



Herramientas Case

COLECCION CULTURA INFORMATICA

INSTITUTO NACIONAL DE ESTADISTICA E INFORMATICA©

Elaborado por la Sub-Jefatura de Informática

Dirección Técnica de Desarrollo Informático

Teléfono 433-4223 - Anexos 181 - 315

Telefax 433-5568

INTERNET postmast@inei.gob.pe

Impreso en los Talleres de la Oficina de Impresiones de la Oficina Técnica de Difusión Estadística y Tecnología Informática del Instituto Nacional de Estadística e Informática (INEI)

Edición

: 300 Ejemplares

Domicilio, Redacción y Talleres

: Av. Gral. Garzón N° 658 - Jesús María

Noviembre de 1999

875-99-OI-OTDETI-INEI

Presentación

El Instituto Nacional de Estadística e Informática (INEI), como ente rector del sistema Nacional de Informática, continuando con la publicación de la Colección "Cultura Informática", presenta en esta oportunidad su Vigésimo Segundo Número titulado: **"Herramientas Case"**.

La presente publicación trata temas como qué son las Herramientas Case, cuales son sus beneficios, que procesos se debe seguir para adoptar estas Herramientas en las organizaciones, qué puntos deben considerarse para seleccionar una Herramientas Case, cual es el futuro de las Herramientas Case, entre otros.

El Instituto Nacional de Estadística e Informática, pone a disposición de sus lectores la presente publicación, esperando sea de su agrado y utilidad.

Econ. Félix Murillo Alfaro
JEFE
INSTITUTO NACIONAL DE ESTADISTICA
E INFORMATICA

Índice

1.	<i>Herramientas CASE:</i> <i>El mejor soporte para el proceso de desarrollo de Software</i>	5
2.	<i>Qué son las Herramientas CASE</i> <i>Glosario de Definiciones Básicas de CASE</i>	7 10
3.	<i>Historia de las Herramientas CASE</i> <i>Evolución de las Herramientas CASE</i>	11 13
4.	<i>Clasificación de las Herramientas CASE</i> <i>Rango de las Herramientas CASE</i>	15 23
5.	<i>Componentes y Funcionalidades de una Herramienta CASE</i>	25
6.	<i>Ejemplos de Herramientas CASE más utilizadas</i>	29
7.	<i>Adopción de Herramientas CASE en las Organizaciones</i>	41
8.	<i>Puntos Importantes a considerar cuando se selecciona una Herramienta CASE</i> <i>Estrategias de Implantación de una Herramienta CASE</i> <i>Consideraciones Importantes</i>	47 49 50
9.	<i>Tendencias Tecnológicas y del Mercado de las Herramientas CASE</i>	51

1. Herramientas Case

El mejor soporte para el proceso de desarrollo de software

Las computadoras afectan nuestras vidas nos guste o no. Utilizamos computadoras en nuestra vida diaria, la mayor parte del tiempo sin reconocer conscientemente que estamos haciéndolo. Las utilizamos en aplicaciones domésticas como microondas, televisión, vídeo casseteras o fuera de nuestras casas en máquinas para tarjetas de crédito, por ejemplo.

La verdad es que no podemos escapar de las computadoras.



El rápido incremento en performance de las computadoras junto al dramático decremento en tamaño y costo, dio como resultado una explosión de tecnología, generándose una larga variedad de aplicaciones que éstas pueden soportar.

Desde el inicio de la escritura de software, ha existido un conocimiento de la necesidad de herramientas automatizadas para ayudar al diseñador del software. Inicialmente, la concentración estaba en herramientas de apoyo a programas como traductores, recopiladores, ensambladores, procesadores de macros, y montadores y cargadores.

Este conjunto de aplicaciones que pueden informatizarse, aumentó dramáticamente en un breve espacio de tiempo, causando una gran demanda por nuevo software a desarrollar. A medida que se escribía nuevo software, habían ya en existencia millones y millones de líneas de código que necesitaban se mantenidas y actualizadas.

Esto causó a la industria de las computadoras muchos problemas, no podía cubrir el incremento de la demanda con los métodos que se estaban usando. Esto fue reconocido como una crisis de software. Para superar este problema en el proceso de desarrollo de software, se introdujeron metodologías para intentar crear estándares de desarrollo.

Hay también otra manera en la que la industria se ha ayudado a superar las dificultades de uso de esta tecnología disponible. La industria de computadoras ha desarrollado un soporte automatizado para el desarrollo y mantenimiento de software. Este es llamado Computer Aided Software Engineering (CASE).



2. Qué son las Herramientas CASE



Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Como es sabido, los estados en el Ciclo de Vida de desarrollo de un Software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.

CASE se define también como:

- *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.*
- *La sigla genérica para una serie de programas y una filosofía de desarrollo de software que ayuda a automatizar el ciclo de vida de desarrollo de los sistemas.*
- *Una innovación en la organización, un concepto avanzado en la evolución de tecnología con un potencial efecto profundo en la organización. Se puede ver al CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales.*

Variaciones en el significado de CASE.

Computer		
Aided	Assisted	Automated
Software	Systems	
Engineering		

La realización de un nuevo software requiere que las tareas sean organizadas y completadas en forma correcta y eficiente. Las Herramientas CASE fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software.

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



También permite a las compañías 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.

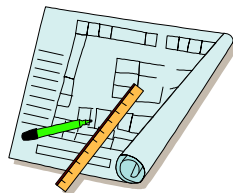


Las herramientas CASE también permiten a los analistas tener más tiempo para el análisis y diseño y minimizar el tiempo para codificar y probar.

La introducción de CASE integradas está comenzando a tener 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.*

La principal ventaja de la utilización de una herramienta CASE, es la mejora de la calidad de los desarrollos realizados y, en segundo término, el aumento de la productividad. Para conseguir estos dos objetivos es conveniente contar con una organización y una metodología de trabajo, además de la propia herramienta.

La mejora de calidad se consigue reduciendo sustancialmente muchos de los problemas de análisis y diseño, inherentes a los proyectos de mediano y gran tamaño (lógica del diseño, coherencia, consolidación, etc.). La mejora de productividad se consigue a través de la automatización de determinadas tareas, como la generación de código y la reutilización de objetos o módulos.



Glosario de Definiciones Básicas de CASE:

CASE: *Ayuda por Computadora a la Ingeniería de Software.*

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

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

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

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

PUESTO DE TRABAJO para CASE: *Una estación de trabajo técnica, diseñada a 32 bits o computadora personal equipada con Herramientas Case que automatiza varias funciones del ciclo.*

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.*

3. Historia de las Herramientas CASE

Las Herramientas CASE tienen su inicio con el simple procesador de palabras que fue usado para crear y manipular documentación. Los setentas vieron la introducción de técnicas gráficas y diagramas de flujo de estructuras de datos. Sobre este punto, el diseño y especificaciones en forma pictórica han sido extremadamente complejos y consumían mucho tiempo para realizar cambios.

La introducción de las herramientas CASE para ayudar en este proceso ha permitido que los diagramas puedan ser fácilmente creados y modificados, mejorando la calidad de los diseños de software. Los diccionarios de datos, un documento muy usado que mantiene los detalles de cada tipo de dato y los procesos dentro de un sistema, son el resultado directo de la llegada del diseño de flujo de datos y análisis estructural, hecho posible a través de las mejoras en las Herramientas CASE.



Pronto se reemplazaron los paquete gráficos por paquetes especializados que habilitan la edición, actualización e impresión en múltiples versiones de diseño. Eventualmente, las herramientas gráficas integradas con diccionarios de base de datos para producir poderosos diseños y desarrollar herramientas, podrían sostener ciclos completos de diseño de documentos.



Como un paso final, la verificación de errores y generadores de casos de pruebas fueron incluidos para validar el diseño del software. Todos estos procesos pueden saberse integrados en una simple herramienta CASE que soporta todo el ciclo de desarrollo.

La primera herramienta comercial se remonta a 1982, aunque algunos especialistas indican que algunos ejemplos de herramientas para diagramación ya existían.

No fue sino 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 beneficiadas por la ayuda de las CASE. Estos beneficios consistían, por ejemplo, en el aumento en la productividad. El objetivo en 1985 para muchos vendedores era producir software más rápidamente. Las herramientas del CASE serían una familia de métodos favorablemente estructurados para planeamiento, análisis y diseño. Esto llevaría a la generación automática de código para desarrollo de software vía una especificación formalmente diseñada. Esto traería como beneficio:

- ☒ *Una mejora en la calidad, fiabilidad, utilidad y rendimiento.*
- ☒ *El entorno de producción de documentación para software mejora la comunicación, mantenimiento y actualización.*
- ☒ *Hace el trabajo de diseño de software más fácil y agradable.*
- ☒ *La promesa futura de reemplazar realmente a los ingenieros de software especializados.*
- ☒ *Reducción del costo de producción de software.*

Con estos objetivos en mente, la industria destinó millones en producción de Herramientas CASE.

Así como esta enorme suma de dinero fue gastada en Herramientas CASE, hubo también trabajo de investigación a nivel mundial en diferentes instituciones como Universidades, Instituciones Gubernamentales y de Defensa. La industria de Herramientas CASE está creciendo y esta tomando cada vez mayor importancia.

Evolución de las Herramientas CASE

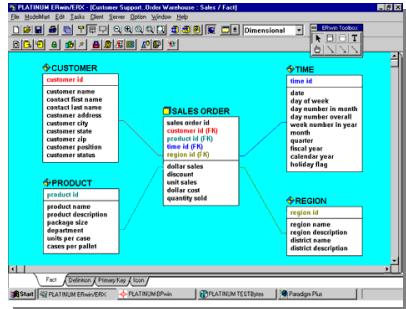
<i>A inicios de los 80's</i>	:	<i>Ayuda en la documentación por computadora. Diagramación asistida por computadora. Herramientas de análisis y diseño.</i>
<i>A mediados de los 80's</i>	:	<i>Diseño automático de análisis y pruebas. Repositorios automáticos de información de sistemas.</i>
<i>Al final de los 80's</i>	:	<i>Generación automática de código desde especificaciones de diseño.</i>
<i>A inicios de los 90's</i>	:	<i>Metodología Inteligente. Interface de Usuario reusable como una metodología de desarrollo.</i>



4. Clasificación de las Herramientas Case

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada. Podrían clasificarse atendiendo a:

- Las plataformas que soportan.
- Las fases del ciclo de vida del desarrollo de sistemas que cubren.
- La arquitectura de las aplicaciones que producen.
- Su funcionalidad.



Las herramientas CASE, en función de las fases del ciclo de vida abarcadas, se pueden agrupar de la forma siguiente:

1. **Herramientas integradas, I-CASE** (Integrated CASE, CASE integrado): abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son llamadas también CASE workbench.
2. **Herramientas de alto nivel, U-CASE** (Upper CASE - CASE superior) o front-end, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.
3. **Herramientas de bajo nivel, L-CASE** (Lower CASE - CASE inferior) o back-end, dirigidas a las últimas fases del desarrollo: construcción e implantación.
4. **Juegos de herramientas o Tools-Case**, son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

Tipo de Case	Ventajas	Desventajas
I – Case	<ul style="list-style-type: none"> ➤ Integra el ciclo de vida. ➤ Permite lograr importantes mejoras de productividad a mediano plazo. ➤ Permite un eficiente soporte al mantenimiento de sistemas. ➤ Mantiene la consistencia de los sistemas a nivel corporativo. 	<ul style="list-style-type: none"> ➤ No es tan eficiente para soluciones simples, sino para soluciones complejas. ➤ Depende del Hardware y del Software. ➤ Es costoso.
Upper Case	<ul style="list-style-type: none"> ➤ Se utiliza en plataforma PC, es aplicable a diferentes entornos, ➤ Menor costo 	<ul style="list-style-type: none"> ➤ Permite mejorar la calidad de los sistemas, pero no mejora la productividad. ➤ No permite la integración del ciclo de vida.
Lower Case	<ul style="list-style-type: none"> ➤ Permite lograr importantes mejoras de productividad a corto plazo. ➤ Permite un eficiente soporte al mantenimiento de sistemas. 	<ul style="list-style-type: none"> ➤ No garantiza la consistencia de los resultados a nivel corporativo. ➤ No garantiza la eficiencia del Análisis y Diseño. ➤ No permite la integración del ciclo de vida.

I - CASE

El I-CASE se concibe como el conjunto de cuatro herramientas que tocan las disciplinas que van desde la estrategia de la empresa, y la concepción del sistema de información, hasta el análisis, diseño y la generación de los mismos programas.



Las herramientas I-CASE se basan en una metodología. Tienen un repositorio y aportan técnicas estructuradas para todas las fases del ciclo de vida. Estas son las características que les confieren su mayor ventaja: una mejora de la calidad de los desarrollos.

Beneficios del I - Case:

<i>Desarrollo Tradicional</i>	<i>Case</i>	<i>I - Case</i>
<i>Énfasis en la codificación y pruebas de programas.</i>	<i>Énfasis en análisis y diseño.</i>	<i>Énfasis en el modelamiento empresarial.</i>
<i>Especificaciones basadas en papel.</i>	<i>Especificaciones basadas en diagramas automatizados.</i>	<i>Especificaciones basadas en diagramas automatizados y perfectamente integrados a través de las diferentes etapas de desarrollo.</i>
<i>Codificación manual de programas.</i>	<i>Generación automática de códigos, programas fuente.</i>	<i>Generación automática de códigos y totalmente integrada con la estación de trabajo de diseño representación gráfica de las estructuras de códigos.</i>
<i>Documentación manual.</i>	<i>Generación automática de documentación .</i>	<i>Generación automática de documentación .</i>
<i>Mantenimiento de programas fuentes.</i>	<i>Mantenimiento de especificaciones de diseño y regeneración de códigos</i>	<i>Mantenimiento de especificaciones de diseño y regeneración de códigos</i>
		<i>Uso de una enciclopedia (coordinador de conocimientos) para concentrar toda la información de las diferentes estaciones de trabajo en forma integrada.</i>

Juego de Herramientas o Tools Case.-

Están formadas por un conjunto de herramientas orientadas cada una de ellas a resolver una determinada fase del desarrollo. El ensamblaje de las mismas se efectúa por medio de una serie de interfaces que adaptan las salidas producidas por cada una de ellas de forma que sirva como entrada en la siguiente.

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

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*
- *Tranforms de Transform Logic*
- *Telon Pansophic System*
- *Decase de DEC COBOL 2/ Worbench de Micro Focus*
- *Snap CASE para AS/400*

Otra posible clasificación, utilizando la funcionalidad como criterio principal, es la siguiente:

- ◆ **Herramientas de planificación de sistemas de gestión.** Sirven para modelizar 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 mejor 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.** 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.
 - Herramientas de análisis y diseño (Modelamiento).
 - Herramientas de creación de prototipos y de simulación.
 - Herramientas 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. Ejemplos de estas herramientas son:
 - Herramientas de codificación convencionales.
 - Herramientas de codificación de cuarta generación.
 - Herramientas de programación orientadas 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:
 - Herramientas de análisis estático.
 - Herramientas de codificación de cuarta generación.

- *Herramientas de programación orientadas 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 puede 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 de la gestión del proyecto, en lugar de proporcionar un soporte global para la actividad de gestión. Utilizando un conjunto seleccionado de las mismas se puede: realizar estimaciones de esfuerzo, coste 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.*

Otra clasificación, diferencia las funciones CASE en cinco grupos:

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, etc. El repositorio es un concepto más amplio que el de diccionario de datos y soporta a los demás grupos de funciones. No es fácil encontrar en el mercado productos Case con funcionalidades estrictamente a las de repositorio, ya que, a pesar de su innegable importancia, tienen un carácter auxiliar de los demás grupos de funciones. Cualquier sistema Case poseerá un repositorio propio o bien, trabajará sobre un repositorio suministrado por otro fabricante o vendedor.

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".

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. Aunque el número de fases puede ser variable en función del nivel de detalle que se adopte, pueden de modo simplificado, identificarse las siguientes:

- *Planeamiento.*
- *Análisis y Diseño.*
- *Implantación (programación y pruebas).*
- *Mantenimiento y actualización.*

Los sistemas Case pueden cubrir la totalidad de estas fases 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.

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.

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 estriba también, en la mejora de la calidad de los desarrollos realizados. Determinados sistemas Case enfatizan más sobre este punto que sobre el anterior, introduciendo herramientas que permiten ejercer un control intenso de garantía de calidad del software desarrollado desde las primeras fases de su ciclo de vida.

Rango de las Herramientas Case (*)



Algunas Herramientas CASE son sólo para la fase de Diseño .

Otras, son sólo generadoras de Código



Algunas Herramientas de Análisis y Diseño tienen una visión de Desarrollo orientada a procesos sin la capacidad de modelamiento.

Algunas proveen Herramientas para el modelamiento sin incluir los procesos de Análisis o Diseño.



(*)Fuente: Revista, Tecnología de Punta

5. Componentes y funcionalidades de una herramienta CASE

A continuación se describen los principales componentes de una herramienta CASE y sus funcionalidades.

Repositorio

Base de datos central de una herramienta CASE. El repositorio amplía el concepto de diccionario de datos para incluir toda la información que se va generando a lo largo del ciclo de vida del sistema, como por ejemplo: componentes de análisis y diseño (diagramas de flujo de datos, diagramas entidad-relación, esquemas de bases de datos, diseños de pantallas), estructuras de programas, algoritmos, etc. En algunas referencias se le denomina Diccionario de Recursos de Información.

La mayoría de herramientas CASE poseen un repositorio propio o bien trabajan sobre un repositorio suministrado por otro fabricante o vendedor.

Apoyándose en la existencia del repositorio se efectúan comprobaciones de integridad y consistencia:

- *Que no existan datos no definidos.*
- *Que no existan datos autodefinidos (datos que se emplean en una definición pero que no han sido definidos previamente).*
- *Que todos los alias (referencias a un mismo dato empleando nombres distintos) sean correctos y estén actualizados.*
- *Las características más importantes de un repositorio son:*
 - *Tipo de información. Que contiene alguna metodología concreta, datos, gráficos, procesos, informes, modelos o reglas.*
 - *Tipo de controles. Si incorpora algún módulo de gestión de cambios, de mantenimiento de versiones, de acceso por clave, de redundancia de la información.*

La gestión de cambios y el mantenimiento de versiones, ayudarán en el caso de que convivan diferentes versiones de la misma aplicación, o se tengan que

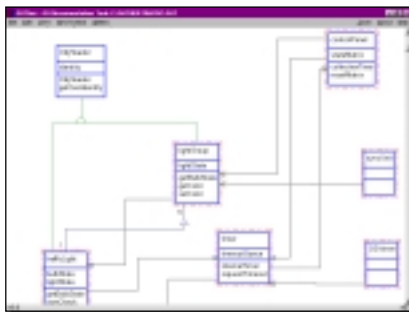
realizar cambios en la versión en producción y en la de desarrollo, simultáneamente.

- *Tipo de actualización. Si los cambios en los elementos de análisis o diseño se ven reflejados en el repositorio en tiempo real o mediante un proceso por lotes (batch). Esto será importante en función a la necesidad de que los cambios sean visibles por todos los usuarios, en el acto.*
- *Reutilización de módulos para otros diseños. El repositorio es la clave para identificar, localizar y extraer código para su reutilización.*
- *Posibilidad de exportación e importación para extraer información del repositorio y tratarla con otra herramienta (formateo de documentos, mejora de presentación) o incorporar al repositorio, información generada por otros medios.*
- *Interfases automáticas con otros repositorios o bases de datos externos.*

Módulos de diagramación y modelización

Algunos de los diagramas y modelos utilizados con mayor frecuencia son:

- *Diagrama de flujo de datos.*
- *Modelo entidad - interrelación.*
- *Historia de la vida de las entidades.*
- *Diagrama Estructura de datos.*
- *Diagrama Estructura de cuadros.*
- *Técnicas matriciales.*



Algunas características referentes a los diagramas son:

- *Número máximo de niveles para poder soportar diseños complejos.*
- *Número máximo de objetos que se pueden incluir para no encontrarse limitado en el diseño de grandes aplicaciones.*
- *Número de diagramas distintos en pantalla o al mismo tiempo en diferentes ventanas.*
- *Dibujos en formato libre con la finalidad de añadir comentarios, dibujos, información adicional para aclarar algún punto concreto del diseño.*
- *Actualización del repositorio por cambios en los diagramas. Siempre resulta más fácil modificar de forma gráfica un diseño y que los cambios queden reflejados en el repositorio.*

- *Control sobre el tamaño, fuente y emplazamiento de los textos en el diagrama.*
- *Comparaciones entre gráficos de distintas versiones. De esta forma será más fácil identificar qué diferencias existen entre las versiones.*
- *Inclusión de pseudocódigo, que servirá de base a los programadores para completar el desarrollo de la aplicación.*
- *Posibilidad de deshacer el último cambio, facilitando que un error no conlleve perder el trabajo realizado.*

Herramienta de prototipado

El objetivo principal de esta herramienta es poder mostrar al usuario, desde los momentos iniciales del diseño, el aspecto que tendrá la aplicación una vez desarrollada. Ello facilitará la aplicación de los cambios que se consideren necesarios, todavía en la fase de diseño.

La herramienta será tanto más útil, cuanto más rápidamente permita la construcción del prototipo y por tanto antes, se consiga la implicación del usuario final en el diseño de la aplicación. Asimismo, es importante poder aprovechar como base el prototipo para la construcción del resto de la aplicación. Actualmente, es imprescindible utilizar productos que incorporen esta funcionalidad por la cambiante tecnología y necesidades de los usuarios.

Los prototipos han sido utilizados ampliamente en el desarrollo de sistemas tradicionales, ya que proporcionan una realimentación inmediata, que ayudan a determinar los requisitos del sistema. Las herramientas CASE están bien dotadas, en general, para crear prototipos con rapidez y seguridad.

Generador de código

Normalmente se suele utilizar sobre ordenadores personales o estaciones de trabajo, por lo que el paso posterior del código al host puede traer problemas, al tener que compilar en ambos entornos.

Las características más importantes de los generadores de código son:

- *Lenguaje generado. Si se trata de un lenguaje estándar o un lenguaje propietario.*

- *Portabilidad del código generado. Capacidad para poder ejecutarlo en diferentes plataformas físicas y/o lógicas.*
- *Generación del esqueleto del programa o del programa completo. Si únicamente genera el esqueleto será necesario completar el resto mediante programación.*
- *Posibilidad de modificación del código generado. Suele ser necesario acceder directamente al código generado para optimizarlo o completarlo.*
- *Generación del código asociado a las pantallas e informes de la aplicación. Mediante esta característica se obtendrá la interfase de usuario de la aplicación.*

Módulo generador de documentación

El módulo generador de la documentación se alimenta del repositorio para transcribir las especificaciones allí contenidas.

Algunas características de los generadores de documentación son:

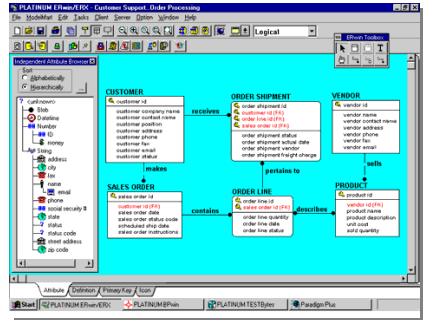
- *Generación automática a partir de los datos del repositorio, sin necesidad de un esfuerzo adicional.*
- *Combinación de información textual y gráfica, lo que hace más fácil su comprensión.*
- *Generación de referencias cruzadas. Con ello se podrá localizar fácilmente en qué partes de la aplicación se encuentra un determinado objeto o elemento, con el fin de analizar el impacto de un cambio o identificar los módulos afectados por un determinado error.*
- *Ayuda de tratamiento de textos. Facilidad para la introducción de textos complementarios a la documentación que se genera de forma automática.*
- *Interfase con otras herramientas: procesadores de textos, editores gráficos, etc.*

6. Ejemplos de Herramientas Case más utilizadas.

A continuación se describen los principales componentes de una herramienta CASE y sus funcionalidades.

ERwin

PLATINUM ERwin es una 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 stored procedure y triggers para los principales tipos de base de datos.

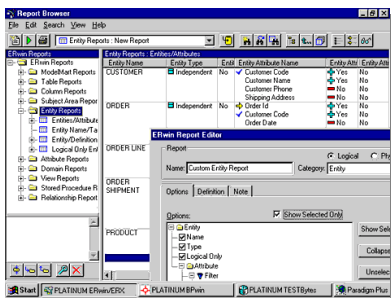


ERwin hace fácil el diseño de una base de datos. Los diseñadores de bases de datos sólo apuntan y pulsan un botón para crear un gráfico del modelo E-R (Entidad-relación) de todos sus requerimientos de datos y capturar las reglas de negocio en un modelo lógico, mostrando todas las entidades, atributos,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

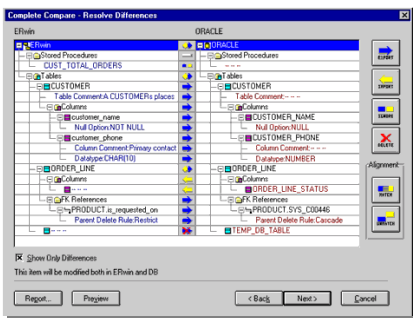
Más que una herramienta de dibujo, ERwin automatiza el proceso de diseño de una manera inteligente. Por ejemplo, ERwin habilita la creación de un diccionario de atributos reusables, asegurando la consistencia de nombres y definiciones para su base de datos.

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.



ERwin 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, ERwin genera automáticamente tablas, vistas, índices, reglas de integridad referencial (llaves primarias, llaves foraneas), 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.)

EasyCASE

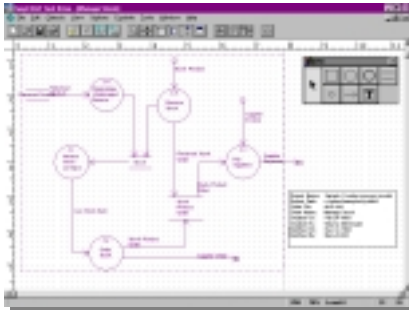
EasyCASE Profesional - el centro de productos para procesos, modelamiento de datos y eventos, e Ingeniería de Base de Datos- es un producto para la generación de esquemas de base de datos e ingeniería reversa - trabaja para proveer una solución comprensible para el diseño, consistencia y documentación 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.



EasyCASE 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 y más.

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.



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

El verdadero poder de EasyCASE se encuentra en el soporte comprensivo al modelamiento de datos, procesos y eventos. Posee desde el editor de diagramas

flexible y un diccionario de los datos integrado en formato dBASE, así como una extensa cantidad de reportes y análisis.

Porque EasyCASE Profesional, 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 asegurar la 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.

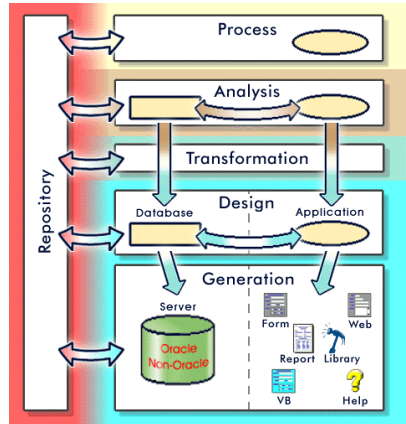
Especificaciones de EasyCASE Profesional:

METODOLOGIAS ESTRUCTURADAS: <ul style="list-style-type: none">• Yourdon/DeMarco• Gane & Sarson• Ward-Mellor• SSADM• Yourdon/Constantine• Chen• Martin• Bachman• Shlaer-Mellor• IDEF1X• Merise• Metrica	BASE DE DATOS QUE SOPORTA: <ul style="list-style-type: none">• Oracle• Paradox• Progress• SQLBase• SQL Server• Sybase• Watcom SQL• Access• ANSI SQL• Clipper• dBASE III, IV, V• DB2• FoxPro• Informix• Otras más...
TIPOS DE DIAGRAMAS: <ul style="list-style-type: none">• 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: <p><i>EasyCASE Professional 4.2 o superior requiere:</i> <i>EasyCASE Database Engineer;</i> <i>PC's 386/486/Pentium y compatibles;</i> <i>Microsoft Windows 3.1 o superior,</i> <i>8 MB RAM, 8 MB de espacio en disco duro;</i> <i>VGA o mejor color;</i> <i>Mouse.</i> <i>Requerimientos para Red: NetBIOS-compatible</i> <i>DOS, MS-Windows o Novell network.</i></p>

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

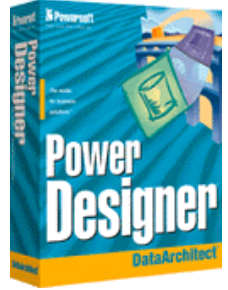
Muchas metodologías diferentes para base de datos y desarrollo de aplicaciones existen actualmente. Oracle Designer no fuerza al uso de alguna metodología específica, pero en cambio proporciona un juego de herramientas que le permiten que use la metodología de desarrollo que elija.

Oracle Designer soporta las siguiente metodologías:

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 (incluido) 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 datawarehousing para el diseño e implementación de una base de datos. Cuenta con soporte para bases de datos tradicionales DBMS y bases de datos en plataformas de sistemas analíticos usando modelados dimensionales, esquemas de "estrella" y "nieve", particionamiento y agregación. También cuenta con un alto desempeño en el indexamiento de esquemas.

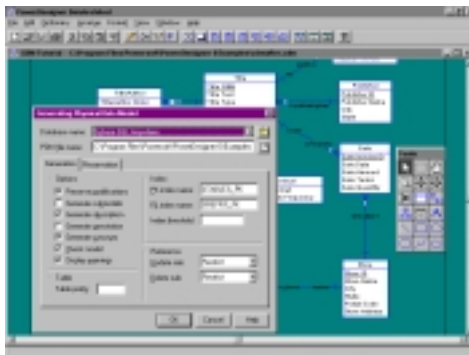
PowerDesigner MetaWorks.

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 de 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 base de datos 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.

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 ultima 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 arriba mencionados.

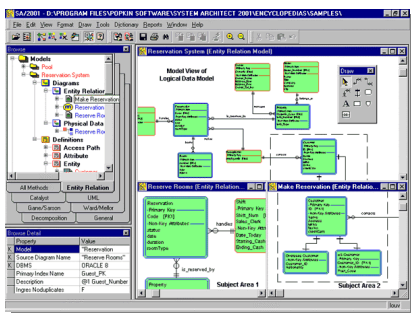
Posee esquemas de seguridad e integridad a través de contraseñas que posibilitan el acceso al sistema en diversos niveles, pudiéndose integrar a la seguridad de la red Novell o Windows/NT de ser necesario. Posee también con un completo Help sensible al contexto.

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 Modelo 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 customización de informes que emplea los recursos gráficos y facilidad de edición de Windows.

PRODUCTO

System Architect Versión 4.0 de 1997. Numero 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. Actualmente se encuentra en evaluación System Architect 2001.

PROVEEDOR

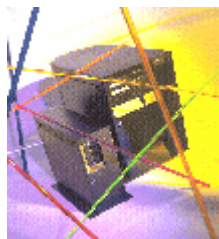
POPKIN Software & Systems Incorporated EUA - Fundada en 1987 y representada en América Latina por CHOOSE TECHNOLOGIES - Sao Paulo - Brasil Rua Alexandre Dumas, y Buenos Aires - Argentina Viamonte 759

REQUISITOS CLIENTE/SERVIDOR

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

SNAP

SNAP es un CASE (Ingeniería de Software Asistida por el Computador) 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 de IBM., totalmente documentados y ajustados a los requerimientos específicos de la organización, en una fracción del tiempo y coste del que se invertiría, si se utilizaran herramientas tradicionales.



SNAP se ha consolidado como el CASE más poderoso 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-RELACION, 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, Método de Desarrollo Acelerado (MDA), Utilitarios y Seguridad.

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 directamente desde otras herramientas (Upper Case), tales como V.A.W. o bien digitado directamente en SNAP. Además, SNAP prevé la facilidad de incorporar o trasladar DDS's existentes al Modelo de Datos. Una vez introducido el Modelo de Datos, 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 gama 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.

7. Adopción de Herramientas CASE en las Organizaciones.

Históricamente, las organizaciones han experimentado problemas con la adopción de Herramientas CASE. Dado que las organizaciones no conocen aún los beneficios de esta tecnología, se desea que el uso de un bien fundamentado proceso de adopción de CASE, ayude a incrementar la sucesiva adopción de estas herramientas.

Es importante ampliar el rango de organizaciones que adquieran tecnologías de computación y desarrollen estándares para el desarrollo de software, diseño de métodos, metodologías y técnicas para llevar adelante el ciclo de vida de los sistemas.

Para ello, se recomienda:



☞ *Identificar los factores críticos en los procesos.*

☞ *Proponer un conjunto de procesos a adoptar.*

☞ *Guiar satisfactoriamente esta adopción teniendo en consideración la organización y su entorno cultural.*

Los factores que hacen crítico el proceso de adopción de tecnología CASE, incluyen:

- ❖ **Objetivos:** *La definición de un conjunto de objetivos claros y medibles, que incluyan objetivos técnicos y de negocio. Un ejemplo podría ser "incrementar en 50% el mejoramiento de la calidad en la actividad de especificación de requerimientos" o "incrementar en un 40% el uso de herramientas CASE en el desarrollo de proyectos".*

- ❖ **Apoyo de la Administración:** Extender la participación activa de la alta gerencia para alentar la adopción de CASE, sin limitar la buena voluntad para obtener los recursos que sean necesarios.



- ❖ **Uso estratégico de herramientas:** Definir una estrategia clara para el uso adecuado de las herramientas.
- ❖ **Desarrollo del Plan para el proceso total de adopción:** Un plan y diseño para el proceso total de posicionar estas herramientas al interior de cada componente de la organización.
- ❖ **Compromiso:** Propiciar que las personas se involucren en el esfuerzo de adopción en forma activa, motivando a los participantes.
- ❖ **Metodología ajustable:** La buena disposición y factibilidad técnica de ajustar, cuando sea necesario, los métodos de la organización y los métodos típicos de usar herramientas CASE, de tal forma que permitan llegar a un conjunto consistente de métodos.



- ❖ **Entrenamiento:** Proveer el entrenamiento e información necesarios y apropiados en cada paso a cada persona envuelta en el proceso de adopción

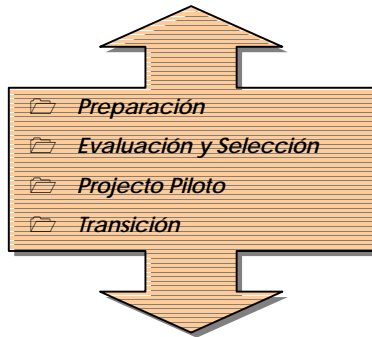
- ❖ **Ayuda de expertos:** Provisión de ayuda experta en el uso de estas herramientas durante el proyecto piloto y continuamente tal como las herramientas se utilicen entre los componentes

de la organización.

- ❖ **Proyecto piloto:** Los resultados de una prueba piloto controlada son prioritarios al tomar una decisión final.
- ❖ **Capacidad de la herramienta:** La capacidad técnica de la herramienta, en cuanto al entorno de hardware y software, de modo que satisfaga los objetivos definidos en el contexto del alcance esperado.

- ❖ **Cambiado moderado:** *Asegurar la viabilidad que la organización pueda operar simultáneamente entre el viejo y nuevo métodos, hasta que los componentes de la organización hayan cambiado totalmente hacia el nuevo método.*

La adopción de herramientas CASE debe incluir los siguientes procesos principales:



Proceso de preparación:

El propósito de este proceso es el establecimiento de los objetivos generales de los esfuerzos de adopción de herramientas CASE y establecer y definir los aspectos de dirección y administración.

Esta etapa comprende:

- *Establecimiento de objetivos,*
- *Verificación de la factibilidad y medición de los objetivos,*
- *Establecimiento de políticas,*
- *Desarrollo de planes.*

Proceso de Evaluación y Selección:

El propósito de este proceso es identificar las herramientas CASE más adecuadas, que sean las candidatas y aseguren que las herramientas recomendadas cumplirán con los objetivos de la organización.



Este proceso está compuesto por:

- *Iniciación: Define los objetivos y requerimientos para la evaluación de las herramientas CASE.*
- *Estructuración: Elaborar un conjunto de requerimientos basados en cada herramienta CASE.*
- *Evaluación: Producir reportes de evaluación técnica.*
- *Selección: Identificar las más adecuadas herramientas entre todas las candidatas.*

Proceso de Proyecto Piloto:



El propósito de este proceso es ayudar en la validación del trabajo desarrollado en el proceso inicial de adopción de CASE, y para determinar si la actual capacidad de la herramienta es lo que la organización necesita.

Este proceso está compuesto por:

- *Iniciación: Definir planes, procedimientos, recursos y capacitación para realizar el proyecto piloto.*
- *Performance: Ejecutar un proyecto controlado en cada nueva herramienta que pueda ser probada.*
- *Evaluación: Provee los resultados de la evaluación de la performance del proyecto piloto.*
- *Decisión: Liderar el proceso de adopción, dejando la herramienta o performance a un segundo proyecto piloto e identificando la experiencia en el aprendizaje de la organización para el proceso de transición.*

Proceso de Transición:

El propósito de este proceso es minimizar las interrupciones durante el cambio de los actuales procesos hacia la nueva tecnología, basados en el máximo aprovechamiento de las experiencias del proyecto piloto.

Este proceso está compuesto por:

- *Iniciación: Definir planes, procedimientos y recursos que permitan la transición hacia el uso de las herramientas.*
- *Entrenamiento: Capacitar a los usuarios de la nueva herramienta.*
- *Institucionalización: Aplicar progresivamente la herramienta en segmentos largos del entorno, como parte normal de la práctica organizacional.*
- *Monitoreo y apoyo continuo: Identificar si el proceso de adopción funciona correctamente y asegurar el entrenamiento, así como otros recursos que sean necesarios durante el período de transición.*
- *Evaluación y culminación: Medir el resultado de la adopción de las herramientas CASE y proveer a la organización la experiencia y conocimiento para futuros proyectos de adopción.*



8. **Puntos Importantes a considerar cuando se selecciona una Herramienta CASE.**

Seleccionar una Herramienta CASE no es una tarea simple. No existe una 'mejor' herramienta respecto de otra. Hay numerosas historias respecto al uso de CASE y las fallas que pueden producirse. Las fallas o las respuestas satisfactorias están en relación con las expectativas. Si el proceso de evaluación y selección de las Herramientas CASE falla, entonces la Herramienta no cumplirá con las especificaciones o expectativas del negocio. Esto puede ocurrir durante el proceso de implementación o ejecución del producto.

Hay tres puntos comunes que fallan en el proceso de evaluación y selección:

- *El proceso en sí mismo.*
- *Los pre-requisitos necesarios.*
- *Conocer la organización.*

El proceso en sí mismo:

El proceso de evaluación y selección de Herramientas CASE debe aproximarse a un proyecto mayor. El proceso debe definirse cuidadosamente y debe incluir las mejores técnicas de dirección de proyecto. Ninguna selección es igual que otra, porque dos organizaciones no son iguales. Por ejemplo, el proceso de selección para el Ministerio de Defensa puede ser completamente diferente que en una corporación comercial. Aunque hay principios básicos, por ejemplo, todos debemos entender el criterio en el que está basado el proceso de selección, todos deben tener una visión común. Es adecuado limitar el número de vendedores tanto como sea posible, para poder enfocar y entender realmente una determinada herramienta.



Los pre-requisitos necesarios:

El propósito de las herramientas CASE es apoyar y facilitar el desarrollo de software. Debe haber una comprensión clara del propósito de las herramientas que se propongan dentro del ambiente de desarrollo que es compartido por el equipo de la selección. El equipo debe tener una visión común del ambiente de desarrollo de sistemas, resultando la selección de la herramienta adecuada.



Otro requisito previo importante sería tener una metodología de desarrollo de sistemas seleccionada. Sin una metodología, ingresará al largo camino del fracaso. Las herramientas implementan la metodología, no la determinan.

Conocer la organización:

Cuando se está evaluando y seleccionando una herramienta CASE, es importante conocer y entender a la organización. Tal como las personas son únicas, así también las organizaciones son únicas a su propio modo, cada una tiene una personalidad e infraestructura propias. Una empresa podría disciplinarse y alcanzar un nivel alto de madurez en el proceso de diseño de software, mientras otra puede estar en las fases tempranas. Sin tener en cuenta la disciplina y la madurez, es muy importante entender la organización que se verá reflejada en la selección final.



Estrategias de Implantación de una Herramienta CASE

1. *Identificar la magnitud de problemas a resolver en la Institución.*
2. *Identificar el nivel estratégico que deben tener los sistemas.*
3. *Evaluar los recursos de hardware y software disponibles en la Institución y el medio.*
4. *Evaluar el nivel del personal.*
5. *Efectuar un estudio de costo-beneficio definiendo metas a lograr.*
6. *Elegir las herramientas apropiadas para la Institución.*
7. *Establecer un programa de capacitación de personal de sistemas y usuarios.*
8. *Elegir una aplicación que reúna la mayor parte de los siguientes requisitos:*
 - *Gran impacto de resultados.*
 - *Disponibilidad de recursos.*
 - *Mínimo nivel de riesgos.*
 - *Máxima colaboración de usuarios.*
 - *Tamaño reducido de solución.*
9. *Se establecerán interfases de compatibilidad de los nuevos sistemas que deben convivir con los sistemas anteriores.*

Consideraciones Importantes

①. *La elección del Case va a depender de sus estrategias de desarrollo:*

- *Si tiene un gran volumen de aplicativos desarrollados, es conveniente contrastar lo realizado versus las técnicas de Análisis y Diseño.*
- *Si tiene presión por resultados a corto plazo, el empleo de un Lower Case le será de utilidad, si se basa en modelos de datos y procesos claros y definidos.*
- *Si desea realizar proyectos de gran envergadura es recomendable aplicar Upper y Lower Case.*
- *Si trabaja con archivos de grandes dimensiones, es recomendable que el Case soporte el Diseño de Bases de Datos.*
- *Si no tiene formación y experiencia en el manejo de metodologías, es recomendable contar con asesoría especializada, que capacite al personal y supervise los avances de Análisis y Diseño.*

②. *Evalúe la eficiencia del producto en las pruebas unitarias y de integración, y fundamentalmente en las pruebas de sistemas.*

③. *Considere los recursos apropiados para usar el Case, de Hardware (memoria, disco, concurrencia), de Software (versión de Sistema Operativo).*

9. Tendencias Tecnológicas y del Mercado

de las Herramientas CASE

Las principales líneas de evolución hacia las que parecen encaminarse las herramientas CASE son:

- *CASE para sistemas bajo arquitectura cliente/servidor. No hay que confundir el hecho de que una herramienta CASE funcione en un entorno de arquitectura cliente/servidor, con que el sistema desarrollado mediante una herramienta CASE vaya a funcionar bajo dicha arquitectura.*

En la actualidad ya hay ejemplos de los dos casos, herramientas CASE que funcionan bajo un entorno cliente/servidor, en red y con un repositorio centralizado en un servidor y herramientas CASE que generan aplicaciones que funcionan en un entorno cliente/servidor, en las cuales se puede indicar dónde deben residir los componentes de la aplicación en tiempo de ejecución, liberando al programador de aspectos referidos a los protocolos de comunicaciones, seguridad, interfases gráficas de usuario, etc.

La línea de evolución, en este caso, vendrá marcada por versiones mejoradas de la herramienta, que faciliten cada vez más la distribución de los elementos de una aplicación entre los diferentes clientes y servidores y una mayor liberalización del programador, de todos los aspectos que no sean propios de la aplicación (protocolos de red, seguridad, etc.).

- *CASE multiplataforma. Estas herramientas soportan las combinaciones dominantes de diferentes plataformas físicas, sistemas operativos, interfases gráficas de usuario, sistemas de gestión de bases de datos, lenguajes de programación y protocolos de red. En este sentido el futuro podrá ser de apertura creciente a nuevas plataformas y portabilidad más generalizada.*
- *CASE para ingeniería inversa y directa. Ya existen algunas herramientas de este tipo. Su evolución marcará notables mejoras en la obtención de los diseños a partir del código ya existente (ingeniería inversa) y la regeneración del mismo, una vez optimizado el diseño (ingeniería directa).*

- *CASE para trabajo en grupo (groupware). Estas herramientas se centran en el proceso de desarrollo más que en el producto a desarrollar, facilitando la integración de diferentes grupos humanos, pertenecientes incluso a empresas diferentes, trabajando conjuntamente en un gran proyecto. Deberían incorporar las facilidades clásicas de ofimática: correo electrónico, calendarios en línea, planificación de actividades, preparación de documentos, actas de reuniones, etc.*
- *CASE para desarrollo de sistemas orientados a objetos. En la actualidad existen algunas herramientas que cubren alguna de las fases del ciclo de vida de desarrollo de aplicaciones orientadas a objetos (interfase de usuario, análisis, diseño, programación, etc.). El objetivo futuro podría ser cubrir el ciclo de vida completo. Aunque hoy en día, la mayor efectividad se consigue con las herramientas CASE para métodos estructurados, en un futuro no muy lejano esta situación se invertirá a favor de las que soportan objetos. La proliferación de este tipo de herramientas podrá verse retrasada debido al gran número de notaciones y metodologías de orientación a objetos distintas que existen en la actualidad.*

Es importante resaltar que las herramientas actuales permiten generar objetos: modelo "estático" y modelo "funcional", mas no el modelo "dinámico". La programación orientada a objetos puede cambiar la forma que tienen las empresas de hacer negocio y como tal, necesita ser tratada cuidadosamente, tanto por las empresas u organismos, como por los fabricantes de tecnologías que proporcionan las soluciones.

A mediano y largo plazo, otras posibles líneas de evolución serán:

- *La utilización de la tecnología multimedia.*
- *La incorporación de técnicas de inteligencia artificial.*
- *Sistemas de realidad virtual.*



Bibliografía

- ***El Analista de Sistemas***
Grupo de Estudios AS – <http://members.xoom.com/analista/portada.htm>
- ***Revista, Tecnología de Punta***
Ing. Víctor Castañeda Guzmán
- ***Herramientas para el Desarrollo de Sistemas de Información***
Instituto Nacional de Estadística e Informática
- ***Power Designer***
<http://www.sybase.com/products/powerdesigner>
- ***Oracle Designer***
<http://www.oracle.com/tools/designer/quicktour/contents.htm#features>
- ***Erwin***
http://www.platinum.com/products/brochure/als/b_erwin.htm
- ***System Architect***
<http://www.popkin.com/products/sa2001/product.htm>
- ***The Case Tool Home Page***
<http://osiris.sunderland.ac.uk/sst/case2/>
- ***Herramientas CASE***
<http://www.geocities.com/SiliconValley/Bit/6238/index.htm>
- ***Otras páginas Web.***