

Modelamiento de Datos
Conceptos Fundamentales
Profesor: Fernando Guerrero T.
Copyright© 2004

Contenidos.

Parte I. Conceptos Fundamentales.

1. 1. Realidades, Modelos y Lenguajes
2. 2. Estrategia General de Resolución de Problemas
3. 3. Dimensiones de un Sistema Basado en Software
4. 4. Características de los Sistemas de Bases de Datos
5. 5. Procesos de Abstracción en el Modelamiento
6. 6. Propiedades de las Correspondencias entre Clases
7. 7. Concepto de Dato y Modelo de Dato

I. Conceptos Fundamentales.

1 Realidades, Modelos y Lenguajes.

La realidad única, concreta y objetiva no puede ser captada como tal. Aún cuando pudiésemos asumir que esta realidad única existe, cada uno de nosotros la modifica a través del filtro de su percepción. La percepción de cada persona es algo bastante complejo, que está influido, entre otros posibles factores, por el tiempo, espacio y estado de ánimo al momento de realizar la percepción, además

del impacto de experiencias previas, factores ambientales, estructura neuronal y el código genético del individuo. Lo relevante es que para n observadores de un fenómeno, es posible obtener al menos n percepciones distintas (aunque posiblemente no “radicalmente” distintas). Las herramientas que utilizamos para poder comunicar y plasmar nuestras percepciones de realidades se denominan modelos. Los modelos son representaciones de algún fenómeno o hecho del mundo que nos interese (en el caso de la ingeniería de sistemas interesaría por ejemplo modelar organizaciones, datos o procesos de negocio). Para poder expresar estos modelos es que requerimos de los lenguajes. Los lenguajes son herramientas creadas por el hombre (u otros seres) con el fin de comunicarse. Son imprescindibles para poder concebir modelos, pues uno expresa a lo más lo que el lenguaje le permite. Además, los lenguajes son los que permiten comunicar los modelos a otros (que comprenden dichos lenguajes), validarlos, discutirlos y ampliar la percepción del otro sobre un mismo fenómeno. Para efectos de este curso, consideraremos los siguientes componentes de los lenguajes.

1. La sintaxis. Es el conjunto de símbolos permitidos en el lenguaje. (por ejemplo las letras del abecedario o todas las palabras del idioma español)
2. Una gramática. Son las reglas generadoras del lenguaje. (por ejemplo la gramática del español)
3. La semántica. Es el significado asociado al lenguaje (por ejemplo, el significado de las palabras y su interpretación dentro de un contexto dado).

2 Estrategia general de resolución de problemas.

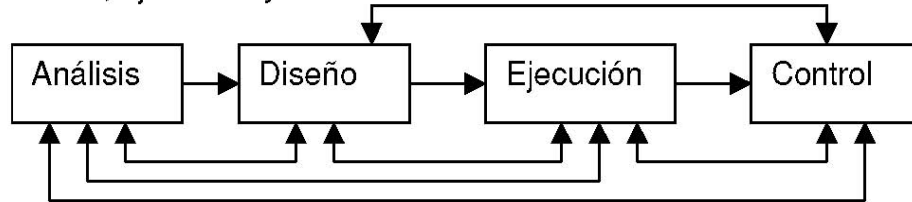
Para poder abordar un problema, primero hay que definirlo. Esto, que pudiese parecer trivial, no lo es para nada. Definir un problema significa comprenderlo y modelarlo. Los problemas son realidades susceptibles de ser modeladas de diversas maneras, como se muestra en el siguiente ejemplo, donde distintos involucrados nos dan su visión acerca de los problemas de una organización.

INVOLUCRADOS	¿CUALES SON LOS PROBLEMAS DE ESTA ORGANIZACIÓN?
--------------	---

Empleado	¿Cómo aumento mis ingresos? ¿Cómo beneficio a mi organización? ¿Cómo aumento mi productividad y mi calidad de trabajo? ¿Cómo quedo mejor con mis jefes y consigo una mejora en mi situación actual?
Ejecutivo	¿Cómo aumento la productividad? ¿Cómo dejo feliz a mi gente sin ir contra las políticas de la empresa? ¿Cómo motivo a mi personal? ¿Cómo aumento la calidad de los productos? ¿Cómo puedo mejorar mis procesos y productos?
Directorio	¿Cómo aumentamos nuestras ganancias? ¿Está bien gestionada nuestra empresa? ¿Cómo aseguramos el crecimiento de la empresa?
Comercialización y Marketing	¿Cómo creo nuevos mercados? ¿Cómo dejo más satisfechos a los clientes? ¿Cómo me posiciono mejor en el mercado? ¿Dónde hay oportunidades?
Matemático	¿Cuál es la función matemática que optimiza esta organización? ¿Cuáles son las variables relevantes?
Secretaria	¿Cómo disminuyo mi carga de trabajo?
Contabilidad	¿Cómo simplifico el proceso contable? ¿Cuáles son los flujos de entrada y de salida? ¿Cómo hago la contabilidad en menor tiempo?
Recursos Humanos	¿Cómo apporto mayores beneficios al personal? ¿Cuáles son las principales técnicas para mantener al personal a gusto?
Administración Financiera	¿Cómo disminuyo costos?

Gerencia	¿Cómo aseguro la misión de la empresa? ¿Cómo aumento la productividad de esta organización? ¿Dónde hay oportunidades de mejora? ¿Cuál debe ser nuestra estrategia? ¿Cómo vamos?
Informáticos	¿Qué procesos son factibles de automatizar? ¿Dónde podemos introducir tecnología de punta? ¿Cómo convengo a los usuarios de que no tienen la razón?
Contratistas	¿Cómo me hago indispensable? ¿Cómo me aseguro trabajo por un periodo mediano?

Básicamente, el proceso de resolver un problema consta de cuatro etapas: análisis, diseño, ejecución y control de la solución.



2.1 Análisis del Problema.

El análisis del problema involucra capturar el máximo de información referente a este, obtener la visión del mismo por parte de los involucrados y generar un modelo para cada una de estas visiones. Se deben refinar estos modelos hasta obtener una o varias representaciones (modelos) del problema, los que posibilitan su análisis. Estos modelos que describen las visiones del problema son los que posibilitarán que se vislumbre alguna solución. Se debe modelar tanto el problema como la situación esperada, de modo de poder diseñar la forma de llegar del estado actual al estado objetivo (problema solucionado). Lo relevante de este punto es que el análisis es una tarea creativa que requiere de capacidades de abstracción y uso de lenguajes que faciliten la tarea de generación de modelos y análisis de los mismos. En el análisis del problema se deben contestar las siguientes preguntas: ¿Cuál es el problema? (¿Cuál es el estado actual considerado insatisfactorio?) ¿En qué forma se consideraría el problema solucionado? (¿Cuál es el estado deseado?) ¿Qué restricciones existen para llegar a esa solución?

2.2 Diseño de la Solución.

El diseño de la solución es el proceso por el cual se determina cuáles son los pasos a seguir para conseguir pasar desde el estado actual (problema) al estado deseado u objetivo. Muchas veces este proceso de diseño involucra a variados componentes, no sólo de índole informática, los cuales no son el objetivo de este curso. En algunas ocasiones, el modelo del estado objetivo es ya el diseño de la solución, como por ejemplo, en el caso de los sistemas software, el modelo de la situación objetivo se traslapa con la primera versión del diseño.

2.3 Ejecución

En la ejecución, se sigue el plan establecido en el diseño para pasar del estado inicial al estado objetivo (definido en el análisis). La ejecución del diseño es en muchos casos la construcción en un lenguaje de programación u otra herramienta de implementación computacional de lo diseñado en la etapa previa.

2.4 Control

En esta etapa, se verifica y valida lo ejecutado. Esto significa que se debe verificar que lo ejecutado corresponde a lo diseñado, y lo diseñado efectivamente permite alcanzar el estado objetivo definido en el análisis. La validación consiste en responder si esta es LA solución.

3 Dimensiones de un sistema basado en software.

El software es otro modelo de la realidad, definido en lenguajes particulares que incluyen códigos de programa, documentación, manuales de usuario, procedimientos administrativos, bases de datos, entre otros. El software no se produce en el sentido clásico, más bien, se desarrolla.

"...ESTA BASADO EN UN RECONOCIMIENTO DEL HECHO QUE,..., CUALQUIER PROGRAMA ES UN MODELO DE UN MODELO DENTRO DE UNA TEORÍA DE MODELOS PARA UNA ABSTRACCIÓN DE ALGUNA PORCIÓN DEL MUNDO O DE ALGÚN UNIVERSO DE DISCURSO." *MEIR LEHMANN, PROCEEDING OF THE IEEE VOL. 68 No. 9 SEPT. 1980.*

Podemos distinguir en un software tres dimensiones, o formas de verlo:

Con respecto al tipo de componente.

Interfaz. La interfaz es aquella componente del software que permite que un usuario interactúe con él. A través de este componente un software recibe las entradas del usuario y le entrega las respuestas. **Lógica de procesamiento.** Este componente es aquel que se preocupa de relacionar las entradas que le llegan a través de la interfaz con el repositorio, generando los resultados que el usuario espera. **Repositorio.** Este componente alberga a todos aquellos componentes que el software utiliza y almacena incluso después de ejecutarse (el repositorio es persistente en el tiempo). En este componente reside información que es persistente y que puede ser compartida con otros componentes de lógica de

procesamiento. **Con respecto al nivel de abstracción:** Conceptual. En este nivel de abstracción nos estamos enfrentando a una representación (modelo) muy cercana a la realidad a modelar, de manera independiente de la plataforma de implementación computacional. De alguna manera, es en este nivel donde deberían encontrarse los modelos que se realicen en la etapa de análisis. Lógico. Este nivel de abstracción se centra en los aspectos centrales del sistema, pero con una visión más cerca de la implementación en una plataforma definida (puede ser “un tipo de” plataformas). Los modelos generados en la etapa de diseño deberían encontrarse mayoritariamente en este nivel de abstracción.

Físico. En este nivel la abstracción ya es mínima. Estamos ante representaciones directas de la implementación de los sistemas. Modelos de nivel físico deberían documentar la etapa de ejecución. **Con respecto al tipo de modelamiento:**

Estático. El componente estático de un sistema software lo constituye la definición de todas sus estructuras, las cuales podrán tomar distintos valores. Dinámico. El componente dinámico de un software lo constituye su comportamiento, es decir, la definición de respuestas (cambios de estado) a ciertos estímulos (eventos).

Funcional. El componente funcional es aquel que define las transformaciones de las entradas, objetos del repositorio, eventos, etc., en respuestas o salidas del sistema software. Esta componente es aquella que realiza las transformaciones.

Estas tres dimensiones se entrecruzan y conforman una matriz de 3x3x3, en cuyas celdas podemos encontrar distintos lenguajes que apoyan el modelamiento de dicha dimensión.

Componente	Interfaz			Lógica de Procesamiento			Repositorio		
	de Nivel Abstracción			de Nivel de Abstracción			de Nivel de Abstracción		
Tipo Modelamiento	Concep tual	Lógi co	Físi co	Concep tual	Lógi co	Físic o	Concep tual	Lógi co	Físic o
Estático									
Dinámico									
Funcional									

Para cada una de estas celdas pueden existir uno o más lenguajes adecuados para modelar realidades vistas bajo el prisma de las tres dimensiones

consideradas.

4 Características de los Sistemas de Bases de Datos.

“UNA BASE DE DATOS CONSISTE EN ALGUNA COLECCIÓN DE DATOS PERSISTENTES E INDEPENDIENTES USADOS POR UNA ORGANIZACIÓN DETERMINADA.” (DATE, 1995)

Los sistemas de bases de datos se caracterizan por poseer una fuerte componente de repositorio. Incluso podemos considerar repositorio y base de datos como sinónimos. Los sistemas de bases de datos, al ser un repositorio, poseen componente estática, dinámica y funcional, y cada una de estas componentes puede ser vista desde los niveles de abstracción conceptual, lógico o físico. Podemos precisar un poco más en la composición de este caso particular de sistema basado en software. Dimensión: Tipo de Modelamiento

Tipo Componente	Componente Sistema Base de Datos	Descripción
Estático	Estructura	Estructuras que constituyen la base de datos.
Estático	Restricciones Estáticas	Reglas que restringen el conjunto de valores (estados) que la base de datos (estructura) puede tomar.
Dinámico	Restricciones Dinámicas	Reglas que restringen las transiciones entre valores (estados) válidos de la base de datos (estructura).
Funcional	Manipulación de los datos.	Definición de los procedimientos por los cuales la base de datos (estructura) cambia de un valor (estado) a otro.

5 Procesos de Abstracción en el modelamiento.

La abstracción es un proceso mental que se aplica al seleccionar algunas características y propiedades de un conjunto de objetos y excluir otras no pertinentes. En otras palabras, se hace una abstracción al fijar la atención en las propiedades consideradas esenciales de un conjunto de cosas y desechar sus diferencias.

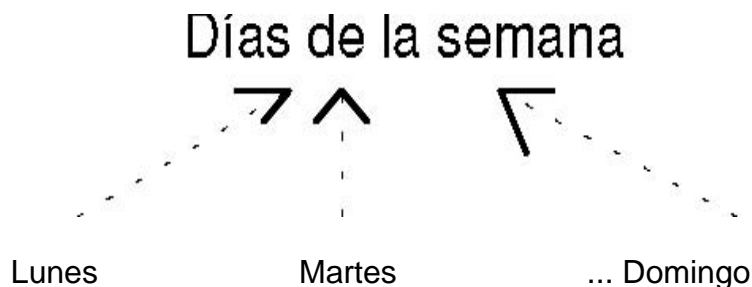
En el modelamiento de datos, se usan tres tipos de abstracciones: clasificación, agregación y generalización.

5.1 Abstracción de clasificación.

Se utiliza para definir un concepto como una clase de objetos de la realidad caracterizados por propiedades comunes.

Se representa gráficamente como un árbol de un nivel que tiene como raíz la clase y como hojas los elementos de la clase. Las ramas del árbol se representan por líneas discontinuas. Cada rama del árbol indica que un nodo hoja es un miembro (ES_MIEMBRO_DE) la clase que representa la raíz.

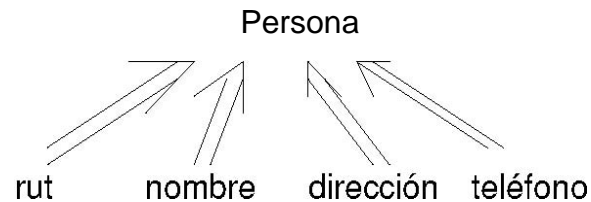
Un mismo objeto real puede clasificarse de varias maneras.



5.2 Abstracción de Agregación.

Define una nueva clase a partir de un conjunto de (otras, no necesariamente

distintas) clases que representan sus partes componentes. Se representa por un árbol de un nivel en el cual todos los nodos son clases; la raíz representa la clase creada por agregación de las clases representadas en las hojas. Cada rama del árbol indica que una clase hoja es una parte de (ES_PARTE_DE) la clase representada por la raíz. Para distinguirla de la agregación de clasificación, las ramas dirigidas están representadas por líneas dobles que van de los componentes a los objetos agregados.



La clasificación y la agregación son las dos abstracciones básicas utilizadas para construir estructuras de datos dentro de la base de datos y dentro de los lenguajes convencionales de programación. La clasificación es el procedimiento utilizado cuando, partiendo de elementos individuales de información, se identifican tipos de campos o atributos. La agregación es el procedimiento mediante el cual se reúnen tipos de campos relacionados en grupos como por ejemplo tipos de registros.

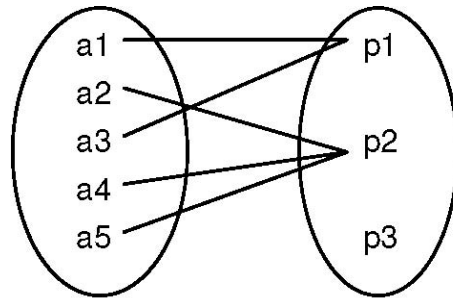
5.3 Abstracción de generalización.

Define una relación de subconjunto entre elementos de dos o más clases. Cada generalización se representa con un árbol de un nivel, en el que todos los nodos son clases, con la clase genérica como raíz y las clases subconjunto como hojas; cada rama del árbol expresa que una clase hoja es un (ES_UN) subconjunto de la clase raíz. Para distinguir la generalización de otras abstracciones, se usa una flecha sencilla apuntando hacia la raíz. Esta abstracción, a pesar de ser muy común e intuitiva, no se usa en muchos modelos de datos. Sin embargo es muy útil por su cualidad fundamental de herencia: en una generalización, todas las abstracciones definidas para la clase genérica son heredadas por las clases subconjunto.

6 Propiedades de las correspondencias entre las clases.

6.1 Agregación Binaria.

Es una correspondencia que se establece entre dos clases. Se puede representar una agregación binaria entre dos clases mediante la descripción de éstas como conjuntos y el trazado de una línea entre un elemento de cada conjunto para representar que están agregados.



Automóviles Personas Representación para la agregación "POSEE".

Al observar la figura, se puede decir que la persona p1 posee los autos a1 y a2, la persona p2 posee los autos a2, a4 y a5, mientras que la persona p3 no posee autos. De esto último se puede observar que no es obligatorio que todas las personas posean autos, pero al parecer todos los autos deben tener un dueño. Esta última característica es propia de cada agregación, y se refieren a la cardinalidad de correspondencia entre las clases.

6.1.1 Cardinalidad mínima.

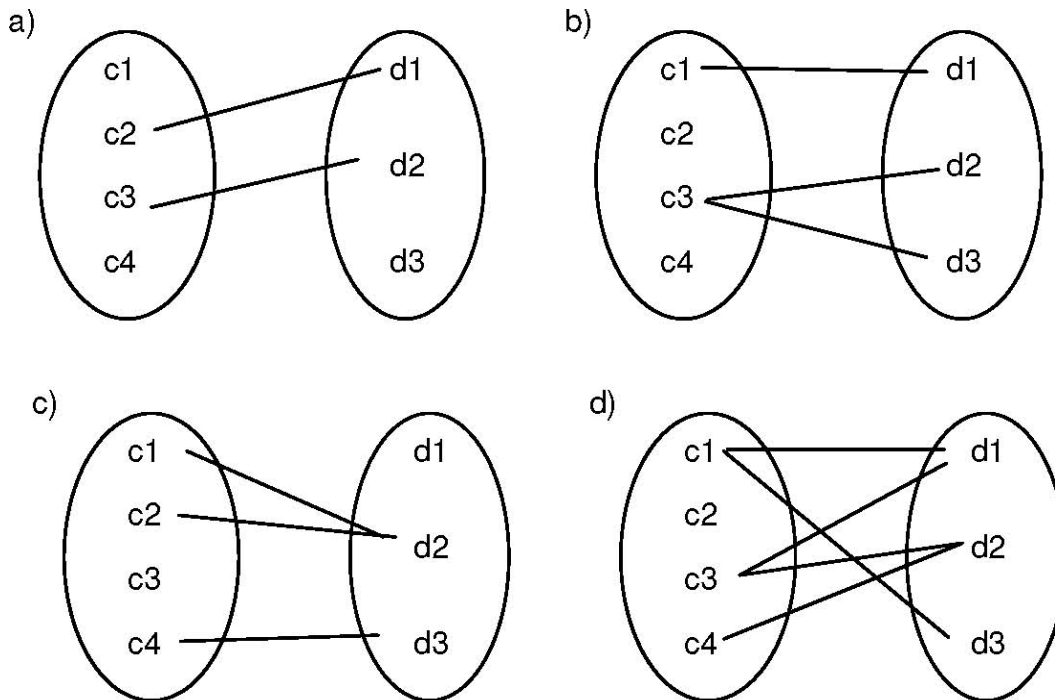
Consideremos una agregación A entre las clases C y D. La cardinalidad mínima de C en A, denotada por $\text{card-min}(C,A)$, es el menor número de correspondencias en las que cada elemento de C puede tomar parte. Análogamente se define $\text{card-min}(D,A)$.

Si $\text{card-min}(A,B)=0$, entonces se dice que la clase A tiene una participación opcional en la agregación B. Si $\text{card-min}(A,B)>0$, entonces se dice que la clase A tiene una participación obligatoria en la agregación B.

6.1.2 Cardinalidad máxima.

Consideremos una agregación A entre las clases C y D. La cardinalidad máxima de C en A, denotada por $\text{card-máx}(C,A)$, es el mayor número de correspondencias en las que cada elemento de C puede tomar parte. Análogamente se define $\text{card-máx}(D,A)$.

Si $\text{card-max}(C,A)=1$ y $\text{card-max}(D,A)=1$, se dice que la agregación es de uno a uno. Si $\text{card-max}(C,A)=n$ y $\text{card-max}(D,A)=1$, se dice que la agregación es de uno a muchos. Si $\text{card-max}(C,A)=1$ y $\text{card-max}(D,A)=n$, se dice que la agregación es de muchos a uno. Si $\text{card-max}(C,A)=n$ y $\text{card-max}(D,A)=m$, se dice que la agregación es de muchos a muchos. Nótese que n y m representan valores mayores que 1.



a) Agregación Uno a Uno. $\text{Card}(C,A)=(x,1)$ y $\text{Card}(D,A)=(x,1)$, con x en $\{0,1\}$.

b) Agregación Uno a Muchos. $\text{Card}(C,A)=(x,n)$ y $\text{Card}(D,A)=(y,1)$, con x en $\{0,1,\dots,n\}$ e y en $\{0,1\}$.

c) Agregación Muchos a Uno. $\text{Card}(C,A)=(x,1)$ y $\text{Card}(D,A)=(y,n)$, con x en $\{0,1\}$ e y en $\{0,1,\dots,n\}$.

d) Agregación Muchos a Muchos. $\text{Card}(C,A)=(x,n)$ y $\text{Card}(D,A)=(y,m)$, con x e y en $\{0,1,\dots,n\}$, n y m valores indefinidos mayores que uno.

6.2 Agregación n-aria.

Es una correspondencia establecida entre tres o más clases. Se mantiene las definiciones de cardinalidades máxima y mínima.

Cardinalidad Mínima. Consideremos una agregación A entre las clases C_1, C_2, \dots, C_n . La cardinalidad mínima de C_i en A , denotada por $\text{card-min}(C_i, A)$, es el menor número de correspondencias en las que cada elemento de C_i puede tomar parte.

Cardinalidad Máxima. Consideremos una agregación A entre las clases C_1, C_2, \dots, C_n . La cardinalidad máxima de C_i en A , denotada por $\text{card-max}(C_i, A)$, es el mayor número de correspondencias en las que cada elemento de C_i puede tomar parte.

6.3 Generalizaciones.

Una abstracción de generalización establece una correspondencia entre la clase genérica (raíz) y las clases subconjunto. Considérese la clase Persona como una generalización de las clases Mujer y Hombre; cada elemento de éstas corresponde exactamente a un elemento de la clase Persona. En esta generalización, cada elemento de la clase Persona corresponde a un elemento de la clase Mujer o la clase Hombre, pero nunca a ambas. Esto es una característica de cada generalización y se denomina cobertura.

6.3.1 Cobertura total o parcial.

La cobertura de una generalización es total (t) si cada elemento de la clase genérica corresponde al menos a un elemento de las clases subconjunto; es parcial (p) si existe algún elemento de la clase genérica que no corresponde a ningún elemento de las clases subconjunto.

6.3.2 Cobertura exclusiva o superpuesta.

La cobertura de una generalización es exclusiva (e) si cada elemento de la clase genérica corresponde, a lo más a un elemento de las clases subconjunto; es

superpuesta (s) si existe algún elemento de la clase genérica que corresponde a elementos de dos o más clases subconjunto diferentes.

Persona

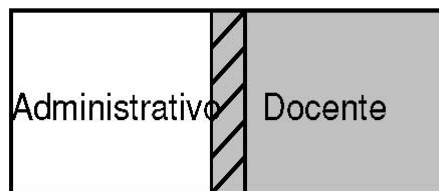


∴ las personas son Hombres o Mujeres, pero no ambos.

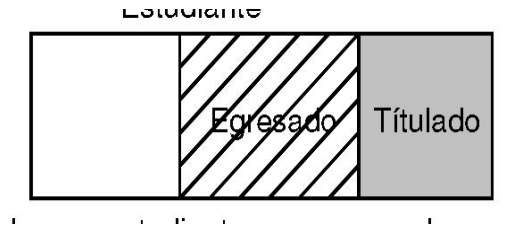
a) total, exclusiva. Todas

b) total, superpuesta. Todos los empleados son Administrativos o Docentes, pudiendo haber empleados desempeñando ambas funciones.

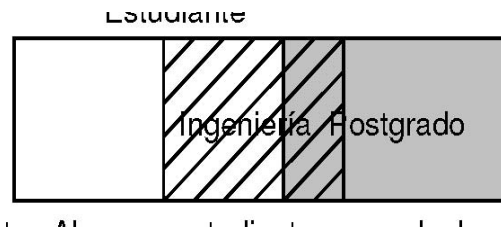
Empleado



c) parcial, exclusivo. Algunos estudiantes son egresados, mientras que otros están titulados, pero no hay ningún estudiante en ambas situaciones.



d) parcial, superpuesta. Algunos estudiantes son de Ingeniería y otros son de postgrado, y hay algunos estudiantes que son de ingeniería y también participan en postgrado.



7 Concepto de Dato y Modelo de Dato.

7.1 El significado de dato.

La percepción del mundo puede ser descrita como una sucesión de fenómenos. Desde el comienzo de los tiempos el hombre ha tratado de descubrirlos, ya sea que los entienda completamente o no.

La descripción de estos fenómenos es llamada DATO. Los datos corresponden al registro discreto (no continuo) de hechos acerca de un fenómeno, con lo cuál ganamos información acerca del mundo que nos rodea (Información: incremento del conocimiento que puede ser inferido de los datos).

Usualmente el dato y su significado son registrados juntos, ya que el lenguaje natural es lo suficientemente poderoso para hacerlo. Por ejemplo "el kilo de pan cuesta \$460" registra el valor (460) y su significado o semántica (valor del kilo de pan en pesos).

En ciertos casos los datos están separados de su semántica. Por ejemplo, una planilla de notas es una tabla de datos. Su interpretación está implícita y se supone que quien la lee conoce su significado.

El uso del computador para procesar datos ha traído consigo una mayor separación entre los datos y su interpretación. Mucha de la interpretación de los datos está explícita. Consideremos por ejemplo un programa que calcula integrales definidas, este programa recibe valores de entrada y genera valores como salida. Sin embargo, el programa en si no tiene conocimiento si el problema resuelto es de termodinámica o electromagnetismo.

Han habido dos razones para separar los datos de su significado:

- . • los computadores no manejan (bien) el lenguaje natural, que es la mejor forma de dar interpretación y significado a un dato.
- . • el almacenamiento del significado de los datos ocupa espacio, e inicialmente este era escaso y costoso.

Así, tradicionalmente la interpretación de los datos se deja al usuario y al sistema manual externo al computador.

En muchos sistemas la interpretación de datos se encuentra en los programas que hacen uso de ellos, de modo que los datos pasan a ser una simple colección de valores.

Por otra parte, supongamos que algo de la semántica de los datos se codifica junto con ellos . Así los datos no son solo valores, sino que también tienen una semántica, y los datos están más cerca de la interpretación del mundo. Ellos forman una "vista" del mundo, la que no es exacta ni concreta, sino que usualmente es bastante abstracta.

Los datos no son estáticos, y corresponden a un mundo que está en constante cambio. La flexibilidad en la interpretación de los datos permite capturar los aspectos dinámicos del mundo y al mismo tiempo, proveer una estructura estable

para los datos. Esta flexibilidad se puede tener de dos formas:

- El sistema puede permitir que los mismos datos sean vistos de diferente forma. Por ejemplo, diferentes aplicaciones puedan usar los mismos datos y dar su propia semántica.
- Diferentes datos pueden ser vistos de la misma forma. Por ejemplo, se quiere ver a los gerentes, secretarías y empleados sólo como trabajadores de una organización, no importando su cargo. Aquí la interpretación debe ser lo suficientemente abstracta para que diferentes vistas del mundo se vean de la misma forma.

7.2 Modelamiento de Datos.

Es aparente que una interpretación del mundo es necesaria, la que debe ser suficientemente abstracta para que no sea afectada por la dinámica del mundo (los pequeños cambios), y debe ser suficientemente robusta para poder representar como los datos y el mundo se relacionan. Una herramienta como esta es llamada *modelo de datos*, el cual permite representar en forma más o menos razonable alguna realidad. El modelo de datos permite realizar abstracciones del mundo, permitiendo centrarse en los aspectos macros, sin preocuparse de las particularidades; así nuestra preocupación se centra en generar un esquema de representación, y no en los valores de los datos.

Los modelos de datos nos permiten capturar parcialmente el mundo, ya que es improbable generar un modelo que lo capture totalmente.

Sin embargo, se puede tener un conocimiento relativamente completo de la parte del mundo que nos interesa. Así, un modelo captura la cantidad de conocimiento tal que cumpla con los requerimientos que nos hemos impuesto previamente.

DATO: la siguiente tupla: <nombre del objeto, propiedad del objeto, valor, tiempo>
--

Esta definición es correcta, ya que cada vez que se describe un fenómeno, éste se refiere a un objeto (nombre del objeto) y ciertas características (propiedades del objeto) el cual tiene un valor en un momento determinado (tiempo).

- ◆ Ejemplo. El *precio del pan* es \$460 . nombre: precio del pan propiedades : (unidad, \$), entero no negativo. valor: 460 tiempo: hoy

En general, el modelar un objeto no se considera el tiempo, sino que éste se considera implícito en la semántica de él.

◆ Ejemplo. Consideremos el caso de una matriz:

nombre: matriz_coeficiente propiedades:

+, -, *, a[i,j] ◆ R valor : [1 2]

[3 4]

ESQUEMA

Para una aplicación particular de un modelo de datos, el modelamiento de la realidad se llama esquema.

Un esquema es una definición genérica que identifica categorías (ejemplo: libro, autor, etc.), sus propiedades (nombre, título) y sus relaciones (escrito).

Por ejemplo, un modelo de datos simple es una archivo (tabla). Aplicando este modelo a una situación particular se puede tener el siguiente esquema:

Persona (Nombre, Edad, Dirección), donde Persona es el nombre genérico de una entidad, y Nombre, Edad y Dirección son nombres genéricos para los atributos.

7.2.1 Modelo de Datos.

Un modelo de datos define las reglas por las cuales los datos son estructurados. Esta estructuración, sin embargo, no da una interpretación completa acerca del significado de los datos y la forma en que serán usados. Las operaciones que se permiten efectuar a los datos deben ser definidos.

◆ Ejemplo. Una lista puede ser tratada como pila o fila, dependiendo de las operaciones que se permitan sobre ella.

Generalmente las operaciones están relacionadas con la estructura de los datos y tienen validez en el contexto en que fueron definidos.

BASE DE DATOS

Una colección de datos estructurados en una forma particular, según un esquema particular, se denomina base de datos.

Todo modelo de datos debe poder capturar las propiedades estáticas y dinámicas de una realidad. Las propiedades estáticas corresponden a lo que es relativamente invariante en el tiempo, son siempre verdadero y no cambia en el tiempo. ♦ Ejemplo. Que los precios se midan en \$ es relativamente invariante.

Las propiedades dinámicas corresponden a la naturaleza evolutiva del mundo. Por esto, para todo modelo debe ser posible capturar los dos tipos de propiedades.

MODELO DE DATOS

Se define el modelo de datos M consistente de dos partes:

G : un conjunto de reglas que lo generan.

O : un conjunto de operaciones.

El conjunto de reglas G expresan las propiedades estáticas de un modelo de datos y corresponden a lo que se denomina generalmente Data Definition Language (DDL). Este define las estructuras permitidas para el modelo de datos M .

El conjunto G se puede dividir en dos:

- G_s : las estructuras permitidas.
- G_c : las restricciones del modelo.

Así, G_s genera las categorías y estructuras para un modelo, y G_c las restricciones.

Utilizando esta última notación, un esquema S consiste de dos partes: una estructura S_s y restricciones S_c , donde S_c es una lista explícita de restricciones que no deben ser violadas.

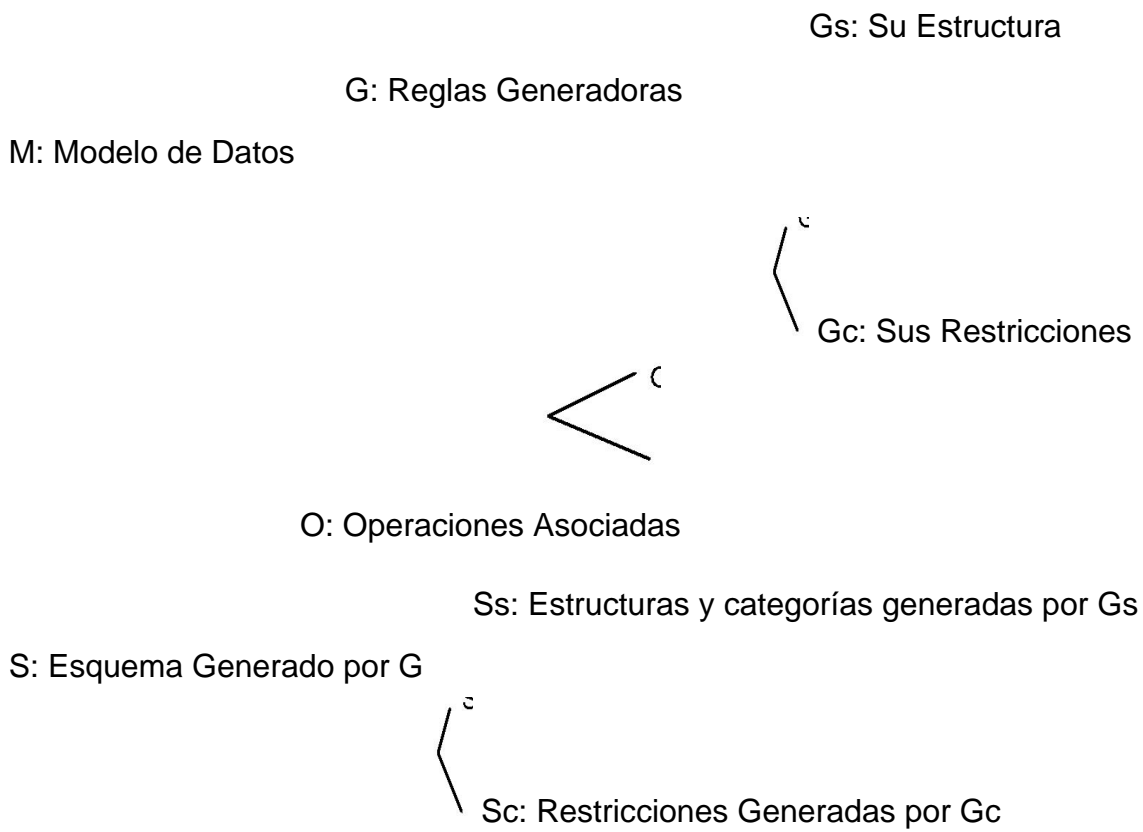
Por ejemplo, en la definición de la entidad persona, un atributo particular CI (Cédula de Identidad) puede ser declarado como clave, esto es, en un instante dado no puede haber dos personas con el mismo valor para CI . S_s no prohíbe dos ocurrencias, pero S_c sí.

Un modelo de datos también puede tener restricciones que son inherentes a él, las que generalmente se incorporan en Ss (la estructura).

◆ Ejemplo, sólo se permite relaciones entre objetos mediante una estructura de árbol.

Las reglas de generación G son generadoras de un conjunto de esquemas S, en el que cada uno de ellos define estructuras y restricciones particulares para los datos. Hoy muchas bases de datos D en términos de la ocurrencia del esquema S, pero todos tienen la misma estructura genérica y obedecen a las mismas restricciones definidas en S.

En resumen:



D: Base de Datos ◆◆ Ocurrencia del esquema S. Las restricciones Sc deben ser siempre válidas, para toda ocurrencia D del esquema S.

7.2.2 Operaciones.

Las propiedades dinámicas de un modelo de datos son expresadas por un conjunto de operaciones O , las que generalmente son llamadas Data Manipulation Language (DML). Estas propiedades definen las acciones permitidas para una base de datos, tal que transforme la ocurrencia D_i en la ocurrencia D_j .

No todas las operaciones definidas en O causan cambios en la base de datos, pero si causan un cambio en el estado de ella. El estado de una base de datos no es un objeto de ella, pero está asociado a la base de datos, y cambia como resultado de una operación.

◆ Ejemplo. Consideremos el modelo de datos tipo Tabla; este modelo tiene como una de sus operaciones la operación "read", que permite recorrer la tabla en forma secuencial. Esta operación no cambia el contenido de la Base, pero si su estado (cambia el registro actual, que ahora es el siguiente).