

ESTIMACIÓN DE TIEMPOS

Uno de los aspectos clave a la hora de planificar un proyecto, ya sea un nuevo desarrollo o el mantenimiento evolutivo de un sistema ya existente, es la correcta estimación del esfuerzo necesario para llevarlo a cabo. Existen diversas aproximaciones para el cálculo del esfuerzo de un proyecto de desarrollo de software: basadas en puntos de función, basadas en casos de uso, métodos empíricos como COCOMO, etc. En la práctica, la experiencia en proyectos previos de similares características es fundamental y es la que puede determinar la bondad de la estimación (*estimación por analogía*).

El presente documento describe un *modelo de estimación del esfuerzo* que permite **estandarizar y simplificar el mecanismo de estimación** de proyectos software.

Este modelo es el que está aceptado y se está utilizando en el servicio de Aplicaciones del ASIC desde enero de 2013 para el desarrollo de los diferentes proyectos tanto internos como externos, y es el fruto de la experiencia en planificación de proyectos software de los responsables del área.

1. Definición del modelo

El Modelo de Estimación propuesto se basa en los siguientes elementos conceptuales:

- **Componentes Físicos:** Elementos que hay que crear o modificar (Tabla B.D, Procesos, Ventanas Estáticas, Ventanas Dinámica, Informes e Interfaces)
- **Ponderadores:** Valores que permiten afinar las características de los componentes anteriores, en concreto:
 - **Lenguaje:** Tecnología a emplearse (Java, JSP, Oracle...)
 - **Actividad:** Actividad a desarrollar (Diseño o Codificación)
 - **Complejidad:** Estableciendo diferentes grados para cada componente y tecnología (Simple, Media, Compleja)

A continuación definimos cada uno de los componentes, independientemente del lenguaje o aplicación, así como el alcance y la complejidad de cada uno de ellos.

1.1 Componentes y Ponderadores

Tabla Base de Datos

Descripción: Tablas sobre las que se requiere almacenar información, de las cuales el usuario final tiene constancia. Representa un grupo de datos lógicamente relacionado.

Alcance: Se contabilizan todas aquellas Tablas que sean creada o modificadas en su estructura (eliminación o creación de nuevos atributos). *En fase de análisis no se ha definido la estructura de la base de datos, en su lugar, la estimación puede realizarse sobre las clases del modelo de clases UML.*

Complejidad: La complejidad se obtendrá en función del número de relaciones y atributos:

- Simple: Hasta 3 relaciones y menos de 25 atributos
- Media: Hasta 5 relaciones e igual o más de 25 atributos
- Compleja: Más de 5 relaciones e igual o más de 25 atributos.

Ventana Dinámica

Descripción: Interfaces de usuario que para su construcción se requiera recoger datos del Sistema. Estas interfaces presentarán interacción con el usuario.

Alcance: Se contabilizan todas aquellas Ventanas de nueva construcción o aquellas que se ven modificadas por la inclusión/eliminación de campos, cambio de la lógica interna de procesamiento o inclusión/eliminación de las acciones a realizar sobre los datos. No se incluirá en esta estimación el coste del diseño e implementación del modelo de datos subyacente.

Complejidad: La complejidad se obtendrá en función de campos y validaciones a realizar:

- Simple: 1-8 campos sin seguridad y/o validaciones 1-5 (campos obligatorios)
- Media:
 - ✓ 1-8 campos con seguridad y/o 6-10 validaciones
 - ✓ 9-15 campos sin seguridad y 1-5 validaciones (campos obligatorios)
- Compleja:
 - ✓ 9-15 campos con seguridad y/o >11 validaciones complejas
 - ✓ 16-20 campos sin seguridad y 1-5 validaciones (campos obligatorios)

Informes

Descripción: Se incluyen los siguientes objetos asociados.

- Tablas para la generación de informes
- Proceso de extracción de datos y cálculo y actualización asociado
- Soporte gráfico: salida concreta, además de la propia pantalla: (xls, pdf, txt, etc.)
- Interfaz de lanzamiento

Si cualquiera de los objetos asociados al informe tuviera una complejidad que justifique su gestión individualizada, se deberá contemplar la posibilidad de realizar una estimación por separado de dicho componente.

Alcance: Se contabilizan aquellos informes de nueva creación así como aquellos que vean alterados los campos a mostrar, la lógica de procesamiento o Entidades accedidas.

Complejidad: La complejidad vendrá definida por el número de salidas de texto y columnas:

- Simple:
 - ✓ 1-8 columnas de salida, con tipo de salida TXT
 - ✓ 1-5 columnas de salida, con otros tipos de salida distintos a TXT.
- Media:
 - ✓ 9-15 columnas de salida, con tipo de salida TXT
 - ✓ 6-12 columnas de salida, con otros tipos de salida distintos a TXT
- Compleja:
 - ✓ 16-30 columnas de salida, con tipo de salida TXT
 - ✓ 13-25 columnas de salida, con otros tipos de salida distintos a TXT

Procesos

Descripción: Funcionalidades desde el punto de vista del usuario a desarrollar derivadas de procesos Batch, procesos de control o cargas iniciales de Base de Datos, que no se traducen en procesos on-line.

Alcance: Se contabilizarán aquellos Procesos de nueva creación así como aquellos que sean modificados en la lógica de procesamiento, almacenes referidos, inclusión/eliminación/modificación de los flujos de datos entrantes o salientes.

Complejidad: La complejidad se obtendrá en función del número de almacenes y flujos:

- Simple: 0 almacenes y hasta 4 flujos
- Media:
 - ✓ 0 almacenes y 5 o 6 flujos
 - ✓ 1 almacén y 3 a 5 flujos
 - ✓ 2 almacenes y hasta 4 flujos
- Compleja:
 - ✓ 0 almacenes y más de 6 flujos
 - ✓ 1 almacén y más de 5 flujos
 - ✓ 2 almacenes y más de 4 flujos
 - ✓ Más de 2 almacenes

Interfaces

Descripción: Interfaces con otros sistemas, considerándose los procesos de envío de datos desde un sistema origen a otro destino.

Alcance: Se contabilizarán aquellos Procesos de nueva creación así como aquellos que sean modificados en la lógica de procesamiento, almacenes referidos, inclusión/eliminación/modificación de los flujos de datos entrantes o salientes.

Complejidad: La complejidad se obtiene en función del número de entidades referidas y el número de campos implicados

- Simple: Hasta 2 entidades y menos de 50 campos
- Media: Hasta 5 entidades o hasta 2 entidades y más de 50 campos
- Compleja: 6 o más entidades

1.2 Actividades durante el ciclo de vida

Para la Línea de Mantenimiento Evolutivo y Desarrollos Nuevos se define el siguiente ciclo de vida con cada una de las tareas específicas de cada fase:

Fase de Análisis (requerimientos y evaluación)
<ul style="list-style-type: none"> • Análisis de Requisitos
Fases de Diseño+Codificación (incluye pruebas Unitarias)
<ul style="list-style-type: none"> • Diseño técnico • Modelo físico de datos • Realización de Planes de Pruebas • Desarrollo Software • Pruebas Unitarias
Fase de Pruebas (pruebas integradas de desarrollo)
<ul style="list-style-type: none"> • Ejecución y documentación de los planes de prueba
Documentación Desarrollo
<ul style="list-style-type: none"> • Confección / actualización de manuales
Fase de Implantación
<ul style="list-style-type: none"> • Soporte implantación en Preproducción
<ul style="list-style-type: none"> • Soporte implantación en Producción

2. Cumplimentación de la hoja de estimación

Se deberán tener las siguientes consideraciones a la hora de cumplimentar la hoja de estimación de esfuerzo:

2.1 En relación con la actividad

- **Actividad Diseño, incluye:**
 - Diseño Técnico: arquitectura, interfaces, componentes, ...
 - Modelo físico de datos.
 - Diseño de planes de pruebas.
- **Actividad Codificación, incluye:**
 - Desarrollo software.
 - Pruebas unitarias.
 - Pruebas integradas desarrollo.
- **El resto de actividades del ciclo de vida:** no se consideran en las tablas de estimación del esfuerzo, pero tienen que contabilizarse. Se considerará la siguiente relación de esfuerzo entre las actividades principales del ciclo de vida.

Análisis	15%
Diseño	15%
Codificación	35%
Pruebas	20%
Documentación	10%
Implantación	5%

Tabla 1: Relación porcentual de esfuerzo entre las actividades del ciclo de vida

Es decir, una vez calculado el esfuerzo en Diseño y Codificación se aplicará un 15% para calcular el esfuerzo de planificación y análisis, un 10% para calcular el esfuerzo de documentar el proyecto y un 5% para su implantación.

2.2 En relación con la complejidad

Para determinar la complejidad de la tarea a estimar, tanto en diseño como en codificación, se deberá tener en cuenta la descripción del **punto 1.1** de cada uno de los componentes.

2.3 En relación con la tecnología

La hoja (**ver tabla 2**) de estimación de esfuerzo recoge las horas previstas, tanto de diseño como de codificación, para una tecnología determinada para un programador medio (programador *senior* con tres años de experiencia). En caso de que la tecnología a utilizar no esté incluida en la tabla puede aproximarse a la más parecida. Si el equipo de desarrollo no tiene experiencia en la tecnología debería aplicarse un factor corrector.

DESARROLLO			
LENGUAJE	ACTIVIDAD	DIFICULTAD	HH
Java	Diseño	Simple	1
Java	Diseño	Media	2
Java	Diseño	Compleja	3
Java	Codificación	Simple	4
Java	Codificación	Media	5
Java	Codificación	Compleja	8
JSF/JSP/XHTML	Diseño	Simple	1
JSF/JSP/XHTML	Diseño	Media	2
JSF/JSP/XHTML	Diseño	Compleja	3
JSF/JSP/XHTML	Codificación	Simple	4
JSF/JSP/XHTML	Codificación	Media	6
JSF/JSP/XHTML	Codificación	Compleja	8
Oracle	Diseño	Simple	1
Oracle	Diseño	Media	2
Oracle	Diseño	Compleja	3
Oracle	Codificación	Simple	3
Oracle	Codificación	Media	5
Oracle	Codificación	Compleja	7

Tabla 2: HOJA DE ESTIMACIÓN DE ESFUERZO (Horas-Hombre) POR LENGUAJE, ACTIVIDAD Y DIFICULTAD

3. Ejemplo completo de estimación de esfuerzo

Para estimar el esfuerzo en diseño y codificación, previamente debe tenerse una idea aproximada de:

- **Los componentes:** que implementarán un conjunto de los requisitos funcionales identificados en la declaración de alcance o de casos de uso (**ver punto 1.1**).
- **Un diseño preliminar (prototipado):** de las interfaces de los formularios de cada módulo dará una idea de la complejidad de estos.

Seguidamente vamos a ver la secuencia de pasos a realizar para aplicar el modelo de estimación propuesto en los apartados anteriores sobre un caso de estudio. En primer lugar describiremos someramente el caso de estudio.

3.1 Caso de estudio

Supongamos que vamos a implementar un **servicio de llamadas de emergencia** para atenciones hospitalarias.

- Entre las muchas funcionalidades que se han identificado se encuentra **“nueva emergencia”**. Esta gestión deberá realizarse por parte de los “operadores telefónicos” cada vez que una persona (“paciente”) llame al servicio con una emergencia. Esta funcionalidad requiere:

- ✓ **Comprobar si el paciente está registrado en el sistema (“Busca Paciente”)**: En caso contrario se deberá...
- ✓ **Dar de alta el paciente (“Alta Paciente”)**: en caso de que no esté registrado en el sistema, y...
- ✓ **Crear un nuevo registro de emergencia**: Se completa “Nueva Emergencia” introduciendo los síntomas del paciente y las especialidades a las que pertenecen. El sistema automáticamente asignará la ambulancia y el hospital de destino que minimicen el tiempo de traslado del paciente.

La parte del diagrama de casos de uso que modela esta funcionalidad se muestra en la **Figura 1**.

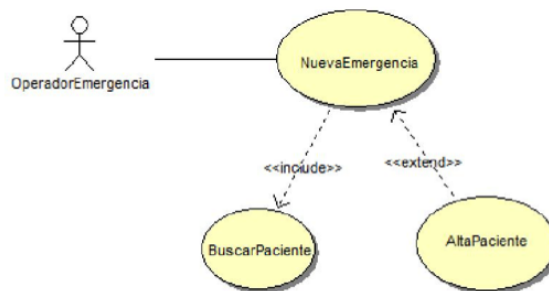


Figura 1. Caso de uso Nueva Emergencia

El modelo conceptual de datos se muestra en la **Figura 2**. Es necesario tener una idea de las entidades involucradas y las relaciones entre ellas para hacer una estimación de la complejidad.

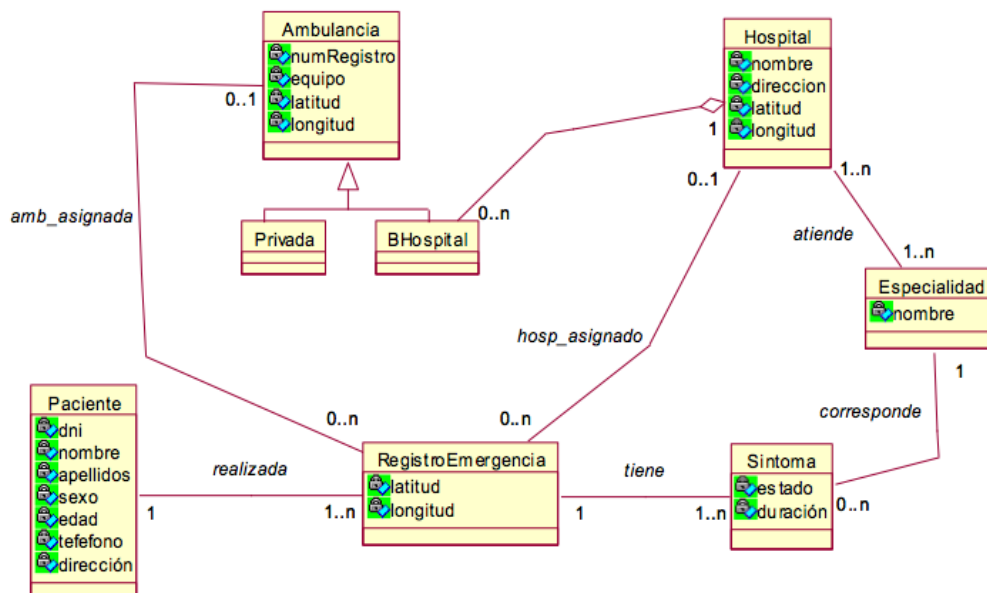
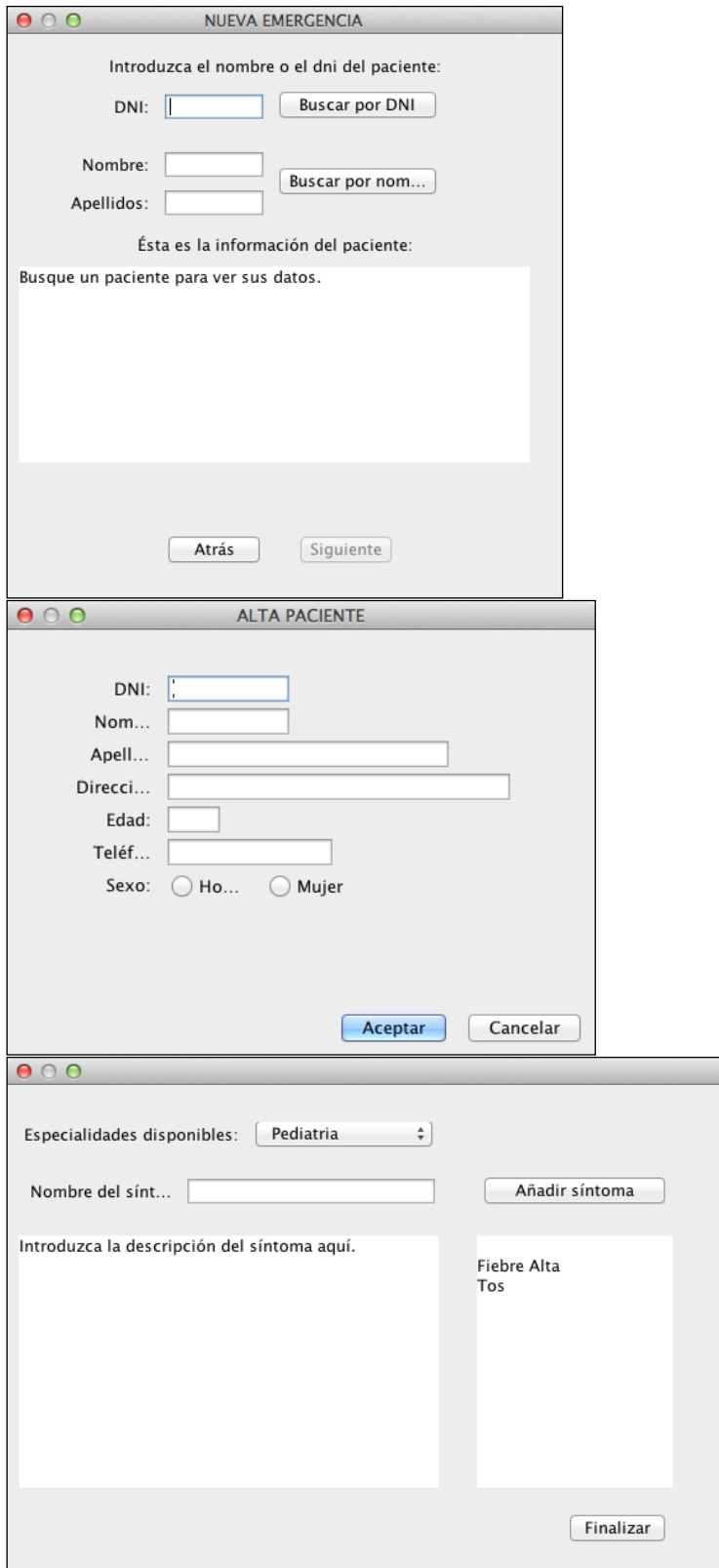


Figura 2. Modelo de clases del Servicio de Llamadas de Emergencias

Se ha realizado un diseño previo de los formularios que se muestra en la **Figura 3**: el operador introduce el identificador del paciente (dni o nombre), si no existe se abre el formulario Alta

Paciente para registrar al paciente en el sistema. A continuación se recogen los datos sobre los síntomas que tiene el paciente, que se relacionan con una especialidad médica. Al pulsar Finalizar el sistema asigna automáticamente una ambulancia y un hospital de destino.



The image displays three overlapping screenshots of a medical emergency management software interface.

NUEVA EMERGENCIA

Introduzca el nombre o el dni del paciente:

DNI:

Nombre:

Apellidos:

Ésta es la información del paciente:

Busque un paciente para ver sus datos.

ALTA PACIENTE

DNI:

Nom...

Apell...

Direcci...

Edad:

Teléf...

Sexo: ☐ Ho... ☐ Mujer

Especialidades disponibles:

Nombre del sínt...

Introduzca la descripción del síntoma aquí.

Fiebre Alta
Tos

Figura 3. Formularios del caso de uso Nueva Emergencia

Los **componentes** a tener en cuenta en el desarrollo del módulo serán, según los elementos enumerados en el apartado 1.1:

- **Tablas de la base de datos:** Se modificarán las tablas Paciente, Registro Emergencia y Síntoma. El número de relaciones es de 5, y el número de atributos inferior a 25. Según las directrices del apartado 1.1, **la complejidad será MEDIA.**
- **Ventana dinámica:** Se consideran todos los formularios asociados al caso de uso o componente. Es posible que alguno de ellos se reutilice posteriormente. Por ejemplo, el formulario Alta Paciente podría formar parte de otro caso de uso del sistema. En ese caso, el esfuerzo se computará únicamente una vez.

En los formularios hay un total de 15 campos o controles gráficos. Se consideran tanto:

- ✓ los que recogen datos del usuario (entrada)
- ✓ como los que visualizan información (salida).

Al menos los siguientes campos requieren de algún tipo de validación:

- ✓ En Nueva Emergencia hay que verificar si DNI o Nombre y Apellidos existen en la base de datos.
- ✓ En Alta Paciente hay que verificar que el DNI no esté ya registrado, y los otros 6 campos pueden considerarse obligatorios, con validaciones de formato.
- ✓ En la última ventana, debe cargarse la lista de especialidades y comprobar que se introduce al menos un síntoma. Hacen un total de 12 campos sobre los que se realiza algún tipo de validación.

Según las directrices del apartado 1.1, **la complejidad sería COMPLEJA.**

Supongamos que la parte de la aplicación correspondiente al operador es una aplicación de escritorio, diseñada con una arquitectura multicapa. Se va a desarrollar en Java, y el servidor de base de datos es Oracle. La **Tabla 3** muestra el cálculo del esfuerzo según las consideraciones previas, aplicando la información referente al modelo de estimación de la **Tabla 2** (ver apartado 2.3).

Requisito	Componente	Tecnología	Complejidad	HH Diseño	HH Codificación
R1.1. Nueva Emergencia	Base de Datos	Oracle	Media	2	5
R1.1. Nueva Emergencia	Ventana dinámica	Java	Compleja	3	8
			Total HH:	18 HH	

Tabla 3. Estimación de esfuerzo para el caso de uso Nueva Emergencia (Diseño + Codificación)

Como se observa el esfuerzo resultante es de 18 Horas-Hombre. Para calcular el esfuerzo total del proyecto, una vez estimado el esfuerzo de cada componente del sistema, hay que sumarle

el esfuerzo necesario para las tareas de análisis de requisitos, documentación e implantación (véase **Tabla 1** del apartado 2.1). Suponiendo que solamente se fuera a desarrollar este componente el esfuerzo final sería el indicado en la **Tabla 4**.

ACTIVIDAD	%	HH	JORNADAS (7h30m)
Requerimientos y Evaluación	15%	3,85	0,51
Diseño+ Codificación + Pruebas	70%	18,00	2,40
Documentación Desarrollo	10%	2,57	0,34
Implantación	5%	1,29	0,17
TOTALES	100%	25,71	3,42

Tabla 4. Estimación de esfuerzo total del proyecto

4. ¿Qué tengo que preparar?

El objetivo de esta práctica consiste en realizar una estimación del esfuerzo del proyecto **del caso de estudio**, utilizando la metodología descrita en los apartados anteriores. Para la realización de este trabajo necesitaremos analizar los prototipos desarrollados en la sesión anterior. El trabajo a realizar puede dividirse en las siguientes partes:

1. Dividir el trabajo en los diferentes módulos de los que consta **el caso de estudio**.
2. Hacer un listado de las diferentes funciones o casos de uso de los que consta cada módulo.
3. Utilizando los prototipos desarrollados en la sesión anterior, determinar los componentes (tablas de datos, ventana dinámica, informes, etc.) que será necesario utilizar para cada funcionalidad requerida, así como su complejidad (ver apartado 1.1).
4. Recopilamos esta información, y utilizando la **Tabla 2** (véase apartado 2.3) creamos tablas con los esfuerzos (H-H) de Diseño y Codificación para cada una de las funcionalidades del caso de estudio. Estas tablas tendrán una estructura similar a la de la **Tabla 3** (véase apartado 3), pero replicadas para los diferentes casos de uso o requisitos funcionales identificados.
5. Sumaremos todos estos esfuerzos para determinar el esfuerzo total de Diseño+Codificación+pruebas para todo el proyecto. Después utilizaremos la información del apartado 2.1 para extrapolar los esfuerzos del resto de fases del proyecto (análisis, documentación, implantación, etc.), y determinar el esfuerzo total del proyecto. Toda esta información quedará reflejada en una tabla similar a la **Tabla 4** (véase apartado 3).

Toda esta información se recogerá en un documento **“Estimación de esfuerzo”** que se añadirá a la declaración de Alcance.

Los criterios que se tendrán en cuenta para evaluar esta parte de la declaración de alcance serán los siguientes:

- Se ha estimado correctamente la complejidad de los diferentes componentes utilizados, utilizando los criterios del apartado 1.1.
- Se ha estimado el esfuerzo H-H de desarrollo (diseño, codificación y pruebas) por cada componente que implementa un caso de uso definido en la declaración de alcance.
- Se han tenido en cuenta en la estimación todos los requisitos funcionales expuestos en el encargo del proyecto.
- Se ha estimado el esfuerzo global de dedicación al proyecto en H-H, aplicando correctamente los porcentajes para todas las fases del ciclo de vida.