



Preparador Informática

www.preparadorinformatica.com

TEMA 50. INFORMÁTICA

**ANÁLISIS DE SISTEMAS:
MODELIZACIÓN CONCEPTUAL DE
DATOS. TÉCNICAS DESCRIPTIVAS.
DOCUMENTACIÓN**

TEMA 50 INF: ANÁLISIS DE SISTEMAS: MODELIZACIÓN CONCEPTUAL DE DATOS. TÉCNICAS DESCRIPTIVAS. DOCUMENTACIÓN

1. INTRODUCCIÓN

2. MODELO DE DATOS

3. ANÁLISIS DE SISTEMAS: MODELIZACIÓN CONCEPTUAL DE DATOS

4. TÉCNICAS DESCRIPTIVAS PARA EL MODELO CONCEPTUAL DE DATOS

4.1. MODELO ENTIDAD/RELACIÓN (E/R)

4.1.1. ENTIDADES

4.1.2. RELACIONES

4.1.3. ATRIBUTOS

4.2. MODELO E/R EXTENDIDO

5. DOCUMENTACIÓN

6. CONCLUSIÓN

7. BIBLIOGRAFÍA



1. INTRODUCCIÓN

Con el incremento de la complejidad de los programas informáticos, su generalización y la competencia en la industria del software se hizo necesario desarrollar sistemas de calidad que potenciases aspectos como la reutilización, la eficiencia o la documentación del software desarrollado. En definitiva, aplicar criterios de ingeniería al desarrollo de software, utilizando metodologías, herramientas y criterios de desarrollo para mejorar los tiempos del desarrollo y la calidad de los resultados. Esto dio lugar a la “ingeniería del software”.

La **ingeniería de software** es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el **desarrollo de las aplicaciones informáticas (software)**.

La ingeniería de software incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema. La fase de análisis es una de las más importantes en la ingeniería del software, de ahí la importancia que tiene el tema actual. En este tema nos centraremos en esta fase de Análisis de sistemas donde vamos a ver cómo realizar la modelización conceptual de datos, distintas técnicas descriptivas y conceptos relacionados con la documentación del proceso de análisis de datos, pero previamente partiendo del concepto de modelo de datos.

2. MODELO DE DATOS

En informática, un **modelo de datos** es un lenguaje utilizado para describir las estructuras de los datos (tipos de datos y relaciones entre ellos), las restricciones de integridad (condiciones que deben cumplir los datos, según las necesidades de nuestro modelo basado en la realidad) y las operaciones de manipulación de los datos (insertado, borrado, modificación de datos). Para clasificar los modelos debemos pensar en el nivel de abstracción, es decir, en lo alejado que esté del mundo real por lo que encontramos modelos de datos conceptuales, lógicos y físicos.

En este tema nos centramos en los modelos de datos **conceptuales** que son aquellos que describen las estructuras de datos y restricciones de integridad. Se

utilizan durante la etapa de análisis de un problema dado, y están orientados a representar los elementos que intervienen y sus relaciones.

3. ANÁLISIS DE SISTEMAS: MODELIZACIÓN CONCEPTUAL DE DATOS.

Las primeras tareas que se llevan a cabo en cualquier proyecto, independientemente del ciclo de vida utilizado, son las referentes al análisis del sistema. El análisis del software se centra en tres aspectos:

- Identificar y representar los datos que manejar el sistema y sus relaciones.
- Modelar los tratamientos que se realizarán sobre esos datos.
- Analizar las distintas funcionalidades que debe implementar la aplicación.

El análisis conceptual de datos constituye la primera fase del diseño de los datos que manejará el sistema y proporciona una visión fija de la información del dominio del problema. Es el proceso en que se representa la estructura estática de las entidades que representan los datos y sus relaciones.

Los datos de un sistema se almacenan habitualmente en una base de datos. La finalidad del análisis conceptual de datos es generar un esquema conceptual a partir de los requisitos, que represente los elementos del universo y facilite su posterior traducción a un modelo de datos que se pueda implementar en un sistema de base de datos o sistema de información donde se vayan a almacenar los datos del sistema. El esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, y es independiente del tipo de SGBD que se vaya a utilizar. El más importante y utilizado actualmente es el modelo Entidad-Relación y el modelo Entidad-Relación extendido que es una variante del anterior (ambos los explicaremos en apartados posteriores de manera detallada).

Para construir un modelo conceptual es necesario partir de la información recopilada acerca del dominio del problema e identificar las distintas entidades que contendrán los datos que manejará el sistema. Para ello se seguirán una serie de etapas: Identificar las entidades, identificar las relaciones que se establecen entre las entidades, identificar los atributos de las entidades y el dominio de estos atributos, identificar los campos clave de cada entidad, determinar las jerarquías de generalización entre entidades si las hubiera,

generar el modelo y, por último, revisar el modelo conceptual con el usuario para su validación.

4. TÉCNICAS DESCRIPTIVAS PARA EL MODELO CONCEPTUAL DE DATOS

Existen distintas técnicas descriptivas para realizar el modelo conceptual de datos, pero sin duda la más importante en la actualidad es el modelo Entidad/Relación (E/R) y el modelo Entidad/Relación extendido en los cuales nos centraremos en este apartado.

4.1. MODELO ENTIDAD/RELACIÓN (E/R).

El modelo E/R es uno de los modelos de datos conceptuales más extendidos en las metodologías de diseño de base de datos. Fue propuesto por Peter P. Chen en 1976 aunque existen otros autores que han investigado y propuesto posteriormente nuevas aportaciones sobre el mismo.

El modelo E/R puede ser usado como una base para una vista unificada de los datos adoptando el enfoque más natural del mundo real que consiste en entidades y relaciones. Se basa en llegar a un nivel de abstracción que permita definir los elementos que componen nuestra base de datos. Se componen básicamente de **Entidades** (E) y **Relaciones** (R), donde ambas contienen atributos que almacenan información.

4.1.1. ENTIDADES

Una entidad es un objeto real o abstracto sobre el que queremos almacenar información en nuestra base de datos. (Ej: una persona). La estructura genérica de un conjunto de entidades con las mismas características se denomina tipo de entidad. Existen dos clases de entidades: **regulares**, que tienen existencia por sí mismas, y **débiles** cuya existencia depende de otra entidad. Las entidades deben cumplir las siguientes tres reglas:

- Tienen que tener existencia propia.
- Cada ocurrencia de un tipo de entidad debe poder distinguirse de las demás.
- Todas las ocurrencias de un tipo de entidad deben tener las mismas características (atributos).

La representación gráfica de un tipo de entidad regular es un rectángulo etiquetado con el nombre del tipo de entidad. Un tipo de entidad débil se representa con dos rectángulos concéntricos con su nombre en el interior.



4.1.2. RELACIONES

Una relación es una asociación entre entidades, sin existencia propia en el mundo real que estamos modelando, pero necesaria para reflejar las interacciones existentes entre entidades. La estructura genérica de un conjunto de relaciones denomina tipo de entidad.

La relación puede ser regular, si asocia tipos de entidad regulares, o débil, si asocia un tipo de entidad débil con un tipo de entidad regular. Dentro de las relaciones débiles se distinguen la dependencia en existencia y la dependencia en identificación.

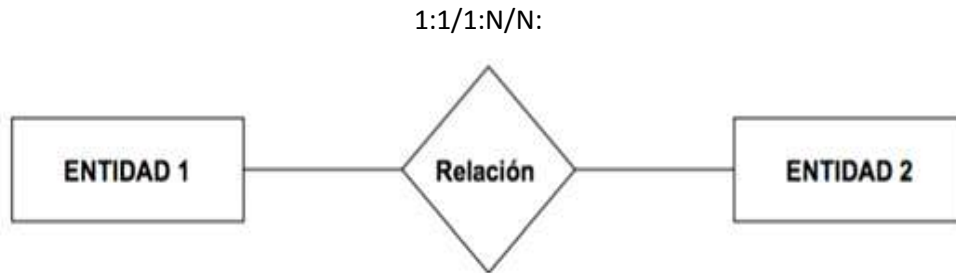
Se dice que la dependencia es en existencia cuando las ocurrencias de un tipo de entidad débil no pueden existir sin la ocurrencia de la entidad regular de la que dependen. Se dice que la dependencia es en identificación cuando, además de lo anterior, las ocurrencias del tipo de entidad débil no se pueden identificar sólo mediante sus propios atributos, sino que se les tiene que añadir el identificador de la ocurrencia de la entidad regular de la cual dependen.

Una relación se caracteriza por:

- **Nombre:** que lo distingue unívocamente del resto de relaciones del modelo.
- **Tipo de correspondencia:** es el número máximo de ocurrencias de cada tipo de entidad que pueden intervenir en una ocurrencia de la relación que se está tratando. Conceptualmente se pueden identificar tres clases de relaciones:
 - Relaciones 1:1: Cada ocurrencia de una entidad se relaciona con una y sólo una ocurrencia de la otra entidad.
 - Relaciones 1:N: Cada ocurrencia de una entidad puede estar relacionada con cero, una o varias ocurrencias de la otra entidad.
 - Relaciones M:N: Cada ocurrencia de una entidad puede estar relacionada con cero, una o varias ocurrencias de la otra entidad y cada ocurrencia de

la otra entidad puede corresponder a cero, una o varias ocurrencias de la primera.

Se representa por un rombo unido a las entidades relacionadas por dos líneas rectas a los lados. El tipo de correspondencia se representa gráficamente con una etiqueta 1:1, 1:N o M:N, cerca de alguno de los vértices del rombo.



El **grado de una relación** es el número de entidades que participan en una relación. En función del grado se pueden establecer diferentes tipos de relaciones: unarias, binarias, ternarias y n-arias.

4.1.3. ATRIBUTOS

Es una propiedad o característica de un tipo de entidad o de un tipo de relación. Se trata de la unidad básica de información que sirve para identificar o describir la entidad. Un atributo se define sobre un dominio. Cada tipo de entidad ha de tener un conjunto mínimo de atributos que identifiquen unívocamente cada ocurrencia del tipo de entidad. Este atributo o atributos se denomina identificador principal o clave primaria. Ya que puede haber varias claves y necesitamos elegir una, lo haremos atendiendo a estas normas:

- Que sea única.
- Que se tenga pleno conocimiento de ella.
- Que sea mínima, ya que será muy utilizada por el gestor de base de datos.

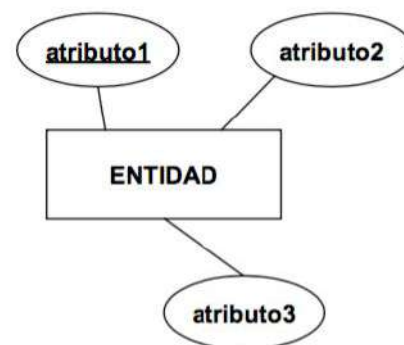
Se pueden definir restricciones sobre los atributos, según las cuales un atributo puede ser:

- Univaluado, atributo que sólo puede tomar un valor para todas y cada una de las ocurrencias del tipo de entidad al que pertenece.
- Multivaluado: Es aquel que tiene más de una ocurrencia para un determinado valor de la clave.

- Obligatorio, atributo que tiene que tomar al menos un valor para todas y cada una de las ocurrencias del tipo de entidad al que pertenece.

Un atributo se representa mediante una elipse, con su nombre dentro, conectada por una línea al tipo de entidad o relación.

Para representar las claves primarias se subrayarán aquellos atributos que la formen.



4.2. MODELO E/R EXTENDIDO

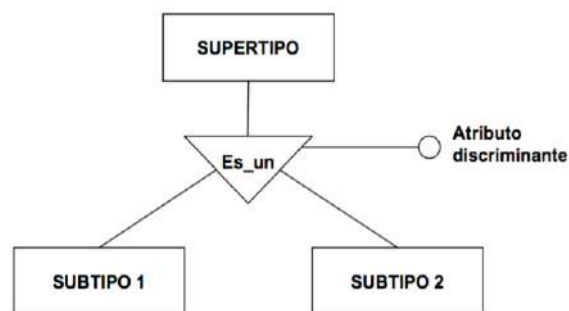
Además de los elementos comentados en apartados anteriores, existen extensiones del modelo entidad/relación que incorporan determinados conceptos o mecanismos de abstracción para facilitar la representación de ciertas estructuras del mundo real.

La **cardinalidad**: representa la participación en la relación de cada una de las entidades afectadas, es decir, el número máximo y mínimo de ocurrencias de un tipo de entidad que pueden estar interrelacionadas con una ocurrencia de otro tipo de entidad. La cardinalidad máxima coincide con el tipo de correspondencia. La representación gráfica de las cardinalidades se realiza mediante una etiqueta del tipo (0,1), (1,1), (0,n) o (1,n), que se coloca en el extremo de la entidad que corresponda. Si se representan las cardinalidades, la representación del tipo de correspondencia es redundante.



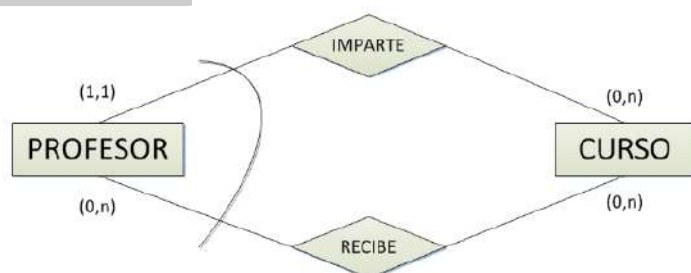
La **generalización** permite abstraer un tipo de entidad de nivel superior (supertipo) a partir de varios tipos de entidad (subtipos); en estos casos los atributos comunes y relaciones de los subtipos se asignan al supertipo. Por ejemplo: generalizar los tipos hombre y mujer obteniendo el supertipo persona.

La **especialización** es la operación inversa a la generalización, en ella un supertipo se descompone en uno o varios subtipos, los cuales heredan todos los atributos y relaciones del supertipo, además de tener los suyos propios. Un ejemplo es el caso del tipo persona, del que se pueden obtener los subtipos licenciado, ingeniero y doctor.

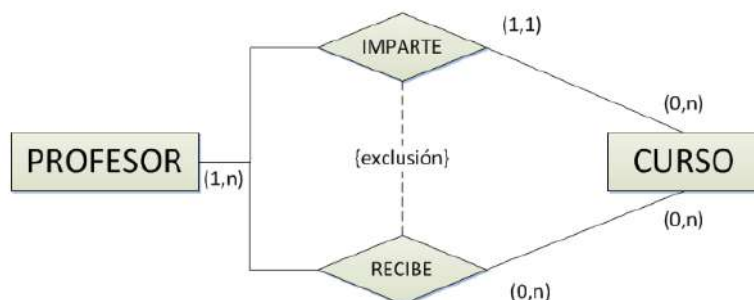


La existencia de supertipos y subtipos, en uno o varios niveles, da lugar a una jerarquía, que permitirá representar una restricción del mundo real. La representación de las jerarquías se realiza mediante un triángulo invertido, con la base paralela al rectángulo que representa el supertipo y conectando a éste y a los subtipos. Si la división en subtipos viene determinada en función de los valores de un atributo discriminante, éste se representará asociado al triángulo que representa la relación.

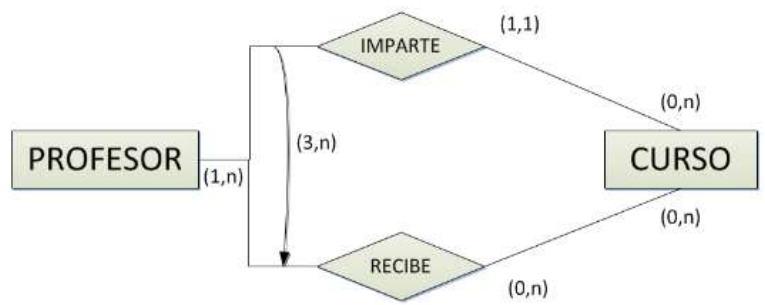
Exclusividad: Entre dos tipos de entidades puede existir más de un tipo de relación. Aunque se dice que una relación es exclusiva cuando la existencia de una relación entre dos tipos de entidades implica la no existencia de las otras relaciones.



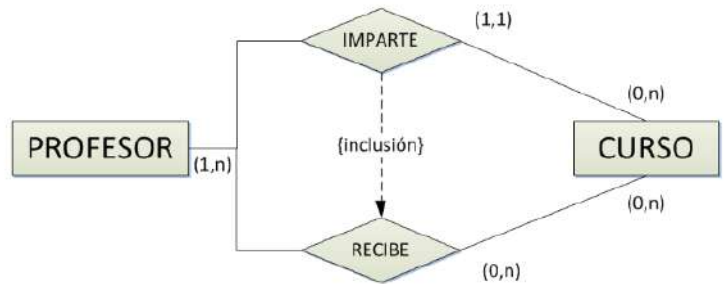
Exclusión: Entre dos tipos de relaciones R1 y R2. Significa que E1 está relacionada con E2 bien mediante R1, o bien mediante R2 pero que no pueden darse ambas relaciones simultáneamente.



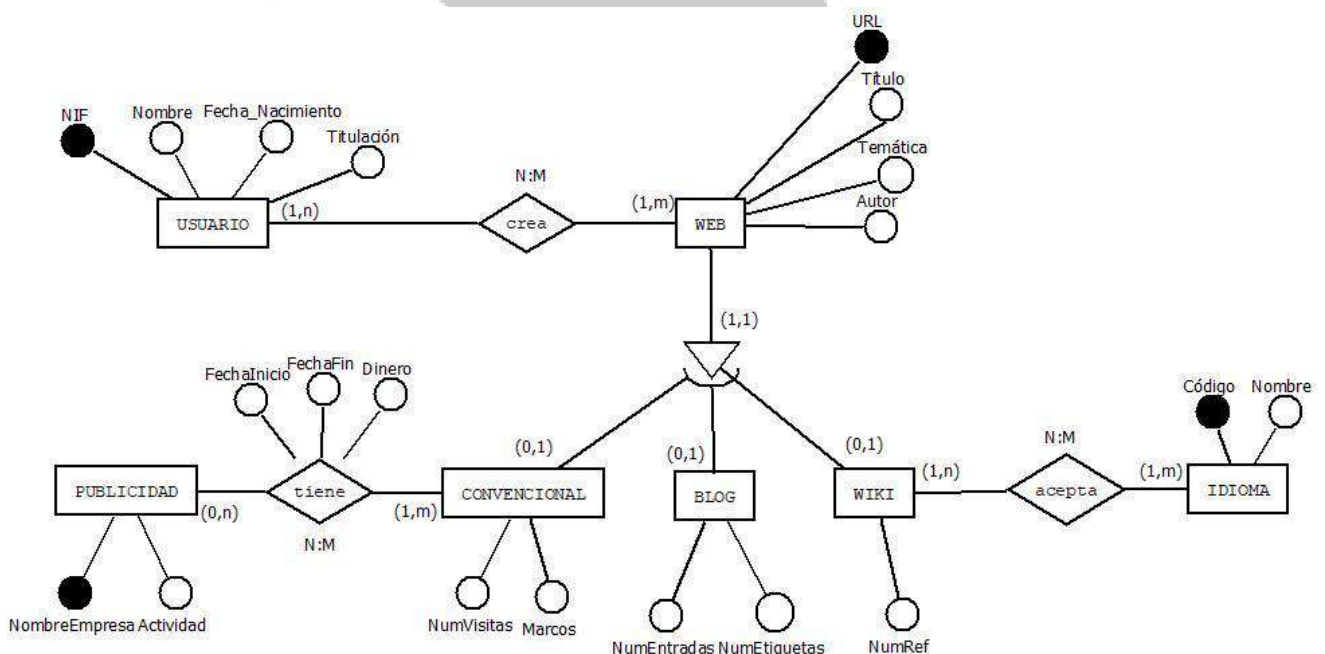
Inclusividad entre dos tipos de relaciones R1 y R2 respecto a la entidad E1. Para que la entidad E1 participe en la relación R2 debe participar previamente en la relación R1.



Inclusión entre dos tipos de relaciones R1 y R2. Para que la entidad E1 participe en la relación R2 con E2 debe participar previamente en la relación R1.



Para terminar este apartado voy a mostrar un ejemplo de diagrama entidad relación:



Existen distintas notaciones para representar los elementos del modelo entidad relación como por ejemplo IE ó UML cada una de ellas con sus propias particularidades, aunque en ellas seguimos utilizando el modelo teórico de Chen y en general los elementos que permiten representar son comunes a los estudiados.

5. DOCUMENTACIÓN

La documentación de software es una etapa transversal a todas las demás etapas del diseño: se iniciará en los primeros pasos del ciclo de vida y continuará hasta la entrega final del producto. Cada fase, proyecto e infraestructura tendrá sus propios procedimientos de documentación, pero hay una serie de manuales y documentos que son comunes a las diferentes etapas de los ciclos de vida.

Para poder realizar la modelización conceptual de datos hemos partido de la documentación generada en las etapas anteriores del proceso de desarrollo y ahora en este proceso también se generará documentación que a su vez se utilizará en fases posteriores, especialmente en el diseño lógico de datos para la elaboración del modelo lógico de datos.

La documentación más destacable en esta fase es la siguiente:

- Modelo conceptual de datos del sistema (habitualmente representado con el diagrama entidad relación o diagrama entidad relación extendido).
- Además, se deben incluir las especificaciones que no pueden ser representadas en el diagrama en documentos que den, por ejemplo, más información sobre las entidades (si queremos usar un alias para las entidades, por ejemplo) ó más información sobre los atributos (el dominio de los mismos, por ejemplo).
- Diccionario de datos: El diccionario de datos es la “base de datos” del sistema: en él se registrará toda la información manejada por el sistema junto con su significado, relaciones, restricciones y cualquier otra información relevante acerca de cada uno de los datos. Este diccionario de datos es un documento que se irá refinando a lo largo del proceso de desarrollo del software

6. CONCLUSIÓN

Una de las fases más importantes en la ingeniería del software es el análisis. Si esta fase se realiza adecuadamente tenemos grandes opciones de que nuestro software cumpla correctamente con los requerimientos del cliente y por lo tanto tengamos un software exitoso.

En esto radica la gran importancia de este tema ya que es imprescindible dominar y realizar adecuadamente la modelización conceptual de datos, ser capaz de utilizar las técnicas descriptivas más importantes para ello y documentar adecuadamente la fase para que en fases posteriores podamos avanzar en el ciclo de desarrollo del software de manera adecuada.

7. BIBLIOGRAFÍA

- Pressman, Roger. ***Ingeniería del software: un enfoque práctico***. Editorial McGraw-Hill
- Piattini, Mario G. y otros. ***Análisis y diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software***. Editorial Ra-Ma.
- De Miguel A, y Piattini M: **Fundamentos y modelos de BBDD**. Edit. Ra-Ma
- https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm
- <https://www.infor.uva.es/~jvalvarez/docencia/pt7seccion3.pdf>