

¿Qué es una herramienta CASE?

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

También se puede definir 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.

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.

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

Componentes y funcionalidades de una Herramienta CASE

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. HERRAMIENTAS CASE
- 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.
- Interface con otras herramientas: procesadores de textos, editores gráficos, etc.

Estructura general de una herramienta case

La estructura CASE se basa en la siguiente terminología:

- CASE de alto nivel son aquellas herramientas que automatizan o apoyan las fases finales o superiores del ciclo de vida del desarrollo de sistemas como la planificación de sistemas, el análisis de sistemas y el diseño de sistemas.
- CASE de bajo nivel son aquellas herramientas que automatizan o apoyan las fases finales o inferiores del ciclo de vida como el diseño detallado de sistemas, la implantación de sistemas y el soporte de sistemas.
- CASE cruzado de ciclo de vida se aplica a aquellas herramientas que apoyan actividades que tienen lugar a lo largo de todo el ciclo de vida, se incluyen actividades como la gestión de proyectos y la estimación.

Herramientas/entornos CASE (y meta-CASE)

- **Rational Rose:** <http://www.rational.com/>

- **Dome (herramienta CASE/MetaCASE):**
<http://www.htc.honeywell.com/dome/index.htm>
- **MetaEdit (herramienta CASE/MetaCASE):** <http://www.metacase.com/>
- **ArgoUML (herramienta CASE - open source):** <http://argouml.tigris.org/>

Herramienta UML (Lenguaje Unificado de Modelado)

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.