

TEMA 58

AYUDAS AUTOMATIZADAS PARA EL DESARROLLO DE “SOFTWARE” (HERRAMIENTAS CASE). TIPOS. ESTRUCTURA. PRESTACIONES.

ÍNDICE

1. INTRODUCCIÓN
2. PRESTACIONES
3. CLASIFICACIÓN
4. COMPONENTES
5. INTEGRACIÓN
 - 5.1. El diccionario de datos
6. CASE Y EL CICLO DE VIDA CLÁSICO
7. UTILIZACIÓN DE CASE
 - 7.1. Structured Architec
 - 7.2. Exceleator
8. ELECCIÓN DE HERRAMIENTA
9. BIBLIOGRAFÍA

1. INTRODUCCIÓN

A finales de los años setenta hay un fuerte tirón en la demanda de aplicaciones institucionales de gran complejidad con requerimientos muy exigentes a veces difíciles de identificar. Estos dos factores, aumento de demanda y aumento de complejidad, contribuyen a una importante crisis del software, que no es capaz de dar solución con las herramientas y metodologías existentes a las peticiones de los usuarios, a esto se le puede sumar el estado en que se encontraba el hardware.

A partir de estos momentos hay una evolución en varias líneas o frentes:

- Surgen nuevas técnicas metodológicas como superación del ciclo de vida clásico (técnicas estructuradas).
- Empiezan a surgir nuevas herramientas de desarrollo basadas en el ordenador.
- El hardware evoluciona, aparecen nuevos entornos, máquinas cada vez más pequeñas y a la vez más potentes sobre las que son posibles nuevos sistemas operativos y nuevos entornos de desarrollo basados en las nuevas metodologías y herramientas.

En este escenario en los años ochenta nace el término **CASE**, Computer Aided Software Engineering, o Ingeniería de Software Asistida por Ordenador, una evolución dentro del desarrollo de sistemas de información que pretende ser la solución a la crisis del software tal y como la hemos descrito anteriormente, es decir, en el entorno institucional con aplicaciones cada vez más complejas.

CASE no es identificable con una herramienta sofisticada ni con una determinada metodología, si hubiera que definirlo podríamos decir que es una nueva "filosofía" que intenta conducir el desarrollo de software hacia una disciplina real de ingeniería, filosofía porque está por encima de una simple agregación de herramientas y métodos, ingeniería porque está a favor de métodos disciplinados y bien definidos que poco tienen que ver con la intuición y/o la artesanía. Es evidente que un cambio de filosofía lleva más tiempo que el aprendizaje de una determinada técnica y por supuesto un alto volumen de inversiones.

Desde un punto de vista más práctico, CASE puede percibirse, en principio, como:

- La automatización del desarrollo del software.
- La informatización de la informática.
- La redefinición del entorno software.

Profundizando más podemos estudiar CASE basándose en sus objetivos, entre los que podemos citar:

- Conseguir la aplicación práctica de las metodologías estructuradas.
- Mejorar la calidad del software producido.
- Acelerar el proceso de desarrollo.
- Facilitar la reutilización del software.
- Crear estándares de documentación.
- Disminuir y simplificar el mantenimiento de programas.
- Incrementar la "portabilidad" de las aplicaciones.
- Realizar prototipos en entornos complejos.

Para conseguir sus objetivos, CASE se basa tanto en el nuevo hardware, mucho más potente, como en las nuevas metodologías y herramientas, redefiniendo, como se ha dicho, un nuevo entorno de desarrollo, basado fundamentalmente en la utilización de metodologías actuales, herramientas CASE y el diccionario de datos integrado.

2. PRESTACIONES

Automatizando nuestras herramientas podemos aumentar sensiblemente los beneficios derivados del uso de las mismas, esta es definitiva la idea que da origen a este tipo de herramientas, generadores de aplicaciones, códigos y pantallas, sistemas de documentación automatizada, sistemas de diccionarios de datos...

Los beneficios son múltiples, todos los que teníamos y otros entre los que se podrían citar:

Disminución de tiempo que trae consigo cualquier automatización, cabe pensar, por ejemplo, en el desarrollo por prototipos.

Automatización de tareas rutinarias como el dibujo de diagramas en el caso del Análisis Estructurado.

Garantizar procesos consistentes asegurándonos que ciertos procedimientos cumplan las reglas establecidas, como por ejemplo la explosión de los diagramas de flujo de datos.

Registro de los detalles del sistema, se pueden capturar y almacenar los datos y detalles de un sistema durante su desarrollo con objeto de recuperarlos y procesarlos con distintas finalidades en sucesivos momentos del proyecto o, incluso, en proyectos diferentes.

3. CLASIFICACIÓN

En general, podemos clasificarlas en tres categorías:

- Herramientas de alto nivel.
- Herramientas de bajo nivel.
- Herramientas integrales e integradas.

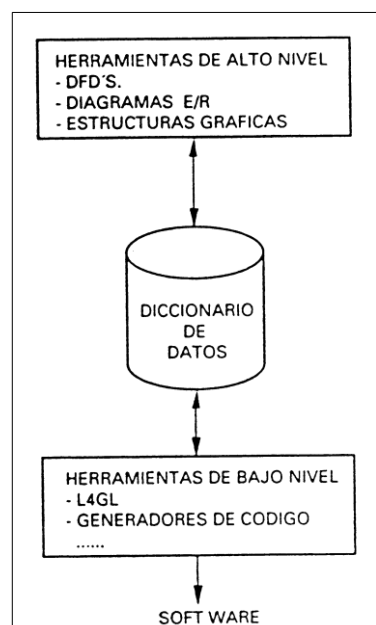


Figura 58.1. Herramientas de alto y bajo nivel

Herramientas de alto nivel. Las herramientas de alto nivel, también denominadas de tipo “front-end”, automatizan las fases correspondientes del proyecto, ayudando al analista en la preparación de especificaciones y descripciones del sistema, proporcionando, en general, un soporte gráfico, del que podrían ser un ejemplo representativo los diagramas de flujo del sistema.

Herramientas de bajo nivel. Este tipo de herramientas, denominadas también de tipo “back-end”, ayudan al analista a describir los algoritmos y lógica de programas, descripción física de datos e interacciones con los dispositivos de entrada y salida, convirtiendo los diseños lógicos del software en código, en un lenguaje de programación que, en definitiva, conforma la aplicación informática, por este motivo también se conoce a estas herramientas como de “programación asistida por ordenador”.

Herramientas integradas. La actividad del análisis es un todo que comienza en los requerimientos del usuario «traducidos» a especificaciones de «alto nivel», dando lugar a un proceso de descomposición que finaliza en unas especificaciones de «bajo nivel» programables, por tanto, lo deseable sería una herramienta que combinara los dos tipos vistos hasta ahora.

En general, las herramientas de tipo integral han sido poco corrientes, quedando como responsabilidad para el analista el paso del “front-end” al “back-end”. Es evidente que la integración, además de ahorrar tiempo, daría seguridad a la transición alto-bajo nivel.

Las herramientas integrales surgen generalmente asociadas con metodologías de desarrollo concretas y/o vinculadas a un determinado fabricante, la clave de la transición suele estar en el *diccionario de datos*, pieza clave en este entorno, así como en el ambiente CASE.

4. COMPONENTES

A continuación se exponen, con excepción del diccionario de datos que, a pesar de que puede englobarse como un componente más, debido a su importancia y trascendencia, se tratará en un punto aparte, los componentes más comunes de una herramienta CASE:

Utilidades de «diagramación». Las facilidades para producir diagramas de flujo de datos son fundamentales como apoyo a la metodología de análisis estructurado, las herramientas de diagramación ofrecen la capacidad de dibujar diagramas, guardando todos los detalles de forma interna, como soporte al análisis y documentación de requerimientos de la aplicación. Las herramientas CASE suelen incorporar uno o varios métodos de análisis estructurado.

La capacidad de reutilizar, cambiando y volviendo a dibujar los diagramas facilita enormemente una actividad que los analistas consideran aburrida y pesada.

Generadores de interfaces. Basándose en la facilidad de crear menús y pantallas para la demostración del sistema, así como formatos de informes, los generadores de interfaces son fundamentales en la preparación de prototipos de este entorno, así como en los demás métodos de desarrollo.

Generadores de código. De acuerdo con las especificaciones de bajo nivel, los generadores de código obtienen código ejecutable, si bien esta conversión aún no está muy perfeccionada.

En conexión con el diccionario integrado se puede conseguir además código reutilizable, de manera que ante un cambio en las especificaciones, se pueda volver a generar el código a partir de los detalles registrados en el diccionario.

Utilidades de administración. Algunas herramientas CASE ayudan al jefe de proyecto a planificar y llevar un control sobre todo el proceso de ejecución temporalizando y asignando recursos a ellas, también ayudan en la realización de informes.

5. INTEGRACIÓN

Aunque lo deseable es que las herramientas CASE sean integrales, algunas se presentan como elementos discretos considerables por separado y otras como un grupo integrado de herramientas. Comercialmente esto se conoce como “toolkit” y “workbench”, respectivamente.

La integración puede conseguirse de tres formas en general:

- Mediante una interfaz uniforme para todas las actividades.
- Mediante la transferencia de datos entre los distintos componentes.
- Creando enlaces, bien manuales o automáticos, bajo la responsabilidad del analista.

En cualquier caso es importante señalar que la integración siempre pasará por el diccionario de datos.

5.1. El Diccionario de Datos

El diccionario de datos es un **catálogo, depósito de información o repositorio**, que contiene información amplia de todos los componentes del sistema, datos, procesos, flujos de datos, entidades, etc.

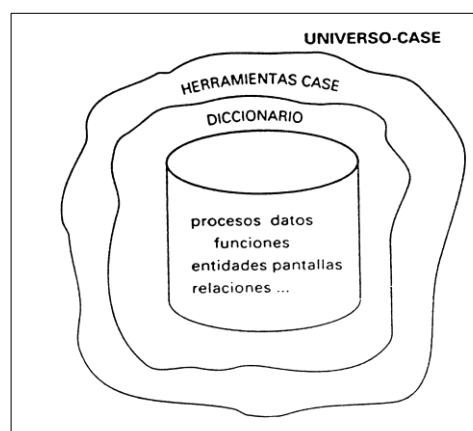


Figura 58.2. El diccionario de datos

Está basado en una arquitectura de bases de datos fácilmente accesible aportando funciones básicas para el uso de la herramienta en general y facilidades sobre el uso del propio diccionario en particular.

Entre las *funciones* que aporta se pueden citar:

- Integración de componentes.
- Compartición de información.
- Estandarización de la documentación.
- Generación de la documentación.
- Generación de código.
- Reutilización y control de proyectos.
- Gestión y control de proyectos.

Entre las *facilidades* que puede proveer están:

- Control de seguridad.
- Concesión de privilegios.
- Bloqueos.
- Auditoría.
- Control de cambios.

6. CASE Y EL CICLO DE VIDA CLÁSICO

Si consideramos un ciclo de vida clásico simplificado compuesto por análisis, diseño, codificación y pruebas, podemos ver cómo los tiempos dedicados a cada actividad que de por sí en términos absolutos se reducen, se distribuyen de manera distinta con la aplicación, primero de técnicas estructuradas, y después de CASE.

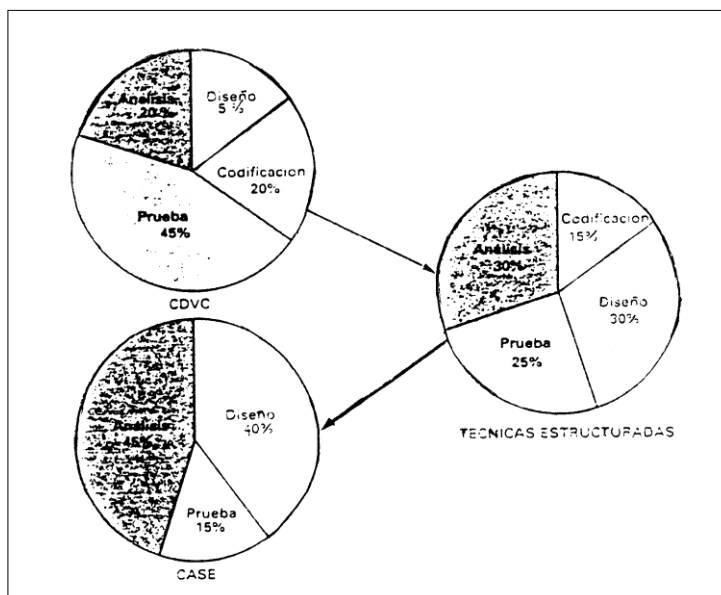


Figura 58.3. Del CDVC a CASE

7. UTILIZACIÓN DE CASE

En principio habría que decir que no todas las herramientas son iguales ni corren en los mismos entornos ni tienen la misma orientación o finalidad, esta tema no pretende dar una guía de uso para las herramientas ni mucho menos.

Hay herramientas muy completas adaptadas a varios entornos y con una orientación amplia como es el caso de **Excelerator** (de Index Technologies), algunas ofrecen incluso versiones educacionales de un alcance más reducido y orientación más concreta como Structured Architect (de Meta Systems); por último, algunas están fuertemente vinculadas a un fabricante como, por ejemplo, Predict CASE (de Software AG) u Oracle CASE (de Oracle), en este punto hablaremos de forma muy general de las dos primeras.

7.1. Structured Architect

La versión educacional de “arquitecto estructurado” es un ejemplo de herramienta simple, fácil de usar y asequible que puede utilizarse sobre PC XT/AT y compatibles, está diseñado para implementar exclusivamente la metodología de DeMarco para el análisis estructurado.

Su núcleo es también el diccionario denominado aquí enciclopedia, ofreciendo un entorno gráfico integrado con ella, con lo que permanentemente se encuentra actualizada sin admitir redundancias, utiliza para su manejo una técnica de menús. Además tiene los siguientes elementos:

- **Editor de diagramas** (Edit diagram). Permite dibujar, editar, definir e introducir información sobre los flujos de datos, puede introducirse información descriptiva para datos u objetos utilizando su propio editor de textos y posee procesos de análisis e impresión mediante los cuales se puede comprobar la completitud y consistencia de los diagramas balanceando padres e hijos por ejemplo.

- **Enciclopedia** (Encyclopedia). Permite construir y mantener el diccionario de datos. Enciclopedia y editor están integrados de forma que la información en uno de ellos está disponible en el otro, la enciclopedia captura, ordena y cataloga información que posteriormente puede ser visualizada o impresa, está dividida en dos partes: enciclopedia para procesos y fuentes y enciclopedia de datos, accesibles mediante menús.

- **Análisis** (Analyses). Se pueden realizar distintas funciones en este sentido, entre ellas:

Balancear un proceso con su diagrama: asegurándonos que todos los flujos que entran o salen de un diagrama padre, también entran o salen de un diagrama hijo.

Identificación de datos obsoletos.

Identificar procesos independientes: no tienen diagrama del que dependen o «padre».

Identificar procesos sin descripción.

Identificar datos sin descripción.

Identificar procesos sin entrada o salida de datos.

- **Informes** (reports). Los informes se pueden obtener por pantalla o impresora, podemos obtener informes diversos, entre ellos:

Listar todos los objetos.

Listar los objetos de un tipo especificado, procesos, datos, fuentes...

Listar requerimientos.

Estructura de procesos.

Estructura de datos.

Resumen de procesos.

Imprimir todos los diagramas.

7.2. Excelerator

Excelerator es una de las herramientas más reconocidas y comunes en la actualidad, soporta varias metodologías estructuradas y está disponible en varios entornos (PC XT/AT/PS-2, SUN y APOLLO), su uso es bastante simple mediante menús.

El núcleo es el diccionario de proyectos, en torno al cual existen diferentes *funciones*, que son:

- **Gráficas** (Graphics), permite realizar varios tipos de diagramas y cuadros para análisis y diseño, a través de ellas el analista ve la lógica del sistema desde distintas ópticas y niveles de abstracción, puede facilitar la comunicación entre usuarios y técnicos.
- **Diccionario** (XIdictionary), es la puerta de acceso al núcleo, tiene depositada información sobre datos y procesos, diagramas, pantallas e informes, permite la producción de informes sobre el contenido de todas las entradas al mismo, a través de él se coordina toda la información requerida en la construcción del sistema y se realiza la gestión del proyecto.
- **Pantallas e informes** (Screens and reports), permite realizar diseños y pruebas de pantallas e informes, es de gran utilidad en el desarrollo incremental o de prototipos.
- **Documentación** (Documentation), permite crear el documento de especificaciones del sistema por completo para imprimirlo a través de él o enlazando con un procesador de textos.
- **Análisis** (Analysis), realiza la validación de los gráficos verificando su estructura y consistencia, ayudando así a la localización tanto de errores como de datos duplicados a corto plazo, puede realizar e imprimir listados sobre entidades del depósito de datos, así como informes del proyecto.
- **Interfaz** (XLD Interface), facilita el manejo de funciones para varios usuarios desde el punto de vista de la integridad y seguridad, permitiendo o no importación de archivos, bloqueo, etc.

- **Administración** (Housekeeping), aporta facilidades para crear y mantener el ambiente del proyecto como el establecimiento de dispositivos y opciones de salida, la adición y cambio de niveles de autorización, usuarios y proyectos, etc.

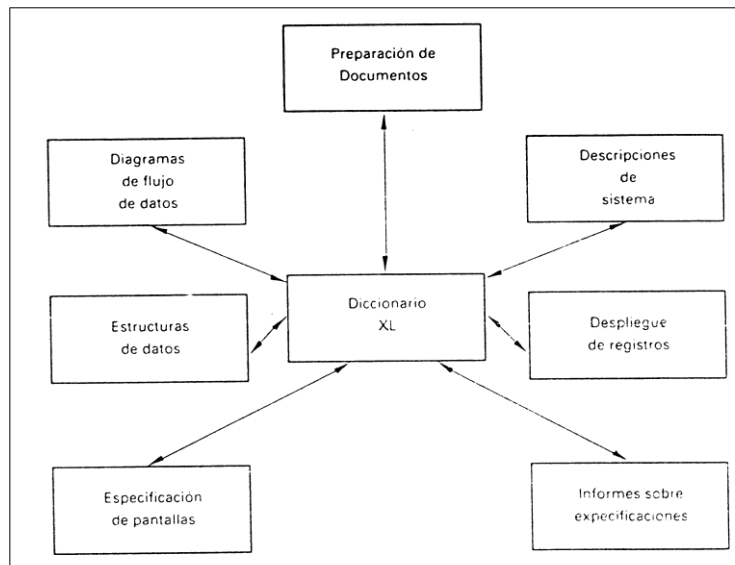


Figura 58.4. El núcleo de EXCELERATOR

8. ELECCIÓN DE HERRAMIENTAS

La respuesta a esta pregunta es muy sencilla, la que se tenga, el problema es que realmente se disponga de alguna.

En las grandes empresas (incluidas las de servicios informáticos) la evolución en el desarrollo llevará a este tipo de herramientas, en empresas pequeñas esta evolución aún tardará en llegar.

Las herramientas CASE no suelen ser baratas, por lo que es difícil que los centros de enseñanza dispongan de ellas si no es bajo versiones educacionales o acuerdos de este tipo; así pues, en este caso, la educación, la recomendación es utilizar lo que se tenga e intentar llegar a acuerdos con las distintas casas comerciales para obtener versiones educacionales.

Como se ha dicho, es en el entorno de las grandes empresas donde la evolución en el Desarrollo llevará casi con toda seguridad a las herramientas CASE, en este caso será la alta dirección la que decidirá cuál utilizar; sin embargo, como en todo cambio, existen algunos *obstáculos* para la implementación de CASE. Los más comunes son:

- Las herramientas CASE son caras.
- Los beneficios no son tan claros.
- Las herramientas son muy diversas.
- La formación es muy cara.

9. BIBLIOGRAFÍA

Pressman, Roger S.
Ingeniería del Software
 Mc Graw-Hill, 1993

López, Antonio
Metodologías de Desarrollo
 Ra-ma, 1990