

ANÁLISIS Y DISEÑO ASISTIDO POR ORDENADOR: HERRAMIENTAS CASE

11.1.- Introducción

Bajo el nombre de herramientas CASE (Computer Aided Software Engineering) se agrupan una *serie de productos destinados a la automatización de la producción de software*, esto es, al *desarrollo asistido de sistemas informáticos*.

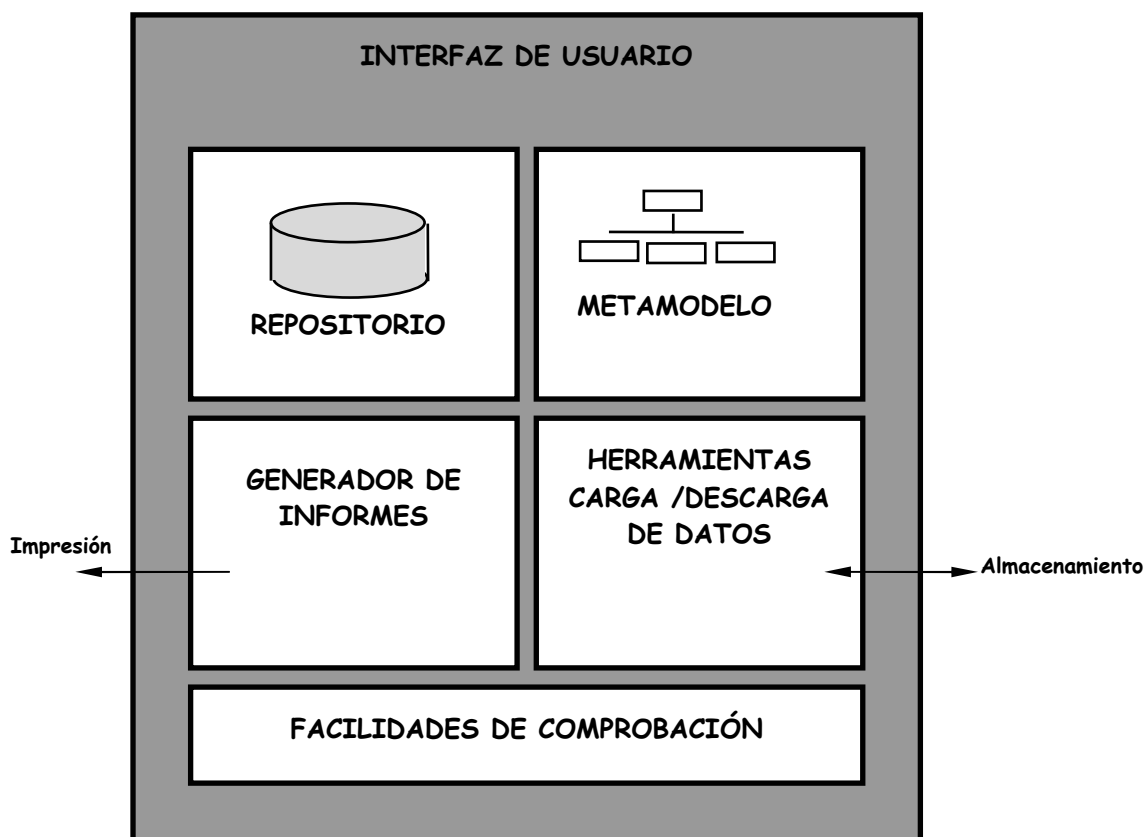
La tecnología CASE supone la "*informatización de la Informática*", es decir, la "automatización del desarrollo del software", contribuyendo a elevar la productividad y la calidad en el desarrollo de sistemas de información.

Este nuevo enfoque a la hora de construir software persigue mejorar la calidad y la productividad de los sistemas de información, para lo que se plantean los siguientes objetivos:

- * *Permitir la aplicación práctica de metodologías estructuradas, lo que resulta muy difícil sin emplear herramientas*
- * *Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones*
- * *Simplificar el mantenimiento de programas*
- * *Mejorar y estandarizar la documentación*
- * *Aumentar la portabilidad de las aplicaciones*
- * *Facilitar la reutilización de componentes software*
- * *Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos*

De forma esquemática, una herramienta CASE se compone de los siguientes elementos:

- **Repositorio** (diccionario) donde se almacenan los elementos definidos o creados por la herramienta, y cuya gestión se realiza mediante el apoyo de un SGBD o de un sistema de gestión de ficheros
- **Metamodelo** (no siempre visible), que constituye el marco para la definición de las técnicas y metodologías soportadas por la herramienta
- **Generador de Informes**, que permite obtener la documentación que describe el sistema de información desarrollado; documentación que está asociada a las técnicas y metodologías empleadas.
- **Carga/Descarga de datos**, facilidades que permiten cargar el repositorio de la herramienta CASE con los datos procedentes de otros sistemas, o bien generar a partir de la propia herramienta esquemas de bases de datos, programas, etc. que pueden, a su vez, alimentar otros sistemas.
- **Comprobación de errores**, facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta
- **Interfaz de usuario**, que constará de editores de texto y herramientas de diseño gráfico que permitan, mediante la utilización de un sistema de ventanas, iconos y menús, definir los diagramas, matrices, etc. que incluyen las distintas metodologías.



11.2.- Categorías de herramientas CASE

En general, pueden distinguirse tres tipos de herramientas:

- * **Herramientas de gestión**, encargadas de la estimación, planificación y seguridad del proyecto
- * **Herramientas técnicas**, que, tradicionalmente se dividen en:
 - * *CASE frontales o superiores*, que abarcan las primeras fases del diseño
 - * *CASE dorsales o inferiores* cuyo objetivo es el diseño detallado y la generación de código
- * **Herramientas de soporte**, como el sistema repositorio/diccionario, control, configuración, seguridad, etc.

Se denomina **ICASE** (Integrated CASE) a las herramientas que engloban aspectos tanto de CASE frontales como de CASE dorsales, contemplando, por tanto, todo el ciclo de desarrollo.

Así mismo, se denomina **IPSE** (Integrated Programming Support Environment) a aquellas herramientas que, además, incluyen componentes para la gestión de proyectos y la gestión de configuración.

11.3.- Repositorio/Diccionario de recursos de información

Los repositorios y los sistemas de diccionarios de recursos de información (**SDRI**) se configuran en los entornos CASE como un elemento esencial para el desarrollo de los sistemas de información.

Históricamente nacen con la necesidad de describir los datos en una aplicación, así como algunas de sus características y restricciones, es decir su **semántica**.

Posteriormente, la necesidad de centralizar las descripciones de los datos para conseguir sistemas mas coherentes, eficientes y adaptables a cambios, surgen productos que permiten una mejor gestión de los datos, ofreciendo una gran independencia entre éstos y las aplicaciones: son los **Sistemas Gestores de Bases de Datos (SGBD)**.

Como un componente de los SGBD aparece el **directorío de datos**, encargado de describir dónde y cómo se almacenan los datos de la base, el modo de acceso y otras características físicas de los mismos. El directorío contiene, pues, las especificaciones necesarias para pasar de la representación externa de los datos a la representación interna de los mismos, en formato legible para la máquina; En definitiva, el objetivo principal del directorío de datos es transmitir al SGBD la información necesaria para poder acceder a los datos contenidos en la base.

Con independencia de los directoríos surgen los **diccionarios de datos** que reúnen información sobre los datos almacenados (metadatos): descripciones, estructuras, consideraciones de seguridad, uso de los datos, etc., que los usuarios necesitan para comprender el significado, esto es, el aspecto lógico de los datos.

A los paquetes que llevan a cabo tanto las funciones del diccionario como las funciones de directorío se les denomina: **diccionario/directorío de datos**.

La complejidad de los Sistemas de información (SI) ha conllevado la aparición de herramientas CASE que contienen un diccionario, llamado **enciclopedia** o **repositorio** donde se almacenan los datos generados durante el ciclo de vida de un SI: esquemas, grafos, matrices, información relativa a la gestión de proyectos, gestión de configuraciones, etc.

El nuevo concepto **diccionario de recursos de información (DRI)** pretende ser el eslabón final en la evolución de los "almacenes de datos". El DRI constituye el *depósito integrado de todos los datos sobre la organización, automatizados o no, que son utilizados para efectuar las labores de planificación, control y operación que permitan a la empresa cumplir sus objetivos*. Engloban, por tanto, las capacidades y funciones de todos los almacenes de datos anteriores

Es importante observar la existencia de dos tipos diferenciados de repositorios:

repositorios (con minúscula).- almacenan objetos para una herramienta CASE en particular. Constituyendo el núcleo de las herramientas CASE

Repositorios (con Mayúscula).- basados en estándares e implementan un modelo de información abierto y extensible, soportando un entorno integrado de ingeniería de software. Pretenden ser el corazón de toda la arquitectura de información de la empresa **sirviendo de soporte para la integración de sistemas**.

11.3.- CONTENIDO DEL DICCIONARIO DE RECURSOS DE INFORMACIÓN

Para cumplir sus objetivos, un DRI debe contener, entre otras, información relativa a las siguientes áreas:

- * **Organización** a la que va a servir, los "modelos" de empresa y "reglas de negocio"

- * *Usuarios de la organización y del sistema de información*
- * *Subsistemas que integran cada uno de los sistemas considerados*
- * *Componentes físicos y elementos de entorno que integran cada subsistema*
- * *Estructuras de datos, ficheros y BD de cada uno de ellos, así como las interrelaciones existentes entre estos datos y las reglas de integridad*
- * *Seguridad y confidencialidad de los datos*
- * *Programas, aplicaciones o rutinas de cada subsistema*
- * *Metodologías a emplear*
- * *Gestión del ciclo de vida software*

También suele contener información acerca de los propios metadatos, es decir "meta-metadatos".

Los diccionarios deben ser extensibles, de modo que cada usuario pueda definir nuevos objetos que pueda utilizar de la misma manera que los facilitados de forma estándar por el sistema.

11.4.- Herramientas de Análisis y Diseño

Dentro de esta categoría destacan las herramientas que permiten crear y modificar diagramas E/R, diagramas de flujo de datos, diagramas de estructura, diagramas de clases, etc.

También se consideran dentro de esta categoría herramientas de prototipado como:

1. Diseñadores de pantallas
2. Generadores de Menús
3. Generadores de Informes
4. Lenguajes de especificación ejecutables

Un aspecto a considerar en estas herramientas es la capacidad de análisis y verificación de especificaciones que soporte la herramienta, no sólo sintáctica, sino también semánticamente, como por ejemplo el balanceo de niveles en los DFD o la capacidad de normalizar un diagrama de datos (normalmente hasta 3FN)

Todas estas herramientas operan sobre un repositorio donde se va almacenando la información necesaria para el funcionamiento de la misma herramienta, cuyo contenido, mas o menos completo, dependerá de dicha herramienta.

Ejemplos de herramientas de este tipo son: DESIGNER/2000 de ORACLE, EASY CASE de Evergreen, EXCELERATOR de Intersolv, OBJECT MAKER de Mark IV, etc.

11.5.- Generación de código y documentación

Aunque no se puede generar en todos los casos el 100 % del código de los programas, la tecnología CASE aumenta considerablemente la productividad ya que genera buena parte del código y al programador le quede sólo por completar la implementación de los programas, o refinarlos para conseguir un mejor rendimiento.

Una cuestión importante a considerar en la generación de código es la integración en el entorno CASE de herramientas de prueba y depuración, así como facilidades de importación/exportación hacia SGBD y diccionarios externos

Las herramientas CASE soportan la creación automatizada de un conjunto muy variado de documentación obtenido a partir de la información almacenada en el directorio, que va desde la descripción textual en una especie de pseudocódigo hasta diagramas mas o menos complejos.

11.6.- Herramientas de prueba

Suelen conocerse con las siglas CAST (Computer Aided Software Testing) y las funcionalidades que suelen soportar este tipo de herramientas son:

- * **Gestión de pruebas:**
 - * *Predecir el coste y tiempo de las pruebas*
 - * *Planificar pruebas de productos, actividades y recursos*
 - * *Monitorizar pruebas*
 - * *Generar informes*
- * **Definir objetivos y requisitos de prueba:**
 - * *Registrar requisitos y /u objetivos*
 - * *Verificar requisitos*
- * **Diseñar pruebas:**
 - * *Diseñar, generar y documentar los casos de prueba*
 - * *Rastrear requisitos*
 - * *Trazar caminos, ramas y sentencias de casos de prueba*
 - * *Predecir la salida esperada*
- * **Construir entornos de ejecución de pruebas:**
 - * *Verificar código*
 - * *Implementar código*
 - * *Generar elementos de prueba*
- * **Ejecutar pruebas:**
 - * *Registrar ejecuciones de casos de prueba*
 - * *Repetir ejecuciones*
 - * *Registrar casos de prueba y salidas reales*
- * **Evaluar pruebas:**
 - * *Fallos de las pruebas*
 - * *Estadísticas de fallos*
 - * *Calidad de las pruebas*
 - * *Calidad del software (funcionalidad, Rendimiento, etc.)*

Entre estas herramientas de pruebas destacan LOGISCOPE, VERIFY, STW, etc.

11.7.- Herramientas de gestión de configuración

En entornos de desarrollo complejos, especialmente si se integran diversas herramientas de ingeniería de software, se hace imprescindible la incorporación de una herramienta capaz de gestionar la configuración de los sistemas. Este tipo de herramientas ofrecen cuatro tipos de capacidades:

- **Control de versiones**, es decir, la capacidad de proporcionar almacenamiento y acceso controlado a los datos, así como de registrar los cambios sobre los mismos y poder recuperar versiones anteriores
- **Construcción**, que consiste en automatizar las tareas asociadas a la compilación y enlace de un sistema con el fin de generar los ejecutables.
- **Trazabilidad de requisitos y análisis de impacto**, que permiten, respectivamente, rastrear un requisito hasta su implementación, y conocer los elementos del sistema que se ven afectados ante un cambio
- **Capacidades generales**, apartado en el que se engloba: la gestión del proceso, capacidades de trabajo en grupo, interfaces con otras herramientas, generación de informes, consultas, etc.

Otro aspecto que deben soportar estas herramientas es la **fusión** (merging) de elementos variantes. En este caso la herramienta, a la hora de fusionar dos variantes puede:

- Dejar sólo un elemento en caso de que ambos sean iguales
- Dejar sólo el elemento que haya sufrido cambios con respecto a la versión base
- Pedir al usuario que, de forma interactiva, vaya resolviendo los conflictos en cualquier otro caso

Para almacenar las distintas versiones de un elemento, las herramientas de gestión de configuración suelen emplear un **almacenamiento incremental** (delta) que consiste en guardar sola la primera (o la última) versión del elemento de forma completa, almacenando, para el resto de las versiones, exclusivamente las diferencias entre éstas y la versión inmediatamente anterior (o posterior)

Aunque muchos entornos de desarrollo tienen sus propias facilidades de gestión de configuración, pueden considerarse como herramientas específicas AIDE-DE- CAMP de Software Maintenance & Development Systems, CCC de Softtool Corporation o CMVC de IBM.

11.8.- Herramientas de Ingeniería inversa

En este apartado destacan distintas herramientas, como las que llevan a cabo:

- **Ingeniería inversa de datos.**- Capaces de extraer información del código fuente que describe la estructura de los elementos de datos, construyendo diagramas E/R partiendo de esquemas relacionales, jerárquicos o ficheros
- **Ingeniería inversa de procesos.**- Permiten aislar la descripción lógica de las entidades y las reglas de negocio a partir del código de los programas

- **Reestructuración del código fuente.**- Modifican el formato del código fuente o implantan un formato estándar. Su funcionamiento puede resumirse de la siguiente manera:
 - Se construye un flujograma a partir del programa no estructurado
 - Se estructura el flujograma (por ejemplo eliminando GOTO)
 - Se genera un nuevo programa
- **Redocumentación.**- Permiten generar diagramas a fin de que se comprenda mejor el código
- **Análisis de código.**- Funcionalidades que van desde la indentación automática del código fuente hasta la posibilidad de ir visualizando dinámicamente las llamadas existentes en el mismo.

Las herramientas de ingeniería inversa se han desarrollado fundamentalmente para lenguaje COBOL aunque existen también para FORTRAN y SGBD jerárquicos y relacionales. Algunos ejemplos son: BACHMAN TOOL SET de Bachman Information Systems, COBOL/SRE de Andersen Consulting, etc.

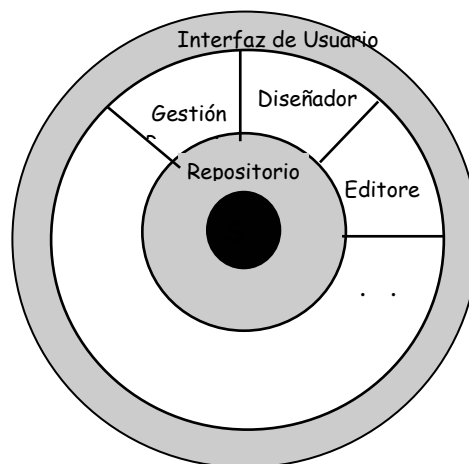
11.9.- Integración de herramientas CASE

Como se ha observado, no suele ser habitual que un mismo fabricante ofrezca las mejores herramientas para las distintas tareas a realizar durante todo el ciclo de vida, por lo que se hace preciso integrar diversos productos.

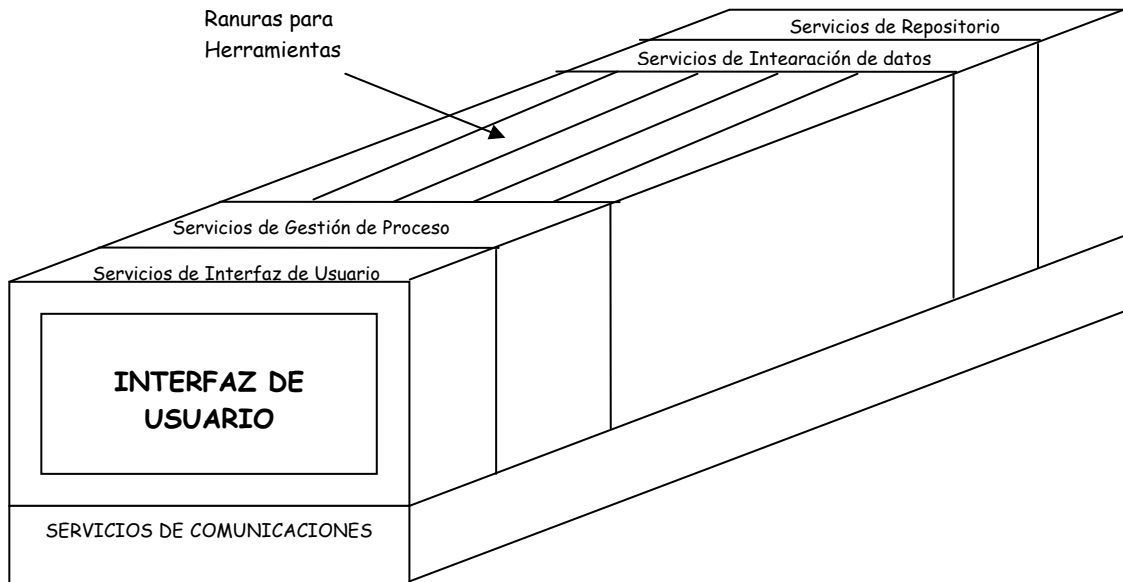
Tradicionalmente, se distinguen tres clases de integración:

- **Integración de datos.**- Posibilidad de que una herramienta utilice o entregue información a otras herramientas, bien sea mediante un mecanismo de transferencia de datos, bien a través del repositorio
- **Integración de control.**- Mediante la cual una herramienta puede invocar a otras y comunicarles eventos
- **Integración de presentación.**- Que se refiere a la homogeneidad y consistencia de la interfaz de usuario (por ejemplo, imponiendo que todas las herramientas accedan a la ayuda de la misma manera).

Como **modelos de integración** suelen considerarse los siguientes:



- * El **Modelo cebolla** cuyo núcleo es el sistema operativo, sobre el que se basa el repositorio que aglutina las diferentes herramientas del entorno, sobre las que se sitúa la interfaz de usuario, que proporciona una interacción homogénea entre el usuario y el sistema.



- * El **Modelo Tostadora** en el que las herramientas se encajan en el entorno como si fuesen las tostadas de una tostadora.

11.10.- Resumen

Las herramientas CASE constituyen un elemento muy importante en la mejora del software, pero no son suficientes si no van acompañadas de las técnicas y las metodologías adecuadas, no se implantan de la forma correcta o si no se completan con otros aspectos relativos a la calidad. En la actualidad, el entorno CASE se ha integrado con otros aspectos que también persiguen la mejora del software, como son las métricas, el modelo de proceso del software, modelos para evaluar la capacidad de desarrollo y calidad o la gestión de la configuración.