

## Capítulo 2

# Planificación

---

La planificación consiste en la estimación del tiempo y esfuerzo durante las fases de desarrollo, con el fin de incrementar los niveles de servicio, calidad y coste.

Para ello, dividimos el ciclo de vida del proyecto en una serie de tareas a las cuales se les asignará las dos estimaciones siguientes:

**Estimación Inicial** empleadas en los inicios del desarrollo. Son las menos exactas, pero se emplean como primera aproximación a la viabilidad del proyecto.

**Estimación Final** expresan la duración y el esfuerzo real empleado, cuyos valores se compararán para evaluar la exactitud de la estimación, empleando el *Error Relativo de la estimación*, RE.

Las estimaciones realizadas en días, considerando una dedicación media de 4 horas al día y una persona, son:

Tarea	Estimación Inicial	Estimación Final	RE
Búsqueda de Documentación	20 días	30 días	33.3 %
Introducción y Planificación	5 días	6 días	16.6 %
Elicitación de Requisitos	15 días	18 días	16.6 %
Análisis de Requisitos	20 días	18 días	-11.1 %
Diseño del Sistema	25 días	20 días	-25 %
Prototipo del Diseño	5 días	5 días	0 %
Instalación de la Plataforma	3 días	2 días	-50 %
Implementación	50 días	62 días	19.5 %
Pruebas	10 días	10 días	0 %
Presentación	2 días	3 días	33.3 %
TOTAL	155 días	174 días	11 %
<i>Error Relativo Medio</i>			3,3 %

Cuadro 2.1: Planificación del proyecto

## Capítulo 3

# Estimación del Coste del Proyecto

---

Existen una serie de métricas propuestas por la Ingeniería del Software para determinar el esfuerzo de un proyecto, el alcance del mismo y la productividad de sus programadores. Vamos a aplicar algunas de las mismas a este desarrollo, para calibrar su dificultad y rendimiento obtenido.

### Las métricas orientadas a tamaño

La métrica del software es un factor realmente importante en el análisis de un proyecto. Las métricas orientadas al tamaño proporcionan medidas directas del software y del proceso por el cual se desarrolla. Se basan en la medición del número de Líneas De Código – LDC - que contiene el desarrollo, entendiendo por línea de código una sentencia del lenguaje de programación (se excluyen comentarios y líneas en blanco de los fuentes). Una forma de clasificarlos es atendiendo al número de líneas de código, como se muestra en la tabla:

Categoría	Programadores	Duración	LDC	Ejemplo
Trivial	1	0 – 4 semanas	< 1k	Utilidad de ordenación
Pequeño	1	1 – 6 meses	1k – 3k	Biblioteca de funciones
Media	2 – 5	0,5 – 2 años	3k – 50k	Compilador de C
Grande	5 – 20	2 – 3 años	50k – 100k	SO pequeño
Muy grande	100 – 1000	4 – 5 años	100k – 1M	Grandes SO
Gigante	1000-5000	5 – 10 años	> 1M	Sistema de Distribución

Cuadro 3.1: Categoría de un proyecto en función de sus líneas de código

La aplicación ha generado más de 16.100 líneas de código, con lo que nos enfrentamos a un proyecto software con una clasificación de complejidad media, para el cual se necesitarían de 2 a 5 programadores trabajando de medio año a 2 años.

## El Método COCOMO

Una metodología que se encarga de medir proyectos software es COCOMO. La metodología COCOMO (CONstructive COst MOdel) se debe a Barry Boehm, y está orientada a líneas de código.

Hay una jerarquía de modelos COCOMO: básico, intermedio y avanzado, la cual se aplica a tres tipos diferentes de software:

**Orgánico** proyectos relativamente sencillos, menores de 50.000 líneas de código. Se tiene experiencia en proyectos similares y se encuentra en un entorno estable.

**Semiacoplado** proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable, y las restricciones intermedias.

**Empotrado** proyectos bastante complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica. Se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Dado que sólo se va a emplear una variable para la estimación (la línea de código), se empleará COCOMO básico, ya que es un modelo univariable estático, con lo que se obtiene una valoración objetiva del esfuerzo realizado. Este proyecto será considerado como software orgánico, ya que posee menos de 50.000 líneas de código.

La ecuación del esfuerzo de COCOMO básico tiene la siguiente forma:

$$E = \text{Esfuerzo} = a \text{ KLDC } b \text{ (persona x mes)}$$

donde KLDC es el número de líneas de código, distribuidas en millares, para el proyecto.

La ecuación del tiempo de desarrollo es:

$$T = \text{Tiempo de duración del desarrollo} = c \text{ Esfuerzo } d \text{ (meses)}$$

Por su parte los coeficientes  $a$ ,  $b$ ,  $c$  y  $d$  se obtienen empíricamente del estudio de una serie de proyectos, y sus valores son:

Proyecto de software	$a$	$b$	$c$	$d$
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3	1,12	2,5	0,35
Empotrado	3,6	1,2	2,5	0,32

Cuadro 3.2: Coeficientes COCOMO

En el desarrollo del proyecto se han codificado 16,1 miles de líneas de código.

$$Esfuerzorealizado = 2,4 * 16,1^{1,05} = 44,4 personas - mes$$

$$T = 2,5 * 44,4^{0,38} = 10,57mes$$

Nº de personas para desarrollar el proyecto = E/T= 44,4 / 10,57 ~ 4,2 personas

Con todo este estudio, debemos concluir que se ha realizado un gran esfuerzo, pues la métrica nos dice que habiendo desarrollado un programa de este tamaño hubiesen sido necesarias cuatro personas, en lugar de solo una como ha sido el caso, para desarrollar el Proyecto en tan solo nueve meses.

Además se puede comparar este desarrollo con el de un compilador, tarea nada desdeñable, y que ha sido el desarrollo de varios alumnos para obtener el título de ingeniero correspondiente al ciclo superior.