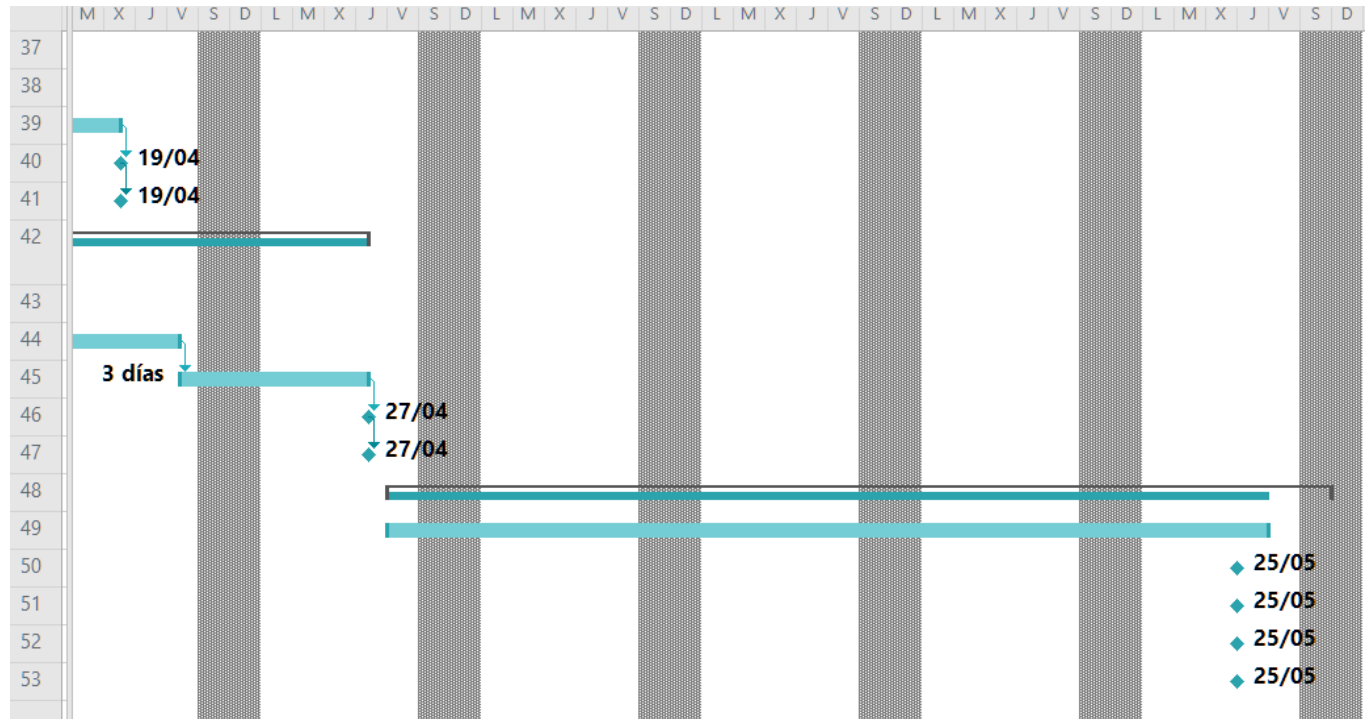
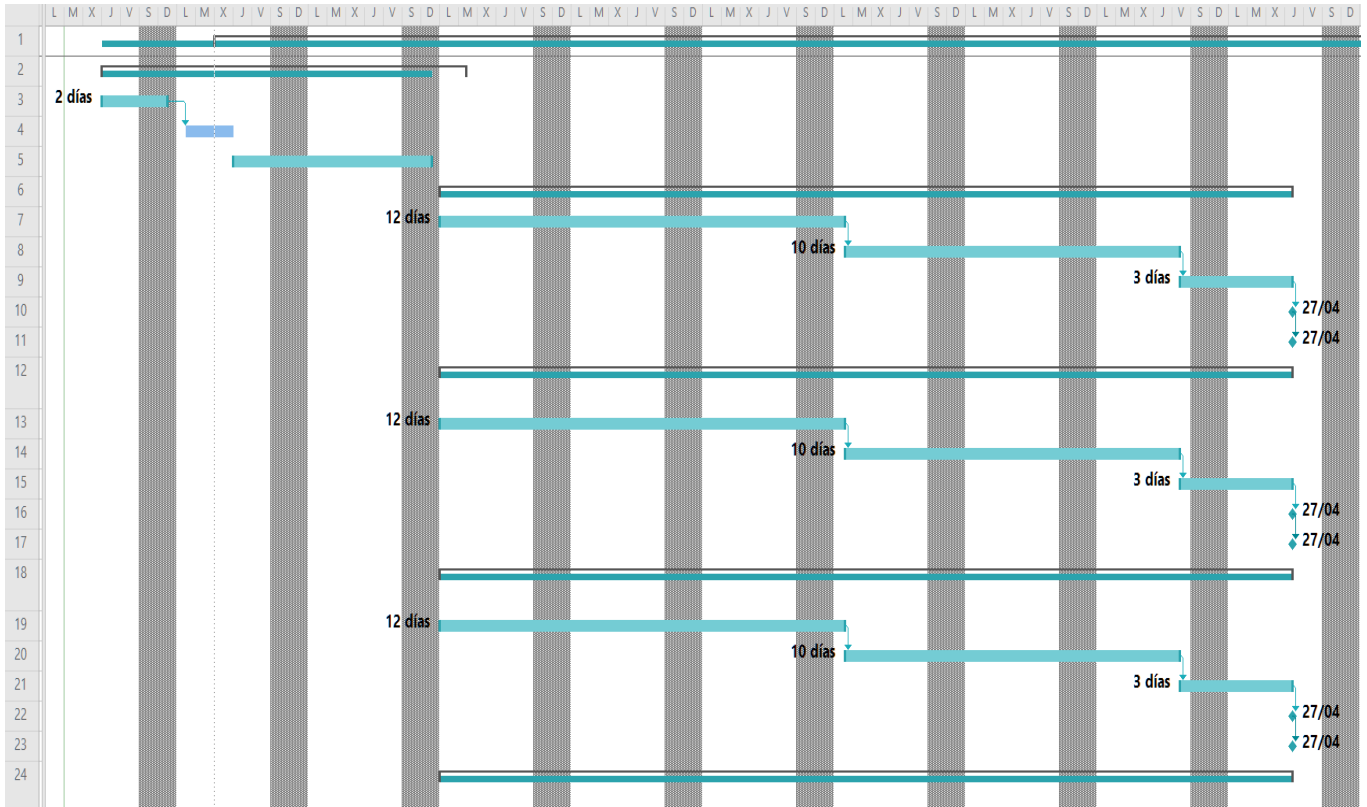
Columna1	Columna2	Columna3	Columna4	Columna5	Columna6	Columna7	Columna8	Columna9	Columna10
AE ->	COM cliente	Planificacion y gestion del riesgo	Ingenieria		Const y adap			Eval. Cliente	
Accion			Analisis	diseño	Codif	Prueba	Ensemb.	Intal.	Eval
Proyecto			1.1 i: 23.02.17 f: 26.02.17 r: Daniel, Jagoba, Valentin e: Analisis de proyecto	1.2 i: 26.02.17 f: 01.03.17 r: Daniel, Jagoba, Valentin e: Diseño de proyecto					
Cierre de Requisitos			1.3 i: 02.03.17 f: 08.03.17 r: Todo el equipo	1.4 i: 09.03.17 f: 12.03.17 r: Todo el equipo					
Modulo de uso de catalogo (MUC)				2.1 i: 13.03.17 f: 03.04.17 r: Daniel,Alejandro Nieto e: Diseño MUC	2.2 i: 04.04.17 f: 21.04.17 r: Daniel,Alejandro Nieto e: Codificación MUC	2.3 i: 21.04.17 f: 27.04.17 r: Alejandro Nieto, Daniel e: Codificación MUC	2.4 i: 28.04.17 f: 28.04.17 r: Daniel,Alejandro Nieto e: Ensamblaje MUC	2.5 i: 03.05.17 f: 03.05.17 r: Daniel,Alejandro Nieto e: Instalación MUC	2.6 i: 03.05.17 f: 10.05.17 r: e:
Modulo de gestión de Clientes (MGC)				3.1 i: 13.03.17 f: 03.04.17 r: Valentin, Daniel e: Diseño de MGC	3.2 i: 04.04.17 f: 21.04.17 r: Daniel, Valentin e: Codificación MGC	3.3 i: 21.04.17 f: 27.04.17 r: Valentin, Daniel e: Codificación MGC	3.4 i: 28.04.17 f: 28.04.17 r: Daniel, Valentin e: Ensamblaje MGC	3.5 i: 03.05.17 f: 03.05.17 r: Daniel, Valentin e: Instalación MGC	3.6 i: 03.05.17 f: 10.05.17 r: e:
Modulo de Gestión de Empleados (MGE)				4.1 i: 13.03.17 f: 03.04.17 r: David,Alejandro Nieto e: Diseño de MGE	4.2 i: 04.04.17 f: 21.04.17 r: David, Alejandro Nieto e: Codificación MGE	4.3 i: 21.04.17 f: 27.04.17 r: Alejandro Nieto, David e: Codificación MGE	4.4 i: 28.04.17 f: 28.04.17 r: David, Alejandro Nieto e: Ensamblaje MGE	4.5 i: 03.05.17 f: 03.05.17 r: David, Alejandro Nieto e: Instalación MGE	4.6 i: 03.05.17 f: 10.05.17 r: e:
Modulo de Gestión de productos (MGP)				5.1 i: 13.03.17 f: 21.04.17 r: Alejandro Povedano, Alejandro Pedrosa, e: Diseño de MGP	5.2 i: 04.04.17 f: 21.04.17 r: Alejandro Povedano, Alejandro Pedrosa, e: Codificación MGP	5.3 i: 21.04.17 f: 27.04.17 r: Alejandro Povedano, Alejandro Pedrosa, e: Codificación MGP	5.4 i: 28.04.17 f: 28.04.17 r: Alejandro Povedano, Alejandro Pedrosa, e: Ensamblaje MGP	5.5 i: 03.05.17 f: 03.05.17 r: Alejandro Povedano, Alejandro Pedrosa, e: Instalación MGP	5.6 i: 03.05.17 f: 10.05.17 r: e:
Modulo de Gestión de descuentos (MGD)				6.1 i: 13.03.17 f: 03.04.17 r: Jagoba, Miriam e: Diseño de MGD	6.2 i: 04.04.17 f: 21.04.17 r: Jagoba, Miriam e: Codificación MGD	6.3 i: 21.04.17 f: 27.04.17 r: Jagoba, Miriam e: Codificación MGD	6.4 i: 28.04.17 f: 28.04.17 r: Jagoba, Miriam e: Ensamblaje MGD	6.5 i: 03.05.17 f: 03.05.17 r: Jagoba, Miriam e: Instalación MGD	6.6 i: 03.05.17 f: 10.05.17 r: e:
Modulo de Gestión de favoritos (MGF)				7.1 i: 13.03.17 f: 03.04.17 r: Alejandro Sevilla, e: Diseño de MGF	7.2 i: 04.04.17 f: 15.04.17 r: Alejandro Sevilla, e: Codificación MGF	7.3 i: 15.04.17 f: 18.04.17 r: Alejandro Sevilla, e: Codificación MGF	7.4 i: 28.04.17 f: 28.04.17 r: Alejandro Sevilla, e: Ensamblaje MGF	7.5 i: 03.05.17 f: 03.05.17 r: Alejandro Sevilla, e: Instalación MGF	7.6 i: 03.05.17 f: 10.05.17 r: e:
Modulo de Gestión de pedidos (MGPE)				8.1 i: 13.03.17 f: 03.04.17 r: Daniel, Enrique, e: Diseño de MGPE	8.2 i: 04.04.17 f: 21.04.17 r: Daniel, Enrique, e: Codificación MGPE	8.3 i: 21.04.17 f: 27.04.17 r: Daniel, Enrique, e: Codificación MGPE	8.4 i: 28.04.17 f: 28.04.17 r: Daniel, Enrique, e: Ensamblaje MGPE	8.5 i: 03.05.17 f: 03.05.17 r: Daniel, Enrique, e: Instalación MGPE	8.6 i: 03.05.17 f: 10.05.17 r: e:
Proyecto	9.1 16.05.17 r: Daniel e: Opinión y revisión SRS	9.2 16.05.17 r: Jagoba e: Cierre gestión	9.3 16.05.17 r: Alejandro Nieto e: Cierre ingeniería		9.4 16.05.17 r: Valentin e: Cierre código			9.5 17.05.17 r: Daniel e: Entrega Proyecto	

4.2. Gráfico Gantt

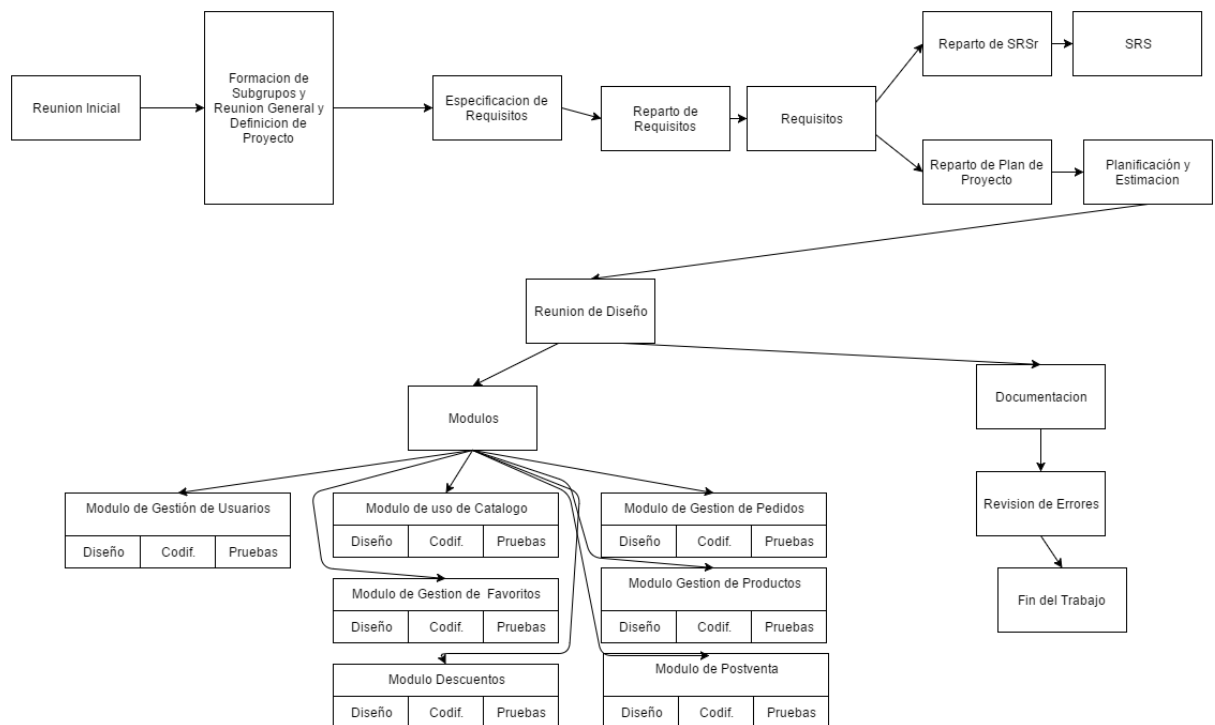
1		✈	✈	49 días	mié 01/03/17	mié 31/05/17	
2		✈	✈ Proyecto	10 días	jue 23/02/17	mar 14/03/17	
3		✈	1.1 Análisis de proyecto	2 días	jue 23/02/17	dom 26/02/17	
4	📅	✈	1.2 Diseño de Proyecto	2 días	lun 27/02/17	mié 01/03/17	3
5		✈	1.3 Cierre de requisitos	6 días	jue 02/03/17	dom 12/03/17	
6		✈	✈ Módulo de uso de catálogo (MUC)	25 días	lun 13/03/17	jue 27/04/17	
7		✈	2.1 Diseño MUC	12 días	lun 13/03/17	lun 03/04/17	
8		✈	2.2 Codificación MUC	10 días	lun 03/04/17	vie 21/04/17	7
9		✈	2.3 Codificación MUC	3 días	vie 21/04/17	jue 27/04/17	8
10		✈	2.4 Ensamblaje MUC	0 días	jue 27/04/17	jue 27/04/17	9
11		✈	2.5 Instalación MUC	0 días	jue 27/04/17	jue 27/04/17	10
12		✈	✈ Módulo de uso de gestión de clientes (MGC)	25 días	lun 13/03/17	jue 27/04/17	
13		✈	3.1 Diseño MGC	12 días	lun 13/03/17	lun 03/04/17	
14		✈	3.2 Codificación MGC	10 días	lun 03/04/17	vie 21/04/17	13
15		✈	3.3 Codificación MGC	3 días	vie 21/04/17	jue 27/04/17	14
16		✈	3.4 Ensamblaje MGC	0 días	jue 27/04/17	jue 27/04/17	15
17		✈	3.5 Instalación MGC	0 días	jue 27/04/17	jue 27/04/17	16
18		✈	✈ Módulo de Gestión de Empleados (MGE)	25 días	lun 13/03/17	jue 27/04/17	
19		✈	4.1 Diseño MGE	12 días	lun 13/03/17	lun 03/04/17	
20		✈	4.2 Codificación MGE	10 días	lun 03/04/17	vie 21/04/17	19
21		✈	4.3 Codificación MGE	3 días	vie 21/04/17	jue 27/04/17	20
22		✈	4.4 Ensamblaje MGE	0 días	jue 27/04/17	jue 27/04/17	21
23		✈	4.5 Instalación MGE	0 días	jue 27/04/17	jue 27/04/17	22

24		✈	▫ Módulo de Gestión de Productos (MGP)	25 días	lun 13/03/17	jue 27/04/17	
25		✈	5.1 Diseño MGP	12 días	lun 13/03/17	lun 03/04/17	
26		✈	5.2 Codificación MGP	10 días	lun 03/04/17	vie 21/04/17	25
27		✈	5.3 Codificación MGP	3 días	vie 21/04/17	jue 27/04/17	26
28		✈	5.4 Ensamblaje MGP	0 días	jue 27/04/17	jue 27/04/17	27
29		✈	5.5 Instalación MGP	0 días	jue 27/04/17	jue 27/04/17	28
30		✈	▫ Modulo de Gestión de Descuentos (MGD)	25 días	lun 13/03/17	jue 27/04/17	
31		✈	6.1 Diseño de MGD	12 días	lun 13/03/17	lun 03/04/17	
32		✈	6.2 Codificación de MGD	10 días	lun 03/04/17	vie 21/04/17	31
33		✈	6.3 Codificación de MGD	3 días	vie 21/04/17	jue 27/04/17	32
34		✈	6.4 Ensamblaje MGD	0 días	jue 27/04/17	jue 27/04/17	33
35		✈	6.5 Instalación MGD	0 días	jue 27/04/17	jue 27/04/17	34
36		✈	▫ Modulo de Gestión de Favoritos (MGF)	21 días	lun 13/03/17	mié 19/04/17	
37		✈	7.1 Diseño de MGF	12 días	lun 13/03/17	lun 03/04/17	
38		✈	7.2 Codificación de MGF	7 días	lun 03/04/17	sáb 15/04/17	37
39		✈	7.3 Codificación de MGF	2 días	lun 17/04/17	mié 19/04/17	38
40		✈	7.4 Ensamblaje MGF	0 días	mié 19/04/17	mié 19/04/17	39
41		✈	7.5 Instalación MGF	0 días	mié 19/04/17	mié 19/04/17	40
42		✈	▫ Modulo de Gestión de Pedidos (MGPE)	25 días	lun 13/03/17	jue 27/04/17	
43		✈	8.1 Diseño de MGPE	12 días	lun 13/03/17	lun 03/04/17	
44		✈	8.2 Codificación de MGPE	10 días	lun 03/04/17	vie 21/04/17	43
45		✈	8.3 Codificación de MGPE	3 días	vie 21/04/17	jue 27/04/17	44
46		✈	8.4 Ensamblaje MGPE	0 días	jue 27/04/17	jue 27/04/17	45
47		✈	8.5 Instalación MGPE	0 días	jue 27/04/17	jue 27/04/17	46

42	✈	▴ Modulo de Gestión de Pedidos (MGPE)	25 días	lun 13/03/17	jue 27/04/17	
43	✈	8.1 Diseño de MGPE	12 días	lun 13/03/17	lun 03/04/17	
44	✈	8.2 Codificación de MGPE	10 días	lun 03/04/17	vie 21/04/17	43
45	✈	8.3 Codificación de MGPE	3 días	vie 21/04/17	jue 27/04/17	44
46	✈	8.4 Ensamblaje MGPE	0 días	jue 27/04/17	jue 27/04/17	45
47	✈	8.5 Instalación MGPE	0 días	jue 27/04/17	jue 27/04/17	46
48	✈	▴ Proyecto	16,63 días	vie 28/04/17	sáb 27/05/17	
49	✈	9.1 Opinión y revisión SRS	15,13 días	vie 28/04/17	jue 25/05/17	
50	✈	9.2 Cierre Gestión	0 días	jue 25/05/17	jue 25/05/17	
51	✈	9.3 Cierre Ingeniería	0 días	jue 25/05/17	jue 25/05/17	
52	✈	9.4 Cierre Código	0 días	jue 25/05/17	jue 25/05/17	
53	✈	9.5 Entrega Proyecto	0 días	jue 25/05/17	jue 25/05/17	



4.3. Red de tareas



4.4. Tabla de uso de recursos

Recursos	Octubre	Noviembre	Diciembre
Personal	11	11	11
Ordenadores	11	11	11
Google Drive	x	x	x
Balsamiq		x	x
Draw.io		x	x

WhatsApp	x	x	x
----------	---	---	---

5. Recursos del proyecto

5.1. Personal

- **Director del proyecto:** Daniel Reyes Tubón
- **Subdirectores:** Jagoba Montes Larrabaster, Valentín Campillo Martínez, Gonzalo Pablos Sánchez
- **Subgrupos:**
 1. Daniel Reyes Tubón, Alejandro Pedrosa García, Gonzalo Pablos Sánchez.
 2. Jagoba Montes Larrabaster, Alejandro Nieto Vila, Alejandro Sevilla Romero, David García Fernández.
 3. Valentín Campillo Martínez, Alejandro Povedano Atienza, Enrique Fuertes Franco, Miriam Cabana Ramírez.

5.2. Hardware y software

Hardware:

Cualquier ordenador que pueda compilar un programa en java.

Software:

- Microsoft Windows: Sistema operativo predominante en los ordenadores de los componentes del grupo.
- Microsoft Word / Libre Office / Open Office: Usado para los documentos a entregar del proyecto.
- Microsoft Project: Usado para la gestión del proyecto, así como cosas más específicas como el gráfico Gantt.
- Google Drive: Plataforma en la nube usada para que todos los miembros puedan acceder a los documentos, organizada por carpetas de desarrollo, producción y versiones.
- Eclipse: Programa que se usará para implementar el código java de la aplicación.
- Balsamiq: Usado para realizar las pantallas de la aplicación.
- WhatsApp: Herramienta principal de comunicación. Se han usado 3 grupos (uno por cada subgrupo) para facilitar la toma de decisiones y la propagación adecuada de las

mismas.

- Skype: Usado para conferencias entre el director y los subdirectores.
- Draw.io /IBM RSA: Usados para los diagramas UML del SRS.
- Campus Virtual de la UCM: Documentación para hacer el proyecto.
- Internet: Documentación auxiliar sobre temas concretos.

5.3. Lista de recursos

- Personal (especificado en el 5.1)
- Ordenadores
- Salas Laboratorios UCM
- Microsoft Windows
- Microsoft Word
- Microsoft Excell
- LibreOffice
- OpenOffice
- Microsoft Project
- Google Drive
- Eclipse
- Balsamiq Mockups
- WhatsApp
- Skype
- Draw.io
- IBM RSA
- Campus Virtual UCM
- Internet

6. Organización del personal

6.1. Estructura de equipo

Para organizar nuestro equipo, compuesto de 11 personas, usaremos un modelo Descentralizado Controlado (DC), según la organización de equipos de Mantei. Este modelo tiene las siguientes características:

- Tiene un director definido que coordina tareas específicas y subdirectores de equipos que tienen responsabilidades sobre sub-tareas.
- La solución de problemas es una actividad del grupo, pero la implementación de soluciones se reparte entre los subgrupos por el director de equipo, determinándolo con los subdirectores.
- La comunicación entre subgrupos e individuos es horizontal, pero también existe comunicación vertical a lo largo de la jerarquía de control del grupo.

Existe un director de grupo, Daniel Reyes Tubón, elegido por consenso y que se encarga de organizar al resto del grupo y repartir el trabajo entre los distintos subgrupos, acordándolo previamente con los subdirectores.

Cada uno de los cuatro subgrupos está formado por tres o cuatro miembros, siendo alguno de ellos el elegido como subdirector.

Entre ellos se encargan de que cada miembro tenga las tareas adecuadas para que se cumplan a tiempo todas las entregas y la distribución sea lo más equitativa posible. Los cuatro subgrupos del equipo están reflejados en el 5.1 del Plan de Proyecto.

Para comunicarnos verbalmente hemos usado la aplicación WhatsApp. De esta manera, cada miembro del equipo tiene dos grupos:

- Uno general, en el que estamos los once miembros, y que se usa para informar acerca de las entregas finales y resolver dudas generales.
- Uno de cada subgrupo, por el cual se nos reparten las tareas a cada uno.

Para organizar el organigrama del equipo hemos tenido en cuenta las siguientes circunstancias:

- Dificultad del problema

La dificultad se ha visto reducida al organizar el equipo mediante el modelo Descentralizado Controlado, encargándose cada subgrupo de su tarea.

- Duración del equipo

El equipo llegará a su fin con la entrega final del proyecto.

- Fecha de entrega

Por definir, aunque estimamos que a finales de mayo de 2017.

- Calidad y fiabilidad

Para satisfacer al cliente se probará cada módulo de la aplicación de forma independiente y luego la aplicación una vez implementadas todas sus funcionalidades.

- Comunicación

Aparte de los grupos de WhatsApp, canal de comunicación principal, se realizan reuniones entre los distintos miembros del grupo/subgrupo, dependiendo de la disponibilidad de cada uno. Además de reuniones entre los subdirectores para organizar el trabajo. También se usa Google Drive para compartir las distintas partes hechas por cada integrante y para editar los documentos.

6.2. Informes de gestión

En la documentación del proyecto, se lleva un informe de cada parte realizada por cada uno, almacenada en las distintas carpetas en el Google Drive compartido del grupo.

Las competencias de cada uno las podemos agrupar en tres:

- Director de proyecto : Se encarga de la gestión de proyecto, reparte las tareas entre

los distintos grupos, y posteriormente revisará que todas las tareas se han realizado, se encarga de poner en común soluciones o problemas que puedan surgir, además se encargará de realizar las tareas asignadas propias de la entrega

- Subdirectores de proyecto : Su función principal es la de hacer de enlace entre el director de proyecto y los integrantes del subgrupo, al igual que el director de proyecto deberán poner en común soluciones y problemas, repartirán dentro de su grupo las tareas y posteriormente las revisarán, además de realizar las tareas propias de la entrega.
- Resto de los integrantes del grupo: Su función es aportar ideas, problemas y soluciones y realizar las tareas asignadas propias de la entrega y verificar que son correctas de acuerdo a los estándares.

Todos los integrantes son responsables al menos de una parte de cada entrega ya que se les asignará alguna tarea y tendrán que realizarla correctamente

Esta gestión se hace acompañando el documento de entrega de otro documento donde quedan registradas las tareas a realizar, responsables y responsabilidades de cada uno de los integrantes, además se deja constancia del trabajo realizado por cada uno.

7. Mecanismos de seguimiento y control

7.1. Garantía de calidad y control

Hemos tenido en cuenta la garantía de calidad y control mediante las actividades realizadas todo el cuatrimestre.

La garantía de calidad realiza las funciones de información de la gestión. Lo usamos para obtener una visión más completa del proyecto y poder comprobar que lo realizado va a cumplir cada uno de los objetivos y requisitos que le pedimos, así como para garantizar la relación con los estándares del desarrollo de software.

Después de ello realizamos el control de calidad que consiste en distintas comprobaciones y tests realizados durante el proceso de software para asegurar de que cada uno de los módulos cumple las condiciones que se le habían asignado (RTF).

Específicamente, en nuestro grupo, no existen una o más personas dedicadas específicamente al control de calidad del proyecto. Sin embargo, se espera que cada persona, al realizar su trabajo, lo haga comprobando que todo cumple unos requisitos

mínimos y unos patrones de calidad que sean comunes a todo el plan.

También usamos la estructura específica grupal para hacernos cargo de la garantía de calidad. Es decir, al terminar una tarea específica, esta es revisada para comprobar que se ha realizado de manera correcta. Si algo no cumple los estándares de calidad será comunicado al grupo para que pueda ser corregido antes de realizar la entrega.

El control de cambios se realiza en documento complementario a cada uno de los presentados en cada entrega y normalmente con el título “*read me*” en la que se especifican el cambio realizado, la persona que lo ha hecho y la versión del documento al que se ha aplicado. De esta forma sabemos en cada momento quién ha hecho cada caso y ha alterado cualquier parte del proyecto, y se tienen responsables por si ha habido algún error.

Cabe destacar que durante el desarrollo al realizar las revisiones técnicas formales o RTF solo nos encargamos de realizar unos test con el fin de ver que todo lo que se está implementando funcione. Además, antes de realizar cualquier implementación gracias a toda la documentación elaborada y preparada podemos preparar estrategias para llevar a cabo la implementación del software, también nos permite ver cuál es el progreso del proyecto. En resumen las RTFs nos permiten ir evaluando el trabajo que vamos realizando pero gracias a la SQA realizada anteriormente del desarrollo del proyecto podemos ser más precisos a la hora de tomar decisiones y llevar a cabo acciones. Las RTFs están dentro de la SQA por lo tanto si llevamos a cabo la garantía de calidad con la SQA ya tenemos unas RTFs pero si solo hacemos RTFs no tenemos la SQA.

7.2. Gestión y control de cambios

Con este propósito de tratar la GCS, hemos elegido dos herramientas que sumadas a las soluciones de equipo nos proporcionan la capacidad suficiente para tratar con los Cambios.

Por un lado, **Google Drive** que nos permite trabajar en paralelo sobre documentos.

Y por otro lado, **GitHub** que nos permite trabajar en paralelo y mantener controlados los cambios sobre el código y versiones, utilizando el sistema de control de versiones GIT.

Aunque un gran número de personas del equipo no ha tenido nunca contacto con esta herramienta se impartirán nociones básicas de cómo trabajar con esta plataforma por parte de los miembros que la conocen, se espera realizar todo correctamente a través de las funcionalidades que ofrece la plataforma y revisiones de la directiva; se estará ayudando en todo lo posible con el fin de que todos sean capaces de trabajar sin ningún tipo de problema.

A continuación, explicamos cómo con estas herramientas y la organización del equipo de trabajo conseguiremos, vamos a solventar las 5 actividades de CGS:

Identificación de ECSs

En este contexto, vamos a tratar la GCS centrándonos en dos elementos fundamentales, que a juicio del equipo, han sido identificados como **ECSs** pues son expuestos al cambio constantemente:

- La Documentación generada (SRS, Plan de Proyecto)
- Código generado para la implementación.

Para cada uno de ellos se marca una **línea base**.

- En el caso de la documentación, en Google Drive hemos creado para cada Entrega un directorio, que dentro siempre se compone de 2 subdirectorios **“DESARROLLO”** y **“PRODUCCIÓN”**.

En **PRODUCCIÓN**, es donde se ubican las **líneas base**, es decir contiene aquellos documentos, que han sido revisados y aprobados por la directiva del equipo. Sólomente el director del equipo tiene permisos de edición sobre dichos documentos y por tanto, para realizar algún cambio sobre ellos, hace falta la resolución de la directiva.

Mientras que en **DESARROLLO**, se ubican aquellos documentos sobre los cuales todo el equipo trabaja en paralelo y que están sometidos al cambio constante.

- En el caso del código, la plataforma GitHub, nos ofrece la posibilidad de mantener una línea base usando ramas, es decir el director del equipo crea un repositorio que contiene el proyecto, dado que cada repositorio está formado por una o varias ramas, la línea base se establece en la rama principal (master) sobre la que sólomente el director podrá hacer cambios, una vez se hayan aprobado por la directiva, se envía una solicitud para mezclar la rama que contiene los cambios aprobados con la rama principal (pull request).

Control de Versiones

- En el caso de la documentación, en Google Drive hemos creado un directorio distinto para cada entrega, que una vez realizada la misma, se mantiene la versión estable de todos los documentos y recursos generados en el subdirectorio **PRODUCCIÓN**.

- En el caso del código, la plataforma GitHub, al usar una estructura en ramas, permite crear nuevas ramas que se pueden configurar de la **misma manera que la línea base inicial** (master 2.0) constituyendo una nueva versión, a partir de la cual se pueden colgar otras ramas donde se pueden ir haciendo las modificaciones correspondientes a la nueva versión, proporcionando un control sobre las versiones al regirse por las mismas reglas descritas para la línea base. Por supuesto en estos casos se debe incluir en el repositorio un fichero con la descripción de la versión que se está usando en cada momento que será distinto para cada una de ellas.

Control de Cambios

- En el caso de la documentación, en Google Drive, dentro del directorio de **Desarrollo** que contiene los ECSs hemos creado además sendos documentos, con una lista de tareas a ejecutar, cada una de ellas cuenta con:
 - **Tema:** título de la tarea
 - **Responsables:** uno a más miembros a los que se les ha adjudicado la tarea
 - **Estado:** que varía entre los valores “pendiente”, “Editando-[Nombre del miembro]”, “Editado con Dudas”, “Editado”. Que permite saber en todo momento quien está a cargo y en qué estado se encuentra la tarea.
 - **Descripción:** que especifica las actividades de la tarea.
- En el caso del código, la plataforma GitHub, como ya hemos mencionado permite una estructura en ramas, con lo cual para los miembros del equipo lo que se hace es, clonar la rama de la línea base (máster) a una nueva rama (dev). En esta nueva rama se pueden ir haciendo los cambios a cargo del equipo, para controlarlos contamos con un mecanismo que va guardando un identificador único sobre cada cambio publicado, de tal forma que si hay algún problema siempre se puede cargar en la rama una versión anterior del trabajo.
También contamos, con un sistema de etiquetas (issues), donde se van a publicar y asignar las tareas a realizar por cada uno de los miembros, que al terminarla darán parte a la directiva, quienes revisarán el correcto funcionamiento y una vez aprobado, se cierra la issue, y queda listo el cambio para subirlo, a producción (máster).

Auditoría de la configuración

A lo largo del desarrollo del proyecto se van a ir realizando revisiones técnicas formales con el fin de verificar que todo lo desarrollado hasta ese momento funciona correctamente y realiza la tarea que se le tiene asignada. Para esto contamos la ayuda que tiene GitHub que nos permite documentar cuál es el estado de cada tarea(en nuestro caso las llamamos

“ISSUE”) e ir dando ayuda y feedback en caso de que el trabajo en cuestión lo necesite. En esa documentación con el fin de dar una correcta ayuda se quedará reflejado quien realizó esa tarea y durante qué fechas ya que lo más común es que un integrante del proyecto trabajó en varios aspectos y de esta manera sabremos a quién acudir o no en el caso de que necesitemos ayuda.

Informes de estado

Después de realizar una auditoría y de haber realizado cualquier modificación, revisión o actualización de algún aspecto que tenga que ver con el proyecto se realizará su pertinente documentación reflejando quienes y porque han tomado dichas decisiones. Además si se finaliza alguna parte o se crea alguna nueva tarea también se documentan junto con la persona que estará encargada de la misma. También se documentará cuales pueden ser sus posibles repercusiones en el proyecto, se analizaran sus riesgos y se preparan soluciones por si llegase a suceder. Toda esta información se almacenará en una carpeta de drive con el nombre **INFORMES** en la cual tendremos un documento con toda la información que se haya recopilado junto con la fecha de la auditoría con el fin de tener un seguimiento del desarrollo del proyecto.

8. Apéndices

No se tiene pensado añadir ningún apéndice adicional al documento.