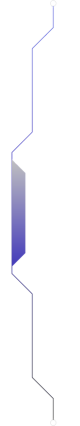
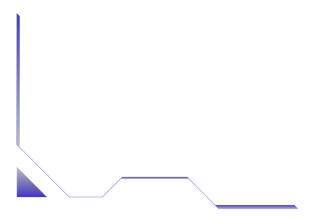
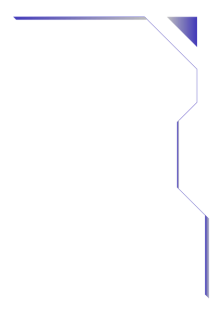
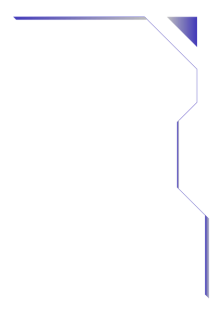
INGENIERÍA DE SOFTWARE I 

MODELADO DE

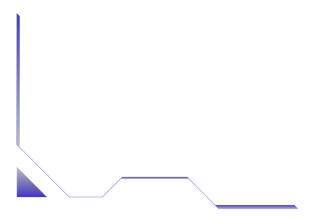
SISTEMAS

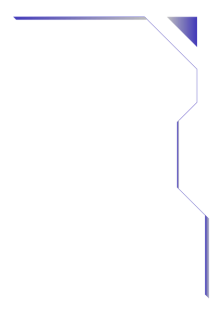


¿Qué es un modelo?

¿Qué es un modelo? 

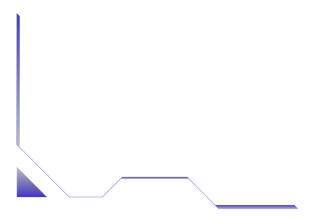
Un modelo es una representación simplificada, abstracta y generalmente idealizada de la realidad, construida con el propósito de comprender, analizar, explicar o predecir el comportamiento de un fenómeno, objeto, sistema o proceso.



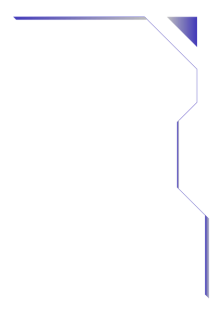
¿Qué es un modelo? 

Un modelo es una representación simplificada, abstracta y generalmente idealizada de la realidad, construida con el propósito de comprender, analizar, explicar o predecir el comportamiento de un fenómeno, objeto, sistema o proceso.

Simplificada: ?

Abstracta: ? 

Idealizada: ?

¿Qué es un modelo? 

Un modelo es una representación simplificada, abstracta y generalmente idealizada de la realidad, construida con el propósito de comprender, analizar, explicar o predecir el comportamiento de un fenómeno, objeto, sistema o proceso.

Simplificada: porque omite detalles irrelevantes para el objetivo que se persigue.

Abstracta: porque no es el objeto real, sino una interpretación de él. Idealizada: porque puede asumir condiciones ideales que no siempre cumplen en la realidad. 

¿Y el modelado de sistemas?

Modelado de sistemas 

Proceso para desarrollar modelos abstractos de un sistema, donde cada modelo presenta una visión o perspectiva diferente de dicho sistema

Se ha convertido en un medio para representar el sistema usando algún tipo de notación gráfica



¿En qué etapa los usamos?

¿En qué etapa los usamos?Durante el proceso de ingeniería de requerimientos para ayudar a derivar los requerimientos de un sistema 

Durante el proceso de diseño para describir el sistema a los ingenieros que implementan el sistema

Después de la implementación para documentar la estructura y la operación del sistema.



Modelos de sistemas Ayudan a aclarar lo que hace el sistema 

Modelos del sistema existente de una organización

Modelos del

sistema que 

se construirá

existente y pueden utilizarse como base para discutir sus fortalezas y debilidades. Posteriormente, conducen a los

requerimientos para el nuevo sistema.

Ayudan a explicar los requerimientos propuestos a otros participantes del sistema. Útiles para discutir las propuestas de diseño y documentar el sistema para la

implementación.



¿Verdadero o falso? 

El aspecto más importante de un

modelo del sistema es que deja fuera

los detalles.





¿Verdadero o falso? 

El aspecto más importante de un

modelo del sistema es que deja fuera

los detalles.





¿Verdadero o falso? El aspecto más importante de un

modelo del sistema es que deja fuera

los detalles.

Un modelo es una abstracción del sistema a estudiar, y no una representación alternativa de dicho sistema. De manera ideal, una representación de un sistema debe mantener toda la información sobre la entidad a

representar.

Una abstracción simplifica y recoge deliberadamente las características más destacadas.



UML (Unified Modeling Language) 

Es un lenguaje de modelado visual estándar utilizado para especificar, visualizar, construir y documentar los artefactos de un sistema software, así como sus procesos y estructuras. No es un lenguaje de programación, sino un lenguaje de representación gráfica para describir cómo funciona un sistema.

UML (Unified Modeling Language) 

Se basa en diagramas que representan distintos aspectos del sistema: Diagramas de actividad, que muestran las actividades incluidas en un proceso o en el procesamiento de datos.

Diagramas de caso de uso, que exponen las interacciones entre un sistema y su entorno.

Diagramas de secuencias, que muestran las interacciones entre los actores y el sistema, y entre los componentes del sistema. Diagramas de clase, que revelan las clases de objeto en el sistema y las asociaciones entre estas clases. 

Diagramas de estado, que explican cómo reacciona el sistema frente a eventos internos y externos.

Perspectivas para modelar Perspectiva externa Se modelan el contexto o entorno del sistema (Diagramas de Contexto) 

Perspectiva

de interacción

Perspectiva

estructural

Perspectiva 

de comportamiento

Se modela la interacción entre un sistema y su entorno, o entre los componentes de un sistema (Diagramas de caso de uso y de secuencias) Se modela la organización de un sistema o la estructura de datos que procese el sistema (Diagramas de clase, Diagramas de Objetos) Se modela el comportamiento dinámico del sistema y cómo responde ante ciertos eventos (Diagramas de estado, Diagrama de Actividad)

Modelos de contexto 

En una primera etapa en la especificación de un sistema, se debe decidir sobre las fronteras del mismo. Esto implica trabajar con los participantes del sistema para determinar cuál funcionalidad se incluirá y cuál la ofrece el entorno.

Hay cuestiones sociales y organizacionales que pueden afectar a la definición de la frontera de un sistema.

Los sistemas externos generan datos para el sistema o consumen datos del sistema. Pueden compartir datos con el sistema, conectarse directamente, a través de una red, o no conectarse en absoluto. Pueden estar físicamente juntos o ubicados en edificios separados.

Modelos de interacción 

Todos los sistemas incluyen interacciones de algún tipo. Éstas son interacciones del usuario, que implican entradas y salidas del usuario; interacciones entre el sistema a desarrollar y otros sistemas; o interacciones entre los componentes del sistema.

Veremos estos modelos, que son complementarios entre ellos: Modelado de Casos de uso: se utilizan para modelar interacciones entre un sistema y actores externos (usuarios u otros sistemas) 

Diagramas de secuencia: para modelar interacciones entre componentes del sistema

Modelos estructurales 

Muestran la organización de un sistema, en términos de los componentes que constituyen dicho sistema y sus relaciones Muestran la estructura estática del sistema y sus partes en diferentes niveles de abstracción

Se crean cuando se discute y diseña la arquitectura del sistema. En UML existen un total de siete tipos de diagramas de estructura.



Modelos estructurales 

Veremos diagramas

de clase para modelar

la estructura estática

de las clases de

objetos.



Modelos de comportamiento 

Son modelos dinámicos del sistema conforme se ejecutan Se muestra lo que sucede o lo que se supone que pasa cuando un sistema responde ante un estímulo del entorno:

datos que llegan y se procesan en el sistema

eventos que activan el procesamiento del sistema

Veremos diagramas de estado y de actividad



Modelos de comportamiento



Diagrama de Clases

Es un modelo estructural

Se utiliza para representar los elementos que componen un sistema de información desde un punto de vista estático

El diagrama define las clases que se utilizarán cuando se pase a la fase de construcción y la manera en que se relacionan las mismas. Muestran el modelo lógico de los datos de un sistema. Los elementos representan algo del mundo real.

El primer paso es identificar los elementos esenciales del sistema y representarlos como clases

El segundo paso es dibujarlos en un recuadro

El tercer paso es asociarlos dibujando una línea entre las clases

Se compone de clases, relaciones e interfaces.





Clase

Elemento principal del diagrama.

Representan conceptos o entidades del negocio.

Define un grupo de objetos que comparten características, condiciones y significado.

La manera más rápida de encontrar clases sobre un enunciado o sobre un negocio, es buscar los sustantivos que aparecen en el mismo.





Clase

Tanto los atributos como las funciones incluyen al principio de su descripción la visibilidad que tendrá.

(+) Pública. Representa que se puede acceder al atributo o función desde cualquier lugar de la aplicación.

(-) Privada. Representa que se puede acceder al atributo o función únicamente desde la misma clase.

(#) Protegida. Representa que el atributo o función puede ser accedida únicamente desde la misma clase o desde las clases que hereden de ella (clases derivadas).





Relación

Una relación identifica una dependencia.

Esta dependencia puede ser:

entre dos o más clases (más común)

una clase hacía sí misma (menos común, pero existen), este último tipo de dependencia se denomina dependencia reflexiva. Las relaciones se representan con una línea que une las clases, esta línea variará dependiendo del tipo de relación

Dependencia común Dependencia reflexiva



Relación - Propiedades

Multiplicidad: es decir, el número de elementos de una clase que participan en una relación.

Se puede indicar un número, un rango

Se utiliza “n” o “ \* “ para identificar un número cualquiera

Nombre de la asociación: en ocasiones se escribe una indicación de la asociación que ayuda a entender la relación que tienen dos clases. Suelen utilizarse verbos





Relación - Propiedades

Si la multiplicidad no se indica explícitamente, está pendiente, no existe multiplicidad predeterminada en UML.





Relación - Propiedades Uno a Uno 

Uno a Muchos



Relación - Propiedades 

Muchos a 

muchos





Relación - Tipos de relaciones 

Asociación: Este tipo de relación es el

más común y se utiliza para representar 

dependencia semántica.

Se representa con una simple línea

continua que une las clases que están

incluidas en la asociación.

«Una mascota pertenece a una

persona».



Relación - Tipos de relaciones Agregación: Es una representación 

jerárquica que indica a un objeto y las partes

que componen ese objeto.

Representa relaciones en las que un

objeto es parte de otro, pero aun así

debe tener existencia en sí mismo.

Se representa con una línea que tiene un

rombo en la parte de la clase que

contiene las otras.

Un ejemplo de esta relación podría ser:

«Una empresa tiene 0 o Muchos Clientes.

La entidad Cliente puede existir si se

destruye la entidad Empresa».





Relación - Tipos de relaciones

Composición: Es similar a la agregación, representa una relación jerárquica entre un objeto y las partes que lo componen, pero de una forma más fuerte. 

Los elementos que forman parte no tienen sentido de existencia cuando el primero no existe.

Cuando el elemento que contiene los otros

desaparece, deben desaparecer todos ya que no tienen sentido por sí mismos sino que dependen del elemento que componen. Además, suelen tener los mismos tiempos de vida.

Se representa con una línea continua con un rombo relleno en la clase que es compuesta.

«Una Empresa está compuesto por empleados, si la empresa se destruye se elimina la entidad

Empleado»



Relación - Tipos de relaciones

Herencia:

También se conoce como generalización

Este tipo de relaciones permiten que una clase (clase hija o subclase) reciba los atributos y métodos de otra clase (clase padre o superclase).

Estos atributos y métodos recibidos se suman a los que la clase tiene por sí misma.





Relación - Tipos de relaciones

Herencia:

Un ejemplo de esta relación 

podría ser la siguiente: En la

figura se especifica que Auto y

Camión heredan de Vehículo, es

decir, Auto posee las

Características de Vehículo

(Precio, VelMax, etc) además

posee algo particular que es

Descapotable, en cambio

Camión también hereda las

características de Vehiculo

(Precio, VelMax, etc) pero posee

como particularidad propia

Acoplado, Tara y Carga.



Relación - Tipos de relaciones 

Herencia: 

Otro ejemplo





Relación - Tipos de relaciones

Dependencia:

Se utiliza este tipo de 

relación para representar

que una clase requiere de

otra para ofrecer sus

funcionalidades.

Se representa con una flecha

discontinua que va desde la

clase que necesita la utilidad

de la otra flecha hasta esta

misma.





Interfaces

Es una entidad que declara una serie de atributos, funciones y obligaciones.

Es una especie de contrato donde toda instancia asociada a una interfaz debe de implementar los servicios que indica aquella interfaz. Dado que únicamente son declaraciones no pueden ser instanciadas. Las interfaces se asocian a clases.

Una asociación entre una clase y una interfaz representa que esa clase cumple con el contrato que indica la interfaz, es decir, incluye aquellas funciones y atributos que indica la interfaz.

Su representación es similar a las clases, pero indicando arriba la palabra <<interface>>.





Buenas prácticas para construir diagramas de clases

Los diagramas de clase pueden tender a volverse incoherentes a medida que se expanden y crecen.

Es mejor evitar la creación de diagramas grandes y dividirlos en otros más pequeños que se puedan vincular entre sí más adelante.

Cuantas más líneas se superpongan en sus diagramas de clase, más se complica utilizarlo.





Cómo construir diagramas de clases

Identificar los nombres de las clase

El primer paso es identificar los objetos primarios del sistema. Las clases suelen corresponder a sustantivos dentro del dominio del problema.

Distinguir las relaciones

El siguiente paso es determinar cómo cada una de las clases u objetos están relacionados entre sí. Busca los puntos en común y las abstracciones entre ellos; esto te ayudará a agruparlos al dibujar el diagrama de clase.

Crear la estructura

Primero, agregar los nombres de clase y vincularlos con los conectores apropiados, prestando especial atención a la cardinalidad o las herencias.

Dejamos los atributos y funciones para más tarde, una vez que esté la estructura del diagrama resuelta.







Una biblioteca contiene libros y revistas.



Puede haber varias copias de un libro. Algunos de los libros son reservados sólo para préstamos a corto plazo. Todos los otros pueden ser prestados a cualquier miembro de la biblioteca por tres semanas. Los miembros de la biblioteca pueden normalmente solicitar hasta seis ítems de una vez, pero miembros del staff pueden solicitar hasta doce ítems a la vez.

Solamente miembros del staff pueden pedir préstamos de revistas. El sistema debe conservar la pista de cuando los libros y revistas son prestados y retornados a la biblioteca.





Identifiquemos sustantivos

Una biblioteca contiene libros y revistas.

Puede haber varias copias de un libro. Algunos de los libros son reservados sólo para préstamos a corto plazo. Todos los otros pueden ser prestados a cualquier miembro de la biblioteca por tres semanas. Los miembros de la biblioteca pueden normalmente solicitar hasta seis ítems de una vez, pero miembros del staff pueden solicitar hasta doce ítems a la vez.

Solamente miembros del staff pueden pedir préstamos de revistas. El sistema debe conservar la pista de cuando los libros y revistas son prestados y retornados a la biblioteca.





Identifiquemos sustantivos

Una biblioteca contiene libros y revistas.

Puede haber varias copias de un libro. Algunos de los libros son reservados sólo para préstamos a corto plazo. Todos los otros pueden ser prestados a cualquier miembro de la biblioteca por tres semanas. Los miembros de la biblioteca pueden normalmente solicitar hasta seis ítems de una vez, pero miembros del staff pueden solicitar hasta doce ítems a la vez.

Solamente miembros del staff pueden pedir préstamos de revistas. El sistema debe conservar la pista de cuando los libros y revistas son prestados y retornados a la biblioteca.





Clases candidatas

Biblioteca → Nombre del Sistema

Libro

Revista

Copia

Préstamos A Corto Plazo → Evento

Miembro De Biblioteca

Item → libro o revista

MiembroDelStaff

Sistema → Término general





Relaciones entre clases

Libro es un Item

Revista es un Item

Copia es una copia de Libro

Miembro De Biblioteca

Item

Miembro De Staff es un Miembro De Biblioteca





Operaciones

Miembro De Biblioteca pide prestado Copia

Miembro De Biblioteca devuelve Copia

Miembro De Staff pide prestado Revista

Miembro De Staff devuelve Revista





Diagrama



Beneficios de los diagramas de clases

Ilustrar modelos de datos para sistemas de información, sin importar qué tan simples o complejos sean.

Comprender mejor la visión general de los esquemas de una aplicación.

Expresar visualmente necesidades específicas de un sistema y divulgar esa información en toda la empresa.

Crear diagramas detallados que resaltan cualquier código específico que será necesario programar e implementar en la estructura descrita.

Ofrecer una descripción independiente de la implementación sobre los tipos empleados en un sistema que son posteriormente transferidos entre sus componentes.



¿Preguntas?