

**1.0 Introducción**

El objetivo de este capítulo orientado a la Dirección y Gestión de Proyectos Software es introducir la necesidad del uso de Técnicas y Herramientas de Ingeniería del Software, entre las que se incluyen Estándares de Dirección de Proyectos y Metodologías para el Desarrollo de Sistemas Informáticos, en el mundo de la empresa.

Para ello, y tras analizar algunas definiciones de Ingeniería del Software para centrar el tema, se aborda el estudio de dos metodologías con distinta orientación, aunque similares en sus objetivos finales: la implantación de un “modus operandi” que permita eliminar costes de mantenimiento de las aplicaciones desarrolladas.

La primera, una Metodología clásica de Gestión de Proyectos, ataca el problema por dos vías diferentes: la introducción de una serie de Estándares, no solo para el desarrollo de proyectos, sino de ámbito general en toda la Instalación, y una segunda vía consistente en la descomposición jerárquica de un Proyecto en Fases, para simplificar su evolución y realizar su control.

En nuestra opinión, toda buena Metodología debe de cubrir no solo la etapa de desarrollo propiamente dicho de un proyecto, sino que además, debe de abarcar todo el ámbito de la Empresa, facilitando el desarrollo de estándares que engloben todo el conjunto de las labores diarias, puramente informáticas o no, que se llevan a cabo en un Centro de Proceso de Datos. Más que una serie de normas de obligado cumplimiento, se debe de intentar que la Metodología sea una cierta mentalidad existente, compartida por todo el personal de la Organización. Hay que saber crear el ambiente necesario de colaboración y lograr las motivaciones oportunas hasta poder sentir que el método es bueno y beneficioso para todos.

Por lo tanto, y para dar ese ámbito general que se pretende, hemos dividido la

---

Metodología en cinco grandes Etapas: Una primera parte dedicada a la *Organización* necesaria para crear el ambiente que se desea y soportar la metodología, una segunda parte de *Planificación*, donde estudiaremos cuidadosamente lo que debe de hacerse y cómo debe de hacerse, una etapa de *Control*, obviamente necesaria en cualquier metodología, que nos indique cómo vamos y qué bien lo estamos haciendo, la etapa del *Desarrollo* propiamente dicho de nuestro Proyecto y por último, una parte que cubra la *Documentación* necesaria y suficiente en el desarrollo de nuestras Aplicaciones.

La segunda Metodología estudiada, *PDM80 (Prototype Development Methodology 80)*, desarrollada en Los Angeles, California, por Computomata Intl. Co., está orientada al desarrollo de prototipos y apoyada en tecnología de Base de Datos, en la que se concentra un gran esfuerzo en la involucración del Usuario en la definición del problema que se va a abordar, y en la ejecución de las pruebas, y donde se potencia el uso de lenguajes de cuarta generación utilizados como lenguajes de consulta para interrogar a la Base de Datos con el fin de verificar la estructura del prototipo desarrollado, asegurándose de que su diseño responde a las definiciones especificadas.

La idea fundamental de esta Metodología es el desarrollo de prototipos. Un prototipo es un modelo de lo que al final se corresponderá con la Base de Datos definitiva, que se someterá a pruebas para comprobar su funcionalidad. De estas pruebas surgirán modificaciones que darán origen a un segundo prototipo, versión mejorada y posiblemente ampliada del primero, el cual se volverá a probar, repitiéndose el proceso hasta alcanzar el prototipo definitivo y su Base de Datos asociada.

La responsabilidad y ejecución de estas pruebas recae, como ya se ha mencionado, en el propio usuario, quien deberá de comprobar que el modelo de Base de Datos desarrollado es capaz de resolver todos los problemas planteados en el momento de la definición de los requerimientos del proyecto.

Como consideraciones finales, se trata de mostrar la dificultad existente para la implantación de cualquier tipo de método y de control, necesitándose para ello del total apoyo de la alta dirección de la empresa, y se insiste, una vez más, en la necesidad de

---

una Metodología, sea ésta cual sea.

Es nuestra intención que el estudio de este tema sirva para despertar el interés por la utilización de técnicas de Ingeniería del Software para el desarrollo de proyectos, ya que, a la larga, creemos que es el único camino viable para mantenerse a flote en el complicado mundo de la Informática.

## 1.1 **Ingeniería de Sistemas Software: Algunas Definiciones Básicas**

Para comenzar a hablar de Ingeniería del Software, o de Ingeniería de Sistemas Software, parece procedente dar algunas de las definiciones extendidas. Una definición bastante general podría ser la siguiente:

“La Ingeniería del Software es una parte de la Informática que proporciona métodos y técnicas para desarrollar y mantener software de calidad con el fin de resolver problemas”

Abarca las áreas:

\* Construcción de Compiladores y Sistemas Operativos

\* Desarrollo de Aplicaciones de:

- Negocios
- Investigación científica
- Medicina
- Producción
- Banca
- Control de tráfico
- Meteorología
- Leyes
- Etc.

---

Además de esta definición, podemos encontrar entre los muy diversos autores que han estudiado el tema, otras definiciones, todas ellas perfectamente válidas, y que difieren en que inciden y dan más importancia a alguno de los aspectos de la Ingeniería del Software que a otros. Veamos tres de ellas, entre las más relevantes.

**Definición 1:**

“La Ingeniería del Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de Sistemas Software”. [Zelkowitz, 1978]

Esta definición hace énfasis en principios y metodologías, entendiendo por principios las leyes y reglas bajo las cuales operan las máquinas, los sistemas, etc., y por metodologías el conjunto de normas que nos indican lo que debemos hacer y su secuencia en el tiempo. Por ejemplo, un principio podrá ser: “Los programas software deben de proporcionar mensajes de error cuando algo va mal”, mientras que la metodología nos dirá: “hay que chequear cada entrada y ver si es válida, y en caso contrario notificar al usuario”.

Además, incorpora los aspectos y tareas de mantenimiento, ya que el software es *algo vivo*, que continua operando desde su puesta en producción. Lo que implica tener en consideración dos aspectos:

1. La necesidad de comprender el software desde su nacimiento hasta su muerte, es decir, durante todo el Ciclo de Vida
2. El modo de emplear ese conocimiento y comprensión del software para crearlo y mantenerlo.

---

**Definición 2:**

“La Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de ordenador y de la documentación asociada necesaria para desarrollarlos, operar con ellos y mantenerlos”. [Boehm, 1976]

Esta definición destaca el uso del conocimiento existente sobre el desarrollo del software y no en el desarrollo de nuevas técnicas y metodologías. El Ingeniero del Software se interesa en encontrar soluciones prácticas a problemas reales.

Además, afirma que el software está formado por los programas de ordenador y por la documentación asociada (incluyendo manuales de operación, del Usuario, etc.) Los Sistemas Software no son solamente código y algunas líneas de comentarios en el programa. Por el contrario, cualquier cosa necesaria para desarrollar, operar y mantener el programa es parte del Sistema Software.

**Definición 3:**

“La Ingeniería del Software trata con la implementación de principios de Ingeniería y métodos para obtener, de un modo económico, software fiable y que funcione en máquinas reales”. [Bauer, 1972]

En esta definición se destacan tres aspectos prácticos, relacionados fuertemente con la Calidad del Software:

1. Se desea un software que funcione donde queremos que funcione.
2. Que lo haga según las especificaciones acordadas con los Usuarios
3. Que sea asequible en sus costes.

---

### **Algunas consideraciones sobre las tres definiciones anteriores:**

1. Todas contienen información útil para determinar el significado de lo que es Ingeniería del Software, de sus actividades, productos y filosofía.
2. Ninguna es incorrecta, aunque algunas son más útiles que otras, dependiendo de las circunstancias.
3. Todas enfatizan varios aspectos del proceso de desarrollo del software y de sus objetivos.
4. De ellas se desprende que la Ingeniería del Software es una disciplina en evolución, y por tanto existen varias definiciones para el mismo término. Es posible que con el tiempo se acuñen definiciones estándar, aunque también puede suceder que no sea así en todos los casos.

### **Conclusiones:**

Dado que continuamente están apareciendo nuevos problemas en nuevas áreas de investigación, habrá que desarrollar de modo continuo nuevas técnicas para resolverlos. Por ello, la Ingeniería del Software es un campo en constante desarrollo y con un crecimiento altamente acelerado.

Baste como ejemplo citar que varios investigadores en Inteligencia Artificial han comenzado a explorar nuevos métodos y técnicas de Ingeniería del Software para aplicar a los desarrollos que están llevando a cabo en esta área, como por ejemplo las Metodologías de Desarrollo de Sistemas Expertos.

Una vez conocidas y analizadas las distintas definiciones de la Ingeniería del Software, ya nos podemos adentrar en el paso siguiente: el estudio de los métodos usados por la Ingeniería del Software, es decir, el estudio de algunas de las Metodologías de Desarrollo de Sistemas Informáticos.

---

## 1.2 *Necesidad de una Metodología: Ventajas e Inconvenientes*

Aunque parezca mentira, todavía existen grupos de Proceso de Datos que operan sin unos Estándares de ámbito total, formalmente reconocidos y definidos, y donde las personas que los forman realizan el trabajo “a su manera”, de la mejor forma posible, por supuesto, pero sin ningún tipo de control serio de los Proyectos por parte de la Dirección.

Este “modus operandi” ha causado, y seguirá causando, a la larga no pocos problemas de difícil solución cuando se afronta el consabido problema del mantenimiento de las Aplicaciones.

Además, con la influencia cada vez mayor de los Ordenadores Personales en el medio informático, se han creado grupos de desarrollo “paralelo” por parte de los propios usuarios, que han dado en desarrollar sus propias aplicaciones por su cuenta y riesgo, y por supuesto, sin la utilización de ningún tipo de método para ello.

Como se verá más adelante, el coste de detección y posterior corrección de un error se dispara exponencialmente cuanto más adentrados nos encontremos en el desarrollo de un proyecto, siendo máximo cuando la Aplicación se encuentra ya en fase de Producción, es decir, cuando ya está en pleno funcionamiento.

Por lo tanto, se vuelve de vital importancia la detección de errores en las fases más tempranas del desarrollo, donde su corrección será mucho más fácil y mucho menos costosa. En otras palabras, se plantea la necesidad de implementación de una Metodología de Desarrollo de Sistemas que nos facilite esa detección y que nos suministre un modo “común” de hacer las cosas en nuestra Instalación.

Por supuesto que en Informática, como en todo orden de cosas, no hay almuerzo gratis, y cada objetivo tiene su coste. Por lo tanto, si estamos realmente dispuestos a implantar una metodología en nuestro Centro de Proceso de Datos, debemos de ser totalmente conscientes del tipo de “aventura” en la que nos estamos metiendo y debemos de

---

conocer cuales son los inconvenientes con los que nos vamos a encontrar (nos los van a echar en cara, ¡seguro!) y también de las ventajas que vamos a obtener del esfuerzo invertido, para convencernos a nosotros mismos, en primer lugar, de los beneficios indudables que se van a obtener, para estar así en disposición de convencer a los de, si fuera necesario.

Entre los más serios inconvenientes con los que toparemos están, a nuestro juicio, la imagen de burocracia exagerada que puede dar el uso de una metodología si no se estudia con cuidado el proceso que se pretende implantar. Todo método lleva su papeleo incorporado, y deberemos de conocerlo para poder minimizarlo y no caer en un exceso de control que podría originar un rechazo y una vuelta “a los orígenes”. Nos encontraremos con una gran pérdida de tiempo en reuniones, preparación de informes, creación de documentación..... y vuelta a reunirse de nuevo. A poco que nos descuidemos, invertiremos más tiempo en reuniones de trabajo que en el propio desarrollo, y habremos caído en el polo opuesto al que nos encontrábamos inicialmente.

Así como los inconvenientes hasta ahora mencionados son en cierta medida evitables, también habremos de pagar un alto coste en tres puntos que no se pueden evitar, coste que deberemos de asumir si queremos que nuestra implantación sea un éxito: El desarrollo e implantación de Estándares en la Instalación, la formación adecuada del personal y un seguimiento activo de post-implantación, que irá disminuyendo paulatinamente con el tiempo.

Ante todo lo expuesto, alguien podría preguntarse si vale realmente la pena embarcarse en semejante aventura y si no sería mejor “dejar el mundo como está”. Parece, pues, conveniente pasar a analizar algunas de las ventajas más importantes, que sin duda obtendremos, antes de tomar la decisión de si deberemos coger el barco o quedarnos en puerto.

Desde luego, nos vamos a encontrar con una estructura y con una operativa “común” para toda la Instalación. Una forma igual de hacer las cosas en todo el entorno, de manera que cualquiera pueda y sea capaz de entender y modificar con facilidad lo que



---

otro ha desarrollado. Pensamos que para cualquier persona que haya trabajado o esté trabajando en un Centro de Proceso de Datos en la sección de Mantenimiento, tan solo esta ventaja ya justificaría los inconvenientes mencionados anteriormente.

Por otra parte, cualquier metodología va a descomponer un Proyecto en partes (normalmente llamadas Fases), lo cual facilita enormemente su desarrollo y proporciona además una serie de ventajas a la hora de asignar costes a cada una de ellas, permitiendo también un control estricto de los cambios que, inevitablemente, se van a producir en los requerimientos iniciales de comienzo del Proyecto.

Por todo esto, y por lo que a continuación pasaremos a detallar en este tema, se consigue una reducción sustanciosa de los costes de mantenimiento, saca sin fondo de cualquier instalación de Proceso de Datos.

Queda aquí reflejada superficialmente la problemática con la que nos encontraremos al intentar implantar cualquier metodología de gestión de proyectos. En las siguientes líneas intentaremos estudiar dos metodologías que pueden cumplir las funciones ya detalladas. La primera, Metodología de Gerencia de Proyectos, basada en el esquema tradicional de Desarrollo de Aplicaciones, y la segunda, PDM80, orientada al Desarrollo de Prototipos.

---

### 1.3 *Metodología de Gestión de Proyectos: Estándares para Dirección de Proyectos*

Toda buena metodología debe de cubrir no solo la etapa de desarrollo propiamente dicho de un proyecto, sino que además, debe de abarcar todo el ámbito de la Instalación, desarrollando estándares que engloben todo el conjunto de las labores diarias, puramente informáticas o no, que se llevan a cabo en un CPD. Más que una serie de normas de obligado cumplimiento, se debe de intentar que la metodología sea una cierta “mentalidad” existente y compartida por todo el personal de la Instalación. Hay que saber crear el ambiente necesario de colaboración y lograr las motivaciones oportunas hasta poder sentir que el método es “bueno” y beneficioso para todos.

Por lo tanto, y para dar ese ámbito general que se pretende, dividiremos la metodología en cinco grandes Etapas que posteriormente se detallarán: Una primera parte dedicada a la *Organización* necesaria para crear el ambiente que se desea y soportar la metodología, una segunda parte de *Planificación*, donde estudiaremos cuidadosamente lo que debe de hacerse y cómo debe de hacerse, una etapa de *Control*, obviamente necesaria en cualquier metodología, que nos indique cómo vamos y qué bien lo estamos haciendo, la etapa del *Desarrollo* propiamente dicho de nuestro Proyecto y por último, una parte que cubra la *Documentación* necesaria y suficiente en el desarrollo de nuestras Aplicaciones.

#### 1.3.1 *Organización de la Empresa y del Sistema de Información*

En este apartado trataremos de definir una estructura completa de responsabilidades para evitar los fallos que se podrían derivar de una falta de definición clara en los requerimientos, motivo que ha causado el abandono de no pocos proyectos. Nos aseguraremos de dejar claro quién tiene la responsabilidad del proyecto, quién definirá sus requerimientos, quién tendrá la autoridad para hacer una u otra cosa, quién informará a quién, quién pagará por el desarrollo, etc., etc.

Esta definición de responsabilidades deberá de estar perfectamente clara aún tratándose de proyectos pequeños, debiendo de asegurarnos de que todo el mundo sabe quién tiene

---

la autoridad y cual es ésta, antes de comenzar un nuevo proyecto.

### 1.3.1.1 Estructura de la Organización

Para lograr el control apetecible en el desarrollo de proyectos, es necesario disponer en la Instalación de una Estructura que se pueda usar en todos los proyectos que se vayan a realizar, independientemente de su tamaño, por lo que la misma deberá de ser lo suficientemente flexible para permitir el logro de este objetivo, sin perder, por otra parte las ventajas que una buena organización nos aportaría.

La solución propuesta se muestra en la figura 2.1, donde, para máxima flexibilidad, se hace uso de un Equipo de Fase para el desarrollo de proyectos, como se verá más adelante.

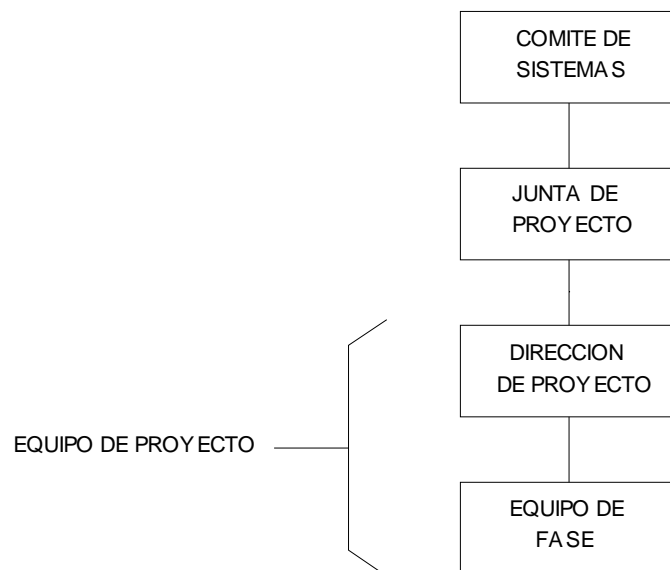


Figura 1.1 Jerarquía del Equipo de Fase

---

#### 1.3.1.1.1 Comité de Sistemas

En primer lugar necesitaremos de un equipo de personas, pertenecientes a la alta dirección de la empresa, al que podremos llamar *Comité de Sistemas*, encargado de la planificación a largo y medio plazo de todos los sistemas informáticos de nuevo desarrollo, y que indicará las directrices a seguir por Proceso de Datos. Normalmente, este equipo, llámese Comité de Sistemas o de cualquier otra manera, ya se encuentra presente en cualquier empresa, aunque tal vez, y sobre todo si se trata de una empresa pequeña, no esté perfectamente delimitado en cuanto a sus atribuciones y responsabilidades, sino que puede tratarse de una figura difusa y de actuación casi espontánea, por lo que se deberá, si este es el caso, de definir claramente quien lo forma, cuales son sus atribuciones y cual es el ámbito de su actuación.

Así, el Comité de Sistemas deberá de estar formado por una dirección de usuarios de alto nivel, por representantes de la dirección general de la empresa y de la dirección de Informática, y entre sus primordiales atribuciones podríamos destacar, entre otras, la decisión de la política de adquisición de equipos, el control de los recursos informáticos necesarios para cubrir la demanda de los usuarios, la coordinación de los requerimientos de estos en todos los proyectos, el control de dichos proyectos, la designación de los miembros de la *Junta de Proyecto*, etc.

#### 1.3.1.1.2 Junta de Proyecto

En un segundo nivel, destacaremos la ya citada *Junta de Proyecto*, dependiente de la dirección de Informática, y cuya misión es el seguimiento de todos los proyectos abordados por el CPD, y que estará formada por el director de Informática, un representante del Usuario y un representante de Informática, los cuales se encargarán de la dirección y control de los proyectos y actuarán como unión entre el Comité de Sistemas y la *Dirección de Proyecto*.

Normalmente, en proyectos pequeños no será necesario disponer de una Junta de Proyecto de tres personas, por lo que consideraremos que cada una de las figuras

---

representan funciones y no personas, y que una persona puede desempeñar más de una función, si la situación así lo requiere.

#### 1.3.1.1.3 Dirección de Proyecto

La dirección y seguimiento día a día de un proyecto estará a cargo de la *Dirección de Proyecto*, donde las funciones a cubrir serán las de un Director o *Gerente de Proyecto*, un Jefe de Equipo, un Sustituto y un Administrador del Proyecto. Nuevamente recalcaremos que las figuras aquí mencionadas representan funciones que pueden ser cubiertas por la misma persona.

El Gerente de proyecto tendrá categoría de directivo y sus responsabilidades abarcan la asignación de tareas y objetivos, preparación de presupuestos y evaluación de los miembros del *Equipo de Fase* al final de cada fase del proyecto.

El Jefe de Equipo es el responsable técnico del proyecto y deberá de colaborar con el Gerente de Proyecto en la toma de decisiones de orden técnico, asesorándole si fuese necesario.

El Sustituto actúa como apoyo del Jefe de Equipo, realizando labores de Control de Calidad en proyectos de larga duración. Esta función puede ser realizada por el propio Jefe de Equipo en proyectos pequeños.

Por último, el Administrador se encargará, como es lógico, de la parte financiera, área en la cual el Gerente de Proyecto puede ser inexperto, realizando labores administrativas, elaborando planes bancarios, etc.

---

#### 1.3.1.1.4 Equipo de Fase

Dado que las tareas y los resultados de cada una de las fases en que se va a dividir un proyecto no son iguales, las habilidades para desarrollar una u otra fase tampoco son las mismas, por lo que el Equipo de Fase estará formado por personas distintas en cada fase, lo que nos lleva a la necesidad de establecer un equipo por cada fase del proyecto. El tamaño del Equipo dependerá principalmente del tamaño del proyecto, y su función es realizar el desarrollo de cada fase del Proyecto.

Además del Equipo de Fase puede existir un *Grupo de Soporte*, con funciones de asesoramiento en la dirección técnica del proyecto, pudiendo también realizar labores de evaluación de paquetes de software existentes y un grupo de *Coordinación Técnica*, cuya labor primordial sería la de ayudar en la estructuración y validación de los planes y estimación del proyecto, identificar problemas potenciales y revisar la duración de los proyectos.

Una vez más, nuestro estándar deberá de definir las funciones a cubrir y no las personas.

### 1.3.2 *Planificación del Proyecto*

Bajo este apartado, examinaremos una propuesta de estándar de planificación de proyectos que sea lo suficientemente flexible para que nos sirva para cualquier tipo de proyecto y para cualquier clase de herramienta de planificación, y asimismo que permita la planificación progresiva a medida que se van conociendo más datos sobre el proyecto.

#### 1.3.2.1 **Estimación del Proyecto**

El primer paso será establecer una estimación del proyecto que nos permita determinar

---

la mejor solución posible (normalmente habrá varias) a nuestro problema. Para ello, deberemos de determinar el coste de cada solución para poder conocer los costes comparativos y conocer así cual es la alternativa mejor.

A la hora de dar esta estimación, se deberá de tener en cuenta que ésta se hace en un momento en el que se sabe muy poco del proyecto y se cuentan con muy pocos datos, por lo que deberemos de basar nuestra investigación en datos históricos, siendo de vital importancia la experiencia en otros proyectos similares.

De todos modos, cualquier estimación se deberá de desglosar en fases, que coincidirán con la división en fases del propio proyecto, como se verá en el apartado correspondiente dedicado al desarrollo del proyecto.

Existen en la actualidad una gran variedad de herramientas de estimación de costes que simplifican el problema de los cálculos que hay que realizar, pero que no evitan la necesidad de un fuerte conocimiento de planificación de proyectos para poder utilizarlas.

### **1.3.2.2 Planes del Proyecto**

El segundo paso consistirá en establecer la planificación adecuada para nuestro proyecto. Consideraremos tres planes: Un Plan de Fase, un Plan de Disponibilidad de Recursos y un Plan de Desviación.

#### **1.3.2.2.1 Planes de Fase**

Por cada fase del proyecto, estableceremos dos tipos de planes: Un *Plan Técnico* y un *Plan Financiero*. El primero mostrará las actividades, tareas, recursos y asignación de tiempos y el segundo será el encargado de calcular el coste de la fase.

Estos planes deberán de estar disponibles antes del comienzo de cada fase, con la excep-

---

ción del plan para la primera fase, que se lleva a cabo como una de las primeras tareas de la misma.

El paquete de planificación debería de estar formado por una descripción, un plan gráfico (Pert, ROI, etc.), un documento con los supuestos sobre los que se ha basado el plan y un estudio de los problemas potenciales que se hayan detectado.

#### 1.3.2.2.2 Plan de Disponibilidad de Recursos

Es muy importante establecer un calendario que nos muestre la disponibilidad de los recursos de la organización para las fechas previstas del desarrollo del proyecto. Este plan deberá de incluir datos relativos a la disponibilidad y no disponibilidad de cada recurso, contemplando también cuestiones tales como vacaciones, períodos de formación del personal, asignación compartida con otros proyectos, etc.

#### 1.3.2.2.3 Plan de Desviación

La aceptación del Plan de Fase por parte del Usuario y de Informática implica la adquisición de ciertos compromisos por ambas partes. El Usuario se compromete a aceptar los costes derivados del desarrollo de la fase, e Informática, por ese precio, se compromete a llegar a un cierto grado de desarrollo del proyecto en una cierta fecha y en un determinado control de calidad. Si alguno de estos factores comenzase a variar de manera sensible, es responsabilidad del Gerente de Proyecto el notificarlo a la Junta de Proyecto, de la que se recordará que forma parte un representante del Usuario a nivel de dirección.

El plan describirá el problema causante de la desviación, así como las causas del mismo, el impacto de la desviación en la totalidad del proyecto y las recomendaciones para su solución.

Este plan abarcará hasta el fin de la fase correspondiente, ya que los trabajos de recuperación serán incorporados en el plan de la fase siguiente, como una tarea de la



---

misma.

### **1.3.3 Control**

Esta etapa, la tercera de la metodología, considera la necesidad de llevar a cabo una supervisión sobre diversos aspectos del proyecto, elaborando los estándares necesarios y atacando el problema de las comunicaciones orales y escritas en un entorno de equipo y proyecto.

#### **1.3.3.1 Desarrollo de Estándares**

Puede surgir en este momento la pregunta sobre la necesidad de elaborar estándares para el control de un proyecto. Como se decía cuando hablábamos sobre las ventajas e inconvenientes de una metodología, habrá que pagar un precio por los estándares, y puede plantearse de nuevo la duda de la conveniencia de pagar o no dicho precio.

A nuestro modo de ver, es totalmente necesario disponer de estándares de control de proyectos para poder contestar a una serie de interrogantes que se nos pueden plantear en cualquier momento, tales como saber con qué calidad estamos trabajando, o conocer si aún estamos produciendo lo que el usuario realmente quiere, o bien determinar si nos encontramos dentro del calendario previsto o si, por el contrario, se ha producido una desviación; o simplemente para darnos cuenta de si nuestros costes se disparan por no poder descubrir y fijar los problemas a tiempo. El poder dar respuesta a cualquiera de estas preguntas es una razón de por sí suficiente para justificar el coste de desarrollo de estándares de control de proyectos en cualquier organización.

Por otra parte, deberemos de imponernos una serie de limitaciones a la hora de especificar nuestros estándares, pues de lo contrario podríamos caer en una excesiva rigidez de operación que podría originar un contraproducente rechazo de los mismos. Así, los estándares nos deberán de servir tanto para proyectos largos como para proyectos pequeños, deberán de ser compatibles con el resto de estándares de la Instalación y evitarán al máximo la burocracia.

---

### 1.3.3.2 Comunicación Técnica: Informes y Reuniones

Otro aspecto importante de Control es el recoger información sobre el proyecto, tanto técnica como financiera, y hacer interpretar esta información por la gente adecuada. La solución para el manejo de esta información la llevaremos a cabo por medio de Informes y Reuniones.

Dentro de los Informes, consideraremos cinco tipos diferentes: Hoja diaria, Plan de Fase, Informe de Estado, Resumen de Fase y Peticiones de Modificación. En cuanto a Reuniones, las tendremos de dos clases: de Revisión de Fase y Walkthroughs.

#### 1.3.3.2.1 Hoja Diaria u Hoja de Tiempos

Elaborada por los miembros del Equipo de Fase para el Gerente de Proyecto, es, como su nombre indica, de frecuencia diaria y recogerá el esfuerzo dedicado a las distintas tareas de la Fase que se esté acometiendo.

#### 1.3.3.2.2 Matriz Esfuerzo / Plan de Fase

La cumplimenta el Jefe de Equipo para el Gerente de Proyecto y es de frecuencia semanal. Comparará el esfuerzo realizado contra el Plan de Fase, anotándose los trabajos realizados así como las desviaciones.

#### 1.3.3.2.3 Informe de Estado

Del Gerente de Proyecto a la Junta de Proyecto, y de frecuencia mensual, consiste en un breve informe del estado del proyecto donde se deberá de hacer constar el “status” técnico y los problemas más significativos. Llevará además un informe sobre los trabajos realizados en las últimas dos semanas y los trabajos a realizar en las dos semanas siguientes.

---

#### 1.3.3.2.4 Resumen de Fase

Es una forma resumida de pasar la información en un documento muy breve. Su objetivo fundamental es pasar información financiera a la Junta de Proyecto.

#### 1.3.3.2.5 Peticiones de Modificación

Su origen es normalmente un usuario y su destino es la Junta de Proyecto, y la frecuencia es aleatoria. Es un tipo de informe vital si se desea que el proyecto esté bajo control. Una vez que han sido firmados los acuerdos sobre presupuesto y especificaciones de un proyecto, solo la Junta de Proyecto tiene la autoridad de aceptar cualquier cambio y solo el cliente puede aceptar los nuevos costes adicionales que se deriven de dicha modificación.

Por lo tanto, cualquier modificación deberá de estar perfectamente documentada y razonada, para poder determinar su impacto en el proyecto y su coste probable.

No debemos olvidar que la implementación de cambios sin un control ni una autorización adecuada han sido, y probablemente seguirán siendo, la causa de que fallen una gran cantidad de proyectos y no cumplan sus especificaciones de tiempo y/o presupuesto.

#### 1.3.3.2.6 Reunión de Revisión de Fase

Es la reunión formal de la Junta de Proyecto y el Gerente de Proyecto, y para la cual se deberán de definir unos estándares muy claros. Su propósito es que el Gerente de Proyecto tenga la ocasión de dar un informe de la marcha del proyecto y pueda hacer toda clase de preguntas que inevitablemente surgirán durante la ejecución del mismo. Puede, asimismo, presentar planes a la Junta para su aprobación o para que considere posibles cambios de dirección o mejoras, o bien aprovechar la reunión para solicitar la adjudicación de nuevos recursos si fueran necesarios. También se deberá de convocar

---

esta reunión para terminar el proyecto, bien porque ha sido finalizado, bien porque la Junta de Proyecto considera que existe una razón válida para que el proyecto deje de ser interesante.

Obligatoriamente deberán de asistir a esta reunión los miembros de la Junta de Proyecto, el Gerente de Proyecto y un Auditor Interno o representante de Control de Calidad, pudiendo ser asistentes opcionales el Jefe de Equipo, especialistas técnicos y el Administrador del Proyecto, si fuera necesario.

Se deberá de celebrar esta reunión obligatoriamente al comienzo del proyecto, al final de cada fase, al final del proyecto y en el caso de producirse desviaciones en el mismo, como ya se ha comentado anteriormente, pudiendo convocarse opcionalmente a intervalos predeterminados durante una fase de proyecto larga, en los comienzos próximos de una fase, o cuando hay revisión de solicitudes de modificación, entre otras ocasiones.

#### 1.3.3.2.7 Walkthroughs

Con este nombre se conoce el segundo tipo de reuniones de utilidad para el control del proyecto. Su función principal no es establecer ninguna clase de intercomunicación entre partes, como parece que debería de ser el objetivo de una reunión, sino que lo que pretende es establecer una evaluación del trabajo de una persona por parte del Equipo de Proyecto. Por considerarlo de interés especial, se proporciona información detallada en el Apéndice A, al final de este capítulo.

Así, un Walkthrough es una herramienta muy poderosa para evaluar la calidad del producto que se está produciendo, siempre que se evite la tentación de utilizarlo para evaluar a la persona cuyo trabajo se está revisando.

Por lo general, un walkthrough se convoca cuando un miembro del Equipo (normalmente un programador) considera que una parte de su trabajo está completa y puede ser revisada. En este momento, invita a otros miembros del Equipo para que

---

revisen su trabajo y le ayuden a detectar errores potenciales. Esta es la labor fundamental de un walkthrough: la detección de errores. Por tanto debe de evitarse por todos los medios el utilizarlo para otros propósitos, (como por ejemplo evaluar a los programadores) que indudablemente invalidarían los resultados.

Durante la reunión, uno de los miembros, actuando como secretario, irá tomando notas de todos los errores encontrados o supuestamente detectados, en una Lista de Acción (ver figura 1.3), documento base del walkthrough, la cual entregará al programador convocante al final de la misma. En esta Lista se especificará *solamente* que se han detectado ciertos errores potenciales, no figurando en absoluto ni cuantos fueron, ni quién los encontró, ya que se debe de insistir una vez más que el objetivo primordial de un walkthrough es la detección de errores y no la evaluación de programadores.

Tras la reunión, el programador dispone de dos días para revisar y/o corregir los errores encontrados, especificando su solución en la propia Lista de Acción, indicando si se han resuelto o si, por el contrario, algún punto en cuestión no constituía un error, debiendo entregar de nuevo la Lista a los participantes con la finalidad de que éstos estén enterados de los cambios realizados y puedan conocer la totalidad del proceso.

Por esto, también se suelen emplear este tipo de reuniones para la formación de nuevos miembros del equipo, ya que constituyen un excelente método de aprendizaje.

Las funciones a desempeñar en un walkthrough son las de revisado, revisador o revisadores, moderador y secretario, pudiendo ser realizadas varias funciones por la misma persona.

El tiempo máximo de un walkthrough suele estipularse en dos horas, pasado el cual está comprobado que ya no se obtienen los resultados apetecidos de detección de errores.

---

### 1.3.4 *Desarrollo del Proyecto*

En los apartados anteriores se ha planteado la conveniencia de dividir un proyecto en Fases que nos permitirán simplificar notablemente el problema inicial. El desarrollo del proyecto, tal y como aquí se entiende, es el que contempla cualquier metodología de desarrollo de sistemas.

Así, bajo este apartado de desarrollo del proyecto y siguiendo las pautas de las metodologías clásicas de desarrollo, consideraremos su división en seis fases: *Definición de Requerimientos, Diseño Externo, Diseño Interno, Desarrollo, Pruebas e Instalación y Seguimiento*, fases que la mayoría de las metodologías coinciden en establecer, aunque puedan variar ligeramente sus nombres y quizás alguna de sus tareas asociadas. Las tres primeras suelen agruparse bajo el nombre genérico de *Diseño* y las tres últimas bajo el de *Implantación*.

A su vez, dividiremos cada Fase en Actividades, éstas en Subactividades y éstas a su vez en Tareas, nivel más bajo en la descomposición jerárquica de un proyecto informático.

Comentaremos a continuación de forma breve los objetivos fundamentales de cada una de las Fases, dando al final un esquema resumido de las subdivisiones antes mencionadas.

#### 1.3.4.1 **Definición de Requerimientos**

El objetivo de esta primera fase consiste en dejar perfectamente claro cuales son los requerimientos del usuario para el proyecto que se está abordando, cual va a ser el coste que estará dispuesto a afrontar y qué va a obtener por parte de Informática por ese coste. En otras palabras, establecerá el compromiso formal entre el Usuario e Informática con respecto al desarrollo del proyecto.

---

El papel del usuario durante esta fase es fundamental, correspondiéndole la mayor parte del trabajo en el desarrollo de la misma, ya que será él mismo el que deberá de especificar cuales son los objetivos del proyecto.

#### **1.3.4.2 Diseño Externo o Diseño Conceptual**

Esta fase corre principalmente a cargo del Analista Funcional, (de ahí que algunos autores la definan con el nombre de Análisis Funcional), con el apoyo del Analista del Usuario, y consiste fundamentalmente en realizar un estudio detallado de todas las funciones y procesos del proyecto que se pretende desarrollar.

Habrà que definir cuales seràn las entradas y salidas de la Aplicaci3n, definir los requisitos de datos, realizar el dise1o l3gico de la Base de Datos, especificando sus entidades, sus atributos y las relaciones entre ambos, determinar los componentes del Sistema, identificando criterios de rendimiento y situaciones de prueba, etc., y finalmente emitir un informe con las especificaciones funcionales para su revisi3n por la Junta de Proyecto. La preparaci3n de este informe serà la 3ltima tarea de todas y cada una de las Fases.

#### **1.3.4.3 Dise1o Interno o Dise1o F3sico**

El Analista T3cnico es el responsable de la ejecuci3n de la mayor3a de las tareas de esta fase, compartiendo ocasionalmente esta labor con el Analista Funcional y el Gerente de Proyecto.

Su misi3n consiste en preparar todo el soporte informàtico del proyecto, realizando, entre otras, las labores de dise1o de Bases de Datos, Esquemas, Subesquemas, procesos del Sistema, procesos de Seguridad y Control, procesos de Prueba, etc.

Como una de las 3ltimas Actividades de esta fase, realizarà un Anàlisis de Costes y

---

Beneficios, donde deberá de resumir el resultado económico y estimar los costes de la instalación del Sistema una vez terminado, para presentar este estudio a la Dirección.

#### **1.3.4.4 Desarrollo**

En esta fase, llevada a cabo por el Gerente de Proyecto y el Equipo de Fase en su mayor parte, se realizará el trabajo de programación de todas las Unidades de Trabajo (programas) que conformarán el Sistema.

En general, se deberá de organizar el proyecto estableciendo estándares técnicos para su desarrollo, se completará el diseño técnico y de la Base de Datos, se confirmará el análisis de costes/beneficios, se prepararán los datos para las pruebas, y se establecerá el plan de pruebas, se desarrollarán los manuales para el usuario y se entrenará al personal que deberá de operar con el Sistema.

Es, con mucho, la más laboriosa y larga de las fases, y donde más recursos se consumen en cualquier proyecto.

#### **1.3.4.5 Pruebas**

Como su nombre indica, en esta fase se verificará el producto que se ha elaborado para validar su correcto funcionamiento.

Corre fundamentalmente a cargo del Analista del Usuario y del Analista Técnico, y entre otras, se deberán de realizar pruebas de integración con el resto de los Sistemas ya en funcionamiento, y llevar a cabo las pruebas del Usuario.

Como fin de fase, se emitirá un informe para la Junta de Proyecto.



---

#### **1.3.4.6 Implantación y Seguimiento**

La responsabilidad de esta última fase recae principalmente en el Analista Funcional y en el Analista Técnico, así como en el personal de Producción, y su misión consiste en llevar el Sistema terminado y probado a un Estado de Producción Normal.

Como actividades de esta fase podemos citar la conversión de los ficheros que sean necesarios, el completar la documentación para futuras mejoras, la vigilancia de la producción durante un período razonable de tiempo y el paso del nuevo Sistema a un estado rutinario de operación.

Como una de las últimas tareas, deberemos de transferir la responsabilidad del Sistema al Grupo de Producción, que será el encargado de la explotación del Sistema en lo sucesivo.

#### **1.3.5 Documentación**

Identificaremos en este apartado la necesidad de alcanzar unos estándares en la documentación de cualquier proyecto, ya sea grande o pequeño, definiendo cual debe de ser la estructura de dicha documentación.

Los objetivos que la documentación debe de cubrir serán el dejar constancia de toda la información técnica y de gestión que se ha ido recogiendo a lo largo del proyecto, obtener los manuales del usuario, ser capaz de registrar cualquier cambio en las especificaciones y preparar información sobre el uso del Sistema en los manuales del operador.

---

En los proyectos pequeños, los estándares de documentación serán de tal naturaleza que no constituyan una carga innecesaria, y en cualquier caso deberán de evitar duplicidades de información y deberán de encajar con los trabajos de cada fase y ser medibles al final de la misma, ya que es fundamental que la documentación se vaya obteniendo a medida que se va desarrollando el proyecto y no debe de considerarse nunca como un trabajo a realizar “a posteriori”.

Consideraremos dos tipos de documentación: Documentación de Negocios o de Gestión y Documentación Técnica.

### **1.3.5.1 Documentación de Negocios**

En general, la documentación de gestión consistirá en Planes, Informes, Conformidad Firmada de Fin de Fase, Documentación de Entrevistas de Revisión y Walkthroughs y Peticiones de Modificaciones.

Junto con estos impresos, una Auditoría seguirá la pista del proyecto, revisando los objetivos del mismo, así como los cambios producidos, los pasos previstos para alcanzar dichos objetivos, el control de fechas, recursos, costes, y finalmente, el rendimiento alcanzado contra el rendimiento previsto.

### **1.3.5.2 Documentación Técnica**

Esta consistirá, principalmente, en documentar los programas, rutinas, ficheros, bases de datos, etc., utilizados durante el desarrollo del proyecto. Se deben de definir unos estándares claros de documentación, ya que la tendencia innata es a no documentar, teniendo siempre presente que la *documentación es para el lector*, y no olvidando asimismo que debemos documentar siempre por adelantado y nunca después de haber terminado un proceso.

---

### 1.3.5.3 Estructura de la Documentación

En general, de un proyecto deberemos de documentar entre otros los puntos que a continuación se detallan:

1. Visión general del Proyecto
2. Sistema Actual, si existe
3. Nuevo Sistema
4. Documentos de Pruebas
5. Manual del Usuario
6. Manual de Instalación

Es labor de cada empresa el detallar los formularios e impresos necesarios para cubrir cada uno de estos apartados, debiendo de estar perfectamente estandarizados cada uno de ellos.

Por otra parte, sería muy conveniente el poder disponer de un paquete que gestione la documentación, (nos estamos refiriendo fundamentalmente a un Diccionario de Datos), como herramienta para descargar un poco la laboriosa y siempre tediosa tarea de la documentación de un proyecto.

---

### 1.3.5.3.1 Índice del Manual del Usuario

Como referencia, se incluye en este apartado un “esqueleto” sugerido para desarrollar el Manual del Usuario de una Aplicación cualquiera. Dado que se intenta cubrir la mayoría de las posibilidades, es probable que para alguna Aplicación concreta sobren algunos de los puntos expuestos.

#### **0 Índice**

#### **1 Introducción**

##### 1.1 Descripción general de la Aplicación

(Comentario breve sobre el origen, la finalidad y el ámbito de la Aplicación)

##### 1.2 Objetivos y ventajas

(Descripción del propósito de la Aplicación y de las ventajas principales a obtener con el uso de la misma)

##### 1.3 Usuarios y desarrolladores de la Aplicación

(Indicar a quién va dirigida la Aplicación y el Equipo de Trabajo que la ha desarrollado)

#### **2 Requerimientos de Hardware y Software**

##### 2.1 Requerimientos de Hardware

(Procesador, memoria RAM, espacio en disco para datos y programas y dispositivos hardware necesarios, como ratón, impresoras, modems, etc.)

##### 2.2 Requerimientos de Software

(Describir las necesidades de software, tanto de Sistemas, como por ejemplo la versión del Sistema Operativo necesaria, como de Aplicación,

---

indicando en qué lenguaje ha sido desarrollada, si necesita licencia run-time para su ejecución, etc.)

### **3 Procedimientos de Instalación, arranque y finalización de la Aplicación**

#### 3.1 Contenido de los diskettes de instalación

(Índice del contenido de los diskettes de instalación. No es necesario detallar la función de cada programa o base de datos)

#### 3.2 Procedimiento de Instalación

(La Aplicación deberá de disponer de un procedimiento de instalación lo más automático que sea posible. De ser así, en este punto se indicará el modo de arranque del procedimiento de instalación y las sucesivas pantallas de toma de datos sobre la instalación que puedan aparecer; en caso contrario, detallar de manera muy clara los pasos que debe de dar el usuario para instalar la Aplicación, como crear directorios, copiar ficheros, etc.)

#### 3.3 Arranque de la Aplicación

(Se explicará, lo más brevemente posible, cómo se ejecuta la Aplicación)

#### 3.4 Finalización de la Aplicación

(Se explicará, lo más brevemente posible, cómo se termina la Aplicación. Se incluirán recomendaciones en el sentido de no apagar el PC ni utilizar Ctrl + Alt + Del para finalizar la Aplicación, etc.)

### **4 Descripción de Menús, Pantallas y Procesos**

#### 4.1 Elementos comunes a toda la Aplicación

(Se describirá el funcionamiento de los Menús, Menús Desplegables, Cajas, Selectores, Botones, etc., comunes a todas las pantallas de la

---

Aplicación.)

#### 4.2 Menús, pantallas y procesos de la Aplicación

(Comenzando en la pantalla principal, se describirán todos los menús, submenús y pantallas de la Aplicación, de izquierda a derecha y de arriba a abajo, detallando los procesos que se realizan en cada pantalla, excepto los originados por los elementos comunes explicados en el punto anterior)

#### 4.3 Menús y pantallas de ayuda

(El último de los procesos del menú principal será el de la ayuda de la Aplicación. Se describirán sus pantallas y procesos con igual criterio que en el apartado anterior)

### **5 Descripción de Procedimientos del Usuario**

#### 5.1 Procedimientos de la Aplicación

(Descripción detallada de todos los procedimientos que el usuario deba de realizar con la Aplicación. Incluir las entradas y salidas de cada proceso, así como los controles de validación de datos que realizan)

#### 5.2 Procedimientos Especiales

(Incluir procedimientos de backup y recuperación de la información, señalando frecuencia recomendada de los backups y cualquier otra consideración de interés. Incluir también procesos de reconstrucción periódica de índices de las Bases de Datos, etc.)

### **6 Soporte Técnico y Mantenimiento**

#### 6.1 Soporte Técnico

(Indicar como obtener apoyo técnico tanto para el hardware como para el

---

software)

## 6.2 Mantenimiento

(Indicar las características y responsabilidades de mantenimiento de la Aplicación y del hardware)

# 7 Apéndices

## 7.1 Glosario de términos

(Relación de términos propios de la Aplicación o términos técnicos, o en general cualquier otro, que sean de interés para el usuario)

## 7.2 Índice de abreviaturas

(Se incluirá un índice de todas las abreviaturas empleadas, indicando su significado, aunque éste pueda parecer obvio, como por ejemplo PC)

## 7.3 Índice de formularios

(Relación de formularios necesarios para la captura de datos o para cualquier otro propósito. Los formularios deberán de estar numerados, indicándose su número, su nombre y su finalidad. Se incluirá una muestra de cada formulario)

## 7.4 Índice de listados

(Relación de listados originados por la Aplicación. Los listados deberán de estar numerados, indicándose su número, su nombre y su finalidad. Se incluirá una muestra de cada listado)

## 7.5 Descripción del hardware

(Si no se dispone de los manuales del fabricante del hardware utilizado, impresoras, modems, fax, etc., se deberá de explicar el funcionamiento de cada dispositivo, indicando las operaciones de mantenimiento, como

---

cambio de toner en las impresoras, y los errores más frecuentes que pueden originar, su causa probable y su posible solución)



---

### 1.3.5.3.2 Índice del Manual de Referencias Técnicas

Se muestra aquí el índice de un Manual de Referencias Técnicas, con las mismas salvedades y consideraciones del punto anterior.

#### **0 Índice**

#### **1 Introducción**

##### 1.1 Descripción del Manual

(Comentario breve sobre el contenido del Manual)

##### 1.2 Notación y Convenciones

(Simbología empleada en el Manual, si procede)

##### 1.3 Terminología

(Indicar el significado de cualquier término técnico utilizado en el Manual y que deba de ser aclarado)

#### **2 Bases de Datos**

##### 2.1 Relación general

(Listado de todas las Bases de Datos de la Aplicación, con el nombre y una descripción breve de su función)

##### 2.2 Descripción detallada

(Documentar cada Base de Datos indicando su nombre, ubicación (path) y estructura: campos que la forman, tamaño, tipo, índices, etc., así como los valores por defecto de cada campo y su validación)

---

### **3 Ficheros**

#### 3.1 Relación general

(Listado de todos los Ficheros de la Aplicación, con el nombre y una descripción breve de su función)

#### 3.2 Descripción detallada

(Documentar cada Fichero indicando su nombre, ubicación (path) y estructura: campos que lo forman, tamaño, tipo, índices, etc., así como los valores por defecto de cada campo y su validación)

### **4 Programas**

#### 4.1 Relación general

(Listado de todos los Programas de la Aplicación, con el nombre y una descripción breve de su función)

#### 4.2 Descripción detallada

(Documentar cada Programa indicando su nombre, ubicación (path) y una descripción general de los procesos que realiza. Si es necesario, incluir descripción detallada de los procesos que, por sus características, así lo requieran)

### **5 Pantallas**

#### 5.1 Relación general

(Listado de todas las Pantallas de la Aplicación, con el nombre y una descripción breve de su función)

#### 5.2 Descripción detallada

(Describir cada Pantalla su nombre, ubicación (path) y estructura: objetos que la forman: botones, menús desplegables, selectores, etc., indicando la

---

función que realiza cada uno)

## **6 Informes**

### 6.1 Relación general

(Listado de todos los Informes de la Aplicación, con el nombre y una descripción breve de su función)

### 6.2 Descripción detallada

(Describir cada Informes indicando su nombre, ubicación (path) y los campos que lo forman, indicando además las diferentes maneras de visualizar el informe: pantalla, impresora o ambas, así como las posibles clasificaciones de los datos, etc.)

## **7 Referencias Cruzadas**

### 7.1 Relación de Bases de Datos por Programa

(Listado de todas las Bases de Datos que utiliza cada Programa)

### 7.2 Relación de Ficheros por Programa

(Listado de todos los Ficheros de Entrada y /o Salida de cada Programa)

### 7.3 Relación de Procesos por Programa

(Listado de todos los Procesos, a nivel macro, de cada Programa)

### 7.4 Relación de Pantallas por Programa

(Listado de todas las Pantallas que utiliza cada Programa)

### 7.5 Relación de Informes por Programa

(Listado de todos los Informes que obtiene cada Programa)

### 7.6 Arbol de llamadas por Programa

---

(Listado de todos programas que llaman y son llamados por cada Programa)

## **8 Diagramas**

- 8.1 Diagrama Entidad / Relación  
(Diagrama E/R, o equivalente, de la Aplicación)
- 8.2 Diagrama de Flujo de Datos  
(DFD de la Aplicación, con los niveles de detalle que sean necesarios)
- 8.3 Diagrama de Cruce Entidad / Relación y Diagrama de Flujo de Datos  
(Diagrama para verificar la consistencia del Diagrama E/R y el DFD)
- 8.4 Diagrama de Flujo de Programas  
(DFP de la Aplicación, indicando las entradas y salidas de cada programa y el encadenamiento entre programas)
- 8.5 Otros Diagramas  
(Incluir cualquier otro diagrama que se considere de interés)

---

### 1.3.6 *Resumen*

A continuación se muestra un resumen de las distintas Fases, Actividades y Tareas que conforman la Metodología de Gestión de Proyectos propuesta.

## **METODOLOGIA DE GESTION DE PROYECTOS**

### **Fase 1: Definición de Requerimientos**

#### Actividad 1.1 Estudio de Viabilidad

- Tarea 1.1.1 Definición del Problema
- Tarea 1.1.2 Estudio del Sistema Actual

#### Actividad 1.2 Planes y Requerimientos

- Tarea 1.2.1 Identificar funcionalidades y necesidades de información
- Tarea 1.2.2 Identificar requisitos de ejecución, seguridad y control

#### Actividad 1.3 Evaluación del software de Aplicación

- Tarea 1.3.1 Revisar software preseleccionado
- Tarea 1.3.2 Definir modificaciones funcionales
- Tarea 1.3.3 Evaluar documentación y arquitectura técnica

#### Actividad 1.4 Identificación del equipo y software de Sistema

- Tarea 1.4.1 Identificar y evaluar alternativas
- Tarea 1.4.2 Establecer estrategia

---

## **Fase 2: Análisis Lógico**

### Actividad 2.1 Análisis del Usuario (conceptual y funcional)

- Tarea 2.1.1 Definir entradas y salidas
- Tarea 2.1.1 Definir funciones y procesos
- Tarea 2.1.3 Definir requisitos de datos
- Tarea 2.1.4 Diseñar Bases de Datos (diseño lógico)
- Tarea 2.1.5 Especificar componentes del Sistema
- Tarea 2.1.6 Emitir especificaciones funcionales para su aprobación/revisión

## **Fase 3: Diseño Físico**

### Actividad 3.1 Diseño técnico del Sistema

- Tarea 3.1.1 Diseñar arquitectura técnica
- Tarea 3.1.2 Diseñar Bases de Datos (diseño físico)
- Tarea 3.1.3 Diseñar procesos del sistema
- Tarea 3.1.4 Diseñar procesos de seguridad y control

### Actividad 3.2 Respaldo Técnico

- Tarea 3.2.1 Diseñar procesos de prueba y requerimientos de prueba
- Tarea 3.2.2 Determinar requerimientos de recursos

### Actividad 3.3 Plan de Instalación

- Tarea 3.3.1 Identificar pasos de la instalación
- Tarea 3.3.2 Establecer requerimientos de personal
- Tarea 3.3.3 Desarrollar enfoque de conversión de software y/o de datos
- Tarea 3.3.4 Desarrollar plan de trabajo de la instalación

### Actividad 3.4 Selección de Hardware y Software

- Tarea 3.4.1 Establecer criterios de selección
- Tarea 3.4.2 Preparar solicitudes de propuesta
- Tarea 3.4.3 Evaluar propuestas
- Tarea 3.4.4 Negociar condiciones del contrato

### Actividad 3.5 Análisis de Costes y Beneficios

- Tarea 3.5.1 Estimar costes de instalación
- Tarea 3.5.2 Estimar costes y beneficios operativos

- 
- Tarea 3.5.3 Documentar consideraciones intangibles
  - Tarea 3.5.4 Resumir resultado económico

#### Actividad 3.6 Revisión y aprobación por la Junta de Proyecto

- Tarea 3.6.1 Publicar especificaciones funcionales y técnicas
- Tarea 3.6.2 Preparar informe y revisar con la Junta de Proyecto
- Tarea 3.6.3 Aprobar Proyecto y revisar prioridad

### **Fase 4: Desarrollo**

#### Actividad 4.1 Organización

- Tarea 4.1.1 Organizar el Proyecto
- Tarea 4.1.2 Establecer estándares
- Tarea 4.1.3 Completar programa de trabajo
- Tarea 4.1.4 Establecer programa de formación de operadores y usuarios

#### Actividad 4.2 Diseño detallado

- Tarea 4.2.1 Completar diseño técnico
- Tarea 4.2.2 Diseñar Programas
- Tarea 4.2.3 Completar diseño de Bases de Datos
- Tarea 4.2.4 Confirmar análisis de costes/beneficios
- Tarea 4.2.5 Revisar estructura de diseño
- Tarea 4.2.6 Preparar datos comunes de pruebas

#### Actividad 4.3 Desarrollo de procedimientos para el usuario

- Tarea 4.3.1 Desarrollar procedimientos de seguridad y control
- Tarea 4.3.2 Completar diseño de formularios
- Tarea 4.3.3 Desarrollar manuales de los usuarios
- Tarea 4.3.4 Obtener formularios
- Tarea 4.3.5 Entrenar al personal de operaciones

#### Actividad 4.4 Preparar la conversión

- Tarea 4.4.1 Completar plan de conversión
- Tarea 4.4.2 Desarrollar procedimientos de conversión
- Tarea 4.4.3 Crear ficheros de conversión
- Tarea 4.4.4 Planificar pruebas del Sistema
- Tarea 4.4.5 Crear modelo de prueba del Sistema

---

#### Actividad 4.5 Desarrollar Software de Sistemas

- Tarea 4.5.1 Desarrollar o modificar software de Sistemas
- Tarea 4.5.2 Desarrollar medio de prueba

#### Actividad 4.6 Instalación de hardware y software

- Tarea 4.6.1 Preparar el lugar de operación
- Tarea 4.6.2 Obtener suministros
- Tarea 4.6.3 Instalar el hardware
- Tarea 4.6.4 Instalar el software

### **Fase 5: Prueba del Sistema**

#### Actividad 5.1 Pruebas del Sistema

- Tarea 5.1.1 Ejecutar pruebas de integración
- Tarea 5.1.2 Ejecutar pruebas del usuario

### **Fase 6: Instalación y Seguimiento**

#### Actividad 6.1 Conversión de ficheros

- Tarea 6.1.1 Convertir ficheros
- Tarea 6.1.2 Completar documentación de mejoras futuras
- Tarea 6.1.3 Vigilar la operación normal del sistema
- Tarea 6.1.4 Transferir el Sistema a un estado rutinario de operación

Estas son, en líneas generales las Fases, Actividades y Tareas primordiales en que se deberá de descomponer un Proyecto desde un punto de vista de una Metodología, aunque no figuran todas las posibles pues su extensión desbordaría la prevista para este tema.



---

## 1.4 Metodología de Desarrollo de Prototipos: PDM80

### 1.4.1 *Definición y ámbito de PDM80*

PDM80 (Prototype Development Methodology 80), desarrollada por Computomata Intl. Co, es una metodología orientada al desarrollo de prototipos y apoyada en tecnología de Base de Datos, en la que se concentra un gran esfuerzo en la involucración del usuario en la definición del problema que se va a abordar, y en la ejecución de las pruebas, y donde se potencia el uso de lenguajes de cuarta generación utilizados como lenguajes de consulta para interrogar a la Base de Datos con el fin de verificar la estructura del prototipo desarrollado, asegurándose de que su diseño responde a las definiciones especificadas.

La idea fundamental de esta metodología es el desarrollo de prototipos. Un prototipo es un modelo de lo que al final se corresponderá con la Base de Datos definitiva, que se someterá a pruebas para comprobar su funcionalidad. De estas pruebas surgirán modificaciones que darán origen a un segundo prototipo, versión mejorada y posiblemente ampliada del primero, el cual se volverá a probar, repitiéndose el proceso hasta alcanzar el prototipo definitivo y su Base de Datos asociada.

La responsabilidad y ejecución de estas pruebas recae, como ya se ha mencionado, en el propio usuario, quien deberá de comprobar que el modelo de Base de Datos desarrollado es capaz resolver todos los problemas planteados en el momento de la definición de los requerimientos del proyecto.

### 1.4.2 *Descomposición de un Proyecto: Fases*

PDM80, al igual que la Metodología de Gerencia de Proyectos ya estudiada, descompone la totalidad del proyecto en seis fases para simplificar la tarea de resolución del mismo. Estas fases son *Auditoría Operacional*, *Diseño Conceptual*,

---

*Definición de la Base de Datos, Análisis Heurístico, Test de Entorno y Revisión de Rendimiento / Calidad*, fases que, como se verá, coinciden bastante con las hasta ahora planteadas.

#### **1.4.2.1 Auditoría Operacional**

Esta primera fase tiene por objeto el auditar la información de la empresa relativa al problema, con el fin de recabar todos los datos necesarios para su resolución. Como primera tarea se deberá de establecer la amplitud y el calendario del proyecto, pasando a continuación a identificar los procesos involucrados en el mismo, construyéndose modelos de actividad (a un macro-nivel en estos momentos) para cada uno de los procesos.

También deberán de prepararse en esta fase las entrevistas con los usuarios, para poder determinar las actividades a desarrollar, obteniendo copias de formularios, procedimientos y políticas de la empresa referentes a los procesos antes identificados.

Se trata, en suma, de recabar la mayor cantidad de información posible sobre el problema que se intenta resolver con medios informáticos.

#### **1.4.2.2 Diseño Conceptual**

El objetivo de esta fase es construir un modelo de información que refleje el esquema conceptual de la Base de Datos. Es muy importante que este modelo esté lo más ajustado posible a la realidad para que el diseño posterior de la Base de Datos pueda cumplir con sus objetivos.

Se comenzará por establecer un plan, calendario y presupuesto, pasando a continuación a construir el modelo del Sistema actual o modelo “As Is”, es decir, tal y como es en la actualidad. A continuación se buscarán nuevos modelos de información, evaluando, por

---

ejemplo, otras alternativas de software, que nos permitirán construir el modelo de información “Should be”, es decir, como debería de ser.

Para finalizar la fase se identificarán los requisitos de mantenimiento y de la Aplicación, se construirá una matriz y un plan de implementación, en el que se definirán Esquemas y Subesquemas de la Base de Datos y se definirán los prototipos a desarrollar posteriormente, emitiendo un informe de fin de fase como última de las tareas de la misma.

### **1.4.2.3 Definición de la Base de Datos**

Como su nombre indica, esta fase tiene por objeto el definir la Base de Datos que será soporte de nuestro prototipo.

Se comenzará por definir un plan e identificar y seleccionar las Entidades que van a ser utilizadas, los Atributos de las mismas y la relación entre ambos, lo que nos permitirá establecer un perfil de la Base de Datos en el Diccionario de Datos, herramienta base de toda la documentación.

El siguiente paso será la identificación de rutinas de proceso para los datos que las requieran, pasándose a continuación al proceso de Normalización, llevando la Base de Datos a Tercera Forma Normal.

Se convierte, entonces, el modelo conceptual de la Base de Datos en el modelo de datos soportado por el Gestor de Base de Datos de que se disponga en la Instalación, definiendo con los lenguajes adecuados el Esquema y los Subesquemas ya definidos desde un punto de vista lógico en la fase anterior.

La conclusión de esta fase es el establecimiento de la Base de Datos Prototipo, que será utilizada en las fases siguientes.

---

#### **1.4.2.4 Análisis Heurístico**

En estos momentos, estamos preparados para realizar la simulación exhaustiva del posterior funcionamiento de la Base de Datos, por lo que entregaremos nuestra Base de Datos Prototipo al usuario para realizar las demostraciones heurísticas. Estas demostraciones consistirán en una serie de preguntas a la Base de Datos utilizando lenguajes de consulta (query languages), para verificar que la estructura diseñada se ajusta al problema definido con anterioridad.

Como resultado de las demostraciones, se podrán evaluar las respuestas de la Base de Datos a las preguntas realizadas, lo que nos permitirá finalizar la estructura interna de la Base de Datos y definir del mismo modo la estructura de mantenimiento.

Como fin de fase, se codificarán e implementarán los programas de mantenimiento y se desarrollarán las políticas de implementación, procesos y formularios necesarios.

#### **1.4.2.5 Test de Entorno**

Se llevarán a cabo, en esta fase, las pruebas de integración del Sistema que se ha desarrollado, con los sistemas ya existentes en la Instalación, realizándose ejecuciones en paralelo con el Sistema anterior con el objeto de verificar resultados.

También se definirán en esta fase los Informes estándar que se obtendrán de ella, así como de los Informes eventuales que podrían ser necesarios en un momento dado.

---

#### **1.4.2.6 Revisión Rendimiento / Calidad**

La última de las fases de PDM80 consiste en una Auditoría del rendimiento y la calidad del producto, antes de completar su desarrollo.

Se deberán de identificar parámetros de rendimiento, compromisos, el perfil uso / respuesta, y afinar la Base de Datos.

Asimismo se identificarán los planes para la reorganización y recarga de la Base de Datos, se revisarán los controles de seguridad, la validez y consistencia de los datos, etc., estableciéndose un plan de auditorías periódicas de la Base de Datos, que aseguren la calidad de las Aplicaciones.

Como fin de fase, se comprobará que toda la documentación es la adecuada, tanto la de la Base de Datos como la de los Sistemas de Aplicación.

---

## 1.5 Bibliografía

[Andersen, 1984]. Arthur Andersen & Co. *Method / I*. 1984.

[Bentley, 1982] Colin Bentley. *Computer Project Management*. C. Heyden & Son Ltd. 1982.

[Bohem, 1976] B.W.Bohem. *Software Engineering*. IEEE Trans. Computers, Diciembre 1976, pp. 1226-1241.

[Cardenas, 1985] Alfonso Cárdenas. *PDM80. Prototype Development Methodology*. Computomata Intl. Co. 1985.

[Caridad, 1991] Serafín Caridad. *Metodologías de Desarrollo de Sistemas en un Ambiente de Control de Calidad*. Separata de Los Medios Informáticos de la Gestión Empresarial Pública y Privada. Publicaciones de la Fundación Alfredo Brañas. Velograf. S.A. 1991.

[Caridad, 1983] Serafín Caridad. *Metodología de Programación Estructurada*. Banco Pastor, S.A. 1983.

[Caridad - Souto, 1985] Serafín Caridad y Ramón Souto. *Control de Calidad de Seguimiento Proyectos*. Banco Pastor, S. A. 1985.

[Grafton, 1986] William Grafton. *Test to Production Migration Procedures for DB/DC*. Computomata Intl. Co, 1986.

[Mills, 1976] Harlan D. Mills. *Improved Programming Technologies*. IBM. IPTO Support Group. 1976.

[Peat, 1980] Peat, Marwick, Mitchell & Co. *System Development Manual*. 1980.

[Souto, 1986] Ramón Souto. *Procedimiento de Migración de Aplicaciones a Producción*. Banco Pastor, S. A. 1986.

---

## **1.6 Prácticas**

Para completar el estudio de este capítulo, se recomiendan las prácticas siguientes:

### **1.6.1 *Formación de Equipos de Trabajo***

Una buena idea sería formar Equipos de Trabajo para acometer el desarrollo de un Proyecto, siguiendo las pautas expuestas en cualquiera de las metodologías estudiadas, o de cualquier otra metodología que se conozca o de la que se disponga.

### **1.6.2 *Análisis de una Aplicación***

Como ejemplo, se podría abordar el diseño y desarrollo de una Aplicación cualquiera que sea bien conocida.

---

## 1.7 **Apéndice A: Walkthroughs Estructurados**

En términos generales, un Walkthrough es un tipo de reunión, con unas características especiales y con unas pautas predeterminadas, que puede realizarse durante la Fase de Análisis de una Aplicación o durante la Fase de Desarrollo de la misma, es decir, en cualquier momento del Ciclo de Vida de un Proyecto.

Los Walkthroughs que vamos a detallar a continuación son los que se llevan a cabo durante la Fase de Desarrollo, concretamente para la depuración y puesta a punto de Programas.

### 1.7.1 *Consideraciones Generales*

Un Walkthrough Estructurado de Programación es el nombre dado a una serie de revisiones, cada una de ellas con distintos objetivos, hechas durante las distintas fases del desarrollo de un programa. Se pueden especificar hasta cuatro clases diferentes de Walkthroughs:

De Diseño

De Preparación de los Test del Programa y de los Test de Datos

De Codificación

De Pruebas del Programa

Estos Walkthroughs se realizarán, como es lógico, en el momento que sea oportuno, dependiendo del grado de desarrollo del Programa.

Como en definitiva se trata de revisar el trabajo de una persona, es muy importante que se cumplan las normas que se dan a continuación para evitar que estas personas, cuyo trabajo va a ser comprobado, se sientan incómodas o incluso preocupadas por culpa del Walkthrough, con lo que no solo no se alcanzarían los beneficios que la reunión proporciona sino que incluso podrían producir efectos totalmente contrarios y muy perjudiciales para el desarrollo del software.



---

### **1.7.2 Walkthroughs Obligatorios**

Normalmente, y de acuerdo con los Estándares definidos en las organizaciones que utilizan este tipo de reuniones, algunos de los Walkthroughs anteriores se consideran obligatorios. Por lo general, suelen ser obligatorios los de Codificación y Pruebas. El primero se realiza normalmente después de la primera compilación sin errores, y el segundo después de las pruebas del programa.

Por lo general, la realización o no del resto de los Walkthroughs dependerá del criterio del programador, que puede convocarlos si lo considera oportuno, aunque esto depende por supuesto de los Estándares definidos por cada empresa.

### **1.7.3 Asistentes**

Las normas relativas a las personas que pueden asistir a un Walkthrough se pueden resumir en los puntos que se a continuación se citan. Debemos de recordar que es de gran importancia que la totalidad de estas normas y de las que se verán a continuación se cumplan, para la buena marcha del Walkthrough.

- 1 Las personas que participan en la reunión deberán de ser elegidas por el programador que convoca el Walkthrough.
- 2 El número de asistentes, incluido el programador cuyo trabajo va a ser revisado, será de dos a tres personas.
- 3 El Personal Directivo no podrá formar parte de un Walkthrough.
- 4 El Gerente de un Proyecto no podrá formar parte de los Walkthroughs de su propio Proyecto, aunque sí de los demás.

---

#### **1.7.4 Duración**

- 1 La duración máxima de un Walkthrough será de dos horas.
- 2 Si en ese tiempo no se han podido cubrir todos los puntos, se convocará un nuevo Walkthrough más adelante.

#### **1.7.5 Preparación de la Reunión**

- 1 Como mínimo 48 horas antes de la reunión, el programador entregará a los participantes una Guía de Convocatoria de Walkthrough (ver figura 1.2), con los datos necesarios para llevar a cabo la reunión, así como un índice de los puntos a cubrir en la misma.
- 2 Además, el programador entregará copias de todo el material a revisar a cada uno de los participantes.

#### **1.7.6 La Reunión**

- 1 Se seleccionará un participante para que actúe como Moderador y otro como Secretario, aunque ambas funciones podrán ser realizadas por la misma persona.
- 2 La misión del Moderador será la de coordinar la Reunión y la de tomar nota de las acciones a llevar a cabo en la Lista de Acción, que entregará al programador al final de la Reunión.
- 3 La parte principal de un Walkthrough es la detección de errores y la revisión de los estándares. Sin embargo, se podrá comenzar la reunión con una breve explicación por parte del programador del trabajo que ha realizado, si existe una razón que así lo justifique.
- 4 Bajo ningún concepto se intentará buscar soluciones a un problema. Esa será

---

labor posterior del programador.

- 5 Cuando se encuentre un punto conflictivo, el programador podrá explicar su trabajo. Si se llega a la conclusión de que en ese punto existe un problema, se anotará en la Lista de Acción.
- 6 No se entrará en discusión sobre errores triviales (errores de sintaxis, puntuaciones omitidas, etc.), sino que se subrayarán en una de las copias del material, y al final se le entregarán al programador.
- 7 Todos los miembros participarán de una manera activa en la Reunión.

### **1.7.8 Trabajo Posterior**

- 1 El programador escribirá en la Lista de Acción las resoluciones tomadas, dando explicaciones breves pero suficientes para indicar las acciones que ha llevado a cabo.
- 2 Esta Lista de Acción se hará llegar a los participantes en un plazo no superior a los dos días a partir de la fecha de celebración de la Reunión.
- 3 Cada vez que se lleve a cabo un Walkthrough, se deberá enviar notificación a la estructura jerárquica superior que corresponda (normalmente a Control de Calidad, si existe en la empresa), utilizando para ello la Guía de Convocatoria de Walkthrough.
- 4 En dicha Guía no se mencionará si hubo errores, ni cuantos, ni quien los encontró, sino solamente la fecha de la Reunión, los participantes, el nombre del programa y el tipo de Walkthrough, dado que *la misión de la Reunión será única y exclusivamente la detección de errores, y en ningún caso se utilizará como método de evaluación de programadores.*

---

### **1.7.9 Interrupción de un Walkthrough**

- 1 Si durante la Reunión se detecta un “Error Grave” (Ej.: Error de concepto que obligue a reconsiderar el diseño, etc.), se cancelará la Reunión en ese momento. El programador, con tiempo suficiente, rehará el material, y convocará un nuevo Walkthrough.

### **1.7.10 Consideraciones Finales**

- 1 Los programas pequeños, que se realicen en menos de dos días, no deberían de tener que pasar ningún Walkthrough obligatoriamente. No obstante, podrán ser objeto de una Reunión si el programador lo considerase necesario.
- 2 Las reuniones rápidas (informales) entre programadores no precisarán ser notificadas ni necesitarán de una Guía de Convocatoria. Se incluyen en este tipo de reuniones las revisiones de código entre programadores, en las que un programador revisa el código de otro y viceversa.

Hasta aquí, hemos revisado las normas y la mecánica operativa de los Walkthroughs y queda por mencionar los beneficios que de dichas reuniones se derivan. En primer lugar, los Walkthroughs constituyen el medio idóneo para revisar el cumplimiento de los Estándares de una Organización y para verificar que el trabajo está siendo desarrollado con un nivel de calidad aceptable, y como beneficio añadido, sirven también para formar a nuevos programadores (que pueden participar como oyentes en los Walkthroughs) en la disciplina de programación de la empresa. También podríamos añadir que los Walkthroughs Estructurados proporcionan seguridad y tranquilidad a los programadores, ya que su trabajo ha sido refrendado por sus propios compañeros, libremente elegidos por ellos. Pero quizá lo más importante sea que los Walkthroughs constituyen un estupendo modo no solo de localizar errores sino de evitar incluso que se lleguen a cometer antes de que el programa pase a un estado de producción, donde ya será mucho más costoso cualquier cambio que se deba de realizar.

---

Por otra parte, cualquier tipo de revisión del trabajo de una persona se convierte en un tema cuando menos delicado, muy difícil a veces de realizar sin herir la sensibilidad de quien va a ver su esfuerzo sometido a juicio. Por ello insistimos una vez más que es de todo punto necesario que las normas de los Walkthroughs se cumplan estrictamente para que se puedan alcanzar los beneficios esperados. De lo contrario, el porvenir de los Walkthroughs en cualquier organización se verá seriamente cuestionado, y con el tiempo terminarán incluso por desaparecer.

### **1.7.11 Formularios**

Como complemento se muestran dos formularios: uno para convocatorias de Walkthroughs y una Lista de Acción. Estos formularios se pueden tomar como ejemplo, aunque lo más probable es que cada Organización diseñe los suyos propios.

CONVOCATORIA DE WALKTHROUGH		Fecha: ..../..../....																		
Convoca: Fecha: Lugar: Hora: Programa: Tipo de Walkthrough:																				
Material que se acompaña: 1 2 3 4																				
Objetivos: Por favor, revisar: 1 2 3 4																				
Asistentes: <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%;"></th> <th colspan="2" style="text-align: center;"><u>Presente</u></th> </tr> <tr> <th></th> <th style="text-align: center;">Si</th> <th style="text-align: center;">No</th> </tr> </thead> <tbody> <tr> <td>1</td> <td style="text-align: center;">Si</td> <td style="text-align: center;">No</td> </tr> <tr> <td>2</td> <td style="text-align: center;">Si</td> <td style="text-align: center;">No</td> </tr> <tr> <td>3</td> <td style="text-align: center;">Si</td> <td style="text-align: center;">No</td> </tr> <tr> <td>4</td> <td style="text-align: center;">Si</td> <td style="text-align: center;">No</td> </tr> </tbody> </table>				<u>Presente</u>			Si	No	1	Si	No	2	Si	No	3	Si	No	4	Si	No
	<u>Presente</u>																			
	Si	No																		
1	Si	No																		
2	Si	No																		
3	Si	No																		
4	Si	No																		

Fig. 1.2 Formulario de convocatoria de un Walkthrough

---

LISTA DE ACCION DE WALKTHROUGH		Fecha: ..../..../....
Convoca:		
Fecha:		
Lugar:		
Hora:		
Programa:		
Tipo de Walkthrough:		
Cuestiones		Resolución

Fig. 1.3 Lista de acción de un Walkthrough