

EINA

CASE - AMDC Object Oriented Development tools

RESUMEN

◆ Herramientas CASE

- Origen
- Evolución
- Tipos

◆ AMDC

◆ Arquitectura EINA

- Módulos integrantes
- Destinatarios
- Soporte al ciclo de vida

◆ MÓDULOS EINA

- Kernel
 - Plantillas Variables
 - Gestión de Cambio
 - Ingeniería de Requisitos
- Control de Versiones
- UML Lenguaje de Modelación
- OCL Lenguaje de Restricciones
- Generación de Código

◆ MÓDULOS EINA (Cont.)

- Reusabilidad
- Seguimiento de Errores
- Base de Conocimiento
- Modelación Relacional
- COCOMO Control de Costos
- Gestión de Equipos Humanos

◆ Otros Módulos EINA

- Generación de Documentos
- Meta-Modelación Gráfica
- Métodos Formales

◆ EINA Paso a Paso

◆ Lo que falta

EINA CASE - AMDC OO Tools, es un producto tecnológicamente avanzado, robusto y sofisticado de uso crítico para el desarrollo automático de sistemas informáticos... *¡todavía en construcción!*

EINA ha sido concebido y construido bajo la dirección del profesor **Allen Peralta**, Doctor en Informática por la Universitat Politècnica de Catalunya, Licenciado en Informática por la Universidad Autónoma de Barcelona, Bachelor of Science (Licenciado) en Computer Technology por el New York Institute of Technology y con estudios de postgrado del Master of Science en Investigación Operativa por el Polytechnic Institute of New York. Es Premio Nacional de Informática CCS 1975, coautor de 8 libros y autor de numerosas publicaciones científicas.

Gran parte del desarrollo de Eina lo han realizado universitarios de informática, que aceptaron como trabajo de fin de curso algunos de los módulos componentes del proyecto y los desarrollaron bajo la supervisión del equipo de ingenieros responsables de Eina, quienes posteriormente integraron estos módulos en el producto final.

En la actualidad, la primera versión del producto está realizada en un 88,6% y su desarrollo se ha detenido, por lo que los socios propietarios de la empresa buscan fórmulas viables para concluirlo.

EINA

CASE - AMDC Object Oriented Development tools

Herramientas CASE

Computer-Aided Software Engineering, (**CASE** Tools o Herramientas CASE) Ingeniería de Software asistida por ordenador, es un conjunto de herramientas, lenguajes de programación y medios técnicos en general que permiten el desarrollo de aplicaciones de forma automática.

Las Herramientas CASE reducen costos, aceleran el desarrollo y proporcionan una extensa documentación que se puede utilizar para fu-

Las primeras Herramientas CASE aparecieron con gran expectativa en los años '60.

Hace 40 años los lenguajes de desarrollo de sistemas informáticos eran nulas o muy escasas. Los primeros resultados obtenidos frustraron las expectativas.

Los Modernos lenguaje, **no-procedimentales**,

- Lenguajes **declarativos**, orientados a la creación de un enunciado preciso del problema
- Lenguajes **funcionales**, a partir de funciones elementales se construyen unas más complejas.
- Lenguajes de Programación **Orientados al Objeto** (Object-oriented programming languages - OOPL)

Estos lenguajes permiten que los desarrolladores se centren en el problema a desarrollar y acometan proyectos complejos y seguros.

Tipos de Herramientas CASE:

Los tipos de herramientas CASE se clasifican por los productos que se obtiene con su utilización. Ellos son:

♦ **Upper CASE:** Facilitan la elaboración de la Gestión del Proyecto y su Análisis.

- **DFD** (Data Flow Diagrams),
- **ERD** (Entity Relationship Diagrams),
- Flujo de Trabajo,
- Orientación a Objetos,
- Diccionario de Datos,
- Reglas de Negocio,
- Herramientas de Análisis

♦ **Lower CASE:** Cercanas al código de Máquina

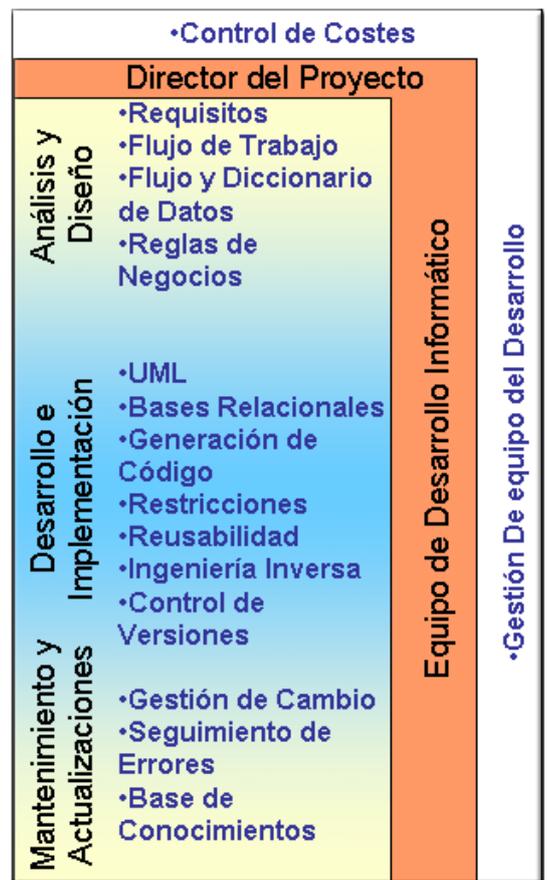
- Generación de Código,
- Generación de Formularios,
- Generación de Informes,
- Generación de Documentos, e
- Importación/Exportación de Utilidades

Desde muy temprano, las herramientas de bajo nivel han tenido buenas ofertas.

EINA es una herramienta CASE de nivel medio / alto

AMDC

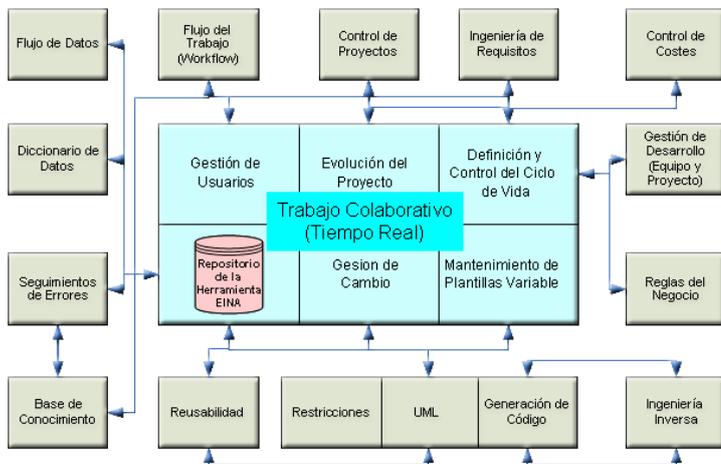
La tecnología **AMDC** (**A**nalysis, **M**odeling, **D**esign, and **C**onstruction tools) (Herramientas para el Análisis, Modelación, Diseño y Construcción), es la evolución de la tecnología CASE. Da soporte a metodologías formalizadas que ayudan a generar los requisitos de una aplicación, a definir los datos y las especificaciones de programación (tanto a las orientadas al objeto como a las no orientadas al objeto).



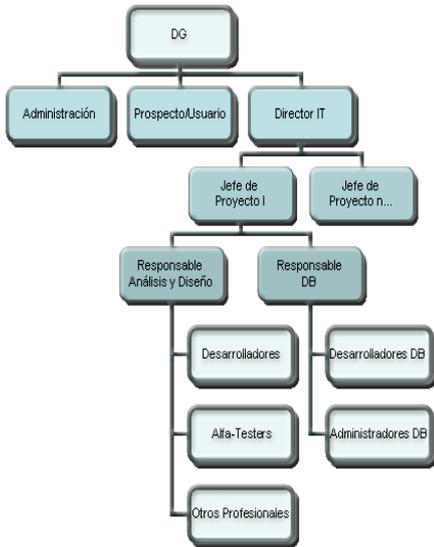
Las herramientas **AMDC** orientadas al Objeto suministran soporte a una o más metodologías orientadas a l objeto y a algunas o **todas las fases del ciclo de vida** de un desarrollo software, incluyendo la habilidad, si está completamente integrada en la metodología, de construir aplicaciones de dominios y/o componen-

EINA

CASE - AMDC Object Oriented Development tools



Arquitectura Eina



Destinatarios

EINA es una herramienta para ser utilizada por **todos** los miembros del equipo de un proyecto informático, sean del nivel que sean, a fin de realizar todas las tareas necesarias para construir programas para sus usuarios finales:

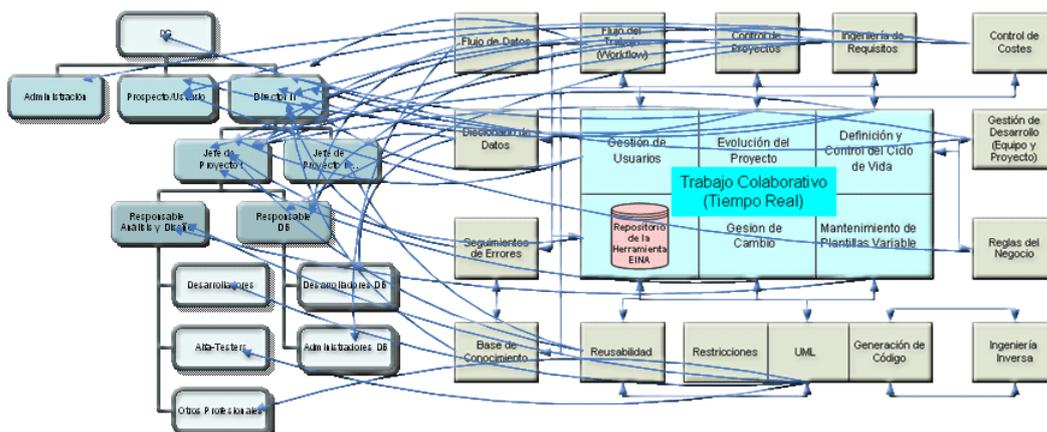
- **El director técnico** para definir la metodología de trabajo de sus técnicos.
- **El jefe de proyecto** para controlar el proyecto, la productividad y los costes.
- **El analista** para analizar las necesidades funcionales del usuario y proponer una solución.
- **El programador** para realizar las tareas de desarrollo de software.
- **y/o todas aquellas personas** que son **afectadas por el proyecto** asumido por el desarrollador, activando los módulos necesarios.

Arquitectura EINA

EINA es un producto de tecnología **AMDC** integrada, basado en la **Orientación a Objetos**, que está desarrollado en módulos independientes con funcionalidades específicas e íntimamente integrados

EINA está construida con los siguientes módulos:

- Kernel
- Requirements Engineering
- Graphical Version Control
- UML (Unified Modeling Language)
- OCL (Object Constraint Language)
- Language Support
- Reusability Management
- Error and Change Tracking
- Software Knowledge Management
- Database modeled with MDA
- COCOMO II Cost Control
- Team Management
 - Document Generation
 - Graphical Meta-Modeling
 - Formal Methods



Ciclo de Vida

EINA

CASE - AMDC Object Oriented Development tools

MÓDULOS EINA

KERNEL

Trabajo Colaborativo

- Soporta un ciclo de vida completo, y en un entorno tan competitivo como el actual.
- La herramienta ha sido diseñada para ser utilizada por todos los informáticos, sea cual sea su nivel o especialidad, de forma cooperativa, dando soporte al trabajo en equipo tanto local como globalmente, en equipos distribuidos en diferentes geografías



Desarrollado al 100%
Integrado al 100%

MÓDULO CENTRAL

Esencial del sistema. Gestiona:

- ◆ el repositorio o base de datos de la herramienta CASE,
- ◆ el mantenimiento de los usuarios del sistema y sus privilegios,
- ◆ la definición de ciclos de vida y su asociación con cada proyecto,
- ◆ la evolución de los proyectos,
- ◆ el mantenimiento de las plantillas variables,
- ◆ la gestión del cambio.
- ◆ Trabajo colaborativo en tiempo real



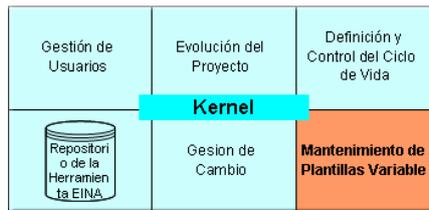
Desarrollado al 100%
Integrado al 100%

MÓDULOS EINA (Cont.)

KERNEL

Mantenimiento de plantillas variables

- ◆ Es la posibilidad de extensión documental que posee Eina siguiendo los perfiles o mecanismos de extensión de UML.
- ◆ Estas plantillas, a los usuarios interesados en otros tipos de informática adicional, les permiten añadir entre otras cosas;
 - un módulo características de "error tracking",
 - centro de coste,
 - fecha de petición de corrección,
 - coste, etc..



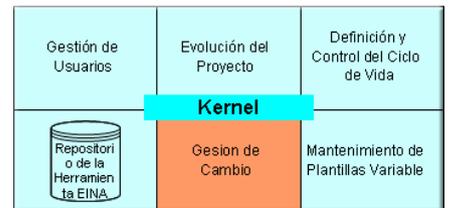
Desarrollado al 100%
Integrado al 100%

MÓDULOS EINA (Cont.)

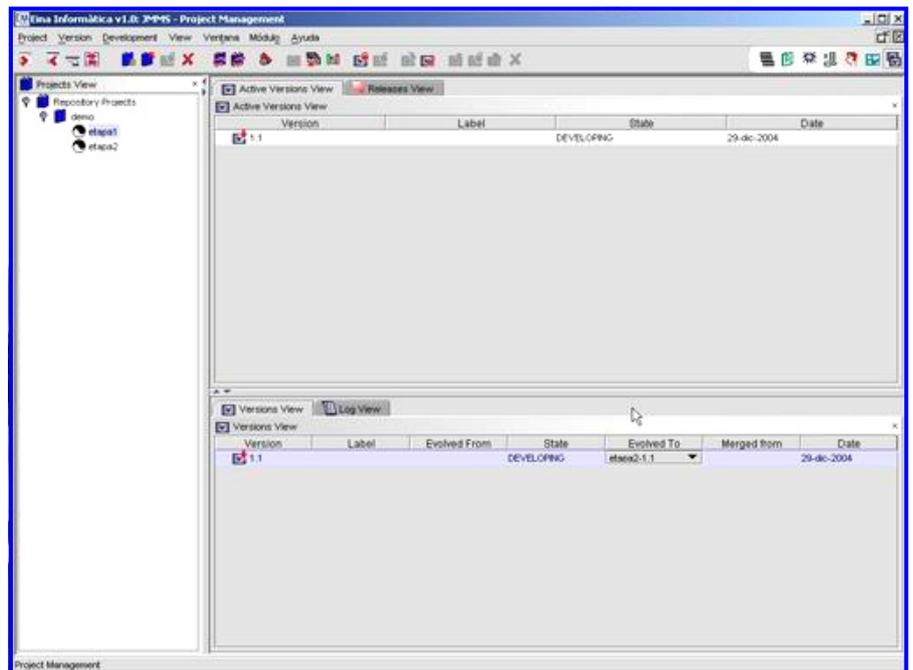
KERNEL

Gestión del cambio

- ◆ Es la funcionalidad que permite controlar la evolución de desarrollo de un sistema. Ejemplo: cuando un diagrama de análisis evoluciona a un diagrama de diseño, aunque son dos elementos independientes, la gestión de cambio registra esta evolución.
- ◆ De la misma forma, cuando se relaciona un requisito de un contrato con su implementación en un caso de uso, se realiza un tipo de control de seguimiento llamado "trazabilidad".



Desarrollado al 100%
Integrado al 100%

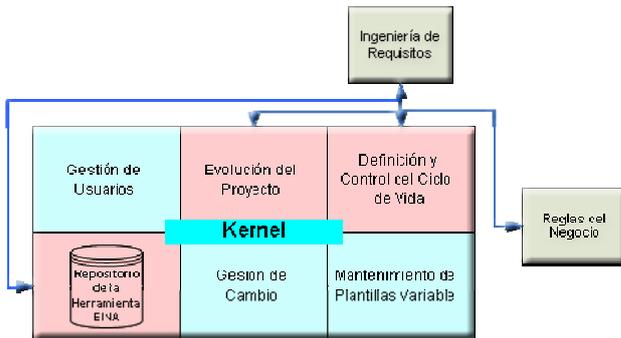


Apariencia Real

EINA

CASE - AMDC Object Oriented Development tools

MÓDULOS EINA

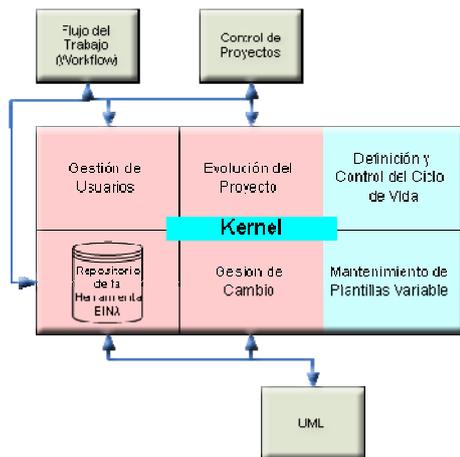


Ingeniería de requisitos (Requirements Engineering)

- ◆ Es el registro y control de los requisitos de los usuarios, que normalmente se reflejan en un contrato.
- ◆ El módulo de ingeniería de requisitos permite **registrar** todos los requisitos de un contrato, establecer trazas con su implementación en UML y planificar la tarea para su implementación.
- ◆ El módulo de **control** de la Ingeniería de requisitos garantiza que cada requisito se realiza tal como se especifica en el contrato. (El coste de un error en el proceso de gestión de un requerimiento es entre 10 y 20 veces superior al de un error en análisis, diseño o programación.)

Desarrollado al 90%

Integrado al 100%



Control de Versiones (Graphical Version Control)

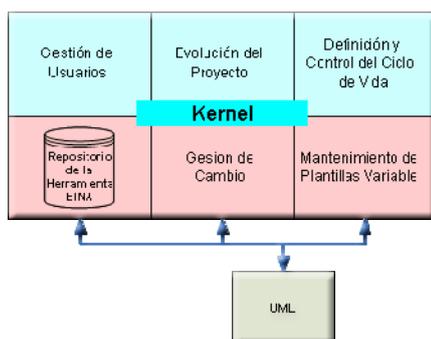
Eina incorpora un control de versiones o gestión del cambio tanto de la programación como de los diagramas o de cualquier otro artefacto.

Eina compara versiones para hallar sus diferencias. Estas comparaciones sobre datos textuales son de fáciles realización.

Mostrar las diferencias por diagramas requiere mostrar diferencias gráficas, textuales y ocultas en las plantillas de documentación.

Desarrollado al 100%

Integrado al 100%



UML (Unified Modeling Language)

Este módulo permite la diagramación de todos los modelos de UML 2.0. Así mismo implementa la versión parcial actual del Diagram Interchange. Eina da soporte a la versión 2.0 con todos sus requisitos gráficos y/o de meta-modelo.

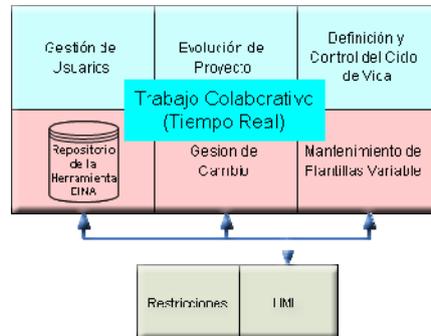
Desarrollado al 100%

Integrado al 100%

EINA

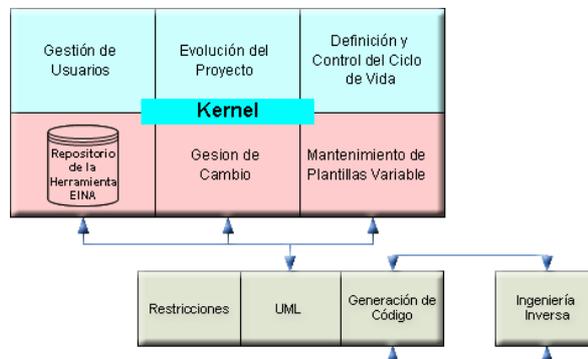
CASE - AMDC Object Oriented Development tools

MÓDULOS EINA



Generación de Código (Language Support)

- ◆ EINA está diseñada para dar soporte a todos los lenguajes comerciales orientados a objetos del tipo "round-trip engineering". Es decir, además de poder generar código, poder importar código modificando los diseños del repositorio o creando nuevas versiones. Este round-trip se obtiene sin lenguajes de marcación (mark-up languages) que entorpecen la lectura del programa y cuya modificación es siempre muy problemática.
- ◆ La parametrización de la generación de código se realiza mediante la utilización de plantillas variables (templates), lo que permite al usuario adaptar dicha generación a su gusto y preferencias.
- ◆ La versión actual soporta Java, Smalltalk, C++, C# y Cobol 200x.
- ◆ Durante el desarrollo de los modelos de diseño se puede especificar un "language environment" con lo que la herramienta se adapta al máximo a las particularidades de dicho lenguaje evitando en lo posible errores sintácticos.



OCL (Object Constraint Language)

- ◆ Forma parte del estándar de UML y al igual que este la versión actual es la 2.0. La utilidad de OCL fija restricciones o condiciones sobre los modelos de forma textual.
- ◆ Las restricciones se pueden poner sobre el meta-modelo y/o sobre el modelo.
- ◆ Mediante OCL se pueden definir estándares de empresa, de lenguaje, proyecto, etc. Que serán comprobados mediante un intérprete. El intérprete además de la devolución de datos booleanos también permite la devolución de datos numéricos, conjuntos, etc. por lo que puede usarse como query sobre los meta-modelos y/o los modelos.
- ◆ Otra ventaja que se obtiene del query OCL son métricas sobre las aplicaciones. Las métricas permiten obtener cifras de productividad y de calidad, al valorar la complejidad del sistema y su relación con el número de errores ocurridos a lo largo de la vida del programa.

Desarrollado al 100%
Integrado al 100%

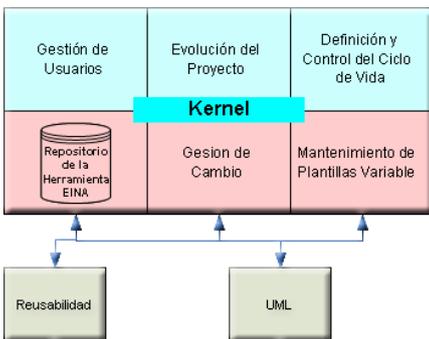
EINA

CASE - AMDC Object Oriented Development tools

MÓDULOS EINA

Gestión de Reusabilidad (Reusability Management)

- ◆ La gestión de la reusabilidad es esencial en un entorno orientado a objetos para incrementar la productividad y calidad de los desarrollos. Un activo o componente que se reutilice y que anteriormente haya sido certificado permite programar menos al reutilizar el componente existente y además como su corrección ha sido certificada aumenta la calidad de la aplicación al contener esta menos errores.
- ◆ La gestión de la Reusabilidad se inicia con un proceso de certificación que permite establecer las políticas del producto a desarrollar; le sigue un proceso de catalogación de los componentes con palabras controladas. El componente se puede ubicar físicamente en la biblioteca o almacenarlo en disco externo.
- ◆ La reusabilidad de EINA es amplia y además de catalogar componentes informáticos también puede catalogar plantillas de documentos, manuales de procesos, etc. Esto se realiza definiendo categorías de componentes con diferentes niveles de información.



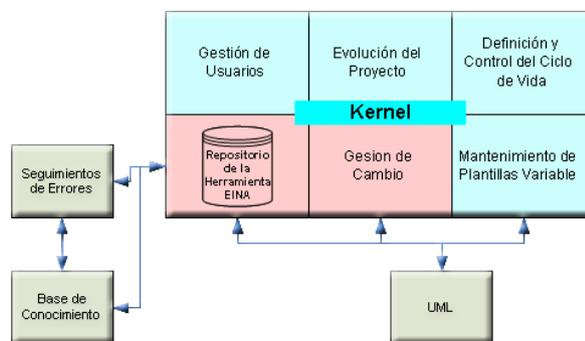
Desarrollado al 100%
Integrado al 75%

Seguimiento de Errores y Cambios (Error and Change Tracking)

- ◆ Cuando se identifica un error se registra en la base de datos y se inicia el proceso de corrección.

- ◆ Por cada error se registra:
 - el usuario,
 - una descripción del problema,
 - fecha en la que se ha detectado el error,
 - gravedad del mismo,
 - centro de coste al que repercutir la corrección y
 - técnico al que se le asigna la corrección.
- ◆ La corrección de un error sigue el siguiente protocolo:
 - El técnico realiza una evaluación del coste (y tiempo) necesario para corregir el error,
 - establece una traza entre el error en el programa y aquellos módulos que deban ser modificados y
 - se le asigna una tarea para su corrección según la importancia y severidad del error.
 - una vez corregido el error se registra la fecha de terminación (para poder tener cifras de tiempo de respuesta a los usuarios) y
 - se liberará el nuevo release al usuario.

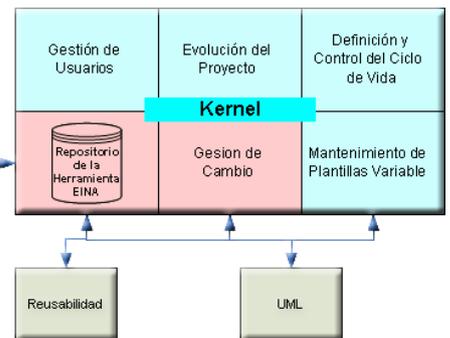
- ◆ El "change tracking" funciona de la misma forma pero además tiene distintos centros de coste asociados, prioridades inferiores, etc.



Desarrollado al 90%
Integrado al 100%

Base de Conocimiento (Software Knowledge Management)

- ◆ El mismo módulo que permite la catalogación de componentes informáticos facilita la catalogación de cualquier tipo de documento relacionado o no con un componente. Estos documentos pueden estar asociados a otros de forma que se crea una red de documentos, o correlaciones, que describen parte del conocimiento informático de la organización. Los documentos pueden ser internos o externos y si están en formato XML pueden hacerse catalogaciones mucho más eficaces



Desarrollado al 100%
Integrado al 100%

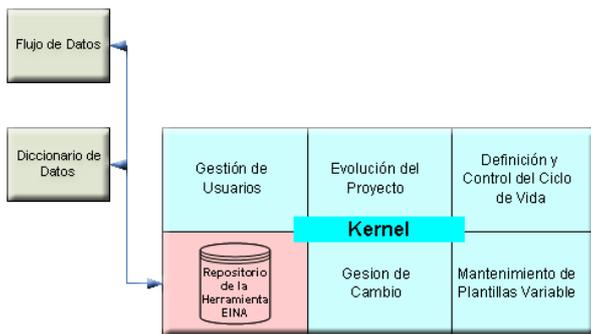
EINA

CASE - AMDC Object Oriented Development tools

MÓDULOS EINA

Modelado Relacional con MDA (Database modeled with MDA)

- ◆ Son muchas las herramientas AMDC de análisis y diseño que dan soporte al proceso de diseñar la base de datos y de construir unos objetos o componentes que permitan a la aplicación acceder a la base de datos.
- ◆ Algunas situaciones requieren un round-trip entre el SQL y el diseño de la base de datos que con frecuencia no soportan las herramientas AMDC. Como cuando surge la necesidad de introducir un cambio en una tabla (añadir uno o más datos) para obtener una información circunstancial. Se introduce el cambio y se realiza el



programa necesario. Pero el cambio no queda registrado. Si esto ocurre con cierta frecuencia no se puede obtener el nuevo diseño de la base de datos a partir del SQL textual.

- ◆ El módulo EINA para bases de datos permite tanto las funciones del desarrollador como las funciones que requiere el gestor de base de datos y que no están basadas en UML sino en el modelo relacional.

Desarrollado al 50%
Integrado al 50%

EINA informática soporta COCOMO II (COCOMO II Cost Control)

- ◆ En el control de costes, de un proyecto informático, intervienen muchos parámetros:
 - entorno de desarrollo, infraestructuras hardware y software,
 - experiencia del equipo de desarrolladores en dicho entorno,
 - experiencia y/o conocimiento del equipo de desarrolladores en la aplicación y en el tipo de aplicación,
 - estabilidad del software de base y del hardware, etc.

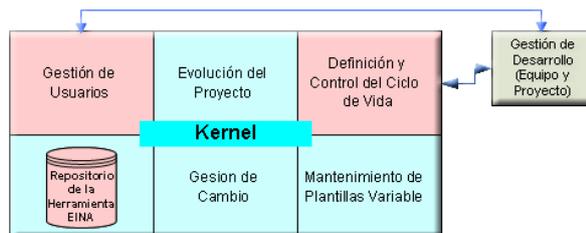
◆ Todo esto se necesita por cada informático que participa en el equipo de desarrollo, incluyendo a jefe de equipo.

◆ Otros aspectos, como el tamaño de la aplicación, el ciclo de vida, el tamaño del equipo, etc. influyen en

la estimación del coste y en el coste real obtenido.

- ◆ COCOMO II (Constructive COst Model) es el modelo desarrollado por Barry Boehm que mejor estima los costes de las aplicación

Desarrollado al 100%
Integrado al 0%



Gestión de Equipos Humanos (Team Management)

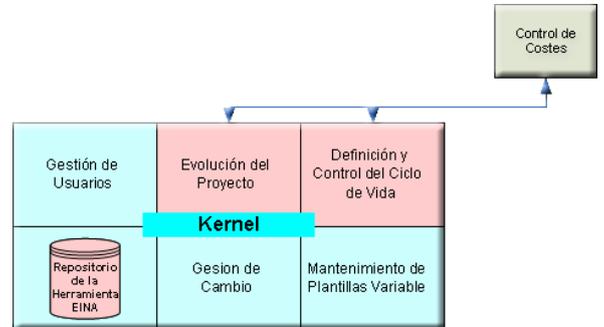
- ◆ El sistema de Team Management permite:

- planificar las tareas a desarrollar,
- cuantificar las cargas de trabajo asignadas a cada persona,
- detectar las desviaciones en tiempo y presupuesto, etc.

- ◆ Este módulo está íntimamente ligado a cada una de las etapas de desarrollo de un sistema (análisis, diseño, programación), lo que permite obtener métricas individuales o colectivas sobre el desarrollo y con ellas mejorar las futuras estimaciones y componer equipos humanos de desarrollo más ajustados a cada proyecto.

◆ Este módulo enlaza con los centros de coste.

Desarrollado al 90%
Integrado al 90%



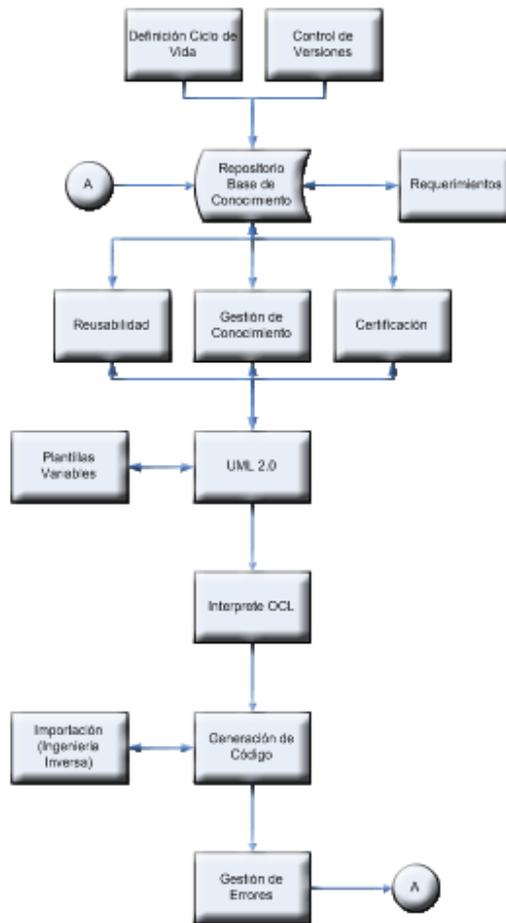
EINA

CASE - AMDC Object Oriented Development tools

OTROS MÓDULOS EINA

A diferente nivel de diseño y/o desarrollo se encuentran los siguientes módulos:

- ◆ **Document Generation** El módulo de documentación permite definir e integrar el tipo de documento a generar; Ayudas, Guías de Usuario, Manuales técnicos etc.
- ◆ **Graphical Meta-Modeling** Además de los estándar, existen muchos otros diagramas útiles que pueden complementar a UML 2.0. Unos ejemplos de estos diagramas son los jerárquicos para mostrar organigramas de personal de una organización, flujo documental en grandes organizaciones, etc. El módulo Graphical Meta-Modeling de EINA permitirá definir diagramas a nivel de meta-modelo. Es decir, dar las figuras, asociaciones y reglas de aplicación para crear un diagrama.
- ◆ **Formal Methods** Los métodos formales son una forma matemática de describir y/o desarrollar aplicaciones que garantizan que no existen errores o muy pocos. Se utilizan muy poco por que son muy complejos de aplicar y requieren una gran formación. EINA informática pretende ofrecer un método formal compatible con UML y que permita el desarrollo de sistemas críticos



EINA Paso a Paso

- ◆ **Gestión de Proyectos:** Que permite la Definición del Ciclo de Vida de un proyecto, su creación y actualización y el control de versiones.
- ◆ **Gestión de Requisitos:** Que permite la captura y gestión y actualización de los requisitos de los proyectos.
- ◆ **Modelación de Datos:** Que da soporte al diseño de bases de datos.
- ◆ **Modelación de Aplicaciones:** Permite los Análisis, diseños y desarrollos de aplicaciones con UML.
- ◆ **Plantillas Variables:** Módulo de soporte del resto de Módulos.
- ◆ **Reusabilidad:** Repositorio de todos los objetos y artefactos generados con Eina.
- ◆ **Generación de Código:** permite la ejecución en máquina de los sistemas modelados, incluyendo las respectivas restricciones.
- ◆ **Control de Restricciones:** Generación y gestión de las restricciones inherentes a un proyecto.
- ◆ **Gestión de Errores:** permite la identificación, registro y trazabilidad de errores
- ◆ **Generación de documentación:** soporte avanzado a la creación de informes

Lo que Falta de EINA

El producto, en el nivel de desarrollo que se encuentra (estimado en un 88,6%), funciona correctamente de acuerdo con lo proyectado, cumple el propósito funcional de sus creadores y ha superado con éxito el "Alfa-Test" y el "Beta-Test".

En la actualidad (tercer trimestre del año 2006), hace 18 meses que el desarrollo del producto está detenido. Para su comercialización definitiva es necesario:

- ◆ Concluir el código y actualizar la tecnología,
- ◆ Formalizar y traducir al inglés los manuales de; Uso, Programación, Guías y Tutoriales,
- ◆ Diseñar e implementar la política de venta y la estrategia de marketing,
- ◆ Lanzar el producto al mercado.

(Comente con nosotros los detalles - Contáctenos)

Solicite más información: