



DESARROLLO DE SOFTWARE 3ER SEMESTRE

PROGRAMA DE LA ASIGNATURA:
ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

UNIDAD 2
REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO
ORIENTADO A OBJETOS

CLAVE:

INGENIERÍA: TSU:
15142313 16142313

CIUDAD DE MÉXICO, NOVIEMBRE DEL 2016

UNIVERSIDAD ABIERTA Y A DISTANCIA DE MÉXICO





UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

ÍNDICE

UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS.....	3
PRESENTACIÓN DE LA UNIDAD.....	3
PROPÓSITO	4
COMPETENCIA ESPECÍFICA	4
2.1. LEVANTAMIENTO DE REQUERIMIENTOS.....	5
2.1.1. INTRODUCCIÓN A LOS REQUERIMIENTOS	12
2.2. DOCUMENTACIÓN PARA LEVANTAMIENTOS Y ESPECIFICACIONES	14
2.2.1. DOCUMENTACIÓN.....	16
2.2.2. ESPECIFICACIONES.....	18
2.3. ESTÁNDARES DE ESPECIFICACIÓN	19
2.3.1. FASES DE LA ESTANDARIZACIÓN	20
2.3.2. ANÁLISIS DE RESTRICCIONES	24
2.4. MODELOS DEL DESARROLLO DE SOFTWARE.....	25
2.4.1. ÁGILES	27
2.4.2. PREDICTIVOS	28
CIERRE DE LA UNIDAD	30
PARA SABER MÁS	31
FUENTES DE CONSULTA.....	32



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

PRESENTACIÓN DE LA UNIDAD

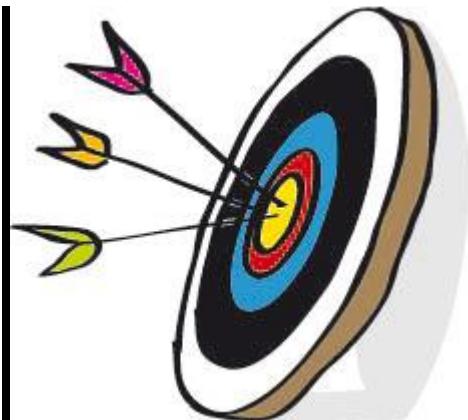
Cuando se va a diseñar un sistema es importante conocer lo que se desea diseñar y esa información la va a proporcionar el cliente; dicha información son los requerimientos. En muchas de las ocasiones, el cliente conoce de manera muy específica lo que desea que su sistema realice, pero en otras, sólo tiene la idea general. Por lo tanto, como diseñadores es necesario recopilar toda esta información; para lograr recabar en forma adecuada los requerimientos del sistema es posible utilizar herramientas como entrevistas, pues éstas permiten realizar documentos que contendrán todas las especificaciones que solicite nuestro cliente.

Para realizar entrevistas es necesario efectuar una estandarización y conocer las restricciones que existen para llevar a cabo lo que el cliente solicita. Una vez que se obtienen las restricciones y se conocen los requerimientos del cliente, se decide el modelo en cual se va a trabajar, ya sea ágil o predictivo.

Por lo tanto, en la presente unidad, podrás conocer todo respecto a los requerimientos para realizar el análisis de un diseño orientado a objetos.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS PROPÓSITO



- Conocer la manera de obtener los requerimientos para lograr un buen análisis de un sistema con programación orientado a objetos, de lo cual se logrará una información simplificada y organizada, para así obtener un buen diseño.

Al término de esta unidad lograrás:

COMPETENCIA ESPECÍFICA



Distinguir los requerimientos del sistema orientado a objetos en su etapa de análisis para definir su diseño mediante técnicas y estándares de especificación.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

2.1. LEVANTAMIENTO DE REQUERIMIENTOS



Requerimientos. Tomada de <https://goo.gl/24ssFP>

Para realizar el diseño de un sistema orientado a objetos, es necesario detallar la definición de un problema en forma de requisitos, lo cual se realiza de manera repetitiva y progresiva. Por un lado, supone la planificación, realización y evaluación de las entrevistas con los clientes y usuarios finales del sistema, que son los portadores de la información necesaria para conocer el problema y definir el proyecto. Por otro lado, supone la identificación y descomposición reiterada (hasta el nivel de detalle que en cada caso sea necesario) de los problemas y necesidades expresados por el cliente y los usuarios, para así redactar un conjunto de requisitos formales. Para obtenerlos, se utiliza la entrevista.

La entrevista es una conversación dirigida por objetivos entre un entrevistador, miembro del equipo de desarrollo, y un entrevistado, que suele ser el cliente o un usuario final (Kendall, 2011).



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

En una entrevista es importante crear desde el principio un clima de cordialidad y confianza, atendiendo siempre a las opiniones del entrevistado. Él es la fuente de información principal; de la relación que se establezca entre los participantes de la entrevista dependerá la facilidad o dificultad para conocer sus necesidades. Es bueno tomar en cuenta que pueden surgir dificultades y malos entendidos, por ejemplo, la resistencia al cambio, usuarios que ven el nuevo sistema como una amenaza para su futuro trabajo, expertos reticentes a compartir conocimientos, etc.

Durante la realización de una entrevista lo habitual es la toma de notas, para redactar más tarde el informe de evaluación de la misma. En otras ocasiones es conveniente grabar audio y video para no perder detalle de lo solicitado, pero para la realización de los mismos es necesario contar con autorización del entrevistado, ya que algunas personas suelen incomodarse; el uso de esta tecnología incrementa costos, porque es necesario tener una inversión con equipo que tenga la capacidad para realizarlo.

Realizar una entrevista es más que una simple conversación, requiere cierto nivel de preparación, conocer un poco lo que el cliente necesita sin entrar en detalles, lo cual puede indagarse desde que se establece la petición del servicio hasta que se establece la cita.

Según el tipo de preguntas, existen diferentes *clases de entrevista*:

- Inductiva: se comienza con preguntas cerradas, para ir pasando, durante la entrevista, a las preguntas abiertas.
- Deductiva: al principio se hacen preguntas abiertas y se va acotando la entrevista con preguntas cada vez más cerradas.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

- Mixta: se utilizan ambos tipos de preguntas indistintamente¹

La realización de una entrevista debe prepararse con anticipación y si es necesario se hará una segunda o tercera entrevista. Para preparar una entrevista puedes apoyarte con técnicas de cuestionarios, ya sea de preguntas abiertas o cerradas.

Las preguntas abiertas dan al cliente la opción de ampliar su respuesta y no basta con un simple sí o un no, a diferencia de las preguntas cerradas.

Algunos ejemplos de preguntas abiertas son los siguientes:

- ¿Cuál es el objetivo del sistema?
- ¿Quiénes son los usuarios? ¿A qué población va dirigido el sistema?
- ¿En qué momentos se va a utilizar? ¿Se usará periódicamente, en diversos ciclos?
- ¿Qué le parece nuestra propuesta frente a otras que ha recibido?
- ¿Qué servicios le gustaría recibir de su sistema?

Para poder formular preguntas cerradas es necesario anticipar las posibles alternativas de respuesta. Algunos ejemplos de preguntas cerradas:

- ¿Firmamos el contrato?
- ¿Le gusta nuestro producto?

A partir de las entrevistas y reuniones con el cliente, se define el problema; es decir, se obtiene la *especificación de requisitos*. En ella se describe lo que el

¹ Si deseas ampliar conocimientos de entrevistas desde cómo prepararla, consulta la página 89 del libro de Kendal (2005); la bibliografía completa la encontrarás al final de este documento.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

sistema nuevo debe realizar, además de que se ha averiguado, mediante un estudio de viabilidad, si el sistema se puede desarrollar o no.

Una vez que ya conoces toda la información de lo que tu cliente demanda, te presentamos una serie de técnicas que puedes emplear para tu especificación de requisitos:

- **Organizar:** Bajar todas las preguntas a un formato, por ejemplo, de requerimientos, en donde coloques el nombre, la descripción y una clave de acceso rápido.
- **Limpiar:** dejar en un segundo plano toda aquella información que te proporcionó el cliente que no tenga ningún funcionamiento o afecte tu diseño.
- **Reelaborar:** si después de organizar tu información te surgieron algunas dudas; en muchas ocasiones es necesaria una segunda entrevista con preguntas específicas.
- **Separar:** una vez que ya conoces todos los requerimientos del cliente y los tienes organizados en tu formato, separa aquellos que se refieran a la misma parte o módulo del sistema.

Vamos a suponer que el cliente está solicitando un sistema para controlar las ventas y el almacén de su producto vía telefónica; por lo tanto, concertó una cita para hablar de detalles y los requerimientos. Entonces, se debe comenzar planteando posibles preguntas (abiertas o cerradas) sobre detalles más



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

específicos del diseño; por ejemplo: ¿qué datos requiere que el sistema tenga de cada producto, aparte de clave de descripción y precio? Para el control del almacén, ¿qué datos de existencia requiere de cada producto?

Una vez que se recopile la información, debes vaciarla en un formato de requerimientos y **organizar** lo solicitado; es un hecho que mucha de esa información que se recabó no se va a ocupar, entonces habrá que **limpiarla** y no considerarla; en caso de que algo no se haya preguntado o por si surgen más dudas es necesario **reelaborar**, y por ultimo **separar** en los posibles módulos (por ejemplo: ventas y almacén).

Esta información se apoya en tres pilares o consta de tres partes principales:

- Procesos.
- Datos.
- Eventos.

Los **procesos** dicen qué hay que hacer. Los **datos** indican con qué hay que hacerlo. Los **eventos** muestran cuándo hay que hacerlo. Por lo tanto, estos tres puntos provocan la ejecución de la operación adecuada en cada momento. Son los que activan la aplicación, la ponen en marcha, y propagan esa activación a lo largo de la aplicación, desencadenando la ejecución de otras operaciones. Los eventos llevan el control de la aplicación introduciendo el dinamismo necesario para su ejecución. Los eventos tienen mucha relación con la interfaz de la aplicación, porque a través de la interfaz se introducen los eventos.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

Los tres pilares son complementarios, no tienen más importancia uno que otro, se necesitan los tres. En algunos sistemas predomina más uno que otro, pero siempre están presentes.

Para especificar cada uno de los tres pilares se utilizan modelos. Un modelo es una representación abstracta de la realidad. Por tanto, como resultado del análisis se obtendrán:

- **Modelo de procesos:** recoge funciones, actividades, tareas, acciones, debe realizar la aplicación y manejar los datos. “Un modelo de procesos de software es una representación abstracta de un proceso del software, cada modelo de proceso representa un proceso desde una perspectiva en particular y así proporciona sólo información parcial sobre ese proceso” (Sommerville, 2006).

Existen muchos modelos que se enfocan al proceso (algunos de ellos los viste en la unidad 1 de esta materia): en espiral, en cascada o incremental, que se basan en la representación de los procesos a diseñar.

- **Modelo de datos:** como el mismo nombre del modelo lo dice, están enfocados a la representación de los datos, los cuales se manejan en las materias de bases de datos; el concepto es el siguiente:

Un modelo de datos es una serie de conceptos que puede utilizarse para describir un conjunto de *datos* y *operaciones* para manipular los mismos. Cuando un modelo de datos describe un conjunto de datos, se está describiendo un conjunto de conceptos de una



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

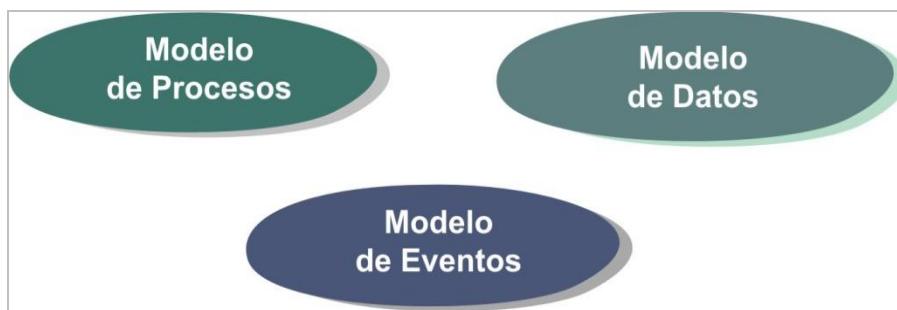
realidad determinada y se llama modelo conceptual, los conceptos de un modelo de datos.

Hay dos tipos de modelos de datos: modelos conceptuales usados en el diseño de bases de datos y modelos lógicos apoyados por los sistemas de gestión de bases de datos (Ceri, 1994).

- **Modelo de eventos:** indica en qué momento debe ejecutarse cada acción. Para construir cada modelo hay diferentes técnicas, algunas son complementarias.

Los modelos enfocados al evento se enfocan en lo que sucede en cada situación. Un ejemplo sería el despliegue de un menú.

Algunos de estos modelos son los desarrollados con técnicas UML, el cual se abordará de manera ampliada en la unidad 4 de esta materia.



Modelos para diseño de aplicaciones orientadas a objetos



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

En la fase de análisis se pretende que los modelos (de procesos, de datos y de eventos) estén lo suficientemente detallados sin llegar a descender al diseño. El análisis tiene por objetivo entender el problema: las necesidades del cliente, las restricciones que se deben cumplir.

El diseño pretende obtener una solución óptima que cumpla todos los requisitos. Se orienta hacia la máquina, centrándose en cómo crear un sistema software que reúna todas las necesidades y cumpla todas las restricciones.

El levantamiento de requerimientos forma parte fundamental para entender las necesidades del cliente y a partir de ello se pueden generar modelos que nos muestren, de manera abstracta, lo que deseamos, previo al trabajo de diseño, además de que nos sirve como una herramienta para la etapa de análisis.

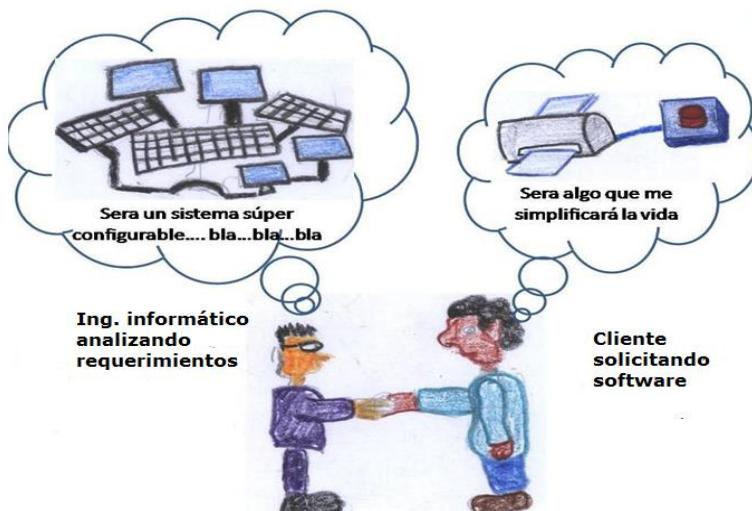
2.1.1. INTRODUCCIÓN A LOS REQUERIMIENTOS

La especificación de requisitos es la primera fase de todo ciclo de vida o metodología de desarrollo de un proyecto informático. Dos son sus objetivos principales:

- Identificar las necesidades del cliente, es decir, conocer y definir el problema.
- Realizar un estudio de viabilidad económica, técnica y legal, a partir del cual se pueda decidir sobre la continuación del proyecto, teniendo en cuenta las alternativas de construcción del mismo que se nos planteen.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS



Necesidades del cliente. Tomada de <http://goo.gl/jbcM3f>

Estos dos objetivos principales se concretan en una serie de acciones a realizar, unas técnicas a aplicar y unos productos a obtener. Resulta obvio (en cualquier contexto, no sólo en el terreno informático) que un primer paso necesario para solucionar un problema es tenerlo definido correcta y detalladamente. Es fundamental centrar la necesidad del cliente para poder definir el problema que se va a resolver, tratando de dejar claro lo que queda dentro y fuera del alcance del mismo. En este momento se está hablando de la definición, desde el punto de vista puramente técnico del proyecto. Un aspecto importante es la identificación de los tipos de usuarios potenciales que tendrá el sistema.

Descubrir tres necesidades del cliente:

- *Ser cliente*
- *Ponerse en comunicación con los clientes*
- *Simular el uso por los clientes*



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

2.2. DOCUMENTACIÓN PARA LEVANTAMIENTOS Y ESPECIFICACIONES

Una vez que se conocen los requerimientos del cliente, el paso a seguir es la manipulación de la información, comenzar con el llenado de formatos o documentos, clarificando las ideas que el cliente expresó.

El siguiente paso es analizar lo que se está solicitando y solicitar una nueva entrevista para firma de documentos de aceptación de requerimientos; el cliente debe verificar que lo redactado en los documentos es lo que solicitó.



Documentación. Tomada de <http://goo.gl/jbcM3f>

A continuación se presenta una idea de cómo se puede organizar y documentar dicha información.

Realizar un documento con el siguiente contenido:

1. Índice:

El documento debe contener un índice general con la ubicación de los temas a desarrollar.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

2. Descripción del proyecto:

Es importante y aconsejable que en un párrafo describas lo que el proyecto o sistema debe realizar y verificar que lo que el cliente solicitó se haya entendido para que posteriormente se pueda diseñar el sistema.

3. Lista de usuarios:

Hacer una lista de las personas que tendrán que manipular el sistema a desarrollar, pero principalmente describir qué es lo que cada una de ellas tendrá derecho a realizar y hasta dónde podrá consultar, es decir, formar una lista de permisos de usuarios.

4. Requerimientos hechos por el cliente:

En el tema anterior, se mencionó el uso de un formato para poder organizar los requerimientos, por lo tanto, es necesario agregar si el requerimiento es funcional o no; por ejemplo: con respecto a lo funcional, agregar un botón que haga el respaldo de la información; un requerimiento no funcional sería el color específico de cierta ventana, que si bien es cierto no afecta el desarrollo del sistema, es el cliente el que lo solicita.²

5. Requerimientos del sistema:

Es importante investigar sobre lo que tu cliente está solicitando, qué requisitos de hardware y de software debe contener como mínimo el equipo en el que va a ejecutarse el diseño que se realice.

² Si deseas conocer más sobre requerimientos funcionales y no funcionales, te invito consultar el capítulo 6 del libro de Sommerville (2005); la bibliografía completa la encontrarás al final de este documento.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

6. Otros:

- Análisis costo-beneficio: juzgar qué tan conveniente es realizar un análisis costo-beneficio previo al diseño del sistema y mostrarlo al cliente.
- Restricciones: dentro de la categoría de otros, insertar en el documento principal las restricciones que puede tener el sistema a desarrollar.
- Contrato: establecer un contrato entre cliente y servidor puede y debe ser un documento importante.
- Cronograma o calendario de actividades.
- Diagramas UML.

Presentar un documento con toda la información organizada de esta manera puede ser de gran ayuda para comenzar a darle forma al desarrollo de un sistema, y por ende, forma parte del análisis del mismo; dicho orden de integración de documentos sólo es una sugerencia de acomodo de información, deja al analista organizar libremente según sus necesidades.

2.2.1. DOCUMENTACIÓN

La documentación de los requerimientos es una parte importante durante el análisis. En la práctica es común describirlos en un documento llamado *especificación de requerimientos* del software; su principal objetivo es comunicarlos, ordenarlos y clasificarlos de forma efectiva.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

Una vez que ya se conocen los requerimientos del cliente que se ordenaron en un documento, es importante saber colocar en cada uno de estos documentos la información requerida, por ejemplo:

Lista de usuarios: entregar un formato que contenga la información de los usuarios que van a entrarán al sistema, puede simplemente ser una tabla que contenga las siguientes columnas: tipo de usuario, descripción, permisos y accesos.

Requerimientos: una forma muy práctica de ordenar los requerimientos que tu cliente te proporcionó, puedes encontrarla en una tabla cuyas columnas se compongan de lo siguiente: clave de requerimiento, nombre del requerimiento, descripción del requerimiento.

Puedes primero colocar todos los requerimientos funcionales y después los requerimientos no funcionales.

Análisis costo beneficio: realizar este tipo de análisis para que le muestres al cliente el beneficio que puede obtener y bajo qué costo.

- **Restricciones:** en un documento describir y ordenar todo lo que consideres puede ser un impedimento para realizar lo que el cliente solicita.
- **Contrato:** un contrato es el asentamiento físico de lo que el cliente solicita, estipula lo que va a entregar el proveedor del servicio y las condiciones monetarias, formas de pago, fechas de entrega, etc.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

- **Cronograma o calendario de actividades:** se trata de representar, de manera gráfica, el tiempo que llevará la elaboración de cada uno de los procesos o requerimientos a cubrir y te puedes apoyar con herramientas como Project³.
- **Diagramas UML:** serán descritos en el último capítulo de esta materia, por lo tanto, ahora sólo se enlistarán:
 - Casos de uso
 - Escenarios del caso de uso
 - Diagramas de actividades
 - Diagrama secuencial
 - Diagrama de clase
 - Diagrama de gráfico de estado

2.2.2. ESPECIFICACIONES

Expresar las características esenciales de un objeto nos permite distinguir a uno de otro. Aparte de distinguir los objetos, las especificaciones también proveen límites conceptuales, permitiendo que se disponga de las características de un objeto que se necesite.

El objetivo principal de las especificaciones es entregar una relación de requerimientos que ayuden a determinar de forma completa y correcta el diseño orientado a objetos. La especificación es el documento final que contiene

³ En la sección *Para saber más* encontrarás una liga que te servirá de guía.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

toda la información de manera completa: diagramas de información y tablas de requerimientos.

La especificación puede realizarse con base en estándares internacionales, con lo cual se desarrollarán softwares con altos niveles de calidad. Existen organismos dedicados a establecer normas y a la vez se encargan de evaluar que dichas normas se cumplan.

Existen estándares relacionados con el desarrollo de software, tales como:

ISO/IEC 25000: en lo que se refiere a calidad del producto la norma ISO/IEC 25000 proporciona una guía para el uso de las nuevas series de estándares internacionales, llamados *Requisitos y Evaluación de Calidad de Productos de Software* (SQuaRE); constituyen una serie de normas basadas en la ISO 9126 y en la ISO 14598 (para la evaluación del software), y su objetivo principal es guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad; establece criterios para la especificación de requisitos de calidad de productos software, sus métricas y su evaluación (Norma ISO/IEC 25000:2005).

El **CMM-CMMI** es un modelo de calidad del software que muestra una tabla de niveles de madurez de las empresas, en cuanto a los procesos que tienen para producir software.

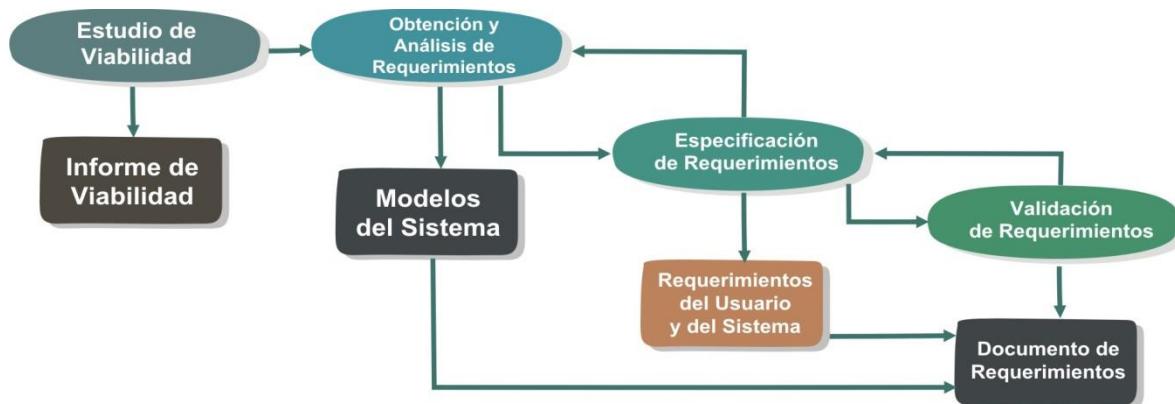
2.3. ESTÁNDARES DE ESPECIFICACIÓN



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

Las especificaciones del software determinan el proceso de comprensión y definición sobre los servicios que se requieren del sistema y de identificación de las restricciones de funcionamiento y desarrollo del mismo. La ingeniería de requerimientos es un proceso crítico en el proceso del software, los errores en esta etapa originan problemas posteriores en el diseño e implementación del sistema.

En la siguiente figura se muestra el proceso de ingeniería de requerimientos. Éste conduce a la producción de un documento de requerimientos, que es la especificación del sistema. Normalmente en este documento los requerimientos se presentan en dos niveles de detalle. Los usuarios finales y los clientes necesitan una declaración de alto nivel de los requerimientos, mientras que los desarrolladores del sistema necesitan una especificación más detallada de éste.



Proceso de ingeniería de requerimientos. Tomada de Somerville (2006)

2.3.1. FASES DE LA ESTANDARIZACIÓN



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

Existen cuatro fases principales en el proceso de estándares de ingeniería de requerimientos:

1. **Estudio de viabilidad.** Se estima si las necesidades del usuario se pueden satisfacer con las tecnologías actuales de software y hardware. El estudio analiza si el sistema propuesto será rentable desde un punto de vista de negocios y si se puede desarrollar dentro de las restricciones de presupuesto existentes. Este estudio debe ser relativamente económico de elaborar. El resultado debe informar si se va a continuar con un análisis más detallado.

Sommerville (2005) define el estudio de viabilidad como un estudio corto y orientado a resolver las siguientes preguntas:

1. ¿El sistema contribuye a los objetivos generales de la organización o empresa?
2. ¿El sistema se puede implantar utilizando tecnología actual dentro de las restricciones de tiempo y presupuesto?
3. ¿El sistema puede integrarse a otros sistemas existentes en la empresa?

Para ayudar a responder las preguntas del estudio de viabilidad, se tienen algunos ejemplos de preguntas posibles:

- ¿Cómo se las arreglaría la organización o empresa si no se implantara el sistema?
- ¿Cuáles son los problemas con los procesos actuales y cómo ayudaría un sistema nuevo a subsanarlos?
- ¿Cuál es la contribución directa que hará el sistema a los objetivos y requerimientos del negocio?



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

- ¿Se puede obtener y transferir la información a otros sistemas de la organización?
- ¿El sistema requiere tecnología que no se ha utilizado previamente en la organización?
- ¿A qué debe ayudar el sistema y a qué no necesita ayudar?

El estudio de viabilidad no debe requerir más de dos o tres semanas. El resultado de este estudio es un informe que recomienda si vale o no la pena seguir con la ingeniería de requerimientos y el proceso de desarrollo del sistema. En el informe se pueden proponer cambios en el alcance, el presupuesto o sugerir requerimientos adicionales de alto nivel. (Gómez, 2011).

2. Obtención y análisis de requerimientos. Es el proceso de obtener los requerimientos del sistema por medio de la observación de los sistemas existentes: discusiones con los usuarios potenciales y proveedores, el análisis de tareas, etc. Esto puede implicar el desarrollo de uno o más modelos y prototipos del sistema que ayudarán al analista a comprender el sistema a especificar.

3. Especificación de requerimientos. Es la actividad de traducir la información recopilada durante la actividad de análisis en un documento que define un conjunto de requerimientos. En este documento se pueden incluir dos tipos de requerimientos: los *requerimientos del usuario*, que son declaraciones abstractas de los requerimientos del cliente y del usuario final del sistema, y los *requerimientos del sistema*, que son una descripción más detallada de la funcionalidad a proporcionar.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

4. **Validación de requerimientos.** Esta actividad comprueba la veracidad, consistencia y completitud de los requerimientos. Durante este proceso, inevitablemente se descubren errores en el documento de requerimientos. Se debe modificar entonces para corregir estos problemas.

Por supuesto, las actividades en el proceso de requerimientos no se llevan a cabo de forma estrictamente secuencial. El análisis de requerimientos continúa durante la definición y especificación, y a lo largo del proceso surgen nuevos requerimientos. Por lo tanto, las actividades de análisis, definición y especificación se entrelazan. En los métodos ágiles como la programación extrema, los requerimientos se desarrollan de forma incremental, conforme a las prioridades del usuario, y la obtención de requerimientos viene de los usuarios que forman parte del equipo de desarrollo.

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

- ¿Están incluidas todas las funciones requeridas por el cliente? (completaa)
- ¿Existen conflictos en los requerimientos? (consistencia)
- ¿Alguno de los requerimientos tienen más de una interpretación? (sin ambigüedad)
- ¿Está cada requerimiento claramente representado? (entendible)
- ¿Pueden los requerimientos ser implementados con la tecnología y el presupuesto disponible? (factible)
- ¿Está la SRS escrita en un lenguaje apropiado? (claridad)
- ¿Existe facilidad para hacer cambios en los requerimientos? (modificabilidad)
- ¿Está claramente definido el origen de cada requisito? (rastreable)



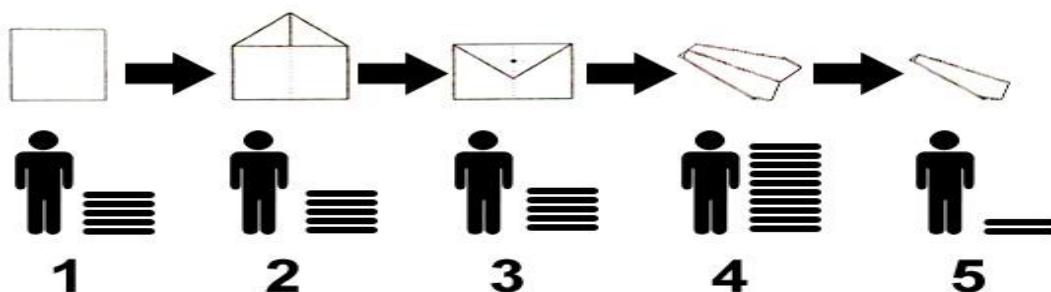
UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

- ¿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.

2.3.2. ANÁLISIS DE RESTRICCIONES

Las restricciones son relaciones entre entidades de un modelo de objetos; el término de entidad incluye los objetos, clases, atributos, enlaces y asociaciones. Una restricción reduce los valores que una entidad puede tomar.



Análisis de restricciones. Tomada de <http://goo.gl/v3mUt9>

- **Restricciones entre objetos.** Determina el estado en el cual los objetos se diferencian uno del otro; ejemplo: horario de entrada de un empleado de oficina no puede ser después de las 9:00 de la mañana, suponiendo que el horario de entrada al trabajo sea a esa hora.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

- **Restricciones entre atributos de un objeto.** Determina sus atributos específicos; ejemplo: el objeto “alumno” sólo debe tener como atributos, nombre completo y no apellido paterno, apellido materno y nombre.
- **Restricciones sobre un objeto a lo largo del tiempo.** Determina el estado del objeto; se especifica que nunca debe de cambiar su estado; ejemplo: el objeto “alumno” no puede aumentar su número de control.

Las restricciones simples pueden situarse en el modelo de objetos; restricciones complejas aparecerán en el modelo funcional. Las restricciones no tienen por qué aparecer inicialmente en el modelo de objetos, éstas irán añadiéndose conforme se vayan concretando en la definición del modelo.

2.4. MODELOS DEL DESARROLLO DE SOFTWARE

Las metodologías se centran en una serie de combinaciones de los modelos de procesos genéricos (cascada, evolutivo, incremental, etc.). Adicionalmente una metodología debe definir con precisión los roles y actividades involucradas, junto con prácticas y guías de adaptación.

Si se recuerda la asignatura Programación orientada a objetos, se explica en ésta el uso de métodos, los cuales son las actividades a realizar por objetos; de igual manera se aplica lo anterior como analogía para los modelos de desarrollo de software, en donde un método es un proceso para hacer una metodología.

La comparación y/o clasificación de metodologías no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalle,



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

información disponible y alcance de cada una de ellas. A grandes rasgos, si tomamos como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño, podemos clasificar las metodologías en dos grupos: *metodologías estructuradas* y *metodologías orientadas a objetos*. Por otra parte, considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de *metodologías tradicionales* (o peyorativamente denominada *metodologías pesadas*, o peso pesado). Otras metodologías, denominadas metodologías ágiles, están más orientadas a la generación de códigos con ciclos muy cortos de desarrollo; se dirigen a equipos de desarrollo pequeños; hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

En 1995 Booch y Rumbaugh proponen el método unificado con la ambiciosa idea de conseguir una unificación de sus métodos y notaciones, que posteriormente se reorienta a un objetivo más modesto, para dar lugar al Unified Modeling Language (UML), la notación OO más popular en la actualidad (Abriz, Huitrón y Ramírez, 2012, p. 90).

Algunos métodos orientados a objetos con notaciones predecesoras de UML son OOAD (Booch), OOSE (Jacobson), Coad & Yourdon, Shaler & Mellor y OMT (Rumbaugh).

Algunas metodologías orientadas a objetos que utilizan la notación UML son Rational Unified Process (RUP), OPEN, MÉTRICA (que también soporta la notación estructurada).



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

2.4.1. ÁGILES

Entre las nuevas técnicas para aplicar las prácticas esenciales de la programación extrema y los métodos ágiles para desarrollar sistemas orientados a objetos se encuentra la metodología ágil. En ésta el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, es bien documentado), y adaptable (permite realizar cambios de último momento).

Según Kendall (2011), el modelado ágil también abarca un conjunto de principios esenciales. Además de los principios esenciales de la programación extrema, el modelado ágil agrega principios tales como "modelar con un propósito", "el software es su meta principal" y "viajar con poco equipaje" (es una forma de decir que poca documentación es suficiente).

Entre las metodologías ágiles identificadas están:

- Extreme Programming
- Scrum
- Familia de Metodologías Crystal
- Feature Driven Development
- Proceso Unificado Rational, una configuración ágil
- Dynamic Systems Development Method
- Adaptive Software Development
- Open Source Software Development



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

Un modelo ágil permite mostrar un resultado de manera rápida al cliente; por consecuencia se dejan los formatos y la documentación para una última entrega, enfocando el 100% del tiempo en el desarrollo del sistema, mostrando al cliente avances sobre el producto mismo.

2.4.2. PREDICTIVOS

Las *metodologías predictivas* son conocidas como metodologías no ágiles y están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también *metodologías tradicionales o clásicas*, en las cuales se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

Todas las propuestas metodológicas antes indicadas pueden considerarse como metodologías tradicionales por el especial énfasis que presentan en cuanto a su adaptación a las condiciones del proyecto (mediante su configuración previa a aplicarse); realizando una configuración adecuada, podrían considerarse ágiles. La inspiración usual para las metodologías ha sido disciplina como las ingenierías civiles o mecánicas.

Lo que se ha venido desarrollando a lo largo de este documento con respecto de lo importante de la documentación y lo que se abordará en la unidad 4 con los modelos UML que se expresan a través de dibujos, marca la importancia de llevar el control a través de documentos, seguidos de los diagramas o gráficos, los cuales en ocasiones se vuelven una actividad a la que hay que invertir tiempo sobre todo en la fase de dibujo de diagramas UML.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

Como los dibujos especifican las piezas y cómo deben unirse, actúan como los fundamentos de un plan de construcción detallado. Dicho plan define las tareas que necesitan hacerse y las dependencias que existen entre estas tareas. Esto permite un plan de trabajo y un presupuesto de construcción razonablemente predecibles. También dice en detalle cómo deben hacer su trabajo las personas que participan en la construcción. Esto permite que ésta requiera menos pericia intelectual, aunque se necesita a menudo mucha habilidad manual.

Así el acercamiento de muchas metodologías se trata tener un plan de trabajo predecible que pueda usar gente del más bajo nivel jerárquico. Para hacerlo, debemos separar el plan de la construcción. Por consiguiente necesitamos entender cómo hacer el diseño de software de modo que la construcción pueda ser sencilla una vez que el plan esté hecho.

¿Qué forma toma este plan? Para muchos, éste es el papel de notaciones de diseño como el UML (Lenguaje de Modelado Unificado). Si podemos hacer todas las decisiones significativas usando UML, podemos armar un plan de construcción y entonces podemos dar planes a los programadores como una actividad de construcción.

Pero aquí surgen preguntas cruciales: ¿es posible armar un plan que sea capaz de convertir el código en una actividad de construcción predecible? Y en tal caso, ¿es la construcción suficientemente grande en costo y tiempo para hacer valer la pena este enfoque?

Todo esto trae a la mente más preguntas. La primera es la cuestión de cuán difícil es conseguir un diseño UML en un estado que pueda entregarse a los programadores. El problema con un diseño tipo UML es que puede parecer muy



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

bueno en el papel, pero resultar seriamente fallido a la hora de la programación. La única verificación que podemos hacer con los diagramas UML es la revisión cuidadosa. Los errores en el diseño sólo se descubren durante la codificación y pruebas, incluso los diseñadores experimentados se sorprenden a menudo cuando convierten dichos diseños en software.

Otro problema es el costo comparativo. Cuando se construye un puente, el costo del esfuerzo en el plan es aproximadamente un 10% del total, la construcción es el resto. En software la cantidad de tiempo gastada codificando es mucho menor. Se sugiere que para un proyecto grande, sólo 15% del proyecto sea fundamentadas en código y pruebas unitarias, una inversión casi perfecta de las proporciones de la construcción del puente. Aun cuando se consideren las pruebas parte de la construcción, el plan es todavía 50% del total. Esto genera una pregunta importante sobre la naturaleza del diseño en software comparado con su papel en otras ramas de la ingeniería.

CIERRE DE LA UNIDAD

Has concluido la unidad 2 del curso; a lo largo de ésta trabajaste con la documentación de requerimientos para el análisis orientado a objetos, comenzando con la parte de levantamiento de requerimientos, que incluye el describir cuáles son los requerimientos y qué actividades necesitas realizar para el levantamiento de los mismos.

También identificaste cuál es la documentación para el levantamiento y qué especificaciones debe cumplir considerando sus estándares, divididos en sus



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

fases y análisis de restricciones. Por último en esta unidad distinguiste qué modelos del desarrollo de software se manejan y si son ágiles o predictivos.

Es aconsejable que revises nuevamente la unidad en caso de que los temas que se acaban de mencionar no te sean familiares o no los recuerdes; de no ser este tú caso, ya estás preparado(a) para seguir con la unidad 3, en donde continuarás con la metodologías de diseño para la generación de sistemas orientados a objetos, tales como Bosh, OOC, OMT y UML.

PARA SABER MÁS



Ideas .Tomada de <http://goo.gl/z2W5z>

Consulta el documento *Crear un plan en Project en cinco pasos sencillos* (2003), en él encontrarás un manual de uso del software Microsoft Project. Disponible en: <http://office.microsoft.com/es-hn/project-help/crear-un-plan-en-project-en-cinco-pasos-sencillos-HA001136153.aspx>



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

FUENTES DE CONSULTA

- Abriz, A., Huitrón, J. M. y Ramírez, C. (2012). *Sistema financiero para empresa distribuidora de gas*. México: UNAM-Facultad de Ingeniería.
- Booch, G. (1996). *Análisis y Diseño Orientado a Objetos con aplicaciones*. México: Pearson Educación.
- Ceri, B. (1994). *Diseño conceptual de bases de datos*. EUA: Addison-Wesley.
- Cueva, J. (2005). Análisis orientado a objetos. En *El proceso de desarrollo de software* (pp. 6-15). Recuperado el 22 de julio de 2011 de <http://www.di.uniovi.es/~cernuda/pfc/ao0.pdf>
- Cueva, J. (2005) *Ingeniería del software*. Madrid: Pearson Educación.
- Fowler, M. (2003). La nueva metodología. Sierra, A. (trad.). *Programación Extrema*. Recuperado el 22 de julio de 2011 de <http://www.programacionextrema.org/articulos/newMethodology.es.html>
- García, S. y Morales, E. (2003). *Desarrollo de aplicaciones informáticas. Análisis y diseño detallado de aplicaciones informáticas de gestión*. México: Thompson.



UNIDAD 2. REQUERIMIENTOS PARA EL ANÁLISIS DEL DISEÑO ORIENTADO A OBJETOS

- Gómez, M. (2011). *Notas del curso Análisis de requerimientos*. México: UAM-Unidad Cuajimalpa-Departamento de Matemáticas Aplicadas y Sistemas-División de Ciencias Naturales e Ingeniería.
- Kendall, E. (2011). *Análisis y diseño de sistemas*. México: Pearson Educación.
- Letelier, P. y Penadés, M. (s. f.). Extreme Programming (XP). En *Metodologías ágiles para el desarrollo de software*. Valencia: Universidad Politécnica de Valencia. Recuperado el 22 de julio de 2011 de <http://www.willydev.net/descargas/masyxp.pdf>
- Norma ISO/IEC 25000:2005. (s. f.). Recuperado el 03 de julio de 2012 de <http://iso25000.com/index.php/25000.html>
- Somerville, I. (2006). *Ingeniería del software*. Madrid: Pearson Educación.
- World Lingo. (2011). Análisis de requisitos. *World Lingo Translations LLC*. Recuperado el 22 de julio de 2011 de http://www.worldlingo.com/ma/enwiki/es/Requirements_analysis