
9. Ingeniería de Software Asistida por Computadora (CASE)

Todo el mundo conoce la historia de los hijos del zapatero: el zapatero esta tan ocupado haciendo zapatos para otros que sus hijos van descalzos. Durante los últimos 20 años, muchos de los ingenieros de software han sido los “hijos del zapatero”. Aunque han construido sistemas complejos que automatizan el trabajo de otros, ellos mismos no han aplicado estas técnicas. De hecho hasta hace poco, la ingeniería de software era fundamentalmente una actividad manual en la que las herramientas se utilizaban únicamente en las etapas finales.

Hoy en día, los ingenieros de software han recibido su primer par de zapatos – la ingeniería de software asistida por computadora (sus siglas en ingles CASE). No hay tanta variedad de zapatos como nos gustaría. Sin embargo, constituyen una pieza indispensable del guardarropa del ingeniero y, con el tiempo, se harán mas confortables, mas fáciles de usar y mas adaptables a las necesidades de cada usuario.

Orígenes del CASE

En 1955, los ingenieros mecánicos y eléctricos trabajan con herramientas manuales: libros y tablas que contenían formulas y los algoritmos necesarios para el análisis de un problema; calculadoras (mecánicas) para realizar los cálculos necesarios y asegurar que el producto iba a funcionar; bolígrafos y lápices, mesas de dibujo y reglas que permita al ingeniero crear los modelos del producto que iba a construir. Se hizo un buen trabajo, pero se hizo a mano.

Paso una década y el mismo grupo de ingeniería comenzó a experimentar con la ingeniería basada en computadora. Muchos se resistieron a utilizar computadoras. Una excusa habitual era: “no me fió de los resultados”. Sin embargo, otros se lanzaron hacia adelante. El proceso estaba cambiando.

Pasamos a 1975. Las formulas y los algoritmos que el ingeniero necesitaba se incorporaron a programas de computadora que se utilizaban para analizar una gran variedad de problemas de ingeniería. La gente confiaba en los resultados de estos programas. De hecho, la mayoría de su trabajo no podía realizarse sin ellos. Las estación es de trabajo graficas, conectadas a potentes computadoras, estuvieron en uso y sustituyeron a las mesas de dibujo y otras herramientas para la creación de modelos de ingeniería. Se estaba construyendo un puente entre la ingeniería y el trabajo de manufactura, creando el primer enlace el diseño asistido por computadora (CAD) y la fabricación asistida por computadora (CAM.)

Volviendo al futuro, encontramos ingeniería asistida por computadora (CAE), diseño asistido por computadora y fabricación integrada por computadora

(CIM, sucesor de CAM) como actividades usuales en la mayoría de las empresas.

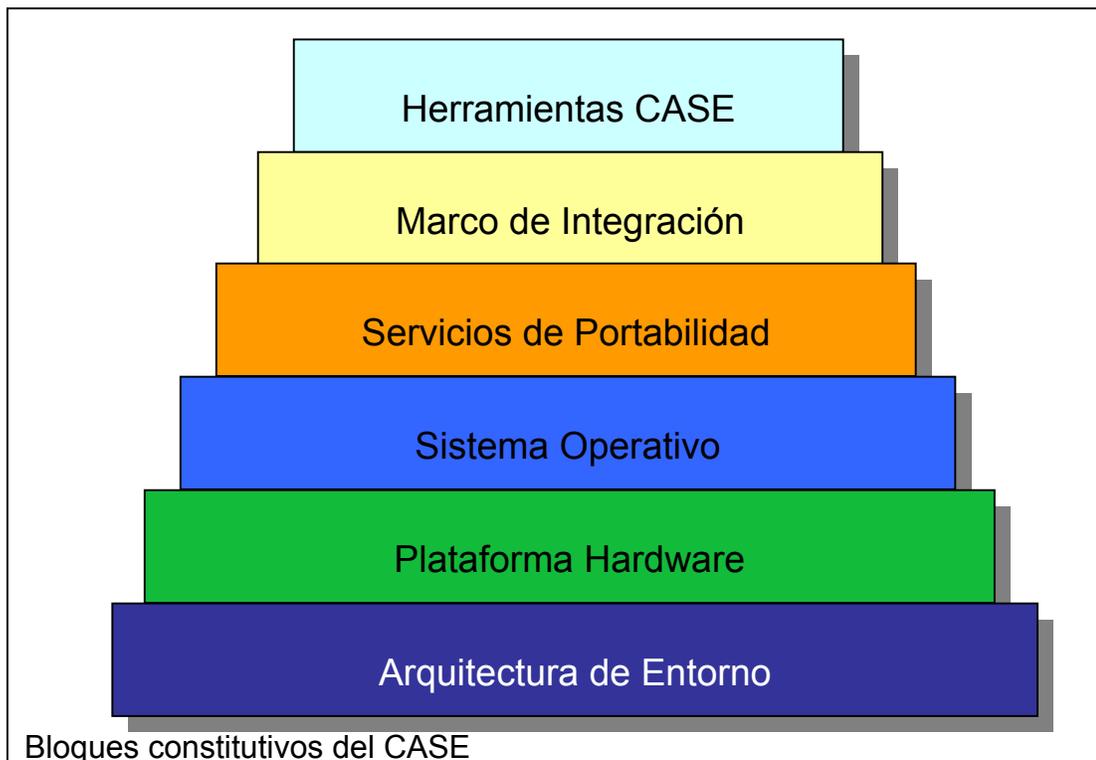
¿Qué es el CASE?

Las herramientas CASE son usadas en algunas de las fases de desarrollo de sistemas de información, incluyendo análisis, diseño y programación. Su objetivo fundamental es proveer un lenguaje para describir el sistema general que sea lo suficientemente explícito para generar todos los programas necesarios.

La CASE supone la aplicación de principios científicos a través de una metodología que ayude a producir software de alta calidad en un tiempo mucho más reducido.

Bloques que Componen el CASE

La Ingeniería del Software Asistida por Computadora (CASE) puede ser tan simple como una herramienta que permite desarrollar una actividad específica, o tan compleja como un "entorno" que integre distintas herramientas, bases de datos, hardware, red, sistemas operativos, estándares y muchos otros componentes.



Cada bloque constituye la base del siguiente, con las herramientas situadas en la cima de la pila. Es interesante ver que el fundamento para un CASE efectivo

tiene poco que ver con las herramientas de ingeniería del software en sí mismas.

Herramientas CASE: se describe con más profundidad en el capítulo 9.1

Marco de integración: Es un conjunto de programas especializados que permiten a cada herramienta CASE comunicarse con las demás.

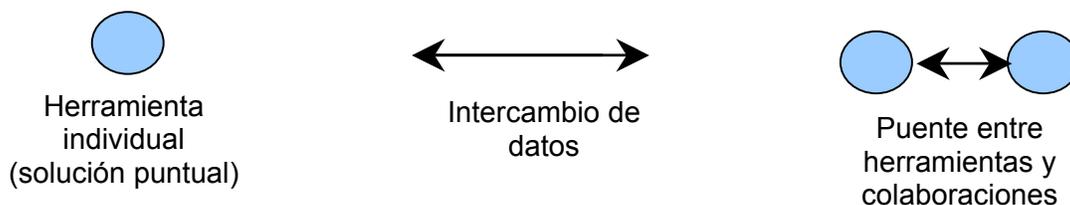
Servicios de portabilidad: Este conjunto constituye un puente entre las herramientas CASE, su marco de integración y la arquitectura de entorno. De esta forma permiten que las herramientas CASE y su marco de integración puedan migrar a través de diferentes plataformas de hardware y sistemas operativos sin problemas de adaptación.

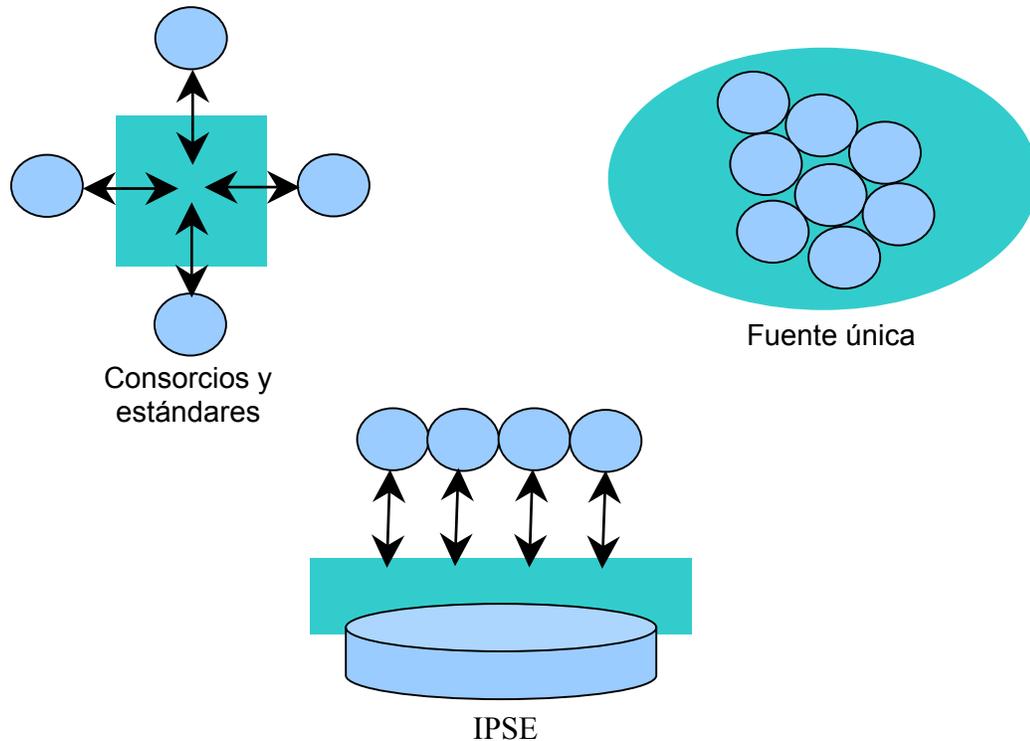
Sistema operativo: Gestiona el hardware, la red y las herramientas; mantiene el entorno unido.

Plataforma hardware: Son las estaciones de trabajo individuales interconectadas mediante la red para que los ingenieros del software puedan comunicarse de forma efectiva.

Arquitectura de entorno: Es la base del CASE, en este bloque se construyen los entornos de la ingeniería del software, engloba los sistemas de software y hardware. Además considera los patrones del trabajo humano que se aplican durante el proceso de ingeniería del software.

En las siguientes figuras se muestran los niveles de integración del CASE. En el nivel más bajo del espectro de integración está la **herramienta individual** (solución puntual). Cuando las herramientas proporcionan facilidades para el **intercambio de datos** el nivel de integración aumenta ligeramente. Estas herramientas generan una salida en un formato estándar compatible con otras herramientas que puedan leer ese formato. En algunos casos, los que construyen herramientas CASE complementarias trabajan juntos para establecer un puente entre ellas (ejemplo: una herramienta de diseño y análisis que se une a un generador de código). La **integración por fuente única** se da cuando el constructor de herramientas CASE integra diferentes herramientas y las vende como un único paquete. Al final del espectro de integración está el entorno de soporte de proyectos integrado (IPSE por sus siglas en inglés).





9.1 Herramientas CASE

Clasificación de las herramientas case

Siempre que se intenta clasificar las herramientas CASE se corren riesgos. Se suele suponer que para crear un entorno CASE efectivo, se deben incluir todas las categorías de herramientas – pero esto es sencillamente falso. Se puede dar lugar a una confusión al situar a una herramienta determinada dentro de una categoría cuando se podría pensar que pertenece a una categoría distinta. La categorización simple tiende a ser llana – esto es, no aparece una interacción jerárquica de las herramientas o de las relaciones entre ellas. Pero aun asumiendo todos estos riesgos, es necesario crear una taxonomía – para comprender mejor el alcance del CASE, y para apreciar mejor donde se pueden aplicar estas herramientas,

Las herramientas CASE se pueden clasificar bajo diferentes enfoques:

- ◆ Por su función
- ◆ Por su papel como instrumentos para el personal técnico o los directivos.
- ◆ Por la arquitectura del entorno que las soporta (hardware y software)
- ◆ Origen

Tomando la funcionalidad como criterio principal se creó la siguiente clasificación:

Herramientas de planificación de sistemas de gestión

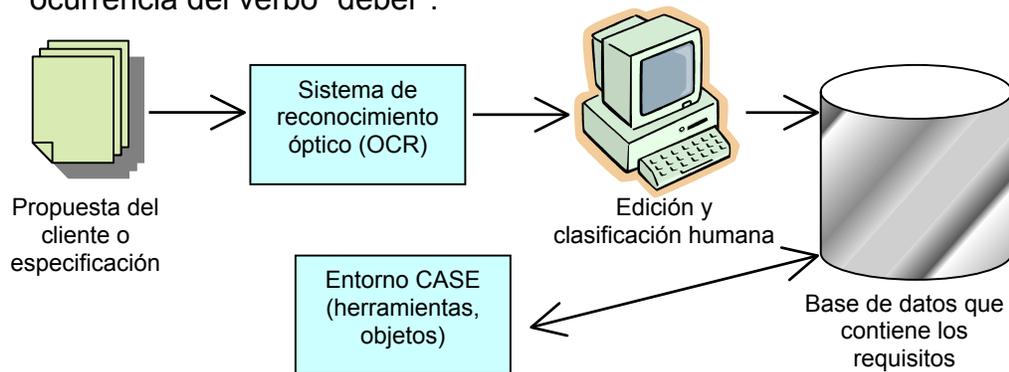
Proporcionan un "metamodelo" del cual se pueden obtener sistemas de información específicos, mediante la modelización de los requisitos de información estratégica de una organización. El objetivo principal de las herramientas de esta categoría es ayudar a comprender mejor como se mueve la información.

Herramientas de gestión de proyectos

Pueden hacer estimaciones útiles de esfuerzo, coste y duración del proyecto, definir una estructura de partición del trabajo, planificación del mismo y hacer el seguimiento de proyectos de forma continua. Además se pueden utilizar para recoger datos que permitan realizar una estimación de la productividad del desarrollo y la calidad del producto.

Herramientas de planificación de proyectos: las herramientas que caen dentro de esta categoría se centran en dos áreas fundamentales: el esfuerzo y coste de un proyecto de software; y la planificación del proyecto.

Herramientas de seguimiento de requisitos: el objetivo de estas herramientas es de proporcionar un enfoque sistemático para aislar requisitos, comenzando con las especificaciones del cliente. La extracción de requisitos puede ser tan sencilla como encontrar cada ocurrencia del verbo "deber".



Herramientas de gestión y medida: las herramientas de medidas actuales se centran a las características del producto y del proceso. Las herramientas orientadas a la gestión parten de medidas

específicas del proyecto que proporcionan una indicación global de la productividad y de la calidad.

Herramientas de soporte

La categoría de herramientas de soporte engloba las herramientas de aplicación y de sistemas que complementan el proceso de ingeniería de software. Estas incluyen herramientas de documentación, herramientas para gestión de redes y software del sistema, herramientas de control de calidad y herramientas de gestión de bases de datos y de configuración del software.

Herramientas de documentación: Las herramientas de producción de documentación y autoedición se utilizan en casi todos los aspectos de la ingeniería del software y representan una oportunidad muy interesante para todos los que desarrollan software. No es raro que una empresa emplee el 20 o el 30 por ciento de su esfuerzo de desarrollo en la documentación. Por esta razón, estas herramientas constituyen una opción importante para aumentar la productividad.

Las herramientas de documentación suelen estar unidas a otras herramientas CASE por medio de una interfaz de datos suministrada por el vendedor. Muchas herramientas de análisis y diseño están unidas a uno o varios sistemas de autoedición, de tal forma que los modelos y textos creados durante el análisis y el diseño puedan ser transmitidos a una herramienta de documentación y añadidos a la especificación creada utilizando la misma herramienta de documentación.

Herramientas para software de sistemas: El CASE es una tecnología de estaciones de trabajo. Por esto, el entorno CASE debe soportar software de redes de comunicación de alta calidad, correo electrónico, boletines electrónicos y otras posibilidades de comunicación.

Herramientas de control de calidad: La mayoría de las herramientas CASE que se venden como orientadas al control de calidad, son en realidad herramientas de medida que comprueban el código fuente para determinar su compatibilidad con lenguajes estándar. Otras herramientas extraen métricas técnicas como base para medir la calidad del software que se está desarrollando.

Herramientas de bases de datos y de GCS: El software de gestión de bases de datos sirve como base para el establecimiento de una base de datos CASE (almacén). Poniendo énfasis en los objetos de la configuración, las herramientas de gestión de bases de datos para

CASE pueden evolucionar de los sistemas relacionales a los sistemas basados en objetos.

Las herramientas CASE pueden ayudar en las cinco tareas principales de la configuración del software, identificación y control de versiones, control de cambios, auditoria y gestión de estados. Las base de datos CASE proporciona un mecanismo para identificar cada elemento de la configuración y relacionarlo con otros elementos.

“La utilización de bases de datos, herramientas de gestión y configuración y herramientas de inspección de componentes, es el primer paso hacia una biblioteca de software que estimulará la reutilización de componentes de software.”

Herramientas de análisis y diseño

Las herramientas de análisis y diseño permiten al ingeniero de software crear un modelo del sistema que se va a construir. El modelo contiene una representación de los datos y del flujo de control, del contenido de los datos, representaciones de los procesos, especificaciones de control y otras representaciones del modelo. Las herramientas de análisis y diseño también permiten la evaluación de la calidad del modelo y ayudan a eliminar errores antes de que se propaguen al diseño, o al código.

Herramientas de AE/DE: la mayoría de las herramientas de diseño y análisis se basan en el método de análisis y diseño estructurado (AE/DE). El AE/DE es una técnica que permite al ingeniero de software crear progresivamente modelos más complejos de un sistema, comenzando en el nivel de requisitos y concluyendo con un diseño de arquitectura.

Herramientas PRO/SIM: las herramientas de creación de prototipos y de simulación (PRO/SIM) proporcionan al ingeniero de software la capacidad de predecir el comportamiento de un sistema en tiempo real antes de que sea construido. Muchas herramientas tienen la capacidad de generar código.

Herramientas para el diseño y desarrollo de interfaces: las herramientas de diseño y desarrollo de interfaces son, en realidad un conjunto de componentes de software, tales como menús, botones, estructuras de ventanas iconos, mecanismos de visualización, controladores de dispositivos y otros elementos de este tipo.

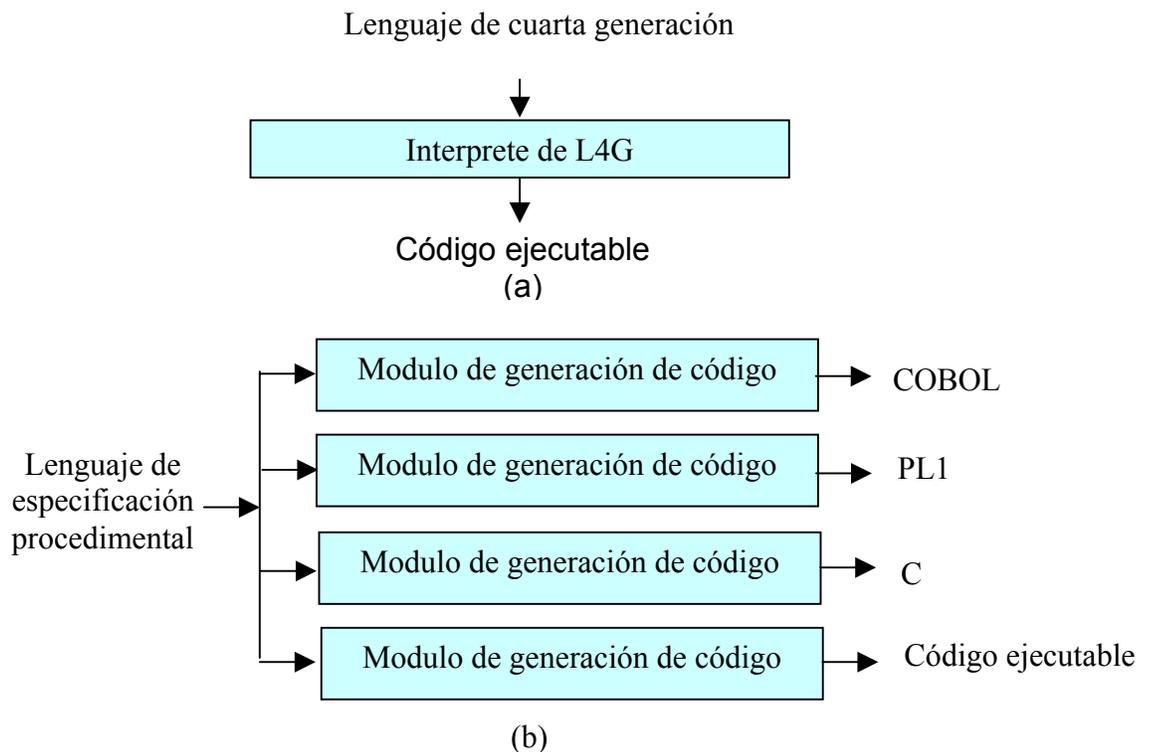
Herramientas de programación

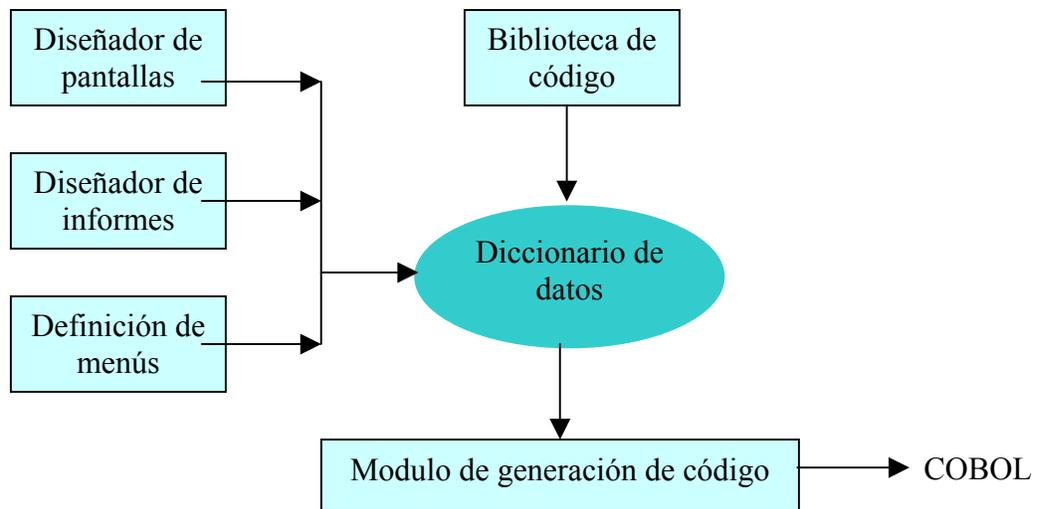
Engloba los compiladores, los editores y los depuradores que se utilizan con los lenguajes de programación convencionales.

Herramientas de codificación convencionales: durante casi 30 años las únicas herramientas disponibles para los programadores eran las herramientas convencionales de programación y por esto, cada problema de ingeniería de software era como un problema de programación. Hoy en día las herramientas convencionales siguen existiendo como una primera línea del desarrollo de software, pero están respaldadas por todas las herramientas CASE.

Herramientas de codificación de cuarta generación: los sistemas de consulta a bases de datos, los generadores de código y los lenguajes de cuarta generación han cambiado la forma de desarrollar sistemas. Idealmente, estas herramientas de generación de código no solo traducirían la descripción de un sistema a un programa operativo sino que también ayudarían a verificar la corrección de la especificación del sistema. De tal forma que la salida resultante satisfaga los requisitos del usuario.

Herramientas de cuarta generación. (a) L4G; (b) generación de código; (c) generación de aplicaciones.



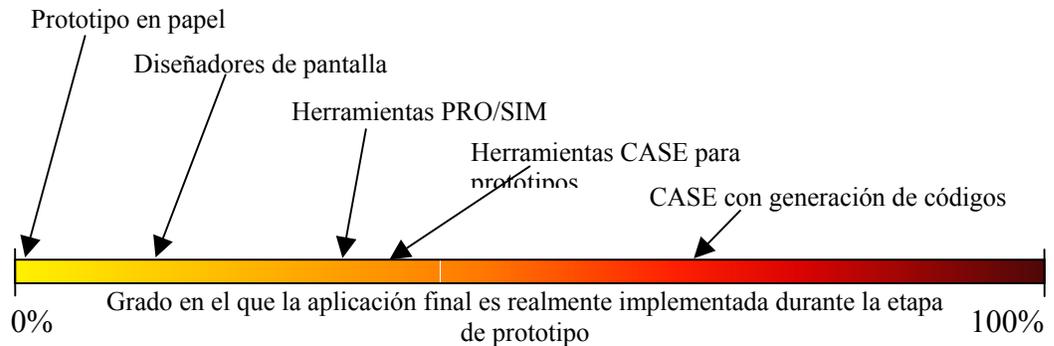


(c)

Herramientas de programación orientadas a objetos: es una de las tecnologías más actuales de la ingeniería de software. Los entornos de programación orientados a objetos suelen estar unidos a lenguajes de programación específicos como: C++, Eiffel, Objective-C, Smalltalk o Java. Un entorno O-O típico incorpora características de las interfaces de tercera generación (ventanas, ratón, menús desplegables, operaciones sensibles al contexto, multimedia, etc.) con funciones especializadas como la del “inspector” –una función que permite al ingeniero de software examinar todos los objetos contenidos en las bibliotecas de objetos para determinar si pueden o no se utilizadas en la aplicación actual.

Herramientas de creación de prototipos

La realización de prototipos es un paradigma de la ingeniería de software ampliamente utilizado, todas las herramientas de creación de prototipos se sitúan en algún lugar del espectro de implementación que se muestra en la figura.



Herramientas de ingeniería inversa: utiliza como entrada el programa fuente para extraer y analizar su arquitectura, su estructura de control, el flujo lógico y la estructura y flujo de datos. Otras herramientas que pertenecen a esta categoría aplican una técnica conocida como partición de programas. Las herramientas de ingeniería inversa han sido denominadas herramientas de “visualización de código”, permitiendo que el ingeniero visualice el programa, y a su vez ayudan a controlar la cantidad de cambios y la productividad de la gente que los realiza.

Herramientas de reingeniería: pueden dividirse en dos subcategorías – de **reestructuración de código**, que aceptan como entrada código fuente si estructurar y realizan el análisis de ingeniería inversa reestructurando el código y agostándolo a los conceptos modernos de programación estructurada; de **revisión de datos**, que analizan las definiciones de los datos o una base de datos descrita en un lenguaje de programación o en lenguaje de descripción de base de datos, traducen esta descripción a una notación gráfica que puede ser analizada por el ingeniero de software. Al trabajar con las herramientas de reingeniería, se puede modificar la estructura lógica de la base de datos, normalizar los archivos resultantes y generar automáticamente un nuevo diseño físico de la base de datos.

9.2 Integración de Herramientas

Todos los entornos CASE implementan **mecanismos de ejecución** y **mecanismos de comunicación**. Para ilustrar las características de estos mecanismos se seguirá el *Entorno de Herramientas Portables Comunes* (por su siglas en inglés, PCTE), uno de los diferentes estándares para entornos CASE.

Los mecanismos de ejecución y de comunicación se denominan **mecanismos básicos** – se definen funciones para manipular

entidades que existen dentro del contexto de desarrollo de software. Estas entidades incluyen objetos (datos, código fuente, documentos, dispositivos) y herramientas que operan sobre los objetos.

Los mecanismos de ejecución constituyen una forma de ejecutar un proceso desde su contexto estático independientemente que se trate de un programa ejecutable o interpretable; a su vez los mecanismos de comunicación gestionan la comunicación entre procesos, estableciendo colas de mensajes que permiten comunicarse a las diferentes herramientas, por ejemplo: la realización de una tarea a cargo de la herramienta CASE "A" puede generar un "suceso" que active la herramienta CASE "B", para ejecutar B se debe utilizar un mecanismo de arranque, pero para pasar información de A a B se requiere un mecanismo de comunicación.

9.3 Ejemplos de herramientas CASE

A continuación se muestra una lista de herramientas CASE, solo se muestran las que se consideraron mas comunes, algunos de nosotros ya hemos trabajado con una o mas de estas herramientas, pero ahora sabemos que son las herramientas CASE, y también sabemos lo útiles y la gran ayuda que podemos obtener de ellas.

Compañía	Observaciones
Eclipse	Eclipse es una plataforma abierta para la integración de herramientas de desarrollo construidas por una comunidad abierta y proveedores de herramientas. Trabaja y se desarrolla con los criterios de código abierto y licencia libre
ORACLE	Oracle es una de las principales compañías fabricantes de herramientas CASE. Entre sus productos mas utilizados se encuentran: JDeveloper, Designer y Forms Developer.
SYBASE - Powersoft's	PowerDesigner 7.0: herramienta CASE de Análisis y Diseño incluye capacidades de generación relacional y con orientación a objetos
VISIO - MICROSOFT	Descripción visual de procesos y ayuda a la reingeniería de procesos
Software Engineering Lab.	ASADAL - Herramienta CASE especializada en Sistemas de Tiempo Real
IMB Software	Rational ClearCase – Herramienta para el manejo de versiones.
Emutool y EmuDump	EmuTool y EmuDumpGenTool - Simulador de APZ para centrales telefónicas AXE utilizadas y desarrolladas por Ericsson

Bibliografía y Referencias

-
- ◆ Ingeniería del Software 6ª edición
Ian Sommerville & Addison Wesley
 - ◆ Apuntes de informática
<http://usuarios.lycos.es/klauzen/notas.htm>
 - ◆ Revista Espacios de Venezuela
<http://www.revistaespacios.com/a99v20n02/30992002.html>
 - ◆ Introducción a los Sistemas y Herramientas CASE - Computer Aided Software Engineering
<http://ceds.nauta.es/informes/case04.htm>:
 - ◆ IBM Software – Rational Clearcase - Product Overview
<http://www-306.ibm.com/software/awdtools/clearcase/>