

Informe sobre la Planificación

Proyecto Grupo 7 - "Gualet"

Fecha creación: 06/05/2021

Índice de contenidos

1. Historial de Revisiones	Pág. 3
2. Introducción	Pág 3
3. Descripción general de la Planificación	
3.1 Proceso software	Pág. 4
3.2 Planificación	Pág. 5
3.3 Cambios y adaptaciones	Pág. 7

1. Historial de Revisiones

Fecha	Versión	Autor	Organización	Descripción
6/5/21	0.1	Grupo 7	UMA	Creación del informe de Planificación
1/6/21	1.0	Grupo 7	UMA	Puesta en común y cambios según los altercados de la planificación
8/6/21	1.1	Grupo 7	UMA	Errores corregidos para su perfecta entrega

2. Introducción

El desarrollo de cualquier proyecto o proceso software debe ser ordenado, planificado y moldeable a las distintas circunstancias, claro, esto si lo que se busca es un proceso hecho bajo la filosofía de la ingeniería del software a la cual nos ceñimos por su superioridad y calidad para la creación o mejora de todo proyecto. Por tanto, tener una planificación es fundamental para nosotros.

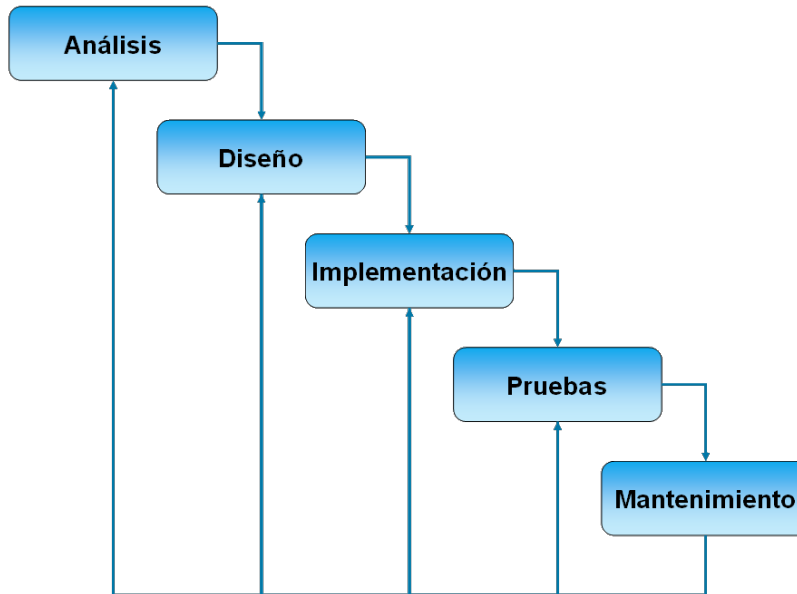
Pressman lo define de la siguiente manera: ***“El objetivo de la planificación del proyecto es proporcionar un marco conceptual que permita al gerente hacer estimaciones razonables de recursos, costo y calendario.”***

Es en ello en lo que a versar este informe, un documento para la explicación y concreción de todas las actividades del proyecto, el proceso software, el enfoque que seguiremos y los distintos cambios o inconvenientes que hayamos sufrido en el camino.

3. Descripción general de la planificación

3.1 Proceso software

Llamamos proceso software a una representación abstracta (simplificada) de un proceso software ∞ Propuestos inicialmente para poner orden el caos del desarrollo de software. Debido a la amplia diversidad de tipos de aplicaciones software Para nuestro proyecto hemos utilizado el modelo de proceso software **Modelo Cascada**.



también conocido como **secuencial** o **ciclo de vida de un programa** (denominado así por la posición de las fases en el desarrollo de esta, que parecen simular la caída de una cascada hacia las siguientes fases), es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase. Este modelo fue el primero en originarse y es la base de todos los demás modelos de ciclo de vida.

En nuestro caso, los requisitos estaban ya definidos desde un principio, lo que hace al modelo secuencial una muy buena opción, dado que no habrá que volver a las etapas anteriores (sin embargo sabemos por la teoría que nunca podemos dar por realizada una etapa, así que esto en un futuro nos dará problemas).

Análisis: en esta etapa se abarca la reunión con el cliente, en donde nos dejará en claro cuales son los requisitos del producto software. Llegando a un acuerdo entre las posibilidades que existen para lo que nos propone.

Diseño: dados todos los requisitos, entramos en la tarea de modelación UML, los diagramas que nos servirán de lienzo para dibujar nuestro software. Serán en esta etapa definidos los siguientes:

- Diagrama de Clases
- Diagrama de Casos de Uso
- Diagramas de Secuencia

Implementación: daremos comienzo luego del diseño a pasar lo “abstracto” a lo “físico” es decir, pasaremos los informes de diagramas y requisitos al código en Java, usando el IDE Eclipse y el repositorio GitHub para la comunicación de ficheros.

Pruebas y Mantenimiento: “finalizado” el proyecto (entre comillas porque un software siempre requerirá de mantenimiento y cuidado durante su evolución) empezaremos las pruebas con ayuda de la librería JUnit, para encontrar aquellos fallos de código o si los requisitos han sido implementados tal y como se especifico en su informe.

Dependiendo del caso, tendremos que volver a alguna etapa anterior y resolver los conflictos ocurridos.

3.2 Planificación

Con ayuda de la plataforma online *Trello* hemos organizado de manera eficiente las distintas actividades que vamos a realizar, cuando lo haremos y quienes.

<https://trello.com/grupo7proyectointroduccionalaingenieriadelsoftware>

El tablero en Trello ha sido dividido en

Hecho

Todas las actividades que han sido realizadas

En curso

Las que se encuentren en plena realización (máximo una semana)

Futuro

Actividades que aún no se han planificado o de futura aprobación

Las actividades en cuestión son:

Actividad	Reuniones por día
➤ Creación del repositorio en GitHub	1 día
➤ Planificación	1-2días
➤ Diagrama de Requisitos	1-2días
➤ Diagrama de Casos de Uso	1-2días
➤ Diagrama de Clases	1-2días
➤ Diagramas de Secuencia	1-2días
➤ Asignación de tareas (programación)	1día
➤ Comenzar la programación	20-30días
➤ Reunión para la presentación del producto	2-3días
➤ Presentación del proyecto	

En cuanto a la programación se asignará de la siguiente manera:

Desarrolladores: Base de Datos, Clases e Interfaz.

Testers: Pruebas unitarias con JUnit y intentar “romper” el código en busca de fallos.

Coordinadores: Ayudaran a los dos anteriores y a poner todo en conjunto.

3.3 Cambios y adaptaciones

Durante la programación (es decir fase de implementación) nos hemos encontrado con el mayor problema: esperar a que otros tengan que terminar sus partes para que los otros puedan continuar.

En especial ocurrió con la implementación de la Base de Datos, donde tuvimos que esperar a que los compañeros asignados a esa parte terminaran de hacerla para poder empezar con la GUI, la cual necesitaba acceder a un fichero de datos.

También destaca el uso de la Programación Extrema, donde cuando un compañero se quedaba atascado, otro se ponía con él y empezaban a buscar los errores hasta solucionarlos. Bastante útil fue su uso, logró avanzar el proyecto más rápido de lo esperado.

Por la fecha en la que se modifica este informe no se ha cambiado nada de lo establecido en los requisitos. Sin embargo ha sido necesario pactar con el cliente y avisarle que dos de los requisitos funcionales no podrán ser, por ahora, implementados estos son el **Asistente Virtual** y **Recordar Usuario**, debido a la complejidad y falta de tiempo en la que nos encontramos. En un futuro estás esperamos que sean implementadas

