

RUP



METODOLOGÍA PURA

- Es una metodología cuyo fin es entregar un producto de software.
- Se estructura todos los procesos y se mide la eficiencia de la organización.
- El RUP es un conjunto de metodologías adaptables al contexto y necesidades de cada organización.
- Describe como aplicar enfoques para el desarrollo del software, llevando a cabo unos pasos para su realización.
- Se centra en la producción y mantenimiento de modelos del sistema

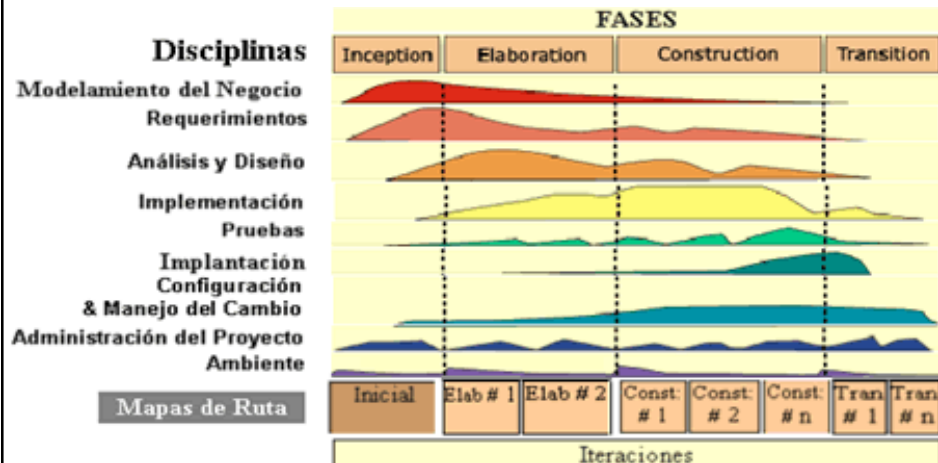
RUP



Es un proceso de desarrollo de software el cual utiliza el lenguaje unificado de modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

RUP



RUP - Principales características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

RUP



Esfuerzo en actividades según fase del proyecto

El ciclo de vida RUP es una implementación del Desarrollo en espiral.

Fue creado ensamblando los elementos en secuencias semi-ordenadas.

El ciclo de vida organiza las tareas en fases e iteraciones.

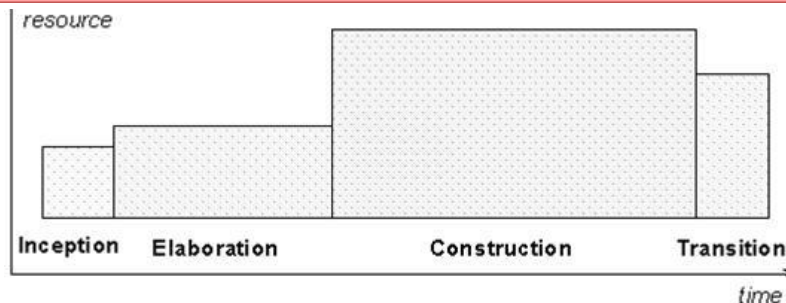
RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

RUP



Fases del ciclo de vida del RUP:

- 1. Fase de Inicio**
- 2. Fase de Elaboración**
- 3. Fase de Desarrollo**
- 4. Fase de Cierre / Transición**



RUP**1. Fase de Inicio**

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.

RUP**2. Fase de Elaboración**

En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

RUP**3. Fase de Desarrollo**

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

RUP**4. Fase de Cierre / Transición**

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

La metodología RUP tiene 6 principios clave:



- 1. Adaptación del proceso:** El proceso debe adaptarse a las características de la organización para la que se está desarrollando el software.
- 2. Balancear prioridades:** Debe encontrarse un balance que satisfaga a todos los inversores del proyecto.
- 3. Colaboración entre equipos:** Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, entre otros.
- 4. Demostrar valor iterativamente:** Los proyectos se entregan, aunque sea de una forma interna, en etapas iteradas. En cada iteración se evaluará la calidad y estabilidad del producto y analizará la opinión y sugerencias de los inversores.
- 5. Elevar el nivel de abstracción:** Motivar el uso de conceptos reutilizables.
- 6. Enfocarse en la calidad:** La calidad del producto debe verificarse en cada aspecto de la producción.

Disciplina de desarrollo de RUP



Determina las etapas a realizar durante el proyecto de creación del software.

Ingeniería o modelado del negocio: Analizar y entender las necesidades del negocio para el cual se está desarrollando el software.

Requisitos: Proveer una base para estimar los costos y tiempo de desarrollo del sistema.

Análisis y diseño: Trasladar los requisitos analizados anteriormente a un sistema automatizado y desarrollar una arquitectura para el sistema.

Implementación: Crear software que se ajuste a la arquitectura diseñada y que tenga el comportamiento deseado.

Pruebas: Asegurarse de que el comportamiento requerido es correcto y que todo lo solicitado está presente.

Despliegue: Producir distribuciones del producto y distribuirlo a los usuarios.

Disciplina de soporte RUP



Determina la documentación que es necesaria realizar durante el proyecto.

Configuración y administración del cambio: Guardar todas las versiones del proyecto.

Administración del proyecto: Administrar los horarios y recursos que se deben de emplear.

Ambiente: Administrar el ambiente de desarrollo del software.

Distribución: Hacer todo lo necesario para la salida del proyecto.

Elementos del RUP



Actividades: Procesos que se han de realizar en cada etapa/iteración.

Trabajadores: Personas involucradas en cada actividad del proyecto.

Artefactos: Herramientas empleadas para el desarrollo del proyecto. Puede ser un documento, un modelo, un elemento del modelo.

RUP - Artefactos



RUP en cada una de sus fases (pertenecientes a la estructura estática) realiza una serie de artefactos que sirven para comprender mejor tanto el análisis como el diseño del sistema (entre otros). Estos artefactos (entre otros) son los siguientes:

Inicio

- Documento Visión
- Diagramas de caso de uso
- Especificación de Requisitos
- Diagrama de Requisitos

Elaboración:

- Diagramas de caso de uso

Construcción:

- Documento Arquitectura que trabaja con las siguientes vistas:

RUP - Artefactos



Construcción:

- Documento Arquitectura que trabaja con las siguientes vistas:

VISTA LOGICA → Diagrama de clases
 → Modelo E-R (Si el sistema así lo requiere)

VISTA DE IMPLEMENTACION → Diagrama de Secuencia
 → Diagrama de Estados
 → Diagrama de Colaboración

VISTA CONCEPTUAL → Modelo de dominio

VISTA FISICA → Mapa de comportamiento a nivel de hardware.

El Proceso Unificado de Rational



- Un moderno modelo de proceso derivado de la labor sobre el UML y procesos asociados.
- Normalmente se describe a partir de 3 de perspectivas
 - Una perspectiva dinámica, que muestra las fases del modelo en el tiempo;
 - Una perspectiva estática que muestra las actividades de proceso;
 - Una perspectiva práctica que sugiere buenas prácticas a utilizar durante el proceso.

Modelado del software



Ejemplos

1- Construcción de una cucha para un perro:

Puede hacerlo una sola persona

Requiere:

- Modelado mínimo
- Proceso simple
- Herramientas simples

2- Construcción de una casa:

-Construida eficientemente y en un tiempo razonable de un equipo

- Modelado
- Proceso bien definido
- Herramientas más sofisticadas

3- Construcción de un rascacielos

Contexto de desarrollo

- Determinar configuración del proceso
- Recursos necesarios
- Herramientas más sofisticadas aún.

"El modelado captura las partes esenciales del sistema"



¿Qué es UML?

UML = Unified Modeling Language
Un lenguaje de propósito general para el modelado orientado a objetos
Documento "OMG Unified Modeling Language Specification"

UML combina notaciones provenientes desde:
Modelado Orientado a Objetos
Modelado de Datos
Modelado de Componentes
Modelado de Flujos de Trabajo (Workflows)

Motivación



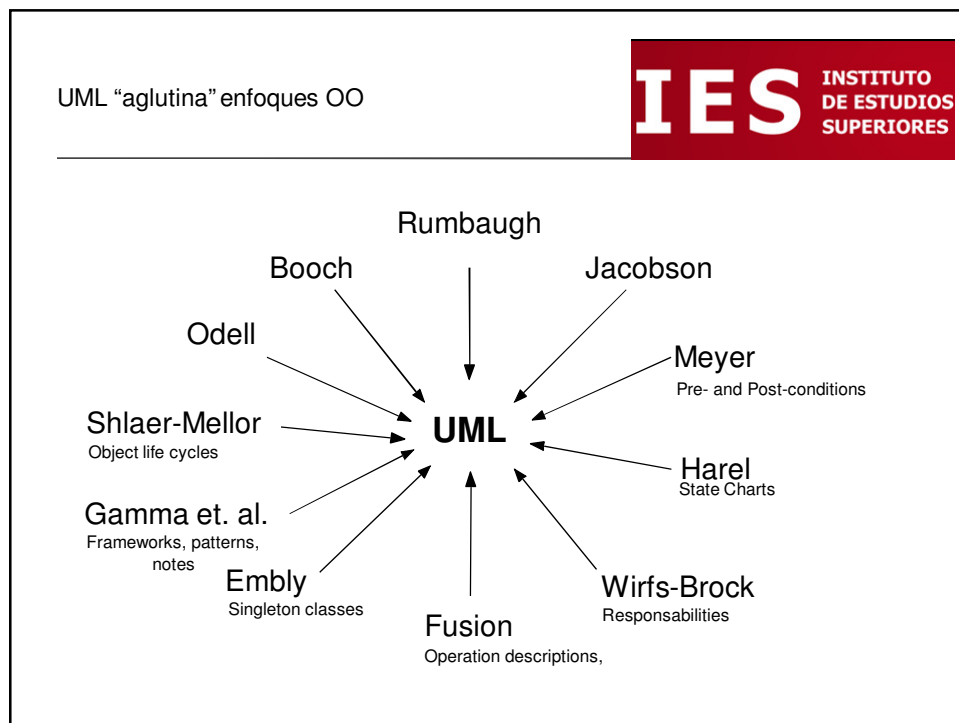
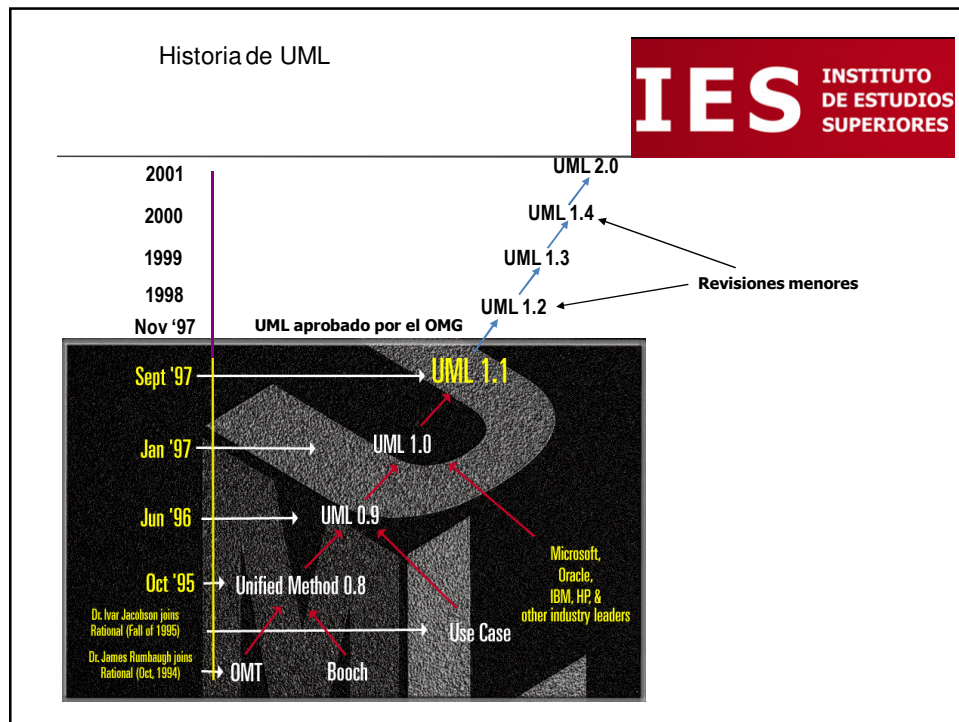
Diversos métodos y técnicas OO, con muchos aspectos en común pero utilizando distintas notaciones
Inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc.
Pugna entre distintos enfoques (y correspondientes gurús)
Establecer una notación estándar

Historia de UML



Comenzó como el “Método Unificado”, con la participación de Grady Booch y Jim Rumbaugh. Se presentó en el OOPSLA’95

El mismo año se unió Ivar Jacobson. Los “Tres Amigos” son socios en la compañía Rational Software. Herramienta CASE Rational Rose



Aspectos Novedosos



Definición semi-formal del Metamodelo de UML

Mecanismos de Extensión en UML:

Stereotypes

Constraints

Tagged Values

Permiten adaptar los elementos de modelado,
asignándoles una semántica particular

Inconvenientes en UML



Definición del proceso de desarrollo usando UML.

UML no es una metodología

Falta integración con respecto de otras técnicas
tales como patrones de diseño, interfaces de
usuario, documentación, etc.

“Monopolio de conceptos, técnicas y métodos en
torno a UML”

Perspectivas de UML



UML será el lenguaje de modelado orientado a objetos estándar predominante los próximos años

Razones:

Participación de metodólogos influyentes

Participación de importantes empresas

Aceptación del OMG como notación estándar

Evidencias:

Herramientas que proveen la notación UML

"Edición" de libros

Congresos, cursos, etc.

Modelos y Diagramas



Un modelo captura una vista de un sistema del mundo real. Es una abstracción de dicho sistema, considerando un cierto propósito. Así, el modelo describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle.

Diagrama: una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos

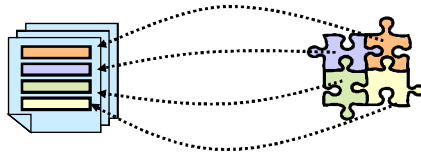
OMG UML 1.4 Specification

... Modelos y Diagramas



Un proceso de desarrollo de software debe ofrecer un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés

El código fuente del sistema es el modelo más detallado del sistema (y además es ejecutable). Sin embargo, se requieren otros modelos ...



Cada modelo es completo desde su punto de vista del sistema, sin embargo, existen relaciones de trazabilidad entre los diferentes modelos

Diagramas de UML

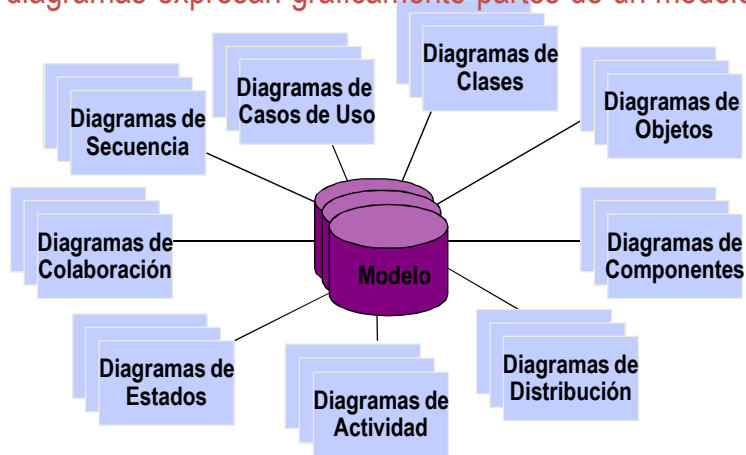


- ☐ Diagrama de Casos de Uso
- ☐ Diagrama de Clases
- ☐ Diagrama de Objetos
- ☐ Diagramas de Comportamiento
- ☐ Diagrama de Estados
- ☐ Diagrama de Actividad
- ☐ Diagramas de Interacción
- ☐ Diagrama de Secuencia
- ☐ Diagrama de Colaboración
- ☐ Diagramas de implementación
- ☐ Diagrama de Componentes
- ☐ Diagrama de Despliegue

... Diagramas de UML

IES INSTITUTO
DE ESTUDIOS
SUPERIORES

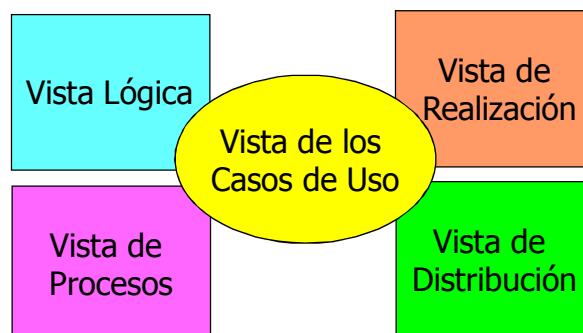
Los diagramas expresan gráficamente partes de un modelo



Organización de Modelos

IES INSTITUTO
DE ESTUDIOS
SUPERIORES

4+1 vistas de Kruchten (1995)



Este enfoque sigue el browser de Rational Rose

Organización de Modelos



-El modelado de sistemas es el proceso para desarrollar modelos abstractos de un sistema, donde cada modelo presenta una visión o perspectiva diferente de dicho sistema. En general, el modelado de sistemas se ha convertido en un medio para representar el sistema usando algún tipo de notación gráfica, que ahora casi siempre se basa en notaciones en el Lenguaje de Modelado Unificado (UML).

-Los modelos se usan durante el proceso de ingeniería de requerimientos para ayudar a derivar los requerimientos de un sistema, durante el proceso de diseño para describir el sistema a los ingenieros que implementan el sistema, y después de la implementación para documentar la estructura y la operación del sistema.

Organización de Modelos



Es posible desarrollar modelos tanto del sistema existente como del sistema a diseñar:

- 1. Los modelos del sistema existente se usan durante la ingeniería de requerimientos:** Ayudan a aclarar lo que hace el sistema existente y pueden utilizarse como base para discutir sus fortalezas y debilidades. Posteriormente, conducen a los requerimientos para el nuevo sistema.
- 2. Los modelos del sistema nuevo se emplean durante la ingeniería de requerimientos:** para ayudar a explicar los requerimientos propuestos a otros participantes del sistema. Los ingenieros usan tales modelos para discutir las propuestas de diseño y documentar el sistema para la implementación. En un proceso de ingeniería dirigido por modelo, es posible generar una implementación de sistema completa o parcial a partir del modelo del sistema.

El aspecto más importante de un modelo del sistema es que deja fuera los detalles. Un modelo es una abstracción del sistema a estudiar, y no una representación alternativa de dicho sistema. De manera ideal, una representación de un sistema debe mantener toda la información sobre la entidad a representar. Una abstracción simplifica y recoge deliberadamente las características más destacadas. Por ejemplo, en el muy improbable caso de que este libro se entregue por capítulos en un periódico, la presentación sería una abstracción de los puntos clave del libro. Si se tradujera del inglés al italiano, sería una representación alternativa. La intención del traductor sería mantener toda la información como se presenta en inglés.

Desde diferentes perspectivas, usted puede desarrollar diferentes modelos para representar el sistema. Por ejemplo:

1. Una perspectiva externa, donde se modelen el contexto o entorno del sistema.
2. Una perspectiva de interacción, donde se modele la interacción entre un sistema y su entorno, o entre los componentes de un sistema.
3. Una perspectiva estructural, donde se modelen la organización de un sistema o la estructura de datos que procese el sistema.
4. Una perspectiva de comportamiento, donde se modele el comportamiento dinámico del sistema y cómo responde ante ciertos eventos.

El UML tiene numerosos tipos de diagramas y, por lo tanto, soporta la creación de muchos diferentes tipos de modelo de sistema. Sin embargo, un estudio en 2007 (Erickson y Siau, 2007) mostró que la mayoría de los usuarios del UML consideraban que cinco tipos de diagrama podrían representar lo esencial de un sistema.

1. **Diagramas de actividad**, que muestran las actividades incluidas en un proceso o en el procesamiento de datos.
2. **Diagramas de caso de uso**, que exponen las interacciones entre un sistema y su entorno.
3. **Diagramas de secuencias**, que muestran las interacciones entre los actores y el sistema, y entre los componentes del sistema.
4. **Diagramas de clase**, que revelan las clases de objeto en el sistema y las asociaciones entre estas clases.
5. **Diagramas de estado**, que explican cómo reacciona el sistema frente a eventos internos y externos.

El Lenguaje de Modelado Unificado es un conjunto compuesto por 13 diferentes tipos de diagrama para modelar sistemas de software. Surgió del trabajo en la década de 1990 sobre el modelado orientado a objetos, cuando anotaciones similares, orientadas a objetos, se integraron para crear el UML. Una amplia revisión (UML 2) se finalizó en 2004. El UML es aceptado universalmente como el enfoque estándar al desarrollo de modelos de sistemas de software. Se han propuesto variantes más generales para el modelado de sistemas.

... Organización de Modelos



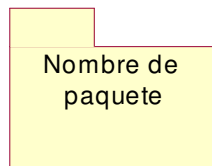
Propuesta de Rational Unified Process (RUP)

- M. de Casos de Uso del Negocio (Business Use-Case Model)
- M. de Objetos del Negocio (Business Object Model)
- M. de Casos de Uso (Use-Case Model)
- M. de Análisis (Analysis Model)
- M. de Diseño (Design Model)
- M. de Despliegue (Deployment Model)
- M. de Datos (Data Model)
- M. de Implementación (Implementation Model)
- M. de Pruebas (Test Model)

Paquetes en UML



Los paquetes ofrecen un mecanismo general para la organización de los modelos/subsistemas agrupando elementos de modelado
Se representan gráficamente como:



... Paquetes en UML

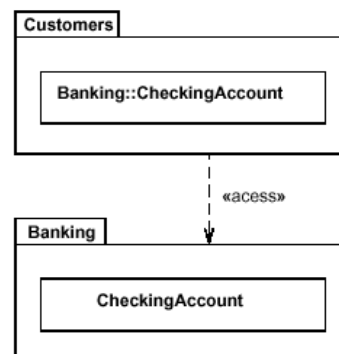


Cada paquete corresponde a un submodelo (subsistema) del modelo (sistema)
Un paquete puede contener otros paquetes, sin límite de anidamiento pero cada elemento pertenece a (está definido en) sólo un paquete
Una clase de un paquete puede aparecer en otro paquete por la importación a través de una relación de dependencia entre paquetes

... Paquetes en UML



Todas las clases no son necesariamente visibles desde el exterior del paquete, es decir, un paquete encapsula a la vez que agrupa
El operador "::" permite designar una clase definida en un contexto distinto del actual



... Paquetes en UML

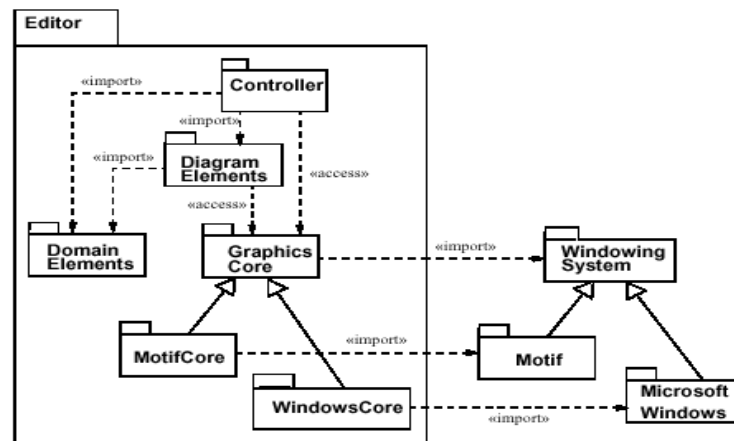
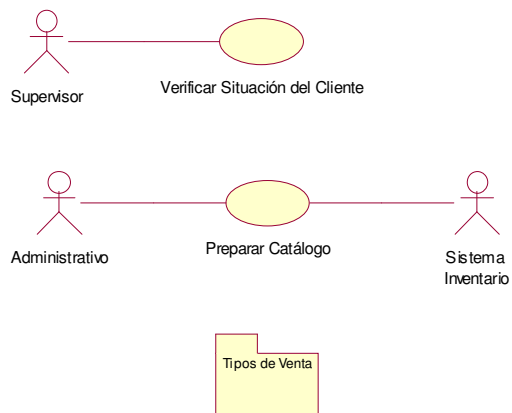


Diagrama de Casos de Uso

Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje

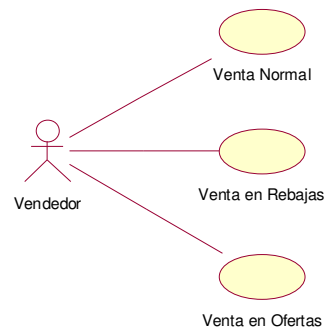
No pertenece estrictamente al enfoque orientado a objeto, es una técnica para capturar de requisitos



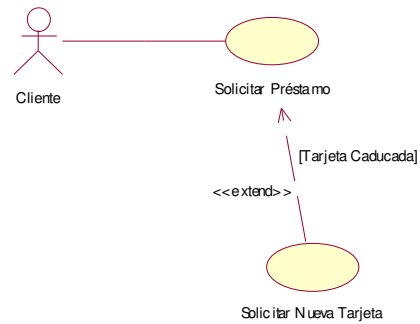
... Ejemplos



En el paquete tipos de venta:



Otro Ejemplo



... Ejemplos

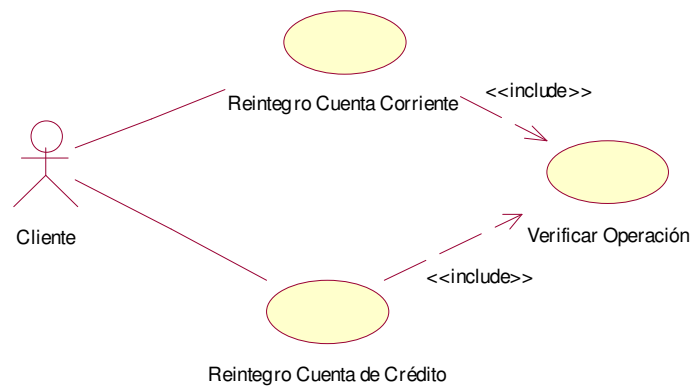


Diagrama de Secuencia

IES INSTITUTO DE ESTUDIOS SUPERIORES

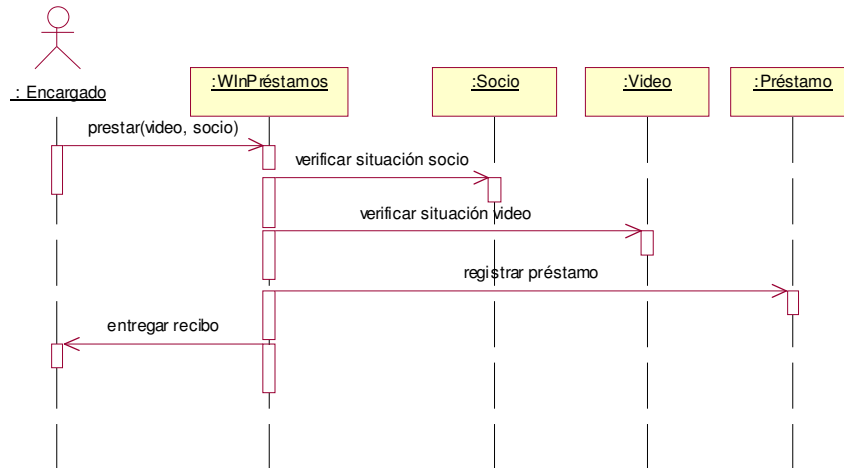


Diagrama de Colaboración

IES INSTITUTO DE ESTUDIOS SUPERIORES

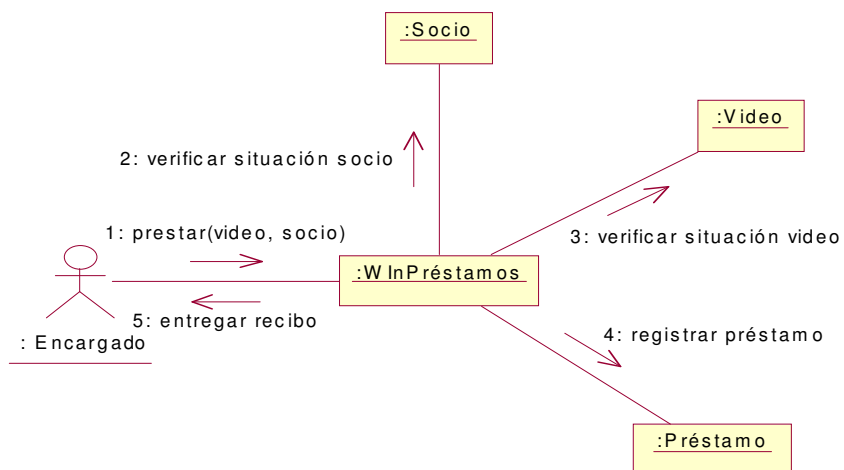


Diagrama de Clases



El Diagrama de Clases es el diagrama principal para el análisis y diseño

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia

La definición de clase incluye definiciones para atributos y operaciones

El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones