



GESTIÓN DE PROYECTOS INFORMÁTICOS

METODOLOGÍA

Creación: Marzo de 1999

Revisión: Nov. de 2000

Revisión: Febrero de 2006

Revisión: Marzo de 2008

Revisión: Julio de 2011

Revisión: Diciembre de 2014

Ult. Revisión: Mayo de 2016

© **iT-South Consultores**

Av. Alicia Moreau de Justo 740 piso 3º of. 1

C1107AAN Puerto Madero, Bs.As. - Argentina

Teléfono +54 +11 4334-0033 4343-3157

Email: info@itsouth.com.ar

Web: www.itsouth.com.ar

INDICE

INDICE	2
OBJETIVO	3
ALCANCES	3
REQUERIMIENTOS	3
INTRODUCCIÓN	4
DIAGRAMA DE PROCEDIMIENTOS	5
Fases y Etapas	6
Fase 1: Estudio de Factibilidad	9
Diagrama Detallado de Procedimientos de la Fase 1	12
Fase 2: Análisis y Diseño Global	13
Diagrama Detallado de Procedimientos de la Fase 2	16
Fase 3: Desarrollo	22
Diagrama Detallado de Procedimientos de la Fase 3	25
Fase 4: Puesta en Producción	28
Diagrama Detallado de Procedimientos de la Fase 4	30

OBJETIVO

Reglamentar los procedimientos para dirigir y gestionar un proyecto mediano o grande (hasta cierto punto). Entendiéndose por tal a un proyecto que supere las 1000 horas y que no exceda las 30000, con un equipo que varíe entre dos y quince personas.

ALCANCES

Desde que aparece la posibilidad de presentar una propuesta hasta la puesta en producción del sistema.

Este documento está destinado a una instrucción interna sobre la metodología para todos los miembros del equipo del proyecto, aunque está abierto para presentar al cliente del potencial proyecto a fin de que verifique la calidad del proceso.

REQUERIMIENTOS

- ✓ Experiencia personal en gestión de proyectos, necesaria para implementar con éxito esta metodología.
- ✓ Conocimiento de las herramientas necesarias para controlar la gestión del proyecto: entre las más importantes están Project para el plan global y Asana para la asignación de tareas en cada iteración y GitLab para el soporte de incidentes.
- ✓ Conocimiento tecnológico general del ámbito de desarrollo del proyecto, a fin de poder tomar las decisiones correctas.
- ✓ Se recomienda estar al corriente del contenido de los siguientes documentos:
 - *“Resumen de Gestión de Proyectos Informáticos” (RGPI)*
 - *“Metodología de Programación”*
 - *“Modelo de Propuesta Preliminar” (PropuestaBase.doc)*
 - *“Modelo de Minuta de Relevamiento” (MinutaRelev.doc)*
 - *“Modelo de Acta de Entrega” (ActaEntrega.doc)*
 - *“Modelo de Acta de Inicio/Fin de Etapa” (ActaEtapa.doc)*
 - *“Modelo de Plan de Proyecto” (PlanModelo.mpp)*
 - *“Modelo de Análisis de Requerimientos” (AnalisisReqModelo.doc)*
 - *“Planillas de Diseño” (_EDabm_blanco.doc, _EDproc_blanco.doc, _EDrut_blanco.doc, _EDcons_blanco.doc, _EDlist_blanco.doc, _EDbr_blanco.doc, _EDmenu_blanco.doc, _EDnotas_blanco.doc)*
 - *“Planillas de Control de Calidad” (_CC_blanco.doc)*
 - *“Modelo de Evaluación de Proyecto” (EvaluacionProyecto.doc)*

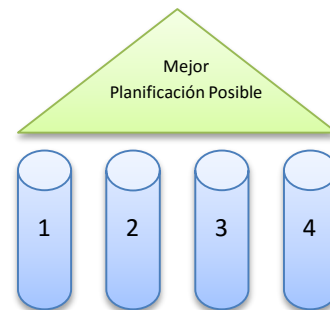
INTRODUCCIÓN

El 65% de los proyectos se atrasa entre un 25 y un 50% de su duración total, dicho porcentaje se incrementa proporcionalmente al tamaño del proyecto. Existen múltiples medios para tratar de evitar, o al menos reducir, la probabilidad de que se atrase como así también el porcentaje de retraso (si llega a ocurrir).

Se denomina *planificación exitosa* a aquella que no sufre retrasos, o al menos los retrasos y contratiempos que sufre son razonablemente reducidos, los costos están más o menos acordes a los planificados, y el producto entregado cumple con la planificación en cuanto a las prestaciones y la calidad deseadas. Esto también se conoce como *desarrollo eficiente*, y a esto es a lo que se apunta mediante la aplicación de la metodología presentada.

Para conseguir este tipo de planificación, también llamada “*Mejor Planificación Posible*”, hay que basarse en los cuatro pilares de la planificación:

1. Evitar los **errores clásicos**
2. Respetar las **bases teóricas del desarrollo**
3. Realizar una eficiente **gestión de riesgos**
4. Utilizar **métodos orientados a la planificación**



Son muchos los aspectos que hay que tener en cuenta para lograr la solidez de estos cuatro pilares. En esta metodología se expondrán lineamientos generales y se hará mención de la mayoría de los mencionados métodos, sin embargo no se entrará en detalle más de lo necesario. Para mayor información referirse al documento “*Resumen de Gestión de Proyectos Informáticos*”, que es un compilado de un trabajo de *Steve McConnell* (consultor de renombre, ingeniero jefe de software en una importante consultora estadounidense y contratista de empresas del calibre de *Microsoft Corporation*). En esta metodología se hará referencia en distintas oportunidades a dicho documento mediante la sigla *RGPI*.

Todos los temas expuestos a continuación están redactados para un lector que pueda ser perfectamente un candidato para la dirección de un proyecto, con la finalidad de que pueda utilizar este documento a modo de guía para la gestión y control del mismo.

DIAGRAMA DE PROCEDIMIENTOS

La metodología presentada está basada en el uso de procedimientos y herramientas estándares y en la experiencia práctica de los autores de la misma. Su ciclo de vida es el denominado ***cascada modificada con entrega por etapas***.

Este modelo utiliza *técnicas de desarrollo incremental* combinadas con *metodologías ágiles*, aprovechando todas las ventajas del **control** del modelo en cascada, pero permitiendo entregas parciales de módulos con la **flexibilidad** de las metodologías ágiles, con lo que se lograría ir depurando un módulo a medida que se desarrollan otros. Los módulos críticos del proyecto serían siempre los primeros a tener en cuenta en la planificación global.

Cabe destacar que, a diferencia de las **metodologías ágiles puras**, este **ciclo de vida combinado** brinda un mayor control y visibilidad del proyecto a corto, mediano y largo plazo. Metodologías como *scrum* son utilizadas en cada iteración (para llegar a una entrega parcial) y en la etapa final de ajustes post-entrega.

Las principales ventajas de este modelo son:

- ✓ Se puede ir entregando el sistema rápidamente una vez finalizado el análisis, con lo cual se obtiene una **temprana retroalimentación** que ayuda a reducir los riesgos del proyecto.
- ✓ Se puede **controlar mejor la planificación**, ya que se manejan varios hitos de entrega parcial del producto, y no uno solo al final. Esto da margen suficiente para una replanificación en caso de ser necesaria. Además, mejora la visibilidad del proyecto para la directiva y el cliente.
- ✓ Se realiza una **mejor gestión de riesgos** debido al fluido contacto con el cliente y la consecuente reducción de zonas oscuras del sistema, encarando en las primeras iteraciones las partes más complejas o importantes del sistema.
- ✓ Se consigue una gran **satisfacción del cliente**, ya que este recibirá periódicamente los frutos del desarrollo, logrando indirectamente que se involucre íntimamente con el sistema, ya que se sentirá partícipe y responsable de una parte muy importante del proyecto, que es la aprobación de su funcionalidad.
- ✓ **Mejora en la calidad** por tener que depurar hasta conseguir una “*calidad de entrega*” cada 15 o 20 días gracias al uso de scrum.

El único inconveniente es que es un modelo que invita a los cambios en los requerimientos. Por ello hay que complementarlo con un buen **grupo de control de cambios** (ver más información al respecto en el documento de *RGPI*). En cuanto a los cambios aceptados, hay que considerar en qué etapa serán agregados, ya que habrá que replanificar. Además, este modelo requiere una muy buena planificación para evitar que se realice primero una entrega que dependa de algún otro módulo que todavía no haya sido entregado.

Es aconsejable realizar un plan de entregas quincenales o a lo sumo mensuales. Se recomienda utilizar un método de **hitos miniatura** (ver más información al respecto en el documento de *RGPI*) para la planificación y un modelo de **entrega evolutiva mediante scrum** (especie de prototipado evolutivo que tiende a mejorar el módulo entregado hasta que este es aceptable) después de cada entrega. Se deben planificar primero las entregas de las partes más importantes del sistema, para eliminar riesgos y dar un rumbo claro al proyecto.

No todas las etapas tienen que entregarse al cliente, siempre servirán para comprobar (quizás ante la directiva) el estado del proyecto. Cada entrega debe estar completa y con un nivel aceptable de calidad.

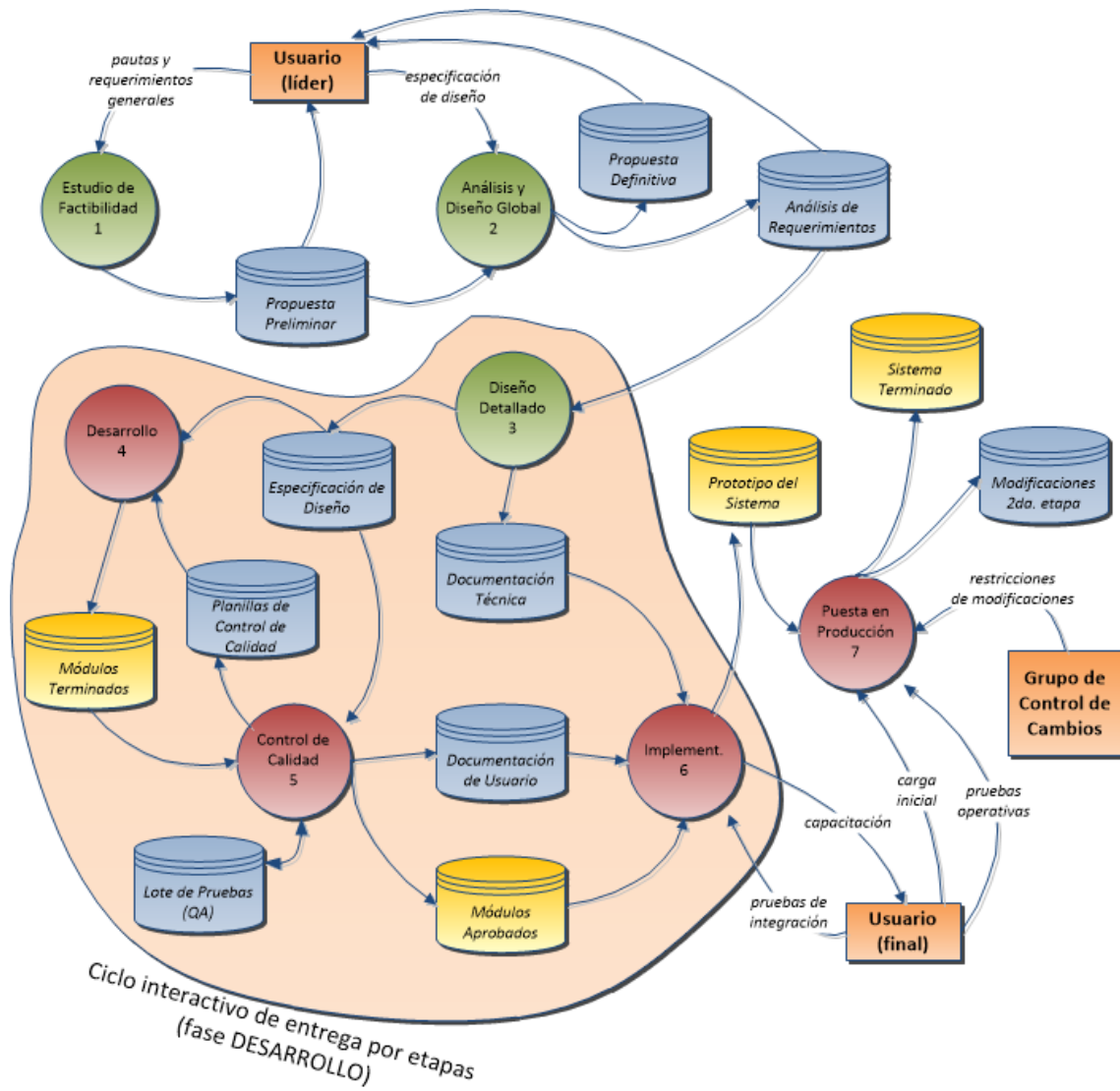
Esta metodología es totalmente independiente de la arquitectura (desktop, web, mobile, service, etc.) y de la plataforma de desarrollo (Net, Java, PHP, etc.), ya que apunta al proceso en sí mismo.

FASES Y ETAPAS

Como todas las metodologías, esta se divide en **FASES** y **ETAPAS**, las cuales se desarrollarán a continuación.

Las fases a desarrollar son:

- 1) **Estudio de Factibilidad:** tendrá como objeto relevar los requerimientos en líneas generales y confeccionar una “*Propuesta Preliminar*”.
- 2) **Análisis y Diseño Global:** se analizará a bajo nivel cada requerimiento documentado en la etapa anterior y se realizará un diseño global de los principales circuitos del sistema. El resultado de esta etapa será el “*Análisis de Requerimientos*”, con el cual se confeccionará la “*Propuesta Definitiva*”.
- 3) **Desarrollo:** se compone de varias etapas bien diferenciadas y repetitivas por cada submódulo del sistema (llamadas *iteraciones*, las cuales se llevan a cabo mediante *scrum*):
 - a) **Diseño Detallado:** se elaborarán las “*Especificaciones de Diseño*” de cada funcionalidad del sistema. Esta tarea puede llegar a incluir el desarrollo del maquetado de la funcionalidad en cuestión.
 - b) **Desarrollo:** se codificará cada una de las funcionalidades especificadas en la etapa anterior.
 - c) **Control de Calidad:** se realizarán los controles de calidad de las funcionalidades desarrolladas. Esta etapa incluye el ajuste de los módulos con calidad inaceptable.
 - d) **Implementación:** se lleva a cabo la integración de las funcionalidades, la carga inicial y la confección del ejecutable o instalador a entregar al cliente, el cual es instalado en un entorno de testing para que los usuarios puedan probar el funcionamiento del módulo entregado. También se dicta la capacitación relativa al módulo entregado.
- 4) **Puesta en Producción:** esta etapa tiene lugar cuando ya se terminaron de desarrollar todos los módulos del sistema. Su función es realizar los ajustes requeridos para que el sistema pueda llevarse a producción, realizar la carga inicial y migración definitiva de los datos, parametrizar el sistema, e implementarlo en el servidor adecuado.



Algunas aclaraciones sobre el gráfico:

- ☞ Por razones de claridad visual, no se grafican todos los detalles. Los mismos se aclaran al desarrollar cada fase.
- ☞ Los rectángulos representan a entidades externas. A los efectos de simplificar el esquema, no se identifica al equipo del proyecto en relación a su participación en cada fase, sin embargo se deja en claro que el mismo participa en mayor o menor grado, en la totalidad del proyecto. Al desarrollar cada fase se detallarán sus implicados.
- ☞ La entidad externa *USUARIO* representa en forma genérica al cliente. Generalmente, esto involucra un líder de proyecto por parte del cliente, un responsable de sistemas, y uno o más usuarios finales.
- ☞ Los círculos representan las fases del proceso. El orden de ejecución está indicado por su número de subíndice.
- ☞ El cilindro representa un objeto tangible que en la práctica se transformará en un entregable que podrá ser comprobado por el cliente y marcará el fin de alguna etapa. Dicho objeto puede ser un documento, un prototipo o el software desarrollado. Se diferencia con colores los tipos de entregables (documentos por un lado y productos software por otro).
- ☞ Las flechas representan el flujo de la información. Se aclara la índole de la información que circula en cada flujo sólo cuando circula entre entidades y procesos.
- ☞ La fase de desarrollo implica que los procesos que encierra se repiten secuencialmente para el diseño, codificación, prueba e implementación de cada módulo o etapa, utilizando una o más iteraciones para el control de dicho ciclo.

Siempre y cuando la modalidad de trabajo del cliente lo permita, es conveniente confeccionar un **“Acta de Inicio/Fin Etapa”** al inicio y fin de cada fase (ver documento *ActaEtapa.doc* para obtener un modelo de la misma). Esto servirá para dejar establecido sin lugar a dudas, cuáles fueron los tiempos reales del proyecto. Dichas actas deberán firmarse por todos los responsables del proyecto.

FASE 1: ESTUDIO DE FACTIBILIDAD

Esta fase tendrá como objetivo **relevar y analizar en líneas generales** los requerimientos y características globales de la aplicación a desarrollar, a fin de establecer el *alcance* del sistema y el *esfuerzo* necesario para su desarrollo, de forma de poder conformar una propuesta lo más precisa posible.

En los casos en que sea factible, es conveniente **solicitar un documento escrito** conteniendo los mencionados requerimientos. Este documento suele ser un manual de procedimientos, una especificación del usuario, o una especificación de diseño especialmente confeccionada para la aprobación del presupuesto interno para el desarrollo del nuevo sistema.

Esta fase suele tener una **duración media** de una o dos semanas desde la primera entrevista con el cliente hasta la entrega del documento final. Generalmente, no son necesarias más de una o dos entrevistas (una al principio muy general y otra después de analizar todos los requerimientos). Sin embargo esto dependerá de lo precisa que sea la documentación entregada por el cliente.

Los **recursos implicados** en esta fase son generalmente de altos niveles jerárquicos:

- ✓ Líder de proyecto de la consultora
- ✓ Analista técnico y/o funcional de la consultora
- ✓ Líder de proyecto del cliente
- ✓ Responsable de sistemas del cliente
- ✓ Representante de los usuarios finales

La **salida o cierre** de esta fase lo conforma el documento que denominaremos **"Propuesta Preliminar"**. Dicha propuesta está desarrollada íntegramente en un documento de Word, el cual se recomienda tomar como modelo para el desarrollo de una nueva propuesta (ver documento *PropuestaBase.docx*). La misma debe contener lo siguiente:

- 1 Software.
 - Datos técnicos del software a utilizar, el motor de bases de datos, los requerimientos de hardware, la tecnología de desarrollo, el esquema de distribución de las distintas partes del producto, etc.
 - Breve descripción funcional de la aplicación a desarrollar, incluyendo un diagrama de casos de uso general del sistema, sin olvidar mencionar expresamente los alcances de la aplicación, los potenciales clientes y las ampliaciones futuras o expectativas para con el software a obtener en el corto, mediano y largo plazo.
- 2 Descripción de la implementación y capacitación a brindar.
- 3 Documentación a presentar, tanto técnica como de usuario.
- 4 Opciones de mantenimiento postventa.
- 5 Plan de actividades mencionando:

- La configuración del equipo de trabajo (cantidad, perfil y participación en horas de cada recurso involucrado).
 - El diagrama estimado de tiempos (representado mediante un Gantt en el que se especifiquen las fases y etapas principales, sin entrar en detalle de cada tarea en particular, pero indicando los recursos involucrados en cada una de las etapas).
 - Un cronograma de actividades e hitos principales.
 - La modalidad de trabajo propuesta, sin olvidar aclarar la dependencia existente entre las distintas fases del ciclo de vida del proyecto (por ejemplo, que no se comenzará el desarrollo hasta no tener la aprobación escrita del análisis).
- 6 Condiciones generales de contratación incluyendo:
- Propiedad intelectual (del aplicativo y las bibliotecas).
 - Confidencialidad de la información.
 - Tareas no cotizadas en el presupuesto: por ejemplo, se recomienda no incluir la etapa de migración de datos, al menos hasta concebir el modelo de datos del nuevo sistema.
- 7 Cotización.
- Importe en horas (totales por proyecto y parciales por recurso y etapa) según el diagrama de tiempos antes mencionado.
 - Forma y condiciones de pago.
 - Aclaración sobre el carácter *preliminar* de la propuesta (es decir, que la misma puede variar entre un 20 y un 30%, dependiendo de la claridad de los requerimientos y del tamaño del proyecto). Esta propuesta revestirá un carácter de definitiva al finalizar la fase de Análisis. Se recomienda en esta etapa, solicitar la opinión del cliente (específicamente la del representante de sistemas) en cuanto a su propia estimación de tiempos, cantidad y calidad de los recursos necesarios. También es recomendable basarse en la propia experiencia en sistemas afines tecnológicamente y de alcances similares.

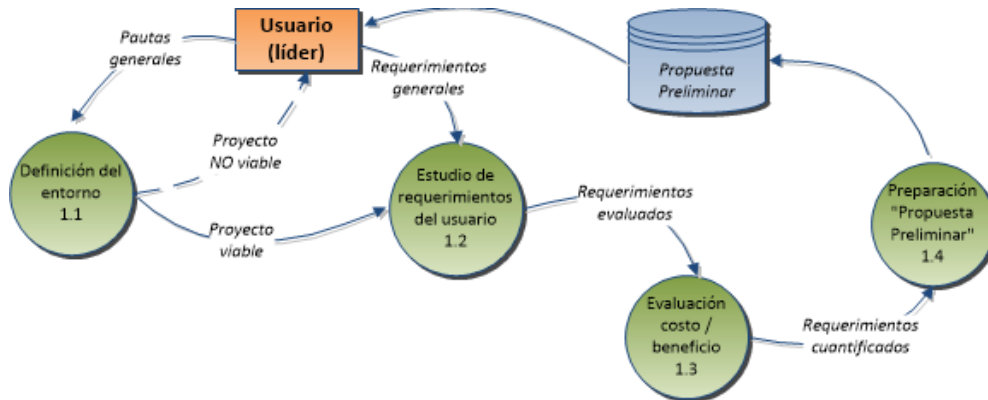
En cuanto al **equipo de trabajo**, la conformación del mismo dependerá del tamaño del proyecto, su complejidad tecnológica y la cantidad de meses en que se desee finalizar. Sin embargo, se puede tomar la siguiente tabla como una primera aproximación:

Cantidad de Horas (hasta)	Complejidad Tecnológica	Cantidad de meses (aprox.)	Recursos	
			Cantidad	Perfil
2000	Baja	4-6	4	1 líder de proyecto 1 analista funcional 2 programadores
	Media	6-8	5	1 líder de proyecto 1 analista funcional 3 programadores
5000	Media	8-10	6	1 líder de proyecto 1 analista funcional 1 diseñador 3 programadores
	Alta	10-12	7	1 líder de proyecto 1 analista funcional 1 diseñador 1 jefe de desarrollo 3 programadores
10000	Alta	12-15	9-12	1 líder de proyecto 1 analista funcional 1-2 diseñadores 1 jefe de desarrollo 4-6 programadores 1 testeador
15000	Alta	14-18	14-17	1 líder de proyecto 1-2 analistas funcionales 2-3 diseñadores 2 jefes de desarrollo (2 equipos) 6-8 programadores 2 testeadores
20000	Alta	20-24	19	1 líder de proyecto 2 analistas funcionales (2 equipos) 4 diseñadores 2 jefes de desarrollo (2 equipos) 8 programadores 2 testeadores

ACLARACION: El tipo de perfil “jefe de desarrollo” hace referencia a un analista programador especializado de nivel senior que tendrá un perfil técnico, siendo un programador especialista en el ámbito tecnológico en el que se desarrolle la aplicación en particular.

DIAGRAMA DETALLADO DE PROCEDIMIENTOS DE LA FASE 1

Para llevar a cabo esta fase, puede seguirse el siguiente esquema de procedimientos, teniendo en cuenta todo lo enunciado precedentemente en este capítulo.



1.1: Esta etapa se implementa a través de una reunión con el cliente en la cual deberá establecerse el entorno del sistema, cuyos elementos dependen de la arquitectura:

- Plataforma (sistema operativo)
- Hardware (servidores, terminales, dispositivos móviles)
- Tipo de red (topología, alcance, protocolo)
- Herramientas de desarrollo
- Motor de base de datos
- Arquitectura tecnológica de desarrollo (cantidad de capas y elementos que la conforman, frameworks o herramientas complementarias, esquema de distribución del producto, comunicación de los componentes, potenciales usuarios del sistema, planeamientos futuros)
- Requerimientos de seguridad, conectividad, velocidad, confiabilidad, etc.

1.2: Esta segunda etapa puede requerir una o dos reuniones, dependiendo de la complejidad y el alcance del proyecto. El objetivo es relevar en líneas generales los requerimientos del usuario. Paso seguido, se estudian los mismos a fin de determinar el alcance del proyecto y sus características sobresalientes como así también los principales circuitos administrativos.

1.3: Esta etapa se realiza sin la intervención del cliente. Su finalidad es determinar la cantidad y calidad de los recursos necesarios y el tiempo total del proyecto, con lo cual se puede cuantificar el mismo. Para realizar todo esto, es conveniente basarse en la experiencia personal, en la de analistas colegas y en proyectos anteriores que sean similares en alcance y tecnología utilizada, ya que en realidad se cuenta con demasiada poca información/tiempo como para hacer un estudio más detallado que brinde una estimación más precisa.

1.4: Este proceso se realiza sin la intervención del usuario. El objetivo del mismo es confeccionar la "Propuesta Preliminar", la cual reunirá toda la información recopilada e incluirá una serie de temas clave, los cuales ya se mencionaron anteriormente dentro de este capítulo.

FASE 2: ANÁLISIS Y DISEÑO GLOBAL

Esta fase está compuesta básicamente por dos etapas:

1. **Relevamiento:** el objetivo es relevar al más bajo y elemental nivel de detalle todos y cada uno de los requerimientos del sistema, a fin de poder realizar un diseño global del mismo.
2. **Diseño global:** su objetivo es desarrollar un modelo lógico del sistema, de forma tal que se pueda pasar a la siguiente fase sin más información que la contenida en dicho modelo. Es decir, que debe ser autosuficiente. No es necesario especificar en detalle el diseño de cada funcionalidad del sistema, pero sí deben identificarse claramente todas ellas.

Esta fase suele tener una **duración media** que puede variar entre cuatro y ocho semanas (de una a dos semanas para el relevamiento y el resto para el diseño global), dependiendo principalmente de la frecuencia de las entrevistas, de la claridad de los requerimientos solicitados, de la complejidad de la aplicación, de la capacidad y dedicación de todos y cada uno de los miembros del equipo, de la experiencia en aplicaciones similares, entre otros factores. Es recomendable tratar de calcular el tiempo de la etapa en base a la cantidad de entrevistas necesarias, ya que es más simple determinar el tiempo que requiere cada entrevista (incluyendo el relevamiento en sí y el análisis y documentación de los datos relevados).

Los **recursos implicados** en esta fase son todos los implicados de alguna forma con el proyecto, ya que es aquí donde hay que analizar a fondo la aplicación a desarrollar para no pasar por alto ningún aspecto que luego pueda ocasionar graves problemas en el diseño del sistema.

De esta etapa depende en gran parte el **éxito o fracaso** del proyecto. Por ello, se recomienda solicitar a la directiva del cliente que preste el mayor apoyo al proyecto y recomiende hacer lo mismo a cada uno de los subordinados que se encuentren implicados. Además de esto último, es aconsejable utilizar algunas técnicas que permitan poner al usuario final del lado del proyecto, ya que es lógico que el mismo se sienta un poco reticente a los cambios. Esto se logra:

- ✓ haciéndolo participe en las reuniones de relevamiento
- ✓ explicándole cómo va a ser el sistema
- ✓ comentándole las decisiones tomadas y el por qué (para que no se sienta excluido)
- ✓ pidiéndole opinión cada vez que sea oportuno
- ✓ tratando de entusiasmarlo con alguna de las funcionalidades del sistema que significarán para él una reducción en su carga de trabajo, la posibilidad de contar con cierta información que hasta el momento no tenía, una reducción en los tiempos de proceso, o alguna otra característica destacada

La **salida o cierre** de esta fase lo conforma el documento que denominaremos “**Análisis de Requerimientos**” (ver modelo en el documento *AnalisisReqModelo.docx*), el cual se utilizará como base de referencias para todo planteamiento posterior relacionado a las características cubiertas o no por el sistema. Dicho documento contendrá:

1 Aspectos generales.

- Definición del equipo del proyecto, indicando sus responsabilidades y etapas en las que participará.
- Especificación del programa de entrevistas realizado (conviene confeccionar este plan de antemano y entregarlo al cliente, pero en caso de no poder hacerse así, se confecciona esta parte a medida que se avanza en el relevamiento) indicando la fecha de la reunión, el tema tratado, los presentes y la minuta que resume los temas desarrollados.
- Definición de los servicios con que cuenta el área de sistemas del cliente (esto es muy importante porque seguramente estas áreas estarán involucradas en el proyecto de una u otra forma):
 - Infraestructura (responsables del hardware y centro de cómputos).
 - Administrador de base de datos.
 - Administrador de redes y comunicaciones.
 - Seguridad informática.
 - UX, usabilidad, control de procesos (o áreas similares).
 - Control de calidad.

2 Aspectos técnicos.

- Definición del entorno de desarrollo de la aplicación:
 - Herramientas de desarrollo (lenguajes, frameworks, herramientas complementarias de la arquitectura, generadores de informes, control de código fuente, documentación, seguimiento del proyecto).
 - Sistema operativo.
 - Motor de base de datos y ORM (en caso de utilizar).
 - Tipo de red (topología, protocolo, distribución física).
 - Características del servidor (o servidores).
 - Características de los puestos de trabajo o dispositivos móviles.
 - Cantidad, perfil y ubicación de los potenciales usuarios del sistema.
- Definición de la estructura de la aplicación:
 - Estructura general de la aplicación (tipo y formato de los componentes).

- Esquema de distribución de los distintos componentes del sistema.
 - Políticas de la empresa.
 - Metodología de desarrollo empleada.
 - Ciclo de vida del proyecto.
 - Esquema de seguridad interno del aplicativo.
 - Forma de acceso a los datos.
 - Integridad de la base de datos.
 - Herramientas de consulta de información.
 - Depuración de tablas, históricos, tracking, manejo de excepciones.
- 3 Aspectos lógicos.
- Definición de módulos y alcances.
 - Interfases con otros sistemas.
 - Modelo de datos del sistema.
 - Diagrama Entidad Relación del mismo.
 - Diccionario de datos (si hubiera).
 - Detalle de todas y cada una de las funcionalidades a incluir en el sistema (prototipo de menús).
 - Glosario de términos.
 - Diagramas de Casos de Uso, Clases, Componentes y Despliegue a fin de acotar claramente el funcionamiento de las principales funcionalidades.
 - Diagrama de Gantt conteniendo un plan minucioso y detallado de todas y cada una de las tareas a desarrollar desde el comienzo del proyecto hasta la puesta en producción del sistema.
- 4 Notas del Relevamiento.
- Anexo conteniendo cualquier especificación de diseño que haya sido entregada por el cliente.
 - Anexo conteniendo todas las minutas del relevamiento debidamente firmadas por los responsables del proyecto.

Se adjuntará a este “Análisis de Requerimientos” la “**Propuesta Definitiva**”, basada en la “Propuesta Preliminar” modificada, corrigiendo la conformación del equipo del proyecto, los tiempos de cada etapa por cada recurso, las fechas de entrega, los hitos principales, los costos en horas y las condiciones y formas de pago.

- Modelo de datos del sistema actual (si hubiera)
- Estructura de los datos a migrar al nuevo sistema (si corresponde)

Deberán realizarse reuniones de dos tipos: generales y detalladas. Las generales son por lo común con jefes de área y se tratan aspectos a grandes rasgos. Las detalladas deberán realizarse con los usuarios finales, que son por lo común los expertos que conocen las particularidades de cada proceso operativo.

En esta etapa hay dos aspectos fundamentales. El primero es tener en cuenta todas las recomendaciones referentes a las técnicas “Teory-W” y “Negociación Conveniente”, a fin de lograr el mayor provecho posible de las reuniones y tratar de evitar conflictos entre los miembros del equipo del proyecto. Para más detalles, referirse al documento *RGPI*. El segundo aspecto a tener en cuenta es mantener bajo control el *conjunto de prestaciones* del aplicativo, a fin de que los entrevistados no se desvíen y se pierda el objetivo principal de la aplicación, llevando los costos del proyecto a dimensiones incalculables. En este último caso es conveniente tomar nota de los requerimientos pero establecer que se realizarán en una siguiente versión del producto.

La duración de la entrevista no debería exceder las cuatro horas, ya que incluso los analistas bajarían su rendimiento debido al cansancio. Para realizar una reunión provechosa es aconsejable desarrollar un temario previo al comienzo de la reunión y entregar una copia a cada invitado a la misma, en lo posible vía mail para que conste la invitación a cada participante.

2.2: Una vez finalizada la entrevista, en lo posible el mismo día, deberán analizarse los datos recopilados y confeccionar una minuta de todos los temas tratados y las decisiones tomadas. Es conveniente realizar estas entrevistas por la mañana, así se podrá contar con toda la tarde para realizar el análisis de la misma. La mencionada minuta (tomar como modelo el documento *MinutaRelev.docx*) deberá detallar:

- Fecha de la reunión de relevamiento
- Participantes de la reunión
- Invitados ausentes
- Objetivo del relevamiento
- Enunciado de los temas tratados (temario)
- Resumen de la información recopilada y decisiones tomadas, recalcando especialmente aquellos puntos que puedan afectar el diseño del sistema
- Temas pendientes de definición y próximos pasos

Una vez analizada toda la información relevada y confeccionada la minuta correspondiente, es conveniente desarrollar el temario para la siguiente reunión, sin olvidar incluir en dicho temario un espacio al principio de la reunión para resumir la minuta anterior y aclarar cualquier duda que pudiera existir. Una vez hecho todo esto, deberá pasarse por mail a todos los participantes de la reunión la minuta desarrollada para que puedan leerla con detenimiento a fin de que puedan objetar cualquier aspecto en la siguiente reunión. También deberá pasarse por el mismo medio las invitaciones a la siguiente reunión y el temario de la misma. Es aconsejable que cada minuta esté

debidamente firmada por todos los participantes de cada reunión y las mismas se incluyan como un anexo al “Análisis de Requerimientos”.

- 2.3:** Es aconsejable iniciar este proceso luego de finalizada la etapa de Relevamiento, ya que de lo contrario no se poseerá información consistente en que basarse. Este proceso se realiza en paralelo con el **2.4**.

Basándose en las minutas del relevamiento y en toda la documentación recibida, deberá realizarse el análisis o diseño global del sistema y confeccionarse:

- Los diagramas de Casos de Uso (CU) para determinar el alcance funcional
- Diagramas de Clases (DC) para establecer el modelo de objetos
- Diagramas de Componentes (DCO) o de distribución intercapas (*layer diagrams*) para determinar la estructura lógica de la aplicación.
- Diagramas de Despliegue (DDE) para visualizar la distribución física de la solución.
- El modelo de datos (MD)
- El diagrama de Entidad-Relación (DER)
- Los diccionarios de datos de las tablas (DDT), campos (DDC) y funcionalidades (DDF)
- Diagramas de actividades (DA) o estados (DE) en los casos en que sea necesario aclarar un determinado proceso o algoritmo
- El glosario de términos

Este proceso es el núcleo de todo el ciclo de vida del proyecto. Si se hace un buen trabajo, el proyecto será casi seguramente exitoso, caso contrario, es muy probable que fracase. No deberá escatimarse tiempo en esta etapa. Es conveniente realizar un trabajo detallado y minucioso, sin presiones de ningún tipo.

Otro punto fundamental que debe llevarse a cabo en esta etapa es la estimación de los tiempos necesarios para realizar cada programa en particular. Al respecto, cabe mencionar que da muy buenos resultados invitar a cada analista y programador a realizar estimaciones independientes de tiempos de desarrollo para cada programa. Luego deberán compararse las mismas sacando un promedio, un máximo y un mínimo por cada tarea, discutir cada estimación y realizar un plan con el *valor esperado* de cada tarea (ver fórmula en el punto **2.6**). Tener presente que una mala estimación puede hacer ver como retrasado un proyecto que en realidad está dentro de los tiempos lógicos, simplemente está mal estimado.

- 2.4:** Esta actividad es complementaria a la del punto **2.3**. El objetivo principal de contar con el mapa del sistema (lo que sería el prototipo de menús) es el de verificar que no se pase por alto ninguna funcionalidad requerida, y que el usuario pueda ver en un lenguaje que pueda entender todos los aspectos que estarán incluidos en el aplicativo. Se destaca el carácter de *mapa* o *prototipo* y no de sistema de menús definitivo.

- 2.5:** Dado que el mayor riesgo para un proyecto es no poder cumplir con su planificación, es imprescindible realizar una eficiente gestión de riesgos de planificación. Dicha gestión deberá

identificar cada riesgo, medir su peso y probabilidad de ocurrencia, para así determinar su impacto en el sistema. Con dicha información deberá confeccionarse la lista de los principales riesgos del sistema. Los tiempos derivados de la presente lista deberán incluirse en el plan del proyecto, lo cual tendrá como consecuencia que se propongan distintas fechas de fin del proyecto:

- Mejor caso: se alcanzará esta fecha en el caso en que no se produzca ningún riesgo. Esta fecha debería ser la meta, aunque rara vez se pueda alcanzar.
- Caso esperado: es el caso realista, el cual asume que se producen un número razonable de desvíos.
- Peor caso: es el resultado de que la mayoría de los riesgos ocurran. Si se realizó un buen plan y una eficiente gestión del proyecto difícilmente se llegue a este caso.

Para más información sobre cómo realizar una “Gestión de Riesgos”, referirse al documento *RGPI*.

2.6: El objetivo de esta etapa es confeccionar un detallado *plan del proyecto*, que abarque desde el comienzo del proyecto (fase estudio de factibilidad) hasta la puesta en producción definitiva del sistema. La idea de incluir en el plan fases ya completadas al momento de entregar este documento (fases 1 y 2), es la de poder brindar al cliente un plan íntegro para que puedan tener una visión total del proyecto encarado.

Este plan de proyecto no se parecerá en nada al *diagrama estimado de tiempos* de la *Propuesta Preliminar*, sino que en este caso se tratará de un detallado *diagrama de Gantt*, el cual contenga el 100% de las tareas a realizar. No podrá realizarse ninguna tarea que no esté en el plan, y no debe omitirse en dicho plan ninguna actividad que deba llevarse a cabo. Un plan impreciso, incompleto o mal controlado es el primer paso para que un proyecto se salga de control. Es absolutamente imprescindible apegarse al plan durante todo el transcurso del proyecto, y no abandonarlo sin importar la presión ejercida por la directiva, ya que de esta forma se entraría en un caos y el proyecto quedaría fuera de control y nadie podría saber dónde está parado ni mucho menos cuanto falta para terminar.

Dado el ciclo de vida utilizado (entrega por etapas), la fase de desarrollo debe ser subdividida en una serie de entregas por etapas. Dichas entregas deberán tener una periodicidad promedio de 15 o 20 días, para lograr visibilidad y control. Los módulos críticos, más problemáticos o con requerimientos imprecisos deben ser los primeros en ser entregados para obtener una temprana retroalimentación que permita corregir cualquier desvío que pueda producirse. Además, con entregas tan frecuentes se consigue involucrar al usuario desde el principio, logrando que se ponga del lado del proyecto y no en su contra, ya que cada entrega irá acompañada de una capacitación y un testeado del módulo entregado. El único punto crítico a tener en cuenta es el de ser muy meticuloso al diagramar las entregas debido a las dependencias entre los distintos módulos.

En cuanto a la estimación de tiempos para cada tarea, en especial para las tareas de programación, se recomienda realizar varias estimaciones independientes (líder del proyecto, analista funcional, desarrolladores). Luego deberán unificarse para su análisis, a fin de determinar valores máximos, mínimos y promedio para cada tarea. Después de discutir y ratificar o rectificar cada una de ellas, se tomará el *valor esperado* resultante para la asignación del tiempo correspondiente:

$$VE_{(TAREA.X)} = \frac{4 * MEDIA + MAX + MIN}{3}$$

Cada entrada en el prototipo de menús conformará una tarea para la fase de Desarrollo, la cual deberá ser evaluada (en cantidad de horas) y asignada a algún recurso para su realización. Sin embargo, estas no son las únicas tareas a incluir en el *plan del proyecto*. A modo orientativo, se muestra en la siguiente tabla una serie de fases, etapas y subetapas a considerar en un plan de proyecto completo (puede tomarse un modelo de *PlanModelo.mpp*).

Tener en cuenta que el total de horas que arroje el proyecto deberá ser consistente con las horas presupuestadas y las fechas de entrega propuestas.

FASE/ETAPA/TAREA/SUBTAREA/HITO	DURACIÓN (DÍAS)	RECURSOS INTERVINIENTES
✓ FASE 1: ESTUDIO DE FACTIBILIDAD	4	
✓ Relevamiento del entorno tecnológico	0,5	LP,USR
✓ Relevamiento de los requerimientos generales	0,5	LP,USR
✓ Estudio de los requerimientos relevados	2	LP
✓ Preparación de la "Propuesta Preliminar"	1	LP
◆ Entrega de la "Propuesta Preliminar"	0	LP
✓ FASE 2: ANÁLISIS Y DISEÑO GLOBAL	20...60	
✓ Relevamiento	5...20	
✓ Relevamiento General	3	
✓ Definición del equipo del proyecto	0,5	LP
✓ Relevamiento de requerimientos generales	1	LP,AF,USR
✓ Relevamiento de aspectos tecnológicos	0,5	LP,AF,USR
✓ Preparación del "Programa de Entrevistas"	1	LP,AF
◆ Entrega del "Programa de Entrevistas"	0	LP
✓ Relevamiento Detallado	2...15	
✓ Relevamiento Submódulo 1...n (ciclo repetitivo)	2	
✓ Relevamiento general	0,5	LP,AF,USR
✓ Relevamiento detallado con el usuario	0,5	LP,AF,USR
✓ Análisis pormenorizado	1	LP,AF,USR
◆ Entrega de la Minuta	0	LP
✓ Diseño Global	15...50	
✓ Análisis de las minutas del relevamiento	2	LP,AF
✓ Definición de los casos de uso	5	LP,AF
✓ Definición del modelo de objetos (diagramas de clase, componentes y despliegue)	20	LP,AF
✓ Definición del modelo de datos		LP,AF
✓ Creación del DER		LP,AF
✓ Creación de los DD y glosarios		LP,AF
✓ Preparación del prototipo de menús	1	LP,AF
✓ Preparación del GANTT del proyecto	2	LP
✓ Preparación del "Análisis de Requerimientos"	3	LP
◆ Entrega del "Análisis de Requerimientos"	0	LP
✓ FASE 3: DESARROLLO	40...200	
✓ Preparación del Entorno	8	
Implementar modelo de datos	2	AF,DBA
Replicar entorno global del sistema	1	AF,PR
Preparar el prototipo de menús	1	PR
Adaptar rutinas generales	2	PR
Adaptar esquema de seguridad	2	PR
✓ Desarrollo Submódulo 1...n (ciclo repetitivo)	15...30	
Diseño detallado (especificaciones de diseño)	2...5	AF
✓ Codificación	5...15	
Formulario 1	1...10	PR
Formulario n	1...10	PR
Documentación técnica	1	AF

	Documentación de usuario	1	AF
	Control de calidad	1	LP,AF
	Ajustes	1...3	PR
◆	Entrega del submódulo	0,5	AF,PR
	Dictado de capacitación al usuario	0,5	LP,AF,USR
	Vuelco de datos en entorno de prueba	0...5	AF,PR
	Pruebas por parte del usuario	2...5	USR
◆	Aprobación de funcionalidad	0	USR
	Prototipo del Sistema	1...?	
	Compilación de los componentes	1	PR
	Preparación del servidor de aplicaciones	1	AF,USR
	Potenciales riesgos de la planificación	0...40	
◆	<i>Instalación del Prototipo del Sistema (mejor caso)</i>	0	LP,AF,PR
◆	<i>Instalación del Prototipo del Sistema (caso esperado)</i>	0	LP,AF,PR
◆	<i>Instalación del Prototipo del Sistema (peor caso)</i>	0	LP,AF,PR
	FASE 4: PUESTA EN PRODUCCIÓN	20...?	
	Vuelco de datos definitivo en entorno de prueba	1...10	AF,PR
	Pruebas integrales del sistema	10...20	USR
✕	Ajustes al Sistema	?	
	A determinar	?	AF,PR
✕	Nuevos Requerimientos	?	
	A determinar	?	AF,PR
◆	<i>Aprobación definitiva de la funcionalidad</i>	0	USR
	Puesta en Marcha	4...15	
	Preparación del ambiente de producción	2	AF,USR
✕	Vuelco de datos definitivo en producción	1...10	AF,USR
	Instalación del sistema en producción	1	AF,PR
◆	<i>Fin del Proyecto</i>	0	LP,USR

ACLARACIONES:

La duración es aproximada y corresponde a días hábiles de 8 horas de trabajo. Cabe aclarar que hay tareas cuya duración varía dependiendo de cada proyecto. En dichos casos se fijó un intervalo orientativo (desde...hasta). Por otro lado, existen ciertas tareas que son imposibles de estimar hasta alcanzar cierta etapa dentro del proyecto. En estos casos se indicó un signo de interrogación (?) en el tiempo estimado.

✓ significa que la tarea o etapa ya fue completada.

◆ significa que se trata de un hito.

✕ significa que se recomienda dejar esta tarea fuera de presupuesto, ya que no se puede estimar.

LP representa al líder de proyecto de la consultora.

AF representa al analista (o analistas) funcional de la consultora.

PR representa a un recurso de programación

DBA representa al administrador de DB del cliente.

USR representa a los usuarios del sistema (líderes, usuarios finales y de sistemas).

Al confeccionar el plan hay que tener muy en cuenta las dependencias entre las distintas tareas.

2.7: Esta tarea deberá necesariamente realizarse al finalizar todas las anteriores. La misma consiste en recopilar toda la documentación anteriormente confeccionada y armar los dos documentos que conformarán el cierre de esta fase, los cuales ya fueron mencionados anteriormente dentro de este capítulo:

- El “Análisis de Requerimientos”
- La “Propuesta Definitiva”

FASE 3: DESARROLLO

Esta fase es la de mayor duración de todo el proyecto. La misma se subdivide en varias etapas bien diferenciadas, sin embargo con un objetivo en común: **diseñar, codificar, probar e implementar el sistema** analizado en las etapas anteriores. Para este ciclo se utilizará una metodología ágil (*scrum* es la más adecuada por el momento), ya que es posible que los requerimientos sufran cambios y se deben conseguir entregables en plazos muy cortos, con tareas y asignaciones cambiantes hasta cierto punto. Los pasos a ejecutar en cada iteración o por cada entregable a conseguir son:

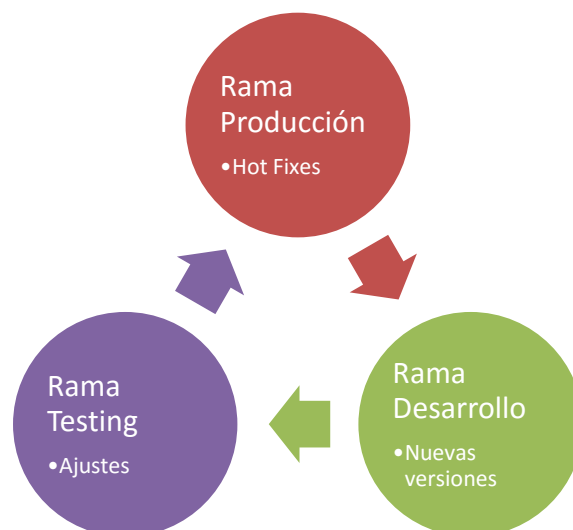
1. **Diseño Detallado:** el objetivo es confeccionar, en base al “Análisis de Requerimientos” un documento que sirva a modo de manual técnico detallado (“Especificación de Diseño”) para cada funcionalidad que los programadores tengan que desarrollar. Dependiendo de la arquitectura, es posible que esto incluya el maquetado de la pantalla asociada a la funcionalidad en cuestión.
2. **Desarrollo:** consiste en la codificación de cada funcionalidad especificada. Incluye además las pruebas básicas unitarias detalladas en la “Metodología de Programación”.
3. **Control de Calidad:** su objetivo es llevar a cabo las pruebas unitarias y de integración de todos los componentes del submódulo, como así también ingresar el lote de pruebas que el usuario deberá realizar (casos de aceptación).
4. **Implementación:** tiene el objetivo de desplegar el sistema para que sea accesible desde los equipos destinados a realizar las pruebas de usuario, capacitar a los mismos y preparar el entorno para que puedan ejecutarse las pruebas de aceptación.

A los efectos de poder trabajar en forma dinámica, flexible y remota, se utilizará alguna herramienta de control de código fuente (por ejemplo GIT) utilizando ramas (*branches*) para manejar al menos tres versiones:

1. **Producción:** es la rama que contiene la versión de los fuentes que están operativos. Esta rama se usa para generar *fixes* por *bugs*.
2. **Testing:** es la rama que contiene los fuentes de la versión de pruebas de usuario.
3. **Desarrollo:** es la rama que actualmente tiene los fuentes en desarrollo.

Las tres ramas podrían avanzar en paralelo y, cada vez que se cierra un ciclo, se realiza un *merge* y se pasa de desarrollo a testing, de testing a producción y de producción a desarrollo nuevamente.

Con este sistema, se mantiene un tracking de cada *release* liberada y los fuentes asociados a la misma.



Es muy difícil establecer la **duración promedio** de esta fase, ya que varía enormemente de un proyecto a otro. Sin embargo, se puede dar como parámetro de referencia una duración mínima de un mes y hasta quizás un año. En caso de manejarse proyectos con una duración mayor a un año, es aconsejable realizar una división en subproyectos, manejados con un equipo independiente c/u de ellos.

Los **recursos implicados** no son los mismos en todas las etapas. Sin embargo, cabe destacar que todo el equipo del proyecto participa en la fase, en menor o mayor grado. En la especificación de proceso del diagrama de procedimientos se aclara en cada etapa los recursos intervinientes.

Cabe destacar en este capítulo la importancia de **apegarse al plan del proyecto**. En caso de algún atraso en alguno de los hitos de control, deberán analizarse los motivos de tal eventualidad, corregirse el o los errores y en caso necesario, replanificar. En este último caso, es probable que amerite realizar alguna reunión con el cliente para discutir el tema. Jamás hay que abandonar el plan por más presión que exista, ya que la alternativa es el caos total.

Durante esta fase es donde se producirá el mayor número de desvíos, por innumerables razones. Lo que hay que hacer es una muy **eficiente gestión del proyecto**, siguiendo de cerca los **riesgos de la planificación**, verificando si se produjo alguno de ellos o bien quizás algún otro riesgo que no se tuvo en cuenta. Ante cualquier eventualidad, el procedimiento apropiado es el descrito en el párrafo anterior. Es conveniente actualizar la tabla de “*Riesgos de la Planificación*” a medida que los mismos se vayan produciendo, como así también tomar nota de los mismos en la “*Planilla de Seguimiento del Proyecto*”. A continuación, se muestra un ejemplo de ambas tablas o planillas:

1. Tabla de Riesgos de la Planificación (medidas expresadas en hs/proyecto)

Id	Riesgo	Plan de Prevención	Probab. de Ocurrencia (%)	Magnitud de la Pérdida (sem.)	Impacto del Riesgo (sem.)	Impacto real (sem.)
1	Nuevos requerimientos	Crear un grupo de control de cambios que apruebe únicamente los requerimientos necesarios.	50	3	1,5	2,43
2	Planificación demasiado optimista	Hacer varias estimaciones individuales, compararlas y discutir las con todo el equipo.	40	2	0,8	0,25
Total de semanas de retraso:				5	2,3	2,68

La última columna de la tabla anterior es la que se va actualizando durante el control de la gestión de riesgos del proyecto. Como puede verse en el ejemplo, en cuanto al riesgo “1”, su impacto real superó el planificado. Esto debe tenerse muy en cuenta para nuevos proyectos.

2. Tabla de Seguimiento del Proyecto (medidas expresadas en hs/hombre)

Fecha Ocurrencia	Categoría (RI, EC, AA, OB)	Impacto Aprox. (hs/hombre)	Descripción de la Desviación	Plan de Acción
11/10/2000	RI 8	2	Tareas no contempladas en el plan (creación del menú general del sistema).	Se hizo utilizando las horas consideradas para pruebas y ajustes para la entrega, por lo tanto no generó desfasajes, pero la tarea debería haberse incluido.
25/10/2000	EC	0	Poco tiempo para el control de calidad	En la primera entrega se controló todo con dos días de tiempo para los ajustes. En la segunda y tercer entrega, quedó menos de un día para los ajustes. Aunque salió todo relativamente bien, esto puede traer

				problemas de calidad inaceptable. Por otro lado, prácticamente todos las funcionalidades necesitan uno que otro ajuste. Se tendrá en cuenta para las siguientes entregas.
25/11/2000	OB	0	Entregas en tiempo y forma	En las últimas dos entregas (la 4ta. y 5ta.) se corrigió ese corto margen del que siempre se disponía al hacer una entrega. En estos casos se llegó a tiempo y con holgura para corregir cualquier problema de calidad. Por otro lado, la calidad general de las funcionalidades aumentó, quizás por la familiarización de los recursos con las herramientas.
Total desfasaje acumulado:		2		

Cada vez que se realiza una anotación en esta planilla, debe tomarse nota de la equivalencia en la planilla anterior. Tener en cuenta que debe transformarse el impacto de *hs/hombre* a *hs/proyecto*. Esto debe hacerse teniendo en cuenta el número de recursos implicados en el desvío en el momento en que este se produjo. En cuanto a la categoría, se utilizan los siguientes mnemotécnicos:

- ✓ RI x: riesgo número x, referenciado en la planilla de “*Riesgos de la Planificación*”
- ✓ EC: error clásico cometido
- ✓ AA: aviso de alerta para tener en cuenta (similar a un error clásico)
- ✓ OB: observación destacada, generalmente favorable

Se recomienda prestar especial atención a los pedidos de **nuevos requerimientos**, ya que el ciclo de vida utilizado (entrega por etapas con scrum) invita a ello. Lo que hay que hacer es explicarle al cliente cómo afectan los nuevos requerimientos al plan, a los tiempos y a los costos. En caso de que los mismos sean aceptados de cualquier forma, hay que incluir los requerimientos en el plan y tomar nota de tales sucesos (incluso puede redactarse un informe de incidencia), ya que los mismos seguramente traerán alguna consecuencia cuando no se alcancen las fechas o se superen los costos estipulados en la propuesta original. Si es posible, se recomienda incluir en el plan los nuevos requerimientos al final del hito de “*Fin de Proyecto*”, en una 5ta. Fase, denominada “*Mantenimiento*”.

Se recalca la importancia de la **toma de medidas** en la evolución del proyecto. Es fundamental contar con documentación escrita sobre el resultado del proyecto, tomando notas sobre:

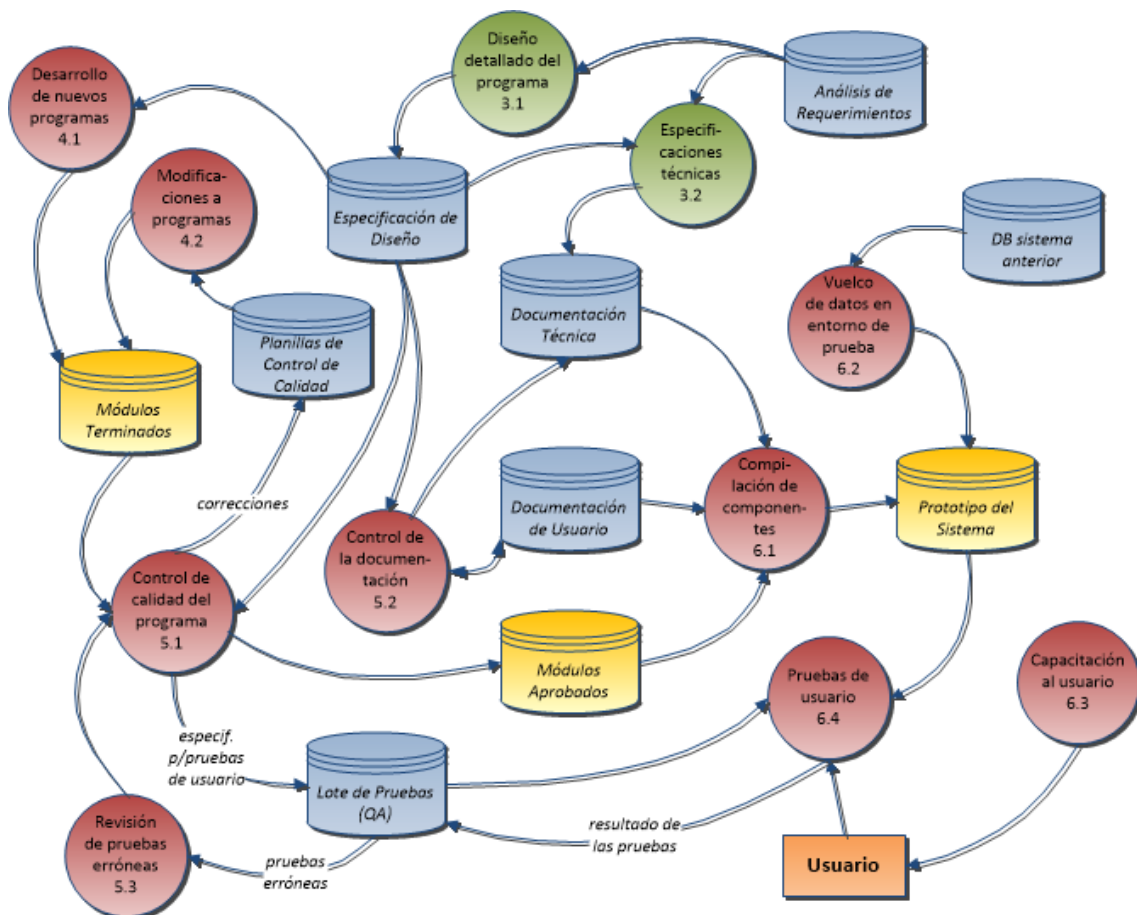
- ☞ Tiempos (de desarrollo, tarea, fase o etapa)
- ☞ Costos (totales y por etapa)
- ☞ Cumplimiento del plan
- ☞ Desvíos producidos (impacto, motivo del desvío, acciones tomadas para controlarlo)
- ☞ Riesgos ocurridos y errores cometidos (impacto, origen, medidas tomadas p/su solución)
- ☞ Beneficios detectados

Esta información servirá para realizar un mejor proyecto en la próxima oportunidad. La mejor forma de tomar medidas con respecto a los tiempos de desarrollo, es en base a las especificaciones de diseño. En cuanto a las medidas con respecto al cumplimiento o desvíos en el plan, tiempos y costos de cada

etapa, se recomienda incluir el Gantt del proyecto en el software de control de código fuente. De este modo, siempre se podrá comparar el plan original y sus subsiguientes alteraciones. Para tomar medidas con respecto a los riesgos ocurridos, errores cometidos, desvíos y beneficios detectados, lo mejor es anotar cada uno de ellos en la “Planilla de Seguimiento del Proyecto”, la cual debe ser actualizada a medida que se van gestionando los riesgos de la planificación. Toda esta información se utilizará al finalizar el proyecto para confeccionar un documento de “Evaluación del Proyecto”, cuya finalidad es resumir el resultado del mismo con el objetivo de poder tener una visión objetiva de los logros alcanzados y los errores cometidos (ver modelo en el documento *EvaluacionProyecto.doc*).

La **salida o cierre** de esta fase no es ni más ni menos que el “**Prototipo del Sistema**”. Se entiende por *prototipo* a un aplicativo 100% funcional y que cumple con todos los requerimientos documentados en el “Análisis de Requerimientos”. El carácter de *prototipo* se debe a que el mismo está sujeto a testeos y ajustes para la aprobación definitiva del sistema.

DIAGRAMA DETALLADO DE PROCEDIMIENTOS DE LA FASE 3



Para llevar a cabo esta fase, puede seguirse el siguiente esquema de procedimientos, teniendo en cuenta todo lo enunciado precedentemente en este capítulo.

ACLARACION: Se describirán los procesos en el orden lógico de ocurrencia de cada uno, que no necesariamente coincide con la numeración de los mismos.

3.1: Esta etapa consiste en la creación de la “Especificación de Diseño” de cada programa (formulario, consulta, listado, proceso, rutina o método) a desarrollar. Para confeccionar la misma, deberá utilizarse la plantilla modelo para cada tipo de programa a diseñar (EDabm_blanco.docx,

Edbr_blanco.docx, EDproc_blanco.docx, EDrut_blanco.docx, EDlist_blanco.docx, EDcons_blanco.docx). La base necesaria para su creación podrá obtenerse en el “Análisis de Requerimientos” (especificación de procesos, diagramas de casos de uso, diagramas de clase, DERs, diccionario de datos, Gantt del proyecto, minutas del relevamiento, etc.). Esta especificación debe ser autosuficiente para que un recurso con el perfil adecuado y el conocimiento de la “Metodología de Programación”, pueda desarrollar el objeto especificado sin necesidad de información adicional. Dependiendo de la arquitectura utilizada, es posible que esta especificación cuente además con un wireframe o maquetado de la funcionalidad a desarrollar. No se deberá dar nada por obvio, todo deberá anotarse en la especificación. La misma deberá subirse al repositorio de control de código fuente y asignarse la tarea al recurso de programación que la llevará a cabo. En esta etapa interviene pura y exclusivamente el analista funcional y los diseñadores (si hubiera).

- 3.2:** Esta etapa se realiza en conjunto con la anterior. Su objetivo es crear o actualizar la documentación técnica del sistema. Dicha documentación podrá ingresarse en diferentes sitios dependiendo del objeto a documentar (información relativa a los diccionarios de datos en línea, manual de procedimientos, información técnica específica, etc.).
- 4.1:** Con la especificación de diseño y la documentación anexa, el programador (único recurso implicado en esta etapa) deberá codificar y probar el objeto especificado. Es recomendable realizar una reunión entre el analista a cargo de la ED y el programador asignado para estudiar la funcionalidad a desarrollar y evitar malos entendidos. Las pruebas a realizar son las denominadas “pruebas unitarias” de cada programa. Deberán respetarse todas las reglas de programación estipuladas en el documento “Metodología de Programación”. En el caso de proyectos grandes, es muy probable que exista un programador jefe, analista técnico y/o funcional que deberán seguir estrechamente el desarrollo de cada funcionalidad. Una vez que se termine con un programa, listado o rutina en particular, deberán anotarse las horas insumidas y cualquier otro dato solicitado en las especificaciones de diseño y pasar la tarea a la siguiente etapa en el software de control de avance del proyecto.
- 5.1:** Una vez que se vayan terminando las funcionalidades del módulo o submódulo, el analista funcional encargado del mismo deberá realizar las denominadas “pruebas de integración” (ver más información al respecto en la “Metodología de Programación”). Las mismas deberán basarse en la especificación de diseño. Como resultado de estas pruebas deberán registrarse todos los *issues* detectados y asignárselos al desarrollador para su resolución. Tener en cuenta que en caso de detectar un error de diseño y no un error de codificación, lo que deberá hacerse es modificar la especificación de diseño correspondiente (utilizando el *control de revisiones*), y asignarse dicha especificación al recurso que deberá efectuar las correcciones. Además, tendrán que ingresarse las especificaciones correspondientes al lote de pruebas que el usuario tendrá que testear cuando llegue el momento.
- 5.2:** Paralelamente al control del programa desarrollado, deberá controlarse la documentación de usuario del mismo. En caso de que la misma se encuentre incompleta o desactualizada, el mismo analista tendrá la tarea de completarla. Deben tenerse muy en cuenta las modificaciones a las especificaciones de diseño y el contenido de las planillas de control de calidad, ya que ambos elementos pueden influir en el proceso de desactualización de la documentación, tanto técnica como de usuario.
- 4.2:** Tomando las planillas de control de calidad (*issues* o incidentes reportados) o las especificaciones de diseño modificadas (con el control de correcciones habilitado), el programador deberá realizar los cambios especificados.

- 6.1:** Una vez codificados y aprobados todos los componentes del módulo o submódulo a entregar, los mismos deberán “compilarse” a fin de crear los correspondientes entregables. Dichos entregables pueden componerse de uno o varios ejecutables y librerías adjuntas, además de una serie variable de manuales de usuario, técnicos y de procedimientos. El encargado de realizar esta etapa, es por lo general uno de los analistas programadores en estrecha relación con el analista técnico o funcional a cargo del módulo. Dicho módulo deberá desplegarse en el entorno que se haya definido para realizar las pruebas de usuario. Es conveniente acompañar la entrega por un **“Acta de Entrega”** (ver modelo en el documento *ActaEntrega.docx*), en la que constan los detalles del módulo entregado, el responsable de las pruebas y los plazos establecidos para las mismas. Dicha acta deberá ser firmada por los responsables de la consultora y el cliente.
- 6.2:** Una vez instalado el módulo en el ambiente de prueba, deberán migrarse o ingresarse manualmente los datos necesarios para lograr que el módulo entregado sea funcional a fin de que el usuario pueda testear el mismo. Se recomienda tener muy en cuenta el impacto de la mencionada carga inicial dentro del plan del proyecto, ya que insumirá un valioso tiempo de los analistas, programadores y usuarios.
- 6.3:** En esta etapa, el líder de proyecto y/o el analista funcional deberán dictar la capacitación al usuario, siempre referente al módulo entregado y no a todo el sistema. Es conveniente, aunque no estrictamente necesario, que se haya finalizado el proceso **6.2** antes de dictar la mencionada capacitación.
- 6.4:** En esta etapa interviene pura y exclusivamente el usuario. El mismo tendrá la tarea de tomar cada uno de los casos de aceptación (lote de pruebas) del módulo, ejecutarlos y anotar sus resultados, junto con cualquier comentario o sugerencia pertinentes.
- 5.3:** Una vez que el usuario vaya terminando con los testeos, el analista debe recopilar los datos de las pruebas de usuario que arrojen resultados erróneos y registrar los correspondiente *issues* (planillas de control de calidad) o bien modificar las planillas de diseño (si correspondiera a un error de diseño y no de codificación), según se especificó en el proceso **5.1**.

FASE 4: PUESTA EN PRODUCCIÓN

El objetivo de esta fase es poder realizar todas aquellas tareas que sean necesarias para **poner en producción el sistema**. Básicamente, las mismas consisten en realizar la migración definitiva de los datos o carga inicial, las pruebas de integración o paralelo, y los ajustes que sean indispensables para la puesta en funcionamiento del sistema. El uso de *scrum* es muy recomendable en esta etapa, ya que cada iteración tendrá por objetivo obtener una versión final del sistema para que pueda ponerse en producción.

La **duración** de esta fase dependerá de lo extensas que sean las pruebas, de la flexibilidad en el diseño del sistema para poder adaptarse rápidamente a los cambios solicitados, y de la cantidad de cambios que sean aprobados para realizarse dentro de esta etapa. Como media, se puede tomar una duración estimada de un mes.

Los **recursos involucrados** en esta fase son todos los implicados de alguna forma con el proyecto, especialmente el usuario, excepto quizás algunos recursos de programación que tal vez ya no sean requeridos.

Al usuario le corresponde la **aprobación definitiva de la funcionalidad** del sistema, a fin de tomar la decisión de ponerlo o no en producción. Al respecto, es conveniente aclarar que cuando decimos *usuario*, este término puede representar a varias personas de diferentes sectores. Por ello, es muy probable que en algunos usuarios se encuentre cierta resistencia al cambio, lo cual se debe a innumerables razones, las cuales no viene al caso mencionar. Lo importante es decir como disolver esa resistencia:

- ✓ Hacer partícipes en el proyecto a los usuarios clave (los más experimentados y con mayor influencia en el área) desde el principio, logrando que actúen como defensores del sistema y no como acusadores, contribuyendo a buscar soluciones y no problemas.
- ✓ Realizar entregas parciales de los módulos a todos los sectores, sin dejar ninguno totalmente aislado para el final, a fin de que no se sientan relegados.
- ✓ Dictar una muy buena capacitación, acompañada de ser posible de un manual de procedimientos, con el fin de eliminar las dudas y miedos que el usuario pueda tener.
- ✓ Guiar, controlar y apoyar a cada usuario encargado de realizar las pruebas de integración y/o paralelo para que no se sienta abandonado y por consiguiente aumente su miedo a comenzar a utilizar un nuevo sistema.
- ✓ Solicitando a la directiva que nos dé su apoyo, y en caso necesario, que presione a los usuarios para que hagan todo lo necesario para la implementación del nuevo sistema.

Cabe destacar que en esta fase corresponde realizar todos los **ajustes** al sistema que sean necesarios para que el mismo pueda ponerse en producción. También es muy común que aquí aparezcan varios **nuevos requerimientos** o cambios en los mismos. Al respecto, es conveniente tener en cuenta las siguientes recomendaciones:

- ☞ Es probable que esta etapa se utilice no sólo para realizar los ajustes que el usuario solicite, sino también para pulir cosas que hayan quedado en el tintero. Por ello, es imprescindible no subestimar la etapa y contar con varios recursos de programación.

- ☞ Hay que clasificar muy bien los pedidos de los usuarios en:
- Ajuste necesario por error del sistema
 - Cambio de requerimiento necesario para la operatoria normal del sistema
 - Nuevo requerimiento necesario para la operatoria normal del sistema
 - Ajuste o nuevo requerimiento para lograr una mejora en el sistema
 - Cambio de requerimiento no imprescindible
 - Nuevo requerimiento no imprescindible

El primero de los ítems debe realizarse dentro de la etapa de “Ajustes”. Los dos siguientes, corresponde realizarlos en la etapa de “Nuevos Requerimientos”. Ambas etapas deben calcularse desde el principio del proyecto en base a una cantidad máxima de horas que se utilizarán para dichas tareas. Una vez completado el cupo de horas disponibles, no se aceptarán más cambios de requerimientos, a no ser que sean realmente imprescindibles. En este último caso, habrá que hacer una reunión con la directiva para tratar el tema y tomar una decisión. Tener muy en cuenta que esta fase es muy difícil de controlar, y por lo general, es aquí donde se produce un número tal de desvíos que consigue atrasar la puesta en producción varias semanas o incluso meses.

Para resolver lo expresado en el párrafo anterior, se recomienda incluir después del hito de “Fin de Proyecto”, una **FASE 5: MANTENIMIENTO**. Se incluirán dentro de esta fase todos los nuevos requerimientos o cambios en los mismos. De esta forma, se podrá conseguir terminar con el proyecto dando un cierre poniéndolo en producción y entrar en una nueva etapa de mantenimiento, lo cual consigue reducir la presión y dar tranquilidad a la directiva de que el proyecto llegó a su término en tiempo, forma y costos.

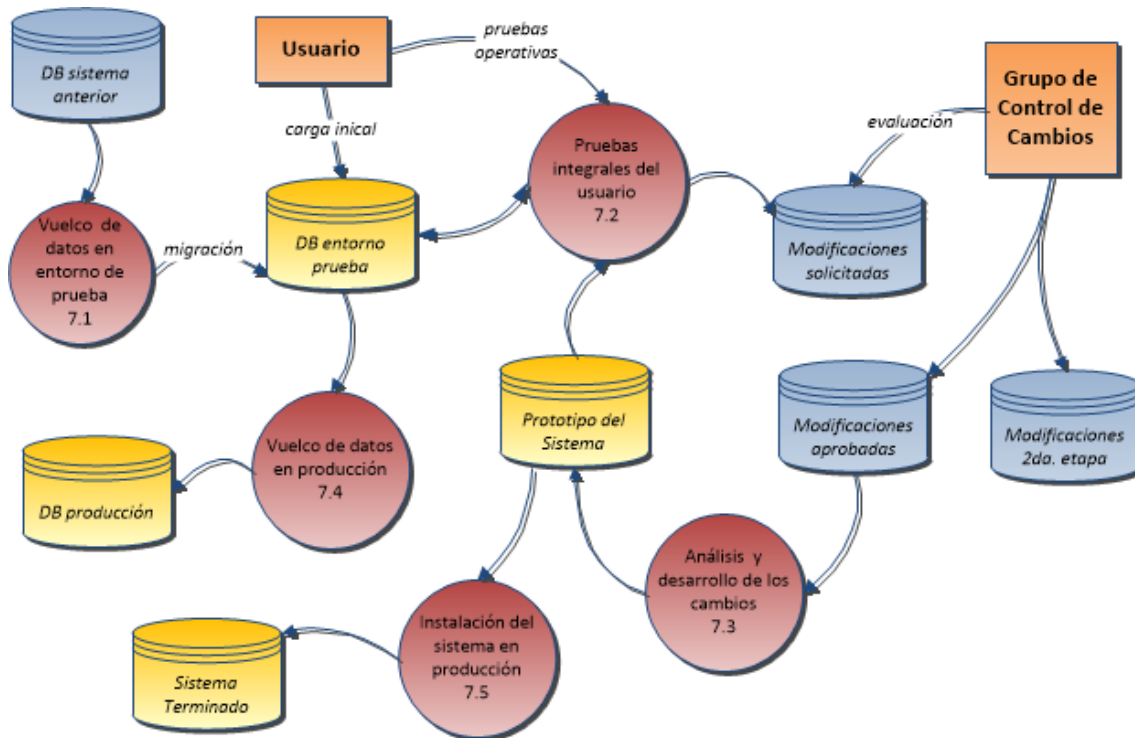
El encargado de evaluar cada uno de los cambios solicitados, clasificarlo y categorizarlo, es el **Grupo de Control de Cambios**. Referirse al documento *RGPI* para informarse al respecto.

Es muy importante para el cliente (no sólo para el usuario final), poder contar en esta etapa con el **apoyo del equipo del proyecto**: el líder controlando que todo salga bien y manejando las prioridades, el analista resolviendo los problemas eventuales (de datos y programa) y los programadores codificando las soluciones lo antes posible. Este apoyo debería negociarse de entrada, con respecto a su duración y costos. Sin embargo, se puede estimar una duración aproximada que puede oscilar entre las dos y cuatro semanas.

La **salida o cierre** de esta fase es nada más ni nada menos que el “**Sistema Terminado**”.

DIAGRAMA DETALLADO DE PROCEDIMIENTOS DE LA FASE 4

Para llevar a cabo esta fase, puede seguirse el siguiente esquema de procedimientos, teniendo en cuenta todo lo enunciado precedentemente en este capítulo.



7.1: Esta etapa consiste en realizar el vuelco de datos a la base del entorno de prueba, a fin de que los usuarios puedan realizar las pruebas operativas del sistema. Dicho vuelco puede originarse en una migración a partir de la base anterior, en cuyo caso ya debería haberse considerado y analizado el tema. Al respecto, siempre es conveniente solicitar que el cliente realice una exportación (como sea) de los datos que le solicitamos a un determinado formato y que la consultora los importe. En ningún caso es recomendable meterse en la base anterior y sacar la información, ya que pueden cometerse innumerables errores. Paralelamente, es común que ciertas tablas deban ser cargadas a mano por los usuarios. En general, lo que se intenta migrar en forma automática son aquellos datos que son demasiado voluminosos. Tampoco es conveniente migrar los archivos históricos (al menos no en esta etapa), ya que suelen tener numerosos errores.

7.2: En esta etapa interviene casi exclusivamente el usuario, si bien el equipo de desarrollo deberá prestarle un gran apoyo. Las pruebas se realizarán con la base migrada y el prototipo del sistema terminado. Como resultado de las mismas, se genera una lista (probablemente muy numerosa) de cambios a realizar. Dicha lista debe ser evaluada por el grupo de control de cambios, clasificando los requerimientos básicamente en dos planillas diferentes:

- Modificaciones aprobadas: deben ser realizadas antes de la puesta en producción definitiva.
- Modificaciones 2da. etapa: entran en el plan dentro de la fase 5 (mantenimiento post-producción).

De la exhaustividad de las pruebas realizadas dependerá lo bien que pueda llegar a salir la puesta en producción definitiva.

- 7.3:** El analista funcional y el líder de proyecto deberán priorizar, analizar y diseñar cada modificación a realizar, para entregársela al equipo de desarrollo, quién deberá codificar la solución, probarla y devolverla al usuario para su testeo. Hay que tratar de acotar al máximo esta etapa, ya que la misma se puede prolongar indefinidamente.
- 7.4:** En cuanto el líder de usuarios apruebe la funcionalidad operativa del sistema, deberán tomarse las medidas correspondientes para su puesta en producción. El primer paso es realizar la migración definitiva de los datos en la base de datos de producción. Esto puede ser relativamente complejo debido a políticas de seguridad informática, espacios en los discos de producción, carga de trabajo de los servidores, etc. Por ello, es aconsejable evaluar todos estos temas con la suficiente antelación. En cuanto al origen de los datos a migrar, se puede optar por dos caminos:
- Utilizar la base de datos empleada para las pruebas, con lo cual hay que eliminar con seguridad todos los datos de las pruebas realizadas. Por lo general, esto puede hacerse únicamente si las pruebas fueron relativamente sencillas y con un éxito considerable.
 - Volver a migrar todo desde la base de datos anterior, tal como se hizo con la base de pruebas.
- 7.5:** Paralelamente a la migración, debe tomarse el prototipo del sistema modificado e instalarse en el servidor de producción. Debe tenerse en cuenta que es posible que las terminales que vayan a hacer uso del sistema no sean las mismas que las utilizadas durante las pruebas, incluso puede ocurrir que se haya planificado poner nuevo hardware el cual todavía no está disponible. Por ello, debe contarse con un colchón de tiempo razonable para realizar esta tarea.

iT South Consultores

Av. Alicia Moreau de Justo 740 Piso 3º Of. 1

C1107AAN Capital Federal, Argentina

Teléfono +54 (11) 4334-0033 / 4343-3157

Email: info@itsouth.com.ar

www.itsouth.com.ar

