



ISBN: 978-987-1896-01-1

Facultad Regional Tucumán
Universidad Tecnológica Nacional – U.T.N.
Argentina

2012

Editorial de la Universidad Tecnológica Nacional – edUTecNe

<http://www.edutecne.utn.edu.ar>

edutecne@utn.edu.ar

© [Copyright] La Editorial de la U.T.N. recuerda que las obras publicadas en su sitio web son *de libre acceso para fines académicos y como un medio de difundir el conocimiento generado por autores universitarios*, pero que los mismos y edUTecNe se reservan el derecho de autoría a todos los fines que correspondan.

A mis hijos, Ulises y Gabriel, fuente inagotable de inspiración constante.

Muchas gracias por ser parte de mi vida.

Gustavo Gabriel Maigua

Para que mi hijo lo asimile y promueva en el futuro.

Emmanuel Fernando López

Contenido

PREFACIO

CAPÍTULO I. INTRODUCCIÓN A LA DIRECCIÓN DE PROYECTOS

CAPÍTULO II. FASES DE UN PROYECTO

CAPÍTULO III. FASE DE PLANEACIÓN

CAPÍTULO IV. FASE PRODUCTIVA

CAPÍTULO V. GESTIÓN DE RIESGOS

CAPÍTULO VI. RECURSOS HUMANOS

CAPÍTULO VII. REDMINE, SOFTWARE LIBRE DE GESTIÓN DE PROYECTOS

CONCLUSIONES

NOTAS

BIBLIOGRAFÍA Y REFERENCIAS WEB

ANEXO I: Casos De Éxito

ANEXO II: Herramientas De Financiamiento De Proyectos

ANEXO III: Índice De Figuras

PRÓLOGO

Ante la demanda mundial de formación de Ingenieros en todas sus especialidades y en particular los ingenieros orientados a la informática, existe un problema más, que es la necesidad de formar (preparar) líderes de equipos en cualquier Dirección de Gestión de Proyectos Informáticos.

Esto implica que no tan solo se necesita al ingeniero formado en la especialidad para ocupar un puesto de trabajo sino que también se necesita a un ingeniero que sepa liderar y/o trabajar en equipo para alcanzar el Objetivo que se plantee en una Organización.

Este libro busca llegar mediante una lectura fácil, con los mecanismos y pasos que debe seguir el líder, no tan solo de la Dirección de Gestión Informática, sino también el líder de grupos por debajo de la Dirección.

Es evidente esta falta de formación y es por ello que las Universidades están apuntando en sus diseños curriculares, la enseñanza de emprendedurismo, trabajo en equipo, planes de negocios, formación en Recursos Humanos, etc.

En el seno del Consejo Mundial de Decanos de Ingeniería, (Global Engineering Dean Council) uno de los debates fue precisamente este tema. Y la conclusión a la que se llegó después de un importante intercambio de ideas fue que, además de saber identificar todos los parámetros que puedan surgir en el desarrollo de un Proyecto, los Líderes deben tener una fuerte dosis de INNOVACION para alcanzar el Objetivo.

Es por ello que resulta necesario que los egresados adquieran estos conocimientos durante la formación académica y un factor, no menor es, la capacitación de los docentes en estos temas actuales.

Estimo entonces que este libro puede ser de gran ayuda no tan solo para alumnos, ingenieros o estudiantes avanzados que requieran esta técnica de trabajo sino también como lectura de referencia para profesores que deseen involucrarse en estos conceptos.

Por último, quiero felicitar a los jóvenes autores, a quienes tuve la oportunidad de ser su profesor y conocerlos más de cerca, por el nivel de compromiso y profesionalismo que significó el desafío de escribir este Libro. Los éxitos no llegan por azar, uno los construye día a día a partir del momento que "elige" trabajar sobre el mismo.

Ing. Walter Fabián Soria

Decano de la Facultad Regional Tucumán de la Universidad Tecnológica Nacional
Miembro del Comité Ejecutivo del Consejo Mundial de Decanos de Ingeniería

PREFACIO

Objetivos

Se intentó escribir un libro de fácil lectura en el que el lector va a encontrar definiciones sencillas pero concisas sobre los aspectos más importantes de la dirección y gestión de proyectos informáticos.

Hemos tomado como documentación de referencia desde estándares técnicos reconocidos como PMI (PMBOK) o Métrica v3, hasta principios universales de liderazgo y conducción de equipos.

Esta obra incluye un caso de estudio de ejemplo para que el lector tenga una referencia práctica sobre los diferentes conceptos teóricos que se van desarrollando.

Esperamos que este libro sea una gran ayuda para quienes deseen incorporar herramientas efectivas en la dirección y gestión de proyectos informáticos.

¿A Quién Se Dirige Este Libro?

La obra *Buenas Prácticas en la Dirección y Gestión de Proyectos Informáticos* va dirigida a todos aquellos que pretendan introducirse en el arte de la Dirección y Gestión de Proyectos Informáticos, así como para todos aquellos profesionales con experiencia que necesiten un curso de actualización de conocimientos. Está especialmente recomendado por sus autores para:

- Estudiantes.
- Tecnólogos.
- Directores de Proyectos.
- Responsables de Proyectos con mayor o menos responsabilidad en los mismos.
- *Staff* en Project Management.
- Y en general entusiastas de la Dirección y Gestión de Proyectos Informáticos con una formación limitada en gestión de proyectos o que empiecen de cero.

Organización Del Libro

Los contenidos de este libro fueron divididos en siete capítulos que van introduciendo paulatinamente al lector en la Dirección de Proyectos Informáticos. Se agregó un caso de estudio para explicar de manera práctica los contenidos de los capítulos III, IV y V.

En *capítulo I* nos centramos en la definición de conceptos básicos que le servirán al lector para abordar la lectura de los restantes capítulos del libro. En este mismo capítulo se realiza un análisis sobre la necesidad de la dirección y gestión de proyectos fundamentado en conceptos teóricos y experiencia de los autores.

En el *capítulo II*, se definen las principales fases de un proyecto estándar como Planeación, Ejecución y Mantenimiento. Las mismas son explicadas en un alto nivel de abstracción para que el lector pueda diferenciar entre una y otra fase.

En el *capítulo III* se desarrolla a un nivel más detallado la Fase de Planeación. Se incluye la concepción inicial, la definición del problema y la planificación del proyecto. Como resultado final de esta fase se obtendrá el estudio de viabilidad que permitirá determinar si un proyecto es factible o no. También se agregó, al finalizar este capítulo, una breve explicación sobre los Contratos Informáticos.

En el *capítulo IV* se desarrolla la Fase Productiva. En la primera parte de este capítulo se divide el contenido en las etapas de Análisis, Diseño, Implementación y Pruebas. En este capítulo el lector podrá encontrar detalles puntuales que le permitirán una ejecución, seguimiento y control más preciso para dirigir un proyecto informático. En la segunda parte de este capítulo se desarrolla la Fase de Mantenimiento del proyecto. Esta fase representa el proceso de mejora y optimización del software después de su entrega al usuario final, así como también la corrección y prevención de los defectos. Se define claramente los diferentes tipos de mantenimiento y su distribución del coste.

Luego de hacer una descripción detallada de las principales Fases de un Proyecto, en el *capítulo V* se describen los detalles más importantes de la Gestión de Riesgo y los principales beneficios de realizar un control y seguimientos de los eventos que pueden afectar el alcance de los objetivos del proyecto.

En el *capítulo VI*, detallamos las principales características de un equipo de personas que llevaran a cabo un proyecto informático. Es sabido que un Director de Proyecto tiene a su equipo como herramienta principal para el éxito, por eso se detallan las principales características y habilidades requeridas en el recurso humano involucrado en un proyecto informático, tanto desde el punto de vista técnico como así también del emocional.

Finalmente, en el *capítulo VII*, se agregó documentación de una herramienta libre para la gestión de proyectos llamada Redmine. Se considero oportuno comentar a los lectores sobre las potencialidades que le pueda dar una herramienta informática de gestión a un proyecto informático y Redmine es un buen ejemplo para compartir.

Como elementos extras a los contenidos, se agrego documentación importante en forma de *Anexos* que brindaran mayor detalle de algunos temas significativos referidos a los contenidos del libro.

Convenciones Utilizadas

Al final de los capítulos se presentará un depurado conjunto de conclusiones acerca de las Buenas Prácticas en la Dirección y Gestión de Proyectos presentadas.

Además, en este manual se añade un formato de notas especiales a modo de explicación y de resalte de la información e ideas más importantes:



IMPORTANTE

Es importante resaltar que podremos encontrar estos cuadros a lo largo del libro. Pueden aparecer resaltando puntos importantes de aprendizaje; con información complementaria; o destacando ciertas técnicas o recomendaciones útiles.

Agradecimientos

Al **Ing. Gustavo Rego** por sus aportes en el contenido teórico, al **Ing. Daniel Talebi** por los casos de estudios referenciados y al **Ing. Joaquín Igon** por la información provista sobre las líneas de financiamiento de proyecto.

CAPÍTULO I. INTRODUCCIÓN A LA DIRECCIÓN DE PROYECTOS

“Cristaliza tus metas. Elabora un plan para alcanzarlas. Fíjate una fecha límite. Entonces, con suprema confianza, lleva adelante tu proyecto”. (Paul J. Mayer)

1.1 ¿Por qué es necesaria la dirección y gestión de proyectos?

1.1.1 Lo único constante es el cambio

Durante la última década, la informática en su conjunto se ha ido simplificando, sin embargo, los entornos de desarrollo han seguido un camino contrario y son cada día más complejos. Hace algún tiempo, un desarrollo típico estaba formado por un par de docenas de transacciones, algunos utilitarios y poco más. Todo era sencillo porque había pocas cosas donde elegir, casi siempre la elección dependía del propio suministrador del hardware y tales plataformas de desarrollo eran estables y no se producían cambios durante algún tiempo importante. El trabajo era realizado de manera *“artesanal”* por un programador, que interactuaba directamente con el cliente, y obtenía por el mismo la retroalimentación para medir la calidad.

Hoy en día todo esto ha cambiado; las plataformas de desarrollo proliferan y ofrecen nuevas posibilidades que hacen que la oferta sea más atractiva y se cambia de entornos en función de las nuevas posibilidades que los mismos ofrecen. Los usuarios son mucho más exigentes, demandan unas prestaciones antiguamente no soñadas, desde nuevos dispositivos móviles, y esto hace que en muchas ocasiones el diseño gráfico y su interfaz genere más trabajo de programación que los propios algoritmos para los que se concibe el programa. La necesidad de desarrollar un software normalmente multiplataforma dificulta, no sólo el propio desarrollo sino también las pruebas de aceptación del mismo. Hoy no basta con que una aplicación funcione, sino que debe funcionar en diferentes sistemas operativos y bajo diferentes condiciones.

Como si no tuviéramos demasiado con tener que solucionar los problemas de hoy, aparecen las preguntas sobre la tecnología del mañana: ¿Qué sistemas tendremos dentro de unos años?, ¿cómo conseguir un entorno de desarrollo del que se pueda migrar fácilmente al acontecer futuro?, ¿Qué haremos con las aplicaciones existentes en un futuro?, ¿Serán obsoletas?. Estas preguntas que nos formulamos ahora mismo deberán ser consideradas por el director del proyecto para conocer el grado de realización del software con vistas a un mantenimiento futuro.

1.1.2 Evolución histórica del concepto de calidad en la industria del software

La industria del software es una industria joven que ha evolucionado rápidamente, aunque no tanto como para alcanzar la madurez que tiene la industria tradicional. Es importante recordar que es evidente la influencia de la industria tradicional sobre la industria del software. La siguiente tabla propuesta por Marciniak, pone de manifiesto esta relación, comparando las diferentes fases superadas por el control de calidad:

Etapa	Descripción	Industria del Software	Industria en general
Artisanos	Se fían de la creatividad y del buen trabajo artesanal	Años sesenta	Antes del siglo XIX
Inspección	Supervisores inspeccionan la calidad antes de la liberación de producto	Años setenta	Siglo XIX
Control estadístico del proceso	Cuantificación de la calidad del producto, técnicas de muestreo	Pocas evidencias de uso	Años treinta
Aseguramiento de la calidad	Uso de estándares en los sistemas de calidad para los procesos	Años ochenta	Años cincuenta
Conformidad con la calidad	Calidad total: se eliminan derroches y minimizan costes	Años noventa	Años ochenta
Calidad dirigida al cliente	Calidad total dirigida hacia el cuidado del cliente y del servicio	Pocas evidencias de uso	Años noventa
Calidad dirigida al mercado	Calidad total dirigida hacia el cliente existente así como a clientes en potencia	Pocas evidencias de uso	Algunas evidencias de uso

Figura 1. Evolución histórica del concepto de calidad en la industria del software

1.1.3 Tomar el timón del barco

Podríamos comparar un proyecto software con un barco que está zarpando. Si no se han planificado bien los recursos previamente, será muy difícil que se pueda llegar a destino. Se deben detectar y organizar todas las tareas. A cada tarea se le debe asignar recursos materiales y humanos para que pueda ser ejecutada en un determinado periodo, considerando siempre un uso eficiente de los recursos. En el caso de que la planificación haya sido correcta y el proyecto comenzó (el barco ha zarpado), será muy importante un control y seguimiento continuo de los recursos humanos y materiales en el transcurso del tiempo. Una mala administración de los mismos podrá hacer que los recursos se acaben antes de llegar a destino. Esto dejará al barco (proyecto) varado en medio del océano. Existen diferentes técnicas para hacer el seguimiento de un proyecto y por lo tanto es importante que el capitán del barco, es decir el director del proyecto, las conozca y las aplique. Generalmente estas técnicas son útiles y necesarias, casi tanto como una brújula en el mar.

En el mejor caso en que la planificación, el seguimiento y el control sean ejecutados efectivamente, nada nos libra de que una noche cualquiera nos llegue una tormenta que pueda desestabilizar el viaje y el barco se hunda, es decir que el proyecto pueda fracasar. Estas

podrían ser todas las variables externas que podrían afectar el proyecto: como un aumento inflacionario, una huelga general, aumento excesivo en los precios de producto, etc. En este caso se deberá tener un buen análisis de riesgo para actuar ante cualquier evento con un plan de contingencia apropiado.

Es importante que un director de proyectos informáticos incorpore herramientas gerenciales para poder planificar y ejecutar el proyecto de una manera profesional.

1.1.4 Demanda laboral en Argentina

El "mundo" tecnológico en la Argentina ya emplea casi a la misma cantidad de personas que la industria automotriz.

El fuerte crecimiento que el sector informático experimentó en los últimos años lo posicionó "cabeza a cabeza" con la industria automotriz en cuanto a la creación de empleos registrados.

Así, mientras la industria automotriz (entre terminales y autopartistas) cuenta con 77.362 empleos registrados, las actividades de Informática (desarrollo e implementación de software, consultoría, suministros de programas, servicios relacionados con bases de datos, procesamiento de datos y servicios de soporte, mantenimiento y reparación) les dan trabajo a unas 76.500 personas, señalan los datos del Ministerio de Trabajo al primer trimestre de 2010.

Hasta noviembre del 2010, la incorporación de perfiles "tecnológicos" no paró de crecer. La búsqueda de profesionales en el área de IT se incrementó entre un 40 y un 50% entre enero y octubre del 2010 en comparación con el mismo período de 2009, según estimaciones de la consultora Michael Page International Argentina.

La contracara del fenómeno

Sin embargo, la contracara es la falta de recursos calificados para cubrir los diferentes puestos que requieren las empresas, ya sean multinacionales o Pymes. El siguiente gráfico detalla que el grueso de los empleados de esta industria cuenta con un nivel universitario completo (38%) o incompleto (31%).



Figura 2. Estructura ocupacional real

Los más buscados

En la actualidad, las empresas buscan profesionales no sólo orientados a lo técnico y operativo, sino que también estén enfocados al negocio, con la habilidad de vincular conocimientos especializados con aspectos referentes al presupuesto y la perspectiva de crecimiento económico de una compañía.

Según (Michael Page), el perfil más solicitado en este momento es el de los especialistas que ocupen cargos gerenciales en el área de IT; y por eso es tan importante recibir formación en la disciplina de la Dirección y Gestión de Proyectos.

1.1.5 Formalizando aptitudes gerenciales a Informáticos

Se podría decir que, actualmente, casi **no existe una mesa de decisión donde no participe un experto en informática** como miembro estratégico en la implementación de nuevas tecnologías para obtener un crecimiento en la organización. Esta forma sistemática de trabajar exige el establecimiento de un conjunto de métodos, procedimientos, herramientas y técnicas que hagan posible la realización del proyecto según las etapas y actividades previstas y faciliten su control por parte del responsable del mismo.

Hoy en día dirigir un proyecto de software exige mucho más que conocimientos técnicos. Se necesita tener capacidades de planificación y liderazgo. Entre las cualidades que se deberían buscar para el Director del Proyecto caben destacar las siguientes: mente estructurada y lógica, liderazgo y aceptación por el grupo de trabajo, conocimiento del sector de la actividad del proyecto, y madurez y formación específica en aspectos gerenciales (dirección por objetivos).

Las universidades del primer mundo han comenzado a ver la necesidad de proveer herramientas gerenciales a los directores de proyectos informáticos, ya que esta rama de la industria viene generando numerosos puestos de trabajo. Esto ha comenzado verse plasmado en nuevas ofertas de postgrado de máster en Dirección de Proyectos Informáticos.

1.1.6 Necesidad de la dirección y gestión del software

Podría decirse que la industria del software ha crecido de manera exponencial en los últimos años. Este crecimiento se dio también con la cantidad de puestos de trabajo generados. Los roles fueron cambiando durante este crecimiento desde un producto artesanal a un producto profesional. El “artesano” informático se convirtió en Director de Proyectos, y ahora tiene la responsabilidad de adquirir aptitudes gerenciales para afrontar los desafíos futuros.

1.2 Conceptos básicos

1.2.1 Proyecto

1.2.1.1 Definición

Un **proyecto** es un esfuerzo temporal acometido para crear un único servicio o producto.

“Conjunto de actividades planificadas, ejecutadas y supervisadas con el fin de alcanzar un fin común con recursos finitos”

Hay que recordar que uno de los recursos finitos más importante es el tiempo. La naturaleza temporal de los proyectos indica un principio y un final definidos. El final se alcanza cuando se logran los objetivos del proyecto o cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto.

1.2.1.2 Características de un proyecto

Es imprescindible, para llevar a cabo un proyecto con éxito, discernir claramente un objetivo a cumplir. Para ello es necesaria la intervención de personas especialistas que se encarguen de desarrollar cada una de las fases de las que consta.

1.2.1.3 Tipos de proyectos

Los proyectos pueden ser de diversa índole, existen múltiples clasificaciones y entre ellas podemos considerar:

1. Técnicos y no técnicos.
2. Unipersonales y multipersonales.
3. Monodisciplinarios y multidisciplinarios.
4. Monocontrato o multicontrato.
5. Resultados: tangibles o intangibles.
6. Rentabilidad económica o rentabilidad social.
7. Con fines claros: proyectos espaciales.
8. Proactivos y Reactivos.
9. Internos y Externos.
10. De mayor o menor envergadura.
11. Inversión propia o externa (privada/pública) o mixta.
12. De investigación y desarrollo.

1.2.1.4 Entorno del proyecto

El conjunto de condiciones en las que se va a realizar el proyecto se conoce como **entorno**. El entorno del proyecto puede cambiar fácilmente, en especial el contexto socio-económico.

La influencia del entorno sobre el proyecto será más intensa cuanto más duras sean las condiciones económicas y sociales. A continuación se muestra una imagen ilustrativa:

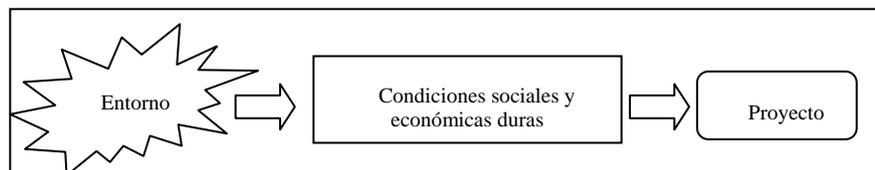


Figura 3. Influencia del Entorno en el Proyecto

Las épocas de crisis que traen condiciones sociales y económicas duras terminan haciendo que el entorno influya en el proyecto de diferentes maneras.

Por otra parte, también es importante tener en cuenta que la influencia del proyecto sobre el entorno será más intensa cuanto mayor sea la masa del proyecto y su grado de innovación.

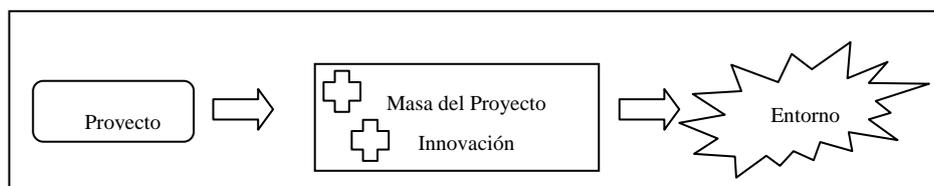


Figura 4. Influencia del Proyecto en el Entorno

Un ejemplo de esta influencia pueden ser los teléfonos móviles, que fueron insertados en el entorno, produciendo una gran influencia en el mismo.

1.2.2 Dirección de Proyectos

La dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo.

Evidentemente, supone gozar de una visibilidad más amplia sobre los recursos y los objetivos, y difícilmente estaremos dispuestos a renunciar a la gestión de nuestros propios recursos aunque, en muchos casos, implique cierta responsabilidad adicional y, por qué no decirlo, algún que otro dolor de cabeza.

Asumido lo anterior, ¿por qué los jóvenes profesionales son tan reacios a dirigir y a gestionar los proyectos en los que participan? Es evidente que al optar por una carrera profesional no vinculada con la gestión, el individuo se decanta por actividades técnicas, pero eso no ha de significar renunciar a la perspectiva que supone participar en las tareas de dirección y gestión de los proyectos en los que trabaje.

Participar en la gestión y dirección de un proyecto supone conocer los recursos y objetivos del mismo, así como las limitaciones a tener en cuenta, y este conocimiento nos permite optimizar nuestro trabajo y adaptarlo a los requisitos más específicos y más relevantes del proyecto. La dirección pasa de ser una tarea aislada a constituirse en una herramienta para mejorar el desarrollo de las actividades técnicas.

El cumplimiento de los requisitos del proyecto se logra mediante la aplicación e integración adecuadas de procesos de la dirección de proyectos agrupados lógicamente en lo que se determina FASE.

Las FASEs principales de un proyecto son:

- Iniciación
- Planificación
- Ejecución
- Seguimiento y Control
- Cierre.

1.2.3 Proyecto Informático

Un Proyecto Informático es un sistema de cursos de acciones simultáneas y/o secuenciales que incluye personas, equipamientos de hardware, software y comunicaciones, enfocadas en obtener uno o más resultados deseables sobre un sistema de información.

Los recursos más frecuentemente utilizados que caracterizan a un sistema de información, son los componentes de la Tecnología de la Información (TI) como puede ser el uso de Hardware, Software y Comunicaciones. Considerando entonces, la importancia que la informática tiene en los planes estratégicos de cualquier empresa moderna, no solamente se debe tener en cuenta la evolución de los recursos de la tecnología de la información, sino también las distintas metodologías para el desarrollo de los sistemas de información.

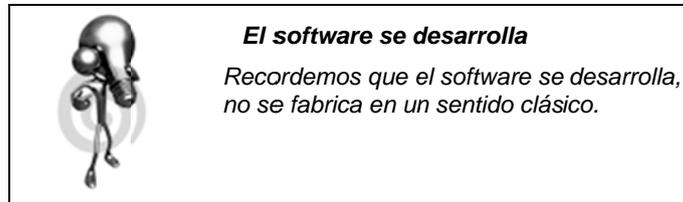
1.2.4 Tipos de Proyectos Informáticos

Existen diferentes clasificaciones de los tipos de proyectos informáticos. A continuación listamos los principales tipos de proyectos informáticos:

- Software
 - Metodologías, Ingeniería del software, etc.
 - Software empotrado.
- Hardware
 - Velocidad de Proceso, S.O., Servicios, etc.
- Comunicaciones y Redes
 - Protocolos, Buses, Cableado, etc.
- Instalaciones de Hardware
 - Peso de los equipos, Instalación de aire acondicionado, Suelo flotante, Extinción de incendios, Conectividad externa, etc.
 - CPD's, Sites de Internet, etc.
- Sistemas de Misión Crítica
 - Industrial, Médica, Nuclear, Militar, Aeronáutica, etc.
 - Tiempo real, Esquemas productivos, etc.
- Auditorías
 - Sistemas, Seguridad, Calidad, Legislación ...
- Peritajes
 - Civiles, Penales, Laborales...
- Consultoría y Asesoría
 - Sobre cualquier actividad.
- Seguridad Informática (ISO 17799)
 - Seguridad de la Información.
- Reingeniería de Proyectos

- De cualquiera de los tipos.

Los proyectos de desarrollo de software se diferencian de los otros proyectos de ingeniería tradicional en la **naturaleza lógica** del producto software.



En todos los proyectos de ingeniería la buena calidad se adquiere mediante un buen diseño, pero en el caso del software la etapa de construcción incide pobremente en su calidad, no así en la construcción de hardware o de una obra civil. Otra diferencia es que el software no se estropea, el paso del tiempo o males del entorno no inciden en el aumento de la tasa de fallos.

Así, no se puede gestionar un proyecto de desarrollo de software como si se tratara de un proyecto de fabricación.

La gestión del proyecto de software es el primer nivel del proceso de ingeniería de software, porque cubre todo el proceso de desarrollo. Para conseguir un proyecto de software fructífero se debe comprender el ámbito del trabajo a realizar, los riesgos en los que se puede incurrir, los recursos requeridos, las tareas a llevar a cabo, el esfuerzo (coste) a consumir y el plan a seguir.

1.2.5 Dirección y Gestión de Proyectos Informáticos

La dirección y gestión de proyectos es la aplicación del enfoque de sistemas para la administración de tareas tecnológicas complejas o de proyectos cuyos objetivos se establecen explícitamente en términos de tiempo, costes y parámetros de realización.

Sin entrar en metodologías de trabajo concretas, podemos decir que para gestionar adecuadamente un proyecto de desarrollo de software es recomendable disponer de las siguientes **herramientas**:

1. Un **sistema de planificación** que nos permita organizar el proyecto en función de hitos, tareas y subtareas, con asignación y control de tiempos y recursos materiales y humanos. Idealmente el sistema de planificación debe permitirnos también hacer el seguimiento y reajustar la planificación en función de la evolución del proyecto. Este componente debe permitir definir un proyecto como una sucesión de hitos que a su vez se descompongan en tareas y subtareas, con asignación de tiempo y recursos a cada una. Además de definir la planificación, el sistema debe proporcionar mecanismos para hacer el seguimiento de la misma y modificar la planificación

cuando sea necesario. Para que esto sea posible es recomendable disponer de herramientas para llevar el control de los tiempos estimados y empleados para cada tarea; para poder controlar de verdad la evolución del proyecto es importante que las personas que trabajan en el proyecto vayan reportando el tiempo que dedican a cada tarea y actualicen el estado de las mismas con relativa frecuencia. Para un proyecto normal puede ser suficiente con actualizar semanalmente, aunque el control de tiempos siempre es más fiable si se completa **diariamente**.

2. Un **sistema de gestión documental**, que nos servirá para almacenar y mantener los documentos obtenidos o generados durante el desarrollo del proyecto y acceder a ellos cómodamente. Cada hito, tarea o subtarea puede implicar la obtención o generación de documentación (actas de reuniones, documentos de diseño, etc.). Idealmente, el sistema de gestión de proyectos debe permitir que almacenemos esa documentación en el propio sistema.
3. Un **sistema de control de versiones**, que se utilizará para permitir el desarrollo concurrente y para mantener la historia del código fuente y parte de la documentación producida en el proyecto. Al tratarse de proyectos informáticos lo normal es que se trabaje con código fuente y con documentos que van evolucionando a lo largo del desarrollo y que deben ser modificados por múltiples personas, por lo que resulta casi imprescindible disponer de un sistema de control de versiones que permita mantener la historia de los ficheros generados y que más de una persona trabaje concurrentemente sobre el mismo código.
4. Un **sistema de gestión de incidencias** que se empleará para hacer el seguimiento de los errores detectados y sus correcciones, tanto aquellos reportados por los responsables de la prueba del software como por los desarrolladores o los usuarios finales. Este tipo de sistema también se puede utilizar como sistema de seguimiento de tareas de corta duración asociadas a fases del proyecto, a errores detectados o a cambios relacionados con solicitudes de mejora solicitadas por el cliente.

1.2.6 Rol del Director del Proyecto

El director del proyecto es la persona asignada por la organización ejecutante para alcanzar los objetivos del proyecto. Varias de las herramientas y técnicas para dirigir proyectos son específicas a la dirección de proyectos. Sin embargo, comprender y aplicar los conocimientos, herramientas y técnicas que se reconocen como buenas prácticas **no** es suficiente para gestionar los proyectos de un modo eficaz y eficiente. Además de las habilidades específicas a un área y de las competencias generales en materia de gestión requeridas para el proyecto, la dirección de proyectos efectiva requiere que el director del proyecto cuente con las siguientes características:

- **Conocimiento y habilidades gerenciales.** Se refiere a lo que director del proyecto sabe sobre la dirección de proyectos.
- **Desempeño.** Se refiere a lo que el director del proyecto puede hacer o lograr si aplica los conocimientos en dirección de proyectos.
- **Liderazgo.** Establecer la unidad de propósito y la orientación de la dirección de la organización. Crear y mantener un ambiente interno, en el cual el personal pueda llegar a involucrarse totalmente en el logro de los objetivos de la organización.

Dirigir un proyecto por lo general implica:

- Abordar las diversas necesidades, inquietudes y expectativas de los interesados según se planifica y efectúa el proyecto.
- Liderar efectivamente un grupo humano.
- Equilibrar las restricciones contrapuestas del proyecto que se relacionan, entre calidad, costes y duración.
- Identificar y gestionar riesgos.

(Domingo Ajenjo), en el libro “Dirección y Gestión de Proyectos”, hizo una de las mejores representaciones de la labor del Director de Proyecto. La siguiente figura 5 lo muestra:



Figura 5. La Labor del Director de Proyectos

Esta imagen representa que el camino del director de proyecto es dificultoso, que tiene un objetivo, pero el alcance del mismo puede ser afectado por contingencias e imprevistos. Un buen Director de Proyectos puede minimizar posibles problemas de contingencias e imprevistos realizando un correcto Análisis de Riesgos, como se indica en el capítulo V del libro. Esta imagen también muestra que el Director de Proyectos avanzará en el tiempo teniendo en cuenta siempre los plazos y costes, los cuales deben ser controlados constantemente. Con recursos materiales y humanos, un director de proyectos podrá alcanzar sus objetivos, por eso la importancia de hacer una selección correcta de los mismos antes de comenzar el proyecto. Finalmente, el gráfico muestra que un director de proyectos siempre será juzgado, por lo que tiene que estar preparado para recibir los resultados finales y hacer un análisis correcto del mismo.

1.2.7 Gestión

“Articular el método para alcanzar un objetivo único y no repetitivo en un plazo con principio y fin claros utilizando las técnicas que nos proporciona la gestión.”

Las principales tareas a realizar son: planificar y establecer estrategias adecuadas, organizar a los miembros y equipos para lograr los objetivos que queremos alcanzar, y controlar y comprobar si se están alcanzando dichos objetivos. La organización de un proyecto consiste en diseñar la estructura con la que vamos a establecer las dependencias entre individuos, departamentos, cosas... dentro del proyecto. Asimismo, debemos asignar las tareas más idóneas para esas capacidades y el tiempo estimado para cumplir las tareas o funciones.

1.2.8 La empresa

1.2.8.1 Definición de empresa

Una **empresa** es el ejercicio profesional de una actividad económica planificada con el objetivo de ofrecer un producto o servicio.

Para crear una empresa se necesita:

- **Negocio:** cierta actividad comercial que genera beneficios.

- Sociedad mercantil: sociedad que tiene por objeto la realización de uno o más actos de comercio. Es reconocida como una unidad jurídica, y cuenta con un patrimonio propio.
- Explotación: conjunto de procesos tecnológicos por los que a partir de unos elementos se obtienen ciertos productos o resultados.
- Planta o establecimiento: unidad espacial o física donde se localiza la actividad económica o comercial.

1.2.8.2 Dimensiones de la empresa

- Dimensión funcional: actividad organizada y alternativa al mercado con ánimo de lucro.
- Dimensión técnico-económica: actividad productiva de bienes y servicios. Se encarga de la transformación de productos.
- Dimensión económico-financiera: actividad económica que genera valor y dinero.
- Dimensión jurídico-mercantil: actividad generadora de relaciones.
- Dimensión social: actividad compuesta por relaciones humanas y de poder. Se ocupa de la comunicación y la relación entre las personas que forman la empresa.

1.2.8.3 El empresario

Podemos encontrar varias definiciones para empresario:

- Persona que asume riesgos.
- Persona que cede el capital para iniciar la actividad empresarial.
- Persona que establece cómo se realiza la combinación de factores dentro de la empresa.
- Persona que fija objetivos a conseguir, establece los medios que llevarán a conseguir esos objetivos y plantea las medidas económicas oportunas y más favorables para ello.

1.2.8.4 El directivo

Se encarga de la dirección de la empresa. Se ocupará, entre otras, de las siguientes cuestiones:

- **Globalización de la economía** (eliminación de barreras políticas, aduaneras,...).
- **Descentralización** a nivel de países y de estructura de la propia empresa.

- **Orientación hacia procesos** y no hacia funciones.
- **División del trabajo:** cada vez más y más rápido.
- **Trabajo en red:** mejora la coordinación y se asimilan más fácilmente los cambios.
- **Trabajo en grupo:** mejora la eficiencia y la productividad.
- **Definición de objetivos** y del camino a seguir para conseguirlos (*management*).

CAPÍTULO II. FASES DE UN PROYECTO

“Los hombres no viven juntos porque sí, sino para acometer juntos grandes empresas”. (José Ortega y Gasset)

Antes de comenzar el desarrollo de un proyecto hay que tener en cuenta y estudiar todas y cada una de las fases por las que pasa. Según el PMBOK, *“Las fases del proyecto son divisiones dentro del mismo proyecto, donde es necesario ejercer un control adicional para gestionar eficazmente la conclusión de un entregable mayor”*

En este capítulo vamos a redactar una descripción básica de las diferentes fases de un proyecto. Sin entrar en discusión con las diferentes metodologías que existen actualmente, nosotros preferimos simplificar la explicación al lector diciendo que en la mayoría de los proyectos se pueden distinguir tres grandes secciones de trabajo: **planeación**, **ejecución** y **mantenimiento**.



Figura 6. Fases Principales de un Proyecto

Tal como se muestra en la figura 6, la planeación es donde está la fase de definición del proyecto y donde entra la planificación del proyecto.

- Planeación
 - Definición del problema.
 - Planificación del proyecto.

Cuando finalizamos la Planeación, nos tenemos que hacer la siguiente pregunta obligada:
¿Es viable el proyecto?

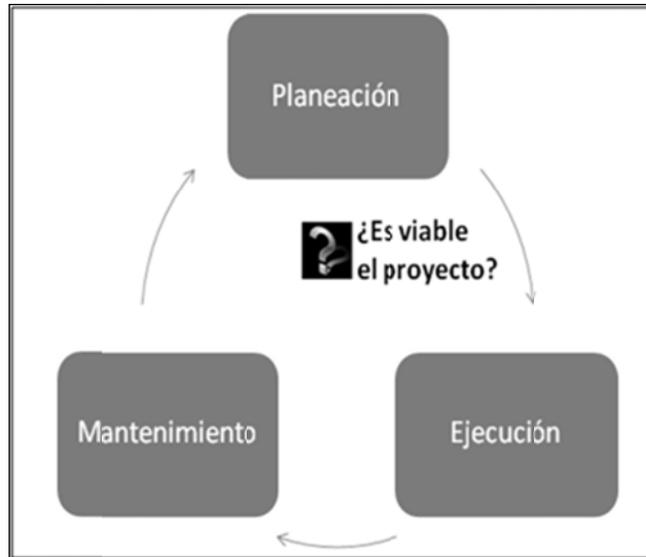


Figura 7. Secuencia de Fases

Consecuentemente con la figura 7, la ejecución es donde se trata la puesta en marcha, la fase productiva y la conclusión del proyecto.

- Ejecución del proyecto
 - Puesta en marcha.
 - Fase productiva.
 - Conclusión del proyecto.

Una vez realizado y entregado el proyecto, se inicializa la fase de Mantenimiento, que trae características propias del proyecto y el entorno en el que fue implantado.

2.1 Planeación

En la **planeación** de un proyecto es importante tener claro tanto el problema que se pretende solucionar, el producto que se quiere obtener, como el servicio que se pretende proporcionar. A su vez es necesario evaluar tanto los costes económicos como los recursos humanos.

Fases de la planeación:

- **Inicio y Definición del Problema**, es decir, clarificar el producto o servicio a obtener.

- **Planificación del Proyecto;** atender a las necesidades que aparecerán a lo largo del desarrollo, anticipando el curso de las tareas a realizar, la secuencia en que se llevarán a cabo, los recursos y el momento en que serán necesarios.

Esta fase es explicada detalladamente en el capítulo III del libro.

2.1.1 Inicio y Definición del problema

El origen de un proyecto suele surgir a partir de una necesidad que se convierte en un problema. En este punto hay que trabajar con los usuarios, directores de empresa y clientes, ya que será de ellos de quien tendremos que obtener la información para saber donde reside el problema y donde está la oportunidad. A la definición del problema muchas veces no se le da la importancia que se merece.

Hay que tener en cuenta que todo el proyecto se basará en esta definición y es mejor que quede clara. Debe ser revisada por todos los usuarios, directores de empresa y clientes. En esta fase se definen, entre otras cosas, los límites del proyecto.



Acta de Constitución de Proyecto

Es altamente recomendable crear un Acta de Constitución del Proyecto, basado en PMI, que autorice formalmente un proyecto o una fase, y en la misma, documentar los requisitos iniciales que satisfacen las necesidades y expectativas de los interesados. Para más información ver el caso de estudio presentado en esta obra.

2.1.2 Planificación del Proyecto

En esta fase se deberán identificar todas las cosas necesarias para poder alcanzar el objetivo marcado, considerando las tres dimensiones sobre los que se apoyará el desarrollo de todo el proyecto:

- **Calidad:** viene dada por las especificaciones.
- **Coste,** valorado en el presupuesto.
- **Duración:** asignada en el calendario de trabajo.

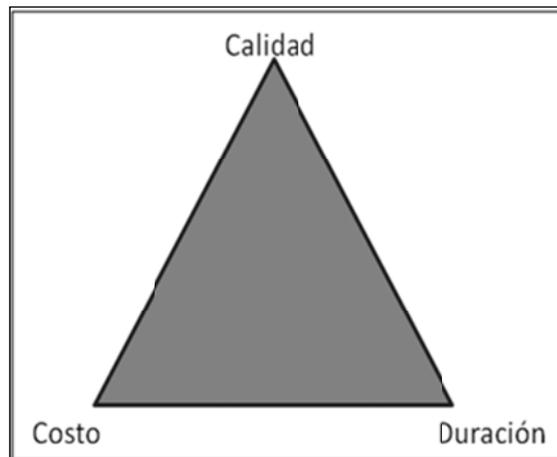


Figura 8. Relación de Calidad, Costo y Duración de un Proyecto

Hay quien separa el análisis de la aplicación de la propia planificación, por entenderse que la primera es una tarea técnica, mientras que la planificación es una tarea de gestión. Aun siendo así, tienen que realizarse de forma simultánea.

Hay que **planificar** el trabajo antes de llevarlo a cabo, antes de realizar el análisis de los trabajos asociados a éste, pero difícilmente se podrá realizar la planificación de todo el proyecto.

Tareas a realizar en la etapa definición del problema:

- **Estimar** el tamaño de la aplicación a desarrollar. Estudiar el sistema actual.
- Discutir y **analizar** lo que se desea obtener.
- Identificar las **tareas** a realizar.
- **Asignar recursos** a cada tarea.
- Crear un **calendario** de las tareas.
- Realizar un **estudio económico**.



Presentación de Oferta al Cliente

*Es importante resaltar que podremos presentar una **Oferta** de manera responsable al cliente solo cuando tengamos realizada una planificación correcta del proyecto.*

2.2 Ejecución

En la ejecución de un proyecto, se trata de llevar a cabo el plan previo. La ejecución se verá fuertemente influida por la planificación, es decir, una mala planificación supondrá una mala ejecución.

Fases de la ejecución:

- Puesta en marcha.
- Fase productiva.
- Conclusión del proyecto. Esta fase es explicada detalladamente en el capítulo IV del libro.

2.2.1 Puesta en marcha

En esta fase se ha de organizar el equipo de desarrollo, los mecanismos de comunicación, la asignación de roles y de responsabilidades a cada persona.

Las principales tareas son:

- Ajustar a las disponibilidades actuales las necesidades de personal de la fase de planificación.
- Establecimiento de la estructura organizativa.
- Definir responsabilidades y autoridad.
- Organizar el lugar de trabajo. En muchas ocasiones el comienzo de un proyecto tiene tareas como instalación de equipamientos, acondicionamiento de locales, etc.
- Puesta en funcionamiento del equipo. Organizar reuniones más o menos informales para que se conozcan las personas del equipo en el caso de que esto no sea así; esto evitará malentendidos y conflictos durante la ejecución del proyecto.
- Divulgación de los estándares de trabajo y sistemas de informes. Al comenzar el proyecto, las personas están más receptivas y esta es una razón para introducir los nuevos métodos de trabajo.



Ajustar Proyecto

Es estratégico ajustar los requisitos, recursos y personas a las disponibilidades actuales para evitar inconvenientes y desperdicios de recurso en la Fase Productiva.

2.2.2 Fase productiva

En esta fase se realiza la actividad gruesa del proyecto. En la misma se realizarán actividades de análisis, diseño, implementación y pruebas. Todo el equipo de trabajo asignado al proyecto estará abocado a las diferentes actividades designadas.

En esta fase productiva ya no debería existir alguna duda respecto a las especificaciones, recursos y personas en situación de trabajo. Éstas deben llevar a cabo cada una de las tareas que se les ha asignado. Aunque esta última frase es correcta, también lo es saber que existen riesgos en todos los proyectos, por lo tanto, es muy importante hacer un seguimiento del mismo y actuar rápidamente ante cada evento que pueda alterar el curso normal del proyecto.

El responsable del proyecto debe tomar medidas de rendimiento, revisar los informes de los empleados, mantener reuniones para identificar los problemas antes de que aparezcan y en este caso poner en práctica las acciones correctivas y preventivas necesarias.

2.2.3 Conclusión del Proyecto

En esta fase se trata de dar por finalizado el proyecto y entregar el producto definitivamente al cliente.

Las principales actividades a realizar son éstas:

- Revisar las desviaciones del proyecto e identificarlas para evitarlas en futuros proyectos.
- Reasignar el personal a los nuevos proyectos o reintegrarlos en los departamentos de partida.
- Es interesante documentar las relaciones entre los empleados para futuros proyectos.

2.3 Fase de Mantenimiento

Es el proceso de mejora y optimización del software después de su entrega al usuario final (es decir; revisión del programa), así como también la corrección y prevención de los defectos.

La fase de mantenimiento es la fase que viene después de la implantación. Esta fase es explicada detalladamente en la sección **4.8** de este libro.

CAPÍTULO III. FASE DE PLANEACIÓN

“No es el plan lo que importa, sino la planificación.” (Dr. Graeme Edwards)

3.1 Descripción de la Fase de Planeación

En esta fase se incluye la concepción inicial, la definición del problema y la planificación del proyecto. Como resultado final de esta fase se obtendrá el estudio de viabilidad que permitirá determinar si un proyecto es factible o no. En este libro se detallarán herramientas para determinar una factibilidad técnica, de gestión y económica. Un proyecto podría ser factible técnicamente, es decir que se puede realizar con los recursos adquiridos para el proyecto. También podría tener factibilidad de gestión, es decir, que los tiempos estimados son idóneos y que los recursos asignados estarán disponibles para afrontar el esfuerzo estimado. Por último, también podríamos decir que un proyecto es factible económicamente si el precio obtenido es afrontable por el cliente, y si la forma de pago determinada por el plan financiero no produce ninguna pérdida. Estas tres variantes son sumamente importantes en la mayoría de los proyectos, pero es importante recalcar que no son las únicas. Por ejemplo, un proyecto podría ser factible en los tres puntos anteriores, pero no serlo a nivel político en el ámbito que se desempeña. Es decir, si el proyecto tiene intereses políticos y estos no son cumplimentados, el proyecto dejaría de ser factible.

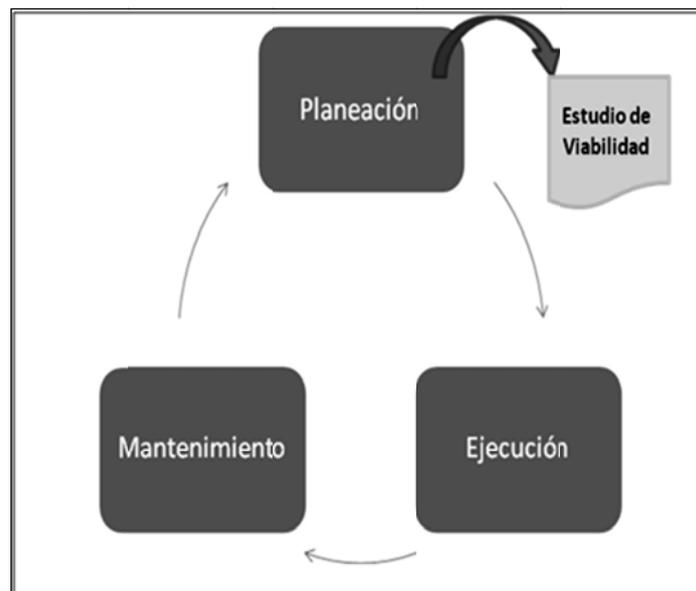


Figura 9. La fase de planeación genera el Estudio de Viabilidad

3.2 Inicio del Proyecto

Un proyecto siempre comienza cuando se descubre una necesidad en el entorno. Esta necesidad puede venir desde un potencial cliente. Esta necesidad puede generar un proyecto nuevo o una nueva versión de un proyecto (producto) ya existente.

Pero que exista esa necesidad no quiere decir que directamente comenzaremos a trabajar en el proyecto sin realizar un análisis previo.

Lo primero que tenemos que analizar es si tenemos capacidades técnicas, recursos (físicos, temporales y humanos) para poder realizar un proyecto que pueda satisfacer esa necesidad. Sin dejar de considerar siempre, si el proyecto es de nuestro interés o no. Para esto, antes que nada, tenemos que realizar un relevamiento de información pertinente para definir algunos puntos claves:

- Objetivos del proyecto.
- Potencial Cliente.
- Sector del mercado que se beneficiaría con el proyecto.
- Necesidad de Negocio.
- Oportunidades de Negocio que justifiquen la inversión.
- Beneficios cuantitativos y cualitativos que brindaría el proyecto.
- Identificación del producto o servicio que se brindará.
- Responsable del proyecto.
- Principales Hitos y Eventos del proyecto (previo a la planificación).
- Interesados en el proyecto.
- Suposición o restricciones que se puedan identificar del proyecto.

Esta información se plasma en el **Acta de Constitución del Proyecto** y registro de interesados. Cuando el acta de constitución del proyecto recibe aprobación, el proyecto se considera autorizado oficialmente. Aunque el equipo de dirección del proyecto pueda colaborar en la redacción de este acta, la aprobación y el financiamiento se manejan fuera de los límites del proyecto.



Acta de Constitución de Proyecto

Es altamente recomendable formalizar esto en un documento. El "Acta de Constitución de Proyecto" de PMI, es un herramienta útil para esta formalización. Para aprender cómo desarrollar un Acta de Constitución de proyecto recomendamos revisar la sección "3.3.1 Desarrollar el Acta de Constitución del Proyecto" del PMBOK.

Es estratégico involucrar a los clientes y a otros interesados durante la iniciación, ya que esto mejora la probabilidad de contar con propiedad compartida, con la aceptación de los entregables y con la satisfacción del cliente y demás interesados.

Un Acta de Constitución de Proyecto representa una documentación formal de la necesidad de negocio que el proyecto se comprometió a abordar. Además esto permite verificar los criterios de éxito y revisar la influencia y los objetivos de los interesados en el proyecto. Con este documento ya se puede **tomar una decisión sobre la necesidad de continuar, posponer o suspender el proyecto.**

3.3 Definición del Problema

Como se dijo anteriormente, el origen de un proyecto suele surgir a partir de una necesidad que se convierte en un problema. Muchas veces el cliente cree que puede necesitar algo pero no ha identificado claramente cuál es su problema. Es responsabilidad del equipo de trabajo ayudar a encontrar y definir el problema que está incitando el desarrollo del proyecto. En este punto hay que trabajar con los usuarios, directores de empresa, clientes y todos los interesados en el proyecto, ya que serán ellos de quienes tendremos que obtener la información para saber cuál es el problema.

Una adecuada definición del problema requiere de la capacidad para diferenciar entre causas, síntomas y necesidades para ser resueltas técnicamente.

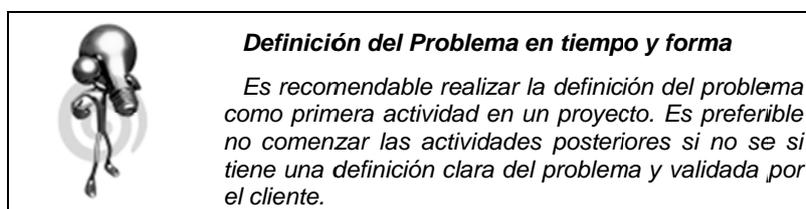
El problema permite conocer y delimitar el terreno de lo desconocido, y es decisivo en el resultado final: una definición incorrecta nos lleva a encontrar una posible solución incorrecta. Su planteamiento adecuado es necesario para determinar el alcance del proyecto y las posibles vías de solución.

Hay que tener en cuenta que todo el proyecto se basará en esta definición y es mejor que quede bien clarificada. Debe ser revisada por todos los interesados en el proyecto.

Tareas a realizar en la fase definición del problema:

- **Estudiar** el sistema y contexto actual.
- Discutir y **analizar** lo que se desea obtener.

- Clarificar las **áreas de la empresa** que se verán afectadas.
- **Definir** el problema y sus componentes, aclarando:
 - Que es fundamental.
 - Que es deseable.
 - Que es opcional.
- Identificar al/los **responsable/s** del proyecto.
- Crear una **declaración clara** de lo que se va a solucionar.
- **Validar** la definición inicial del problema con los interesados en el proyecto.



3.3.1 Caso de Estudio: Definición del Problema

Situación actual.

Un gran problema que aqueja a los laboratorios de análisis clínicos de gran envergadura (en la provincia de Tucumán) no es la limitación en la infraestructura de los mismos, sino más bien es la incompleta administración de la información necesaria para el desarrollo operativo del negocio. Convirtiéndose estos problemas en un factor limitante para su crecimiento.

Esta problemática de información encuentra su origen, de forma parcial, en la amplia implementación de sistemas informáticos en las organizaciones desde hace aproximadamente 15 años atrás (1995), los cuales estaban caracterizados por carecer de metodologías de desarrollo que permitiesen realizar un análisis y diseño profesional, y en donde gran parte de las organizaciones de Tucumán tuvieron acceso a un software a bajo coste que gestionase la información de su negocio, lo que no implica la satisfacción de las necesidades reales. Esto condujo a una mala gestión de la información, resultando en disminución del nivel de eficiencia de las organizaciones y carencia de satisfacción general. Como alternativa a esto, hoy en día las nuevas tecnologías permiten la integración de servicios relacionados a la información, brindando una nueva gama de funcionalidades que aumenta la satisfacción general del usuario. El Sistema de Gestión Bioquímica se analiza y diseña para solucionar este factor limitante, considerándose como un sistema abocado a las necesidades reales de los Laboratorios Bioquímicos y por lo tanto de comercialización factible. El siguiente diagrama ilustra como las diferentes causas interactúan generando el problema.

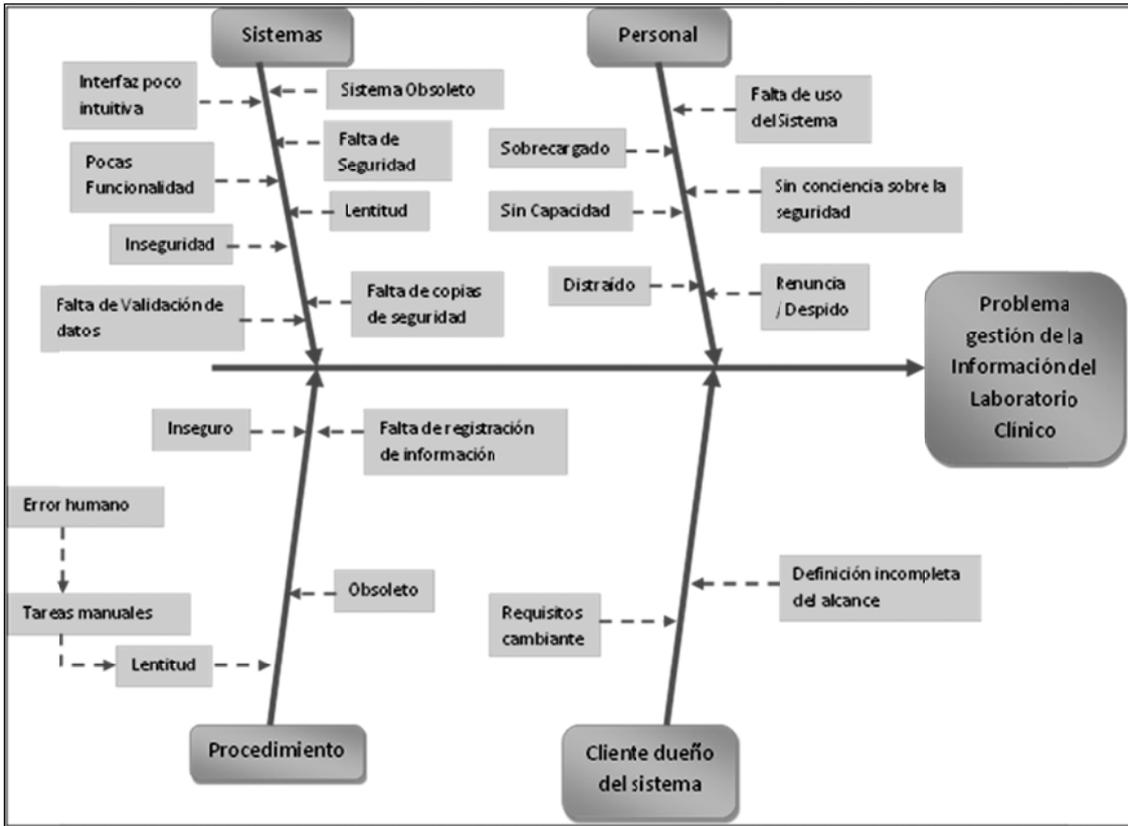


Figura 10. Diagrama Causa-Efecto

Tomando como referencia las características de uno de los mejores sistemas del mercado, realizamos la siguiente comparativa:

Parámetro	Sistema de Gestión - Proyecto General	Omega 3000
Soporte para múltiples áreas o sectores del laboratorio	Si, es independiente del área	Si, especializado en Hematología y Clínica en General
Orientación a la Web	Si	Si
Orientación a escritorio	Si	Si
Orientación a móviles	Si, pero no prevista en el corto plazo	No, pero si sería posible
Conexión con equipos analizadores	Si, pero se desarrollará de acuerdo a los equipos con los que cuenten cada laboratorio	Si, con más de 400 controladores de equipos analizadores
Exportación de datos a Excel	Si	No mencionado explícitamente, pero generalmente si soportan
Exportación de datos en XML	Si	No
Integración con LDAP	No	Si
Módulo de estadísticas	Si	Si, de avanzada
Módulo de Auditoría	No	Si
Módulo de Dosis	No	Si
Módulo de Urgencias	No	Si
Flujo de estado de protocolo personalizado	Si	No, posee un flujo básico
Flujo de estado de prácticas de protocolo personalizado	Si	No, posee flujo básico
Módulo para Facturación con Colegio Bioquímico de Tucumán	Si, requerirá implementar un módulo en el colegio de bioquímicos	No
Definir los datos que se requieren registrar de acuerdo a los tipos de clientes	Si, requiere modificación del diseño interno	No
Definir reglas de validación personalizadas	Si, requiere modificación del diseño interno	No
Posibilidad de agregar funciones personalizadas	Si, requiere redefinición del alcance	No

Figura 11. Diferencia contra Soluciones

Estos son solo algunos parámetros de comparación, no debe considerarse una comparación exhaustiva, ya que se deben realizar pruebas completas de funcionales, de carga de sistema, de usabilidad, de seguridad, y de integridad entre otras.

Si bien se observa que Omega 3000 es un producto que ofrece una gran gama de funcionalidades que nuestra propuesta no iguala, observamos que al ser tan general se pierde lo particular y los requerimientos menores pero necesarias, tal como lo es la posibilidad de facturación de análisis al Colegio de Bioquímicos de Tucumán. Esto evidencia la insatisfacción de las necesidades reales de los laboratorios, ya que no se adapta realmente a lo que el laboratorio necesita. Otro ejemplo de esta situación es el llamado a sala de extracción de paciente de estudios de VIH. Normalmente, los sistemas no contemplan que se les asigne un número de turno para la extracción, por lo que los pacientes son llamados por sus nombres. Ahora esto crea un potencial conflicto ya que la realización de análisis de VIH es confidencial según lo establece la Ley 23.798 en su artículo 2 inciso E. Y para el registro del paciente “Se utilizará, exclusivamente, un sistema que combine las iniciales del nombre y del apellido, día y año de nacimiento. Los días y meses de un solo dígito serán antepuestos de número cero” según lo establece el DECRETO REGLAMENTARIO N° 1.244/91 DE LA LEY N° 23.798 art. 2 inciso E.

3.3.2 Entregable – Acta de Constitución del Proyecto

Consiste en desarrollar un documento que autoriza formalmente un proyecto, y en documentar los requisitos iniciales que satisfacen las necesidades y expectativas de los interesados. Se define el alcance inicial y se comprometen los recursos financieros iniciales. Se identifican los interesados internos y externos que van a interactuar y ejercer alguna influencia sobre el resultado global del proyecto, y se documenta información relevante relativa a sus intereses. Este es uno de los procesos de más importantes de esta fase y ya nos permitirá definir una estrategia de gestión de estos interesados, intentando disminuir el impacto negativo que puedan ocasionar en el proyecto. Si aún no fue nombrado, se seleccionará el director del proyecto.

Este documento establece una relación de cooperación entre la organización ejecutante y la organización solicitante (o cliente, en el caso de proyectos externos). De preferencia se asigna un Director de Proyecto durante la elaboración del acta de constitución del proyecto, pero siempre antes de comenzar la planificación. Se recomienda que el director del proyecto participe en la elaboración del acta de constitución del proyecto, ya que ésta le otorga la autoridad para asignar los recursos a las actividades del proyecto. Los proyectos son autorizados por alguien externo al proyecto, tal como un patrocinador, una oficina de dirección de proyectos (PMO) o un comité ejecutivo del portafolio. El iniciador del proyecto o el patrocinador debe encontrarse a un nivel apropiado para financiar el proyecto. Cualquiera de ellos elaborará el acta de constitución del proyecto o delegará esta tarea al director del proyecto (generalmente ocurre lo segundo). El proyecto queda autorizado con la firma del iniciador en el acta. Los proyectos se autorizan en función de las necesidades internas de la empresa o de influencias externas. Esto normalmente desencadena la realización de un análisis de necesidades, de un caso de negocio o la descripción de la situación que el proyecto

abordará. La elaboración del acta de constitución de un proyecto vincula el proyecto en cuestión con la estrategia y el trabajo en curso de la organización.

La figura 12 muestra los factores que intervienen en la generación del Acta de Constitución del Proyecto.



Figura 12. Factores que intervienen en la generación del Acta de Constitución del Proyecto

Objetivos que persigue:

- Autorizar formalmente el proyecto.
- Nombrar el Director de Proyecto.
- Describir la necesidad de negocio y su justificación.
- A partir de la necesidad de negocio, identificar los objetivos del proyecto.
- Identificar las aéreas involucradas y recursos necesarios.

Contenido del Acta de Constitución del Proyecto:

- Necesidades del negocio y oportunidades de negocio que justifiquen la inversión.
- Finalidad del proyecto (justificación) – Presupuesto resumido.
- Descripción del producto o servicio.
- Objetivos del proyecto propiamente dicho en términos de TIEMPO, COSTE Y CALIDAD.
- Director de proyecto nombrado.
- Participación de áreas funcionales y miembros del equipo / Influencia de los interesados.
- Cronograma de Hitos resumido y Supuestos y Restricciones.

3.3.3 Caso de Estudio: Acta de Constitución

<i>Id proyecto</i>	Nombre proyecto	<i>Cliente interno o externo</i>	Código de documento
1	Sistema de Gestión Bioquímica	Clinical Lab	T045
<i>Patrocinador del proyecto</i>		<i>Elaborado por</i>	<i>Fecha</i>
		Daniel Talebi – Pablo Ortiz	16/10/2010

Necesidad de negocio / oportunidades de negocio que justifiquen la inversión

Actualmente casi la totalidad de los laboratorios de análisis clínicos en San Miguel de Tucumán, cuentan con un sistema informático que les permite realizar la administración de los procesos que se llevan a cabo en el laboratorio, por lo tanto juegan un papel vital en la organización. Pero para que una organización se expanda, no solo debe aumentar la cantidad de personal contratado, sino también la infraestructura necesaria para el desarrollo de las actividades. Resulta así que con el pasar del tiempo los sistemas informáticos realizados para la mayoría de los laboratorios fueron quedando obsoletos, dificultando el proceso de expansión y satisfacción de los clientes.

Así es evidente la necesidad de contar con un Sistema de Gestión desarrollado con las últimas tecnologías del mercado y metodologías de trabajo, brindando una nueva gama de funcionalidades y servicios para la organización y el cliente.

Beneficios cuantitativos y cualitativos de la necesidad de negocio

Mayor velocidad de atención al cliente mediante el acceso web. Menores costos asociados a la atención personal del cliente. El nuevo sistema proveerá acceso remoto a las funcionalidades del mismo. El sistema estará basado en la utilización de las últimas tecnologías, aprovechando el uso de internet para establecer nuevos canales de comunicación para el personal de la organización y los clientes del laboratorio. Consulta de información de análisis vía internet. Mejora la imagen de la organización ante cliente.

Descripción de producto o servicio que el proyecto generará

El Sistema de Gestión Bioquímica realizará la administración de los análisis clínicos, gestión de usuarios de tipo empleado y clientes (particulares, laboratorios y centros de diálisis), cuenta corriente de clientes, y facturación, control de stock, registro de novedades y agenda.

En lo referente a los análisis clínicos, el sistema será capaz de registrar la recepción de muestras y su transición a lo largo del procesamiento hasta que los resultados están disponibles para ser informados a los clientes, mediante un informe impreso o publicándolos en internet. Para los laboratorios clientes se ofrecerá la posibilidad de cargar las prácticas que desean derivar de forma remota (mediante internet). Estos análisis generarán un registro en los movimientos diarios del registro de caja, así también como en la cuenta asociada a cada cliente, los cuales el sistema se encargará de registrar. Además deberá contar con un sistema de alertas, en el cual si los resultados de las prácticas evidencian un riesgo en la salud del paciente se comunicará de forma urgente, a quien corresponda, dicha información, en lo posible de forma automática. El sistema deberá registrar todas aquellas novedades asociadas a un protocolo o de carácter general.

Es necesario que el sistema cuente con manejo de recordatorios, como medio de ayuda en la realización de las actividades de los empleados.

Asimismo el sistema se encargará de gestionar a los tres tipos de clientes: Particulares (con y sin Obra Social), Laboratorios y Centros de Diálisis, manejando no tan solo la información filiatoria de cada uno de ellos, sino también información sobre su cuenta económica.

Así los dos tipos de usuarios, empleados y clientes, accederán al sistema utilizando nombre identificador y clave. La utilización de las distintas funcionalidades podrá configurarse según los permisos que se desee otorgar a cada usuario, también será posible crear perfiles de usuarios.

El registro de los movimientos de caja deberá incluir la emisión y recepción de facturas y otros tipos de comprobantes.

Del mismo modo, el sistema deberá realizar el manejo de listas de precios para cada práctica de acuerdo a la procedencia que la solicite, y administrar los precios que las obras sociales (que tienen convenios con Laboratorio Tucumán) pagan por las prácticas.

Otra de las funcionalidades del sistema será la asignación de tareas según distintos criterios, a los perfiles de técnicos o de bioquímicos, y el seguimiento del estado de las mismas, el control de stock de los insumos y sus consumos, registro básico de los médicos que ordenan los análisis. En la

parte de comunicación con equipos, se deberá proveer de los medios (interfaz) para permitir la carga automática de datos provenientes de los equipos que procesan las muestras, ya sea por conexión directa o indirecta. Así los equipos (o sus intermediarios) deberán cargar los datos en el sistema, y no que el sistema extraiga los datos de los equipos.

En todos los procesos se pondrá especial énfasis en la flexibilidad del sistema ante nuevos cambios.

Igualmente es importante realizar un control de trazabilidad, por lo que se registrará toda operación en la que se modifiquen los datos del sistema, como así también es importante proveer estadísticas para el Sistema de Gestión de Calidad del Laboratorio Tucumán.

El sistema no realizará la liquidación de sueldos de los empleados, control de asistencia, ni manejará información impositiva de la actividad económica (cálculos de impuestos, I.V.A, ingresos brutos y demás). No poseerá la base de datos de clientes de las obras sociales. No realizará llamadas telefónicas ni envío de mensajes de textos. La registración de las llamadas telefónicas deberá realizarse manualmente en el sistema. En caso que se considere necesario que el sistema envíe correos electrónicos, se supondrá que se dispone de una conexión a internet adecuada y que el proveedor del servicio de correo electrónico brinda un servicio óptimo y sin interrupciones. El control de stock requerirá que un empleado registre en el sistema el consumo de un insumo, con excepción del control de impresiones. No se proveerá de ningún otro producto que no sea el software desarrollado junto con sus manuales correspondientes y código fuente. También se incluirá solo un curso de Capacitación en Uso del Sistema.

<i>Hito / evento</i>	<i>Fecha propuesta</i>	<i>Descripción del hito o evento</i>	<i>Responsabilidad</i>
Módulo de Gestión de Usuario	A definir	Conjunto de funcionalidades relacionadas con la administración del personal que hace uso del sistema	Daniel Talebi – Pablo Ortiz
Módulo de Gestión de Cliente	A definir	Conjunto de funcionalidades relacionadas con la administración de los clientes del laboratorio	Daniel Talebi – Pablo Ortiz
Módulo de Seguridad	A definir	Conjunto de funcionalidades que proporcionan los medios para proteger la integridad, operatividad, confidencialidad y autenticidad de la información del sistema	Daniel Talebi – Pablo Ortiz
Módulo de Análisis Clínicos	A definir	Conjunto de funcionalidades relacionadas con la administración de los análisis clínicos que se realizan en el laboratorio	Daniel Talebi – Pablo Ortiz
Módulo de Facturación	A definir	Conjunto de funcionalidades relacionadas con la administración de los movimientos económicos de la realización de análisis clínicos a clientes	Daniel Talebi – Pablo Ortiz
Módulo de Control de Stock	A definir	Conjunto de funcionalidades que administra y controla el stock de los recursos controlables del laboratorio	Daniel Talebi – Pablo Ortiz
Módulo Otros	A definir	Conjunto de funcionalidades que no están enmarcadas dentro de las otras categorías	Daniel Talebi – Pablo Ortiz

<i>Miembro de equipo</i>	<i>Empresa / sector</i>	<i>Roles y responsabilidades</i>
Pablo Javier Ortiz	Interno	Analista Funcional
Daniel Talebi	Interno	Líder de Proyecto

<i>Subcontratistas / proveedores</i>	<i>Producto / servicio</i>
A definir	Programación del Sistema

<i>Interesados en el proyecto</i>		
<i>Nombre</i>	<i>Sector / empresa</i>	<i>Comentarios</i>
Laboratorios de Análisis Clínicos	General	El mercado puede verse beneficiado por contar con un Sistema Informático más, que se adapte a las necesidades del laboratorio y se desarrolle con las últimas tecnologías
Colegio de Bioquímicos Tucumán	General	Puede beneficiarse estableciendo un convenio para ofrecer a los laboratorios un sistema informático
Sanatorio	General	
Gobierno	General	

Suposiciones

En el alcance del sistema SGB no estarán incluidos los siguientes ítems:

- Liquidación de sueldos a empleados.
- Control de asistencia
- Manejo de información impositiva de actividades económicas (cálculos de impuestos, I.V.A, ingresos brutos y demás)
- Poseer la base de datos de clientes de las obras sociales.
- No realizará llamadas telefónicas ni envío de mensajes de textos.
- La registración de las llamadas telefónicas deberá realizarse manualmente en el sistema.

En caso que se considere necesario que el sistema envíe correos electrónicos, se supondrá que se dispone de una conexión a internet adecuada y que el proveedor del servicio de correo electrónico brinda un servicio óptimo y sin interrupciones. El control de stock requerirá que un empleado registre en el sistema el consumo de un insumo, con excepción del control de impresiones. No se proveerá de ningún otro producto que no sea el software desarrollado junto con sus manuales correspondientes y código fuente. También se incluirá solo un curso de Capacitación en Uso del Sistema.

Restricciones

El sistema SGB no realizará:

- Liquidación de sueldos de los empleados
- Control de asistencia
- No manejará información impositiva de la actividad económica (cálculos de impuestos, I.V.A, ingresos brutos y demás).
- No poseerá la base de datos de clientes de las obras sociales.
- No realizará llamadas telefónicas ni envío de mensajes de textos.
- La registración de las llamadas telefónicas deberá realizarse manualmente en el sistema.
- En caso que se considere necesario que el sistema envíe correos electrónicos, se supondrá que se dispone de una conexión a internet adecuada y que el proveedor del servicio de correo electrónico brinda un servicio óptimo y sin interrupciones.
- El control de stock requerirá que un empleado registre en el sistema el consumo de un insumo, con excepción del control de impresiones.
- No se proveerá de ningún otro producto que no sea el software desarrollado junto con sus manuales correspondientes y código fuente.
- También se incluirá un curso de Capacitación en Uso del Sistema.

Firmas autorizantes		
Nombre	Fecha	Firma
Daniel Talebi	23/10/2010	_____
Pablo Javier Ortiz	23/10/2010	_____

3.4 Estudio de Viabilidad

Como ya hemos enunciado en los apartados *Inicio y definición del problema*, debemos analizar la información con la que contamos y determinar cuál es el problema, así como detectar la oportunidad. Entonces habiendo pasado por este propósito, nos planteamos determinar hasta qué punto nos resulta interesante este trabajo. Puede que el futuro proyecto no encaje adecuadamente en el tipo de actividad que desempeña la empresa, o puede que las condiciones de alcance, plazo o precio del mismo no se adecuen a nosotros.

Antes de ponerse a preparar una oferta para el cliente hay que tener en cuenta que **Preparar una oferta supone tiempo, esfuerzo y, por tanto: Dinero.**

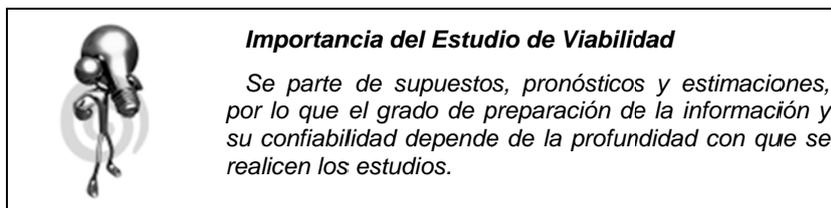
Y no están solo el coste laboral y material de asignar recursos a preparar una oferta. También hay que evaluar el coste de oportunidad de las personas y medios que, de no estar preparando esa oferta, posiblemente estarían trabajando en algún proyecto sí remunerado.

Por eso hay que considerar, a grosso modo, si realmente nos interesa y tenemos posibilidades de conseguir el trabajo.

1. ¿Disponemos de los **conocimientos necesarios** para realizar los trabajos en cuestión y, en caso negativo, podemos obtenerlos a tiempo y a coste razonable?
2. ¿Disponemos de los **recursos materiales y humanos** necesarios para realizar los trabajos, o podemos adquirirlos o subcontratarlos a un coste razonable?
3. ¿Somos **competitivos** en el mercado, y podemos ofrecer al menos lo mismo, o más, que otros posibles ofertantes?
4. ¿Tiene el cliente algún **proveedor favorito**, candidato principal a adjudicarse el contrato?. En caso afirmativo, ¿tenemos alguna posibilidad de batirle, en precio o condiciones (técnicas, financieras, temporales, etc.), y que el proyecto siga siendo rentable?
5. ¿El contrato, en el precio y las condiciones previstas, es consistente y **económicamente rentable**?, es decir, si en dichas condiciones es posible obtener un beneficio de su realización.
6. Concurrir a la oferta y obtener el contrato, ¿qué **coste de oportunidad** tiene?, ¿puede impedirnos atender a otros clientes o a otros proyectos más interesantes?

Aunque alguna de las preguntas anteriores nos pudiera hacer desistir de ofertar, hay que considerar también si: **Obtener el contrato ¿dotaría a nuestra empresa de una mejora competitiva, en términos de adquisición de nuevos conocimientos, nuevas tecnologías, o mejora de la reputación que luego pueda facilitar acceder a nuevos contratos?**

La evaluación de dicha oportunidad se denomina **Estudio de Viabilidad**, y es este el que supone el éxito o fracaso del proyecto; en cierta manera es un proceso de aproximaciones sucesivas, donde se define el tipo de trabajo a realizar, el coste que supondrá realizarlo y el tiempo que se precisará para ello. Y de ahí se extrae el potencial precio de venta y, por tanto, el margen comercial resultante.



Si bien dependiendo de la metodología utilizada, de las características del proyecto y demás variables, el lector puede considerar realizar numerosos estudios que le significarán o no resultados más precisos de esa evaluación exhaustiva de la oportunidad, para los fines de este libro solo veremos los estudios de:

1. **Viabilidad técnica**, en la que analizaremos si estamos lo suficientemente cualificados, desde el punto de vista técnico, y que sabemos cómo realizar las tareas necesarias, y como resolver los posibles problemas que puedan aparecer durante el trabajo.
2. **viabilidad de gestión**, preguntándonos si seremos capaces de terminar el trabajo en el periodo de tiempo adecuado, organizando las tareas oportunamente y gestionando los recursos (humanos y materiales) de la manera correcta.
3. **Viabilidad económica**, estableciendo si todo lo anterior lo realizaremos por un precio adecuado, con que márgenes de ganancia, cuánto costara el proyecto, a cuánto lo venderemos, y cómo será el plan financiero.

Estos tres estudios devengarán en sus respectivos documentos de oferta técnica, de gestión y económica, los cuales se desarrollan en los siguientes apartados. No obstante, cabe aclarar que una oferta correcta, basada en un estudio de viabilidad, puede variar mucho en alcance y esfuerzo. Hay ofertas que se transmiten verbalmente, en cuestión de segundos o minutos, y otras pueden llevar meses, incluso años, e involucran a docenas de personas y miles de horas de trabajo con mucha documentación. Por lo que es importante considerar la decisión de ofertar o no, ya que detrás de una oferta hay mucho trabajo a realizar, que en la mayoría de los casos es absorbido por la empresa que desarrollara el producto y no por el cliente.

Si se decidió ofertar, es importante remarcar que la correcta realización de estos estudios es fundamental ya que nos proveerá de información vital, además habrá que tener en cuenta la

información subjetiva, aspectos más sutiles sobre las relaciones con los actores del proyecto, clientes y demás, y los intereses de la empresa.

El estudio de viabilidad tiene por objetivo **servir de herramienta a nivel de dirección técnica de proyecto para la toma de decisiones ejecutivas en el clima en el cual el proyecto se desarrollará, de manera de determinar la conveniencia o no de presentar la oferta y competir por el contrato.**

El estudio de viabilidad lleva a imaginar varias situaciones ("casos de utilización"). Cada caso permite la evaluación de los riesgos del proyecto.



Estudio de Viabilidad: un tema de expertos

Es recomendable la participación de expertos en el Estudio de Viabilidad, donde su conocimiento empírico será de gran importancia en el resultado final generado por dicho estudio.

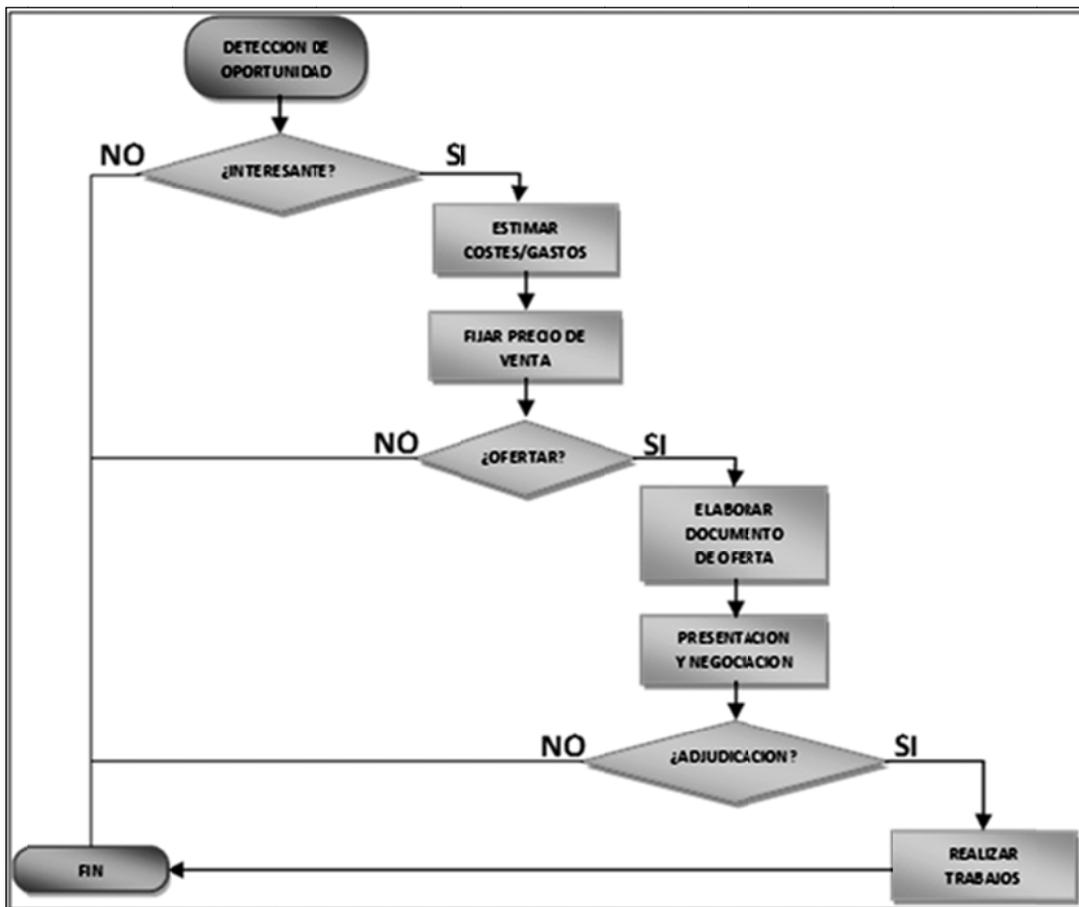


Figura 13. Procedimiento de preparación de una oferta

3.4.1 Oferta Técnica

La Oferta Técnica describe el problema a resolver, cómo pensamos resolverlo, qué riesgos se anticipan, y cómo se abordarán llegado el caso.

Podemos establecer que una forma de desarrollar este documento, más allá de que varíe considerablemente de un proyecto a otro, es comenzando por una descripción sintetizada de nuestra interpretación del problema, una descripción sucinta de en qué consistirá el trabajo; la descripción de las actividades previstas. A partir de la experiencia podemos decir que es recomendable presentarla gráficamente ya que permite una visión rápida y clara de todo lo que se realizará, descripción de los recursos involucrados en la ejecución de los trabajos (materiales, herramientas y sistemas), listar los entregables, los que pueden ser documentos, productos y cualquier otra cosa que se genere como resultado de la realización de una tarea, una breve reseña de los potenciales riesgos que estén fuertemente relacionados a factores externos que puedan comprometer el éxito. Finalmente sobre el escenario mostrar cuales requisitos son satisfechos y cuáles no.

En general más vale excederse ligeramente en extensión y detalle que quedarse corto y dar la sensación de descuido o incapacidad técnica. No obstante el exceso también es malo, y puede abrumar al cliente con detalles de muy bajo nivel, que luego pueden no corresponderse con los reales, que surgen durante la ejecución del proyecto.

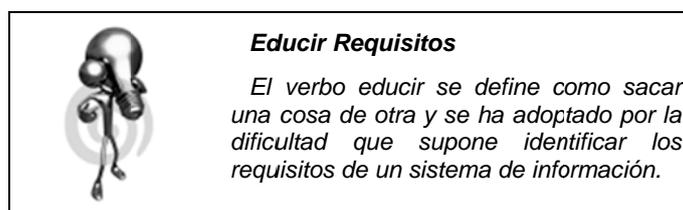
El propósito de esta parte del documento de oferta es **múltiple**:

- Mostrar al potencial cliente que hemos entendido su problema, sus necesidades y sus requisitos específicos.
- Anticiparle una potencial solución (sin llegar a resolver el problema, eso se hará durante el contrato), para mostrarle nuestra capacidad técnica.
- Identificar los puntos conflictivos del trabajo a realizar, y anticipar posibles soluciones.
- Justificar el tiempo y el dinero que consideramos necesario para ejecutar el proyecto.

3.4.1.1 Pasos para alcanzar una oferta técnica

Considerando la figura 13, en este punto nos encontraríamos elaborando el documento de oferta, habiendo decidido ofertar.

La primera aproximación a lo que será el documento de oferta técnica surge a partir de **analizar el trabajo a realizar**, dicho análisis es preliminar.



La manera más habitual de realizar dicho análisis es descomponer el proyecto en actividades independientes, compuestos por tareas y, si es preciso, sub-tareas. Esta descomposición permite:

1. **Analizar** detallada y organizadamente el trabajo a realizar, definiendo el alcance técnico del proyecto en base a la definición del problema.
2. **Educir Requisitos** preliminares en base a la información del proyecto obtenida.
3. **Formalizar los requisitos** previos y los **resultados** a obtener de cada tarea (incluyendo documentación, tareas interrelacionadas, medios materiales, etc.) en una Especificación de Requisitos de Software preliminar.
4. **Validar** la Especificación de Requisitos de Software preliminar generada con el cliente.
5. **Asignar un responsable** a cada tarea.

En la figura 14 observamos (marcado con el círculo) la relación entre la **estimación de gastos / costes y el precio del proyecto** y el documento de oferta técnica que determina claramente el alcance del proyecto.

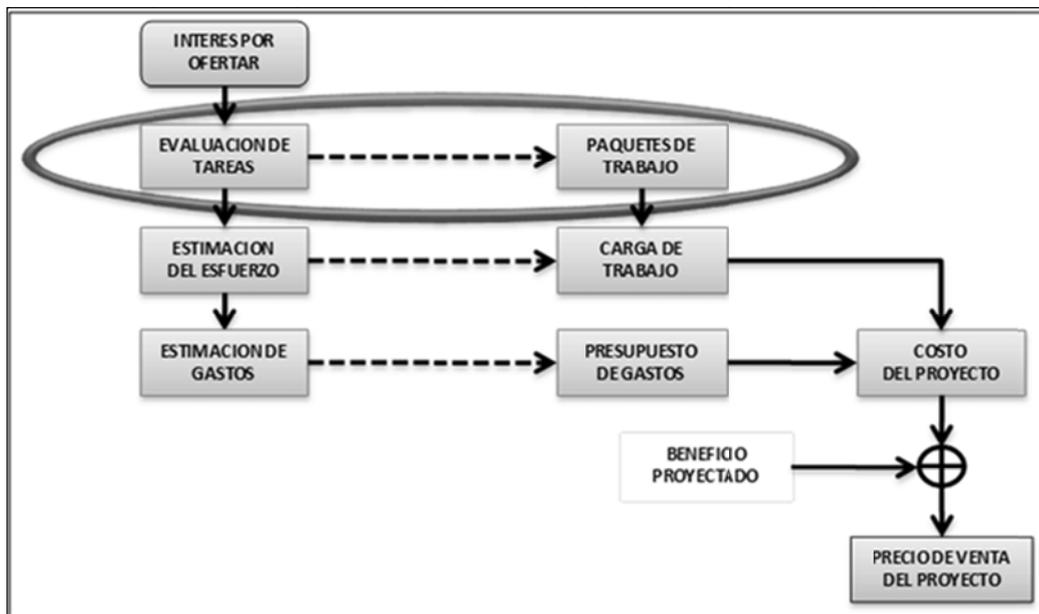


Figura 14. Evaluación de Tareas y Paquetes de Trabajo

Una oferta técnica debe contener esta información como mínimo:

- Introducción, donde se sintetice nuestra comprensión del problema.
- El alcance del proyecto, donde describa en qué consistirá el trabajo.

- Descripción preliminar de los requisitos funcionales y no funcionales del proyecto, que definan claramente lo que el sistema va a realizar.
- Descripción clara de lo que NO realizará el sistema, para evitar confusiones en el futuro.
- Lista de entregables que generará esta oferta técnica.
- Identificación de riesgos potenciales, suposiciones y dependencias.



3.4.2 Oferta de Gestión

3.4.2.1 Descripción

La Oferta de Gestión describe la organización, la metodología y los procedimientos que se seguirán para llevar a cabo el proyecto, indicando claramente las actividades a realizar, los recursos involucrados y el tiempo que se estimará para cada actividad. Este documento debe incluir la información necesaria para justificar la solvencia organizativa y de gestión de nuestra propuesta. Generalmente contendrá:

- Descripción de paquetes de trabajo, y responsable de cada uno de ellos.
- Lista de resultados entregables del proyecto.
- Planificación temporal propuesta: Gantt y/o PERT.
- Plan de reuniones.
- Equipo de trabajo. Perfil profesional del personal con mayor responsabilidad en el proyecto.
- Referencias de la empresa en proyectos similares.

Con respecto a los criterios de valoración de las ofertas, por lo general suele valorarse en términos de la solvencia técnica del equipo de trabajo propuesto, la experiencia de la empresa en trabajos similares y el plazo de ejecución de los trabajos, entre otros factores.

3.4.2.2 Pasos preliminares para alcanzar una oferta de gestión

Considerando las implicaciones de la oferta de gestión, analizaremos la información con la que contamos previamente para generar dicho documento, la cual surge a partir de **estimar el esfuerzo requerido**, dicha estimación es preliminar.

La estimación de horas permite conocer cuál será una de las partidas del coste de la realización del trabajo ofertado. Permite consolidar la información de esfuerzo derivada de la descripción de tareas a sub-tareas que, junto con los demás gastos de proyecto conforma el coste total del mismo.

A la hora de dimensionar el esfuerzo, es conveniente clasificar el mismo según las distintas **categorías profesionales** del equipo de trabajo, pues el coste real es muy distinto para cada una de ellas (una hora de secretaria nos cuesta mucho menos que una hora ingeniero senior).

También es conveniente valorar el esfuerzo teniendo en cuenta las heterogeneidades del equipo de trabajo (no todas las personas rinden igual), y los posibles problemas que pueden surgir, y que hagan que el coste se incremente.

Generalmente se utiliza una planilla tipo matricial la cual permite estimar por separado cada categoría y reducir el error global. Además con ello se facilita simultáneamente el control del avance de las diferentes tareas y la detección de sobreesfuerzos en cualquiera de las actividades.

Este tipo de estimación permite realizar comprobación adicional sobre las hipótesis manejadas, que puedan proporcionar una indicación de que alguno de los pasos realizados no fue correcto. Así por ejemplo:

- El **número total de horas** resultante tiene que ser consistente con el equipo de trabajo previsto y con la duración del proyecto. Como regla general, el esfuerzo total debe de ser **comparable o ligeramente inferior** al producto del número de horas de trabajo disponibles en el periodo de ejecución del contrato, multiplicado por el número de personas que componen el equipo de trabajo.
- La **distribución de la carga de trabajo entre categorías** tiene que ser consistente con el tipo y alcance del proyecto. En un proyecto de desarrollo de software, la mayor parte puede y debe hacerla personal de categoría IJ (ingeniero junior) o inferior.
- Algo mucho más complejo y sutil de determinar, es la **proporción de esfuerzo entre tareas** o, más concretamente entre aéreas funcionales del proyecto. Empíricamente podemos entender que un proyecto dedicado a desarrollar un producto concreto, debería tener asignado la mayor parte del esfuerzo a los paquetes de trabajo relacionados al desarrollo del producto en sí; una valoración muy apartada de esto podría indicar una estimación incorrecta.

Una vez definidas las tareas en la Oferta Técnica, es necesario realizar los siguientes pasos para desarrollar una Oferta de Gestión:

1. Se **estima** el tiempo necesario para realizar cada tarea.
2. Se **clasifica** el equipo de trabajo (responsables asignados a cada tarea) en las diferentes categorías profesionales.
3. Se **agrupan** las tareas (utilizando algún criterio, pudiendo agrupar tareas relacionadas, o algún otro).
4. Se genera **una matriz** la cual tiene dos entradas, las tareas y los responsables de las mismas y en cada intersección asigna el valor estimado. La matriz generada describe el tiempo necesario (total y parcial) en cada caso. A continuación se muestra un ejemplo de dicha matriz.

PT	Descripción de acciones y/o Actividades	Estimación de horas por categoría						TOTAL
		DP	JP	AF	AF	AC	AT	
100	Relevar las variables que influyen en el dominio del proyecto		3	11	11	6	6	37
200	Relevar Subsecretaría de Economía y Hacienda		2	8	8	2	4	24
210	Relevar la normativa interna vigente		10	3	3	10	2	28
220	Relevar las Direcciones intervinientes		6	10	10	5	5	36
230	Relevar los recursos disponibles			20	25		13	58
240	Relevar sistemas existentes		8	45	45		23	121
300	Analizar la información recaudada	3	6	35	35	7		86
400	Crear Plan de Factibilidad	2	8	25	25			60
SubTotal		5	43	157	162	30	53	
TOTAL								450

Figura 15. Matriz de Estimación

5. Se **definen las fechas** de comienzo y fin para cada tarea.
6. Se **definen** dependencias entre las tareas.
7. Se genera un documento de **planificación temporal** (generalmente gráfico, Gantt y/o PERT).
8. Se **estiman los recursos necesarios** para la realización de todas las tareas.
9. Se revisa el proyecto para lograr su optimización utilizando técnicas apropiadas (ejemplo: camino crítico, distribución equitativa de recursos, minimizar gastos, etc.)

Como ya se detalló previamente, para obtener el precio del proyecto, se debe sumar el **coste del proyecto** y el **beneficio proyectado**. Pero a su vez para obtener el **coste del proyecto** debemos sumar la **carga de trabajo** y el **presupuesto de gastos**. En la oferta de gestión trabajamos en estimar dicha **carga de trabajo**.

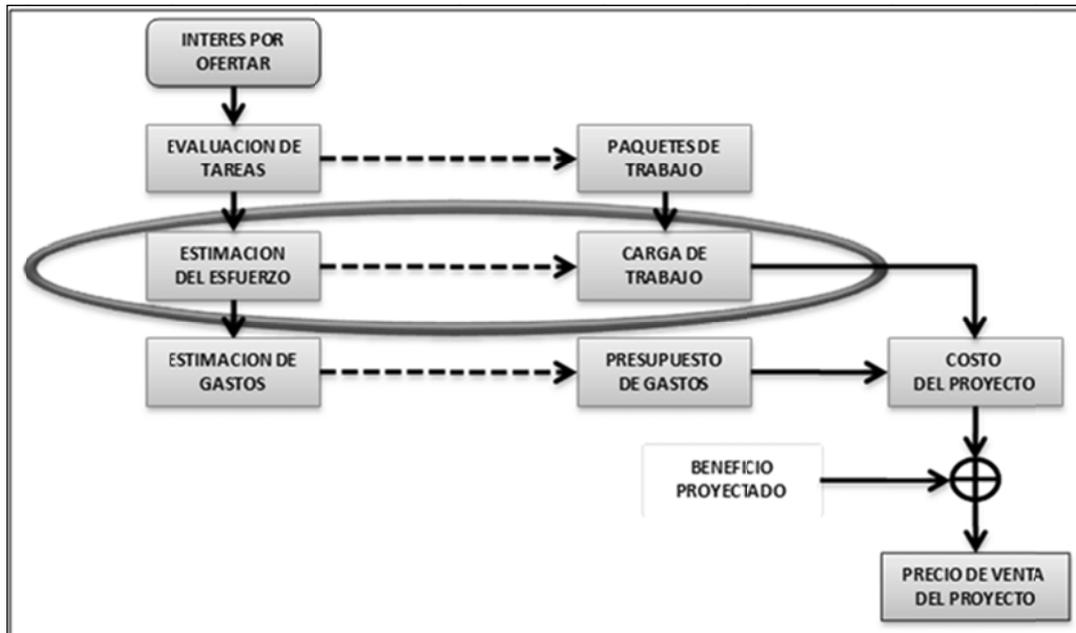


Figura 16. Estimación del Esfuerzo y Carga de Trabajo

3.4.3 Oferta Económica

3.4.3.1 Descripción

La Oferta Económica es uno de los puntos críticos del proyecto desde el lado del cliente. Desde la primera entrevista, el cliente nos dejará saber sus intenciones sobre el conocimiento del monto total del proyecto. Recién en esta etapa estaremos en condiciones de proveer información válida y bien fundamentada sobre el coste de un producto o servicio.

En este documento se puede saber cuánto costará realizar el proyecto, en cuanto podríamos venderlo y cuál es la estrategia financiera para la recuperación de la inversión, entre otros puntos importantes.

Hay que tener en cuenta que muchas ofertas se desechan directamente, sin llegar a mirar el documento técnico ni de gestión, porque el precio no se considera adecuado. Este razonamiento lo puede considerar un cliente si no tiene una oferta técnica o de gestión lo suficientemente convincente como para fundamentar el desembolso económico.

Particularmente, los análisis de viabilidad económica pueden ser de carácter previo, simultáneo o prolongado. Los análisis de carácter previo se limitan al objeto esencial de la toma de decisiones conteniendo un pronóstico de viabilidad. No obstante, en la mayoría de los casos

el análisis es simultáneo; en él no sólo se realiza un pronóstico, sino que se realiza un seguimiento del desarrollo del proyecto incluyendo la propuesta y ejecución de medidas paliativas y correctoras durante la ejecución del proyecto. Esta segunda fase de los análisis de viabilidad corresponde a un nivel que funciona de dirección ejecutiva. Incluso, en determinados casos el análisis económico alcanza al seguimiento del proyecto finalizado, incluyendo los gastos de conservación, y mantenimiento.

Es de mucha utilidad hacer un cálculo estimativo de la inversión y del coste operativo del proyecto (en términos de recursos materiales y humanos), los plazos considerados y las eventuales ganancias generadas por la inversión. Basándose en estos cálculos aproximados, se puede determinar si se debe continuar con el proyecto.



Consejo

*Es altamente recomendable no brindar información sobre el precio del proyecto hasta que **no** esté presentada la oferta técnica y de gestión. Con esta información el cliente podría tener una noción más clara de la complejidad del proyecto y así lograr una mejor predisposición para aceptar el precio final.*

Una Oferta Económica debe contener una estimación económica que a su vez contiene:

- El precio de venta del proyecto básico.
- Precio de las mejoras adicionales al proyecto básico.
- La forma de pago, incluyendo:
 - Distribución.
 - Medio.
 - Financiación.
- Validez de la oferta (lo más normal es que sea válida por un periodo dado).
- Inclusión o no de determinados impuestos, entre ellos el IVA.
- Precio al que se facturarán, los recursos adicionales que el Cliente pueda solicitar, como complemento a los necesarios para la realización del proyecto.

Para obtener el precio del proyecto, se debe sumar el **coste del proyecto** y el **beneficio esperado**. Pero a su vez para obtener el **coste del proyecto** debemos sumar la **carga de trabajo** y el **presupuesto de gastos**. En la oferta económica trabajamos en establecer los montos que configuran el **precio total del proyecto**.

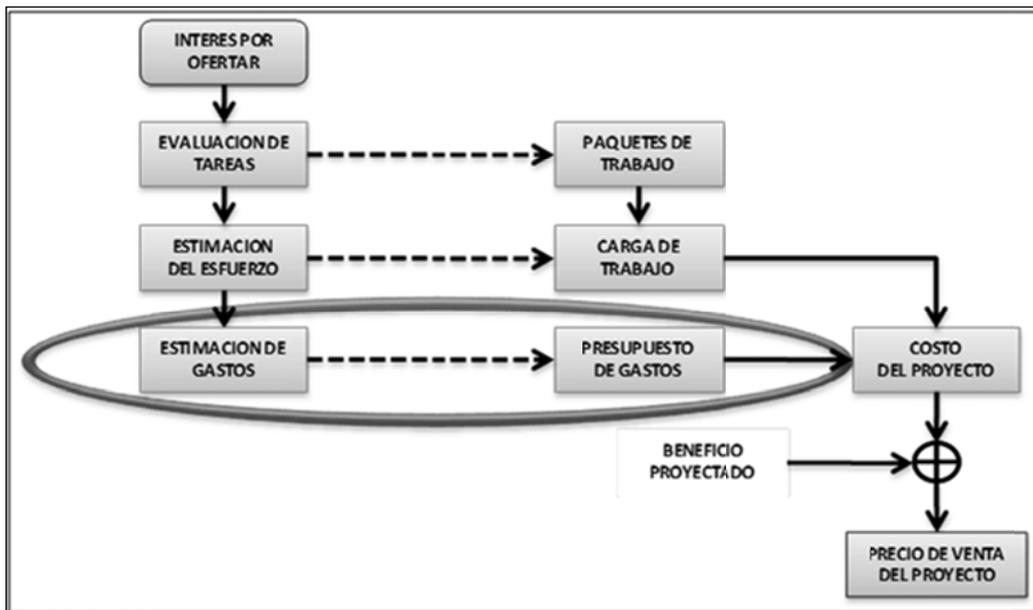


Figura 17. Estimación de Gastos y Presupuesto de Gastos

3.4.3.2 Pasos preliminares para alcanzar una oferta económica

Dentro del desarrollo del documento de oferta económica se incluye la estimación económica dentro de la que habiendo ya definido la estimación del esfuerzo, podremos **estimar los costes**. Dicha estimación es preliminar.

Evaluar el coste de ejecutar un desarrollo de software es una tarea compleja. El alto valor añadido del proceso, el carácter intangible del producto, y la dificultad para adecuar procesos informáticos a necesidades de usuarios (con cierta dosis de subjetividad en muchos casos) hacen que la valoración económica en este tipo de proyectos pueda resultar más compleja que en otros entornos proyectuales.

La estimación económica incluye los costes tanto de materiales y bienes a adquirir a terceros como por viajes y desplazamientos, comidas, gastos financieros, uso y mantenimiento de ordenadores, etc. Cada organización estructura de manera diferente estos costes, por lo general por el tipo de proyectos que realiza, definiendo con más nivel de detalle aquellos tipos de costes más significativos de sus actividades.

3.4.3.3 Secciones de la Estimación Económica

A continuación desarrollaremos tres de las partidas que propone, como clasificación de los diferentes costes, *Domingo Ajenjo* en su libro "Dirección y Gestión de Proyectos". Cabe aclarar que la cuarta partida referente a los costes de personal perteneciente a la organización, ya fue planteada en el apartado de Oferta de Gestión.

- **Subcontrataciones:** Son costes de personal, ajeno a la empresa pero que interviene, bajo nuestra responsabilidad ante el cliente. Por lo general se utilizan dos

modalidades de subcontratación: volumen de trabajo a un precio fijo, o un trabajo concreto a un determinado precio por hora. En cualquier caso, al personal subcontratado no le son aplicables los gastos de la empresa.

- **Costes internos:** Costes de consumibles, servicios informáticos, etc., que son proporcionales al esfuerzo real que se dedique y al consumo de recursos concretos.
- **Otros costes:** En esta partida se incluyen todos los demás costes y gastos que no tienen cabida en los tipos anteriores. Dentro de ella, cada empresa hará las clasificaciones que estime oportunas, como por ejemplo:
 - **Material y equipos:** Son los elementos físicos necesarios para completar el proyecto. Pueden ser para uso propio, o suministros que, por lo general, se le entregan al cliente al terminar el proyecto.
 - **Viajes y estancias:** Son costes en los que incurre el personal para realizar el trabajo, incluyendo los desplazamientos a la oficina del cliente o a cualquier otro sitio que sea necesario, las dietas correspondientes, etc.
 - **Otros gastos:** Se incluyen aquí, por último, los gastos adicionales que no tenían cabida en los apartados anteriores.

El detalle de estos ítems representados en la figura 18, se muestra como estimación económica de la siguiente manera:

1. Considerando los **valores obtenidos (#1)**, por categorías, en la matriz de estimación de esfuerzo (que fue mostrada en la figura del apartado anterior de Oferta de Gestión), **agregaremos los costes unitarios por categoría profesional (#2)**. Obtendremos el coste de cada categoría de manera individual, durante el proyecto. El error de valoración cometido parcialmente será siempre menor que el error de una valoración global.
2. Al agregar los costes parciales de cada categoría se obtiene una **valoración global del desarrollo del proyecto (#3)**.
3. Al punto anterior debe **sumársele costes asociados (#4,#5 y #6)** de gestión y dirección, subcontrataciones, materiales y bienes a adquirir a terceros como por viajes y desplazamientos, comidas, gastos financieros, uso y mantenimiento de ordenadores, etc.
4. Al coste del proyecto alcanzado en el punto anterior deberemos **sumarle un porcentaje de beneficio esperado (#7)**, así como aplicarle porcentajes de tasas del mercado (ejemplo.: I.V.A), con lo que obtendremos el **precio final del producto (#8)**.

ESTIMACION ECONOMICA DEL ESTUDIO						
					PRECIO VENTA (SIN IVA):	26.393 Pesos (Arg)
					ESFUERZO :	367 Horas
					MARGEN :	20 %
1. ESTIMACION DE COSTES Y GASTOS :						
1.A COSTES DE PERSONAL						
CATEG.	HORAS	COSTE/HR	COEF IC.	CONTING.	COSTE	
CAT_1 (DI)	5	50,00	1,20	2,00%	306	
CAT_2 (JP)	43	35,00	1,20	4,00%	1.878	
CAT_3 (AF)	157	30,00	1,20	5,00%	5.935	
CAT_4 (AF)	162	30,00	1,20	5,00%	6.124	
Asesor MSMT	53	35,00	1,20	4,00%	2.315	
SUBTOTAL	367	35,91	1,37	5,42%	14.242	
1.B SUBCONTRATACIONES						
SUBC_1 (AC)	30	25		8,00%	750	
SUBTOTAL					810	
1.C COSTES VARIOS						
ORDENADOR :	367	6,00	1,00	5,00%	2.312	
CONSUMIBLES :	367	1,50	1,00	5,00%	578	
SUBTOTAL		7,50			2.890	
1.D OTROS GASTOS						
MATERIAL / EQUIPO :	538,5				539	
VIAJES / ESTANCIAS :	450				450	
SUBTOTAL				3,00%	1.018	
TOTAL COSTES Y GASTOS					18.961	
2. PRECIO DE VENTA :						
2.A MARGEN						
PORCENTAJE SOBRE	MARGEN	SOBRE	VALOR			
PERSONAL	30,00%	14.242	4.273			
SUBCONTRATACIONES	15,00%	810	122			
COSTES VARIOS	20,00%	2.890	578			
OTROS GASTOS	20,00%	1.018	204			
TOTAL	27,30%	18.961	5.176			
PORCENTAJE FIJO	20,00%	18.961	3.792			
2.B PRECIO DE VENTA						
PRECIO DE VENTA, Pesos, CON IVA al ...	16,00%		26.393			

Figura 18. Estimación Económica

Generalmente se utilizan plantillas predefinidas para realizar la carga de estos valores, las cuales realizan estos cálculos de forma automática.



Consejo

Es altamente recomendable contar con el asesoramiento de profesionales especializados en el tema, ya que hay cuestiones subyacentes que no se verán en profundidad en esta obra, y como dijimos este es un punto crítico para el éxito del proyecto.

3.4.3.4 Entregable – Documento de la Oferta

Si hemos llegado hasta acá, es que hemos decidido presentar la oferta al Cliente y, de esta manera, optar por la adjudicación del contrato.

El documento de oferta es aquel que se remite al potencial Cliente ofreciéndole un bien o servicio a cambio de una contraprestación económica. Por lo general, responde a una solicitud del propio Cliente (mediante petición directa o restringida, etc.), aunque también puede ser una oferta no solicitada (si se detecta que el Cliente tiene una necesidad no cubierta, y estamos en condiciones de satisfacerla. En este caso es interesante evaluar la capacidad real de contratación que tenga el Cliente antes de prepara la oferta).

Llegado el momento de compilar el documento de oferta, de lo que se trata es de mostrar al potencial Cliente que somos el equipo más idóneo para realizar el proyecto, es decir:

- Que estamos suficientemente capacitados, desde el punto de vista técnico, y que sabemos cómo ejecutar las tareas necesarias, y cómo resolver los posibles problemas que puedan aparecer durante el trabajo. Ésta es la **oferta técnica**.
- Que seremos capaces de terminar el trabajo en el periodo de tiempo adecuado, organizando las reuniones intermedias oportunas y gestionando los recursos (humanos y materiales) de la manera correcta. Ésta es la **oferta de gestión**.
- Que todo lo haremos por un precio adecuado. Ésta es la **oferta económica**.

El documento de oferta también sirve como referencia contractual. Todo lo que se dice en el documento de oferta que se va a hacer, incluido en el precio propuesto, después hay que hacerlo.

Debe adecuarse en estructura a lo solicitado (verbalmente o mediante escrito de petición de oferta) por el Cliente pero, en general, es de esperar que contenga, al menos:

- Una **introducción**.
- Los **antecedentes y propósito** del trabajo ofertado.
- El **alcance** práctico del mismo (indicando claramente qué no forma parte del alcance).
- La **oferta técnica**, que mostrará al Cliente la capacidad técnica del ofertante.
- La **oferta de gestión**, incluyendo planificación temporal de la ejecución del trabajo, esfuerzo a utilizar y personal clave.
- La **oferta económica**, indicando claramente cuál es el precio final para el contratante, los demás gastos que pudieran derivarse, así como el momento y la forma de pago del proyecto.

En cuanto a la **extensión** del documento de oferta, varía enormemente en función del tipo de Cliente, la relación mantenida con él en el pasado y, sobre todo, el tipo y volumen del contrato.

En general, es conveniente no dedicar a la preparación de la oferta un esfuerzo excesivo (pues significa coste de oportunidad para otros proyectos). Una regla general puede ser no dedicar más del 5% del volumen del contrato, aunque el carácter estratégico, el interés o las relaciones con el Cliente pueden modificar significativamente el alza dicho umbral.

3.4.4 Caso de Estudio: Estudio de Viabilidad

3.4.4.1 Oferta Técnica

Introducción y Propósito.

El propósito de esta especificación es definir de forma preliminar todas las funcionalidades, restricciones y características que se desean implementar para el Sistema de Gestión Bioquímica que se desarrollará para Clinical Lab.

Este documento está sujeto a revisiones sucesivas por el grupo de usuarios hasta alcanzar la aprobación del mismo, con lo que servirá de base para la realización de las ofertas Técnicas, de Gestión y Económica incluidas Estudio de Factibilidad que se lleva a cabo a pedido de Clinical Lab.

Alcance.

El Sistema se llamará *Sistema de Gestión Bioquímica*. El mismo permitirá administrar los recursos, procesos e información que Clinical Lab utiliza para brindar servicios a sus clientes.

El *Sistema de Gestión Bioquímica* permitirá administrar la recepción de muestras biológicas provenientes de los clientes, a continuación, será posible administrar la información en los procesos de análisis e interpretación. Finalmente, se podrá gestionar la validación de resultados y entrega de los mismos de forma impresa, verbal (solo como excepción y bajo registro) o digital.

En todos los procesos se pondrá especial énfasis en el registro de modificación de la información.

El objetivo del Sistema es gestionar de forma eficaz y eficiente los recursos e información, permitiendo un seguimiento detallado de los procesos a medida que avanzan por las distintas etapas.

El Sistema no incluye la entrega de otro producto que no sea el software desarrollado junto a sus manuales correspondientes y el código fuente, tampoco incluye el mantenimiento continuo ni instalación de algún servicio extra. Se incluirá solo la correspondiente Capacitación en Uso del Sistema.

Definiciones, acrónimos y abreviaturas.

Término	Significado
SGB	Sistema de Gestión Bioquímica
CL	Clinical Lab
Usuario	Personal de CL que interactuará con el Sistema
Cliente	Persona o Entidad que utiliza los servicios de CL
Muestra	Muestra biológica utilizada como elemento de estudio en los análisis clínicos
Práctica	Es la orden de determinación de los valores de parámetros definidos, utilizando un método específico, sobre una muestra determinada brindada por el paciente.
Protocolo	Conjunto de prácticas en una fecha determinada y para un paciente específico
Procedencia	Lugar de donde proviene una muestra. Utilizada para identificar al laboratorio derivante o centro de diálisis.
Tarea	Actividades a desarrollar por personal del laboratorio de manera tal de realizar una práctica sobre una muestra

Resumen.

Este documento posee información de requerimientos del SGB, define de forma preliminar funcionalidades, características principales de usuarios, restricciones, suposiciones, dependencias y requisitos futuros del sistema que se desea construir.

También define todos los requisitos necesarios en esta versión preliminar, categorizándolos por funcionalidad, rendimiento, desarrollo, etc.

Este documento está basado según el Standard de IEEE 830 -1999 para la definición de requisitos de software.

Descripción General – Perspectiva del producto.

El Sistema de Gestión Bioquímica no forma parte de ningún sistema mayor, se proyecta que el mismo cubra todas las necesidades internas de Laboratorio Tucumán solicitadas y validadas durante los meses de Abril y Mayo del 2010.

Descripción General – Funcionalidad del producto.

El Nuevo Sistema de Gestión Bioquímica realizará la administración de los análisis clínicos, gestión de usuarios de tipo empleado y clientes (particulares, laboratorios y centros de diálisis), cuenta corriente de clientes, y facturación, control de stock, registro de novedades y agenda.

En lo referente a los análisis clínicos, el sistema será capaz de registrar la recepción de muestras y su transición a lo largo del procesamiento hasta que los resultados están disponibles para ser informados a los clientes, mediante un informe impreso o publicándolos en Internet. Para los laboratorios clientes se ofrecerá la posibilidad de cargar las prácticas que desean derivar de forma remota (posiblemente mediante internet). Estos análisis generarán un registro en los movimientos diarios del registro de caja, así como en la cuenta asociada a cada cliente, los cuales el sistema se encargará de registrar. Además deberá contar con un sistema de alertas, en el cual si los resultados de las prácticas evidencian un riesgo en la salud del paciente se comunicará de forma urgente, a quien corresponda dicha información, y en lo posible de forma automática. El sistema deberá registrar todas aquellas novedades asociadas a un protocolo o de carácter general.

Es necesario que el sistema cuente con manejo de recordatorios, como medio de ayuda en la realización de las actividades de los empleados.

Así mismo el sistema se encargará de gestionar a los tres tipos de clientes: Particulares (con y sin Obra Social), Laboratorios y Centros de Diálisis, manejando no tan solo la información filiatoria de cada uno de ellos, sino también información sobre su cuenta económica.

Así los dos tipos de usuarios, empleados y clientes, accederán al sistema utilizando nombre identificador y clave. La utilización de las distintas funcionalidades podrá configurarse según los permisos que se desee otorgar a cada usuario, también será posible crear perfiles de usuarios.

El registro de los movimientos de caja deberá incluir la emisión y recepción de facturas y otros tipos de comprobantes.

Del mismo modo, el sistema deberá realizar el manejo de listas de precios para cada práctica de acuerdo a la procedencia que la solicite, y administrar los precios que las obras sociales (que tienen convenio con Laboratorio Tucumán) pagan por las prácticas.

Otra de las funcionalidades del sistema será la asignación de tareas según distintos criterios, a los técnicos o bioquímicos, y el seguimiento del estado de las mismas, el control de stock de los insumos y sus consumos, registro básico de los médicos que ordenan los análisis. En la parte de comunicación con equipos, se deberá proveer de los medios (interfaz) para permitir la carga automática de datos provenientes de los equipos que procesan las muestras, ya sea por conexión directa o indirecta. Así los equipos (o sus intermediarios) deberán cargar los datos en el sistema, y no que el sistema extraiga los datos de los equipos.

En todos los procesos se pondrá especial énfasis en la flexibilidad del sistema ante nuevos cambios.

Igualmente es importante realizar un control de trazabilidad, por lo que se registrará toda operación en la que se modifiquen los datos del sistema, como así también es importante proveer estadísticas para el Sistema de Gestión de Calidad del Laboratorio Tucumán.

Como parte del Sistema, se brindará una capacitación en el uso del software y se entregará los manuales de usuario correspondientes y código fuente.

Características de los Usuarios.

Tipo de usuario	Gerente General
Formación	Título de grado en Bioquímica. Conocimientos de informática, administración de empresas y contabilidad.
Actividades	Validación de resultados de análisis. Control General.

Tipo de usuario	Gerente de Calidad
Formación	Conocimientos básicos de Bioquímica. Conocimientos en Administración de Empresas y Administración de Recursos.
Actividades	Control de Calidad de los Procesos y Personal.

Tipo de usuario	Bioquímico
Formación	Título de grado en Bioquímica.
Actividades	Procesamiento de muestras. Realización de análisis. Interpretación de resultados

Tipo de usuario	Técnico
Formación	Conocimientos de Bioquímica.
Actividades	Preparación y procesamiento de muestras.

Tipo de usuario	Secretaria
Formación	Secundario Completo. Conocimientos básicos de bioquímica.
Actividades	Recepción de clientes y muestras. Información de análisis y resultados. Cobros y reintegros de prácticas.

Restricciones.

El Sistema no realizará:

- Liquidación de sueldos de los empleados.
- Control de asistencia.
- Manejo de información impositiva de la actividad económica (cálculos de impuestos, I.V.A, ingresos brutos y demás).
- Llamadas telefónicas ni envío de mensajes de textos, pero si se podrá registrar manualmente la realización de las mismas.

Otras consideraciones:

- No poseerá la base de datos de clientes de las obras sociales.
- El control de stock requerirá que un empleado registre en el sistema el consumo de un insumo (previamente cargado en el Sistema), con excepción del control de impresiones.
- Los equipos de análisis (o sus intermediarios) deberán cargar los datos en el sistema, y no que el sistema extraiga los datos de los equipos.
- Al importar los datos de una base de datos de un sistema anterior, se solicitará al cliente exportarla en un formato de campos el cual será especificado debidamente en su momento.

El Sistema no incluye la entrega de otro producto que no sea el software desarrollado junto a sus manuales correspondientes y código fuente, tampoco incluye el mantenimiento continuo ni instalación de algún servicio extra. La configuración del

software con los parámetros específicos de la actividad a realizar estará a cargo del cliente. Se incluirá solo un curso de Capacitación en Uso del Sistema.

Suposiciones y dependencias.

El buen funcionamiento de servicios disponibles a través de internet estará ligado a la calidad de la conexión entregada por el proveedor de Internet.

En caso que se considere necesario que el sistema envíe correos electrónicos, se supondrá que el proveedor del servicio de correo electrónico brinda un servicio óptimo y sin interrupciones.

La integridad de los datos proveniente de los distintos equipamientos electrónicos utilizados en el procesamiento y análisis de muestras dependerá en cada caso de la interfaz de comunicación entre el SGB y el equipo o su intermediario. Algunas de las interfaces existentes fueron diseñadas por el fabricante, otras se diseñaron a medida.

Evolución previsible del sistema.

En ésta versión preliminar del documento no se registran requisitos futuros referentes a diseño, desarrollo, metodologías, lenguajes, normas, hardware o software.

Requisitos comunes de las interfaces.

El Sistema de Gestión Bioquímica interactuará con distintos sistemas mediante interfaces de comunicación que se especificarán en las siguientes sub-secciones.

Interfaces de Usuario.

Número de requisito	1.
Requisito	El Sistema deberá proporcionar una interfaz de usuario amigable y flexible orientada a la Web.
Descripción	Se utilizará una interfaz de usuario basada en páginas, menús y formularios.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	2.
Requisito	El Sistema deberá proporcionar interfaces de usuario apropiadas para cada perfil de usuario.
Descripción	Se diseñarán las interfaces según los permisos y funcionalidades requeridas por cada perfil de usuario.
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

Número de requisito	3.
Requisito	El Sistema deberá proporcionar interfaces con buena navegabilidad, evitando ventanas en exceso.
Descripción	La navegabilidad del Sistema deberá permitir a los Usuarios recorrer y cambiar de página de forma rápida. Detalles sobre la navegabilidad serán especificados en una versión futura de éste documento.
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	4.
Requisito	El Sistema poseerá una interfaz intuitiva.
Descripción	El Sistema deberá presentar una interfaz sencilla y entendible que resulte intuitiva a cualquier Usuario.
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Interfaces de hardware.

Número de requisito	1.
Requisito	Se deberá contar con computadoras que posean los puertos adecuados para la conexión con cada equipo analizador o su intermediario.
Descripción	Se analizarán las posibilidades de utilizar la interfaz existente o diseñar una nueva.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	2.
Requisito	El Sistema deberá los elementos necesarios para poder conectarse a internet y estar comunicado en red entre varios terminales.
Descripción	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Interfaces de software.

Número de requisito	1.
Requisito	El Sistema deberá proporcionar una interfaz de software de comunicación para que los equipos (o sus intermediarios) puedan conectarse y cargar los datos automáticamente.
Descripción	Debido a que se busca que la mayoría de las tareas estén automatizadas, se requiere que el Sistema establezca un protocolo de comunicación por el cual los resultados de las prácticas se puedan cargar de forma automática en el sistema. Los equipos a conectarse son todos aquellos disponibles en el laboratorio.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Interfaces de comunicación.

Número de requisito	1.
Requisito	El Sistema deberá poseer una interfaz para el envío de correos electrónicos u otro tipo de comunicación digital, según corresponda.
Descripción	En base a un análisis de mayor profundidad se determinará si estos métodos de comunicación resultan necesarios
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Requisitos Funcionales.

Número de requisito	1.
Requisito	El Sistema deberá administrar Altas, Bajas, Modificaciones y Consultas de Pacientes Particulares.
Descripción	Uno de los tipos de clientes del laboratorio son los particulares, es decir aquellas personas que concurren de manera personal para la realización de estudios. Ellos pueden ser con o sin obra social. Se deberá administrar la información filiatoria que permita una correcta identificación.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	2.
Requisito	El Sistema deberá administrar Altas, Bajas, Modificaciones y Consultas de Laboratorios Clientes.
Descripción	Los laboratorios clientes son aquellos laboratorios que debido a su capacidad y/o complejidad no pueden realizar ciertas prácticas, y por tal motivo derivan la realización de dicho estudio a CL. Al igual que el resto de los clientes, se debe administrar la información filiatoria
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	3.
Requisito	El Sistema deberá administrar Altas, Bajas, Modificaciones y Consultas de Centros de Diálisis.
Descripción	Los Centros de Diálisis son organizaciones que tratan a pacientes con enfermedades renales, quienes deben someterse a controles en determinados períodos. De esta manera el Centro de Diálisis envía un conjunto de muestras (de sus pacientes) para ser analizadas. Normalmente se realizan planes anuales de realización de estudios, debido al gran volumen que representan los mismos.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	4.
Requisito	El Sistema deberá administrar Altas, Bajas, Modificaciones y Consultas de Protocolos.
Descripción	Los Protocolos son representados el conjunto de estudios que deben realizarse para un cliente, en una fecha determinada. La función principal es que las prácticas correspondientes a un mismo protocolo sean validadas en forma conjunta, asegurándose que sean coherentes bajo un criterio bioquímico. Además los protocolos ayudan a la organización operativa, contable, evaluación de calidad, entre otros.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	5.
Requisito	El Sistema deberá registrar la devolución de señas o reintegros de dinero a pacientes particulares con obra social
Descripción	En caso que un cliente (particular con obra social) informe que la autorización de la obra social está en trámite, se le solicita una seña al momento del pago del protocolo. Al momento de traer la autorización de la obra social se le devuelve la seña.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.4.4.2 Oferta de Gestión

Infraestructura Organizativa.

La Entidad Ejecutora es una organización matricial, dedicada a la ejecución de Proyectos, por tal motivo no existe una estructura fija del personal, sino por el contrario los grupos se van conformando a medida que son necesarios, dependiendo de la envergadura del proyecto y los plazos de ejecución. Aun así, en cada proyecto se pueden identificar mínimamente distintos grupos de trabajo que conforman el equipo del proyecto.



Figura 19. Grupos de trabajo que conforman el equipo del proyecto

Grupo de Dirección y Gestión.

- 1 Director de Proyecto: Es el responsable máximo de la ejecución del proyecto, tendrá a su cargo todas las cuestiones globales del proyecto, tales como la

coordinación de los equipos de desarrollo y la gestión en la obtención de los recursos necesarios para el proyecto.

- 1 Líder de Proyecto: es el responsable del equipo de desarrollo, tendrá a su cargo la labor de gestión de los recursos y la asignación de tareas. Es el segundo en la jerarquía de responsabilidad. Manejará la planificación temporal y económica del proyecto.

Grupo de Desarrollo.

- 1 Ingeniero Senior: es el profesional que cuenta con los conocimientos y experiencia necesaria para desarrollar **todas** las actividades en un proyecto.
- 3 Ingenieros Juniors: son aquellos que siendo profesionales no cuentan con los conocimientos y experiencia necesaria para ser un Senior, pero están en proceso de aprendizaje.

Grupo de Análisis.

- 2 Analistas Funcionales: son aquellos que se encargarán de realizar el análisis de la situación actual y relevar cuáles son las necesidades del cliente.

Grupo de Calidad.

- 1 QualityAssurance: es el responsable de realizar las pruebas de calidad en el desarrollo de software, configurando los escenarios de prueba e informando los resultados de dichas pruebas.

Otros.

- 1 Diseñador: es el encargado de realizar el diseño gráfico de la interfaz del sistema.
- 1 IT Junior: es el encargado de asesorar de la configuración de la infraestructura tecnológica existente y de la futura.

En esta configuración mínima planificada para este proyecto se estima que se requerirá un total de 11 profesionales en los 8 puestos de trabajo.

Proceso Productivo.

El proceso de desarrollo del Proyecto comprende las etapas del ciclo de vida de un Proyecto, cumpliendo con las etapas de: Inicio, Planificación, Ejecución, Seguimiento y Control, y Cierre, a la cual se le suma una etapa de Capacitación realizada entre la Ejecución y Control y el Cierre.

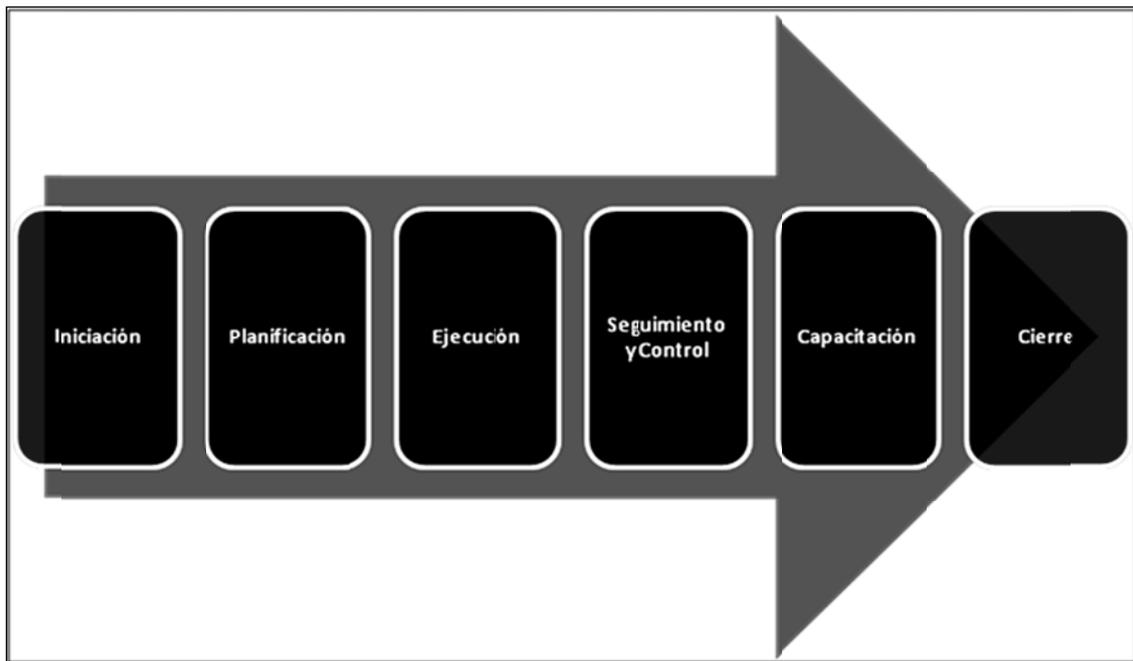


Figura 20. Ciclo de vida del proyecto

Etapas.

- **Iniciación:** Desarrollar el término de abertura del proyecto. Desarrollar la declaración del alcance preliminar del proyecto.
- **Planificación:** Desarrollar el plan de administración del proyecto.
- **Ejecución:** Orientar y administrar la ejecución del proyecto.
- **Seguimiento y Control:** Inspeccionar y controlar el trabajo del proyecto. Control integrado de cambios.
- **Capacitación:** Desarrollar un Curso de Capacitación de las funciones del sistema que se implementaron.
- **Cierre:** Finalizar el proyecto.

Las etapas que componen al proyecto se basan en lo establecido por el Project Management Institute (PMI).

Debido a que el proyecto es de gran envergadura, se dividirá en fases de ejecución secuenciales y con solapamiento.

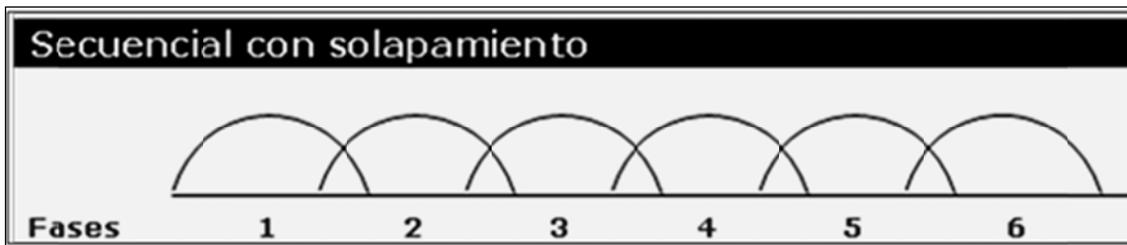


Figura 21. Fases de ejecución de proyectos

Fases.

El Proyecto se encontrará dividido en fases, de manera tal de realizar los avances de una manera organizada y completa. Así se encontrará dividido en 4 Fase:

1. Fase 1: Módulos de Usuarios y Seguridad.
 - Duración Aproximada: 95 Días Hábiles.
2. Fase 2: Módulo de Análisis Clínicos.
 - Duración Aproximada: 176 Días Hábiles.
3. Fase 3: Módulos de Facturación y Control de Stock.
 - Duración Aproximada: 69 Días Hábiles.
4. Fase 4: Módulo de Funciones Varias.
 - Duración Aproximada: 18 Días Hábiles.

Cada Módulo será entregado al cliente para que realice las pruebas correspondientes que aseguren que el producto cumple con todas sus exigencias.

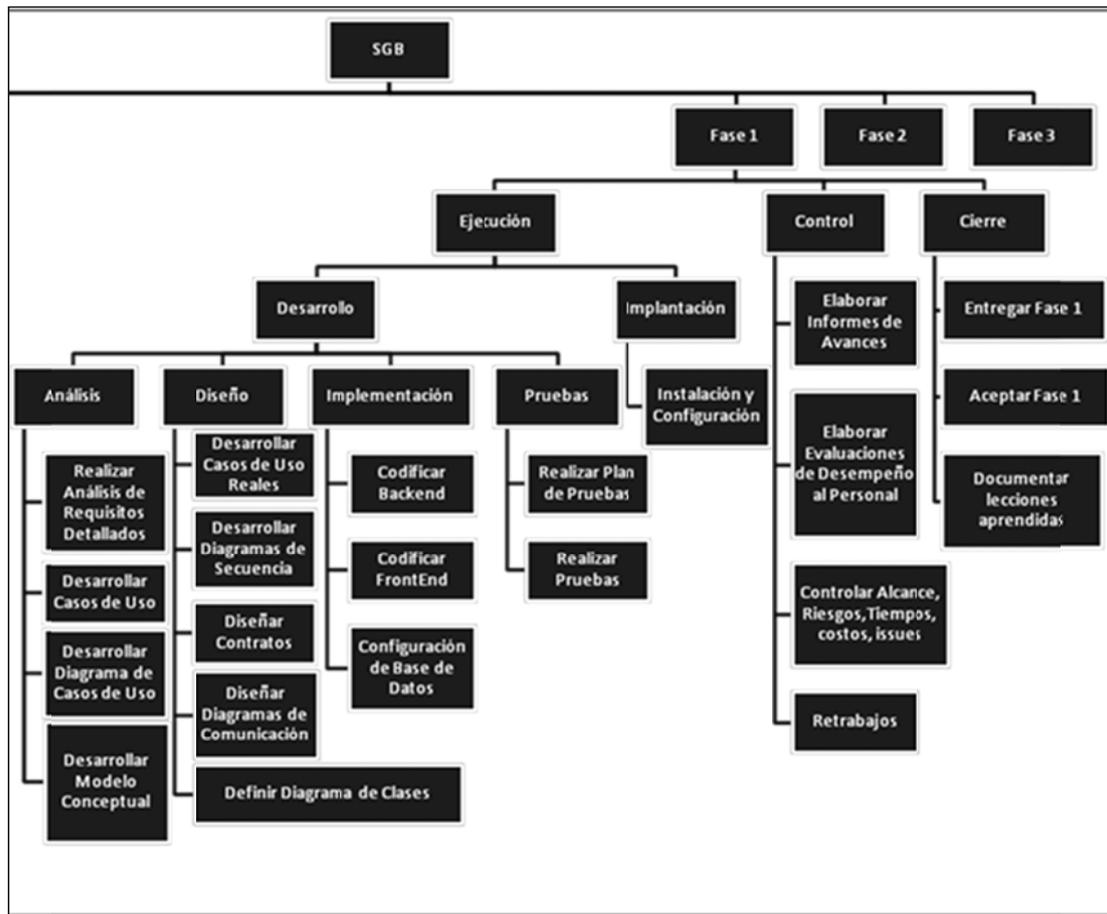


Figura 22. Estructura de Desglose de Trabajo (para mejorar la visualización se suprimió las fases Iniciación y Planificación de la presente figura)

Gantt.

Las actividades descritas en la Estructura de Desglose de Trabajo son organizadas temporalmente considerando los criterios establecidos, generalmente es ilustrado a partir de un diagrama de Gantt.

3.4.4.3 Oferta Económica

- **Precio de Venta:**
 - Precio Final U\$D 115.234.
 - Precio sin IVA U\$D 95.236.
 - IVA U\$D 20.000.
- **Validez de Oferta:** 30 Días
- **Forma de Pago y Financiación:** A definir durante la negociación.

ESTIMACION ECONOMICA DE PROYECTO						
TITULO :	Sistema de Gestión Bioquímica		REFERENCIA :			
CLIENTE :	Clinical Lab		DIRECTOR :	Juan Perez		
FECHA INICIO :	01/01/2012		FECHA FIN :	01/01/2013		
			PRECIO VENTA:	95.236	Dolares	
			ESFUERZO :	2.487	Horas	
			MARGEN :	50	%	
1. ESTIMACION DE COSTES Y GASTOS :						
1.A COSTES DE PERSONAL						
CATEG.	HORAS	COSTE/HR	COEFIC.	CONTING.	COSTE	
Director	124	25,57	1,00	10,00%	3.488	
Lider Proyecto	373	19,43	1,00	10,00%	7.972	
Ing. Senior	293	17,05	1,00	10,00%	5.495	
Ing. Junior	338	11,93	1,00	10,00%	4.436	
A. Funcional	482	11,93	1,00	10,00%	6.325	
Diseñador	377	11,93	1,00	10,00%	4.947	
QA Calidad	344	12,27	1,00	10,00%	4.643	
IT	156	12,95	1,00	10,00%	2.222	
SUBTOTAL	2487	14,45	1,00	10,00%	39.529	
1.B SUBCONTRATACIONES						
SUBC_1					0	
SUBC_2					0	
SUBTOTAL				8,00%	0	
1.C COSTES VARIOS						
ORDENADOR :	2487	4,00	1,00	5,00%	10.445	
CONSUMIBLES :	2487	0,75	1,00	5,00%	1.959	
OTROS :	150	11,00	1,00	5,00%	1.733	

Figura 23. Oferta Económica

3.5 Contrato Informático



Una vez que la viabilidad de nuestro proyecto está demostrada, el siguiente paso será crear un contrato que formalice el trabajo que vamos a realizar. Hay que recordar que no hay proyecto sin contrato. En este momento del libro vamos a desarrollar conceptos básicos de contratos informáticos que un Director de Proyectos debería conocer.

3.5.1 Definición de Contrato Informático

El contrato informático es una contratación cuyo objeto son bienes y servicios informáticos.



Bienes informáticos:

- *Todos aquellos elementos que en conjunto conforman el soporte físico del elemento informático (hardware).*
- *Los bienes inmateriales que proporcionan las órdenes, datos, procedimientos e instrucciones, y que en conjunto conforman el soporte lógico del elemento informático (software).*



Servicios informáticos:

Todos aquellos servicios que sirven de apoyo y complemento a la actividad informática en una relación de afinidad directa con ella.

Los contratos informáticos están formados por elementos dispares que exigen la mezcla o unión de dos o más tipo de contratos para poder configurar sus características, siendo su objeto múltiple y diversificado, pudiendo darse multitud de figuras que desequilibrarían cualquier relación tipo que se pueda pensar. Todo ello debido a la pluralidad de las partes que intervienen y la dispersión de intereses entre ellas, así como a la particularidad de determinadas cláusulas que forman parte de este tipo de contratos.

Asimismo, el desconocimiento por el usuario, en términos generales, de las posibilidades y límites de la informática, hace que no todo en el contrato pueda estar basado en el principio de la autonomía de la voluntad de los contratantes.

En muchas ocasiones, son contratos de adhesión, en los que una de las partes fija las cláusulas del contrato y lo otra se adhiere a las mismas, sin tener posibilidad de modificar ninguna de ellas. Estos contratos de adhesión son producto de la contratación en masa que, frecuentemente, violan los derechos de los consumidores de bienes y servicios informáticos por el gran desequilibrio que se produce al faltar la emisión libre de voluntad por una de las partes en la fijación de las cláusulas del contrato.

En algunos casos, como el de las conocidas contrataciones llave en mano, sería adecuada la aplicación de la teoría del resultado en la contratación informática, en un claro arrendamiento de obra. Ahora bien, ello implica que los resultados se especifiquen en el contrato definiendo cuales son, dentro de unos límites razonables, o dicho de otra forma, cuando la función básica de tratamiento de la información sea cumplida aunque se puedan dar algunos comportamientos de la misma que, sin tener gran carga sobre la aplicación, no sean los adecuados o cuenten con algunos errores o fallos.

En definitiva, la contratación informática en general, adolece de determinadas características que la hacen extremadamente complicada en la redacción de los contratos y en la fijación de los derechos y obligaciones de las partes. A ello hay que añadir a inexistencia de una normativa adecuada a los mismos y la dificultad en la fijación del objeto cuando son contratos complejos. Es por ello, que se deben redactar teniendo en cuenta un equilibrio de prestaciones y evitar en lo posible la existencia de cláusulas oscuras.

3.5.2 Partes de un contrato informático

En la contratación informática se ven involucrados varios elementos, a los que podemos denominar complementarios, que se interrelacionan entre sí.

Existen cuatro partes principales en un contrato informático:

1. Contratantes.
2. Parte expositiva.
3. Cláusulas o pactos.
4. Anexos.

A continuación se detallan estas partes principales de un contrato informático.

3.5.2.1 Los contratantes

Los contratantes son las personas físicas o jurídicas interesadas en realizar el proyecto. No es lo mismo la contratación informática realizada entre profesionales de la informática, que la contratación informática realizada entre un profesional de la informática y un tercero.

Por ello, la identificación y situación profesional de los intervinientes reviste gran importancia, debiendo fijar, no solamente quien adquiere cada responsabilidad proveniente de la contratación y a quien representa, sino también qué conocimientos o formación profesional, o empresarial, relacionada con el tema objeto del contrato, tiene cada uno debido a la obligación existente, desde la óptica de una buena fe contractual, de informar correctamente a la otra parte y de proporcionar claridad a las cláusulas y obligaciones del contrato.

La formación de la voluntad y las responsabilidades de cada una de las partes, tienen una relación con la identificación personal y profesional de las mismas, que la convierten en dato de gran importancia en este tipo de contratos.

3.5.2.2 Parta expositiva

En esta parte se expone, de forma clara y concreta, el por qué y el para qué del contrato. Es importante señalar que dentro de los contratos informáticos es imprescindible fijar de forma sencilla, el por qué se realiza el contrato y cuáles han sido los condicionantes o circunstancias que han movido a las partes a unirse mediante esta relación contractual.

Para ello, se fijarán los intereses de cada cual, especificando las necesidades de uno y la oferta del otro; dejando bien claro que es lo que ofrece una parte y que es lo que acepta la otra y debiendo existir una coincidencia real sobre el objeto, o concepto que de él y de su utilidad respecto al fin perseguido, tienen cada una de las partes.

Por otro lado es de especial interés establecer claramente el negocio jurídico en el cual luego, de acuerdo con la teoría general para ese negocio en el ordenamiento, se pueda subsumir el caso e interpretar el contrato.

3.5.2.3 Cláusulas o Pactos

Partiremos del principio de buena fe y, estableceremos una "obligación" de colaboración en ambos sentidos; el suministrador debe colaborar con el usuario y, lo que es igual de importante, el usuario debe colaborar con el suministrador. Es altamente recomendable destacar la participación activa de ambas partes como requisito fundamental para alcanzar los tiempos estimados.

Además, el usuario debe respetar y seguir las directrices que, respecto al bien contratado y su implementación en el circuito de información, le indique el suministrador y, consecuentemente, utilizar el equipo informático o los programas, siguiendo las instrucciones que, para su óptima utilización, le señale. El suministrador, por su parte, se exonera de responsabilidad en el caso de que exista una anomalía consecuencia del incumplimiento por parte del usuario de estas instrucciones de funcionamiento o manejo.

Estas cláusulas o pactos han de cumplir los siguientes requisitos, aunque son orientativos:

- Obligaciones de las partes, claras y concisas.
- El deber de asesoramiento.
- El cumplimiento del plazo.
- La formación del usuario.
- Prohibición de subarrendar.
- Sustitución del equipo.
- Definición de términos o conceptos oscuros.
- El mantenimiento preventivo.
- Cláusulas de garantía.

3.5.2.4 Los Anexos

Es fundamental que los contratos informáticos vayan acompañados de unos Anexos que incorporados a ellos y con la misma fuerza de obligar, contengan diferentes desarrollos de elementos que forman parte sustancial del contrato.

Es altamente recomendable agregar como anexo el Acta de Constitución del Proyecto y el Plan de Gestión del Proyecto.

3.5.3 Tipos de Contratos Informáticos

Ante la gran diversidad de contratos informáticos que existen en la actualidad, dividiremos su estudio en dos grupos diferenciados. El primero, respecto al objeto, debido a las características especiales de los distintos objetos sobre los que pueden versar estos contratos -ya sea hardware, software, servicios de mantenimiento y formación, o llave en mano- que llevan a la necesidad de su estudio y tratamiento individualizado.

El segundo, respecto al **negocio jurídico**, debido a que los contratos informáticos, más comúnmente realizados, se han llevado a cabo bajo el paraguas protector de una determinada figura jurídica en la que han encontrado acomodo; pero casi todos los casos, ha sido necesario adecuar el objeto del contrato al negocio jurídico realizado.

3.5.3.1 Por el Objeto

Por el objeto del contrato distinguiremos contratos de hardware, contratos de software, contratos de instalación llave en mano y contratos de servicios auxiliares.

1. Contratos de Hardware. En los que hay que conceptualizar como hardware todo aquello que, físicamente, forme parte del equipo, considerando como tal, también, a los equipos de comunicaciones u otros elementos auxiliares para el funcionamiento del sistema que se va a implementar.

2. Contratos de Software. Hay que diferenciar en el momento de analizar una contratación de software, si se trata de un software de base o de sistema, o se trata de un software de utilidad, o de aplicación o usuario, ya que este último, debe responder a unas necesidades particulares, las del propio usuario, el que encarga la aplicación, y que, por tanto, tendrán que quedar claramente especificadas en el contrato; sin embargo, el software de base o sistema y el software de utilidad responden a unas características generales que son las del propio sistema o las de la utilidad a la que sirven y es un producto ya conformado de antemano que no se somete a peticiones o particularidades del usuario.

3. Contratos de instalación llave en mano. En los que irán incluidos tanto el hardware como el software, así como determinados servicios de mantenimiento y de formación del usuario.

4. Contratos de servicios auxiliares. Como pueden ser, el mantenimiento de equipos y programas o la formación de las personas que van a utilizar la aplicación respecto a equipos, sistemas o aplicaciones.

3.5.3.2 Por el Negocio Jurídico

De acuerdo con el negocio jurídico del contrato, existirán tantos tipos de contratos como negocios jurídicos se realicen sobre este objeto. Así, algunos de los más utilizados en el campo de la informática son los llamados de venta, de arrendamiento financiero, de alquiler, de opción de compra, de mantenimiento, de prestación de servicios, de arrendamiento de obra, de préstamo, o de depósito.

1. De venta. Cuando sea un contrato en el que el suministrador, o vendedor en este caso, se obliga a entregar una cosa determinada, un bien informático, y la otra parte, comprador, a pagar por él a un precio cierto (art. 1445 CC). La venta también puede ser de servicios.

2. De arrendamiento financiero. Mediante el que se requiera que participen tres partes, el suministrador, vendedor del equipo informático, una entidad o intermediario financiero que compra el bien, para un tercero que es el usuario, y el usuario del bien que lo poseerá, pero lo tendrá en régimen de arrendamiento financiero hasta que haya cumplido con unas determinadas características o requisitos.

3. De alquiler. El arrendamiento sobre bienes informáticos es un arrendamiento tipo de los regulados en el Código Civil, art. 1543 y ss., caracterizado porque el suministrador se obliga a dar al usuario el goce o uso de un bien informático durante un tiempo determinado y por un precio cierto.

4. De opción de compra. Aunque la opción de compra no está definida en nuestro ordenamiento y solamente se recoge para bienes inmuebles en la legislación hipotecaria (art.14), nuestra doctrina y jurisprudencia la tienen bien delimitada exigiendo que para que exista este tipo de contrato, tienen que darse tres requisitos principales:

- Respecto al optante, que le debe conceder la decisión unilateral de la realización de la opción de compra.
- Precio de compraventa, que debe quedar perfectamente señalado para el caso de que el optante decida acceder a dicha compraventa.
- Plazo del ejercicio de la opción de compra, Debe quedar determinado con claridad en el acuerdo de las partes.

5. De mantenimiento. Puede ser tanto de equipos como de programas, o incluso, mantenimiento integral en el que se puede incluir un servicio de formación, asesoramiento y consulta.

6. De prestación de servicios. En los que incluiríamos análisis, especificaciones, horas máquina, tiempo compartido, programas, etc., que los podríamos clasificar como unos contratos de arrendamientos de servicios. El arrendamiento de servicios se da cuando una parte se obliga con la otra a prestarle unos determinados servicios, con independencia del resultado que se obtenga mediante la prestación.

7. De ejecución de obra, consistente en el compromiso de una de las partes, en nuestro caso el suministrador del bien o servicio informático, a ejecutar una obra, y de la otra parte realizar una contraprestación en pago por la obra llevada a cabo.

8. De préstamo, caracterizado porque una parte entrega a otra el bien informático para que use de él durante un tiempo determinado y lo devuelva una vez cumplido ese tiempo y de Comodato, consistente en un tipo de contrato de préstamo en el que el suministrador transfiere el uso del bien informático prestado. El Código Civil (art. 1740), se refiere al *comodato* como un contrato de préstamo, en el que una de las partes entrega a la otra alguna cosa no fungible para que use de ella por cierto tiempo y se la devuelva, indicando que es esencialmente gratuito. En el caso de que se acuerde entre las partes una retribución, deja de ser comodato para pasar a ser un arrendamiento de cosas.

9. De depósito, que se constituye de acuerdo con lo establecido en el Código Civil (art. 1758), desde que una persona recibe una cosa ajena con la obligación de guardarla y restituirla, siendo un contrato gratuito, salvo pacto contrario (art.1760), pero que en el caso de cumplirse los requisitos establecidos en el Código de Comercio (art.303), se trata de un depósito mercantil, en el que el depositario tendrá derecho a exigir retribución por el depósito, salvo pacto contrario (art.304), con las obligaciones para el depositario de conservación de la cosa, en este caso, del bien informático, de acuerdo con lo establecido en los arts.306 y concordantes del mismo cuerpo legal.

CAPÍTULO IV. FASE PRODUCTIVA

“Nadar siempre es fácil cuando uno está fuera del agua.” (PhD Ing. Alexander Prieto León)

4.1 Descripción de la Fase Productiva

Esta es la Fase de desarrollo del trabajo en sí. Esta fase se divide en cuatro etapas las cuales deben ser ejecutadas según el modelo elegido que puede ir desde un modelo en cascada (no muy frecuente en la actualidad) hasta un modelo iterativo e incremental, incluso un modelo de metodología ágil como puede ser Scrum. Estas etapas son Análisis, Diseño, Implementación y Pruebas. Todas estas etapas son desarrolladas en las siguientes secciones de este capítulo.



Scrum

Scrum es una metodología para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Durante la ejecución del proyecto, se debe poner énfasis en la comunicación para tomar decisiones lo más rápido posible en caso de que surjan problemas. Además, se deberán organizar regularmente (una vez por semana preferentemente) reuniones para administrar el equipo del proyecto, es decir, discutir regularmente el progreso del proyecto y determinar las prioridades para las siguientes semanas. Existen diferentes formatos para representar el reporte semanal. En la figura siguiente se muestra un ejemplo del formato de un reporte semanal.

Proyecto	ID	Responsable	Confeccionó
	1		
Documento			Fecha
Reporte de Trabajo Semanal			

Reporte de Trabajo Semanal

A. Periodo:

Periodo
Desde: 01/01/2009 **Hasta:** 08/01/2009

B. Proyecto:

Modulo	Estado	Actividades	Próximos Pasos

El color de la fuente de la Columna Estado será:
Verde: Si el proyecto avanza sin problemas
Amarillo: Si el proyecto tiene algún alerta o potencial problema futuro
Rojo: Si surgió algún problema grave o escasez de recursos en el proyecto

Figura 24. Formato de reporte semanal

En esta fase se ejecutarán aquellos procesos necesarios para completar el trabajo definido en el plan para la dirección del proyecto a fin de cumplir con las especificaciones del mismo. Este grupo de proceso implica coordinar personas y recursos, así como integrar y realizar las actividades del proyecto de conformidad con el plan para la dirección del proyecto.

Durante la ejecución del proyecto, los resultados (desempeño hasta el momento) pueden requerir que se actualice la planificación y que se vuelvan a definir las fechas claves. Esto puede incluir cambios en la duración prevista de las actividades, cambios en la disponibilidad y productividad de recursos, así como en los riesgos no anticipados. Tales variaciones pueden afectar el plan para la dirección del proyecto o los documentos del proyecto, y pueden requerir un análisis detallado y el desarrollo de respuestas de dirección de proyectos apropiadas. Los resultados del análisis pueden generar la solicitud de cambios que, en caso de ser aprobados, podrían modificar el plan para la dirección del proyecto u otros documentos del proyecto.



Dirigir y Gestionar la Ejecución del Proyecto

Es el proceso que consiste en ejecutar el trabajo definido en el plan para la dirección del proyecto, para cumplir con los objetivos del mismo.

Las actividades que abarcan la Dirección y Gestión de la Ejecución del proyecto son:

- Realizar las actividades necesarias para cumplir con los requisitos del proyecto.
- Crear los entregables del proyecto.
- Reunir, capacitar y dirigir a los miembros del equipo asignado al proyecto.
- Obtener, gestionar y utilizar los recursos, incluyendo materiales, herramientas, equipos e instalaciones.
- Implementar los métodos y normas planificados.
- Establecer y gestionar los canales de comunicación del proyecto, tanto externos como internos al equipo del proyecto.
- Generar la información relevante del proyecto, tal como coste, cronograma, avance técnico y de calidad y el estado, a fin de facilitar las proyecciones.
- Emitir las solicitudes de cambio y adaptar los cambios aprobados al alcance, a los planes y al entorno del proyecto.
- Gestionar los riesgos e implementar las actividades de respuesta a los mismos.
- Recopilar y documentar las lecciones aprendidas e implementar las actividades aprobadas de mejora del proceso.

El director del proyecto dirige el desempeño de las actividades planificadas del proyecto. El proceso *Dirigir y Gestionar la Ejecución del Proyecto* se ve directamente afectado por el área de aplicación del mismo. **Los entregables se producen como salidas de los procesos ejecutados para cumplir con el trabajo planificado y programado.**

En las próximas secciones se detallarán todas las etapas y actividades incluidas en la Fase de Ejecución.



Experiencia en la Dirección de Proyectos

El conocimiento empírico es un elemento muy importante para la dirección de proyectos efectiva.

4.2 Análisis: Ingeniería de Requisitos

4.2.1 Descripción

Aquí comienza el desarrollo propiamente dicho. En esta primera etapa se realiza un análisis profundo del sistema para determinar claramente **QUÉ** es lo que va a hacer el sistema. La Fase de Análisis consiste en el estudio del sistema de información, en el contexto en que se encuentra en ese momento, y la definición de las necesidades que poseen los usuarios, y que deberán ser satisfechas por medio del software a desarrollar.

Se deberán buscar las respuestas para preguntas tales como:

- ¿Qué es lo que debe hacer el sistema?
- ¿Qué es lo que debe hacer el software a desarrollar?
- ¿En qué condiciones debe funcionar?
- ¿Qué limitaciones tendrá?
- ¿Cómo se implantará?
- ¿Qué recursos materiales, temporales y humanos hacen falta para ello?
- ¿Cómo se verificará que el sistema funciona correctamente?

Ingeniería de requisitos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado. **Su principal propósito es hacer que los requisitos mismos alcancen un estado óptimo antes de llegar a la fase de diseño en el proyecto.** Los requisitos deben ser medibles, comprobables, sin ambigüedades o contradicciones, etc. El sentido común dice que no puedo diseñar si no conozco claramente QUÉ va a hacer el sistema.

Desde un punto de vista conceptual, las actividades en la etapa de análisis son cinco:

1. **Extracción/Educción de requisitos:** consiste en hallar e identificar los requisitos que debe satisfacer un determinado sistema de información. Esto se realiza por medio de entrevistas o comunicación con clientes o usuarios, para saber cuáles son sus deseos. El verbo *educir* se define como *sacar una cosa de otra* y se ha adoptado por la dificultad que supone identificar los requisitos de un sistema de información. Aunque aparentemente dichos requisitos vienen dados por el cliente, la realidad es que la mayoría de ellos deben ser investigados por el analista.
2. **Analizar requisitos:** detectar y corregir las falencias comunicativas, transformando los requisitos obtenidos de entrevistas y cuestionarios, en condiciones apropiadas para ser tratados por el diseño.

3. **Documentar requisitos:** igual que todas las etapas, los requisitos deben estar debidamente documentados.
4. **Verificar los requisitos:** consiste en comprobar el correcto funcionamiento de un requisito en la aplicación.
5. **Validar los requisitos:** comprobar que los requisitos implementados se corresponden con lo que inicialmente se pretendía.

Un **analista de sistemas** es aquel individuo responsable de investigar, planear, coordinar y recomendar opciones de software y sistemas para cumplir los requerimientos de una empresa de negocios. Juega un rol vital en el proceso de desarrollo de los sistemas. Un analista de sistemas exitoso debe adquirir cuatro habilidades: analítica, técnica, gerencial, e interpersonal. Las habilidades analíticas permiten al analista de sistemas entender a la organización y sus funciones, las cuales le ayudan a identificar oportunidades, analizar y resolver problemas. Las habilidades técnicas ayudan al analista de sistemas a entender el potencial y las limitaciones de las tecnologías de la información. El analista de sistemas debe ser capaz de trabajar con varios lenguajes de programación, sistemas operativos, y plataformas hardware de computadoras. Las habilidades gerenciales ayudan al analista de sistemas a administrar proyectos, recursos, riesgos, y cambio. Las habilidades interpersonales ayudan al analista de sistemas a trabajar con los usuarios finales así como con analistas, programadores, y otros profesionales de los sistemas.



Analista Funcional

El análisis de un sistema informático debe ser realizado por una persona con experiencia y capacidad de análisis y educación de requisitos. Para este rol se requiere de habilidades y aptitudes especiales para comprender e inferir los requisitos del sistema. Este rol es comúnmente conocido como Analista Funcional.

Es muy común encontrar también el término *Analista Orgánico*, que se diferencia del Analista Funcional. El analista se encarga del estudio de los requisitos, objetivos y funciones a realizar por el sistema, documentando todos estos aspectos según la metodología establecida. El Analista Funcional es el que se encarga de la fase de análisis propiamente dicha, y el Analista Orgánico se encarga de la fase de diseño, por lo que en algunos medios se emplea simplemente el término "Diseñador" para esta categoría.



Analista Orgánico

El analista orgánico está más cerca del desarrollo, del diseño. Su labor es diseñar lo definido por el analista funcional. Documenta su trabajo y el resultado de su trabajo es lo que emplean directamente los programadores.

4.2.2 Técnicas principales para la Educación de Requisitos

La Educación de requisitos puede ser un proceso largo y arduo para el que se requiere de habilidades y aptitudes especiales para poder obtener los requisitos. Los nuevos sistemas cambian el entorno y las relaciones entre la gente, así que es importante identificar a todas las personas implicadas, considerar sus necesidades y asegurar que entienden las implicaciones de los nuevos sistemas. Los analistas pueden emplear varias técnicas para obtener los requisitos del cliente.

Técnicas de Educación	Descripción
Entrevistas	Las entrevistas son un método común. Por lo general no se entrevista a toda la gente que se relacionará con el sistema, sino a una selección de personas que represente a todos los sectores críticos de la organización, con el énfasis puesto en los sectores más afectados o que harán un uso más frecuente del nuevo sistema. Los requisitos que surgen de las entrevistas a menudo se contradicen unos a otros o se formulan desde la ignorancia de los detalles del funcionamiento del sistema, sus potencialidades, interdependencias o limitaciones; por lo que se debe trabajar con los mismos para corregir sus fallos. Las entrevistas pueden ser personales o grupales. También puede ser abiertas o cerradas.
Cuestionarios	Esta técnica consiste en elaborar un conjunto de preguntas para obtener información sobre un determinado tema. Las preguntas podrán ser abiertas o cerradas. Es importante determinar claramente las personas que deberían responder al cuestionario, estas deberían ser una muestra representativa del todo.
Estudio de Documentación	Esta técnica consiste en estudiar los documentos existentes tales como formularios, archivos de auditoría (logs) y la documentación del sistema informático. Toda documentación sobre la cual podrían surgir requisitos, deberá ser cuidadosamente estudiada por los analistas.
Lluvia de Ideas	Esta técnica consiste en juntar a un grupo de personas idóneas sobre un determinado tema, crear un ambiente de estimulación y provocar que las ideas fluyan. Es importante no criticar las ideas cuando se aplica esta técnica. En primera instancia se documentan todas las ideas, luego se las analiza y finalmente se documentan las ideas que tengan buen fundamento para ser consideradas.
Observación	Esta tarea consiste en la observación de las tareas que realiza el usuario, estos no son siempre conscientes de las tareas que realizan y como las realizan. Esta técnica consume tiempo, pero puede ser muy útil cuando se quiere discernir sobre la complejidad de la tarea en un área. El observador actúa pasivamente sin tener intervención en el sistema.
Demostración de tareas	Es una variante de la entrevista y la Observación. Se le pide a un usuario que muestre como realiza una tarea específica. El Observador puede consultar lo que crea conveniente durante la demostración, es decir que tiene una aptitud activa con respecto al sistema.
Prototipos	Esta técnica consiste en desarrollar una versión simple de una parte del sistema. El usuario validará los requisitos por medio de interacción con el prototipo. Es importante que el usuario sepa que el prototipo no es el producto final, sino sólo la representación de una parte del mismo. El prototipo podrá ser: <ul style="list-style-type: none"> • Incremental: que luego podrá mejorarse hasta convertirse en la solución final. • Desechable: que será parte de la solución final, solo una herramienta para validar requisitos

Figura 25. Técnicas de educación



Educcion de Requisitos

Se refiere a la captura y descubrimiento de los requisitos. Es una tarea más humana que técnica, en la que se identifica a los interesados y se establecen las primeras relaciones entre ellos y el equipo de desarrolladores.

Cuando sea necesario, el analista empleará una combinación de las técnicas de Educción para establecer los requisitos exactos de las personas implicadas, para producir así un sistema que resuelva las necesidades del negocio.



Preguntas Abiertas y Cerradas

*Una pregunta abierta es una pregunta que admite un número **ilimitado** de respuestas. Por ejemplo: “¿Cómo cree usted que debería funcionar el sistema? “*

*Una pregunta cerrada es una pregunta que admite un número **limitado** de respuestas. Por ejemplo: “¿Cuántas opciones deben estar disponibles en el submenú Ayuda? “*

4.2.3 Especificación de requisitos del software o ERS

Una especificación de requisitos software es una descripción completa del comportamiento del sistema a desarrollar. Incluye un conjunto de casos de uso que describen todas las interacciones que se prevé que los usuarios tendrán con el software. También contiene requisitos no funcionales (o suplementarios). Los requisitos no funcionales son los requisitos que imponen restricciones al diseño o funcionamiento del sistema (tal como requisitos de funcionamiento, estándares de calidad, o requisitos del diseño).



Formalizar los Requisitos en la IEEE-830

Las estrategias recomendadas para la especificación de los requisitos del software están descritas por el IEEE 830-1998. Este estándar describe las estructuras posibles, contenido deseable, y calidades de una especificación de requisitos del software.

Principalmente, los requisitos se dividen en:

- **Funcionales:** son los que el usuario necesita que efectúe el software. Ej: “el sistema deberá validar si el usuario está autorizado para realizar ventas”.
- **No funcionales:** especifican cuan “bien” un sistema debe realizar sus funciones: Ej: “el sistema deberá imprimir 20 registros por segundo”.

Una **mala** Especificación de Requisitos provoca:

- Que el sistema no satisfaga a los usuarios.
- Desacuerdo entre usuarios y desarrolladores.
- Puede ser imposible demostrar si el software cumple o no los requisitos. Esto puede traer consecuencias legales.
- Gastos innecesarios en tiempo y dinero en construir un software no idóneo.

El análisis es una tarea crítica dentro del desarrollo de software, no existe otra tarea que con mayor capacidad de afectar al sistema cuando se hace mal. Es crítico encontrar aquí lo errores, ya que si no son encontrados en esta fase, se puede propagar por todo el ciclo de desarrollo aumentando los costes para reparar. Cuanto más tarde se descubren los errores en la Especificación de requisitos, más crecen los costes de forma exponencial.

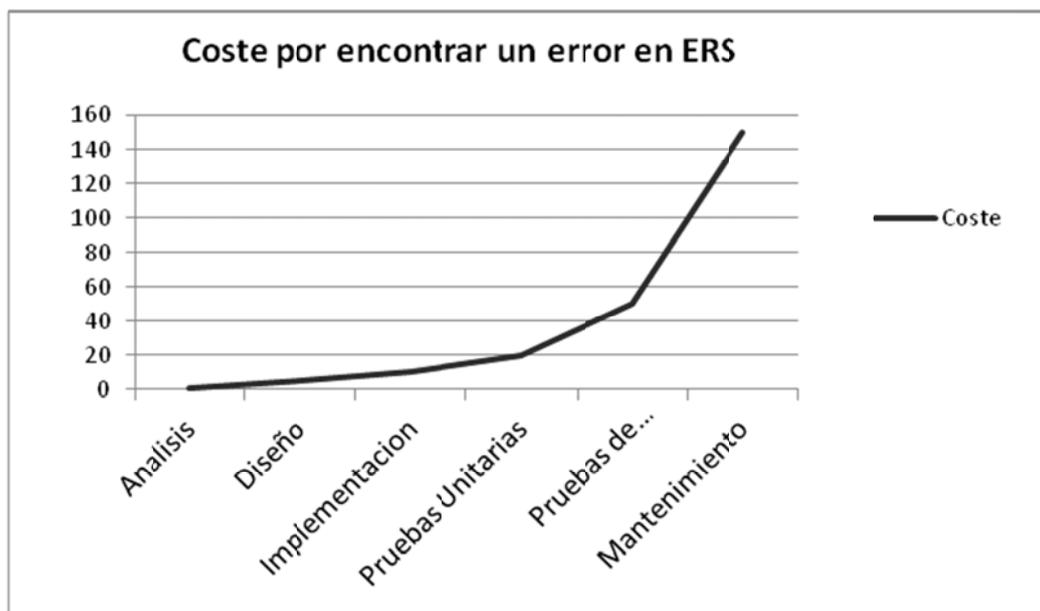


Figura 26. Coste por encontrar un error en ERS

4.2.4 Identificación de las personas involucradas

Debido a que los cambios que introduce un sistema nuevo tienden a afectar a más de un tipo de usuario, los analistas de requisitos han de tomar en consideración a todos los implicados para que se obtengan y depuren sus requisitos de la forma más exacta posible. Entre los implicados hay que considerar:

- Organizaciones que integran la organización del analista que está diseñando el sistema.
- Organizaciones o sistemas de respaldo.
- Dirección.
- Usuarios.

4.2.5 Problemas para obtener los requisitos

4.2.5.1 *Relacionados con las personas involucradas*

Estos son algunos de los típicos problemas que podrán dificultar la determinación de los requisitos:

- Los usuarios no tienen claro lo que desean.
- Los usuarios no se involucran en la elaboración de requisitos escritos.
- La comunicación con los usuarios es lenta.
- Los usuarios no participan en revisiones o son incapaces de hacerlo.
- Los usuarios no comprenden los problemas técnicos.
- Los usuarios no entienden el proceso del desarrollo.
- Los usuarios insisten en nuevos requisitos después de que el coste y la programación se hayan fijado.

Esto puede conducir a la situación donde las exigencias del consumidor cambian, incluso cuando el desarrollo del producto ya está en marcha.

4.2.5.2 *Relacionados con los analistas*

La correcta redacción de las Especificaciones de requisitos Software es imprescindible para el correcto desarrollo del proyecto. Por ello, en su redacción hay que **evitar**:

- Uso de terminología ambigua en la redacción de los documentos de requisitos.
- Sobre especificación de los requisitos.
- Escritura poco legible, voz pasiva, abuso de negaciones.
- Uso de verbos en condicional, expresiones subjetivas.
- Ausencia de términos y verbos del dominio de la aplicación.
- Que el personal técnico y los usuarios finales puedan tener diversos vocabularios y puedan llegar a creer incorrectamente que están de acuerdo, no dándose cuenta del desacuerdo hasta que se provee el producto final.

4.2.5.3 Relacionados con los desarrolladores

Los problemas posibles causados por los desarrolladores durante el análisis de requisitos son:

- Los desarrolladores pueden intentar encajar el sistema en un modelo existente, en vez de desarrollar un sistema adaptado a las necesidades del cliente.
- El análisis de requisitos se puede realizar a menudo por los ingenieros o programadores, en vez de personal con las habilidades de relación con la gente y el conocimiento apropiados para entender las necesidades de un cliente correctamente.

4.2.6 Entregable – Especificación de Requisitos de Software

4.2.6.1 Introducción

Se presenta el formato de Especificación de Requisitos Software según la versión de 1998 del estándar IEEE 830. En la IEEE se indica que un buen documento de requisitos debe contemplar toda la información presentada en el estándar y, aunque propone una organización de dicha información, no exige estrictamente este formato.

Como resultado de esta fase se debe producir un documento de especificación de requisitos en el que se describa lo que el futuro sistema debe hacer. Por tanto, no se trata simplemente de una actividad de análisis, sino también de síntesis. Asimismo, se define requisito como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado. Esta definición se extiende y se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación.

En la determinación de los requisitos no sólo deben actuar los analistas, es muy importante la participación de los propios usuarios, porque son éstos los que mejor conocen el sistema que se va a automatizar. Así pues, el documento de especificación de requisitos debe ser legible por el cliente, con lo que se evita el malentendido de determinadas situaciones, ya que el cliente participa activamente de la extracción de dichos requisitos. Basándose en estos requisitos, el ingeniero de software procederá al modelado de la futura aplicación.

4.2.6.2 Objetivos de la ERS

Los principales objetivos que se identifican en la especificación de requisitos software son:

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software: El cliente debe participar activamente en la especificación de requisitos, ya que éste tiene una visión mucho más detallada de los procesos que se llevan a cabo. Asimismo, el cliente se siente partícipe del propio desarrollo.
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente: En muchas ocasiones el cliente no sabe exactamente qué es lo que quiere. La ERS

permite al cliente definir todos los requisitos que desea y al mismo tiempo los desarrolladores tienen una base fija en la que trabajar. Si no se realiza una buena especificación de requisitos, los costes de desarrollo pueden incrementarse considerablemente, ya que se deben hacer cambios durante la creación de la aplicación.

- Servir de base para desarrollos de estándares de ERS particulares para cada organización:
 - Cada entidad puede desarrollar sus propios estándares para definir sus necesidades. Una buena especificación de requisitos software ofrece una serie de ventajas entre las que destacan el contrato entre cliente y desarrolladores (como ya se ha indicado con anterioridad), la reducción del esfuerzo en el desarrollo, una buena base para la estimación de costes y planificación, un punto de referencia para procesos de verificación y validación, y una base para la identificación de posibles mejoras en los procesos analizados.
 - La ERS es una descripción que debe decir ciertas cosas y al mismo tiempo debe decir las de una determinada manera. Una ERS forma parte de la documentación asociada al software que se está desarrollando, por tanto debe definir correctamente todos los requerimientos, pero no más de los necesarios. Esta documentación no debería describir ningún detalle de diseño, modo de implementación o gestión del proyecto, ya que los requisitos se deben describir de forma que el usuario pueda entenderlos. Al mismo tiempo, se da una mayor flexibilidad a los desarrolladores para la implementación. Así pues, el grado y el lenguaje utilizado para la documentación de los requisitos estarán en función del nivel que el usuario tenga para entender dichas especificaciones.

4.2.6.3 **Características de una buena ERS**

Las características deseables para una buena especificación de requisitos software que se indican en el IEEE son las siguientes:

- **Correcta.**
 - La ERS es correcta si y sólo si todo requisito que figura en ella refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
- **No ambigua.**
 - Un documento es no ambiguo si y solo si cada requisito descrito tiene una única interpretación. Cada característica del producto final debe ser descrita utilizando un término único y, en caso de que se utilicen términos similares en

distintos contextos, se deben indicar claramente las diferencias entre ellos. Incluso se puede incluir un glosario en el que indicar cada significado específicamente. Los analistas deben poner un cuidado especial a la hora de especificar los requisitos. El hecho de utilizar el lenguaje natural para hacer la ERS comprensible a los usuarios supone un riesgo muy elevado, porque el lenguaje natural puede llegar a ser muy ambiguo.

- **Completa.**

- Debe incluir todos los requisitos significativos del software (relacionados con la funcionalidad, ejecución, diseño, atributos de calidad o interfaces externas).
- Debe existir una definición de respuestas a todas las posibles entradas, tanto válidas como inválidas, en todas las posibles situaciones.
- Debe cumplir con el estándar utilizado. Si hay alguna parte del estándar que no se utiliza, se debe razonar suficientemente el porqué no se ha utilizado dicho apartado.
- Deben aparecer etiquetadas todas las figuras, tablas, diagramas, etc, así como definidos todos los términos y unidades de medida empleados.

La ERS debe ser siempre completa, aunque en ocasiones esto no será posible. Por ejemplo si todavía no se han determinado los formatos de los informes finales o por cualquier razón se está esperando la publicación de un reglamento sobre impuestos.

- **Verificable.**

- Un requisito se dice que es verificable si existe algún proceso no excesivamente costoso por el cual una persona o una máquina pueda chequear que el software satisface dicho requerimiento.
- Ejemplos:

No verificables	<ul style="list-style-type: none"> • El producto debería funcionar bien. • El producto debería tener una buena interfaz de usuario.
Verificable	<ul style="list-style-type: none"> • La salida se suministra dentro de los 20 segundos siguientes al evento E el 60% de las veces, y en los 30 segundos siguientes en el 100%.

Figura 27. Ejemplo de atributos Verificables/No verificables

- **Consistente.**

- Es consistente si y solo si ningún conjunto de requisitos descritos en ella son contradictorios o entran en conflicto. Se pueden dar tres casos:

1. Requisitos que describen el mismo objeto real utilizando distintos términos.
2. Las características especificadas de objetos reales. Ejemplo: Un requisito establece que todas las luces son verdes y otro que son azules.
3. Conflicto lógico o temporal entre dos acciones determinadas. Se llega a un punto en el que dos acciones serían perfectamente válidas. Ejemplo: ¿sumar o multiplicar?

- **Clasificada.**

- Los requisitos deben estar clasificados en la ERS. No todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por diversos criterios:
 - Importancia: Pueden ser esenciales, condicionales u opcionales.
 - Estabilidad: Cambios que pueden afectar al requisito.
- Lo ideal es el establecimiento de prioridades, de modo que la implementación de un requisito de menor prioridad no emplee excesivos recursos.

- **Modificable.**

- El documento debe permitir que cualquier cambio pueda realizarse de manera fácil, completa y consistente. Para ello, es deseable tener una organización coherente y fácil de usar en la que aparezca el índice o una tabla de contenidos fácilmente accesible. También es deseable evitar la redundancia, es decir que no aparezca un mismo requisito en más de un lugar de la ERS. No es un error, pero si se tiene que modificar alguna cosa será mucho más cómodo si no tenemos que buscar el mismo requisito en varios lugares.

- **Explorable (trazabilidad).**

- El documento debe permitir determinar el origen de cada requisito de manera clara tanto hacia atrás (origen que puede ser un documento, una persona etc.) como hacia delante (componentes del sistema que realizan dicho requisito). Cuando un requisito de la ERS representa un desglose o una derivación de otro requisito, se debe facilitar tanto las referencias hacia atrás como hacia adelante en el ciclo de vida. Las referencias hacia adelante de la ERS son especialmente importantes para el mantenimiento del software. Cuando el código y los documentos son modificados, es esencial poder comparar el conjunto total de requisitos que puedan verse afectados por estas modificaciones.

- **Utilizable.**

- En la ERS también se deben tener en cuenta las necesidades de mantenimiento. El personal que no ha intervenido directamente en el desarrollo debe ser capaz de encargarse de su mantenimiento. Así, dicha ERS actúa a modo de plano de la aplicación, permitiendo incluso modificaciones que no requieran un cambio en el diseño. En ocasiones, el equipo de desarrollo supone unos conocimientos que el personal que se encargue del mantenimiento no tiene por qué tener. Por esta razón es necesaria una correcta documentación de las funciones, ya que si no se conoce en detalle su origen, difícilmente podrán ser modificadas.

4.2.6.4 Esquema de la ERS definida en el IEEE 830-1998

La figura 28 muestra la estructura de la ERS propuesta por el IEEE en su estándar 830 [IEEE, 1998]:

1. Introducción
1.1. Propósito
1.2. Ámbito del Sistema
1.3. Definiciones, Acrónimos y Abreviaturas
1.4. Referencias
1.5. Visión general del documento
2. Descripción General
2.1. Perspectiva del producto
2.2. Funciones del producto
2.3. Características de los usuarios
2.4. Restricciones
2.5. Suposiciones y dependencias
2.6. Requisitos futuros
3. Requisitos Específicos
3.1. <u>Interfases externas</u>
3.2. Funciones
3.3. Requisitos de rendimiento
3.4. Restricciones de diseño
3.5. Atributos del sistema
3.6. Otros requisitos
4. Apéndices
5. Índice

Figura 28. Estructura de ERS IEEE 830



Profundizar en la IEEE-830

Es altamente recomendable profundizar en cada uno de los apartados que se definen en el estándar estudiado. Para lo cual una lectura detallada de las especificaciones de los requisitos del software que se encuentran descritas por IEEE 830-1998 es menester.

4.2.7 Conclusión del Análisis

La Etapa de Análisis es una etapa muy importante en el proyecto. Es altamente recomendable trabajar con Analistas Funcionales experimentados que permitan definir claramente “QUE” es lo que el sistema debería hacer. Todo esto debe estar plasmado en una Especificación de Requisitos de Software. Es muy importante utilizar un documento estándar para registrar los requisitos. La IEEE 830 es una muy buena opción.

El análisis y la especificación de los requisitos puede parecer una tarea relativamente sencilla, pero, en realidad, el contenido del análisis es muy denso y abundan las malas interpretaciones o la falta de información. Es muy difícil evitar la ambigüedad.

Finalmente el equipo desarrollador deberá entregar la Especificación de Requisitos como el entregable principal de esta etapa. Esta será la entrada principal para que los diseñadores puedan comenzar a trabajar en la Etapa de Diseño.

El documento de especificación de requisitos software supone una especie de contrato entre cliente y desarrolladores en el que unos indican sus necesidades, mientras que los otros se limitan a implementar lo que se indica en el documento. Principalmente por esta razón tiene tanta importancia la fase de análisis de requisitos. Por tanto, **de la calidad del documento de ERS dependerá el desarrollo y calidad del producto final.** La existencia de un estándar, como la ERS (IEEE 830) permite la coherencia en la especificación de requisitos y ayuda a no dejar cabos sueltos.

4.2.8 Caso de Estudio: Análisis

4.2.8.1 Requisito nº33: ABMC de Protocolos

Título	El Sistema deberá administrar Altas, Bajas, Modificaciones y Consultas de Protocolos.
Descripción	Los protocolos son los documentos que describen la realización de un conjunto de prácticas a un determinado paciente, como así también las muestras biológicas necesarias que han sido recepcionadas para la realización de dichas prácticas. Cada vez que el paciente solicita la realización de estudios se le crea el protocolo nuevo.
Prioridad	Alta/Esencial

4.2.8.2 Caso de Uso: Registrar Nuevo Protocolo a Cliente

Actores	Usuario recepcionistas
Propósito	Registrar un nuevo protocolo para un cliente en particular, el cual tendrá un conjunto de prácticas a realizar con las muestras biológicas suministradas por el cliente
Visión General	Cuando un cliente llega al laboratorio, el usuario recepcionista lo atiende. El cliente le indica las prácticas que debe realizarse, el usuario recepcionista crea un nuevo protocolo con las prácticas indicadas y recepciona las muestras biológicas necesarias para las prácticas
Tipo	Primario /Esencial
Precondiciones	Que el usuario recepcionista haya iniciado sesión en el sistema Que el usuario recepcionista tenga permisos para registrar nuevos protocolos Que existan prácticas en el sistema Que las prácticas indiquen qué tipos de muestras requieren Que el cliente se encuentre registrado en el sistema
Postcondiciones	Se ha creado un protocolo con sus prácticas
Referencia	Requisito Nº 33

Curso Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1 – El usuario recepcionista busca y selecciona en el sistema al cliente	2 – El sistema valida que el usuario tenga los permisos correspondientes, busca al cliente y comprueba que no haya sido eliminado del sistema
3 – El usuario recepcionista selecciona e ingresa el código de una práctica	
3.a Si está autorizada por una obra social se carga el número de orden y se selecciona la obra social por cada práctica	4 – El sistema valida los datos, selecciona la práctica y la agrega a la lista de prácticas del protocolo
Se repiten los pasos 3 y 4 mientras hayan más prácticas	
5 – El usuario recepcionista las registra muestras requeridas, por el conjunto de prácticas, en el sistema.	6 – El sistema valida los datos ingresados, comprueba que las muestras se corresponden con las prácticas derivadas y agrega las muestras al protocolo
7 -El usuario recepcionista ingresa el médico ordenante y si existe una observación	8 – El sistema valida los datos ingresados El Sistema informa el total a pagar
9 –El usuario recepcionista indica el monto del pago, finaliza y guarda los cambios del protocolo.	10 – El sistema registra el nuevo protocolo, sus prácticas, sus muestras y su pago. 11 – El sistema informa que la operación se ha llevado a cabo satisfactoriamente

Flujo Alterno

1 – El cliente no se encuentra registrado en el sistema. Utilizar Caso de Uso: Registrar Nuevo Cliente
 2- El sistema comprueba que el usuario no tiene los permisos para ejecutar esta función. El sistema informa del error y cancela la operación
 2 – El sistema comprueba que el cliente ha sido previamente eliminado. El sistema informa del error y solicita que realice nuevamente la selección.
 4 – El sistema detecta que la práctica solicitada no existe. Informa del error y descarta el ingreso.
 4 – El sistema detecta que la práctica ya ha sido cargada previamente. Informa de la situación y descarta la selección.
 4 – El cliente no tiene obra social e informa que autorización de una obra social para uno o más prácticas. El sistema detiene la registración y abre el menú de Asociar Cliente a Obra Social.
 5 – El cliente no posee las muestras biológicas necesarias. El usuario recepcionista omite este paso, continúa con los pasos del proceso y al final deriva al cliente al área de extracción, modifica el estado del protocolo y de las prácticas asociadas a las muestras pendientes. Cuando el cliente entregue las muestras se utiliza el caso de uso: registrar muestras de protocolo.
 6 – El sistema detecta que se ha registrado una muestra innecesaria para las prácticas a realizar. Informa que la muestra no es necesaria para realizar las prácticas solicitadas y descarta la registración
 6 – El sistema detecta que las muestras ingresadas no son suficientes para realizar todas las prácticas solicitadas. Informa del error y pregunta el usuario recepcionista si desea cambiar el estado de las prácticas y del protocolo o si quiere eliminar las prácticas del protocolo.
 8 – El sistema detecta que los datos ingresados no son válidos. Informa del error y solicita reingreso
 1,3,5,7,9 – El cliente cancela la operación. El sistema descarta la información sin realizar ningún cambio en el sistema

Entradas:

- DNI o apellido y nombre del paciente.
- Códigos de Prácticas.
- Código de Autorización de Obra Social y Código de Obra Social.
- Código de muestras.
- Observación.

Procesos:

- Validar los datos ingresados.
- Calcular el precio total del Protocolo.
- Generar el número identificador del Protocolo.
- Imprimir código de barras del número de protocolo para rotular a los tubos con las muestras biológicas.
- Generar el número identificador de cada Práctica de un Protocolo.
- Asociar cada Práctica de un Protocolo con dicho Protocolo.
- Generar el número identificador de cada Muestra de un Protocolo.
- Asociar cada Muestra de un Protocolo con dicho Protocolo.

- Calcular Fecha y Hora.
- Asociar el Protocolo al Cliente.
- Verificar que las muestras requeridas para realizar las prácticas han sido recepcionadas.
- Almacenar los datos.
- Asignar Estado Inicial del Flujo de Estados a Protocolo, Prácticas de Protocolo y Muestras de Protocolo.

Salidas:

- Número de Protocolo.
- Fecha de Recepción.
- Hora de Recepción.
- Número de cada Práctica del Protocolo.
- Número de cada Muestra del Protocolo.
- Fecha Estimada de Finalización.

4.3 Diseño

4.3.1 Descripción

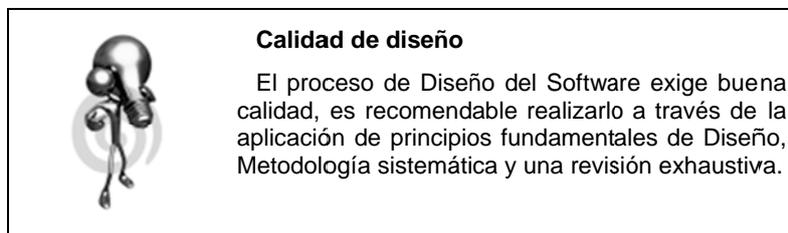
Es el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación, a fin de proporcionar una completa descripción de **cómo** se realizará el software. En el análisis nos centrábamos en **qué** se tiene que hacer, mientras que en el diseño nos centramos en **cómo** se tiene que realizar. Por lo tanto, en esta etapa se verá **cómo** almacenar los datos, **cómo** se van a implementar los procesos y **cómo** se van a diseñar las interfaces; todas estas cuestiones definidas en el análisis. **El Diseño del Software es un proceso y un modelado a la vez.** Es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir. A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas:

- El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente.
- Debe ser una guía que puedan leer y entender los que construyan, prueben y mantengan el Software.

- El Diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación.

Para evaluar la calidad en el diseño, se deben establecer criterios técnicos como son:

- Debe presentar una organización jerárquica que haga un uso inteligente del control entre los componentes del software.
- Debe ser modular, es decir, se debe hacer una partición lógica del Software en elementos que realicen funciones y subfunciones específicas.
- Debe contener abstracciones de datos y procedimientos.
- Debe producir módulos que presenten características de funcionamiento independiente.
- Debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y el entorno exterior.
- Debe producir un diseño usando un método que pudiera repetirse según la información obtenida durante el análisis de requisitos de Software.



La fase de diseño comienza con la aprobación formal de los requisitos del sistema por parte del usuario o cliente. Tiene por objeto definir la estructura del software a desarrollar a partir del modelo establecido en la fase de análisis. Lo que se obtiene asignando funciones a componentes de software, y definiendo los flujos de datos y de control entre dichas componentes. Como resultado de esta fase se obtiene el documento de diseño.

Encierra las siguientes etapas:

- ✓ **El diseño de los datos.** Transforma el modelo de dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el Software.
- ✓ **El Diseño Arquitectónico.** Define la relación entre cada uno de los elementos estructurales del programa.

- ✓ **El Diseño de la Interfaz.** Describe como se comunica el Software consigo mismo, con los sistemas que operan con él y con los operadores y usuarios que lo emplean.

A continuación se detallan algunos aspectos de interés aplicables a la sub-fase de diseño de alto nivel:

1. Durante el diseño de la arquitectura se adoptará y comenzará a utilizar una metodología de diseño de software concreta y reconocida.
2. La arquitectura del software se descompondrá en unidades de menor complejidad. Preferentemente se implementará una metodología tipo *top-down*, de diseño descendente.
3. Para cada elemento o componente del software se indicarán y describirán, al menos, los datos de entrada, las funciones que desempeña y los datos de salida.
4. Se estimarán durante esta sub-fase los requisitos computacionales (CPU, memoria, disco, etc.) para el entorno de desarrollo y el entorno operativo.
5. El resultado de esta sub-fase será revisado formalmente durante una reunión de revisión.
6. El resultado de esta sub-fase se plasmará en el documento de diseño de alto nivel, que contendrá, al menos, los principales componentes del software y las interfaces entre ellos.
7. Este documento incluirá una tabla donde se muestre la correspondencia entre requisitos de software y elementos del diseño de alto nivel.



Metodología de Diseño de Sistemas

Es altamente recomendable utilizar una metodología concreta y reconocida para el diseño del sistema de información, así como para las demás fases del desarrollo del software. MÉTRICA versión 3 puede ser utilizada libremente con la única restricción de citar la fuente de su propiedad intelectual.

4.3.2 Fallos en el Diseño de Sistemas

El diseño real de un sistema falla al no captar los requerimientos esenciales del cliente. La información puede no ser proporcionada lo suficientemente rápida para ser útil, también puede venir en un formato imposible de digerir y usar, o puede representar los elementos equivocados de datos. Un sistema puede ser diseñado con una interface pobre. Un sistema de información será juzgado como un fracaso si su diseño no es compatible con la estructura, cultura y metas de la institución cliente.

4.3.3 Entregable – Fase de Diseño

4.3.3.1 *Introducción*

El Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

4.3.3.2 *Relaciones de los artefactos del Diseño con los del Análisis*

El término Artefacto, en conexión con el desarrollo de software, está mayormente asociado a métodos o procesos de desarrollo específicos, como el Proceso Unificado. Un artefacto es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software. Otros se enfocan en el proceso de desarrollo en sí mismo, como planes de proyecto, casos de negocios o enfoque de riesgos. El código fuente compilado para el testeado se suele considerar un artefacto, ya que el ejecutable es necesario para el plan de testeado. En ocasiones un artefacto puede referirse a un producto terminado, como el código o el ejecutable, pero más habitualmente se refiere a la documentación generada a lo largo del desarrollo del producto en lugar del producto en sí. Los artefactos pueden variar en su necesidad de mantenimiento y actualización.

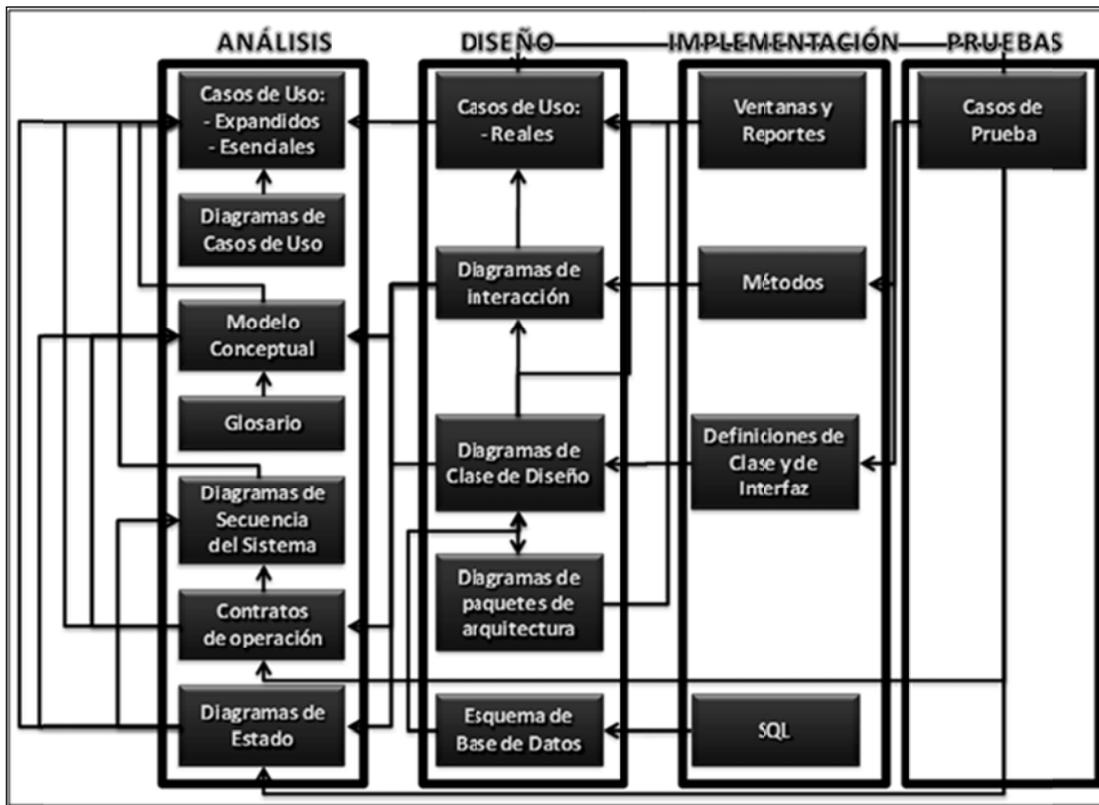


Figura 29. Fases de ciclo de desarrollo de software

Como podemos observar en la figura 29, UML y Patrones presenta las diferentes fases del ciclo de vida de desarrollo iterativo de software a partir de artefactos interrelacionados. Particularmente en las fases de Análisis y Diseño existen cuestiones vinculadas, si bien los Casos Esenciales de Uso son desarrollados en la fase de Análisis, este enfoque propone los Casos Reales de Uso en la fase de Diseño, esto muestra la íntima interrelación entre fases.

Presenta numerosos artefactos, pero si nos imaginamos UML como una caja de herramientas con su martillo, destornillador, alicates, etc., siguiendo con la analogía, si vamos a colgar un cuadro no usaremos todas las herramientas de nuestra caja, posiblemente sólo usemos el martillo para clavar el clavo. Lo mismo pasa con UML, una vez que conozcamos las herramientas usaremos en cada momento las más adecuadas a nuestras necesidades. De esta manera podemos indicar que los artefactos más utilizados en la fase de Análisis, sin que esto signifique la excepción de los demás, son los Casos de Uso y en la de Diseño El Diagrama de Clases, que para los fines de este libro se considerará como el entregable necesario para continuar con la fase de Implementación.

4.3.3.3 De la ERS al Diagrama de Clases

Hasta el momento hemos definido un entregable en la fase de Análisis denominado Especificación de Requisitos de Software (ERS) a través del estándar IEEE 830, que como vimos en su estructura, los diferentes usuarios se definen en el apartado 2.3 *Características de*

Usuarios, y seguidamente identifica requisitos en la forma “**El sistema deberá...**” en el apartado 3 *Requisitos Específicos*.

Si consideramos por otro lado el UML podemos notar que realiza a través de Casos de Uso una especificación de requisitos funcionales de software muy similar al del estándar IEEE 830, o que en esencia cumple con identificar, validar y documentar los requisitos de software. Señala el uso de dos elementos: **el actor** (usuario) y el **caso de uso** (requisito). El actor representa una entidad externa que interactúa con el sistema. Las entidades externas podrán ser personas u otros sistemas, mientras que el caso de uso hace referencia al/los requisito/s del sistema a construir, detallando su comportamiento, el cual se traduce en resultados que pueden ser observados por el actor. Los casos de uso describen las cosas que los actores quieren que el sistema haga “**El sistema deberá...**”, por lo que un caso de uso debería ser una tarea completa desde la perspectiva del actor.

Los resultados de la especificación de requisitos son dos productos: el catalogo de requisitos y el documento de especificación de requisitos de software. El primero de ellos contiene la lista de requisitos de software clasificada por tipo y prioridad; y el segundo de ellos, especifica el comportamiento del sistema a un grado de detalle mayor.

La figura 30 muestra la relación existente entre la Especificación de Requisitos de Software (ERS) según IEEE 830 y UML.

ERS IEEE 830		UML	
Número de requisito	1.	Nombre	Consultar Bibliografía
		Autor	Joaquin García
Requisito	El Sistema deberá administrar Consultas de Bibliografía.	Fecha	xx/xx/xxxx
		Descripción: Permite consultar Bibliografía en la Biblioteca	
Descripción	Cada Consulta deberá tener relacionada uno o más Libros. Un mismo Libro puede ser utilizado para varias Consultas.	Actores: Usuario logueado	
		Precondiciones: El usuario debe haberse logueado en el sistema	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esen- cial <input type="checkbox"/> Media/Des- eado <input type="checkbox"/> Baja/ Opcional	Flujo normal: <ul style="list-style-type: none"> • El actor pulsa el botón consultar bibliografía • El sistema muestra una caja de texto para introducir el titulo de la obra y otra zona de mayor tamaño para introducir una descripción breve. • El actor introduce el titulo de la obra y una descripción breve. • El sistema comprueba la validez de los datos y muestra la obra. 	
		Flujo alternativo: <ul style="list-style-type: none"> • El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija. 	
		Poscondiciones: La consulta a sido almacenad en el sistema.	

Figura 30. Comparativo ERS IEEE 830 y UML

De esta manera y salvando las distancias de ambos enfoques podemos encontrar relaciones muy fuertes que le permiten al Director de Proyecto contar con información suficiente como para ingresar a la Fase de Diseño con elementos suficientes como para generar el Diagrama de Clases (entregable).

En este libro se mostró como entregable de la fase de Análisis a la ERS IEEE 830, porque su formato con una pequeña labor de síntesis es ideal para acompañar al contrato, además de ser un estándar internacional reconocido que da prestigio a nuestro equipo de trabajo a la hora de negociar con el Cliente.

4.3.3.4 *Diagrama de Clases*

Si bien siguiendo UML y Patrones podemos decir que una vez en la Fase de Diseño definiremos los Casos Reales de Uso, prepararemos los Diagramas de Colaboración (Diagramas de Comunicación en otras versiones), asignaremos responsabilidades GRASP y estableceremos la Visibilidad entre Objetos para recién crear el Diagrama de Clases, para los fines de este libro, es éste último el documento principal que servirá de entregable para la fase de Implementación, sin dejar de lado que pueda complementarse con los demás.

Clases

Representa un conjunto de entidades que tienen en común propiedades, operaciones, relaciones y semántica.

Una clase es un constructor que define la estructura y comportamiento de una colección de objetos denominados instancias de la clase.

En UML la clase está representada por un rectángulo con tres divisiones internas, son los elementos fundamentales del diagrama:

- **Nombre:** Identifica a la clase.
- **Atributo:** Representa una propiedad de una entidad. Cada atributo de un objeto tiene un valor que pertenece a un dominio de valores determinado.
 - **Sintaxis del atributo es:** *Visibilidad <nombre>: tipo = valor {propiedades}*.
Donde visibilidad es +público, #protegido o –privado.
- **Métodos:** El conjunto de operaciones que describen el comportamiento de los objetos de una clase.
 - **Sintaxis de una operación es:** *Visibilidad nombre (lista de parámetros): tipo que retorna {propiedades}*.

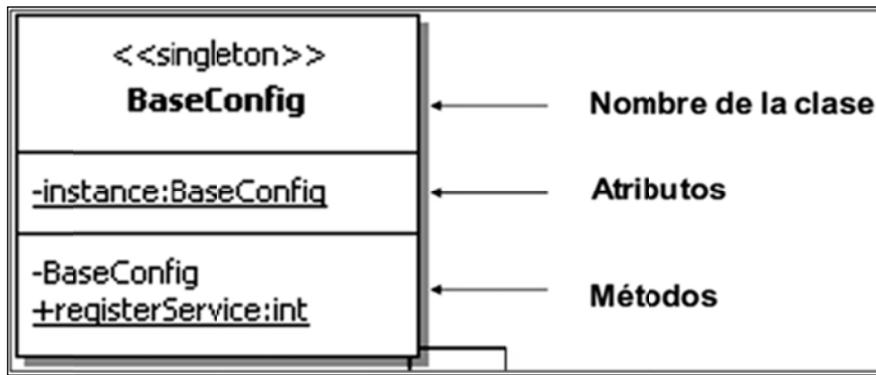


Figura 31. Ejemplo de Clase

Características:

- El lenguaje utilizado para especificar las clases es el mismo que el lenguaje de programación (UML).
- Visibilidad de atributos y operaciones de la clase.
- Las clases se mapean directamente en el lenguaje de programación.
- Las relaciones entre clases tienen un significado directo en el lenguaje de programación. Por ejemplo, la generalización se implementa como un mecanismo de Herencia.
- Los métodos de una clase de diseño son los métodos de una clase implementada.
- Las clases que provean interfaces pueden implementarse directamente en algunos lenguajes de programación. Por ejemplo, una clase de diseño en Java puede implementar una interfaz.
- Una clase de diseño puede ser activa, en el sentido de que los objetos instanciados pueden mantener su propio hilo de ejecución (thread) concurrentemente con otros objetos.

4.3.3.5 Crear Diagramas de Clases del Diseño

Para elaborar un diagrama de clases del diseño, aplique la siguiente estrategia con su equipo:

1. Identifique todas las clases que participan en la solución de software. Para ello analice los diagramas de interacción.
2. Dibújelas en un diagrama de clases.
3. Duplique los atributos provenientes de los conceptos asociados del modelo conceptual.

4. Agregue los nombres de los métodos analizando los diagramas de interacción.
5. Incorpore la información sobre los tipos a los atributos y a los métodos.
6. Agregue las asociaciones necesarias para dar soporte a la visibilidad requerida de los atributos.
7. Agregue flechas de navegabilidad a las asociaciones para indicar la dirección de la visibilidad de los atributos.
8. Agregue las líneas de relaciones de dependencia para indicar la visibilidad no relacionada con los atributos.

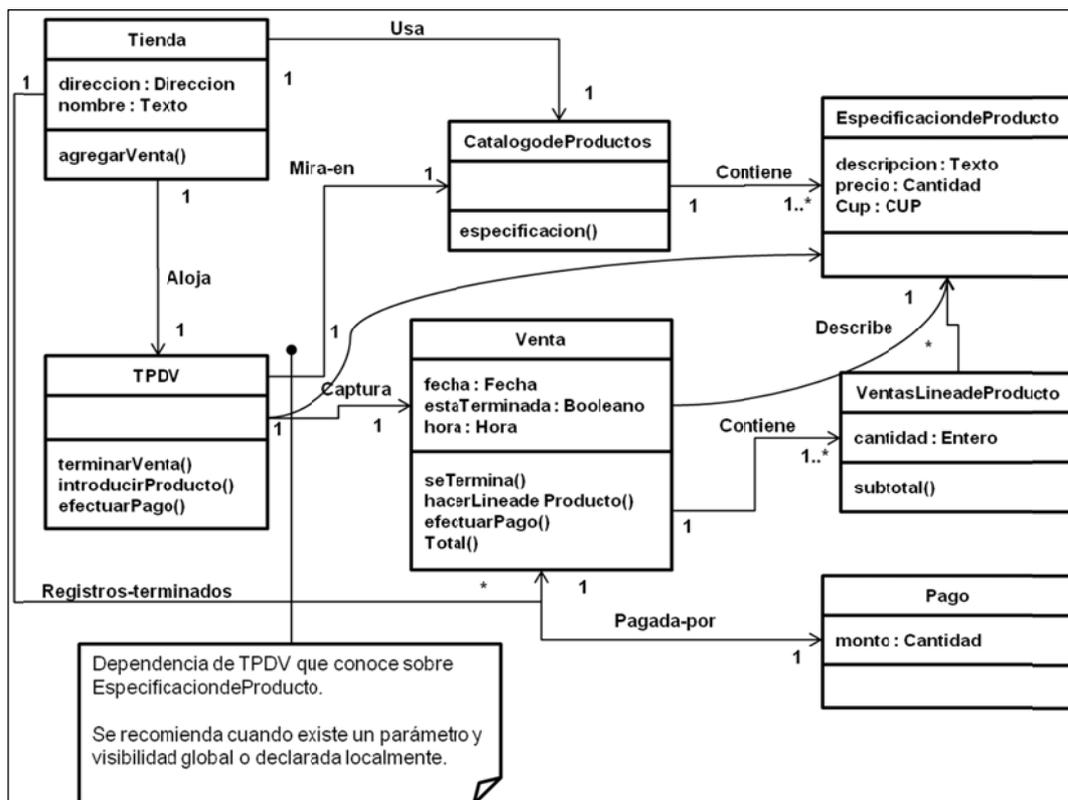


Figura 32. Ejemplo de Diagrama de Clases del Diseño

4.3.4 Resumen

Éste documento generado será el entregable para la fase de implementación, se presentó acá por considerarse el más relevante, pero generalmente se encuentra acompañado del Plan de Pruebas de Programas entre otros que permitirán al equipo de la siguiente fase comenzar con su trabajo. Asimismo cabe aclarar que al tratarse de un proceso de desarrollo iterativo e incremental presenta como ventaja la posibilidad de introducir los resultados de un ciclo anterior al iniciar el siguiente.

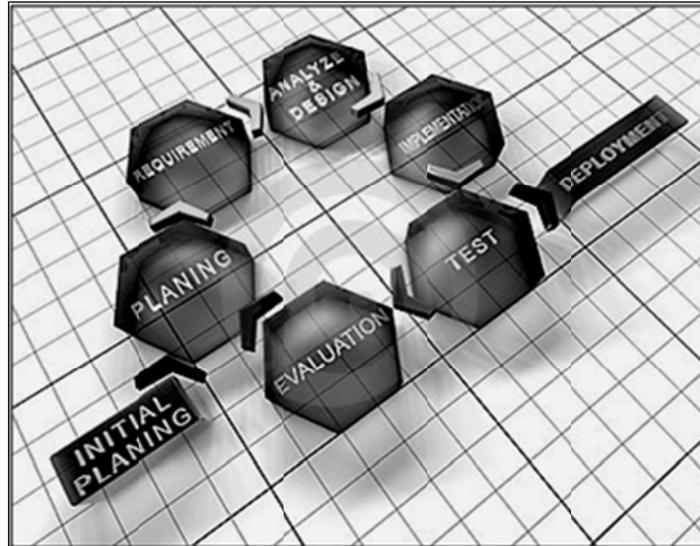


Figura 33. Modelo iterativo e incremental de desarrollo de sistemas

4.3.5 Conclusión de la Fase de Diseño

Antes de comenzar con el desarrollo de cualquier proyecto, se debe conducir un estudio de Sistemas para detectar todos los detalles de la situación actual de la empresa. La información reunida con este estudio sirve como base para crear varias estrategias de Diseño. Los administradores deciden entonces qué estrategias seguir. Los Gerentes, empleados y otros usuarios finales que se familiarizan con el uso del sistema tienen un papel muy importante en el desarrollo de sistemas.

4.3.6 Caso de Estudio: Diseño

4.3.6.1 Caso de Uso Real: Registrar Nuevo Protocolo

Actores	Usuario recepcionistas
Propósito	Registrar un nuevo protocolo para un cliente en particular, el cual tendrá un conjunto de prácticas a realizar con las muestras biológicas suministradas por el cliente
Visión General	Quando un cliente llega al laboratorio, el usuario recepcionista lo atiende. El cliente le indica las prácticas que debe realizarse, el usuario recepcionista crea un nuevo protocolo con las prácticas indicadas y recepciona las muestras biológicas necesarias para las prácticas
Tipo	Primario /Real
Precondiciones	Que el usuario recepcionista haya iniciado sesión en el sistema Que el usuario recepcionista tenga permisos para registrar nuevos protocolos Que existan prácticas en el sistema Que las prácticas indiquen qué tipos de muestras requieren Que el cliente se encuentre registrado en el sistema
Postcondiciones	Se ha creado un protocolo con sus prácticas
Referencia	Requisito N° 33
Curso Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1 – El usuario ingresa al menú de registrar protocolo	2 – El sistema busca en la base de datos y carga en los combobox correspondientes a los cliente particulares, laboratorios clientes, prácticas, muestras y obras sociales
3 – El usuario recepcionista selecciona de un combobox el cliente (sea particular o laboratorio cliente), si se trata de un particular se selecciona la obra social (si tuviere) de un combobox, se ingresa en un textbox el número de orden y el médico en un textbox. Se seleccionan de un combobox múltiple las prácticas a realizarse, se seleccionan de un combobox múltiple las muestras que se dejan para realizar las prácticas, se ingresa una observación en un textbox, y se pulsa el botón <i>Grabar Protocolo</i>	4 – El sistema valida que el usuario tenga los permisos correspondientes, recupera los id's de los combobox, comprueba que existan en el sistema y que no estén eliminados, valida el formato y valor de los textbox ingresados. El sistema crea el protocolo en el sistema y lo graba en la base de datos. El sistema muestra información del protocolo.
5 – El usuario ingresa en un textbox el pago que se realizará sobre el protocolo y pulsa el botón <i>Registrar Pago</i>	6 – El sistema valida el monto ingresado, comprueba que no sea mayor al total y modifica el atributo deuda del protocolo 5 – El sistema informa que la operación se ha llevado a cabo satisfactoriamente y muestra

Flujo Alterno

- 3 – El cliente no se encuentra registrado en el sistema. Utilizar Caso de Uso: Registrar Nuevo Cliente
- 2- El sistema comprueba que el usuario no tiene los permisos para ejecutar esta función. El sistema informa del error y cancela la operación
- 4 – El cliente no tiene obra social e informa que autorización de una obra social para una o más prácticas. El sistema detiene la registración y abre el menú de Asociar Cliente a Obra Social.
- 4 – El cliente no posee las muestras biológicas necesarias. El usuario recepcionista omite este paso, continúa con los pasos del proceso y al final deriva al cliente al área de extracción, modifica el estado del protocolo y de las prácticas asociadas a las muestras pendientes. Cuando el cliente entregue las muestras se utiliza el caso de uso: registrar muestras de protocolo.
- 4 – El sistema detecta que se ha registrado una muestra innecesaria para las prácticas a realizar. Informa que la muestra no es necesaria para realizar las prácticas solicitadas y descarta la registración
- 4 – El sistema detecta que las muestras ingresadas no son suficientes para realizar todas las prácticas solicitadas. Informa del error y pregunta el usuario recepcionista si desea cambiar el estado de las prácticas y del protocolo o si quiere eliminar las prácticas del protocolo.
- 4 – El sistema detecta que los datos ingresados no son válidos. Informa del error y solicita reingreso
- 6 – El sistema detecta que los datos ingresados no son válidos. Informa del error y solicita reingreso
- 1,3,5 – El cliente cancela la operación. El sistema descarta la información sin realizar ningún cambio en el sistema

4.3.6.2 Diagrama de Secuencia

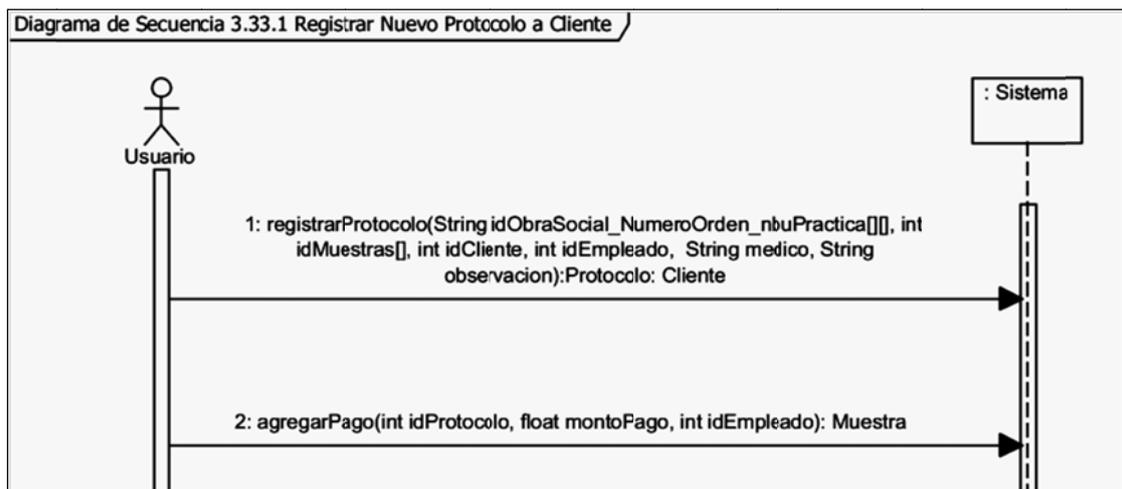


Figura 34. Diagrama de Secuencia para el caso de estudio que se viene desarrollando

4.3.6.3 Diagrama de Comunicación

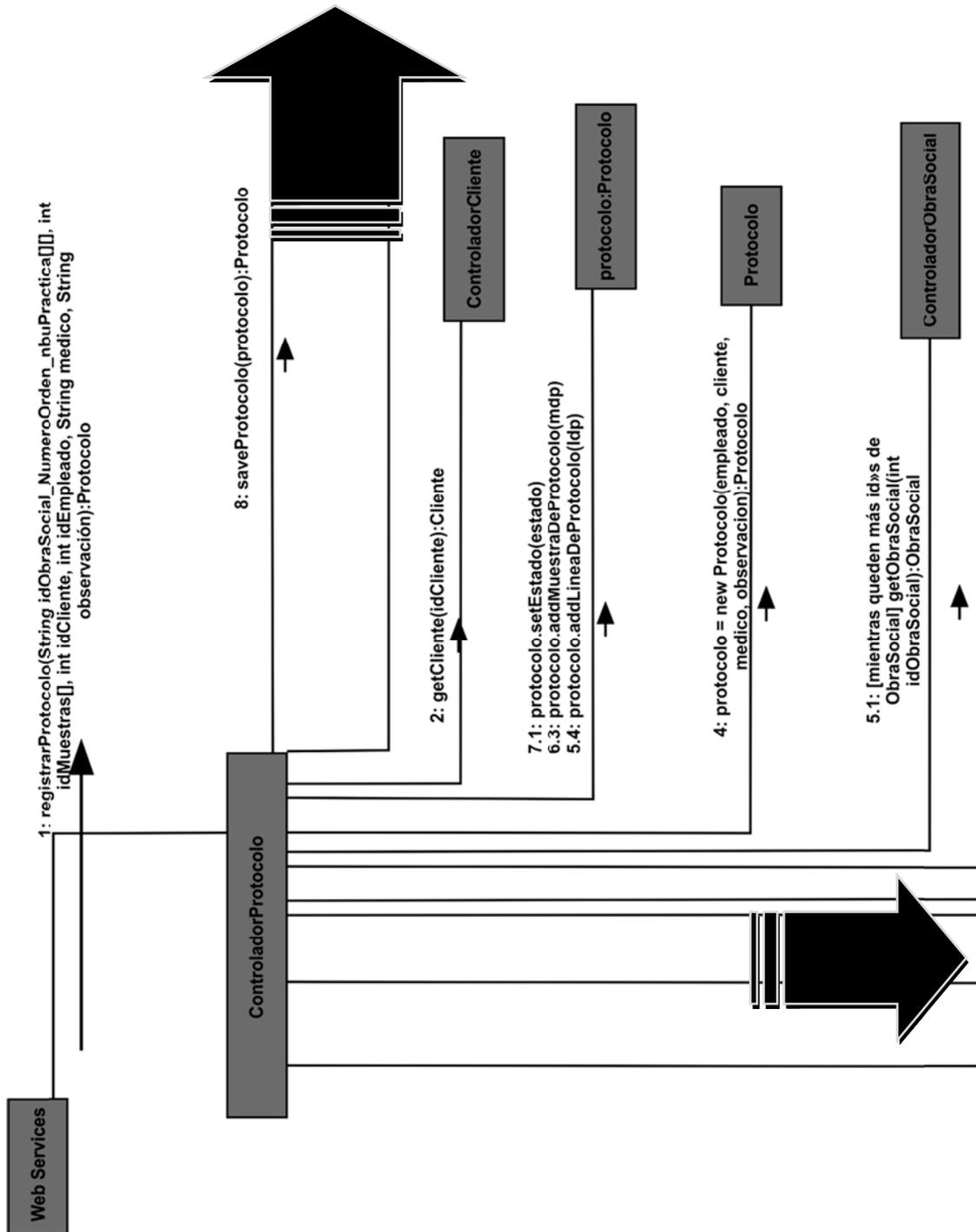


Figura 35. Diagrama de comunicación para el caso de estudio (I)

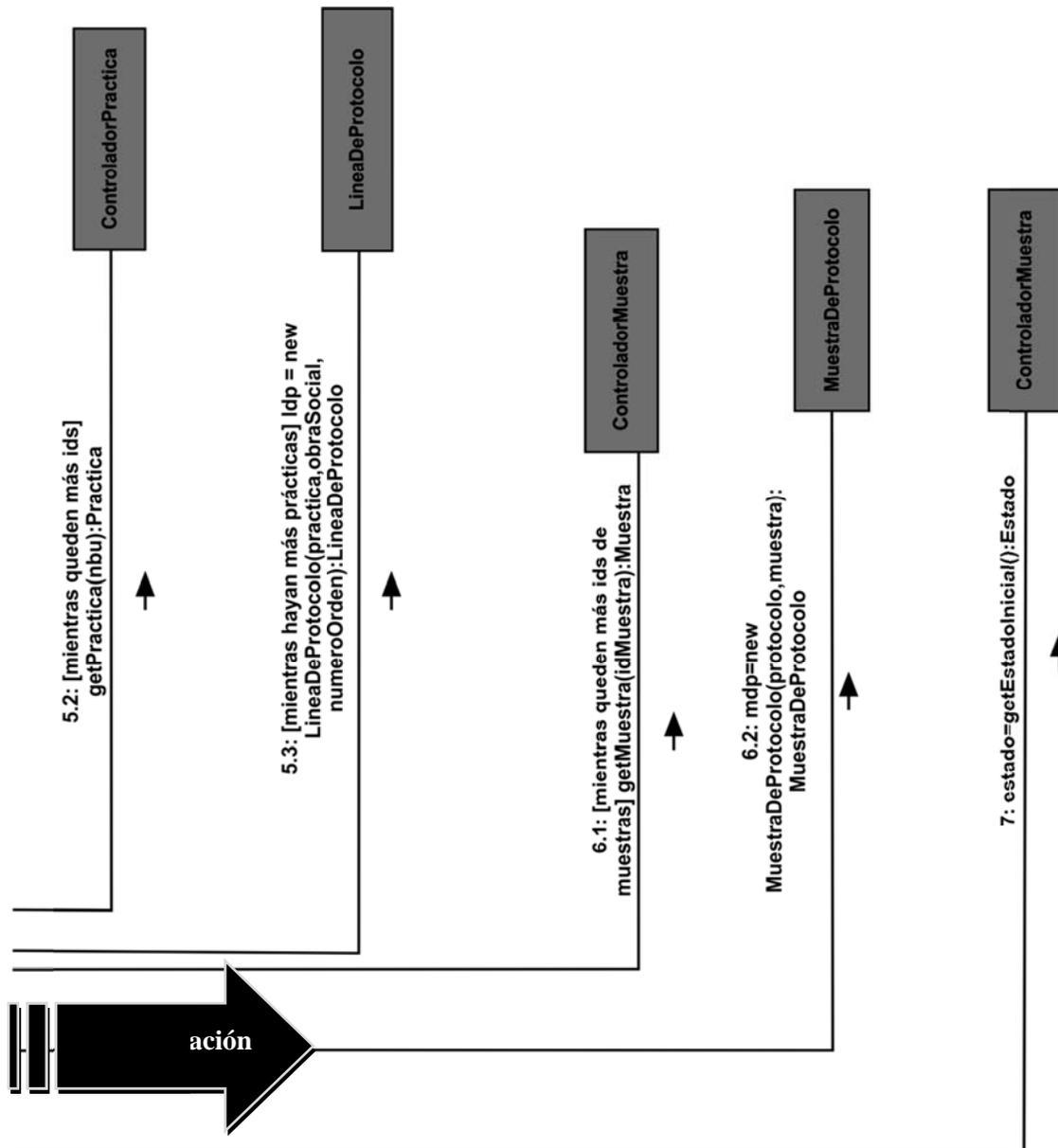


Figura 36. Diagrama de comunicación para el caso de estudio (II)

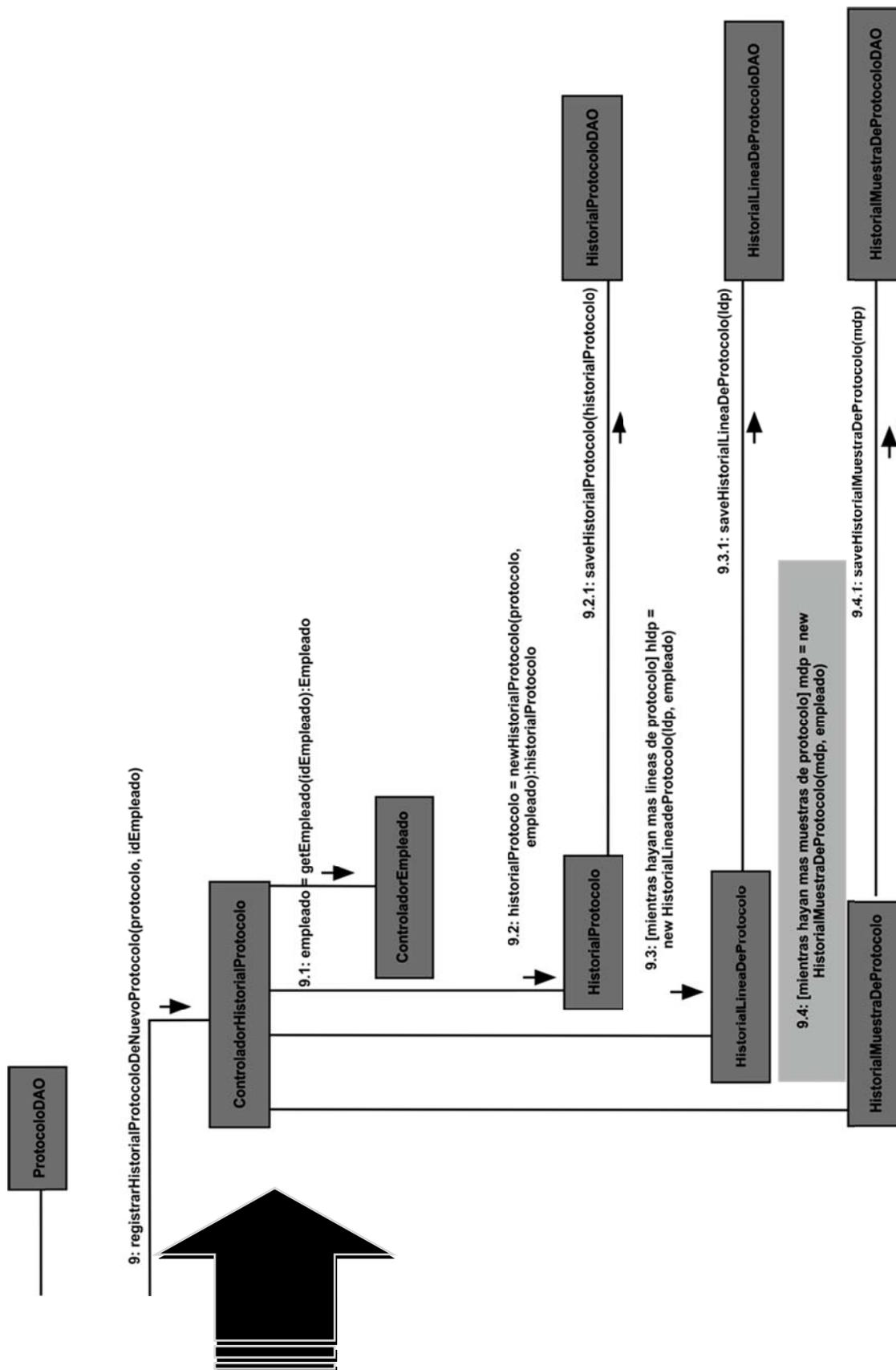


Figura 37. Diagrama de comunicación para el caso de estudio (III)

4.3.6.4 Diagrama de Clases

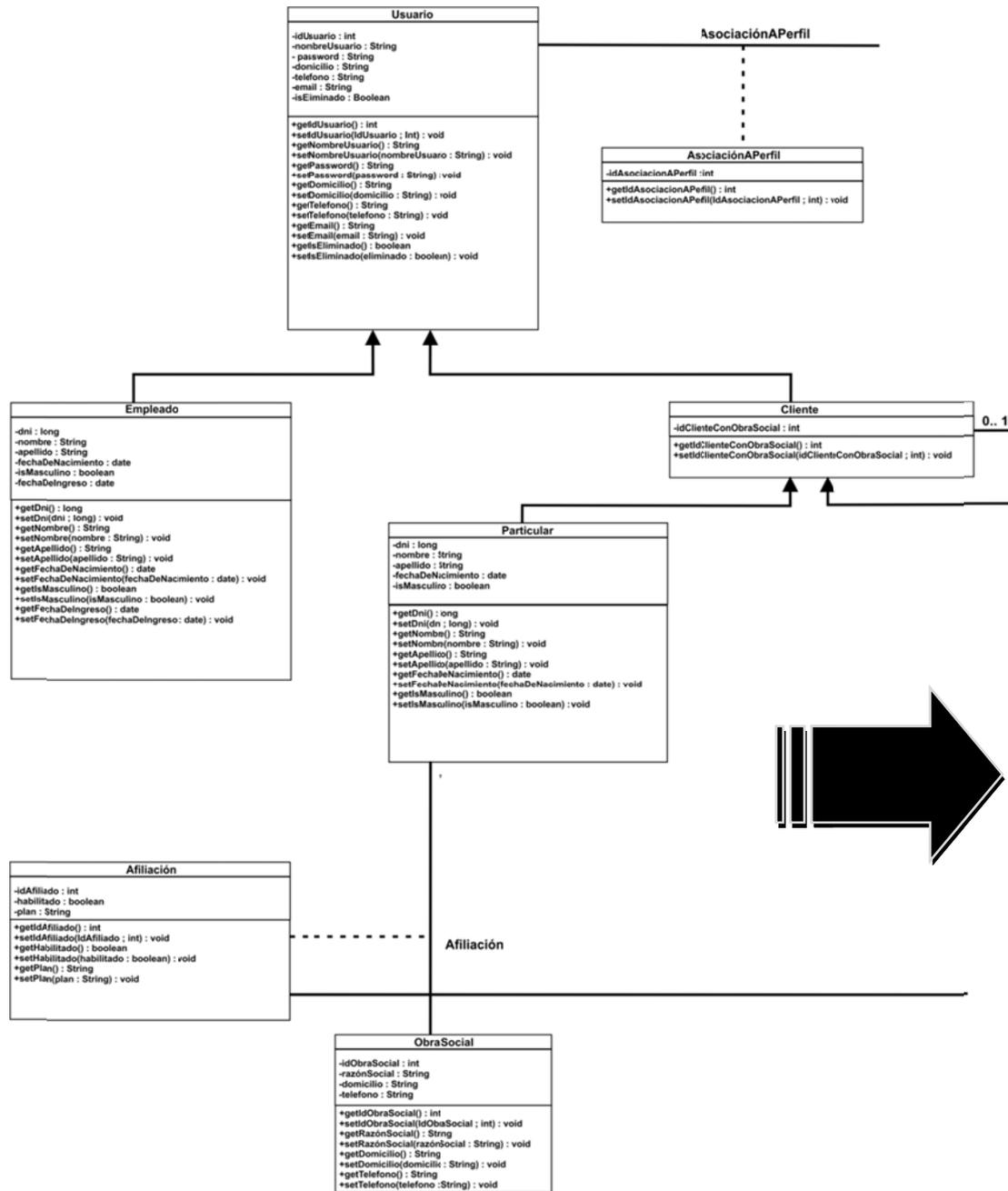


Figura 38. Diagrama de Clases para el caso de estudio (I)

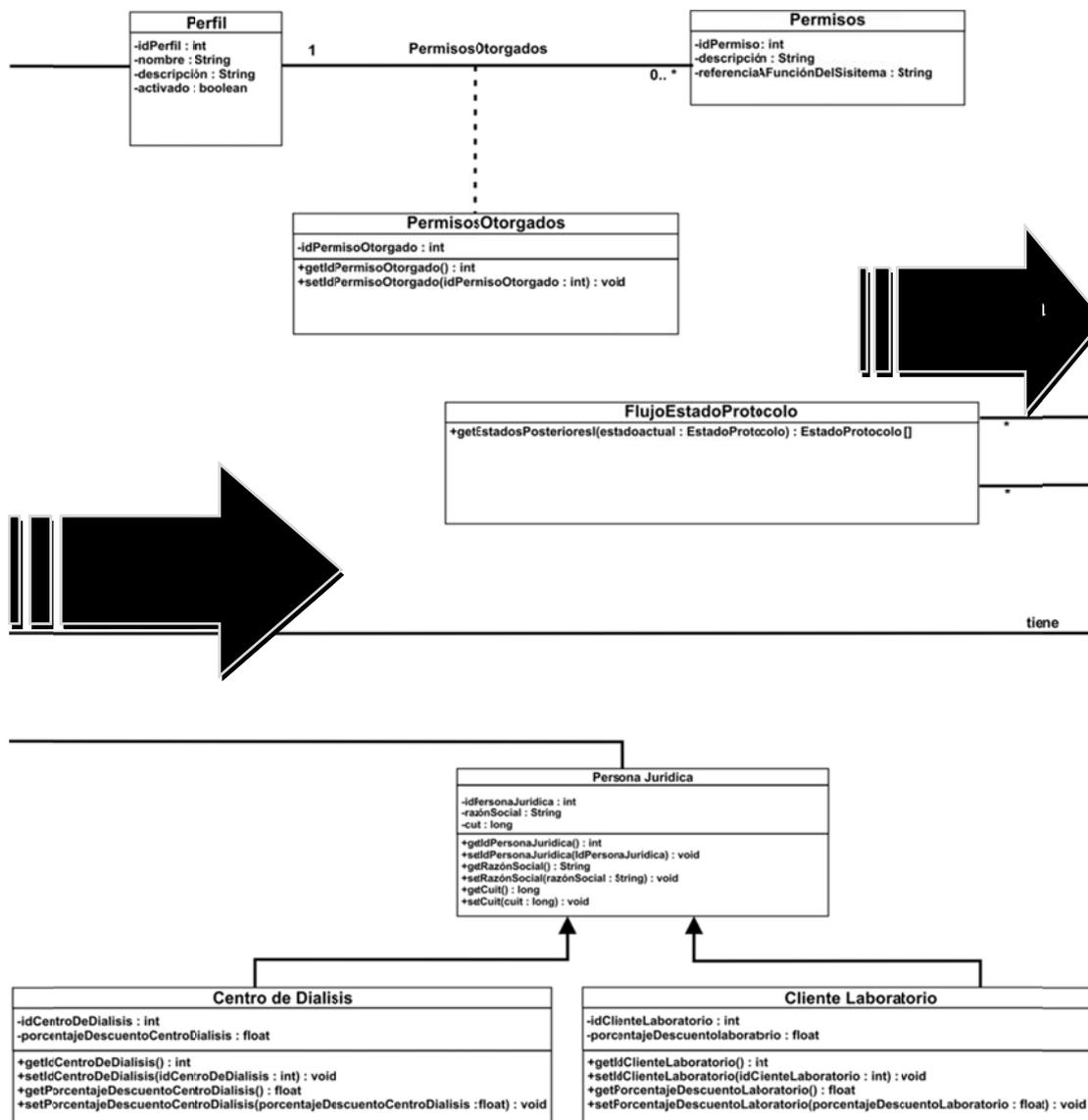


Figura 39. Diagrama de Clases para el caso de estudio (II)

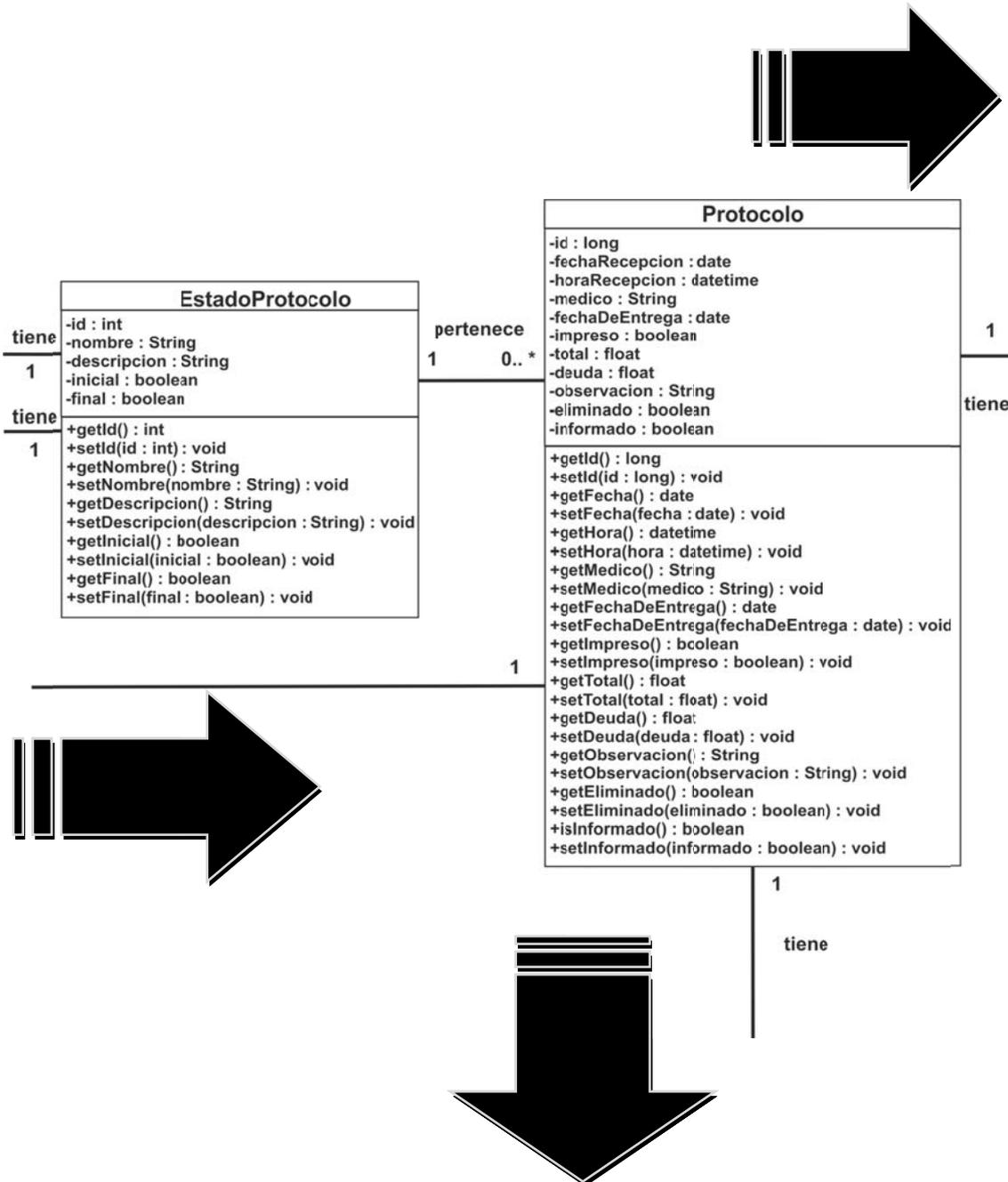


Figura 40. Diagrama de Clases para el caso de estudio (III)

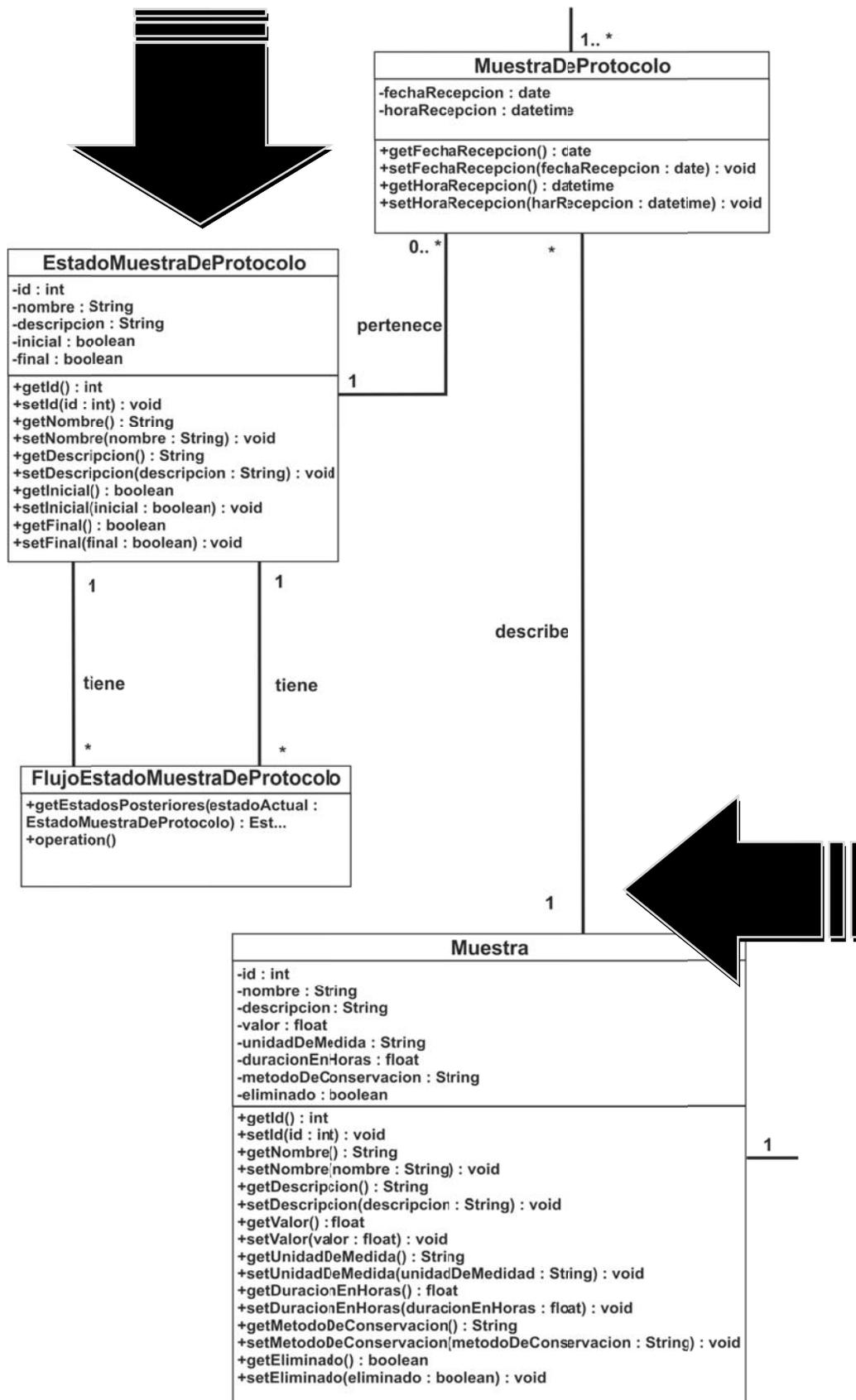


Figura 41. Diagrama de Clases para el caso de estudio (IV)

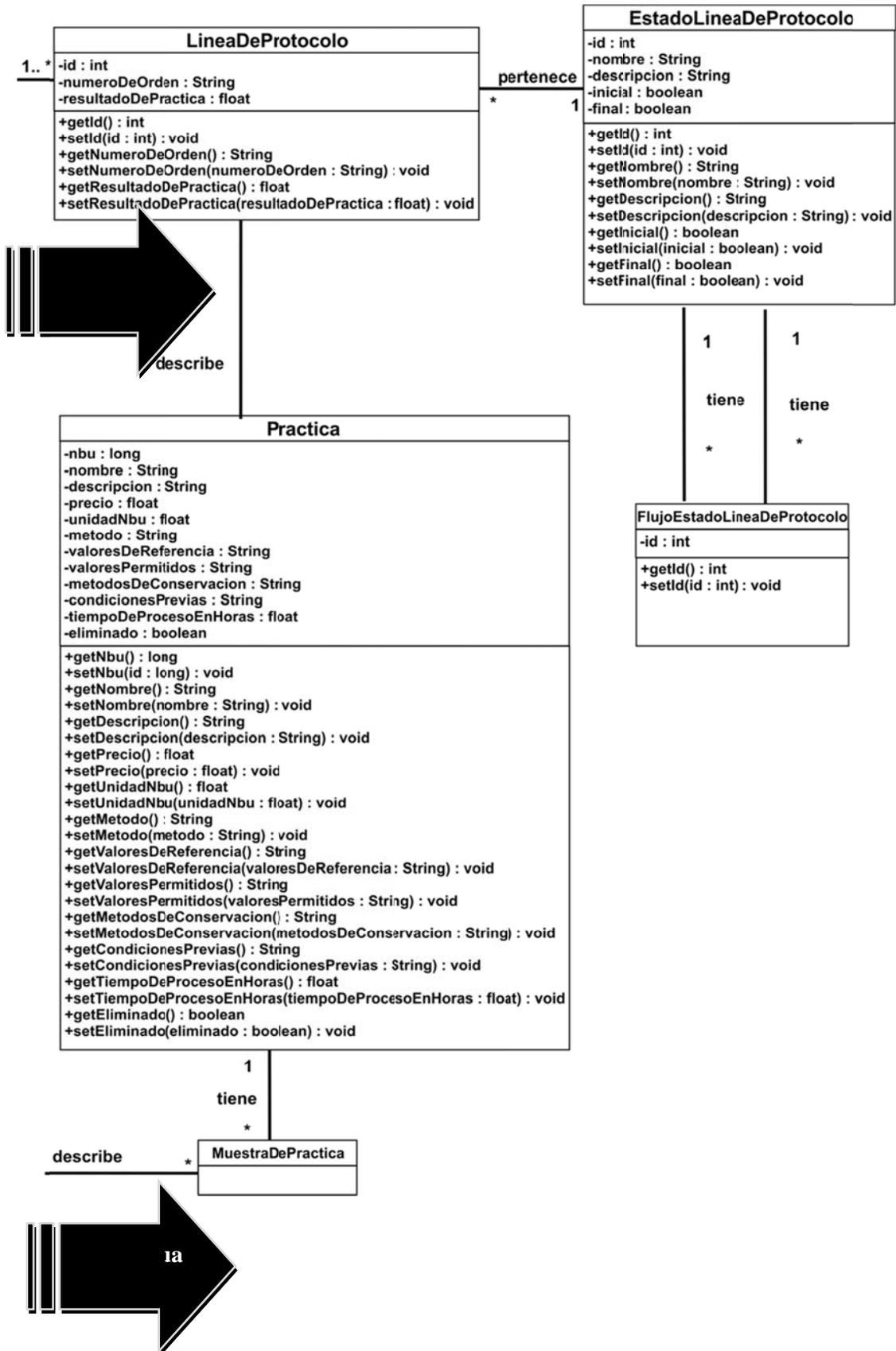


Figura 42. Diagrama de Clases para el caso de estudio (V)

4.4 Implementación

4.4.1 Descripción

En este proceso se genera el código de los componentes del Sistema de Información, se desarrollan todos los procedimientos de operación y seguridad, y se elaboran los manuales de usuario final y de explotación con el objetivo de asegurar el correcto funcionamiento del sistema para su posterior implantación.

Para conseguir dicho objetivo, se realizan las pruebas unitarias, las pruebas de integración de los subsistemas y componentes y las pruebas del sistema, de acuerdo al plan de pruebas establecido. Esto se encuentra desarrollado en la sección 4.5 Pruebas.

Así mismo, se define la formación del usuario final y, si es necesario, se construyen los procedimientos de migración y carga inicial de datos.

En las tareas desarrolladas en esta fase se asegura la disponibilidad de la infraestructura necesaria para la generación del código de los componentes y procedimientos del sistema de información.

Una vez configurado el entorno de construcción, se realiza la codificación y las pruebas de los distintos componentes que conforman el sistema de información, en las actividades:

- **Generación del Código de los Componentes y Procedimientos**, que se hace según las especificaciones de construcción del sistema de información, y conforme al plan de integración del sistema de información.
- **Ejecución de las Pruebas Unitarias**, dónde se llevan a cabo las verificaciones definidas en el plan de pruebas para cada uno de los componentes.

Una vez construido el sistema de información y realizadas las verificaciones correspondientes, se lleva a cabo la integración final del sistema de información, comprobando tanto las interfaces entre subsistemas y sistemas externos como los requisitos, de acuerdo a las verificaciones establecidas en el plan de pruebas para el nivel de pruebas del sistema.

En esta fase también se genera la documentación del usuario final. Se trata de la formación necesaria para que los usuarios finales sean capaces de utilizar el sistema de forma satisfactoria; se especifica en la actividad Definición de la Formación de Usuarios Finales.

Si se ha establecido la necesidad de realizar una migración de datos, la construcción y pruebas de los componentes y procedimientos relativos a dicha migración y a la carga inicial de datos se realiza en la actividad *Construcción de los Componentes y Procedimientos de Migración y Carga Inicial de Datos*.

La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad, así como su posible reutilización.

4.4.2 Preparación del entorno de generación y construcción

El objetivo de esta actividad es asegurar la disponibilidad de todos los medios y facilidades para que se pueda llevar a cabo la construcción del sistema de información. Entre estos medios, cabe destacar la preparación de los puestos de trabajo, equipos físicos y lógicos, gestores de bases de datos, bibliotecas de programas, herramientas de generación de código, bases de datos o ficheros de prueba, entre otros.

Las características del entorno de construcción y sus requisitos de operación y seguridad, así como las especificaciones de construcción de la estructura física de datos, se establecen en la actividad *Generación de Especificaciones de Construcción*, y constituyen el punto de partida para la realización de esta actividad.

4.4.2.1 Implantación de la Base de Datos Física o Ficheros

La implementación de la base de datos física, es una tarea muy importante en esta fase. Estas son las actividades a desarrollar:

- Crear los elementos del sistema gestor de base de datos o sistema de ficheros.
- Reservar el espacio de almacenamiento, definiendo, entre otros, los dispositivos físicos a emplear, tamaño de los bloques, tipo de registro físico, zona de desbordamiento, opciones de almacenamiento de datos, etc.
- Inicializar la base de datos o ficheros, cargando los datos considerados necesarios en el espacio de almacenamiento previamente definido.

4.4.2.2 Preparación del Entorno de Construcción

En esta tarea se prepara el entorno en el que se construirán los componentes del sistema de información, contemplando aspectos tales como:

- Bibliotecas o librerías a utilizar.
- Herramientas: generadores de código, editores, compiladores, verificadores sintácticos, montadores de enlace.
- Puestos de trabajo.
- Implementación de los procedimientos de operación y seguridad propios del entorno de construcción, de acuerdo a los requisitos de seguridad y operación establecidos.

4.4.3 Generación del código de los componentes y procedimientos

El objetivo de esta actividad es la codificación de los componentes del sistema de información, a partir de las especificaciones de construcción obtenidas en la fase de Diseño,

así como la construcción de los procedimientos de operación y seguridad establecidos para el mismo.



Generación del código de los componentes y procedimientos

En paralelo a esta actividad, se desarrollan las actividades relacionadas con las pruebas unitarias y de integración del sistema de información. Esto permite una construcción incremental, en el caso de que así se haya especificado en el plan de pruebas y en el plan de integración del sistema de información.

4.4.3.1 Generación del Código de los Componentes

En esta tarea se genera el código correspondiente a cada uno de los componentes del sistema de información. Para generar el código fuente se tienen en cuenta los estándares de nomenclatura, codificación y calidad utilizados por la organización y recogidos en el catálogo de normas.

Con el fin de verificar que el código fuente especifica de forma correcta el componente, se realiza su ensamblaje o compilación, verificando y corrigiendo los errores sintácticos, y el enlace del código objeto obtenido con las correspondientes bibliotecas.



Codificación

En este paso se traduce el algoritmo ya estructurado, verificado y comprobado a mano, al lenguaje de programación que vaya a utilizarse. Sólo se convierten las acciones del algoritmo en instrucciones de computadora usando la sintaxis de un lenguaje particular, pero requiere de conocimientos del lenguaje y de sumo cuidado en la colocación de las instrucciones, las que deben apegarse y seguir fielmente a la lógica del algoritmo y la semántica y sintaxis del lenguaje.



Compilación

Compilación o corrección de los errores sintácticos y semánticos del código, es la eliminación de los errores "gramaticales" según las reglas de construcción de instrucciones particulares del propio lenguaje (la sintaxis). Puede hacerse a medida que se traduce, pero es mejor al final para no perder la secuencia de la codificación. Al terminar debe tenerse el código libre de los errores antes mencionados.

Para realizar la compilación puede hacerse uso de un compilador, el cual es un programa especial que analiza todo el código fuente y detecta los errores antes mencionados ocasionados durante la codificación o la digitación. Los fallos de lógica que puedan existir en nuestro programa no son detectados por este software. Los errores que sí son evidenciados por el compilador deben corregirse modificando el programa fuente.

Este trabajo de corrección de los errores sintácticos y semánticos del código, que deriva en un código libre de errores, puede efectuarse también mediante un intérprete, en lugar de con el empleo de un compilador. Un traductor intérprete toma un programa fuente, y lo traduce y ejecuta línea a línea, instrucción a instrucción, según sea necesario, y normalmente sin guardar el resultado de la traducción. El compilador realiza la traducción completa y a continuación ejecuta el código. A la hora de depurar la aplicación, puede resultar más interesante el trabajo con un intérprete antes que con un compilador, de ahí la observación realizada.

4.4.3.2 Generación del Código de los Procedimientos de Operación y Seguridad

El objetivo de esta tarea es generar los procedimientos de operación y administración del sistema de información, así como los procedimientos de seguridad y control de acceso, necesarios para ejecutar el sistema una vez que se haya implantado y esté en producción. Para la generación de dichos procedimientos se tienen en cuenta, también, los estándares y normas de la instalación recogidos en el catálogo de normas.

4.4.4 Construcción de los componentes y procedimientos de Migración y Carga Inicial de datos

El objetivo de esta actividad es la codificación y prueba de los componentes y procedimientos de migración y carga inicial de datos, a partir de las especificaciones recogidas en el plan de migración y carga inicial de datos obtenidos en el proceso *Diseño del Sistema de Información*.

Previamente a la generación del código, se prepara la infraestructura tecnológica necesaria para realizar la codificación y las pruebas de los distintos componentes y procedimientos asociados, de acuerdo a las características del entorno de migración especificado en el plan de migración y carga inicial de datos.

Finalmente, se llevan a cabo las verificaciones establecidas en la especificación técnica del plan de pruebas propio de la migración.

4.4.4.1 Preparación del Entorno de Migración y Carga Inicial de datos

Se dispone el entorno en el que se van a construir los componentes y procedimientos de migración y carga inicial de datos, considerando las bibliotecas o librerías a utilizar, herramientas o utilidades específicas para la conversión, y compiladores, entre otros.

Se determinan los datos necesarios para realizar las pruebas de los componentes y procedimientos asociados y se configura el entorno de acuerdo a dichas necesidades.

4.4.4.2 Generación del código de los componentes y procedimientos de Migración y Carga Inicial de Datos

El objetivo de esta tarea es la generación del código correspondiente a los procedimientos y componentes necesarios para llevar a cabo la migración, definidos en el plan de migración y carga inicial de datos. Para generar el código fuente se tienen en cuenta los estándares de

nomenclatura y codificación utilizados por la organización y recogidos en el catálogo de normas para este tipo de componentes.

4.4.4.3 Realización y Evaluación de las Pruebas de Migración y Carga Inicial de Datos

El objetivo de esta tarea es efectuar las pruebas de los distintos componentes y procedimientos de migración y evaluar su resultado. Esta evaluación recoge el grado de cumplimiento de las mismas, y consiste en:

- Comparar los resultados obtenidos con los esperados
- Identificar el origen de cada problema detectado para poder remitirlo a quien proceda, determinar la envergadura de las modificaciones y qué acciones deben llevarse a cabo para resolverlo de forma satisfactoria.
- Indicar si el plan de pruebas debe volver a realizarse total o parcialmente, y si será necesario contemplar nuevos casos de prueba no considerados anteriormente.

4.4.5 Elaboración de los Manuales de Usuario

El objetivo de esta tarea es elaborar la documentación de usuario, tanto usuario final como de explotación, de acuerdo a los requisitos establecidos y recogidos en la Especificación de Requisitos de Software.

Se deben especificar aspectos relativos a los tipos de documentos a elaborar y estándares a seguir en la generación de los mismos, y para cada uno de ellos:

- Formato y soporte en el que se desarrollarán.
- Estructura.
- Distribución y mantenimiento de la documentación y número de copias a editar.

4.4.6 Definición de la formación de usuarios finales

En esta actividad se establecen las necesidades de formación del usuario final, con el objetivo de conseguir la explotación eficaz del nuevo sistema.

Para la definición de la formación hay que tener en cuenta las características funcionales y técnicas propias del sistema de información, así como los requisitos relacionados con la formación del usuario final.

El producto resultante de esta actividad es la especificación de la formación de usuarios finales, que consta de los siguientes elementos:

- Esquema de formación.
- Materiales y entornos de formación

4.4.7 Entregable – Fase de Implementación

4.4.7.1 Introducción

Una vez concluidos los diagramas de clases del diseño destinados al ciclo de desarrollo actual en la aplicación dispondremos de suficientes detalles para generar un código que utilizaremos en la capa del dominio de los objetos.

Los artefactos del UML creados en la fase de diseño –los diagramas de colaboración y los de clases del diseño- servirán de entrada en el proceso de generación de código.

4.4.7.2 La programación y el proceso de desarrollo

Si se quiere reducir el riesgo y aumentar la probabilidad de conseguir una aplicación adecuada, el desarrollo debería basarse en un suficiente modelado del análisis y diseño antes de iniciar la codificación. Ello no significa que durante la programación no tengan cabida los prototipos ni el diseño: las modernas herramientas de desarrollo ofrecen un excelente ambiente para examinar rápidamente métodos alternos, normalmente vale la pena dedicar poco o mucho tiempo al diseño por la codificación.

Pero la parte esencial de la aplicación –por ejemplo el modelo conceptual básico, las capas arquitectónicas, las principales asignaciones de responsabilidades y las interacciones más importantes de los objetos- se determina más satisfactoriamente en una investigación formal y en el proceso de diseño que “apresurándose a codificar”. Esto origina sistemas que son más difíciles de entender, ampliar y de darle mantenimiento, sin que brinden soporte a un proceso susceptible de repetirse eficazmente.

Dicho lo anterior, a menudo lo más conveniente es omitir la fase de diseño para realizar un poco de programación exploratoria a fin de describir un diseño funcional y luego regresar a la fase formal de diseño.

La creación de código en un lenguaje orientado a objetos no forma parte del análisis ni del diseño orientado a objetos; es meta final en sí misma.

Una ventaja del análisis, del diseño y la programación orientados a objetos –cuando se emplean junto con un proceso de desarrollo como el que hemos recomendado- es que ofrece una guía completa de principio a fin para realizar la codificación a partir de los requerimientos. Los artefactos que se introducen en otros posteriores en una forma rastreable y útil culminarán finalmente en una aplicación funcional. Con ello no queremos decir que el camino sea fácil ni que podamos seguirlo mecánicamente, pues existen demasiadas variables. Pero la guía constituye un punto de partida para experimentar e intercambiar opiniones.

Creatividad y cambio durante la fase de construcción.

La toma de decisiones y el trabajo creativo se realizan en gran medida durante las fases de análisis y diseño. Veremos que la generación de código es un proceso de traducción bastante mecánico.

No obstante, en general la fase de programación no es un paso fácil de la generación de código, todo lo contrario. En realidad los resultados obtenidos durante el diseño son un primer paso incompleto; en la programación y en las pruebas se efectuará multitud de cambios y se descubrirán y resolverán problemas detallados. **Los artefactos del diseño cuando están bien hechos, producen un núcleo flexible que crece con elegancia y fuerza para atender los problemas que surjan en la programación.**

Cambios de código, las herramientas CASE y la ingeniería inversa.

Es conveniente que los diagramas generados en la fase de diseño sean actualizados semi-automáticamente para que incluyan los cambios en la siguiente fase de codificación. En teoría esto debería hacerse con una herramienta CASE (Computer-aided software engineering, ingeniería de software asistida por computadora), la cual puede leer el código fuente y producir automáticamente –por ejemplo- los diagramas de clases y de colaboración. Este es un aspecto de la ingeniería inversa, o sea la actividad consistente en generar modelos lógicos partiendo de un código fuente ejecutable.

4.4.7.3 Mapeo de diseños para codificación

Para implementar un lenguaje de programación orientado a objetos se requiere escribir un código fuente para:

- Definiciones de Clases.
- Definiciones de Métodos.

Creación de las definiciones de clases a partir de los diagramas de clases del diseño.

Los diagramas de clases del diseño describen por lo menos el nombre de las clases, las superclases, las etiquetas de los métodos y los atributos simples de una clase. Esa información es suficiente para formular una definición básica de una clase en un lenguaje orientado a objetos.

Creación de métodos a partir de los diagramas de colaboración.

Un diagrama de colaboración muestra los mensajes que se envían en respuesta a una llamada al método. La secuencia de los mensajes se traduce en una serie de enunciados en la definición del método.

4.4.7.4 Orden de la implementación

Es necesario implementar las clases (y, en teoría probarlas totalmente de manera individual), de la menos acoplada a la más acoplada. Por ejemplo, si observamos la figura 43, las clases posibles para implementar son *Pago* o *EspecificaciondeProducto* vienen después de las clases

que dependen únicamente de las implementaciones anteriores: *CatalogodeProductos* o *VentasLineadeProducto*.

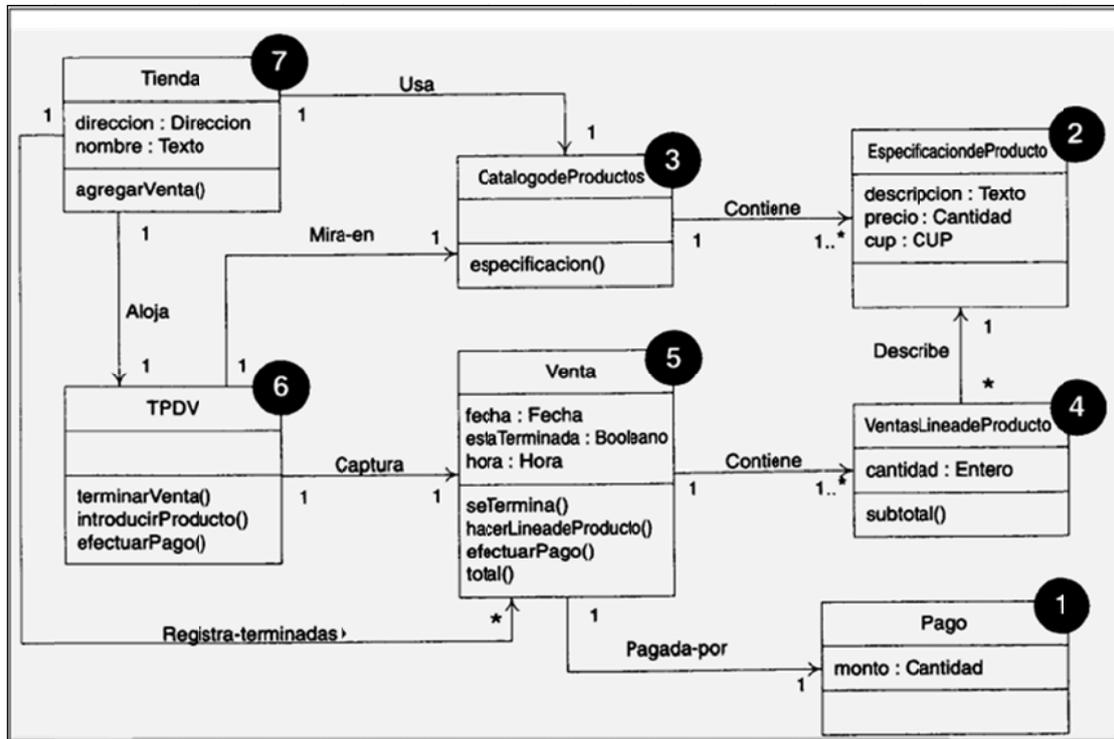


Figura 43. Ejemplo de orden de implementación

4.4.7.5 Resumen

Es muy sencillo el proceso de traducir a definiciones de clases los diagramas del diseño y también el proceso de traducir a métodos los diagramas de colaboración. En la fase de programación todavía pueden tomarse muchas decisiones, efectuarse muchos cambios de diseño y realizarse una exploración muy extensa; pero, en teoría, la arquitectura global y las decisiones más importantes ya se establecieron totalmente antes que comience la fase de codificación.

4.4.7.6 Conclusión de la Implementación

Una vez codificado, documentado y pasados los tests en cada ciclo de desarrollo iterativo, habiendo realizado las pruebas unitarias y los test de integración; con toda esta información se debe confeccionar un **Documento de diseño de Programas y Codificación** el cual acompañara a los **Manuales de Operador y Usuario** así como los de **Formación**. Los mismos configurarán los elementos más importantes del paquete de entregable que servirán de entrada a la siguiente fase.

4.4.8 Caso de Estudio: Implementación

La arquitectura del sistema está desarrollada en capas de acuerdo a la responsabilidad de cada una de ellas. También vale aclarar que se estructuró el sistema para que sea fácilmente adaptable a otra base de datos (que sea soportada por Hibernate) o a otra interfaz gráfica (que sea compatible con web services).

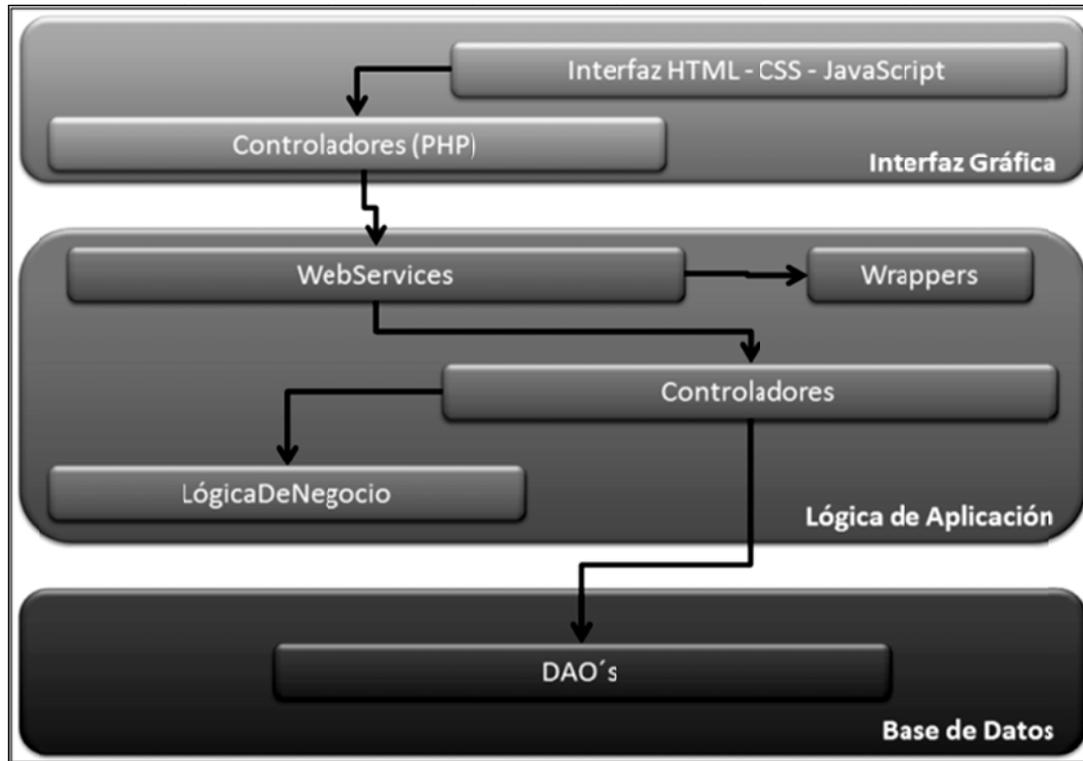


Figura 44. Arquitectura del caso de estudio presentado

4.4.8.1 Estructura de Código



Figura 45. Estructura del código del caso de estudio presentado



Figura 46. BasesDeDatos.Controlador (1)



Figura 47. LogicaDeNegocio (2)

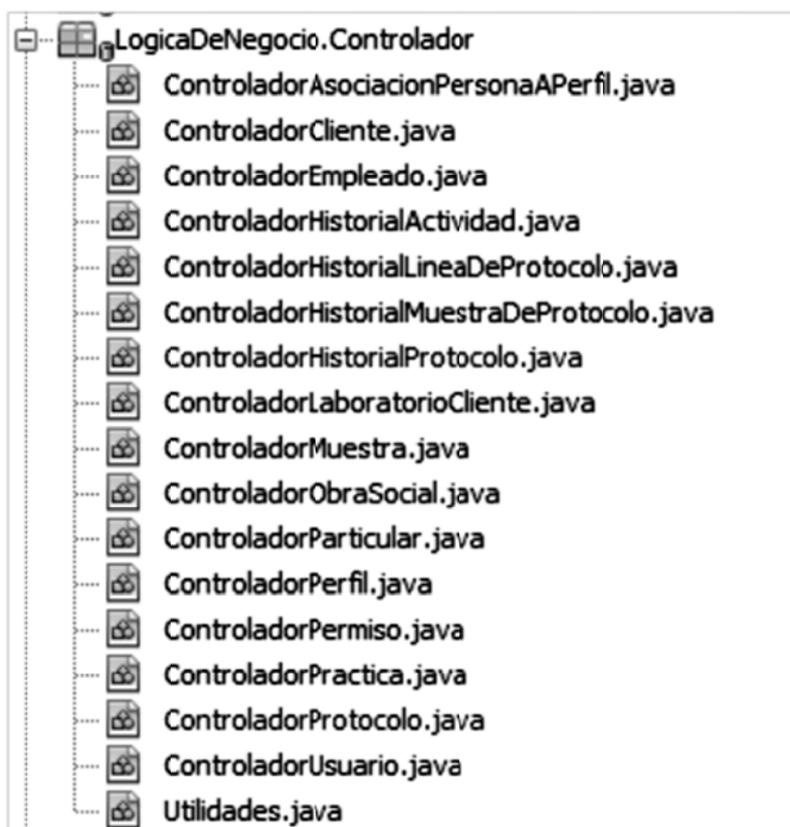


Figura 48. LogicaDeNegocio.Controlador (3)

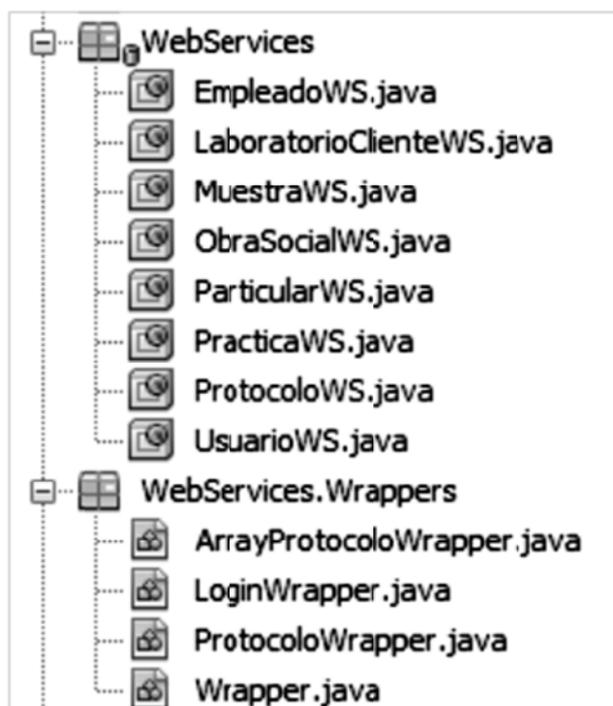


Figura 49. WebServices/WebServicesWrappers (4)

4.4.8.2 Clase *Protocolo.java*

Nota: El siguiente código de la clase *Protocolo* se presenta solo con fines demostrativos, por motivo no se incluyen ciertos atributos, métodos, importaciones y anotaciones necesarios para el funcionamiento adecuado de la clase.

```

/**
 * Sistema de Gestión Bioquímica
 * Autores: Daniel Talebi - Pablo Javier Ortiz
 * Universidad Tecnológica Nacional
 * Facultad Regional Tucumán
 * 2011
 * Prohibida su copia, distribución parcial o total a título oneroso o
 gratuito
 */

package LogicaDeNegocio;
/**
 * @author Daniel Talebi - Pablo Ortiz
 */
@Entity
public class Protocolo{
    private long id;
    private String fechaDeRecepcion;
    private String horaDeRecepcion;
    private String medico;
    private String fechaDeEntrega;
    private boolean impreso;
    private float total;
    private float deuda;
    private String observacion;
    private boolean eliminado;
    private boolean informado;
    @ManyToOne
    private Cliente cliente;
    @OneToMany
    private List<MuestraDeProtocolo> muestrasDelProtocolo;
    @OneToMany
    private List<LineaDeProtocolo> lineasDelProtocolo;

    public Protocolo(Cliente cliente, Empleado empleado,String medico,
String observacion, ArrayList<LineaDeProtocolo> lineas,
ArrayList<MuestraDeProtocolo> muestras) {
        this.cliente = cliente;
        this.medico = medico;
        this.observacion = observacion;
        this.lineasDelProtocolo = lineas;
        this.muestrasDelProtocolo = muestras;
    }
}

```

```

    this.impreso = false;
    this.eliminado = false;
    this.informado = false;
    //Calculamos el total
    for(LineaDeProtocolo indice:this.lineasDelProtocolo){
        this.total += indice.getPractica().getPrecio();
    }
    this.deuda = this.total;
}
}

```

4.4.8.3 Clase ControladorProcolo.java – Método Registrar Protocolo

```

/**
 *
 * @param idObraSocial_NumeroOrden_nbuPractica
 * @param idCliente
 * @param fechaDeRecepcion
 * @param horaDeRecepcion
 * @param medico
 * @param fechaDeEntrega
 * @param observacion
 * @return
 * @throws ProtocoloAlredyExistsException
 * El array idObraSocial_NumeroOrden_nbuPractica tiene el
siguiente formato:
 * -----
 * |int idObraSocial | String NumeroOrden | long nbuPractica |
 * |-----|
 * |          |          |          |
 * -----
 */
public static Protocolo registrarProtocolo(String
idObraSocial_NumeroOrden_nbuPractica[][] , int idMuestras[] , int
idCliente, String medico, String observacion, int idEmpleado) throws
ProtocoloCreationException{
    Protocolo protocolo = null;
    try {
        //Validar Datos
        //Verificar que los array´s tengan datos
        //Buscar al cliente
        Cliente cliente =
ControladorCliente.getClienteById(idCliente);
        //Bucle para recorrer todas los id´s de práctica
        //HACER LA CLASE LDP Y MDP
        protocolo = new Protocolo(cliente, medico, observacion);
        //Iteramos para crear las líneas de protocolo
    }
}

```

```

        for (int i = 0; i <
idObraSocial_NumeroOrden_nbuPractica.length; i++) {
            //Buscamos los id's del array
            int idObraSocial =
Integer.parseInt(idObraSocial_NumeroOrden_nbuPractica[i][0]);
            String numeroOrden =
idObraSocial_NumeroOrden_nbuPractica[i][1];
            long nbuPractica =
Long.parseLong(idObraSocial_NumeroOrden_nbuPractica[i][2]);
            //Recuperamos la Obra Social mediante el id
            ObraSocial obraSocial =
ControladorObraSocial.getObraSocialById(idObraSocial);
            //Recuperamos la Práctica mediante el nbu
            Practica practica =
ControladorPractica.getPracticaByNbu(nbuPractica);
            //Creamos la Línea de Protocolo
            LineaDeProtocolo linea = new
LineaDeProtocolo(protocolo, practica, obraSocial, numeroOrden);
            //Agregamos esa línea a la lista de líneas del
protocolo
                protocolo.addLineaDeProtocolo(linea);
            }
            //Recuperamos las muestras y la agregamos al ArrayList
            for (int i=0;i<idMuestras.length;i++){
                Muestra muestra =
ControladorMuestra.getMuestraById(idMuestras[i]);
                MuestraDeProtocolo muestraDeProtocolo = new
MuestraDeProtocolo(protocolo, muestra);
                protocolo.addMuestraDeProtocolo(muestraDeProtocolo);
            }
            Protocolo resultado = null;
            resultado = ProtocoloDAO.saveProtocolo(protocolo);
            //Mandamos el registro al Historial de Protocolo

ControladorHistorialProtocolo.registrarHistorialProtocoloDeNuevoProtoc
olo(protocolo, idEmpleado);
            //Volvemos a buscar y asignar el cliente, ya que
necesitamos que se
            //carguen el arraylist de AsociacionesAPerfil en los
objetos usuarios
            try {

resultado.setCliente(ControladorCliente.getClienteById(resultado.getCl
iente().getId()));
                } catch (ClienteNotFoundException ex) {
                    System.out.println(ex);

Logger.getLogger(ProtocoloDAO.class.getName()).log(Level.SEVERE, null,
ex);
            }
            return protocolo;
        } catch (ClienteNotFoundException ex) {

```

```

Logger.getLogger(ControladorProtocolo.class.getName()).log(Level.SEVERE,
null, ex);
    throw new ProtocoloCreationException(ex.getMessage());
} catch (ObraSocialNotFoundException ex) {

Logger.getLogger(ControladorProtocolo.class.getName()).log(Level.SEVERE,
null, ex);
    throw new ProtocoloCreationException(ex.getMessage());
} catch (PracticaNotFoundException ex) {

Logger.getLogger(ControladorProtocolo.class.getName()).log(Level.SEVERE,
null, ex);
    throw new ProtocoloCreationException(ex.getMessage());
} catch (MuestraNotFoundException ex) {

Logger.getLogger(ControladorProtocolo.class.getName()).log(Level.SEVERE,
null, ex);
    throw new ProtocoloCreationException(ex.getMessage());
} finally{
    return protocolo;
}
}

```

4.4.8.4 Clase ProtocoloDAO – Método saveProtocolo

```

public static Protocolo saveProtocolo(Protocolo objeto){
    Protocolo resultado = null;
    try{
        session = HibernateUtil.iniciarOperacion();
        long idProtocolo = (Long) session.save(objeto);
        session.beginTransaction().commit();
        resultado = (Protocolo) session.createQuery("from
Protocolo where id = " + idProtocolo).uniqueResult();
        ArrayList<MuestraDeProtocolo> muestrasDelProtocolo =
(ArrayList<MuestraDeProtocolo>) session.createQuery("from
MuestraDeProtocolo where protocolo_id = " + resultado.getId()).list();
        resultado.setMuestrasDelProtocolo(muestrasDelProtocolo);
    } catch (HibernateException ex) {
        System.out.println(ex);
        session.getTransaction().rollback();
        throw new
HibernateException("ProtocoloDAO.saveProtocolo(objeto): Ocurrió un
error en la capa de acceso a Datos.\n" + ex);
    } finally{
        session.close();
        return resultado;
    }
}
}

```

4.4.8.5 Clase *ProtocoloWs.java* – Método *registrarProtocolo*

```

/**
 * Web service operation
 */
@WebMethod(operationName = "registrarProtocolo")
public ProtocoloWrapper registrarProtocolo(@WebParam(name =
"IdObraSocial_NumeroOrden_NbuPractica")
String idObraSocial_NumeroOrden_NbuPractica, @WebParam(name =
"idMuestras")
int stringMuestras[], @WebParam(name = "idCliente")
int idCliente, @WebParam(name = "medico")
String medico, @WebParam(name = "observacion")
String observacion, @WebParam(name = "idEmpleado")
int idEmpleado) {
    Protocolo protocolo = null;
    ProtocoloWrapper resultado = null;
    //long resultado = -1;
    try {
        //Para armar el array de las obras sociales con numero
de orden
        //autorizando a una práctica determinada, usamos un
string[]
        //que tiene como separador de campo el "-"

        //Creamos un array con tantas filas como indique la
variable filas
        //y con 3 columnas, ya que es el idObraSocial,
NumeroOrden y NbuPractica
        System.out.println("La longitud del array es: " +
idObraSocial_NumeroOrden_NbuPractica);
        String filas[] =
idObraSocial_NumeroOrden_NbuPractica.split("#");
        String datos[][] = new String[filas.length][3];
        for (int i=0;i<filas.length;i++){
            String columna[] = filas[i].split("-");
            datos[i][0] = columna[0];
            datos[i][1] = columna[1];
            datos[i][2] = columna[2];
        }
        /*System.out.println("La longitud de datos es: " +
datos.length);
        for(int i=0;i<datos.length;i++){
            for(int j=0;i<3;j++){
                System.out.println(datos[i][j]);
            }
        }*/
        /*String auxiliar[] = stringMuestras.split("#");
        int idMuestras[] = new int[auxiliar.length];

```

```

        for (int i=0;i<auxiliar.length;i++){
            idMuestras[i] = Integer.parseInt(auxiliar[i]);
        }*/
        protocolo =
ControladorProtocolo.registrarProtocolo(datos, stringMuestras,
idCliente, medico, observacion, idEmpleado);
        /**
         * Eliminamos todas las referencias adicionales que no
tienen sentido
         * mostrar ahora
         */

protocolo.getCliente().setListaDeAsociacionAPerfiles(null);
        for(LineaDeProtocolo lineaActual :
protocolo.getLineasDelProtocolo()){
            lineaActual.setProtocolo(null);
        }
        for(MuestraDeProtocolo muestraActual :
protocolo.getMuestrasDelProtocolo()){
            muestraActual.setProtocolo(null);
        }

        //Envolvemos al protocolo recién creado
        resultado = new ProtocoloWrapper(true, null,
protocolo);
        } catch (Exception e) {
            System.out.println(e);
            resultado = new ProtocoloWrapper(false, e.toString(),
null);
            return resultado;
        }
        return resultado;
    }
}

```

4.5 Pruebas

"El testing puede probar la presencia de errores, pero no la ausencia de ellos." (Edsger Dijkstra)

4.5.1 Descripción

Según Wikipedia, las pruebas de software (en inglés *testing*) son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un software. Es una fase del desarrollo de software consistente en probar las aplicaciones construidas. Las Pruebas de Software consisten en ejecutar un programa y mediante técnicas experimentales descubrir que errores tiene. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

Hay muchos planteamientos a la hora de abordar el proceso de pruebas de software, pero para verificar productos complejos de forma efectiva se requiere de un proceso de investigación más que seguir un procedimiento al pie de la letra. Una definición de **"testing"** es: proceso de evaluación de un producto desde un punto de vista crítico, donde el "tester" (persona que realiza las pruebas) somete el producto a una serie de acciones inquisitivas, y el producto responde con su comportamiento como reacción. **Es necesario** realizar las pruebas en un entorno de pruebas separado físicamente del de producción. Para crear un entorno de pruebas en una máquina independiente de la máquina de producción es necesario crear las mismas condiciones que en la máquina de producción.

En general, los informáticos distinguen entre errores de programación (o "bugs") y defectos de forma.

- **Defecto de forma:** el programa no realiza lo que el usuario espera.
- **Error de programación:** fallo en la semántica de un programa de ordenador. Éste podría presentarse, o no, como un defecto de forma si se llegan a dar ciertas condiciones.

Una práctica común es que el proceso de pruebas de un programa sea realizado por un grupo independiente de "testers" al finalizar su desarrollo y antes de sacarlo al mercado. Actualmente, viene siendo muy popular distribuir de forma gratuita una versión no final del producto para que sean los propios consumidores los que la prueben. En ambos casos, a la versión del producto en pruebas y que es anterior a la versión final, se le denomina **"Beta"** o "Demo".

Puede además existir una versión anterior en el proceso de desarrollo llamada “**Alpha**”, en la que el programa, aunque incompleto, dispone de funcionalidad básica y puede ser testado.

4.5.2 La importancia de la detección oportuna

En la cadena de valor del desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallos. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan con la calidad de un producto bien desarrollado. Las aplicaciones de software han crecido en complejidad y tamaño, y por consiguiente también en costes. Hoy en día es crucial verificar y evaluar la calidad de lo construido para minimizar el coste de su reparación. Cuanto antes se detecte un fallo, más barato será su corrección.



Proceso de Prueba realizado por expertos

El proceso de prueba es un proceso técnico especializado, de investigación, que requiere de profesionales altamente capacitados en lenguajes de desarrollo, métodos y técnicas de pruebas y herramientas especializadas. El conocimiento que debe manejar un ingeniero de prueba es muchas veces superior al del desarrollador de software.

4.5.3 Tipos de Pruebas

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de prueba. El proceso de pruebas, sus objetivos y los métodos y técnicas usados se describen en un plan de pruebas.

Para esto existen diferentes tipos de pruebas tales como:

- Pruebas unitarias.
- Pruebas funcionales.
- Pruebas de integración.
- Pruebas de validación.
- Pruebas de sistema.
- Caja blanca (sistemas).
- Caja negra (sistemas).
- Pruebas de aceptación.
- Pruebas de regresión.
- Pruebas de carga.
- Pruebas de prestaciones.
- Pruebas de recorrido.

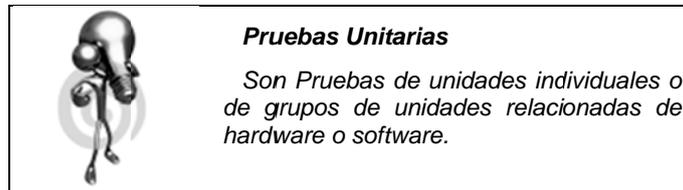
- Pruebas de mutación.
- Pruebas concurrentes.

En este libro solo se va a desarrollar una introducción sobre las Pruebas Unitarias, Pruebas de Integración, Pruebas de Sistemas y Pruebas de Regresión.

4.5.4 Ejecución de las Pruebas Unitarias

En esta actividad se realizan las pruebas unitarias de cada uno de los componentes del sistema de información, una vez codificados, con el objeto de comprobar que su estructura es correcta y que se ajustan a la funcionalidad establecida.

En el plan de pruebas se ha definido el entorno necesario para la realización de cada nivel de prueba, así como las verificaciones asociadas a las pruebas unitarias, la coordinación y secuencia a seguir en la ejecución de las mismas, y los criterios de registro y aceptación de los resultados.



4.5.4.1 Preparación del Entorno de las Pruebas Unitarias

En esta tarea se preparan todos los recursos necesarios para realizar las pruebas unitarias de cada uno de los componentes del sistema de información.

Para ello, se asegura la disponibilidad del entorno y de los datos necesarios para ejecutar estas pruebas, se preparan las bibliotecas o librerías oportunas para la realización de las mismas, así como los procedimientos manuales o automáticos necesarios, conforme a la especificación del entorno definida en el plan de pruebas.

4.5.4.2 Realización y Evaluación de las Pruebas Unitarias

El objetivo de esta tarea es comprobar el correcto funcionamiento de los componentes del sistema de información, codificados previamente, conforme a las verificaciones establecidas en el plan de pruebas para el nivel de pruebas unitarias.

Para cada verificación establecida, se realizan las pruebas con los casos de pruebas asociados, efectuando el correspondiente análisis y evaluación de los resultados, y generando un registro conforme a los criterios establecidos en el plan de pruebas.

Seguidamente, se analizan los resultados de las pruebas unitarias, evaluándose las mismas para comprobar que los resultados son los esperados. Si los resultados no son los esperados hay que proceder a realizar las correcciones pertinentes.

4.5.5 Ejecución de las Pruebas de Integración

El objetivo de las pruebas de integración es verificar si los componentes o subsistemas interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida, y se ajustan a los requisitos especificados en las verificaciones correspondientes.

La estrategia a seguir en las pruebas de integración se establece en el plan de pruebas, donde se habrá tenido en cuenta el plan de integración del sistema de información.

4.5.5.1 Preparación del Entorno de las Pruebas de Integración

En esta tarea se disponen todos los recursos necesarios para realizar las pruebas de integración de los componentes y subsistemas que conforman el sistema de información.

Para ello, se asegura la disponibilidad del entorno y de los datos necesarios para ejecutar estas pruebas, se preparan las bibliotecas o librerías que se estimen oportunas para la realización de las mismas, así como los procedimientos manuales o automáticos asociados, conforme a la especificación del entorno definida en el plan de pruebas.

4.5.5.2 Realización de las Pruebas de Integración

El objetivo de esta tarea es verificar el correcto funcionamiento de las interfaces existentes entre los distintos componentes y subsistemas, conforme a las verificaciones establecidas para el nivel de pruebas de integración.

Para cada verificación establecida, se realizan las pruebas con los casos de pruebas asociados, efectuando el correspondiente análisis e informe de los resultados de cada verificación, y generando un registro conforme a los criterios establecidos en el plan de pruebas.

4.5.5.3 Evaluación del Resultado de las Pruebas de Integración

El objetivo de esta tarea es analizar los resultados de las pruebas de integración y efectuar su evaluación. Dicha evaluación recoge el grado de cumplimiento de las pruebas y consiste en:

- Comparar los resultados obtenidos con los esperados.
- Identificar el origen de cada problema detectado para poder remitirlo a quien proceda, determinar la envergadura de las modificaciones y qué acciones deben llevarse a cabo para resolverlo de forma satisfactoria.
- Indicar si el plan de pruebas debe volver a realizarse total o parcialmente, y si será necesario contemplar nuevos casos de prueba no considerados anteriormente.

4.5.6 Ejecución de las Pruebas del Sistema

El objetivo de las pruebas del sistema es comprobar la integración del sistema de información **globalmente**, verificando el funcionamiento correcto de las interfaces entre los distintos

subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.

En la realización de estas pruebas es importante comprobar la cobertura de los requisitos, dado que su incumplimiento puede comprometer la aceptación del sistema por el equipo de operación responsable de realizar las pruebas de implantación del sistema, que se llevarán a cabo en el proceso Implantación y Aceptación del Sistema.

4.5.6.1 Preparación del Entorno de las Pruebas del Sistema

En esta tarea se preparan todos los recursos necesarios para realizar las pruebas del sistema, de acuerdo a las características del entorno establecidas en el plan de pruebas.

Para ello se asegura la disponibilidad del entorno y de los datos necesarios para ejecutar estas pruebas, se preparan las bibliotecas o librerías que se estimen oportunas para la realización de las mismas, así como los procedimientos manuales o automáticos asociados.

4.5.6.2 Realización de las Pruebas del Sistema

El objetivo de esta tarea es comprobar la integración de todos los subsistemas y componentes del sistema de información, así como la interacción del mismo con otros sistemas de información con los que se relaciona, de acuerdo a las verificaciones establecidas para el nivel de pruebas del sistema.

Para cada verificación establecida, se realizan las pruebas con los casos de pruebas asociados, efectuando el correspondiente análisis e informe de los resultados y generando un registro conforme a los criterios establecidos en el plan de pruebas.

4.5.6.3 Evaluación del Resultado de las Pruebas del Sistema

El objetivo de esta actividad es analizar los resultados de las pruebas del sistema de información y efectuar su evaluación. Dicha evaluación recoge el grado de cumplimiento de las mismas, y consiste en:

- Comparar los resultados obtenidos con los esperados
- Identificar el origen de cada problema detectado para poder remitirlo a quien proceda, determinar la envergadura de las modificaciones y qué acciones deben llevarse a cabo para resolverlo de forma satisfactoria.
- Indicar si el plan de pruebas debe volver a realizarse total o parcialmente, y si será necesario contemplar nuevos casos de prueba no considerados anteriormente.



Formalización de las pruebas

Es altamente recomendable utilizar una metodología concreta y reconocida para las pruebas del sistema de información, así como para las demás fases del desarrollo del software. El módulo "Construcción del Sistema de Información" de MÉTRICA versión 3 puede ser utilizado.

4.5.7 Pruebas de Regresión

Se denominan Pruebas de Regresión a cualquier tipo de pruebas de software que intentan descubrir las causas de nuevos errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, inducidos por cambios recientemente realizados en partes de la aplicación que anteriormente al citado cambio no eran propensas a este tipo de error. Esto implica que el error tratado se reproduce como consecuencia inesperada del citado cambio en el programa.

Este tipo de cambio puede ser debido a prácticas no adecuadas de control de versiones, falta de consideración acerca del ámbito o contexto de producción final y extensibilidad del error que fue corregido (fragilidad de la corrección), o simplemente una consecuencia del rediseño de la aplicación. Por lo tanto, en la mayoría de las situaciones del desarrollo de software se considera una buena práctica que cuando se localiza y corrige un bug, se grabe una prueba que exponga el *bug* y se vuelvan a probar regularmente después de los cambios subsiguientes que experimente el programa.

Existen herramientas de software que permiten detectar este tipo de errores de manera parcial o totalmente automatizada, la práctica habitual en programación extrema es que este tipo de pruebas se ejecuten en cada uno de los pasos del ciclo de vida del desarrollo del software.

Cuando se lleva a cabo un cambio en un producto software, se deben realizar estas **Pruebas de Regresión** para evitar el “efecto onda”. Con ellas, se comprueba que los cambios sobre un componente del software no introduzcan un comportamiento no deseado o errores en otros componentes no modificados.

Tipos de regresión:

- Clasificación de ámbito
 - Local; los cambios introducen nuevos errores.
 - Desenmascarada; los cambios revelan errores propios.
 - Remota; los cambios vinculan algunas partes del programa (módulo) e introducen errores en ella.
- Clasificación temporal
 - Nueva característica; los cambios realizados con respecto a nuevas funcionalidades en la versión, introducen errores en otras novedades de la misma versión del software.
 - Característica preexistente; los cambios realizados con respecto a nuevas funcionalidades introducen errores en funcionalidad existentes de previas versiones.

4.5.8 Entregable – Fase de Pruebas

4.5.8.1 Introducción

Esta fase, da inicio luego de que las diferentes unidades de diseño han sido desarrolladas y probadas por separado. Durante el desarrollo del sistema se emplean de forma experimental para asegurar que el software no falle, es decir, que funcione de acuerdo a sus especificaciones y a la manera que los usuarios esperan que lo haga, y de esta forma poder detectar cualquier anomalía, antes de que el sistema sea puesto en marcha.

Si el sistema cumple de forma satisfactoria con las pruebas, se procede a realizar la carga de los archivos, base de datos y tablas del nuevo sistema, para de esta forma dar inicio al proceso de aceptación final, durante el cual, el sistema comenzará a funcionar por un lapso determinado de tiempo llamado Periodo de Aceptación. Finalizado el Periodo de Aceptación, se le dará al sistema la aprobación final, para que pase a ser el sistema oficial.

4.5.8.2 Tipos de Pruebas

El director de proyecto debe conocer que tipos de pruebas se realizarán al sistema ya implementado, y la utilidad de cada una de ellas. Las pruebas de unidades son el primer tipo de prueba que se aplica. El siguiente nivel consiste en las pruebas de integración. Esto valida la funcionalidad global de cada etapa de la aplicación parcial. Por último las pruebas de sistema y de aceptación validan el producto final, como lo muestra la siguiente figura 50:

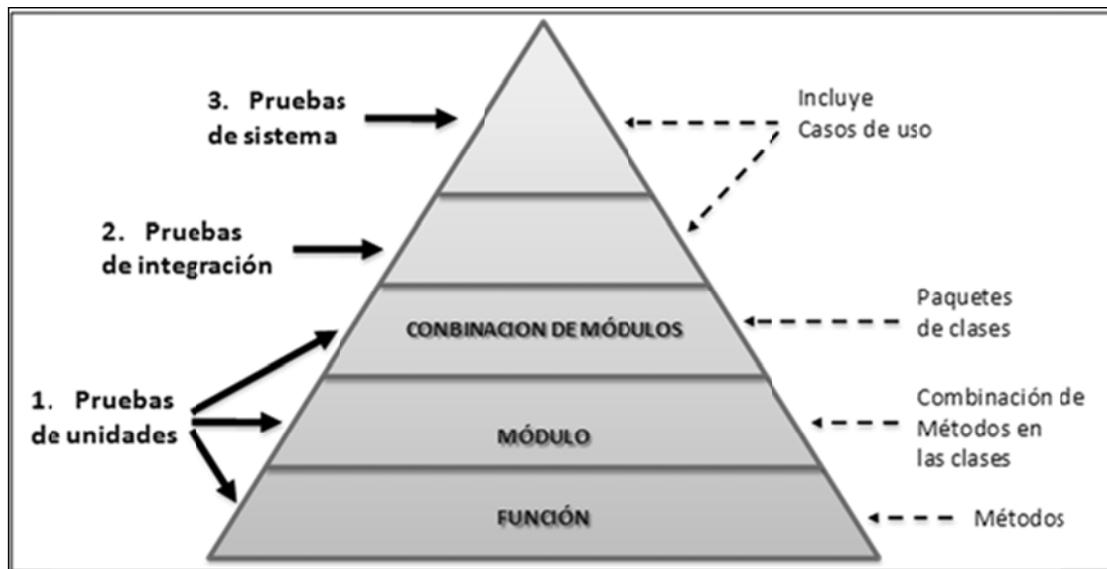


Figura 50. Pruebas de visión global

4.5.8.3 Documentos más usuales en cada fase

A todos estos documentos hay que añadir documentos con la estimación y planificación de la próxima fase y del resto del proyecto. También habrá que ir actualizando el índice de todo el material relacionado. Vamos a listar los más importantes:

- Plan de pruebas del sistema (actualizado).
- Informe de los resultados de las pruebas.
- Descripción de las pruebas, el resultado esperado, resultado obtenido y acciones a tomar para corregir las desviaciones.

Plan de Pruebas

El propósito del plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas. Note que puede haber un plan global que explicita el énfasis a realizar sobre los distintos tipos de pruebas (verificación, integración e integración). Luego de haber ejecutado las pruebas se actualizará los elementos contenidos en el mismo.

Un plan de pruebas incluye:

- ✓ Identificador único del documento.
- ✓ Introducción y resumen de elementos y características a probar.
- ✓ Elementos software a probar.
- ✓ Características a probar.
- ✓ Características que no se probarán.
- ✓ Enfoque general de la prueba.
- ✓ Criterios de paso/fallo para cada elemento.
- ✓ Criterios de suspensión y requisitos de reanudación.
- ✓ Documentos a entregar.
- ✓ Actividades de preparación y ejecución de pruebas.
- ✓ Necesidades de entorno.
- ✓ Responsabilidades en la organización y realización de las pruebas.
- ✓ Necesidades de personal y formación.
- ✓ Esquema de tiempos.
- ✓ Riesgos asumidos por el plan y planes de contingencias.
- ✓ Aprobaciones y firmas con nombre y puesto desempeñado.

Informes de los resultados de las pruebas

Una vez ejecutadas las pruebas se documentará los resultados a partir de la generación de:

1. **El histórico de pruebas (test log)** documenta todos los hechos relevantes ocurridos durante la ejecución de las pruebas, con los siguientes elementos:
 - ✓ Identificador.
 - ✓ Descripción de la prueba: elementos probados y entorno de la prueba.
 - ✓ Anotación de datos sobre cada hecho ocurrido (incluido el comienzo y el final de la prueba).
 - ✓ Fecha y hora.
 - ✓ Identificador de informe de incidente.
 - ✓ Otras informaciones.
2. **El informe de incidente (test incident report)** documenta cada incidente (por ejemplo, una interrupción en las pruebas debido a un corte de electricidad, bloqueo del teclado, etc.) ocurrido en la prueba y que requiera una posterior investigación. Su estructura se define así:
 - ✓ Identificador.
 - ✓ Resumen del incidente.
 - ✓ Descripción de datos objetivos (fecha/hora, entradas, resultados esperados, etc.).
 - ✓ Impacto que tendrá sobre las pruebas.
3. **El informe resumen (test summary report)** resume los resultados de las actividades de prueba (las señaladas en el propio informe) y aporta una evaluación del software basada en dichos resultados. Incluye:
 - ✓ Identificador.
 - ✓ Resumen de la evaluación de los elementos probados.
 - ✓ Variaciones del software respecto a su especificación de diseño, así como las variaciones en las pruebas.
 - ✓ Valoración de la extensión de la prueba (cobertura lógica, funcional, de requisitos, etc.).
 - ✓ Resumen de los resultados obtenidos en las pruebas.
 - ✓ Evaluación de cada elemento software sometido a prueba (evaluación general del software incluyendo las limitaciones del mismo).
 - ✓ Firmas y aprobaciones de quienes deban supervisar el informe.

Comparativo de los resultados de las pruebas

Una vez realizadas las pruebas se deben registrar los resultados y comparar con la planificación, de manera que se identifiquen las posibles desviaciones y establezcan acciones concretas a realizar, las que generaran nuevos requerimientos o modificarán los existentes, para que los desarrolladores los implementen.

4.5.9 Conclusión de la fase de Pruebas

Aunque se han desarrollado miles de herramientas de soporte de esta fase, todas han limitado su éxito a entornos muy concretos, frecuentemente sólo sirviendo para el producto para el que se desarrollaron. Sólo herramientas muy generales como **analizadores de complejidad, sistemas de ejecución simbólica y medidores de cobertura** han mostrado su utilidad en un marco más amplio. Pero al final sigue siendo imprescindible un artista humano que sepa manejarlas.

4.5.10 Caso de Estudio: Pruebas

4.5.10.1 Pruebas Unitarias

Debido a que el software desarrollado es un prototipo cuyo fin es demostrar el objetivo del presente proyecto, la codificación del mismo no evidencia necesariamente las técnicas adecuadas de programación, por tal motivo se opta por no realizar esta prueba, ya que se pueden detectar errores cuya presencia están justificados por la necesidad de simplificación del código.

4.5.10.2 Pruebas de Integración

Basándonos en la antes mencionada, no es posible desarrollar estas pruebas ya que no se realizaron las pruebas unitarias

4.5.10.3 Pruebas del Sistema

Se realiza este tipo de prueba considerando al sistema como una caja negra, a la cual se le proveen entradas y se esperan salidas determinadas, de acuerdo a la funcionalidad propuesta por el requisito desarrollado.

Tal como lo establece el Caso de Uso Real, se proveen dos conjuntos de datos de entrada, especificados en la tabla de pruebas I (figura 51) y la tabla de pruebas II (figura 52), los cuales luego se verán reflejados en las figuras 53 y 54 que demuestran el ingreso de los mismos en el sistema. Los datos de salida esperados están descritos como sigue en la tabla de pruebas III (figura 55) y en la figura 56, se puede corroborar que la prueba se realizó correctamente:

1er Conjunto de Datos	
Cliente Particular (radio button activado)	María Estevez
Laboratorio Cliente (radio button desactivado)	Clinical Lab SA
Médico	Dr. José Correa
Obra Social	OSDE Binario
Número de Orden	AX254684
Prácticas	TC4 Hemograma
Muestras	Sangre Plasma
Observaciones	Trajo sus propias muestras

Figura 51. Tabla de pruebas I

2do Conjunto de Datos	
Monto de Pago	50.75

Figura 52. Tabla de pruebas II

Registrar Nuevo Protocolo

Dar de alta a una nueva protocolo en el sistema

Ciente

Cliente Particular

Laboratorio Cliente

Médico

Obra Social

Número de Orden

NBU Práctica

Muestra

Observaciones

Figura 53. Ingreso de datos, tabla de pruebas I

Registrar Nuevo Protocolo

Monto a Pagar

Figura 54. Ingreso de datos, tabla de pruebas II

Resultados Esperados				
Fecha de Recepción	11/03/2011			
Hora de Recepción	17:32			
Fecha Probable de Entrega	13/03/2011			
Ciente	María Estevez			
Médico	Dr. José Correa			
Total	\$56.75			
Deuda	\$6			
Observaciones	Trajo sus propias muestras			
Práctica	TC4	OSDE Binario	AX254684	\$53.25
	Hemograma	OSDE Binario	AX254684	\$3.5
Muestras	Sangre	11/03/2011		
	Plasma	11/03/2011		

Figura 55. Tabla de pruebas III

Registrar Nuevo Protocolo

Información	Descripción
Número de Protocolo	2
Fecha de Recepción	11/3/2011
Hora de Recepción	5:32
Fecha Probable de Entrega	13/3/2011
Cliente	María Estevez
Médico	SDr. José Correa
Total	\$56.75
Deuda	\$6
Observaciones	Trajo sus propias muestras

Práctica	Obra Social	Número de Orden	Subtotal
TC4	OSDE Binario	AX254684	\$53.25
Hemograma	OSDE Binario	AX254684	\$3.5

Muestras	Fecha de Recepción
<u>Sangre</u>	11/3/2011
<u>Plasma</u>	11/3/2011

Figura 56. Actividad de pruebas III

4.6 Implantación

4.6.1 Descripción

Una vez analizado, diseñado, codificado y probado el software, éste está listo para entrar en la fase de **Implantación**. En esta fase se instala el software sobre la plataforma (hardware) final, se llevan a cabo los test de aceptación especificados, y se comprueba que el programa satisface los requisitos para los que fue concebido. En tales condiciones, el software recibe la aceptación provisional, y comienza su operación y uso.

Durante la fase de implantación se genera el documento de implantación, donde se describen las actualizaciones de instalación y verificación realizadas, así como el documento de aceptación provisional.

A continuación se reseñan algunos aspectos de interés aplicables a esta fase:

- En las técnicas de aceptación participará el usuario y el equipo de desarrollo.
- Las pruebas de aceptación fallidas darán lugar a cambios en el diseño del software.
- La verificación de que el software cumple los requisitos definidos derivará en la aceptación provisional del mismo.
- El resultado de esta fase incluirá el conjunto de informes de aceptación, junto con la documentación asociada a los cambios realizados durante la fase.

Al implantar un Sistema de Información lo primero que debemos hacer es asegurarnos que el Sistema sea operacional, es decir, que funcione de acuerdo a los requerimientos del análisis y permitir que los usuarios puedan operarlo.

Existen varios enfoques de Implantación, pero en cualquiera de ellos se deben mínimamente considerar:

- Usar diferentes estrategias para el entrenamiento de los usuarios.
- El Analista necesita ponderar la situación y proponer un plan de conversión que sea adecuado para la organización.
- El Analista necesita formular medidas de desempeño con las cuales evaluar a los Usuarios.
- Debe convertir físicamente el sistema de información antiguo, al nuevo modificado.
- En la preparación de la Implantación, aunque el sistema esté bien diseñado y desarrollado correctamente, su éxito dependerá de su implantación por lo que es importante capacitar al usuario con respecto a su uso y mantenimiento.

4.6.2 Capacitación de Usuarios del Sistema

La capacitación de usuarios del sistema es enseñar el manejo del sistema a los usuarios que se relacionan u operan en un proceso de implantación



Capacitación de Usuarios

No se debe incluir en una misma capacitación a personas de diferentes niveles de habilidad e intereses de trabajo; debido a que si en una Empresa existen trabajadores inexpertos no se pueden incluir en la misma sección de los expertos ya que ambos grupos quedarán perdidos. "Es como querer conducir dos Barcos con diferentes destinos, con un mismo Mapa de rutas o con el mismo timón".

Aun y cuando la Empresa pueda contratar los Servicios de Instructores externos, el analista es la persona que puede ofrecer la mejor capacitación debido a que conoce el personal y el sistema mejor que cualquier otro. A la falta o imposibilidad del analista la organización puede contratar otros servicios de capacitación como son:

- **Vendedores:** Son aquellos que proporcionan capacitación gratuita fuera de la Empresa de uno o dos días.
- **Instructor pagado externamente:** Son aquellos que pueden enseñar todo acerca de las computadoras pero para algunos usuarios esta no es una capacitación necesaria.
- **Instructores en casa:** Están familiarizados con el personal y pueden adecuar los materiales a sus necesidades, pero les falta experiencia en Sistemas de Información que es realmente la necesidad del usuario.

4.6.3 Objetivos de la Capacitación

Es lograr que los usuarios tengan el dominio necesario de las cosas básicas acerca de los procesos que se emplean para la operación de manera eficiente y segura del sistema.

4.6.4 La Evaluación del Sistema

Se lleva a cabo para identificar puntos débiles y fuertes del Sistema implantado. La evaluación ocurre a lo largo de cualquiera de las siguientes cuatro dimensiones:

- **Evaluación operacional:** Se evalúa la manera en que funciona el Sistema, esto incluye su facilidad de uso, Tiempo de respuesta ante una necesidad o proceso, como se adecuan los formatos en que se presenta la Información, contabilidad global y su nivel de Utilidad.
- **Impacto Organizacional:** Identifica y mide los beneficios operacionales para la Empresa en áreas tales como, Finanzas (Costes, Ingresos y Ganancias), eficiencia en el desempeño laboral e impacto competitivo, rapidez y organización en el flujo de información interna y externa.
- **Desempeño del Desarrollo:** Es la evaluación del proceso de desarrollo adecuado tomando en cuenta ciertos criterios, como tiempo y esfuerzo que concuerden con presupuesto y estándares y otros criterios de Administración de Proyectos. Además se incluyen la valoración de los métodos y las herramientas utilizados durante el desarrollo del Sistema.

4.6.5 Control de configuración en la fase de implantación

En un proyecto de este tipo, todos los elementos y componentes relacionados con el desarrollo del software deben ser objeto de control de configuración, y estar sometidos a los procedimientos y normas relacionados con el mismo.

Cada elemento de configuración estará identificado unívocamente, y definido por su propósito, su funcionalidad, sus requisitos y sus interfaces, indicando para cada uno el número de versión (y revisión) actual.

Como se dijo ya en esta fase: *“Las pruebas de aceptación fallidas darán lugar a cambios. ...”*, cada vez que alguna de las partes (cliente, equipo de trabajo, usuarios) detecte algún fallo o disconformidad en el software o en la documentación generada. Es conveniente generar, como parte de las actividades de control de configuración, una **hoja de notificación de disconformidad**, donde se refleje y describa la misma, la solución recomendada y la solución adoptada. La figura 57 muestra un modelo de notificación utilizado.

Notificación de disconformidad		Nota N°
PROYECTO		Fecha :
		Autor :
Elemento :		
Referencia :		
Edición-versión / revisión:		
Localización del problema :		
Descripción del problema :		
Solución recomendada :		
Respuesta del autor :		
Estado de la disconformidad		
<input type="checkbox"/> CERRAR	<input type="checkbox"/> SEGUIR	<input type="checkbox"/> ACCION

Figura 57. Notificación de disconformidad

La notificación de una incidencia o disconformidad dará lugar, por lo general, a una actuación que responda y (en su caso) resuelva la misma. A menudo pasará por introducir cambios en

alguno de los elementos de configuración (código o documentación). Considerar adecuadamente dichos cambios es vital.

Como respuesta al cambio mismo, se generará un documento en el que se documenten los cambios realizados, **informe de modificación**, el responsable y la fecha, así como las razones y el esfuerzo asociado a las mismas.

En la figura 58 se muestra un posible formato para dicho informe.

Informe de Modificación del Software PROYECTO		Inf. N°	
		Fecha :	
		Autor :	
Elemento :	[Redacted]		
Referencia :	[Redacted]		
Edición-versión / revisión:	[Redacted]		
Cambios implantados :	[Redacted]		
Razones del cambio :	[Redacted]		
Fechas de comienzo y fin del cambio, y esfuerzo necesario :	[Redacted]		
Documentación adjunta			
<input type="checkbox"/> Código fuente	<input type="checkbox"/> Procedimiento de prueba	<input type="checkbox"/> Datos test	
<input type="checkbox"/> Resultados de test	<input type="checkbox"/> Documentación adicional	<input type="checkbox"/> Otros	

Figura 58. Informe de modificación de software

Junto con el software, también los documentos del proyecto sufrirán modificaciones. Un cambio en el software originado por una notificación de disconformidad puede obligar a revisar y actualizar todo el árbol de documentación del proyecto.

Cada elemento objeto de control de configuración es acompañado por una **hoja de control de configuración**, y sirve para reflejar qué versión se está manejando actualmente, y qué cambios y modificaciones incluye con respecto a ediciones anteriores.

En esta figura 59 se muestra el posible formato de dicho documento:

- Versión por configuración (hardware o estaciones de trabajo).
- Llevar a cabo cualquier conversión de datos necesaria.
- Llevar a cabo la instalación del sistema nuevo a producción.
 - Instalación completa desde cero.
 - Instalación en paralelo.
 - Instalación por fases.
- Comenzar el uso de los acuerdos de nivel de servicio.
- Planificar y programar las revisiones post-instalación:
 - Establecer los criterios de:
 - Rendimiento del sistema.
 - Calidad del sistema.
 - Satisfacción del usuario.
 - Calidad y facilidad de manejo de:
 - Manuales de usuario y operador.
 - Formación de usuarios y operadores.
 - Información y datos producidos.
 - Fluidez de la instalación.
 - Costes de desarrollo, instalación, operaciones y mantenimiento.
 - Establecer la planificación y calendario para las revisiones.
 - Asegurar la disponibilidad de:
 - Personal requerido.
 - Documentación requerida.
- Llevar a cabo las revisiones post-instalación:
 - Crear el informe de la revisión post-instalación.
 - Obtener la aprobación firmada de los informes de:
 - Usuarios finales del sistema.
 - Operadores del sistema.
 - Auditoría y garantía de la calidad.
 - Desarrollo de sistemas.
 - Soporte de sistemas y mantenimiento.

- Finanzas.
 - Obtener la carta de aprobación del sistema.
- Establecer el calendario para otras revisiones post-instalación si es necesario.

4.7 Conclusión del Proyecto

4.7.1 Descripción

Una vez finalizado el proyecto, parece evidente la necesidad de analizar los resultados y recapitular el curso de los hechos para hacerse una idea clara de los objetivos cumplidos, de los que no se han alcanzado, y de la utilidad futura en otros proyectos, del trabajo realizado.

4.7.2 Objetivos del cierre del Proyecto

Según *Domingo Ajenjo*, a grandes rasgos el cierre del proyecto tiene como objetivo principal:

- Analizar desde la perspectiva económica el resultado del proyecto, es decir, hacer balance de los recursos consumidos y los beneficios alcanzados.
- Diagnosticar el funcionamiento de la empresa, identificando las desviaciones entre el resultado y las previsiones iniciales, y encontrando las razones de dichas desviaciones.
- Corregir, para futuros proyectos, las actuaciones inadecuadas que dieron lugar a las desviaciones anteriores.

Y como objetivos secundarios:

- Consolidar como parte del *know-how* de la empresa los resultados técnicos del proyecto, incluyendo los conocimientos adquiridos, la tecnología utilizada, la documentación y los productos desarrollados., etc.
- Identificar las nuevas oportunidades comerciales nacidas de la ejecución del proyecto recién terminado, y organizar las actividades comerciales precisas para dar continuidad, mediante nuevos contratos, al proyecto anterior.

Asimismo para alcanzar la conclusión efectiva del proyecto se debe considerar:

- 1) La ACEPTACION: El proyecto no puede darse por terminado hasta que el Cliente revise y acepte el resultado del mismo.
- 2) Se debe crear un INFORME DE CIERRE DEL PROYECTO: Es en teoría la última documentación del proyecto. Es como una foto de fin de curso en la que se aprecia

quién termina y quién no (además, por las caras uno puede llegar a intuir si el curso fue bueno o malo).

- 3) Realizar un análisis de INDICADORES OBJETIVOS DEL RESULTADO DEL PROYECTO: Lo más probable es que la documentación de cierre del proyecto se curse, de oficio, a los responsables de la empresa, quienes la analizarán en busca de estos indicadores para evaluar la marcha de los proyectos.

En esta fase se trata de dar por finalizado el proyecto y entregar el producto definitivamente al cliente. Las principales actividades a realizar son estas:

- Revisar las desviaciones del proyecto e identificarlas para evitarlas en futuros proyectos.
- Reasignar el personal a los nuevos proyectos o reintegrarlos en los departamentos de partida.
- Es interesante documentar las relaciones entre los empleados para futuros proyectos.

4.7.3 Acciones a seguir por el director del proyecto para efectivamente concluir con el proyecto

1. El director deberá ir cancelando relaciones con proveedores.
2. Respecto de los recursos humanos, deberá ir dejando sin efecto los vínculos generados entre dichos recursos y las actividades asignadas en este proyecto, liberándolos para otros proyectos.
3. Una vez realizado el punto anterior el director solicitará al cliente la aceptación del proyecto.
4. Deberá además poner a disposición del cliente de forma ordenada y accesible toda la información relativa al proyecto.
5. El director de proyecto llevará adelante una reunión, con informe final, aceptación y algún documento para formalizar esta fase del proyecto (tipo acta de cierre). En esta reunión se firmará este documento aceptando ambas partes que se cumplió con lo establecido.
6. Se darán por finalizadas las relaciones internas y externas vinculadas al proyecto.

J.Meredith y S. Mantel en 1989 describen en su trabajo tres formas de concluir un proyecto:

- **Por extinción**, donde el proyecto ah sido llevado a cabo, de forma exitosa o habiendo fracasado y se ah acordado su finalización.

- **Por inclusión**, generalmente se supone que el proyecto ha sido un éxito y se institucionaliza en la organización. En este caso el proyecto es visto como una transformación. Cobra mucha importancia el proceso de transición del proyecto.
- **Por integración**, supone generalmente la distribución del personal, el equipo y el material en la organización principal.

La labor del Director de Proyectos en cuanto a la conclusión es muy intensa, así que es importante tener presente el tipo de conclusión para poder anticiparse a las tareas. Asimismo no debe conformarse con haber finalizado el proyecto, siempre debe pensar cómo hacerlo mejor, ya sea porque existen nuevas oportunidades siguiendo la evolución del proyecto en la organización donde se ha implantado, o bien porque otros clientes pueden solicitar algo similar. Además siempre debe ser creativo, aplicando nuevos modelos y tecnologías adaptando diferentes soluciones a los problemas que surjan, cualquier cosa que haga mejor la idea original, más eficaz y más rentable (mejores resultados a un menor costo), le permitirán desarrollar sus competencias al máximo preparándolo para los vertiginosos futuros desafíos.



Concluir el proyecto sin sobresaltos

Es altamente recomendable dejar bien establecidos, de manera concreta y sin ambigüedades los requisitos de la solución adoptada, y su aprobación por parte del cliente de manera escrita por medio de algún documento. Sin este requisito sería muy difícil llevar adelante una conclusión efectiva del proyecto, ya que el cliente podría solicitar mejoras de manera infinita.

4.8 Mantenimiento

4.8.1 Descripción

El mantenimiento del software es una de las actividades más comunes en la ingeniería del software. Es el proceso de mejora y optimización del software **después de su entrega** al usuario final (es decir; revisión del programa), así como también la corrección y prevención de los defectos. Es importante recordar:

*“El proyecto no finaliza cuando se entrega el producto, sino cuando el cliente está satisfecho.”
(Craig Larman)*

Desarrollar un mantenimiento efectivo del software es una de las bases para obtener un alto grado de satisfacción del cliente. El mantenimiento de software es una de las fases en el ciclo de vida de desarrollo de sistemas, que se aplica al desarrollo de software. La fase de mantenimiento es la fase que viene después de la implantación.



Mantenimiento del software

El mantenimiento del software puede entenderse como el "proceso de modificar un sistema o componente software después de la entrega al cliente o usuario, para corregir defectos, mejorar el rendimiento o atributos, adaptarlo a un cambio de entorno, o ampliar su funcionalidad" [IEEE, 1990].

Durante este capítulo vamos a describir la información más importante a tener en cuenta por un Director de Proyectos Informáticos en la fase de mantenimiento. No se profundizará en las técnicas o metodologías para realizar el mantenimiento del software, sino que solo considerarán los puntos más importantes para la Dirección de Proyectos Informáticos.

“Una vez que un sistema pasa a formar parte de la vida diaria de la empresa, cada programa, cada procedimiento y cada estructura de datos, se convierte en una pieza del negocio que, como tal, deberá funcionar en forma constante, exacta y confiable. La operación del negocio ahora dependerá del funcionamiento del sistema, por lo que las tareas de mantenimiento cobran vital importancia. Durante la fase de mantenimiento, se ponen en práctica todas las políticas y los procedimientos destinados a garantizar la operación continua de los sistemas para asegurar su uso efectivo, con el fin de que éstos se constituyan en una verdadera herramienta de apoyo al logro de los objetivos estratégicos de la empresa.” (Lloren Fábregas)

Una vez que el nuevo sistema esté operativo, el auditor de sistemas independiente de las otras fases de la vida del sistema, revisará lo siguiente:

- Determinar si el programa ha logrado los requerimientos de los objetivos; se debe prestar especial atención a la utilización y la satisfacción de los usuarios finales, ya que ellos constituirán un indicador excelente.
- Verificar que se miden, analizan e informan adecuadamente a la gerencia los beneficios identificados con el estudio de factibilidad.
- Revisar las solicitudes de cambios a los programas que se han realizado, para evaluar el tipo de cambios que se exigen al sistema, el tipo de cambios puede indicar problemas de diseño, programación o interpretación de los requerimientos de usuario.

La fase de mantenimiento involucra cambios en el software con el objetivo de corregir defectos y dependencias encontradas durante su uso, añadir nueva funcionalidad para mejorar la usabilidad y aplicabilidad del software.

En un ambiente formal de desarrollo de software, la organización o equipo de desarrollo tendrá algún mecanismo para documentar y rastrear defectos y deficiencias. El software, al igual que otros productos, generalmente es entregado con un conjunto de defectos y deficiencias. Las deficiencias conocidas son normalmente documentadas en una carta de consideraciones operacionales o notas de entrega. Así que los usuarios del software serán

capaces de trabajar evitando las deficiencias conocidas y conocerán cuándo el uso del software sería inadecuado para tareas específicas.

Las personas involucradas en la fase de mantenimiento del software esperan trabajar en estos defectos conocidos, ubicarlos, y preparar una nueva entrega del software, conocido como una versión de mantenimiento, la cual resolverá los temas pendientes.

4.8.2 Tipos de Mantenimiento

A continuación se señalan los tipos de mantenimientos existentes, definidos tal y como se especifican para la metodología MÉTRICA v3:

- **Perfectivo:** son las acciones llevadas a cabo para mejorar la calidad interna de los sistemas en cualquiera de sus aspectos: reestructuración del código, definición más clara del sistema y optimización del rendimiento y eficiencia.
- **Evolutivo:** son las incorporaciones, modificaciones y eliminaciones necesarias en un producto software para cubrir la expansión o cambio en las necesidades del usuario.
- **Adaptativo:** son las modificaciones que afectan a los entornos en los que el sistema opera, por ejemplo, cambios de configuración del hardware, software de base, gestores de base de datos, comunicaciones, etc.
- **Correctivo:** son aquellos cambios precisos para corregir errores del producto software.

Cabe señalar que, de estos 4 tipos de mantenimiento, solamente el correctivo y el evolutivo entran en el ámbito de MÉTRICA versión 3, ya que los otros dos requieren actividades y perfiles distintos a los del proceso de desarrollo.

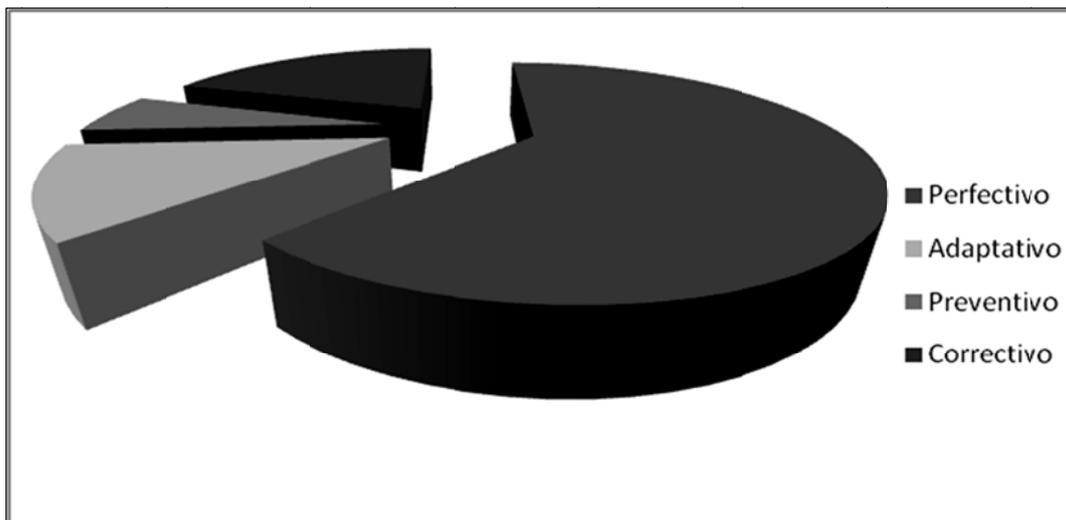


Figura 60. Costes relativos a cada tipo de mantenimiento

Según la definición anterior existen diferentes tipos de mantenimiento, como se puede apreciar en figura anterior, la cual representa los tipos de mantenimiento y los costes relativos.

Si dentro de los costes globales de mantenimiento hacemos un estudio de los costes asociados a cada tipo de mantenimiento, comprobamos que el mantenimiento correctivo no es la actividad que consume más recursos. **La mayoría de los estudios han identificado el mantenimiento perfectivo como la actividad dominante en los centros de proceso de datos.**

Si analizamos las actividades que deben realizar los programadores, podemos fijarnos en las siguientes:

- Estudiar las peticiones.
- Estudiar la documentación.
- Estudiar el código.
- Implementar el cambio.
- Realizar Pruebas.
- Actualizar la documentación del programa.

4.8.3 Distribución del Coste

Vamos a ver la importancia de la fase de mantenimiento. Los modelos del ciclo de vida tradicionales representan el mantenimiento como una fase que comienza una vez que se han finalizado las pruebas. Multitud de estudios indican que es la fase más costosa del ciclo de vida del software, lo que da lugar a que la mayor parte del presupuesto de los centros de proceso de datos (CPD) se destine a mantener los sistemas existentes. Algunos estudios ya señalaban, en 1987, que el 80% del presupuesto de los CPD se dedicaba a actividades de mantenimiento, y que en 1995 podía alcanzar el 95%. Algunas empresas han sobrepasado este porcentaje hasta llegar al límite de recursos (lo que se ha denominado barrera de mantenimiento), imposibilitando nuevos desarrollos. Por ello, **podemos asegurar que el mantenimiento es la fase dominante del ciclo de vida.**

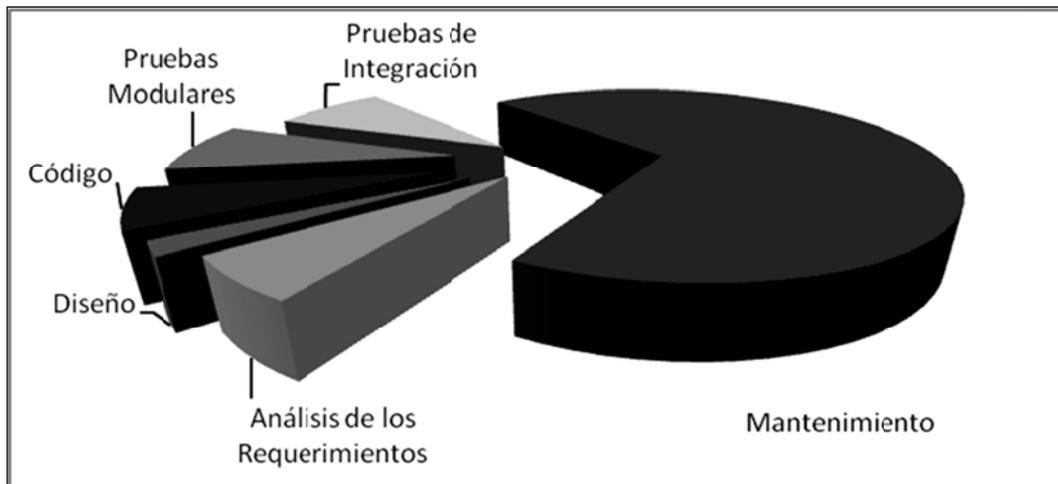


Figura 61. Costes relativos a cada fase

4.8.4 Factores que afectan al coste

Veamos algunos factores que afectan directamente a estos costes:

- **Inexistencia de los métodos, técnicas y herramientas** que puedan proporcionar una solución global de mantenimiento. Los ciclos de vida del software y, por lo tanto, las metodologías de desarrollo que los siguen, no reflejan la importancia del mantenimiento en términos de esfuerzo y coste necesarios ni de las actividades que hay que realizar durante esta fase. En la actualidad, prácticamente todos los métodos existentes se centran en el desarrollo de nuevos sistemas en vez de reparar o mejorar los sistemas existentes.
- La **complejidad de los sistemas** se incrementa paulatinamente por la realización de continuas modificaciones. Además, durante la vida de un sistema hay una pérdida de información, puesto que cada vez hay menos personas que conocen el software.
- **La documentación del sistema es defectuosa o inexistente.** En el caso de existir, muchas veces no se actualiza a medida que cambia el sistema. La programación es de baja calidad; muchas veces no es estructurada y/o no sigue ningún estilo estandarizado.
- **Se considera el mantenimiento como una actividad poco creativa;** a diferencia del desarrollo, y, por lo tanto, más sencilla y menos importante, que puede realizarse por personal con menor experiencia, menor soporte de herramientas y menor esfuerzo de gestión. No es raro descubrir que la primera actividad que realizan los nuevos programadores contratados por las empresas suele ser el mantenimiento de sistemas, a pesar del desconocimiento de los mismos.
- **Las actividades del mantenimiento se suelen realizar bajo presión de tiempo.** Normalmente los programadores (ya que el mantenimiento siempre se realiza sobre código) tienen poco tiempo para realizar las modificaciones, por lo que a veces no se

actualiza la documentación. Las correcciones imperfectas dan lugar a nuevos esfuerzos de corrección de futuro.

- **Poca participación del usuario durante el desarrollo del sistema.** Cuando se entrega el sistema al usuario, no satisface sus necesidades, lo que da lugar a un gran esfuerzo de mantenimiento posterior.



Efecto Iceberg

*Es importante tener en cuenta el **Efecto Iceberg**, es decir, en el momento que se hace el mantenimiento no se cuenta muchas veces con el factor económico (¿Cuánto dinero se invertirá en el mantenimiento?), y una vez se comienza a desarrollar la fase de mantenimiento en la aplicación, comienzan a surgir nuevos requerimientos, el efecto del iceberg (en la superficie se ve solo una parte de lo que realmente es su tamaño).*

4.8.5 Distribución del Tiempo

Las actividades pueden agruparse en funciones más genéricas, como:

- Comprensión del software y de los cambios que hay que realizar: un programador que quiera cambiar el software necesita comprenderlo (conocer el propósito, la estructura interna, los requisitos de operación...). En caso contrario, existe un gran riesgo de introducir nuevos defectos que, además, son los más costosos de reparar. Como veremos en la figura siguiente gran parte del tiempo se dedica al estudio para comprender el software, por lo que la utilización de cualquier herramienta que ayude a este propósito aumentará de forma considerable la productividad de los programadores.
- Modificación del software incorporando los cambios: implica la creación y modificación de estructuras de datos, lógica de proceso, interfaces, etc., así como la actualización de la documentación. Los programadores se deben asegurar de que los cambios no afectarán a la coherencia del programa y que no incrementarán su complejidad. En definitiva, deben conocer su impacto sobre el sistema para evitar posibles efectos laterales.
- Realización de pruebas para validar los cambios: se debe comprobar la corrección del software después de realizar cualquier cambio mediante la realización de pruebas selectivas. Pensar que un pequeño cambio no necesita la realización de pruebas puede acarrear graves problemas de calidad y de fiabilidad.

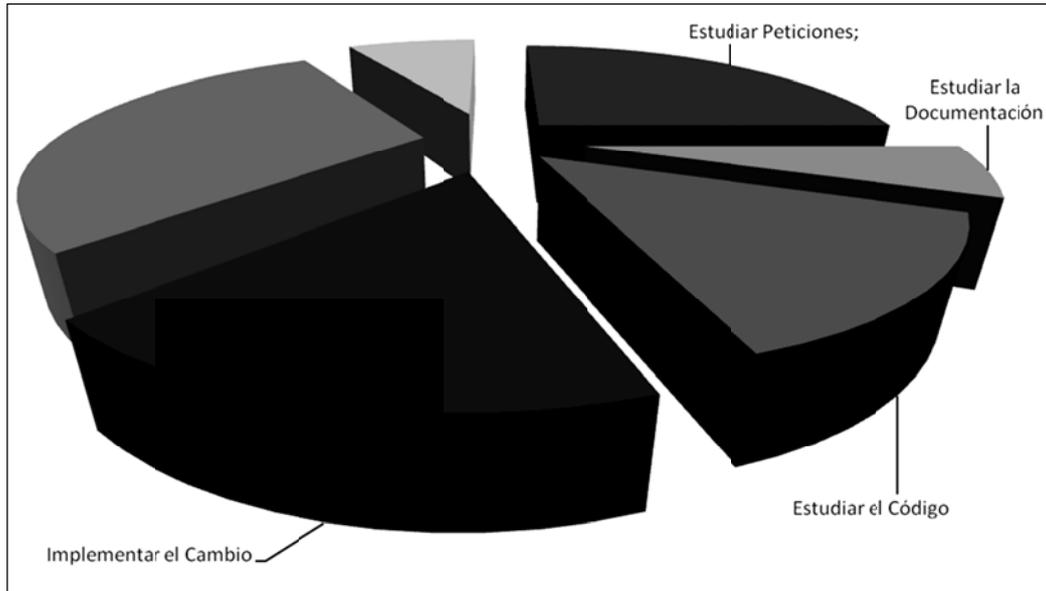


Figura 62. Distribución del tiempo de Mantenimiento

4.8.6 Tareas Usuales en la fase de Mantenimiento

- Implementar los cambios del sistema:
 - Utilizar los procedimientos de implementación de versiones, o
 - Implementar versiones de emergencia y después utilizar los procedimientos de versiones formales de forma retroactiva.
- Asegurarse de que el sistema continúa solucionando las necesidades de los usuarios:
 - Utilizar los acuerdos de niveles de soportes.
 - Revisiones regulares de requerimientos del nivel de acuerdo.
 - Revisiones regulares de cómo el sistema está alcanzando sus objetivos.
 - Llevar a cabo revisiones regulares del sistema.
 - Utilizar los procedimientos y contenido de las revisiones post instalación.

CAPÍTULO V. GESTIÓN DE RIESGOS

*“El significado de la vida no es la seguridad, las grandes oportunidades son arriesgadas.”
(Shirley Hufstедler)*

5.1 Descripción

La Gestión de los Riesgos del Proyecto incluye los procesos relacionados con llevar a cabo la planificación de la gestión, la identificación, el análisis, la planificación de respuesta a los riesgos, así como su monitoreo y control en el proyecto. Los **objetivos** de la Gestión de los Riesgos del Proyecto son aumentar la probabilidad y el impacto de eventos positivos, y disminuir la probabilidad y el impacto de eventos negativos en el proyecto.

Actividad	Descripción
Planificar la Gestión de Riesgos	Proceso por el cual se define cómo realizar las actividades de gestión de riesgos para un proyecto.
Identificar Riesgos	Proceso por el cual se determinan los riesgos que pueden afectar al proyecto y se documentan sus características.
Realizar Análisis de Riesgos	Proceso que consiste en priorizar los riesgos para realizar otros análisis o acciones posteriores, evaluando y combinando la probabilidad de ocurrencia y el impacto de dichos riesgos. El análisis deberá ser cualitativo en todos los casos, y en algunos casos también deberá ser cuantitativo.
Planificar la Respuesta a los Riesgos	Proceso por el cual se desarrollan opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.
Seguimiento y Control de los Riesgos	Proceso por el cual se implementan planes de respuesta a los riesgos, se rastrean los riesgos identificados, se monitorean los riesgos residuales, se identifican nuevos riesgos y se evalúa la efectividad del proceso contra riesgos a través del proyecto.

Figura 63. Actividades en la gestión de riesgos

Estos procesos interactúan entre sí y con los procesos de las otras áreas de conocimiento. Cada proceso puede implicar el esfuerzo de una o más personas, dependiendo de las necesidades del proyecto. Cada proceso se ejecuta por lo menos una vez en cada proyecto y en una o más fases del proyecto, en caso de que el mismo esté dividido en fases. Aunque los procesos se presentan aquí como elementos diferenciados con interfaces bien definidas, en la práctica se superponen e interactúan de formas que no se detallan aquí.



Riesgo

Un riesgo es un evento o condición incierta que, si sucede, tiene un efecto en por lo menos uno de los objetivos del proyecto. Los riesgos de un proyecto se ubican siempre en el futuro. Los objetivos pueden incluir el alcance, el cronograma, el coste y la calidad.

Un riesgo puede tener una o más causas y, si sucede, uno o más impactos. Una causa puede ser un requisito, un supuesto, una restricción o una condición que crea la posibilidad de consecuencias tanto negativas como positivas. Por ejemplo, la causa podría ser contar con una cantidad limitada de personal asignado para el análisis del proyecto. El evento de riesgo es que un analista funcional abandone el proyecto, y por consiguiente la cantidad limitada de personal disponible asignado al proyecto sea menor a la planificada, afectando a los tiempos destinados al análisis del proyecto.

Si alguno de estos eventos inciertos se produce, puede haber un impacto en el coste, el cronograma o en el desempeño del proyecto. Las condiciones de riesgo podrían incluir aspectos del entorno del proyecto o de la organización que pueden contribuir a poner en riesgo el proyecto, tales como prácticas deficientes de dirección de proyectos, la falta de sistemas de gestión integrados, la concurrencia de varios proyectos o la dependencia de participantes externos que no pueden ser controlados.

Los riesgos del proyecto tienen su origen en la incertidumbre que está presente en todos los proyectos. Los riesgos conocidos son aquéllos que han sido identificados y analizados, lo que hace posible planificar respuestas para tales riesgos. Los riesgos desconocidos específicos no pueden gestionarse de manera proactiva, lo que sugiere que el equipo del proyecto debe crear un plan de contingencia. Un riesgo del proyecto, que ha ocurrido, también puede considerarse un problema.

Existe un ambiente de **incertidumbre** cuando falta el conocimiento seguro y claro respecto del desenlace o consecuencias futuras de alguna acción o situación, lo que puede derivar en riesgo cuando se aprecia la perspectiva de una contingencia con posibilidad de generar pérdidas o la proximidad de un daño. La **incertidumbre supone cuantificar hechos mediante estimaciones para reducir riesgos futuros**, y aunque su estimación sea difícil no justificará su falta de información.

El riesgo supone un hecho externo, que puede acontecer o no en algún momento determinado. Por lo que el riesgo puede ser contemplado como elemento de incertidumbre que puede afectar a la actividad empresarial, pudiendo ser motivado por causas externas o internas a la empresa.

La incertidumbre varía para cada sujeto y para cada actividad a desarrollar. Esta diferencia cualitativa y cuantitativa de intensidad de la incertidumbre se encuentra relacionada con el grado de información e identificación del problema. La única forma de reducir el riesgo o al

menos sus consecuencias, se consigue mediante su identificación lo más clara posible, lo que permite poner en marcha todas aquellas acciones necesarias para intentar anularlos o minimizarlos con el uso de los conocimientos y de las técnicas que han servido para convertir, en algún grado, los riesgos en previsible. Por lo que riesgo también se puede definir como la valoración económica de la incertidumbre.



Las organizaciones y el riesgo

Las organizaciones perciben los riesgos como el efecto de la incertidumbre sobre los objetivos del proyecto y de la organización.

Las personas y los grupos adoptan actitudes frente al riesgo que influyen la forma en que responden a ellos. Estas actitudes frente al riesgo son motivadas por la percepción, las tolerancias y otras predisposiciones, que deben hacerse explícitas siempre que sea posible.

Debe desarrollarse un método coherente en materia de riesgos **para cada proyecto**, y la comunicación sobre el riesgo y su gestión debe ser abierta y honesta. Las respuestas a los riesgos reflejan el equilibrio percibido por una organización entre tomar y evitar los riesgos.



Tolerancia de riesgo

Las organizaciones y los interesados están dispuestos a aceptar diferentes niveles de riesgo. Los riesgos que constituyen una amenaza para el proyecto pueden aceptarse si se encuentran dentro de los límites de tolerancia y si están en equilibrio con el beneficio que puede obtenerse al tomarlos. Por ejemplo, la adopción de un cronograma de ejecución rápida es un riesgo que se corre para obtener el beneficio de una fecha de finalización más temprana.

Para tener éxito, la organización debe comprometerse a tratar la gestión de riesgos de una manera proactiva y consistente a lo largo del proyecto. Debe hacerse una elección consciente a todos los niveles de la organización para identificar activamente y perseguir una gestión eficaz durante la vida del proyecto. Los riesgos existen desde el momento en que se concibe un proyecto. Avanzar en un proyecto sin adoptar un enfoque proactivo en materia de gestión de riesgos aumenta el impacto que puede tener la materialización de un riesgo sobre el proyecto y que, potencialmente, podría conducir al fracaso.

5.2 Planificación de la Gestión de Riesgos

Planificar la Gestión de Riesgos es el proceso por el cual se define cómo realizar las actividades de gestión de riesgos para un proyecto. Una planificación cuidadosa y explícita

mejora la probabilidad de éxito de los otros procesos de gestión de riesgos. La planificación de los procesos de gestión de riesgos es importante para asegurar que el nivel, el tipo y la visibilidad de gestión de riesgos sean acordes tanto con los riesgos como con la importancia del proyecto para la organización.



Importancia de la planificación de la Gestión de Riesgos

La planificación de los procesos de gestión de riesgos es importante para asegurar que el nivel, el tipo y la visibilidad de gestión de riesgos sean acordes tanto con los riesgos como con la importancia del proyecto para la organización. La planificación también es importante para proporcionar los recursos y el tiempo suficientes para las actividades de gestión de riesgos y para establecer una base acordada para evaluar los riesgos.

El proceso Planificar la Gestión de Riesgos debe iniciarse tan pronto como se concibe el proyecto y debe completarse en las fases tempranas de planificación del mismo.

En la Planificación de la Gestión de Riesgos se establecen los criterios que serán utilizados para la Gestión de Riesgos y la forma en que serán llevados adelante los demás procesos de Gestión de Riesgos durante el proyecto.

La planificación de la Gestión del Riesgo incluye la definición de:

- Metodología.
- Responsables.
- Frecuencia.
- Categorías de Riesgos.
- Definición de probabilidad e impacto.
- Tolerancia de riesgos.

5.3 Identificación de Riesgos

Identificar los Riesgos es el proceso por el cual se determinan los riesgos que pueden afectar el proyecto y se documentan sus características. Entre las personas que participan en la identificación de riesgos se pueden incluir: el director del proyecto, los miembros del equipo del proyecto, el equipo de gestión de riesgos (si está asignado), clientes, expertos en la materia externos al equipo del proyecto, usuarios finales, otros directores del proyecto, interesados y expertos en gestión de riesgos. Si bien estas personas son a menudo participantes clave en la

identificación de riesgos, se debería fomentar la identificación de riesgos por parte de todo el personal del proyecto.

Identificar los Riesgos es un proceso **iterativo** debido a que se pueden descubrir nuevos riesgos o pueden evolucionar conforme el proyecto avanza a lo largo de su ciclo de vida. La frecuencia de iteración y quiénes participan en cada ciclo varía de una situación a otra. El formato de las declaraciones de riesgos debe ser consistente para asegurar la capacidad de comparar el efecto relativo de un evento de riesgo con otros eventos en el marco del proyecto. El proceso debe involucrar al equipo del proyecto de modo que pueda desarrollar y mantener un sentido de propiedad y responsabilidad por los riesgos y las acciones de respuesta asociadas. Los interesados externos al equipo del proyecto pueden proporcionar información objetiva adicional.

Para favorecer un proceso completo y sistemático de identificación de riesgos se sugiere el uso de una estructura de categoría de riesgos como las que se detalla en la figura 64:

Categoría	Descripción
De la Organización	<ul style="list-style-type: none"> • Dependencias del Proyecto • Recursos • Financiación • Priorización
Dirección de Proyectos	<ul style="list-style-type: none"> • Estimación • Planificación • Control • Comunicación
Técnico	<ul style="list-style-type: none"> • Requisitos • Tecnología • Complejidad e interfaces • Rendimiento y Fiabilidad • Calidad
Externo	<ul style="list-style-type: none"> • Subcontratistas y Proveedores • Regulatorio • Mercado • Cliente • Condiciones Climáticas

Figura 64. Estructura de categoría de riesgos

5.4 Análisis de Riesgos

Los riesgos pueden ser analizados de manera cualitativa o cuantitativa.

Realizar el Análisis Cualitativo de Riesgos es el proceso que consiste en priorizar los riesgos para realizar otros análisis o acciones posteriores, evaluando y combinando la probabilidad de ocurrencia y el impacto de dichos riesgos.

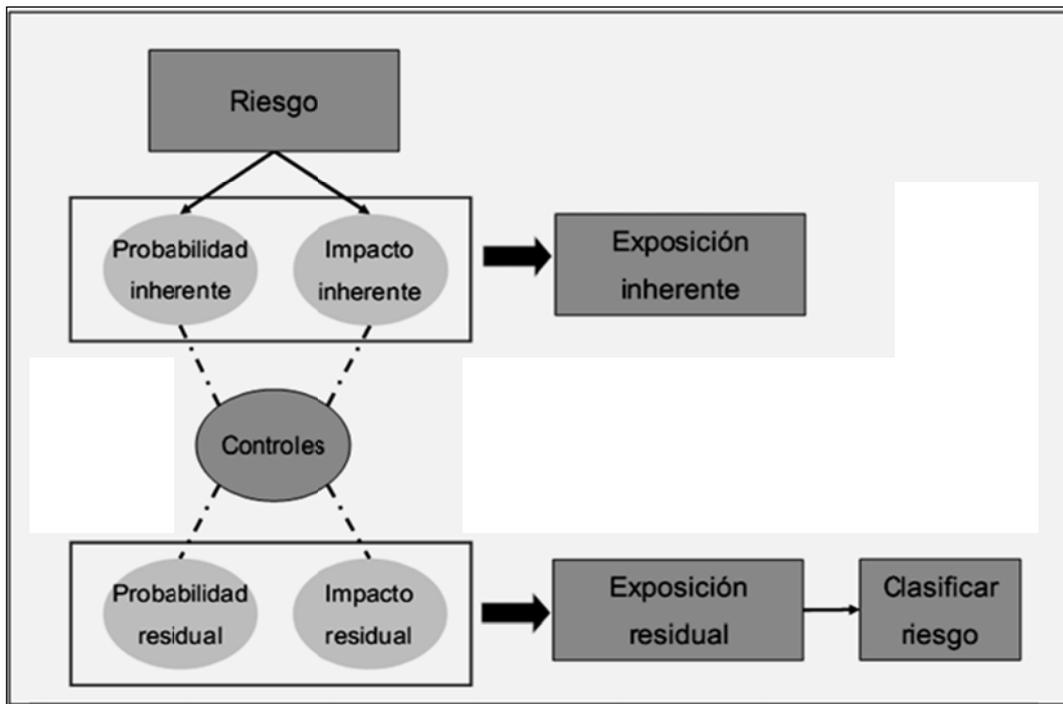


Figura 65. Análisis de Riesgos

Las organizaciones pueden mejorar el desempeño del proyecto concentrándose en los riesgos de alta prioridad. El proceso *Realizar el Análisis Cualitativo de Riesgos* evalúa la prioridad de los riesgos identificados usando la probabilidad relativa de ocurrencia, el impacto correspondiente sobre los objetivos del proyecto si los riesgos se presentan, así como otros factores, tales como el plazo de respuesta y la tolerancia al riesgo por parte de la organización asociados con las restricciones del proyecto en cuanto a costes, cronograma, alcance y calidad. Estas evaluaciones reflejan la actitud frente a los riesgos, tanto del equipo del proyecto como de otros interesados. Por lo tanto, una evaluación eficaz requiere la identificación explícita y la gestión de las actitudes frente al riesgo por parte de los participantes clave en el marco del proceso *Realizar el Análisis Cualitativo de Riesgos*. Cuando estas actitudes frente al riesgo introducen parcialidades en la evaluación de los riesgos identificados, debe ponerse atención en evaluar dicha parcialidad y en corregirla.

La definición de niveles de probabilidad e impacto puede reducir la influencia de parcialidades. La criticidad temporal de acciones relacionadas con riesgos puede magnificar la importancia de un riesgo. Una evaluación de la calidad de la información disponible sobre los riesgos del proyecto también ayuda a clarificar la evaluación de la importancia del riesgo para el proyecto.

		Exposición				
Probabilidad	5 - Muy Alta	2,6	3,2	3,8	4,4	5,0
	4 - Alta	2,2	2,8	3,4	4,0	4,6
	3 - Media	1,8	2,4	3,0	3,6	4,2
	2 - Baja	1,4	2,0	2,6	3,2	3,8
	1 - Muy baja	1,0	1,6	2,2	2,8	3,4
		1 - Muy bajo	2 - Bajo	3 - Medio	4 - Alto	5 - Muy alto
		Impacto				

Figura 66. Probabilidad de Impacto de Riesgo

En la figura anterior se ve claramente cómo se puede clasificar los riesgos según la probabilidad de ocurrencia e impacto en la organización. Un riesgo muy frecuente que tenga un impacto muy alto en la organización, seguramente será valuado con una puntuación alta (zona roja en la figura), lo que indica que deberá ser tratado de alguna manera.



Beneficios del Análisis Cualitativo de Riesgos

Realizar el Análisis Cualitativo de Riesgos es por lo general un medio rápido y económico de establecer prioridades para la planificación de la respuesta a los riesgos, y sienta las bases para realizar el análisis cuantitativo de riesgos, si se requiere. El proceso Realizar el Análisis Cualitativo de Riesgos debe ser revisado durante el ciclo de vida del proyecto para mantenerlo actualizado con respecto a los cambios en los riesgos del proyecto.

Realizar el Análisis Cualitativo de Riesgos puede conducir al proceso Realizar el Análisis Cuantitativo de Riesgos o directamente al proceso Planificar la Respuesta a los Riesgos.



Análisis Cuantitativo de Riesgos

Realizar el Análisis Cuantitativo de Riesgos es el proceso que consiste en analizar numéricamente el efecto de los riesgos identificados sobre los objetivos generales del proyecto.

5.5. Planificar la respuesta a los riesgos

Planificar la Respuesta a los Riesgos es el proceso por el cual se desarrollan opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.

Se realiza después del proceso Realizar el Análisis de Riesgos. Incluye la identificación y asignación de una persona (el “propietario de la respuesta a los riesgos”) para que asuma la responsabilidad de cada respuesta a los riesgos acordada y financiada. El proceso Planificar la Respuesta a los Riesgos aborda los riesgos en función de su prioridad, introduciendo recursos y actividades en el presupuesto, el cronograma y el plan para la dirección del proyecto, según se requiera.

Las respuestas a los riesgos planificadas deben adaptarse a la importancia del riesgo, ser rentables con relación al desafío por cumplir, realistas dentro del contexto del proyecto, acordadas por todas las partes involucradas y deben estar a cargo de una persona responsable.

También deben ser oportunas. A menudo, se requiere seleccionar la mejor respuesta a los riesgos de entre varias opciones. Los riesgos incluyen las amenazas y las oportunidades que pueden afectar el éxito del proyecto, y se debaten las respuestas para cada una de ellas.

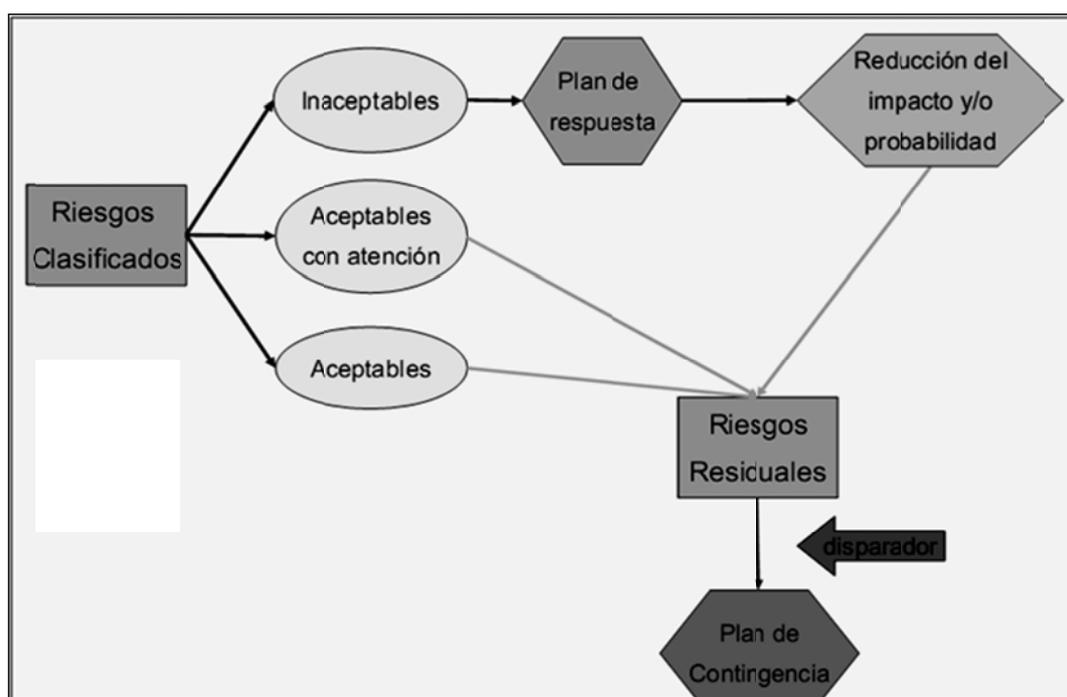


Figura 67. Planificación de la respuesta a los riesgos

Cada riesgo tendrá alguna de las siguientes respuestas:

- **EVITAR:** cambiar las acciones del proyecto para evitar el riesgo.
- **REDUCIR/MITIGAR:** minimizar la posibilidad de ocurrencia.
- **TRANSFERIR:** a un tercero. No lo elimina, solo es transferido.
- **ACEPTAR:** pasiva o activamente. En este caso es necesario un Plan de Contingencia.

La siguiente tabla muestra claramente cuáles son las respuestas que se deben adoptar según la clasificación de cada riesgo:

	PROBABILIDAD BAJA	PROBABILIDAD ALTA
IMPACTO BAJO	NO CONSIDERAR	ACEPTAR (CONTINGENCIA)
IMPACTO ALTO	RIESGOS ASEGURABLES (TRANSFERIR)	EVITAR MITIGAR

Figura 68. Clasificación de respuestas a un riesgo

5.6 Seguimiento y control de riesgos

Una vez identificados los riesgos del proyecto, es necesario realizar un seguimiento sobre estos, además de supervisar los riesgos residuales, identificar nuevos riesgos, ejecutar planes de respuesta a los riesgos, y evaluar su efectividad a lo largo del ciclo de vida del proyecto.

En la siguiente figura vemos como debemos realizar un seguimiento estricto a los Riesgos Clasificados en forma *Aceptables con atención*, y como debemos realizar un seguimiento a los Riesgos Clasificados en forma *Aceptables*. Para los riesgos *Inaceptables*, debemos tener un plan de respuesta, que debe ser evaluado previamente.

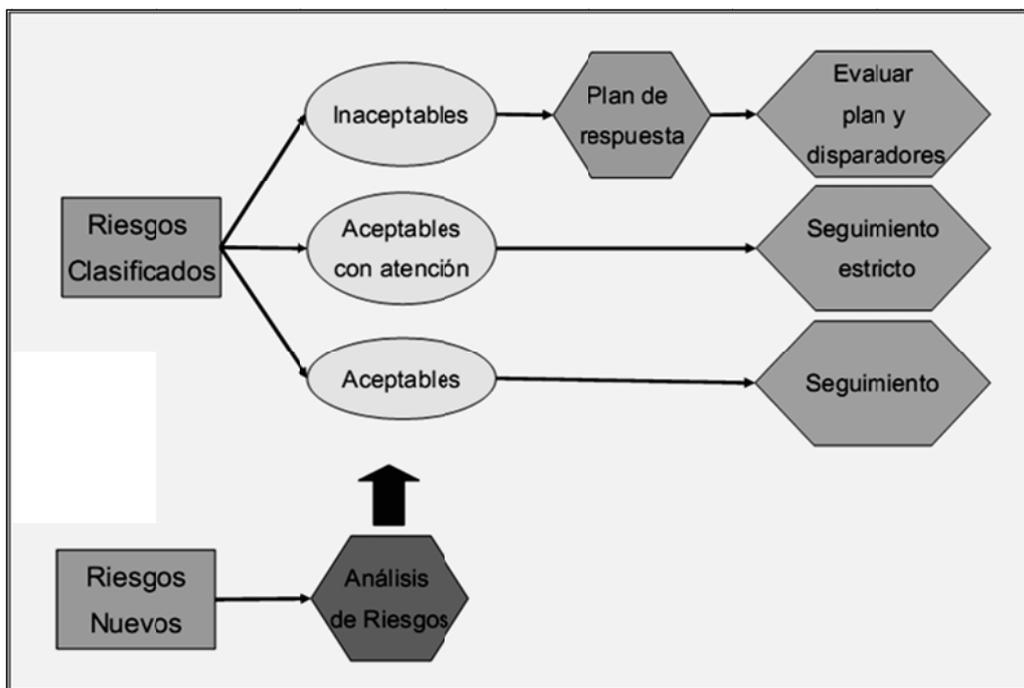


Figura 69. Seguimiento y control de riesgos

5.7 Beneficios de la Gestión de Riesgos

Una gestión de riesgos puede ser costosa de implantar al principio, pero si es realizada correctamente, trae importantes beneficios para el proyecto y el equipo de trabajo. A continuación se detallan los principales beneficios:

- Incrementa la probabilidad de éxito del proyecto.
- Provee una visión general de los riesgos, desafíos y problemas del proyecto.
- Reduce los costes del proyecto (planes de acción mantienen relación coste/beneficio adecuada).
- Reduce los tiempos del proyecto.
- Minimiza sorpresas y problemas.
- Evita que ocurran problemas, o si ocurren, evita su propagación.
- Genera ventaja competitiva.
- Aumenta rentabilidad.
- Incrementa el nivel de confianza de los interesados del proyecto, al conocer las debilidades y fortalezas.

En resumen se puede concluir que una buena gestión de riesgos es fundamental para que el proyecto tenga altas probabilidades de éxito.

CAPÍTULO VI. RECURSOS HUMANOS

6.1 Introducción

Según el PMBOK, la Gestión de Proyectos es una disciplina organizada en diferentes Áreas de Conocimiento, y una de ellas es la Gestión de los Recursos Humanos del Proyecto.

La Gestión de los Recursos Humanos del Proyecto incluye los procesos que organizan y dirigen el equipo del proyecto. El equipo del proyecto está compuesto por las personas a quienes se han asignado roles y responsabilidades para concluir el proyecto. Si bien es común hablar de la asignación de roles y responsabilidades, los miembros del equipo deberían participar en gran parte de la planificación y toma de decisiones del proyecto. La participación temprana de los miembros del equipo **aporta experiencia** durante el proceso de planificación y **fortalece** el compromiso con el proyecto como veremos.



Intereses comunes

Es muy importante identificar los intereses de cada miembro del equipo y determinar si se alinean con los intereses del proyecto. Si esto sucede, tendremos un equipo comprometido y motivado con los objetivos del proyecto.

El tipo y el número de miembros del equipo del proyecto a menudo pueden cambiar a medida que avanza el proyecto. Los miembros del equipo del proyecto pueden denominarse “personal del proyecto”. Los procesos de Gestión de los Recursos Humanos del Proyecto incluyen:

Planificación de los Recursos Humanos	• Identifica y documenta los roles del proyecto, las responsabilidades y las relaciones de informe, y también crea el plan de gestión del personal.
Adquirir el Equipo del Proyecto	• Obtiene los recursos humanos necesarios para completar el proyecto.
Desarrollar el Equipo del Proyecto	• Mejora las competencias y la interacción de los miembros del equipo para lograr un mejor rendimiento del proyecto.
Gestionar el Equipo del Proyecto	• Hace un seguimiento del rendimiento de los miembros del equipo, proporciona retroalimentación, resuelve polémicas y coordina cambios a fin de mejorar el rendimiento del proyecto.

En este capítulo se va a presentar un estudio diferente de los Recursos Humanos de un proyecto. Nos vamos a centrar primero en la figura del Director del Proyecto y vamos a demostrar la necesidad e importancia de un buen *Project Manager* en la actualidad. A continuación, vamos a tratar de ofrecerle al lector, personificado como Director de Proyectos presente o futuro, una guía para gestionar eficazmente su equipo de trabajo, sacando el mayor provecho del mismo haciendo uso de sus capacidades interpersonales a través de la Inteligencia Emocional. Presentaremos este concepto y su importancia en la gestión, y a continuación veremos cómo desarrollarlo y ponerlo en práctica para llegar a conseguir equipos de alto rendimiento, utilizando principios y técnicas a través de la Dirección, el Liderazgo y la Comunicación en el Proyecto.

Nuestro **objetivo final** será conseguir, para el conjunto de Recursos Humanos de nuestro proyecto, y de forma constante:

- Entusiasmo.
- Motivación.
- Compromiso.
- Y determinación para lograr y cumplir con las metas.

A lo largo del capítulo se podrán encontrar referencias a diferentes autores, los cuales son eminencias en las disciplinas que vamos a presentar, y que aquí sólo pretendemos introducir y descubrir para el lector. Los autores de esta obra recomendamos el estudio añadido de los trabajos de estos autores si se pretende profundizar en los mismos.

6.2 Perfiles más comunes en un Proyecto Informático

El objetivo de este apartado es presentar al lector una enumeración de los perfiles que aparecen a lo largo del desarrollo de un Proyecto Informático, como introducción previa a la figura del Director de Proyectos que después se tratará en detalle.

Aunque el número, estructura y funciones de los agentes participantes en el desarrollo de un sistema van a depender del tamaño, complejidad y características particulares de dicho sistema, así como del tipo del proyecto, pueden establecerse de forma general los perfiles de los integrantes siguientes, que pueden corresponder cada uno de ellos a distintas personas o a una misma (por ejemplo, en proyectos pequeños una misma persona puede hacer la función de analista y de programador).

En primer lugar, en todo desarrollo están implicados dos grupos de personas:

- Los pertenecientes a la organización usuaria o promotora del proyecto. Será la organización que use el sistema cuando éste esté construido.
- Las pertenecientes a la organización de desarrollo o empresa contratista. Es la organización encargada de la realización del sistema.

Existen dos particularidades destacables siempre que exista un proyecto:

- *Desarrollo interno.* Ocurre cuando la organización usuaria y la de desarrollo coinciden. En este caso, el departamento de Sistemas de Información o el departamento de Informática es el encargado de realizar un desarrollo para otro departamento o departamentos de su empresa.
- *Audidores externos.* Ocurre cuando la empresa promotora del proyecto y la contratista deciden de mutuo acuerdo contratar a una tercera firma, especialista en auditoría informática e independiente de ambas, que intervenga en aquellos casos en los que se hayan previsto procedimientos extraordinarios de control (auditorías).

Es muy probable que la composición del personal perteneciente a las organizaciones participantes (cliente, desarrolladora y auditora si la hubiera) sea la siguiente:

1. Comité de Dirección.

- Constituido por los responsables (directivos) de la organización usuaria o promotora del proyecto y de la organización u organizaciones encargadas de su desarrollo. Al frente de este comité se encontrará el Director del Proyecto.

2. Director del Proyecto o Project Manager (usados indistintamente en adelante).

3. Comité de Garantía de Calidad.

- Está integrado por personal con conocimientos y experiencia en metodologías de desarrollo del departamento de Informática de la organización usuaria o, en caso de que no existiera dicho departamento o así se quisiera, de una organización externa, independiente de la que desarrolla el sistema (empresa auditora). Este comité se encargará de controlar la evolución del proyecto a través del análisis de los productos generados a lo largo de las diferentes fases de desarrollo, determinando si el producto obtenido es el adecuado a los requerimientos especificados. Desarrolla su trabajo de forma independiente, realizando comprobaciones y análisis de los productos generados. En ningún caso el personal de este comité compaginará labores de dirección, desarrollo, etc. en el mismo proyecto, ya que perdería su independencia y no se podría asegurar la imparcialidad.

4. Equipo de Desarrollo Técnico.

- Formado por las personas que se van a encargar directamente del desarrollo. Dichas personas pertenecerán a la organización contratada para desarrollar el sistema, aunque si la organización promotora dispone de departamento de Informática y lo desea, podrían formar también parte de este equipo personal dicho departamento de la organización promotora. Al frente del equipo se sitúa el Jefe del Proyecto, siendo el resto de personal implicado en el desarrollo: *los Analistas, Diseñadores, Programadores y Técnicos de Sistemas.*

5. Personal del Área de Explotación.

- Se trata de personas del departamento de Informática de la organización usuaria con conocimientos sobre el entorno en que se explotará el sistema cuando se instale. Estas personas colaborarán con el equipo de desarrollo para concretar determinados aspectos, relacionados principalmente con la definición del entorno tecnológico del sistema (equipos, software de base, etc.) y con la construcción de la base de datos que posiblemente utilizará el sistema. En este último caso conviene destacar la conveniencia de la existencia de un Administrador de tal base de datos, que será una persona del área de explotación experta en esta labor y que se responsabilizará de la seguridad, confidencialidad y ajustes de dicha base de datos, de forma que se pueda asegurar un rendimiento óptimo cuando el sistema funcione en explotación.

6. Personal del Área de Usuario Final.

- Estará formado por personas que utilizarán el sistema una vez terminado. Se recurre a ellas para determinar con detalle las funciones que deberá realizar el sistema. Normalmente hay, al menos, dos categorías de usuarios:
 - Usuario responsable: encargado de fijar requisitos generales y de dar el visto bueno al software final y a los manuales.
 - Usuario final: en el que delegará el responsable para definir los requisitos detallados.

Vamos a ver ahora, de una forma más formal, la relación de participantes que define la metodología MÉTRICA versión 3, del Ministerio de Administraciones Públicas de España. Recordamos que esta metodología ha sido concebida para abarcar el desarrollo completo de Sistemas de Información sea cual sea su complejidad y magnitud, por lo cual su estructura y los perfiles de los participantes que intervienen deberán adaptarse y dimensionarse en cada momento de acuerdo a las características particulares de cada proyecto. Los perfiles establecidos se muestran en la siguiente tabla:

Perfil	Participantes
<i>Directivo</i>	<ul style="list-style-type: none"> ➤ Comité de Dirección. ➤ Comité de Seguimiento. ➤ Directores de usuarios. ➤ Usuarios Expertos.
<i>Jefe de Proyecto</i>	<ul style="list-style-type: none"> ➤ Jefe de Proyecto. ➤ Responsable de Implantación. ➤ Responsable de Mantenimiento. ➤ Responsable de Operación. ➤ Responsable de Sistemas. ➤ Responsable de Seguridad. ➤ Responsable de Calidad.
<i>Consultor</i>	<ul style="list-style-type: none"> ➤ Consultor. ➤ Consultor Informático. ➤ Consultor de las Tecnologías de la Información. ➤ Consultor de Sistemas de Información. ➤ Especialista en Comunicaciones. ➤ Técnico de Sistemas. ➤ Técnicos en Comunicaciones.
<i>Analista</i>	<ul style="list-style-type: none"> ➤ Analista. ➤ Administrador de Bases de Datos. ➤ Equipo de Arquitectura. ➤ Equipo de Formación. ➤ Equipo de Implantación. ➤ Equipo de Operación. ➤ Equipo de Seguridad. ➤ Equipo de Soporte Técnico. ➤ Equipo de Proyecto. ➤ Grupo de Aseguramiento de la Calidad.
<i>Programador</i>	<ul style="list-style-type: none"> ➤ Programador.

Figura 70. Participantes en cada perfil definido según Métrica V3

6.3 El Director del Proyecto

6.3.1 Definiciones

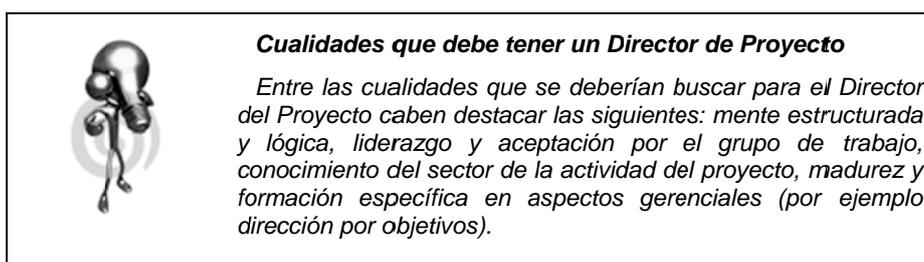
Se trata de un directivo de alto nivel, con amplia experiencia en **planificación y gestión** de Sistemas de Información, nombrado por lo general por acuerdo entre las organizaciones implicadas. Normalmente, se trata de un directivo de la organización encargada del desarrollo del sistema. Es el responsable de los **aspectos técnicos, contractuales, administrativos y**

financieros del proyecto. Dirige y supervisa la realización completa y correcta de las tareas asignadas.

Este responsable máximo del proyecto, debe ser una persona con amplios conocimientos informáticos y experiencia en el desarrollo del tipo de sistemas al que pertenezca el proyecto. Define y dirige las tareas ejecutadas por los miembros del equipo, incluyendo las fechas de terminación de tareas o subtareas. Proporciona guía y asistencia, y coordina que el producto final sea el adecuado.

Dependiendo de la magnitud de los desarrollos y de su escalabilidad, en nuestra carrera profesional podremos encontrar que el Director del Proyecto delegue responsables en áreas (calidad, seguridad, mantenimiento, etc.) según la necesidad del proyecto, o por el contrario, que el propio Jefe del Proyecto sea considerado y llamado también Director del Proyecto. No obstante, debemos recordar que se trata de perfiles diferentes con sus responsabilidades y actividades independientes.

Según MÉTRICA v3 (España, MAP, 2000), el Director del Proyecto realiza la estimación del esfuerzo necesario para llevar a cabo el proyecto, selecciona la estrategia de desarrollo, determina la estructura del mismo seleccionando los procesos principales de la metodología a emplear (como MÉTRICA), fija el calendario de hitos y entregas y establece la planificación del proyecto. Es el encargado de dirigir el proyecto, realizando las labores de seguimiento y control del mismo, revisión y evaluación de resultados y coordinación del equipo del proyecto. Tal y como indica también MÉTRICA v3, se ocupa también de la gestión y resolución de incidencias que puedan surgir durante el desarrollo del proyecto así como de la actualización de la planificación inicial. Entre sus funciones se encuentran la elaboración de los informes de seguimiento y el archivo de la documentación de gestión del proyecto una vez que este ha finalizado.

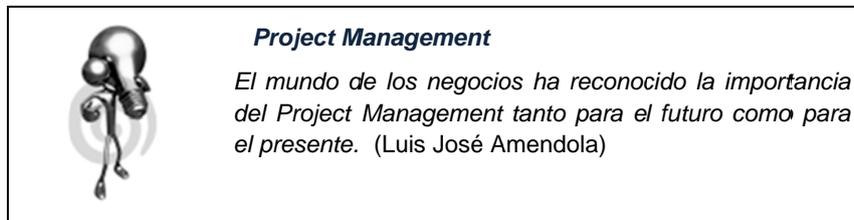


6.3.2 ¿Por qué un Project Manager?

El Dr. Engineering Management Luis José Amendola (2006), prácticamente logra dar una respuesta a esta pregunta cuando nos explica en su obra cómo entender la Dirección de Proyectos.

El Project Management, en el entorno industrial, aplica técnicas y herramientas de Project Management para gestionar los proyectos dentro de la estrategia del negocio y demostrar los logros que con estas se consiguen.

Un proyecto opera bajo plazos, costes, riesgo, calidad, y **factor humano**. Como observamos día a día, y como defiende Luis José Amendola, los proyectos son cada vez más grandes y más complejos. Hay gente que desde el papel del director del proyecto ha pasado de ser un experto técnico a un integrador de factores donde hoy en día juega un papel muy importante la conducta humana y el sentido común.



Por esta necesidad del factor humano, de un tiempo a esta parte, se viene demostrando un cambio en el estilo de dirección que ha venido predominando hasta cierto tiempo atrás. En la actualidad, si queremos una dirección de proyectos eficaz, está demostrado que el flujo de trabajo debe ser horizontal. Con una organización vertical tradicional y cadenas estrictas de mando, los trabajadores no tienen ninguna oportunidad de participar en la toma de decisiones ni en otras áreas funcionales de la empresa, lo que les impide sentirse más partícipes y motivados. Sin embargo, empleando una dirección horizontal, el trabajo se organiza a través de grupos funcionales de trabajo en el que existe la colaboración inherente de unos con otros, posibilitando además una mejora en cuanto a la coordinación y comunicación entre directivos y empleados. **El flujo de trabajo horizontal genera productividad, eficiencia y efectividad, además de por supuesto rentabilidad.**

¿Por qué resulta importante esto? Porque se necesita a una persona, un perfil nuevo, capaz de entender todas las unidades funcionales de la empresa y la interacción entre sí, para ser capaz de llevar adelante una **dirección eficaz** a través de un flujo de trabajo horizontal. Antiguamente, esta figura no existía, y los responsables de los proyectos sólo debían ocuparse de entender las operaciones de su unidad independiente.

Ahora nos encontramos con la necesidad del manejo de una gran cantidad de información y conocimiento. Podemos simplificar a la dirección de proyectos como conocimientos, gráficos e intuición, según la mayoría de los autores contrastados. Es primordial que tengamos clara la misión, el alcance, los objetivos y el reparto de cada proyecto antes de comenzar. El éxito del Project Manager requiere compromiso, dedicación, entrega y aprender día a día de las metodologías, técnicas y herramientas nuevas. Sólo así, desplegando los principios del Project Management, aseguraremos la igualdad y consistencia en la ejecución de los proyectos que llevemos a cabo. En el complejo y competitivo mundo en el que nos movemos, debemos buscar constantemente la excelencia como directores de proyectos, y como dice un viejo proverbio chino, la excelencia debe ser un hábito, no un acto.

El claro cambio necesario de la dirección vertical por la dirección horizontal, nos ofrece ya un buen argumento que justifica la existencia del Project Manager o Director del Proyecto en la empresa.

Dirección Vertical	Dirección Horizontal
<ul style="list-style-type: none"> ➤ El trabajo se divide en funciones, departamentos y tareas, y su realización es llevada a cabo por individuos. ➤ La unidad base de consecución de resultados es la persona y su trabajo. ➤ Cadenas estrictas de mando. ➤ Rigidez. ➤ Poca participación ➤ Menos responsabilidad. ➤ Menos comprensión del conjunto. ➤ Lentitud a la hora de enfrentar retos. ➤ El Director del Proyecto es el que asigna las tareas, mide, evalúa, controla y premia los resultados. 	<ul style="list-style-type: none"> ➤ Realización de tareas por equipos, ligando las actividades de los mismos a las necesidades de los clientes y los proveedores. ➤ Permite unidad de esfuerzos. ➤ El trabajo se organiza en torno a varios procesos de negocio o flujos de trabajo. ➤ Motivación. ➤ Mayor participación. ➤ Comunicación. ➤ Mayor eficiencia en el uso de los recursos. ➤ Mayor rentabilidad del proceso. ➤ Mediante el estilo de Dirección Horizontal, la responsabilidad de la gestión recae en los equipos, y el objetivo del directivo es la mejora continua de los resultados.

Figura 71. Comparativa entre Estilos de Dirección y sus características

Pero no es el único. Nuestra disciplina, la Industria Informática, está en constante evolución, se desarrolla y crece sin parar, y cada día nuevos conocimientos nos inundan. Esto nos lleva a lo siguiente: las Tecnologías de la Información y la Comunicación requieren del conocimiento experto de mucha gente; seguro que el lector está de acuerdo con esta idea. Aquí no es suficiente ni válido el genio de una persona trabajando sola, como pueda ser el caso de otra ciencia en la que un trabajador en armonía desarrolla su trabajo para cumplir un determinado objetivo. ¿A qué nos lleva esto? A la necesidad del trabajo en equipo. Y esta tendencia del trabajo en equipo para la gestión de la tecnología, con frecuencia, necesita un gestor, un gestor del proyecto, **un Project Manager**.

Por todo esto, y como decíamos al principio del libro, el número de empresas que usan la dirección y gestión de proyectos en los últimos años ha aumentado, especialmente en el sector de las TI. Ya no existe la despreocupación de poner al frente de los proyectos a alguien “que sepa” o a alguien “con experiencia en el desarrollo de proyectos”. Ahora, nuestra disciplina y las empresas requieren y exigen de directores de proyectos formados como tal, con numerosas habilidades, de todo tipo.

6.3.3 Habilidades y Características

En la nueva era de las TI, las **habilidades requeridas** para los Project Managers, establecidos ya como una profesión, son:

- Alta sensibilidad para los pequeños detalles.
- Capacidad de integración de factores.
- Generación de planes viables con éxito.
- Persuasión.
- Capacidad de influenciar.
- Capacidad de inspirar.
- Habilidad negociadora.

Otras características básicas deseables, de entre las que deben sobresalir:

- Capacidad de solucionar problemas, enfrentándose a ellos de cara.
- Capacidades de gestión empresarial.
- Competencias técnicas.
- Dotes comunicativas.
- Liderazgo.
- Desarrollado sentido común.
- Alta tolerancia hacia la ambigüedad.
- Buena comprensión del contexto.

6.3.4 Errores comunes de los Directores de Proyectos:

Es importante que también comprendamos los errores en los que podemos caer, ya que esto nos ayudará a centrar los esfuerzos y a evitar fallar en los proyectos.



Un líder debe saber claramente que NO debe hacer

Jim Collins, en el libro Empresas que Sobresalen, detalla que más importante que tener una lista de las cosas que hay que hacer, es tener una lista de las cosas que no hay que hacer.

El autor Gregory M. Horine (2010) ofrece la siguiente clasificación para los errores más comunes que comenten más frecuentemente los directores de proyectos:

1. No comprender exactamente los objetivos de la organización para con el proyecto ni asegurarse de que se cumplen esos objetivos.

2. No manejar bien las expectativas de los implicados a lo largo de la ejecución del proyecto.
3. No llegar a un acuerdo ni lograr la adscripción de los implicados principales en relación a los objetivos y los criterios del éxito del proyecto.
4. No desarrollar un calendario realista que incluya todos los esfuerzos, las interrelaciones de las tareas, las estimaciones completas y los recursos asignados por niveles.
5. No conseguir la aceptación y la adscripción al calendario del proyecto.
6. No decidir firmemente ni comunicar quién es el responsable de determinados asuntos.
7. No utilizar los procedimientos de control de cambios para gestionar el alcance del proyecto.
8. No comunicarse de forma eficiente y consistente con todos los implicados importantes.
9. No ejecutar el plan del proyecto.
10. No atajar a tiempo los riesgos que puede haber en el proyecto.
11. No identificar los riesgos a tiempo ni desarrollar planes de contingencia (respuestas) para esos riesgos.
12. No obtener los recursos adecuados con las características apropiadas en el momento oportuno.
13. No buscar la resolución de los problemas de forma contundente.
14. Mala definición de los requisitos y de la gestión.
15. Gestión insuficiente y falta de liderazgo del equipo del proyecto.



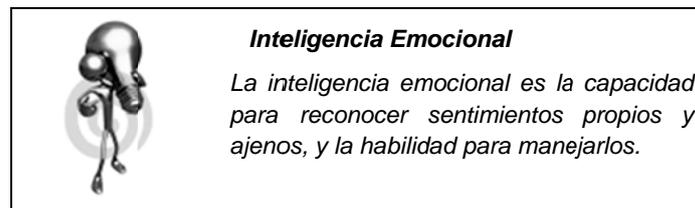
El aprendizaje en buenas prácticas

Ser conscientes y aprender buenas prácticas en la Dirección y Gestión de Proyectos Informáticos es fundamental, de ahí la importancia de textos como el que tiene delante el lector, para tratar de ayudarle en lo que podamos en el desarrollo de su carrera profesional.

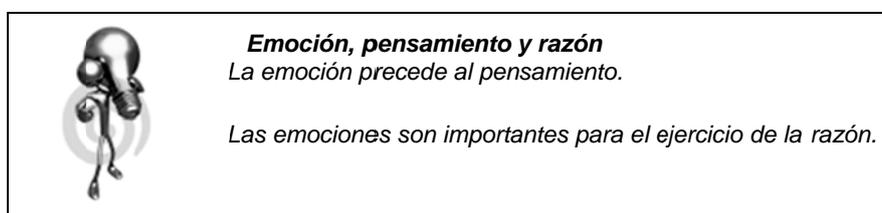
6.4 Inteligencia Emocional aplicada a la Dirección de Proyectos

6.4.1 Introducción

Inteligencia Emocional es una frase que se viene escuchando mucho en los últimos tiempos. Es como una respuesta global a muchos problemas relacionados al recurso humano. Pero, ¿qué es la Inteligencia Emocional?, ¿Por qué es importante para un Director de Proyectos? ¿Cómo sacarle partido para lograr equipos de alto rendimiento? Vamos a tratar de dar respuestas a estas y otras preguntas a lo largo de los siguientes apartados.



El considerado padre de este concepto es Daniel Goleman, que en 1995 publicó su prestigioso libro *Emotional Intelligence*. Sin embargo, otros autores en el pasado ya se habían referido a esta idea, a este concepto. Thorndike, en 1920, utilizó el término *inteligencia social* para referirse a la habilidad de **comprender y motivar** a otras personas. En 1983, Howard Gardner, en su Teoría de las Inteligencias Múltiples, también introdujo la idea de incluir tanto la inteligencia interpersonal (la capacidad para **comprender las intenciones, motivaciones y deseos de otras personas**) y la inteligencia intrapersonal (la capacidad de **comprenderse a uno mismo, apreciar los sentimientos, temores y motivaciones propias**). Otro de los orígenes de la inteligencia emocional está en Joseph Ledoux, como influencia más reciente, a partir de su libro *El cerebro emocional* (1996), en el que divulga sus hallazgos acerca de los circuitos neuronales del cerebro.



Daniel Goleman estima que la inteligencia emocional se puede organizar en cinco capacidades:

1. Conocer las emociones y sentimientos propios.
2. Manejarlos.

3. Reconocerlos.
4. Crear la propia motivación.
5. Y a partir de ahí, gestionar las relaciones, influyendo y manejando en los demás.



Importancia de aprender y desarrollar Inteligencia Emocional

¿Por qué es importante recibir formación en Inteligencia Emocional, y ponerla en práctica como Directores de Proyectos? Por la gran relevancia de las emociones en los resultados del trabajo.

Las **características** de la llamada inteligencia emocional son imprescindibles para un Director de Proyectos que es también un gestor de recursos humanos, y que responsable del rendimiento y de los resultados profesionales de los mismos.

Estas características son: la capacidad de motivarnos a nosotros mismos y a los demás, de perseverar en el empeño a pesar de las posibles frustraciones, de controlar los impulsos, de diferir las gratificaciones, de regular nuestros propios estados de ánimo, de evitar que la angustia interfiera con nuestras facultades racionales, y la **capacidad de empatizar y confiar en los demás, logrando que los demás también confíen en nosotros.**

El concepto de "Inteligencia Emocional" enfatiza el papel preponderante que ejercen las emociones dentro del funcionamiento psicológico de una persona cuando ésta se ve enfrentada a momentos difíciles y tareas importantes: los peligros, las pérdidas dolorosas, la persistencia hacia una meta a pesar de los fracasos, el enfrentar riesgos, o los conflictos con un compañero en el trabajo. En todas estas situaciones hay una involucración emocional que puede resultar en una acción que culmine de modo exitoso o bien interferir negativamente en el desempeño final. **Cada emoción ofrece una disposición definida a la acción, de manera que el repertorio emocional de la persona y su forma de operar, influirán decisivamente en el éxito o fracaso que obtenga en las tareas profesionales (y personales) que emprenda.**

Últimamente, se les ha dado a los factores emocionales la importancia debida en el tiempo y en el espacio, incluyéndolos en el óptimo desempeño de las actividades profesionales, donde las personas, como individuos, como gerentes y como líderes, tienen cada uno de ellos sus diferencias en muchos aspectos y áreas, pero que como seres humanos están dentro de los Principios de la Inteligencia Emocional.

“Las condiciones intelectuales no son la única garantía de éxito en el ámbito profesional, sino tan sólo un factor, que unido a las necesidades emocionales del personal cubiertas como equipo, desarrollará el desempeño y los resultados de todo líder y trabajador motivándolo emocionalmente a ser productivo.” (Daniel Goleman).

Numerosos autores, y también Goleman, nos hablan de las *Competencias Emocionales*. Estos autores definen el éxito de gerentes líderes y trabajadores, en personas de alto nivel de desempeño, con destrezas, habilidades técnicas y emocionales bien desarrolladas; que han alcanzando la capacidad de dar sentimientos que cada vez se hacen más competitivos y necesarios en la familia, en la sociedad, y en el trabajo (gerencia).

Las competencias emocionales que más se repitieron como decisivas en el éxito de los líderes y sus empresas, fueron clasificadas en cuatro categorías:

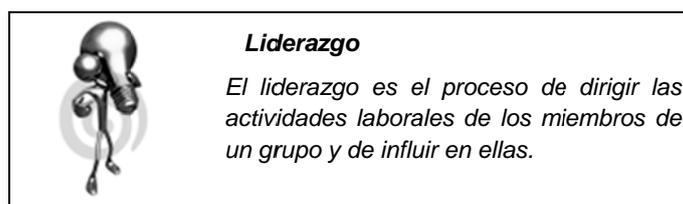
CATEGORÍA	DESCRIPCIÓN	COMPETENCIAS
Autoconciencia	Se trata de la habilidad para reconocer y comprender los propios estados emocionales, sentimientos, rasgos, así como su efecto en las demás personas.	Las competencias que se miden y desarrollan en esta categoría son: <ul style="list-style-type: none"> ✓ Autoconfianza. ✓ Capacidad para despertar estados emocionales alegres.
Autorregulación	Es la habilidad para controlar y redireccionar impulsos y estados emocionales negativos, unido a la capacidad para suspender juicios y pensar antes de actuar.	<ul style="list-style-type: none"> ✓ Autocontrol. ✓ Confiabilidad. ✓ Conciencia. ✓ Adaptabilidad. ✓ Orientación a resultados. ✓ Iniciativa.
Empatía	Se trata de las habilidades para sentir y palpar las necesidades de otros y de la propia organización, unida a la apertura para servir y cubrir las inquietudes de quienes le rodean.	<ul style="list-style-type: none"> ✓ La empatía. ✓ La conciencia organizacional. ✓ Orientación al servicio.
Socialización	Engloba el dominio de estrategias y formas de relacionarse afectiva y efectivamente con las demás personas, creando redes de relaciones, construyendo climas agradables, abiertos, y efectivos en sus conversaciones.	<ul style="list-style-type: none"> ✓ Desarrollo de la persona. ✓ Liderazgo. ✓ Influencia. ✓ Comunicación. ✓ Gerencia del cambio. ✓ Manejo de Conflictos. ✓ Construcción de redes. ✓ Cooperación en equipo.

Figura 72. Clasificación de Competencias Emocionales

6.4.2 Liderazgo

Vamos a ver en primer lugar varias definiciones sobre *Liderazgo*.

Según *Wikipedia*, el liderazgo es el **conjunto de capacidades** que una persona tiene para **influir** en un grupo de personas determinado, haciendo que este equipo **trabaje con entusiasmo en el logro de metas y objetivos**. También se entiende como la capacidad de tomar la iniciativa, gestionar, convocar, promover, incentivar, motivar y evaluar a un grupo o equipo. En la administración de empresas y proyectos, el liderazgo es el ejercicio de la actividad ejecutiva en un proyecto, de forma eficaz y eficiente, sea éste personal, gerencial o institucional (dentro del proceso administrativo de la organización).

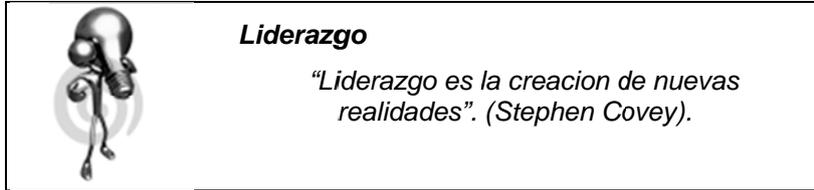


Esta última definición tiene cuatro implicaciones importantes. Implica que haya una persona (líder o no) que pueda influir y motivar a los demás (seguidores). De ahí que en los estudios sobre liderazgo se haga énfasis en la capacidad de persuasión e influencia. Tradicionalmente, a la suma de estas dos variables se le ha denominado **carisma**. Sin embargo, los estudios actuales en psicología y sociología, han concluido que el carisma no tiene la importancia que históricamente se le había otorgado y que también hay otros factores (**habilidades directivas**) que son más determinantes a la hora de construir el verdadero liderazgo. El autor Stephen Covey define "*Liderazgo es la creación de nuevas realidades*".

Luis José Amendola (2006), nos aporta también una definición de Liderazgo en su genial obra "*Estrategias y Tácticas en la Dirección y Gestión de Proyectos*", y después nos alienta a llegar al concepto de **COACHING**, ampliamente difundido por los estudiosos del Liderazgo. Para el Dr. Amendola, el rol del liderazgo trabaja constantemente en la expansión de su rol, e incluye actividades tales como:

- Ser un modelo para los demás.
- Amplitud de mente a la hora de comprender y analizar el negocio.
- Superar las barreras del proyecto y anular las posibles interferencias.
- Agente facilitador de la comunicación dentro del equipo.
- Orientación del equipo al cliente.
- Motivar para la excelencia.

Según este autor, un líder efectivo, que es lo que buscamos, tiene muy clara la dirección de su organización, y está enfocado a alcanzar su visión, entiende lo que esto significa para cada persona individualmente, así como la necesidad de modelar esa necesidad y enfocarla.

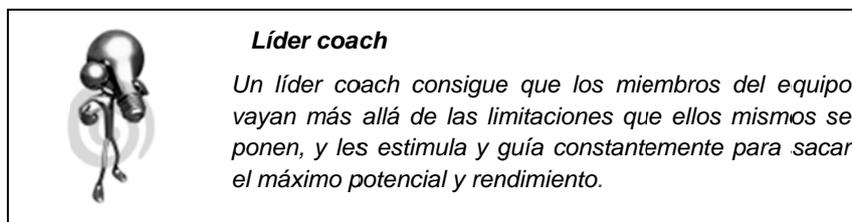


Tal y como muy bien dice Amendola, tradicionalmente, **el eslabón de competencia más débil del directivo de proyectos siempre ha sido la gestión del rendimiento de sus colaboradores.**

La correcta formación técnica contrasta con la falta de habilidades emocionales y por tanto relacionales. El coaching, defiende Amendola, mediante una metodología estructurada, lleva a cabo aproximaciones que nos permiten trabajar en la mejora del rendimiento y en el desarrollo del potencial del equipo del proyecto.

El coaching es por tanto, como dice el autor que venimos referenciando, **“un modelo de liderazgo que tiene la finalidad de desarrollar el potencial de las personas”**, y que se apoya en los siguientes principios:

- Hay que centrarse en el rendimiento futuro, y no en el actual o pasado. Sólo en las posibilidades futuras.
- Hay que creer en los miembros del equipo del proyecto. Si nosotros creemos en ellos, está demostrado que esto tendrá un impacto muy positivo en su desempeño. Por lo que obtendremos lo mejor de las personas.
- La confianza es la gran apuesta del coaching.
- Los miembros del equipo no aprenden del líder, aprenden de ellos mismos y de la experiencia.



El coaching está siendo aplicado por numerosas organizaciones en los últimos tiempos, y se está empezando a aceptar como una ventaja competitiva. Tanto es así, que algunos autores,

como Hendricks (1996), ya se atreven a citar las características más importantes de este modelo de liderazgo:

1. Claridad en la comunicación.
2. Apoyo integral al equipo.
3. Confianza y reconocimiento al equipo.
4. Visión clara de las metas comunes.
5. Empatía.
6. Control del riesgo y seguridad de los miembros del equipo.
7. Paciencia.
8. Confidencialidad.
9. Y respeto.

Volviendo al concepto puro de Liderazgo, en la literatura podemos encontrar gran variedad de clasificaciones de tipos de líderes. Nos vamos a quedar con la siguiente clasificación de **Estilos de Liderazgo** y sus características, sobre la que trabajaremos después:

- **Líder Visionario.**
 - Empatía.
 - Confianza en sí mismo.
 - Agente de cambio.
- **Líder Afiliativo.**
 - Empatía.
 - Creación de relaciones.
 - Resolución de conflictos.
- **Líder Democrático.**
 - Fomento del trabajo en equipo.
 - Gran comunicador.
- **Líder Orientativo.**
 - Empatía.
 - Dominio del potencial de los demás.
 - Autoconsciente.
- **Líder Coactivo.**

- Confianza en sí mismo.
- Ordena a los demás que ejecuten sus deseos.
- Carece de empatía.
- **Líder que Marca La pauta.**
 - Implanta calidad.
 - Iniciativa.
 - Motivación.
 - No ayuda a mejorar, crítica a los que no logran sus objetivos.

Dice Gregory M. Horine (2010), que en la Capacidad de Liderazgo se incluyen capacidades básicas como las habilidades interpersonales y generales, como **forma de tratar a la gente, la adaptabilidad, la flexibilidad, la gestión de personas**, el grado de orientación al cliente, capacidades analíticas, de resolución de problemas, y el talento de tener siempre en mente la perspectiva general.

Esto acerca la Inteligencia Emocional, de la que venimos hablando, al Liderazgo. Y estas características nos llevan también a un importante concepto: el clima de trabajo, lo que se conoce como **clima organizativo** de la organización.

El clima organizativo tiene una importancia predominante en el cómo desempeñan su trabajo los miembros del equipo, y refleja el sentido de las personas acerca de su capacidad. Necesitamos un clima que favorezca el entusiasmo, la motivación, el compromiso, y la determinación. Existen indicadores del clima, e incluyen el grado en el que la comunicación es clara, la flexibilidad con la que cuentan los empleados para realizar sus trabajos como mencionaba Horine, la capacidad para innovar, y el sentido de la responsabilidad para llevar a cabo las tareas.

Daniel Goleman y Cary Cherniss, en su obra *The Emotionally Intelligent Workplace (2005)*, defienden la importancia del liderazgo, y justifican la importancia de un liderazgo emocionalmente inteligente para crear un clima de trabajo próspero que derive en una consecución de los objetivos a través del rendimiento de todos los empleados, y nos brindan una serie de argumentos muy útiles para nosotros, Directores de Proyectos:

- En su obra, ponen de manifiesto el resultado de estudios que revelan que los líderes emocionalmente inteligentes, con grandes competencias emocionales, sobrepasan los objetivos en términos de rendimiento, entre en un 15 y un 20 %.
- “La relación entre puntos fuertes en Inteligencia Emocional en un líder, y el rendimiento de la unidad, parecer estar mediatizado por el clima creado por el líder.”

Sus estudios, han revelado la importancia del clima en la relación con la inteligencia emocional de los individuos que ocupan puestos de Dirección de Proyectos, y esto ha conducido a reconocer el destacado papel de las competencias en Inteligencia Emocional en la eficacia del liderazgo. La siguiente tabla, elaborada por Goleman y Cherniss, recoge los efectos de esta relación:

Estilo de Liderazgo	Competencias en IE	Impacto sobre el clima	Objetivo	Cuándo es apropiado
Visionario	Confianza en uno mismo, empatía, catalizar los cambios, liderazgo, motivación de logro	Enormemente positivo	Movilizar a otros para seguir una visión.	Cuando los cambios requieren una nueva visión, o cuando es necesaria una dirección clara.
Afiliativo	Empatía, establecer vínculos, resolución de conflictos.	Muy positivo.	Crear armonía.	Para cerrar heridas en un equipo, o para motivar en época de estrés.
Democrático	Trabajo en equipo y colaboración, comunicación.	Muy positivo.	Lograr compromiso a través de la participación.	Para conseguir participación o consenso, o para obtener valiosas aportaciones por parte de los empleados.
Orientativo	Desarrollar a los demás, empatía, autoconciencia emocional.	Muy positivo	Acumular fuerza para el futuro.	Para ayudar a un empleado a mejorar el rendimiento, o a desarrollar puntos fuertes a largo plazo.
Coactivo	Motivación de logro, iniciativa, autocontrol emocional.	Altamente negativo.	Acatamiento inmediato.	En una crisis, para iniciar un giro, o con empleados problemáticos.
Marcar la pauta	Meticulosidad, motivación de logro, iniciativa.	Muy negativo.	Ejecutar tareas a muy alto nivel.	Para obtener resultados rápidos de un equipo muy motivado y competente.

Figura 73. Estilo de Liderazgo, Inteligencia Emocional (IE) y efectividad organizativa, de Goleman y Cherniss (2005)

6.4.3 La Comunicación en el proyecto

En un proyecto participan un grupo de personas que interactúan entre sí en busca de un objetivo común y con recursos limitados. Esta interacción se realiza por medio de la comunicación. Por tal motivo, se considera esencial agregar contenidos referidos a la comunicación en el Proyecto.

Las comunicaciones del proyecto son todos los medios y formas de que un proyecto interactúe con todos sus implicados. Esto no sólo incluye los elementos de comunicaciones

formales y estándar, sino que también incluye comunicaciones de gestión del cambio organizacional.



Importancia de la comunicación según PMI

Según el PMI, los directores de proyectos deben pasar el 90% de su tiempo comunicándose.

6.4.3.1 La importancia de la Gestión de las Comunicaciones en el Proyecto

Nos encontramos ante una nueva Área de Conocimiento de la Gestión de Proyectos. Incluye los procesos requeridos para asegurar la generación, recopilación, distribución, almacenamiento, recuperación y disposición final oportuna y apropiada **de la información del proyecto**. Los procesos de Gestión de las Comunicaciones del Proyecto proporcionan los enlaces cruciales entre las personas y la información, que son necesarios para que las comunicaciones sean exitosas. Los directores del proyecto pueden dedicar una cantidad de tiempo excesiva a la comunicación con el equipo del proyecto, los interesados, el cliente y el patrocinador. Todas las personas involucradas en el proyecto deben comprender cómo afectan las comunicaciones al proyecto en su conjunto. Los procesos de Gestión de las Comunicaciones del Proyecto incluyen:

Planificación de las Comunicaciones	<ul style="list-style-type: none"> • Determina las necesidades de información y comunicación de los interesados en el proyecto.
Distribución de la Información	<ul style="list-style-type: none"> • Hace que la información necesaria esté disponible para las personas interesadas en el proyecto en el momento oportuno.
Informar del Rendimiento	<ul style="list-style-type: none"> • Recopila y distribuye información sobre el rendimiento, incluido el informe de estado de la situación, la medición del avance y las proyecciones.
Gestionar a los Interesados	<ul style="list-style-type: none"> • Gestiona las comunicaciones a fin de satisfacer los requisitos de los interesados en el proyecto y resolver polémicas con ellos.

Las comunicaciones del proyecto son importantes no sólo por una razón obvia, que es la de mantener adecuadamente y constantemente informados a los miembros del estado y progreso

del proyecto, sino que también son un factor determinante para el éxito general del mismo.
¿Por qué?

- Porque la calidad y la efectividad de las comunicaciones tendrá un enorme impacto sobre las percepciones de los implicados en relación al proyecto y al papel del director del proyecto de líder del mismo.
- Porque la capacidad del director del proyecto para comunicar, es el factor más sobresaliente que afecta al nivel de gestión y dirección del núcleo del equipo del proyecto.
- Porque ya hay suficientes problemas en la ejecución del proyecto, como para añadir además conflictos como resultado de las malas percepciones, la falta de información o los problemas no existentes, y todo esto consecuencia de una mala comunicación.



Organización y comunicación

Cualquier director de proyectos con experiencia, sabe que hay dos competencias que le salvarán casi siempre: la organización y la comunicación. Si se sobresale en estos campos, especialmente en las comunicaciones del proyecto, se compensarán las flaquezas en los demás.

6.4.3.2 Competencias interpersonales

Las siguientes técnicas son probablemente las más importantes porque afectan a la calidad de todas las comunicaciones del proyecto, aquellas formales y las más frecuentes que ocurren en el día a día del proyecto entre el equipo del proyecto y los implicados. La siguiente lista muestra las competencias interpersonales claves que los buenos comunicadores poseen o deben esforzarse en alcanzar:

- Escuchar con un propósito constructivo.
- Humildad; ser humilde y transmitir esa humildad.
- Pensar con calma antes de responder, y nunca hacerlo en caliente.
- Ponerse en el lugar del otro, y si es posible, que el otro se dé cuenta de ello.
- No juzgar, y evitar términos y tonos que impliquen juicio, culpa o malas acciones de otras partes.
- Interesarse por los demás.
- Intentar comprender lo que hacen, por qué lo hacen y qué problemas están pasando.
- Validar las percepciones antes de responder.
- Mostrar apreciación por su tiempo y el esfuerzo de comprensión.
- Resumir lo que ha dicho el hablante.

- Hacer que la gente se sienta escuchada.
- Centrarse en construir relaciones.
- Tener bajo control las emociones propias.
- No asumir que una respuesta negativa está causada por temas personales, la mayor parte de las veces no será así.
- Siempre que sea posible, evitar interrumpir.
- Asegurarnos de que estamos siendo comprendidos.

6.4.3.3 La dificultad de comunicarse

Barreras de la comunicación efectiva:

- Filtración.
- Percepción selectiva.
- Emociones.
- Lenguaje.
- Cultura nacional.

Buenas prácticas para lograr la superación de las barreras:

- Emplear retroalimentación.
 - Hacer énfasis en comportamientos específicos.
 - Mantener la retroalimentación impersonal.
 - Mantener la retroalimentación orientada a las metas.
 - Dar un tiempo oportuno a la retroalimentación.
 - Asegurarse de que lo entiendan.
 - Dirigir la retroalimentación negativa hacia un comportamiento que quien lo recibe pueda controlar.
- Simplificar el lenguaje.
- Escuchar activamente.
 - Empatía.
 - Aceptación.
 - Intensidad.
 - Disposición de asumir la responsabilidad de escuchar el mensaje completo.

- Establecer contacto visual.
 - Asentir con la cabeza y mostrar una expresión facial adecuada.
 - Evitar acciones o gestos que los distraigan.
 - Hacer preguntas.
 - Parafrasear.
 - Evitar interrumpir al orador.
 - No hablar de más.
 - Hacer transiciones suaves entre los papeles del orador y escucha.
- Restringir las emociones.
 - Vigilar los indicativos no verbales.

6.4.3.4 Manejo de conflictos en la comunicación

Los conflictos son las diferencias incompatibles percibidas que dan como resultado interferencia u oposición. Un conflicto que no pudo ser resuelto a tiempo puede costar el fracaso del proyecto. Por tal motivo, es muy importante que el Director del Proyecto tenga los medios necesarios para identificar conflictos, y habilidad para resolverlos a tiempo.

Prácticas efectivas para la resolución de conflictos:

- Establecer un estilo para manejar el conflicto.
- Tener cuidado al seleccionar los conflictos que se quieren manejar.
- Evaluar a los participantes en el conflicto.
- Evaluar la fuente del conflicto.
- Conocer las opciones.

6.4.4 Potenciar el rendimiento del equipo del proyecto

Una dirección eficaz, y contar con buenas competencias de comunicación, son ingredientes claves para conseguir el éxito de un proyecto. Pero no es suficiente, debemos ir más allá en la búsqueda de la excelencia empresarial, y en busca de nuestras aptitudes y capacidades como líderes y buenos gestores de recursos humanos. Para ello, debemos comprender los principios y técnicas específicas que podemos aplicar para potenciar al máximo el rendimiento del equipo de un proyecto, y de esto nos vamos a ocupar a continuación.

6.4.4.1 Principios de Gestión

Ahora ya conocemos qué es un equipo de alto rendimiento, qué le caracteriza. El siguiente paso tiene que ser lograr el máximo rendimiento de nuestro equipo de trabajo, llegar a alcanzar ese nivel. Y para conseguirlo, conviene revisar varios principios fundamentales de gestión,

primordiales para guiar a nuestro equipo, que también nos ofrece el Sr. Horine en la obra indicada como fuente de este subapartado:

1. **Adaptar el estilo de gestión;** dependiendo de la fase del proyecto, las necesidades particulares del equipo y el entorno del proyecto.
2. **Reclutar a los más preparados.** Para un director de proyectos, tener a la gente adecuada representa el 80% de la batalla. En este sentido, siempre que sea posible, debe ser el propio director de proyectos el que se encargue de la selección del personal, y en la misma medida, será muy importante conseguir la participación de personas con una carrera plagada de éxitos, y con grandes competencias para aportar en el proyecto.
3. **Planificar como equipo.** Este es un factor fundamental, ya que si conseguimos implicar e involucrar a todos los miembros en la planificación del desarrollo, la planificación pasa a ser “su plan”, y “su” calendario. Con esto logramos un mayor compromiso, aceptación y un mayor nivel de responsabilidad. Como vemos, poco a poco vamos obteniendo, siguiendo estos principios, las características de lo que sin duda será un equipo eficaz, un equipo de alto rendimiento.
4. **Proteger al equipo y mantenerlo centrado.** El director del proyecto debe ser quien proteja al equipo de las políticas, ruido y otros factores que puedan distraerlo y ralentizar su progreso; y debe asegurarse también de que los miembros del equipo se mantengan centrados durante todo el tiempo, sin perder nunca de vista el marco general del proyecto: misión, objetivos y prioridades.
5. **Potenciar la productividad.** Tal y como defienden todos los autores del mundo, pocas cosas hay más importantes que asegurarse de que cada miembro del equipo tenga claro qué es lo que tiene que hacer. De cara a la productividad, la importancia es máxima.
6. **Mejorar las competencias comerciales.** Una meta fundamental que pretendemos obtener con cualquier persona del equipo es mejorar sus cualidades comerciales mediante sus experiencias en el proyecto. Debemos buscar formas de mejorar las cualidades de nuestros miembros, y ayudarles a crecer como profesionales.
7. **Aplicar los talentos individuales.** Consta de:
 - a. Buscar y encontrar los talentos de cada persona, los que se pueden aplicar al proyecto.
 - b. Entender también sus flaquezas.
 - c. Analizar las emociones y comprender qué es lo que lleva a moverse a cada persona, sus factores de motivación, lo que les importa. Una vez conocido esto, será posible no sólo posicionar mejor a la gente, sino que nos permitirá

recompensar y reconocer mejor a los individuos. Con el tiempo, esto se convertirá en un hábito de vida.

- d. Alinear las funciones y responsabilidades con el punto fuerte de cada miembro del equipo en la medida que sea posible. El “punto fuerte” es la combinación de los talentos naturales y las motivaciones personales.

8. **Reconocer y recompensar.** Formas de realizarlo, no de conseguirlo:

- a. El director del proyecto debe hacer como si fuese el representante o relaciones públicas de cada miembro del equipo. No sólo debemos ofrecer respuestas oportunas, debemos ofrecer apreciación por cada miembro personalmente. Y es más efectivo si se realiza durante todo el desarrollo del proyecto, según avanza, y no solamente como revisión anual o final del proyecto. Necesitamos y debemos mantener vivo el compromiso, la motivación y el espíritu profesional de los miembros.
- b. Celebrar los éxitos del equipo. Esto conseguirá un gran impulso.
- c. Recompensa.

9. **Cohesionar al equipo.** Todo lo visto anteriormente tiene sentido si facilitamos y conseguimos la sinergia del equipo. La mayor parte de los equipos pasan por las fases de Integración, Normalización, Agitación y Realización, pero se pueden poner en marcha muchas acciones para influir positivamente en este proceso, y es recomendable centrarse en lo siguiente:

- a. *Construir relaciones.* Por ejemplo, establecer excursiones o salidas de equipo, comidas, reuniones y otras cosas para que las relaciones puedan evolucionar y los miembros se conozcan y ganen confianza unos en otros. Esto es una práctica habitual adoptada por la Dirección de Proyectos Informáticos (DPI) de la Universidad Tecnológica Nacional, en la Facultad Regional de Tucumán, y ha venido cosechando muy buenos resultados en los últimos tiempos.
- b. *Establecer procedimientos de equipo.* Determinando las reglas, directrices y protocolos que sean necesarias para establecer una productividad de equipo.

6.4.4.2 **Técnicas para potenciar el rendimiento de los equipos**

Una vez que conocemos los principios primordiales para guiar el rendimiento del equipo, observemos unas cuantas técnicas contrastadas para conseguir nuestro objetivo (Horine, 2010):

- Llevar a cabo reuniones de apertura para el equipo, al principio de cada fase. Es una excelente forma de restablecer las expectativas sobre el contexto del proyecto. Y es

conveniente utilizar también minireuniones de apertura al principio de cada fase del proyecto, no sólo al comienzo del proyecto, para restablecer las expectativas.

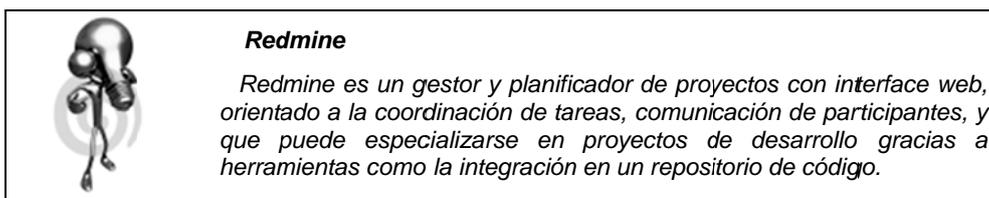
- Crear grupos de trabajo, permitiendo compartir ideas y experiencias, solucionar problemas y aumentar la sinergia del equipo.
- Utilizar sabiamente el tiempo de las reuniones.
- Desarrollar la descripción de los componentes del equipo. Lo importante de esta cuestión no es perder el tiempo en documentar esto, sino el hecho de realizarlo con el equipo para desarrollar estas directrices y procedimientos. De este modo, al igual que con el plan general del proyecto y el calendario, formará parte de su trabajo, es decir, será suyo también.
- Desarrollar, establecer y comunicar normas, aplicando el conocimiento experto.
- Utilizar la experiencia.
- Resolver los conflictos al instante. Los equipos de alto rendimiento no permiten que persistan los conflictos o los problemas, porque si ocurre puede afectar a la productividad del equipo. Como director de proyectos necesita facilitar la resolución de esos problemas inmediatamente. Esto no quiere decir que no escuche y que haga juicios precipitados, significa que se traten, no que se eviten.
- Preparar al equipo para las interacciones directas con el cliente.
- Establecer un repositorio del proyecto. Por ejemplo, a través de una Wiki que permita mejorar y facilitar la productividad del equipo, compartiendo conocimientos y protegiendo a la vez los activos del proyecto.
- Desarrollar costumbres de equipo para crear unidad. Un buen ejemplo podría ser ir a comer todos juntos un día de la semana.
- Asignar tareas efectivas con aceptación clara de responsabilidad y calendario.
- Planificar para la orientación. Existe un período inicial de orientación para cada miembro del equipo que se une al proyecto. El objetivo del director del proyecto debe ser acelerar ese período y obtener la máxima productividad por su parte lo antes posible.

CAPÍTULO VII. REDMINE, SOFTWARE LIBRE DE GESTIÓN DE PROYECTOS

“La construcción exitosa de toda máquina depende de la perfección de las herramientas empleadas.” (Charles Babbage)

7.1 Descripción

Cualquier organización en crecimiento, que maneje una cierta cantidad de proyectos de diversa índole, necesita una herramienta que le permita la gestión eficiente de sus proyectos. En este capítulo se presentará una solución basada en la herramienta de software libre REDMINE. A través de esta herramienta cada equipo de trabajo podrá darle seguimiento a sus actividades, accediendo desde un navegador web.



Redmine es una herramienta de gestión de proyectos software con interface web. Una vez instalada, el administrador da de alta los proyectos a través de la interface web, puede dar de alta a los desarrolladores y jefes de proyecto (o pueden darse de alta ellos mismos a través de la interface web).

Una vez dados de alta los proyectos y sus jefes, estos pueden definir los hitos del proyecto y las tareas a realizar para cada uno de estos hitos.

Los desarrolladores tienen en su página de entrada una lista de las tareas que tienen asignadas. Es una única lista conjunta de las tareas de todos los proyectos. Según van trabajando en las tareas, pueden ir marcando el tiempo que estiman que les llevará la tarea, el tiempo que han trabajado en ella y/o el porcentaje que creen que tienen realizado.

Con esta información, en el hito correspondiente del proyecto se muestra una "barra de progreso" horizontal, en la que una parte aparece en color verde, indicando el número de tareas terminadas, mientras que el resto aparece sin color, indicando lo que queda pendiente. Esta barra de progreso da una idea bastante aproximada de cuánto llevamos hecho y cuánto queda por hacer. Por supuesto, será más aproximada si nos molestamos en meter los tiempos estimados en las tareas y estimamos bien.

7.2 Funcionalidades

7.2.1 Gestión de múltiples proyectos

Se pueden gestionar múltiples proyectos desde una sola interface con una ventana de navegador. La navegación es muy sencilla y se puede saltar y cambiar de proyecto en cualquier momento. Además, cada proyecto puede tener una configuración totalmente diferente y el usuario tener un rol distinto en cada uno. Los proyectos pueden definirse como privados o públicos, visibles para todo el mundo. Es el administrador quien da acceso a cada miembro. También dentro de cada proyecto pueden definirse varios subproyectos.

The screenshot shows the 'Nuevo proyecto' (New Project) form in Redmine. The form is titled 'Gestor de Proyectos DPI' and includes a search bar and a dropdown menu for 'Ir al proyecto...'. The form fields are:

- Nombre ***: Text input field with a note '30 caracteres como máximo.'
- Proyecto padre**: Dropdown menu.
- Descripción**: Rich text editor with a toolbar (Bold, Italic, Underline, Strikethrough, Code, Bulleted List, Numbered List, Indent, Outdent, Link, Unlink, Table, Table of Contents, Print, Help) and a 'Formato de texto: Ayuda' link.
- Identificador ***: Text input field with a note 'Longitud entre 1 y 20 caracteres. Letras minúsculas (a-z), números y signos de puntuación permitidos. Una vez guardado, el identificador no puede modificarse.'
- Sitio web**: Text input field.
- Público**: Checkbox.
- Cliente ***: Dropdown menu with the text '-- Por favor seleccione --'.
- Descripción de Cliente**: Text input field.
- Estado ***: Dropdown menu with the value 'En producción'.

A 'Terminado' button is located at the bottom left of the form.

Figura 74. Nuevo Proyecto en Redmine

7.2.2 Personalización de proyectos

En Redmine cada proyecto es totalmente personalizable, pudiendo encontrar proyectos muy distintos entre sí según sus objetivos. Lo más importante son los módulos que se pueden desactivar o activar para cada proyecto: wiki, foro, noticias, peticiones, control del tiempo, documentos, ficheros o repositorio, aunque hay módulos comunes a todos los proyectos como el de *actividad* y *vistazo*. Si un proyecto está enfocado a notificar incidencias, se puede configurar para incluir solo peticiones, si se busca un proyecto más colaborativo, la wiki y las

noticias son una buena opción, e incluso se puede habilitar un proyecto solo con un foro. La personalización de un proyecto puede ser realizada como se indica en la figura anterior.

7.2.3 Gestión de roles

Cada usuario de Redmine puede pertenecer a un determinado rol, con el cual interactuará en el proyecto pudiendo acceder a diversas funcionalidades determinadas por un administrador del gestor. Es decir, que un usuario posee un rol en cada proyecto. Con ese rol puede tener acceso a diferentes funcionalidades del sistema, según sea la configuración determinada sobre los permisos de ese rol. Una persona puede acceder con más permisos a un proyecto que a otro, porque simplemente en un proyecto tenga un rol distinto que en otro. A continuación se muestra una imagen ilustrativa de la configuración de permisos de un determinado rol, en este caso el Rol del Analista Funcional.

The image shows a web interface for configuring a role named 'Analista Funcional'. At the top, there is a text input field containing 'Analista Funcional' and a checkbox labeled 'Se pueden asignar peticiones a este perfil' which is checked. Below this is a section titled 'Permisos' (Permissions) containing several categories of permissions, each with a list of options and checkboxes:

- Proyecto**
 - Crear proyecto
 - Administrar miembros
 - View rate
 - Modificar proyecto
 - Administrar versiones
 - Seleccionar módulos del proyecto
 - Crear subproyectos
- Foros**
 - Administrar foros
 - Editar mensajes propios
 - Enviar mensajes
 - Borrar mensajes
 - Modificar mensajes
 - Borrar mensajes propios
- Budget module**
 - View budget
 - Manage budget
- Calendario**
 - Ver calendario
- Customer module**
 - View customer
 - Edit customer
 - Assign customer
 - See customer list
- Documentos**
 - Administrar documentos
 - Ver documentos

Figura 75. Permisos del rol analista-funcional

7.3 Sistema flexible de seguimiento de tareas

Una de las mecánicas más útiles para el desarrollo de un proyecto en Redmine son las peticiones y su visualización. Por defecto, el sistema tiene tres tipos de peticiones: tipo Error, tipo Tarea y tipo Soporte. El Administrador del sistema puede crear cualquier otro tipo de petición que desee.

Las peticiones deben asignarse a un determinado miembro del proyecto. Además, las peticiones tienen información que son propias de cualquier actividad que se desarrolle en un proyecto, como por ejemplo: una fecha de inicio y fin para esa petición, control del tiempo en las horas utilizadas, porcentaje realizado, prioridad, etc. El sistema también permite que se

pueda enlazar con la subida de un fichero, y encajar en una categoría (se pueden definir tantas como se quieran). A continuación se muestra una imagen con una petición de tipo Requerimiento de ejemplo:

Requerimiento #2335

Req 51. El Sistema deberá utilizar el servidor Mongrel.

Añadido por González, Gastón Mariano hace alrededor de 1 año. Actualizado hace 9 meses.

Estado: Incorporado
 Prioridad: Normal
 Asignado a: Moreira Mora, Pablo Federico
 Categoría: -
 Versión prevista: Implementación
 Esfuerzo: Media
 Riesgo: Media

Fecha de inicio: 16.06.2010
 Fecha fin:
 % Realizado: 100%
 Tiempo dedicado:
 Tipo: Restriccion de Diseño
 Obligatoriedad: Obligatorio

Descripción
 El mismo es bastante seguro, con características de autenticación y negociación de contenidos, útiles para el sitio.

Subtareas
 Peticiones relacionadas

Historico

- Actualizado por Moreira Mora, Pablo Federico hace 11 meses
 - Versión prevista establecido a Implementacion
- Actualizado por Díaz, Natalia hace 9 meses
 - Tema cambiado Req 38. El Sistema deberá utilizar el servidor Mongrel. por Req 51. El Sistema deberá utilizar el servidor Mongrel.

Terminado

Figura 76. Detalle de una petición

Con todos estos datos, pueden visualizarse las peticiones de manera personalizada estableciendo filtros, y servir así de informes de tareas o incidencias. Además dentro de un proyecto pueden establecerse versiones y asignar tareas a determinadas versiones; así, conforme se marquen tareas completadas, las versiones irán completando su porcentaje automáticamente. A continuación se muestra un ejemplo de una consulta de peticiones filtradas según un determinado valor.

Peticiones

Filtros
 Estado: todos
 Tipo: igual
 Requerimiento

Opciones
 Aceptar Anular Guardar

#	Tipo	Estado	Prioridad	Tema	Autor	Asignado a	Actualizado	Categoría	Fecha de in
3167	Requerimiento	Abortada	Normal	Req 50. "El Sistema deberi poder cargarse en Notebook's, PC's o dispositivos similares con MODEM, Ethernet Cableado o Wireless como configuración mínima".	Díaz, Natalia	Moreira Mora, Pablo Federico	07.10.2010 15:21		07.10.201
3166	Requerimiento	Abortada	Normal	Req 47. "El Sistema deberi desarrollarse utilizando el lenguaje HTML, PHP, Javascript y CSS".	Díaz, Natalia	Moreira Mora, Pablo Federico	07.10.2010 15:22		07.10.201
3143	Requerimiento	Incorporado	Normal	Req 43. "El sistema deberi permitir al usuario con perfil de Administrador eliminar una provincia de un pais determinado".	Díaz, Natalia	Villafañe, César	10.11.2010 19:28		03.10.201
3142	Requerimiento	Incorporado	Normal	Req 42. "El sistema debe	Díaz, Natalia	Villafañe, César	18.11.2010 12:07		03.10.201

Terminado

Figura 77. Tareas organizadas por medio de filtros

Entre las diferentes consultas que el sistema permite realizar, se encuentra la consulta de “Tiempo dedicado” que permite consultar las horas de trabajo registradas por cada miembro. Esta consulta puede ser optimizada incluso, consultando las horas trabajadas en cada petición asignada. Esto permite al Director del Proyecto tener información correcta sobre la participación de cada miembro del equipo en el proyecto.

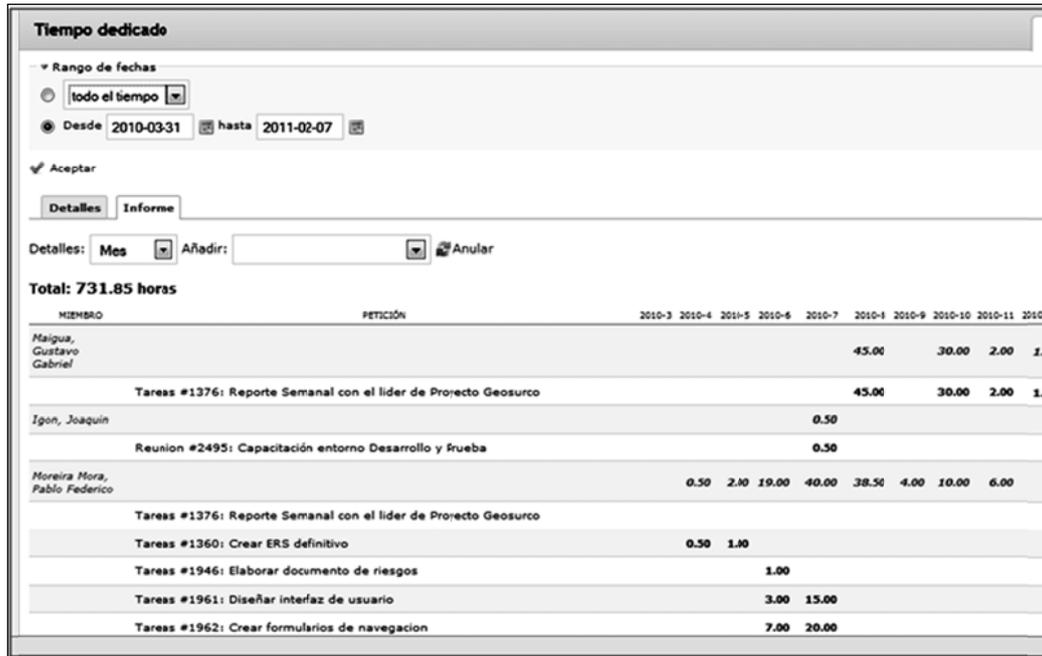


Figura 78. Tiempo dedicado de los miembros a un proyecto, dividido por peticiones

7.4 Uso de calendario y Diagrama de Gantt

Redmine incluye un calendario para visualizar todas las peticiones a lo largo de un mes elegido, indicando claramente el día de inicio y de fin de cada petición.

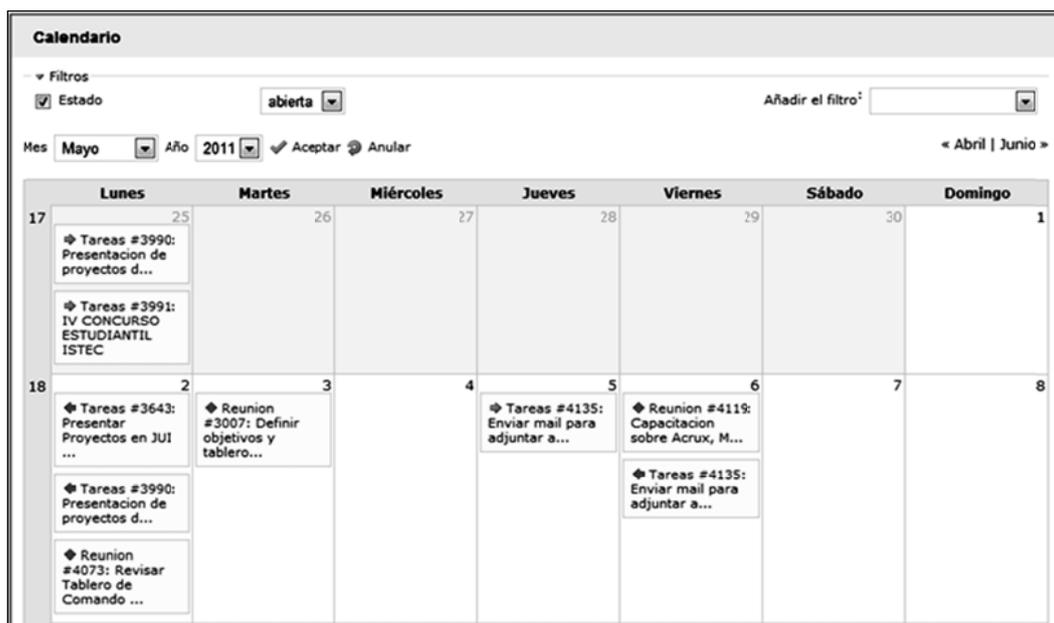


Figura 79. Calendario de actividades mensuales

Del mismo modo, Redmine ofrece una vista en diagrama de Gantt, que va marcando el porcentaje completado conforme avanzan los días. Las peticiones que se visualizan en ambos casos están sujetas a los filtros definidos por el usuario. Redmine utiliza una combinación de colores que permite al usuario tener información rápida sobre cualquier anomalía. Una demora en una determinada actividad será reflejada con el color rojo. El azul indica que las actividades se desarrollan normalmente.



Figura 80. Diagrama de Gantt de un proyecto

7.5 Notificaciones

Configurando previamente el servidor de correo SMTP, Redmine permite enviar notificaciones por correo electrónico en todos los proyectos, definiendo antes los eventos que activan estos avisos. Además cada usuario en su configuración puede elegir recibir notificaciones de cualquier evento, o solo las relacionadas con él (por ejemplo uno de los campos de las peticiones son las personas en seguimiento). Puede configurarse además el servidor de correo entrante, permitiendo así actualizar peticiones simplemente por email e incluso crear nuevas peticiones. Toda la actividad de cada proyecto también puede exportarse en Atom, para ser seguida desde un lector RSS. Si ninguna de estas opciones es favorita, en todos los proyectos existe el módulo de "Actividad", que refleja todo el flujo por días en el proyecto y lo muestra en una lista.

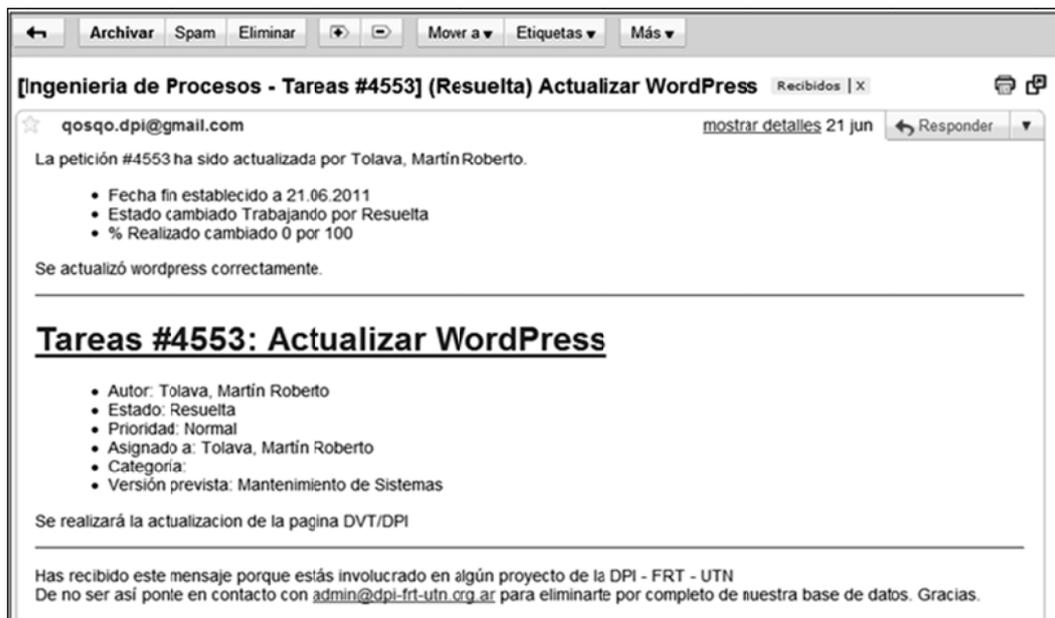


Figura 81. Notificación de una tarea en el correo electrónico

7.6 Administración de noticias, documentos, archivos y ficheros

A través de la herramienta se pueden generar noticias en los proyectos, visibles para todos los miembros, asimismo les permite a los usuarios autorizados del proyecto gestionar los documentos, con la subida de los mismos. Esta misma funcionalidad se puede utilizar para la subida de archivos particulares y ficheros de distinto formatos; estos últimos se utilizan para ser indexados en las peticiones y secciones del gestor.

7.7 Presupuesto del proyecto

La herramienta provee una plugin llamado "Budget Module" que permite calcular el presupuesto del proyecto según el uso de los recursos utilizados. Entre otras cosas, esto permite cotizar la hora de cada recurso humano utilizada, para que el director del proyecto disponga de forma clara sobre los gastos del proyecto realizados hasta una determinada fecha, y de esta forma contrastarlo con la planificación realizada. A continuación se muestra una hoja de presupuesto de un proyecto indicando claramente los costes de cada etapa.

Presupuestos							
Presupuesto total		15.470 €					
Presupuesto mano de obra:		12.620 €					
Presupuesto gastado mano de obra:		830 €					
Presupuesto remanente mano de obra:		11.790 €					
Progreso:		43%					
Puntuación presupuesto:		38					
Excesos:		0 €					
Falta en Deliverables:		12.303 €					
Siguiente entrega:		más de 1 año					
Finalización:		más de 1 año					
Beneficio potencial:		1.250 €					
#	IMPORTANCIA	ASUNTO	PRESUPUESTO	PRESUPUESTO DE MANO DE OBRA	GASTADO	ENTREGA	PROGRESO
▶ 7	94	Presupuesto Fase Analisis	4.970 €	3.570 €	305 €	06.04.2010	<div style="width: 43%;"></div>
▶ 8	0	Presupuesto Fase Diseño	5.750 €	4.900 €	0 €	06.04.2010	<div style="width: 0%;"></div>
▶ 9	70	Presupuesto Fase Implementacion	1.750 €	1.750 €	525 €	06.04.2010	<div style="width: 43%;"></div>
▶ 10	0	Presupuesto Fase Prueba	3.000 €	2.400 €	0 €	06.04.2010	<div style="width: 0%;"></div>
(1-4/4) Por página: 25, 50, 100							

Figura 82. Hoja de presupuestos de un proyecto

7.8 Exportación a distintos formatos

Los informes de peticiones que pueden generarse añadiendo filtros, y que permiten visualizar las diferentes tareas de un proyecto, pueden exportarse en PDF o formato CSV, pudiendo así imprimirlos posteriormente en un formato organizado. Esto permite a un Director de Proyecto, disponer rápidamente de información generada por el proyecto para ser presentada ante quien lo solicite.

Otro punto interesante es que las páginas de la wiki también pueden exportarse en HTML o TXT.

7.9 Análisis gráfico

El sistema permite realizar análisis gráficos de la gestión de proyectos según sea su actividad en el transcurso del tiempo. Esta información presentada en forma de gráficos puede ser muy

útil para hacer un análisis correcto de la situación. A continuación se muestran algunas graficas generadas por el sistema:

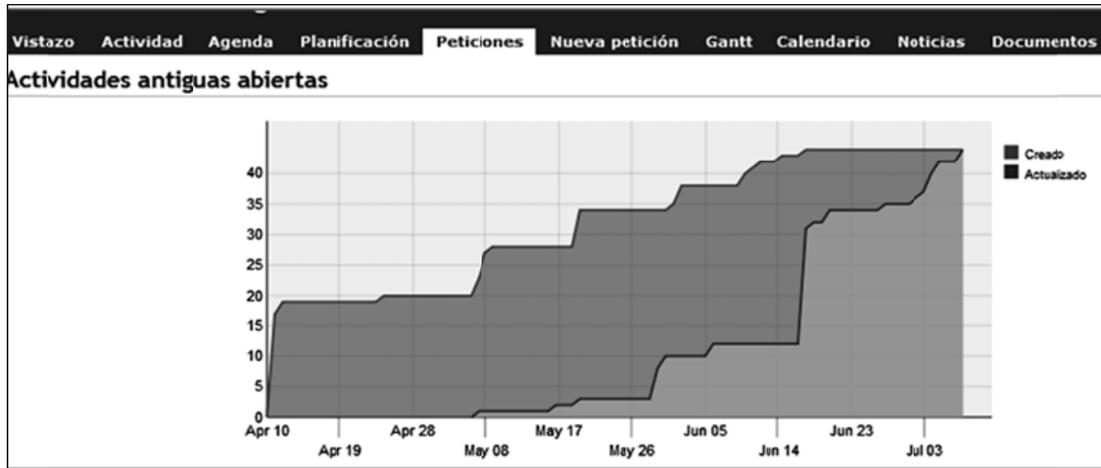


Figura 83. Peticiones creadas comparadas con las peticiones actualizadas

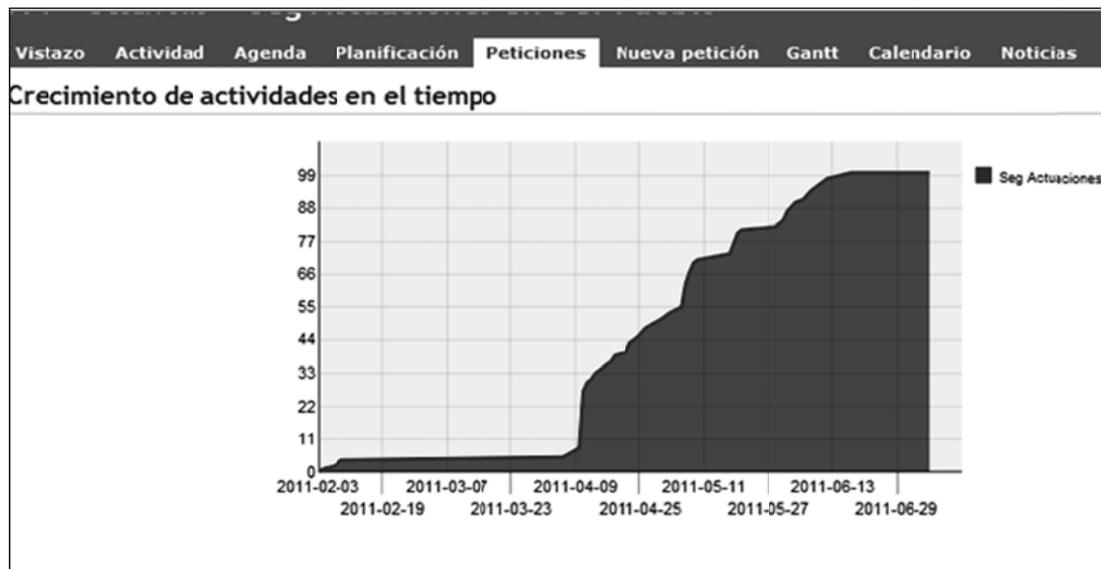


Figura 84. Crecimiento de actividades en el tiempo

7.10 Otras características

Por razones de contenido no se pueden detallar todas las funcionalidades de Redmine en este capítulo. Por tal motivo es importante comentar como otras funcionalidades importantes a destacar:

- la **página personal** de cada usuario, que ofrece una vista personalizable con información de todos los proyectos donde está participando, como un calendario global, o peticiones asignadas.

- la **integración con Repositorios** como CMS, Subversión, etc.
- la posibilidad de **definir campos personalizados** para cada módulo.
- la **barra de búsqueda global**.
- la posibilidad de ampliar la funcionalidad con **decenas de extensiones**.
- la **integración con diferentes bases de datos** MySQL, SQLite y PostgreSQL.

7.10.1 Usabilidad

7.10.1.1 Diseño de la interface

El diseño de la aplicación tiene una interface web muy sencilla y que la hace fácil de manejar. Para cada proyecto existen una serie de pestañas fijas en la parte superior que organizan los diferentes módulos. Dentro de cada módulo se muestra la información correspondiente de forma limpia y ordenada, y a la derecha se suelen incluir una serie de opciones variables según la ventana. El uso es muy simple y destaca la facilidad para configurar los proyectos, la visibilidad del módulo de "Actividad", y los colores elegidos que no recargan la herramienta. Además incluye algunos temas o skins y otros que pueden descargarse.



Figura 85. Interfaz de Redmine con su menú de funcionalidades

7.10.1.2 Facilidad de uso

La aplicación es muy sencilla de usar y aunque el ámbito principal puede ser el empresarial o el de desarrollo de software, cualquier usuario medio podría usar Redmine para administrar sus propios proyectos o tareas. La navegación web se hace muy intuitiva, e incluso puede recordar a la de algunos blogs.

La estructura, y la profundidad de las opciones y configuración están muy bien elegidas, encontrando cada cosa en su lugar indicado. Puede requerir varios minutos conocer todo en profundidad, pero la funcionalidad principal es palpable.

7.10.2 Accesibilidad

Redmine no está dotado con funciones de fácil acceso para personas con problemas de accesibilidad de cualquier tipo. De todas formas la aplicación puede integrarse perfectamente con cualquier tecnología de asistencia del sistema operativo, y con cualquier opción relacionada con el navegador de internet.

7.10.3 Portabilidad/Adaptabilidad

7.10.3.1 Plataformas disponibles

Redmine es una aplicación servidor multiplataforma basada en *Ruby on Rails*. Los únicos requisitos para instalar Redmine en una máquina son: una base de datos (que puede ser *MySQL*, *PostgreSQL* o *SQLite*), y *Ruby* y *Ruby on Rails* en sus versiones apropiadas. Si una máquina sostiene esto, puede instalarse Redmine independientemente de la plataforma. *Rails* funcionará sobre cualquier sistema operativo.

A nivel de cliente, Redmine puede ser accedido desde cualquier plataforma o sistema operativo, tan solo hace falta conexión a la red apropiada y un navegador de internet.

7.11 Plugins

Redmine dispone de un gran número de plugins que rondan la cifra de 100 extensiones. Estos plugins están disponibles en la siguiente dirección web: www.redmine.org/wiki/redmine/Plugin_List

En la lista del anterior enlace vienen los autores de cada uno, una pequeña descripción, versión compatible y de dónde obtenerlo. Pinchando en cada uno se visualiza una ficha específica para cada plugin, con información adicional como la instalación, la actualización o capturas.

Son muy variados y se adaptan a distintas necesidades. Se pueden destacar algunos dedicados a generar gráficos (Charts), a establecer tiempos para cada tarea (Timesheet) o a crear salas de chat (Chat).

Para más información sobre los plugin, como por ejemplo tutoriales y guías de desarrollo para crear propios, hay una página específica en la wiki de Redmine: www.redmine.org/wiki/redmine/Plugins

También son interesantes los plugins de terceros, extensiones que utilizan otras aplicaciones para integrarse con Redmine de alguna manera: www.redmine.org/wiki/redmine/ThirdPartyTools

Por ejemplo hay disponibles del entorno de desarrollo Eclipse, el navegador Firefox o para el teléfono iPhone.

7.12 Licencia/Distribución

La licencia de la aplicación es GPL v2 (GNU General Public License, version 2), cuyos términos se pueden consultar en el siguiente enlace: www.gnu.org/licenses/gpl-2.0.html

Resumidamente define a la aplicación como software libre, con libertad de uso, modificación y distribución.

	<p>Descarga de Redmine</p> <p><i>Redmine puede descargarse de forma libre y gratuita desde su página oficial (que a su vez está montada en Redmine): www.redmine.org/wiki/redmine/Download</i></p>
--	--

Desde el enlace están disponibles las últimas descargas, entre ellas la última versión estable de la herramienta, siempre en código fuente. Existen paquetes para Debian y distribuciones derivadas, e instaladores para otras plataformas, pero no se proporcionan directamente desde la plataforma de Redmine, si no que se consideran paquetes de terceros. Estos paquetes no tienen por qué estar siempre actualizados con la última versión, pero en concreto la versión para este análisis proviene de un paquete *.deb* de una versión de Redmine con solo un mes de antigüedad.

Desde la propia página no se ofrecen servicios o soporte sobre Redmine porque está mantenido por una comunidad de voluntarios, pero existe una gran cantidad de empresas que trabajan con ella y ofrecen instalaciones, soporte, formación o desarrollos.

Se puede probar Redmine en una demo online habilitada desde la propia web: demo.redmine.org

	<p>Demo online de Redmine</p> <p><i>Se puede probar Redmine en una demo online habilitada desde la propia demo.redmine.org</i></p>
---	---

Licencia de módulos/extensiones: Los plugins de Redmine pueden o no tener la misma licencia de la aplicación. De hecho, hay algunos con licencia MIT. Aunque todos se pueden considerar libres, se recomienda leer la licencia de cada caso particular.

La herramienta debe ir acompañada de un completo plan de mejoras de información, para que los miembros de la organización aprendan a manejar la herramienta de forma rápida y eficaz. Esto se ofrece a partir de un servicio completo.

7.13 Comparación con otras aplicaciones de Gestión de Proyectos

En este capítulo se presentaron las principales funcionalidades de utilizar Redmine como herramienta de gestión de proyecto. El uso de dichas funcionalidades se convierte en ventajas competitivas para un Director de Proyectos. Es importante recordar que los autores del libro se basaron en esta herramienta debido a su experiencia en el uso de la misma. Con la intención de que el lector no conozca solamente esta herramienta, se agrega una comparación entre Redmine y otras herramientas similares. En la tabla siguiente se muestra una comparación entre las principales aplicaciones, de código abierto bajo plataforma WEB, para gestión de proyectos de Software.

Aplicación	Administración de Proyecto	Colaborativo	Seguimiento de peticiones	Administración de Recursos	Administración de Documentación
dotProject	X				
eGroupWare	X	X	X	X	X
Project.net	X	X	X	X	X
Redmine	X	X	X	X	X
Trac	X	X			
Readysset	X				

Figura 86. Principales aplicaciones, de código abierto bajo plataforma Web, para gestión de proyectos de software

7.14 Conclusión del empleo de Redmine

Las bondades de disponer una herramienta para la gestión de proyectos como Redmine, hacen que sea una alternativa válida para que un director de proyectos decida gestionar su proyecto por medio de una herramienta informática. Redmine es una herramienta eficiente para cualquier organización en crecimiento, que maneje una cierta cantidad de proyectos de diversa índole.

La experiencia en el uso de la misma y su posibilidad de adaptación a los procesos de cualquier organización, hacen que esta herramienta sea útil para un equipo de trabajo. Utilizar una herramienta correcta para la gestión de proyectos permitirá tener una mejor planificación, seguimiento y control de un proyecto, y con todo esto, un producto o servicio de mejor calidad.

CONCLUSIONES

A diario, en diferentes organizaciones, muchas personas asumen el papel de directores de proyectos. No todas las personas están preparadas para dirigir un proyecto. Resulta clave saber implicar a la dirección para dotar al gestor de proyectos de la relevancia que realmente tienen en los resultados de la organización. Cada vez se requiere más formación para un director de proyecto. Actualmente, la gestión de los proyectos depende en exceso de la experiencia del gestor de proyectos, por lo que es necesario disponer de un conjunto de buenas prácticas para la gestión de proyectos.

Una mala gestión de proyectos desemboca a menudo en la no definición de necesidades de usuario final, en excesos de costes y en retrasos en la entrega de los proyectos. En otras palabras, la calidad del proyecto no será buena. Las causas de estos problemas pueden ser omisiones realizadas durante el desarrollo de sistemas, definición imprecisa de objetivos, estimaciones de costes prematuras, deficientes técnicas de estimación, mala gestión de tiempo, falta de liderazgo y comunicación, etc. Entre las funciones básicas de la dirección de proyectos se incluyen la planificación de las tareas de proyecto, la elección del equipo de proyecto, la organización y la planificación de los esfuerzos del proyecto, la dirección del equipo y el control de la evaluación del proyecto. El director del proyecto es el máximo responsable del proyecto.

Las Metodologías en la Gestión de Proyectos se empieza a aplicar en empresas u organizaciones que adquieren un tamaño o complejidad elevados, o cuando la empresa está en un punto de madurez empresarial en el que ya no está tan preocupada por su subsistencia sino por su productividad y su analítica para mejorar.

Hay muchos tipos de Metodologías (ágiles, ligeras, pesadas), un buen Gestor de Proyectos debe conocer unas cuantas en profundidad (modo de aplicación, cómo, por qué) para poder elegir la adecuada para su Proyecto. El Gestor de Proyectos elegirá la Metodología adecuada en función del Proyecto, de las necesidades y de las Personas que participen en el desarrollo de ese proyecto.

El director de proyectos es un equilibrista que tiene que balancearse siempre entre recursos humanos y materiales, tratando de equilibrar costes y plazos, previniendo riesgos y eventos que puedan afectar el curso normal del proyecto. Un director de proyectos debe ser un facilitador de soluciones constantemente.

Dirigir un proyecto informático es un arte que requiere conocimientos, habilidades y experiencia. En el transcurso de este libro, hemos intentado proveer al lector de un conjunto de conocimientos básicos para la gestión de proyectos, basado en las principales bibliografías disponibles en el mercado. Muchas veces hicimos énfasis en la importancia de adquirir ciertas habilidades que permitan resolver determinadas situaciones de la manera más eficiente posible.

Algunas habilidades pueden ser innatas, como la capacidad de liderazgo, y otras podrán ser adquiridas con un buen entrenamiento, como la capacidad de comunicar eficazmente. También, hemos intentado agregar contenidos de experiencia de los autores en cada uno de los temas abordados, y aunque nuestra intención es la mejor, y el lector repasara este libro varias veces, tenemos que prevenir que la experiencia se la hace andando, acertando y cometiendo errores. Solo el tiempo puede brindarnos esta herramienta de manera eficaz. Recomendamos al lector, seguir esta bibliografía, pero también aprender de su ejercicio diario.

Durante este libro hemos desarrollado las principales fases de un proyecto informático, detallando las actividades comúnmente llevadas a cabo en cada fase y los puntos más importantes a tener en cuenta por el director de proyectos. Además se ha desarrollado un caso de estudio para clarificar los conceptos teóricos descritos.

También hemos redactado información adicional sobre sistemas informáticos para la dirección de proyectos, haciendo énfasis en la tecnología disponible por la comunidad del software libre. Las herramientas para Gestión de Proyectos deben adaptarse a los niveles de desarrollo de los grupos, servir como guía en la gestión, facilitar el seguimiento continuo, integrarse con el resto de la empresa, ser entornos colaborativos con acceso desde cualquier lugar y permitir la Gestión del Conocimiento, de Tareas y de Personas.

Podríamos concluir que hemos reunido un conjunto de Buenas Prácticas en Dirección y Gestión de Proyectos Informáticos, y que las mismas han sido plasmadas en este libro en los diferentes capítulos desarrollados con el fin de proveer de herramientas que permitan al lector gestionar y dirigir un proyecto con calidad.

NOTAS

Casos

Cary Cherniss.....	189
Charles Babbage.....	199
Craig Larman.....	154
Daniel Goleman.....	183, 184, 189
Domingo Ajenjo	22, 53, 152
Dr. Graeme Edwards.....	31
Edsger Dijkstra	130
Gregory M. Horine.....	180, 189
Howard Gardner	182
Jim Collins	180
José Ortega y Gasset.....	25
Lloren Fábregas	154
Luis José Amendola	177, 186
Michael Page.....	14
Paul J. Mayer.....	11
PhD Ing. Alexander Prieto León.....	74
Shirley Hufstedler	160
Stephen Covey.....	186
Thorndike.....	182

BIBLIOGRAFÍA Y REFERENCIAS WEB

Recopilación de fuentes:

- AJENJO, DOMINGO ALBERTO. 2005. Dirección y gestión de proyectos; Un enfoque práctico. 2 ed. España, RA-MA.
- PROJECT MANAGEMENT INSTITUTE. 2008. Guía De Los Fundamentos Para La Dirección De Proyectos (PMBOK Guide). 2 ed. Estados Unidos, ANSI.
- GUTIÉRREZ DE MESA, JOSÉ ANTONIO - PAGÉS ARÉVALO, CARMEN. 2008. Planificación y gestión de proyectos informáticos 2 ed. España. Universidad de Alcalá
- COLLINS, JIM. 2002. Empresas que sobresalen. 2 ed. Estados Unidos, Grupo Editorial Norma.
- PRESSMAN, ROGER. 2005. Ingeniería del Software, un enfoque práctico. 6 ed. Estados Unidos, McGraw-Hill.
- WIEGERS, KARL. 1999. Software Requirements. 2 ed. Estados Unidos, Microsoft Press.
- JACOBSON, IVAR; BOOCH, GRADY; RUMBAUGH, JAMES. 2000. El Proceso Unificado de Desarrollo de Software; La guía completa del Proceso Unificado escrita por sus creadores. Madrid, Editorial PEARSON EDUCACION.
- PETERS, THOMAS; WATERMAN, ROBERT. 1992. En busca de la EXCELENCIA; Experiencias de las empresas mejor gerenciadas de los Estados Unidos. 5 ed. Estados Unidos, Grupo Editorial Norma.
- MINISTERIO DE ADMINISTRACIONES PÚBLICAS DEL GOBIERNO DE ESPAÑA. 2006. Métrica V3; Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información. 1 ed. España.
- COVEY, STEPHEN R. 2005. Los siete hábitos de las personas altamente efectivas. 2 ed. Estados Unidos, PAIDOS.
- AMENDOLA, LUIS JOSE. 2006. Estrategias y Tácticas en la Dirección y Gestión de Proyectos. 1 ed. España. Editorial UPV.
- HORINE, GREGORY. 2009. Gestión de Proyectos. 1 ed. España, Editorial Anaya Multimedia.
- GOLEMAN, DANIEL. 1996. Inteligencia Emocional. 1 ed. Estados Unidos, Editorial Kairós.

- GOLEMAN, DANIEL; CHERNISS, CARY. 2001. Inteligencia Emocional en el Trabajo. 1 ed. Estados Unidos, Editorial Kairós.
- LARMAN, CRAIG. 2003. UML y Patrones. Introducción al análisis y diseño orientado a objetos y proceso. 2 Ed. España, Editorial Pearson
- FRANCES, MONICA; VERNA ETCHERBER, ROBERTO; SCARAFFIA ,CRISTINA. 2011. Promover / Dinamizar verbos de la vinculación. 1ra Edición. Buenos Aires, Editorial edUTecNe
- PAE – Portal de Administración Electrónica del Gobierno de España
<http://administracionelectronica.gob.es>

ANEXO I: Casos De Éxito

1. Introducción

En este anexo I intentaremos mostrar las empresas que ya sea en el desarrollo del proyecto de constitución de sí mismas como entidades de facturación, como en el transcurso de la serie de operaciones que incluyen sus procesos de negocios, gestionaron eficientemente los factores claves para convertirse en exitosas, y lo que es más importante aún, **lograron mantenerse**. Puntualmente se realizará un seguimiento al caso de éxito de tres de ellas, que podrían considerarse las más reconocidas a nivel mundial *Google, Apple y Amazon*.

2. Empresas exitosas a nivel mundial

A nivel mundial existe un ranking que realiza la revista Fortune de EEUU cada año. Como se puede comprobar en este ranking, el volumen económico de las principales empresas del mundo supera con creces al PIB de la mayoría de los países de la Tierra. En cuanto a la fuente de esta información, Fortune es una publicación de negocios (dependiente de Time Warner) fundada en 1.930, cuatro meses después del "crack "de Wall Street. La revista Fortune es conocida por su elaboración de rankings de riqueza, mejores compañías para trabajar y todo tipo de estudios relacionados con el mundo de las finanzas.

Edición 2010			General	Mapa	Gráficas	Mosaico
Posición	Empresa	Facturación [millones (\$)]				
1º	 Wal-Mart Stores	408.214				
2º	 Exxon Mobil	284.650				
3º	 Chevron	163.527				
4º	 General Electric	156.779				
5º	 Bank of America	150.450				
6º	 ConocoPhillips	139.515				
7º	 AT&T	123.018				
8º	 Ford	118.308				
9º	 J.P. Morgan Chase & Co.	115.632				
10º	 Hewlett-Packard	114.552				

Figura 87. Clasificación de las 10 empresas con mayor facturación bruta de todo el planeta en el año 2010, según datos extraídos de la revista Fortune

Como observamos en el *top ten* solo encontramos una empresa del sector informático, HP en el puesto N° 10. Es también sabido que conseguir información relacionada a las actividades financieras de las empresas informáticas es difícil, aún para esta importante revista. Así que vamos a ver información filtrada por otras variables, más representativas que la facturación bruta.

Ranking de las empresas con mayor precio por acción en el Nasdaq Composite (Febrero 2011).

Este informe contiene los datos de todas las empresas que participan en el promedio compuesto del Nasdaq. Este índice, también conocido como *Nasdaq Composite*, es un importante indicador bursátil de Estados Unidos. Este indicador bursátil incluye todos los valores (tanto nacionales como extranjeros) que cotizan en el mercado Nasdaq, un total de más de 5000 empresas. El peso de las empresas en el índice se basa en la capitalización de su mercado, con determinadas normas de limitación de la influencia de los mayores componentes. La alta ponderación de valores tecnológicos dentro del mercado Nasdaq, ha hecho este índice muy popular. En este listado se muestran las compañías con mayor precio por acción en el

momento del cierre mensual a febrero de 2011, información muy valiosa para el análisis histórico.

Posición	Empresa	Empresa, Fundador	Empresa, Facturación	Empresa, Sector	Empresa, Año de fundación	Precio cierre [\$]
1º	Google	Larry Page	23.650	Informática	1.998	613,40
2º	Priceline			Informática		453,88
3º	Mitsui & Company					363,96
4º	Apple	Steve Jobs	36.537	Informática	1.976	353,21
5º	Intuitive Surgical			Investigación y desarrollo		327,95
6º	CME Group Inc.					311,28
7º	Netflix Inc.					206,67
8º	First Citizens BancShares Inc.					202,00
9º	ATRION Corporation					176,34
10º	National Western Life Insurance Company					174,88
11º	Amazon			Informática		173,29

Figura 88. Listado con las compañías con mayor precio por acción a Febrero de 2011

En este caso podemos observar que cuando consideramos un índice tan importante como Nasdaq Composite, que en su composición es mucho más abarcativo nos encontramos con muchas más empresas del sector informático ocupando lugares de privilegio. El monstruo Google a la cabeza, siguiéndolo Apple, y como queriendo entrar al top ten, Amazon. Ahora si nos enfocamos en una región y puntualmente dentro de un sector, podemos darnos cuenta que la supremacía se mantiene, con pequeñas diferencias.

¿Cuál es la mejor empresa tecnológica de Estados Unidos?

En esta encuesta figuran todas las empresas de tecnología que cotizan en el índice NASDAQ. Desde empresas de hardware, software, internet, energías renovables, automoción, biotecnológicas... Estados Unidos es hoy en día la mayor fuente de compañías revolucionarias que tiene La Humanidad. La cuna de las grandes compañías tecnológicas se centra especialmente en zonas como Silicón Valley o el área de Boston, sometidas a la gran influencia de universidades de prestigio como Stanford o el Instituto Tecnológico de Massachusetts (MIT). Sin duda, un entorno difícil de repetir en otros lugares.

Posición	Empresa	Empresa, Fundador	Empresa, Facturación	Empresa, Año de fundación
1º	 Google	 Larry Page	23.650	1.998
2º	 Microsoft	 Bill Gates	58.437	1.975
3º	 Apple	 Steve Jobs	36.537	1.976
4º	 Intel		35.127	
5º	 Cisco Systems		36.117	
6º	 Oracle		23.252	1.977
7º	 eBay		8.727	
8º	 Amazon			
9º	 Dell		52.902	
10º	 Adobe			

Figura 89. Principales Empresas Tecnológicas de EEUU

Ahora si bien cambiaron los competidores, continuamos observando la tendencia, con algunos mejores posicionamientos, el caso de Apple y Amazon.

Ranking de las empresas más admiradas del mundo según la revista Fortune (2010)

Anualmente la revista Fortune hace una encuesta entre empresarios para evaluar la reputación de las empresas de todo el mundo. Sin importar el sector de procedencia, este ranking muestra las 10 empresas más valoradas y admiradas por los empresarios que fueron encuestados.

Edición	2010	General	Mapa	Gráficas	Mosaico	Personalizado
Posición	Empresa	Empresa, Fundador	Empresa, Facturación	Empresa, Sector	Empresa, Año de fundación	
1º	 Apple	 Steve Jobs	36.137	Informática	1.976	
2º	 Google	 Larry Page	23.650	Informática	1.998	
3º	 Berkshire Hathaway		112.493			
4º	 Johnson & Johnson		61.697			
5º	 Amazon			Informática		
6º	 Procter & Gamble		79.697	Comercio al por menor		
7º	 Toyota	 Kiichiro Toyoda	230.200		1.933	
8º	 Goldman Sachs Group		51.673			
9º	 Wal-Mart Stores		408.114	Comercio al por menor		
10º	 The Coca-Cola Company	 John Pemberton	30.590	Alimentación, bebidas y tabaco	1.892	

Figura 90. Empresas más valoradas y admiradas

3. Casos de éxito

A partir de lo expuesto trataremos de mostrar particularmente las características más sobresalientes de cada una de estas empresas que las llevaron a alcanzar y mantener la supremacía.

3.1 GOOGLE: Cultura corporativa + innovación

El uso del buscador Google está tan extendido que la edición 2007 del diccionario Merriam-Webster contiene el verbo transitivo *"to google"*: *buscar una información concreta en Internet a través de un buscador, particularmente Google*. ¿Cómo ha conseguido Google este liderazgo en el mercado de los buscadores?, ¿y cómo espera mantenerlo? Pues, sobre todo, gracias a su cultura de empresa y su constante apuesta por la innovación.

Para analizar la trayectoria de esta empresa, nos remontamos al año 1995. Por entonces, los buscadores respondían a la filosofía de directorio, y eran personas quienes principalmente se encargaban de la categorización de las páginas web. Pero la ingente cantidad de información online hizo impracticable la indexación manual. En diciembre de 1995 apareció el primer buscador basado en la exploración automática y exhaustiva de la web, AltaVista. Unos meses antes, Sergey Brin y Larry Page se habían conocido en la Universidad de Stanford. Juntos, desarrollaron un algoritmo para la búsqueda de información en grandes bases de datos. El motor de búsqueda de Brin y Page ordenaba las páginas con contenidos similares según el

número de enlaces que les apuntaban; en su opinión, cuánto más apuntada era una página, más popular resultaba y, por tanto, más arriba en la lista de resultados merecía estar.

Durante 1998, los estudiantes de Stanford rechazaron varias ofertas de compra de su tecnología y, finalmente, decidieron crear su propia empresa con dinero de amigos y familiares. En verano de ese año, el cofundador de Sun Microsystems, Andy Bechtolsheim, les firmó un cheque por valor de 100.000 dólares a nombre de Google Inc., y los dos jóvenes emprendedores constituyeron una empresa en una cochera de Menlo Park, en la ciudad Silicón Valley en California. Fue entonces cuando la empresa quedó registrada como tal. En 1999, los grandes fondos de capital riesgo pusieron sus ojos en la empresa y, con la financiación apropiada, le dieron el empujón definitivo, que la ha convertido en el líder de las búsquedas en Internet.

En el año 2000, Google ya superaba en tráfico al buscador AltaVista y, en 2002, AOL (America OnLine) lo eligió como motor de búsqueda por defecto para sus más de 34 millones de miembros. En agosto de 2004, Google salió a bolsa con un precio de 85 dólares por acción y obtuvo un capital total de 1.200 millones de dólares. Desde entonces, el valor de la empresa en bolsa no ha dejado de crecer hasta superar la barrera psicológica de los 500 euros el 21 de noviembre de 2006.

A cualquier usuario que se le pregunte ¿y tú porque usas Google? va a contestar algo parecido a:

- Casi siempre encuentro lo que busco sin tener que perder tiempo en visitar sitios que no me interesan.
- Es muy rápido.
- Es muy fácil de usar.

Algo más que publicidad

A mediados de los años noventa, la mayoría de buscadores obtenían sus ingresos de la publicidad que insertaban en sus páginas de resultados de búsquedas.

Google decidió diferenciar totalmente entre los resultados de una búsqueda y los anuncios pagados; así, los resultados de una búsqueda en Google aparecen en una lista situada a la izquierda y los anuncios "breves textos sin imágenes" aparecen a la derecha de la página. Dichos anuncios presentan cierta relevancia para los internautas, ya que están relacionados con sus búsquedas.

Pero los ingresos de Google procedían principalmente de la concesión de licencias de su tecnología de búsqueda a terceras páginas.

Al poco tiempo el buscador lanzó AdWords, un programa de publicidad "autoservicio" que se puede activar online con una tarjeta de crédito en cuestión de minutos.

Google sólo cobra al anunciante cuando un navegante haga clic en el anuncio, independientemente del número de veces que éste aparezca en la página de Google.

Cada vez que un internauta utiliza una palabra, Google decide qué anuncio publica de entre los anunciantes que han apostado por dicha palabra. Esta decisión es fruto de la puja y su estimación de las posibilidades de clic y, por tanto, de su ingreso. AdWords permite crear una campaña desde cinco euros, lo que ha conseguido atraer a pequeños negocios.

En 2003, Google introdujo AdSense. Este programa permite a cualquier página web de terceros incorporar anuncios que Google haya concertado siguiendo la filosofía AdWords. Al convertirse en mayorista de publicidad, Google ofrece a cualquier página con tráfico la posibilidad de cobrar por publicidad, quedándose con una comisión.

El marketing de búsqueda funciona tan bien que, en enero de 2007, el cobro de licencias de su tecnología supuso menos del 1% de los ingresos totales de Google, que ya en 2005 ascendieron a 6.139 millones de dólares.

Pero Google no ha limitado su negocio al segmento de la búsqueda, sino que ha entrado en el escritorio de los usuarios de la mano de aplicaciones como Google Pack.

Siguiendo con la filosofía de no cobrar al usuario y de que la monetización ya aparecerá con el tiempo, todos estos servicios son gratuitos.

Otro de los segmentos en los que Google ha entrado es en el de los servicios dirigidos a desarrolladores de páginas web, entre ellos herramientas de ayuda para la gestión de campañas publicitarias.

Organizarse para innovar.

Una de las claves del éxito de Google ha sido su capacidad para poner en práctica rápidamente ideas innovadoras sin dejar de ser el mejor motor de búsqueda del mercado. Así, la cultura corporativa de Google puede resumirse en cinco puntos:

- 1) Pensar al revés. La empresa está convencida de que los modelos de negocio aparecerán sobre la marcha y prioriza la tecnología por encima del negocio.
- 2) Actuar permanentemente en beta. De esta forma, los clientes se convierten en aliados en el aprendizaje del producto.
- 3) Innovación continua y ejecución veloz, en lugar de la perfección absoluta.
- 4) Renuncia al marketing formal.
- 5) Creación de reglas propias, como demostró con su salida a bolsa en una subasta en Internet.

Con esta filosofía, Google ha implantado una serie de iniciativas para fomentar la creatividad de sus ingenieros, presionados por el día a día. Para ello ha institucionalizado la regla del 20%, que permite a los ingenieros dedicar el 20% de su tiempo a trabajar en proyectos que les apetezca e ilusione. Productos como Google News, Gmail o la red social Orkut son fruto de esta iniciativa.

3.2 GOOGLE Y APPLE

¿Por qué Google y Apple tienen negocios rentables y exitosos y yo no puedo ganar dinero con mi idea de negocios?

Ideas de negocios.

Esta imagen resume muy claramente uno de los factores del éxito de Google y Apple como modelos de negocios. Ambos se concentraron en lanzar al mercado productos simples para no confundir a sus potenciales clientes y así poder vender sus productos y ganar dinero con ellos.

En el extremo superior de la imagen está el concepto de los productos Apple que radica en que simplemente con un botón se podría acceder a todo un mundo de tecnología.

La foto siguiente es la simplicidad de la página de búsquedas de Google, podemos recordar otros motores de búsqueda como Yahoo que eran más complicados. Al final de la foto está el producto complejo en contraposición de la simplicidad de Google y Apple.

La foto evidentemente tiene un toque de humor pero refleja con claridad el concepto de que toda empresa debe de seguir que cuanto más fácil y simple sea su producto, más fácil será transmitir su idea de negocios a sus potenciales clientes y así podrá vender más y ganar más dinero.

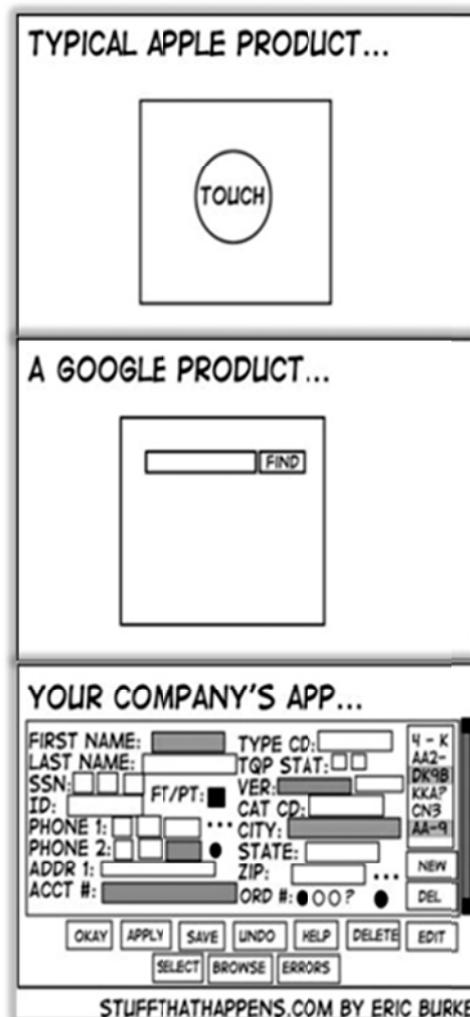


Figura 91. Simplicidad de Google y Apple

ANEXO II: Herramientas De Financiamiento De Proyectos

1. Introducción

En este anexo se intentará dar un panorama general de las herramientas para la financiación de proyectos que existen en el medio público y privado, sus objetivos y alcances.

Cabe precisar que en el contexto habitual donde se desarrollaron y se adquirieron la mayor parte de las experiencias volcadas al presente libro, el término PYMES identifica a Pequeñas y Medianas Empresas que son el motor del desarrollo regional y fuente de trabajo e innovación permanente, las cuales tienen un volumen de producción y plantillas de RRHH menor al de las empresas constituidas, y están directamente relacionadas a la gestión de proyectos. El director de proyectos en su proceso de desarrollo profesional incursiona inicialmente en proyectos desarrollados por estas pymes, principalmente porque es requisito fundamental para que las mismas puedan posicionarse con una oferta competitiva importante, que sus proyectos sean planeados, ejecutados y mantenidos como lo venimos mostrando en este libro. Asimismo financiar dichos proyectos requiere de la aplicación de las competencias con las que cuenta un director de proyectos, que en definitiva debe lograr alcanzar los objetivos con recursos limitados, como lo son los económicos. Por lo que este anexo se desarrollará entre estos contenidos.

Primeramente podemos decir que en general el sector financiero privado en Argentina no se ha mostrado demasiado predispuesto a ofrecer créditos blandos para el desarrollo de proyectos innovadores y/o para la financiación de capital de riesgo. En ese marco, una herramienta clave de financiamiento está representada por los bancos, siendo una de las principales fuentes de financiación externa para las pymes (Pequeñas y Medianas Empresas); en particular cuando superaron la etapa de iniciación y las utilidades no son suficientes para autofinanciarse. La mayoría consisten en préstamos bancarios cuyas tasas de interés suelen ser más altas que las que se ofrecen en las líneas de los Estados nacional o provincial, y uno de los inconvenientes que tienen los pequeños empresarios para acceder al crédito es la dificultad para presentar garantías reales que respalden la operación; es decir, están limitadas en su capacidad de obtener préstamos, porque el valor de sus activos o el monto de su capital societario, no cubre los requisitos de patrimonio computable en relación con el crédito solicitado.

En cuanto a las herramientas financieras del Gobierno, se ofrecen alternativas de financiamiento a partir de sus políticas de apoyo a las micro y pequeñas empresas. En general son programas que tienden a dar soporte a la inversión en bienes de capital, activos de trabajo, incorporación de procesos de calidad e innovación y modernización tecnológica. Estos programas, además de créditos, ofrecen bajo ciertas circunstancias que son específicas de la línea, la posibilidad de subsidios o Aportes No Reembolsables (ANR) y crédito fiscal.

Es para destacar que tanto los subsidios que se otorgan como el crédito fiscal, en todos los casos e independientemente del organismo que lo otorgue, deben contar, para su realización, con una efectiva rendición de gastos. Por lo tanto, es una ayuda para financiar nuevos proyectos o emprendimientos, fomentando la modernización y la innovación tecnológica, pero no puede ser utilizado para quien no dispone de capital para comenzar con el proyecto. En estos casos, se sugiere tratar de tomar créditos dentro del mismo sistema de apoyo a empresas por parte del Gobierno, pues se otorgan a tasas de interés subsidiadas, notablemente inferiores a la que ofrecen los bancos privados.

Tanto en uno como en otro caso, es importante mantenerse atento a las actualizaciones que año a año se van produciendo en las líneas o programas, sobre todo en los que del Gobierno depende. De acuerdo con la dinámica de las realidades empresarias del medio y del mundo laboral, se contemplan nuevas situaciones como, es el caso de los emprendedores que no estaban contenidos en las líneas tradicionales de financiamiento en la actualidad fueron incorporados.

Es importante tener un breve conocimiento de la existencia de estas líneas de financiamiento que ofrece el gobierno nacional Argentino, para poder aprovechar estas oportunidades de financiamiento de proyectos.

2. Organismos nacionales con herramientas de financiación de proyectos

Todos los organismos nacionales tienen como requisito indispensable, para la solicitud ya sea de créditos, subsidios o aportes no reembolsables, la presentación de un **documento de proyecto**, que debe ajustarse a los términos y condiciones fijadas en las bases de la línea y en los formularios diseñados a tal fin. Las bases, como los formularios mencionados, están disponibles en las páginas web de cada organismo.

Es importante tener en cuenta bajo que modalidad se presentan los proyectos. Así podemos encontrar las siguientes modalidades:

- **Ventanilla permanente:** los proyectos a financiar no tienen fecha límite. Por lo tanto, sin plazos determinados, el proyecto puede presentarse en cualquier momento.
- **Ventanilla no permanente** o convocatoria pública: Las bases y condiciones de la convocatoria fijan una fecha de cierre, esta fecha constituye el límite para la presentación del proyecto. En general, hay una convocatoria por año.

También hay que destacar que las líneas de financiamiento se destinan en su gran mayoría a pymes. El criterio que se utiliza para saber si una empresa es o no pyme y cuál es su

clasificación es la Resolución SEPYME 21/2010 (la que puede ser consultada desde www.sepyme.gob.ar/legislacion/):

“Serán consideradas micro, pequeñas y medianas empresas aquellas cuyas ventas totales expresadas en pesos argentinos (\$) no superen los valores establecidos en el cuadro que se detalla a continuación:

TAMAÑO	Sector				
	Agropecuario	Industria y Minería	Comercio	Servicios	Construcción
Micro Empresa	610.000	1.800.000	2.400.000	590.000	760.000
Pequeña Empresa	4.100.000	10.300.000	14.000.000	4.300.000	4.800.000
Mediana Empresa	24.100.000	82.200.000	111.900.000	28.300.000	37.700.000

Figura 92. Clasificación de empresas

Se entenderá por ventas totales anuales, el valor de las ventas que surja del promedio de los últimos tres balances o información contable equivalente adecuadamente documentada, excluidos el Impuesto al Valor Agregado, el impuesto interno que pudiera corresponder y deducidas las exportaciones que surjan de los mencionados balances o información contable hasta un máximo del treinta y cinco por ciento (35%) de dichas ventas.

En los casos de las empresas cuya antigüedad sea menor que la requerida para el cálculo establecido en el párrafo anterior, se considerará el promedio proporcional de ventas anuales verificado desde su puesta en marcha”.

Entre los organismos para tener acceso a fondos en Argentina encontramos:

2.1 Ministerio de Ciencia, Tecnología e Innovación Productiva (MinCyT)

Este ministerio tiene en su ámbito las siguientes reparticiones, que son las que operan con las líneas de financiamiento que se presentan en su página: www.mincyt.gov.ar.

- Agencia Nacional de Promoción Científica y Tecnológica (**ANPCYT**).
www.agencia.gov.ar.



Figura 93. Portal ANPCYT

Es un organismo descentralizado que, aunque depende administrativamente del MinCyT, está gobernado por un directorio propio que se renueva periódicamente. Dispone de fondos del Tesoro Nacional, Prestamos del Banco Interamericano de Desarrollo (BIRF), del recupero del financiamiento reembolsable y provenientes de convenios de cooperación con organismos o instituciones nacionales e internacionales. Los recursos públicos son en parte otorgados a la Agencia para su administración como responsable de la aplicación de la Ley 23877 de Promoción de la Innovación Tecnológica; y de la Ley 25922 de Promoción de la Industria del Software.

Las líneas de financiamiento de la Agencia cubren una amplia variedad de destinatarios; desde científicos dedicados a la investigación básica, hasta empresas interesadas en mejorar su competitividad a partir de la innovación tecnológica.

Promueve el financiamiento de proyectos tendientes a mejorar las condiciones sociales, económicas y culturales en la Argentina a través de **cuatro** fondos:

- **Fondo para la Investigación Científica y Tecnológica (FONCyT)**. Tiene como misión apoyar a actividades cuya finalidad es la generación de nuevos conocimientos científicos y tecnológicos, tanto en temáticas básicas como aplicadas, desarrollados por investigadores pertenecientes a instituciones públicas y privadas sin fines de lucro radicadas en el país.
- **Fondo Fiduciario de Promoción de la Industria del Software (FONSOFT)**. Se creó a partir de la sanción de la Ley 25922 de Promoción de la Industria del Software. Esta sostenido por el presupuesto nacional y financia diferentes actividades a través de convocatorias de créditos y subsidios:

- Proyectos de investigación y desarrollo relacionados con actividades comprendidas en el régimen de promoción (creación, diseño, desarrollo, producción e implementación y puesta a punto de sistemas de software).
 - Programas de nivel terciario o superior para la capacitación de recursos humanos.
 - Programas para la mejora de la calidad de los procesos de creación, diseño, desarrollo y producción de software.
 - Programas de asistencia para la constitución de nuevos emprendimientos.
- **Fondo Argentino Sectorial (FONARSEC).** Apoya proyectos y actividades cuyo objetivo sea desarrollar capacidades críticas en áreas de alto impacto potencial y transferencia permanente al sector productivo: salud, energía, agroindustria, desarrollo social, TI, nanotecnología, biotecnología. El objetivo es acelerar el desarrollo de proyectos públicos-privados; crear o expandir centros de investigación orientados al sector productivo, desarrollando una fuerte plataforma local que pueda ser compartida por varias empresas y/o instituciones.
 - **Fondo Tecnológico Argentino (FONTAR).** Financia proyectos dirigidos al mejoramiento de la productividad del sector privado a partir de la innovación tecnológica. Entre las líneas de FONTAR que se utilizan mencionamos:
 - **CONVOCATORIA PÚBLICA**
 - Aportes No Reembolsables ANR
 - Programa de Crédito Fiscal
 - Créditos Regionales
 - **VENTANILLA PERMANENTE**
 - ANR Patentes.
 - Créditos a Empresas (CAE) Fontar-Bice.
 - Aportes Reembolsables a Instituciones (ARAI).
 - Artículo 2º - Créditos para proyectos de Modernización.
 - Proyectos Integrados de Aglomerados Productivos (PI-TEC).
- **Consejo Federal de Ciencia y Tecnología (COFECyT).**



Figura 94. Portal www.cofecyt.gov.ar

El Consejo Federal de Ciencia y Tecnología (COFECYT), es un cuerpo de elaboración, asesoramiento y articulación estratégica de políticas y prioridades nacionales y regionales que promueven el desarrollo armónico de las actividades científicas, tecnológicas e innovadoras en todo el país.

Su presidencia es ejercida por el ministro de Ciencia, Tecnología e Innovación Productiva. Es coordinado por su secretaria general y está integrado por las máximas autoridades de las provincias y la Ciudad Autónoma de Buenos Aires, con competencia en temas de ciencia, tecnología e innovación productiva que adhieren a la Ley 255467 de Ciencia, Tecnología e Innovación.

Las líneas de financiamiento del COFECYT son todas por convocatoria pública y los recursos administrados a través de ellas son aportes no reembolsables (ANR). En ningún caso estas subvenciones pueden exceder el 70% del coste total del proyecto. El resto de los costes debe ser aportado como contrapartida por los beneficiarios y/o terceros.

Dichas líneas se mencionan a continuación:

- **PFIP – Proyectos Federales de Innovación Productiva.** Esta línea tiene por objetivo dar solución, a partir de la generación y transferencia del conocimiento, a problemas sociales y productivos concretos, de alcance municipal, provincial o regional, identificados como prioritarios por las autoridades provinciales en Ciencia y Tecnología acreditadas ante el COFECYT.
- **ASETUR – Apoyo Tecnológico al Sector Turismo.** Línea de financiamiento desarrollada especialmente para dar impulso a centros turísticos regionales que requieran innovación tecnológica y que hayan sido

seleccionados conjuntamente por las autoridades de aplicación de cada provincia y el área de Turismo, en consonancia con el Plan Estratégico Sustentable 2006-2016. Los proyectos financiados son aquellos que requieren de mejoras tecnológicas con las que marcar una diferenciación y mejor oferta turística. Los proyectos deben beneficiar a la mayor cantidad de actores implicados en el centro turístico correspondiente.

- **DETEM – Proyectos de Desarrollo Tecnológico Municipal.** Su objetivo general es jerarquizar la calidad de vida del municipio a través del desarrollo tecnológico a nivel local y mejores prácticas de gestión, con el fin de dar respuesta a las demandas y necesidades sociales, para asegurar así el desarrollo sustentable, en concordancia con las políticas y estrategias provinciales.
- **PFIP-ESPRO – Proyectos Federales de Innovación Productiva-Eslabonamientos Productivos.** Destinado a fomentar el acercamiento de la ciencia y la tecnología a las necesidades concretas de la producción nacional. El principal objetivo es apoyar el desarrollo competitivo de las cadenas de valor de todo el territorio nacional en correspondencia con las estrategias de desarrollo regional. En este sentido, la superación de las debilidades y desafíos tecnológicos representan un gran impulso para el crecimiento productivo desde una perspectiva específicamente sectorial. Paralelamente, se busca articular el funcionamiento de diversas cadenas de valor a partir de la capitalización y potenciación de los efectos de la incorporación de innovación tecnológica en una cadena sobre el desarrollo de otra.

2.2 Ministerio de Industria, Secretaria de la Pequeña y Mediana Empresa y Desarrollo Regional (SEPYME)



Figura 95. Portal www.sepyme.gov.ar

Esta secretaria orientada a las PYMES tiene un conjunto de líneas de financiamiento que son interesantes en el momento de llevar adelante un proyecto, ya sea para la creación de una nueva empresa o para modernizar tecnológicamente una existente. Podemos nombrar:

- **Programa de Acceso al Crédito y la Competitividad – PACC Apoyo a empresas:** Es un programa por el que las pymes que invierten en asistencia técnica para lograr mejoras en la competitividad, innovación de productos y procesos, ascenso en la escala tecnológica y certificaciones de calidad, pueden obtener un reintegro, por parte de la SEPYME, de hasta el 60% u 80 % y hasta \$130.000.
- **Programa de Desarrollo Emprendedor Capital Semilla:** El programa para el Desarrollo de Jóvenes Emprendedores apoya a jóvenes de 18 a 35 años, que tengan una idea proyecto o un Plan de Negocios en los sectores de industria, servicios industriales, TI e investigación y desarrollo con el aporte de Capital Semilla (Préstamo de Honor), el cual funciona en la modalidad de concurso de proyectos.
- **Programa de Fomento Financiero para Jóvenes Emprendedores-Empresas Madrinas:** Uno de los mayores impedimentos para la constitución o consolidación de emprendimientos de jóvenes consiste en la dificultad para conseguir financiación. El programa Madrinas, es una herramienta que promueve la constitución de alianzas entre jóvenes emprendedores y empresas consolidadas, la empresa madrina financia hasta el 100% del Proyecto y la SEPYME le restituye el 50 % de la inversión mediante un bono de crédito fiscal; sobre el 50% restante la ley propone:
 - Que la empresa realice al emprendedor una cesión a fondo perdido.
 - Que la Madrina obtenga una participación accionaria del emprendimiento (Hasta el 49%)
 - Que la Madrina otorgue al joven emprendedor un crédito blando.
- **PACC Emprendedores. Componente de Apoyo a la Actividad Emprendedora del Programa de Acceso al Crédito y la Competitividad:** Los destinatarios de esta línea son emprendedores con proyectos próximos a iniciarse, o pymes de menos de dos años de existencia desde la fecha de la primera venta realizada. Excluye a aquellas empresas cuyas actividades sean de intermediación, financieras, de seguros o servicios jurídicos o contables. Puede recuperar hasta el 85% del coste de la elaboración de su plan de negocios, estudio de mercado, diseño de procesos operativos o administrativos, inscripciones y certificaciones, adquisición de equipamiento e insumos, diseño y adquisición de packaging y comunicación institucional, entre otras actividades. Puede recibir hasta un monto máximo de \$110.000.

Conclusión

Como vimos en este anexo, existe una amplia gama de alternativas a la hora de buscar financiamiento para los proyectos en los que estará involucrado el director de proyectos, ya sean proyectos puntuales dentro de una pyme o la constitución de la pyme como proyecto en sí. Es importante que el director de proyectos logre combinar sus competencias y habilidades profesionales, así como los contenidos técnicos expuestos en capítulos anteriores con el aprovechamiento de las oportunidades mostradas en materia de financiamiento de proyectos, ya que es fundamental para que los proyectos concluyan exitosamente que el arte de Dirección y Gestión de Proyectos Informáticos sea dinamizada en todas sus dimensiones.

ANEXO III: Índice De Figuras

Figura 1. Evolución histórica del concepto de calidad en la industria del software	7
Figura 2. Estructura ocupacional real	9
Figura 3. Influencia del Entorno en el Proyecto.....	11
Figura 4. Influencia del Proyecto en el Entorno.....	11
Figura 5. La Labor del Director de Proyectos.....	17
Figura 6. Fases Principales de un Proyecto.....	20
Figura 7. Secuencia de Fases	21
Figura 8. Relación de Calidad, Costo y Duración de un Proyecto	23
Figura 9. La fase de planeación genera el Estudio de Viabilidad	26
Figura 10. Diagrama Causa-Efecto	30
Figura 11. Diferencia contra Soluciones.....	31
Figura 12. Factores que intervienen en la generación del Acta de Constitución del Proyecto	33
Figura 13. Procedimiento de preparación de una oferta	42
Figura 14. Evaluación de Tareas y Paquetes de Trabajo	44
Figura 15. Matriz de Estimación	47
Figura 16. Estimación del Esfuerzo y Carga de Trabajo.....	48
Figura 17. Estimación de Gastos y Presupuesto de Gastos.....	50
Figura 18. Estimación Económica	52
Figura 19. Grupos de trabajo que conforman el equipo del proyecto	60
Figura 20. Ciclo de vida del proyecto	62
Figura 21. Fases de ejecución de proyectos.....	63
Figura 22. Estructura de Desglose de Trabajo.....	64
Figura 23. Oferta Económica.....	65
Figura 24. Formato de reporte semanal	73
Figura 25. Técnicas de educación.....	77
Figura 26. Coste por encontrar un error en ERS.....	79
Figura 27. Ejemplo de atributos Verificables/No verificables	83
Figura 28. Estructura de ERS IEEE 830	85
Figura 29. Fases de ciclo de desarrollo de software.....	93
Figura 30. Comparativo ERS IEEE 830 y UML.....	94
Figura 31. Ejemplo de Clase	96
Figura 32. Ejemplo de Diagrama de Clases del Diseño.....	97
Figura 33. Modelo iterativo e incremental de desarrollo de sistemas	98
Figura 34. Diagrama de Secuencia para el caso de estudio que se viene desarrollando	100
Figura 35. Diagrama de comunicación para el caso de estudio (I)	101
Figura 36. Diagrama de comunicación para el caso de estudio (II)	102

Figura 37. Diagrama de comunicación para el caso de estudio (III)	103
Figura 38. Diagrama de Clases para el caso de estudio (I)	104
Figura 39. Diagrama de Clases para el caso de estudio (II)	105
Figura 40. Diagrama de Clases para el caso de estudio (III)	106
Figura 41. Diagrama de Clases para el caso de estudio (IV)	107
Figura 42. Diagrama de Clases para el caso de estudio (V)	108
Figura 43. Ejemplo de orden de implementación	116
Figura 44. Arquitectura del caso de estudio presentado	117
Figura 45. Estructura del código del caso de estudio presentado	117
Figura 46. BasesDeDatos.Controlador (1)	118
Figura 47. LogicaDeNegocio (2)	118
Figura 48. LogicaDeNegocio.Controlador (3)	119
Figura 49. WebServices/WebServicesWrappers (4)	119
Figura 50. Pruebas de visión global	132
Figura 51. Tabla de pruebas I	136
Figura 52. Tabla de pruebas II	136
Figura 53. Tabla de pruebas III	138
Figura 54. Actividad de pruebas I	137
Figura 55. Actividad de pruebas II	137
Figura 56. Actividad de pruebas III	139
Figura 57. Notificación de disconformidad	142
Figura 58. Informe de modificación de software	143
Figura 59. Hoja de estado del documento	144
Figura 60. Costes relativos a cada tipo de mantenimiento	150
Figura 61. Costes relativos a cada fase	152
Figura 62. Distribución del tiempo de Mantenimiento	154
Figura 63. Actividades en la gestión de riesgos	156
Figura 64. Estructura de categoría de riesgos	160
Figura 65. Análisis de Riesgos	161
Figura 66. Probabilidad de Impacto de Riesgo	162
Figura 67. Planificación de la respuesta a los riesgos	163
Figura 68. Clasificación de respuestas a un riesgo	164
Figura 69. Seguimiento y control de riesgos	164
Figura 70. Participantes en cada perfil definido según Métrica V3	170
Figura 71. Comparativa entre Estilos de Dirección y sus características	173
Figura 72. Clasificación de Competencias Emocionales	178
Figura 73. Estilo de Liderazgo, Inteligencia Emocional (IE) y efectividad organizativa, de Goleman y Cherniss (2005)	183
Figura 74. Nuevo Proyecto en Redmine	193
Figura 75. Permisos del rol analista-funcional	194

Figura 76. Detalle de una petición.....	195
Figura 77. Tareas organizadas por medio de filtros.....	195
Figura 78. Tiempo dedicado de los miembros a un proyecto, dividido por peticiones.....	196
Figura 79. Calendario de actividades mensuales.....	197
Figura 80. Diagrama de Gantt de un proyecto.....	197
Figura 81. Notificación de una tarea en el correo electrónico.....	198
Figura 82. Hola de presupuestos de un proyecto.....	199
Figura 83. Peticiones creadas comparadas con las peticiones actualizadas.....	200
Figura 84. Crecimiento de actividades en el tiempo.....	200
Figura 85. Interfaz de Redmine con su menú de funcionalidades.....	201
Figura 86. Principales aplicaciones, de código abierto bajo plataforma Web, para gestión de proyectos de software.....	204
Figura 87. Clasificación de las 10 empresas con mayor facturación bruta de todo el planeta en el año 2010, según datos extraídos de la revista Fortune.....	213
Figura 88. Listado con las compañías con mayor precio por acción a Febrero de 2011.....	214
Figura 89. Principales Empresas Tecnológicas de EEUU.....	215
Figura 90. Empresas más valoradas y admiradas.....	216
Figura 91. Simplicidad de Google y Apple.....	219
Figura 92. Clasificación de empresas.....	222
Figura 93. Portal ANPCYT.....	223
Figura 94. Portal <i>www.cofecyt.gob.ar</i>	225
Figura 95. Portal <i>www.sepyme.gob.ar</i>	226

ÍNDICE GENERAL

CAPÍTULO I. INTRODUCCIÓN A LA DIRECCIÓN DE PROYECTOS.....	6
1.1 ¿Por qué es necesaria la dirección y gestión de proyectos?.....	6
1.1.1 Lo único constante es el cambio	6
1.1.2 Evolución histórica del concepto de calidad en la industria del software	6
1.1.3 Tomar el timón del barco	7
1.1.4 Demanda laboral en Argentina	8
1.1.5 Formalizando aptitudes gerenciales a Informáticos	9
1.1.6 Necesidad de la dirección y gestión del software	10
1.2 Conceptos básicos	10
1.2.1 Proyecto	10
1.2.1.1 Definición	10
1.2.1.2 Características de un proyecto.....	10
1.2.1.3 Tipos de proyectos	10
1.2.1.4 Entorno del proyecto	11
1.2.2 Dirección de Proyectos.....	12
1.2.3 Proyecto Informático.....	12
1.2.4 Tipos de Proyectos Informáticos.....	13
1.2.5 Dirección y Gestión de Proyectos Informáticos	14
1.2.6 Rol del Director del Proyecto	15
1.2.7 Gestión	17
1.2.8 La empresa	17
1.2.8.1 Definición de empresa	17
1.2.8.2 Dimensiones de la empresa	18
1.2.8.3 El empresario.....	18
1.2.8.4 El directivo	18
CAPÍTULO II. FASES DE UN PROYECTO.....	20
2.1 Planeación	21
2.1.1 Inicio y Definición del problema.....	22
2.1.2 Planificación del Proyecto	22
2.2 Ejecución	24
2.2.1 Puesta en marcha.....	24
2.2.2 Fase productiva	25

2.2.3 Conclusión del Proyecto	25
2.3 Fase de Mantenimiento	25
CAPÍTULO III. FASE DE PLANEACIÓN	26
3.1 Descripción de la Fase de Planeación.....	26
3.2 Inicio del Proyecto	27
3.3 Definición del Problema	28
3.3.1 Caso de Estudio: Definición del Problema	29
3.3.2 Entregable – Acta de Constitución del Proyecto	32
3.3.3 Caso de Estudio: Acta de Constitución	34
3.4 Estudio de Viabilidad	40
3.4.1 Oferta Técnica	43
3.4.1.1 Pasos para alcanzar una oferta técnica	43
3.4.2 Oferta de Gestión	45
3.4.2.1 Descripción	45
3.4.2.2 Pasos preliminares para alcanzar una oferta de gestión	46
3.4.3 Oferta Económica.....	48
3.4.3.1 Descripción	48
3.4.3.2 Pasos preliminares para alcanzar una oferta económica.....	50
3.4.3.3 Secciones de la Estimación Económica	50
3.4.3.4 Entregable – Documento de la Oferta	52
3.4.4 Caso de Estudio: Estudio de Viabilidad	54
3.4.4.1 Oferta Técnica	54
3.4.4.2 Oferta de Gestión	60
3.4.4.3 Oferta Económica	64
3.5 Contrato Informático.....	65
3.5.1 Definición de Contrato Informático	66
3.5.2 Partes de un contrato informático	67
3.5.2.1 Los contratantes	67
3.5.2.2 Parta expositiva	67
3.5.2.3 Cláusulas o Pactos	68
3.5.2.4 Los Anexos	68
3.5.3 Tipos de Contratos Informáticos	69
3.5.3.1 Por el Objeto	69
3.5.3.2 Por el Negocio Jurídico	69
CAPÍTULO IV. FASE PRODUCTIVA.....	72

4.1	Descripción de la Fase Productiva.....	72
4.2	Análisis: Ingeniería de Requisitos.....	75
4.2.1	Descripción.....	75
4.2.2	Técnicas principales para la Educación de Requisitos.....	77
4.2.3	Especificación de requisitos del software o <i>ERS</i>	78
4.2.4	Identificación de las personas involucradas.....	79
4.2.5	Problemas para obtener los requisitos	80
4.2.5.1	Relacionados con las personas involucradas	80
4.2.5.2	Relacionados con los analistas	80
4.2.5.3	Relacionados con los desarrolladores.....	81
4.2.6	Entregable – Especificación de Requisitos de Software.....	81
4.2.6.1	Introducción	81
4.2.6.2	Objetivos de la <i>ERS</i>	81
4.2.6.3	Características de una buena <i>ERS</i>	82
4.2.6.4	Esquema de la <i>ERS</i> definida en el IEEE 830-1998.....	85
4.2.7	Conclusión del Análisis	86
4.2.8	Caso de Estudio: Análisis	87
4.2.8.1	Requisito nº33: ABMC de Protocolos.....	87
4.2.8.2	Caso de Uso: Registrar Nuevo Protocolo a Cliente	87
4.3	Diseño.....	89
4.3.1	Descripción.....	89
4.3.2	Fallos en el Diseño de Sistemas.....	91
4.3.3	Entregable – Fase de Diseño	92
4.3.3.1	Introducción	92
4.3.3.2	Relaciones de los artefactos del Diseño con los del Análisis.....	92
4.3.3.3	De la <i>ERS</i> al Diagrama de Clases.....	93
4.3.3.4	Diagrama de Clases	95
4.3.3.5	Crear Diagramas de Clases del Diseño	96
4.3.4	Resumen.....	97
4.3.5	Conclusión de la Fase de Diseño	98
4.3.6	Caso de Estudio: Diseño	99
4.3.6.1	Caso de Uso Real: Registrar Nuevo Protocolo	99
4.3.6.2	Diagrama de Secuencia	100
4.3.6.3	Diagrama de Comunicación	101
4.3.6.4	Diagrama de Clases	104
4.4	Implementación	109

4.4.1	Descripción	109
4.4.2	Preparación del entorno de generación y construcción	110
4.4.2.1	Implantación de la Base de Datos Física o Ficheros	110
4.4.2.2	Preparación del Entorno de Construcción	110
4.4.3	Generación del código de los componentes y procedimientos	110
4.4.3.1	Generación del Código de los Componentes	111
4.4.3.2	Generación del Código de los Procedimientos de Operación y Seguridad	112
4.4.4	Construcción de los componentes y procedimientos de Migración y Carga Inicial de datos.....	112
4.4.4.1	Preparación del Entorno de Migración y Carga Inicial de datos	112
4.4.4.2	Generación del código de los componentes y procedimientos de Migración y Carga Inicial de Datos	112
4.4.4.3	Realización y Evaluación de las Pruebas de Migración y Carga Inicial de Datos.....	113
4.4.5	Elaboración de los Manuales de Usuario	113
4.4.6	Definición de la formación de usuarios finales	113
4.4.7	Entregable – Fase de Implementación	114
4.4.7.1	Introducción	114
4.4.7.2	La programación y el proceso de desarrollo	114
4.4.7.3	Mapeo de diseños para codificación.....	115
4.4.7.4	Orden de la implementación.....	115
4.4.7.5	Resumen.....	116
4.4.7.6	Conclusión de la Implementación	116
4.4.8	Caso de Estudio: Implementación.....	117
4.4.8.1	Estructura de Código	117
4.4.8.2	Clase Protocolo.java	120
4.4.8.3	Clase ControladorProcolo.java – Método Registrar Protocolo	121
4.4.8.4	Clase ProtocoloDAO – Método saveProtocolo	123
4.4.8.5	Clase ProtocoloWs.java – Método registrarProtocolo.....	124
4.5	Pruebas.....	126
4.5.1	Descripción	126
4.5.2	La importancia de la detección oportuna	127
4.5.3	Tipos de Pruebas	127
4.5.4	Ejecución de las Pruebas Unitarias.....	128
4.5.4.1	Preparación del Entorno de las Pruebas Unitarias.....	128
4.5.4.2	Realización y Evaluación de las Pruebas Unitarias	128
4.5.5	Ejecución de las Pruebas de Integración.....	129

4.5.5.1	Preparación del Entorno de las Pruebas de Integración	129
4.5.5.2	Realización de las Pruebas de Integración	129
4.5.5.3	Evaluación del Resultado de las Pruebas de Integración	129
4.5.6	Ejecución de las Pruebas del Sistema.....	129
4.5.6.1	Preparación del Entorno de las Pruebas del Sistema.....	130
4.5.6.2	Realización de las Pruebas del Sistema	130
4.5.6.3	Evaluación del Resultado de las Pruebas del Sistema.....	130
4.5.7	Pruebas de Regresión.....	131
4.5.8	Entregable – Fase de Pruebas	132
4.5.8.1	Introducción	132
4.5.8.2	Tipos de Pruebas	132
4.5.8.3	Documentos más usuales en cada fase.....	132
4.5.9	Conclusión de la fase de Pruebas.....	135
4.5.10	Caso de Estudio: Pruebas	135
4.5.10.1	Pruebas Unitarias	135
4.5.10.2	Pruebas de Integración	135
4.5.10.3	Pruebas del Sistema	135
4.6	Implantación.....	139
4.6.1	Descripción	139
4.6.2	Capacitación de Usuarios del Sistema.....	140
4.6.3	Objetivos de la Capacitación	141
4.6.4	La Evaluación del Sistema	141
4.6.5	Control de configuración en la fase de implantación	142
4.6.6	Tareas Usuales de la Implantación del Sistema	144
4.7	Conclusión del Proyecto.....	146
4.7.1	Descripción	146
4.7.2	Objetivos del cierre del Proyecto	146
4.7.3	Acciones a seguir por el director del proyecto para efectivamente concluir con el proyecto.....	147
4.8	Mantenimiento	148
4.8.1	Descripción	148
4.8.2	Tipos de Mantenimiento	150
4.8.3	Distribución del Coste	151
4.8.4	Factores que afectan al coste.....	152
4.8.5	Distribución del Tiempo	153
4.8.6	Tareas Usuales en la fase de Mantenimiento	154

CAPÍTULO V. GESTIÓN DE RIESGOS	156
5.1 Descripción	156
5.2 Planificación de la Gestión de Riesgos	158
5.3 Identificación de Riesgos	159
5.4 Análisis de Riesgos.....	160
5.5. Planificar la respuesta a los riesgos.....	162
5.6 Seguimiento y control de riesgos	164
5.7 Beneficios de la Gestión de Riesgos	165
CAPÍTULO VI. RECURSOS HUMANOS.....	166
6.1 Introducción	166
6.2 Perfiles más comunes en un Proyecto Informático	167
6.3 El Director del Proyecto	170
6.3.1 Definiciones.....	170
6.3.2 ¿Por qué un Project Manager?.....	171
6.3.3 Habilidades y Características.....	173
6.3.4 Errores comunes de los Directores de Proyectos:	174
6.4 Inteligencia Emocional aplicada a la Dirección de Proyectos	176
6.4.1 Introducción	176
6.4.2 Liderazgo	179
6.4.3 La Comunicación en el proyecto	183
6.4.3.1 La importancia de la Gestión de las Comunicaciones en el Proyecto	184
6.4.3.2 Competencias interpersonales.....	185
6.4.3.3 La dificultad de comunicarse.....	186
6.4.3.4 Manejo de conflictos en la comunicación.....	187
6.4.4 Potenciar el rendimiento del equipo del proyecto	187
6.4.4.1 Principios de Gestión.....	187
6.4.4.2 Técnicas para potenciar el rendimiento de los equipos	189
CAPÍTULO VII. REDMINE, SOFTWARE LIBRE DE GESTIÓN DE PROYECTOS.....	192
7.1 Descripción	192
7.2 Funcionalidades.....	193
7.2.1 Gestión de múltiples proyectos.....	193
7.2.2 Personalización de proyectos.....	193
7.2.3 Gestión de roles.....	194
7.3 Sistema flexible de seguimiento de tareas.....	194
7.4 Uso de calendario y Diagrama de Gantt.....	196
7.5 Notificaciones.....	198

7.6 Administración de noticias, documentos, archivos y ficheros	198
7.7 Presupuesto del proyecto	199
7.8 Exportación a distintos formatos	199
7.9 Análisis gráfico.....	199
7.10 Otras características	200
7.10.1 Usabilidad	201
7.10.1.1 Diseño de la interface.....	201
7.10.1.2 Facilidad de uso	201
7.10.2 Accesibilidad.....	202
7.10.3 Portabilidad/Adaptabilidad	202
7.10.3.1 Plataformas disponibles	202
7.11 Plugins	202
7.12 Licencia/Distribución.....	203
7.13 Comparación con otras aplicaciones de Gestión de Proyectos	204
7.14 Conclusión del empleo de Redmine.....	204
CONCLUSIONES	206
NOTAS	208
BIBLIOGRAFÍA Y REFERENCIAS WEB	210
ANEXO I: Casos De Éxito.....	212
ANEXO II: Herramientas De Financiamiento De Proyectos	220
ANEXO III: Índice De Figuras.....	230

Buenas Prácticas en la Dirección y Gestión de Proyectos

Autores:

Gustavo Gabriel Maigua [ver CV]

Emmanuel Fernando López [ver CV]

Diseño de Tapa: *Lic. Lorena Milani*

Diseño de Interior: *Ing. Gustavo Gabriel Maigua*

Corrección de Estilo: *Ing. Gustavo Rego*

ISBN 978-987-

Universidad Tecnológica Nacional – República Argentina

Rector: *Ing. Héctor C. Brotto*

Vicerrector: *Ing. Carlos E. Fantini*

Facultad Regional Tucumán – U.T.N.:

Decano: *Ing. Walter Fabián Soria*

edUTecNe – Editorial de la Universidad Tecnológica Nacional

Coordinador General: *Ing. Ulises J. P. Cejas*

Director de Ediciones: *Ing. Eduardo Cosso*

Coordinadores del Comité Editorial: *Ing. Juan Carlos Barberis y Dr. Jaime Moragues (a.c.)*

Áreas Pre-prensa y Producción: *Tec: Bernardo H. Banega e Ing. Carlos Busqued*

Área Comercialización: *Fernando Horacio Cejas*

Buenos Aires

2012