

Herramientas de modelamiento

Felipe Alfonso González L.

Continuidad en Ingeniería en Informática, IACC Chile., 2020-2022

Instituto Superior de Artes y
Ciencias de la Comunicación,
IACC
Av. Salvador 1318, Metro Santa
Isabel, Providencia, Santiago.
Chile.

f.alfonso@res-ear.ch – felipe.alfonso.glz@gmail.com – <https://twitter.com/felipealfonsog>
<https://glzengrg.com> - <https://freeshell.de/felipe> - <https://linkedin.com/in/felipealfonsog>

Herramientas de modelamiento

Introducción.

El principal **objetivo del modelado de datos es comprender** el flujo y todos los elementos que componen **la gestión del** proceso computacional, visto a nivel informático, el modelo de datos se transformará en variables que se programarán **durante el desarrollo de cualquier** diseño de software. **proyecto.**

En este sentido, **comprender** los datos y comprender los elementos **básicos de un modelo de datos es fundamental** para **traducirlos en una estructura o** modelo que se implementará en **la base de datos.** Esta semana se definirá UML como **el lenguaje estándar para el modelado de datos, con el objetivo** de adquirir **habilidades** para modelar **en** forma de trabajo que se **adapte** a las necesidades de las organizaciones **actuales.**

1.- ¿Qué tipo de modelado sugiere de los vistos en el contenido de la semana? Argumente su planteamiento.

¿Que es modelo relacional?

Modelo relacional. Aquí, los datos se estructuran utilizando relaciones o estructuras matemáticas similares a una cuadrícula que tienen columnas y filas. Básicamente, es una tablaⁱ.

El modelo que seria ideal en el software que tenemos como desafío para su administración es el modelo relacional.ⁱⁱ

Estructura.

En un modelo relacional, las tablas se encuentran separadas por estructuras físicas de almacenamiento. Estas estructuras pueden ser administradas y

gestionadas de tal manera que no se vea afectado el acceso a los datos mediante una aplicación. Existen 2 tipos de modelos relacionales; el logio y el físico. Las operaciones lógicas permiten que una aplicación especifique el contenido que necesita, mientras que las operaciones físicas determinan cómo se debe acceder a esos datos y luego realizar la tarea.

Fueron creados en 1970 por Edgar Frank Codd, empleado de IBM en San José (California). Su funcionamiento es realmente sencillo. Se basan en la teoría de conjuntos y la lógica de predicados.

Están representados por tablas de datos, **por lo que** las filas son **casos** diferentes y las columnas **son los** campos **estudiados.** Es **ampliamente** utilizado para **la gestión dinámica de datos.**

Términos formales del modelo relacional

Existe una amplia gama de términos formales que corresponden **a** expresiones informales. **Es conveniente** conocerlos para **conocerlos.** En la **práctica, utilizamos** expresiones **simples que son** más fáciles de entender.

La relación, que es el término formal, tiene **un** equivalente informal **en la tabla.**

Una tupla no es más que un registro **representado** en las filas de la tabla y el atributo es una columna o **un campo.**

La cardinalidad se refiere al número de filas o registros y **los grados al** número de columnas o campos.

Finalmente, la clave primaria es un identificador único **para cada instancia.**

Cómo funciona el modelo relacional

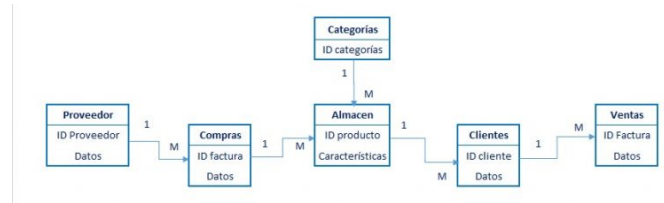
Aunque lo veremos en el **último ejemplo,** **mostraremos** cómo funciona. **Tenga** en cuenta que las propias tablas **han establecido** relaciones entre los datos, de ahí el nombre. En **realidad,** se rige por

algunas reglas simples como sigue:

Las tablas son el **corazón** del modelo y los datos deben **estar** representados en ellas. La **tabla** también se **utiliza** para **calcular** los **resultados** de **otros**.

El orden de cada columna **está** determinado por el tipo de consulta que se **está** **ejecutando**. Por **lo tanto**, no se **requiere** un **pedido** inicial, ya que cada relación se **compone** de un conjunto de datos **único**. Como **dijimos** antes, las filas son datos o casos, las columnas **son campos** o variables. Por otro lado, cada celda es un registro **con** dos dimensiones.

Es **esencial tener** un identificador único (clave primaria) **para** cada registro. Esto **le** permite establecer **una relación** entre dos o más tablas, **usándola** como clave externa.



Las tablas **estarían completas**, si **tomáramos** a los clientes **como ejemplo**, con los datos de cada **cliente** en una fila y el tipo (apellido, **nombre** ...) en una columna. La **letra 1** significa "**desde uno**" y la **letra M** significa "**hasta un número**". Esto se refiere al **hecho de que** en este modelo relacional las salidas de cada tabla son un solo **nivel de datos**, pero pueden **involucrar** más de uno de los anteriores.

Ventajas y desventajas del modelo relacional

Este modelo relacional, como todo **lo demás**, tiene **sus ventajas y desventajas**.

Entre las **ventajas**, podemos destacar las siguientes;

Por un lado, **cuenta con procedimientos para evitar** la **duplicación** de datos. Por otro **lado**, **asegura** la integridad referencial **al eliminar** todo lo relacionado con el registro cuando sea necesario. **También** permite la **estandarización**.

Por **otro lado**, entre las **limitaciones o limitaciones**, tenemos **las** siguientes:

La **primera** es que tiene limitaciones en la **representación** gráfica o en el sistema de **geolocalización**. En segundo **lugar**, los bloques de texto no se **manejan** de **manera** eficiente.

Ejemplo de plantilla de relación

Suponga que queremos crear **una plantilla** con proveedor, compra, **inventario** (con **catálogo**), cliente y ventas. Cada tabla representa un grupo de datos similar. **Tienen** un **montón** de datos en columnas (campos) y cada uno de ellos (tupla) **va** en una fila. **Lo** hemos **simplificado**, solo **con el propósito** de mostrar lo **que hemos visto** en este artículo. **podría** ser algo **como esto**

2.- ¿Considera importante el uso de UML en el sistema a diseñar? ¿Qué ventajas generará el uso de UML?

Hoy en día, UML ("**lenguaje de modelado unificado**") se **refuerza como un** lenguaje estándar **para analizar y diseñar** sistemas de **cálculo**.ⁱⁱⁱ

Entre los lenguajes orientados a objetos más utilizados **en la actualidad** se encuentran Java y C #, **así como** **lenguajes** más antiguos como C y SmallTalk, aunque **la programación** en todos estos lenguajes requiere experiencia previa **en sintaxis y bloques**. el **modelo utilizado** en todos es el mismo: Objetos.

Lo anterior permite **realizar** un análisis en UML sea **cual sea** el lenguaje en el que finalmente se **implemente el sistema** (Java, C #, C, SmallTalk), **la misma funcionalidad** permite **al personal que no conoce el lenguaje**. análisis y diseño de un sistema. Conceptos / Diagramas

Algunos conceptos básicos orientados a objetos incluyen:

Un modelo es un **resumen** del problema que **está tratando de resolver**.

Un dominio es el mundo **del que surge** el problema.

Un modelo **está formado por objetos** que interactúan entre sí **mediante el envío de mensajes**.

Cada objeto **tiene sus propias características** (**propiedades**) y **las operaciones** que puede realizar (métodos); **los valores** asignados a un objeto en un momento **dado determinan** su estado.

Una clase es un **modelo** para describir un **grupo de propiedades** (**propiedades**) y **objetos de comportamiento** (métodos).

Los objetos son instancias de **clases**.

Los 9 diagramas que forman la base de UML se enumeran a continuación y muestran cómo se diseñó el sistema:

UseCase

Clases

Objetos

Secuencia

Colaboración

Estado (Statechart)

Actividad

Componentes

Implementación

El uso de UML.

Modelado de procesos de negocios El modelado de procesos de negocios es una parte esencial de cualquier proceso de desarrollo de software. Permite al analista **comprender las líneas generales** y los procedimientos que **rigen las operaciones de**. Este modelo **proporciona** una descripción de dónde **encajaría** el sistema de software **en consideración** en la estructura **organizativa** y las **operaciones de rutina de**. **Creación de un sistema de software en mediante la captura de las operaciones** manuales **habituales y procesos** automatizados que se incorporarán **al** sistema, con **los costos** y beneficios asociados. Como modelo **de negocio preliminar**, permite al analista capturar los **hechos más importantes**, entradas, recursos y **productos relacionados** con el proceso de negocio^{iv}. **Se puede construir** un modelo completamente **rastreado al vincular luego** elementos de diseño (como casos de uso) al modelo **comercial** a través de **conexiones magnéticas**, **conexiones** de implementación **que generalizan los procesos comerciales** a los requisitos funcionales y **posiblemente** a los artefactos de software que **realmente se construirán**.

Debido a que el modelo de proceso de negocio es **a menudo más amplio** que la **sección del sistema de información considerado**, también permite al analista identificar claramente **qué cae** dentro del alcance del sistema propuesto y **qué se hará** de otras **formas** (por ejemplo, manual proceso).

UML es **una herramienta gráfica** que **se utiliza** para especificar **métodos o procedimientos a**

implementar por el sistema, **a través** de una serie de diagramas.

Nos proporciona una serie de herramientas que **nos permiten visualizar** el programa en sus **distintas fases** o procesos, **para delimitarlos** y organizarlos de **manera** que la persona que va a desarrollar el sistema **pueda entender**.

Cabe **señalar** que UML no es un lenguaje de **programación**, sino **un** sistema que permite modelar la estructura de **un programa**.

Las personas que nunca han programado **con uml** siempre lo **encuentran** una **pérdida** de tiempo, pero **deberían** dedicarlo **al menos** una semana, vale la **pena**.

Ventajas de la **programación en UML**^v:

Mejor tiempo de desarrollo **general (50% o más)**.

Modelado de sistemas (no solo software) utilizando conceptos orientados a objetos. Establecer conceptos y artefactos ejecutables.

Liderar el desarrollo de **escalabilidad** en sistemas **críticos complejos**.

Cree un lenguaje de modelado **utilizable** tanto por humanos como por máquinas.

Mejor soporte **para la planificación** y el control de proyectos.

Altamente reutilizable y minimiza los costos.

Actualice o cambie fácilmente el software programado.



Elementos de modelado avanzados^{vi}.

Los elementos de modelado **discutidos** hasta ahora **le permiten** representar cualquier esquema de base de datos de aplicaciones tradicional. Pero para representar algunas características del mundo real **de manera más precisa y directa**, se utilizan otros elementos **del** modelo **para permitir** más potencia y flexibilidad. En esta **sección**, **veremos** algunos elementos que **nos permiten** representar fácilmente realidades **que son difíciles de expresar** utilizando los elementos **del**

modelo vistos en el capítulo anterior. 1.

Generalización / **Especialización** En algunos casos, puede **resultar** interesante representar la información de un dominio de forma **ascendente**. Es decir, **definir** los tipos de **entidades** según **diferentes** niveles de abstracción y **relacionar** tipos de **entidades** más **específicos** con los de carácter más general. **Por lo tanto, el conocimiento común definido en las clases comunes** se puede reutilizar en nuevas clases.

Por ejemplo, supongamos **que el tipo de entidad "Persona"** en un entorno **académico**. Sus atributos son nombre, apellidos, **documento de identidad**, fecha de nacimiento, sexo, teléfono y correo electrónico. En la base de datos queremos tratar con diferentes tipos de personas: profesores, **administradores** y **alumnos**. Los estudiantes tienen algunas características **únicas** que **están relacionadas** con la base de datos y que los **maestros** y **administradores no comparten**, como las **identificaciones de los estudiantes** y el **primer año de admisión**. **Por parte de los profesores, nos gusta archivar** sus áreas de **especialización**. Y del personal **administrativo**, queremos **tener en cuenta su puesto** y **el departamento en el que trabaja**. Es decir, todas las entidades tienen un conjunto **común de** características, pero también tienen características diferentes que **deberían** poder representar en el modelo de base de datos. Además, podemos ver que cada uno de los tres colectivos tiene una semántica **bien** definida y **es diferente** a los demás. Cuando nos encontramos con casos como **en el ejemplo anterior**, **podemos** crear una jerarquía de tipos de entidad que **nos permite** definir una clase en diferentes niveles de abstracción. Por ejemplo, en el **primer caso**, **podemos** crear una jerarquía de tipo de entidad con Persona como la **clase más general** y Estudiante, Profesor y Administración como clases **concretas**.

Generalización / especialización¹ es la relación entre un tipo **común de** entidad, que **llamamos** superclase, (o tipo de entidad padre), y un tipo de entidad más específico, denominado subclase (o tipo de entidad hijo).

En notación UML, esta relación **está** representada **por una flecha hueca** que **va** de las subclases a la superclase. La superclase define la información más general y **tiende** a incluir características comunes a todas sus subclases, como **nombre de** atributo, apellido, DNI, fecha de nacimiento, **género**, teléfono y **el correo electrónico** del ejemplo anterior, es **común a** todas las subclases y se define en la clase **padre**. Las subclases, por otro lado, definen información más **específica** y **tienden** a definir características que no son comunes a **otras** subclases,

como **una propiedad de campo especial** de la clase **Teacher**. Las subclases se denominan especiales de la superclase y la superclase se denomina generalización de las subclases. **Las propiedades** o relaciones de la superclase se **pasan** a las clases **secundarias**. **A esto se le llama** herencia de propiedad. Por **lo tanto**, cualquier entidad que pertenezca a cualquier subclase puede tener **valores** para **las propiedades definidas** en la superclase. **Las propiedades** de la subclase **se denominan propiedades concretas** o locales. Solo las entidades que pertenecen a una determinada **subclase** **pueden** tener un valor asignado a estos atributos.

Diseño conceptual de bases de datos

En notación UML^{vii}, esta relación **está** representada **por una flecha hueca** que **va** de las subclases a la superclase. La superclase define la información más general y **tiende** a incluir características comunes a todas sus subclases, como **nombre de** atributo, apellido, DNI, fecha de nacimiento, **género**, teléfono y **el correo electrónico** del ejemplo anterior, es **común a** todas las subclases y se define en la clase **padre**. Las subclases, por otro lado, definen información más **específica** y **tienden** a definir características que no son comunes a **otras** subclases, como **una propiedad de campo especial** de la clase **Teacher**. Las subclases se denominan especiales de la superclase y la superclase se denomina generalización de las subclases. **Las propiedades** o relaciones de la superclase se **pasan** a las clases **secundarias**. **A esto se le llama** herencia de propiedad. Por **lo tanto**, cualquier entidad que pertenezca a cualquier subclase puede tener **valores** para **las propiedades definidas** en la superclase. **Las propiedades** de la subclase **se denominan propiedades concretas** o locales. Solo las entidades que pertenecen a una determinada **subclase** **pueden** tener un valor asignado a estos atributos.

Lenguajes de modelado conceptual

Como se mencionó, existen varios lenguajes que permiten **la representación de diagramas de conceptos**. Algunas **de las características** importantes que deben **reunir** estos lenguajes son: • **Expresivos**: deben ser **lo suficientemente** expresivos para permitir **la representación de** diferentes conceptos del mundo real y las relaciones entre estos conceptos. • **Sencillez**: los **diagramas** resultantes deben ser simples y fáciles de entender. El usuario de la base de datos **debe** poder **comprender** los conceptos **presentados**. • **Representación del esquema**: el modelo debe utilizar una notación **esquemática que sea fácil** de entender y útil para visualizar el **mapa** conceptual. • **Forma**: la representación del modelo debe ser precisa, formal y no **debe** presentar ambigüedad. **No** es fácil **encontrar**

¹ La relación de generalización/especialización también se denomina relación 'es-un' o 'es-una' (o 'is a' en inglés) por la manera en que se hace referencia al concepto: un estudiante 'es una' persona, un profesor 'es una' persona.

Casas Roma, J. y i Caralt, J. C. (2014). *Diseño conceptual de bases de datos en UML*. Barcelona, Spain: Editorial UOC. Recuperado de <https://elibro.net/es/ereader/iacc/57635?page=109>.

estos rasgos y algunos de ellos a menudo entran en conflicto con otros. Quizás el modelo más conocido y más utilizado hasta la fecha es el modelo de correlación de entidades, también conocido como modelo ER. Sin embargo, ahora el uso del lenguaje de modelado unificado (UML) ha ganado un gran poder en el diseño de esquemas de bases de datos conceptuales. El lector notará que usamos el término modelo y lenguaje indistintamente. Es cierto que en general el término y el modelo lingüístico representan cosas distintas: el modelo representa las abstracciones de la realidad los lenguajes son artefactos que se utilizan para representar patrones. Sin embargo, en el campo de las bases de datos, los modelos (o modelos de datos) se definen como un conjunto de conceptos, relaciones y reglas que se utilizan para definir esquemas conceptuales. Por lo tanto, en el dominio de la base de datos, podemos usar los términos modelo y lenguaje de manera intercambiable. A continuación, estudiaremos el modelado ER y UML, luego se enumerarán las metodologías y estrategias a seguir durante la fase de modelado conceptual.

El lenguaje UML El lenguaje unificado de modelado¹⁵ (UML) es un lenguaje de propósito general diseñado para modelar sistemas de software. El estándar fue creado y es mantenido por el Object² Grupo de administración³. Se agregó por primera vez a la lista de tecnologías utilizadas por OMG en 1997 y desde entonces se ha convertido en el estándar de la industria para modelar sistemas de software. UML es un lenguaje gráfico diseñado para especificar, visualizar, modificar, construir y documentar un sistema. Permite la visualización estándar de varios artefactos, incluidas operaciones, agentes, lógica empresarial y esquema de base de datos.

Para el diseño de bases de datos conceptuales, estamos particularmente interesados en los diagramas de clases, que nos permiten representar información del dominio del habla. Sin embargo, cada diagrama UML se describe brevemente a continuación, aunque su alcance está más allá del alcance de este texto. Esquemas estructurales: permiten representar la información relevante de un sistema de información y organizar sus componentes.

a) Un diagrama de clases es un diagrama estático que describe la estructura de un sistema en términos de clases de sistema, las propiedades de estas clases y las relaciones establecidas entre ellas (llamados enlaces en terminología UML). Estos diagramas son uno de los pilares del desarrollo orientado a objetos,

pero también han demostrado excelentes habilidades de modelado de datos. Por este motivo, adquieren cada vez más importancia en el diseño conceptual de bases de datos⁴.

b) Diagrama de objetos que muestra instancias (u objetos del mundo real) y las relaciones entre ellos de acuerdo con un diagrama de clases. Las instancias de un sistema cambian con el tiempo, por lo que los gráficos de objetos se pueden considerar como una instantánea que muestra diferentes instancias de un sistema y cómo se relacionan en un momento dado. Por ejemplo, en algún momento la clase "Usuario" puede tener dos instancias, una representando al usuario "José María" con la contraseña "caf " y la otra representando al usuario "Ana" con la contraseña "c0n1 ch ". c) Diagrama de componentes que ilustra la organización y las dependencias entre los componentes del sistema. En un entorno de base de datos, se utilizan para modelar espacios de tabla o particiones. Estas estructuras no se tratan en este libro porque forman parte del diseño físico de la base de datos. d) Diagramas de implementación que muestran la distribución de los componentes del sistema y su relación con los componentes de hardware disponibles. Diagramas de comportamiento: ayudan a representar las funciones que realiza un sistema de información y cómo las realizan los diferentes niveles de abstracción. Es decir, definen la funcionalidad del sistema de información.

a) Los diagramas de casos de uso se utilizan para modelar interacciones funcionales entre usuarios y sistemas.

b) Diagrama de secuencia que representa las interacciones entre diferentes objetos a lo largo del tiempo. Muestran el flujo de mensajes a lo largo del tiempo entre diferentes objetos.

c) Los diagramas de colaboración representan interacciones entre objetos como una secuencia de mensajes secuenciales. Estos diagramas llaman la atención sobre la organización estructural de los objetos que envían y reciben mensajes.

d) El diagrama de estado describe cómo cambia el estado de un objeto en respuesta a varios eventos externos.

e) Los diagramas de actividad presentan una vista dinámica del sistema y modelan el flujo de control de una actividad a otra.

Diagramas y vistas.

UML define varios modelos para la representación de los sistemas que pueden verse y manipularse mediante un conjunto de diagramas diferentes:

- Diagramas de estructura.

promueve el uso de la tecnología orientada a objetos median- te guías y especificaciones para estas guías.

⁴ El dominio del discurso, universo del discurso, o simplemente domi- nio, es la información relevante en un determinado contexto.

² En inglés, unified modeling language (UML).

³ El Object Management Group (OMG) es un consorcio dedicado a esta- blecer y promover varias especificaciones de tecnologías orientadas a objetos, como UML, XMI o CORBA. Es una organización sin ánimo de lucro que

- Diagrama de clases.
- Diagrama de estructuras compuestas.
- Diagrama de componentes.
- Diagrama de despliegue.
- Diagrama de objetos.
- Diagrama de paquetes.
- Diagramas de comportamiento.
- Diagrama de casos de uso.
- Diagrama de actividad.
- Diagramas de interacción.
- Diagrama de secuencia.
- Diagrama de comunicación o colaboración.
- Diagrama de visión global de la interacción.
- Diagrama de tiempo.
- Diagrama de maquina de estados.

Una vista es un subconjunto de las construcciones de modelado de UML que representa un aspecto del sistema.

Los diagramas UML se pueden organizar en las siguientes vistas [Rumbaugh et al., 2007].

- Vista estática.
- Diagramas de clases.
- Vista de casos de uso.
- Diagramas de casos de uso.
- Vista de interacción.
- Diagramas de secuencia.
- Diagramas de comunicación.
- Vista de actividad.
- Diagramas de actividad.
- Vista de la máquina de estados.
- Diagramas de máquina de estados.
- Vista de diseño.
- Diagramas de estructuras.
- compuestas.
- Diagramas de colaboración.
- Diagramas de componentes.
- Vista de despliegue.
- Diagramas de despliegue.
- Vista de gestión del Modelo.
- Diagramas de paquetes.
- Perfiles.
- Diagramas de paquetes.

Las vistas se pueden agrupar en áreas conceptuales.

Área	Vista
Estructural	Vista estática
	Vista de diseño
	Vista de casos de uso
Dinámica	Vista de máquina de estados
	Vista de actividad
	Vista de interacción
Física	Vista de despliegue
Gestión	Vista de gestión del modelo
	Perfiles

Vista de casos de Uso.



Introducción.

- La vista de casos de uso captura la funcionalidad de un sistema, de un subsistema, o de una clase, tal como se muestra a un usuario exterior.
- Reparte la funcionalidad del sistema en

transacciones significativas para los usuarios ideales de un sistema.

- Los usuarios del sistema se denominan actores y las particiones funcionales se conocen con el nombre de casos de uso.
- La técnica que se utiliza para modelar esta vista es el diagrama de casos de uso.

Características.

Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente por Ivar Jacobson [Jacobson, 1987], [Jacobson et al. 1992] e incorporada a UML.

Modela la funcionalidad del sistema tal como la perciben los agentes externos, denominados actores, que interactúan con el sistema desde un punto de vista particular.

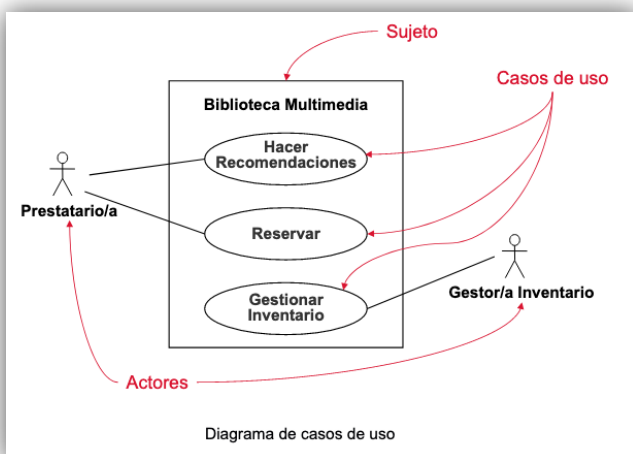
Sus componentes principales son:

- Sujeto: sistema que se modela.
- Casos de uso: unidades funcionales completas.
- Actores: entidades externas que interactúan con el sistema.

El sujeto se muestra como una caja negra que proporciona los casos de uso.

El modelo de casos de uso se representa mediante los diagramas de casos de uso.

Características.



Actores.

Un actor es un clasificador que modela un tipo de rol que juega una entidad que interacciona con el sujeto pero que es externa a él.

- Un actor puede tener múltiples instancias físicas.
- Una instancia física de un actor puede jugar diferentes papeles. Los actores se comunican con el sujeto intercambiando mensajes (señales, llamadas o datos).

Notación.

- Se representan con el icono estándar de “stickman” o “monigote” con el nombre del actor (obligatorio) cerca del símbolo, normalmente se pone encima o debajo.
- También se puede representar mediante un símbolo de clasificador con el estereotipo «actor».
- Los nombres de los actores suelen empezar por mayúscula.
- Se pueden usar otros símbolos para representar tipos de actores, por ejemplo para representar actores no humanos.

Tipos de actores [Larman, 2002].

Principales.

- Tiene objetivos de usuario que se satisfacen mediante el uso de los servicios del sistema.
- Se identifican para encontrar los objetivos de usuario, los cuales dirigen los casos de uso.

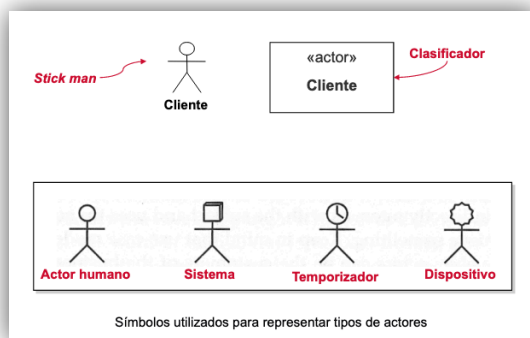
De apoyo.

- Proporcionan un servicio al sistema.
- Normalmente se trata de un sistema informático, pero podría ser una organización o una persona.
- Se identifican para clarificar las interfaces externas y los protocolos.

Pasivos.

- Está interesado en el comportamiento del caso de uso, pero no es principal ni de apoyo.
- Se identifican para asegurar que todos los intereses necesarios se han identificado y satisfecho.
- Los intereses de los actores pasivos algunas veces son sutiles o es fácil no tenerlos en cuenta, a menos que estos actores sean identificados explícitamente.

Actores.



Casos de uso.

Un caso de uso se define como un conjunto de acciones realizadas por el sistema que dan lugar a un resultado observable.

El caso de uso especifica un comportamiento que el sujeto puede realizar en colaboración con uno o más actores, pero sin hacer referencia a su estructura interna.

El caso de uso puede contener posibles variaciones de su comportamiento básico incluyendo manejo de errores y excepciones.

Una instanciación de un caso de uso es un escenario que representa un uso particular del sistema (un camino).

Características de los casos de uso:

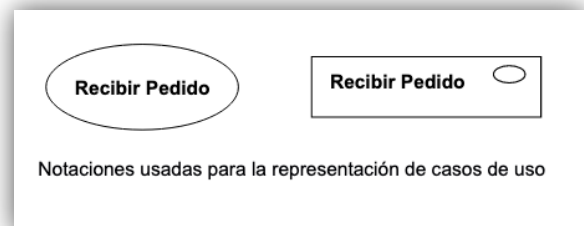
- Un caso de uso se inicia por un actor.
- Los casos de uso proporcionan valores a los actores.

- La funcionalidad de un caso de uso debe ser completa.

El comportamiento de un caso de uso se puede describir mediante interacciones, actividades, máquinas de estado.

Notación:

- Elipse con el nombre del caso de uso dentro o debajo de ella. Se puede colocar algún estereotipo encima del nombre y una lista de propiedades debajo.
- La representación alternativa es la del símbolo del clasificador con una elipse pequeña en la esquina superior derecha.



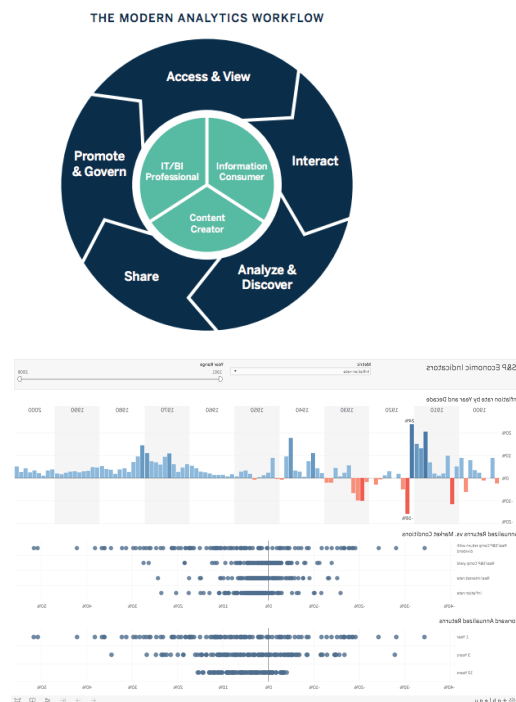
¿Qué es la inteligencia de negocios? Guía sobre la inteligencia de negocios y por qué es importante.^{viii}

La inteligencia de negocios (BI) combina análisis de negocios, minería, visualización, herramientas e infraestructura de datos, además de prácticas recomendadas para ayudar a las empresas a tomar decisiones basadas en los datos. En la práctica, implementar la inteligencia de negocios moderna implica contar con una vista integral de todos los datos de la organización. Además, consiste en usar estos datos para impulsar el cambio, eliminar las ineficiencias y adaptarse rápidamente a los cambios del mercado o la demanda.

Es importante tener en cuenta que esta es una definición muy moderna de la BI. Además, la BI tiene una historia muy breve como palabra de moda. Originalmente, la inteligencia de negocios tradicional surgió en la década de 1960 como un sistema para compartir información entre organizaciones. En la

década de 1980 se desarrolló aún más junto con los modelos informáticos. Se utilizó para tomar decisiones y transformar datos en información antes de convertirse en un producto específico de los equipos de BI con soluciones de servicio basadas en TI. En las soluciones de BI modernas, se priorizan factores como el análisis de autoservicio flexible, los datos gobernados en plataformas confiables, la capacitación de los usuarios corporativos y la rapidez para obtener información.

Im1 – im 2



Desarrollo.

Una universidad trabajaba con el software Microsoft Dynamics como CRM (Sistema de manejo de relaciones con el cliente). Se requiere realizar varios procesos, desde la integración del mencionado CRM con otro tipo de herramientas que suplan las necesidades de la universidad, hasta dar forma a esa visión única que representaría a cada uno de los estudiantes. Y por supuesto, para solventar el problema del pulido de las iniciativas de marketing, habría que definir una serie de métricas de resultado y KPI que permitieran a la entidad sacar el máximo provecho de todas sus iniciativas. Hacer esto tendría como efecto una mejora notable en la universidad, la cual ya había sufrido problemas debido a la mala gestión de la información. Por un lado, se había sufrido la duplicación de datos, la presencia de detalles erróneos sobre los alumnos e información que, en general, era de mala calidad. Todos estos datos no solo presentaban estos errores, sino que además se encontraban desperdigados en distintas aplicaciones y resultaba complicado tener una visión de todo ello en paralelo. En otro orden de cosas, también se había intentado realizar algunas iniciativas de integración de

aplicaciones sin éxito.

1.- ¿Cómo aplicaría una data warehouse para resolver la problemática presentada en la universidad?

Primero que nada:

Data warehouse: pasos para construirlo con éxito:

El data warehouse es un sistema donde se almacena una gran cantidad de datos. Tiene como propósito analizar los datos y obtener resultados para facilitar la toma de decisiones, y antes de resolver esta problemática debemos tomar las siguientes situaciones o características:

Im4



El *data warehouse* o almacén de datos, como también se le conoce, es **un sistema de gestión de datos** diseñado para habilitar y soportar actividades empresariales. Su capacidad analítica permite a las organizaciones obtener valiosos datos de su negocio en tiempo real y mejorar la toma de decisiones.

Para construir un *data warehouse* exitoso se requiere seguir los siguientes pasos:

1. Comprender el problema

El equipo de desarrollo de *data warehouse* debe realizar un diagnóstico de la cantidad de datos que maneja la empresa y las transformaciones digitales que se deben implementar, en caso sea necesario, antes de crear un sistema de gestión de datos. Conociendo a profundidad estos dos puntos, el *data warehouse* será un aliado potente para los altos mandos.

2. Identificar la fuente de los datos

Un *data warehouse* extrae datos de diversas fuentes. El equipo de desarrollo debe identificar todos los *data marts* (versión especial del almacén de datos centrado en un tema o área específica de negocio) que se

utilizarán para el procesamiento de datos. Las fuentes de datos pueden ser de cualquier tipo: otras bases de datos (SQL/NoSQL), aplicaciones, redes sociales, encuestas, archivos de Excel, etc.

3. Crear el modelo de data warehouse

Una vez establecidos los *data marts* y definidos los datos, se crea una base de datos central para todos los elementos que estarán incluidos en el *data warehouse*. Esta base de datos central debe ilustrar todos los objetos -y sus propiedades- que irán en el almacén de datos.

4. Seleccionar las herramientas ETL adecuadas

Las herramientas ETL (Extracción, Transformación y Carga) se encargan de extraer los datos de las fuentes, transformarlos y cargarlos al *data warehouse*. En otras palabras, son pequeños programas que permiten transformar distintos formatos de fuentes a una que pueda procesar la base de datos central. Es muy importante seleccionar una herramienta ETL que tenga un manejo sencillo y represente, de manera visual, todas las transformaciones. Entre las herramientas más conocidas podemos mencionar: [Microsoft - SQL Server Integrated Services \(SSIS\)](#), [IBM InfoSphere Information Server](#), [Amazon Redshift](#) y [Oracle GoldenGate](#).

5. Emplear analítica inteligente

En esta etapa, se necesitan crear gráficos, cuadrículas y/o tablas para visualizar los datos y tomar las mejores decisiones. En esta etapa existen programas que permitirán generar reportes como [Oracle BI](#), [Microsoft PowerBI](#), entre otros.

6. Documentar

Para que un *data warehouse* sea exitoso, necesita ser parte funcional de las operaciones de una empresa y evolucionar a medida que el negocio también crezca. Por lo tanto, una buena documentación de cómo se ha venido desarrollando el almacén de datos y los logros que se han alcanzado gracias a él garantizarán su continuidad.

La implementación de un *data warehouse* proporciona a las empresas una visión más clara y detallada del negocio, lo que permita a los altos mandos tomar las mejores decisiones en el menor tiempo posible.^{ix}

Gerencia del conocimiento:

¿En dónde se aplican el warehouse y CRM?

Aunque el data warehouse se emplea para la

inteligencia de negocios o BI, también puede utilizarse para la gestión de relaciones con los clientes o CRM.

¿Y cuáles son sus aplicaciones?

¿Y el data warehouse?

Por su parte, los almacenes de datos, debido a su potencial, tienen aplicaciones profundamente arraigadas en todas las industrias que utilizan datos históricos para la predicción, el análisis estadístico y la toma de decisiones. Los almacenes de datos se concentran en la gestión de riesgos y la reversión de políticas. Además, analizan los datos de los consumidores, las tendencias del mercado, las regulaciones e informes gubernamentales y, lo que es más importante, la toma de decisiones financieras. Los almacenes se utilizan principalmente para analizar patrones de datos y tendencias de clientes, además de mantener registros de participantes ya existentes.

Además de las industrias antes mencionadas, el almacén de datos también se utiliza en la manufactura y distribución, el sector retail, la industria de seguros, de gobierno y educación, marketing, transporte, entre muchas otras. Por último, una integración de CRM con el almacenamiento de datos puede proporcionar numerosos beneficios a las organizaciones.⁵

1. Industria bancaria/financiera. Los almacenes de datos se concentran en la gestión de riesgos y la reversión de políticas. Además, analizan los datos de los consumidores, las tendencias del mercado, las regulaciones e informes gubernamentales y, lo que es más importante, la toma de decisiones financieras.

2. Industria de bienes de consumo. Se utilizan para la predicción de tendencias de consumo, gestión de inventarios, investigación de mercado y publicidad. También se lleva a cabo un análisis en profundidad de las ventas y la producción. Además de ello, se intercambia información de socios comerciales y clientes.

3. Atención de salud. Este es uno de los sectores más importantes que utiliza data warehouse. Todos sus registros financieros, clínicos y de empleados se envían a los almacenes. Estos ayudan a crear estrategias y predecir los resultados, rastrear y analizar los comentarios de sus servicios, generar informes de pacientes, compartir datos con compañías de seguros vinculadas, servicios de asistencia médica, etc.

4. Industria de seguros. Los almacenes se utilizan principalmente para analizar patrones de datos y tendencias de clientes, además de mantener registros de participantes ya existentes. El diseño de ofertas y

⁵ Resumen realizado por alumno.

promociones hechas a medida del cliente también es posible a través del data warehouse.

Todo en un lugar.

Además de las industrias antes mencionadas, el almacén de datos también se utiliza en la manufactura y distribución, el sector retail, la industria de seguros, de gobierno y educación, marketing, transporte, entre muchas otras.

Al integrar datos de múltiples sistemas de origen, permite una visión central de toda su empresa.

Im7



Por último, una integración de CRM con el almacenamiento de datos puede proporcionar numerosos beneficios a las organizaciones. Especialmente en la reducción en el costo para adquirir clientes, en el costo de venta, el costo de servicio y el tiempo de entrega. Desde la gestión de permisos, la recopilación de *insights* del consumidor y la atención al usuario, los usos del data warehouse en la gestión de relaciones con los clientes son sumamente importantes.^x

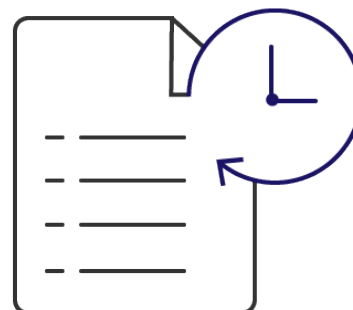
Veneficios de implementación de un DataWareHhouse en una Universidad un en una Empresa en general.
Beneficios de Data Warehouse.

Conoce la historia

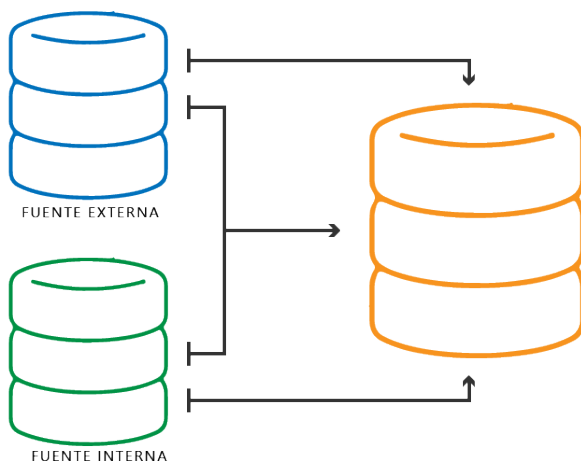
Mantiene el historial de su información incluso si los sistemas transaccionales de origen no lo hacen.

Ayuda a que se puedan tomar mejores decisiones al integrar y relacionar fuentes de datos internos y/o externos.

Im8



Im5

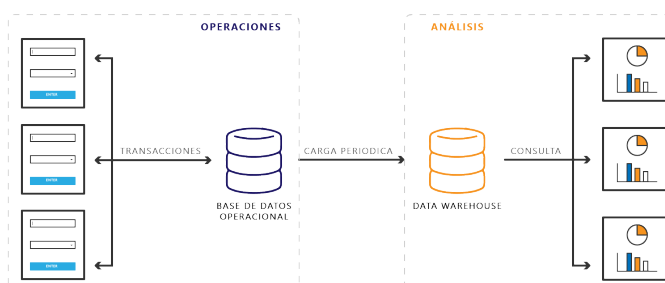


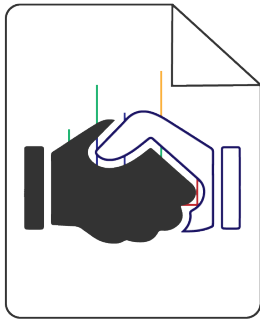
Información Oportuna y Confiable

Un **Data Warehouse** se emplea para hacer el trabajo analítico, dejando las bases de datos transaccionales libres para centrarse en la operación.

El **Data Warehouse** se actualiza periódicamente aplicando reglas de negocio y calidad de datos, ofreciendo información confiable al momento.

Im9

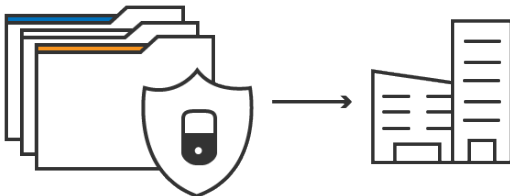




Una sola versión de la verdad

Toda su empresa obtendrá información confiable desde un repositorio único, evitando la manipulación con un criterio individual.

Im10



La velocidad es la clave

Estructura los datos reduciendo el volumen y maximizando el rendimiento para consultas analíticas complejas, sin afectar a los sistemas transaccionales.

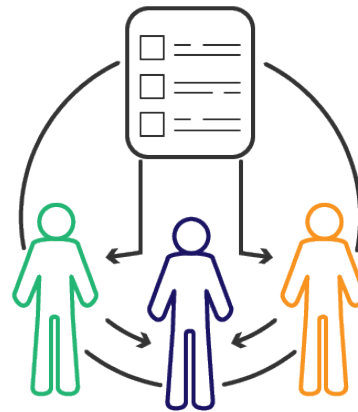
Im11



Gobierno de Datos

Un **Data Warehouse** al estar orientado a modelos de negocio, permite gobernar el acceso a la información basado en roles y/o funciones de la empresa.

Im12



2.- ¿Sugiere un modelo estrella o copo de nieve? *Argumente.*

Yo argumentaría un modelo de copo de nieve, puede que sea mas enrevesada su implementación; pero tal como, es ideal cuando hay muchas tablas de grandes dimensiones, y sea necesario determinar aspectos como *redundancia, normalización, integridad de datos y otros significativos elementos a considerar*; tal como se exponen en sitios web como estos⁶:

En la etapa de construcción es mejor modelar las tablas relacionales en una gran estructura desnormalizada compuesta por **tabla de hechos**, y tablas más pequeñas que definirán las n-dimensiones llamadas **tablas de dimensiones**.

Modelo Copo de Nieve

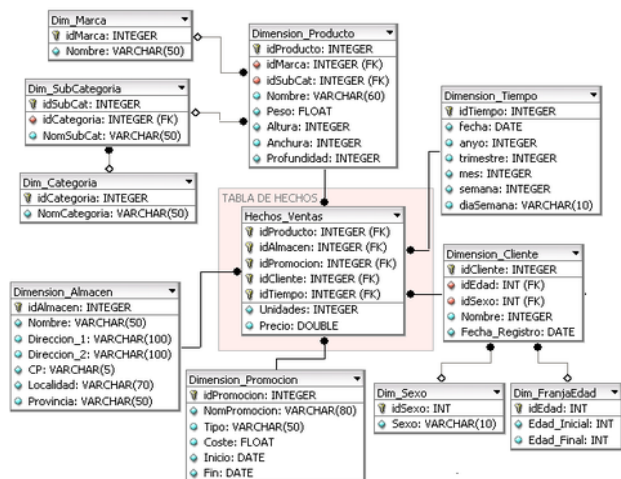
El modelo copo de nieve es una variación o derivación del modelo estrella. En este modelo la tabla de hechos deja de ser la única relacionada con otras tablas ya que existen otras tablas que se relacionan con las dimensiones y que no tienen relación directa con la tabla de hechos. El modelo fue concebido para facilitar el mantenimiento de las dimensiones, sin embargo esto hace que se vinculen más tablas a las secuencias SQL, haciendo la extracción de datos más difícil así como vuelve compleja la tarea de mantener el modelo.

Ejemplo de modelo Copo de Nieve. *(Tomada de Wikipedia)*

Img3

Construcción)

⁶ <https://revistadigital.inesem.es/informatica-y-tics/guia-construir-datawarehouse/> (Fases de



Si se desea más orden en ese aspecto se puede utilizar el modelo copo de nieve sin embargo al existir más relaciones en el modelo este se volvería poco eficiente para buscar la información además de volverse complejo de mantener. Por eso es muy recomendable definir bien que se espera del Datawarehouse para utilizar uno de los dos modelos, factores como tamaño, uso y velocidad de proceso pueden hacer tomar un modelo u otro.^{xi}

3.- ¿Qué ventajas tendrá la universidad al implementar el data warehouse?

Creo que primero que nada es importante definir que es un DataWarehouse, antes de resolver la problemática en cuestión.

¿Que es?^{xii}

Un Data warehouse permite a una organización o empresa ejecutar análisis potentes en grandes volúmenes de datos históricos de formas que una base de datos estándar simplemente no puede.

Hoy en día, un Data warehouse puede estar alojado en un dispositivo dedicado o en la nube, y la mayoría de los data warehouses han agregado capacidades de análisis y herramientas de visualización y presentación de datos. Todo esto entrega lo necesario para entregar los recursos para la toma de decisiones en una corporación.⁷

Pasado y presente del Data Warehouse.

Históricamente, los data warehouses se habían formado utilizando datos repetitivos estructurados que eran filtrados antes de entrar en el data warehouse. Sin embargo, **en los últimos años, el data warehouse ha evolucionado debido a**

información contextual que ahora se puede adjuntar a los datos no estructurados y que también puede ser almacenada.

Aquellos primeros datos relacionales estructurados no podían ser mezclados y emparejados para temas analíticos con datos textuales no estructurados. Pero con el advenimiento de la contextualización, estos tipos de análisis ahora sí pueden hacerse de forma naturales y fácil.

En el data warehouse, *datos no repetitivos, como los comentarios en una encuesta, correos electrónicos y conversaciones, se tratan de forma diferente a las ocurrencias repetitivas de datos, como el flujo de clics, mediciones o el procesamiento máquina o analógico.* Los datos no repetitivos son datos basados en textos que fueron generados por la palabra escrita o hablada, leída y reformateada y, lo que es más importante, ahora puede ser contextualizada. Con el fin de extraer cualquier sentido de los datos no repetitivos para su uso en el Data Warehouse, deben tener el contexto de los datos establecidos.

En muchos casos, el contexto de los datos no repetitivos es más importante que los datos en sí. En cualquier caso, los datos no repetitivos no pueden utilizarse para la toma de decisiones hasta que se haya establecido el contexto.

Data Lakes y Data Warehouses: ¿mutuamente exclusivos o partners perfectos?.

Los data lakes han surgido en el paisaje de Data Management en los últimos años, sin embargo, **data lake no es necesariamente un reemplazo del data warehouse.** En cambio, **complementan los esfuerzos existentes y dan soporte al descubrimiento de nuevas preguntas.** Una vez que se descubren esas preguntas se optimizan las respuestas. Y optimizar puede significar moverse fuera del data lake para ir a un data mart o al data warehouse.

Características:

Características de un Datawarehouse.

La ventaja principal de estos sistemas se basa en la estructura de la información, que supone el almacenamiento de los datos de forma homogénea y fiable, en una estructura basada en la consulta y el

⁷ Resumen realizado por alumno.

tratamiento jerarquizado de la misma, y en un entorno diferenciado de los sistemas operacionales. Según Bill Inmon, quien acuñó el término, el Datawarehouse se caracteriza por ser:

- **Integrado:** Los datos almacenados tienen que ser integrados en una estructura consistente, lo que conlleva a la eliminación de las inconsistencias existentes entre sistemas de operaciones.
- **Temático:** Para facilitar el acceso a los datos y el entendimiento de ellos se suelen organizar por temas.
- **Histórico:** Permite almacenar datos en momentos concretos, para realizar análisis de tendencias en esos momentos o periodos de tiempo y poder realizar comparaciones entre esos valores tomados.
- **No volátil:** Esto implica que la información almacenada en el Datawarehouse es para ser leída pero no modificada. Por lo que la información es permanente.

Otra característica importante del datawarehouse es que contiene metadatos, es decir, datos sobre los datos. Los metadatos permiten saber la procedencia de la información, su periodicidad de refresco, su fiabilidad, forma de cálculo, etc.

Los metadatos serán los que permitan simplificar y automatizar la obtención de la información desde los sistemas operacionales a los sistemas informacionales.

Introducidos los conceptos anteriores, vamos a exponer a continuación un ejemplo que iremos elaborando a partir de las diferentes fases de construcción de un datawarehouse.

*“La compañía de alquiler de vehículos **Rent4you** desea diseñar un datawarehouse como sistema de soporte de ayuda a la toma de decisiones estratégicas. La compañía tiene varias sucursales, en las cuales ofertan alquileres de una amplia variedad de vehículos como, coches, motos, caravanas, furgonetas, etc.*

El datawarehouse pretende centralizar la información de todas las sucursales con el objetivo de definir nuevas estrategias de negocio. Se quiere analizar tanto las ventas (alquileres realizados en cada sucursal) así como los gastos para poder estudiar cuáles son los puntos de venta que más beneficios aportan, los perfiles de los clientes por tipo de vehículo para ajustar los seguros, la eficiencia de los empleados por sucursal, etc.

El datawarehouse se va a alimentar de grandes bases

de datos operacionales como son la de ventas, gastos, clientes y administración de RRHH.”

¿Cómo crear un data warehouse?^{xiv}

Estudio preliminar.

Como todo proyecto, el diseño e implementación de un almacén de datos debe ser viable y rentable para ser llevado a cabo. Como punto inicial de este caso, el lector deberá evaluar si el proyecto debe ser implementado o no en función de distintos parámetros. Para ello, el lector debe hacer un análisis del periodo de retorno de inversión, de la tasa interna de rentabilidad, del descuento de flujos financieros y del retorno de la inversión. Para facilitar dicho análisis, se van a asumir ciertas hipótesis sobre el proyecto.

Supongamos que un hospital básico tiene la intención de desplegar, progresivamente, el proyecto de almacén de datos incorporando más usuarios al sistema conforme pasan los años, de forma que el primer año tan solo habrá 25 usuarios y se incorporan de 10 en 10 los siguientes años, alcanzando los 65 usuarios en el quinto año. Supongamos que la inversión inicial del proyecto es 76.000 euros y que los costes de los siguientes años incluyen mantenimiento, evolución y nuevas licencias del sistema, siendo 16.700 euros anuales para los siguientes tres años. Consideremos también que la compañía considera que es capaz de generar beneficios derivados de la implantación del almacén de datos (como, por ejemplo, resultantes de liberar recursos por la automatización de tareas). Los beneficios estimados son 20.000 euros en el primer año, 25.000 en el segundo y 35.000 en cada uno de los tres años restantes. Finalmente, consideremos que la tasa de descuento es del 10 % a lo largo del periodo de cinco años considerado. Bajo el contexto descrito, la dirección general del hospital se pregunta si vale la pena invertir en el proyecto de diseño de un almacén de datos para la gestión de urgencias y hospitalización, y en qué medida el proyecto podría ser rentable.

Diseño de data warehouse.

A partir del análisis del contexto del caso y de las fuentes de datos disponibles de hospitalización, se deberá diseñar un almacén de datos que permita realizar el seguimiento estratégico y operativo del área de hospitalización de un hospital general básico tal y como se ha descrito en el contexto. Con el objetivo de poder recorrer todas las fases de este tipo de proyectos, la actividad se centrará tan solo en la hospitalización. Mediante la metodología de diseño de un data warehouse, se deberá plantear:

1) El análisis de requerimientos: describir las

necesidades de la organización

E identificar las preguntas que debe responder el sistema para el área de hospitalización.

2) El análisis de fuentes de datos: dónde se deben revisar las fuentes de datos proporcionadas, qué tipo de información contienen, cuál es su formato, y qué cantidad representa para la carga inicial.

3) El análisis funcional: proponer el tipo de arquitectura para la factoría de información adecuada para el proyecto (por ejemplo, si es necesario un data mart operacional o una estructura de carga intermedia).

4) Diseño para el modelo conceptual, lógico y físico del almacén de datos: dónde se deben identificar, diseñar e implementar las tablas de hecho, las dimensiones y atributos que pueden representar la información de hospitalización para un hospital general, así como aquellas estructuras consideradas en el punto 3.

Para el desarrollo del data warehouse, es preciso definir correctamente los hechos (facts), las dimensiones de análisis (dimensions) y los atributos que nos permitan tener el nivel de granularidad suficiente para la medida y presentación de los objetivos que se definan en el análisis de requerimientos. Carga de datos Se pide el desarrollo de los procesos de carga del data warehouse propuesto en la segunda parte de este caso. A partir del análisis de fuentes de datos, se deberán diseñar los procesos de extracción, transformación y carga de los datos existentes en los ficheros proporcionados, que contienen información operativa. Se deberá tener en cuenta que esta es una carga inicial de almacén de datos, por lo que se espera que, teniendo en cuenta el período que comprenden los datos y su cantidad, se hagan estimaciones sobre las necesidades de arquitectura del almacén de datos (por ejemplo, tiempo de carga, estimaciones de crecimiento, etc.). Por lo tanto, las tareas a realizar son: Explotación de datos. Por último, se debe diseñar un modelo OLAP para el análisis multidimensional de la información disponible en el data warehouse que permita a un analista de negocio investigar el rendimiento operativo del área de hospitalización de un hospital general básico. Para ello, debe:

- A partir del análisis de requerimientos:
 - Identificar qué usuarios son los susceptibles de usar análisis multidimensionales, teniendo en cuenta que este tipo de usuarios son avanzados y, por lo tanto, buscar analizar ellos mismos la información.

- Proponer el tipo de preguntas que debe responder el análisis multidimensional para estos usuarios.
- El sistema debe responder cómo mínimo a las siguientes preguntas:
 - Evolución de las hospitalizaciones.
 - Evolución de las hospitalizaciones por tipo de alta.
 - Evolución de las hospitalizaciones por meses y años.
 - Evolución de las hospitalizaciones por servicio del hospital.
- Diseñar e implementar un análisis multidimensional que responda al contexto descrito.
- Mostrar cómo el sistema responde a las preguntas mínimas de los usuarios.

¿Cómo planificar un proyecto de inteligencia de negocio?^{xv}.

Ejecución de un proyecto de inteligencia de negocio^{xvi}.

En principio, ejecutar un proyecto es llevar el plan a la realidad, realizar los procesos, actividades y tareas previstos. Sin embargo, como veremos, este camino no es lineal, no basta con aplicar las previsiones del plan o las reglas de un manual. Ejecutar es conseguir que las cosas se hagan, conocer y controlar el progreso y tomar las medidas de corrección que correspondan. Por eso, ejecución y control van muy unidos. Durante la ejecución, se despliega un mayor número de recursos y, en muchos proyectos, se actúa sobre la organización del cliente; por ello, los costes y los riesgos aumentan muy rápidamente. Existe una gran variedad de proyectos BI y, por lo tanto, metodologías más o menos desarrolladas para cada uno de ellos, en muchas ocasiones por los propios fabricantes. Aquí hemos adoptado una visión que creemos que cubre una mayoría de los proyectos de inteligencia de negocio:

- Los proyectos de implantación de una solución completa de inteligencia de negocio, ya sea basada en el desarrollo a medida, la parametrización de software estándar o una combinación de ambas.
- Los proyectos de implantación de cuadros de mando, ya sean basados en la dirección por

objetivos o en el supuestamente popular cuadro de mando integral (balanced scorecard, BSC), que tienen sus características propias y distintivas. En los proyectos, se suele pensar que si se gestiona la elaboración de un producto ya se maneja el proyecto. Confiamos que, a estas alturas, tendréis claro que gestionar el proyecto es mucho más que construir e implementar unos productos BI. Ahora bien, en el momento de la ejecución es cuando se integra la mayor parte de las actividades y de las metodologías propias de cada producto BI.

Los componentes y temas clave de la ejecución de la implantación de un proyecto de inteligencia de negocio.

Antes de empezar con la descripción de los procesos y herramientas, nos parece importante hacer algunas reflexiones sobre cuáles son los elementos y temas clave de la ejecución de un proyecto de inteligencia de negocio, por encima de las metodologías y herramientas que utilicemos y de los productos o servicios que vayamos a entregar. Puede decirse que la ejecución de un proyecto se sostiene sobre cuatro pilares:

- Las metodologías y conocimientos técnicos propios de cada tipo de proyecto (de su producción) y, en paralelo, el aseguramiento y control de la calidad de los productos, en nuestro caso, los proyectos de BI.
- Las habilidades de liderazgo, comunicación, negociación y gestión del cambio. Son habilidades profesionales de gestión interpersonal, que no podemos desarrollar en profundidad aquí.
- Las necesidades de control interno y reporte, tanto de los aspectos técnicos como de los progresos en el tiempo y los aspectos económicos, que trataremos en el apartado “Seguimiento y control del proyecto”.
- Habilidades específicas de atención de incidencias, resolución de problemas y toma de decisiones.

Inves_4pi



Fuente: Rodríguez, García y Lamarca (2007)

La ejecución, especialmente en proyectos de aplicación de las TIC, es la fase en la que se despliegan las habilidades, conocimientos y metodologías específicas de la profesión y de cada clase de proyectos. “La competencia profesional no es suficiente para asegurar el éxito si los aspectos gerenciales no funcionan. Y al contrario, ninguna clase de ayuda administrativa puede asegurar el éxito si falta la competencia profesional. Los dos (gestión y capacidad profesional) son cruciales para el éxito”. Andersen y otros (2006) Dirección o ejecución del proyecto no quiere decir control. Control no es dirección, es solo una parte de la dirección. La dirección de proyecto incluye saber organizar y asignar recursos, comunicar y motivar a los equipos, relacionarse con el cliente y con las diferentes partes involucradas. Dirección tiene que ver con las personas; es, en buena medida, el lado humano del proyecto.

Cómo usar un modelo H2PAC.

Un proyecto de inteligencia de negocio (o business intelligence) se basa en la construcción o parametrización de un determinado componente de inteligencia de negocio (data warehouse, cuadros de mando, soluciones analíticas, etc.) o de una solución completa de un determinado proveedor. En la fase de ejecución de un proyecto de inteligencia de negocio es donde aparecen las principales diferencias con la gestión de un proyecto TIC. En esta fase se despliegan las metodologías y los instrumentos propios para construir un sistema de inteligencia de negocio. Es durante la ejecución de un proyecto cuando la planificación se convierte en una realidad y se deben gestionar los recursos y las situaciones inesperadas en el trabajo del día a día del proyecto. Es el momento de revisar y hacer un seguimiento de la planificación inicial, aprobando aquello que funciona y realizando los cambios que sean necesarios. Además, para que el proyecto sea un éxito debe ir acompañado también de un cambio, a veces profundo, en la cultura de la organización, que debe ser liderado desde la dirección. También se deben desarrollar otras capacidades propias dentro de la empresa, así como crear nuevos roles y responsabilidades. La actividad que se presenta tiene como objetivo trabajar los principales conceptos que se barajan en las etapas de desarrollo y ejecución en la gestión de un proyecto. La actividad se basa en un proyecto de inteligencia de negocio del entorno financiero y está estructurada en dos bloques. En el primer bloque se

contextualizan y trabajan los objetivos relacionados con la etapa de desarrollo de la gestión de proyecto de inteligencia de negocio. En el segundo bloque se muestran tres situaciones que se pueden presentar en la gestión del día a día de un proyecto de inteligencia de negocio.

En la realización de la actividad se alcanzarán los siguientes objetivos:

- Identificar las diferentes etapas en la gestión de un proyecto de inteligencia de negocio.
- Conocer los cuatro pilares de la ejecución de un proyecto de inteligencia de negocio.
- Saber identificar el grupo de interesados o stakeholders de un proyecto.
- Saber identificar y gestionar los principales riesgos de un proyecto.
- Desarrollar las habilidades básicas de comunicación y trabajo en equipo.
- Planificar los hitos principales en la gestión de un proyecto.
- Saber identificar las partidas de coste.
- Realizar un prototipaje.
- Identificar las fuentes de datos de un proyecto de inteligencia de negocio.
- Identificar los procesos ETL.

La importancia de los data warehouse.^{xvii}

Como hemos mencionado anteriormente, un data warehouse es un almacén de datos e información. Normalmente se aloja en el servidor de una empresa o en la nube. Los datos de diferentes aplicaciones de procesamiento de transacciones online (OLTP) y de otras fuentes son recogidos y seleccionados. Posteriormente sirven para realizar análisis y informes relacionados con el Business Intelligence. Gracias a esto, las grandes compañías toman decisiones para establecer objetivos y estrategias.

Las principales funciones de un **almacén de datos** consisten en integrar datos que provienen de las bases de datos distribuidas por las diferentes unidades de la empresa. Para ello es necesario facilitar una descripción global y análisis de toda la organización en el data warehouse.

También sirven para separar los datos usados en operaciones diarias y los usados en el almacén para distintos propósitos. Algunos de ellos son para divulgación, como soporte en la toma de decisiones o para análisis. Ambos tipos no deben coincidir en la misma base de datos porque los objetivos por los cuales se van a utilizar son distintos.

Tipos de estructuras de un data warehouse

Las estructuras de los almacenes de datos pueden ser de tres tipos:

- Básica. Proporciona datos brutos que se almacenan junto con metadatos.
- Básica con un área de ensayo. Ofrece un espacio donde los datos se pueden “limpiar” antes de entrar al data warehouse.
- Básica con un área de ensayo con data marts. Estos sistemas están diseñados para un tipo de negocio en concreto.

Ventajas y desventajas de los almacenes de datos

El uso de almacenes de datos presenta una serie de pros y contras. Entre sus ventajas destacan:

- Facilitan el acceso a una gran variedad de datos.
- Favorecen el funcionamiento de las aplicaciones de los sistemas de apoyo en la toma de decisiones, como informes de tendencia o de excepción.
- Pueden trabajar en conjunto y aumentar el valor operacional de las aplicaciones empresariales.

Ahora bien, entre sus inconvenientes se remarca el alto precio que pueden costar a lo largo de su vida útil debido a su mantenimiento. Además, suelen quedarse obsoletos en pocos años y no retornan el tipo de información demandada.

Un data warehouse en la nube.

Una de las tendencias que no ha parado de crecer en los últimos años es el traslado de los data warehouses a la nube. Eso sí, estos no se adhieren a la arquitectura tradicional de los almacenes de

datos, sino que cada uno tiene una arquitectura única.

Este tipo de almacén presenta una serie de ventajas muy importantes. Aquí te dejamos algunas:

- **Ahorro.** La inversión inicial es mucho menor que el almacén de datos tradicional. La nube permite liberar recursos que anteriormente se habrían destinado a tareas de gestión de data warehouse.
- **Velocidad.** Permite poner rápidamente en marcha los sistemas, lo que provoca una mayor agilidad en el trabajo de los usuarios y una mayor productividad.
- **Escalabilidad.** Los servicios en la nube pueden ampliarse o reducirse fácilmente.
- **Análisis.** La nube puede ser el medio ideal para fusionar datos de los almacenes locales con estas nuevas fuentes de big data. De esta manera, las compañías pueden alcanzar altos niveles de conocimiento.

Base de datos o data warehouse: ¿cuál es la diferencia?

En el paisaje empresarial de 2019, los datos son la única moneda que cuenta. El éxito de una empresa el año que viene y los siguientes dependerá por completo del volumen de datos que recopile, de su precisión y de las posibilidades de creación de informes que permitan, así como de su capacidad para analizar dichos datos, derivar información útil de ellos y actuar en consecuencia. La decisión se reduce a una base de datos o un data warehouse, pero empecemos por explicar qué es y por qué se utiliza cada uno.

Mejora de un data warehouse con cubos.^{xviii}

Para gestionar todos los datos integrados de un data warehouse, muchas empresas emplean cubos para poder crear rápidamente informes y análisis. Un cubo es una sección multidimensional de datos creada a partir de las tablas de un data warehouse. Cada cubo contiene métricas contextuales, pertinentes y útiles para su área empresarial particular. Cuando hace una consulta para obtener respuestas basadas en sus datos, esta se dirige al cubo apropiado.

Informes que antes tardaban 5 minutos en crearse ahora se elaboran en cuestión de segundos, y los usuarios finales ya no tienen por qué comprender la compleja red de referencias que vincula una colección de tablas. Cuando las organizaciones comienzan a recopilar información de varias bases de datos, el tamaño de los datos crece de forma exponencial. Al realizar una consulta estándar en grandes conjuntos de datos de una base de datos relacional activa, se producen problemas de rendimiento graves que no solo afectan a la productividad, sino que pueden llevar a los usuarios a dejar de lado los informes. Cuando esto sucede, se descarta información importante porque los usuarios sencillamente no tienen tiempo para esperar a que los datos se compilen.

Alto retorno de la inversión.

Al combinarlos con software de BI para, en esencia, democratizar los datos y reducir el personal en funciones de análisis y creación de informes, las empresas recuperan su inversión más rápido que nunca.

Desempeño mejorado.

En vez de dedicar una valiosa capacidad de computación a la edición y gestión de registros de datos aislados, el objetivo de los data warehouses es permitir cuanto antes el acceso a los datos, así como su cotejo y análisis, de modo que se puedan tomar decisiones empresariales esenciales al momento, sin que los responsables pierdan un tiempo valioso esperando a que las consultas se carguen. Si busca una solución que lo ayude a analizar los datos de dichas transacciones, le recomendamos encarecidamente que se plantee el uso de un data warehouse. Además de contar con una extensa biblioteca de plantillas de informes y cuadros de mando, Jet Analytics está pensado para ofrecerle información valiosa sobre sus datos desde el primer día. En los próximos años, la calidad, coherencia y accesibilidad de los datos será lo que marque la diferencia para empresas de todos los tamaños.⁸

⁸ Resumen realizado por alumno, desde:
[https://insightsoftware.com/es/blog/database-vs-data-](https://insightsoftware.com/es/blog/database-vs-data-warehouse-whats-the-difference/)

ⁱ La guía sencilla para la diagramación de UML. - Microsoft 365 Team. Septiembre 24, 2019.

ⁱⁱ Enrique Rus Arias (15 de agosto, 2020). Modelo relacional. Economipedia.com

ⁱⁱⁱ Importancia de UML - <https://bit.ly/3ztj1iW>

^{iv} <https://bit.ly/3hRy7ZQ> - Modelando el Proceso de Negocio - Autor: Geoffrey Sparks, Sparx Systems, Australia

^v Ventajas y significados de UML - <https://bit.ly/3ELNmxk>

^{vi} Casas Roma, J. y i Caralt, J. C. (2014). *Diseño conceptual de bases de datos en UML*. Barcelona, Spain: Editorial UOC. Recuperado de <https://elibro.net/es/ereader/iacc/57635?page=108>.

^{vii} Casas Roma, J. y i Caralt, J. C. (2014). *Diseño conceptual de bases de datos en UML*. Barcelona, Spain: Editorial UOC. Recuperado de <https://elibro.net/es/ereader/iacc/57635?page=42>.

^{viii} Referencia desde: <https://www.tableau.com/es-es/learn/articles/business-intelligence>.

^{ix} <https://www.esan.edu.pe/apuntes-empresariales/2020/07/data-warehouse-pasos-para-construirlo-con-exito/> - (Data warehouse: pasos para construirlo con éxito).

^x <https://www.esan.edu.pe/apuntes-empresariales/2018/04/gerencia-del-conocimiento-en-donde-se-aplican-el-warehouse-y-crm/> (Gerencia del conocimiento: ¿en dónde se aplican el warehouse y CRM?).

^{xi} <http://biverano2011.blogspot.com/2011/09/modelo-estrella-y-modelo-copo-de-nieve.html> (Business Intelligence).

^{xii} Referencia desde: <https://www.ibm.com/mx-es/analytics/data-warehouse>.

^{xiii} Referencia desde:

<https://revistadigital.inesem.es/informatica-y-tics/guia-construir-datawarehouse/>.

^{xiv} Curto Díaz, J. y Conesa Caralt, J. (Ed.) (2015). ¿Cómo crear un data warehouse?. Barcelona, Editorial UOC. Recuperado de <https://elibro.net/es/ereader/iacc/114035?page=20>.

^{xv} Rodríguez, J. R. González Farran, X. y Guitart, I. (2016). ¿Cómo planificar un proyecto de inteligencia de negocio?. Barcelona, Spain: Editorial UOC. Recuperado de <https://elibro.net/es/ereader/iacc/58548?page=6>.

^{xvi} Rodríguez, J. R. González Farran, X. y Guitart, I. (2016). ¿Cómo planificar un proyecto de inteligencia de negocio?. Barcelona, Spain: Editorial UOC. Recuperado de <https://elibro.net/es/ereader/iacc/58548?page=25>.

^{xvii} <https://escuelafintech.com/data-warehouse/>. (Fintech School)

^{xviii} <https://insightsoftware.com/es/blog/database-vs-data-warehouse-whats-the-difference/> - (insightsoftware.com)