Metodología Orientada a Objetos (OMT). Rumbaugh

El análisis y diseño orientado a objetos constituye una nueva forma de pensar acerca de problemas empleando modelos que son útiles para comunicarse con expertos en esa aplicación, modelar empresas, preparar documentación, diseñar programas y bases de datos.

> Un modelo es una abstracción de algo, cuyo objetivo es comprenderlo antes de construirlo. Dado que los modelos omiten los detalles no esenciales, es más sencillo manipularlos que manipular la entidad original. La abstracción es una capacidad humana fundamental que nos permite enfrentarnos a la complejidad. Los ingenieros, artistas y artesanos han estado construyendo modelos durante miles de años para probar los diseños antes de ejecutarlos. El desarrollo de sistemas hardware y software no es una excepción. Para construir sistemas complejos, el desarrollador debe abstraer distintas vistas del sistema, construir modelos utilizando notaciones precisas, verificar que los modelos satisfacen los requisitos del sistema y añadir, gradualmente, detalles para trasformar los modelos en una implementación. (Rumbaugh, 1996)

La esencia del análisis y diseño orientado a objetos es la identificación y organización de conceptos del dominio de la aplicación, y no de su presentación final en un lenguaje de programación, es decir, es un proceso conceptual independiente de sí el lenguaje es orientado a objetos.

El uso del análisis y diseño orientado a objetos puede facilitar mucho la creación de prototipos, y las técnicas de desarrollo evolutivo de software. Los objetos son inherentemente reutilizables, y se puede crear un catálogo de objetos que podemos usar en sucesivas aplicaciones. De esta forma, podemos obtener rápidamente un prototipo del sistema, que pueda ser evaluado por el cliente, a partir de objetos analizables, diseñados e implementados en aplicaciones anteriores. Y lo que es más importante, dada la facilidad de reutilización de estos objetos, el prototipo puede ir evolucionado hacia convertirse en el sistema final, según vamos refinado los objetos de acuerdo a un proceso de especificación incremental.

Cabe resaltar que los sistemas construidos hoy en día son más complejos que los sistemas construidos en los años 70s y 80s. La complejidad funcional es menos preocupante de como lo era antes, lo que ahora ha tomado una prioridad alta es el modelar la comprensión del dominio del problema y las responsabilidades del sistema, por lo que metodologías como la OMT (Object Modeling Technique) se han convertido en una herramientas necesaria y de mucha importancia para el desarrollo de software.

La metodología OMT fue creada por James Rumbaugh y Michael Blaha en 1991, mientras James dirigía un equipo de investigación de los laboratorios General Electric. Cabe resaltar que Rumbaugh se unió a Rational Software en 1994, y trabajó allí con Ivar Jacobson y Grady Booch ("los Tres Amigos") para desarrollar UML. Más tarde fusionaron sus metodologías de desarrollo de software, OMT, OOSE y Booch en el Proceso Unificado Racional (RUP), una de las metodologías más utilizadas en la actualidad.

En su momento, OMT fue una de las metodologías de análisis y diseño orientada a objetos, más maduras y eficientes. La gran virtud aportada por esta metodología fue su carácter de abierta (no propietaria), que le permitió ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Lo que facilitó su evolución para acoplarse a las necesidades futuras de la ingeniería de software.

Tiene una fase de diseño no muy compleja y se centra mucho en un buen análisis.

Divide el ciclo de vida del software en cuatro fases consecutivas:

1. **Análisis de objetos:** se centra en entender y modelar el problema en el dominio de la aplicación.
2. **Diseño del sistema:** se determina la arquitectura del sistema en términos de subsistemas.
3. **Diseño de objetos:** se refina y optimiza el análisis de objetos para implementarlo.
4. **Implementación:** se codifica y prueba lo ya diseñado.

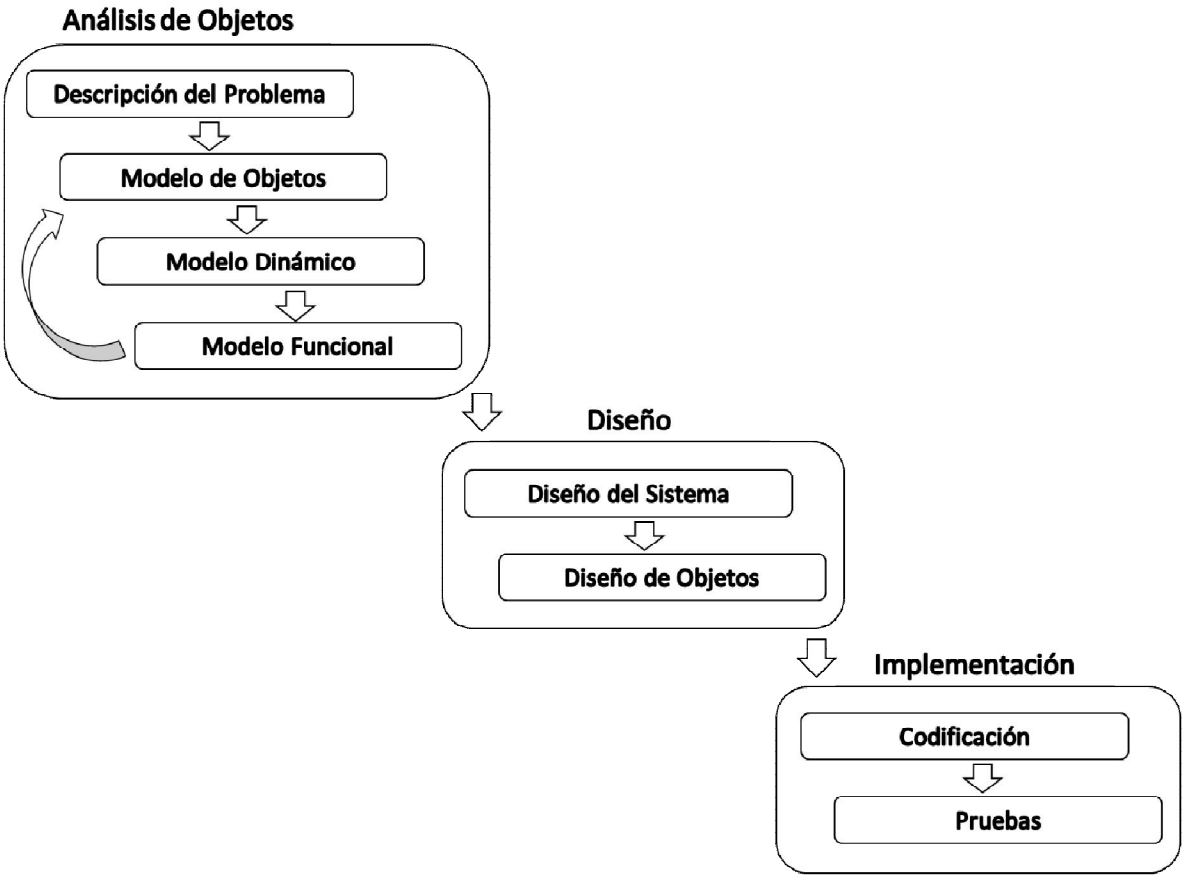


Figura 1. **Ciclo de vida OMT**

Aun cuando la descripción de esta técnica es lineal, el proceso de desarrollo real es iterativo. Donde se repiten los pasos de desarrollo con grados de detalle cada vez más finos, logrando así que cada iteración añada o clarifique características en vez de modificar un trabajo ya realizado, por lo tanto, existe menos posibilidades de introducir incongruencias y errores.

1. Análisis de Objetos

En primer lugar, se describe el problema. Se obtienen unos requisitos que no den lugar a dudas (rendimiento, funcionalidad, contexto, …).

En segundo lugar se hacen los diagramas de objetos. En él se define la estructura de los objetos y clases así como las relaciones que los unen. Comprende tanto un diagrama de clases como un diccionario de datos que las explique.

Posteriormente se crea un modelo dinámico para describir los aspectos de control y evolución del sistema. Incluye un diagrama de flujo de sucesos del sistema y un diagrama de estado por cada clase que tenga un comportamiento dinámico.

Después se crea un modelo funcional que describa las funciones, los valores de entrada y salida, e imponga las restricciones pertinentes. Se suelen usar los diagramas de flujo de datos orientados a objetos.

Por último, se verifican todos los modelos creados y se itera para conseguir un refinamiento de los tres modelos.

Resumiendo, los pasos a seguir son:

1. Se establece la definición del problema.
2. Se construye un modelo de objetos.
3. Se construye un modelo dinámico.
4. Se construye un modelo funcional.
5. Se verifican, iteran y refinan los tres modelos.

a) Modelo de Objetos

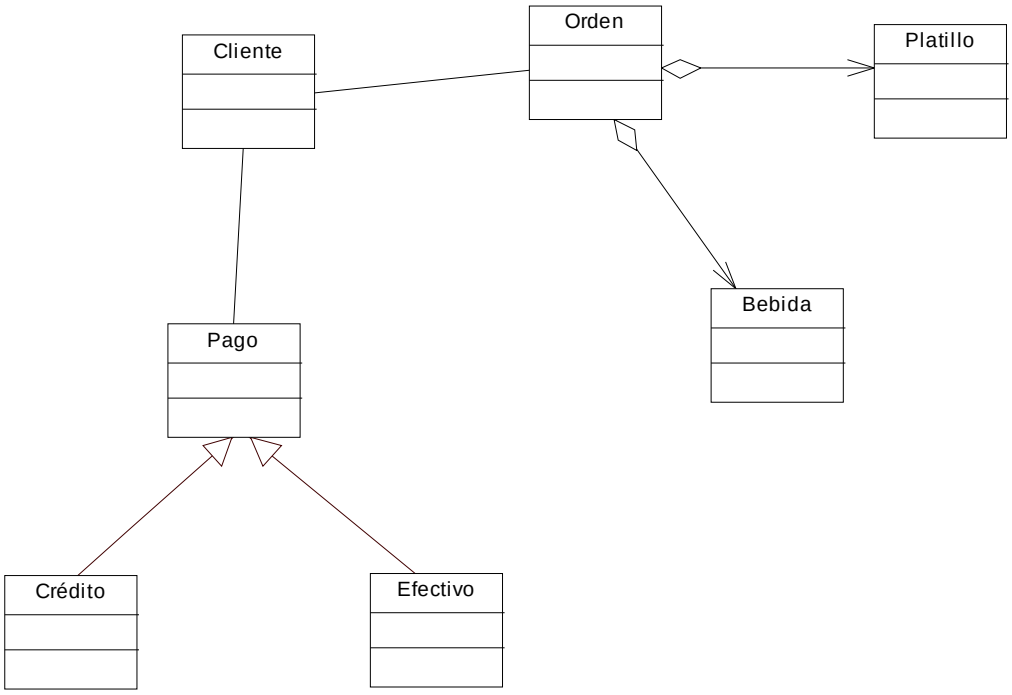
Los pasos para construir el modelo de objetos son los siguientes:

1. Identificación de objetos y/o clases.
2. Crear un diccionario de datos.
3. Identificación de las asociaciones y agregaciones entre los objetos.
4. Identificación de atributos y enlaces.
5. Organización y simplificación de las clases empleando herencia.
6. Verificación de las vías de acceso necesarias para llevar a cabo las probables consultas.
7. Realizar las iteraciones necesarias para el refinamiento del modelo.
8. Agrupar las clases en módulos.

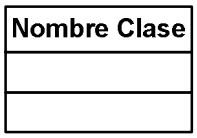
Modelo de objetos = Diagrama de modelo de objetos + diccionario de datos.

- **Diagrama de clases**

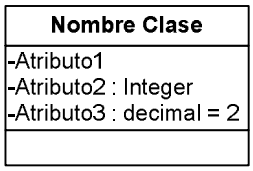
En él se describen las clases que se descubrieron para el sistema analizado en términos del dominio del problema. Además se especifican los atributos y operaciones que distinguen a cada una de las clases y las relaciones con las que podemos conocer su responsabilidad en el sistema.



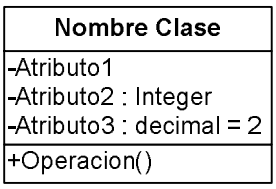
Notación: dentro del diagrama de clases, una clase se representa mediante un rectángulo, donde pueden existir tres separaciones, en la primer parte se coloca el nombre de la clase, en la segunda y tercera parte se pueden agregar los atributos y las operaciones, pero si no se desea agregar ninguno de ellos, es porque no son tan importantes para la comprensión del sistema, entonces el rectángulo solo se queda con el nombre de la clase.



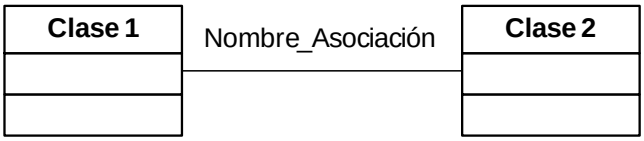
Atributos: Es un dato que distingue a una clase y que puede almacenar valores para el mismo en cada instancia que genere la clase. Debe tener un nombre y el tipo de dato que va a recibir. Los atributos se colocan en la segunda parte del rectángulo de la clase, primero se coloca el nombre del atributo, después precedido de dos puntos (:) el tipo de dato que recibirá y en algunos casos se podrá especificar el valor inicial que recibe precedido por un signo de igual (=).



Operaciones: Las operaciones son funciones que pueden realizar las instancias de una clase. Mediante ellas se pueden visualizar cuales son las responsabilidades de cada clase dentro del sistema. Se colocan en la tercera parte del rectángulo de la clase y debe de contener el nombre de la operación que puede ir seguida de una lista de argumentos entre paréntesis y de un tipo de dato que regresará precedido de dos puntos (:).

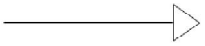


Relaciones (asociaciones): Representan los enlaces entre las instancias dentro del diagrama. Se representan mediante una línea que conecta a las instancias junto con el nombre de la asociación que por lo general es un verbo.



Cuadro 1

Tipos de asociaciones

Nombre	Símbolo	Descripción
Asociación		Asociación bidireccional entre clases con roles y multiplicidad
Agregación		Dicha relación representa la asociación que hay entre los componentes y la clase que se compone de ellos.
Composición		Relación es parte de
Dependencia		Relación de dependencia
Generalización		Es una relación entre una superclase que hereda sus características (atributos y operaciones) y subclases que harán suyas dichas características.

b) Modelo Dinámico

Los pasos para construir el modelo dinámico son los siguientes:

1. Preparación de escenarios de secuencias típicas de iteración.
2. Identificación de sucesos que actúan entre objetos.
3. Preparar un seguimiento de sucesos para cada escenario.

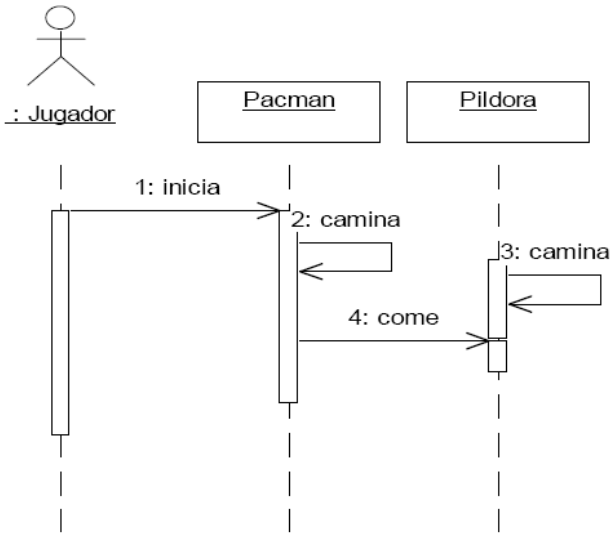
4. Construcción de un diagrama de estado para cada objeto.

5. Comparación de los sucesos intercambiados entre objetos para verificar la congruencia.

Modelo dinámico = Diagrama global de flujo de sucesos + Diagrama de estados

- **Diagrama de Flujo de sucesos o Traza de sucesos**

Una traza de eventos es una lista ordenada de eventos entre diferentes objetos (actores) asignados a columnas en una tabla. Se utiliza para identificar mensajes entre los actores de un cierto problema; de esta forma se pueden ver qué eventos afectan directamente a cada actor. Este diagrama muestra la ocurrencia de los eventos a través del tiempo, e indica un escenario que luego deberá ser incluido en el diagrama de estado. Los estados en este diagrama son los intervalos que ocurren entre cada evento; por lo que ayuda bastante en la identificación de los estados.



Suceso: Es un evento que ocurre en un determinado momento del sistema y por medio del cual se pueden transmitir valores entre los objetos.

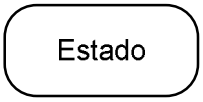
Desarrollar el diagrama de traza de sucesos es el siguiente paso para representar los escenarios después de haberlos realizado.

Cada objeto se muestra como una línea vertical. Los sucesos son representados mediante una flecha que va desde el objeto emisor al objeto receptor, y en el cual se puede incrementar el tiempo de arriba hacia abajo, según avanza este. Mediante el diagrama de traza de sucesos se muestra la forma en cómo los objetos se comunican entre sí enviándose mensajes, visto de otra forma, son peticiones de operaciones a realizar que un objeto le pide a otro.

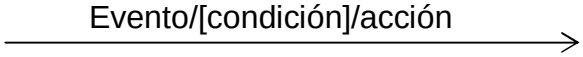
- **Diagrama de Estados**

Relaciona sucesos y estados. Un diagrama de estados se representa mediante estados, restricciones, condiciones y acciones.

Estados: Los estados representan las respuestas de los objetos a varios sucesos en determinado tiempo dentro del sistema. Dicha respuesta puede cambiar el estado del objeto. Se representan mediante cuadros redondeados que contienen un nombre.

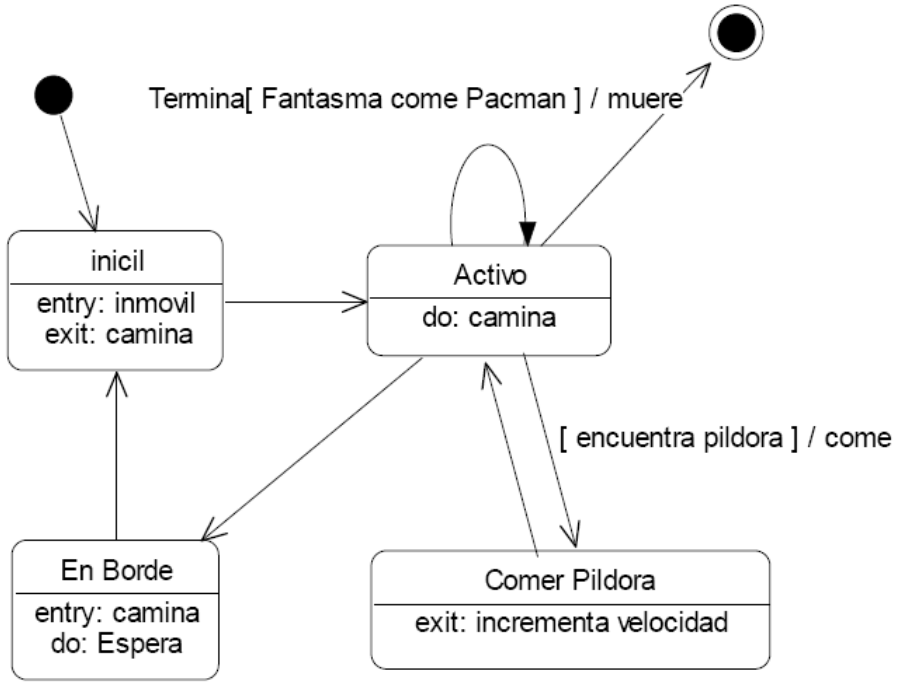


Transiciones: Se representan mediante flechas que salen del estado receptor hasta el estado destino y el nombre que se coloca en la flecha es el nombre del suceso que dio lugar a dicha transición, cada transición que sale de un estado corresponde a un suceso distinto, lo cual indica que no deben de existir sucesos duplicados dentro de un estado.



Condiciones: Una condición se puede pensar como una protección en las transiciones, debido a que si se cumple dicha condición la transición se dará y podrá pasar el objeto de un estado a otro, si dicha condición no se cumple inclusive podría pasar a otro estado mediante otra transición o quedarse en el estado receptor hasta que la condición se cumpla.

Acción: Es una operación que va asociada a un suceso y se representa mediante una barra "/" y el nombre de la acción, después del nombre de la transición.



c) Modelo Funcional

Los pasos para construir el modelo funcional son los siguientes:

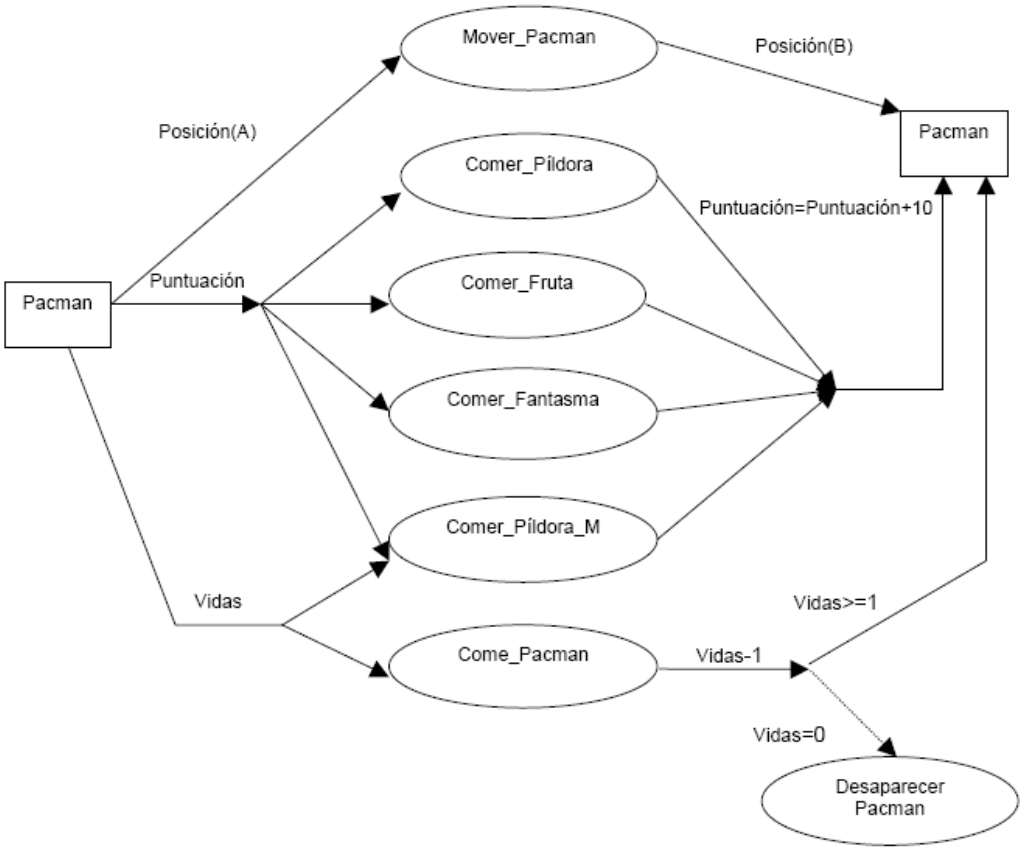
1. Identificación de los valores de entrada y de salida.
2. Construcción de diagramas de flujo de datos que muestren las dependencias funcionales.
3. Descripción de las funciones.
4. Identificación de restricciones.

5. Especificación de los criterios de optimización.

Modelo Funcional = Diagrama de flujo de datos + restricciones

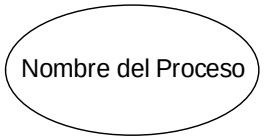
Mediante el modelo funcional se puede observar los resultados que se tienen de un cálculo de valores, especificando solamente entradas y salidas de los valores, mas no como son calculados estos.

El modelo funcional consta básicamente de diagramas de flujo de datos. Los diagramas de flujo de datos son grafos que muestran el flujo de valores de datos a través de procesos los cuales modifican dichos valores para transformarlos en otros.

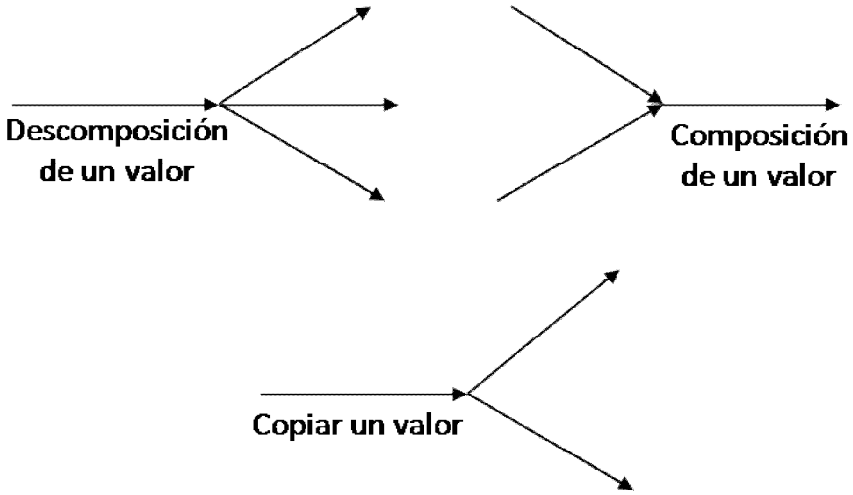


Los diagramas de flujo están compuestos de:

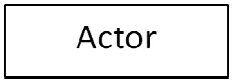
Procesos: se representan mediante una elipse, los procesos tienen como entrada datos, los cuales serán transformados, por lo cual un proceso es visto como un método de una operación aplicada a una clase de objetos.



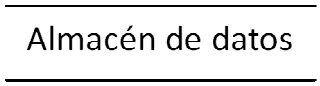
Flujo de datos: un flujo de datos conecta la salida de un proceso a la entrada de otro. Se representa en el diagrama por medio de una flecha, la cual puede llevar el nombre o el tipo de dato. Además de trasladar los datos a otros procesos, los flujos de datos pueden usarse para copiar un valor, realizar la composición de un agregado y así como su inverso.



Actores: los actores son objetos que consumen y producen datos generando operaciones por sí mismos, estos se encuentran siempre en las fronteras del diagrama indicando entradas y salidas de datos. Los actores también son llamados terminadores debido a que su función principal es hacer concluir el flujo de datos. En el diagrama son representados mediante rectángulos.



Almacenes de datos: son objetos cuya tarea es permitir el almacenamiento y acceso de datos. Se representan en el diagrama mediante unas líneas paralelas que tienen el nombre del almacén.



Teniendo ya la representación del sistema en los tres modelos, se lleva a cabo una iteración general de cada uno de ellos.

1. Se deben comparar los tres modelos con la definición del problema y con el conocimiento en el dominio de la aplicación.
2. Se añaden las operaciones claves descubiertas durante la preparación del modelo funcional.
3. Se hace una verificación entre clases, atributos, asociaciones y operaciones de tal manera que resulten congruentes.
4. Comprobar los modelos utilizando escenarios. Se desarrollan escenarios más detallados, incluyendo condiciones de error.
5. Se realiza una iteración de los pasos anteriores, hasta considerar satisfactorio el análisis.

Documento de Análisis = Definición del problema + modelo de objetos + modelo dinámico + modelo funcional.

El documento abarca el análisis, el diseño y la implementación. Contiene una notación gráfica para expresar modelos orientados a objetos, es posible modelar, diseñar e implementar tanto a objetos en el dominio de la aplicación como a objetos en el dominio de la computadora.

2. Diseño del Sistema.

El diseño del sistema es la estrategia de alto nivel para resolver el problema y construir una solución, incluye decisiones acerca de la organización del sistema (arquitectura del sistema) en subsistemas, la asignación de subsistemas a componentes de hardware y software y decisiones fundamentales conceptuales y de política que son las que constituyen el marco de trabajo para el diseño detallado.

El modelo de diseño debe ser razonablemente eficiente y práctico a la hora de codificar, tratando detalles de bajo nivel que se omiten en el modelo de análisis.

Los pasos a seguir son:

1. Organizar el sistema en subsistemas.
2. Identificar la concurrencia inherente en el problema.
3. Asignar los subsistemas a procesadores y a tareas.
4. Seleccionar la estrategia básica de implementación de los almacenes de datos, en términos de estructuras de datos, archivos y bases de datos.
5. Identificar los recursos globales y determinar los mecanismos para controlar el acceso a tales recursos.
6. Seleccionar una aproximación para implementar el control del software.
7. Consideraciones de condiciones de contorno.
8. Establecimiento de prioridades de compensación.

3. Diseño de Objetos

En el Diseño de Objetos se toman las decisiones necesarias para construir un sistema sin descender a los detalles particulares de un lenguaje o sistema de base de datos. El diseño de objetos es el comienzo de un desplazamiento con respecto al mundo real, en el modelo de análisis, aproximándose a la orientación a la computadora, necesaria para una implementación práctica.

Rumbaugh sugiere las siguientes etapas:

1. Obtención de las operaciones para el modelo de objetos a partir de los demás modelos.
2. Diseño de algoritmos para la implementación de las operaciones.
3. Optimización de las vías de acceso a los datos.
4. Implementar el control del software completando la aproximación seleccionada durante el diseño del sistema.
5. Ajuste de la estructura de clases para incrementar la herencia.
6. Diseño de la implementación de las asociaciones.
7. Se determina la representación exacta de los atributos que son objetos.
8. Empaquetamiento de las clases y asociaciones en módulos.

4. Implementación del Sistema.

Durante la implementación se codifican, tanto las estructuras en el dominio de la aplicación como las estructuras en el dominio de la solución. La base que la sustenta es el diseño de objetos.

El código puede ser una simple transición de las decisiones de diseño a las características propias de un lenguaje.

Aquí se den hacer pruebas para determinar si el sistema está siendo construido correctamente.

Referencias

- Metodologías para Análisis y Diseño Orientado a Objetos y MDA. Disponible en [http://www.scribd.com/doc/12848359/Metodologias-Para-Analisis-y-Diseno-Orientado-a-Objetos-y-MDA]

- Rumbaugh, James et al. (1996). Modelado y diseño orientados a objetos. Madrid: Prentice Hall, 1996.

- Técnica de Modelado de Objetos (OMT) (James Rumbaugh). Disponible en [http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%C3%B1o%20orientado%20a%20objetos/rumbaugh.pdf]