



**Universidad Autónoma del Estado de
Hidalgo**

Instituto de Ciencias Básicas e Ingeniería

Licenciatura en Sistemas Computacionales

PANORAMA GENERAL DE LAS “HERRAMIENTAS CASE”

MONOGRAFÍA:

**Para obtener el título de Licenciatura en Sistemas
Computacionales**

PRESENTA:

P.L.S.C Blanca Elizabeth Vargas Martínez

ASESOR:

Lic. Luis Islas Hernández

INDICE

Introducción	I
Objetivo General	III
Objetivos Particulares	III
Justificación	IV
Capítulo 1: Introducción a las herramientas CASE	1
1.1 Definiciones	1
1.2 Objetivos de las Herramientas CASE	6
1.3 Beneficios de las Herramientas CASE	7
1.4 Historia de las herramientas CASE	8
1.5 Características comunes de una herramienta CASE	16
1.6 Clasificaciones de las Herramientas CASE	18
1.6.1 Fases del ciclo del vida del desarrollo de sistemas	19
1.6.2 Su funcionalidad	24
1.6.3 Repositorio	27
1.6.4 Re-ingeniería	28
1.6.5 Soporte del ciclo de vida	28
1.6.6 Soporte de proyecto	29
1.6.7 Mejora continua de calidad	29
1.6.8 Categoría de las herramientas CASE	30
1.7 Principales Herramientas CASE	31
1.7.1 Herramientas de análisis y diseño	33
1.7.2 Generación de código y documentación	34
1.7.3 Herramientas de prueba	34
1.7.4 Herramientas de gestión de configuración	36
1.7.5 Herramientas de ingeniería inversa	36
1.8 Integración de las herramientas CASE	37
1.9 Implantación de las herramientas CASE	39
1.10 Causas del fracaso en la adopción de CASE	40

1.11 Herramientas CASE Cliente/Servidor	43
1.11.1 Arquitectura Cliente/Servidor	43
1.11.2 Evolución de la tecnología Cliente/Servidor	45
1.11.3 Descripción del panorama general del uso de herramientas CASE en aplicaciones Cliente/Servidor	46
1.11.4 Clasificación de las herramientas CASE Cliente/Servidor	50
 Capítulo 2: Descripción de algunas herramientas CASE	 51
2.1 ERwin	52
2.2 EasyCASE	54
2.3 Oracle	57
2.4 Oracle Designer	58
2.5 PowerBuilder de PowerSoft	61
2.6 PowerDesigner	62
2.7 System Architect	64
2.8 SNAP	66
2.9 Visual Basic	69
2.10 Visual FoxPro y Visual C++	69
2.11 Cool Stuf, de Sterling Software	70
2.12 Informix	70
2.13 Opal, de Computer Associates	71
 Capítulo 3: Evaluación de algunas herramientas CASE	 73
3.1 Características Generales	75
3.1.1 ERwin 3.0	75
3.1.2 ER/Studio 2.5	75
3.1.3 System Architect 4.0	75
3.1.4 Power Designer 6.1	76
3.2 Diagramas	76
3.2.1 ERwin 3.0	76
3.2.2 ER/Studio 2.5	77

3.2.3 System Architect 4.0	77
3.2.4 Power Designer 6.1	78
3.3 Ayuda	78
3.3.1 ERwin 3.0	78
3.3.2 ER/Studio 2.5	79
3.3.3 System Architect 4.0	79
3.3.4 Power Designer 6.1	80
3.4 Esquema de la Base de Datos	80
3.4.1 ERwin 3.0	80
3.4.2 ER/Studio 2.5	81
3.4.3 System Architect 4.0	81
3.4.4 Power Designer 6.1	82
3.5 Código que genera el lenguaje	83
3.5.1 ERwin 3.0	83
3.5.2 ER/Studio 2.5	84
3.5.3 System Architect 4.0	84
3.5.4 Power Designer 6.1	84
3.6 Ingeniería hacia delante y de reversa	85
3.6.1 ERwin 3.0	85
3.6.2 ER/Studio 2.5	87
3.6.3 System Architect 4.0	87
3.6.4 Power Designer 6.1	89
3.7 Sincronización de la base de datos	89
3.7.1 ERwin 3.0	89
3.7.2 ER/Studio 2.5	90
3.7.3 System Architect 4.0	91
3.7.4 Power Designer 6.1	92
3.8 Diseño	93
3.8.1 ERwin 3.0	93
3.8.2 ER/Studio 2.5	93
3.8.3 System Architect 4.0	93

3.8.4 Power Designer 6.1	94
3.9 Web	94
3.9.1 ERwin 3.0	95
3.9.2 ER/Studio 2.5	95
3.9.4 Power Designer 6.1	95
3.10 Metodología	95
3.10.1 ERwin 3.0	95
3.10.2 ER/Studio 2.5	95
3.10.3 System Architect 4.0	96
3.10.4 Power Designer 6.1	96
3.11 Reportes	96
3.11.1 ERwin 3.0	96
3.11.2 ER/Studio 2.5	96
3.11.3 System Architect 4.0	97
3.11.4 Power Designer 6.1	97
 Capítulo 4: Casos de uso	 99
4.1 Grupo Unión FENOSA México	100
4.2 Centro de Cómputo Académico de la UAEH en el área de Desarrollo de Software	102
4.3 IDS S.A de C.V	104
 Conclusiones	 107
Glosario de términos	115
Bibliografía	118
Referencias electrónicas	119

Introducción

En la actualidad la evolución del software, ha ayudado en gran medida a las empresas para poder realizar su trabajo.

Los desarrolladores de software también han buscado la forma de simplificar el análisis, mantenimiento y modificaciones del software.

A medida que se desarrollaba un nuevo software, ya se tenían en existencia millones de líneas de código, que necesitaban ser mantenidas y actualizadas por esta razón, se desarrollaron herramientas que facilitaran estas tareas.

Actualmente existen herramientas de ayuda como son las herramientas CASE, las cuales ayudan en la realización de cualquiera de las etapas del desarrollo del software.

En este trabajo se mostrará:

- **CAPITULO I:** Lo relacionado con las herramientas CASE, como es, la definición dada por diversos autores, beneficios y ventajas que tienen, los antecedentes históricos que les dieron origen, su evolución, la clasificación e integración, así como las herramientas que son de tipo cliente / servidor.
- **CAPITULO II:** Las características generales de las herramientas ErWin, EasyCASE, Oracle Designer, PowerDesigner, System Architect, SANP, Visual Basic, Visual FoxPro, Visual C++, Developer / 2000, Cool Stuf, que son herramientas CASE
- **CAPITULO III:** La evaluación de lo que son características generales, diagramación, ayuda, esquema de la base de datos, código que genera el

lenguaje, ingeniería hacia delante y de reversa, sincronización de la base de datos, diseño web, metodología y reportes que presentan las herramientas ERwin 3.0, ER/Studio 2.5, System Architect 4.0 y PowerDesigner 6.1

- **CAPITULO IV:** Se presenta el análisis de tres organizaciones mexicanas, que han hecho uso de las herramientas CASE, Grupo Unión Fenosa México, el Centro de Computo Académico de la UAEH e IDISA S.A de C.V.

Objetivo General

Documentar el origen, herramientas más comunes y aplicaciones de las herramientas CASE para poder aplicarlas durante el ciclo de vida del software.

Objetivos Particulares

- Mostrar el origen de las Herramientas CASE.
- Presentar sus aplicaciones en la industria del Software.
- Presentar los beneficios, ventajas y desventajas que tienen al hacer uso de las Herramientas CASE.
- Conocer características particulares de ErWin, EasyCASE, Oracle Designer, PowerDesigner, System Architect, SANP, Visual Basic, Visual FoxPro, Visual C++, Developer / 2000, Cool Stuf, que son herramientas CASE

Justificación

Este trabajo esta realizado para que los interesados en el desarrollo de software tengan un conocimiento sobre la existencia de herramientas, que pueden ayudarles con su trabajo en cualquiera de las fases del ciclo de vida del software, teniendo en cuenta de que para cada fase se pueden utilizar diversas herramientas dependiendo de qué se desee hacer.

Las herramientas CASE, son una tecnología que puede ayudar y en muchos casos facilitar el desarrollo del software.

Esto no quiere decir que cualquier persona que las utilice podrá considerarse un desarrollador de software, debido a que para hacer uso se debe tener conocimientos básicos como son, el análisis (por ejemplo la realización de diagramas), la codificación aunque es cierto que algunas de las herramientas generan código, no lo hacen a un 100% y todavía se verifican algunos posibles errores.

En México, la mayoría de los desarrolladores de software han hecho uso de estas herramientas, pero no las conocen como tales, por tal motivo no se les saca el mayor provecho posible.

Capítulo 1

Introducción a las herramientas CASE

En este capítulo se presenta la definición que manejan diversos autores a lo que son las herramientas CASE, los objetivos que tienen en general estas herramientas, así como los beneficios al hacer uso de ellas.

Se presentan los antecedentes históricos, que dieron origen a la creación de las herramientas CASE, una clasificación que pertenece a alguna de las fases del ciclo de vida del software, para poder identificar herramientas que podamos utilizar en esa fase.

De igual forma la integración y causas del fracaso de estas herramientas, así como herramientas para cliente/servidor.

1.1 Definiciones.

De acuerdo a Roger S. Pressman¹.

Hasta hace un tiempo, la ingeniería del software era fundamentalmente una actividad manual, en la cual las herramientas² se utilizaban solamente en las últimas fases del proceso.

Los ingenieros del software reconocen que necesitan herramientas más variadas, (las herramientas manuales no sirven para satisfacer las demandas de los sistemas basados en computadoras).

CASE puede tener diversas definiciones como:

- Son un complemento de la caja de herramientas del ingeniero del software que le proporciona, la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Al igual que las herramientas de ingeniería y diseño, asistidos por computadora que utilizan de otras disciplinas, las herramientas CASE ayudan a asegurar que la calidad sea algo diseñado antes de llegar a construir el producto.
- De acuerdo con Efraim Turban, Mc Clean y Ephraim³.
 - CASE (Computer-Aided Systems Engineering) cuyo significado en español es Ingeniería del Software Asistida por Computadora, es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias del desarrollo de sistemas y al igual que las

¹ [S. Pressman, 1998]

² En muchos casos, las únicas herramientas disponibles para el ingeniero del software eran compiladores y editores de texto. Estas herramientas nada mas abarcaban la codificación, una actividad que no debería de ocupar más del 20% del proceso global del software.

³ [James, 2001]

herramientas CAD (Diseño Asistido por Computadora) o CAM (Manufactura Asistida por Computadora) su objetivo es, acelerar el proceso para el que han sido diseñadas, en el caso de CASE para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas (ver Figura 1.1).

- Henry David Crockett de Portland State University, "Las herramientas CASE se ven simplemente como herramientas, que cualquiera puede escoger y utilizar para desarrollar un sistema de información, su selección e implementación casi siempre llevará a una reducida productividad y calidad."

La selección e implementación de herramientas CASE, son un proceso de múltiples etapas que permiten errores fatales en cada una. Uno de los errores más comunes es escoger una herramienta CASE que apoye un método desconocido para los diseñadores".

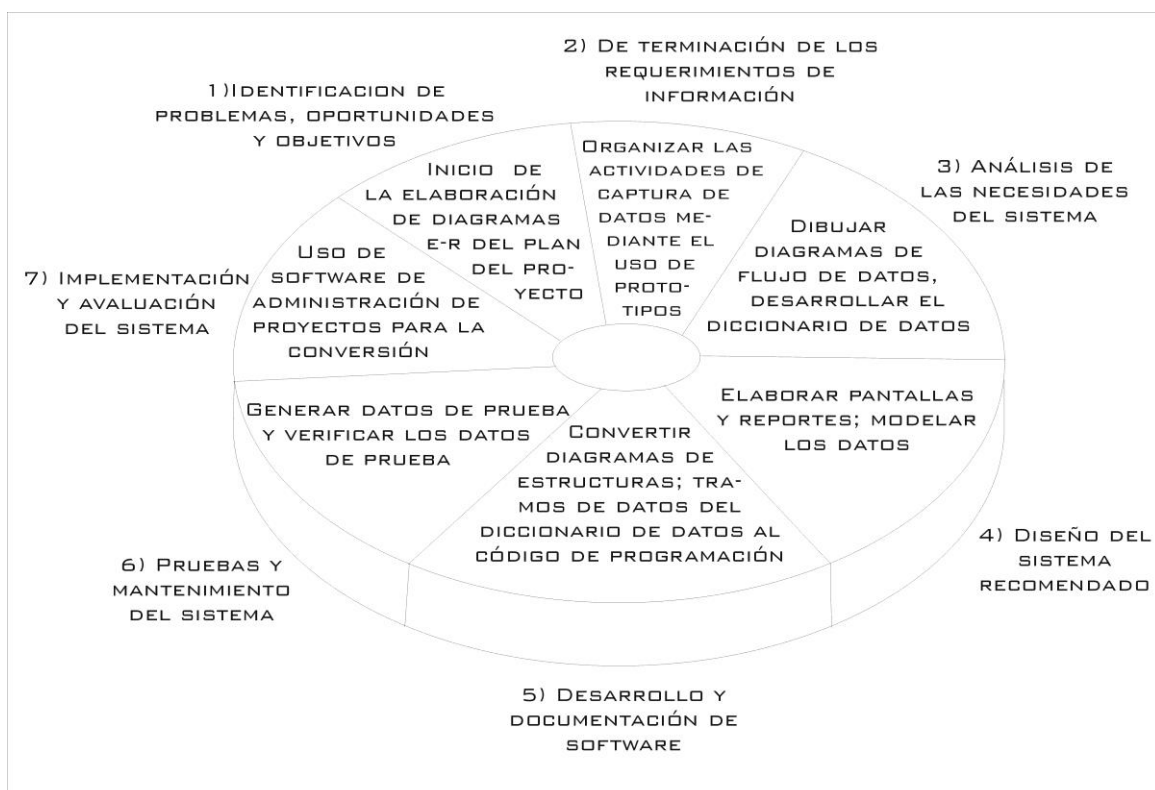


Figura 1.1: Ciclo de vida del desarrollo del software.

- Alan Chimura de CASE Associates, "Las herramientas CASE incluyen, manejadores, métodos, técnicas, disciplina e instrucciones, todos trabajando juntos."
- Bajo el nombre de herramientas CASE se agrupan una serie de productos destinados a la automatización de la producción del software, es decir, al desarrollo asistido de sistemas informáticos [López-Fuensalida, 1990].
- Las herramientas CASE abarcan cada etapa del proceso de ingeniería y cada actividad que se desarrolla a lo largo del mismo. CASE está formado por un conjunto de bloques, que comienzan en el nivel del hardware y del sistema operativo y acaban en cada una de las herramientas.
- CASE se refiere a herramientas para el desarrollo de sistemas que constan de cinco componentes:
 - Herramientas de diagramación.
 - Depósito de información.
 - Generadores de interfaces.
 - Generadores de código.
 - Herramientas de administración.

Las herramientas CASE, hacen hincapié en las actividades de alto nivel, aunque el objetivo a largo plazo es abarcar las actividades de análisis, diseño y desarrollo.

- De acuerdo con el Instituto Nacional de Estadística e Informática⁴ son:

⁴ [Sub – Jefatura, 2006].

- Conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores.
 - Conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.
 - Filosofía de desarrollo de software, que ayuda a automatizar el ciclo de vida del desarrollo de los sistemas.
 - Unión de las herramientas automáticas de software, y las metodologías de desarrollo de software formales.
 - Es el uso de la ayuda computarizada en el proceso del desarrollo del software.
-
- La tecnología CASE corresponde a la ingeniería del software asistido por computadora. Es un conjunto de herramientas de software para automatizar las tareas del desarrollo del software. La idea es proporcionar un conjunto integrado de herramientas, que enlazan y automatizan todas las fases del ciclo de vida de software y su administración. Es una combinación de herramientas, software y metodología enfocada sobre el problema de producción de software [Cuevas, 1991].
 - La tecnología CASE es la automatización del desarrollo de Software, propone un nuevo concepto de ciclo de vida del software basado en la automatización. La idea básica, es proporcionar un conjunto de herramientas, bien integradas que ahorren trabajo, enlazando y automatizando todas las fases del ciclo de vida del software, facilita el seguimiento de las metodologías a nivel detallado al automatizar el dibujo de los diagramas y la generación de la documentación del sistema (ver Figura 1.2).

- La tecnología CASE es una combinación de herramientas de software y de metodologías, más aun, CASE es diferente de las primitivas tecnologías del software, por que se centra en el problema de la productividad del software y no solamente en la implantación de soluciones [McClure, 1992].

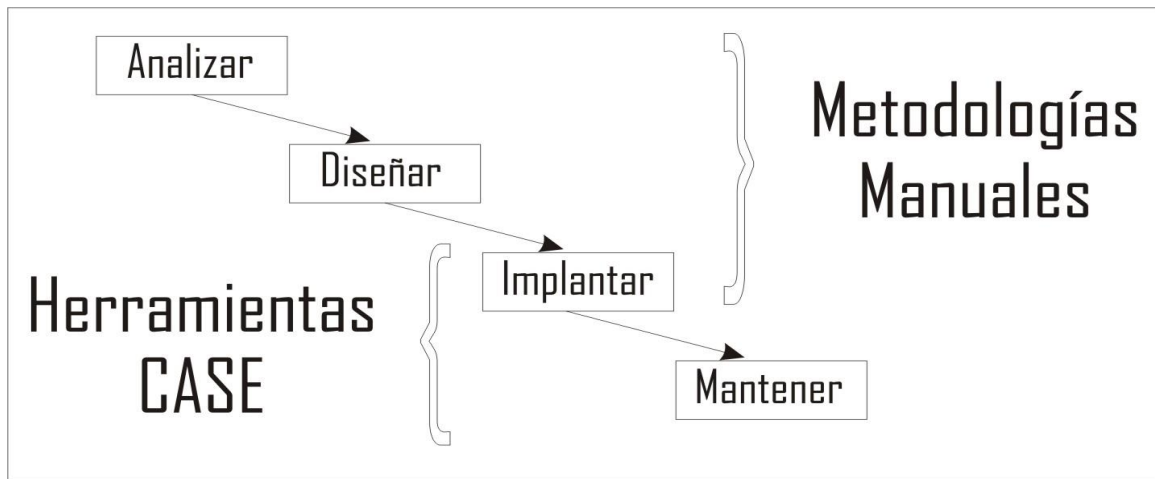


Figura 1.2: Las tecnologías tradicionales de software son de 2 tipos: manuales y con herramientas aisladas. La tecnología CASE es una combinación de herramientas y metodologías totalmente integradas, con especial énfasis en la automatización del ciclo de vida del software.

1.2 Objetivos de las Herramientas CASE.

- De acuerdo con Carma McClure⁵
 - Proporcionar un entorno de desarrollo interactivo con un tiempo de respuesta rápido, recursos dedicados y una comprobación de errores desde el principio
 - La automatización de muchas tareas de desarrollo y mantenimiento de software.

⁵ [McClure, 1992]

- Una programación visual proporcionada por potentes interfaces gráficos.
- De acuerdo con Roger S. Pressman⁶
 - Aumentar la productividad en el desarrollo.
 - Dar calidad a los productos desarrollados.
 - Reducir el costo del software.
 - Automatizar los chequeos de errores.
 - Acelerar el desarrollo de las aplicaciones.
 - Automatizar tareas de desarrollo.
 - Automatizar la generación de documentación.
 - Dar portabilidad al software.
 - Implantar metodologías de desarrollo.
 - Datos reutilizables y compartidos.
 - Administrar el proyecto.
 - Ingeniería hacia atrás (reingeniería).

1.3 Beneficios de las Herramientas CASE⁷.

- Permite las técnicas estructuradas.
- Impone las ingenierías del software y de la información.
- Aumenta la calidad del software mediante comprobación automática.
- Favorece la realización de prototipos.

⁶ [S. Pressman, 1998]

⁷ [McClure, 1992]

- Simplifica el mantenimiento del programa.
- Acelera el proceso de desarrollo.
- Libera al profesional de la principal parte creativa en el desarrollo del software.
- Anima al desarrollo evolucionado y gradual.
- Posibilita la reutilización de los componentes del software.

1.4 Historia de las herramientas CASE.

Desde que a finales de los años sesentas se acuña el término "Crisis del Software" (Es el hecho de que el software que se construye, no solamente no satisface los requerimientos, ni las necesidades pedidos por el cliente, sino que además excede los presupuestos y los horarios de tiempos), numerosos expertos han venido ocupándose del tema, proponiendo distintas técnicas, metodologías y herramientas para manejar esta situación [Piattini, 2004].

En los años ochentas, surgieron varios factores que favorecieron el estado actual del mercado, para las herramientas de desarrollo asistido de sistemas informáticos.

Por una parte, el acercamiento de la información a un número de usuarios, ha provocado cambios organizativos importantes en el mundo laboral, la oferta cada vez más diversa permite a los usuarios poder elegir un entorno apropiado a sus necesidades.

Este hecho provocó una gran demanda de profesionales informáticos, que el mundo académico no era capaz de satisfacer, a esto hay que añadir una cultura informática cada vez mayor en los usuarios, hace que su interés por resolver sus problemas a través de la computadora sea muy grande.

Este factor produjo en las empresas una cartera de pedidos y aplicaciones que el departamento de informática no es capaz de desarrollar.

Por otro lado, una forma generalmente anárquica de desarrollar sistemas, ha producido unas aplicaciones en explotación de baja calidad que absorben entre un 60% y un 80% del tiempo de los informáticos en activo mantenimiento.

Ante esta situación caótica, en la década de los ochentas, el mundo informático reaccionó para tratar de solucionar el problema [López-Fuensalida, 1990].

A mediados de esta década se llevo a una nueva industria llamada CASE (siglas que significan Computer – Aided Software Engineering: Ingeniería de Software Asistida por Computadora) [Yourdon, 1993].

Estas herramientas representan, parte de las primeras herramientas de desarrollo de software, basadas en computadoras personales y los primeros intentos para automatizar el análisis y el diseño de tareas.

Las primitivas herramientas CASE, se dirigieron principalmente a la automatización de la documentación y la comunicación como una mejora. Clave de la productividad del software [McClure, 1992].

La primera herramienta comercial aparece en 1982, aunque algunos especialistas indican que, algunos ejemplos de herramientas para diagramación ya existían.

No fue hasta 1985 en que las herramientas CASE, se volvieron realmente importantes en el proceso de desarrollo de software. Los proveedores prometieron

a la Industria que muchas actividades serían beneficiadas⁸ con la ayuda de las herramientas CASE. [Sub – Jefatura, 2006]

A finales de los años ochentas se produce un considerable aumento en la venta de las herramientas CASE y empieza la etapa de asimilación de la tecnología, que fracasa, debido a las limitaciones de la "Primera Generación" de productos, las falsas expectativas sobre sus posibilidades y su incorrecta implantación.

Además de la resistencia al cambio que habitualmente se produce, cuando aparece un nuevo paradigma que obliga a realizar un importante esfuerzo de formación y, especialmente en adaptación de la mentalidad de los usuarios.

La tecnología CASE ha experimentado la clásica evolución que sufren aquellos paradigmas⁹, que se ofrecen como la solución universal capaz de resolver todos los problemas del desarrollo de sistemas de información.

A mediados de los noventas surge una "Segunda Generación" de herramientas, (algunas de las cuales ya no aparecen bajo el término CASE, para no recordar el fracaso anterior), que superan gran parte de las limitaciones existentes en la primera generación.

La tecnología CASE supone la "automatización del desarrollo del software", contribuyendo con esto, a elevar la productividad y la calidad en el desarrollo de sistemas de información.

Este enfoque persigue los siguientes objetivos a la hora de construir software:

- Permitir la aplicación práctica de metodologías estructuradas.
- Mejorar la calidad del software.

⁸ Estos beneficios consistían, por ejemplo, en el aumento en la productividad.

⁹ Técnicas estructuradas, inteligencia artificial, lenguajes de cuarta generación y, la orientación a objetos.

- Facilitar la realización de prototipos, y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo visual de las aplicaciones, mediante la utilización de gráficos.

De una manera muy esquemática, se puede afirmar que una herramienta CASE se compone de los siguientes elementos (ver Figura 1.3) [Piattini, 2004].

- Repositorio o Depósito centralizado de información.

Es un punto clave para una alta productividad, permite obtener información para los realizadores cuando se necesita y directamente utilizable [Cuevas, 1991].

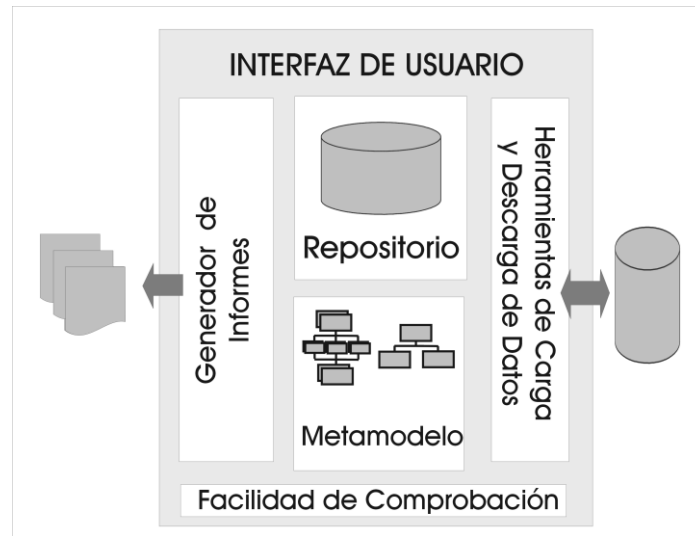


Figura 1.3: Componentes de una herramienta CASE

Es el diccionario donde:

- Almacenan los elementos definidos o creados por la herramienta, y que se basa en un SGBD (Sistema Gestor de Bases de Datos) o en un sistema de gestión de ficheros. [Piattini, 2004].
- Contiene detalles sobre los componentes del sistema, tales como datos, flujo de datos y procesos.
- Incluye información que describe el volumen y frecuencia de cada una de las actividades [James, 2001].
- Almacena los tipos de información del sistema, la interrelación entre los componentes de información, las reglas para usar o procesar los componentes [Cuevas, 1991].

Aunque los diccionarios son diseñados para que el acceso a la información sea de manera sencilla, también incluyen controles y medidas de protección que preservan la exactitud y consistencia de los detalles del sistema como:

- Niveles de autorización.
- Validación de procesos.
- Procedimientos para verificar la consistencia de las descripciones.

Asegurando que el acceso a las definiciones y las revisiones hechas ocurran en forma apropiada y acorde con procedimientos ya establecidos.

El diccionario suele guardar los siguientes tipos de información [James, 2001]:

- Registro y elementos: Detalles de elementos, datos y registros.
- Datos: Detalles relacionados con almacenes y flujos de datos, etc. También se incluyen tablas de códigos definidos por el usuario junto con su significado así como el uso de otros nombres (alias).
- Procesos: Procesos, funciones y módulos del sistema.
- Gráficas: Diagramas de flujo de datos, gráficas estructuradas, diagramas para modelos de datos, diagramas estructurados, diagramas de relación entre entidades y gráficas de presentación.
- Pantallas e informes: Definiciones y composición del diseño de los informes, diseños de pantallas y formas para la entrada de datos. También es posible preparar informes relacionados con la entrada de datos en pantalla.
- Entidades de otro tipo: Informes de las especificaciones contenidas en el diccionario, listados de entidades, nombres de usuarios y descripciones de documentos.

- Metamodelo o Herramientas de administración (no siempre visible).
 - Define las técnicas y metodologías soportadas por la herramienta CASE [Piattini, 2004].
 - Ayudan a los gerentes de proyecto a mantener la efectividad y eficiencia de todo el proceso de desarrollo de una aplicación.
 - Ayuda a los gerentes de desarrollo a calendarizar las actividades de análisis y diseño así como la asignación de recursos a las diferentes actividades del proyecto [James, 2001].
- Generador de informes o Herramientas para diagramación.
 - Da soporte al análisis y documentación de los requerimientos de una aplicación [James, 2001].
 - Permite obtener toda la documentación que describe el sistema de información desarrollado (Documentación que está asociada a las técnicas y metodologías). [Piattini, 2004]
 - Ofrece la capacidad de dibujar diagramas y cartas, además de guardar los detalles en forma interna.
 - Cuando es necesario realizar cambios, la naturaleza de estos se describe en el sistema, el cual puede entonces volver a dibujar todo el diagrama de manera automática. La capacidad para cambiar y volver a dibujar elimina una actividad que los analistas encuentran tediosa y poco deseable [James, 2001].
- Herramienta de carga/descarga de datos o Generadores de código.

- Permite cargar el repositorio de la herramienta CASE con datos provenientes de otros sistemas. [Piattini, 2004]
- Automatiza la preparación de software.
- Incorpora métodos que permiten convertir las especificaciones del sistema en código ejecutable.

La generación de código no ha sido perfeccionada, estos generadores producen aproximadamente el 75% del código fuente de una aplicación. El resto debe ser escrito por los programadores, a lo cual se le conoce como codificación manual.

Los mayores beneficios se obtienen cuando los generadores de código se encuentran integrados con un depósito central de información. Debido a que la información alcanza el objetivo de crear un código que pueda volverse a emplear. Cuando las especificaciones cambian, se puede volver a generar el código al alimentar los detalles del diccionario de datos a través del generador de código. El contenido del diccionario puede emplearse de nuevo para preparar el código ejecutable [James, 2001].

- Interfaz de usuario o Generador de interfaces.

Son los medios que permiten a los usuarios [James, 2001]:

- Interactuar con una aplicación ya sea para dar entrada a información y datos o para recibir información.
- Ofrece la capacidad para preparar imitaciones y prototipos para las interfaces con los usuarios.

- Soportan la rápida creación de menús de demostración para el sistema, de pantallas de presentación y formato de los informes.
- Incluyen de editores de texto y herramientas de diseño gráficos, que permitan mediante la utilización de un sistema de ventanas, iconos y menús, con la ayuda del ratón, definir los diagramas, matrices, etc. que incluyen las distintas metodologías.
- Comprueba los errores, facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta.

1.5 Características comunes de una herramienta CASE¹⁰.

- Operaciones iniciales.

Los sistemas CASE almacenan información por proyecto, cada aplicación es considerada como un proyecto. La información que describe cada proyecto se mantiene por separado de la de otros.

- Menú principal de funciones.

Muchas herramientas CASE permiten que el usuario seleccione una acción señalando su nombre o un número sobre la pantalla, ya sea a través de un dispositivo apuntador o por el posicionamiento de una barra luminosa por medio de las teclas de flechas y tabulador contenidas en el teclado.

- Dibujo de diagramas de flujo de datos.

¹⁰ [James, 2001]

Los diagramas de flujo de datos son uno de muchos tipos de diagramas y cartas disponibles en las herramientas CASE, es muy sencillo modificar los diagramas.

- Diccionario por proyecto.

A medida que se formulan las especificaciones y la documentación, toda la información con respecto al proyecto se acumula en el diccionario de datos. Parte de la información, la graba directamente la persona que hace uso de la herramienta, otra parte es grabada automáticamente.

Una vez que la información se encuentra en el diccionario, puede volver a ser utilizada por el mismo proyecto en forma repetida, sin necesidad de definirla de nuevo.

Dentro del diccionario, las entradas se pueden añadir, modificar, listar, borrar y cambiar nombre. También es posible enlistar el contenido del diccionario con informes preformateados. Se tiene acceso a la información contenida en el diccionario desde cualquier parte.

- Pantallas e informes.

Varias herramientas CASE, proporcionan un método rápido y sencillo para desarrollar prototipos de pantallas para que los usuarios finales trabajen con ellas. El analista puede diseñar y ejecutar pantallas y reportes con el apoyo de un menú, definir la distribución de una pantalla o reporte, el analista puede generar un reporte basándose en datos de prueba proporcionados al sistema

- Herramientas para análisis y documentación.

Algunas herramientas CASE ofrecen características tales como un conjunto de reportes que validan las descripciones del sistema.

- Utilerías.

La información utilizada por el sistema se encuentra descrita por las funciones de utilería, estas funciones permiten definir las contraseñas de los usuarios, los privilegios de acceso y los procedimientos de respaldo. Las utilerías también proporcionan funciones de respaldo y recuperación. Con ellas es posible copiar o volver a crear una parte o todo el diccionario del proyecto a partir de una copia de respaldo

1.6 Clasificaciones de las Herramientas CASE¹¹.

Las Herramientas CASE fueron desarrolladas para automatizar la organización de las tareas y que sean completadas en forma correcta, eficiente y facilitar las tareas coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software.

La mejor razón para la creación de estas herramientas es el incremento en la velocidad de desarrollo de los sistemas. Por esto, las compañías pueden desarrollar sistemas sin encarar el problema de tener cambios en las necesidades del negocio, antes de finalizar el proceso de desarrollo.

También les permite competir más efectivamente usando estos sistemas desarrollados nuevamente para compararlos con sus necesidades de negocio actuales. En un mercado altamente competitivo, esto puede hacer la diferencia entre el éxito y el fracaso. Permiten a los analistas tener más tiempo para el análisis y diseño, y minimizar el tiempo para codificar y probar.

¹¹ [James, 2001]

La introducción de CASE integradas tiene un impacto significativo en los negocios y sistemas de información de las organizaciones. Con un CASE integrado, las organizaciones pueden desarrollar rápidamente sistemas de mejor calidad para soportar procesos críticos del negocio y asistir en el desarrollo y promoción intensiva de la información de productos y servicios.

Estas herramientas pueden proveer muchos beneficios en todas las etapas del proceso de desarrollo de software, algunas de ellas son:

- Verificar el uso de todos los elementos en el sistema diseñado.
- Automatizar el dibujo de diagramas.
- Ayudar en la documentación del sistema.
- Ayudar en la creación de relaciones en la Base de Datos.
- Generar estructuras de código.

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada.

Estas podrían clasificarse según:

1.6.1 Fases del ciclo de vida del desarrollo de sistemas¹².

- Herramientas integradas, I-CASE (Integrated CASE, CASE integrado):

¹² [Sub – Jefatura, 2006]

Son herramientas computacionales que apoyan cada una de las distintas etapas del ciclo de desarrollo de un sistema de información. Son llamadas también CASE workbench.

Generan gráficamente la mayoría de los conceptos relacionados a cada etapa del ciclo de desarrollo (Análisis, diseño, construcción, pruebas) (ver Figura 1.4).

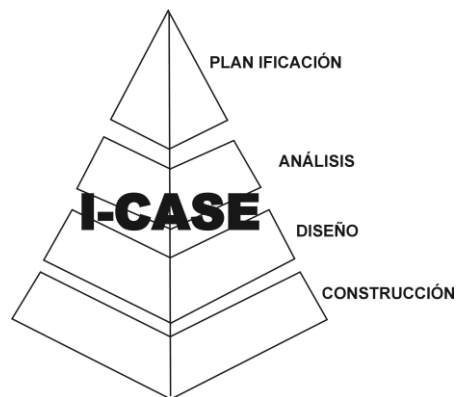


Figura 1.4: Conjunto de herramientas I-CASE

Estas herramientas se aplican a todas las etapas en forma integral, es decir, si se modifica algo del análisis automáticamente se afectará el diseño y el código.

Un entorno I-CASE debería cumplir los siguientes requisitos:

- Proporcionar un mecanismo para compartir la información de ingeniería del software entre todas las herramientas contenidas en el entorno.
- Hacen posible que un cambio en un elemento de información se siga hasta los demás elementos relacionados.
- Proporcionar un control de versiones y una gestión de configuración general para toda la información de la ingeniería del software.

- Permitir un acceso directo y no secuencial a cualquiera de las herramientas contenidas en el entorno.
- Establecer un entorno automatizado para un contexto de procedimientos.
- Capacitar a los usuarios de cada una de las herramientas para experimentar una utilización consistente en la interfaz hombre-máquina.
- Permitir la comunicación entre ingenieros del software.
- Recoger métricas de gestión y técnicas para mejorar el proceso y el producto.

Ventajas:

- Integran el ciclo de vida.
- Permiten lograr importantes mejoras de productividad a mediano plazo.
- Permiten un eficiente soporte al mantenimiento de sistemas.
- Mantienen la consistencia de los sistemas a nivel corporativo.
- Dan énfasis en el modelamiento empresarial.

- Dan especificaciones basadas en diagramas automatizados y perfectamente integrados a través de las diferentes etapas de desarrollo.
- Generan automáticamente códigos totalmente integrados con la estación de trabajo de diseño, representan gráficamente las estructuras de códigos.
- Generan automáticamente la documentación.
- Dan mantenimiento de especificaciones de diseño y regeneración de códigos.
- Hacen uso de una enciclopedia coordinadora de conocimientos para concentrar toda la información de las diferentes estaciones de trabajo en forma integrada.

Desventajas:

- No son tan eficientes para soluciones simples, sino para soluciones complejas.
 - Dependen del hardware y del software
 - Son costosas.
-
- Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior) o front-end.

Estas herramientas son orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.

Por ejemplo, las herramientas de modelado de diagramas entidad-relación y diagramas UML.

Ventajas:

- Se utilizan en plataforma PC, es aplicable a diferentes entornos.
- Tienen menor costo.

Desventajas:

- Permiten mejorar la calidad de los sistemas, pero no mejora la productividad.
 - No permiten la integración del ciclo de vida.
- Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior) o back-end.

Estas herramientas son dirigidas a las últimas fases del desarrollo: construcción e implantación.

Ventajas:

- Permiten lograr importantes mejoras de productividad a corto plazo.
- Permiten un eficiente soporte al mantenimiento de sistemas.

Desventajas:

- No garantizan la consistencia de los resultados a nivel corporativo.
- No garantizan la eficiencia del análisis y diseño.
- No permiten la integración del ciclo de vida.
- Juegos de herramientas o Tools-CASE

Estas son el tipo más simple, las que están formadas por ese conjunto de herramientas, orientadas cada una de ellas a resolver una determinada fase del desarrollo.

Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

En el mercado actual se pueden citar entre otras, algunas herramientas distribuidas por fases de desarrollo como:

- Análisis y Diseño.
 - Prokit Workbench de McDonnell-Douglas.
 - Desing Aid Nastec.
 - Analyst/Designer Toolkit de Yourdon.
 - Excelerator de Index Technology.
 - Pose de Computer System Advisers.
- Diseño de Archivos y Base de Datos.
 - Chen Toolkit de Chen & Associates.
 - IDMS/Architec de Cullinet Software.
 - Autmate Plus de LBMS.
 - Case Designer de Oracle.
 - Synon.

- Oracle.
- Informix.
- Programación.
 - APS de Sage Software.
 - Transforms de Transform Logia.
 - Telon Pansophic System.
 - Decase de DEC COBOL 2/ Worbench de Micro Focus.
 - CASE para AS/400.

1.6.2 Su funcionalidad.

- Herramientas de planificación de sistemas de gestión.

La función principal es para modelar los requisitos de información estratégica de una organización. Proporcionan un "metamodelo" del cual se pueden obtener sistemas de información específicos.

Su objetivo principal es ayudar a comprender de mejor forma cómo se mueve la información entre las distintas unidades organizativas. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas actuales no satisfacen las necesidades de la organización.

- Herramientas de análisis y diseño.

Estas permiten al desarrollador crear un modelo del sistema que se va a construir y también la evaluación de la validez y consistencia de este

modelo. Proporcionan un grado de confianza en la representación del análisis y ayudan a eliminar errores con anticipación. Algunas de ellas son:

- De análisis y diseño (Modelamiento).
 - De creación de prototipos y de simulación.
 - Para el diseño y desarrollo de interfaces.
 - Máquinas de análisis y diseño (Modelamiento).
- Herramientas de programación.

Se engloban aquí los compiladores, los editores y los depuradores de los lenguajes de programación convencionales. Algunos ejemplos de estas herramientas son:

- De codificación convencional.
 - De codificación de cuarta generación.
 - De programación orientada a los objetos.
- Herramientas de integración y prueba.

Sirven de ayuda a la adquisición, medición, simulación y prueba de los equipos lógicos desarrollados. Entre las más utilizadas están:

- De análisis estático.
 - De codificación de cuarta generación.
 - De programación orientada a los objetos.
- Herramientas de gestión de prototipos.

Los prototipos son utilizados ampliamente en el desarrollo de aplicaciones, para la evaluación de especificaciones de un sistema de información, o para

un mejor entendimiento de cómo los requisitos de un sistema de información se ajustan a los objetivos perseguidos.

- Herramientas de mantenimiento:

La categoría de herramientas de mantenimiento se pueden subdividir en:

- Herramientas de ingeniería inversa.
 - Herramientas de reestructuración y análisis de código.
 - Herramientas de reingeniería.
- Herramientas de gestión de proyectos.

La mayoría de las herramientas CASE de gestión de proyectos se centran en un elemento específico, en lugar de proporcionar un soporte global para la actividad.

Utilizando un conjunto seleccionado de las mismas se puede:

- Realizar estimaciones de esfuerzo.
- Costo y duración.
- Hacer un seguimiento continuo del proyecto.
- Estimar la productividad y la calidad, etc.

Existen también herramientas que permiten al comprador del desarrollo de un sistema, hacer un seguimiento que va desde los requisitos del pliego de prescripciones técnicas inicial, hasta el trabajo de desarrollo que convierte estos requisitos en un producto final. Se incluyen dentro de las herramientas de control de proyectos las siguientes:

- Herramientas de planificación de proyectos.

- Herramientas de seguimiento de requisitos.
 - Herramientas de gestión y medida.
- Herramientas de soporte.

Se engloban en esta categoría las herramientas que recogen las actividades aplicables en todo el proceso de desarrollo, como las que se relacionan a continuación:

- Herramientas de documentación.
- Herramientas para software de sistemas.
- Herramientas de control de calidad.
- Herramientas de bases de datos.

1.6.3 Repositorio.

Funcionan en torno a un repositorio central, siendo éste el núcleo fundamental que contiene todas las definiciones de objeto y sus relaciones.

Los objetos pueden ser especificaciones del sistema en forma de diagramas de flujo de datos, diagramas entidad-relación, esquemas de bases de datos, diseños de pantallas.

El repositorio es un concepto más amplio que el de diccionario de datos y soporta a los demás grupos de funciones.

1.6.4 Re-ingeniería.

Los sistemas CASE permiten establecer una relación estrecha y fuertemente formalizable entre los productos generados a lo largo de distintas fases del ciclo

de vida, permitiendo actuar en el sentido especificaciones-código (ingeniería "directa") y también en el contrario (ingeniería "inversa"). Ello facilita la realización de modificaciones en la fase más adecuada en cada caso y su traslado a las demás. Al conjunto de facilidades proporcionadas por la ingeniería "directa" e "inversa" se le denomina "re-ingeniería".

1.6.5 Soporte del ciclo de vida.

El ciclo de vida de una aplicación o de un sistema de información se compone de varias etapas, que van desde la planificación de su desarrollo hasta su implantación, mantenimiento y actualización.

Los sistemas CASE pueden cubrir la totalidad de las fases del ciclo de vida del software o bien especializarse en algunas de ellas.

En este último caso se pueden distinguir sistemas de "alto nivel" ("Upper CASE"), orientados a la autonomía y soporte de las actividades correspondientes a las dos primeras fases y, sistemas de "bajo nivel" ("Lower CASE"), dirigidos hacia las dos últimas. Los sistemas de "alto nivel" pueden soportar un número más o menos amplio de metodologías de desarrollo.

1.6.6 Soporte de proyecto.

Este tipo de funciones hace referencia al soporte de actividades que se producen durante el desarrollo, derivadas fundamentalmente del trabajo en grupos, tales como:

- Facilidades de comunicación.
- Soporte a la creación.

- Modificación e intercambio de documentación.
- Herramientas personales.
- Controles de seguridad, etc.

Los sistemas CASE pueden conceder a estas cuestiones una importancia variable por lo cual el soporte de proyecto constituye un factor de diferenciación.

1.6.7 Mejora continua de calidad.

Aunque frecuentemente se asocia a los sistemas CASE con la mejora de la productividad en el desarrollo de aplicaciones, debe tenerse en cuenta que una de las principales ventajas, consta también en la mejora de la calidad de los desarrollos realizados.

Determinados sistemas CASE introducen herramientas que permiten ejercer un control intenso de garantía de calidad del software desarrollado desde las primeras fases de su ciclo de vida. [Sub – Jefatura, 2006]

1.6.8 Categoría de las herramientas CASE

Estas se pueden clasificar de la siguiente forma (ver Figura 1.13).

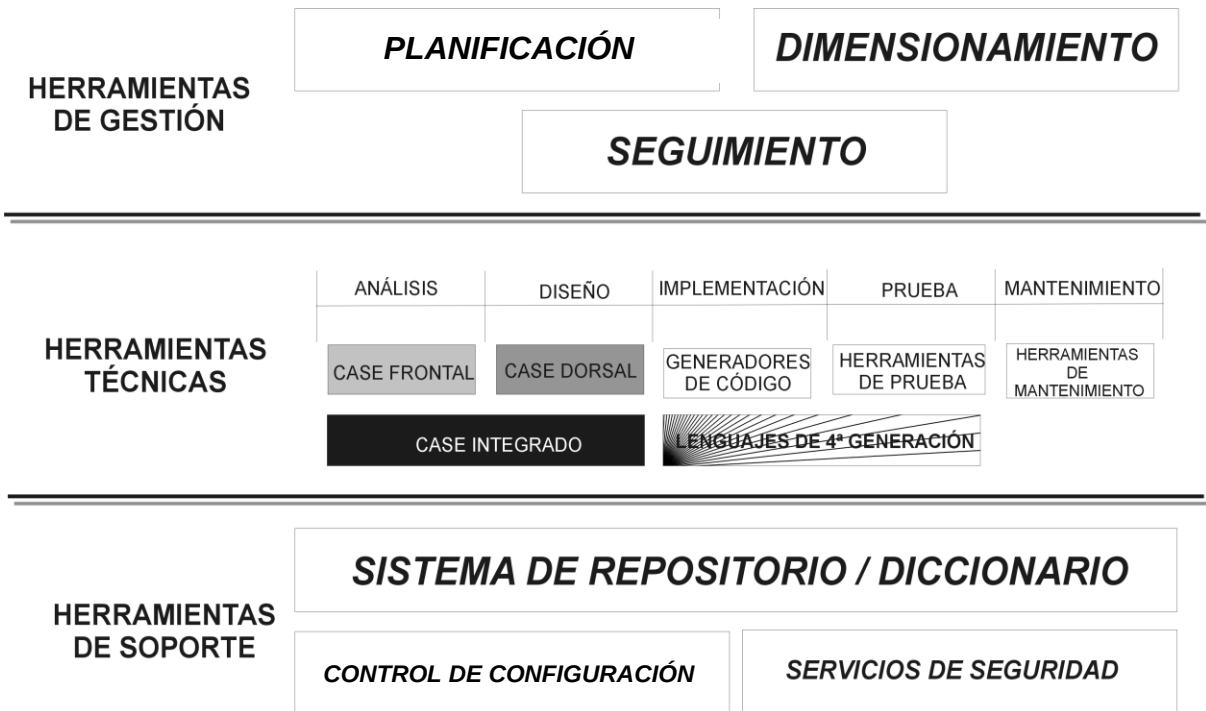


Figura 1.13: Categoría de las herramientas CASE

- **Herramientas de Gestión:**

Estas son encargadas de la estimación, planificación y seguimiento del proyecto.

- **Herramientas Técnicas:**

Estas se dividen tradicionalmente en:

- CASE frontales (front-end) o Superiores (Upper CASE).
- CASE dorsales (back-end) o Inferiores (Lower CASE).

- **Herramientas de Soporte.**

Sirven para tener en buen funcionamiento, los sistemas en cualquier etapa del ciclo de vida del software

1.7 Principales Herramientas CASE.

- Herramientas de análisis y diseño.
 - Crea diagramas de flujo de datos, entidad-relación, etc.
 - Genéricas o específicas para metodologías de diseño de sistemas propietarios.
- Manejadores de código o aplicaciones.
 - Algunas convierten directamente las especificaciones en código.
 - A menudo tienen funciones de arrastrar y soltar para crear aplicaciones e interfaces.
- Herramientas para elaborar prototipos.
 - Generadores de pantallas y menús.
 - Generadores de informes / 4GL.
- Apoyo a lenguajes de programación.
 - Plantillas para secuencias de códigos comunes en lenguajes específicos.
 - Bibliotecas de subrutinas para funciones comunes.
- Herramientas para pruebas.
 - Producen datos para los programas de prueba.
 - Vigilan la ejecución de programas.
 - Verifican los diagramas del análisis del sistema para integridad y consistencia.

- Herramientas para el seguimiento del problema.
 - Identificar la responsabilidad para corregir errores y seguir el progreso en la solución de los mismos.
- Herramientas para la administración de los cambios / control de versión.
 - Depósito de diferentes versiones de código, con funciones de "verificación de salida" y "verificación de entrada".
 - Permite el acceso solo al personal autorizado.
 - Mantienen información acerca de los cambios entre versiones de los programas.
- Herramientas para la administración de proyectos.
 - Método de la trayectoria crítica (diagramas PERT).
 - Diagramas de Gantt.
 - Seguimiento de tiempo y gastos.
- Herramientas para estimación.
 - Estiman los requerimientos y costos de personal para proyectos de desarrollo de sistemas.
- Generadores de documentación.
 - Crean diagramas de flujo, otra documentación para sistemas con documentación pobre o inexistente.
- Herramientas de ingeniería inversa.

- Ayudan a reestructurar el código en sistemas heredados.
- Herramientas de reingeniería de procesos de negocios.
 - Analizan y mejoran los procesos existentes.
 - Diseñan nuevos procesos.

1.7.1 Herramientas de análisis y diseño.

Las herramientas de análisis y diseño, son la parte más importante de las herramientas CASE, en cuanto a número de licencias vendidas y de fabricantes, su principal objetivo es, ayudar a la definición de requisitos del sistema y sus propiedades.

Dentro de esta categoría destacan las herramientas que permiten crear y modificar:

- Diagramas E/R,
- Diagramas de flujo de datos,
- Diagramas de estructura de cuadros,
- Diagramas de clases, etc.

También son muy importantes las herramientas de prototipado como:

- Los diseñadores de pantallas,
- Generadores de menús,

- Generadores de informes y lenguajes de especificación ejecutables.

Un aspecto importante es, la capacidad de análisis y verificación de especificaciones que soporta la herramienta, no sólo sintáctica sino también semántica, como por ejemplo, la capacidad de normalizar un diagrama de datos.

Todas estas herramientas operan, sobre un repositorio donde se va almacenando la información necesaria para el funcionamiento de la misma herramienta.

1.7.2 Generación de código y documentación.

A partir de las especificaciones del diseño se puede generar código (por ejemplo, en C o COBOL) como los esquemas de bases de datos (sentencias de definición en SQL) convenientes.

Actualmente las herramientas CASE, ofrecen interfaces con diversos lenguajes de cuarta generación para la construcción de sistemas de manera rápida.

Las herramientas CASE, también soportan la creación automatizada de un conjunto muy variado de documentación, que va desde la descripción textual de un pseudocódigo, hasta diagramas más o menos complejos.

1.7.3 Herramientas de prueba.

Las herramientas de prueba se conocen también por las siglas CAST (Computer Aided Software Testing), y es un área bastante reciente dentro de la tecnología CASE.

En Poston y Sexton (1992) se presentan algunas funcionalidades que suelen tener este tipo de herramientas:

- Gestión de pruebas:
 - Predecir costo y tiempo de las pruebas.
 - Planificar pruebas de productos, actividades y recursos
 - Monitorizar pruebas.
 - Generar informes.
- Definir requisitos y objetivos de prueba:
 - Registrar requisitos y/u objetivos.
 - Verificar requisitos.
- Diseñar pruebas:
 - Diseñar, generar y documentar los casos de prueba.
 - "Rastrear" (tracing) requisitos.
 - Trazar caminos, ramas y sentencias de casos de prueba.
 - Predecir la salida esperada.
- Construir entornos de ejecución de pruebas:
 - Verificar código.
 - Implementar código.
 - Generar elementos de prueba: módulos impulsores (drivers), módulos ficticios (stubs), interfaces, etc.
- Ejecutar pruebas:

- Fallos de las pruebas.
- Estadísticas de fallos.
- Calidad de las pruebas.
- Calidad del software (funcionalidad, rendimiento, etc.)

1.7.4 Herramientas de gestión de configuración.

En entornos de desarrollo complejos, especialmente integran diversas herramientas de ingeniería de software, se hace imprescindible la incorporación de una herramienta capaz de gestionar la configuración de los sistemas.

Este tipo de herramientas ofrece distintos tipos de capacidades:

1. Control de versiones, capacidad de proporcionar almacenamiento y acceso controlado a los datos, así como de registrar los cambios sobre los mismos y poder recuperar versiones anteriores.
2. Construcción, consiste en automatizar las tareas asociadas a la compilación y enlace de un sistema con el fin de generar ejecutables.
3. Trazabilidad de requisitos y análisis de impacto, permiten respectivamente, rastrear un requisito hasta su implementación, y conocer los elementos del sistema que se ven afectados ante un cambio.

1.7.5 Herramientas de ingeniería inversa.

- Ingeniería inversa de datos, son capaces de extraer la información del código fuente, que describe la estructura de los elementos de datos; construyendo así diagramas Entidad-Relación partiendo de esquemas relacionales en red, o incluso ficheros.

- Ingeniería inversa de procesos, permiten aislar la lógica de las entidades, y las reglas del negocio a partir del código.
- Reestructuración de código fuente, modifican su formato o implantan un formato estándar.
- Redocumentación, permiten generar diagramas a fin de que se comprenda mejor el código.
- Análisis de código, sus funcionalidades van desde la indentación automática del código fuente hasta la posibilidad de ir visualizando dinámicamente las llamadas del mismo.

1.8 Integración de las herramientas CASE.

Las herramientas CASE se pueden utilizar de diversas formas:

- Como herramientas aisladas, donde sólo debe abordarse la compatibilidad de los elementos del entorno.
- En pequeños grupos, que se comunican directamente, para los que la integración está predefinida de manera propietaria.
- En presencia de un marco de integración más amplio, donde habrá que abordar la capacidad de la herramienta para utilizar servicios relevantes.

Tradicionalmente se distinguen tres clases de integración:

- Integración de datos, que establece la posibilidad de que una herramienta utilice o entregue información a otras herramientas, bien sea mediante un mecanismo de transferencia de datos o a través de un repositorio.
- Integración de control, situación en la que una herramienta puede invocar a otras y comunicarles eventos.
- Integración de presentación, que se refiere a la homogeneidad y consistencia de la interfaz de usuario, por ejemplo, que en todas las herramientas se acceda a la ayuda de la misma manera evitando que el usuario se encuentre con sorpresas.

Actualmente existen diferentes estándares de integración para las herramientas CASE, entre los que destacan los desarrolladores por ANSI X3H6C (Tool Integration Models Comité), OMG (CORBA, Common Object Request Broker Architecture) y el IEEE (The Institute of Electrical and Electronics Engineers. Inc); aunque, podemos destacar dos modelos de integración:

- Modelo "CEBOLLA" (ver Figura 1.14), cuyo núcleo es el sistema operativo sobre el que se basa el repositorio que agrupa diferentes herramientas del entorno sobre las cuales se sitúa la interfaz de usuario que proporciona una interacción homogénea entre el usuario y el sistema.
- Modelo "TOSTADORA", propuesto por la ECMA (ver Figura 1.15), en el que las herramientas se encajan en el entorno como si fuesen las tostadas en una tostadora.

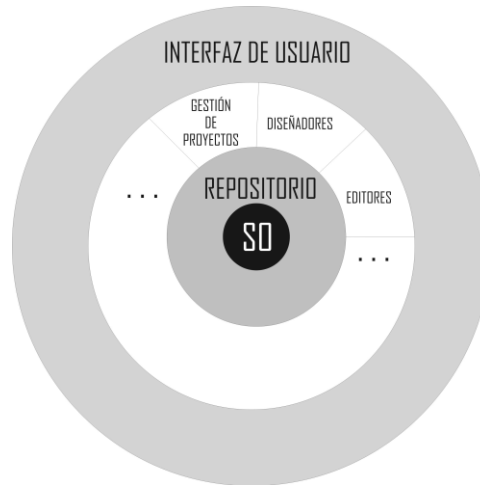


Figura 1.14: Modelo "CEBOLLA" para entornos CASE integrados

Por lo que respecta para la integración, además del diccionario que constituye el elemento básico de coordinación y comparación de información, existen otras denominadas "ARMAZONES" (Frameworks).

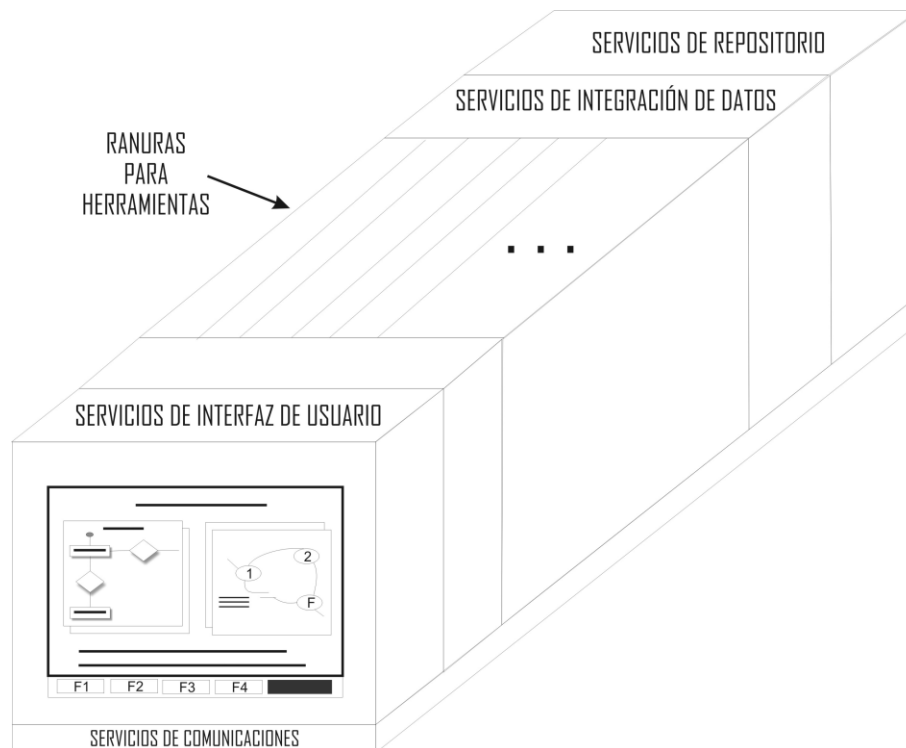


Figura 1.15: Modelo de referencia ECMA

1.9 Implantación de las herramientas CASE.

La tecnología CASE está experimentando un avance en distintas áreas, resolviendo bastantes de los defectos que presentaba la primera generación de productos. Sin embargo, el principal desafío de esta tecnología sigue siendo su adopción por parte de los profesionales y las empresas. Se calcula que en la práctica se abandona la tecnología CASE en un alto porcentaje: el 70% de las herramientas y técnicas se deja de utilizar un año después de su introducción, el 25% lo emplea sólo un grupo o una persona dentro de la empresa, mientras que el 5% restante sí se emplea ampliamente, pero no al 100% de su capacidad.

1.10 Causas del fracaso en la adopción de CASE.

1. Deficiencia de la propia tecnología.

Un gran número de empresas que empezaron a utilizar herramientas CASE en los años ochentas, posteriormente las abandonaron debido a sus inconvenientes, entre los cuales destacan:

- Soporte parcial del ciclo de vida, lo que permite automatizar sólo parte de las actividades de desarrollo, mientras que otras se siguen realizando de forma tradicional.
- Incompatibilidad entre herramientas, incluso entre distintas versiones de la misma herramienta que no siempre se encuentran "sincronizadas" en todas las plataformas hardware/software sobre las que actúan.

- Escasa integración entre las mismas herramientas, así como entre éstas y el resto del entorno: SGBD, lenguajes de cuarta generación, generadores de informes, etc.
- Poca confianza en el vendedor / distribuidor, ya que algunas empresas de CASE son relativamente pequeñas y corren el peligro de desaparecer o ser absorbidas.
- Escasa e inadecuada documentación generada por la herramienta.
- Gran abundancia de herramientas, señalada muchas veces como inconveniente, ya que produce una especie de "bloqueo" a la hora de comprar la herramienta.
- Funcionamiento deficiente en entornos multiusuarios, ya que muchas herramientas surgieron para ordenadores personales.
- Poca capacidad de adaptación a los usuarios.
- Un alto costo, no sólo de la herramienta sino también de la plataforma que exige.

Todas estas deficiencias pueden, actualmente, ser superadas en mayor o menor medida, evaluando varias herramientas, considerando, si fuera posible, un cambio en el hardware / software utilizado intentando cuantificar el coste de la "no adopción" (con especial énfasis en el mantenimiento) y valorando los beneficios que el CASE puede soportar.

2. Deficiencias en la aplicación de la teología a los problemas (incorrecta implantación).

Otra causa del fracaso se debe a la aplicación de herramientas CASE a problemas para los que no están preparadas debido a que:

- Soportan una sola metodología por ejemplo, especializada en el desarrollo de aplicaciones de gestión, y que se pretende emplear para construir sistemas de tiempo real.
- No soportan la técnica más adecuada, por ejemplo en el diseño de bases de datos muy grandes puede ser conveniente emplear la "integración de vistas", que muchas herramientas CASE no soportan.
- Metodologías y herramientas que funcionan relativamente bien en proyectos pequeños o medianos, pueden fracasar en proyectos grandes y viceversa.
- La selección se centra sólo en factores técnicos por lo que la herramienta resulta insuficiente para los aspectos relativos a la gestión que todo desarrollo lleva consigo.

Las medidas más eficaces para afrontar estos problemas pueden ser: comprender y analizar los distintos tipos de metodologías y herramientas existentes (junto a su "escalabilidad"), utilizando las herramientas adecuadas a cada problema, lo que supone un gran esfuerzo en formación e inversión en consultoría.

3. Deficiencias de la propia organización (falsas expectativas)

Independientemente de las deficiencias citadas anteriormente que son propias de las herramientas en sí mismas, muchas veces los fracasos en la adopción de herramientas CASE son debidos a problemas de la propia organización. En definitiva, la adopción de la "filosofía" CASE es, como la

transferencia de cualquier otra tecnología, un tema más "cultural" que tecnológico.

Las causas de fracaso más notables en esta área:

- Actitud por parte de los directivos, que pretenden introducir la tecnología CASE como la "salvación" de todos los males del desarrollo sin contar con una base metodológica
- Infravaloración del esfuerzo requerido (económico, formación, aceptación).
- Incapacidad para encontrar las metodologías y herramientas adecuadas del nivel de madurez de la organización.
- Inadecuada formación.
- No medir la productividad ni la rentabilidad que resulten de la aplicación de la tecnología.

Estas deficiencias se pueden superar con una gestión adecuada de las expectativas, siendo realista (conociendo la cultura de la empresa y su historia frente a los cambios tecnológicos) y con una buena gestión.

1.11 Herramientas CASE Cliente/Servidor.

A la relación entre las herramientas CASE y la arquitectura Cliente/Servidor se puede determinar al plantear las siguientes preguntas:

- ¿Cuáles son las influencias de las herramientas CASE en las empresas desarrolladoras de sistemas de información Cliente/Servidor?

- ¿Cuáles son las tendencias actuales de las empresas fabricantes de sistemas Cliente/Servidor?

1.11.1 Arquitectura Cliente/Servidor.

Con la aparición de las redes de ordenadores en empresas y universidades, ha surgido en el mundo de la informática la tecnología Cliente/Servidor. Una gran cantidad de organizaciones que ya cuentan con un número considerable de aplicaciones Cliente/Servidor en operación: Servidores de Bases de Datos y Manejadores de Objetos Distribuidos.

Esta arquitectura es una tecnología de bajo costo que proporciona recursos compartidos, escalabilidad, integridad, encapsulamiento de servicios, etc; pero al igual que toda tecnología, el desarrollo de aplicaciones requiere que la persona tenga conocimientos, experiencia y habilidades en procesamiento de transacciones, diseño de base de datos, redes de computadoras y diseño gráfico de interfase.

Clientes y Servidores son entidades lógicas separadas que trabajan junto en una red, para cumplir una tarea. Todo sistema Cliente/Servidor tiene las siguientes características:

- Servicio: Cliente/Servidor es principalmente una relación entre ejecución de procesos de máquinas separadas. El servidor de procesos es un proveedor de servicios; el cliente es un consumidor de servicios. En esencia, Cliente/Servidor provee una separación limpia de funciones basadas en la idea de servicios.

- Recursos Compartidos: Un servidor puede ofrecer servicios a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.
- Protocolos Asimétricos: Hay una relación de muchos a uno entre clientes y servidores. Los clientes siempre inician el diálogo para solicitar un servicio. Los servidores están esperando pasivamente por solicitudes de clientes.
- Localidad Transparente: El servidor es un proceso que puede permanecer en la misma máquina como el cliente o en una máquina diferente de la red.
- Intercambio de Mensajes: Los clientes y servidores se acoplan a sistemas que actúan recíprocamente por un mecanismo de pase de mensaje (message passing).
- Encapsulación de Servicios: El servidor es un especialista. Un mensaje le dice a un servidor qué servicio es solicitado; éste entonces le indica al servidor como realizar el trabajo. Los servidores pueden ser actualizados sin afectar la interface de pase de mensajes con los clientes.
- Escalabilidad: Los sistemas Cliente/Servidor pueden ser escalados horizontalmente o verticalmente. La escalabilidad horizontal significa agregar o quitar estaciones de trabajo cliente con sólo un impacto en la ejecución. Una escalabilidad vertical significa emigrar a una máquina servidora más grande y más rápida o múltiples servidores.
- Integridad: El código y el dato del servidor es centralmente mantenido, el cual resulta un mantenimiento más barato y guardando la integridad de los datos compartidos.

1.11.2 Evolución de la tecnología Cliente/Servidor.

La primera generación Cliente/Servidor fue causada por los NOS (Network Operating System). Los NOS facilitan a las aplicaciones compartir archivos, impresoras y otros dispositivos conectados a la red; desempeñan su trabajo extendiendo el alcance del sistema operativo. Se podría llamar a la primera generación de Cliente/Servidor la "generación Netware".

La segunda generación del Cliente/Servidor fueron las aplicaciones centradas en bases de datos. La tecnología predominante es el "servidor de bases de datos SQL". Sin embargo, también se experimentan otras dos grandes generaciones tecnológicas causadas por el GroupWare y los TP monitors.

La tercera generación de Cliente/Servidor son los objetos distribuidos. Los objetos rodean la tecnología de la primera y segunda generación, y añaden un nuevo valor considerable. Tienen el único potencial de distribuir inteligencia entre clientes y servidores donde más se requiere.

1.11.3 Descripción del panorama general del uso de herramientas CASE en aplicaciones Cliente/Servidor.

- CASE al nivel de Estructura de Costos.

Las empresas desarrolladoras, al decidir adoptar una herramienta CASE, asimilan una serie de costos tangibles e intangibles que afectan el proceso de desarrollo de las futuras aplicaciones Cliente/Servidor.

Dichos costos se pueden diferenciar en 3 tipos:

- Precio de Venta. Las herramientas CASE, por su complejidad de desarrollo y su alto nivel de especialización, son muy costosas. Sin embargo, las herramientas más caras resultan más baratas para la empresa desarrolladora si ésta posee una gran cantidad de recursos humanos destinados a proyectos. La razón es que la licencia de las herramientas costosas es única, en cambio, la de las otras herramientas es por máquina instalada. Esta variación en el precio incide, en la toma de decisión de la Alta Gerencia en relación a cuál herramienta debe elegir para un proyecto determinado.
- Costo de Entrenamiento de Personal. La gran complejidad que poseen las herramientas CASE también se traduce en un aumento de los costos de desarrollo de sistemas, debido a los costos generados por la curva de aprendizaje del personal y los costos por entrenamiento. Este incremento se reduce con el tiempo, a medida que los desarrolladores adquieren más destreza en el uso de la herramienta y son más productivos.

El costo es considerablemente elevado y en muchas ocasiones esto ha provocado que algunas empresas dejen de usar las herramientas CASE por considerarlas improductivas.

Un factor que influye en la inclinación de la curva de aprendizaje es un bajo nivel de restricción de la herramienta CASE. Una herramienta que posea pocas restricciones, "puede sobrecargar a un analista al ofrecer más opciones de las que es capaz de manejar. El resultado final puede ser que la herramienta CASE no sea usada apropiadamente"

- Costo de Adopción de la Metodología Asociada a la Herramienta CASE. Debido a que toda herramienta CASE posee una metodología de trabajo asociada, y muy específica; se genera un costo de

desarrollo adicional por adoptar una herramienta cuya metodología sea diferente a la imperante en la empresa. Ello puede generar, a su vez, brotes de hostilidad del personal hacia la herramienta. La implementación de las herramientas CASE integradas en una organización puede ser muy bien recibida por el personal deseoso para su utilización, bien educado en un fondo teórico, y ser apaciblemente introducido en la mecánica de la herramienta a través de un excelente entrenamiento y soporte durante el mismo. La misma herramienta en otro lugar puede ser recibida con hostilidad, con el personal sintiendo que ha sido obligado por la Gerencia. Uno de los errores más comunes es el de elegir una herramienta CASE que soporte un método que no sea familiar a los desarrolladores.

- CASE al nivel de I Rango de Aplicación (CASE Cliente/Servidor).

Características deseables de una herramienta CASE Cliente/Servidor.

Una herramienta CASE Cliente/Servidor provee modelos de datos, generación de código, registro del ciclo de vida de los proyectos, múltiples repositorios de usuarios, comunicación entre distintos ingenieros. Las principales herramientas son KnowledgeWare's Application Development Workbench, TI's Information Engineering Facility (IEF), y Andersen Consulting's Foundation for Cooperative Processing.

Por otra parte, una herramienta CASE Cliente/Servidor debe:

- Proporcionar topologías de aplicación flexibles. La herramienta debe proporcionar facilidades de construcción que permita separar la aplicación (en muchos puntos diferentes) entre el cliente, el servidor y más importante, entre servidores.

- Proporcionar aplicaciones portátiles. La herramienta debe generar código para Windows, OS/ 2, Macintosh, Unix y todas las plataformas de servidores conocidas. Debe ser capaz, a tiempo de corrida, desplegar la versión correcta del código en la máquina apropiada.
- Tener un Control de Versión. La herramienta debe reconocer las versiones de códigos que se ejecutan en los clientes y servidores, y asegurarse que sean consistentes. También, la herramienta debe ser capaz de controlar un gran número de tipos de objetos incluyendo texto, gráficos, mapas de bits, documentos complejos y objetos únicos, tales como definiciones de pantallas y de informes, archivos de objetos y datos de prueba y resultados. Debe mantener versiones de objetos con niveles arbitrarios de granularidad; por ejemplo, una única definición de datos o una agrupación de módulos.
- Crear código compilado en el servidor. La herramienta debe ser capaz de compilar automáticamente código 4GL en el servidor para obtener el máximo performance.
- Trabajar con una variedad de administradores de recurso. La herramienta debe adaptarse ella misma a los administradores de recurso que existen en varios servidores de la red; su interacción con los administradores de recurso debería ser negociable a tiempo de ejecución.
- Trabajar con una variedad de software intermedio. La herramienta debe adaptar sus comunicaciones cliente/servidor al software intermedio existente. Como mínimo la herramienta debería ajustar los temporizadores basándose en, si el tráfico se está moviendo en una LAN o WAN.

- Soporte multiusuarios. La herramienta debe permitir que varios diseñadores trabajen en una aplicación simultáneamente. Debe gestionarse los accesos concurrentes a la base de datos por diferentes usuarios, mediante el arbitrio y bloqueos de accesos a nivel de archivo o de registro.
- Seguridad. La herramienta debe proporcionar mecanismos para controlar el acceso y las modificaciones a los que contiene. La herramienta debe, al menos, mantener contraseñas y permisos de acceso en distintos niveles para cada usuario. También debe facilitar la realización automática de copias de seguridad y recuperaciones de las mismas, así como el almacenamiento de grupos de información determinados, por ejemplo, por proyecto o aplicaciones.
- Desarrollo en equipo, repositorio de librerías compartidas. Debe permitir que grupos de programadores trabajen en un proyecto común; debe proveer facilidades de check-in/ check-out registrar formas, widgets, controles, campos, objetos de negocio, DLL, etc; debe proporcionar un mecanismo para compartir las librerías entre distintos realizadores y múltiples herramientas; gestiona y controla el acceso multiusuario a los datos y bloquea los objetos para evitar que se pierdan modificaciones inadvertidamente cuando se realizan simultáneamente.

1.11.4 Clasificación de las herramientas CASE Cliente/Servidor.

Las herramientas CASE Cliente/Servidor se pueden clasificar en dos grupos: las más modestas y baratas (Visual Basic, Power Builder, Delphi, ERwin), y las llamadas herramientas integradas (IEF, Oracle CASE). Su costo está en proporción directa con su rango de aplicabilidad para desarrollar sistemas de

información. Se ha demostrado que las herramientas del primer grupo no sirven para desarrollar sistemas de complejidad muy grande (Sistemas distribuidos, multiplataformas, o cualquier otro que consuma gran cantidad de recursos durante su desarrollo). Esto influye claramente en las políticas de desarrollo de una empresa que posea alguna herramienta, de forma tal que se han desarrollado metodologías para elegir la herramienta CASE más acorde a las características del proyecto a llevar a cabo. Si bien la diversidad de herramientas CASE es bastante marcada, las empresas fabricantes están mostrando varias tendencias fundamentales de integración, a saber:

- Las futuras versiones de las herramientas CASE integradas serán más abiertas, es decir, admitirán en su metodología el uso de herramientas más pequeñas, además, cada vez se vislumbran acuerdos para utilizar estándares conocidos (como OLE), se hacen públicos más y más acuerdos de integración de tecnologías de diferentes fabricantes.
- Las herramientas CASE cada vez más facilitan la centralización de los archivos fuente y de documentación de los proyectos en entes llamados repositorios, donde puedan almacenarse eficientemente durante una o más fases del ciclo de desarrollo de un sistema.

Capítulo 2

Descripción de algunas herramientas CASE¹³.

En este capítulo se presentan las características principales de herramientas consideradas como herramientas CASE las cuales son:

- ERwin
- Easy CASE
- Oracle
- Oracle Dsigner
- PowerBiulder de PowerSoft
- PowerDesigner
- System Architect
- SNAP
- Visual Basic
- Visual FoxPro y Visual C++
- Cool Stuf de Sterling Software
- Informix
- Opal de Computer Associates.

¹³ [Sub – Jefatura, 2006]

2.1 ERwin

Herramienta de diseño de base de datos. Brinda productividad en diseño, generación, y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada, ERwin permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la base de datos. Genera automáticamente las tablas y miles de líneas de procedimiento almacenado y triggers para los principales tipos de base de datos.

Hace fácil el diseño de una base de datos. Los diseñadores de bases de datos sólo apuntan y pulsan un botón para crear un gráfico del modelo E-R (Entidad-Relación) de todos sus requerimientos de datos y capturan las reglas de negocio en un modelo lógico, mostrando todas las entidades, atributos, relaciones, y llaves importantes.

Más que una herramienta de dibujo, automatiza el proceso de diseño de una manera inteligente. Por ejemplo, habilita la creación de un diccionario de atributos reusables, asegurando la consistencia de nombres y definiciones para su base de datos.

Se mantienen las vistas de la base de datos como componentes integrados al modelo, permitiendo que los cambios en las tablas sean reflejados automáticamente en las vistas definidas. La migración automática garantiza la integridad referencial de la base de datos.

Establece una conexión entre una base de datos diseñada y una base de datos, permitiendo transferencia entre ambas y la aplicación de ingeniería reversa. Usando esta conexión, genera automáticamente tablas, vistas, índices, reglas de

integridad referencial (llaves primarias, llaves foráneas), valores por defecto y restricciones de campos y dominios.

ERwin soporta principalmente bases de datos relacionales SQL y bases de datos que incluyen Oracle, Microsoft SQL Server, Sybase, DB2, e Informix. El mismo modelo puede ser usado para generar múltiples bases de datos, o convertir una aplicación de una plataforma de base de datos a otra.

Especificaciones Técnicas:

- Software de Aplicación Compatibles:
 - NetDynamics,
 - PowerBuilder,
 - PROGRESS,
 - Visual Basic

- Bases de Datos Compatibles:
 - CA-Clipper,
 - CA-OpenIngres,
 - DB2 for MVS y DB2 for OS/390,
 - DB2 UDB,
 - dBASE, FoxPro,
 - HiRDB,
 - Informix,
 - InterBase,
 - Microsoft Access,
 - Microsoft SQL Server,
 - Oracle,
 - Paradox,
 - Rdb,

- Red Brick Warehouse,
 - SAS,
 - SQL Anywhere,
 - SQLBase,
 - Sybase,
 - Teradata.
- Sistemas Operativos Compatibles:
 - Windows NT,
 - Windows 95,
 - Windows 98
- Requerimientos Técnicos:
 - Mínimo 10 MB de espacio de disco duro,
 - 16 MB RAM (32 MB RAM recomendado para modelos largos.)

2.2 EasyCASE

EasyCASE Profesional es un producto para la generación de esquemas de base de datos e ingeniería inversa trabaja para proveer una solución comprensible para el diseño, consistencia y documentación del sistema en conjunto.

Esta herramienta permite automatizar las fases de análisis y diseño dentro del desarrollo de una aplicación, para poder crear las aplicaciones eficazmente, desde procesamiento de transacciones a la aplicación de bases de datos de cliente/servidor, así como sistemas de tiempo real.

Permite capturar los detalles de diseño de un sistema y comunicar las ideas gráficamente, para que sean fáciles de ver y entender. Para un diseño legítimo y modelamiento de datos, procesos y eventos, permite crear y mantener diagramas de flujo de datos, diagramas de entidad-relación, mapas de estructura, etc.

Posee herramientas de corrección avanzadas que permiten revisiones generales en minutos, en lugar de horas o días. Permite re-usar diagramas o partes de diagramas para economizar el diseño de un proyecto.

Soporta una amplia variedad de metodologías estructuradas, permitiendo escoger los métodos más apropiados para realizar las tareas, determina los tipos de esquemas según la metodología del proyecto seleccionado y notifica errores a medida que el modelo se está construyéndose.

Soporta el modelamiento de datos, procesos y eventos; posee desde el editor de diagramas flexible y un diccionario de los datos integrado en formato dBASE, hasta una extensa cantidad de reportes y análisis.

EasyCASE Profesional, es una herramienta multi-usuario, es ideal para aquellos que necesitan compartir datos y trabajar en un proyecto con otros departamentos. El equipo completo puede acceder proyectos localizados en el servidor de la red concurrentemente. Para seguridad de los datos, existe el diagrama y diccionario de los datos que bloquean por niveles al registro, al archivo y al proyecto, y niveles de control de acceso.

Especificaciones de EasyCASE Profesional:

- METODOLOGIAS ESTRUCTURADAS:
 - Yourdon/DeMarco
 - Gane \& Sarson

- Ward-Mellor
 - SSADM
 - Yourdon/Constantine
 - Chen
 - Martin
 - Bachman
 - Shlaer-Mellor
 - IDEF1X
 - Merise
 - Metrica
- BASE DE DATOS QUE SOPORTA:
 - Oracle
 - Paradox
 - Progress
 - SQLBase
 - SQL Server
 - Sybase
 - Watcom SQL
 - Access
 - ANSI SQL
 - Clipper
 - dBASE III , IV, V
 - DB2
 - FoxPro
 - Informix
- TIPOS DE DIAGRAMAS:
 - Data Flow Diagrams (DFDs)

- Transformation Schema (real-time DFDs)
 - Structure Charts (STCs)
 - State Transition Diagrams (STDs)
 - Entity Relationship Diagrams (ERDs)
 - Data Model Diagrams (DMDs)
 - Data Structure Diagrams (DSDs)
 - Entity Life History Diagrams (ELHs)
 - Logical Data Structure Diagrams (LDSs)
-
- REQUERIMIENTOS DEL SISTEMA:

EasyCASE Professional 4.2 o superior requiere:

- EasyCASE Database Engineer;
- PC's 386/486/Pentium y compatibles;
- Microsoft Windows 3.1 o superior,
- 8 MB RAM,
- 8 MB de espacio en disco duro;
- VGA o mejor color;
- Mouse. Requerimientos para Red:
 - NetBIOS-compatible DOS,
 - MS-Windows o Novell network.

2.3 Oracle.

Siguiendo la orientación al Web, Oracle en la actualidad está enfocada directamente a su Arquitectura de Computación de Redes (NCA), considerada como un servidor universal de datos, aprovechando lo mejor de los tres mundos: Web, cliente/servidor y orientación a objetos. Sus herramientas de desarrollo son básicamente tres:

- Developer/2000, herramienta tipo RAD (Rapid Application Development), presenta ventajas como sencillez, orientada a cliente/servidor y desarrollar ambientes Web. Genera software basado en Visual Basic y Java para que pueda correr en cualquier navegador de internet. Developer/2000 funciona sólo en Oracle, pero soporta básicamente las bases de datos SQL Server de Microsoft e Informix.
- Oracle J-Dveloper, un generador de software de objetos en Java que pueden correr en cualquier navegador de internet y permite reutilizarlos.
- Designer/2000, herramienta de modelado de alto nivel para procesos, entidad-relación, work flow y modelados funcionales. La principal diferencia de esta herramienta es que manteniendo un modelado de alto nivel puede generar la aplicación final y luego realiza reingeniería de reverso para actualizar el repositorio central.

2.4 Oracle Designer.

Oracle Designer es un juego de herramientas para guardar las definiciones que necesita el usuario y automatizar la construcción rápida de aplicaciones cliente/servidor flexibles y gráficas.

Integrado con Oracle Developer, Oracle Designer provee una solución para desarrollar sistemas empresariales cliente/servidor de segunda generación.

Sofisticadas aplicaciones cliente/servidor pueden ser 100% generadas usando la lógica de la aplicación y el módulo de componentes reusables. Oracle Designer también habilita la captura del diseño de sistemas existentes, salvaguardando la versión actual.

Todos los datos ingresados por cualquier herramienta de Oracle Designer, en cualquier fase de desarrollo, se guardan en un repositorio central, habilitando el trabajo fácil del equipo y la dirección del proyecto.

En el lado del Servidor, Oracle Designer soporta la definición, generación y captura de diseño de los siguientes tipos de bases de datos, por conexión nativa de Oracle y por conectividad ODBC :

- Oracle8, Oracle7
- Personal Oracle Lite
- RDB
- ANSI 92
- DB2/2 and MVS
- Microsoft SQL Server
- Sybase

Muchas metodologías diferentes para bases de datos y desarrollo de aplicaciones existen actualmente. Oracle Designer no fuerza al uso de alguna metodología específica, pero en cambio proporciona un juego de herramientas que le permiten que use la metodología de desarrollo que elija.

Oracle Designer soporta las siguientes metodologías:

- Desarrollo Rápido de Aplicaciones (RAD)
- Ingeniería de la Información (IE)
- Modelamiento de Procesos Asistido
- Captura de Diseño Asistido

Las herramientas de Oracle Designer se agrupan en áreas que reflejan las necesidades primarias de sus tipos diferentes de usuarios:

- Requisitos para el Modelamiento de Sistemas:
 - Uso de las herramientas en esta área:
 - Procesos para el modelo del negocio,
 - Re-examinar los métodos usados para conseguir las metas de la organización;
 - Crear representaciones diagramáticos de los procesos del negocio;
 - Detalles de los registros;
 - Describir los requisitos del negocio en detalle;
 - Crear modelos diagramáticos de las entidades, funciones y flujos de datos en los sistemas que constituyen la organización.
- Generadores de Diseños Preliminares:
 - Uso de transformadores para generar los diseños preliminares de los modelos creados anteriormente.
- Diseño y Generación:
 - Uso de las herramientas en esta área:
 - Diseño de sistemas que reúnan los requisitos comerciales de una organización;
 - Proveer un ambiente de desarrollo para los ingenieros de sistemas y diseñadores;
 - Crear componentes del lado del servidor y aplicaciones del lado del cliente, desde definiciones grabadas en el Repositorio de Datos.

- Utilitarios :
 - Uso de las herramientas en esta área:
 - Ingresar y editar la información en el Repositorio;
 - Mostrar las relaciones entre los elementos en el Repositorio de Datos;
 - Generar etiquetas predefinidas y personalizadas en el Repositorio;
 - Administrar el Repositorio de datos;
 - Escribir sentencias interactivas en SQL.

2.5 PowerBuilder de PowerSoft.

Con 30 manejadores de base de datos, ofrece dos opciones de conectividad:

- ODBC de Microsoft
- Conectividad nativa.

Una de las características principales¹⁴ de este producto es que comparte el mismo idioma de cada manejador.

Incluye entre otros módulos el Optima++, herramienta RAD basada en componentes que combina desarrollos cliente/servidor e Internet con el rendimiento de C++. Asimismo, ofrece un módulo opcional CASE Power Design que genera modelos lógicos y físicos de los distintos manejadores que soporta para acelerar los desarrollos. También cuenta con la herramienta Info Maker que

¹⁴ Muy apreciada por los usuarios, quienes dicen es mejor con Oracle e Informix que sus propias herramientas

ellos definen "como la estrellita" que permite de manera muy sencilla que los usuarios finales puedan hacer minería de datos.

Power Builder cuenta con conectividad para aplicaciones Java a través del controlador JDBC, desarrollado por Sybase y puede construir aplicaciones sobre cualquier plataforma.

2.6 PowerDesigner

PowerDesigner es una suite de aplicaciones de Powersoft para la construcción, diseño y modelado de datos a través de diversas aplicaciones.

Es la herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y conceptual, que dan a los desarrolladores Cliente/Servidor la más firme base para aplicaciones de alto rendimiento.

Esta suite cuenta con los siguientes productos:

- PowerDesigner ProcessAnalyst. Permite analizar el flujo de datos de toda la empresa, a través de los departamentos hasta el usuario final.
- PowerDesigner DataArchitect. Provee a los diseñadores de las bases de datos una manera eficiente para la creación inteligente, depuración e ingeniería de reversa del modelado, tanto conceptual como físico de los datos.
- PowerDesigner AppModeler. Permite el diseño y ajuste de los componentes de objetos y datos en aplicaciones de uso común como PowerBuilder, Power++, Visual Basic y Delphi, ajustando el modelo de base de datos. Junto con la aplicación de servidor PowerDynamo se pueden publicar las bases de datos en Internet/Intranet directamente del modelo de base de datos. Esta herramienta también puede generar páginas de servidor activas para MS Internet Information Server v.3.0.
- PowerDesigner WarehouseArchitect. Provee un poderoso almacén de datos para el diseño e implementación de una base de datos; además, con

soporte DBMS y bases de datos en plataformas de sistemas analíticos usando modelados dimensionales, esquemas de “estrella” y “nieve”, particionamiento y agregación, también, tiene un alto desempeño en el indexamiento de esquemas.

- PowerDesigner MetaWorks. Permite fácilmente ver y compartir la información del modelado de datos con una definición constante de objetos. También puede comparar y mezclar dos modelos de datos paso a paso.
- PowerDesigner Viewer. Crea reportes de los modelos físicos, conceptuales y procesos del modelado de la base de datos. También permite generar reportes para Internet en HTML. Este producto cuenta con demos directos del sitio de Sybase en Internet para su evaluación.

Además de todas estas características, PowerDesigner ofrece las posibilidades de: Soporte para tipos de datos abstractos.

PowerDesigner soporta la identificación de tipos de datos abstractos con ingeniería inversa de aplicaciones para Oracle8.

Soporte para usuarios de bases de datos. Los usuarios de bases de datos pueden ser recogidos de una lista existente y luego almacenados en un modelo físico de datos. Ahora, es posible añadir nuevos usuarios y también asignar usuarios como propietarios y vistas.

Mayor selectividad en ingeniería inversa. PowerDesigner permite seleccionar no sólo las tablas que se desean cargar, sino todo tipo de objetos de la base de datos.

Cálculo del tamaño de las bases de datos. Puede calcular y definir el tamaño definitivo de bases de datos de nuevo diseño y construcción, incluyendo tamaños detallados de índices y tablas.

2.7 System Architect.

System Architect posee un repositorio único que integra todas las herramientas, y metodologías usadas. En la elaboración de los diagramas, el System Architect conecta directamente al diccionario de datos, los elementos asociados, comentarios, reglas de validaciones, normalización, etc.

Posee control automático de diagramas y datos, normalizaciones y balanceamiento entre diagramas “Padre e Hijo”, además de balanceamiento horizontal, que trabaja integrado con el diccionario de datos, asegurando la compatibilidad entre el Modelo de Datos y el Modelo Funcional.

System Architect es considerado un Upper Case, que puede ser integrado a la mayoría de los generadores de código. Traduce modelos de entidades, a partir de la enciclopedia, en esquemas para Sybase, DB2, Oracle u Oracle 7, Ingress, SQL Server, RDB, XDB, Progress, Paradox, SQL Base, AS400, Interbase, OS/2, DBMS, Dbase 111, Informix, entre otros. Genera también Windows DDL, definiciones de datos para lenguaje C/C++ y estructuras de datos en Cobol. En esta última versión del System Architect es posible a través de ODBC, la creación de bases de datos a partir del modelo de entidades, para los diversos manejadores de bases de datos anteriormente mencionados.

Posee esquemas de seguridad e integridad a través de contraseñas que posibilitan el acceso al sistema en diversos niveles, pudiéndose integrar a la seguridad de la red Novell o Windows/NT de ser necesario. Posee también con un completo Help sensible al contexto. System Architect posee un módulo específico para Ingeniería Reversa desde las Bases de Datos SQL más populares,

incluyendo Sybase, DB2, Infonmix, Oracle y SQL Server (DLL), además de diálogos (DLG) y menús (MNU) desde Windows.

La Ingeniería Reversa posibilita la creación, actualización y manutención, tanto del modelo lógico como de su documentación. A través de ODBC, el System Architect logra leer bases de datos y construir el modelo lógico o físico (diagrama), alimentando su diccionario de datos con las especificaciones de las tablas y de sus elementos de datos, incluyendo las relaciones entre tablas y su cardinalidad. System Architect posee múltiples metodologías para diseño y análisis, incluyendo: Análisis Estructurado (DFD) en los modelos De Marco/Yourdon y Gane/Sarson, análisis de tiempo real en el modelo Ward \& Mellor; análisis esencial de sistemas; análisis orientado a objetos (OOA) en los modelos UML, Booch('91 y '94), Coad/Yourdon, Rumbaugh, Shaler/Mellor; Diagrama de entidad - relación (DER) en los modelos Peter Chen, James Martin, Bachman o Booch, Gráfico de Estructuras, Diagramas de Descomposición, Planeamiento Estratégico de informaciones, entre otras.

Estas metodologías para Desarrollo de Sistemas, proporcionan amplio soporte para la construcción de los Modelos Conceptual, Funcional y Operacional.

System Architect es una herramienta CASE de última generación, creada específicamente para la arquitectura "Cliente/Servidor", por eso posee control total de versiones, y de acceso, así como la administración completa de múltiples equipos de desarrollo.

Independientemente de la topología de la red de comunicación, System Architect es operable sobre Novell, Windows NT y OS2, y posee interfaces específicas con diversos utilitarios "front-end" como PowerBuilder, Visual Basic, SQL Windows, etc.

Posee más de 130 informes estandarizados, entre los cuales: Referencias cruzadas, datos y expresiones, definiciones, etc. Además de tener un módulo específico para personalización de informes que emplea los recursos gráficos y facilidad de edición de Windows.

- PRODUCTO:

System Architect Versión 4.0 de 1997.

Número de copias instaladas: 50.000, instaladas a más de 3.500 Clientes.

Principales clientes: AT&T, Xerox, Microsoft, Novell, American Airlines, IBM, Pepsi, 3M, Toshiba, GE, etc.

- PROVEEDOR

POPKIN Software \& Systems Incorporated EUA

Fundada en 1987 y representada en América Latina por CHOOSE TECHNOLOGIES - Sao Paulo - Brasil Rua Alexandre Dumas, y Buenos Aires - Argentina Viamonte 759

- REQUISITOS CLIENTE/SERVIDOR

- Windows
- IBM PC/AT 486DX-16 MB RAM
- Disco rígido con al menos 50 Mb Libres
- CD-ROM
- VGA Color
- Mouse

2.8 SNAP

SNAP es una herramienta CASE para el desarrollo de aplicaciones en Sistemas AS/400 de IBM. Proporciona el ambiente integral de trabajo, brindando la posibilidad de construir sistemas de inmejorable calidad, adheridos a los estándares S.A.A (Systems Application Architecture) de IBM., totalmente documentados y ajustados a los requerimientos específicos de la organización, en una fracción del tiempo y costo del que se invertiría, si se utilizaran herramientas tradicionales.

SNAP se ha consolidado como la herramienta CASE más poderosa y con mejor historial de resultados, disponible para desarrollo de sistemas en el AS/400 de IBM. Genera los programas nativos de mejor rendimiento en AS/400. Así mismo, proporciona dos ambientes de trabajo y genera aplicaciones nativas y/o Cliente/Servidor con el mismo esfuerzo de desarrollo.

En su arquitectura, SNAP implementa, de manera adecuada, el esquema metodológico de Entidad-Relación, facilitando las herramientas y guías necesarias para construir aplicaciones que exploten al máximo las virtudes y potencial del AS/400 en su modalidad nativa, siguiendo los lineamientos técnicos y de presentación que propone la filosofía S.A.A. de IBM.

SNAP se compone de cuatro grandes áreas:

- Modelo de Datos

En el Modelo de Datos, el analista introduce el diseño conceptual o representación de la estructura de información de la aplicación, siguiendo, paso a paso la metodología Entidad-Relación. Este diseño puede ser transferido desde otras herramientas (Upper Case), tales como VAW (Visible Analyst)¹⁵ o digitado directamente en SNAP; contando además, que

¹⁵ Es una de las principales herramientas CASE disponibles para dar soporte completo al Análisis y Diseño de los proyectos Informáticos.

éste prevé la facilidad de incorporar o trasladar DDS existentes al Modelo de Datos; una vez introducido, SNAP genera automáticamente la base de datos, compuesta por fuentes DDS y objetos nativos AS/400.

Al mismo tiempo el analista, sin necesidad de digitar ni una línea de código, queda habilitado para especificar y generar la base de programas que dan mantenimiento integral al modelo, incluyendo aquellos de soporte de consulta, integridad referencial, navegación por listas, informes necesarios para conformar un sistema.

Estos programas automáticos se generan utilizando todos los recursos y objetos disponibles que se administran y controlan desde el Repositorio Central, quedando a su vez registrados para su uso posterior por otros objetos o programas. Todos los programas generados en el Modelo de Datos quedan totalmente funcionales; sin embargo, el analista tiene la posibilidad de ajuste por medio del Modelo de Desarrollo Acelerado (MDA), segundo gran componente de SNAP.

- Método de Desarrollo Acelerado (MDA)

El MDA es una plataforma de trabajo para ajustar, en forma individual, los programas generados automáticamente en el Modelo de Datos. El MDA permite crear y mantener programas nuevos.

Al igual que con el Modelo de Datos, al trabajar con MDA, el analista se sirve estratégicamente de los elementos del Repositorio Central y lo actualiza con el resultado de su gestión.

El MDA proporciona las herramientas necesarias para ajustar programas individuales, con un altísimo grado de productividad y rendimiento, sin necesidad de recurrir a lenguajes tradicionales. Estas herramientas

incluyen, entre otras, un módulo de especificación de procesos y lógica, y un formateador de informes.

- Utilitarios

SNAP dispone de una serie de utilitarios para ayudar al analista a administrar y controlar el proceso de desarrollo de aplicaciones.

Entre las funciones que se ofrecen están las herramientas para definir los estándares de la organización en SNAP, utilitario de regeneración automática de sistemas como consecuencia de cambios en el modelo de datos, utilitarios de administración y control de repositorio, comandos para salvar, restaurar y recrear sistemas o elementos del repositorio central, documentación integral, y una variedad de utilitarios misceláneos de apoyo a la gestión de los analistas.

- Seguridad

SNAP incorpora un ambiente muy sofisticado para controlar y ayudar a la administración del proceso de desarrollo de sistemas. Se soportan los elementos necesarios para proteger, hasta en cinco niveles, las distintas definiciones y recursos del repositorio central.

2.9 Visual Basic

Es una evolución de su antecesor Basic y como su nombre lo indica, es un ambiente de desarrollo más visual. A partir de la versión 5.0 cuenta con un compilador original de códigos y está más orientado a ambientes cliente/servidor e incluye soporte e integración a aplicaciones Internet/intranet a través de la tecnología ActiveX. La popularidad de Visual Basic se debe a su simplicidad ya

que en cuanto a conectividad hay otros que lo superan, soporta FoxPro, Oracle, e Informix vía ODBC y aún cuando no está orientada a objetos porque no soporta polimorfismos, cumple algunas de las reglas de esta tecnología al permitir reutilizar componentes para el desarrollo de aplicaciones personalizadas.

2.10 Visual FoxPro y Visual C++

Las herramientas de desarrollo orientadas a objetos con que Microsoft cuenta son Visual FoxPro y Visual C++, siendo ahora lo más reciente InterDev. De tales herramientas, esta última es la primera que ayuda a los desarrolladores de aplicaciones basadas en Web en la construcción de sitios sofisticados totalmente interactivos. InterDev disminuye el ciclo de desarrollo al soportar los lenguajes de Internet Java y Visual Basic Script interconectándose con otros lenguajes como C++ o Visual Basic a través de componentes ActiveX, además, puede interactuar totalmente con FrontPage 97¹⁶. De esta manera ambos pueden trabajar en equipo para la construcción de sitios Web.

2.11 Cool Stuf, de Sterling Software

Esta herramienta cuenta con un módulo para generar ingeniería de software tradicional, así mismo, una línea de productos para desarrollo de aplicaciones cliente/servidor de múltiples capas y para ambientes distribuidos. Además puede generar aplicaciones para Internet/intranets, soporta métodos orientados a objeto UML y cuenta con interfaces MQSeries de IBM o DCE. Cool Stuf cubre todo el ciclo de vida del producto desde la reingeniería de los procesos del negocio, análisis, diseño, distribución de procesos de datos y generación automática de código que puede ser en C++, Java o Cobol. Para ello se apoya en la metodología de James Martin, así como también en metodologías basadas en Orientación a

¹⁶ Herramienta orientada a usuarios finales y diseñadores.

Objetos. Una desventaja de esta es que utilizar una herramienta CASE del tipo Cool Stuf toma más tiempo el desarrollo de software en las primeras fases de análisis y diseño, se asegura la calidad de la aplicación, el entendimiento y la documentación, así como también minimiza el mantenimiento.

2.12 Informix

Es otra de las empresas que también cuenta con su herramienta de desarrollo NewEra orientada a la plataforma cliente/servidor y es totalmente orientada a objetos; además posee dos formas de generar aplicaciones: en forma compilada y en interpretada. Ésta última disminuye considerablemente los tiempos de desarrollo. NewEra cuenta con una característica de particionamiento que permite al desarrollador decidir qué parte de la aplicación se va a ejecutar en la PC y qué parte en el servidor y esto se hace desde el mismo lenguaje y no a través de procedimientos almacenados. Su conectividad con otras plataformas se realiza por medio de drivers ODBC, específicamente para Informix, Oracle, Sybase.

2.13 Opal, de Computer Associates

Herramienta de desarrollo que sirve para preservar toda la inversión existente en las aplicaciones que tiene una empresa en funcionamiento y le agrega nuevo valor al integrar diferentes fuentes de información no sólo de ambiente mainframe sino cliente/servidor, AS/400 y todo de manera interactiva y más amigable. Presenta un ambiente de desarrollo gráfico que tiene capacidad de comunicación con cualquier terminal 3270, VT100 y 5250 e integra cualquier base de datos relacional que tenga un controlador ODBC.

Sin embargo, no es un maquillador de pantalla, ya que además de contar con una interfaz tipo Windows permite al usuario crear sus propios temas. Una de las

ventajas principales de Opal es CODE, es qué permite desarrollar una aplicación por una sola vez, independientemente del ambiente bajo el cual vaya a ser ejecutada y esa aplicación va a servir para un ambiente cliente/servidor, así como también para verlo a través de Internet e intranet. Cabe destacar que múltiples y diferentes fuentes de datos en la misma aplicación Opal pueden ser conectadas con una sesión 3270, VT100 y por otro lado estar accediendo a una base de datos Oracle cliente/servidor y toda esta información converge en un sólo punto que va a ser la aplicación Opal y luego se despliega de acuerdo a lo que se requiere.

Opal está compuesto por tres elementos:

- Integrator, ambiente de desarrollo orientado a objetos;
- Opal Player runtime, que permite ejecutar la aplicación para diversas plataformas y para Internet.
- Opal Server, para optimizar las comunicaciones entre la aplicación Opal que está corriendo en el cliente y los requerimientos de información hacia las fuentes de datos.

Capítulo 3

Evaluación de algunas herramientas CASE¹⁷.

En este capítulo se presenta una comparación entre cuatro herramientas CASE en lo que es:

- Las características generales
- Los diagramas
- Ayuda
- ***El esquema de la base de datos***
- ***El código que genera el lenguaje***
- ***La ingeniería hacia delante y de reversa***
- ***La sincronización de la base de datos***
- ***El diseño***
- ***La web***
- ***La metodología***
- Y los reportes

Sabemos que las herramientas CASE, son de gran utilidad en el proceso de planeación y que además, la información estará disponible para ser manipulada durante las etapas de desarrollo y mantenimiento del ciclo de vida del sistema, es considerada, como la mejor forma de diseñar diagramas y de almacenar el trabajo de desarrollo de un sistema en un repositorio, el cual actúa como un puente para ligar varias herramientas, mientras la información en este, puede ser usada para analizar la totalidad de un diseño, es decir, como algo que permite desarrollar sistemas en nuevas formas usando elementos existentes.

¹⁷ [Dirección, 2006]

Considerando que con el uso de herramientas CASE se tiende a tener pocos errores de análisis y diseño, además, de que las pruebas al sistema toman mucho menos tiempo, por lo que es recomendable hacer uso de ésta, seleccionando la metodología de desarrollo. Se tiene la ventaja de que aún cuando debido a la evolución constante de estos productos será difícil escoger la herramienta óptima.

Herramientas CASE seleccionadas para su evaluación:

- ERwin 3.0,
- ER/Studio 2.5,
- System Architech 4.0
- Power Designer 6.1

3.1 CARACTERÍSTICAS GENERALES

3.1.1 ERwin 3.0

ERwin es una herramienta para modelar, que ayuda a diseñar bases de datos de alto desempeño para cliente/servidor y Web/intranet, así como aplicaciones de almacenado de datos. La herramienta ERwin no solo ayuda a diseñar modelos de datos lógicos, también construye automáticamente estructuras de datos físicos con la información del diagrama. Cuando el modelo de datos esta listo para usarse, simplemente se selecciona el servidor donde se quiere construir la base de datos y se eligen las opciones de generación de esquema que se quieran incorporar. En minutos, ERwin automáticamente construye la base de datos física, incluyendo todas las tablas, índices, procedimientos almacenados, triggers de integridad referencial y otros componentes necesarios para manejar exitosamente los datos usados en la organización.

3.1.2 ER/Studio2.5

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejas. ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes.

3.1.3 System Architect 4.0

System Architect es una herramienta poderosa de modelado estructurado de datos, tiene la capacidad de identificar y clasificar personal para autorizar su entrada al sistema.

Los usuarios de red trabajan en un diagrama de proyecto y una llave de registro de diccionario de datos. Soporta la mayoría de los paquetes de red incluyendo Novell, 3Com, Banyan, DecNet, LAN Manager, STARLAN y otras. Define propiedades para cualquier entrada de diccionario, incluyendo definiciones, símbolos y diagramas. Construye ligas entre varios objetos del diccionario. Especifica y define requerimientos, planes de prueba, cambio de requerimientos, objetivos de negocios, metas, así como símbolos o grupo de símbolos que son afectados.

3.1.4 Power Designer 6.1

Es una herramienta para crear bases de datos y aplicaciones cliente/servidor basadas o no en Web. Permite a los diseñadores de aplicaciones complejas de cliente/servidor tener una descripción general de los procesos particulares para comprender mejor a la organización; exporta información del modelo físico y extiende atributos al diccionario de 4GL. Importa atributos extendidos de PowerBuilder. Soporta definición de atributos extendidos para PowerBuilder, Progress, Uniface, PowerHouse, Axiant, y NS-DK.

3.2 DIAGRAMAS

3.2.1 ERwin 3.0

Los diagramas de modelos de datos en ERwin se usan para generar o actualizar bases de datos relacionales, o sea, sistemas de almacenamiento de datos ordenados en tablas. Cuando se crea un diagrama ERwin, el modelo de la información se representa por entidades (gente, lugares y cosas), atributos (hechos acerca de una entidad, tales como nombre de la persona, dirección, edad, etc.), y relaciones entre entidades. Cada entidad corresponde a una tabla en la base de datos, con instancias que corresponden a los renglones de la tabla y atributos correspondientes a encabezados de columnas. Las relaciones, usadas por DBMS (data base management system) para ligar renglones de datos en tablas diferentes, están representadas como frases verbales en una línea conectando a dos entidades. Cuando se actualiza una base de datos física, ERwin automáticamente genera un script de definición de datos SQL, para crear tablas de bases de datos, incluyendo llaves, restricciones y códigos trigger SQL para reforzar la integridad referencial entre tablas relacionadas.

3.2.2 ER/Studio 2.5

La creación de diagramas es clara y rápida. Tiene la posibilidad de realizar diagramas con desempeño rápido, también, es posible cambiar el estilo de las líneas, los colores, tipos de letra, niveles de acercamiento, y modelos de despliegue, crear subvistas para separar y manejar áreas importantes.

ER/Studio automáticamente mantiene todas las dependencias entre subvistas y el diagrama completo. El Explorer Navigator facilita el trabajo hasta con los diagramas más grandes. Se usa el Internet Explorer para encontrar y seleccionar entidades, con un solo click inmediatamente enfoca una ventana de diagrama.

3.2.3 System Architect 4.0

El área de dibujo de diagramas puede ser del tamaño de 64" x 60" a 50" x 150". Es posible ver los diagramas en las siguientes opciones: tamaño actual, página completa, área usada, o porcentaje de reducción.

Permite la edición de un diagrama en cualquier modo de vista, seleccionar y mover objetos individualmente o usando el ratón para obtener la porción del diagrama que se desee, y cambiar el tamaño de objetos individuales proporcionalmente o no proporcionalmente usando el ratón. La herramienta Leveling Automatically nivela diagramas y usa un mecanismo simple para cambiar la herencia en cualquier dirección. Automáticamente crea diagramas descompuestos de la herencia del Data Flow Diagrams (Diagramas de flujo de datos).

3.2.4 Power Designer 6.1

PowerDesigner cuenta con herramientas para la creación y control de diagramas como son:

- Off-page Connector; que representa los flujos de entradas y salidas en un proceso
- Business Rules que define las reglas de uso para Procesos, Almacenamiento de datos, Entidades externas, y Flujos de datos
- CRUD Matrix, que define el efecto de un proceso de datos en términos de Crear, Leer, Actualizar, y Borrar operaciones (CRUD).

3.3 AYUDA

3.3.1 ERwin 3.0

La herramienta de ERwin, Workspace contiene una ventana de diagrama donde se crea el diagrama del modelo de datos y provee varias herramientas de cliente usadas en el proceso de modelado, por lo que se describen algunas recomendaciones para facilitar a cualquier primerizo su uso:

- Hacer click en la barra de menú de ERwin para conocer información de cómo usar los menús y editores de ERwin.
- Hacer click en la barra de herramientas para obtener rápidamente una idea de cuanto se puede hacer en ERwin sin siquiera abrir un menú.
- Hacer click en la barra de herramientas de color y tipo de letra para ver como se cambia el texto y los colores usados en un diagrama rápidamente.
- Hacer click en la caja de herramientas para aprender más acerca de cada símbolo usado en un diagrama ERwin.
- Hacer click en el Subject Area List Box en el lado derecho de la barra de herramientas para aprender como usar las áreas de sujeto ERwin para subdividir un modelo de datos grande.

3.3.2 ER/Studio 2.5

Ya sea que se inicie un nuevo diseño o se mantenga uno existente, ER/Studio está equipado con elementos de ayuda para hacer el trabajo de manera efectiva.

Las barras de herramientas tienen algunas sugerencias para el uso de las mismas, además de contar con ayuda en línea sensible al contexto.

3.3.3 System Architect 4.0

La ayuda en línea de System Architect es extensiva, e incluye tutoriales del modelado de datos, BPR (Business Process Reengineering – Reingeniería de Procesos), análisis estructurado.

3.3.4 Power Designer 6.1

La ayuda de Power Designer es sensible y adecuada al contexto.

3.4 ESQUEMA DE LA BASE DE DATOS

3.4.1 ERwin 3.0

Para diseñar un modelo de datos, ERwin proporciona propiedades específicas de DBMS y del servidor de editores que permiten definir nombres físicos y propiedades para las tablas, columnas y relaciones que genera cuando se crea el esquema de la base de datos, cuando ERwin crea un esquema de datos, genera un script de cliente DDL (Data Definition Language) usando la sintaxis correcta de SQL para el servidor seleccionado, donde se aprecia el código que genera ERwin y, si se desea, se puede modificar antes de que se cree la base de datos. Si el servidor soporta elementos avanzados como procedimientos almacenados y triggers de integridad referencial, ERwin proporciona editores de plantillas especiales y macros para guardar la información en tiempos determinados para

acelerar la creación de estos objetos en el servidor. En varios servidores, se pueden hasta crear objetos de almacenamiento físico como espacios de tablas y segmentos de enrolamiento de ERwin así como especificar la ubicación y parámetros de almacenamiento para las tablas de bases de datos que ERwin genera en el servidor.

3.4.2 ER/Studio 2.5

Las capacidades de diseño que contiene, ayudan a crear un diseño lógico que puede transformarse en cualquier número de diseños físicos. Como resultado, se puede mantener un diseño lógico normalizado mientras se desnormalizan los diseños físicos para su desempeño. ER/Studio mantiene ligas entre todos los niveles de su diseño por lo tanto puede mezclar cambios en cualquier dirección entre ellos.

ER/Studio revisa la normalización y la compilación con la sintaxis de la plataforma de la base de datos. Se pueden desplegar los modelos de datos usando la notación IDEF1X o IE.

ER/Studio permite tomar por omisión las opciones para todos los diagramas así como realizar cambios al momento de la ejecución.

3.4.3 System Architect 4.0

System Architect proporciona todos los elementos para diseñar un nuevo sistema o modificar un sistema actual. Es posible crear modelos lógicamente normalizados y modelos de datos físicamente desnormalizados usando el conjunto de herramientas de System Architect, también, se puede crear un modelo conceptual de las entidades y especificar su relación con otras; al avanzar el proyecto, se pueden incluir llaves primarias, atributos, reglas, restricciones de integridad

referencial, triggers personalizados, y cualquier otra información que se elija para mantenerla en el modelo.

Si se diseña un nuevo sistema usando un diccionario amplio de datos es posible especificar los requerimientos de los datos antes de comenzar el modelado mientras se está construyendo el modelo, o después de haber completado el diseño lógico.

Si se está modificando un sistema existente es posible usar la ingeniería de reverso de System Architect para crear un diagrama de modelo de datos físicos para el sistema actual. System Architect crea automáticamente un diagrama entidad-relación de un modelo de datos físico. Entonces se puede modificar el diagrama entidad-relación, creando un modelo lógico normalizado del nuevo sistema; una vez que se ha completado el diseño lógico, se pueden generar modelos físicos. Si se planea implementar una base de datos desnormalizada, se puede documentar el proceso de desnormalización usando diagramas vista local. System Architect mantiene ligas entre el modelo lógico, las vistas lógicas, y el modelo físico; por lo tanto los cambios al modelo lógico se reflejan automáticamente en el modelo físico. Al final, se tienen dos modelos físicos separados: uno del sistema actual y otro del sistema propuesto.

Para cualquier tipo de proyecto en el que se esté trabajando, System Architect proporciona flexibilidad para completar el trabajo. Se pueden elegir modos de despliegue en cualquier tiempo durante el proceso de diseño: conceptual, basado en llaves, totalmente atribuido, o despliegue físico, también, una vez que se ha completado el modelo lógico, se pueden ejecutar una serie de reglas revisadas y reportes de normalización para validar la integridad del diseño. System Architect prueba las Formas Normales: Primera, Segunda, Tercera, y Boyce Codd.

3.4.4 Power Designer 6.1

Data Architect proporciona capacidades de modelado de datos tradicional, incluyendo diseño de bases de datos, generación, mantenimiento, ingeniería de reversa y documentación para arquitecturas de bases de datos. Permite que los diseñadores de bases de datos creen estructuras de datos flexibles, eficientes y efectivas para usar una ingeniería de aplicación de bases de datos, también proporciona un diseño conceptual de modelo de datos, generación automática de modelo de datos, diseño de normalización física, sistema de manejo de bases de datos múltiples (DBMS) y soporte de herramientas de desarrollo, y elementos de reportes con presentación y calidad. El diseño se realiza en dos niveles:

- Nivel conceptual: entidades, relaciones, dominios, tipos de datos conceptuales, identificadores, y reglas de negocios. Basado en la notación de Ingeniería de Información de Martin.
- Nivel físico: tablas, columnas, dominios, llaves primarias, llaves foráneas, llaves alternadas, índices, restricciones de integridad referencial declarativa, vistas, parámetros de almacenamiento físico, reglas de negocios, triggers y procedimientos almacenados.

3.5 CÓDIGO QUE GENERA EL LENGUAJE

3.5.1 ERwin 3.0

ERwin combina bases de datos Back - End y desarrollo de aplicaciones Front - End en un ambiente unificado. Tiene soporte para Multi - Clientes, ERwin genera formas de entrada de datos en Visual Basic, DataWindows de Power Builder y PROGRESS SmartObjects del mismo modelo de datos, logrando que los desarrolladores incorporen aplicaciones altamente productivas en tres de los ambientes de desarrollo de bases de datos. ERwin extiende el editor estándar

Column Property Editor de tal forma que se pueden asignar rápidamente propiedades de columna del lado del cliente, tales como tipo de control por omisión. Despliega formato y reglas de validación de cliente para cada columna y genera formas de entrada de datos en uso y otros componentes de aplicación directamente del mismo modelo ERwin que crea la base de datos Back - End.

Para simplificar aún más el desarrollo de aplicaciones en Visual Basic, Logic Works también ofrece DataBOT(tm), un robot de software avanzado que genera dinámicamente todo el código de acceso de datos SQL requeridos, permitiendo hasta que los programadores novatos creen rápidamente aplicaciones sofisticadas de bases de datos de alto desempeño en los ambientes actuales.

3.5.2 ER/Studio 2.5

Genera otros objetos de base de datos como: vistas, procedimientos almacenados, defaults, reglas, y tipos de datos de usuario, con lo anterior ayuda a la auto ordenación de tipos de objetos para eliminar errores de dependencia al construir la base de datos. Tiene una opción para generar código fuente o para construir bases de datos. Soporte para crear bases de datos para Servidores SQL; y otra, para incluir código SQL y verificar la creación de objetos, además de la opción para agregar encabezados de comentarios.

3.5.3 System Architect 4.0

Genera archivos de definición de almacenamiento de trabajo (.WKS) y sección de pantalla (.SCS) para implementaciones de COBOL. System Architect /Power Builder Link permite el intercambio de la información de diseño entre System Architect y Power Builder, incluyendo DDL y atributos extensos.

3.5.4 Power Designer 6.1

Mediante el incremento del modelo de la base de datos, AppModeler genera instantáneamente objetos, componentes Data - Ware, y hasta aplicaciones básicas listas para ejecutarse inmediatamente en Power Builder, Power++, Visual Basic, Delphi, y Web - Based Objects. El AppModeler permite a los desarrolladores: diseñar modelos de bases de datos físicas o crearlas instantáneamente a través de la ingeniería de reversa de bases de datos existentes, generar, documentar y mantener bases de datos, generar rápidamente objetos de aplicación y componentes de datos para Power Builder 4.0 y 5.0; Visual Basic 3.0, 4.0, y 5.0; Delphi 2.0; Power++; y el Web.

Generación de objetos Power Builder. Soporta todas las ediciones de Power Builder 4.0 y 5.0, genera objetos personalizables y componentes basados en modelos de bases de datos físicos y plantillas que se encuentran dentro de las librerías de clases de su elección, objetos ventana y ventana de datos basadas en tablas, vistas y relaciones de llaves primarias-foráneas y hace ingeniería de reverso a los atributos. Incluye plantillas personalizables para la librería Power Builder Foundation Class (PFC).

Generación de objetos en Visual Basic. Soporta todas las ediciones de Visual Basic 3.0, 4.0, y 5.0. Incluye Add - In de Visual Basic para la fácil manipulación de plantillas predeterminadas personalizables, genera formas basadas en tablas, vistas, y relaciones de llaves primarias-secundarias, proyectos basados en modelos de propiedades y controles tales como menús, listas, etc.

Generación de objetos Delphi. Soporta todas las ediciones de Delphi 2.0. Incluye Add - In de Delphi para una manipulación de plantillas personalizables predefinidas. Genera aplicaciones y objetos (proyectos, formas, y controles) de tablas, columnas y referencias.

3.6 INGENIERÍA HACIA DELANTE Y DE REVERSA.

3.6.1 ERwin 3.0

- Ingeniería hacia delante

El desarrollo del modelo de Logic Works usa información específica en un modelo de datos ERwin para acelerar la creación y mantenimiento de soporte, migración y documentación de bases de datos relacionales.

El proceso de generar una base de datos físicos de un modelo de datos lógico se llama ingeniería hacia delante (Forward - Engineering).

A la Generación de un modelo lógico a partir de una base de datos física se llama ingeniería de reversa (Reverse - Engineering).

Es posible llevar a cabo la ingeniería hacia delante en ERwin (que se llama operación de diagrama) mediante la generación automática de un esquema de base de datos física directamente del modelo de datos sin codificar laboriosamente las definiciones de datos necesarias en SQL y los triggers.

- Ingeniería de reverso

La capacidad de ingeniería de reversa de ERwin, la cual puede inferir exitosamente las relaciones entre tablas, permite que se capture rápidamente la estructura de una base de datos existente y convertirla en un modelo lógico independiente del DBMS. Se puede usar el modelo de datos para análisis detallado, se define sobre tiempo y lo distribuye como parte de la documentación requerida a través del ciclo de desarrollo. Si se desea migrar la base de datos existente de una plataforma a otra, ERwin

puede hacer la ingeniería de reverso de la base de datos existente, crear un modelo de datos, modificar o agregar nuevos elementos según sea necesario y después construir la base de datos física en cualquier ambiente de servidor de los que soporta.

3.6.2 ER/Studio 2.5

- Ingeniería hacia adelante

Una vez que se ha diseñado la base de datos, se puede construir o generar código fuente para todo o para parte de los diseños de la base de datos. Propiamente hace la secuencia de la creación de tipos de objetos diferentes para asegurar eficiencia y construir bases de datos libres de errores.

Actualiza una base de datos del diagrama. ER/Studio permite aplicar cambios de diseño del modelo de datos directamente a la base de datos. Cuando se comparan las diferencias entre los dos, formula una estrategia de alteración inteligente que implementa el diseño de las modificaciones mientras se preserva la tabla con los datos existentes, privilegios de objetos, y dependencias en la base de datos.

- Ingeniería de reverso

Cuenta con ingeniería de reverso, cuando necesite iniciar un trabajo de una base de datos existente, ER/Studio puede hacer una ingeniería de reverso al esquema completo para cualquier plataforma de bases de datos. La operación de la ingeniería de reverso extrae eficientemente definiciones de objetos y construye un modelo de datos gráfico.

3.6.3 SYSTEM ARCHITECT 4.0

- Ingeniería hacia delante

Común a los anteriores.

- Ingeniería de reverso

Genera y hace ingeniería de reverso en los triggers, reglas, defaults, tipos de datos definidos, dispositivos y bases de datos. Elementos de la Ingeniería de reverso de SYSTEM ARCHITECT:

- Captura datos del sistema de bases de datos más popular: Access, DB2, Informix, Oracle, SQL Server o SYBASE.
- Ingeniería de reverso de un archivo DDL, a través del manejador ODBC, o a través de la interface directa del SQL Server.
- Genera diagramas de entidad-relación que incluyen definiciones para llaves, no llaves y relaciones.
- Genera diagramas de modelo de datos físicos que incluyen definiciones para tablas, columnas y constantes.
- Crea llaves primarias, índices alternados, y rutas de acceso de los índices en la estructura de la base de datos.
- Importa triggers, procedimientos almacenados, defaults, y roles de definiciones de bases de datos existentes.
- Captura, modifica, y regenera esquema usando SYSTEM ARCHITECT RDE con el Schema Generator.

- Captura, modifica, y regenera pantallas y menús de aplicaciones de Windows usando el SYSTEM ARCHITECT RDE con el SYSTEM ARCHITECT Screen Painter

3.6.4 Power Designer 6.1

- Ingeniería hacia delante

Común a los anteriores.

- Ingeniería de reverso

Visualiza estructuras de bases de datos existentes directamente vía ODBC o usando archivos de script DDL. Genera el modelo conceptual del modelo físico.

Retro-documentación de bases de datos existentes. Re-orientación de la base de datos existente a un DBMS diferente. Interfaz con herramientas de desarrollo: exporta información del modelo físico y extiende atributos al diccionario de 4GL, importa atributos extendidos de Power Builder, soporta definición de atributos extendidos para Power Builder, Progress, Uniface, Power House, Axiant, y NS-DK.

3.7 SINCRONIZACIÓN DE LA BASE DE DATOS

3.7.1 ERwin 3.0

El manejo de cambio comprensivo de ERwin y los elementos de generación de script de alteración facilita el almacenamiento del modelo de datos y la base de datos física de manera sincronizada. Durante la sincronización, ERwin desempeña una comparación comprensiva de todos los objetos físicos y lógicos, incluyendo definiciones de tablas y columnas, llaves, índices y parámetros de almacenamiento físico, resaltando cualquier discrepancia y permitiendo que se migren las definiciones de la base de datos modelo al modelo de la base de datos. Si se exportan cambios de un modelo a una base de datos, ERwin genera scripts de alteración de SQL para actualizar el esquema de la base de datos. ERwin automáticamente revisa todas las dependencias del esquema y traduce y preserva los datos existentes cuando se actualiza el esquema, facilitando los cambios de diseño migrados para activar el desarrollo, probar y producir bases de datos. Además de la sincronización del modelo de toda la base, ERwin soporta sincronización de modelo a modelo y modelo a script.

3.7.2 ER/Studio 2.5

- Sincronización entre el diagrama físico y el lógico.
- Mezcla entre cualquier par de diagramas físicos para la misma plataforma de bases de datos.
- Comparación lado-a-lado de las diferencias.
- El usuario puede decidir que diferencias mezclar o ignorar.
- Objetos reusables.
- Construir atributos reusables.
- Aplicarlos a atributos y columnas.
- Propagación global de actualizaciones.
- Construir tipos de datos personalizables.
- Submodelado.
- Crear cualquier número de subvistas personalizables sobre un diagrama físico o lógico.

- Cualquier objeto puede existir en cualquier número de subvistas (relaciones de muchos a muchos entre objetos y subvistas).
- Crear rápidamente subvistas eligiendo un área del diagrama.
- Control independiente sobre el despliegue de la subvista, incluyendo posición del objeto, colores y letras.
- Utilidad de búsqueda rápida.
- Editores en tabla para evitar la necesidad de poner en modo cascada los diálogos.
- Diferenciación de color de llaves primarias y secundarias inherentes.
- Sombreado de cajas de entidad.

3.7.3 System Architect 4.0

El Generador de esquemas (Schema Generator) automatiza la creación y mantenimiento de esquemas para DBMS de SQL y 4GL System Architect Schema Generator traduce modelos de datos (ER o diagramas físicos DM), IDEF1X y diagramas de clases orientadas a objetos en las definiciones de los esquemas para manejo de sistemas con múltiples bases de datos. La transferencia de esquemas en la DBMS se puede llevar a cabo en vivo, vía conexión ODBC, o a través del uso de un archivo. DDL intermedio. System Architect Schema Generator simplifica el desarrollo y asegura la integridad de los diseños permitiendo crear y mantener esquemas de bases de datos para una amplia variedad de administradores de sistemas de bases de datos SQL y no SQL. Cuando se genera un DBMS múltiple, los tipos de datos definidos en el modelo de datos se mapean automáticamente a los tipos correctos del DBMS seleccionado. Se usa el System Architect Schema Generator para generar:

- DDL para la creación de tablas y mantenimiento desde el repositorio de System Architect.

Definir Create Table o Alter Table, llaves primarias, y llaves secundarias.

- Crear definiciones para reglas, procedimientos almacenados, defaults, y mensajes.
- Triggers de integridad referencial y restricciones.
- Usar tipos de datos definidos por el usuario.
- Definiciones de Disk Init, Disk Mirror, Add Segment, y Create Database.
- Índices de rutas de acceso y llaves.
- Definiciones de tipos para C y libros de COBOL.
- El generador de esquema (Schema Generator) prueba las palabras reservadas de COBOL y automáticamente modifica con extensiones -XX.

3.7.4 Power Designer 6.1

Soporta más de 30 DBMS, incluyendo Sybase SQL Server, Oracle, Informix, Ingress, Sybase SQL Anywhere, Microsoft SQL Server, SQLBase, Progress, Access, Paradox, FoxPro, etc. Creación directa de bases de datos vía ODBC o usando archivos de scripts DDL. Elige la generación del modelo entero, ciertos submodelos, u objetos individuales. Reglas de negocios definidas de usuario pueden integrarse en triggers y procedimientos almacenados. Para el mantenimiento de la base de datos, incluye:

- Modelos de archivo.

- Comandos alterados generados para preservar datos contenidos.
- Mantenimiento vía ODBC o a través de archivos de scripts DLL.
- Elegir para generar el modelo entero, ciertos submodelos, u objetos individuales.

3.8 DISEÑO

3.8.1 ERwin 3.0

Complete-Compare es una tecnología de Logic Works que cambia la forma de modelar la interacción de modelos o bases de datos. Provee de una comparación comprensiva de todas las diferencias, resaltando cualquier discrepancia, los cambios pueden migrarse de un modelo-a- base de datos o de una base de datos-a-un modelo, así como también integra la base de datos en el desarrollo del proceso de la aplicación, cuando se ligan las herramientas de desarrollo, sincroniza el Back - End de la base de datos con las formas del Front - End.

3.8.2 ER/Studio 2.5

ER/Studio ayuda a prolongar la inversión que se ha hecho. Soporta el proceso de diseño interactivo inherente en el ciclo de vida de la aplicación.

3.8.3 System Architect 4.0

System Architect cuenta con la herramienta Screen Painter, con la que se pueden diseñar pantallas y menús para aplicaciones de Windows, y pantallas de caracteres para aplicaciones de COBOL. Los archivos de Windows .DLG y .MNU se pueden generar automáticamente por System Architect, incluyendo posición, leyenda, hot key, orden de tablas, y número de identificación para cada control o elemento de menú incluido en la aplicación original de Windows. Por otra parte, pueden crearse pantallas usando controles estándar de Windows, incluyendo botones de presión, cajas de revisión, botones de opciones, cajas combo, cajas de listas, cajas de texto. Se pueden generar archivos de diálogos (.DLG), encabezado

(.H), y forma de Visual Basic (.FRM) para aplicaciones de Windows, usa una rutina simple de captura para crear un nuevo menú de cualquier menú de aplicación de Windows, dibujar menús usando los elementos del menú, submenú y separadores, agregar accesos directos para elementos del menú y submenú, los elementos del submenú y del menú se activan usando el cursor, los diálogos seleccionados se abren de elementos seleccionados del menú.

3.8.4 Power Designer 6.1

MetaWorks es un sistema diseñado para proveer los módulos gráficos de Power Design con la habilidad de compartir y almacenar modelos de datos en un solo punto de control, el Diccionario MetaWorks se ejecuta en una PC y almacena los modelos de datos en un servidor de bases de datos, que puede ser Sybase, SQL Anywhere o cualquier otro como Oracle, Informix, DB2, MS SQL Server y CA OpenIngres. MetaWorks provee de tres funciones principales: Data Model y Submodel Extraction/Consolidation, Project (or Dictionary) Management, y Environment Administration. El MetaBrowser presenta vista de árbol en una línea jerárquica de la aplicación bajo el estudio (base de datos, proyecto, modelo, objeto, y submodelo), expande o colapsa vista de objetos, crea, modifica, borra o imprime objetos seleccionados, habilita comparación entre modelos del mismo tipo, en el nivel de objeto, trabaja con listas de objetos a través de cualquier proyecto, modelo o submodelo.

3.9 WEB

3.9.1 ERwin 3.0

Logic Works Net Results Suites integra el modelado de datos ERwin con las mejores herramientas de desarrollo actuales, lo que provee de una solución

integral para diseñar aplicaciones de Web y el servidor de base de datos Back - End. ERwin / NetDynamics Suite combina el ambiente de desarrollo de aplicación NetDynamics Java-based, ayuda a que los desarrolladores construyan rápidamente aplicaciones de Web de alto desempeño.

3.9.2 ER/Studio 2.5

Publicación automática en el Web. ER/Studio puede documentar automáticamente un diagrama entero, generando un conjunto integrado de reportes HTML sofisticados que múltiples usuarios pueden compartir en Internet.

3.9.3 Power Designer 6.1

Genera reporte en archivo tipo HTML.

3.10 METODOLOGÍA

3.10.1 ERwin 3.0

Soporta metodología para diagramas de Relación-Entidad, modelos de datos IDEF1, y Yourdon/DeMarco.

3.10.2 ER/Studio 2.5

Soporta metodología de Yourdon, con diagramas relación-entidad y modelos IDEF1.

3.10.3 System Architect 4.0

El editor de reporte System Architect también incluye más de 125 reportes predefinidos, incluyendo reportes de metodología para Diagramas de Relación-Entidad, modelos de datos IDEF1, Diagramas de Flujo de Datos para Word & Mellor, Gane & Sarson y Yourdon/DeMarco.

3.10.4 Power Designer 6.1

Soporta Modelos Funcionales y Notaciones de Diagramas de Flujo Modelo Funcional de Objeto (OMT) Yourdon/DeMarco Gane & Sarson SSADM (Análisis de sistema estructurado y metodología de diseño, Structured System Analysis & Design Methodology).

3.11 REPORTES

3.11.1 ERwin 3.0

Los modelos de datos de ERwin pueden editarse, verse e imprimirse de diferentes maneras. Se incluye un RPTwin, que es un generador de reportes gráfico y fácil de usar y un Report Browser built-in con opciones predefinidas de reportes, que proveen un control completo sobre la apariencia y contenido de los reportes.

Además, una sola plantilla de interfaz permite aplicar un estándar de diseño y desplegar preferencias para todos los modelos.

3.11.2 ER/Studio 2.5

Calidad de presentación en los reportes. Además de los reportes de HTML, ER/Studio puede generar reportes de alta calidad con un formato de texto amplio que esta disponible para presentaciones profesionales.

3.11.3 System Architect 4.0

El editor de reportes System Architect es un elemento estándar, esta herramienta permite especificar reportes personalizados para el proyecto con un front-end fácil de usar. El editor de reporte System Architect también incluye más de 125 reportes pre-definidos, incluyendo reportes de metodología para Diagramas de Relación-Entidad, modelos de datos IDEF1x, Diagramas de Flujo de Datos para Ward & Mellor, Gane & Sarson y Yourdon/DeMarco, reportes definidos por el usuario, los reportes pre-escritos, también pueden ser modificados con el editor de reportes GUI, si un nuevo reporte es necesario el reporte de GUI provee de una forma fácil y rápida de crear reportes personalizables.

Todos los reportes pueden incluir una imagen del diagrama relevante, el editor de reportes GUI también provee de flexibilidad agregada para controlar las letras, bordes, orientación, encabezados, y más. Cuando el reporte está listo crea un archivo ASCII y/o RTF en lugar del reporte impreso, se usa el Editor de Reportes System Architect para:

- Reportar propiedades definidas por el usuario.
- Crear reportes de matrices para analizar los datos del repositorio.
- Crear algo específico, leer, actualizar, borrar (CDRU).
- Desempeñar Análisis de Afinidad.

3.11.4 Power Designer 6.1

Creación flexible de reportes estructurados a través de plantilla de reportes. Estructura de árbol de elementos seleccionados para facilitar la organización. Objetos Drag – And - Drop con estructura de árbol para facilitar los ajustes, salva plantillas de reportes, vista previa del reporte antes de imprimirlo, seleccionar un lenguaje por omisión para el reporte, dirigir la impresión o exportar a Microsoft Word, Word Perfect, PageMaker, etc.

Capítulo 4

Casos de Uso

En este capítulo se presentan tres organizaciones que han hecho uso de las herramientas CASE en algún momento, se presentan antecedentes de la organización así como las ventajas y desventajas que han tenido o tuvieron al hacer uso de estas herramientas

4.1 GRUPO UNIÓN FENOSA MÉXICO¹⁸

UNION FENOSA es una compañía española de gas y electricidad presente en 12 países.

UNION FENOSA presta servicios a casi 8,5 millones de clientes en todo el mundo, que consumen 50.000 millones de kWh eléctricos y más de 38.000 millones de kWh de gas.

La potencia total instalada por el Grupo para atender las necesidades de sus clientes asciende a 9.952 MW. El 27,4% del total de esta capacidad de generación se opera fuera de España.

En el ámbito internacional, el Grupo es una empresa de referencia en la producción y suministro de electricidad en Iberoamérica. Cuenta además con importantes intereses en el negocio gasista y participa en el desarrollo de grandes infraestructuras de gas en Egipto y Omán.

A 31 de diciembre de 2005, UNION FENOSA era la decimoquinta mayor empresa del IBEX 35 con una capitalización de 9.576 millones de euros. A lo largo del año el reconocimiento al desarrollo del Plan Estratégico se tradujo en una mejora de la cotización del 62%.

UNION FENOSA ha alcanzado el mayor beneficio de su historia superando los 800 millones de euros. El excelente comportamiento operativo de todos sus negocios y las plusvalías por la venta parcial de sus intereses en el sector de las telecomunicaciones han impulsado este resultado.

¹⁸ [uef.es,2004]

En UNION FENOSA trabajan 16.893 profesionales, de los cuales el 51,2% prestan sus servicios fuera de España. Los recursos humanos son el pilar fundamental para el desarrollo y la consolidación del Grupo, y donde reside todo el conocimiento acumulado en su proceso de transformación.

UNION FENOSA impulsa las mejores prácticas internacionales en la gestión de infraestructuras energéticas, armonizando sus actuaciones y el desarrollo del talento de sus profesionales con la preocupación por el medio ambiente y el desarrollo sostenible.

UNION FENOSA EN MÉXICO dispone de tres centrales de ciclo combinado en México ver Figura 1.16, con una capacidad total instalada de 1.550 MW. El parque de generación de UNION FENOSA, formado por las plantas de Hermosillo, Naco Nogales y Tuxpan, cubre aproximadamente el 3,1% de la demanda eléctrica del país, lo que posiciona a UNION FENOSA como el tercer productor independiente de energía de México.



Figura 1.16: Centrales de Ciclo Combinado en México

La central de Hermosillo, en el estado de Sonora, se encuentra en operación desde 2001. Las centrales de Tuxpan y Naco Nogales están en operación desde 2003.

Todas las centrales cuentan con contratos de compraventa de capacidad y energía eléctrica firmados con la Comisión Federal de Electricidad (CFE) por un periodo de 25 años. En el año 2005 UNION FENOSA concluyó la implantación para todas sus instalaciones de producción de México de los Sistemas de Gestión Medio Ambiental y de Calidad, certificados por AENOR, según las normas ISO 14001 e ISO 9001, respectivamente.

En UNION FENOSA EN MÉXICO no es utilizada ninguna herramienta CASE, dado que la experiencia que han tenido con este tipo de herramientas no ha sido del todo satisfactoria.

Aunque estas herramientas permiten generar aplicaciones rápidamente, no responden adecuadamente a las exigencias de rendimiento y escalabilidad que impone las necesidades de UNION FENOSA.

Además de que existe un efecto de “Caja Negra” que hace que el código generado automáticamente por las herramientas CASE quede fuera de su control.

Por las razones anteriores, en UNION FENOSA y en su sucursal de servicios profesionales SOLUZIONA, utilizan una metodología de desarrollo y conjunto propio de herramientas de ayuda al desarrollo, aunque las herramientas se basan en RATIONAL ROSE + ECLIPSE cuentan con generadores automáticos de código para persistencia, objetos de negocios e interface de usuario.

4.2 Centro de Cómputo Académico de la UAEH en el área de Desarrollo de Software

Su principal actividad es la automatización y sistematización de las actividades realizadas dentro del Centro de Cómputo Académico y las Dependencias de la UAEH que así lo soliciten, para poder facilitar la labor de cualquier área, minimizando tiempos y esfuerzos de actividades, así también optimiza los recursos informáticos y genera las herramientas necesarias para el manejo y control de la información. Dando al mismo tiempo nuevos trazos hacia la mejora continua y un control de calidad que cumpla con las expectativas de quien haga solicitud de sus servicios.

En el Centro de Computo Académico de la UAEH en el área de Desarrollo de Software utilizan herramientas CASE, desde hace cerca de 6 años, las herramientas CASE que utilizan son BPwin, Erwin, Embarcadero 6.0.

Dado a la experiencia que han tenido con este tipo de herramientas han encontrado diversas ventajas como son:

- **Análisis del sistema:** Plasman todos los procesos administrativos, permitiéndoles dar a conocer de forma mas sencilla y explicita lo procesos necesarios para los proyectos que se están desarrollando.
- **Costos:** Al hacer uso de estas herramientas se han ahorrado el gasto de contratar mas personal para hacer el diseño de los sistemas, ahorran tiempo y cuando hay algún error no tienen que volver a realizar todo, sino solo modificar el modulo en el cual se hayan equivocado o sea necesario modificar.
- Pueden realizar el modelo Entidad-Relación para cualquier gestor de bases de datos, creando a su vez el diccionario de datos.

Hasta este momento no han tenido ningún problema con estas herramientas, por lo cual no encuentran ninguna desventaja.

Han hecho uso de estas herramientas en diversos proyectos como son:

- Sistema Integral Administrativo.
- El sistema de Encuestas.
- La Evaluación de Cursos.
- Y actualmente se esta trabajando en el Sistema de Evaluación de la Dirección Universitaria de Idiomas.

Utilizando la plataforma de SQL Server 2000 y Delphi de la versión 5 a la 7.

4.3 IDS S.A de C.V¹⁹

Es una empresa mexicana que cuenta con una exitosa trayectoria desde 1982 en la práctica constante y especializada en servicios de consultoría, desarrollo y capacitación en tecnologías de información. Su crecimiento corporativo ha sido constante por arriba, incluso del promedio de la industria en términos de ingresos y personal. Tienen participación en un gran número de proyectos dentro de los sectores financiero, asegurador, comercial, manufactura, telecomunicaciones, servicios y gobierno, estableciendo relaciones de largo plazo con nuestros clientes a través de sociedades tecnológicas y estratégicas que les aportan beneficios y contribuciones tangibles.

En IDS han desarrollado con éxito más de 500 grandes proyectos, que, por su tamaño o complejidad, son referencia en la industria. Las soluciones desarrolladas abarcan las principales plataformas y tecnologías que existen en el mercado. Para concretar estos logros han invertido mas de siete millones de horas hombre de

¹⁹ [ids, 2005]

trabajo, que se traducen en una gran experiencia; valor que distingue a IDS en el mercado.

Con el fin de alcanzar y consolidar su liderazgo, han cimentado su estrategia de operación en dos activos principales: “clientes” y “su gente”, ambos parte integral de su misión, visión, valores y principios.

Misión

Ser una empresa en tecnologías de información rentable y en constante crecimiento. Que busca ser reconocida y respetada en el medio por su calidad, eficiencia, espíritu de servicio y sobre todo, por su compromiso con sus clientes, al punto de dejarlos a todos ampliamente satisfechos y referenciables. Ser un lugar de trabajo agradable, donde se respete al individuo y se trabaje en equipo; donde abunde el reto, la superación, el reconocimiento y donde se ofrezcan excelentes oportunidades de desarrollo personal, técnico, profesional y económico.

Visión

Ser socios, proveedores y aliados tecnológicos de nuestros clientes ofreciendo servicios de soluciones integrales en tecnologías de información a través de intervenciones proactivas y con un rol estratégico.

Valores y Principios

Nuestro éxito está fundamentado en la calidad, compromiso y ética de cada uno de los que integramos ids. Nuestra gente comparte valores y principios que, al ponerlos en práctica le dan solidez y rumbo hacia fines comunes.

En IDS S.A. de C.V utilizan herramientas CASE, desde hace cerca de 4 años, las herramientas CASE que utilizan son Rational Rose.

Debido a la experiencia que han tenido en el uso de esta herramienta encontraron las siguientes ventajas:

- Facilitar el desarrollo de sistemas, modularizando los desarrollos, y definiendo tiempos de desarrollo.
- Mejoramiento en la lógica funcional de los sistemas.
- Mejor diseño de los sistemas en base a las mejores prácticas de desarrollo hechas por diversas empresas.
- Permite un mejor entendimiento de los sistemas entre las personas involucradas permitiendo tener un mejor enfoque de que se quiere.

De igual forma han encontrado algunas desventajas como son:

- Requieren de capacitación para su uso.
- Es necesaria tener experiencia en el diseño de sistemas para poder sacar el mejor provecho de dichas herramientas.
- Las mejores herramientas son muy costosas, reduciendo el acceso de personal calificado a dichas herramientas así como su utilización en proyectos de software.

Conclusiones

Entre los beneficios ofrecidos por la tecnología CASE podemos encontrar:

- Soporte para el desarrollo de prototipos de sistemas.

El desarrollo de prototipos toma varias formas, en ocasiones se desarrollan diseños para pantallas y reportes con la finalidad de mostrar la organización y composición de los datos, encabezados y mensajes, los ajustes necesarios al diseño se hacen con rapidez para alterar la presentación y las características de la interface. Sin embargo, no se prepara el código fuente, de naturaleza orientada hacia procedimientos, como una parte del prototipo.

El desarrollo de prototipos puede producir un sistema que funcione. Las características de entrada y salida son desarrolladas junto con el código orientado hacia los procedimientos y archivos de datos.

Muchas herramientas CASE soportan las primeras etapas del desarrollo del prototipo y, otras brindan apoyo durante todo el proceso de desarrollo del prototipo. Las que proporcionan la capacidad para generar código soportan de hecho todo proceso, ya que el código puede ser generado al inducir la actividad de generación después de cambiar las especificaciones o requerimientos.

- Generación de código

Algunas herramientas CASE tienen la capacidad de producir el código fuente. La mejor ventaja es la disminución del tiempo necesario para preparar un programa.

La generación del código asegura una estructura estándar y consistente para el programa y disminuye la salida de varios tipos de errores, mejorando de esta manera la calidad.

La generación del código permiten volver a utilizar el software y las estructuras estándares para generar dicho código. Ninguna de las herramientas que existen en el presente es capaz de generar un código completo.

- Mejora en la habilidad para satisfacer los requerimientos del usuario

Las herramientas CASE disminuyen el tiempo de desarrollo, una característica importante para los usuarios, éstas también afectan la naturaleza y cantidad de interacción entre los encargados del desarrollo y el usuario. Las descripciones gráficas y los diagramas, así como los prototipos de reportes y la composición de las pantallas, contribuyen a un intercambio de ideas más efectivo.

- Soporte interactivo para el proceso de desarrollo

Las herramientas CASE soportan pasos interactivos al eliminar el tedio manual de dibujar diagramas, elaborar catálogos y clasificar. Como resultado de esto, se anticipa que los analistas repasarán y revisarán los detalles del sistema con mayor frecuencia y en forma más consistente.

- Facilidad para la revisión de aplicaciones esto nos ahorra más tiempo.

Las herramientas CASE proporcionan un beneficio substancial para las organizaciones al facilitar la revisión de las aplicaciones, contar con un depósito central agiliza el proceso de revisión ya que éste proporciona bases para las definiciones y estándares para los datos. Las capacidades de generación interna, si se encuentran presentes, ya que contribuyen a modificar el sistema por medio de las especificaciones más que por los ajustes al código fuente.

Las herramientas CASE tienen puntos débiles significativos, que van desde la confiabilidad en los métodos estructurados hasta su alcance limitado, los cuales amenazan con minar los beneficios potenciales descritos con anterioridad.

- Falta de niveles estándar para el soporte de la metodología

Aún no aparece un conjunto “estándar” de herramientas CASE, por tanto, se debe tener precaución al seleccionar una herramienta de este tipo.

Las herramientas CASE que existen en el presente, tienen una de las siguientes características:

- Son independientes de la metodología.
- Permiten que los usuarios definan sus propias metodologías.
- Soportan una metodología.
- Soportan las metodologías más diseminadas.

En todas ellas existen ciertos compromisos, las herramientas que son independientes de la metodología, no pueden fomentar el uso de las reglas y estándares de la misma, éstas quizá proporcionen los componentes de una metodología por ejemplo: diagramas de flujos de datos, un diccionario de datos y facilidades para la descripción de procesos; pero no el marco de referencia, reglas y procedimientos que en realidad constituyen el núcleo de la metodología. Aunque se puede llevar a cabo acciones básicas para la validación de diseños y diagramas para detectar componentes faltantes, éstas son sólo funciones mecánicas, por otra parte, esta clase de herramientas no puede proporcionar ayuda metodológica o pedir al usuario que realice tareas necesarias para la metodología que aún esta sin terminar.

Estas herramientas mejoran la productividad al efectuar tareas tediosas y de documentación, aunque ellas no puedan asegurar buenos resultados. Desde el punto de vista funcional, las capacidades que brindan para garantizar la calidad son mínimas.

- Conflictos en el uso de los diagramas

Las herramientas difieren en el uso que hacen los diagramas. Algunas son herramientas exclusivamente para gráficas, que se abocan al dibujo de diagramas para el análisis de entrada y salida de datos; este tipo de herramientas puede restringir ya sea el proceso de desarrollo normal seguido por una organización o el estilo particular de trabajo de los analistas.

Otros vendedores de herramientas consideran los diagramas como documentación y aceptan entradas por medio de formas o lenguajes de especificación y, en ocasiones, en forma gráfica; por tanto, se debe tener cuidado cuando se selecciona una herramienta para apoyar los métodos existentes en una organización.

- Diagramas no utilizados

Los productos CASE emplean gráficas para modelar y generar informes sobre el análisis y desarrollo de sistemas. Una de las afirmaciones de los vendedores de herramientas es que las presentaciones gráficas y la documentación mejoran la comunicación entre los miembros del equipo de desarrollo, propician una calidad mayor de la entrada proporcionada por el cliente y mejoran la productividad de desarrollo de software; sin embargo los investigadores han encontrado que, en algunos casos, las herramientas gráficas, automatizadas o manuales, no se emplean del todo, o tal vez, no se utilicen en la forma que deberían emplearse. Por otra parte, algunos analistas prefieren para algunas tareas un lenguaje estructurado o descriptivo.

Muchos profesionales de los sistemas de información no hacen uso de herramientas gráficas en el desarrollo de software; más bien las emplean para automatizar la producción de informes y documentación del sistema, como los diagramas de flujo utilizados por los programadores para documentar un programa una vez terminado.

- Función limitada

Aunque una herramienta puede apoyar varias fases del ciclo de vida de desarrollo de sistemas o adaptarse a diferentes metodologías de desarrollo, por lo general, su enfoque principal está dirigido hacia una fase o método específico. Por ejemplo, los encargados de desarrollar un nuevo producto pueden afirmar que éste apoya todo el proceso de análisis y diseño, sin embargo, las capacidades de comprobación y verificación de errores del producto quizá sean más rigurosas ya sea en el área de análisis o en la de

diseño, pero no en ambas. Algunos productos están dirigidos hacia el diseño de bases de datos para la organización y desarrollo de aplicaciones que giren en su entorno, omitiendo el soporte para pantallas de presentación visual, los informes sobre requerimientos o las necesidades de seguridad, otros tienen la capacidad de generar el código hacen mayor hincapié en el desarrollo de prototipos como el principal método de desarrollo de sistemas de información. Muchas herramientas para la fase de desarrollo recalcan el mantenimiento y la reestructuración del código, pero ofrecen un soporte débil durante la fase de análisis para la determinación y especificación de requerimientos.

- Alcance limitado

Aunque muchas herramientas basadas en computadoras incluyen la capacidad de verificar las especificaciones para determinar su complementos o consistencia, virtualmente no llevan a cabo ningún análisis de los requerimientos de la aplicación, por tanto, el alcance de las actividades de desarrollo asociado con las herramientas existentes es bastante limitado.

La mayor parte de productos CASE describe (documenta) pero no analiza. De poca ayuda es proporcionar una regla de inclusión en los mejores enfoques y una regla de exclusión para los que son poco satisfactorios. No ofrecen o evalúan, soluciones potenciales para los problemas relacionados con sistemas. Y tampoco existe una garantía clara para que dos analistas que utilicen los mismos métodos aplicados a información idéntica, formulen recomendaciones igualmente aceptables.

La herramienta CASE ERwin es la que ofrece muchas ventajas de acuerdo a las características que presenta, ERwin es la herramienta más sencilla, rápida y ligera para correr, pues está basada en un componente que sólo cubre la base de datos.

Aunque finalmente, ésta decisión dependerá de las necesidades del usuario final o desarrollador del sistema, quien sin lugar a dudas determinará que herramienta puede asistirle en su trabajo. Es decir, tiene libertad de seleccionar herramientas de desarrollo que están disponibles en el mercado de acuerdo a gustos y necesidades.

Por otra parte, hay que considerar que en algunos se obtienen elevados costos tanto en la adquisición de herramientas y costos de entrenamiento de personal, como a la falta de adaptación de tal herramienta a la arquitectura de la información y a metodologías de desarrollo utilizadas por la organización. Además de que, algunas herramientas CASE no ofrecen o evalúan soluciones potenciales para los problemas relacionados con sistemas o virtualmente no llevan a cabo ningún análisis de los requerimientos de la aplicación.

Glosario de términos

4LG o L4G: Lenguaje de Cuarta Generación.

CAD: Diseño asistido por computadora, abreviado DAO pero más conocido por las siglas inglesas CAD (Computer Aided Design), se trata básicamente de una base de datos de entidades geométricas (puntos, líneas, arcos, etc.) con la que se puede operar a través de una interfaz grafica. Permite diseñar en dos o tres dimensiones mediante geometría alámbrica, esto es, puntos, líneas, arcos, splines, superficies y sólidos para obtener un modelo numérico de un objeto o conjunto de ellos.

CAJA NEGRA: Saber que entra y que sale (la interfaz de módulo) y no como lo hace (su implementación).

CAM: Fabricación asistida por computadora, también conocida por las siglas inglesas CAM (Computer Aided Manufacturing), hace referencia al uso de un extenso abanico de herramientas basadas en los ordenadores que ayudan a ingenieros, arquitectos y otros profesionales dedicados al diseño en sus actividades.

CASE: Acrónimo ingles de Computer Aided Software Engineering, que viene a significar Ingeniería de Software Asistida por Computadora. Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación de parte del código automática, documentación o detección de errores entre otras

HERRAMIENTA CASE: Una herramienta del software que automatiza (por lo menos en parte) una parte del ciclo de desarrollo de software.

KIT de HERRAMIENTAS CASE: un conjunto de herramientas CASE integradas que se han diseñado para trabajar juntas y automatizar o proveer ayuda automatizada al ciclo de desarrollo de software, incluyendo el análisis, diseño, codificación y pruebas.

METODOLOGIA CASE: Una automatizable metodología estructurada que define una disciplina e ingeniería como un acercamiento a todos o algunos aspectos del desarrollo y mantenimiento de software

PARADIGMA: Un paradigma esta constituido por los supuestos teóricos generales, las leyes y las técnicas para su aplicación que adoptan los miembros de una determinada comunidad científica.

PLATAFORMA DE HARDWARE PARA CASE: Una arquitectura de hardware con uno, dos o tres sistemas puestos en línea, que proveen una plataforma operativa para las herramientas CASE.

PROTOTIPO: Un prototipo es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar su uso, probarlo en situaciones reales y explorar su uso.

Un prototipo puede ser cualquier cosa, desde un trozo de papel con sencillos dibujos a un complejo software.

PUESTO DE TRABAJO PARA CASE: Una estación de trabajo técnica, diseñada a 32 bits o computadora personal equipada con herramientas CASE que automatizan varias funciones del ciclo.

RDB: Herramienta para el intercambio de información entre ficheros ISO2709 y varios tipos de formatos RDB. Permite la exportación entre ISO2709 y RDB, la

extracción selectiva de información desde tablas RDB, la reunión de información de varias bases en una sola, la eliminación de registros duplicados, y la visualización del contenido de tablas RDB

SGBD: Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

SGBDOO: Sistemas Gestores de Bases de Datos Orientada a Objetos.

SGBDR: Sistemas Gestores de Bases de datos Relacional.

SISTEMA CASE: Un conjunto de herramientas CASE integradas que comparten una interface de usuario común y corren en un ambiente computacional común.

TECNOLOGIA CASE: Una tecnología de software que mantiene una disciplina de la ingeniería automatizada para el desarrollo de software, mantenimiento y dirección de proyecto, incluye metodologías estructuradas y herramientas automatizadas.

Bibliografía

[Cuevas, 1991], Cuevas Agustín, Gonzalo, “Ingeniería del Software. Practica de la programación (Programación estructurada basada en la metodología WARNIER)”, Editorial RA-MA, 1ª impresión, 1991.

[Dirección, 2006], Dirección General de Servicios de Cómputo Académico, “Herramientas CASE para el Análisis y Diseño Estructurado.”, Subdirección de Sistemas – DCCA – UNA, 2006.

[James, 2001], James Wetherber, Efraim Turban, Mc Clean, Ephraim, “Tecnologías de Información para la Administración.”, CECSA, 1ª edición en Español, 2001.

[López-Fuensalida, 1990], López – Fuensalida, Antonio, “Metodologías de desarrollo. (En el camino hacia el CASE)”, Editorial RA-MA, 1ª impresión, 1990.

[McClure, 1992], McClure, Carma, “CASE. La automatización del Software”, Editorial RA-MA, 1ª edición, 1992.

[Piattini, 2004], Piattini, M., Calvo – Manzano, J., Cervera, J. y Fernández, L. “Análisis y Diseño de Aplicaciones Informaticas de Gestion: Una perspectiva de Ingenieria de Software”, Editoria RA-MA, 2004.

[S. Pressman, 1998], S. Pressman, Roger, “Ingeniería del Software. Un enfoque práctico”, Editorial Mc. Graw-Hill, 4ª edición, 1998.

[Sub – Jefatura, 2006], Sub – Jefatura de Informática, “Herramientas CASE”, Instituto Nacional de Estadística e Informática, 2006.

[Yourdon, 1993], Yourdon, Edgard, "Análisis Estructurado Moderno", Editorial Prentice Hall Hispanoamericana, S.A, 1ª edición, 1993

Referencias Electrónicas

[ids, 2005], <http://www.ids.com.mx/acerca.html>, © IDS Comercial 2005

[uef,2004], http://www.uef.es/webuf/ShowContent.do?contenido=CON_01_01&audiencia=1, © 2004 UNION FENOSA, S.A. Seguridad e información legal

[uef.es,2004], http://www.uef.es/webuf/ShowContent.do?contenido=CON_14_04_01&audiencia=, © 2004 UNION FENOSA, S.A. Seguridad e información legal