

Unidad 5

UNIDAD 5 MANTENER Y DOCUMENTAR UNA BASE DE DATOS

Prof. Mg. Ing. Roxana Martínez



UAIOnline
Ultra»»



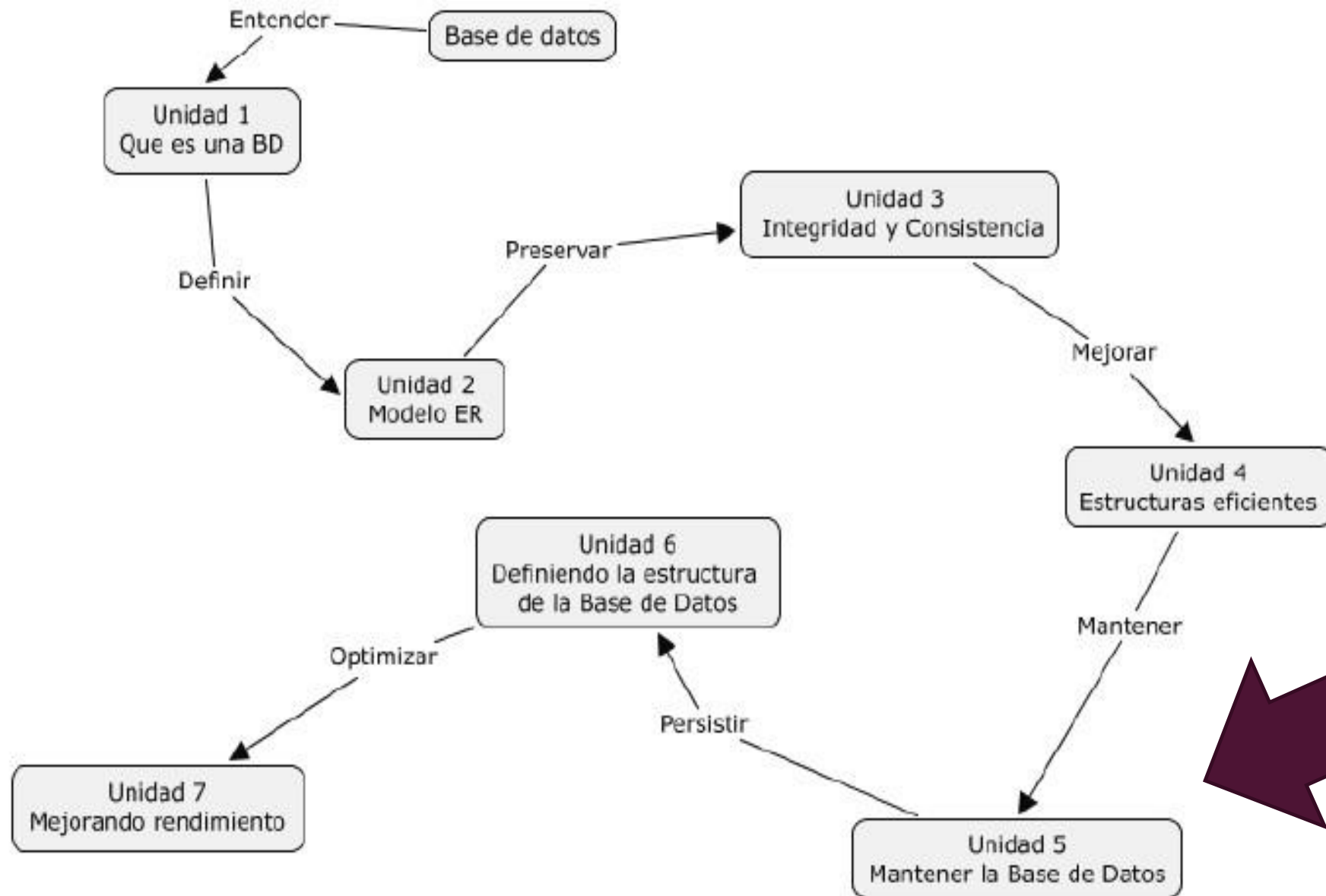
DIAGRAMA DE ENTIDAD RELACIÓN

Unidad 5

■ OBJETIVOS

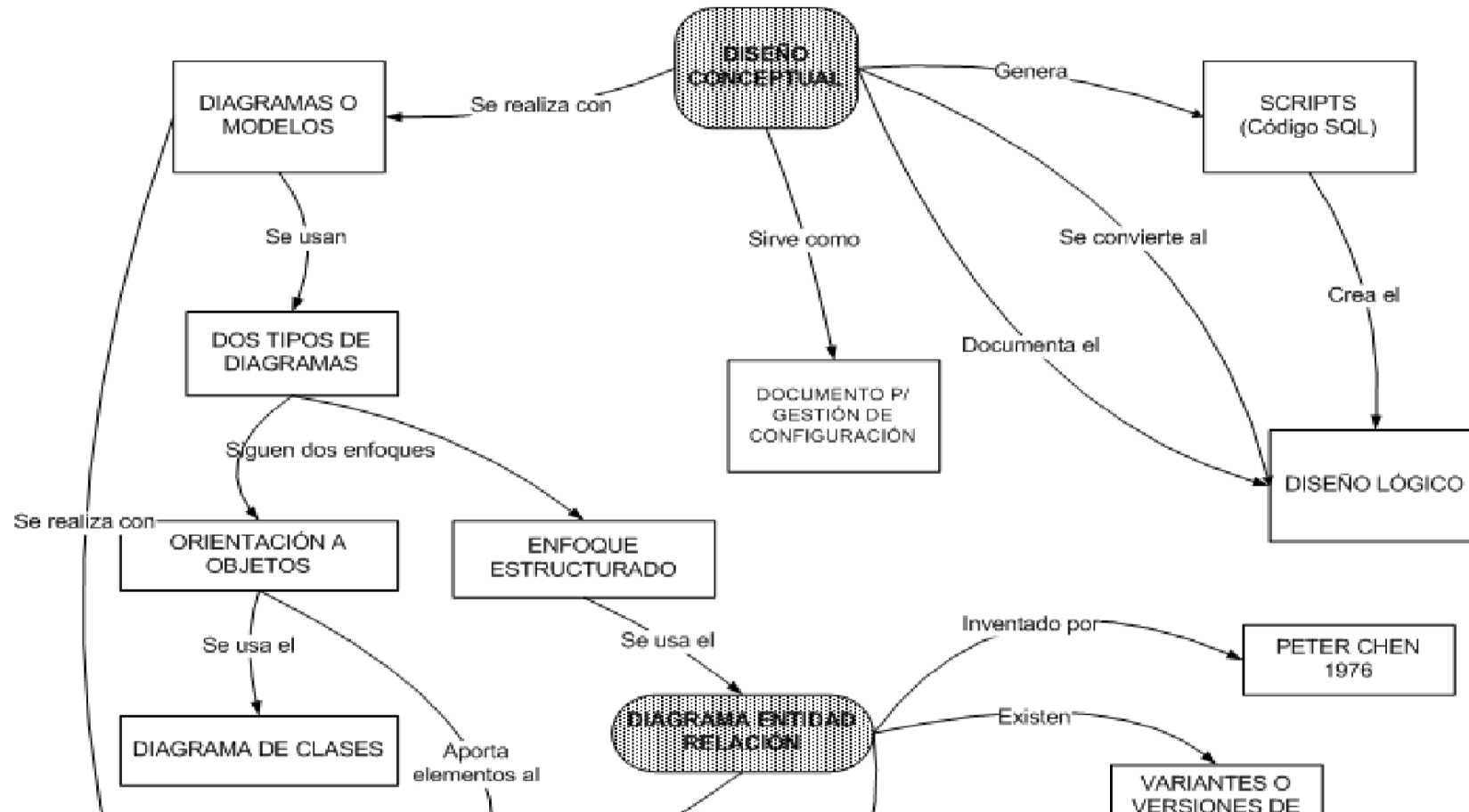
- Comprender los conceptos de Reingeniería y reestructuración de datos. Ingeniería inversa. Herramientas para la ingeniería inversa. Herramientas CASE. Reingeniería orientada a objetos. Que es un diccionario de datos.
- Documentar y mantiene una base de datos. Reestructuración de datos. Ingeniería inversa. Herramientas para la ingeniería inversa. Herramientas CASE. Importancia de los metadatos.

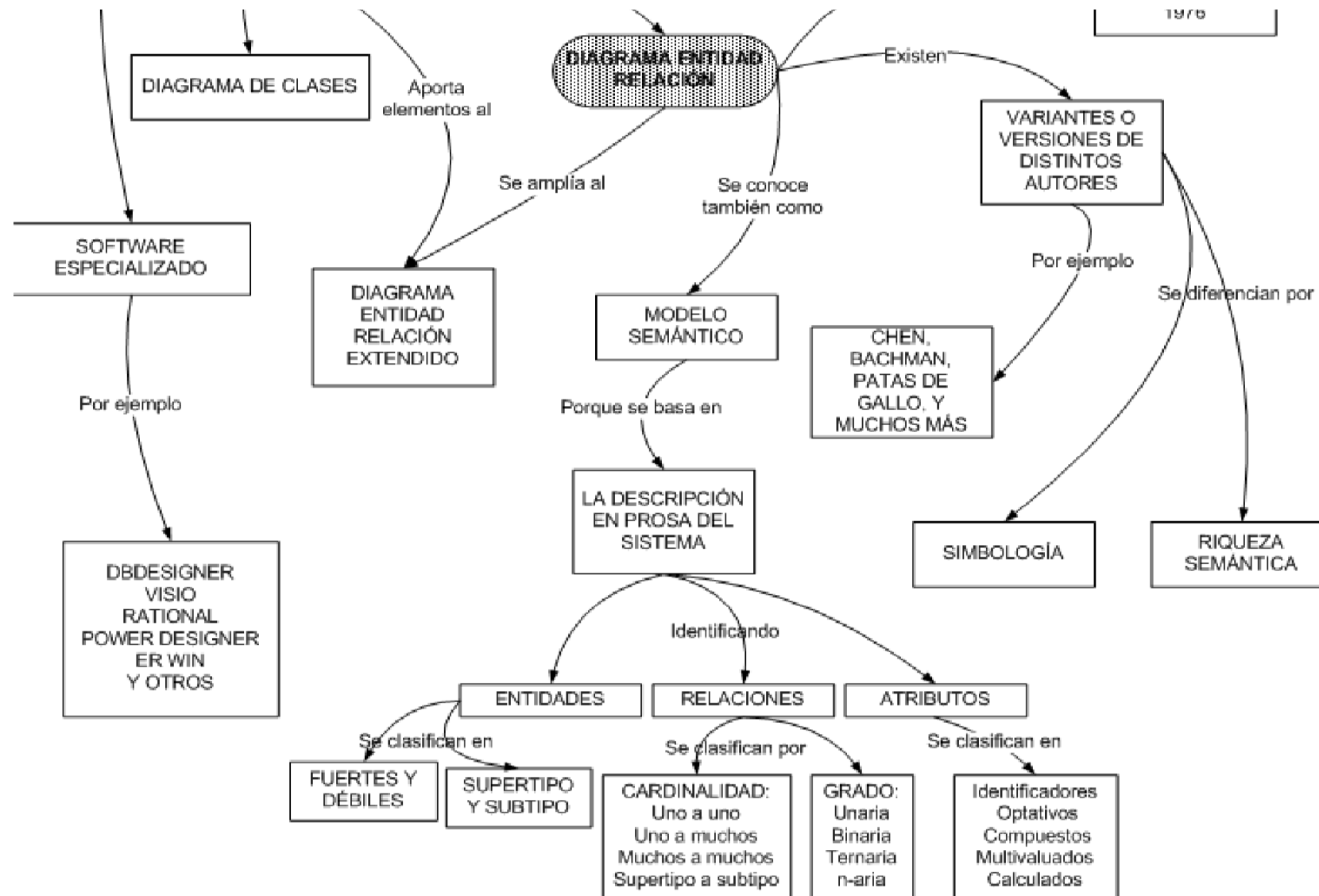




DISEÑO CONCEPTUAL DE BASES DE DATOS

MAPA CONCEPTUAL





Reingeniería



UAIOnline
Ultra»»



REINGENIERÍA

Reingeniería *es el rediseño fundamental y radical de los procesos del negocio*, para lograr mejoras dramáticas en medidas críticas del desempeño, tales como costo, calidad, servicio y rapidez.

La reingeniería es un proceso total de readecuación de las Organizaciones a las nuevas y exigentes condiciones en un entorno cada vez más difícil de controlar.

REINGENIERÍA

Se fundamenta en las tres "C":

Cliente

Competencia

Cambio: Busca que las Empresas sean más efectivas.



Eficiencia + eficacia = efectividad



Reingeniería

Organizacional

ELEMENTOS CLAVE DE LA REINGENIERÍA

- Identificación de los procesos empresariales más importantes, a cualquier nivel, con el propósito de **re-conceptualizar**.
- Enfoque global**, a fin de salvar conceptual y funcionalmente las fronteras que establecen entre unidades que concurren en un mismo proceso.
- Apoyo y compromiso** de cambio por parte e la alta gerencia y de los trabajadores.
- Uso de nueva tecnología**, como elemento sinérgico y dinamizador de los cambios.

REINGENIERÍA

Objetivos de la Reingeniería

Persigue definir criterios de simplificación y optimización que permiten alcanzar las metas del cambio

REINGENIERÍA: CRITERIOS

- Racionalizar las operaciones
- Reducir los costos
- Mejorar la calidad
- Aumentar los ingresos
- Mejorar la orientación hacia los clientes basándose en:
 - Definir responsabilidades funcionales
 - Distribución de carga de trabajos
 - Reducción de tiempo
 - Apoyo de otras áreas
 - Evitar duplicidad e inconsistencia
 - Orientación al servicio al cliente

REINGENIERÍA DEL SOFTWARE

¿Qué es Reingeniería del Software?

Se puede definir como: *“modificación de un producto software, o de ciertos componentes, usando para el análisis del sistema existente técnicas de Ingeniería Inversa y, para la etapa de reconstrucción, herramientas de Ingeniería Directa, de tal manera que se oriente este cambio hacia mayores niveles de facilidad en cuanto a mantenimiento, reutilización, comprensión o evaluación.”*

REINGENIERÍA

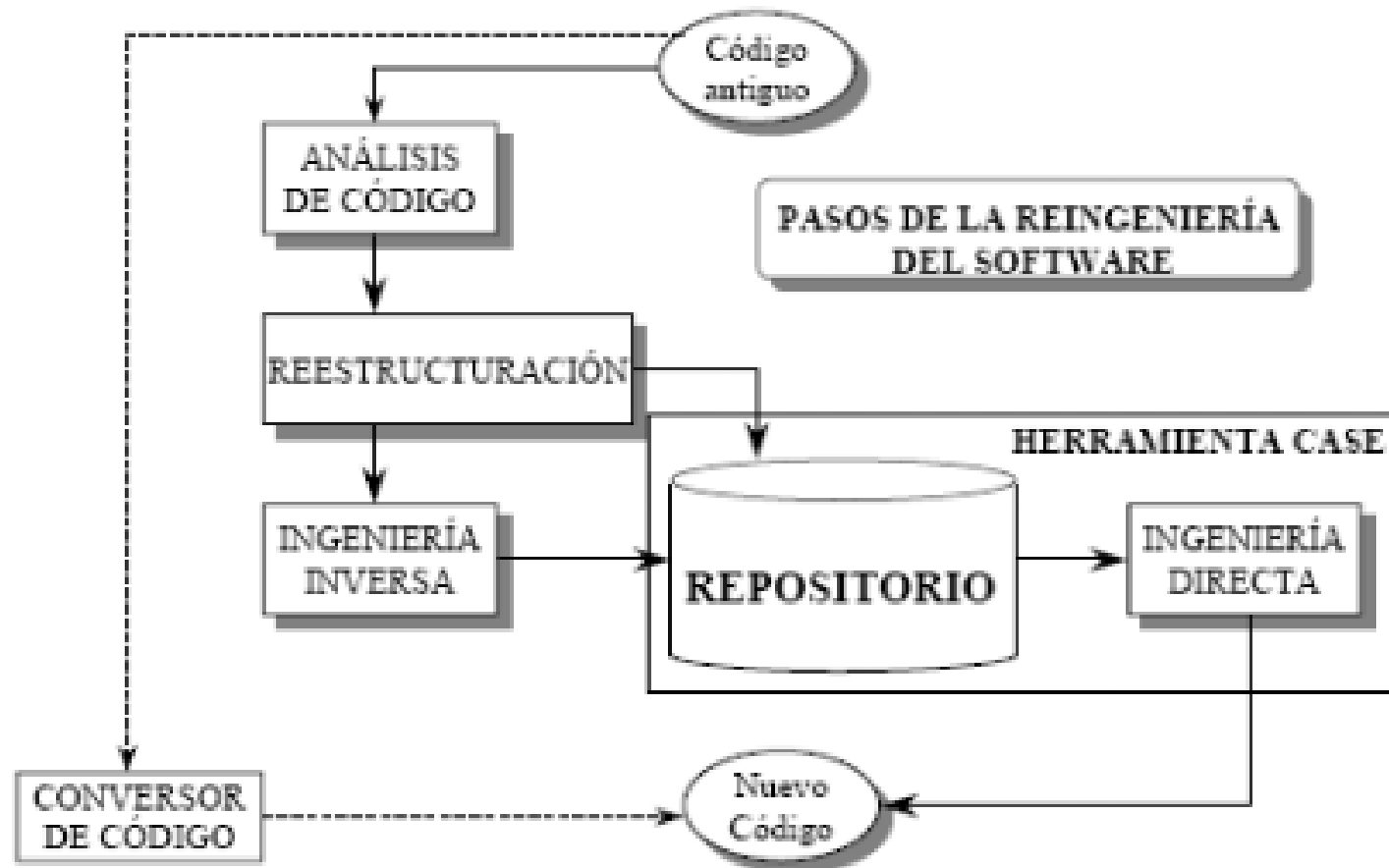
Beneficios de aplicar Reingeniería a un producto existente:

- Pueden reducir los riesgos evolutivos de una organización.
- Puede ayudar a las organizaciones a recuperar sus inversiones en software.
- Puede hacer el software más fácilmente modificable
- Amplía las capacidades de las herramientas CASE.
- Es un catalizador para la automatización del mantenimiento del software.
- Puede actuar como catalizador para la aplicación de técnicas de inteligencia artificial para resolver problemas de reingeniería.

La reingeniería del software involucra diferentes actividades como son:



PASOS DE LA REINGENIERÍA DEL SOFTWARE



ANÁLISIS DE INVENTARIOS

Todas las organizaciones de software deberían tener un inventario de todas sus aplicaciones.

Los candidatos a la reingeniería aparecen cuando se ordena esta información en función de su importancia para el negocio, es entonces cuando es posible asignar recursos a las aplicaciones candidatas para el trabajo de reingeniería.

REESTRUCTURACIÓN DE DOCUMENTOS

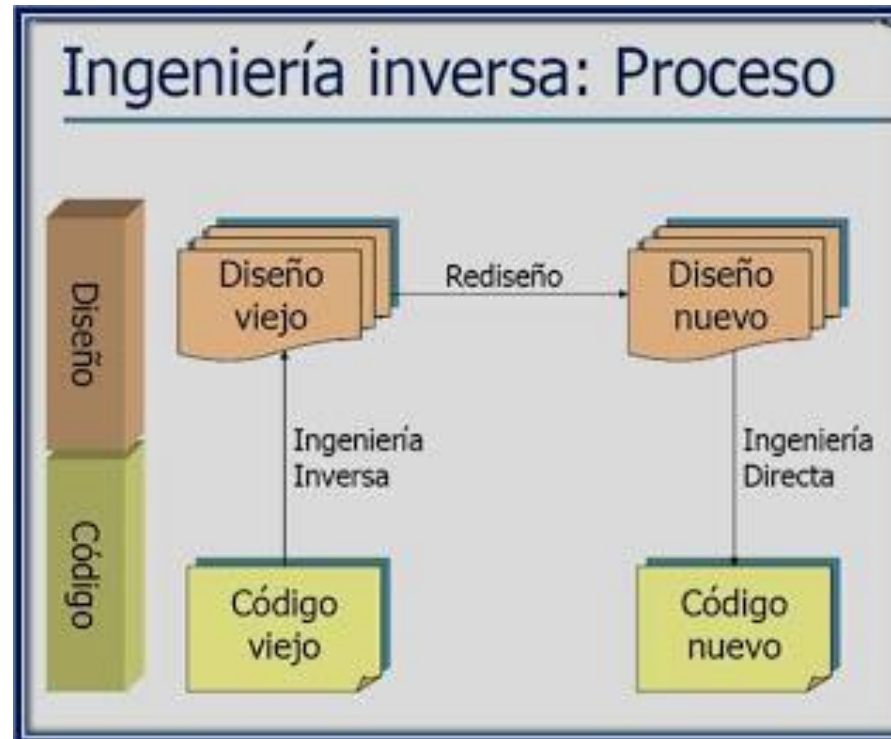
La documentación débil es la marca de muchos sistemas heredados. ¿Pero que se hace acerca de ellos? ¿Cuáles son las opciones? Crear documentación consume mucho tiempo, si el sistema funciona vivirá con lo que tenga.

La documentación debe actualizarse pero se tiene recursos limitados. Se utiliza un enfoque de “documentar cuando se toque”.

El sistema es crucial para el negocio y debe volver a documentarse por completo incluso en este caso un enfoque inteligente es recortar la documentación a un mínimo esencial. Cada una de estas opciones es viable. Una organización de software debe elegir la más apropiada para cada caso.

INGENIERÍA INVERSA

Ingeniería inversa con éxito precede de una o más especificaciones de diseño y fabricación para el producto, mediante el examen de ejemplos reales de ese producto.



INGENIERÍA INVERSA

La ingeniería inversa del software es el proceso de análisis de un programa con el fin de crear una representación de programa con un nivel de abstracción más elevado que el código fuente. ***La Ingeniería inversa es un proceso de recuperación de diseño.*** Con las herramientas de la ingeniería inversa se extraerá del programa existente información del diseño arquitectónico y de proceso, e información de los datos.

REESTRUCTURACIÓN DE CÓDIGO

El tipo más común de reingeniería es la reestructuración de código, se puede hacer con **módulos individuales** que se codifican de una manera que dificultan comprenderlos, probarlos y mantenerlos.

Llevar a cabo esta actividad requiere analizar el código fuente empleando una herramienta de reestructuración, se indican las violaciones de las estructuras de programación estructurada, y entonces se reestructura el código (esto se puede hacer automáticamente).

El código reestructurado resultante se revisa y se comprueba para asegurar que no se hayan introducido anomalías. Se actualiza la documentación interna del código.

REESTRUCTURACIÓN

Reestructuración de datos / I

La reestructuración de datos es una actividad de reingeniería a gran escala. En la mayoría de los casos, la reestructuración de datos comienza con una actividad de ingeniería inversa.

La arquitectura de datos actual se analiza con minuciosidad y se define los modelos de datos necesarios, se identifican los objetivos de datos y los atributos, y después se revisa la calidad de las estructuras de datos existentes.

REESTRUCTURACIÓN

Reestructuración de datos /2

Cuando la estructura de datos es débil (por ejemplo, actualmente se implementan archivos planos, cuando un enfoque relacional simplificaría muchísimo el procesamiento), se aplica una reingeniería a los datos.

Dado que la arquitectura de datos tiene una gran influencia sobre la arquitectura del programa, y también sobre los algoritmos que lo pueblan, los cambios en datos darán lugar invariablemente a cambios o bien de arquitectura o bien de código.

INGENIERÍA DIRECTA

En un mundo ideal, las aplicaciones se reconstruyen utilizando un “motor de reingeniería” automatizado. En el motor se insertaría el programa viejo, que lo analizaría, reestructuraría y después regeneraría la forma de exhibir los mejores aspectos de la calidad del software.

Después de un espacio de tiempo corto, es probable que llegue a aparecer este “motor”, pero los fabricantes de **CASE** han presentado herramientas que proporcionan un subconjunto limitado de estas capacidades y que se enfrentan con dominios de aplicaciones específicos.

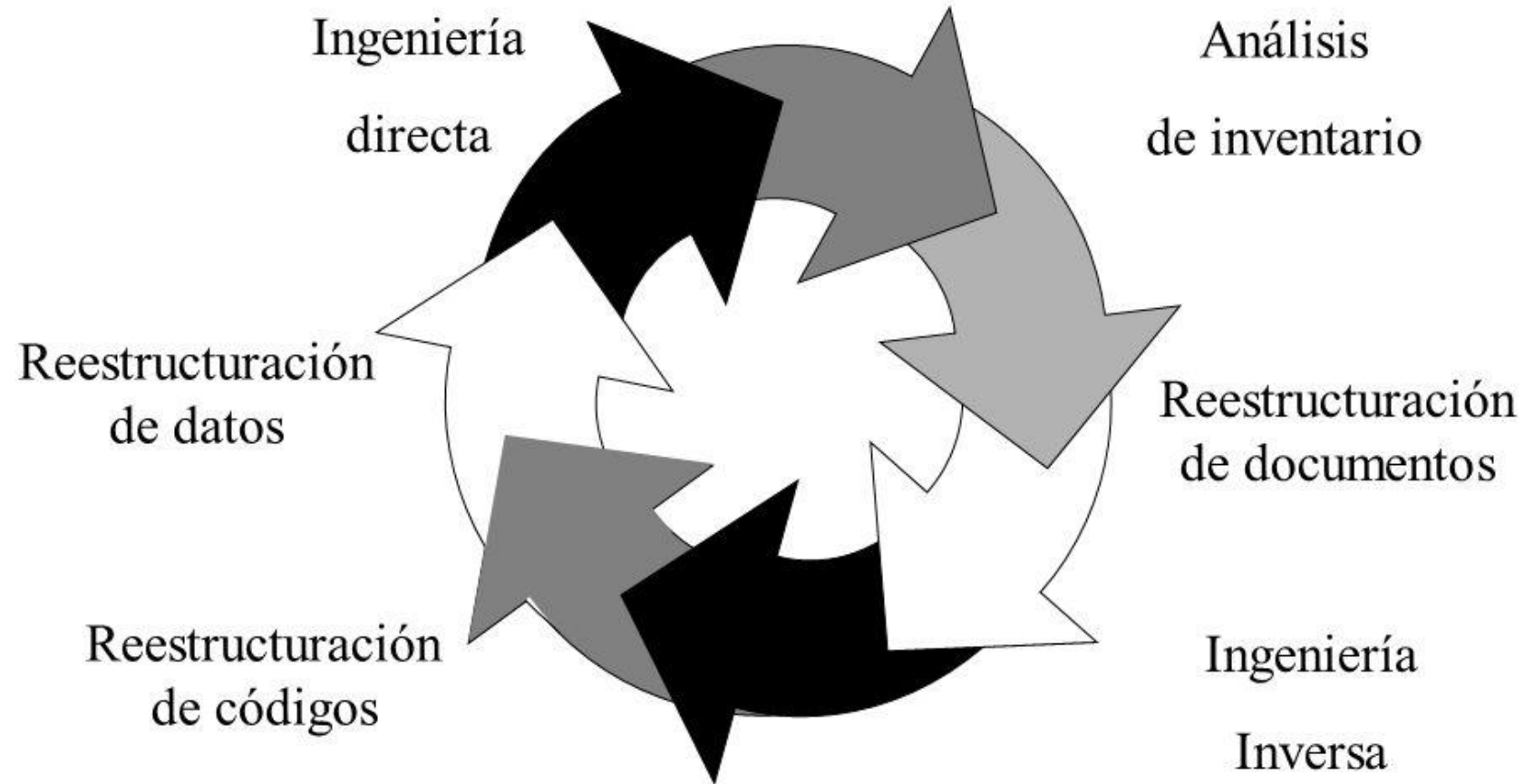
INGENIERÍA DIRECTA

Lo que es más importante, estas herramientas de reingeniería cada vez son más sofisticadas.

La ingeniería directa no solo recupera la información de diseño a partir del software existente, también utiliza esta información para alterar o reconstruir el sistema existente con la finalidad de mejorar su calidad global.

En la mayoría de los casos el software sometido a reingeniería vuelve a implementar la función del sistema existente y también añade nuevas funciones o mejoras.

Modelo de proceso de reingeniería de software



Mantenimiento de Software



Tipos de datos

Clasificación de tipos

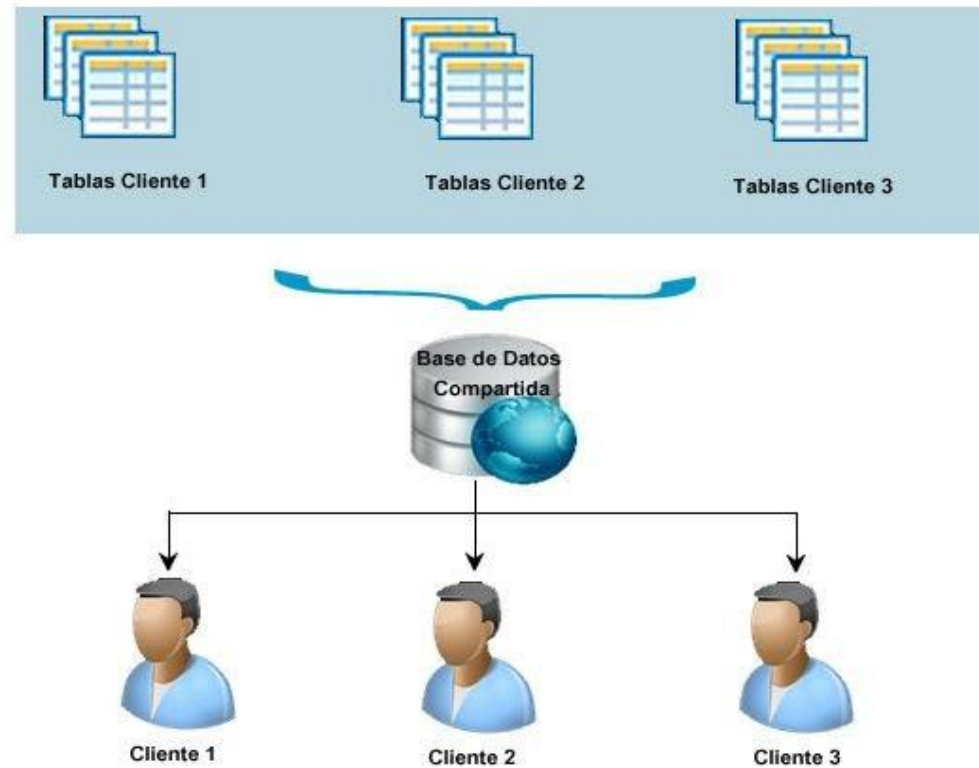
✓ Simples

- ❖ Estándar: `int`, `float`, `double`, `char`, `bool`
Conjunto de valores predeterminado
- ❖ Definidos por el usuario: *enumerados*
Conjunto de valores definido por el programador

✓ Estructurados

- ❖ Colecciones homogéneas: *arrays*
Todos los elementos del mismo tipo
- ❖ Colecciones heterogéneas: *estructuras*
Los elementos pueden ser de tipos distintos

Estructura de una Base de datos



TIPOS DE INGENIERÍA INVERSA

Ingeniería inversa de datos:

Se aplica sobre algún código de bases datos (aplicación, código SQL, etc.) para obtener los modelos relacionales o sobre el modelo relacional para obtener el diagrama entidad-relación.

Ingeniería inversa de lógica o de proceso:

Cuando la ingeniería inversa se aplica sobre código de un programa para averiguar su lógica o sobre cualquier documento de diseño para obtener documentos de análisis o de requisitos.

Ingeniería inversa de interfaces de usuario:

Se aplica con objeto de mantener la lógica interna del programa para obtener los modelos y especificaciones que sirvieron de base para la construcción de la misma, con objeto de tomarlas como punto de partida en procesos de ingeniería directa que permitan modificar dicha interfaz.

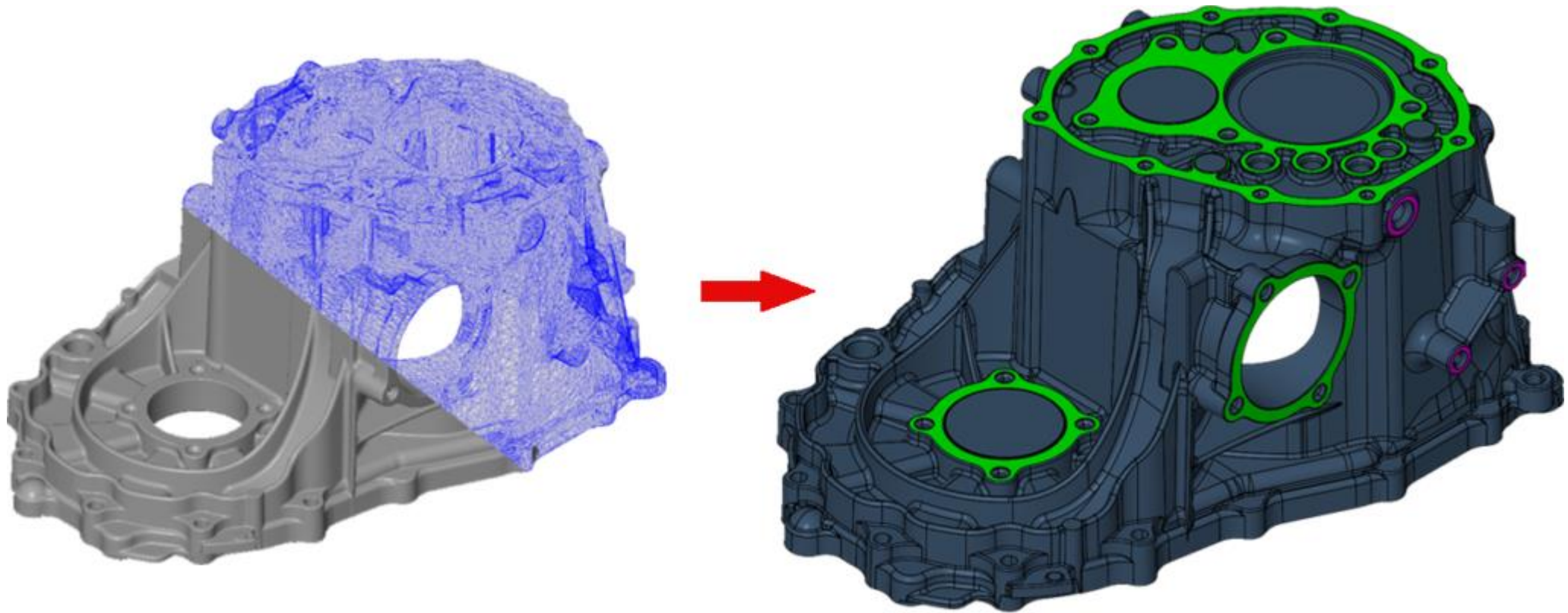
La ingeniería inversa es un proceso que persigue el fin último de obtener detalles e información a partir de un producto o pieza, y de esta manera definir su composición y funcionamiento, además descubrir la manera en que dichas piezas interactúan entre sí.

INGENIERÍA INVERSA



Este método de producción se conoce así debido a que **se desarrolla en el sentido inverso al método habitual** de la ingeniería, método por el cual se utilizan datos técnicos y modelados para la producción de una pieza determinada.

BENEFICIOS DE LA INGENIERÍA INVERSA



HERAMIENTAS PARA LA INGENIERÍA INVERSA

Los Depuradores.

Las Herramientas de Inyección de Fallos.

Los Desensambladores.

Los compiladores Inversos o Decompiladores.

Las Herramientas CASE.

HERAMIENTAS PARA LA INGENIERÍA INVERSA

Los Depuradores:

La depuración de programas es el proceso de identificar y corregir errores de programación.

HERAMIENTAS PARA LA INGENIERÍA INVERSA

Herramientas de Inyección de Fallos:

En las pruebas de software , inyección de fallos es una técnica para mejorar la cobertura de una prueba mediante la introducción de fallos para probar las rutas de código, en particular de tratamiento de errores rutas de código, que de otra manera rara vez seguir. A menudo se utiliza con las pruebas de estrés y es ampliamente considerado como una parte importante del desarrollo robusto software. Pruebas Robustez (también conocido como Sintaxis Testing, Fuzzing o pruebas de la pelusa) es un tipo de inyección de fallos utilizada para la prueba de vulnerabilidades en interfaces de comunicación tales como los protocolos, parámetros de línea de comando, o APIs.

HERAMIENTAS PARA LA INGENIERÍA INVERSA

Los Desensambladores:

Es un programa de computador que traduce el lenguaje de máquina a lenguaje ensamblador, la operación inversa de la que hace el ensamblador.

El código fuente en lenguaje ensamblador generalmente permite el uso de constantes y comentarios del programador. Estos son generalmente eliminados, por el ensamblador, del código ensamblado a código de máquina. De esta manera, un desensamblador operando sobre el código de máquina, produciría el desensamblado careciendo de estas constantes y comentarios; la salida desensamblada se vuelve más difícil de ser interpretada por un humano que el código fuente con comentarios original.

HERAMIENTAS PARA LA INGENIERÍA INVERSA

Los compiladores Inversos o Decompiladores.

Un decompilador es un programa de ordenador que realiza la operación inversa a un compilador, es decir, traduce código de bajo nivel de abstracción a un lenguaje de mayor nivel de abstracción.

Según su nivel de abstracción

Lenguajes Máquina :

están escritos en lenguajes directamente inteligibles por la máquina (computadora), ya que sus instrucciones son cadenas binarias (0 y 1).

Lenguajes de bajo nivel

Los lenguajes de bajo nivel son lenguajes de programación que se acercan al funcionamiento de una computadora. El lenguaje de más bajo nivel por excelencia es el código máquina.

Lenguaje de alto nivel

Los lenguajes de alto nivel son normalmente fáciles de aprender porque están formados por elementos de lenguajes naturales, como el inglés

HERAMIENTAS PARA LA INGENIERÍA INVERSA: HERRAMIENTAS CASE

Aunque no es fácil y **no existe una forma única de clasificarlas**, las herramientas CASE se pueden clasificar teniendo en cuenta los siguientes parámetros:

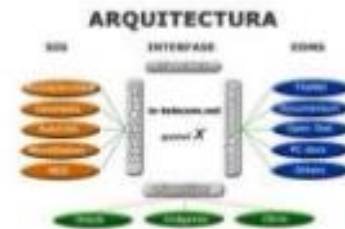
Por Las
plataformas
que
soportan.



Por Las fases
del ciclo de
vida del
desarrollo de
sistemas que
cubren.



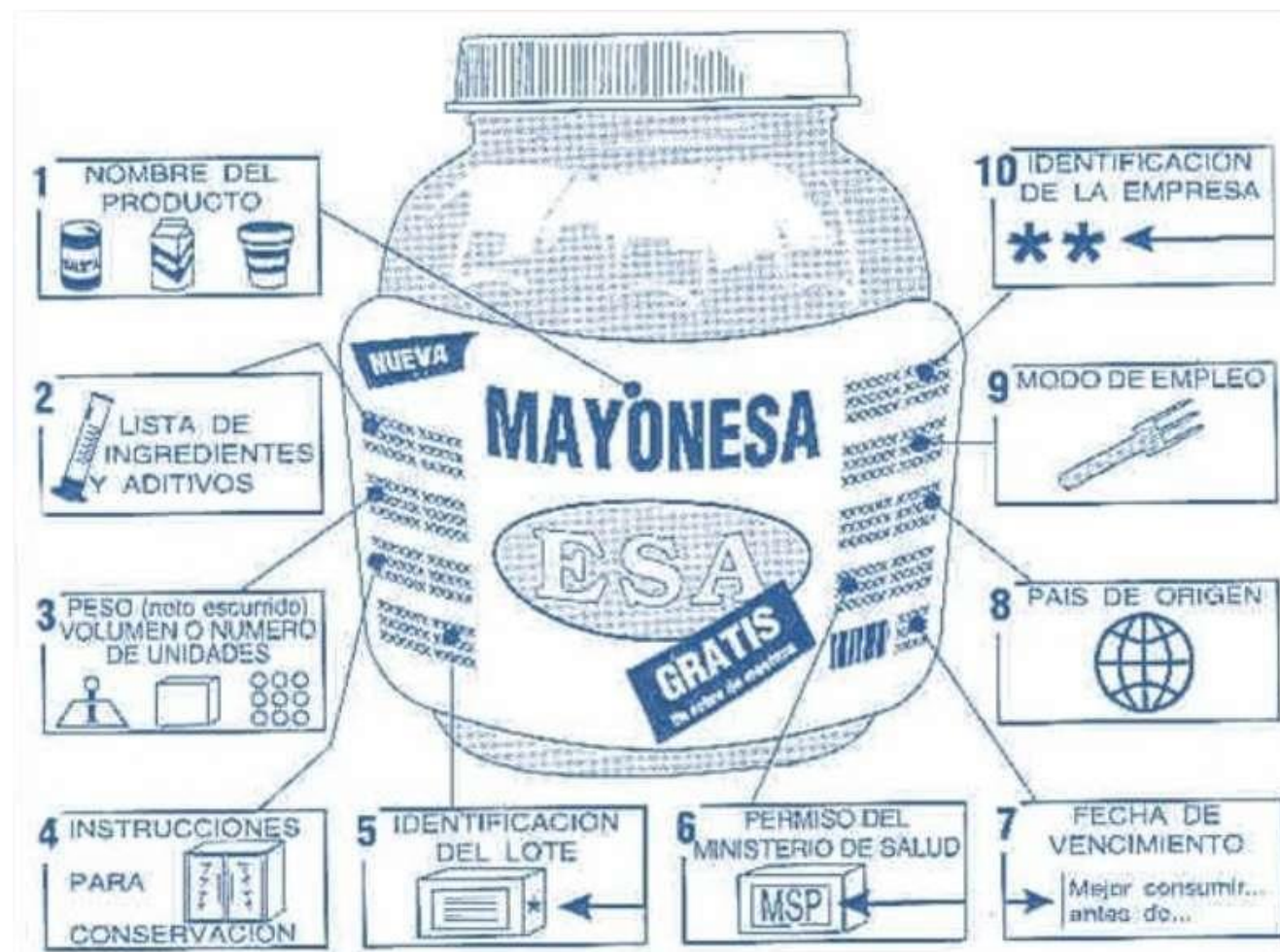
Por La
arquitectura de
las aplicaciones
que producen.



Por Su
funcionalidad



METADATO



DICCIONARIO DE DATOS

Cliente					
Llave	Nombre	Campo	Tipo	Tamaño	Descripción
PK	Código Empleado	cdcl	Número	4	Almacena Código del cliente
	Nombre	nbcl	Texto	50	Almacena nombre del cliente
	Apellidos	apcl	Texto	50	Almacena apellidos del cliente
	Teléfono	tlcl	Número	10	Almacena teléfono del cliente
	Género	gncl	Texto	10	Almacena Género del cliente
	E_mail	emclu	Hipervínculo	100	Almacena correo electrónico del cliente
	E_mail Opcional	emcld	Hipervínculo	100	Almacena correo electrónico del cliente opcional
	Dirección	drcl	Memo	100	Almacena dirección del cliente
FK	Código barrio	cdbrr	Número	4	Almacena código del barrio
FK	Código Ciudad	cdcdd	Número	4	Almacena código de la ciudad
FK	Código Localidad	cdl	Número	4	Almacena código de la localidad



¿Consultas?



UAI

**Universidad Abierta
Interamericana**