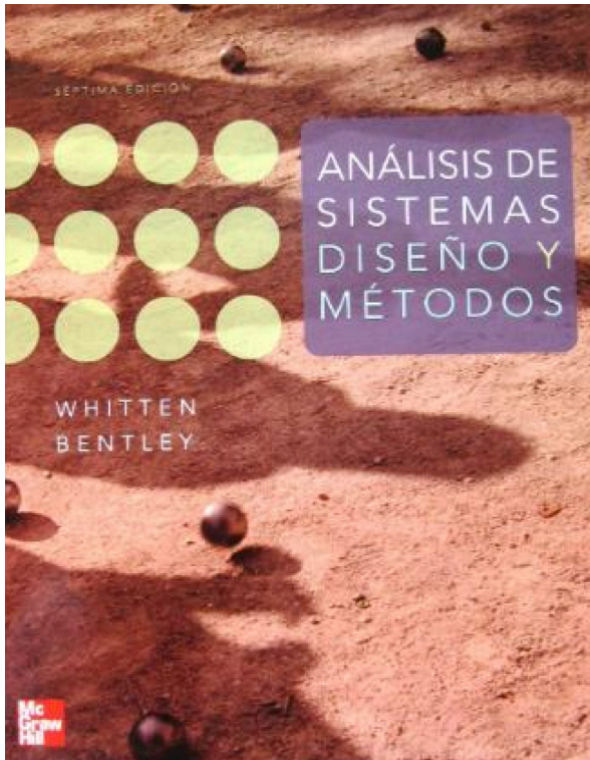
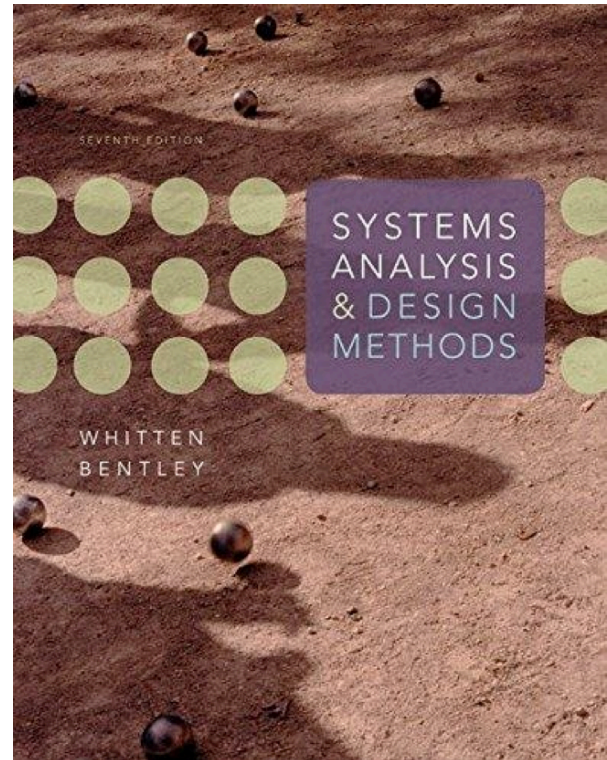


Reseñas Bibliográficas

Whitten, J. L. & D. Bentley. 2008. **Análisis de Sistemas: diseño y métodos**. Séptima edición, McGraw-Hill-Interamericana, México. 569 p. ISBN: 978-970-10-6614-0, Área: Computación.



Whitten, J. L. & D. Bentley. 2005. **System Analysis and Design Methods**. Seventh edition, McGraw-Hill / Irwin, U.S.A. ISBN: 978-0073052335, Área: Computación. 747 p.



La versión en español del libro está organizada en tres partes y 15 capítulos (569 páginas), mientras que la versión en inglés es más completa y consta de cuatro partes y 20 capítulos (747 páginas). El enfoque principal de los autores son las herramientas y técnicas del análisis y diseño de sistemas de información.

Es un libro con una estructura bien definida; sin embargo, algunos usuarios del sitio Amazon opinan que es difícil de leer, por lo que es importante mencionar que es una obra para reforzar las primeras etapas del ciclo de vida del desarrollo de sistemas de

información, asumiendo que el lector posee nociones en diversos tipos de desarrollo, como el desarrollo rápido, el orientado a objetos y el desarrollo basado en prototipos.

Los autores proponen un marco de referencia para el desarrollo de sistemas de información, el cual se va construyendo con cada capítulo y se basa en cuatro perspectivas: los jugadores (equipo), los impulsores de negocios, los impulsores de tecnología y el proceso de desarrollo, partiendo de la premisa que los sistemas están entrelazados y es difícil distinguir entre los sistemas de negocios y

Remedios Fabián Velasco - Instituto de Industrias, Universidad del Mar campus Puerto Escondido. Ciudad Universitaria, Carretera Puerto Escondido - Sola de Vega, Puerto Escondido, 71980, Oaxaca, México.

*Autor de correspondencia: rfabian@zicatela.umar.mx

los sistemas de información que soportan esos negocios.

Cada capítulo se estructura con el marco de referencia, un caso introductorio, el desarrollo de los temas, un mapa de aprendizaje, un resumen, preguntas de aprendizaje, problemas y ejercicios, proyectos e investigación, casos cortos, ejercicios de equipos e individuales y concluye con una sección de lecturas recomendadas. Definitivamente, un texto con valioso material académico.

En la parte uno “El contexto de proyectos de desarrollo de sistemas” se examinan las cuatro perspectivas involucradas y su naturaleza en el desarrollo de sistemas de información. En el primer capítulo, “El contexto de métodos de análisis y diseño de sistemas”, se definen los conceptos de sistemas, los jugadores y los siete tipos de sistemas de información. El énfasis del capítulo es la definición de cada perspectiva. Dentro de la primera perspectiva, considera jugadores a todas las personas involucradas en el desarrollo de un sistema de información; en el marco de referencia muestra la visión de los cuatro tipos de jugadores esenciales, el propietario, los usuarios, los diseñadores y los constructores. Entender que existen diversos involucrados y la visión de cada uno permite entender la expectativa de sus necesidades.

En la versión en inglés, además de lo mencionado, se desglosan las otras tres perspectivas. La perspectiva de los impulsores de negocios son las tendencias que dan origen a un sistema de información, resaltando las siguientes: globalización de la economía, comercio electrónico y negocios electrónicos, seguridad y privacidad, colaboración y asociaciones, gestión de los activos del conocimiento, mejora continua, gestión de la calidad total y rediseño de procesos de negocios. La perspectiva de los impulsores de la tecnología, que influyen en varias vertientes, la tecnología de salida que puede presentar problemas en el manejo de la información y tecnologías emergentes que pueden presentar oportunidades de negocios, en el capítulo se describen las siguientes tecnologías emergentes: redes y el internet, tecnologías inalámbricas

y móviles, tecnologías de objetos, tecnologías colaborativas y aplicaciones empresariales. La cuarta perspectiva es un proceso de desarrollo de sistemas formal, definido como el conjunto de actividades, métodos, mejores prácticas, entregables y herramientas automáticas que los involucrados usan para el desarrollo y mantenimiento de los sistemas de información y software. Se describe el proceso de siete pasos instaurados en cuatro etapas: inicio, análisis, diseño e implementación. Como los sistemas generalmente no son perfectos y los requisitos cambian, el capítulo cierra con la mejora continua y el soporte del sistema.

El capítulo “Componentes de los sistemas de información” se enfoca en el “producto”, el cual es el sistema de información, junto con la federación de sistemas de información necesarios en una organización. Clasifica dichos sistemas en el *front-office* y el *back-office*, dos tipos de sistemas que alimentan a diversas fuentes de datos. Posteriormente, pasa al marco de trabajo de la arquitectura del sistema de información y define tres componentes fundamentales: conocimiento, procesos y comunicaciones. Cada componente se desglosa en dos vertientes: por el lado del negocio y por el lado de la tecnología. A continuación se trata de desglosarlos: El componente conocimiento, por el lado del negocio, es producto de la información y los datos; por el lado de la tecnología, son las bases de datos que soportan el crecimiento y el conocimiento del negocio. En el componente de procesos, por el lado del negocio, la meta es mejorar los procesos del negocio y servicios; por el lado de la tecnología, el software automatiza y respalda dichos procesos y servicios. El componente de comunicaciones, por el lado del negocio, es mejorar la colaboración de las personas; por el lado de la tecnología, son las interfaces que respaldan la colaboración. Los componentes también se representan en el marco de referencia; en la versión en inglés primero desglosa a detalle cada componente antes de construir el marco final, en la versión en español se omite el desglose.

El capítulo “Desarrollo de sistemas de información” antes de analizar el proceso

típico de desarrollo de sistemas, presenta la importancia de la estandarización en los procesos de desarrollo y cómo el modelo de madurez de la capacidad (CMM) ayuda a las organizaciones a madurar el proceso de desarrollo y disminuir la duración y el costo del proyecto. A lo largo del capítulo desglosa cada nivel del CMM, describe la diferencia entre ciclo de vida de un sistema y la metodología de desarrollo de sistemas para posteriormente listar los 10 principios fundamentales para el desarrollo de sistemas.

El proceso de desarrollo de sistemas que se analiza en el texto se conoce como FAST y es una metodología hipotética creada con fines de aprendizaje. FAST se apoya en el marco de referencia denominado PIECES, donde clasifica los seis problemas que dan origen a los proyectos de desarrollo de sistemas: desempeño (*performance*), información (*information*), economía (*economics*), control o seguridad (*control*), eficiencia de las personas y los procesos (*efficiency*) y servicio a clientes, socios, empleados y demás (*service*).

La metodología FAST usa 8 fases para definir mejor los hitos y los productos periódicos. El capítulo muestra una tabla donde se comparan las fases de FAST (Definición de alcance, Análisis del problema, Análisis de requerimientos, Diseño lógico, Análisis de decisión, Diseño físico e integración, Construcción y pruebas e Instalación y entrega) contra las fases clásicas del ciclo de vida de un software (Inicio, Análisis, Diseño e Implementación) y se concluye que FAST muestra un mayor nivel de detalle. Posteriormente ubica la metodología dentro del marco de referencia y muestra un panorama general de cada fase.

Dentro del capítulo se desglosan las actividades transversales del ciclo de vida, aquellas que traslapan varias fases y son importantes para el éxito de cualquier proyecto, tales como la identificación de hechos, el análisis de factibilidad y la administración del proyecto. El proceso de desarrollo de software tiene diversos enfoques, entre los más conocidos se encuentran el secuencial (cascada) y el interactivo (incremental y espiral). Adicionalmente, hay rutas y estrategias

de desarrollo alternativas como las metodologías adaptativas y la metodología controlada por modelos. Esta última se basa en el modelado de sistemas: con los modelos se mejora la comunicación entre los usuarios, analistas, diseñadores y constructores.

Posteriormente, el capítulo muestra las diversas rutas de la metodología FAST por cada estrategia de implementación de sistemas, ya sea desarrollando software, implantando una aplicación comercial (ejemplifica con un Sistema de Planificación de Recursos Empresariales o ERP) e incluso empleando estrategias híbridas. En cada una de las rutas se listan las ventajas y desventajas. El capítulo termina con una breve descripción de las herramientas CASE para automatizar el proceso de desarrollo de software y describe aplicaciones para ambientes de programación y herramientas para la administración de proyectos y de procesos.

En la versión en español concluye la parte uno, pero la versión en inglés continúa con el capítulo "Gestión de proyectos". A decir de los autores, existe una fuerte demanda en administradores de proyectos para el desarrollo de sistemas de información, que tengan conocimiento de los procesos, herramientas y técnicas de la gestión de proyectos, además de experiencia en el desarrollo de sistemas. Esta actividad transversal permite conocer el tiempo límite del proyecto, el presupuesto y las especificaciones y expectativas de los clientes, incluye actividades de gestión y documentación para la mejora continua de la propia gestión de proyectos.

El enfoque de gestión de proyectos es diferente en cada organización, pero para que funcione en el desarrollo de sistemas es prerequisite tener un proceso de desarrollo bien definido (por ejemplo, CMM). Esto mitiga la causa de diversas fallas en los proyectos de desarrollo de sistemas. Se dedica una sección al cuerpo de conocimientos de la gestión de proyectos (PMBok) haciendo énfasis en las competencias necesarias de los gestores, las funciones de los gestores, herramientas y técnicas como las gráficas de Gantt y PERT, y software de gestión de proyectos. Atiende el

ciclo de vida de la gestión de proyectos explicando las ocho actividades que lo conforman.

La parte dos del libro, "Métodos de análisis de sistemas", desglosa las actividades, herramientas y técnicas para analizar los problemas de negocios, la especificación de requerimientos para un sistema de información y la propuesta de solución de negocios y sistemas. El capítulo "Análisis de sistemas" enfatiza los aspectos no técnicos del negocio, bajo la perspectiva de los propietarios y usuarios del sistema. Uno de los primeros componentes es el repositorio, donde se almacenan toda la documentación y entregables de la tarea de análisis de sistemas; permite un repositorio por cada sistema o un repositorio compartido por múltiples proyectos. El análisis de sistemas se fundamenta en resolver un problema y se apoya de distintos enfoques: el enfoque controlado por modelos, que abarca el enfoque tradicional y el enfoque orientado a objetos; el enfoque acelerado del análisis de sistemas, que comprende el descubrimiento de prototipos y el desarrollo de arquitecturas rápidas; los métodos de identificación de requerimientos, que incluyen las técnicas de identificación de hechos y la planeación en conjunto de requerimientos; los métodos del rediseño de procesos de negocio, muy utilizados en las tendencias de gestión de calidad total (TQM) y mejora continua de procesos (CPI); y finalmente, las estrategias de análisis que integran los enfoques de las metodologías ágiles. El resto del capítulo detalla todas las fases del desarrollo de sistemas.

En la fase de definición del alcance, primera fase del desarrollo clásico de sistemas, se desglosan cinco actividades que ayudan a determinar si un proyecto vale la pena de desarrollar, estas son identificar problemas y oportunidades básicos, negociar alcance base, considerar el valor del proyecto base, desarrollar un programa y presupuesto iniciales y comunicar el plan de proyecto.

En la fase de análisis del problema, cuyo objetivo es entender el dominio del problema, se apoya del marco de referencia PIECES para analizar los componentes del problema. Se requiere entender el dominio del problema,

analizar problemas y oportunidades, analizar procesos de negocios, establecer objetivos de mejora del sistema, actualizar o refinar el plan del proyecto y comunicar resultados y recomendaciones.

En la fase de análisis de requerimientos, el único producto a obtener son las necesidades del sistema a desarrollar, dividido en requerimientos funcionales y no funcionales. El formato más simple es la lista original de los objetivos de mejora de sistemas y una sublista de a) entradas, b) procesos, c) salidas y d) datos almacenados. Consta de las siguientes actividades: identificar y expresar los requerimientos del sistema, priorizar los requerimientos de sistema, actualizar o refinar el plan de proyecto y comunicar la definición de requerimientos.

En la fase del diseño lógico se documentan más los requerimientos de negocios mediante modelos de sistemas que ilustran las estructuras de datos, los procesos de negocios, los flujos de datos y las interfaces de usuarios. Con ello se validan los requerimientos previamente establecidos. La fase incluye las siguientes tareas: los requerimientos funcionales de estructura, los requerimientos funcionales del prototipo, validar requerimientos y definir los casos de pruebas de aceptación.

En la fase de análisis de decisión, el propósito es identificar las soluciones candidatas, analizarlas y recomendar un sistema que será diseñado, construido e implementado. Durante esta fase es imperativo identificar las opciones, analizarlas y ofrecer la mejor solución con base en las siguientes cinco actividades: identificar, analizar y comparar soluciones candidatas, actualizar el plan del proyecto y recomendar una solución.

El capítulo "Técnicas de identificación de hechos para la identificación de los requerimientos" atiende el proceso y las técnicas que el analista de sistemas emplea para identificar, analizar y entender los requerimientos del sistema. Para validar los requerimientos, éstos deben ser consistentes, estar completos, ser alcanzables, necesarios, seguros, rastreables y verificables. El proceso de identificación de

requisitos consiste en: identificación y análisis del problema; identificación de requerimientos; análisis y documentación de requerimientos y la gestión de requerimientos. Continúa con la descripción de siete técnicas para identificación de hechos: muestreo de documentación, formatos, archivos y bases de datos existentes; investigación y visitas *in situ*; observación del ambiente de trabajo; cuestionarios; entrevistas; elaboración de prototipos y planificación conjunta de requerimientos. Se aconseja no ir directamente a la entrevista, sino antes analizar y aprender de los documentos existentes, de la observación y de los cuestionarios enfocados a aclarar las dudas.

En el capítulo “Modelando requerimientos de sistemas con casos de uso” se explica que una actividad de vital importancia es la habilidad de elicitar los requerimientos del sistema de manera entendible. En este sentido, el modelado de casos de uso es una técnica para facilitar el desarrollo centrado en el uso. Esta técnica tiene su origen en el modelado orientado a objetos, sin embargo, también se emplea como herramienta de análisis y diseño en enfoques tradicionales, por ejemplo, para modelar datos y procesos; incluso permite tomar decisiones de interfaz y diseño. En términos generales el modelo describe el conjunto de casos de uso (funciones), los actores (usuarios) y las relaciones entre ellos. Comunica en diagramas y descripciones de alto nivel los eventos del negocio que deben ser procesados por el sistema. El proceso consta de identificar actores del negocio, identificar casos de uso requeridos por el sistema, construir el diagrama con el modelo de casos de uso y documentar las narrativas de cada caso. Posteriormente, los casos de uso pasan por una evaluación y transitan a la matriz de prioridades.

En el capítulo “Análisis y modelo de datos” se estudia el modelo de bases de datos a implementar. Existen varias notaciones para especificar dicho modelo, el más frecuente es el diagrama entidad relación (ERD), este describe las entidades y relaciones descritas por los datos. En términos generales, un buen modelo de datos debe ser simple, esencialmente no redundante, adaptable y flexible a

necesidades futuras. Para mejorar el modelo de datos se emplea el proceso de normalización descrito en el capítulo.

El capítulo “Modelado de procesos”, versa sobre la importancia de los procesos como una manera de entender los sistemas y documentar los requerimientos del negocio o el diseño técnico. Hay dos tipos de modelos: el modelo lógico, que describe lo que es el sistema en términos no técnicos, y el modelo físico, que aparte de mostrar lo que es el sistema, también muestra como está implementado física y técnicamente. Se incluyen diversas técnicas de modelado, como diagramas para estructurar un programa, diagrama de flujo lógico, tablas de decisión y diagramas de flujos de datos (DFD), mismos que se describen a lo largo del capítulo, incluyendo las técnicas para los modelos de proceso y la sincronización con los modelos del sistema.

El siguiente capítulo se titula “Análisis orientado a objetos y modelado usando UML”. La importancia del enfoque de programación orientado a objetos (POO) radica en que promueve la reutilización de código y mantiene los costos de programación bajos, ya que es posible colaborar en diversos grupos separados geográficamente y producir un sistema integrado. El enfoque orientado a objetos (OO) requiere del análisis (AOO) y diseño (DOO), surge del modelado orientado a objetos estandarizado en UML (Lenguaje de Modelado Unificado). UML es un conjunto de modelos empleados para describir un sistema de software, como los planos de construcción. A lo largo del capítulo se explican todos los conceptos que involucra este enfoque y aterriza en las actividades del AOO: modelar las funciones del sistema, encontrar e identificar los objetos del negocio y organizar los objetos e identificar sus relaciones. Apoyándose en UML describe los casos de uso y sus narrativas, los diagramas de actividad, los diagramas de secuencia y concluye con los diagramas de clases.

En el capítulo “Análisis de factibilidad y propuesta del sistema” el analista deberá pensar como agente de negocios y responder ¿por qué invertir en un sistema? La factibilidad

es la medida de qué tanto beneficia un sistema de información a la organización y el análisis de factibilidad es el proceso para obtener dicha medida. En el libro se recomiendan seis categorías para las pruebas de factibilidad: la operacional (qué tanto resuelve), la cultural o política (podrá ser aceptada), la técnica (disponibilidad de recursos técnicos y experiencia), de cronograma (tiempo obligado o deseado), la económica (costo-beneficio) y la factibilidad legal (obligaciones contractuales y marco legal). Examina las técnicas de análisis de costo-beneficio, como el valor del dinero en el tiempo, el reembolso, el retorno de inversión, el valor presente neto y concluye con la elaboración y presentación de las propuestas de solución.

En la parte tres, “Métodos de diseño de sistemas”, se abarcan las actividades intermedias del ciclo de vida, herramientas y técnicas del diseño general y detallado con un énfasis particular en el diseño externo (entradas, salidas e interfaces), y el diseño interno (bases de datos e ingeniería de software).

Capítulo “Diseño de sistemas”. Si el análisis de sistemas enfatiza el problema de negocios, al diseño de sistemas le concierne el aspecto técnico o de implementación del sistema. Los analistas de sistemas sirven como facilitadores para esta fase. Los diseños basados en modelos, como el diseño estructurado, la ingeniería de la información o el DOO, enfatizan en los modelos para documentar aspectos de la implementación de nuevos sistemas. En términos generales, el modelo del diseño del sistema llega a ser el plano para construir e implementar el nuevo sistema. Diseñar un sistema para desarrollar “en casa” requiere cinco tareas: diseñar la arquitectura de la aplicación, diseñar la base de datos del sistema, diseñar la interfaz del sistema, especificar los paquetes del diseño y actualizar el plan del proyecto. Diseñar un sistema empleando un software comercial, también requiere obtener requerimientos y saber qué tanto se acopla un sistema comercial al problema y a los otros sistemas que ya existen en la organización.

El capítulo “Modelado y arquitectura de la aplicación” se enfoca exclusivamente en los

planos del sistema de información y las decisiones que conllevan; por ejemplo, si la información debe estar centralizada o distribuida, si requiere comunicarse con otro sistema, si requiere algún ambiente de programación definida. Los planos se integran de varios diagramas, como el de flujo de datos físicos (DFD), que requieren conocer tanto el proceso físico como el flujo de los datos. La arquitectura de tecnología de la información permitirá establecer estándares, decisiones con respecto a la arquitectura de los sistemas, de los datos, de las interfaces e incluso la arquitectura del proceso de desarrollo del software, y a partir de ahí diseñar una estrategia de aplicaciones para la organización.

En el capítulo “Diseño de bases de datos” se explica que las bases de datos se definen de manera separada de los sistemas de información o programas de aplicación, lo que hace que haya flexibilidad en el manejo de los datos. Una parte del capítulo se dedica a repasar los conceptos necesarios para el diseño de las bases de datos. Posteriormente, refiere técnicas como la normalización, que garantizan que el modelo de datos sea simple, no redundante, flexible y adaptable. Después, puntualiza los modelos relacionales de datos y los diseños de archivos tradicionales; actualmente, algunas tendencias de NoSQL apuntan al diseño de archivos, aunque más sofisticados que los descritos en el texto. También abarca el aspecto de integridad referencial y de datos a través del diseño de controles para la base de datos, así como consideraciones de distribución, replicación y prototipos de las bases de datos.

El capítulo “Diseño de salidas y elaboración de prototipos” inicia la etapa del diseño externo. Las salidas presentan la información a los usuarios del sistema, por lo tanto, son el componente más visible. Estas salidas se diseñan construyendo prototipos. Los prototipos son una especie de maquetas que se construyen rápidamente con datos de prueba, no tienen funcionalidad completa, ni características de seguridad o accesos optimizados a los datos. El capítulo trata sobre el diseño de las salidas, el prototipado a través de aplicaciones

específicas y los pasos esenciales para diseñarlas, desde identificar salidas, hasta la validación y pruebas de estas.

Continúa con el capítulo “Diseño de entradas y elaboración de prototipos”. El diseño de entradas tiene el objetivo de capturar y obtener datos en un formato adecuado para la computadora; en los prototipos iniciales no se consideran estos aspectos. Las entradas se clasifican de acuerdo con dos criterios: cómo se adquieren, ingresan y procesan los datos, y qué método o tecnología se emplea para dichas actividades. Los criterios se desglosan en el resto del capítulo.

El diseño externo concluye con el capítulo “Diseño de la interfaz de usuario”. La interfaz de usuario establece el diálogo entre los usuarios y la computadora. En el capítulo se estudia el diseño de la interfaz de usuario examinando los tipos de usuarios, los factores humanos y los lineamientos de ingeniería que afectan el diseño, combinados con la tecnología que se emplea para el diseño de la interfaz de usuario, así como los estilos y consideraciones gráficas.

Con este capítulo concluye la versión en español del libro. En la versión en inglés continúa la parte tres con el capítulo “Modelado y diseño orientado a objetos usando UML”. En el DOO cada pieza de código existe dentro de una clase de objeto. Las aplicaciones trabajan con clases que envían y reciben mensajes de otras clases. El objetivo del DOO es especificar los objetos y mensajes de los sistemas. Los objetos se engloban en clases de objetos y para el diseño se dividen en clases entidad, clases interfaz, clases de control, clases persistentes y clases del sistema, cada tipo con sus propias características. También se diseñan las relaciones entre clases y la visibilidad de estas. La etapa de DOO viene desde el AOO y el refinamiento del modelo de casos de uso para modelar las interacciones, comportamientos y estados de las clases, generando un modelo de clases más extenso. Con los patrones de diseño y la reusabilidad de los objetos se puede mejorar el modelo y completar el diseño con los diagramas de implementación, como los diagramas de paquetes, de componentes y

de despliegue, para describir la arquitectura física del sistema.

La parte cuatro, “Más allá del análisis y el diseño”, versa sobre el soporte del sistema. Es muy común que los nuevos analistas sean responsables de mantener a los sistemas legados, por lo que es importante conocer estos aspectos del ciclo de vida del desarrollo de sistemas.

Capítulo 19 “Construcción e implementación de sistemas”, la construcción implica el desarrollo, instalación y pruebas del sistema, la implementación es el entregable del sistema en producción (operación diaria). En la fase de construcción explica las cuatro tareas: Tarea uno de construcción y pruebas de redes si fuera un requerimiento del análisis del sistema. Tarea dos de construcción y pruebas de las bases de datos, requiere poblar una base de datos inicial y proveer mecanismos de seguridad y desempeño. Tarea tres instalar y probar paquetes del nuevo software y documentar las modificaciones necesarias. Tarea cuatro escribir y probar los nuevos programas construidos, esta actividad incluye desarrollar o refinar los programas requeridos. La fase de implementación describe cinco tareas: Tarea uno lleva a cabo las pruebas del sistema, se hace la prueba final del sistema donde se prueban todos los programas comprometidos del nuevo sistema y se asegura que funcionen juntos. Tarea dos preparar el plan de conversión, de manera detallada facilita una estrategia para pasar el sistema anterior al nuevo sistema. Tarea tres instalar las bases de datos para el sistema y probarlas. Tarea cuatro entrenar usuarios, preparar la documentación adecuada del nuevo sistema y entrenar a los usuarios para el uso exitoso del nuevo sistema. Tarea cinco convertir al nuevo sistema, básicamente es una retroalimentación de todo el proyecto, aprovechar para corregir pequeños errores identificados por los usuarios y comparar los resultados del anterior y el nuevo sistema.

En el capítulo 20 “Operaciones y soporte de sistemas” se detalla el soporte, el cual consiste en el mantenimiento requerido para corregir cualquier error, omisión o nuevo

requerimiento que pueda surgir. Aun en los mejores diseños, es inevitable que pueda surgir algún error causado por una pobre validación, por requerimientos no contemplados o por el mal uso de los programas, entre otros. Por otro lado, el mantenimiento adaptativo es una respuesta a un nuevo problema de negocio o un nuevo requerimiento de información, generalmente se clasifican como reingeniería. El mantenimiento reactivo permite un cambio tecnológico. En algún punto no habrá costo-beneficio de mantenimiento y es natural que los sistemas tiendan a degradarse hasta llegar a la obsolescencia.

Los libros aquí descritos atienden los métodos y técnicas del análisis y diseño dentro del ciclo de vida del desarrollo de un sistema de información. Son libros esenciales para la enseñanza y el aprendizaje del desarrollo de sistemas. Cabe resaltar que la versión en inglés comprende una perspectiva más amplia de todo el ciclo y lo que involucra más allá del desarrollo de un sistema de información. En la versión en español se omite la parte de administración de proyectos y el paradigma orientado a objetos, cinco capítulos faltantes que en esta reseña se describen.