



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

"ACATLÁN"

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: TADEO TRINIDAD

MA. ADELINA

FECHA: 03/OCTUBRE/2002

FIRMA: [Firma manuscrita]

FACTORES QUE INTERVIENEN EN EL ÉXITO DEL DESARROLLO DE SISTEMAS

ANÁLISIS DE UN SISTEMA DE INFORMACIÓN REAL

CASO PRÁCTICO: EL DESARROLLO DE SISTEMAS EN BANCRECER



MEMORIAS DE DESEMPEÑO PROFESIONAL QUE PARA OBTENER EL TÍTULO DE LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA

MA. ADELINA TADEO TRINIDAD

ASESOR: M. EN C. SARA CAMACHO CANCINO



OCTUBRE, 2002

TESIS CON FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

A mi mamá,

*Por todos esos años de esfuerzos y sacrificios,
y por su ejemplo de fe, esperanza y valor
que hoy han hecho posible la culminación de este trabajo.*

Gracias mami . . .

A Marco,

*Mi amigo y compañero en este camino,
quien no solo comparte la vida
conmigo, sino también la misma profesión.*

A:

*Angie, Vicky, Eva, Memo, José,
Dany, Carlos, Iván, Jorge, Luis, Ivanna,
Chente, Gustavo, Gustavo B., Tere,
Ney, Charo, Max*

A la memoria de mi papá

8

1931
MAY 10 11 15 AM '31

Reconocimientos:

A Sara, por su apoyo incondicional para la realización de este trabajo, por su confianza, impulso y dedicación...

A Gustavo y Angie, por su valiosa contribución en la presentación y la aportación de sus ideas y creatividad...

A Rosa Aida, Oscar, Andrés y Blanca por su colaboración en la revisión de estas memorias...

A BanCrecer, porque todo lo que he aprendido hasta ahora laboralmente, lo he aprendido ahí...

A la UNAM, por ser la cuna de tantos y tantos profesionistas que hoy trabajan por México, y por haberme dado el privilegio de formar parte de su comunidad...

Muchas Gracias.

TEXIS C^oN
FALLA DE ORIGEN

Reconocimientos:

A Sara, por su apoyo incondicional para la realización de este trabajo, por su confianza, impulso y dedicación...

A Gustavo y Angie, por su valiosa contribución en la presentación y la aportación de sus ideas y creatividad...

A Rosa Aida, Oscar, Andrés y Blanca por su colaboración en la revisión de estas memorias...

A BanCrecer, porque todo lo que he aprendido hasta ahora laboralmente, lo he aprendido ahí...

A la UNAM, por ser la cuna de tantos y tantos profesionistas que hoy trabajan por México, y por haberme dado el privilegio de formar parte de su comunidad...

Muchas Gracias.

TEXIS CON
FALLA DE ORIGEN

ÍNDICE

INTRODUCCIÓN	13
CAPÍTULO I. CICLO DE VIDA DE UN SISTEMA. CONCEPTOS BÁSICOS	19
CAPÍTULO II. LA IMPORTANCIA DE SEGUIR UNA METODOLOGÍA DE TRABAJO	33
CAPÍTULO III. LA IMPORTANCIA DE UN BUEN ANÁLISIS DE REQUERIMIENTOS	45
CAPÍTULO IV. LA IMPORTANCIA DEL DESARROLLO DE SOFTWARE CON CALIDAD.....	59
CAPÍTULO V. LA IMPORTANCIA DE LA DOCUMENTACIÓN DE SISTEMAS	73
CAPÍTULO VI. LA IMPORTANCIA DE LA PLANEACIÓN EFECTIVA DE PROYECTOS.....	91
CAPÍTULO VII. LA IMPORTANCIA DE LOS RECURSOS HUMANOS.....	103
CONCLUSIONES	113
 ANEXOS	
Acerca de los anexos	121
Solicitud de inicio de proyecto	125
Solicitud de servicios al área de Sistemas	127
Gráfica de Gantt	129
Checklist	131
Reporte de avance	133
Gráfica de eficiencia	135
Gráfica de esfuerzo	137
Reporte de productividad	139
Reporte de asignación	141
Reporte de liberación	143
Relación de recursos	145
Breve descripción de 'Altamira'	147
Trayectoria profesional	149
Biografía: Jack Kilby	151
Desarrollo de sistemas	153
Plegaria de un programador	155
 BIBLIOGRAFÍA	 159

INTRODUCCIÓN

Como hemos observado, hoy en día la economía global depende más de sistemas automatizados que en épocas pasadas. El empleo de computadoras está aumentando en prácticamente todos los campos del quehacer humano. En una era de aumento constante de los costos, los correspondientes a la computación se han estado reduciendo drásticamente gracias a los avances en la tecnología tanto de hardware como de software. Computadoras que podrían haber ocupado enormes espacios y costado millones de dólares hace algunas décadas ahora pueden grabarse en la superficie de pastillas de silicio más pequeñas que la uña de un dedo y son de bajo costo. Por cierto, el silicio es uno de los materiales más abundantes en la Tierra; es un ingrediente de la arena común.

La tecnología de los chips de silicio ha hecho a la computación tan económica que actualmente están en uso aproximadamente 200 millones de computadoras de propósito general en todo el mundo auxiliando a la gente en los negocios, la industria, el gobierno y su vida personal. Esta cifra va aumentando con los años y ha llevado a los equipos de desarrollo a enfrentarse con una nueva etapa de procesos y estándares de calidad.

En la actualidad, son muchos los procesos de desarrollo de software que existen. Con el pasar de los años, la ingeniería de software ha introducido y popularizado una serie de estándares para medir y certificar la calidad, tanto del sistema a desarrollar, como del proceso de desarrollo en sí. Asimismo, se han publicado muchos libros y artículos relacionados con este tema, con el modelado de procesos del negocio y la reingeniería. Con todo, ha surgido un número creciente de herramientas automatizadas para ayudar a definir y aplicar un proceso de desarrollo de software efectivo.

Sin embargo, todavía quedan las siguientes preguntas:

- ¿Cómo explicamos la alta incidencia de fallos en los proyectos de software?
- ¿Por qué existen tantos proyectos de software víctimas de retrasos, presupuestos sobregirados y con problemas de calidad?
- ¿Cómo podemos tener una producción o una economía de calidad, cuando nuestras actividades diarias dependen de la calidad del sistema?

Tal vez suene ilógico pero, a pesar de los avances que ha dado la tecnología, aún existen procesos de producción informales, parciales y en algunos casos no confiables.

En nuestro país somos partícipes de este problema a diario, en donde se ha vuelto común la compra de sistemas extranjeros, para luego "personalizarlos" supuestamente a la medida de las empresas, como ha sido el caso de 'Altamira', el principal sistema informático de BanCreceer. Tal personalización, la mayoría de las veces, termina retrasando el proyecto en meses, o incluso en años.

El reemplazo de plataformas y tecnologías obsoletas, la compra de sistemas completamente nuevos, las modificaciones de todos o de casi todos los programas que forman un sistema, entre otras razones, llevan a desarrollar proyectos en períodos de tiempo sumamente ajustados y en algunos casos irreales; esto ocasiona que se omitan muchos pasos importantes en el ciclo de vida de desarrollo, entre estos, la definición detallada de los requerimientos.

Debemos estar conscientes que conforme aumenta la complejidad del mundo, los sistemas informáticos también deberán crecer en complejidad. En ellos se encuentran diversos componentes de hardware y software que se comunican a grandes distancias mediante una red, misma que está vinculada a bases de datos que, a su vez, contienen enormes cantidades de información.

Una característica del desarrollo de sistemas contemporáneo es reducir precisamente el período de desarrollo. Cuando los plazos se encuentran muy cerca uno del otro es absolutamente necesario contar con un diseño bien cimentado.

Además, hay otro aspecto de la vida moderna que demanda un diseño sólido: las adquisiciones corporativas. Cuando una empresa adquiere a otra (como es el caso de BanCreceer, recientemente adquirido por Banorte), la nueva organización debe tener la posibilidad de modificar aspectos importantes de un proyecto de desarrollo que esté en progreso (la herramienta de desarrollo, el lenguaje de codificación y otras cosas). Un anteproyecto bien diseñado facilitará la conversión. Si el diseño es sólido, un cambio en la implementación podrá darse sin problemas.

En el presente trabajo se hace un estudio analítico del área de Sistemas en BanCreceer, donde se expone en cada capítulo una serie de sugerencias, observaciones, análisis y comentarios sobre el desempeño profesional en las áreas de informática.

En el Capítulo I se presenta una reseña documental sobre el ciclo de vida de un sistema. El objetivo es presentar la metodología clásica recomendada en el desarrollo de un sistema de información, tomando estos conceptos como una base teórica a partir de la cual se establecerá una comparación con la metodología empleada en la práctica para el desarrollo de proyectos en el área de Sistemas BanCreceer.

El Capítulo II describe brevemente la metodología estándar en dicha Institución, dando una pequeña definición de las responsabilidades a cargo de cada área involucrada en el departamento de Desarrollo de Sistemas. El principal objetivo es resaltar las ventajas de emplear una metodología efectiva en las áreas de trabajo.

El Capítulo III tiene por objeto mostrar algunas técnicas empleadas en la etapa de análisis de requerimientos, así como ciertos conceptos fundamentales para la comprensión de esta actividad. Esencialmente se hace énfasis en esta etapa del ciclo de vida ya que se define como la base para el óptimo desarrollo del sistema.

En el Capítulo IV se trata la problemática que enfrenta un programador para producir software de calidad, se dan además algunas sugerencias y observaciones sobre cómo podemos mejorar nuestras habilidades en esta área de la informática. El principal objetivo es mostrar cómo la programación y las estrategias que se empleen conllevan a desarrollar productos de calidad. En este capítulo se hace mención de las bases de la programación estructurada y de la programación orientada a objetos.

El tema expuesto en el Capítulo V es acerca de la documentación, etapa que frecuentemente es ignorada o no se le da la importancia que merece, siendo que ésta es el principio de la comprensión de la operativa y finalidad de un sistema. El objetivo es rescatar del olvido esta importante actividad y resaltar su necesaria ejecución. Se muestran, a grandes rasgos, las principales características deseables en todo documento sobre la funcionalidad de un sistema.

En el Capítulo VI se hace mención de uno de los principales problemas en la liberación de todo proyecto: la falta de una planeación estratégica, actividad sin la cual no se pueden lograr los objetivos de cumplir a tiempo, en las fechas planeadas y con los costos estimados. El objetivo aquí es mostrar algunos conceptos importantes de la administración de proyectos, que involucra tanto procesos como recursos empleados, y comparar cómo en la realidad son o no considerados y, sobre todo, cuáles han sido las repercusiones de llevarlos a cabo o no.

Por último, en el Capítulo VII se destaca la importancia de los recursos humanos, visto como el factor más importante en toda organización. Aquí se mencionan algunas de las cualidades requeridas en un egresado de las áreas de informática, así como las características de los puestos de trabajo en Sistemas BanCreceer.

También se mencionan algunos aspectos sobre la integración de los grupos de trabajo con personal tanto interno como externo y las ventajas de contar con personal altamente motivado.

Este trabajo está dirigido a la población estudiantil de la carrera de Matemáticas Aplicadas y Computación, tanto alumnos como ex alumnos, así como a los profesionistas en general cuyas carreras y actividades laborales sean afines con el área de sistemas, y es importante señalar que consiste fundamentalmente en un análisis derivado de las actividades desempeñadas día a día en el área de desarrollo de sistemas, y que no se incluirá en el texto documentación alguna sobre código, diagramas o algún otro elemento referente al diseño lógico o físico del sistema informático de BanCrecer.

También, cabe señalar que la investigación teórica presentada en el trabajo es parte del marco teórico con que éste se sustenta, mas no tiene el objeto de mostrar detalladamente todos los conceptos involucrados, ya que el tema es en sí muy amplio, por lo que solo se toman como punto de partida para poder realizar una comparación entre la teoría y la realidad. Sin embargo, se pretende que la teoría expuesta sea suficiente y clara.

Por otro lado, aunque el hardware también es un factor importante que debe considerarse en la planeación de todo sistema, es un tema que no se trata en este trabajo, ya que el alcance del mismo solo abarca la parte del software en cuanto a desarrollo y administración de recursos no técnicos ni de infraestructura.

OBJETIVO GENERAL.

Contribuir en el área académica de sistemas computacionales en general, y en particular en la carrera de Matemáticas Aplicadas y Computación en aquellas materias que abarca la ingeniería de software, al mostrar, mediante casos reales, la trascendencia de los principales factores que forman parte del ciclo de vida de un sistema, con el fin de concientizar a los jóvenes estudiantes y a quienes ya se encuentran laborando en el área, sobre la importancia de contar con sólidas bases teóricas en estos temas para poder llevar a cabo un desempeño satisfactorio en nuestro campo de trabajo.

OBJETIVOS PARTICULARES.

1. Mostrar la problemática de la implantación de un sistema de información real, analizando sus distintas etapas evolutivas de acuerdo a la metodología para el desarrollo de sistemas.
2. Resaltar la importancia que tiene la ingeniería de requerimientos dentro del ciclo de desarrollo de sistemas.
3. Dar a conocer las diferentes alternativas que existen para identificar requerimientos.
4. Presentar algunos conceptos de ingeniería de software primordiales para garantizar la calidad del producto.
5. Enfatizar la importancia de la adecuada planeación de los proyectos y la intervención de los recursos humanos en los mismos.
6. Señalar por qué es importante y cómo nos beneficia la adecuada documentación de un sistema.
7. Proporcionar un apoyo teórico sobre los principales conceptos del ciclo de vida clásico en el desarrollo de sistemas.
8. Realizar una crítica constructiva sobre el desempeño del equipo de trabajo y las actividades realizadas que conllevan a la liberación de los proyectos.

CAPÍTULO II

CICLO DE VIDA DE UN SISTEMA. CONCEPTOS BÁSICOS

"Otro defecto de la naturaleza humana es que todos quieren construir, pero nadie está dispuesto a encargarse del mantenimiento"...

- Kurt Vonnegut

INTRODUCCIÓN

En la actualidad, para muchas organizaciones los sistemas de información basados en computadoras son el corazón de las actividades cotidianas y objeto de gran consideración en la toma de decisiones. Las empresas consideran con mucho cuidado las capacidades de sus sistemas de información cuando deciden ingresar o no en nuevos mercados o cuando planean la respuesta que darán a la competencia.

Todo gira en torno a una visión. Un sistema complejo toma forma cuando alguien tiene la visión de cómo la tecnología puede mejorar las cosas. Los desarrolladores tienen que comprender completamente la idea y mantenerla en mente mientras crean el sistema que le dé forma. El auge de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quien tiene la idea y el desarrollador.

A medida que las computadoras son empleadas cada vez más por personas que no son especialistas en computación, el rostro del desarrollo de sistemas adquiere una nueva magnitud. Los propios usuarios emprenden ya el desarrollo de algunos de los sistemas que ellos emplean. Por ello, es necesario comprender que un proyecto de desarrollo de un sistema computacional consta de varios pasos, los cuales ayudan a traducir las necesidades del cliente en un modelo de sistema que utiliza los siguientes componentes: software, hardware, personas, bases de datos, documentación y procedimientos.

En el presente capítulo se mencionan de manera general las etapas involucradas en el ciclo de vida de un sistema, según el método clásico y cuyas actividades han estado y estarán presentes en cualquier lugar donde exista un sistema.

Cabe mencionar que lo que aquí se presenta no son meras definiciones técnicas, ya que, aunado al proceso de investigación en la bibliografía empleada, estos conceptos se complementan con puntos de vista y comentarios personales, con el objeto de enriquecer y aportar algo más que las definiciones en los libros.

CONCEPTOS FUNDAMENTALES

El análisis de sistemas, en una organización o empresa, es el proceso de estudiar su situación con la finalidad de observar cómo trabaja y decidir si es necesario realizar una mejora. La información reunida durante esta etapa sirve como base para crear varias estrategias de diseño y en esta parte se debe decidir qué estrategias seguir.

El método del ciclo de vida para desarrollo de sistemas es el conjunto de actividades que los analistas, programadores y usuarios realizan para desarrollar e implantar un sistema de información. En general, el ciclo de vida clásico para el desarrollo de sistemas consta de las siguientes actividades:

1. Investigación y análisis preliminar
2. Análisis y determinación de requerimientos
3. Diseño del sistema
4. Desarrollo de software
5. Prueba del sistema
6. Implantación
7. Evaluación

1. INVESTIGACIÓN Y ANÁLISIS PRELIMINAR

La principal finalidad de realizar un análisis preliminar consiste en evaluar las necesidades del proyecto, lo cual no abarca un estudio de diseño ni la recopilación de datos para hacer una descripción completa del sistema; en cambio, sirve para evaluar las ventajas del proyecto propuesto y cimentar bases bajo un juicio sobre su factibilidad de desarrollo.

El objeto de realizar este paso es para evaluar qué tan bien una solicitud de proyecto se enlaza con las necesidades estratégicas y organizacionales de la empresa, tanto en la productividad, mejora en los productos y servicios, como en la toma de decisiones a nivel gerencial.

Cuando se formula la solicitud comienza la primera actividad de sistemas: la *investigación preliminar*. Esta actividad tiene tres partes: aclaración de la solicitud, estudio de factibilidad y aprobación (o rechazo) de la solicitud.

Aclaración de la solicitud. Muchas solicitudes que provienen de los usuarios no están formuladas de manera clara. Por consiguiente, antes de considerar cualquier investigación de sistemas, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea. Si éste tiene una buena idea de lo que necesita pero no está seguro cómo expresarlo, entonces bastará con algunas entrevistas y acaso llamadas telefónicas. Pero si el solicitante pide ayuda sin saber qué es lo que se debe desarrollar o dónde se encuentra el problema, la aclaración del mismo se vuelve más difícil. En cualquier caso, antes de seguir adelante, la solicitud de proyecto debe estar claramente planteada. Por ello se recomienda el llenado de una forma de solicitud de proyectos de sistemas. Como ejemplo, en el anexo 1a se muestra un formato para tal efecto.

Estudio de factibilidad. Un resultado importante de la investigación preliminar es la determinación de que el sistema solicitado sea factible. En la investigación preliminar existen tres aspectos relacionados con el estudio de factibilidad:

1. Factibilidad técnica. El trabajo para el proyecto, ¿puede realizarse con el equipo actual, la tecnología existente de software y el personal disponible? Si se necesita nueva tecnología, ¿cuál es la posibilidad de desarrollarla? En el análisis técnico, el analista evalúa los principios técnicos del sistema y al mismo tiempo recoge información adicional sobre el rendimiento, fiabilidad, características de mantenimiento y productividad. Los resultados obtenidos del análisis técnico son la base para determinar sobre si continuar o abandonar el proyecto, si hay riesgos de que no funcione, no tenga el rendimiento deseado, o si las piezas no encajan perfectamente unas con otras.
2. Factibilidad económica. Al crear el sistema, ¿los beneficios que se obtienen serán suficientes para aceptar los costos?, ¿los costos asociados con la decisión de no crear el sistema son tan grandes que se debe aceptar el proyecto? El análisis económico incluye lo que llamamos, el análisis de costos-beneficios, significa una valoración de la inversión económica comparado con los beneficios que se obtendrán en la comercialización y utilidad del producto o sistema. Muchas veces en el desarrollo de sistemas de computación, estos factores son intangibles y resulta difícil evaluarlos, esto varía de acuerdo a las características del sistema. El análisis de costos-beneficios es una fase muy importante ya que de ella depende la posibilidad de desarrollo del proyecto.
3. Factibilidad operacional. Si se desarrolla e implanta, ¿será utilizado el sistema?, ¿existirá cierta resistencia al cambio por parte de los usuarios que dé como resultado una disminución de los posibles beneficios de la aplicación? El análisis operacional debe ser tan minucioso que pueda proporcionarnos información valiosa para desarrollar un sistema perfectamente acorde a las necesidades del usuario.

Aprobación de la solicitud. Aquellos proyectos que son deseables y factibles deben incorporarse en los planes de la empresa. Después de aprobar la solicitud de un proyecto se estima su costo en cuanto al tiempo necesario para terminarlo y las necesidades de personal; con esta información se determina dónde ubicarlo dentro de la lista existente de proyectos.

Las funciones de quienes trabajan en el análisis preliminar son:

- Aclarar y entender la solicitud del proyecto:
 - ¿Qué se está haciendo?
 - ¿Qué es lo que se necesita?
 - ¿Por qué?
- Determinar la dimensión del proyecto.
- Señalar los costos y beneficios de las alternativas apropiadas.
- Determinar la factibilidad técnica y operativa de otras alternativas.
- Informar los hallazgos a la gerencia con recomendaciones y subrayando la aceptación o rechazo de la propuesta.

Una vez que el analista completa las entrevistas iniciales y determina que deberá realizarse el análisis detallado del sistema, se deberá comunicar formalmente a los usuarios solicitantes y a la gerencia de sistemas, bajo el entendimiento de lo que se debe realizar y el enfoque general hacia esta meta, proporcionando un punto de verificación en el que el solicitante puede evaluar si el analista entiende lo que se requiere dando también a la gerencia de sistemas la oportunidad de evaluar el enfoque y la cantidad de recursos que se van a emplear durante el análisis.

El documento que resulta de este análisis preliminar es el reporte de la propuesta para realizar un análisis detallado, el cual representa un acuerdo entre el analista de sistemas y los usuarios.

Dicho reporte deberá incluir lo siguiente:

- Una definición clara y concisa de las razones para realizar el análisis.
- Un planteamiento específico referente a los requerimientos del desempeño del sistema propuesto.
- Una definición del alcance del análisis.
- Una identificación de los hechos que probablemente necesiten recopilarse durante el análisis y de las fuentes potenciales donde pueden obtenerse esos hechos.
- Un programa que indique los eventos principales del análisis.

Razones para iniciar el análisis de sistemas

Las razones básicas para iniciar un análisis de sistemas son las siguientes:

- a) Necesidad de resolver un problema.- Puede suceder que el actual sistema no esté funcionando como se esperaba, entonces se acude al analista de sistemas para que corrija esta anomalía.
- b) Nuevas necesidades.- Esto ocurre cuando surgen nuevas disposiciones en la organización. Independientemente de la causa que dé origen a la nueva necesidad, el analista identificará las modificaciones o adiciones que deben hacerse al sistema, con el fin de que la empresa pueda satisfacer dicha necesidad.

- c) Implantación de una nueva tecnología.- Puede ser el caso de implantar una nueva plataforma, la cual requerirá adaptaciones y creación de nuevo software, así como la necesidad de actualización de hardware en algunos casos.
- d) Mejoramiento general de los sistemas.- En este caso, el analista debe encontrar el modo de hacer mejor lo que ya se tiene.

2. ANÁLISIS Y DETERMINACIÓN DE REQUERIMIENTOS.

Una vez aceptado el proyecto, el analista de sistemas debe definir el alcance y desarrollar un enfoque profundo de los requerimientos y necesidades de los usuarios.

El análisis de sistemas es un proceso que se lleva a cabo teniendo en cuenta ciertos principios, a saber:

- ❖ Debe presentarse y entenderse el dominio de la información de un problema.
- ❖ Definir las funciones que debe realizar el software.
- ❖ Representar el comportamiento del software a consecuencia de acontecimientos externos.
- ❖ Dividir en forma jerárquica los modelos que representan la información, funciones y comportamiento.
- ❖ Debe partir desde la información esencial hasta el detalle de la implementación.

Solicitud de un análisis de sistemas

Independientemente de las razones para efectuar un análisis de sistemas, todos los proyectos deberán empezar con una forma de solicitud de servicios de sistemas de información. En el anexo 1b se presenta un ejemplo de cómo podría ser dicha forma.

Esta forma no solo sirve para proporcionar información acerca del proyecto de sistemas y sus objetivos, sino también de beneficios anticipados, así como una especie de contrato o acuerdo entre el usuario y el analista de sistemas.

Es importante que esta solicitud la llenen los usuarios y los analistas trabajando conjuntamente, pues es el usuario quien tiene la información completa de las necesidades de entrada y de salida y posiblemente una noción general de los controles necesarios. Los analistas de sistemas deben interactuar con el usuario para refinar, sintetizar y agregar sobre estas ideas. Esta forma determina un punto claro de inicio, eliminando así la posibilidad de varias salidas no requeridas e innecesarias.

Normalmente, los proyectos tienen un alias o un nombre corto llamativo, esto los personaliza y facilita su referencia. Si el usuario que solicita el sistema es quien asigna el nombre al proyecto, entonces éste se convierte en *su* sistema, lo cual contribuye a disminuir la posible resistencia al cambio.

Objetivos del Análisis

*Un análisis de sistemas se lleva a cabo teniendo en cuenta los siguientes objetivos en mente:

- Identificar las necesidades del cliente.
- Evaluar qué conceptos tiene el cliente del sistema para establecer su viabilidad.
- Realizar un análisis técnico y económico.

- Asignar funciones al hardware, software, personal, base de datos, y otros elementos del sistema.
- Establecer las restricciones de presupuestos y planificación temporal.
- Crear una definición del sistema que comprenda el fundamento de todo el trabajo.

El aspecto fundamental del análisis de sistemas es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. Los analistas al trabajar con los empleados y administradores, deben estudiar los procesos de una empresa para reunir detalles relacionados con sus procesos, sus opiniones sobre por qué ocurren las cosas, las soluciones que proponen y sus ideas para cambiar el proceso.

Se recomienda emplear cuestionarios para obtener esta información cuando no es posible entrevistar en forma personal a los miembros de grupos grandes dentro de la organización.

Asimismo, las investigaciones detalladas requieren el estudio de manuales y reportes, la observación en condiciones reales de las actividades del trabajo y, en algunas ocasiones, muestras de formas y documentos con el fin de comprender el proceso en su totalidad.

Conforme se reúnen los detalles, los analistas estudian los datos sobre requerimientos, con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que debe producir, junto con características operacionales tales como controles de procesamiento, tiempos de respuesta y métodos de entrada y salida.

3. DISEÑO DEL SISTEMA.

¿Qué es el diseño de sistemas?

Es el proceso de aplicar ciertas técnicas y principios con el propósito de definir un sistema, con suficiente detalle como para permitir su interpretación y su realización física.

La importancia del diseño de software se puede definir en una sola palabra: *Calidad*. Dentro del diseño es donde se fomenta la calidad del proyecto.

El diseño es la única manera de materializar con precisión los requerimientos del cliente. Es un proceso y un modelado a la vez porque es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del sistema a construir. A su vez, debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente. Debe ser, además, una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el software.

Niveles de diseño

El diseño de un sistema se representa a través de dos fases: el diseño lógico y el diseño físico.

Cuando los analistas formulan un diseño lógico, escriben las especificaciones detalladas del nuevo sistema; esto es, describen sus características: las entradas, salidas, archivos, bases de datos y procedimientos; todos de manera que cubran los requerimientos del proyecto.

Diseño de la entrada

La entrada es el enlace que une al sistema de información con el mundo y sus usuarios. Consiste en el desarrollo de especificaciones y procedimientos para la preparación de datos, la realización de los procesos necesarios para poner los datos en una forma utilizable para su procesamiento.

La entrada de los datos se logra al instruir a la computadora para que los lea, ya sea en documentos escritos, impresos ó por personas que los escriben directamente al sistema.

En este punto, se deben considerar los siguientes aspectos importantes:

- Controlar la cantidad de datos en la entrada, ya que de otra manera su captura podría ser un proceso lento que tome mucho más tiempo que el que necesitan las computadoras para realizar sus tareas. Además, en una transacción existen datos importantes y otros que no lo son, por lo que el analista debe saber cuáles utilizará y que en realidad deben formar la entrada, pues entre menos datos a capturar, menores serán las posibilidades de error.
- Es fundamental mantener la sencillez del proceso. La simplicidad funciona y es aceptada por cualquier usuario. Cuesta trabajo que ellos acepten sistemas complejos o confusos y que no exista ninguna garantía para el éxito al ser instalados.

Diseño de la salida

A menudo, para los usuarios la característica más importante de un sistema de información es la salida¹ que produce. Si ésta no es de calidad, se pueden convencer de que todo el sistema es tan innecesario que eviten su utilización y, por lo tanto, posiblemente ocasionen errores y que el sistema falle.

Al diseñar la salida, se deben emplear ciertos métodos para presentar la información producida por el sistema. Los analistas deben decidir cuándo imprimir, desplegar o presentar su salida en forma audible.

Participación de los usuarios.

Los gerentes y usuarios del sistema también poseen un papel importante en el diseño del sistema; no es solamente el proyecto del analista. Durante el diseño, a algunos usuarios se les pide que revisen los borradores de los informes, que examinen los formatos de entrada y que ayuden en la escritura de los procedimientos para decirles a otras personas cómo utilizar el sistema en forma apropiada.

-La participación del usuario proporciona al analista una retroalimentación importante conforme avanza en el diseño; además asegura que los usuarios tengan un conocimiento no técnico de lo que realizará o no el nuevo sistema.

4. DESARROLLO DE SOFTWARE.

El diseño físico, actividad que sigue al diseño lógico, produce programas de software, datos, archivos y un sistema en marcha. Las especificaciones del diseño indican a los programadores qué debe hacer el sistema, los programadores a su vez escriben los programas que aceptan entradas por parte de los usuarios, procesan los datos, producen los informes y almacenan la información en archivos o bases de datos.

¹ El término "salida" se aplica a cualquier información producida por un sistema, ya sea impresa, desplegada o verbal.

Los encargados de desarrollar software pueden instalar (o modificar y después instalar) software comprado a terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los programadores. Por regla general, los programadores (o analistas programadores) que trabajan en las grandes organizaciones pertenecen a un grupo permanente de profesionales. En las empresas pequeñas, donde no hay programadores, se pueden contratar servicios externos de programación.

Los programadores también son responsables de las pruebas unitarias y de la documentación de los programas y de proporcionar una explicación de cómo y por qué ciertos procedimientos se codifican en determinada forma. La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

Evaluación del diseño físico

Para evaluar la calidad del diseño físico, se deben establecer criterios técnicos como son:

- Presentar una organización jerárquica que haga un uso inteligente del control entre los componentes del software.
- El diseño debe ser modular, es decir, se debe hacer una partición lógica del software en elementos que realicen funciones y subfunciones específicas.
- El diseño debe contener abstracciones de datos y procedimientos.
- Debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y el entorno exterior.

Estos criterios no se consiguen por casualidad. El proceso de diseño y desarrollo del software exige buena calidad a través de la aplicación de principios fundamentales de diseño, metodología y una revisión exhaustiva.

Cuando se va a diseñar un sistema basado en computadoras se debe tener presente que el proceso incluye concebir y planear algo en la mente, así como hacer un modelo².

Los diseñadores son los responsables de dar a los programadores las especificaciones de software completas y claramente delineadas. Una vez comenzada la fase de programación, se encargarán de contestar preguntas, despejar dudas y manejar los problemas que enfrentan los programadores cuando utilizan las especificaciones de diseño.

5. PRUEBA DE SISTEMAS.

Durante la fase de pruebas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan como entradas conjuntos de datos de prueba para su procesamiento y después se examinan los resultados. En ocasiones se permite que varios usuarios utilicen el sistema para que los analistas observen si tratan de emplearlo en formas no previstas. Es preferible descubrir cualquier sorpresa antes de que la organización implante el sistema y dependa de él.

² Un modelo es una representación gráfica de un problema o de una situación del mundo real. El modelado nos sirve para tener puntos de vista desde diferentes perspectivas.

En muchas organizaciones, las pruebas integrales³ son conducidas por personas ajenas al grupo que escribió los programas originales; con esto se persigue asegurar, por una parte, que las pruebas sean completas e imparciales y, por otra, que el software sea más confiable. Para ello es importante realizar una matriz de pruebas y apegarse a ella para trabajar en forma ordenada y no dejar de lado alguna parte del sistema cuya omisión repercute en posteriores errores productivos.

El objetivo de este paso es empezar a operar el nuevo sistema con transacciones reales, a fin de efectuar los ajustes que sean necesarios antes de dar por entregado el sistema al usuario.

6. IMPLANTACIÓN.

La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla. Su principal finalidad es poner en marcha el sistema desarrollado y someterlo a un proceso de evaluación. Durante esta etapa, el equipo de proyecto debe acompañar al usuario y finalmente hacerle la entrega para que comience su operación.

Dependiendo del tamaño de la organización que empleará la aplicación y el riesgo asociado con su uso, puede elegirse comenzar la operación del sistema sólo en un área de la empresa (prueba piloto). Algunas veces se deja que los dos sistemas, el viejo y el nuevo, si es que el primero existe, trabajen en forma paralela con la finalidad de comparar los resultados. En otras circunstancias, el viejo sistema deja de utilizarse determinado día para comenzar a emplear el nuevo al día siguiente. Cada estrategia de implantación tiene sus méritos de acuerdo con la situación que se considere dentro de la empresa. Sin importar cuál sea la estrategia utilizada, los encargados de desarrollar el sistema procuran que el uso inicial del sistema se encuentre libre de problemas.

Las áreas usuarias deben analizar las salidas del nuevo sistema y formular todas aquellas observaciones que pudieran surgir.

Cuando se trata de una conversión, el traspaso de sistema actual a uno nuevo puede realizarse según alguno de estos procesos:

Proceso encadenado

- ◆ En este caso, el sistema actual sigue funcionando y produciendo la información establecida, el nuevo sistema toma las transacciones ya procesadas por el sistema actual y produce sus propios resultados en un período de tiempo más tarde.
- ◆ La información producida por el sistema actual se le distribuye al usuario y la producida por el nuevo sistema queda en poder del equipo de proyecto para su evaluación.
- ◆ Este método se usa cuando el nuevo sistema es muy complejo o vulnerable y se le exige un alto grado de confiabilidad.
- ◆ Permite la comparación con los resultados del sistema actual y facilita el análisis de los resultados del nuevo sistema fuera de línea.

Proceso directo

³ Se denominan pruebas integrales porque lo que se prueba es el sistema en conjunto, para verificar la funcionalidad e interacción entre cada módulo.

- ◆ En este caso se inactiva al sistema actual y las transacciones se procesan en el nuevo sistema.
- ◆ Este método se usa cuando el sistema de información no es complejo y si las dificultades iniciales esperadas se estiman inferiores a las consecuencias de no comenzar en tiempo el nuevo sistema.
- ◆ Además, cuando resulte difícil la simultaneidad de las tareas de los sistemas vigente y nuevo.

Proceso en paralelo

- ◆ Las transacciones se procesan en el sistema actual y en el nuevo sistema simultáneamente.
- ◆ Por razones de costo, se puede inactivar el sistema actual tan pronto como los resultados del nuevo sistema de información sean confiables.
- ◆ Este método se usa cuando la Gerencia (máximo nivel) exige una alta confiabilidad del nuevo sistema o cuando los sistemas son muy diferentes (no habrá duplicaciones en procesos y salidas).
- ◆ Tiene la ventaja de minimizar el riesgo de comenzar con un sistema defectuoso.

Proceso por subsistemas

- ◆ El sistema actual continúa procesando una parte de las transacciones y el nuevo sistema el resto.
- ◆ La secuencia en la que los subsistemas comenzarán su operación debe ser cuidadosamente planificada.
- ◆ Reduce costos y riesgos.

Informe final

El objetivo de este paso es informar a la Gerencia sobre el estado del nuevo sistema y su condición de entrega final. Es un informe por escrito en el que se incluye lo actuado en esta etapa, las observaciones de los usuarios, errores, correcciones y aprobación. Se debe exponer a la Gerencia en presencia de los responsables de las áreas involucradas.

Se requiere aprobación del informe final y de lo actuado para seguir adelante con el mantenimiento o finalizar el ciclo del sistema. Para ello se debe tener toda la documentación del sistema de información. La entrega será tan prolongada y rigurosa como haya sido la prueba del sistema nuevo, la evaluación, el ajuste necesario y la participación del usuario.

El equipo de proyecto debe asegurarse que:

- El nuevo sistema cumple con los resultados esperados
- Las áreas tienen toda la documentación a usar
- Se ha concluido con la capacitación

Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo, las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses. Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones: realizar cambios y modificaciones en el software, archivos o procedimientos para satisfacer las nuevas necesidades de los usuarios. Dado que los sistemas de las organizaciones junto con el ambiente de las empresas experimentan cambios de manera continua, los sistemas de información deben mantenerse siempre al día. En este sentido, la implantación es un proceso en constante evolución.

7. EVALUACIÓN DEL SISTEMA

La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

- Evaluación operacional. Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.
- Impacto organizacional. Identificación y medición de los beneficios para la organización, en áreas tales como finanzas (costos, ingresos y ganancias), eficiencia operacional e impacto competitivo. También se incluye el impacto sobre el flujo de información interno y externo.
- Opinión de los administradores. Evaluación de las actitudes de directivos y administradores dentro de la organización, así como de los usuarios finales.
- Desempeño del desarrollo. La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye la valoración de los métodos y herramientas utilizados en el desarrollo.

Un buen mantenimiento del sistema asegura su operación permanente. Las mejoras son necesarias para evitar la degradación del sistema en el servicio a los usuarios.

COMENTARIOS

En el contexto de este trabajo se considera a un sistema como una combinación de hardware y software que da una solución a un problema de negocios. El desarrollo de sistemas es la creación de programas para un cliente, quien tiene el problema que debe ser resuelto. Un analista es el que estudia y documenta el problema del cliente y lo comunica a los desarrolladores, quienes generarán los programas para resolver el problema y los distribuirán en equipos de computación.

En una empresa, la interacción entre los sistemas y sus módulos, así como sus actividades y funciones es algo muy real. Debido a ello, una habilidad importante que deben desarrollar los analistas es la capacidad de adquirir un panorama global de todas las actividades y operaciones de una organización.

Distinguir y comprender las relaciones entre las diversas funciones de los diferentes departamentos y considerar, a lo largo del proceso de desarrollo, el impacto que éstas tienen en toda la organización, conducirá a los analistas a crear los sistemas de información más útiles; porque se adaptarán a los ya existentes.

Es importante poner atención a la interdependencia de unidades dentro de una organización, ya sea por el enfoque de sistemas o por un panorama global, si se desea que el sistema a desarrollar tenga la mayor utilidad para la empresa que lo solicita.

En el diseño de un sistema se deben definir las entradas, procesos y salidas, así como aquellos datos que serán calculados y los que deben ser almacenados. Asimismo, en esta etapa deben describirse con todo detalle los procedimientos de cálculo y los datos individuales. Los analistas diseñadores seleccionarán las

estructuras de datos y los dispositivos de almacenamiento. Los procedimientos que se escriben indican cómo procesar la información y cómo producir las salidas.

Los documentos que contienen las especificaciones de diseño representan a éste de muchas maneras (diagramas, tablas y símbolos especiales). La información detallada del diseño se proporciona al equipo de programación para comenzar la fase de desarrollo de software, por ello es primordial que el diseño se lleve a cabo siempre con calidad, pues éste será la base fundamental del nuevo sistema.

Desafortunadamente, la evaluación de sistemas no siempre recibe la atención que merece. Sin embargo, cuando se conduce en forma adecuada proporcionará mucha información que puede ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones subsecuentes.

Se deben evaluar los resultados de la operación del nuevo sistema, antes de la entrega final. Se analizarán los errores detectados en la puesta en marcha. Dichos errores pueden ser de diseño y/o desarrollo, así como también, salidas previstas que a pesar de no tener errores, no satisfacen a los usuarios.

Importante:

1. Los errores o insatisfacciones deben ser corregidos sin excepción.
2. Si los errores son críticos convendrá suspender la operación del nuevo sistema, reactivar el anterior y reajustar el nuevo según corresponda.
3. Si los errores no son críticos conviene seguir la operación del nuevo sistema y solucionar el problema en el momento que sea más conveniente.

CAPÍTULO III

LA IMPORTANCIA DE SEGUIR UNA METODOLOGÍA DE TRABAJO

*"Todo esfuerzo bien disciplinado
tiene una recompensa múltiple"...*

- Jim Rohn

INTRODUCCIÓN

Como ya se ha percibido, el desarrollo de sistemas es algo muy común dentro de las empresas y cada día muchas de éstas dependen de los medios automatizados para dar servicio a los clientes o usuarios, y también para contar con una forma más eficiente de organizar la información y/o procesarla.

Sin embargo, en un principio el desarrollo de software se hacía sin usar metodologías o técnicas, esto debido a que los sistemas eran pequeños y de muy escasa complejidad, pero a medida que avanzó la tecnología, las necesidades crecieron provocando que los sistemas de software no fueran de suficiente calidad.

Actualmente, para soportar la puesta en operación de nuevos productos y ofrecer un mejor servicio a los clientes, se requiere que toda organización cuente con una metodología de trabajo que permita una adecuada interacción de las áreas que intervienen en los procesos para el desarrollo y mantenimiento de aplicaciones. No obstante, el seleccionar una metodología de trabajo implica analizar varias de ellas para determinar cuál es la más adecuada al problema u organización en donde se desea aplicar.

En este capítulo se describe la metodología interna que indica los parámetros que definen los estándares en que un proyecto debe ser desarrollado y documentado dentro de BanCrecer, haciendo hincapié en que el objetivo no es presentar las metodologías existentes, sino únicamente mencionar cuál es la que se emplea en el caso práctico aquí expuesto y resaltar algunos aspectos importantes que deben tenerse presentes para utilizarla de la mejor manera.

EL EMPLEO DE UNA METODOLOGÍA DE TRABAJO.

¿Por qué es necesario el empleo de una metodología de trabajo?

Para el desarrollo e implantación de los sistemas es necesario realizar una serie de procesos que parten desde la comprensión del problema hasta la puesta en marcha del producto final.

La manera para desarrollar software desde que se inició la computación, a través de los años ha tenido una evolución lenta. En un principio se trató de aplicar el análisis y diseño de otras especialidades, dando como resultado que el software fuera de mala calidad; se observó mediante la experiencia que se requería de métodos y/o técnicas que ayudaran a los desarrolladores de software a crear éste de mejor calidad.

La necesidad de diseños sólidos ha traído consigo la necesidad de una metodología que los analistas, desarrolladores y clientes acepten como pauta. El empleo de una metodología permite modelar, construir y documentar los elementos que forman un sistema de software de una manera óptima.

Sin embargo, en la etapa de selección de una metodología, el equipo de desarrollo debe de elegir la que más se adapte a la problemática del sistema, indudablemente que no todas las existentes son las adecuadas a cada problema, es aquí donde la ingeniería de software ayuda, sugiriendo diferentes metodologías, e inclusive la combinación de éstas.

Debido a que cada organización cuenta con procedimientos y maneras particulares de poner en práctica el ciclo de vida del desarrollo de sistemas, se recomienda llevar a cabo un proceso de integración de la metodología en la cultura organizacional.

LA IMPORTANCIA DE SEGUIR UNA METODOLOGÍA DE TRABAJO

Para lograr satisfactoriamente este proceso, existen diversos factores críticos de éxito, entre los que podemos citar:

- Contar con el apoyo y compromiso de la alta dirección.
- Promover la participación activa del personal de las áreas involucradas.
- Reconocer que la metodología es una herramienta para lograr los objetivos finales.

Consciente de lo anterior, en el Grupo Financiero BanCrecer se ha impulsado la implantación de una metodología formal de trabajo para las áreas involucradas en los procesos de desarrollo de software, con lo cual el área de Sistemas ha llegado a ser unas de las más competitivas, buscando siempre mejorar la atención a clientes y usuarios a través del uso de un sistema de información amigable y seguro.

Cada uno de los elementos que se incorporan en esta metodología ayudan a identificar y comprender los procesos que se aplican, los elementos que están involucrados, establecer una relación entre agrupamientos de la organización y promover un mantenimiento que asegure el funcionamiento adecuado del sistema.

Definición de la Metodología de trabajo

Dentro de la Institución, se define a la Metodología como un proceso estructurado y repetible para construir aplicaciones que resulten en un valor agregado para la empresa y que apoyen los objetivos estratégicos del negocio.

Esta metodología se compone de rutas de trabajo¹ para la planeación, desarrollo y soporte de los distintos proyectos de la Institución. Las rutas posibles son:

- Planeación de la información.
- Desarrollo.
- Soporte de sistemas de producción.

Estas rutas se dividen, a su vez, en tres niveles:

1) Fases.² Las fases son porciones de la ruta y representan etapas lógicas de actividades, las cuales cubren básicamente el ciclo de vida de un sistema (planeación, análisis, diseño, programación, pruebas, liberación y mantenimiento).

2) Segmentos. Un segmento es la agrupación de varias tareas con actividades similares. El objetivo es usarlo como elemento de avance en los reportes, es decir, el estatus de un proyecto deberá informarse a nivel de proyecto, fase y segmento (Ejemplo, en la fase de desarrollo: codificar programas, realizar plan de pruebas.)

3) Tareas. Las tareas son las actividades a realizar durante cada fase de la ruta. (Ejemplo, en el segmento de pruebas: preparar datos para pruebas, hacer pruebas unitarias, etc).

Los productos terminados generados en cada fase, llamados en la Institución 'entregables', representan la documentación aprobada para la nueva aplicación. En ocasiones es necesario requerir de documentos que ayuden a registrar algún evento, pero que no es obligatorio guardarlos como documentación oficial, éstos son los denominados documentos de trabajo y su uso es opcional.

¹ Una ruta de trabajo es la secuencia de pasos a seguir en cada proyecto.

² Cada una de las actividades mayores en las que se divide un proyecto, con fechas de inicio y término predefinidas.

A continuación, se resumen brevemente en un cuadro sinóptico las distintas fases y segmentos de la metodología empleada en BanCreceer.

Fases y segmentos de la metodología

FASE: PLANEACIÓN Y DEFINICIÓN DEL PROYECTO.

SEGMENTO 1: Iniciar proyecto y definir requerimientos de usuario.

SEGMENTO 2: Elaborar diseño conceptual.

SEGMENTO 3: Obtener aprobación del usuario y de la Dirección.

FASE: ANÁLISIS DE INFORMACIÓN.

SEGMENTO 1: Verificar requerimientos del usuario.

SEGMENTO 2: Definir cómo se medirá el desempeño del sistema.

SEGMENTO 3: Modelado de eventos, procesos y datos.

FASE: DISEÑO LÓGICO DEL SISTEMA.

SEGMENTO 1: Diseñar diálogos, pantallas, formatos y reportes.

SEGMENTO 2: Diseñar la arquitectura lógica del sistema.

FASE: DISEÑO FÍSICO DEL SISTEMA.

SEGMENTO 1: Diseñar la arquitectura física del sistema.

SEGMENTO 2: Diseñar el enfoque de la prueba integral.

SEGMENTO 3: Diseñar las pruebas unitarias.

SEGMENTO 4: Verificar atributos de comportamiento del diseño.

FASE: CONSTRUCCIÓN DE PROGRAMAS.

SEGMENTO 1: Codificar programas.

SEGMENTO 2: Preparar datos para pruebas unitarias.

SEGMENTO 3: Realizar pruebas unitarias.

FASE: ANÁLISIS DEL COMPORTAMIENTO DEL SISTEMA.

SEGMENTO 1: Completar atributos de comportamiento.

FASE: PRUEBA INTEGRAL, LIBERACIÓN Y CAPACITACIÓN.

SEGMENTO 1: Desarrollar procedimientos del sistema.

SEGMENTO 2: Desarrollar material para capacitación.

SEGMENTO 3: Capacitar al personal usuario.

SEGMENTO 4: Crear modelo de la prueba integral.

SEGMENTO 5: Realizar prueba integral.

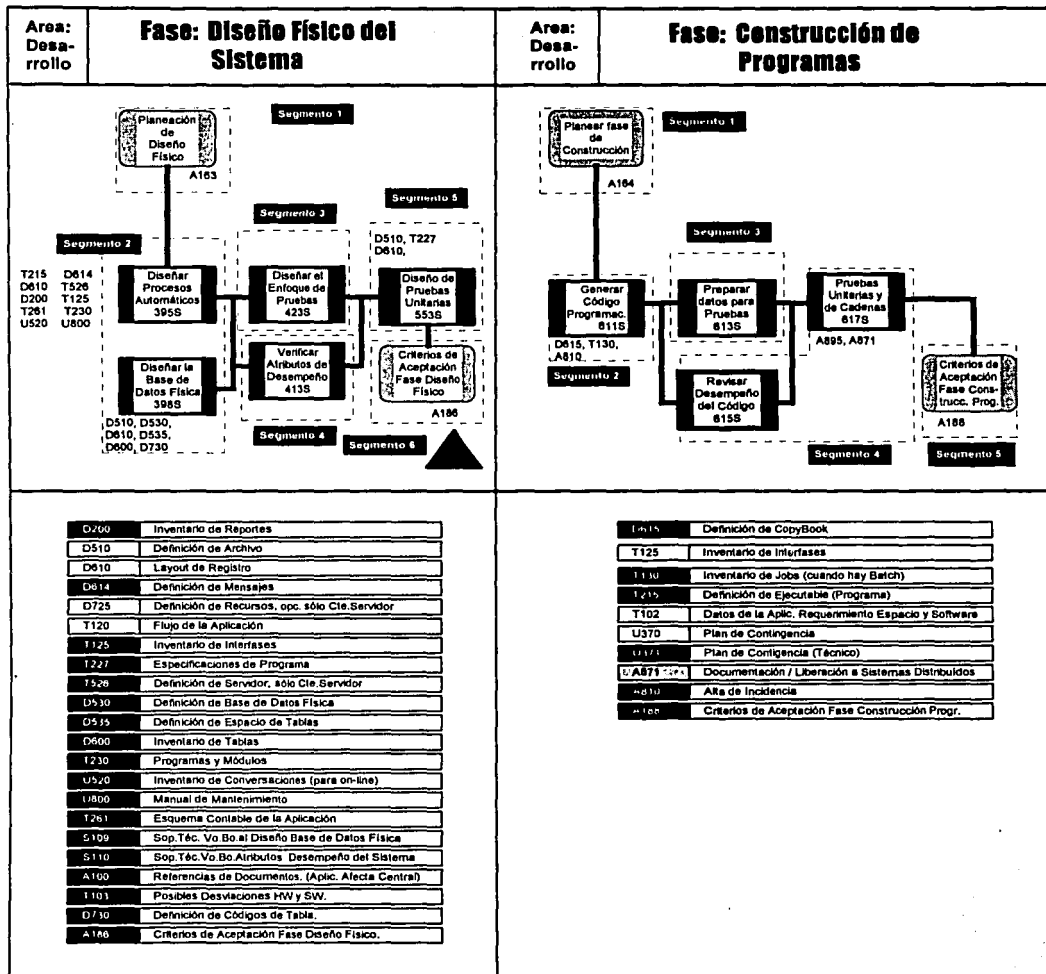
SEGMENTO 6: Elaborar documento de liberación a Producción.

FASE: PUESTA DEL SISTEMA EN PRODUCCIÓN.

SEGMENTO 1: Migración de ambientes.

SEGMENTO 2: Monitorear el sistema en Producción.

LA IMPORTANCIA DE SEGUIR UNA METODOLOGÍA DE TRABAJO



"Diseño Físico del Sistema y Construcción de Programas".

Objetivos de la Metodología

La definición y empleo de esta metodología creada especialmente en BanCreceer tiene principalmente los siguientes objetivos:

- Optimizar los procesos internos de las áreas de sistemas, con el fin de hacer más eficiente el servicio a usuarios para soportar las actividades del negocio, buscando apoyar la posición competitiva de la Institución dentro del mercado.
- Desarrollar aplicaciones usando arquitecturas, técnicas y plataformas en forma organizada.
- Apoyar el desarrollo de equipos de trabajo de tamaño medio que trabajen de manera efectiva con grupos de usuarios múltiples.
- Proveer un enfoque de desarrollo flexible.

Beneficios principales

Los beneficios que representa para el Banco la aplicación de una metodología, nuestra metodología, se resumen en lo siguiente:

- Utilizar una metodología pensada totalmente en función de las necesidades del área (Sistemas).
- Contar con una guía para cada rol, detallando actividades y responsabilidades definidas.
- Contar con una guía basada en los últimos avances dentro del análisis y diseño de sistemas, los cuales van a la par de la evolución de las herramientas comerciales de desarrollo.
- Contar en todo momento con la documentación real y a tiempo de la situación del proyecto, en todas sus áreas.
- Cumplir con las expectativas de los involucrados, principalmente de los usuarios, en costo, tiempo y funcionalidad.
- Cumplir con índices de calidad estandarizados.
- Contar con planes de contingencia en función de los riesgos detectados en cualquier momento.

Alcances

- Todas las áreas que intervienen en el proceso de desarrollo y mantenimiento de sistemas de BanCreceer.
- Todos los sistemas desarrollados, para que cuenten con técnicas, estándares y puntos de control, que permitan su documentación clara, definida y organizada.

Normas generales para el desarrollo de proyectos en BanCreceer

A continuación se mencionan las principales normas que rigen el empleo de la metodología estándar en la Institución, a través de las cuales se hace obligatorio su uso y aplicación, para dar una idea de la importancia que tiene para el desarrollo de software en el Banco.

- Invariablemente, la 'Metodología para el Desarrollo de Nuevas Aplicaciones', deberá aplicarse en los sistemas de la Institución, así como la 'Metodología para el Mantenimiento de Aplicaciones', deberá aplicarse a los sistemas en operación.
- Las actualizaciones o modificaciones a esta metodología, se efectuarán a través del área de Procedimientos, en coordinación con el área de Seguridad y Control de la Información.
- El personal que por motivo de las funciones propias de su puesto, tenga acceso a información derivada de los sistemas que se estén operando, deberá apegarse a las Políticas Generales de Seguridad y Control de la Información.
- Previamente al desarrollo de cualquier proyecto de sistemas, debe someterse a la autorización del Comité de la Dirección.
- El área de Diseño será la responsable de emitir el Aviso de Inicio del Proyecto.
- Cada desarrollo podrá cubrir o no las diferentes fases o tareas de la metodología, dependiendo del proyecto que se trate, guardando el debido cuidado de no omitir fases o tareas que sean indispensables, quedando esto bajo la responsabilidad del líder del proyecto.
- Bajo ningún concepto, la documentación de los procesos definidos en la Metodología deberá estar fuera del software estándar de la Institución para la documentación de proyectos.

Responsabilidades de las áreas involucradas

En este apartado se mencionan brevemente las actividades que son responsabilidad de cada área involucrada con el empleo de la metodología, pues aunque existen algunas otras áreas o departamentos que también están relacionadas con el área de Sistemas, los que se mencionan a continuación son los distintos grupos de trabajo que tienen que ver directamente con el desarrollo de aplicaciones y, por lo tanto, con la metodología.

Esta descripción de áreas y actividades servirá también para dar una mejor idea de cómo están organizados los roles y departamentos y cómo trabaja el personal de sistemas en la realidad, pudiendo hacer una comparación entre lo que dice la teoría en el capítulo I y lo que se menciona en este capítulo a continuación.

METODOLOGÍA.

- Es responsable de proporcionar el apoyo necesario para la definición e implantación de estándares necesarios que hacen más eficiente el desarrollo y mantenimiento de aplicaciones, así como la administración, control y apoyo a usuarios en herramientas tales como Design/1 y Method/1³.

DISEÑO.

- Es el primer contacto del usuario con el área de sistemas y le da soporte en la detección y definición de necesidades. Coordina a las áreas involucradas en la solución, así como las actividades necesarias para su puesta en marcha.
- Definirá en forma cuantificable y medible los requerimientos del usuario, detallando los productos entregables que espera recibir.

³ Estos productos son el estándar utilizado en Bancrecer, y en general, en toda Institución bancaria que maneje con Altamira sus sistemas de información, ambas son para el control y seguimiento de la metodología.

LA IMPORTANCIA DE SEGUIR UNA METODOLOGÍA DE TRABAJO

- Desarrollará un modelo conceptual de la solución que se propone para satisfacer los requerimientos del usuario.
- Definirá todos los productos entregables previstos en el diseño conceptual; esto es, diseño de reportes, pantallas, formatos, archivos, afectaciones o creaciones de bases de datos.
- Obtendrá la aprobación del usuario sobre el diseño definitivo.
- Supervisará que en el desarrollo físico de la aplicación no existan modificaciones al diseño definido.

DESARROLLO

- Desarrollar y mantener los programas para dar soporte al negocio en las distintas plataformas tecnológicas (Desarrollo Central: Cobol, Cics, DB2; Desarrollo Departamental: Visual Basic, Informix, Java, Internet).
- Diseñará y creará físicamente las bases de datos y los archivos que se han previsto.
- Diseñará los procesos que conformarán el nuevo sistema.
- Elaborará los programas correspondientes.
- Elaborará los planes de pruebas unitarias e integrales que serán ejecutadas por el área de Implantación.
- Actualizará la documentación de las aplicaciones, establecida en la Metodología.

SOPORTE TÉCNICO.

- Se encarga de la instalación, soporte y mantenimiento de software y hardware en las distintas plataformas del grupo central y departamental.
- Verificará, junto con las áreas de Desarrollo y Diseño, los atributos de desempeño del nuevo sistema; esto es, el comportamiento en cuanto a velocidad de procesos y manejo de las bases de datos, así como a espacios ocupados.
- Sugerirá mejoras para eficientar el desempeño del nuevo sistema, antes de ponerlo en uso.

IMPLANTACIÓN.

- Probar todos los desarrollos en coordinación con el usuario para garantizar que cumplen con los requerimientos de negocios, así como que se apegan a la normatividad interna y externa.
- Ejecutará todas las pruebas del 'Plan de Pruebas' elaborado por el área de Desarrollo.
- Solicitará las correcciones necesarias cuando se detecten errores al ejecutar las pruebas.
- Documentará las pruebas llevadas a cabo.
- Solicitará el pase al ambiente de Producción si todas las pruebas fueron satisfactorias; no se hará el pase de ningún desarrollo a Producción si no ha sido probado a satisfacción del área de Implantación.
- Elaborará el documento de solicitud de migración de ambientes.

TESIS CON
FALLA DE ORIGEN

PROCEDIMIENTOS.

- Construir, estandarizar y distribuir circulares, normas y procedimientos de la Institución.
- Elaborará los procedimientos operativos, circulares e información respectiva para la red de sucursales y departamentos, con base a la documentación generada por el área de Diseño.
- Llevará el seguimiento y control de la información emitida a usuarios y aquélla pendiente de difundir.

PRODUCCIÓN.

- Realizar la planificación y operación de los sistemas de cómputo, distribución de la información, diseño y monitoreo de servicios de comunicaciones y soporte de la infraestructura de respaldo de los equipos centrales.
- Llevará a cabo físicamente la migración de ambientes.
- Monitoreará el desempeño del nuevo sistema para detectar posibles conflictos.
- Sugirió modificaciones si se detectaran problemas de operación.

INCIDENCIAS

- Analizar, evaluar y documentar todos los problemas escalados por el área de soporte a usuarios (SAS) y coordinar la asignación de las mismas a las áreas de segundo nivel.

COMENTARIOS

El objetivo principal de este capítulo es, como su nombre lo dice, resaltar la importancia del empleo de una metodología de trabajo. ¿Por qué se considera este tema importante?

Veamos el siguiente ejemplo:

"Existió alguna vez un grupo de trabajo en el que la misión estaba *bien definida*, los *recursos humanos* eran más que suficientes, los *recursos materiales* eran abundantes, no había restricciones de *tiempo* y se utilizaba la *tecnología* adecuada. Este grupo trabajaba en la construcción de la Torre de Babel, sin embargo, ¿por qué falló la obra si contaban con los elementos necesarios?"

→ Falló la *comunicación*, la *organización* y la *coordinación*... No tenían una *metodología* de trabajo.

Lo mismo sucede diariamente en la práctica. Tal vez en la mayoría de los proyectos en los que hemos formado parte no siempre contamos con todos los factores descritos en el ejemplo, y no necesariamente por ello terminaremos con un fracaso, o por el contrario, idealmente podría ocurrir que, como en el ejemplo, contásemos con todo lo necesario, pero sin una metodología de trabajo, difícilmente podremos asegurar el éxito. ¿Por qué? Porque en todos los grupos de trabajo, ya sean chicos o grandes, la comunicación, la organización y la coordinación son factores importantísimos que nunca deben perderse de vista. El grado en

que éstos sean empleados determinará el éxito o fracaso que logremos al involucrarnos en algún proyecto, y solamente con una metodología bien definida podremos llevarlos a cabo.

Ahora bien, el tener una metodología bien definida y documentada en la Institución tampoco sirve de mucho si ésta no se lleva a la práctica. Si la gente involucrada no se apega a ella, su utilidad no será mucha. Por ello se menciona que todos debemos estar conscientes del tema, desde los altos niveles hasta el último recurso participante en el desarrollo del sistema.

Se hace mención de esto porque en mi particular experiencia, el tener una metodología perfectamente definida en la Institución no nos ha garantizado el éxito en la liberación de los proyectos apegados a ella. Y esto se ha debido principalmente a que no todos somos conscientes de ella.

Algo que he observado ha sido que debido a la rotación del personal, el conocimiento que se tiene de esta metodología poco a poco se va perdiendo y al final solo quedan unos cuantos que la conocen o han oído hablar de ella. El personal que posteriormente se integra al área ya no conoce los detalles y solo emplea lo que le es absolutamente necesario para el desempeño de sus labores cotidianas. Aunado a ello se encuentra el poco apoyo e interés de algunos de los directivos. Eso ocasiona que paulatinamente, y con las cargas de trabajo, no se cumplan los estándares establecidos y finalmente, todo sucede en detrimento del desarrollo de los proyectos.

Debido a ello, se recomienda que continuamente se desarrolle un plan de capacitación para que el personal aprenda o retome los conceptos de la metodología, ya que ello asegurará que los proyectos se lleven a buen término. Incluso cuando llegue alguien de recién ingreso, debe recibir una plática o por lo menos alguna documentación breve pero precisa de este tema. No hay mejor manera de involucrarlo en los proyectos que haciéndolo partícipe de la metodología, sin permitir que las cargas de trabajo nos hagan olvidarnos de esos aspectos que son parte de nuestra formación profesional dentro de la empresa, ya que justamente esas cargas irregulares de trabajo pueden deberse al poco o nulo uso práctico de la metodología.

A nadie le gustaría que el equipo de desarrollo comenzara a codificar sin más. La estructura y naturaleza de los pasos en un esfuerzo de desarrollo es lo que yo entiendo como metodología. En ese proceso debe quedar claro que antes de comenzar a programar, los desarrolladores tienen que comprender con claridad el problema. Esto requiere que alguien analice los requerimientos, y alguien tiene que convertir tal análisis en diseño.

Sin embargo, establecer una metodología no es solo eso, pues si un analista no tiene contacto con un diseñador, y éste a su vez no tiene contacto con un programador, existirá la posibilidad de que rara vez trabajen juntos para compartir puntos de vista importantes. Es por ello que resalta como un factor clave para el éxito de la metodología el trabajo en equipo. ¿Por qué? El equipo tiene que formarse de analistas para comunicarse con el cliente y comprender el problema, diseñadores para generar una solución, programadores para codificarla e ingenieros de sistemas para distribuirla.

Un proceso de desarrollo tiene que tomar en cuenta todos los procesos anteriores, utilizarlos adecuadamente y asignar la cantidad de tiempo necesaria para cada fase. El proceso también debe dar por resultado diversos productos del trabajo que den indicios de progreso y conformar una estela de responsabilidad.

Finalmente, el proceso deberá asegurar que sus fases no sean discontinuas. En lugar de ello, debe obtenerse información entre las fases para fomentar la creatividad y aumentar la facilidad de innovar, pues en esta época en que cada vez la adquisición de unas empresas por otras para fortalecerse es mayor, la competitividad es un factor determinante, y para estar siempre a la vanguardia, la tendencia debe ser innovar.

La idea de que existan metodologías de trabajo estándar y 'unitalla' es totalmente falsa: una empresa tiene su propia cultura, normatividad, historia y personal.

Ese ha sido el caso de BanCrecer, en donde se fomentó siempre el desarrollo y uso de una metodología a la medida, en la que todas sus áreas de sistemas aportasen siempre su trabajo con calidad.

Sin embargo, lamentablemente el resultado no siempre ha sido el esperado, pues como en el caso de la Torre de Babel, en la mayoría de los proyectos aquí desarrollados ha fallado la comunicación, pero sobre todo, el trabajo en equipo.

CAPÍTULO III

LA IMPORTANCIA DE UN BUEN ANÁLISIS DE REQUERIMIENTOS

*"En la vida de todo problema hay un momento en que
éste es lo bastante grande para verse,
pero aún lo suficientemente pequeño para resolverse"...*

- Mike Leavitt

INTRODUCCIÓN

El análisis de los requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema. De esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

Estudios realizados¹ muestran que más del 53% de los proyectos de software fracasan por no realizar un estudio previo de requisitos. Otros factores, como falta de participación del usuario, especificaciones incompletas y el cambio a los requerimientos, también ocupan sitios altos en los motivos de fracasos.

Cuando se quiere dar a entender mejor lo que vamos a construir, como en el caso de edificios o máquinas, se crea un modelo idéntico, pero en menor escala. Sin embargo cuando aquello que construiremos es un software, nuestro modelo debe tomar una forma diferente. Se deben representar todas las funciones y subfunciones de un sistema. Los modelos se concentran en lo que debe hacer el sistema no en cómo lo hace. Estos modelos pueden incluir notación gráfica, información y comportamiento del sistema.

Si deseamos crear sistemas que nos involucren con este nuevo milenio, ¿cómo debemos manejar tanta complejidad?

La clave está en organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema, lo comprendan y convengan con él.

En este capítulo se analiza la importancia del análisis de requerimientos, el cual debe ser completo, práctico y efectivo, ya que la clara comprensión del problema repercute en un diseño sólido que nos llevará a plasmar fielmente en nuestros desarrollos los requerimientos de los usuarios.

ANÁLISIS DE REQUERIMIENTOS

¿Qué son los requerimientos?

De las muchas definiciones que existen para 'requerimiento', a continuación se presenta la definición que aparece en el glosario de la IEEE².

1. Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
2. Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

- a) Los requerimientos funcionales definen, como la palabra lo indica, las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

¹ Fuente: análisis realizado por la consultoría IDS y presentada a BanCrecer en el 2000.

² Por sus siglas en inglés "Institute of Electrical and Electronics Engineers".

- b) Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, entre otros.

Características de los requerimientos

Las características de un requerimiento son sus propiedades principales. A continuación se presentan las más importantes:

- Necesario: Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.
- Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- Completo: Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- Consistente: Un requerimiento es consistente si no es contradictorio con otro requerimiento.
- No ambiguo: Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición no debe causar confusiones al lector.
- Verificable: Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Dificultades para definir los requerimientos

La mayoría de las veces podemos encontrarnos con ciertos aspectos que dificultan la elaboración de requerimientos bien definidos. Tales factores pueden ser uno o más de los que se listan a continuación:

- ◆ Los requerimientos no son obvios y vienen de muchas fuentes.
- ◆ Son difíciles de expresar en palabras (el lenguaje es ambiguo).
- ◆ Existen muchos tipos de requerimientos y diferentes niveles de detalle.
- ◆ La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
- ◆ Nunca son iguales. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.
- ◆ Los requerimientos están relacionados unos con otros, y a su vez se relacionan con otras partes del proceso.
- ◆ Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas.
- ◆ Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
- ◆ Son difíciles de cuantificar, ya que cada conjunto de requerimientos es particular para cada proyecto.

Ingeniería de requerimientos

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema es llamado Ingeniería de Requerimientos³. La meta de la ingeniería de requerimientos es entregar una especificación de requisitos de software correcta y completa.

³ El término ingeniería es usado en la descripción de metodologías modernas para explicar que se usan disciplinas formadas con técnicas precisas y bien pensadas, antes que inventadas sobre la marcha.

A continuación se darán algunas definiciones para "ingeniería de requerimientos":

- "Es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema." Boehm, 1979.
- "Es el proceso por el cual se transforman los requerimientos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones." STARTS Guide, 1987.
- "Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos." Leite, 1987.
- "Es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto." Rational Software, 1990.

Importancia de la ingeniería de requerimientos

Los principales beneficios que se obtienen de la ingeniería de requerimientos son:

- Permite gestionar las necesidades del proyecto en forma estructurada. Cada actividad consiste de una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados, proporcionando un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto. Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro.
- Mejora la calidad del software. La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (por ejemplo: funcionalidad, facilidad de uso, confiabilidad, desempeño).
- Mejora la comunicación entre equipos. La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- Evita rechazos de usuarios finales, pues obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.

Personal involucrado

Realmente son muchas las personas involucradas en el desarrollo de los requerimientos de un sistema. Es importante saber que cada una de esas personas tienen diversos intereses y juegan roles específicos dentro de la planificación del proyecto; el conocimiento de cada papel desempeñado, asegura que se involucre a las personas correctas en las diferentes fases del ciclo de vida. No conocer estos intereses puede ocasionar una comunicación poco efectiva entre clientes y desarrolladores, que a la vez traería impactos negativos tanto en tiempo como en presupuesto.

Los roles más importantes pueden clasificarse de manera general como sigue:

- **Usuario final:** Son las personas que usarán el sistema desarrollado. Ellos están relacionados con la operatividad, la disponibilidad y la fiabilidad del sistema; están familiarizados con los procesos

LA IMPORTANCIA DE UN BUEN ANÁLISIS DE REQUERIMIENTOS

específicos que debe realizar el software, dentro de los parámetros de su ambiente laboral. Serán quienes utilicen las interfaces y los manuales de usuario.

- **Usuario Líder:** Son los individuos que comprenden el ambiente del sistema o el dominio del problema en donde será empleado el software desarrollado. Ellos proporcionan al equipo técnico los detalles y requerimientos de las interfaces del sistema.
- **Personal de mantenimiento:** Para proyectos que requieran un mantenimiento eventual, éstas personas son las responsables de la administración de cambios, de la implementación y resolución de anomalías. Su trabajo consiste en revisar y mejorar los procesos del producto ya finalizado.
- **Analistas y programadores:** Son los responsables del desarrollo del producto en sí; ellos interactúan directamente con el cliente.
- **Personal de pruebas:** Se encargan de elaborar y ejecutar el plan de pruebas para asegurar que las condiciones presentadas por el sistema son las adecuadas. Son quienes van a validar si los requerimientos satisfacen las necesidades del cliente.

Durante el análisis de requerimientos se deben considerar los siguientes aspectos:

- **Objetivos del negocio y ambiente de trabajo.**

Cuando los objetivos del negocio están definidos, frecuentemente se usan para descomponer el trabajo en tareas específicas. El nuevo sistema cambiará el ambiente de trabajo, sin embargo, es muy difícil anticipar los efectos ocasionados sobre la organización. Los cambios no ocurren solamente cuando se implementa un nuevo software, también ocurren cuando cambia el ambiente que lo rodea (nuevas soluciones a problemas, nuevo equipo para instalar).

- **Punto de vista de los clientes.**

Muchos sistemas tienen varios tipos de clientes. Cada grupo de clientes tiene necesidades diferentes y los requerimientos tienen distintos grados de importancia para ellos. Por otro lado, escasas veces tenemos que los clientes son los mismos usuarios, trayendo como consecuencia que los clientes soliciten procesos que causan conflictos con los solicitados por el usuario. Diferentes puntos de vista también pueden tener consecuencias negativas, tales como datos redundantes, inconsistentes y ambiguos.

- **Barreras de comunicación.**

El análisis de requerimientos depende de una intensa comunicación entre clientes y analistas de requerimientos; sin embargo, existen problemas que no pueden ser resueltos mediante la comunicación. Para remediar esto, se deben abordar nuevas técnicas operacionales que ayuden a superar estas barreras y así ganar experiencia dentro del marco del sistema propuesto.

- **Comprensión de la evolución e integración del sistema.**

Pocos sistemas son construidos desde cero. En la práctica, los proyectos se derivan de sistemas ya existentes. Por lo tanto, los analistas de requerimientos deben comprender esos sistemas, que por lo general son una integración de componentes de varios proveedores. Para encontrar una solución a problemas de este tipo, es muy importante hacer planeamientos entre los requerimientos y la fase de diseño; esto minimizará la cantidad de fallas directas en el código.

- **Documentación de requerimientos.**

Los documentos de análisis de requerimientos son largos. La mayoría están compuestos de cientos o miles de páginas; cada página contiene muchos detalles que pueden tener efectos profundos en el resto del sistema. Normalmente, las personas se encuentran con dificultades para comprender documentos de este tamaño, sobre todo si no lo leen cuidadosamente. Es casi imposible leer un documento de especificación de gran tamaño, pues difícilmente una persona

LA IMPORTANCIA DE UN BUEN ANÁLISIS DE REQUERIMIENTOS

puede memorizar los detalles del documento. Esto causa problemas y errores que no son detectados hasta después de haberse construido el sistema.

En un proceso de análisis de requerimientos efectivo son esenciales diversas actividades, las cuales son aplicadas de manera continua y en orden variado. Dependiendo del tamaño del proyecto y del modelo de proceso de software utilizado para el ciclo de desarrollo, las actividades del análisis de requerimientos varían tanto en número como en nombres.

Actividades de la Ingeniería de Requerimientos para diferentes modelos de procesos de Ingeniería de Software					
MODELO	Oliver and Steiner 1996	EIA* / IS-632	IEEE Std 1220-1994	CMM* Nivel Repetitivo (2)	RUP*
ACTIVIDADES	Evaluar la información disponible	Análisis de requerimientos	Análisis de Requerimientos	Identificación de requerimientos	Análisis del Problema
	Definir métricas efectivas	Análisis funcional	Estudio de los requerimientos	Identificación de restricciones del sistema a desarrollar	Comprender las necesidades de los involucrados
	Crear un modelo del comportamiento del sistema	Síntesis	Validación de requerimientos	Análisis de los requerimientos	Definir el sistema
	Crear un modelo de los objetos	Análisis y control del sistema	Análisis funcional	Representación de los requerimientos	Analizar el alcance del proyecto
	Ejecutar el análisis		Evaluación y estudio de funciones	Comunicación de los requerimientos	Modificar la definición del sistema
	Crear un plan secuencial de construcción y pruebas		Verificación de funciones	Validación de requerimientos	Administrar los cambios de requerimientos
			Síntesis		
			Estudio y evaluación del diseño		
			Verificación física		
		Control			

* EIA: Electronic Industries Alliance

* CMM: Capability Maturity Model

* RUP: Rational Unified Process

Esta tabla⁴ muestra algunos ejemplos de las actividades identificadas para cada proceso. A pesar de las diferentes interpretaciones que cada analista tenga, podemos identificar y extraer cinco actividades principales que son:

- Análisis del problema
- Evaluación y negociación
- Especificación de requisitos de software
- Validación de requisitos
- Evolución

Análisis del Problema

El objetivo de esta actividad es entender las verdaderas necesidades del negocio. Antes de describir qué pasos deben cumplirse en esta actividad, debemos tener una definición clara del término "problema".

Un problema puede ser definido como la diferencia entre las cosas como se perciben y las cosas como se desean. Aquí vemos nuevamente la importancia que tiene una buena comunicación entre desarrolladores y clientes. De esta comunicación depende que entendamos sus necesidades.

Durante el análisis del problema se realizan una serie de pasos para garantizar un acuerdo entre los involucrados basados en los problemas reales del negocio, a saber:

1. Comprender el problema que se está resolviendo.

Es importante determinar quién tiene el problema realmente, considerar dicho problema desde una variedad de perspectivas y explorar muchas soluciones desde diferentes puntos de vista.

Veamos la siguiente necesidad: "El cliente se queja mucho por la enorme fila que debe formar para realizar una transacción bancaria".

Perspectiva del cliente = Pérdida de tiempo

Perspectiva del banco = Posible pérdida de clientes

Posibles soluciones: determinar por qué demoran los cajeros, colocar una nueva caja (implica contratación de nuevos cajeros), abrir una nueva sucursal (involucra personal nuevo y estudio de mercado), realizar transacciones por otros medios (teléfono, Internet, mediante cajeros automáticos, autobancos).

Como puede verse, múltiples soluciones aplican para el mismo problema, sin embargo, sólo una de ellas será la más factible. Las soluciones iniciales, deben ser definidas tomando en cuenta tanto la perspectiva técnica como la del negocio.

2. Construir un vocabulario común.

Debe confeccionarse un glosario en dónde se definan todos los términos que tengan significados comunes (sinónimos) y que serán utilizados durante el proyecto. La creación de un glosario es sumamente beneficiosa ya que reduce los términos ambiguos desde el principio, ahorra tiempo, asegura que todos los participantes de una reunión estén en la misma frecuencia, además de ser reutilizable en otros proyectos.

⁴ Las fuentes utilizadas para recopilar la información mostrada en la tabla se tomaron de un documento que presentó la cosultoría Asteci a BanCreceer.

3. Identificar a todo el personal afectado por el sistema.

Esto evitará que existan sorpresas a medida que avanza el proyecto. Las necesidades de cada afectado son discutidas y sometidas a debate durante el análisis de requerimientos, aunque esto no garantiza que vaya a estar disponible toda la información necesaria para especificar un sistema adecuado.

Para saber quiénes son las personas, departamentos, organizaciones internas o externas que se verán afectadas por el sistema, debemos realizar algunas preguntas.

- ¿Quién usará el sistema que se va a construir?
- ¿Quién desarrollará el sistema?
- ¿Quién probará el sistema?
- ¿Quién documentará el sistema?
- ¿Quién dará soporte al sistema?
- ¿Quién dará mantenimiento al sistema?
- ¿Quién mercadeará, venderá, y/o distribuirá el sistema?
- ¿Quién se beneficiará por el retorno de inversión del sistema?

Como vemos, debe conocerse la opinión de todo aquél que de una u otra forma está involucrado con el sistema, ya sea directa o indirectamente.

4. Definir los límites y restricciones del sistema.

Este punto es importante pues debemos saber lo que se está y no se está construyendo, para así entender la estrategia del producto a corto y largo plazo. Debe determinarse cualquier restricción ambiental, presupuestaria, de tiempo, técnica y de factibilidad que limite el sistema que se va a construir.

Evaluación y negociación de los requerimientos

La diversa gama de fuentes de las cuales provienen los requerimientos hacen necesaria una evaluación de los mismos antes de definir si son adecuados para el cliente. El término "adecuado" significa que ha sido percibido a un nivel aceptable de riesgo, tomando en cuenta las factibilidades técnicas y económicas, a la vez que se buscan resultados completos, correctos y sin ambigüedades.

En esta etapa se pretende limitar las expectativas del cliente apropiadamente, tomando como referencia los niveles de abstracción y descomposición de cada problema presentado.

Los principales pasos de esta actividad son:

- Descubrir problemas potenciales: En este paso se asegura que todas las características descritas anteriormente estén presentes en cada uno de los requerimientos, es decir, se identifican aquellos requerimientos ambiguos, incompletos, inconsistentes, etc.
- Clasificar los requerimientos: En este paso se busca identificar la importancia que tiene un requerimiento en términos de implementación. A esta característica se le conoce como prioridad y debe ser usada para establecer la secuencia en que ocurrirán las actividades de diseño y prueba de cada requisito. La prioridad de cada requerimiento dependerá de las necesidades que tenga el negocio. En base a la prioridad, cada requerimiento puede ser clasificado como mandatorio, deseable o innecesario.

Un requerimiento es mandatorio si afecta una operación crítica del negocio. Si existe algún proceso que se quiera incluir para mejorar los procesos actuales, estamos ante un requerimiento deseable; y si se trata de un requerimiento informativo o que puede esperar para fases posteriores, el requerimiento es catalogado como innecesario.

Una vez hecha esta categorización de los requerimientos, podemos tomar como estrategia general el incluir los mandatorios, discutir los deseables y descartar los innecesarios.

- Antes de decidir la inclusión de un requerimiento, también debe analizarse su costo, complejidad, y una cantidad de otros factores. Por ejemplo, si un requerimiento fuera trivial de implementar, puede ser una buena idea incluirlo por más que éste sea sólo deseable.
- Evaluar factibilidades y riesgos.

Involucra la evaluación de factibilidades técnicas (¿pueden implementarse los requerimientos con la tecnología actual?); factibilidades operacionales (¿puede ser el sistema utilizado sin alterar el organigrama actual?); factibilidades económicas (¿ha sido aprobado por los clientes el presupuesto?).

En la actividad de evaluación y negociación, se incrementa la comunicación entre el equipo de desarrollo y los afectados.

Para que los requerimientos puedan ser comunicados de manera efectiva, hay una serie de consideraciones que deben tomarse en cuenta, entre las principales tenemos:

- ❖ Documentar todos los requerimientos a un nivel de detalle apropiado.
- ❖ Mostrar todos los requerimientos a los involucrados en el sistema.
- ❖ Analizar el impacto que causen los cambios a requerimientos antes de aceptarlos.
- ❖ Establecer las relaciones entre requerimientos que indiquen dependencias.
- ❖ Negociar con flexibilidad para que exista un beneficio mutuo.
- ❖ Enfocarse en intereses y no en posiciones.

Especificación de requisitos de software

La especificación de requisitos de software es la actividad en la cual se genera el documento, con el mismo nombre, que contiene una descripción completa de las necesidades y funcionalidades del sistema que será desarrollado. Describe el alcance y la forma en como hará sus funciones, definiendo los requerimientos funcionales y los no funcionales. También se definen todos los requerimientos de hardware y software, diagramas, modelos de sistemas y cualquier otra información que sirva de soporte y guía para fases posteriores, por lo que servirá como fuente básica de comunicación entre los clientes, usuarios finales, analistas, personal de pruebas, y todo aquel involucrado en la implementación del sistema.

Los clientes y usuarios utilizan la especificación para comparar si lo que se está proponiendo coincide con las necesidades de la empresa. Los analistas y programadores la utilizan para determinar el producto que debe desarrollarse. El personal de pruebas elaborará las pruebas funcionales y de sistemas en base a este documento. Para el administrador del proyecto sirve como referencia y control de la evolución del sistema.

La especificación posee las mismas características de los requerimientos: completa, consistente, verificable, no ambigua, factible, modificable, rastreado, precisa, entre otras.

La estandarización de la especificación de requerimientos es fundamental pues ayudará, entre otras cosas, a facilitar la lectura y escritura de la misma. Será un documento familiar para todos los involucrados, además de asegurar que se cubren todos los tópicos importantes.

Validación de requerimientos

La validación es la actividad que permite demostrar que los requerimientos definidos en el sistema son los que realmente quiere el cliente; además revisa que no se haya omitido ninguno, que no sean ambiguos, inconsistentes o redundantes.

En este punto es necesario recordar que la especificación de los requerimientos debe estar libre de errores, por lo tanto, la validación garantiza que todos los requerimientos presentes en el documento de especificación sigan los estándares de calidad.

No debe confundirse la actividad de evaluación de requerimientos con la de validación de requerimientos. La evaluación verifica las propiedades de cada requerimiento, mientras que la validación revisa el cumplimiento de las características de la especificación de los mismos.

Durante la actividad de validación pueden hacerse preguntas en base a cada una de las características que se desean revisar. A continuación se presentan varios ejemplos:

- ◆ ¿Están incluidas todas las funciones requeridas por el cliente? (completo)
- ◆ ¿Existen conflictos en los requerimientos? (consistencia)
- ◆ ¿Tiene alguno de los requerimientos más de una interpretación? (no ambiguo)
- ◆ ¿Está cada requerimiento claramente representado? (entendible)
- ◆ ¿Pueden los requerimientos ser implementados con la tecnología y el presupuesto disponible? (factible)
- ◆ ¿Están las especificaciones escritas en un lenguaje apropiado? (claro)
- ◆ ¿Existe facilidad para hacer cambios en los requerimientos? (modificable)
- ◆ ¿Está claramente definido el origen de cada requisito? (rastreadable)
- ◆ ¿Pueden los requerimientos ser sometidos a medidas cuantitativas? (verificable)

La validación de requerimientos es importante pues de ella depende que no existan elevados costos de mantenimiento para el software desarrollado.

Evolución

Los requerimientos son una manera de comprender mejor el desarrollo de las necesidades de los usuarios y cómo los objetivos de la organización pueden cambiar, por lo tanto, es esencial planear posibles cambios a los requerimientos cuando el sistema sea desarrollado y utilizado. La actividad de evolución es un proceso externo que ocurre a lo largo del ciclo de vida del proyecto.

Los requerimientos cambian por diferentes razones. Las más frecuentes son:

- Porque al analizar el problema, no se hacen las preguntas correctas a las personas correctas.
- Porque cambió el problema que se estaba resolviendo.
- Porque los usuarios cambiaron su forma de pensar o sus percepciones.
- Porque cambió el ambiente de negocios.
- Porque cambió el mercado en el cual se desenvuelve el negocio.

LA IMPORTANCIA DE UN BUEN ANÁLISIS DE REQUERIMIENTOS

Cambios en los requerimientos impactan no solo en el tiempo inicialmente estimado, también pueden tener impacto en otros requerimientos. Por esto, la administración de cambios involucra actividades como establecer políticas, guardar históricos de cada requerimiento, identificar dependencias entre ellos y mantener un control de versiones.

En vista de que las peticiones de cambios provienen de muchas fuentes, las mismas deben ser enrutadas en un solo proceso. Esto se hace con la finalidad de evitar problemas y conseguir estabilidad en los requerimientos.

Análisis comparativo de las técnicas de ingeniería de requerimientos

El cuadro a continuación muestra las técnicas que pueden ser utilizadas en las diferentes actividades de la ingeniería de requerimientos:

	Análisis del Problema	Evaluación y negociación	Especificación de Requisitos	Validación	Evolución
Entrevistas y Cuestionarios	X				X
Lluvia de Ideas		X			X
Prototipos				X	
Análisis Jerárquico		X			X
Casos de Uso	X		X		X

Haciendo un análisis comparativo entre estas técnicas, podemos acotar lo siguiente:

- Entrevistas / Casos de Uso: Un alto porcentaje de la información recolectada durante una entrevista, puede ser usada para construir casos de uso. Mediante esto, el equipo de desarrollo puede entender mejor el ambiente de trabajo de los involucrados. Cuando el analista sienta que tiene dificultades para entender una tarea, pueden recurrir al uso de un cuestionario y mostrar los detalles recabados en un caso de uso. De hecho, durante las entrevistas cualquier usuario puede utilizar diagramas de casos de uso para explicar su entorno de trabajo.
- Entrevistas / Lluvia de Ideas: Muchas de las ideas planteadas en el grupo, provienen de información recopilada en entrevistas o cuestionarios previos. Realmente la lluvia de ideas trata de encontrar las dificultades que existen para la comprensión de términos y conceptos por parte de los participantes; de esta forma se llega a un consenso.
- Casos de Uso / Lluvia de Ideas: La lista de ideas proveniente del brainstorm⁵ puede ser representada gráficamente mediante casos de uso.

A continuación se muestra una tabla con el detalle de este análisis:

⁵ Término inglés para referirse al conjunto de ideas planteadas en un grupo. Lluvia de ideas.

LA IMPORTANCIA DE UN BUEN ANÁLISIS DE REQUERIMIENTOS

TECNICA	VENTAJAS	DESVENTAJAS
Entrevistas y Cuestionarios	<ul style="list-style-type: none"> • Mediante ellas se obtiene una gran cantidad de información correcta a través del usuario. • Pueden ser usadas para obtener un pantallazo del dominio del problema. Son flexibles. • Permiten combinarse con otras técnicas. 	<ul style="list-style-type: none"> • La información obtenida al principio puede ser redundante o incompleta. • Si el volumen de información manejado es alto, requiere mucha organización de parte del analista, así como la habilidad para tratar y comprender el comportamiento de todos los involucrados.
Lluvia de Ideas	<ul style="list-style-type: none"> • Los diferentes puntos de vista y las confusiones en cuanto a terminología, son aclarados por expertos. • Ayuda a desarrollar ideas unificadas basadas en la experiencia de un experto. 	<ul style="list-style-type: none"> • Es necesaria una buena compenetración del grupo participante.
Prototipos	<ul style="list-style-type: none"> • Ayudan a validar y desarrollar nuevos requerimientos. • Permiten comprender aquellos requerimientos que no están muy claros y que son de alta volatilidad. 	<ul style="list-style-type: none"> • El cliente puede llegar a pensar que el prototipo es una versión del software que será desarrollado. • A menudo, el desarrollador hace compromisos de implementación con el objetivo de acelerar la puesta en funcionamiento del prototipo. • No aplica en sistemas grandes o complejos.
Análisis Jerárquico	<ul style="list-style-type: none"> • Permite determinar el grado de importancia de cada requerimiento. • Ayuda a identificar conflictos en los requerimientos. • Muestra el orden en que deben ser implementados los requerimientos. 	<ul style="list-style-type: none"> • Debe construirse un estándar claro de evaluación, que incluya la participación del cliente.
Casos de Uso	<p>Representan los requerimientos desde el punto de vista del usuario. Permiten representar más de un rol para cada afectado. Identifica requerimientos estancados, dentro de un conjunto de requerimientos.</p>	<ul style="list-style-type: none"> • En sistemas grandes, toma mucho tiempo definir todos los casos de uso. • El análisis de calidad depende de la calidad con que se haya hecho la descripción inicial.

COMENTARIOS

Aunque existen diversas razones que dificultan el mantenimiento del software, la falta de atención al análisis de requerimientos es una de las principales razones por las cuales existen sistemas inadecuados. Aún quedan muchos desafíos que deben ser mejorados, tales como la integración de requerimientos funcionales y no funcionales, la evaluación de especificaciones alternativas así como la formalización de la especificación de requerimientos, entre otras.

Cada actividad y técnica de la ingeniería de requerimientos utilizada individualmente dará diferentes soluciones para diferentes proyectos, incluyendo aquellos casos en los que el dominio y el área del problema son el mismo. Por esta razón, considero que no existe un modelo de proceso ideal para el análisis de requerimientos. Encontrar el método o la técnica perfecta es una ilusión, pues cada método y técnica ofrece diferentes soluciones ante un problema.

En este capítulo se presentaron una serie de actividades y técnicas, que no pertenecen a un modelo de proceso en sí, sino que son una alternativa al material publicado por diferentes autores y que, desde mi punto de vista, son las más importantes.

Debemos recordar que la ingeniería de requerimientos es una actividad que involucra a clientes, usuarios, equipo de desarrollo, administradores de proyectos, etc; por lo tanto, el proceso de análisis de requerimientos no depende solamente de la forma en cómo se percibe el problema, sino también del nivel de experiencia que tengan los involucrados.

Tomando en cuenta la magnitud de comunicación y el trabajo en equipo que debe existir en la ingeniería de requerimientos, considero necesario enfatizar más en cerrar las brechas que todavía existen, incluyendo los siguientes elementos:

- ❖ Factores sociales: involucrar al grupo para compartir sus experiencias.
- ❖ Factores de problemas específicos: el dominio de la estructura y estándares disponibles.
- ❖ Factores organizacionales: tiempo y costo presupuestados.
- ❖ Factores de diseño: por ejemplo, interfases de usuario.

Es importante tomarse el tiempo necesario para conocer a nuestros clientes y usuarios, así como su ambiente de trabajo. Esto también ayuda a establecer una buena relación de trabajo y comunicación entre el equipo de desarrollo y los clientes. Es realmente necesario que los clientes y usuarios participen en la definición de sus requerimientos, pues ellos son los que deciden nuestro destino en el proyecto, deciden si les gustamos o no y además financian el proyecto.

En cuanto al apartado concerniente a la técnica de casos de uso para la ingeniería de requerimientos, puede decirse que éstos son independientes del método de diseño que se utilice, y por lo tanto, del método de programación. Luego de documentar los requerimientos de un sistema con casos de uso, se puede diseñar un sistema "estructurado" (manteniendo una separación entre datos y funciones), o un sistema orientado a objetos, sin que la técnica sea de mayor o menor utilidad en alguno de los dos casos. Esto da más flexibilidad al método y probablemente contribuya a su éxito.

Otro punto a considerar, es la inclusión del término "administración de requerimientos" en la década de los 90. Con esta nueva visión, se busca encontrar una descripción más apropiada de las actividades involucradas, a la vez de enfatizar la importancia de mantener una buena relación entre los afectados y el equipo del proyecto.

CAPÍTULO IV

LA IMPORTANCIA DEL DESARROLLO DE SOFTWARE CON CALIDAD

*"Hombres y mujeres quieren hacer un buen trabajo.
Si se les proporciona el entorno adecuado,
lo harán"...*

- Bill Hewlett, fundador de Hewlett Packard

INTRODUCCIÓN

Como hemos observado a lo largo del tiempo, los avances en la tecnología siempre van parejos con progresos en los lenguajes de programación y con nuevas ayudas para simplificar el uso de la computadora, con lo cual un número mayor de usuarios se beneficia; sin embargo, la necesidad de hacer programas para resolver problemas específicos quizás nunca desaparecerá.

A través de los años se han inventado decenas de lenguajes de alto nivel, pero solo unos cuantos han logrado una amplia aceptación, dependiendo del área en la que se apliquen.

Durante la década de 1960, muchos proyectos de desarrollo de software de gran importancia se toparon con graves problemas. Los procesos generalmente se atrasaban, los costos excedían en mucho los presupuestos y los productos terminados eran poco confiables. Los especialistas comenzaron a darse cuenta de que la creación de software era una actividad mucho más compleja de lo que se habían imaginado. Investigaciones realizadas en esa misma década tuvieron como resultado la aparición de la programación estructurada: un enfoque disciplinado para escribir programas más claros que los no estructurados, más fáciles de probar y depurar, y más fáciles de modificar.

Uno de los resultados más tangibles de las investigaciones mencionadas fue la invención del lenguaje de programación Pascal, por Nicklaus Wirth en 1971. Este lenguaje, llamado así en honor del matemático y filósofo del siglo XVII, Blaise Pascal, fue diseñado para enseñar programación estructurada en ambientes académicos y rápidamente se convirtió en el lenguaje de programación preferido en la mayor parte de las universidades. Desafortunadamente, este lenguaje carece de muchas características necesarias para hacerlo útil en aplicaciones comerciales, industriales y gubernamentales, por lo que no fue aceptado ampliamente en dichos entornos.

En México actualmente en el medio bancario, por ejemplo, se encuentra ampliamente difundido el lenguaje de programación COBOL (Common Business Oriented Language, lenguaje orientado hacia los negocios comunes) el cual fue desarrollado en 1959 por un grupo de fabricantes de computadoras y usuarios gubernamentales e industriales de las mismas. Se usa principalmente para aplicaciones comerciales que requieren una manipulación precisa y eficiente de grandes cantidades de datos. Hoy en día, más de la mitad de todo el software de negocios se sigue programando en Cobol. El sistema central de BanCrecer (Altamira) se codifica en Cobol.

No obstante, el creciente empleo de las computadoras ha conducido a buscar un abaratamiento del desarrollo de software, paralelo a la reducción del costo del hardware obtenido gracias a los avances tecnológicos. Los altos costos del mantenimiento de las aplicaciones en producción normal también han urgido la necesidad de mejorar la productividad del personal de programación.

La computadora debe su derecho de existir, su utilidad, precisamente a su capacidad de efectuar vastos cálculos que no pueden realizar los seres humanos. Deseamos que la computadora efectúe lo que nunca podríamos hacer nosotros, y la potencia de las máquinas actuales es tal, que inclusive los cálculos pequeños escapen al poder de nuestra imaginación limitada debido a la velocidad con que se procesan. Sin embargo, debemos organizarnos de manera tal que nuestros limitados poderes sean suficientes para asegurar que se establecerá el efecto deseado. Esta organización incluye la composición de los programas.

La mayoría de las personas tienen una buena idea de las cosas fascinantes que pueden hacer las computadoras. Nosotros, como programadores, aprendemos cómo ordenar a las computadoras que hagan esas cosas, y es muy importante que, a pesar de hallarnos inmersos en todos los avances tecnológicos que hoy conducen nuestras vidas, no perdamos de vista el desarrollo de software con calidad.

En los principios de la computación, los programadores no realizaban análisis muy profundos sobre el problema para resolver. Si acaso, garabateaban algo en un papel. Con frecuencia comenzaban a escribir el programa desde el principio, y el código necesario iba saliendo conforme se iba requiriendo. Aunque anteriormente tal vez esto agregaba un aura de aventura y atrevimiento al proceso, en la actualidad eso es inapropiado en los negocios de alto riesgo.

En este capítulo se remarca la importancia de la ingeniería de software y se comentan brevemente algunas técnicas de programación que no deben faltar en la consecución del desarrollo de software con calidad.

DESARROLLO DE SOFTWARE

Ingeniería de software

El concepto de ingeniería de software se refiere al conjunto de disciplinas usadas para la especificación, diseño e implantación de software para computadoras. Es una disciplina de la computación que ofrece métodos y técnicas para desarrollar y mantener productos de software que resuelven problemas de todo tipo, siendo su enfoque primario la lógica utilizada en los procesos computarizados.

La metodología de la ingeniería de software se llega a formalizar a finales de los 70's e involucra técnicas para desarrollo de software tales como la programación estructurada, análisis y diseño estructurado y herramientas para soportarlas, llegando al rescate de los desarrolladores que atravesaban por una crisis que se caracterizaba principalmente por los siguientes problemas:

- Imprecisión en la planificación del proyecto y estimación de los costos.
- Baja calidad del software.
- Dificultad de mantenimiento de programas con un diseño poco estructurado.

Sin embargo, y como ya se ha visto, con el surgimiento de esta disciplina los problemas no se solucionaron mágicamente. Actualmente, las organizaciones donde se desarrollan productos de software pueden estar en un nivel de inmadurez del proceso si se cumplen algunos de los siguientes puntos:

- El proceso de desarrollo de software es improvisado.
- Aún si existe un proceso especificado, este no se sigue ni se toma en cuenta rigurosamente.
- La organización es reactiva, enfocada en resolver crisis inmediatas.
- La organización generalmente sobrepasa los presupuestos y los tiempos de entrega.
- En muchos casos, compromete la funcionalidad y la calidad del producto en aras de cumplir con los calendarios.
- No existen bases objetivas para juzgar la calidad del producto ni para resolver los problemas del proceso.
- Cuando un proyecto se retrasa, generalmente se recortan o incluso se eliminan las tareas de revisión y pruebas.
- Los clientes o usuarios finales tienen muy poca participación en el proceso hasta que el producto es liberado.

La ingeniería de software es una tecnología de varios elementos donde cualquiera de sus enfoques debe descansar sobre un esfuerzo de organización de calidad. La administración con calidad fomenta una cultura continua de mejoras de procesos, y es esta cultura la que conduce finalmente al desarrollo de productos exitosos. Los cimientos que son la base de la ingeniería de software están orientados hacia la calidad.

Por otra parte se exige que el software sea eficaz y barato tanto en el desarrollo como en la compra. También se requiere una serie de características como eficiencia, fiabilidad, facilidad de mantenimiento y facilidad de uso.

Objetivos de la ingeniería de software

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas; la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software. Sus principales objetivos son:

- Mejorar la calidad de los productos de software.
- Aumentar la productividad y trabajo de los desarrolladores del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos de software desarrollados en el plazo fijado y dentro del costo estimado.

Para que esos objetivos se cumplan, se deben considerar "las cinco C":

1. Capacidad.

Las actividades de la organización están influenciadas por la capacidad de ésta para procesar transacciones con rapidez y eficiencia. Los sistemas de información basados en computadora mejoran esta capacidad en tres formas:

- (a) Aumentan la velocidad de procesamiento.
- (b) Incrementan el volumen de información a procesar.
- (c) Recuperan rápidamente la información.

2. Costo.

Para determinar si la compañía evoluciona en la forma esperada, de acuerdo con lo presupuestado, se debe llevar a cabo el seguimiento de los costos de mano de obra, bienes y gastos generales. La creciente competitividad del mercado crea la necesidad de mejores métodos para disminuir los costos y relacionarlos con la productividad individual y organizacional. Los sistemas por computadora ayudan a disminuir los costos, ya que toman ventaja de las capacidades de cálculo automático y de recuperación de datos que están incluidos en procedimientos de programas. Muchas tareas son realizadas por programas de cómputo, lo cual deja un número muy reducido de éstas para su ejecución manual, disminuyendo al personal.

3. Control.

Mediante el uso de procedimientos de control por lotes, tratando de que siempre se siga el mismo procedimiento. Cada paso se lleva a cabo de la misma manera, consistencia y con exactitud. A diferencia del ser humano, el sistema no se distrae con llamadas telefónicas, ni olvidos e interrupciones.

4. Comunicación.

La falta de comunicación es una fuente común de dificultades que afectan tanto a clientes como a empleados. Sin embargo, los sistemas de información bien desarrollados amplían la comunicación y facilitan la integración de funciones individuales, por medio de los siguientes mecanismos:

- a) Interconexión (aumento en la comunicación): Muchas empresas aumentan sus vías de comunicación por medio del desarrollo de redes para este fin, dichas vías abarcan todo el país y les permiten acelerar el flujo de información dentro de sus oficinas y otras instalaciones que no se encuentran en la misma localidad. Una de las características más importantes de los sistemas de información para oficinas es la transmisión electrónica de información, como por ejemplo, los mensajes y los documentos vía e-mail.
- b) Integración de áreas en las empresas: Con frecuencia las actividades de las empresas abarcan varias áreas de la organización, la información que surge en un área se necesita en otra.

5. Competitividad.

Los sistemas de información computacionales son un arma estratégica capaz de cambiar la forma en que la compañía compite en el mercado, en consecuencia éstos sistemas mejoran la organización y la ayudan a ganar ventaja competitiva.

Estrategias para el desarrollo de sistemas

En algunos casos los factores que deben considerarse en un proyecto de sistema de información, como el aspecto más apropiado de la computadora o la tecnología de comunicaciones que se va a utilizar, el impacto del nuevo sistema sobre los empleados de la empresa y las características específicas que el sistema debe tener, se pueden determinar de manera secuencial. Todas estas situaciones están determinadas por los siguientes métodos básicos:

a) *Método del ciclo de vida clásico*

Tema que fue expuesto en el primer capítulo.

b) *Método de desarrollo por análisis estructurado*

Muchos especialistas en sistemas de información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método de desarrollo del análisis estructurado tiene como finalidad superar esta dificultad por medio de:

- ⇒ La división del sistema en componentes
- ⇒ La construcción de un modelo del sistema

El método incorpora elementos tanto de análisis como de diseño: el análisis estructurado se concentra en especificar lo que se requiere que haga el sistema o la aplicación. Permite que las personas observen los elementos lógicos (lo que hará el sistema) separados de los componentes físicos (computadora, terminales, sistemas de almacenamiento). Después de esto se puede desarrollar un diseño físico eficiente para la situación donde será utilizado, donde el principal objetivo es generar programas formados por módulos independientes unos de otros desde el punto de vista funcional.

c) *Método del desarrollo por prototipos*

Los prototipos permiten evaluar situaciones extraordinarias donde los encargados de diseñar e implantar sistemas no tienen información ni experiencia, o también donde existen situaciones de riesgo y costos elevados, y aquellas donde el diseño propuesto es novedoso y aún no ha sido probada. La información obtenida con su uso se aplica en un nuevo diseño que se emplea, otra vez, como prototipo y que revela más información valiosa sobre el diseño. El proceso se repite las veces que sea necesario para revelar los requerimientos esenciales del diseño.

En algunos casos donde el sistema no será utilizado frecuentemente o cuando no son muchos los beneficios que se obtienen, puede convertirse el prototipo en el sistema terminado.

d) UML

El Lenguaje Unificado de Modelado (en inglés Unified Modeling Language) es un sistema de notación que se ha convertido en estándar en el mundo de desarrollo de sistemas. Es el resultado del trabajo hecho por Grady Booch, James Rumbaugh e Ivar Jacobson. Está constituido por un conjunto de diagramas, y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Es necesario contar con todos esos diagramas (de casos de uso, de clases y objetos, de estado, de secuencia, etc) dado que cada uno se dirige a cada tipo de persona implicada en el sistema.

Un modelo UML indica qué es lo que supuestamente hará un sistema, mas no cómo lo hará y es, por tanto, independiente del lenguaje de programación que se empleará, aunque es básicamente asociado con los desarrollos en lenguajes de programación orientados a objetos.

Herramientas para el desarrollo de sistemas.

Las herramientas son cualquier dispositivo que, empleándose adecuadamente, mejoran el desempeño del desarrollo de sistemas de información. Las herramientas automatizadas se pueden agrupar como sigue:

1) Herramientas de tipo Front-end

Automatizan las primeras actividades del proceso de desarrollo de sistemas y proporcionan soporte para el desarrollo de modelos gráficos de sistemas y procesos.

Los diagramas de flujo son representativos de este tipo de herramientas.

2) Herramientas para análisis

Estas herramientas ayudan a documentar un sistema existente, ya sea manual o automatizado. También sirven para determinar los requerimientos de una nueva aplicación. Este tipo de herramientas incluyen:

- Herramientas para recolección de datos: capturan detalles que describen sistemas y procedimientos en uso. Documentan procesos y actividades de decisión, se utilizan para apoyar la tarea de identificar requerimientos.
- Herramientas para diagramación: crean representaciones gráficas de sistemas y actividades. Apoyan el dibujo y revisión de diagramas de flujos de datos e iconos asociados con el análisis estructurado. Incluyen programas para representación en diagramas de flujo.
- Herramientas para el diccionario: registran y mantienen descripciones de los elementos del sistema, como grupo de datos, procesos, alimentos de datos, etc. Frecuentemente proporcionan la capacidad de examinar las descripciones del sistema, para decidir si son incompletas o inconsistentes.

3) Herramientas para diseño

Apoyan el proceso de formular las características que el sistema debe tener para satisfacer los requerimientos deseados durante las actividades de análisis. Incluyen:

- Herramientas de especificación: apoyan el proceso de formular las características, como por ejemplo deben tener una aplicación como entradas, salidas, procesamientos específicos de control.
- Herramientas para presentación: se utilizan para describir la posición de datos, mensajes, y encabezados sobre las pantallas de las terminales, informes y otros medios de entradas y salidas.

Los analistas utilizan las herramientas para el diseño de sistemas desde el inicio de la era de las computadoras. Ahora a las herramientas se les está dando un nuevo significado en el diseño de software.

4) Herramientas de tipo back-end

Su finalidad es ayudar al analista a formular la lógica del programa, los algoritmos de procesamiento y la descripción física de datos. También ayudan a la intersección con los dispositivos (para entrada y salida). Estas actividades convierten los diseños lógicos del software en un código de programación; este es el que da existencia a la aplicación.

5) Herramientas para el desarrollo

Ayudan al analista a trasladar los diseños en aplicaciones funcionales. Incluyen:

- Herramientas para ingeniería de software: apoyan el proceso de formular diseños de software, incluyendo procesamientos y controles.
- Generadores de códigos: producen el código fuente y las aplicaciones a partir de especificaciones funcionales bien articuladas.
- Herramientas para pruebas: apoyan la fase evaluación de un sistema. Incluyen facilidades para examinar la correcta operación del sistema.

6) Herramientas integrales

Proporcionan un ambiente que automatiza tareas claves a lo largo del proceso de desarrollo. Estas herramientas facilitan el diseño, administración y mantenimiento del código. Brindan un ambiente eficiente para crear, almacenar, manipular y documentar sistemas.

Reingeniería e ingeniería inversa

Los conceptos de reingeniería e ingeniería inversa están ligados al desarrollo de software a gran escala, donde una mejora en el proceso de este desarrollo supone un aumento en la competitividad de la empresa. Aunque hay que tener en cuenta que esta mejora es, en general, a largo plazo (normalmente de uno a dos años), ambas actividades están orientadas a automatizar el mantenimiento de aplicaciones. Esta es una tarea que consume gran cantidad de recursos, por lo que cualquier reducción en el tiempo y recursos empleados en ella supone una importante mejora en la productividad del proceso. Este es el principal objetivo de la reingeniería.

Se trata de analizar el código o el diseño actual y modificarlo con la ayuda de herramientas automáticas para traducirlos a códigos más estructurados y más eficientes. Dentro de la reingeniería, el proceso de pasar del código a una descripción de más alto nivel es lo que se denomina 'Ingeniería inversa'. La reingeniería e ingeniería inversa prolongan la vida del software.

Dado que es una labor estratégica, es importante conocer cuándo conviene realizar la tarea de reingeniería para una aplicación y cuándo es más rentable sustituirla e implementar una nueva. Las aplicaciones para el primer paso, son aquellas en la que se produce las siguientes situaciones:

- Fallos frecuentes, que son difíciles de localizar
- Son poco eficientes, pero realizan la función esperada
- Dificultades en la integración con otros sistemas
- Calidad pobre del software final
- Resistencia a introducir cambios
- Pocas personas capacitadas para realizar modificaciones
- Dificultades para realizar pruebas
- El mantenimiento consume muchos recursos
- Es necesario incluir nuevos requisitos, pero los básicos se mantienen

Desarrollo de software con y para reuso

El desarrollo de software con reuso consiste en desarrollar una aplicación usando software ya existente. Cualquier profesional lo puede utilizar. Consiste en la construcción de un sistema con la intención de reutilizar partes de él en futuros desarrollos. Con software a gran escala, un buen profesional con experiencia puede desarrollarlo.

Estudios realizados determinan que la práctica de reutilización del software en un proyecto aumenta la productividad durante el desarrollo de dicho proyecto. Sin embargo, la reutilización del software no cubre solo el reuso de códigos, abarca toda una amplia gama de posibilidades en los diferentes niveles, metodología, ciclos de vida, planes del proyecto, especificaciones de requisitos, diseños, arquitectura de software, planes de validación, juegos de prueba y documentación.

Estrategias de programación

Se dice que la orientación a objetos es la metodología de programación clave desde mediados de los años 90. Sin embargo, con frecuencia la mejor manera de crear la estructura interna de dichos objetos, es utilizando las técnicas de la programación estructurada. Además, hay ocasiones en que la mejor manera de la lógica de la manipulación de objetos es con la programación estructurada.

El mundo de la programación se ha vuelto más complejo y exigente. Hoy en día los usuarios quieren aplicaciones que cuenten con las siguientes características:

- Que estén provistas de interfases gráficas y que les permitan aprovechar las capacidades de multimedia: gráficos, imágenes, animación, audio e incluso video.
- Que puedan ejecutarse en redes de computadoras y comunicarse con otras aplicaciones.
- Que aprovechen la flexibilidad y mejoras de rendimiento.
- Que tengan más amplias capacidades de procesamiento de archivos.
- Que no estén limitadas a su escritorio o incluso a la red local de su organización, sino que puedan integrar componentes de la Internet.

- Que puedan escribirse con rapidez, aprovechando un mundo en el que el lema es reutilizar, teniendo acceso a un universo en expansión de componentes de software reutilizables.

Para hacer frente a estas expectativas debemos explotar los recursos a nuestro alcance. A continuación, se muestra una breve reseña de las dos estrategias de programación más utilizadas.

a) *Programación estructurada.*

Los programas computarizados pueden ser escritos con un alto grado de estructuración, lo cual los hace más fáciles de comprender en actividades tales como pruebas, mantenimiento y modificación de los mismos. Mediante la programación estructurada todas las bifurcaciones de control de un programa se encuentran estandarizadas, de forma tal que es posible leer la codificación del mismo desde su inicio hasta su terminación en forma continua, sin tener que saltar de un lugar a otro del programa siguiendo el rastro de la lógica establecida por el programador, como es la situación habitual con codificaciones desarrolladas bajo otras técnicas.

En programación estructurada los programadores deben profundizar más que lo usual al proceder a realizar el diseño original del programa, pero el resultado final es más fácil de leer y comprender. El objetivo de un programador profesional al escribir programas de una manera estructurada es realizarlos utilizando solamente un número de bifurcaciones de control estandarizados.

El resultado de aplicar la sistemática y disciplinada manera de elaboración de programas establecida por la programación estructurada es una programación de alta precisión. Las pruebas de los programas desarrollados utilizando este método se acoplan rápidamente y el resultado final son programas que pueden ser leídos, mantenidos y modificados por otros programadores con mucha mayor facilidad.

Básicamente, la programación estructurada es una técnica en la cual la estructura de un programa, esto es, la interpelación de sus partes, se realiza tan claramente como es posible mediante el uso de tres estructuras lógicas de control:

- a. Secuencia: Sucesión simple de dos o más operaciones.
- b. Selección: Bifurcación condicional de una o más operaciones.
- c. Interacción: Repetición de una operación mientras se cumple una condición.

Estos tres tipos de estructuras lógicas de control pueden ser combinados para producir programas que manejen cualquier tarea de procesamiento de información.

La facilidad de lectura, de comienzo a fin, es una consecuencia de utilizar solamente estas tres estructuras lógicas, y de eliminar la instrucción de desvío de flujo de control.

Un programa escrito de acuerdo a estos principios no solamente tendrá una excelente estructura, sino también una excelente presentación. La estructura del programa es más clara puesto que las instrucciones están más ligadas o relacionadas entre sí, por lo que es más fácil entender lo que hace cada función.

Sin embargo, la programación estructurada concentra los errores en uno de los factores más generadores de fallas en programación: *la lógica.*

No obstante, el seguimiento de las fallas ("debugging") se facilita debido a la lógica más visible, de tal forma que los errores se pueden detectar y corregir más fácilmente.

Un programa que es fácil para leer y está compuesto de segmentos bien definidos tiende a ser simple, rápido y menos expuesto a mantenimiento. Estos beneficios derivan en parte del hecho que, aunque el programa tenga una extensión significativa, en documentación tiende siempre a estar al día, utiliza técnicas de diseño descendentes (top-down) o ascendentes (bottom-up) para fragmentar y disminuir la complejidad funcional.

b) Programación orientada a objetos.

La orientación a objetos ha tomado por asalto y en forma legítima al mundo del software. Como medio para la generación de programas, tiene varias ventajas: fomenta una metodología basada en componentes para el desarrollo de software, de manera que primero se genera un sistema mediante un conjunto de objetos¹, luego se podrá ampliar ese sistema agregándole funcionalidad a los componentes que ya se habían generado o agregándole nuevos componentes, y finalmente se pueden volver a utilizar los componentes que ya se generaron para el sistema cuando se cree uno nuevo, con lo cual se reducirá sustancialmente el tiempo de desarrollo de un sistema.

La orientación a objetos es tan importante para el diseño de software que el OMG (Object Management Group, una corporación no lucrativa que establece las normas para el desarrollo orientado a objetos) predice que los ingresos obtenidos por el software generado bajo esta disciplina, será de 3 millones de dólares en los próximos 3 a 5 años². El UML influye en esto al permitirnos generar modelos de objetos fáciles de usar y comprender, para que los desarrolladores podamos convertirlos en software.

La orientación a objetos es un paradigma que depende de ciertos principios fundamentales. Los objetos, concretos y virtuales, están a nuestro alrededor, ellos conforman nuestro mundo. El software actual simula al mundo (o un segmento de él), y los programas, por lo general, imitan a los objetos del mundo. Si comprendemos algunas cuestiones básicas de los objetos, entenderemos cómo se deben mostrar éstos en las representaciones de software.

Un objeto cuenta con una estructura, es decir, atributos y acciones, que en conjunto se conocen como características o rasgos. Se dice que un objeto es una instancia de una clase, la cual es como una plantilla que sirve para crear objetos.

Aunque en este apartado no entraremos a los detalles de la programación, es importante recordar que el propósito de la orientación a objetos es desarrollar software que refleje particularmente (es decir, que modele) un esquema del mundo. Entre más atributos y acciones se tomen en cuenta, mayor será la similitud del modelo con la realidad.

Sin embargo, la orientación a objetos se refiere a algo más que sólo atributos y acciones, pues también considera otros aspectos básicos. Dichos aspectos se conocen como *abstracción*, *herencia*, *polimorfismo* y *encapsulación*. Y existen aún otros más como el *envío de mensajes*, las *asociaciones* y la *agregación*.

¹ Un objeto es una entidad individual que tiene valores específicos como atributos (propiedades) y acciones (todas las actividades que es capaz de realizar). Es, a su vez, parte de un conjunto más general, al que se denomina clase. Por lo que, una clase es una colección de objetos con atributos y acciones similares.

² Estadística tomada del libro 'Aprendiendo UML en 24 horas', en su edición en español, año 2000.

La abstracción se refiere a quitar las propiedades y acciones de un objeto para dejar sólo aquellas que sean necesarias. Diferentes tipos de problemas requieren distintas cantidades de información, aún si estos problemas pertenecen a un área en común.

La herencia es un aspecto importante de la orientación a objetos: un objeto hereda los atributos y operaciones de su clase, la cual a su vez también puede heredar atributos y acciones de otra, llamada superclase.

El polimorfismo especifica que una acción puede tener el mismo nombre en diferentes clases y cada clase ejecutará tal operación de forma distinta.

Mediante la encapsulación, los objetos ocultan su funcionalidad de otros objetos y del mundo exterior. Cada objeto presenta una interfaz para que otros objetos (y personas) puedan aprovechar su funcionalidad.

También, los objetos funcionan en conjunto mediante el envío de mensajes entre ellos. Los mensajes aquí son peticiones para realizar alguna operación.

A su vez, los objetos se asocian entre sí y esta característica puede ser de distintos tipos. Un objeto en una clase puede asociarse con cualquier cantidad de objetos distintos en otra clase.

La agregación es un tipo de asociación. Un objeto agregado consta de un conjunto de objetos, y una composición de éstos es un tipo esencial de agregación.

Finalmente, lo importante aquí es utilizar los conceptos de la orientación a objetos para ayudarnos a entender el área de conocimiento de nuestros usuarios o clientes (su dominio), y esclarecer sus puntos de vista en términos que ellos puedan comprender.

Todas estas características hacen de la orientación a objetos una metodología totalmente flexible para la generación y reutilización de software, así como dar un enfoque ilimitado en la creación de modelos y programas basados en objetos. Ello permite dar soporte a las necesidades actuales que los usuarios manifiestan en sus sistemas.

COMENTARIOS

Como ya se ha dicho a lo largo del capítulo, en el desarrollo actual de software es muy importante observar ciertas reglas que nos permitan elaborar productos de calidad. Para ello se nos ofrece todo un mundo de alternativas, solo basta con llevarlas a la práctica de la mejor manera posible y ser conscientes de la responsabilidad que conlleva el desarrollo de software.

Entregar productos de calidad, a tiempo y dentro del presupuesto, hará que nuestros clientes confíen en nosotros y asegurará el crecimiento y madurez de la relación de negocio.

Partiendo de mi experiencia personal como programadora, me permito hacer las siguientes recomendaciones:

1. Antes de sentarse a codificar, se debe estudiar el problema en general y ver qué parte nos interesa.

2. Hay que comprender claramente el problema para el cual tenemos que obtener la especificación de las entradas y salidas como resultado. Tiene que quedar claro qué entra, qué sale y las posibles condiciones o restricciones.
3. Una vez comprendido el problema hay que tratar de determinar qué pasos o acciones tenemos que realizar para resolverlo. Esto es, diseñar la solución.
4. Si el problema es bastante complicado, lo mejor es dividirlo en partes más pequeñas, tantas veces sea necesario, para intentar resolverlas por separado.
5. Las ventajas de aplicar esto son:
 - Al dividir el problema en módulos o partes se comprende más fácilmente.
 - Es más fácil realizar modificaciones sobre un módulo en particular que en todo el algoritmo.
 - Los resultados se probarán mucho mejor comprobando si cada módulo da la salida esperada, pues si intentamos probar de golpe todo el programa y si se produce un error no sabremos en qué parte ha sido.
6. Incluso antes de adentrarse en la codificación de los programas, debe quedarnos muy clara la importancia de la creación de nuestros propios algoritmos.

Recordemos que un algoritmo es una secuencia ordenada de pasos - sin ambigüedades -, repetible, que es solución de un determinado problema. Las características fundamentales que debe cumplir todo algoritmo son:

1. Debe ser *preciso e indicar el orden de realización de cada paso*.
2. Debe estar *definido* (si se repiten 'n' veces los pasos se debe obtener siempre el mismo resultado).
3. Debe ser *finito* (debe tener un número finito de pasos).
4. Es *independiente* del lenguaje de programación que se utilice.
5. La definición de un algoritmo debe describir tres partes *Entrada, Proceso, Salida*.
6. La programación es adaptar el algoritmo al lenguaje empleado.
7. El algoritmo es independiente según donde se implemente.
8. El algoritmo trata de resolver problemas mediante programas.

Aprovecho, además, para hacer justicia al lenguaje de programación en que por la naturaleza de mis actividades profesionales he programado en mi estancia en este trabajo, ya que a pesar de que en mis tiempos de estudiante siempre se nos dijo que Cobol es un lenguaje obsoleto, e incluso se eliminó de los planes de estudio y en mi generación se sustituyó por otro 'de mayor difusión' (el profesor prefirió enseñarnos DBaseIII), no está de más señalar que, al menos en el medio de los sistemas de información bancarios y en todos aquellos que requieran manipular grandes cantidades de información comercial, Cobol es y seguirá siendo el lenguaje de programación más solicitado, tan es así que hoy por hoy siguen desarrollándose versiones actualizadas y mejoradas del mismo. En BanCrecer estamos trabajando con la versión 3.3 para OS-390.

Por último, y aún con todo lo que ya se ha dicho a lo largo del capítulo e independientemente del lenguaje utilizado, cabe hacer hincapié en que el desarrollo de programas es, principalmente, una tarea de *dedicación, esfuerzo y creatividad*.

CAPÍTULO V

LA IMPORTANCIA DE LA DOCUMENTACIÓN DE SISTEMAS

*"La verdadera sabiduría no es aquella que se tiene,
sino la que se comparte"...*

- Gertrude Stein

INTRODUCCIÓN

En el ciclo de vida de los sistemas existe otra importante etapa que aunque no se mencionó de manera explícita en las fases del ciclo de vida clásico, también es parte fundamental en todo sistema de información: la documentación.

La documentación del sistema es una actividad inmersa en cada una de las etapas de su ciclo de vida que debería ser considerada parte importante en el desarrollo y evolución del proyecto.

La documentación es esencial para proporcionar entendimiento de un sistema a quien lo vaya a usar para mantenerlo, para permitir auditoría del sistema y para enseñar a los usuarios cómo interactuar con él y a las áreas operativas cómo hacerlo funcionar.

Para ello, en este capítulo se presentan algunas características deseables en la documentación que dará el soporte necesario para la comprensión y mantenimiento de un sistema.

DOCUMENTACIÓN

¿Para qué documentar?

Existen varios tipos de documentación, y entre los más importantes están:

1. Documentación interna en los programas.
Explica la lógica de un programa e incluye descripciones, diagramas de flujo, listados de programas y otros documentos.
2. Documentación para el usuario.
Explica en forma general la naturaleza y capacidades del sistema y cómo usarlo.
3. Documentación para la alta gerencia.
Describe y justifica el desarrollo de un sistema y sus objetivos, además de las fechas y fases de desarrollo.

Muchas organizaciones tienen lo que se conoce como un "programa de documentación", el cual consiste en una política formal, cuya documentación se muestra como algo que debe prepararse en forma rutinaria para cada programa de cómputo, archivo y nuevos sistemas.

La documentación es un registro físico, normalmente por escrito, que contiene los siguientes elementos:

- Políticas y normas referentes al desarrollo del sistema, su implantación, operación y mantenimiento.
- El diseño del sistema de información administrativo.
- Procedimientos para instalar, operar y mantener el sistema de información administrativo.

La documentación adecuada y completa de una aplicación que se desea implantar, mantener y actualizar en forma satisfactoria, es esencial en cualquier sistema de información; sin embargo, frecuentemente es una actividad a la cual se le dedica el menor tiempo y no se le presta la atención adecuada.

Cuando un sistema que se ha desarrollado y ha sido entregado al área de Producción no cuenta con una documentación clara y completa, el personal responsable de su desarrollo será requerido continuamente y no

podrá moverse a otra asignación, lo cual no es recomendable, obstaculizando, incluso, su crecimiento profesional.

Aún cuando las normas de documentación varían de una instalación a otra, es esencial que dentro de una organización se utilice un solo método. El uso de procedimientos y documentación estandarizada proporciona la base de una comunicación clara y rápida, así como un adiestramiento menos costoso del personal de sistemas y reducción de costos de mantenimiento, además, asegura que el sistema opere correctamente y que se utilicen eficientemente los recursos.

En lo particular, considero que toda documentación que se relacione con un sistema, ya sea manual o por computadora, sencillo o complejo, debe reunir los siguientes requisitos básicos:

- Debe ser bien organizada, con secciones claramente indicadas
- Debe ser almacenada en carpetas e incluir un Índice
- Los diagramas deberán ser claros y no aglomerados
- Deberá ser completa
- Debe contener una leyenda o explicación de los términos utilizados
- Siempre debe ser actualizada

El estilo de redacción de los manuales de documentación debe ser concreto y preciso y debe además:

- Definir los términos utilizados
- Utilizar párrafos cortos
- Utilizar títulos y subtítulos
- No emplear frases largas que presenten hechos distintos

El área responsable debe asegurarse de que los estándares sean completos, actualizados, documentados y legibles, así como auditar permanentemente para que se cumplan los estándares. Esta área debe también evaluar si los estándares establecidos son los requeridos y, si es necesario, hacer los cambios pertinentes para que éstos sean los apropiados.

Para evitar confusiones, en las revisiones de la documentación se deben contemplar diferentes tipos de documentos dirigidos a las distintas personas que trabajarán con el sistema para facilitar el mantenimiento del mismo.

Debido a ello, se describen a continuación los documentos más comunes que deben ser realizados a la par del desarrollo del sistema.

a) Documentación administrativa

*Este manual tiene como finalidad el permitir a la alta gerencia tener la información necesaria y suficiente sobre un sistema en particular y servir como fuente de consulta una vez que el sistema ha sido implantado.

Contenido

- Nombre del sistema: Describir el nombre del sistema a implantar en la empresa.
- Equipo encargado del sistema: Nombre del personal encargado del análisis y diseño del sistema.

LA IMPORTANCIA DE LA DOCUMENTACIÓN DE SISTEMAS

- **Resumen administrativo:** Compendio de los puntos que se describen en el manual, el cual tiene como propósito permitir a los altos ejecutivos enterarse en forma somera de la propuesta del sistema. En este punto aparece por primera vez el nombre del sistema, el cual debe ser único, y deberá conservarse invariable en todos los documentos referentes a ese sistema.
- **Planteamiento:** Este punto tiene como finalidad registrar los antecedentes que servirán de partida al desarrollo del análisis del sistema. Se debe mencionar:
 - Dependencia que requirió el trabajo.
 - Personas y / o puestos ocupados por éstas al momento de requerirse el trabajo (acuerdos, disposiciones legales, memorandos, y otros).
 - Condiciones y criterios que normaron el desarrollo del trabajo.
 - Fechas correspondientes.
- **Objetivos del sistema.** Aquí se dejarán establecidos los objetivos que debe cubrir el sistema, en forma clara y precisa para evitar errores de interpretación.
- **Entradas del sistema.** Deben quedar especificados en este punto los documentos fuentes que inician las operaciones del sistema así como la información detallada de aquellos conceptos que serán los datos a captar por el mismo. Se deberán mencionar todos los datos que en forma secundaria originan una entrada importante.

Ejemplo:

Nombre del documento fuente	Módulo o procedimiento donde entra el documento	Usuarios que manejan el documento	Origen del documento
-----------------------------	---	-----------------------------------	----------------------

- **Salidas del sistema.** En este punto, solamente se describirán los resultados de mayor importancia obtenidos a través de todo el proceso. En esta sección se debe dar mayor énfasis a la información que el sistema proporciona cuidando de no hacer tan sólo mención de los resultados a obtener.

Ejemplo:

Nombre de la salida	Destino	Periodicidad en que se genera	Usuarios que lo requieren
---------------------	---------	-------------------------------	---------------------------

- **Diagramación general del sistema.** Es la representación gráfica de las fases del sistema y su flujo a través de las dependencias que intervienen en el mismo, aunque en forma generalizada. La técnica a utilizar y la simbología debe ser seleccionada por los interesados.
- **Explicaciones de las fases del sistema.** Este punto se encuentra relacionado con el anterior ya que lo que se muestra gráficamente, ahora se describe en forma genérica, explicando los procesos que se llevan a cabo en cada dependencia sin profundizar en detalles técnicos o específicos. Se deberán resaltar aquellas fases del proceso en las cuáles se obtengan resultados de importancia así como aquellas que requieran una supervisión especial.
- **Requerimientos del sistema.** Se establecen los recursos, tanto humanos como materiales que son necesarios para poder llevar a cabo el sistema. Presentar costos y descripción, además de las cantidades que se requieran.

LA IMPORTANCIA DE LA DOCUMENTACIÓN DE SISTEMAS

- Estimación de la fecha probable de implantación del sistema. Es necesario que ésta exista, pues es la base de la terminación de todas las actividades para la creación del sistema, tales como: análisis, programación, elaboración de formas, y otros.

Se recomienda utilizar diagramas de Gantt o de Pert para establecer el periodo de las actividades requeridas para el desarrollo del sistema.

Ejemplo:

Escala del tiempo en semanas

Actividades a realizar	1	2	3	4
Presentación de la Propuesta	*****			
Análisis Costo / Beneficio		*****		
Adquisición del Equipo			*****	
Entrenamiento				*****

- Consideraciones generales del nuevo sistema. En este punto se deberán señalar las ventajas, desventajas, y principales diferencias del nuevo sistema con el anterior, tales como seguridad, disminución de costo, ahorro de tiempo, flexibilidad, confiabilidad y otros.

b) Documentación para el usuario

Expone los procesos que el usuario puede realizar con el sistema implantado. Para lograr esto, es necesario que se detallen todas las características que tienen los programas y la forma de acceder e introducir información.

Este tipo de manuales permite a los usuarios conocer el detalle de las actividades que deberán desarrollar para la consecución de los objetivos del sistema. Además, reúne la información, normas y documentación necesaria para que el usuario conozca y utilice adecuadamente la aplicación desarrollada.

Los objetivos de la documentación para el usuario se engloban en los siguientes puntos:

- Conocer cómo preparar los datos de entrada.
- Aprender a obtener los resultados y los datos de salida.
- Servir como manual de aprendizaje.
- Servir como manual de referencia.
- Definir las funciones que debe realizar el usuario.
- Informar al usuario de la respuesta a cada mensaje de error.
- Identificar a los usuarios del sistema: personal que se relacionará con el sistema.
- Definir los diferentes tipo de usuarios: se presentan los diferentes tipos de usuarios que usarían el sistema. Ejemplo: usuarios directos, indirectos.
- Definir los módulos en que cada usuario participará: Se describen los módulos o procesos que se ejecutarán por cada usuario en forma narrativa breve y clara.

La importancia del manual de usuario se debe principalmente a que facilita el conocimiento de:

- Los documentos a los que se puede dar entrada por computadora.

- Los formatos de los documentos.
- Las operaciones que utiliza de entrada y salida de los datos.
- El orden del tratamiento de la computadora con los datos introducidos.
- El momento en que se debe solicitar una operación deseada.
- Los resultados de las operaciones realizadas a partir de los datos introducidos.

Al elaborar el manual de usuario hay que tener en cuenta a quién va dirigido, es decir, el manual puede ser manejado desde el director de la empresa hasta el capturista de datos. Por consiguiente, debe redactarse de forma clara y sencilla para que lo entienda cualquier tipo de usuario.

Contenido

- Diagrama general. Muestra en forma condensada el flujo general de la información y de las actividades que se realizan en el sistema, proporcionando una visión general de éste.
- Diagrama particular detallado. Presenta gráficamente todos los pasos que se efectúan dentro del departamento usuario a quien está dirigido este manual. Deben especificarse los archivos de entrada, salida, los resultados, revisiones y procesos manuales.
- Explicación genérica de las fases del sistema. En este punto se explican en forma específica y detallada todas las operaciones que aparecen en forma gráfica en el diagrama particular y se analizan cada una de las fases señalando:
 - El proceso principal que se desarrolla
 - La entrada de la información
 - La obtención de un resultado parcial
 - El envío de información a otra dependencia
- Instalación del sistema. Se deben proporcionar detalles completos sobre la forma de instalar el sistema en un ambiente particular.
- Iniciación al uso del sistema. En este punto se explica cómo iniciarse en el sistema y cómo se pueden utilizar sus cualidades comunes.

c) Documentación interna de programas.

Objetivos

El principal objetivo de la documentación de programas es facilitar los futuros mantenimientos que se darán al sistema, ya que debe dar una idea clara y una breve definición de la función de cada programa en particular.

Contenido

- Fuentes de entrada. Breve descripción de los datos que sirven de entrada para el procesamiento.
- Fuentes de salida. Breve descripción de los resultados que producirá el programa.
- Bases de datos utilizadas. Accesos y actualizaciones a bases de datos involucradas.
- Módulos y/o subrutinas empleados. Llamados hechos por el programa a otros programas.
- Diagrama de flujo. Visión general de la lógica del programa.
- Descripción de cada uno de sus párrafos. Breve descripción de la funcionalidad de cada párrafo codificado.

En BanCrecer, en la fase de Desarrollo específicamente, tenemos como parte de nuestros entregables, la documentación de los programas, tanto nuevos como modificados. Sin esta actividad completada, no podemos efectuar liberación alguna al ambiente de Producción, y existe un área de control de calidad (llamada Mesa de Control en la institución) encargada de supervisar esto.

Además de esta documentación, tanto técnica como funcional de los programas, uno de nuestros lineamientos es que a parte de que todo programa lleve una estructura estándar en su codificación, debe estar perfectamente documentado internamente. Cada párrafo de código debe ser precedido por una descripción concisa pero clara sobre su funcionalidad.

Aunada a la documentación de los programas, también se realiza una documentación estandarizada y reglamentada acerca de los jcl's¹ en que se ejecutan los programas de los procesos batch, lo cual además de indicar al área de Soporte a la Producción la correcta planificación en que deben calendarizarse los procesos, da una clara idea a los demás programadores para su futuro mantenimiento.

En cuanto a la documentación administrativa y manuales de usuario, existen manuales completos desarrollados tanto por el área de Diseño como por el área de Procedimientos, según sea el caso, lo cual asegura la correcta documentación de los sistemas en Producción, que para hacerla disponible a todas las áreas involucradas a nivel nacional, también se publica en la red interna de la institución.

COMENTARIOS

La documentación de sistemas nos indica qué hacen los sistemas, cómo lo hacen y para quién lo hacen. Generalmente muestra los requerimientos y restricciones del sistema, las características del diseño, así como las características técnicas y la operación del sistema.

Es importante recordar que la documentación de un sistema debe realizarse adecuadamente: bien organizada, actualizada y completa; todos los términos utilizados deben explicarse claramente. Además, debemos asegurar que se encuentre disponible a todos los usuarios de acuerdo a sus necesidades.

Esto es bueno siempre y cuando exista únicamente la documentación necesaria, actualizada y clara, a disposición de todo el personal que requiera tener el acceso a ella.

La importancia de la documentación bien podría ser comparada con la importancia de la existencia de una póliza de seguro; pues mientras todo va bien no existe la precaución de confirmar si nuestra póliza de seguros está o no vigente. Este es comúnmente un problema con el que nos enfrentamos a la hora de dar mantenimiento a los sistemas, pues es hasta que se requiere implantar un cambio o una mejora cuando se recurre a la documentación existente, si es, en el mejor de los casos, que ésta existe, y que se encuentre actualizada, pocas veces sucede. Esta es una mala práctica que debemos erradicar.

Una recomendación importante sobre este tema es que siempre se debe documentar un sistema como si se estuviera a punto de salir para nunca volver. Recordemos que si la documentación del sistema no está actualizada ni completa, difícilmente podremos evitar el estancamiento en un proyecto y poder movernos a otro que nos permita ganar conocimiento y experiencia.

¹ Por sus siglas en inglés Job Control Language, Lenguaje de Control de Tareas.

FALTAN

LAS

PÁGINAS

79|

A

89|

CAPÍTULO VII

LA IMPORTANCIA DE LA PLANEACIÓN EFECTIVA DE PROYECTOS

*"Ser ambiguo y dejar todo para después
son los recursos de los débiles de espíritu"...*

90

- John Updike

ESTA TESIS HA SIDO
DE LA BIBLIOTECA

INTRODUCCIÓN

Si sabemos que una de las necesidades fundamentales en todos los proyectos es que deben desarrollarse de acuerdo a fechas y tiempos estimados, entonces ¿por qué esto es siempre el principal problema en la liberación?

La experiencia ha demostrado que algunas de las principales causas de los retrasos pueden ser:

- Primero, nuestras técnicas de estimación están pobremente desarrolladas, reflejo de hacer suposiciones falsas, como pensar que 'todo saldrá bien'.
- Segundo, nuestras técnicas de estimación caen en la falacia de confundir el esfuerzo con el progreso, con la suposición de que los programadores y los meses son intercambiables.
- Aunado a la inseguridad en nuestras estimaciones, los gerentes del proyecto, en su afán de mejorar las cosas, acortan aún más los tiempos.
- El avance del proyecto es pobremente monitoreado.
- Cuando nos damos cuenta del atraso, la respuesta natural (y tradicional) es aumentar recursos-hombre.

Ciertamente, el costo de un proyecto depende del número de recursos-hombre y el número de meses (tiempo), pero el progreso no.

En el momento de las pruebas del sistema, y debido al exceso de confianza que muchas veces tenemos, suponemos que el número de errores que podrían botar durante las pruebas es menor al que sale en realidad. Ello hace que subestimemos el tiempo en esta fase del sistema.

Estos y muchos otros problemas a los que frecuentemente nos enfrentamos a la hora de poner en marcha el desarrollo de los proyectos nos lleva a replantear nuestras técnicas y estrategias de planeación, tema que será tratado en el presente capítulo.

PLANEACIÓN DE PROYECTOS

¿En qué consiste la planeación de proyectos?

Esta etapa es el punto de partida de todo sistema de información. Para iniciar cualquier actividad se requiere de una previa planeación.

Planificar hace visibles los planes de proyecto, los calendarios y los costos. Para llevar a cabo esta actividad se requiere de tres pasos:

1. Establecer las metas de los sistemas

El proceso de planeación deberá alinear sus actividades con los objetivos de la empresa, enfocando los proyectos hacia las metas estratégicas de la compañía e identificando las áreas en las que probablemente se encontrarán oportunidades con altos beneficios. El objetivo principal es establecer las metas de la

organización y enlazarlas con las metas de los sistemas. A partir de esto empiezan a surgir ideas de proyectos¹ para dar soporte a estas metas, las cuales pueden proponerse como:

- Diseñar e implementar sistemas que apoyen a las metas organizacionales.
- Aprovechar las oportunidades de negocios proporcionadas por las nuevas tecnologías informáticas.
- Seguir una metodología de desarrollo de sistemas que interactúe con los usuarios.

2. Determinar y asignar prioridades a las solicitudes de proyectos de sistemas

No todos los proyectos solicitados son deseables o factibles. Algunas organizaciones reciben tantas solicitudes de sus usuarios que sólo es posible atender unas cuantas. Cuando esto ocurre, la administración decide qué proyectos son los más importantes y el orden en que se llevarán a cabo. Muchas organizaciones desarrollan sus planes para sistemas de información con el mismo cuidado con el que planifican nuevos proyectos y programas de fabricación o la expansión de sus instalaciones.

3. Evaluar los recursos y la capacidad de los sistemas.

La finalidad de este paso es la determinación de la aceptación que tendrán los proyectos de sistemas planeados, así como afirmar que se cuenta con capacidad suficiente durante la etapa de planeación, no solo para solucionar la necesidad actual, sino para respaldarlo por condiciones de funcionamiento permanente hasta que un nuevo proyecto lo reemplace de acuerdo a nuevas necesidades estratégicas por parte de la empresa.

La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. Cuando la evaluación de sistemas se conduce en forma adecuada proporciona mucha información que puede ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones subsecuentes.

Existen proyectos cuyas actividades son muy diversas y tienen muchas dependencias, por ejemplo, creaciones o mudanzas de centros de cómputo. Éstos requieren una estricta definición de la secuencia de desarrollo. Hay otros, en cambio, en los que las actividades son pocas, aunque bastas, y tienen interdependencias muy escasas y evidentes, por ejemplo, desarrollos de software, conversiones, etc. En ambos tipos de proyecto, es importante identificar y dimensionar las actividades y sus productos de la manera más exhaustiva posible.

La gestión de un proyecto presupone establecer condiciones para el desarrollo del mismo. Involucra actividades de: planificación, estimación de recursos, seguimiento y control y evaluación del proyecto.

- La planificación de proyectos se define como la predicción de la duración de las actividades y tareas a nivel individual.
- La estimación se define como la predicción de personal, esfuerzo y costo que se requerirá para terminar todas las actividades y productos conocidos asociados con el proyecto.
- El seguimiento de proyectos es la recolección de datos y su acumulación sobre recursos consumidos, así como los costos generados asociados al mismo.
- Controlar el desarrollo implica actividades de verificación y validación, además facilita la toma de decisiones.

¹ Mucho se ha utilizado el término 'proyecto' a lo largo de este trabajo, sin embargo, aprovechando la naturaleza del tema aquí expuesto, definiremos un proyecto como un conjunto de actividades encaminadas a lograr un resultado en un período y presupuesto determinado.

- La evaluación en los proyectos es una actividad fundamental para la mejora de la productividad, el costo y la calidad del producto final.

El proceso de inicio abarca aquellas actividades de creación de la estructura del proyecto. Durante éste se define el ciclo de vida del software para el proyecto en cuestión y se establecen los planes para su gestión. Se estiman y asignan los recursos necesarios a fin de ejecutar las distintas tareas que demanda el proyecto. Se identifican y seleccionan estándares, metodologías y herramientas para la gestión y ejecución del mismo y, por último, se prepara y establece un plan para su implantación adecuada y oportuna. El plan de gestión del proyecto de software que conducirá el desarrollo se produce como culminación de este proceso.

Mediciones y estimaciones

El software al ser intangible (no tiene peso, ni volumen, ni superficie, etc) se mide a través de diversos aspectos clave en el desarrollo. Las mediciones y estimaciones atacan los tres problemas clave de la ingeniería del software:

1. Estimar costos y recursos en un proyecto de software.
2. Garantizar la calidad del producto final.
3. Mejorar la productividad del ingeniero de software durante el desarrollo.

Para estimar los recursos es necesario tener en cuenta una serie de factores de riesgo que influyen sustancialmente en la precisión de las estimaciones de los recursos humanos necesarios para la realización del proyecto. Los más importantes son:

- Complejidad de la tarea
- Modificaciones permitidas a lo largo del desarrollo
- Experiencia de los desarrolladores
- Duración del proyecto
- Estructuración del problema y de las tareas
- Disponibilidad de datos e información suministrada por el usuario
- Disponibilidad y facilidad de comunicación con el usuario

Además de las fases estándar del desarrollo, hay que tener en cuenta la coordinación y seguimiento del proyecto que suponen una importante carga de trabajo y que son olvidadas durante la planificación o no se le dedica mucho tiempo.

El costo global se compone de las partidas de viajes, hardware (nuevo o actualización), software (en caso de comprar algún paquete para el desarrollo), gastos comunes y personal, que es el más influyente, ya que el costo de un proyecto es directamente proporcional a los recursos humanos.

El proceso engloba todas las actividades y fases que se llevan a cabo durante la realización del proyecto. Se persigue determinar si en cada fase los resultados producidos se corresponden con los esperados y en establecer un control sobre los recursos estimados para cada una de las fases.

El producto incluye cualquier documento o software desarrollado que se genere durante el proceso completo. En las medidas de productos de software existen medidas directas (como es el costo del proyecto, esfuerzo empleado, líneas de código implementadas, etc.) y medidas indirectas (tales como funcionalidad, fiabilidad, eficiencia, facilidad de mantenimiento, etc.).

Sin embargo, en la práctica nos encontramos con que, en general, los proyectos consumen más tiempo del planeado y esto es causado en gran medida por la recurrente subjetividad al medir el grado de avance, pues ello impide precisar las fechas de terminación y, con ello, los costos.

Desde mi punto de vista, considero que debemos enfocarnos a dimensionar objetivamente el esfuerzo requerido para terminar un proyecto. Esto también implica medir objetivamente el grado de avance del proyecto.

También, es muy importante asignar a los recursos adecuadamente, por lo que es necesario conocer anticipadamente las actividades a realizar. Durante el proceso de desarrollo, es fundamental estar alerta del avance e identificar las desviaciones oportunamente para la toma de decisiones. Finalmente, el principal objetivo es terminar el proyecto dentro del tiempo y costo planeados, con la calidad y eficiencia esperadas.

Consideraciones para una adecuada planeación de proyectos

Podemos hacer uso de algunas herramientas que nos permitirán llevar un mejor control sobre el desarrollo de un proyecto y, por ende, una planeación estratégica y efectiva. Tales herramientas se mencionan a continuación:

- **Gráfica de Gantt:** Mediante esta gráfica podemos obtener una visión general acerca del estado del proyecto en la que destaca la fecha comprometida de terminación a nivel de tarea, fase e incluso de proyecto (ver anexo 6a).
- **Checklist:** Es una relación detallada de productos a obtener. Esta herramienta puede ser usada para facilitar la asignación de tareas, además de que podremos medir el esfuerzo estimado para obtener cada producto (ver anexo 6b).
- **Reporte de avance:** Éste debe ser un informe objetivo de logros. Mediante esta herramienta podemos determinar la eficiencia del proyecto (avance vs tiempo), así como evaluar el esfuerzo necesario para concluir el proyecto, lo cual incrementa la posibilidad de terminación oportuna (ver anexo 6c).
- **Gráfica de eficiencia:** Mediante esta gráfica podremos registrar la eficiencia observada en el proyecto e incluso registrar la observación de la tendencia. Con ella podemos visualizar que una mala planeación conduce a reportes de avance no periódicos (ver anexo 6d).
- **Gráfica de esfuerzo:** Con esta gráfica podemos visualizar el esfuerzo requerido semanalmente para concluir a tiempo el proyecto. Es, además, una guía en la toma de decisiones sobre la asignación balanceada de tareas y la cantidad de recursos necesarios (ver anexo 6e).
- **Gráfica de productividad:** La gráfica de productividad es una guía en la toma de decisiones sobre el estado del proyecto. Sirve para el seguimiento de la contribución de cada persona al logro del objetivo, además, sirve también como una guía en la medición de actividades que se realizan fuera de lo planeado.

En BanCrecer, más que una gráfica se realiza semanalmente un reporte de productividad en el que se refleja claramente el desempeño de todos los recursos involucrados en el desarrollo de proyectos o en el mantenimiento de los mismos. Esta misma información al final del mes nos da un reporte concentrado (ver anexo 6f).

LA IMPORTANCIA DE LA PLANEACIÓN EFECTIVA DE PROYECTOS

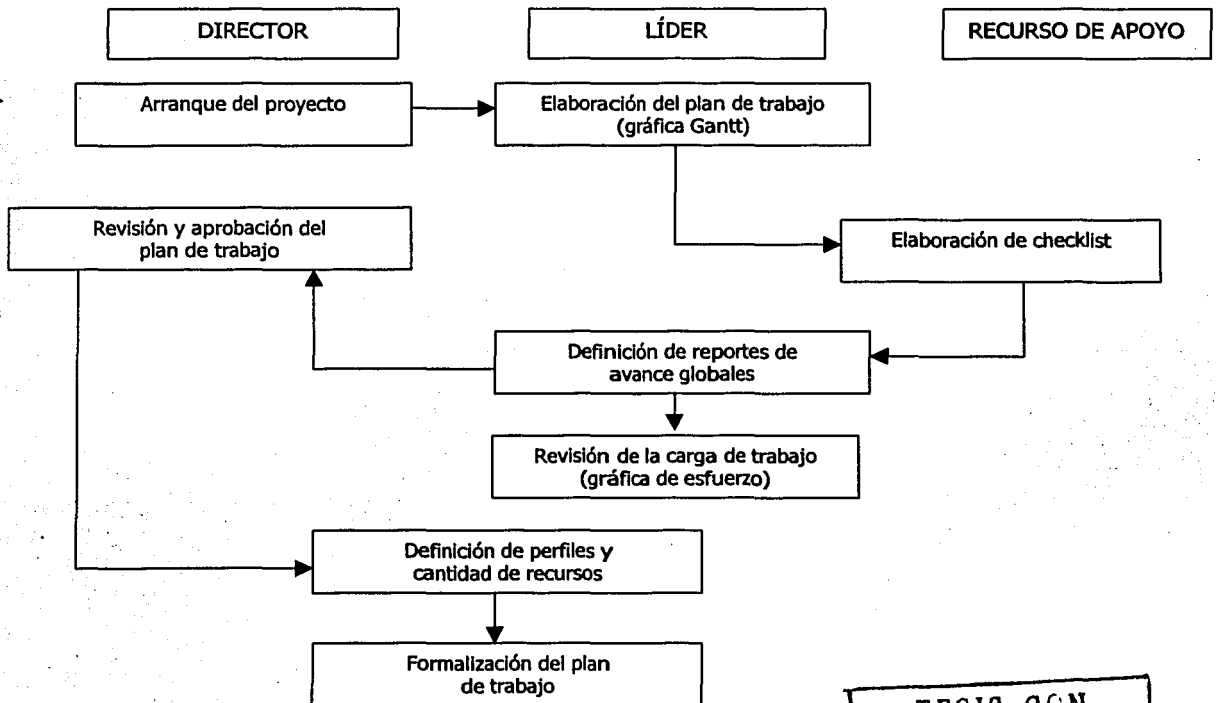
- **Reporte de asignación:** Este reporte proporciona una localización clara y rápida de cada una de las personas en los proyectos, además de que puede ayudarnos en el costeo de los mismos.

En el caso de BanCrecer, el reporte que se utiliza para este propósito es prácticamente el mismo empleado para reportar la productividad, salvo que este concentra a todos los recursos de la Dirección de Desarrollo que obviamente se encuentran asignados a distintos proyectos en cada subdirección (ver anexo 6g).

- **Reporte de liberación:** Este reporte es una guía en la toma de decisiones sobre la fecha de liberación de los recursos, el perfil del recurso próximo a liberarse y evaluar el reforzamiento de proyectos críticos o compromisos para futuros proyectos (ver anexo 6h).
- **Relación de recursos:** Además de las gráficas y reportes mencionados, en BanCrecer se revisa semanalmente la lista de recursos disponibles por cada subdirección y por cada proyecto. Es una proyección sobre la disponibilidad de los recursos requeridos de acuerdo a las necesidades (ver anexo 6i).

Los reportes aquí presentados son sugerencias para llevar un adecuado control de proyectos, sin embargo cada empresa o grupo de trabajo es libre de decidir y de considerar los que mejor se adapten a sus necesidades o, en todo caso, tomar la base de ellos y desarrollar los suyos propios.

Mecánica de aplicación para una efectiva planeación



LA IMPORTANCIA DE LA PLANEACIÓN EFECTIVA DE PROYECTOS

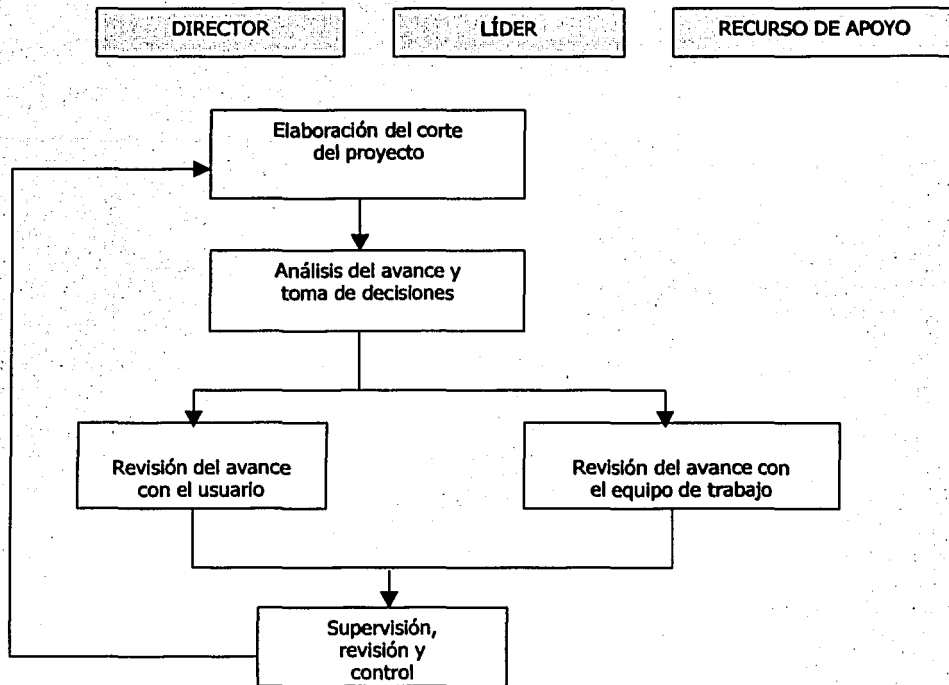
La experiencia aconseja que como parte de una adecuada planeación, se deben observar invariablemente las siguientes actividades:

- Arranque del proyecto.
- Definir uno o varios objetivos.
- Definir criterios de aceptación.
- Identificar factores críticos de éxito.
- Revisar los alcances del proyecto.
- Definir la organización y responsabilidades del equipo de trabajo.

Además, resulta necesario apoyarse con las siguientes herramientas:

- Elaboración del plan de trabajo:
 - Identificar fases críticas o importantes e incluirlas en el Gantt.
 - Definir puntos de control o de revisión.
 - Planear pensando en resultados tangibles (entregables) en plazos cortos.
 - No mezclar diferentes tipos de actividades dentro de la misma fase.
 - El nombre de la fase debe ser un fiel reflejo de los productos que contiene.
- Elaboración de los checklist (productos):
 - Realizar al menos un checklist por fase.
 - Incorporarse solo productos relacionados con la fase.
 - Incorporar productos que se puedan elaborar dentro del tiempo planeado.
 - Estimar el tiempo para la elaboración de cada producto en base a una persona de capacidad y productividad estándar.
 - Considerar el ambiente, lenguajes y herramientas para la estimación del tiempo.
- Definición de los reportes de avance globales:
 - Estos reportes deben hacerse de acuerdo a compañías, áreas o departamentos, equipos de trabajo, actividades o subproyectos, según sea el caso.
 - Considerar estos reportes para la estimación del esfuerzo en recursos, perfil o fechas.
- Revisión de las cargas de trabajo:
 - Identificar incrementos fuertes en el número de recursos requeridos por cada período, lo cual conlleva a ajustar las fechas de inicio y terminación de las fases.
- Revisión y aprobación del plan de trabajo:
 - Conseguir la aprobación del director.
 - Obtener la aprobación del área encargada del control de proyectos.
- Definición de perfiles y cantidad de recursos:
 - Esta definición se debe hacer de acuerdo a las horas agrupadas por fase, actividad o especialidad y de acuerdo a las horas productivas.
 - Existe una cantidad máxima de recursos a asignar y, por lo tanto, una duración mínima.
- Formalización del plan de trabajo:
 - Presentarlo al usuario.
 - Presentarlo al equipo de trabajo.

LA IMPORTANCIA DE LA PLANEACIÓN EFECTIVA DE PROYECTOS



- **Elaboración del corte del proyecto:**
 - Mantener actualizados los planes.
 - Revisar los tiempos reales vs los tiempos planeados.
 - Reportar los productos como terminados de acuerdo a un criterio binario.
- **Análisis del avance y toma de decisiones:**
 - Revisar el avance de cada fase y el avance global del proyecto.
 - Replanear, en caso de retrasos inalcanzables.
 - En los reportes de avance revisar la eficiencia, horas esperadas, horas reales, horas esperadas en el siguiente período y la fecha estimada de terminación.
 - Evaluar el desempeño del equipo de trabajo.
 - Definir los objetivos del siguiente período.
- **Revisión del alcance con el usuario:**
 - Revisar el proyecto basándose en el Gantt preparado.
 - Es muy importante mantener informado al usuario sobre la situación actual del proyecto (avance, logros, probables problemas).
 - Revisar y documentar los cambios al proyecto.
 - Documentar los acuerdos y acciones concluidas (elaborar minutas).
- **Revisión del avance con el equipo de trabajo:**

LA IMPORTANCIA DE LA PLANEACIÓN EFECTIVA DE PROYECTOS

- Revisar las gráficas de Gantt, de eficiencia y de productividad.
 - Revisar los informes sobre la situación actual del proyecto.
 - Revisar la terminación de productos por cada recurso de acuerdo a la asignación del período anterior preguntando ¿qué está terminado? si no ¿cuándo estará?, si está atrasado ¿cuáles son las razones?
 - Asignar nuevos productos a cada recurso de acuerdo al objetivo del período.
 - Documentar compromisos.
- Supervisión, revisión y control:
(Comentarios dirigidos especialmente a líderes de proyecto o a quienes tengan gente a su cargo)
- Controlar el proyecto basándose siempre en un esquema 'supervisión - revisión - control'.
 - Utilizar los planes como herramienta fundamental para coordinar y controlar.
 - El instrumento básico de control es la revisión.
 - La distribución eficaz de las actividades del proyecto depende en gran medida del líder (su experiencia en el área, en la aplicación, etc).
 - Las revisiones son muy importantes porque permiten:
 - Detectar las desviaciones
 - Evitar repetición de errores
 - Detectar omisiones en el plan
 - Corregir estimados deficientes
 - Detectar problemas potenciales
 - Minimizar los malos entendidos (revisiones intermedias)
 - Crear un ambiente de confianza
 - Mantener y/o mejorar la comunicación

COMENTARIOS

Recordemos que si hay un proyecto que tiene un año de retraso no se atrasó de golpe, sino que lo hizo de un día tras otro, tras otro... tras otro.

En BanCrecer no siempre los proyectos se han terminado en el tiempo inicialmente estimado y muchos de aquellos que lo han logrado no siempre se liberaron con la calidad requerida. Además las cargas de trabajo han sido totalmente desbalanceadas no solo entre áreas sino dentro de ellas, pues incluso dentro del área de Desarrollo, las cargas de trabajo no son lo suficientemente equilibradas.

Uno de los mayores problemas que yo he observado ha sido que aún cuando se cuenta con una metodología para llevar a cabo una planeación estratégica, ésta no se ha seguido o muchas veces la planeación se hace de manera poco efectiva y pensada, lo cual lleva a proyectos pésimamente desarrollados.

Aunque finalmente se cumpla con los requerimientos de las áreas usuarias, no siempre se culminan de la manera más óptima y recomendable, cosa que conlleva a un mantenimiento correctivo más que preventivo del producto entregado.

Otro factor determinante en los retrasos ha sido siempre la presión de la alta gerencia, pues cuando un proyecto se turna al área de Sistemas, éste ya fue comprometido en fechas con las áreas usuarias.

LA IMPORTANCIA DE LA PLANEACIÓN EFECTIVA DE PROYECTOS

La rápida y pobremente pensada calendarización de actividades, así como la falta de conocimiento y/o experiencia de algunos analistas encargados de la investigación preliminar y el diseño lógico, ocasiona que se planifiquen mal tanto las actividades a realizar como el número de recursos que deberán ser involucrados.

En mi entorno, se ha observado que el área de Diseño en general consume la mayor parte del tiempo presupuestado para el proyecto y, peor aún, el producto de ese trabajo pocas veces es el adecuado para llevar a cabo un óptimo desarrollo de software. Además, en las etapas de análisis y diseño nunca se involucra a todas las áreas necesarias en un nuevo proyecto, sino hasta el final del ciclo y a marchas forzadas.

Lo que es evidente es que no se trata de una competencia entre áreas por los tiempos y créditos del proyecto, sino que el desarrollo de un sistema debe ser revisado por todos los involucrados desde sus fases iniciales para así llevar un control estricto y observar que se cumplan cada una de las fases de la metodología siguiendo una planeación adecuada. Alguien mencionó alguna vez que: "...la competencia genera violencia y la cooperación genera desarrollo..." Así pues, lo que nosotros necesitamos entre áreas es cooperar, solo así lograremos terminar un producto de calidad para satisfacer las necesidades de nuestros clientes usuarios, que es finalmente nuestro objetivo.

CAPÍTULO VIII

LA IMPORTANCIA DE LOS RECURSOS HUMANOS

*"Una máquina puede hacer el trabajo de cien hombres ordinarios,
pero ninguna puede hacer el trabajo de un hombre extraordinario"...*

- Elbert Hubbard

INTRODUCCIÓN

Al tema del personal no se le otorga a menudo la suficiente importancia, siendo éste el activo más importante de toda organización. Este capítulo se dedica a revelar algunos aspectos de los recursos humanos que generalmente son pasados por alto y sin los cuales no podrá decirse que el desempeño de una empresa es el óptimo en todos sus niveles.

Muchas veces, principalmente en el área de Sistemas, el factor humano es uno de los más castigados ya que aunque se le puede considerar como una de las carreras mejores pagadas en nuestro país, también es una profesión que requiere de mucho sacrificio, pues las jornadas de trabajo en no contadas ocasiones son agotadoras y derivan en el sacrificio del tiempo personal. Esto puede llegar a ser causa del rompimiento de sanas relaciones laborales e interpersonales, ocasionando desmotivación en la gente, entre otras cosas.

A propósito, un factor importante en el ámbito laboral es el grado de compromiso y motivación que pueda tener el personal de una institución, y a su vez, el interés que ésta muestra por su gente. Para tener una idea de ello, podría hacerse un sondeo formulando a cada empleado la siguiente pregunta:

"... En su trabajo, ¿tiene usted la oportunidad de hacer cada día lo que mejor sabe hacer? ..."¹

Evidentemente, mientras mayor sea la tendencia a responder afirmativamente esta pregunta, mejor será el desempeño de las organizaciones cuyos empleados sienten que están utilizando sus fortalezas todos los días, consiguiendo empresas sólidas y poderosas. Esto debido a un aumento considerable en la productividad y consecuentemente, la obtención de la lealtad y retención de los clientes.

Frecuentemente, y tal vez por las cargas de trabajo, al personal pocas veces se le otorga el mérito y la atención que por su importancia merece, y esto no es más que la señal de que algo en la empresa no está bien, pues en su desesperada carrera por lograr la terminación de sus productos, sacrifican al recurso humano, lo cual no solo va en detrimento de su propia imagen y calidad en el trabajo, sino también del producto final.

Los aspectos clave en un sistema y su capacidad de operación están representados por su personal y su tecnología. Depende de cómo se organicen y se conjuguen estos factores para lograr el éxito de los sistemas implantados.

Este capítulo está dedicado al recurso humano, conscientes de que el desarrollo del potencial de cada individuo está determinado por las fortalezas que posee, así que "...los líderes y directivos deben entender y aprovechar este recurso básico del nuevo orden económico mundial y convertirse en gerentes excepcionales del capital humano..."

RECURSOS HUMANOS EN BANCRECER

En el área de Sistemas BanCrecer, la organización de los recursos humanos se basa en la separación de equipos, personal y funciones por proyectos, cada uno de ellos dirigidos a un área de la Dirección o a una necesidad concreta. Cada proyecto agrupa a un determinado número de especialidades, coordinados por un líder de proyectos.

¹ Basada en el libro de M. Buckingham y D. Clifton referido en la bibliografía. Para mayor información se recomienda consultar el texto completo.

Tratar de definir cada uno de los puestos de trabajo resulta complicado debido a la gran variedad que se puede presentar de los mismos. Sin embargo, es vital para el buen funcionamiento de una organización, y en este caso específico, de un centro de cómputo, el definir lo más claramente posible las características de cada puesto de trabajo, indicando cuáles son las funciones, tareas técnicas a realizar, tareas administrativas, las dependencias y puestos dependientes.

Como ejemplo de la complejidad de la definición, se muestra el problema existente entre análisis, programación y diseño. Es evidente la interacción entre estas tres funciones, las cuales se derivan una de otra y están en estrecha relación hasta tal punto que se puede dudar muchas veces sobre las aptitudes y conocimientos que se requieren de sus titulares. Por ello, la composición de un departamento y la definición de funciones o descripción de puestos de trabajo debe ser tal que, además de establecer claramente los cometidos de sus integrantes, no provoque solapamientos ni deje responsabilidades por cubrir.

Sabemos que la adquisición de los conocimientos es una cuestión de enseñanza y de aprendizaje, y que las aptitudes son cualidades personales que los conocimientos desarrollan, refuerzan, afinan y/o afirman, pero jamás crean.

En un recuento global, puede decirse que las aptitudes en los empleos informáticos² se dividen en cuatro categorías, en las que fundamentalmente se distinguen las cualidades intelectuales, de carácter, de competencia y sociales, y a estas categorías viene a unirse la formación académica.

Mediante la adecuada formación especializada es posible incrementar o desarrollar estas cualidades, obteniendo como resultado el informático que la sociedad precisa. Pero cada individuo posee conocimientos y habilidades que pueden y deben ser aprovechados al máximo, contribuyendo a su desarrollo personal y, sobre todo, profesional, lo cual obviamente repercutirá en el mejor desempeño de la empresa. Una gran organización no sólo debe adaptarse al hecho de que cada empleado es diferente, sino que debe aprovechar esas diferencias. Debe estar atenta para reconocer las señales que hablan de los talentos naturales de cada empleado y así ubicar y desarrollar a cada persona de modo que sus talentos se transformen en fortalezas auténticas.

Al cambiar la forma de seleccionar, evaluar, desarrollar y canalizar las carreras de su gente, una organización debe desarrollar la totalidad de su empresa alrededor de las fortalezas de cada persona. De esta manera, la organización estará en condición de mostrar un desempeño muy superior al de sus similares.

Características de los puestos de trabajo típicos en BanCrecer

Se definen a continuación algunos puestos de trabajo típicos que se visualizan de manera general en la organización de personal en el área de Desarrollo de Sistemas en BanCrecer, con el único fin de detallar la forma en que en esta empresa se tienen establecidas las tareas básicas, responsabilidades y dependencias de cada puesto de trabajo.

Con todo, hay que señalar que sólo es un ejemplo de la estructuración del perfil del puesto de trabajo en esta área específica; por lo tanto, no deben tomarse estos ejemplos como definiciones estrictas.

² Se puede admitir que es informático aquel que, por la naturaleza de sus tareas, debe, para ejecutarlas, tener un cierto conocimiento de las posibilidades de los sistemas informáticos, entendiendo que el nivel, el grado, la amplitud de estos conocimientos puede, entre los extremos, variar según la función ejercida.

Director de desarrollo

Funciones:

- Dirigir, planificar y controlar todas las actividades de proceso de datos de las que es responsable ante la Dirección Ejecutiva.

Labores técnicas:

- Proponer objetivos y políticas de trabajo.
- Preparar el programa de trabajos a desarrollar.
- Estudios del hardware y software necesarios, así como la posible subcontratación de servicios.
- Actualizar la estructura organizativa cuando sea necesario.
- Especificar la normalización para el desarrollo y documentación del trabajo.
- Desarrollar normas de procedimiento, tanto internas como para relaciones con los usuarios.

Labores administrativas:

- Preparar informes periódicos o especiales sobre su departamento.
- Presidir las reuniones principales con los usuarios.

Dependencia:

- De la dirección general de la empresa.

Puestos dependientes:

- Subdirectores de área.
- Especialistas y staff.

Relaciones fuera del Departamento:

- Directores.
- Otras unidades de servicios de la empresa.
- Suministros de equipos, software, hardware y servicios.
- Otras empresas de su sector con problemática similar.

Cualidades requeridas:

- Las cualidades esenciales son las requeridas a un nivel de dirección elevado, comprendiendo la posibilidad de «vender sus ideas» a otros, especialmente a la alta dirección.

Origen y formación básica:

- Salvo excepción, la formación básica debe ser idealmente un licenciado o ingeniero informático y, en todo caso, corresponder a la de una escuela técnica superior o licenciatura y comprender una etapa de preparación en los negocios.
- Es indispensable que el director de informática haya tenido la ocasión de dirigir o de trabajar en diferentes servicios de la empresa o en una empresa similar, a fin de que conozca perfectamente las necesidades.
- Es deseable que haya adquirido experiencia en el proceso de datos a un nivel de responsabilidad, por ejemplo, como responsable de los estudios. Si no está experimentado, su subordinado inmediato inferior lo deberá estar obligatoriamente.

Promoción profesional:

- Los puestos de dirección de la empresa.

Subdirector de desarrollo

Funciones:

- Dirigir, planificar y controlar todas las actividades de proceso de datos de las que es responsable ante la Dirección de Desarrollo.

Labores técnicas:

- Vigilar que se cumplan los objetivos y políticas de trabajo propuestos por la Dirección.
- Revisar el programa de trabajos a desarrollar.

Labores administrativas:

- Preparar informes periódicos o especiales sobre su departamento.
- Asistir a las reuniones principales con los usuarios.

Dependencia:

- De la dirección del área.

Puestos dependientes:

- Líderes de proyectos.
- Staff.

Relaciones fuera del Departamento:

- Subdirectores.
- Directores.
- Areas de soporte técnico.

Cualidades requeridas:

- Cualidades esenciales de manejo de personal y sólidos conocimientos de la operativa de su Departamento.

Origen y formación básica:

- Salvo excepción, la formación básica debe ser idealmente un licenciado o ingeniero informático. Es indispensable que haya adquirido experiencia en el proceso de datos a un nivel de responsabilidad.

Promoción profesional:

- Los puestos de dirección de la empresa.

Líder de proyectos de Desarrollo

Funciones:

- Diseño y análisis de sistemas a desarrollar.
- Apoyo a usuarios durante la implantación.

Labores técnicas:

- Estudiar y analizar el sistema existente.
- Efectuar el diseño detallado y preparar las especificaciones.
- Analizar y evaluar los cambios pedidos a una aplicación.
- Realizar trabajos de apoyo a la implantación e impulsar su aceptación por el usuario.

Labores administrativas:

- Proporcionar datos sobre el progreso de la aplicación.

- Preparar datos sobre su actividad personal.
- Presentar informes de avance en los proyectos y en relación a su personal.

Dependencia:

- Del subdirector de desarrollo o del subdirector de proyecto.

Puestos dependientes:

- Analistas y programadores que participen en su aplicación.

Relaciones fuera del Departamento:

- Usuarios.
- Áreas de Soporte Técnico.
- Áreas de Soporte a la Producción.
- Áreas de Diseño e Implantación.
- Áreas de Mantenimiento.

Relaciones en el Departamento:

- Subdirector.
- Analista de aplicaciones.
- Programadores.

Cualidades requeridas:

- Poseer una mente analítica, es decir, capacidad de investigar y encontrar todos los puntos constituyentes de una operación.
- Ser muy detallista.
- Ser muy preciso y ordenado.
- Poseer don de mando.
- Conocimientos prácticos de programación.
- Es deseable que posea un conocimiento teórico del trabajo, pero no es necesario que haya trabajado en el servicio.
- Habilidades de manejo de personal.

Origen y formación básica:

- Debe ser idealmente un licenciado o ingeniero informático.

Promoción profesional:

- Los puestos de subdirección de la empresa.

Analistas y programadores

Funciones:

- Asesorar a otros programadores en la utilización del software específico.
- Colaborar en la definición de las necesidades de equipo y software.

Labores técnicas:

- Desarrollar y mantener los programas para dar soporte al negocio.
- Preparar la documentación.
- Llevar a cabo las pruebas unitarias.
- Monitoreo y seguimiento a los procesos hasta su puesta en producción.

LA IMPORTANCIA DE LOS RECURSOS HUMANOS

Dependencia:

- Del líder de proyecto.

Relaciones fuera del Departamento:

- Diseño e Implantación.

Relaciones en el Departamento:

- Líder de proyecto.
- Otros analistas de sistemas y de aplicaciones.
- Programadores.

Cualidades requeridas:

- Lógica.
- Sentido del detalle.
- Sentido de lo esencial.
- Comprensión de lectura.
- Redacción.
- Trabajo en equipo.

Origen y formación básica:

- Debe ser idealmente un licenciado, ingeniero o técnico en informática y/o carreras afines.

Promoción profesional:

- Los puestos de líder de proyectos.

Por otro lado, se ha observado que cuando se forma un nuevo equipo de trabajo muchas veces tendemos, si esto es posible, a proveerlo excesivamente de personal, y también se han dado casos en que no siempre tenemos los puestos necesarios cubiertos y que únicamente se preparan cuando la necesidad se vuelve evidente.

Es verdad que un sistema informático exige un equipo básico determinado, pero la implantación de un sistema impone cambios sucesivos cuya evolución depende en gran medida de las necesidades y de los medios financieros de las empresas y, por otra parte, de las adaptaciones progresivas, que sólo el tiempo permite cumplir razonablemente.

Aunque se ha expuesto un modelo estándar de actividades diversas en el área de Desarrollo de BanCreceer, es evidente que cada puesto de trabajo debe adaptarse tanto a la empresa como al material de las aplicaciones. Esto es lo que explica que en toda colectividad ninguna cualidad de desempeño requerida está inmovilizada en su definición.

Personal interno y externo

Cuando se habla de incorporar algún esquema de outsourcing o de personal propio de la empresa se deben analizar diversos factores. El reclutamiento del personal del servicio informático depende estrictamente de la política practicada por la empresa. Es, en efecto, posible efectuar un reclutamiento interno o preferir un reclutamiento externo. En muchos casos una política intermedia hace escoger las dos formas.

En el caso de BanCreceer, se ha optado por dicha política intermedia, dejando para las áreas que tienen que ver directamente con la producción y el soporte técnico de las aplicaciones, exclusivamente el reclutamiento de personal interno. En el área de Desarrollo, conviven en el trabajo diario y en el análisis de los sistemas los

equipos híbridos, compuestos tanto de personal interno como de personal externo, éste último integrado por personal no solo de una sino de distintas consultorías. Siempre asegurando que los puestos de líderes de proyectos sean reservados a personal interno, pues según los niveles de escalamiento profesional, éstos serán los que puedan ocupar algún puesto de nivel superior dentro de la empresa.

Debido a la imagen reflejada por otras empresas, se observa que si el reclutamiento interno ofrece, por un lado, la ventaja de proveer personas que conocen bien la empresa, por otro lado puede también presentar el inconveniente de reclutar personas que, por las rutinas adquiridas, no estén en condiciones de elaborar y de llevar a cabo las reorganizaciones que impone el uso de un sistema informático. Ello contribuye a una resistencia al cambio y/o innovación del sistema implantado, lo cual muchas veces lo vuelve obsoleto.

Además, la elección por reclutamiento interno puede dejarse influenciado por factores personales que dependen a la vez del contexto y las circunstancias. En mi experiencia, he observado que por estas razones la contratación de analistas y programadores en BanCrecer ha sido totalmente mixta, para lograr la constitución de equipos pluridisciplinarios y buscando conseguir una buena dosificación.

Por otra parte, habrá que buscar el tiempo necesario para tener en cuenta las personalidades de cada miembro del equipo para realizar una integración coherente y, como consecuencia, duradera.

Cualquiera que sea la forma de reclutamiento, interno o externo, éste debe ser cuidadosamente preparado. La oferta de empleo debe definir con precisión los puestos a proveer y los perfiles de los candidatos a cada función.

Esta nota debe prever las cualidades requeridas para ser candidato, así como la titulación o competencias exigidas. Es evidente que en un reclutamiento interno y en función de los puestos a prever, la elección puede ser más o menos restrictiva según los empleos. Muy a menudo, por otra parte, esa elección propone a tal o a cual miembro del personal ocupar el empleo a proveer. Este procedimiento debe evitarse, ya que puede conducir a parcialidades.

Finalmente, se considera que al momento de elegir al personal necesario para un departamento se deben realizar los siguientes pasos:

- Test de aptitud.
- Entrevista personal.
- Entrevista con el responsable del servicio informático.

Los test de aptitudes técnicas, en particular, no son ni una confesión ni un atentado a la libertad, sino un medio de elección. Son el reflejo tanto de la situación intelectual y psicológica como de las posibilidades eventuales del candidato.

Como todo proceso en el cual los recursos humanos juegan un papel primordial, será vital para el buen desarrollo de un proyecto que *el personal involucrado esté convencido y crea en los procesos de calidad y mejora continua.*

Cuando las empresas consideran al recurso humano como su factor más importante, los individuos ganan porque aquéllas deciden invertir en ellos de varias maneras, invitándolos a prepararse continuamente para lograr ser capaces de dar más en su trabajo y a su vez ganar más por el mismo.

A su vez, si el personal está motivado, las corporaciones se benefician porque al concentrarse en el desarrollo y calidad de sus empleados, sus costos disminuyen significativamente, mientras que aumentan notablemente sus niveles de productividad, crecimiento y rentabilidad.

COMENTARIOS

Si una empresa ya ha contratado al personal que requiere numérica y profesionalmente hablando, es también tema importante saber qué tanto se valoran unos a otros, es decir, cómo valora, motiva e invierte la empresa en su personal y cómo el personal valora y desempeña su trabajo.

Por parte de la empresa es importante que se estimule el trabajo en equipo, la comunicación, la capacitación, el reconocimiento profesional y no solamente la remuneración salarial, ya que si bien es cierto que todos buscamos una compensación económica, también es cierto que existen otros factores, específicamente las relaciones humanas, que si no son tomados en cuenta y hábilmente desarrollados, son un factor determinante que demerita el desempeño del personal y, por lo tanto, de la empresa.

Pero no todo es responsabilidad de la empresa. Nosotros como colaboradores e integrantes de una organización y de un equipo de trabajo también tenemos obligaciones que cumplir, no solo por el pago que estamos recibiendo, sino por nuestros propios valores éticos y profesionales, que son los que hablan de nosotros como egresados de una Universidad y como ciudadanos de un país.

¿Qué tan profundamente hemos reflexionado con respecto al hecho de saber en qué estamos colaborando con nuestra empresa y con nuestro país? Tal vez la respuesta sea muy sencilla: *trabajar*.

Trabajar nos da diariamente la oportunidad de ser mejores cada día. El resultado de este esfuerzo es una persona más completa, madura y segura de sí misma. Más que el simple quehacer de cada día, mediante el trabajo diario vamos conformando nuestro futuro. Debemos trabajar con responsabilidad y optimismo, esto es una exigencia fundamental. El destino de nuestro país está en manos de todos nosotros y de cada uno de los habitantes quienes desde su lugar de trabajo participan en el fortalecimiento de la sociedad entera.

Además, debemos ser capaces de reconocer y administrar óptimamente ese poderoso factor que es el tiempo.

¿Por qué se menciona esto? Porque resulta que quienes trabajamos en Sistemas frecuentemente nos quejamos de las cargas de trabajo tan intensas, de los proyectos mal organizados, de la escasa vida social que nos queda. La llave mágica del manejo del tiempo radica en dos simples palabras: ser eficaz y ser eficiente. Entendiendo por eficacia el dedicarse a la tarea adecuada y por eficiencia el realizarla correctamente.

Hoy más que nunca vale la pena pensar por un momento en las herramientas con las que contamos para hacer eficientemente nuestro trabajo: escuela, capacitación, computadoras, redes, libros, manuales, etc, a la vez de meditar que todo esto no serviría de mucho sin el empuje que día a día le demos todos a nuestro desarrollo personal y profesional.

CONCLUSIONES

*"La habilidad es lo que permite hacer ciertas cosas,
la motivación determina lo que se hace,
la actitud determina cuán bien se hace"...*

- Lou Holz

CONCLUSIONES

Este trabajo es resultado de algunos años de trabajo y desempeño en el área de Desarrollo. Es el resultado de una serie de eventos vividos a lo largo de mi carrera profesional. Y como resultado de esos años, surge ahora este testimonio escrito de mis propias vivencias en el campo laboral.

Rápidamente han transcurrido los años y ha sido tan de prisa que muchas veces no nos percatamos de todas las cosas que han sucedido a nuestro alrededor, sin embargo suceden día con día, ya sea con nuevos productos de software o de hardware que aparecen en el mercado, una nueva computadora en nuestras oficinas, o incluso en la forma de comunicarnos con nuestros compañeros de trabajo.

Estos cambios sin duda trascienden en nuestra forma de pensar y actuar, a la vez que nos preparan para el enorme reto que representa salir de la universidad y ejercer nuestra carrera profesional. Y es precisamente en este último punto donde debemos abundar acerca de la importancia de trabajar por una misma meta: nuestra consolidación como profesionistas, ya que si seguimos preparándonos como hasta ahora lo hemos hecho, sin duda alguna continuaremos contribuyendo a mejorar el nivel académico de nuestra escuela y formando nuevos y mejores profesionistas durante mucho tiempo más, y esto no será por obra de la casualidad, sino por el esfuerzo que imprimamos a nuestro quehacer cotidiano.

Quiero finalizar esta recopilación de memorias de desempeño profesional con una serie de breves comentarios en los que la principal finalidad es, en algunos casos, resumir ciertos puntos que considero importantes de resaltar, en otros casos, se refieren a aquellos temas en los que se desea puntualizar específicamente algún aspecto importante, pero, sobre todo, quiero finalizar con comentarios respecto a cómo percibo la situación actualmente en BanCreceer, la empresa para la cual trabajo, ahora ya perteneciente a Banorte, y, desde mi punto de vista, cuáles son las cosas que yo considero que se pueden hacer de manera diferente para mejorar la productividad y contribuir al desempeño de un trabajo y productos de calidad.

Los aspectos que yo cambiaría obedecen a la observación de los hechos que, en mi humilde opinión, son los que no nos han dejado romper con paradigmas que obstaculizan el logro de nuestros objetivos primordiales, los cuales tienen que ver con la calidad en nuestro trabajo y con el éxito que deseamos lograr al finalizar y liberar un sistema.

Si ya de por sí es gratificante el poner a disposición del usuario un sistema que es utilizado todos los días a nivel nacional, que repercute en la captación de clientes y en el bienestar de la empresa, cuánto más gratificante sería saber que quienes contribuimos a su desarrollo hemos trabajado arduamente en un ambiente de armonía, en equipo tanto con nuestros superiores como con nuestros compañeros, y en un ambiente regido por estándares no solo de desempeño sino también de calidad, en el que todos los días estamos aprendiendo y aportando algo nuevo, con lo cual seguimos dando lo mejor de nosotros mismos para bien del individuo, de la empresa y del país.

A pesar de la moderna tecnología, los avances en la ingeniería de software, la preparación del personal, ¿por qué seguimos enfrentando esos problemas en BanCreceer?

FACTORES QUE INTERVIENEN EN EL ÉXITO DEL DESARROLLO DE SISTEMAS

A través de cada capítulo de este trabajo hemos visto que la observación le enseña al analista:

Lo que debe suceder...	Lo que realmente ocurre...
<ul style="list-style-type: none">• Procedimientos normales de operación• Controles y verificación de calidad• Documentos llenados en forma apropiada• Terminación del trabajo en forma eficiente y a tiempo	<ul style="list-style-type: none">• Retraso al hacer el trabajo• Información que se recuerda de memoria (incorrectamente)• Etapas omitidas• Fotocopias innecesarias• Necesidad de nuevos controles• Información que no se encuentra en el archivo• Documentos mal llenados• Los empleados no están conscientes de los procedimientos prescritos

Mucho tiene que ver la participación de la gente involucrada. Es indispensable que involucremos a todo el personal a participar en los cambios y mejora del trabajo, pues todos debemos estar conscientes y convencidos de que es necesario un cambio no solo de costumbres sino de actitud para conseguir un equipo de trabajo digno de confianza y de respeto.

La participación de la gente involucrada minimiza las dificultades propias de cualquier adaptación, facilita el trabajo diario a partir de una adecuada capacitación, permite crecer con el sistema, y aprovechar sus potencialidades. Además, la participación de todos los involucrados garantiza el éxito del proyecto y nos ayuda a crecer como profesionales y como grupo.

En esta Empresa hemos visto que los problemas derivados de una implantación deficiente pueden ser de diversa índole, afectando todos ellos a la productividad y la eficacia del trabajo realizado, no solamente de un área determinada, sino de todo el equipo de trabajo.

Entre los efectos derivados de una implantación inadecuada de los nuevos sistemas o de las nuevas modificaciones, pueden destacarse:

- Baja del rendimiento y la productividad durante un largo período posterior a la puesta en marcha de los nuevos sistemas.
- Deficiente utilización de los sistemas desarrollados, con el consiguiente desaprovechamiento de las inversiones realizadas en tecnología de la información.
- Pérdida de productividad por repetición de tareas, corrección de errores y necesidad excesiva de trabajo en común por falta de conocimientos individuales.
- Necesidad de dedicar un mayor esfuerzo a tareas de control y seguimiento que aseguren la correcta realización del trabajo.

Además, una implantación deficiente no solo afecta a las áreas de Sistemas, sino que afecta, principal y directamente, al usuario de los sistemas.

Como se ha comentado a lo largo del trabajo, frecuentemente el usuario no es involucrado adecuadamente en el desarrollo de un sistema, *su sistema*. Una buena parte de la resistencia, activa o pasiva, que los usuarios oponen normalmente a cualquier cambio en su forma de trabajar se puede evitar con una correcta estrategia de comunicación que contemple:

- Identificar las expectativas de los usuarios ante el cambio tecnológico a través de cuestionarios, entrevistas o reuniones de grupo.
- Planificar acciones de comunicación que clarifiquen y respondan a las expectativas identificadas: video, charlas, cartas, folletos, artículos, etc.
- Identificación de la población objetiva afectada por la implantación de sistemas.
- Planificación preliminar de la impartición de la formación para cada uno de los sistemas, teniendo en cuenta el calendario de implantación y la estrategia de formación seleccionada para cada caso.

Esto no ha sido posible particularmente en BanCrecer porque siempre se está trabajando en tiempos muy cortos, debido no solo a una mala planeación, sino a un mal análisis de los requerimientos. Ya que el desarrollo de un sistema es el trabajo de todo un equipo, quienes no se encuentran inmersos en su campo de trabajo influyen negativamente en la terminación del producto, afectando, por lo tanto, a todas las demás fases del ciclo de vida del sistema.

En el área de Desarrollo, la cual es mi área de desempeño, también tenemos fallas que a su vez repercuten en la terminación del producto, pero considero, sin embargo, que nos esforzamos para hacer nuestro mejor papel, aunque muchas veces el problema no es por conocimientos, sino de actitud.

Es justo decir que al área llega mucha gente que no tiene la experiencia ni las habilidades necesarias, lo cual es hasta cierto punto comprensible, pero lo que es inconcebible es que muchas veces no traen ni siquiera las bases de la programación.

En otros casos, aún hay gente que piensa que el desarrollo del software es un trabajo para los 'obreros' de los sistemas, pero puedo decir con la suficiente autoridad que ser programador y ser parte del área de Desarrollo me ha dado las habilidades necesarias para ser un recurso capaz e importante en el eslabón de la cadena, ya que, particularmente en BanCrecer, somos los recursos de Desarrollo los que conocemos más a fondo los sistemas y quienes finalmente asesoramos a las áreas de Diseño e Implantación, a las áreas de Metodología e Incidencias y estamos siempre en contacto con las áreas de Soporte Técnico.

Además, ser programador tiene sus ventajas y satisfacciones, pues programar nos da la oportunidad de:

- El gusto de hacer las cosas tal como nosotros las pensamos y hacer que funcionen como queremos.
- El placer de crear cosas útiles (software) para alguien más.
- El saber que lo que estamos haciendo es parte importante de un sistema complejo, como cada pieza en un rompecabezas.
- El aprendizaje continuo. A veces teórico, a veces práctico, a veces ambos.

Sin embargo, muchas veces el exceso de confianza en un programador también afecta el resultado final en el proyecto. El pensar y dar por hecho que todo saldrá bien hace que perdamos la meta. Como programadores, sabemos que toda actividad creativa se divide en tres partes: la idea, la implementación y la interacción. La idea es el principio de todo proyecto, la implementación es llevar a cabo físicamente esa idea, y la interacción será cuando funcione con otros proyectos. Sin embargo, es durante la implementación donde nos encontramos

con que las especificaciones eran incompletas o inconsistentes, lo cual repercute en las fallas del sistema puesto en producción.

Por ello hago hincapié, una vez más, en lo siguiente:

- Se debe identificar y organizar el problema en función del dominio de la aplicación, no de su representación final con un lenguaje de programación.
- Dividir el problema de software en subproblemas menos complejos, interconectados mediante una estructura sencilla, suficientemente independientes para permitir que el trabajo subsecuente pueda proseguir por separado en cada uno.
- Aunque la tecnología siempre está avanzando, tan pronto como liberamos algún diseño se vuelve obsoleto en términos de sus conceptos. La obsolescencia de una implementación debe ser medida contra otras implementaciones existentes.
- El reto y la misión es encontrar soluciones reales a problemas reales de acuerdo a las fechas actuales con los recursos disponibles.

Si se revisa el contenido de este trabajo, veremos que en síntesis podemos listar como alternativas generales de solución para la problemática común en las áreas de Sistemas, los siguientes factores:

- Usar una metodología consistente.
- Definir roles y responsabilidades.
- Contar con un aseguramiento de calidad dinámico.
- Establecer definiciones estandarizadas de trabajo.
- Uso de técnicas de análisis, diseño y programación.
- Planeación estratégica de proyectos.
- Administración efectiva de personal.

En el proceso de desarrollo de sistemas, aspectos tales como la calidad, la integración del proyecto, la estimación de tiempos y costos, así como los recursos humanos, son factores que se encuentran inmersos dentro del entorno de una organización, donde no solo intervienen los elementos del producto de software y los de la tecnología, sino también el cliente, la empresa, el entorno de desarrollo y las personas involucradas. Todos estos elementos se relacionan de manera directa con el proceso de desarrollo de software, quien finalmente, es el elemento de control y en donde se centraliza toda la labor de construir un producto de software de calidad.

Para asegurar un óptimo desarrollo de software se deben observar los siguientes aspectos:

- Metodología: existe, se aplica, es satisfactoria.
- Documentación: existe, está actualizada, es accesible.
- Estándares: se aplican, cómo y quién los controla.
- Participación del usuario.
- Participación de personal externo.
- Control de calidad.
- Entornos real y de prueba.
- Control de cambios.

En el caso específico de BanCrecer, he observado y considero que sería bueno que algunos de los líderes de Desarrollo pasen a ser los líderes responsables de los grupos de trabajo del área de Diseño, capacitándolos para adquirir, desarrollar o mejorar habilidades de negociación, por ejemplo. No quiero decir con esto que el área de Desarrollo se quede sin sus líderes de proyecto ni que se echen fuera a los líderes de Diseño. Con esta aseveración pretendo, primero resaltar por enésima ocasión la importancia de un buen análisis de requerimientos y un diseño lógico perfectamente delineado para llevar a cabo un desarrollo de software de calidad, y ya que he mencionado que en BanCrecer es generalmente el área de Desarrollo la que mejor conoce los sistemas, no sería mala idea tomar a lo mejor de la gente de Diseño y a los responsables de las principales aplicaciones del área de Desarrollo e integrar un equipo de calidad como responsable del análisis de los requerimientos, ello coadyuvaría a una mejor estimación de tiempos, recursos y actividades, y, por lo tanto, a una planeación efectiva de proyectos.

Ahora bien, también considero que a los líderes de proyecto, de todas las áreas, se les debe procurar una capacitación efectiva en términos de habilidades gerenciales, ya que se ha observado que quien llega a ser líder de proyecto, por lo menos en el área de Desarrollo, se lo debe a sus habilidades y conocimientos técnicos, dejando de lado los aspectos clave de la administración de personal, lo cual influye en la motivación o desmotivación de los recursos. Quizá haya alguien que es un recurso excelente técnicamente hablando. Tal vez sea el que mejor conoce una aplicación, pero pudiera no tener la más mínima idea de cómo manejar un grupo de trabajo, de cómo administrar y explotar las fortalezas de su gente. Entonces, no se debe perder un buen técnico y ganar un mal jefe de proyectos. Se ha dado el caso. Esto no quiere decir que se deje de lado y busquemos un líder del proyecto por otra parte, puesto que se ha visto también que hay gente muy buena para administrar a un grupo de personas, pero si no tiene idea de las actividades que son necesarias para llevar a cabo un proyecto, aspectos tales como la codificación de los programas por ejemplo, tampoco será muy bueno a la hora de estimar tiempos y planear las tareas de los recursos.

Lo que se debe buscar es un buen equilibrio entre ambos perfiles y lograr la mejor combinación que satisfaga las necesidades de los individuos y de la empresa, lo cual repercutirá en el desempeño de todo el equipo de trabajo. Por ello se recomienda que se prepare gerencialmente a la gente, cuyos méritos hablan de sus habilidades para dirigir a un grupo de gente a su cargo y no solo conformarse con exigir resultados sin haber hecho una inversión previa en su persona, que a la larga dará los frutos esperados.

Esto se vuelve aún más importante de considerar, puesto que la tendencia ha sido que los subdirectores de área, e incluso los directores, provengan de quienes alguna vez fueron los programadores, los analistas y los líderes de proyecto. Vale la pena mencionar este aspecto ya que resulta estimulante que en BanCrecer sea reconocido el esfuerzo de los empleados de esta manera.

Otro punto que resulta evidente en un área laboral como lo es Sistemas, es el trabajo bajo presión, lo cual repercute además de nuestro desempeño, en nuestra calidad de vida. Casi todos nosotros cuando empezamos a sentir las presiones del trabajo y la falta de tiempo, pretendemos resolver el problema trabajando más intensamente, nos quedamos más tarde en la oficina, vamos a ella el fin de semana o nos llevamos a casa el trabajo que no pudimos realizar en las horas normales. Es como pretender pasar más agua por una tubería, poniendo más presión en la bomba que la envía. Hasta cierto punto lo logramos. El riesgo está en que podemos reventar una tubería si la presión resulta exagerada.

Estos son solo algunos factores que determinan si los sistemas en los que estamos trabajando se terminarán exitosamente. La solución no es más trabajo ni más presión. La solución es un trabajo más inteligente: esto significa jerarquizar, lo cual representa hacer primero lo más importante y valorar y aplicar en mayor o menor medida los conceptos que se han venido manejando a lo largo de este trabajo: una planeación estratégica en los proyectos, trabajar de acuerdo a una metodología, aplicar técnicas de ingeniería de software, mantener una comunicación efectiva entre áreas y realizar responsablemente las fases del ciclo de vida de un sistema.

ANEXOS

120

ACERCA DE LOS ANEXOS...

ANEXO 1a¹. Solicitud de inicio de proyecto.

En este formato se ejemplifica la información mínima requerida para hacer formalmente una solicitud de inicio de un proyecto. Dividido en dos partes, tiene por objeto mostrar información relevante para dar un panorama general de la solicitud, como es, clasificar si corresponderá a un desarrollo nuevo, una modificación o si implica un rediseño. Fecha en que se solicita y quién lo solicita. Esta información de inicio puede darnos una idea sobre la prioridad de lo solicitado. La segunda parte da un esbozo general sobre las implicaciones que representa en un momento dado para el personal de Sistemas. Es también información previa general.

ANEXO 1b. Solicitud de servicios al área de sistemas.

Cuando el desarrollo de un proyecto ha sido aceptado, se recomienda llenar un formato similar a éste, con el que formalmente se hace una petición de análisis al área de Sistemas. Como puede observarse, aquí ya se menciona el nombre del proyecto, quién lo solicitó y quién lo aprobó, dependiendo del estatus, incluso pudiera estar ya en una etapa en que al concluir el análisis previo, el proyecto haya sido rechazado. De cualquier manera, se incluyen los datos y los comentarios de quien tiene a su cargo dicho análisis.

ANEXO 6a. Gráfica de Gantt.

La gráfica de Gantt muestra el grado de avance de un proyecto, fechas de inicio y de terminación, de acuerdo a las horas estimadas y reales invertidas en cada una de las actividades que se hayan planeado.

ANEXO 6b. Checklist.

Un checklist, o listado de productos (o entregables como les llamamos en BanCrece), es un reporte global de cada una de las actividades asignadas a cada recurso, indicando en cada una de las fases quién es el responsable de su realización y el tiempo asignado.

En el ejemplo, la columna 'No' indica el número consecutivo asignado a cada entregable, la columna 'Rec' se refiere al recurso responsable, en este caso se muestran sus iniciales, y enseguida se muestran la fecha en que se asignó y las horas invertidas. Al final, se muestra un detalle tanto de los entregables como de las horas en general terminadas y pendientes.

ANEXO 6c. Reporte de avance.

Como puede observarse, el reporte de avance nos da una visión general del estatus del proyecto al momento de la fecha de corte. Mostrando una comparación entre la semana anterior y la última semana evaluada.

¹ La numeración de los anexos corresponde con el número del capítulo en el que se les hace referencia.

ANEXO 6d. *Gráfica de eficiencia.*

La gráfica de eficiencia muestra por cada semana cuál es el grado de eficiencia del grupo de trabajo, considerando como mínimo deseable el 80, en la escala de 0 a 100. Estos datos son obtenidos de acuerdo a otros reportes que se generan para medir la productividad de los recursos.

ANEXO 6e. *Gráfica de esfuerzo.*

La gráfica de esfuerzo también se realiza con corte semanal y muestra en que momento del proyecto se incrementa el esfuerzo requerido para la realización de las actividades planeadas. Esta gráfica nos da también una visión sobre qué tan bien estamos administrando los recursos y si las actividades fueron correctamente estimadas en tiempo.

ANEXO 6f. *Reporte mensual de productividad.*

El reporte de productividad, aunque se revisa semanalmente, sirve de base para generar un acumulado y medir los resultados cada mes. Aquí se presenta la información relevante sobre los tiempos incurridos por cada uno de los recursos asignados a determinado equipo de trabajo. Tomando como base que cada semana laboral comprende 40 horas, a partir de ahí se detalla cómo han sido éstas trabajadas. Así pues, este reporte da a la Dirección del área un visión detallada del personal: si gozó de vacaciones o incapacidad, si asistió a algún curso, si trabajó horas extras, si incurrió en horas improductivas para el proyecto o actividades en las que estuvo asignado, cuántas horas liberó o produjo realmente al cierre de semana, etc.

ANEXO 6g. *Reporte de asignación.*

Este reporte, similar al anterior, da una visión global no solo de un equipo de trabajo, sino de todo el personal de la Dirección. Indica que al mes de corte, se estimaron 184 horas de trabajo y muestra cómo fueron invertidas esas horas, de acuerdo al reporte de productividad, asimismo indica por cada recurso a qué proyecto y subdirección pertenece, así como si es personal interno o externo en la Empresa. Este reporte es para la Dirección.

ANEXO 6h. *Reporte de liberación.*

El reporte de liberación da información general por cada recurso en el área, pero sobre todo indica en qué proyecto se encuentra asignado actualmente y cuándo estará disponible para nuevos proyectos o actividades.

ANEXO 6i. *Relación de recursos.*

El reporte de relación de recursos es una proyección a futuro sobre el número de recursos y horas que se estiman para la siguiente semana que serán asignados a determinadas actividades. Además, éste nos indica si es personal externo (columnas 'Ast, CP, ... BC²'), si el recurso se encuentra de incapacidad, vacaciones o es becario (columna 'Edo. Rec' o estatus del recurso).

² Abreviación de los nombres de cada consultoría participante.

ANEXO A. Breve descripción de 'Altamira'.

Este anexo tiene la finalidad de hacer una breve reseña del sistema empleado en BanCrecer, Altamira, el cual es la parte medular del desarrollo de este trabajo, pues es a través de él donde he adquirido los conocimientos y la experiencia que me han llevado a la realización de esta memoria. Por ello he considerado importante mencionarlo.

ANEXO B. Trayectoria Profesional.

Aparte del sistema con el que tantas cosas he aprendido, he querido hacer una breve reseña sobre mi trayectoria profesional en la Empresa, mencionando cuál ha sido mi evolución y mis logros dentro de ella.

ANEXO C. Biografía: Jack Kilby.

Me pareció interesante incluir la biografía de Jack Kilby a manera de breviarío cultural, puesto que este personaje ha influido de alguna manera en nuestra vida profesional, al ser inventor de la materia base de nuestra principal herramienta de trabajo: el microprocesador.

ANEXO D. Desarrollo de Sistemas.

Quise incluir esta caricatura de lo que en la realidad parece ser la mejor representación del ciclo de vida en el desarrollo de sistemas. Aunque resulta graciosa, no deja de ser un llamado de atención para ver lo que precisamente debemos evitar y que es justamente lo que pretende esta memoria profesional.

ANEXO E. Plegaria de un programador.

Por último, para quienes nos dedicamos a la creación de software mediante la codificación del mismo, resulta graciosa pero realmente cierta esta 'Plegaria de un Programador'. Ignoro quién haya sido el autor, pero cuando llegó a mis manos no pude evitar identificarme con cada una de las oraciones. Yo creo que toda persona que se precie de ser o haber sido programador alguna vez, sabrá justamente a qué se refiere.

ANEXO 1a

SOLICITUD DE INICIO DE PROYECTO

PARTE I. Llenada por el solicitante.

Modificación

Rediseño

Requerimiento nuevo

Fecha de solicitud: _____

Presentada por: _____

Departamento: _____

Naturaleza de la solicitud: _____

Razones de la solicitud:

Anexos de documentos de apoyo:

PARTE II. Llenada por personal de Sistemas.

Las modificaciones parecen ser:

Menores

Mayores

Extensas

La información puede requerir cantidad adicional de:

Hardware

Software

Personal

Los recursos requeridos serían:

Menores

Mayores

Extensos

Factores de factibilidad (porcentaje):

Técnico

Económico

Operacional

Investigación preliminar desarrollada por: _____

Nombre del proyecto: _____

Fecha de la petición: _____

Estado de la petición:

En revisión

Aprobada

Rechazada

Fecha de inicio en caso de ser aprobada: _____

ANEXO 1b

SOLICITUD DE SERVICIOS AL ÁREA DE SISTEMAS

Nombre del sistema: _____

Nombre del proyecto: _____

Beneficios anticipados: _____

Solicitado por: _____

Departamento: _____

Aprobado por: _____

Departamento: _____

Situación actual de la solicitud:

Aprobada

Aprobada bajo condición

Rechazada

Motivo:

Nombre y firma:

Departamento:

Puesto:

Fecha:

Comentarios:

GO EAST
JUL 20 1978

ANEXO 6a

GRÁFICA DE GANTT

PROYECTO: BCER00035
 FECHA INICIO: 05/04/2002 TERM: 13/06/2002
 FECHA CORTE: 02/05/2002

TIEMPO TRANSCURRIDO												
FECHAS DE CORTE		05 ABR 11 ABR	12 ABR 18 ABR	19 ABR 25 ABR	26 ABR 02 MAY	03 MAY 09 MAY	10 MAY 16 MAY	17 MAY 23 MAY	24 MAY 30 MAY	31 MAY 06 JUN	07 JUN 13 JUN	
FASE												
ELABORACIÓN DE PLAN DE TRABAJO	TP											
	TA	100%										
CRITERIOS DE ACEPTACIÓN	TP											
	TA	100%										
ANÁLISIS	TP											
	TA	100%										
ESPECIFICACIONES	TP											
	TA	100%										
CONSTRUCCIÓN	TP											
	TA	100%										
PRUEBAS UNITARIAS	TP											
	TA	75%										
PRUEBAS INTEGRALES	TP											
	TA	25%										
INSTALACIÓN	TP											
	TA											
PORCENTAJE DE AVANCE							40%					

TP Tiempo planeado
 PA Porcentaje de avance

TESIS CON
 FALLA DE ORIGEN

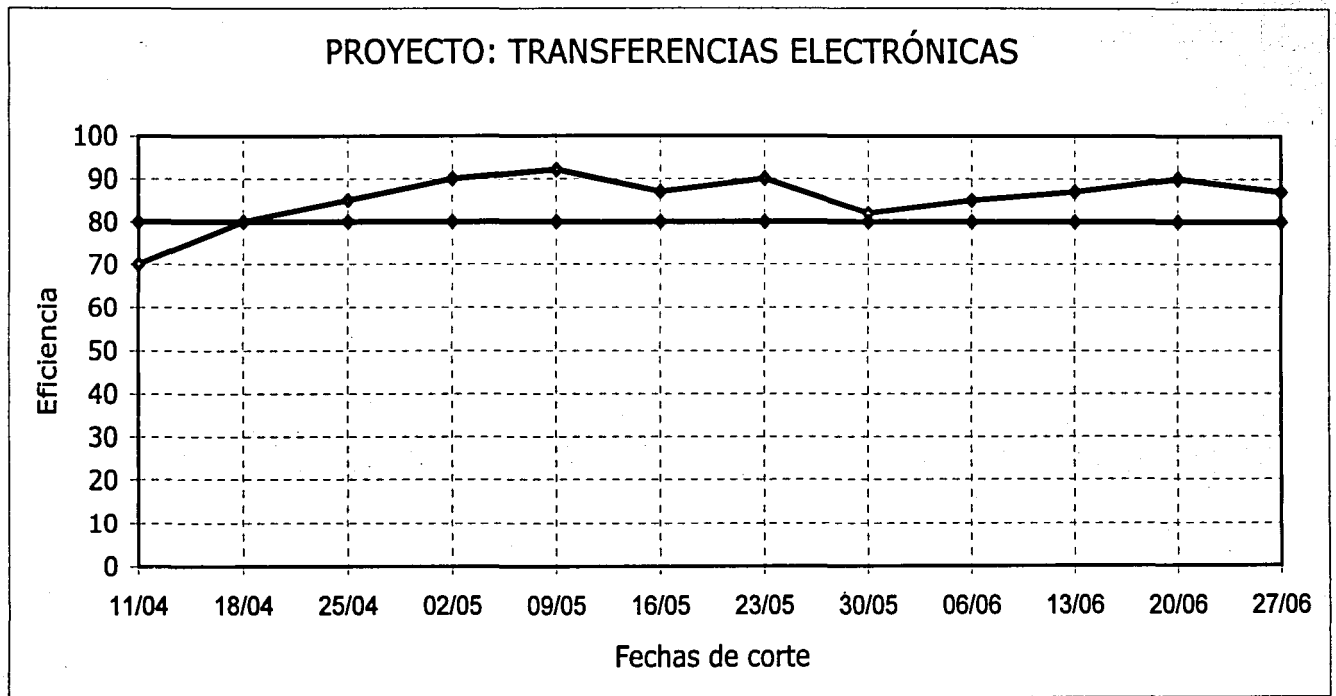
BCER00035 ENTREGABLE	CONSTRUCCIÓN											
	1 CODIFICACIÓN				2 PRUEBA UNITARIA				3 CORRECCIÓN			
	No	Rec	Fecha	Hrs	No	Rec	Fecha	Hrs	No	Rec	Fecha	Hrs
TEF001	1	RMO	18-Abr	20	6	GMR	18-Abr	5	11	RMO	3-May	6
TEF002	2	JAP	18-Abr	20	7	GMR	18-Abr	5	12	JAP	3-May	6
TEF003	3	RMO	20-Abr	18	8	GMR	20-Abr	4	13			5
TEF004	4	JAP	20-Abr	20	9			5	14			6
TEF005	5			10	10			2	15			3
ENTREGABLES												
TOTAL				5				5				5
TERMINADOS				4				3				2
NO TERMINADOS				1				2				3
HORAS												
TOTAL				88				21				26
TERMINADAS				78				14				12
NO TERMINADAS				10				7				14

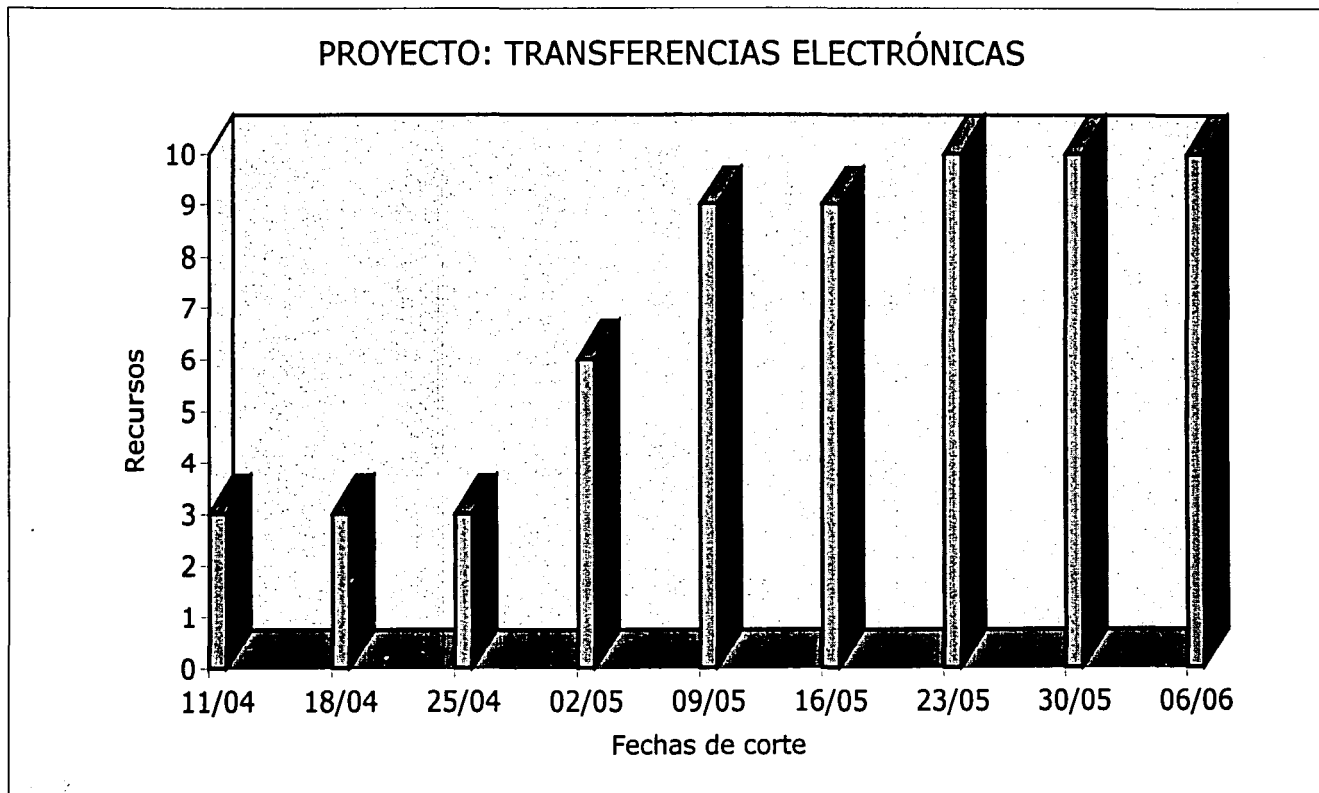
ANEXO 6c

REPORTE DE AVANCE

PROYECTO :	BCER00035	COMESA
SUBPROYECTO :	BCER000351	TRANSFERENCIAS ELECTRÓNICAS
FECHA TÉRMINO:	13/06/2002	
FECHA CORTE :	02/05/2002	

CONCEPTO	SEMANA ANTERIOR	SEMANA ACTUAL
Horas requeridas	12,541	12,541
Avance en horas	11,408	11,684
Horas faltantes	1,133	857
Porcentaje de avance	90	93
Duración total (semanas)	22	22
Tiempo transcurrido (semanas)	17	18
Porcentaje de tiempo transcurrido	77	82
Eficiencia	117	114
Horas esperadas	274	227
Horas reales	276	228
Eficiencia semanal	100	100
Horas esperadas siguiente semana	226	214





NO. 1234
MAY 17 1938

REPORTE MENSUAL DE PRODUCTIVIDAD

COMPARATIVO	CAPACIDAD	INC.	VAC.	BECAS	CURSOS	HORAS	COSTO	OTRAS	ASIG1	EXTRA	EXTRA	PACID	HORAS	ENT	COSTO	OST	PROD	OBSERVACIONES	COMPANIA
TOTAL	TOTAL					IMPROD	ADMIN	ASIG2	PROY	DOBLE	TRIP	MCM	IBERADA	TERM	REAL	ERAD	%		
DESARROLLO (ENERO)																			
																			BanCrecar
(SEM 02 - 10)	56	0	0	0	0	0	56	0	0	0	0	0	0	0	1	0	#1DIV/01	Administración	
(SEM 11 - 17)	40	0	0	0	0	0	40	0	0	0	0	0	0	0	1	0	#1DIV/01	Administración	
(SEM 18 - 24)	40	0	0	0	0	0	40	0	0	0	0	0	0	0	1	0	#1DIV/01	Administración	
(SEM 25 - 31)	40	0	0	0	0	0	40	0	0	0	0	0	0	0	1	0	#1DIV/01	Administración	
TOTAL	176	0	0	0	0	0	176	0	0	0	0	0	0	0	4	0	#1DIV/01		
																			BanCrecar
ARTURO ALCÁNTARA																			
(SEM 02 - 10)	48	0	0	0	0	0	0	0	48	0	0	0	0	36	1	0	75	Soporte al batch	
(SEM 11 - 17)	36	0	0	0	0	0	0	0	36	0	0	0	0	36	0	0	100	Soporte al batch	
(SEM 18 - 24)	36	0	0	0	0	0	0	0	36	0	0	0	0	36	0	0	100	Soporte al batch	
(SEM 25 - 31)	36	0	0	0	0	0	0	0	36	0	0	0	0	36	0	0	100	Soporte al batch	
TOTAL	156	0	0	0	0	0	0	0	156	0	0	0	0	144	1	0	92		
																			BanCrecar
JESUS ARRIETA																			
(SEM 02 - 10)	56	0	0	0	0	0	0	0	56	0	0	0	0	55	1	0	98	TEF	
(SEM 11 - 17)	40	0	0	0	0	0	0	0	40	0	0	0	0	41	0	0	103	TEF	
(SEM 18 - 24)	40	0	0	0	0	0	0	0	40	0	0	0	0	40	0	0	100	TEF	
(SEM 25 - 31)	40	0	0	0	0	0	0	0	40	0	0	0	0	32	0	0	80	TEF	
TOTAL	176	0	0	0	0	0	0	0	176	0	0	0	0	168	1	0	95		
																			Asteci
RAFAEL MUÑOZ																			
(SEM 02 - 10)	56	0	0	0	0	0	0	0	56	0	0	0	0	56	1	0	100	TEF	
(SEM 11 - 17)	40	0	0	0	0	0	0	0	40	0	0	0	0	40	0	0	100	TEF	
(SEM 18 - 24)	40	8	0	0	0	0	0	0	32	0	0	0	0	36	0	0	113	TEF	
(SEM 25 - 31)	40	0	0	0	0	0	0	0	40	0	0	0	0	40	0	0	100	TEF	
TOTAL	176	8	0	0	0	0	0	0	168	0	0	0	0	172	1	0	102		
																			BanCrecar
VICTOR RAMIREZ																			
(SEM 02 - 10)	56	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Vacaciones	
(SEM 11 - 17)	40	0	8	0	0	0	0	0	32	0	0	0	0	32	0	0	100	TEF	
(SEM 18 - 24)	40	0	0	0	0	0	0	0	40	0	0	0	0	40	0	0	100	TEF	
(SEM 25 - 31)	40	0	0	0	0	0	0	0	40	0	0	0	0	40	0	0	100	TEF	
TOTAL	176	0	64	0	0	0	0	0	112	0	0	0	0	112	1	1	100		
																			BanCrecar
GERARDO PACHECO																			
(SEM 02 - 10)	56	0	0	0	0	3	0	0	40	0	0	13	15	40	0	0	104	R13028 / TEF	
(SEM 11 - 17)	40	0	0	0	0	0	0	0	40	0	0	0	0	48	0	0	120	TEF	
(SEM 18 - 24)	40	0	0	0	0	0	0	0	40	0	0	0	0	40	0	0	100	TEF	
(SEM 25 - 31)	40	0	0	0	0	0	0	0	40	0	0	0	0	40	0	0	100	TEF	
TOTAL	176	0	0	0	0	3	0	0	160	0	0	13	15	168	2	2	106		

TESIS CON
 FALTA DE ORIGEN

REPORTE DE ASIGNACIÓN

COMPARATIVO	PUESTO	CAPACIDAD	INC.	VAC.	BECAS	CURSOS	HORAS	COSTO	OTRAS	ASIG.	EXTRA	EXTRA	PACID	HORAS	ENT	PROYECTO	SUBDIR	COMPANIA	DESARROLLO DE SISTEMAS		ENERO	
																			TOTAL	ASIG.	TRIP.	MCM.
Alejandro Cortés Vidaur	Analista	184	0	0	0	0	0	0	0	96	0	0	88	50	160	Contabilidad	Desa-III	Asteci				
Amanda Carrasco Carrasco	Programador	184	0	0	0	0	0	0	0	112	0	0	72	16	160	SAR	Desa-II	Asteci				
Enrique Batalla Brandt	Analista	184	0	24	0	0	0	0	0	160	0	0	0	0	168	Recaudaciones	Desa-II	Asteci				
Francisco Ureña Rosete	Programador	184	0	0	0	0	0	0	0	117	0	0	67	52	110	Contabilidad	Desa-III	Asteci				
Gerardo Salinas Muñoz	Programador	184	0	0	0	0	0	0	0	96	0	0	88	112	152	Colocación	Desa-II	Asteci				
Héctor Froyán Gervasio Bustos	Analista	184	0	0	0	0	0	0	0	0	0	0	184	227	0	Recaudaciones	Desa-II	Asteci				
Israel González Acosta	Analista	184	0	0	0	0	0	0	0	90	0	0	94	134	88	Tesofe	Desa-I	Asteci				
Iván Alfonso Sánchez Vázquez	Analista	184	8	0	0	0	0	0	0	102	0	0	74	72	102	SAR	Desa-II	Asteci				
Jaime López Domínguez	Analista	184	0	16	0	0	4	0	0	53	0	0	111	103	61	TEF	Desa-I	Asteci				
Luis Enrique Cárdenas Alcaraz	Analista-prog	184	0	40	0	0	0	0	0	0	0	0	144	192	5	Tesofe	Desa-I	Asteci				
Mario Ávila Moreno	Programador	184	0	0	0	0	1	0	0	74	0	0	109	50	130	Tesofe	Desa-I	Asteci				
Nelly M. Herrera Mendiola	Programador	184	0	0	0	0	0	0	0	0	0	0	184	147	4	SAR	Desa-II	Asteci				
Oscar Parra García	Analista-prog	184	0	0	0	0	0	0	0	112	0	0	72	95	84	Telmex	Desa-I	Asteci				
Rafael Muñoz Orozco	Analista	184	0	40	0	0	4	0	0	58	0	0	82	42	45	TEF	Desa-I	Asteci				
Saúl Martínez Ventura	Programador	184	0	0	0	0	0	0	0	0	0	0	184	202	0	Telmex	Desa-I	Asteci				
Victor Moreno Montiel	Analista	184	0	40	0	0	0	0	0	120	0	0	24	0	171	Telmex	Desa-I	Asteci				
Aramando Márquez Espinoza	Líder de proyecto	184	0	88	0	0	0	96	0	0	0	0	0	0	0	Recaudaciones	Desa-II	Banco				
Arturo Alcántara Rivas	Programador	168	0	0	0	0	0	0	0	168	0	0	0	0	168	Soporte al Batch	Desa-I	Banco				
Benjamin Rangel Peña	Analista-prog	184	0	0	0	0	0	0	0	0	0	0	184	166	0	Recaudaciones	Desa-II	Banco				
Carlos Ortega Romero	Programador	184	0	0	0	0	0	0	0	0	0	0	184	207	10	SAR	Desa-II	Banco				
Dolores Carvajal Medellín	Líder de proyecto	184	0	0	0	0	0	184	0	0	0	0	0	0	0	SAR	Desa-II	Banco				
Edgar Navarro Peña	Programador	184	0	16	0	0	0	0	0	88	0	0	80	40	144	Recaudaciones	Desa-II	Banco				
Ernestina Licea García	Analista	184	0	0	0	0	0	0	0	184	0	0	0	0	0	Telmex	Desa-I	Banco				
Francisco López Bonilla	Analista-prog	184	0	80	0	0	4	0	0	26	0	0	74	40	26	TEF	Desa-I	Banco				
Gerardo Pacheco Camacho	Analista-prog	184	0	0	0	0	4	0	0	90	0	0	90	64	89	TEF	Desa-I	Banco				
José De Jesús Arieta Peláez	Programador	184	0	0	0	0	4	0	0	44	0	0	136	150	13	TEF	Desa-I	Banco				
Luis Vargas Rosas	Analista	184	0	0	0	0	0	0	0	184	0	0	0	0	0	Telmex	Desa-I	Banco				
María Adelina Tadeo Trinidad	Líder de proyecto	184	0	0	0	0	0	184	0	0	0	0	0	0	0	TEF	Desa-I	Banco				
Mireya Aguirre Olmedo	Analista	184	0	0	0	0	4	0	0	88	0	0	92	75	88	TEF	Desa-I	Banco				
Pedro Martínez García	Analista	184	0	0	0	0	0	0	0	56	0	0	128	0	96	Colocación	Desa-II	Banco				
Ruth Rodríguez Moreno	Programador	184	0	0	0	0	0	0	0	96	0	0	88	77	106	Colocación	Desa-II	Banco				
Susana Michel López	Líder de proyecto	184	184	0	0	0	0	0	0	0	0	0	0	0	0	Colocación	Desa-II	Banco				
Victor Manuel Ramírez Galicia	Analista	184	0	0	0	0	4	0	0	63	0	0	117	75	63	TEF		Banco				
César Cid Contreras	Analista	184	0	0	0	0	0	0	0	80	0	0	104	2	134	Tesofe	Desa-I	Sinapsis				
David Martínez Albiter	Programador	136	0	0	0	0	0	0	0	56	0	0	80	0	56	Proteo	Desa-IV	Sinapsis				
Enrique Villaverde Padilla	Programador	184	0	0	0	104	0	0	0	80	0	0	0	0	86	Proteo	Desa-IV	Sinapsis				
Fabián Ortiz Gil	Programador	184	0	0	0	0	0	0	0	184	0	0	0	0	181	Proteo	Desa-IV	Sinapsis				
José Angel Mejía Reyes	Analista	184	0	0	0	104	0	0	0	64	0	0	16	0	84	Proteo	Desa-IV	Sinapsis				
Marcos Campos Fuentes	Analista-prog	184	0	0	0	0	0	0	0	136	0	0	48	0	136	Proteo	Desa-IV	Sinapsis				
Sandra Almazán Flores	Programador	114	0	0	0	0	0	0	0	114	0	0	0	0	96	Colocación	Desa-II	Sinapsis				
Tania Hidalgo Baeza	Programador	184	0	0	0	0	0	0	0	105	0	0	79	60	144	Colocación	Desa-II	Sinapsis				
Victor Belmont Paredes	Analista	48	0	0	48	0	0	0	0	0	0	0	0	0	0	Tesofe	Desa-I	Sinapsis				

COMPARATIVO	PUESTO	R.H.R.	COSTO		OTRAS		ASIGN.	CLAV.	PROY. C/PROY.	FECHA	FECHA	UBIC.	COMPANIA
			TOTAL	ADMIN.	BASE	PROY.							
Alejandro Cortés Vidaur	Analista	156	0	60	96	BCER00028	Contabilidad	13-Ene-02	10-Feb-02	Desa-III	Asteci		
Amanda Carrasco Carrasco	Programador	156	0	0	156	BCER00029	SAR	15-Dic-01	30-Ene-02	Desa-II	Asteci		
Enrique Batalla Brandt	Analista	156	0	0	156	BCER00015	Recaudaciones	15-Dic-01	15-Ene-02	Desa-II	Asteci		
Francisco Ureña Rosete	Programador	156	0	0	156	BCER00028	Contabilidad	13-Ene-02	10-Feb-02	Desa-III	Asteci		
Gerardo Salinas Muñoz	Programador	156	0	0	156	BCER00002	Colocación	06-Sep-01	14-Ene-02	Desa-II	Asteci		
Héctor Froylán Gervasio Bustos	Analista	156	0	0	156	BCER00015	Recaudaciones	15-Dic-01	15-Ene-02	Desa-II	Asteci		
Israel González Acosta	Analista	156	0	0	156	BCER00031	Tesofe	07-Ene-02	28-Feb-02	Desa-I	Asteci		
Iván Alfonso Sánchez Vázquez	Analista	156	0	54	102	BCER00029	SAR	15-Dic-01	30-Ene-02	Desa-II	Asteci		
Jaime López Domínguez	Analista	156	0	0	156	BCER00035	TEF	05-Abr-02	13-Jun-02	Desa-I	Asteci		
Luis Enrique Cárdenas Alcaraz	Analista-prog	156	0	0	156	BCER00031	Tesofe	07-Ene-02	28-Feb-02	Desa-I	Asteci		
Mario Avila Moreno	Programador	156	0	0	156	BCER00031	Tesofe	07-Ene-02	28-Feb-02	Desa-I	Asteci		
Nelly M. Herrera Mendiola	Programador	156	0	0	156	BCER00029	SAR	15-Dic-01	30-Ene-02	Desa-II	Asteci		
Oscar Parra García	Analista-prog	156	0	44	112	BCER00023	Telmex	14-Ene-02	25-Ene-02	Desa-I	Asteci		
Rafael Muñoz Orozco	Analista	156	0	0	156	BCER00035	TEF	05-Abr-02	13-Jun-02	Desa-I	Asteci		
Saúl Martínez Ventura	Programador	156	0	0	156	BCER00023	Telmex	28-Ene-02	04-Feb-02	Desa-I	Asteci		
Victor Moreno Montiel	Analista	156	0	0	156	BCER00023	Telmex	28-Ene-02	04-Feb-02	Desa-I	Asteci		
Armando Márquez Espinoza	Líder de proyecto	156	156	0	0	BCER00015	Recaudaciones	15-Dic-01	15-Ene-02	Desa-II	Banco		
Arturo Alcántara Rivas	Programador	156	0	0	156	BCER00007	Soporte al Batch	02-Ene-02	02-Abr-02	Desa-I	Banco		
Benjamin Rangel Peña	Analista-prog	156	0	0	156	BCER00015	Recaudaciones	15-Dic-01	15-Ene-02	Desa-II	Banco		
Carlos Ortega Romero	Programador	156	0	0	156	BCER00029	SAR	15-Dic-01	30-Ene-02	Desa-II	Banco		
Dolores Carvajal Medellín	Líder de proyecto	156	156	0	0	BCER00029	SAR	15-Dic-01	30-Ene-02	Desa-II	Banco		
Edgar Navarro Peña	Programador	156	0	0	156	BCER00015	Recaudaciones	15-Dic-01	15-Ene-02	Desa-II	Banco		
Emestina Licea García	Analista	156	0	0	156	BCER00023	Telmex	14-Ene-02	25-Ene-02	Desa-I	Banco		
Francisco López Bonilla	Analista-prog	156	0	0	156	BCER00035	TEF	05-Abr-02	13-Jun-02	Desa-I	Banco		
Gerardo Pacheco Camacho	Analista-prog	156	0	0	156	BCER00035	TEF	05-Abr-02	13-Jun-02	Desa-I	Banco		
José De Jesús Arrieta Peláez	Programador	156	0	0	156	BCER00035	TEF	05-Abr-02	13-Jun-02	Desa-I	Banco		
Luis Vargas Rosas	Analista	156	0	0	156	BCER00023	Telmex	14/01/022	31-Ene-02	Desa-I	Banco		
María Adalina Tadeo Trinidad	Líder de proyecto	156	156	0	0	BCER00035	TEF	05-Abr-02	13-Jun-02	Desa-I	Banco		
Mireya Aguirre Olmedo	Analista	156	0	0	156	BCER00035	TEF	05-Abr-02	13-Jun-02	Desa-I	Banco		
Pedro Martínez García	Analista	156	0	0	156	BCER00002	Colocación	06-Sep-01	14-Ene-02	Desa-II	Banco		
Ruth Rodríguez Moreno	Programador	156	0	0	156	BCER00002	Colocación	06-Sep-01	14-Ene-02	Desa-II	Banco		
Susana Michel López	Líder de proyecto	156	156	0	0	BCER00002	Colocación	06-Sep-01	14-Ene-02	Desa-II	Banco		
Victor Manuel Ramírez Galicia	Analista	156	0	0	63	BCER00035	TEF	05-Abr-02	13-Jun-02	Desa-I	Banco		
César Cid Contreras	Analista	156	0	0	156	BCER00031	Tesofe	07-Ene-02	28-Feb-02	Desa-I	Sinapsis		
David Martínez Albiter	Programador	156	0	0	156	BCER00014	Proteo	14-Nov-01	16-Ene-02	Desa-IV	Sinapsis		
Enrique Villaverde Padilla	Programador	156	0	0	156	BCER00014	Proteo	02-Ene-02	16-Ene-02	Desa-IV	Sinapsis		
Fabián Ortiz Gil	Programador	156	0	0	156	BCER00014	Proteo	02-Ene-02	16-Ene-02	Desa-IV	Sinapsis		
José Angel Mejía Reyes	Analista	156	0	0	156	BCER00014	Proteo	14-Nov-01	16-Ene-02	Desa-IV	Sinapsis		
Marcos Campos Fuentes	Analista-prog	156	0	0	156	BCER00014	Proteo	14-Nov-01	16-Ene-02	Desa-IV	Sinapsis		
Sandra Almazán Flores	Programador	156	0	0	156	BCER00002	Colocación	06-Sep-01	14-Ene-02	Desa-II	Sinapsis		
Tania Hidalgo Baeza	Programador	156	0	0	156	BCER00002	Colocación	06-Sep-01	14-Ene-02	Desa-II	Sinapsis		
Victor Belmont Paredes	Analista	156	0	0	156	BCER00031	Tesofe	07-Ene-02	28-Feb-02	Desa-I	Sinapsis		

144

RELACIÓN DE RECURSOS
Del 05 al 11 de Abril 2002

PROYECTO	ADICIONALES	INDICADORES	VALOR	UNIDAD	MONTO	MONTO	MONTO	MONTO	MONTO	MONTO	TOTAL
					OCURR	OCURR	OCURR	OCURR	OCURR	OCURR	OCURR
GRUPO 1.2											
DESARROLLO DE SISTEMAS I											
Lider de Proyecto											
María Adeline Tadeo Trinidad			1		0	0	40	0	0	0	40
TRANSFERENCIAS ELECTRÓNICAS											
Victor Manuel Ramirez Galicia			1	V	0	0	0	0	0	40	8 32
Mireya Aguirre Olmedo			1		0	0	0	0	0	40	0 40
Francisco López Bonilla			1		0	0	0	0	0	40	0 40
Gerardo Pacheco Camacho			1		0	0	0	0	0	40	0 40
José de Jesús Arrieta Peláez			1		0	0	0	0	0	40	0 40
Rafael Muñoz Orozco	1				0	0	0	0	0	40	0 40
Jaime López Domínguez	1				0	0	0	0	0	40	0 40
TOTAL TRANSFERENCIAS	2	0	0	0	0	0	0	0	0	280	8 272
SOPORTE NOCTURNO AL BATCH											
Arturo Alcánta Rivas			1		0	0	0	0	0	36	0 36
Roberto Nájera García			1		0	0	0	0	0	36	0 36
TOTAL SOPORTE NOCTURNO	0	0	0	0	0	0	0	0	0	36	0 36

ANEXO A

BREVE DESCRIPCIÓN DE 'ALTAMIRA'

Altamira es un sistema modular integrado para el soporte del negocio bancario con tecnología informática de vanguardia, con una plataforma abierta para el desarrollo de todos los sistemas de información de la entidad.

Sus principales características son:

- Relación global del cliente con la entidad.
- Flexibilidad para generar o adaptar productos.
- Incremento de la productividad y mejora de la calidad de servicio.
- Diferentes canales de distribución.
- Eficacia operativa.
- Sistema multientidad y multidivisa.

Este sistema ha facilitado a los usuarios:

- La gestión de las relaciones con los clientes
- El control de información para la toma de decisiones
- La operación de los productos bancarios
- La distribución de información a clientes y terceros
- La efectividad en el trabajo diario

A continuación se muestran los módulos que integran este sistema de información bancario, particularmente en Bancrecer, con el fin de dar una idea global de su dimensión en términos de software.

MÓDULO / APLICACIÓN	No. DE PROGRAMAS
Arquitectura	1,072
Captación	3,170
Colocación	2,614
Contabilidad	703
Seguros	425
Domiciliaciones	481
Extranjero	1,100
Medios de Pago	355
MIS	3,322
Clientes	392
Pensiones	677
Tablas Corporativas	479
Conciliador	250
TOTAL	15,040

ANEXO A

PLATAFORMA Y SOFTWARE UTILIZADO

Altamira es un sistema implantado en un ambiente tipo host con plataforma mainframe de IBM, con terminales o emulaciones tipo 3270, administrado bajo el sistema operativo OS/390 y operado mediante el editor TSO ISPF. El código está desarrollado en lenguaje Cobol, las bases de datos son relacionales, el manejador utilizado es DB2, por lo que la programación es con SQL para los accesos a tablas. La operación on line del sistema se desarrolla en entorno de CICS y la operación en batch se controla mediante jcl's.

Se cuenta con diversas y modernas herramientas, que facilitan el desarrollo de las aplicaciones, entre las que destacan sofisticados softwares para el control y administración de las versiones existentes de código de programación en cada ambiente operativo¹ del Banco.

¹ Cada uno de los ambientes en los que se corren los procesos operativos del Banco. Desarrollo, ambiente empleado para la creación de las aplicaciones. Test, ambiente exclusivo para pruebas del área de Implantación. Formación, ambiente empleado para capacitación. Producción, ambiente real operativo.

ANEXO B

TRAYECTORIA PROFESIONAL

Como parte de los recursos humanos de BanCrecer, quiero exponer lo siguiente:

Hace poco más de ocho años que concluí la carrera, desde entonces he formado parte del personal de este Banco.

Al principio, fui contratada con un puesto de becario por la principal consultoría que en ese entonces ofrecía sus servicios a la Institución, Andersen Consulting, propietaria del sistema 'Altamira'.

Como becaria, aprendí muchas cosas básicas en un ambiente mainframe. Participé en el apoyo para la puesta en marcha de Altamira en BanCrecer, una filial de BanCrecer en Costa Rica. Conocí a mucha gente de España, Costa Rica y, por supuesto, México.

Aunque en un inicio tuve cierto temor por el mundo que se me presentaba: un sistema enorme, mucha gente nueva, jefes buenos y malos, etc., hoy sé que todo ello forma parte de lo que la vida nos ofrece y que nos enriquece personal y profesionalmente.

Afortunadamente salí adelante con éste, mi primer proyecto. En seis meses me ofrecieron un empleo base como empleada del Banco, y desde mi primer puesto de programador junior al actual de líder de proyectos no he dejado de aprender cosas nuevas.

He tenido la fortuna de participar en proyectos de todos tamaños, de distintas formas, en conversiones, nuevos desarrollos, mantenimiento, optimización, en la creación de nuevos módulos para agrandar aún más a Altamira, ajustándolo a las necesidades propias de la Empresa, pues de ser un sistema español creado en Catalunya, hoy el Altamira de BanCrecer es un sistema totalmente personalizado y diferente.

Me enorgullece decir que de todos los bancos en México que cuentan con Altamira, el nuestro es el sistema más estable. Hoy nuestros tiempos de respuesta y atención al cliente son de los mejores. Y en Costa Rica, tuvo mi equipo de trabajo de ese tiempo, la satisfacción de recibir la felicitación y el reconocimiento del presidente general del Banco.

Todo esto sin duda ha sido muy motivante. Es cierto que ha habido buenos y malos momentos, y siempre los habrá, pero la experiencia que ello nos deja es invaluable. El aprendizaje es continuo, como ya he mencionado en las conclusiones del trabajo, a veces teórico, a veces práctico, pero siempre habrá algo que aprender.

He aprendido y estoy aprendiendo aún a administrar a un grupo de gente, pues con nuestros compañeros siempre hay algo nuevo que aprender. El reto es siempre fascinante: conducir a un grupo de trabajo al logro de sus objetivos, para satisfacción personal y de la Empresa.

Puedo decir en resumen que es bueno adaptarse a los cambios, mejor aún, anticiparse a ellos y dar siempre lo mejor de sí mismos. Y aunque son ya varios años de permanencia en esta Institución, ahora bajo una nueva razón social, creo que aún hay muchas cosas que hacer y que aprender, tal vez más adelante llegaré al final de este ciclo y comenzaré uno nuevo, pero este es el resumen de mi vida profesional hasta ahora... gracias a mi preparación como un egresado de la carrera de Matemáticas Aplicadas y Computación.

Proyectos en los que he participado:

ANEXO B

- Liberación y mantenimiento del módulo de Colocación¹ en BanCrecen; 1994.
- Desarrollo de nuevos módulos para BanCrecen; 1995.
- Conversión del módulo de Colocación BanCrecen, primera fase; 1996.
- Conversión del módulo de Colocación BanCrecen, segunda fase; responsable; 1997.
- Mantenimiento y corrección al módulo de Colocación BanCrecer en el área de créditos con recursos propios para el saneamiento del Banco; responsable; 1998.
- Desarrollo del módulo de Colocación Gran Caiman; responsable; 1999.
- Soluciones Definitivas al Batch general; responsable; 2000.
- Optimización General de Procesos; responsable; 2001.
- Transferencias Electrónicas de Fondos; responsable; 2002.
- Domiciliación de Recibos (Cargos Interbancarios); responsable; 2002.
- Soporte al sistema en producción; responsable; 2002.

¹ Módulo para la administración de créditos bancarios.

ANEXO C

GRACIAS POR EL CHIP

JACK KILBY: SU INVENTO CAMBIÓ AL MUNDO

(Tomado de la revista Selecciones, Septiembre 2001. Por T. R. Reid. Págs 48-51)

Para alguien que una vez salió reprobado en un examen de matemáticas, fue maravilloso lo que ocurrió el 10 de diciembre del año 2000 en Estocolmo: Jack Kilby, hombre de trato afable oriundo de Kansas, a quien el Instituto Tecnológico de Massachusetts había rechazado por sacar notas bajas en matemáticas, recibió el premio Nobel de Física, lo cual resulta tanto más notable cuanto que Kilby no es físico ni tiene gran preparación formal en la materia.

Si la Real Academia Sueca de Ciencias decidió pasar por alto este detalle es porque Kilby, después de todo, logró el invento más valioso de la segunda mitad del siglo XX: el chip o microprocesador, e inauguró así la era de la informática.

Podría decirse que este dispositivo de silicio, parte esencial de todos los aparatos electrónicos, se ha vuelto el producto industrial más importante después del petróleo crudo. Sin él no habría computadoras personales, teléfonos celulares, Internet ni videojuegos. El microprocesador, un circuito integrado en una lámina de material semiconductor, generalmente silicio, ha cambiado al mundo tan radicalmente como la bombilla eléctrica, el teléfono y el automóvil, pero, por alguna razón, el hombre que lo inventó nunca obtuvo el reconocimiento del que disfrutaron Edison, Bell y Ford. Ahora, a sus 77 años, Jack Kilby quizá reciba la atención que merece, gracias al premio Nobel.

Jack Kilby se crió en Great Bend, Kansas. Su padre dirigía allí una central eléctrica, y cuando Jack estaba en la escuela de enseñanza media se propuso ser también ingeniero electricista. Puso la mira en la meca de los ingenieros en ciernes, el ITM, y en el verano de 1941 tomó el tren a Cambridge, Massachusetts, a fin de prepararse para el examen de admisión al instituto. Salió reprobado.

Seis décadas después, cuando su retrato está junto al de Edison en el Salón de la Fama de los Inventores Estadounidenses y posee unas 60 patentes y casi todos los premios de ingeniería, Kilby aún recuerda ese fracaso: "La calificación mínima para aprobar era de 500, y saqué 497".

A los pocos meses estalló la Segunda Guerra Mundial y Kilby, que era cabo segundo, fue asignado al taller de reparación de radio de un puesto militar de su país en la India. Al terminar la guerra asistió a la Universidad de Illinois y se recibió de ingeniero electricista. Era una época de auge de la electrónica. En 1947, tres estadounidenses inventaron el transistor, primer dispositivo de material semiconductor comercialmente importante. En seguida hubo cursos de física cuántica y circuitos transistorizados... sólo para físicos.

"No querían poner cosas tan fascinantes al alcance de gente simple como los ingenieros", cuenta Kilby.

Luego de recibirse entró a trabajar en una pequeña fábrica de piezas electrónicas llamada Centralab, porque fue la única empresa del ramo que lo aceptó. Pocos años después envió una solicitud a la Texas Instruments, en Dallas, y en 1958 se enteró con gran alegría de que lo habían contratado. Tenía 34 años.

La Texas Instruments ya era importante, pero no tanto como Kilby iba a volverla. La empresa le encargó la solución del mayor problema de la electrónica en ese tiempo: el de las "interconexiones" o "cableado".

Alentados por la invención del transistor, los ingenieros estaban diseñando circuitos para unos nuevos aparatos electrónicos: computadoras ultraveloces, capaces de sostener redes de comunicación mundiales o enviar cohetes a la luna. Sólo que estas maravillas de la tecnología no existían más que en papel, exigían tantos kilómetros de alambre y tantos millones de conexiones soldadas, que era imposible construirlas.

ANEXO C

En todo el mundo se buscaba la solución, pero Jack Kilby tenía una gran ventaja: "Como era un novato ignorante del tema, no sabía qué consideraban imposible los demás y no descarté ninguna posibilidad".

En el laboratorio de semiconductores encontró la solución: eliminar los alambres. Era una ruptura tan audaz con la historia de los circuitos, que al principio creyó que no funcionaría, pero sabía que todos los elementos básicos de un circuito podían hacerse del mismo material: silicio. Si se fabricaban todos en una misma pieza, las interconexiones podían colocarse, y hasta imprimirse, en una lasca del mismo material. Al no haber alambres ni soldaduras, cabrían muchos componentes en muy poco espacio. Se podía poner todo el circuito de una computadora en una lasca del tamaño de una uña de bebé.

El 24 de julio de 1958, Kilby anotó la idea en su cuaderno de laboratorio: " Los siguientes elementos de circuito podrían hacerse de una sola pieza: resistores, capacitor, condensador de capacitancia distribuida, transistor". Fue esta frase la que le valió el Nobel.

Pero a un ingeniero no le basta escribir la idea. "Los científicos quieren entender las cosas, los ingenieros, hacerlas funcionar", dijo una vez Kilby. Fue así como uno de los ingenieros más nuevos del laboratorio le preguntó tímidamente al jefe si podía construir un prototipo de "circuito integrado". El jefe accedió, pero, como no quería gastar demasiado, le pidió que construyera un circuito sencillo llamado oscilador desfasador, dispositivo de uso común que convierte la corriente directa en corriente alterna.

El 12 de septiembre, un grupo de directivos de la Texas Instruments fue al laboratorio para ver si el curioso circuito integrado de silicio ideado por Kilby funcionaba.

El ingeniero estaba empalmando nerviosamente algunos alambres. Luego de revisar dos veces las conexiones, tomó aliento, se encogió de hombros como diciendo "a ver qué pasa" y encendió el dispositivo.

Al instante apareció en un monitor una serpiente de luz verde: la sinusoide que representa la corriente alterna. El microprocesador funcionaba. Había nacido una era de la electrónica.

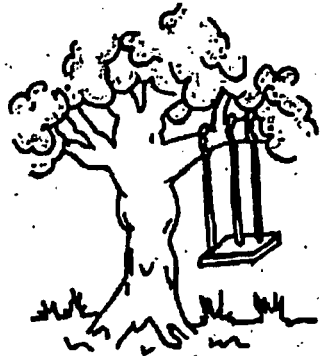
Algunos meses después, Robert Noyce, dio con un dispositivo casi idéntico, pero más fácil de fabricar, de ahí que se le considere coinventor del microprocesador. Más adelante fue cofundador de la Intel, el gigante multinacional de los microprocesadores, y sin duda habría compartido el Nobel con Kilby de no haber muerto en 1990, ya que el premio no se entrega póstumamente.

Hace apenas 43 años, el microprocesador no existía. Ahora representa una industria mundial de 177 mil millones de dólares y se encuentra en todas partes.

Jack Kilby, el estadounidense que puso en marcha esta revolución tecnológica, nunca ha recibido grandes sumas, lo cual no le importa. Es ingeniero, está dedicado a resolver problemas y ha seguido abordando cuestiones de importancia universal. Fue coinventor de uno de los primeros productos comerciales importantes de la tecnología del microprocesador: la calculadora de bolsillo, e intentó construir una célula fotoeléctrica barata.

En Dallas es una celebridad, los medios informativos los llaman el "Edison tejano", pero la mayoría de sus conciudadanos no lo conocen.

Quién sabe por qué, nuestra sociedad saturada de noticias, con su insaciable apetito de rostros nuevos, ha pasado por alto a un verdadero héroe, un hombre que mejoró la vida diaria de todo el mundo con una buena idea.



COMO LO SOLICITÓ EL USUARIO...



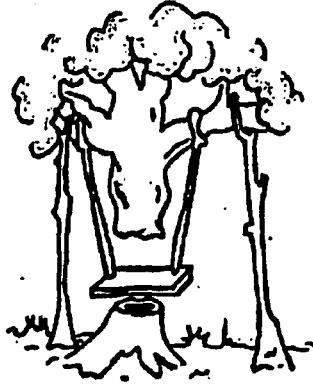
COMO LO PROPUSO EL COORDINADOR DEL PROYECTO...



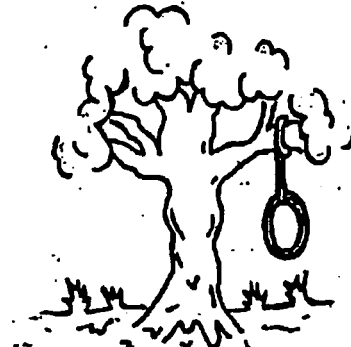
COMO SE ESPECIFICÓ EN EL ANÁLISIS...



COMO LO DISEÑARON...

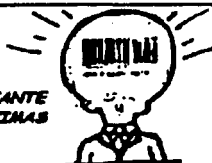


COMO SE IMPLANTÓ...



LO QUE SE NECESITABA...

SEÑOR...



AYÚDAME A TENER MEMORIA DE ELEFANTE QUE AL MENOS RECUERDE LAS 15 ÚLTIMAS VERSIONES DE UN PROGRAMA...

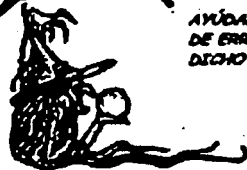
A PODER HACER TODAS LAS COSAS, DESARROLLAR CUATRO PROGRAMAS AL MISMO TIEMPO Y ANALIZAR EL TRUENE DEL SISTEMA...



HAZ QUE NO PIERDA LA PACIENCIA SI TENGO QUE PASARME LAS HORAS HACIENDO UN PROGRAMA QUE FINALMENTE NO SERÁ UTILIZADO...



AYÚDAME A RECORDAR QUE TENGO QUE DOCUMENTAR LOS PROGRAMAS QUE SE VAN A VOLVER A UTILIZAR...



AYÚDAME A PROVEER TODAS LAS CONDICIONES DE ERROR DE UN SISTEMA AÚN CUANDO SE HAYAN DICHO TODAS LAS FUNCIONES...



HAZME SABER SIN QUE NADIE LO DIGA TODO LO QUE QUIEREN QUE HAGA UN PROGRAMA...

AL FINAL DE UN PROYECTO OTÓRNAME EL DON DE ADIVINAR QUE COSA NO ENTENDERÁ EL USUARIO A PESAR DE LA DOCUMENTACIÓN, PARA PODER EXPLICARLO...



PERMÍTEME DESCANSAR COMO PUEDA Y QUE AL DORMIR NO SUEÑE CON LOS PROGRAMAS, A FIN DE NO TENER QUE VENIR A TRABAJAR A CUALQUIER HORA DEL DÍA...



ILUMINA A MI FAMILIA PARA QUE SEPA QUE TENGO HORA DE ENTRADA, MAS NO DE SALIDA...



INCULCA A MIS HIJOS EL RESPETO Y AMOR POR EL PADRE QUE CASI NUNCA LOGRAN VER...

FINALMENTE, DÉJAME APARECER COMO UN GENIO ANTE LOS TRUENES, COMO UN AMBICIOSO ANTE EL TRABAJO Y UN CONFORMISTA ANTE LOS SUELDOS...



BIBLIOGRAFÍA

BIBLIOGRAFÍA

1. Brooks, Frederick; "The mythical man month"; Addison Wesley; 1995.
2. Senn, James A.; "Análisis y Diseño de Sistemas de Información"; 2a Edición; Mc Graw Hill; 2000.
3. Pressman, Roger S.; "Ingeniería de Software"; 4ª Edición; Mc Graw Hill; 2000.
4. Fowler, Martin; "UML Gota a Gota"; 1a edición; Addison Wesley; 1999.
5. Object Management Group; "OMG Unified Modeling Language Specification"; 1999.
6. Malan, Ruth; "Functional Requirements and Use Cases"; Hewlett-Packard Company; 1999.
7. Buckingham, Marcus; Clifton, Donald O.; "Now, discover yours strengths"; The Gallup Organization; 2001.
8. "Curso de Metodología de Planeación y Control de Proyectos"; IDS Comercial; 2000.
9. Publicaciones de diferentes Universidades encontradas en la web:
 - Brackett, Jhon W; "Software Requirements"; Software Engineering Institute Education Program Carnegie Mellon University; 1990.
 - Saiedian, H.; Dale, R.; "Requirements Engineering: Making the connection between the software developer and customer"; Department of Computer Science – University of Nebraska; 1999.
 - Oberg, Roger; Probasco Leslee; Ericsson, Maria; "Applying Requirements Management with Use Cases"; Rational Software Corporation; 1998.
 - Hofmann, Hubert; "Requirements Engineering"; Institute for Informatics – University of Zurich; 1993.
10. Revista "Selecciones"; publicación mensual editada por Reader's Digest México, S.A. de C.V.