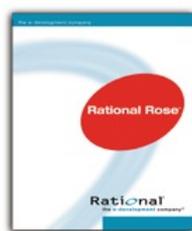


POWER
DESIGNER



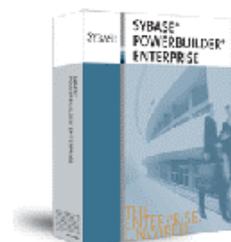
TOGETHER



RATIONAL
ROSE



OPEN CASE 4/0



POWER
BUILDER

SELECCIÓN DE HERRAMIENTAS CASE



Universidad **Alfonso X El Sabio**
Curso 2004-2005
Carrera Ingeniería Informática
Asignatura **Gestión de Sistemas de Información**
Profesora D. Arantazu Cámara
Alumnos J. Javier Masa Lledó 17376
Javier Amorós de la Vega 16245
Juan Fco. Aranda Copado 22399
Sergio Pedrero Garzón 22937
Juan Manuel Callejas 23785
Fernando D. Muñoz Rosales 26843

I. OBJETIVOS.....	3
II. SUPUESTOS DEL CASO	5
III. Clasificación de CRITERIOS	6
IV. CRITERIOS de FILTRO.....	6
V. CRITERIOS GENERALES.....	8
VI. CRITERIOS ESPECÍFICOS.....	9
VII. EJEMPLOS DE PROVEEDORES.....	10
VIII. RESULTADOS DE PROVEEDORES	12
VIII.1 POWERBUILDER.....	12
VIII.2 RATIONAL ROSE	15
VIII.3 BORLAND TOGETHER.....	18
VIII.4 POWER DESIGNER	20
VIII. 5 CASE 4/0.....	23
IX. RESULTADOS FINALES	25
X. INTERNETGRAFIA	26

I. **OBJETIVOS**

Como departamento de tecnología a la gerencia de Ingeniería del Software hemos de encontrar una herramienta CASE apropiada para el desarrollo de software dentro de nuestra empresa. Nuestra empresa dispone de una metodología estructurada con un modelo de ciclo de vida en cascada. Debemos tener en cuenta que la arquitectura en la que se basan los sistemas es Cliente/Servidor con una programación estructurada. Intentaremos analizar la función y necesidad de diversas herramientas CASE, en vista de su futura implantación dentro del departamento.

Consideraremos una unión de sistemas software que utilizan técnicas de diseño y metodologías de desarrollo bien definidas como herramientas CASE. Habremos de controlar como:

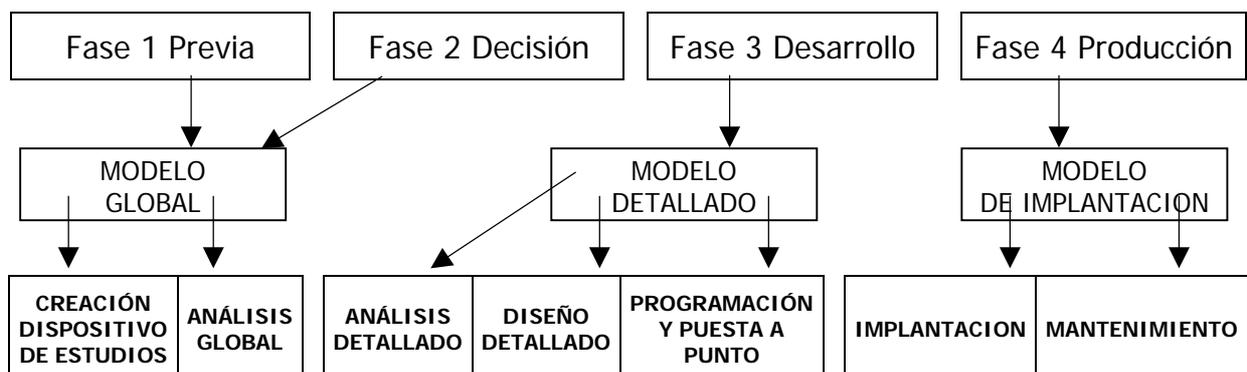
- Construir los modelos que describen la empresa,
- Describir el medio en el que se realizan las actividades,
- Llevar a cabo la planificación,
- El desarrollo del Sistema Informático, desde la planificación, pasando por el análisis y diseño de sistemas, hasta la generación del código de los programas y la documentación.

Por tanto, consideramos que los objetivos necesarios para considerar una herramienta CASE como adecuada son:

1. Mejora de la calidad del software desarrollado
2. Reducir tiempos de desarrollo
3. Reducir tiempos de mantenimiento del software.
4. Mejora la gestión del proyecto
5. Mejora la gestión en cuanto su planificación
6. Mejora la gestión en cuanto su ejecución y control.
7. Por lo tanto, aumento de productividad, tanto en las áreas desarrollo como en las de mantenimiento de estos sistemas.
8. Mejora del archivo de datos (know-how pedia).
9. Mayor facilidad de uso y reducción de la dependencia de analistas y programadores.
10. Integrar las fases de desarrollo (ingeniería del software).

11. Facilita la utilización de las distintas metodologías que desarrollan la propia Ingeniería del Software.
12. Podemos automatizar:
 - a. Desarrollo del software.
 - b. Creación de documentación.
 - c. Generación de código.
 - d. Chequeo de errores.
 - e. Herencias y dependencias.
 - f. Gestión de proyecto.
 - g. Creación de interfaces.
 - h. Creación del flujo del programa (flujogramas y diagramas de secuencias).
 - i. Casos de Uso.
13. Permite:
 - a. La reutilización del código (usabilidad)
 - b. Portabilidad del software
 - c. Estandarización de la documentación.
 - d. Estandarización de la metodología.
 - e. Fácil comprensión (no es lo mismo ver un diagrama que ver el código directamente).

La herramienta CASE a usar deberemos seleccionarla en función del momento que estemos del desarrollo. Así, con un ciclo en cascada y tras tener determinado por parte del protocolo de la empresa la estructura que debe tener cada uno de los desarrollos la idea es evolucionar este protocolo al uso de estas herramientas.



II. SUPUESTOS DEL CASO

Nosotros pertenecemos dentro del departamento de tecnología a la gerencia de Ingeniería del Software. Las aplicaciones que deberemos generar tendrán una estructura cliente/servidor con programación en cascada. Nuestra empresa tendrá una metodología de desarrollo estructurada con un ciclo de vida evolutivo.

La empresa se encuentra dividida en 4 departamentos, por lo que nuestra herramienta deberá dar una respuesta y funcionalidad adecuada para cada uno de estos departamentos.

- **Desarrollo:** departamento al que pertenecen aquellas personas responsables del desarrollo técnico de los proyectos. Las aplicaciones pueden estar desarrolladas por personal de mi empresa como por empresas de servicio externa a las que se contratan.
- **Explotación:** Este departamento donde se implementan, explotan y mantienen las aplicaciones. Este departamento se encargará de la relación directa con el cliente al ser los encargados de la implantación de nuestros sistemas, y del mantenimiento de los mismos. Este departamento valorará positivamente que crea fiabilidad con la arquitectura Cliente/Servidor.
- **Tecnología:** donde se estandarizan las plataformas posibles de desarrollo y de explotación.
- **Soporte al desarrollo:** La funcionalidad de este departamento será el mantenimiento de las herramientas de desarrollo, así como los modelos de información de la empresa. Este departamento exigirá que la herramienta CASE tenga una alta fiabilidad.

Así y tras mirar el esquema que hemos puesto en el punto anterior deberemos hacernos con una herramienta CASE que controle las 4 fases del desarrollo ya que es nuestra obligación la selección de esta herramienta.

III. Clasificación de CRITERIOS

Los criterios se clasifican en:

1. Criterios de Filtro.
2. Criterios Generales.
3. Criterios Específicos.

IV. CRITERIOS de FILTRO

Los criterios filtros son aquellos que consideramos imprescindible que cumplan como mínimo una herramienta CASE para ser tenida en cuenta. Esto permitirá desarrollar una simple checklist que nos permita cribar las herramientas y seleccionar para un análisis más detallado aquellas herramientas que tengan potencial real con respecto a nuestras necesidades.

1. **REPOSITORIO:** Nuestro modelo de empresa que se basa en desarrollo software, y debido a las necesidades que esto conlleva, nuestra herramienta CASE deberá tener un repositorio en forma de base de datos con las siguientes características:
 - Datos: Elementos atributos (campos), asociaciones (relaciones), entidades (registros), almacenes de datos, estructuras, direcciones cliente/servidor, etc.
 - Procesos: Procesos, Funciones, módulos, etc.
 - Conocimiento de la topología de comunicación.
 - Gráficos: DFD (Diagrama de flujo de datos), DER (Diagrama Entidad Relación) DFD (Diagrama de Descomposición Funcional), ED (Diagrama de Estructura), Diagrama de Clases, Diagrama de Cliente/Servidor, Diagrama de Secuencia, Diagrama de Actividad,...
 - Reglas: de conexión, de Gestión, de métodos, ...
2. **DOCUMENTACIÓN:** Requeriremos que la documentación generada por la herramienta CASE se adecue a los estándares utilizados por nuestra empresa, asegurando una correcta metodología a lo largo de todos los procesos software.
3. **ADECUACIÓN AL CICLO DE VIDA:** La estructura de la empresa de la que somos sus ingenieros de Sistemas, definida en el supuesto del caso, considera imprescindible que la herramienta CASE contemple de forma detallada todos los procesos de un ciclo de vida en cascada.
4. **ACCESIBILIDAD:** La información no debe ser local, sino que debe ser posible acceder a ella por toda la red corporativa y, de igual modo, acceder desde fuera de modo seguro.
5. **SEGURIDAD:** La herramienta que elegiremos debe ser usada de forma flexible, desde cualquier máquina y con unos altos niveles de seguridad, que

requiere garantizar que diversos individuos tengan un acceso filtrado a la información, según el nivel desde el que desarrollen. Debemos buscar la integridad de la información.

6. **FIABILIDAD:** Necesitamos una herramienta que no produzca errores, o en el peor de los casos, que sea capaz de solucionarlos sin perjudicar la labor de nuestra empresa. Por ello seleccionaremos tan solo una herramienta CASE que esté lo suficientemente extendida o tenga el aval de una gran empresa, como para garantizarnos una cobertura total en caso de problemas.
7. **PRODUCCIÓN DE DATOS CLIENTE SERVIDOR:** Nuestra empresa se haya especializada en aplicaciones cliente/servidor. Así pues la herramienta CASE deberá tener una parte específica para el desarrollo de este tipo de aplicaciones con gestión de servidores, paquetes y conexiones, así como protocolos de seguridad.
8. **INTERFAZ:** Esta interfaz debe ser controlada por el usuario y deberá cumplir los siguientes estándares:
 - a. Interfaz amigable
 - b. fácil de usar
 - c. robusta
 - d. fácil de aprender
 - e. adaptable
 - f. simple
 - g. consistente
 - h. flexible
 - i. lenguaje natural.

Ya que en nuestra empresa no solo trabajarán informáticos, necesitaremos que la herramienta que utilicemos sea amigable al uso y no perder demasiado tiempo en cursos de formación. Así pues exigiremos como mínimo un interfaz de tipo WIMP¹.

¹ WIMP: Windows, Icons, Mouse, Pull down menus

V. CRITERIOS GENERALES

Los criterios generales son aquellos que coinciden en todas las herramientas CASE analizadas. Realizaremos una evaluación con 6 características comunes de la que intentaremos sacar métricas de desarrollo. Si esto no fuera posible, ya que los programas que vamos a utilizar son versiones de evaluación y muchas de las características están impedidas, en la exposición final nos decantaremos por uno e intentaremos ceñirnos de la mejor forma posible a estos criterios.

- **DIAGRAMAS:**
 - Diagramas de desempeño rápido.
 - Herramientas para creación y control de flujos de entrada y salida.
 - Modelo relacional de datos.
 - Capacidad para el cambio de estilo y aspecto.
- **ESQUEMA DE LA CONEXIÓN CLIENTE/SERVIDOR:**
 - Permitir la conexión tanto en UDP como TCP.
 - Especificar los parámetros de las conexiones.
 - Análisis de los participantes.
- **GENERACIÓN DE CÓDIGO:**
 - Soporte para multivalentes.
 - Código multiplataforma.
 - Revisión y corrección de errores.
 - Autodocumentación.
 - Reusabilidad de código.
 - Generar casos de prueba.
- **INGENIERÍA E INGENIERÍA INVERSA:**
 - Generar clases a partir de un modelo de datos lógico.
 - Generar un modelo de datos lógico a partir de clases implementadas.
- **DISEÑO**
 - Proceso de diseño gráfico.
 - Ayuda para diseñar pantallas, y menús para aplicaciones de distintos sistemas windows.
 - Proveer de distintos módulos gráficos para el diseño de las aplicaciones.
 - Generación de aplicaciones a partir de modelos de datos.
 - Rutinas simples de captura para generación rápida de menús y submenús.

- REPORTEES
 - Posibilidad de ver e imprimir modelos de datos de diferentes maneras.
 - Control completo de la apariencia y contenido de los reportes.
 - Generación de reportes en formato html.
 - Adecuación de reportes a estándar genérico.
 - Aceptación de nuevas plantillas de forma dinámica.
- AYUDA
 - Asistente para aprendizaje.
 - Casos ejemplo.
 - Respuestas a preguntas concretas.
 - Buscador de ayuda.
 - Explicaciones claras.

VI. CRITERIOS ESPECÍFICOS

Los criterios específicos son aquellos que sin darse en todas las herramientas CASE analizadas, consideramos que son suficientemente valiosos como para ser tenidos en cuenta.

- Permitir especificar los parámetros de las conexiones.
- Realizar cambios en las conexiones en tiempo de ejecución.
- Generación de archivos de dialogo.
- Subvistas para separar y manejar áreas importantes en los diagramas, a modo de caja negra.
- Generación automática de código en diferentes lenguajes.
- Mecanismos simples de cambio de herencias.
- Exportacion a formato web o imagen.
- Posibilidad de generar reportes a gusto del usuario.
- Reportes exportables a editores diferentes.

VII. EJEMPLOS DE PROVEEDORES

Para realizar este trabajo nos hemos basado en cinco herramientas CASE, especializadas en el desarrollo de software en pos del manejo cliente/servidor, seleccionadas de <http://www.cs.queensu.ca/Software-Engineering/toolcat.html>. Para ello hemos consultado los datos expuestos en sus respectivas paginas web y consultados foros de opinión y FAQ's. Las herramientas de evaluación a las que hemos accedido presentan taras así que alguno de los puntos quedará desmarcado.

Las cinco herramientas seleccionadas cumplen todos los requisitos de filtro deseadas. Están desarrolladas dentro de una serie de paquetes empresariales de gestión donde la adecuación a un desarrollo software normal se cumplen con creces. Las cinco obtienen buenos resultados con los generales, y, además, poseen diversos requisitos específicos de utilidad.

Las herramientas que vamos a intentar evaluar son:

❖ POWER BUILDER

- Sybase, Inc. 6475 Christie Avenue Emeryville, CA 94608 (510) 922-3500, (510) 922-9441; www.sybase.com
- Precio: 2.833€
- <http://www.sybase.com/products/developmentintegration/powerbuilder>
- Requerimiento mínimo del sistema: Procesador Intel 486. 256 MB RAM. Sistema operativo Windows NT, W98, W95. WME, Windows XP. JDK 1.1, WebServer.
- Frente al POWER DESIGNER, que miraremos en este trabajo también, nos enfrentamos a una herramienta enfocada eminentemente al desarrollo (frente a la herramienta de modelado de datos y aplicaciones); este desarrollo de aplicaciones (RAD) que aumenta la productividad del desarrollador a través de la integración de las funciones de diseño, modelado, desarrollo y gestión. Construye arquitecturas Web y aplicaciones de conectividad limitada. Independientemente de donde tus usuarios se encuentren, mediante el POWER BUILDER podremos suministrar esta información a los usuarios. Soporta integración con J2EE y plataforma .NET

❖ BORLAND TOGETHER 6.1

- Borland Inc.
- Precio: 1500 €
- <http://www.borland.com/>
- Requerimiento del sistema: 512 MB RAM. Entre 70-310 MB de espacio en HD. Procesador Intel Pentium III, 500 Mhz. Sistema operativo Windows XP, W2000. Sistema operativo Linux, Red Hat Linux. Sistema operativo Solaris, ultraSPARC, Solaris 8. Sistema Operativo Mac OS X, Apple G3
- El Together basa el éxito en la tecnología subyacente. Esta diseñada para aumentar el ciclo de vida de las aplicaciones. Sirve para ampliar las capacidades en el análisis de los procesos de desarrollo. Soporta las principales arquitecturas estándares del mercado para el desarrollo y

manejo de aplicaciones empresariales e-business. Integra plataformas y estructuras conceptuales (Sun J2EE platform, Microsoft .NET). Cuenta con tecnología Starbase Corp y TogetherSoft que ayuda en la cartera de soluciones.

❖ RATIONAL ROSE SOFTWARE

- IBM.
- Precio: 1,057 €
- <http://www-306.ibm.com/software/rational/>
- Requerimientos del sistema: Windows 2000, Windows NT, Windows XP. PentiumIII 500 MHz; 500 MHz, mínimo de 384 MB of RAM; 768 MB de espacio en disco. Linux (Linux Red Hat 7.3, 8.0). Solaris (Solaris 2.6, 7, 8, or 9).
- El software de IBM, avalado por la marca, provee una serie de herramientas para modelado de arquitecturas, modelado de diseño, construcción, aplicaciones RAD, testado de componentes,... que maximizan la productividad. En la construcción de aplicaciones de negocio, productos software y de sistemas, así como sistemas embebidos dentro de otros sistemas, el Rational se encuentra dentro de la plataforma de desarrollo de IBM. Esta completa plataforma está especialmente indicada para desarrollo en equipo.

❖ POWER DESIGNER 6.1.

- Sybase, Inc. 6475 Christie Avenue Emeryville, CA 94608 (510) 922-3500, (510) 922-9441; www.sybase.com
- Precio: \$2092.98 (dólares)
- Requerimientos del sistema: para Windows 9x o NT, procesador 486 o mayor, 12MB en RAM, 10 MB en disco duro.
- 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

❖ CASE 4/0

- MicroTOOL GmbH, (+49 30) 467 086-0.
- www.microTOOL.de
- Requerimientos del sistema: para Windows 9x/2000/NT, Pentium II Workstation, 64 MB RAM.
- Case/4/0 integra el análisis gráfico, diseño y herramientas de programación. Como los estándar, case/4/0 genera C/C ++, COBOL, PL/1, Java™, Visual Basic y definiciones para bases de datos relacionadas del diseño de software.

VIII. RESULTADOS DE PROVEEDORES

VIII.1 POWERBUILDER

1.A. DIAGRAMAS

Power Builder es un software de Aplicación que se utiliza para desarrollar aplicaciones de manera rápida y usa tecnologías RAD (Rapid Application Development [desarrollo rápido de aplicaciones]) como cliente/servidor y las iniciativas orientadas al web.

Power Builder se usa como herramienta de desarrollo la cual utiliza y manipula datos de una o varias bases de datos a la vez de diferentes fabricantes de programas de gestión de bases de datos relacionales. Para ello debe conectarse a las base de datos utilizando una conexión para cada base de datos. En Power Builder se pueden crear aplicaciones no sólo para windows, sino también para UNIX y Macintosh. El propietario fabricante de Power Builder es la compañía SYBASE® Inc. ..

1.B. ESQUEMA DE LA BASE DE DATOS

Power Builder es un software de aplicación para el desarrollador de aplicaciones, cuyas características básicas son:

- Soporta una gran variedad de sistemas de gestión de base de datos, tales como: Sybase, Informix, Oracle, Watcom, entre otras.
- Tiene capacidad de acceder a información de múltiples bases de datos y mostrar esa información en una única ventana.
- Se trabaja en ambiente cliente – servidor.
- Posee un objeto inteligente llamado Datawindow que realiza directamente la interfaz con la base de datos, sin requerir que el programador conozca SQL.
- Capacidad de utilizar sentencias SQL combinadas en el código.
- Se puede trabajar en múltiples plataformas, ya que soporta diferentes sistemas operativos y posee drivers nativos para las bases de datos más comerciales.
- La creación de aplicaciones es sencilla.
- Se realiza poca programación.
- Permite usar ventanas, botones y todas las herramientas que presenta el windows facilitando su manejo.
- Para construir la aplicación se utilizan painters (pintores), allí se definen las propiedades de los objetos y se agregan los controles.
- Se puede trabajar con múltiples ventanas.

1.C. GENERACIÓN DE CÓDIGO

Power Builder combina bases de datos back-end y desarrollo de aplicaciones front-end en un ambiente unificado. Tiene soporte para multicientes, Power Builder 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.

Power Builder 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 Power Builder 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.

1.D. INGENIERÍA E INGENIERÍA INVERSA

Mejora del diseño y mantenimiento del modelo de datos: soporta las últimas versiones de los motores de bases de datos y nuevas funciones tales como servicios web y XML en las bases de datos.

Mejora el soporte a UML: incorpora nuevos soportes, tales como J2EE 1.4 así como plantillas y modelos de servicios web.

Mejora el modelo del proceso de negocio: ofrece una visión jerárquica de los procesos diseñados con esta herramienta, que proporcionan una comunicación clara y una mayor flexibilidad.

Nuevo modelo de información Liquidity: soporta el movimiento y la replicación de datos así como la gestión de metadatos, proporcionando un modelo que permite acceder a bases de datos móviles y la sincronización con las bases de datos centrales.

Ofrece colaboración conjunta a través de la tecnología Link and Synch entre todos los modelos soportados: negocio, UML y datos.

Mejora la productividad personal, implementa una serie de generadores automáticos de código, ingeniería inversa y modelado evolutivo, sincronizando el código y el modelo. Lo que se traduce en una menor codificación manual, mantenimiento y esfuerzo de reingeniería.

1.E. DISEÑO

Una aplicación típica que interactúa con un usuario, tal como una hoja de cálculo sobre una computadora personal o un sistema de contabilidad sobre una mainframe, consiste de tres elementos: la presentación, la lógica de la aplicación y los datos.

La presentación es el conjunto de elementos que muestra la aplicación (ventanas, controles, menús) y está enfocada a la interacción con el usuario.

La lógica de la aplicación ejecuta los procesos y determina el flujo de la aplicación.

Los datos son los elementos que procesados proveerán de información al usuario.

Respecto a la configuración y despliegue de la aplicación, los desarrolladores tienen distintas opciones: aplicaciones de escritorio monolíticas corriendo sobre una computadora, aplicaciones multiusuarios corriendo sobre una mainframe, o aplicaciones cliente/servidor.

Todos los objetos o controles que se crean en Power Builder se guardan en una o más librerías asociadas con la aplicación (que están formadas por uno o más de dichos objetos y controles que estarán agrupados en archivos de extensión PBL) cuando se ejecutan las aplicaciones, Power Builder recupera los objetos de las librerías Pintores y editores.

Algunas herramientas que se usen para editar o crear objetos y controles se les llama Pintores. Por ejemplo, se puede crear una ventana en el pintor ventana. Dentro del pintor de ventanas se puede definir las propiedades de las mismas, agregar controles tales como botones y etiquetas, crear miniprogramas (scripts) usando Power Script para la ventana y para los controles que la misma contenga con el fin de instrumentar la funcionalidad que se desee obtener.

Power Builder proporciona pintores para ventanas (windows), menús, DataWindow objetos, objetos visuales y no visuales definidos por el usuario, funciones, estructuras, base de datos, data pipelines, y la aplicación en sí. Para cada uno de estos tipos de objetos, habrá también un editor de código en el que se puede modificar el código directamente.

Para webs, existen editores especializados para HTML y archivos JSP, frames sets, y hojas de estilo.

Interfaz altamente gráfica, intuitiva y herramientas basadas en ventanas.

1.F. REPORTES

Calidad de presentación en los reportes.

Power Builder proporciona la posibilidad de generar fácilmente informes a partir de los diagramas y datos que contiene.

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.

1.G. AYUDA

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.

1.H. WEB

Una aplicación Web que puede contener todos los elementos que se necesite para construir un sitio web -archivos HTML, scripts, imágenes, componentes que se descargan – o una aplicación JavaServer Pages (JSP). Un Web target además contiene configuraciones para construir opciones, realizar conexiones a base de datos y distribución.

La capacidad de crear Web Targets esta disponible únicamente en la versión empresarial de Power Builder (Enterprise Edition)

Objetos. Una aplicación es una colección de objetos. Power Builder provee de diversos tipos de objetos, inclusive objetos gráficos tales como ventanas, menús, botones y los objetos no visuales tales como los datastore, exception y objetos de transacción. En un Web Target, el objeto con el que se trabaje incluye páginas HTML, imágenes y hojas de estilo.

VIII.2 RATIONAL ROSE

2.A. DIAGRAMAS

La creación de diagramas técnicos y de negocios permiten organizar y entender los procesos. La selección de clases, modelos, colección de objetos, casos de estado y demás se realizan desde el menu lateral que permite la selección según el diagrama que estemos utilizando. Las herramientas comunes se encuentran situadas en el centro de la estructura y son accesibles fácil e intuitivamente mediante un clic de ratón. La estructura WIMP de la que ya hemos hablado se cumple también en esta herramienta. Nos comunicamos con la interfaz Rose Extensibility a través de varias posibilidades: mediante Scripts o mediante Rose Automation; en ambos casos las llamadas REI se describen siempre en la ayuda online.

Separa las clases en colores y diferentes objetos dentro de la misma estructura se encuadran de diferente formas y colores. Así se adaptan a la estructura de diagrama que se usa habitualmente en el desarrollo de software, dándole un punto más hacia la comprensión introduciendo los colores.

- El cambio de herramientas se hace con un simple clic derecho de raton y permite la aplicación de cambios en tiempo real a la estructura de desarrollo.
- El cambio de características de clases y objetos se realiza con un rapido menu contextual donde se pueden añadir y quitar campos facilmente. Además, el linkado, compilado y ejecución de codigo se realiza con un simple cambio de ventana.
- El Rational Rose permite la encapsulacion de los objetos, a modo de caja negra, de las partes del diagrama que consideremos oportunos. Asi si existiera una seccion que ya estuviera implementada, probada y demás, la realización de esta "caja negra" simplifica mucho este diseño.
- El estándar UML que utiliza es el **UML 2**.
- Fantástico el diseño del entorno del trabajo. Permite tener en pantalla varias ventanas de uso, separado por pestaña, posee una miniatura donde comprobar como va yendo el diseño, mantiene en todo momento la estructura lateral de objetos de acceso rapido y un entorno gráfico excelente.

2.B. ESQUEMA DE BASE DE DATOS

- Esta característica estaba deshabilitada, sentimos no poder hablar de ella. En el esquema de funcionamiento del Rational Rose de IBM se habla de que soporta todas las funciones del sistema WebSphere² que es un desarrollo software de un entorno de trabajo muy completo. Si esto es así, la evolución natural de este software es el uso conjunto con el desarrollo Eclipse en su versión WebSphere.
- Además el Rational Rose cuenta con una evolución encaminada directamente a este manejo de bases de datos, el IBM Rational Rose Data Modeler ©³. A pesar que nuestra estructura la base de datos no es fundamental, lo que si es cierto es que un manejo intuitivo y automatizado de una base de datos es imprescindible. Posee un entorno de modelado sofisticado con una transformación flexible entre los modelos lógicos y físicos y capacidades avanzadas de modelado visual, permitiendo a los desarrolladores ver como acceder la aplicación a la base de datos. Tanto modelado Entidad-Relacion como modelado Orientado a Objetos se realizan de la misma manera y con la misma notación: Unified Modeling Language.

2.C. GENERACIÓN DE CÓDIGO

- Tras parametrizar e instanciar la generación de código sobre una estructura de datos comprobamos que esta generación se realiza con menos codificación sobre el teclado y se automatiza en

² <http://www-306.ibm.com/software/websphere/>

³ http://www-306.ibm.com/software/info/ecatalog/es_ES/products/H106683S95271V02.html?&S_TACT=none&S_CMP=none

sobremanera (a veces demasiado ya que el programa intenta realizar cosas por su cuenta que se escapan de la voluntad del programador; hay que estar atentos a que no genere más código de lo estrictamente necesario).

A pesar que este producto esta pensado para un ciclo evolutivo en espiral, se adapta muy bien al ciclo en cascada al que la empresa se enfrenta.

Independientemente del tipo de plataforma o aplicación (Eclipse, Java, .NET o o aplicaciones integradas o de informática móvil), Rational le ayuda en todo el proceso de principio a fin.

2.D. INGENIERIA e INGENIERÍA INVERSA.

La creación y edición de diagramas de actividad a partir de un código ya implementado es muy funcional ya que las clases y métodos sobre los que se actúan se ven rápidamente interpretados en diagramas de actividad. Esto no funciona en ingeniería (No por hacer un diagrama de actividad se genera el código del método).

La ingeniería a partir de las clases existentes es la habitual en todas las herramientas CASE hasta ahora.

2.E. DISEÑO

Como hemos hablado hasta ahora el entorno de diseño es muy atractivo y los cambios rápidos en el código conforme vamos desarrollando el modelado es muy bueno.

Un diagramado simple, usando diagramas sin formas, diagramas de topicos y diagramas de exploración, simplifican el uso de la notación UML en el diseño, documentación y comprensión de los elementos de diseño capturados en el modelado UML.

Componentes Drag-and-drop, para facilitar el modelado.

APIs abiertas para poder incluir tus propios desarrollos.

Permite la creación de patrones de diseño de forma sencilla. Permite la exportación de patterns, de plantillas, que me permitan desarrollar el software de igual manera en el futuro.

2.F. REPORTES

Han implementado una herramienta (IBM Rational © QualityArchitect-RealTime) que automatiza la generación, ejecución y exportación de los resultados de los test realizados sobre la estructura UML. Minimiza el esfuerzo de los test y permite este testeo desde un principio ya que esta realizando un test continuo sobre el desarrollo.

Generación HTML, XML y PDF de los reportes.

2.G. AYUDA

El acceso se realiza mediante ayuda orientada y plantillas específicas para cada tarea, que se actualizan de forma periódica online. Incluye soporte técnico, servicios profesionales y formación basada en web con instructor.

Menús contextuales mediante clic derecho de ratón.

2.H. WEB

La creación online de documentación WEB, así como la exportación directamente sobre el web permiten el desarrollo simple de la publicación online para compartir datos estructurados.

VIII.3 BORLAND TOGETHER

3.A. DIAGRAMAS

Posee un potente motor de diagramas que soporta la creación, edición, almacenamiento, documentación e impresión de la mayor parte de los diagramas de **UML 1.3** y además añade otros propios para el modelado de los procesos de negocios y las aplicaciones web en tiempo . Estos son: EJB (Enterprise Java Bean) Assembler, Entity Relationship, Business Process, Robustness Analysis, Web Application, Enterprise Application, TagLib y XML Structure.

3.B. GENERACIÓN DE CÓDIGO

Válido para diversos lenguajes: Corba IDL, Java, C++, C#, VisualBasic y VBNet. La generación de código se realiza a partir del diagrama de clases. El proceso se realiza de forma sincronizada entre diagrama y código del editor. Esto quiere decir que cualquier cambio en el diagrama se traduce inmediatamente a código y viceversa. La sincronización en código esta soportada para todos los lenguajes de generación disponibles.

Incorpora un completo entorno de edición que permite realizar las típicas operaciones de cortar, copiar, pegar, seleccionar, etc.

Puede elegir entre diversos esquemas de funcionamiento del editor: como editor de texto plano, editor de código para el lenguaje Java, editor para C++, para ficheros IDL y para HTML y JSP. Para el resto de lenguajes también se puede usar el editor, la pega es que **no diferenciará las palabras reservadas**, ni permitirá una navegación tan sincronizada entre elementos del diagrama y explorador y código en el editor.

Permite tener accesible varias ventanas de código, que pueden corresponderse con clases de un diagrama que esta abierto, o clases que se pueden seleccionar del explorador de proyectos.

De esta forma cuando alguno de estos elementos se selecciona automáticamente se muestra en el editor. Permite seleccionar como queremos resaltar la sintaxis (tamaño de letra, color, tipo, etc) del lenguaje de programación que se está usando en la ventana actual del editor.

Usar CodeSense para que automáticamente nos complete la definición de métodos soportados, esta opción sólo esta disponible para el lenguaje Java. Usar abreviaturas (snippets) que sustituyen a fragmentos de código, etc.

En cuanto a su uso como entorno de **compilación** es muy útil, ya que permite realizar la compilación y generación de ejecutable de un proyecto sin tener que abandonar Together, salvo que para el lenguaje en cuestión no se disponga de compilador. Permite compilar código de Java y generar el ejecutable, usando un compilador que por defecto se instala con Together. Otra opción es utilizar un compilador externo e indicarlo en las opciones de compilación, especificando su ubicación. No dispone de compilador interno para C++, pero si que permite utilizar un compilador externo asociado. Para el resto de lenguajes no se soporta la compilación ni asociación externa de compiladores. Las opciones que permite la compilación interna son: depurar el código estableciendo breakpoints en la pantalla del editor asociado al código fuente. Poder visionar las acciones y errores en un panel de mensajes. Saltar a la línea correspondiente de código mediante la selección de un error del panel.

En lo que se refiere a la generación de **documentación** ofrece diversas posibilidades de generar y visualizar documentación a partir de los datos introducidos en los modelos. Generar un documento HTML. Imprimir la documentación en una impresora o encapsularla en un fichero .pdf. Otra opción que se permiten es la definición de patrones específicos para generar documentación.

3.D. INGENIERIA E INGENIERIA INVERSA

Permite generar diagramas de clases a partir de código fuente y viceversa. Pero en la versión 6.1 sólo esta disponible para el lenguaje Java.

3.E. DISEÑO

Una herramienta de diseño debe ser de fácil manejo y debe mantener una concordancia entre los diagramas realizados y el código implementado automáticamente. La sencillez de manejo y la visibilidad durante el mismo, consigue que el desarrollo inicial de las aplicaciones se haga de manera sencilla y sea comprensible. Together cumple con estos requerimientos con creces, puesto que además de poseer una concordancia entre ventanas perfectamente complementadas con la facilidad de uso y comprensión, garantizan un buen resultado gracias a la rapidez de adaptación que ofrece Together a los usuarios que no hayan tenido una toma de contacto previa con este tipo de programas.

La coherencia en su conjunto de este programa, dan por tanto un buen resultado en el diseño.

El proceso se realiza de forma sincronizada entre diagrama y código del editor. Esto quiere decir que cualquier cambio en el diagrama se traduce inmediatamente a código y viceversa. La sincronización en código esta soportada para todos los lenguajes de generación disponibles.

3.F. REPORTES

Together permite la creación de reportes específicos y personalizados a petición y gusto del usuario. Lo hace con todo tipo de detalle, cabecera, tipo de paquete adjunto, atributos, son sólo algunas de las características que esta opción presenta. Una vez creada la plantilla del reportes específico, se optimiza, tanto el tiempo como la comodidad de realización de estos reportes.

3.G. AYUDA

La ayuda que ofrece Borland Together es de tipo local y de manera online. Se incluye una completa Documentación API para consultas sobre programación orientada al uso del Together, lo que permite que dentro de una sola aplicación se puedan hacer referencias a diversos tipos de documentación API.

3.H. WEB

Together es capaz de crear páginas web simples en html con los diagramas obtenidos y patrones preestablecidos. Del mismo modo, como herramienta fundamental, destaca el Browser UDDI cuya característica fundamental y más importante es la creación de clientes Proxy basados en archivos WSDL. Programas soportados por J2EE hacen de Together un ejemplo a seguir por sus competidores más directos.

VIII.4 POWER DESIGNER

4.A. DIAGRAMAS

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; y CRUD Matrix, que define el efecto de un proceso de datos en términos de Crear, Leer, Actualizar, y Borrar operaciones (CRUD).

4.B. ESQUEMA DE LA BASE DE DATOS

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 Martin⁴.
- Nivel físico: tablas, columnas, dominios, llaves primarias, llaves foráneas, llaves alternadas, índices, constraits de integridad referencial declarativa, vistas, parámetros de almacenamiento físico, reglas de negocios, triggers y procedimientos almacenados.

4.C. GENERACIÓN DE CÓDIGO

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 PowerBuilder, 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 PowerBuilder 4.0 y 5.0; Visual Basic 3.0, 4.0, y 5.0; Delphi 2.0; Power++; y el Web.

- Generación de objetos PowerBuilder. Soporta todas las ediciones de PowerBuilder 4.0 y 5.0. Genera objetos personalizables de PowerBuilder y componentes basados en modelos de bases de datos físicos y plantillas que se encuentran dentro de las librerías de clases de su elección. Genera objetos ventana y ventana de datos basadas en tablas, vistas y relaciones de llaves primariasforáneas. Genera y hace ingeniería de reverso a los atributos. Incluye plantillas personalizables para la librería PowerBuilder 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 primariassecundarias. Genera proyectos basados en modelos de propiedades. Genera 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.

⁴ <http://sistemas.dgsca.unam.mx/publica/pdf/casestru.pdf>

4.D. INGENIERÍA e INGENIERÍA INVERSA

Ingeniería hacia delante

Común a los anteriores.

Ingeniería Inversa

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 PowerBuilder, soporta definición de atributos extendidos para PowerBuilder, Progress, Uniface, PowerHouse, Axiant, y NS-DK.

4.E. DISEÑO

MetaWorks es un sistema diseñado para proveer los módulos gráficos de PowerDesign con la habilidad de compartir y almacenar modelos de datos en un solo punto de control, el Diccionario MetaWorks. 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.

4.F. REPORTE

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.

4.G AYUDA

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

4.H. WEB

Opcion deshabilitada en la evaluación

VIII. 5 CASE 4/0⁵

5.A. DIAGRAMAS

Case 4/0 asegura que los resultados administrados en una configuración sean consistentes. Los diagramas (como modelos de ER o diagramas de flujo) así como los resultados del desarrollo que no sean representados gráficamente (como elementos de datos o scripts) pueden versionarse en la configuración. La exportación e importación de funciones para la configuración asegura la consistencia en la distribución e integración de los resultados del desarrollo.

5.B. ESQUEMA DE LA BASE DE DATOS

Case 4/0 soporta la implementación en C/C++, COBOL, Java y Visual Basic, genera definiciones de bases de datos para ORACLE, SQL Server, DB2 y MS Access. Además permite la generación de código necesaria para satisfacer las necesidades. El usuario define el diseño de elementos – funciones, relaciones, pantallas, etc.

5.C. GENERACIÓN DE CÓDIGO

Case 4/0 soporta los lenguajes C/C++, COBOL, Java y Visual Basic. El código se escribe en un lenguaje específico de Case 4/0 que es parecido a Basic. Se tiene acceso a la especificación del repositorio case 4/0 mediante un script y utilizando la generación de código. Los cambios que se vayan realizando surten efecto automáticamente en el código generado. El script cifrado con case 4/0 abre una amplia variedad de posibles usos: pantallas de unidad central, código 4GL y la generación de datos de testeo, son sólo unos pocos ejemplos de los que podrían alcanzar nuestros clientes.

Si usted encuentra errores fuera de case/4/0 en el curso de pruebas y eliminación de fallos, usted puede corregirlos inmediatamente. Una vez que su código aumenta y lo ejecuta, usted puede unirlo de nuevo con el sistema case/4/0 simplemente presionando una llave.

El resultado directo de funcionamiento con case/4/0 es el código original compilable.

5.D. DISEÑO

El diseño se basa en el modelo lógico de datos del sistema de análisis.

El diseño de la arquitectura, los pasos consisten en lo siguiente, primero, en la división del sistema en unidades implementables contienen funciones software desarrolladas durante la especificación del problema que puede ser automatizada, junto con los datos utilizados para ello, y segundo, en el desarrollo, en el aspecto de reusabilidad, mediante plantillas de tecnología base.

⁵ CASE 4/0 es un programa open source en el que el código puede ser recompilado.

Pueden ocurrir diversos problemas; redundancias, inconsistencias, diseños incomprensibles, capacidad de mantenimiento pobre.

5.E. REPORTE

Para proyectos grandes y distribuidos en LAN o mediante accesos por servicio remoto, redes privadas, orientadas por transacción. Una interfaz de importación/exportación permite la consistencia en el intercambio de los proyectos distribuidos.

La creación de reportes estructurados a través de una plantilla base de reportes

5.F. AYUDA

La ayuda de Case 4/0 es usual y adecuada al contexto en cada momento.

5.G. WEB

El Editor de Web Case/4/0 hace segura la comunicación que corre suavemente, incluso en equipos distribuidos.

IX. RESULTADOS FINALES

Debido a que el entorno de trabajo sobre el que nos movíamos no era completo, ya que los elementos del trabajo eran versiones de evaluación no podemos desarrollar una conclusión fiable basada en métricas. No podíamos evaluar unas características en unas herramientas y dejarlas sin puntuar en las otras.

En general estamos bastante contentos con tres de las cinco empleadas. Seguramente puntuaríamos por debajo la herramienta Together, ya que se encuentra obsoleta y es menos intuitivo en su diseño que el resto. Además sacaríamos de la decisión final a la herramienta CASE 4/0 ya que se programa en un lenguaje propietario.

De entre las tres restantes la interfaz que supera al resto es la del Rational Rose de IBM ya que es muy intuitiva y completa. El menu contextual que se expande de las opciones es el más completo de los tres.

Asi mismo la forma de publicar entre el Power Builder y el Rational Rose podríamos clasificarlo en empate. El Power Builder te permite seleccionar los campos del informe que quieres sacar, mientras que el Rational Rose permite la generación variada de este reporte. Tal vez sea mas funcional la de la herramienta de IBM, pero el Power Builder mete mucha mas información.

La Ingeniería e Ingeniería Inversa se presenta de forma bastante parecida en los tres que hemos elegido, si bien la característica de permitir crear diagramas de secuencia que presenta la Rational Rose de IBM es un paso más sobre el resto de las herramientas de SyBase.

La generación de código la ganan por goleada el software de Sybase ya que permite regenerar el código en muchos y diferentes lenguajes de programación. El software de IBM centra sus esfuerzos en la codificación de los lenguajes mas importantes y extendidos. Los Power hacen mucho y bien y el Rational presenta pocos pero mejor... Aun asi, elegiríamos el de Sybase ya que no nos habla del lenguaje en el caso.

El esquema de la Base de Datos no podemos evaluarla entre los que nos quedan ya que el Rational Rose cuenta con un software externo para este manejo.

Precio. Aunque no es fundamental si que resulta importante mirar este punto. El software de IBM es mas asequible para poder controlar la inversión y sacarle el máximo rendimiento. Permite multiples licencias (10 para ser exactos)

Si tenemos que elegir uno y a la espera de la nueva versión del Power Designer nos quedaremos con el **Rational Rose de IBM**

X. INTERNETGRAFIA

- <http://www.cs.queensu.ca/Software-Engineering/toolcat.html>
- <http://www.logicworks.com/products/bpwin/bpwin.html>
- <http://www.aisintl.com/case/products/product.html>
- <http://www.sybase.com/products/powerdesigner/>
- <http://www.monografias.com/trabajos6/vica/vica.shtml#obje>
- <http://www.sybase.es/iberia/products/powerdesigner.jsp>
- <http://www.sybase.com/detail?id=1003267>
- <http://www.microsoft.com/spain/office/products/visio/default.mspx>
- <http://www.powerbuilder.org>
- <http://www.abits.com/Borland.htm>
- <http://sistemas.dgsca.unam.mx/publica/pdf/casestru.pdf>
- <http://www-306.ibm.com/software/rational/offerings/design.html>
- http://www.cs.rhul.ac.uk/CompSci/Computers/rational/html/rose_REI_guide/REIGdeChapter2.html
- <http://www.nrt.se/nrt/krav/Losningar/RTM-ROSE%20TIB.pdf>
- <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/datasheets/rsm.pdf>