**PPt1**

**UML**

* UML corresponden a las siglas de Unified Modeling Language (Lenguaje Unificado de Construcción de Moldeos), notación (esquemática en su mayor parte) co que se construyen sistemas por medio de conceptos orientafos a objetos.
* ¿Cómo deberían asignarse las responsabilidades a clases de objetos? ¿Cómo deberían interactuar éstos? ¿Qué papel debe destinársele a cada clase?. Éstas son preguntas claves cuando se diseña el sistema.
* Algunas soluciones ya probadas y eficaces de los problemas de diseñi pueden expresarse (y se han plasmado) como un conjunto de principios, con heurísticas o patrones (fórmulas de soluciones de problemas que codifican los principios aceptados del diseño).

**Asignación de responsabilidades**

* Hay muchas posibles actividades y elementos en el análisis y en el diseño, así como una gran cantidad de principios y directrices. Pero cuál de ellas es la fundamental?
* La habilidad más importante en el análisis y diseño orientados a objetos es **asignar** eficientemente las **responsabilidades** a los componentes de software.
* ¿Por qué es tan importante?
  + Provee solidez
  + capacidad de mantenimiento
  + reutilización de los componentes de software.

**Análisis= Investigación**

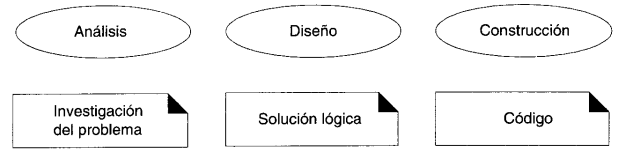
* Para crear una aplicación de software hay que describir el problema y las necesidades o requerimientos: en qué consiste el conflicto y qué debe hacerse. El análisis se centra en una ***investigación del problema***, no en la manera de definir una solución. Por ejemplo, si se desea un nuevo sistema de información computarizada de una biblioteca. ¿Cuáles procesos de la instituciòn se relacionan con us uso?

**Diseño=Solución**

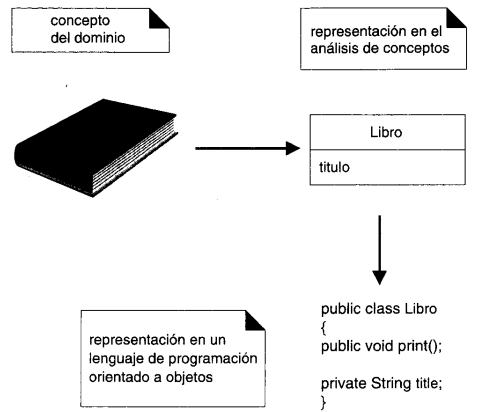
* Para desarrollar una aplicación, también es necesario contar con descripciones detallas y de alto nivel de la solución lógica y saber cómo satisface los requerimientos y las restricciones. El diseño pone de relieve una solución lógica: cómo el sistema cumple con los requerimientos. Por ejemplo: ¿de qué manera el software del sistema de información de la biblioteca capturará y registrará los préstamos de libros?. Así, los diseños se implementan en un software y en hardware.

**Análisis y Diseño orientado a objetos**

* La esencia del análisis y el diseño orientado a objetos consiste en situar el dominio de un problema y su solución lógica dentro de las perspectivas de los objetos (cosas, conceptos o entidades).



* Durante el análisis orientado a objetos se procura ante todo identificar y describir los objetos o conceptos dentro del dominio del problema. Por ejemplo, en el caso del sistema de información de la biblioteca, algunos de los conceptos son *Libros*, *Biblioteca* y *Cliente*.
* Durante el diseño orientado a objetos, se procura definir los objetos lógicos del software que finalmente serán implementados en un lenguaje de programación orientado a objetos. Los objetos tienen atributos y métodos. Así, en el sistema de la biblioteca un **objeto** de software ***Libro*** puede tener un **atributo *título*** y un **método *imprimir***.



**Empresa MicroCaos**

* Imagine ser el fundador y director de MicroCaos, una compañia creada y especilizada en el software que aplica modelos matemáticos de la teoría del caos al análisis del mercado accionario. Esta empresa vende MicroButterfly. En un inicio todos compartían trabajo, contestaban el teléfono, surtía pedidos, escribían programas de computación, sin embargo la empresa creció y hubo dispersión de labores.
  + ¿Cómo se pueden organizar las labores?

**Procesos de Negocio**

* El primer paso consiste en analizar lo que debe hacer una empresa, ésto es sus procesos de negocio, si quiere seguir funcionando: realizar ventas, pagar a empleados y acreedores, desarrollar programas de computación
* Para ello, primero hay que realizar un ***análisis de requerimientos***, en el cual los procesos y las necesidades de los negocios se descubren y se expresan en los ***casos de uso.*** Los caso de uso son descripciones narrativas textuales de los procesos de una empresa o sistema.

**Procesos de Negocio**

|  |  |  |
| --- | --- | --- |
| **Caso de uso** | **:** | **Colocar un pedido** |
| Descrpción | : | Este caso de uso comienza cuando un cliente telefonea a un representante de ventas para hacer una compra de MicroButterfly. El representante anota en una nueva orden la información relativa al cliente y al producto. |

**Identificar los procesos** y registrarlos en los casos de uso n**o es en realidad una actividad del análisis sorientado a objetos**; de ninguna manera se centra en los objetos.

Esto se trata de un paso importante y generalizado en los métodos del análisis y diseño orientado a objetos. Asimísmo, los casos de us forman parte de UML. Vean la analogía:

Procesos de Negocio

|  |  |  |
| --- | --- | --- |
| **Analogía de la empresa** | **Análisis y diseño orientado a objetos** | **Documentos relacionados** |
| ¿Cuáles son los procesos de negocio? | Análisis de requerimientos | Casos de uso |

**Papeles y funciones de la organización**

* El siguiente paso consiste en identificar los papeles de las personas que intervendrán en los procesos: clientes, representantes de ventas, ingeniero de software, entre otros.
* En la perspectiva del análisis y del diseño orientado a objetos, este paso nos recueda el **análisis de dominio orientado a objetos** que expresamos como **modelo conceptual**. Este modelo presenta las diversas categorías de las cosas en el dominio.

**Funciones de los empleados**

* Una vez identificados los procesos de sus empresa y el personal, es el momento de determinar la manera de cumplor los procesos. Se trata de una actividad de diseño, o sea orientada a las soluciones. Junto con los empleados defina responsabilidades de ellos a fin de efectuar las tareas ncesarias para llevar a cabo un proceso. También es necesario definir cómo éstos compartirán o colaborarán en el trabajo.
* Desde el punto de vista del análisis y del diseño orientado a objetos, esta actividad se parece al **diseño orientado** a objetos que pone de relieve la **asignación de responsabilidades**. La asignación significa distribuir las funciones y las reponsabilidades entre varios objetos de software enla aplicación. Del mismo modo que se asignan roles a los empleados, los objetos de software colaboran o interactúan con sus responsabilidades, como lo hacen las personas.
* La descripción de la asignación de las responsabilidades y las interacciones de objetos se expresan gráficamente con **diagramas de diseño** de clases y con **diagramas de colaboración**; unos y otros muestran la definición de clases y el flujo de mensajes entre los objetos de software.

**Un ejemplo del análisis y diseño orientado a objetos**

Se necesita abundante información para explicar cabalmente el análisis y el diseño orientado a objetos. Para no perdernos en muchos detalles, supongamos el siguiente ejemplo. Un juego de datos, en que si un jugador lanza dos datos, y el resultado es siete, ésta gana, de lo contrario pierde.

La notación mostrada forma parte de UML.

**Definición de los casos de uso**

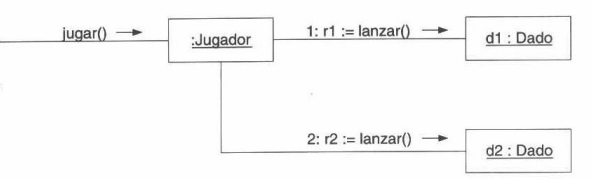
Para entender los requerimientos se necesita, en parte, conocer los procesos del dominio y el ambiente externo, o sea los factores externos que participan en los procesos. Dichos procesos de dominio pueden expresarse en **casos de uso**, o sea, en descripciones narrativas de los procesos del dominio en un formato estructurado en prosa.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Definir los casos de uso** |  | **Definir al modelo conceptual** |  | **Definir los diagramas de colaboración** |  | **Definir los diagramas de diseño de clases** |

Para entender los requerimientos se necesita, en parte, conocer los procesos del dominio y el ambiente externo, o sea los factores externos que participan en los procesos. Dichos procesos de dominio pueden expresarse en **casos de uso**, o sea, en descripciones narrativas de los procesos del dominio en un formato estructurado en prosa.

Como mencionamos anteriormente, los casos de uso no son propiamente un elemento del análisis orientado a objetos; se limitan a describir procesos y pueden ser igualmente eficaces en un proyecto de tecnología no orientada a objetos. No obstante, constituyen un paso preliminar muy útil porque describen las especificaciones de un sistema.

Por ejemplo, supongamos que se desea una simulación en el software del juego de dados. El diagrama de colaboración muestra gráficamente el paso esencial del juego, enviando mensajes a las instancias de las clases *Jugador* y *Dado*.



**Definición del diseño de clases**

Para definir una clase, es preciso contestar varias preguntas:

* + ¿Cómo se conectan unos objetos a otros?
  + ¿Cuáles son los métodos de una clase?

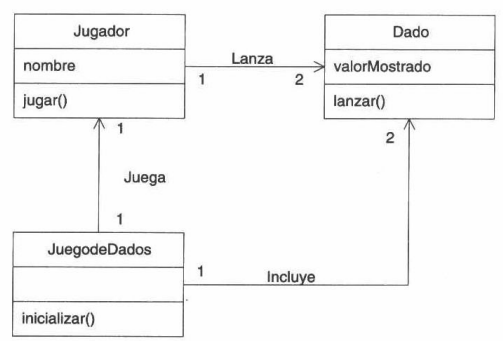
Si se quieren contestar las preguntas anteriores, hay que examinar detenidamente los diagramas de colaboración que indican las conexiones necesarias entre objetos, y también métodos que cada clase de software debe definir. El diagrama de diseño de clases es el que expresa esos detalles. Muestra las definiciones de clase que han de implementarse en el software.

**Diagrama de clases**

Por ejemplo, en el juego de dados, al examinar el diagrama de colaboración, obtenermos el siguiente diagrama del diseño de clases. Puesto que un mensaje *juego* se envía a una instancia *Jugador*, *Jugador* requiere un método jugar, mientras que *Dado* requiere un método *lanzar*.

A diferencia del modelo conceptual, este diagrama no muestra gráficamente conceptos del mundo real; describe únicamente los componentes del software.

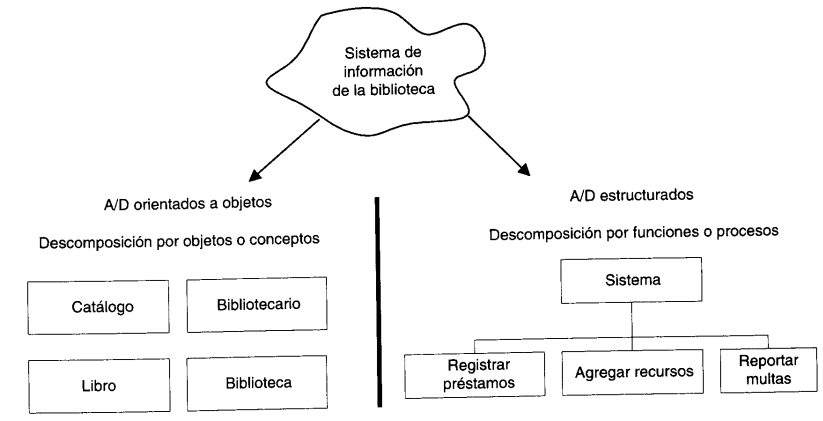
Para indicar de qué manera los objetos se conectan entre sí a través de atributos, una línea con una flecha en la punta indicará un atributo. Por ejemplo, *JuegodeDados* posee un atributo que apunta a una instancia de un *Jugador*



**Diseño orientado a funciones y diseño orientado a objetos**

Generalmente, la manera de afrontar un problema para disminuir la complejidad de ésto y hacerlo más manejable es a través de las descomposición de subproblemas (divide y vencerás).

En contraste en orientación a objetos el problema se divide en objetos y no en funciones.



A veces el límite entre el análisis y el diseño no es todo claro. No obtante, es mejor establecer que el análisis conlleva la etapa de entendimiento (investigación) del problema, mientras que el diseño correspondería a la solución de éste.

Es importante mecionar que el lenguaje UML estandariza los artefactos y la notación, **pero no define un proceso oficial de desarrollo.**