
Diagrama de Estados

1

III. El Paradigma OO: Diagrama de Estados

Diagrama de Estados

- Los Diagramas de Estados representan autómatas de estados finitos, desde el p.d.v. de los estados y las transiciones
- Son útiles sólo para los objetos con un comportamiento significativo

2

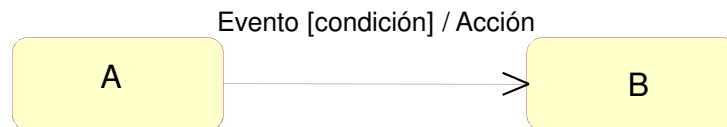
... Diagrama de Estados

- Cada objeto está en un estado en cierto instante
- El estado está caracterizado parcialmente por los valores de algunos de los atributos del objeto
- El estado en el que se encuentra un objeto determina su comportamiento
- Cada objeto sigue el comportamiento descrito en el D. de Estados asociado a su clase
- Los D. De Estados y escenarios son complementarios

3

... Diagrama de Estados

- Estados y Transiciones

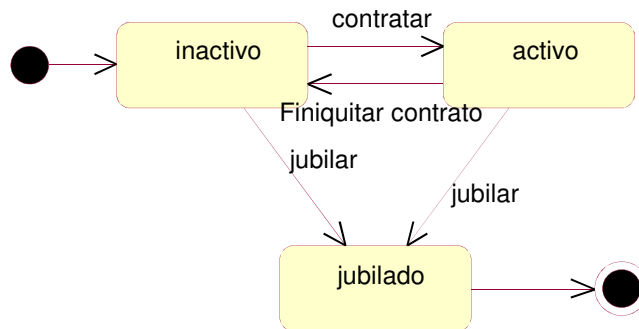


Tanto el evento como la acción se consideran instantáneos

4

... Diagrama de Estados

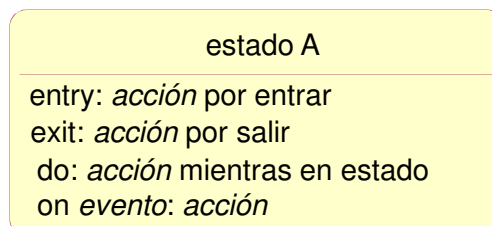
- Ejemplo de un Diagrama de Estados para la clase persona:



5

... Acciones

- Se puede especificar el ejecutar una acción como consecuencia de entrar, salir, estar en un estado, o por la ocurrencia de un evento:



6

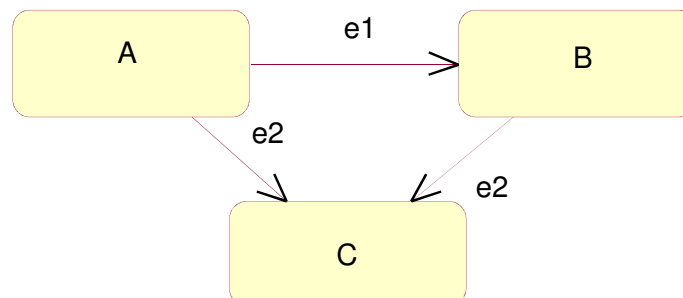
Generalización de Estados

- Podemos reducir la complejidad de estos diagramas usando la generalización de estados
- Distinguimos así entre **superestado** y **subestados**
- Un estado puede contener varios subestados disjuntos
- Los subestados heredan las variables de estado y las transiciones externas

7

Generalización de Estados

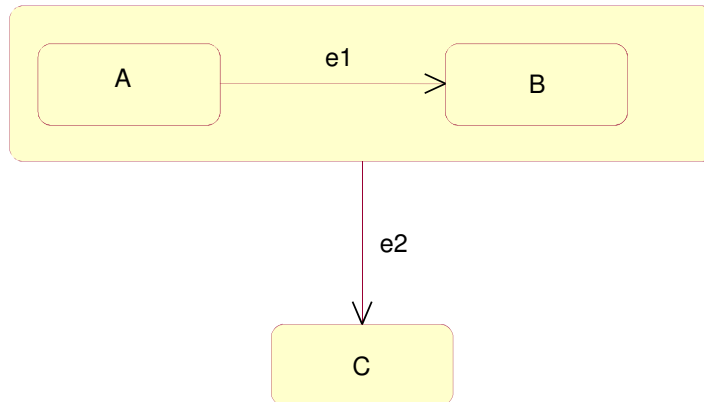
- Ejemplo:



8

Generalización de Estados

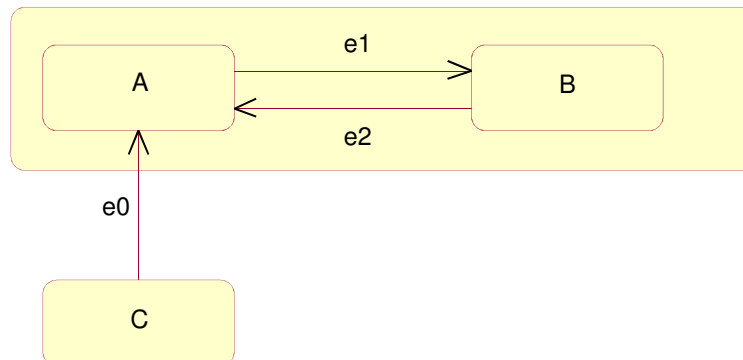
- Quedaría como:



9

... Generalización de Estados

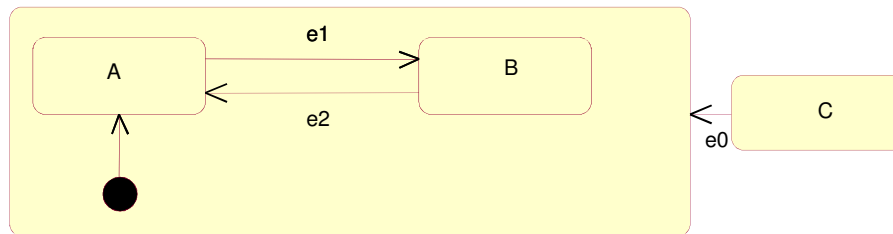
- Las transiciones de entrada deben ir hacia subestados específicos:



10

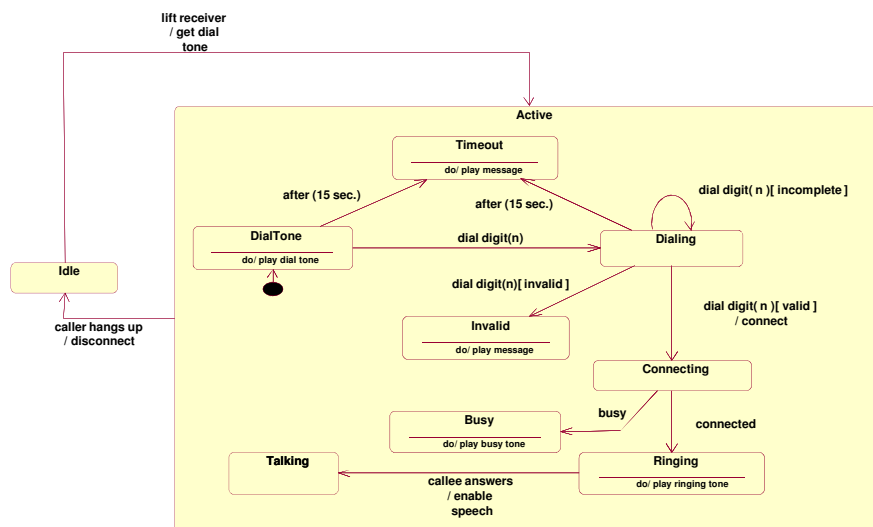
... Generalización de Estados

- Es preferible tener estados iniciales de entrada a un nivel de manera que desde los niveles superiores no se sepa a qué subestado se entra:



11

... Generalización de Estados



12

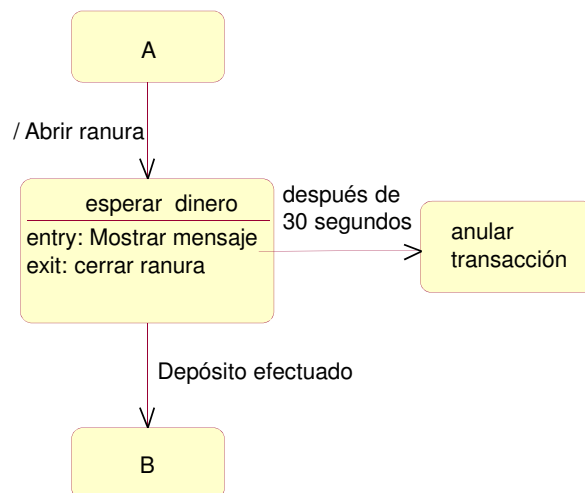
Transiciones temporizadas

- Las esperas son actividades que tienen asociada cierta duración
- La actividad de espera se interrumpe cuando el evento esperado tiene lugar
- Este evento desencadena una transición que permite salir del estado que alberga la actividad de espera. El flujo de control se transmite entonces a otro estado

13

... Transiciones temporizadas

- Ejemplo:



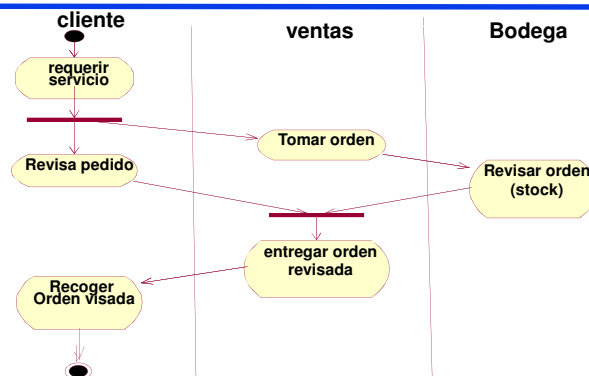
14

Diagrama de Actividad

- El Diagrama de Actividad es una especialización del Diagrama de Estado, organizado respecto de las acciones y usado para especificar:
 - Un método
 - Un caso de uso
 - Un proceso de negocio (Workflow)
- Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad

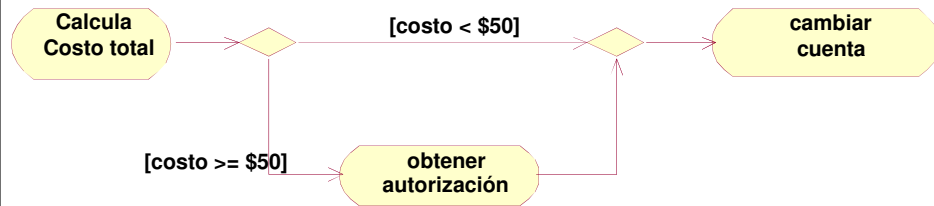
15

Ejemplos



16

... Ejemplos



17

Diagrama de Componentes

18

Diagrama de Componentes

- Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones
- Muestran las opciones de realización incluyendo código fuente, binario y ejecutable

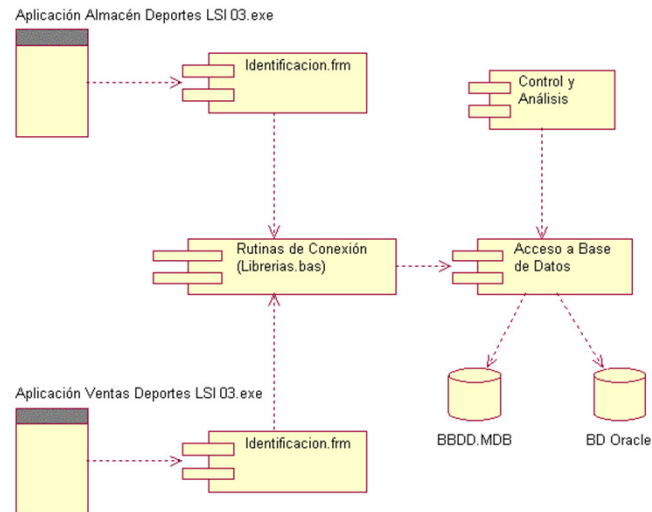
19

...Diagrama de Componentes

- Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc.
- Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente

20

...Diagrama de Componentes



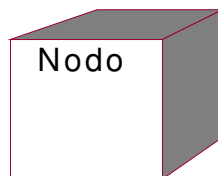
21

Diagrama de Despliegue

22

Diagrama de Despliegue

- Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos



23

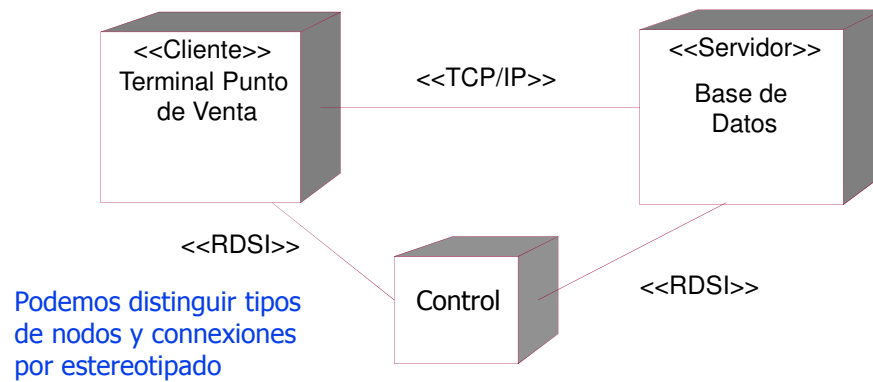
... Diagrama de Despliegue

- Los estereotipos permiten precisar la naturaleza del equipo:
 - Dispositivos
 - Procesadores
 - Memoria
- Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez estereotiparse

24

... Diagrama de Despliegue

- Ejemplo de conexión entre nodos:



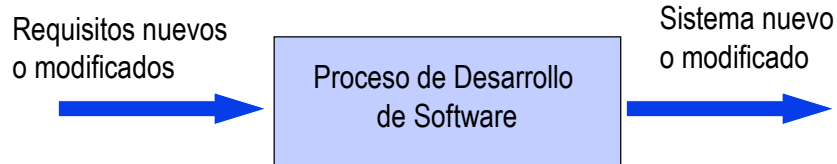
25

Proceso de Desarrollo de SW basado en UML

26

¿Qué es un Proceso de Desarrollo de SW?

- Define **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo



- No existe un proceso de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable

27

Rational Unified Process (RUP)



Rational Unified Process
1998

Rational Objectory Process
1996-1997

Objectory Process
1987-1995

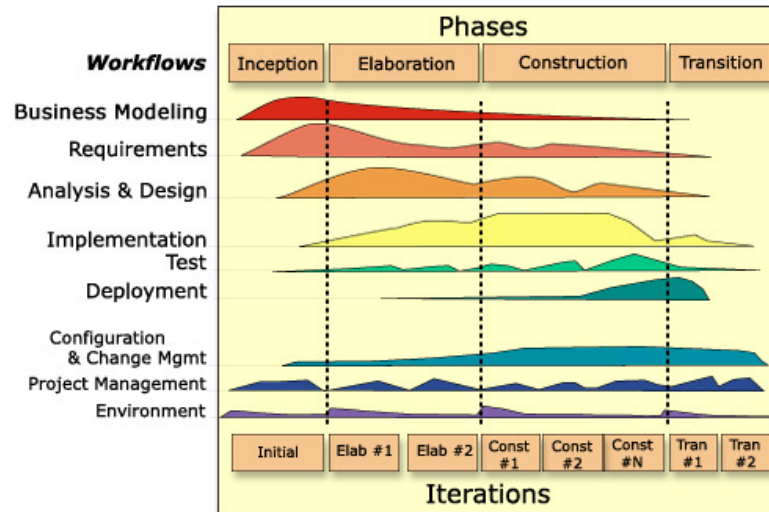
Enfoque Ericsson

- Pruebas funcionales
- Pruebas de desempeño
- Gestión de requisitos
- Gestión de cambios y configuración
- Ingeniería de Negocio
- Ingeniería de datos
- Diseño de interfaces

UML

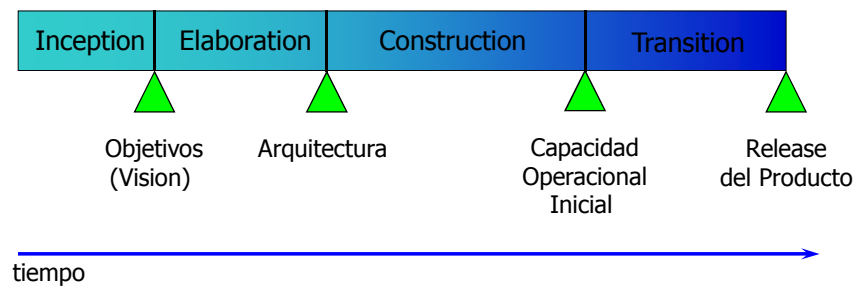
28

Dos Dimensiones



29

Fases e Hitos (Milestones)



30

... Elementos en RUP

Workers

Analyst workers

- Business-Process Analyst
- Business Designer
- Business-Model Reviewer
- Requirements Reviewer
- System Analyst
- Use-Case Specifier
- User-Interface Designer

Developer workers

- Architect
- Architecture Reviewer
- Code Reviewer
- Database Designer
- Design Reviewer
- Designer
- Implementer
- Integrator

Testing professional workers

- Test Designer
- Tester

Manager workers

- Change Control Manager
- Configuration Manager
- Deployment Manager
- Process Engineer
- Project Manager
- Project Reviewer

Other workers

- Any Worker
- Course Developer
- Graphic Artist
- **Stakeholder**
- System Administrator
- Technical Writer
- Tool Specialist

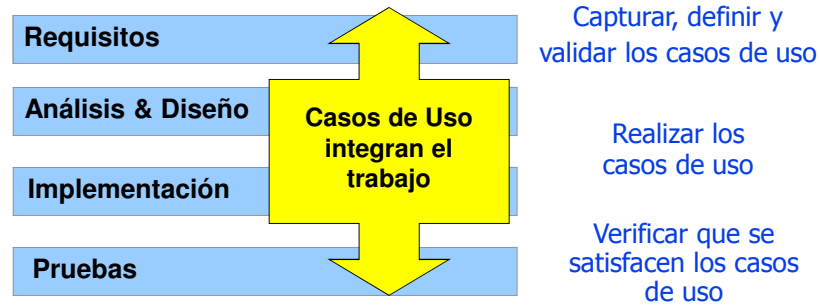
31

Características Esenciales de RUP

- ▣ Proceso Dirigido por los Casos de Uso
- ▣ Proceso Iterativo e Incremental
- ▣ Proceso Centrado en la Arquitectura

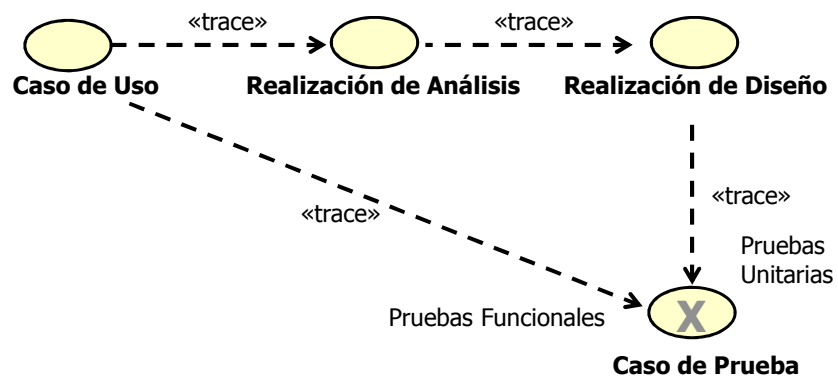
32

Proceso dirigido por los Casos de Uso



33

... Proceso dirigido por los Casos de Uso



[The Unified Software Development Process. I. Jacobson, G. Booch and J. Rumbaugh. Addison-Wesley, 1999]

34

... Proceso dirigido por los Casos de Uso

Estado de aspectos de los Casos de Uso al finalizar cada fase

	Modelo de Negocio Terminado	Casos de Uso Identificados	Casos de Uso Descritos	Casos de Uso Analizados	Casos de Uso Diseñados, Implementados y Probados
Fase de Concepción	50% - 70%	50%	10%	5%	Muy poco, puede que sólo algo relativo a un prototipo para probar conceptos
Fase de Elaboración	Casi el 100%	80% o más	40% - 80%	20% - 40%	Menos del 10%
Fase de Construcción	100%	100%	100%	100%	100%
Fase de Transición					

The Unified Software Development Process. I. Jacobson, G. Booch y J. Rumbaugh. página 358. Addison-Wesley, 1999.

35

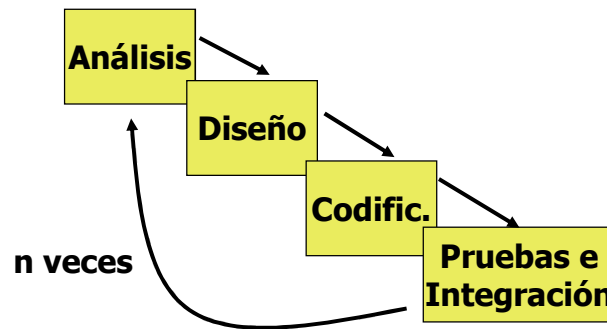
Proceso Iterativo e Incremental

- El ciclo de vida iterativo se basa en la evolución de prototipos ejecutables que se muestran a los usuarios y clientes
- En el ciclo de vida iterativo a cada iteración se reproduce el ciclo de vida en cascada a menor escala
- Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes

36

... Proceso Iterativo e Incremental

- Las actividades se encadenan en una mini-cascada con un alcance limitado por los objetivos de la iteración



37

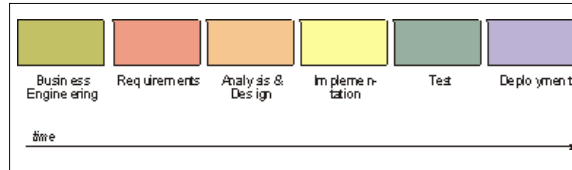
... Proceso Iterativo e Incremental

- Cada iteración comprende:
 - Planificar la iteración (estudio de riesgos)
 - Análisis de los Casos de Uso y escenarios
 - Diseño de opciones arquitectónicas
 - Codificación y pruebas. La integración del nuevo código con el existente de iteraciones anteriores se hace gradualmente durante la construcción
 - Evaluación de la entrega ejecutable (evaluación del prototipo en función de las pruebas y de los criterios definidos)
 - Preparación de la entrega (documentación e instalación del prototipo)

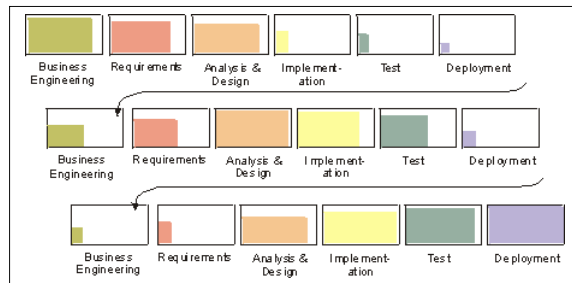
38

Proceso Iterativo e Incremental

Enfoque
Cascada



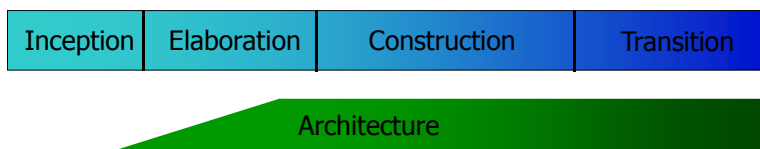
Enfoque
Iterativo e
Incremental



39

Proceso Centrado en la Arquitectura

- ▣ Arquitectura de un sistema es la organización o estructura de sus partes más relevantes
- ▣ Un arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades
- ▣ RUP establece refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo evolutivo



40

Fases del Ciclo de Vida

- El ciclo de vida consiste en una serie de ciclos, cada uno de los cuales produce una nueva versión del producto
- Cada ciclo está compuesto por fases y cada una de estas fases está compuesta por un número de iteraciones
- Las fases son:
 - Inicio o Estudio de oportunidad
 - Elaboración
 - Construcción
 - Transición

41

...Fases del Ciclo de Vida

- Inicio o Estudio de oportunidad (*inception*)
 - Define el **ámbito y objetivos del proyecto**
 - Se define la **funcionalidad** y capacidades del producto
- Elaboración
 - Tanto la **funcionalidad** como el dominio del problema se estudian **en profundidad**
 - Se define una **arquitectura básica**
 - Se **planifica el proyecto** considerando recursos disponibles

42

...Fases del Ciclo de Vida

▪ Construcción

- El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación
- Las fases de estudio y análisis sólo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye (se permiten cambios en la estructura)
- Gran parte del trabajo es programación y pruebas
- Se documenta tanto el sistema construido como el manejo del mismo
- Esta fase proporciona un producto construido junto con la documentación

43

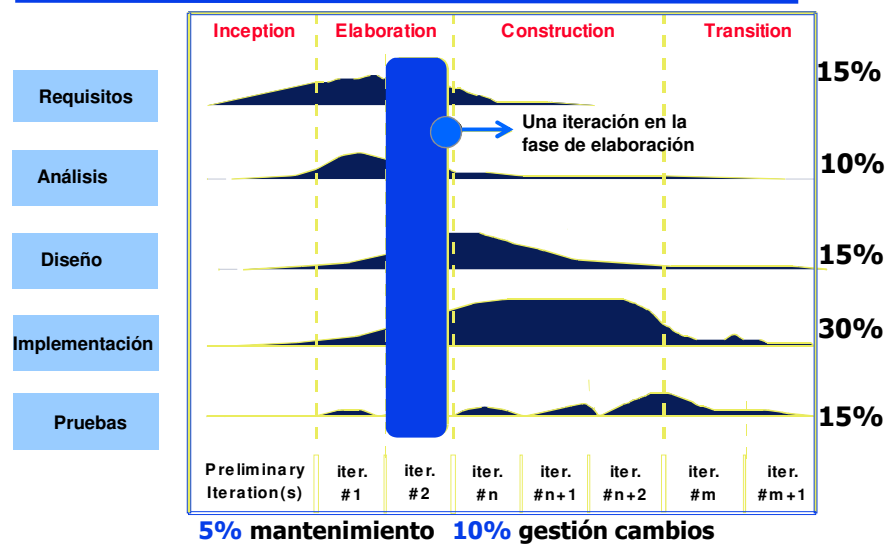
...Fases del Ciclo de Vida

▪ Transición

- Se libera el producto y se entrega al usuario para un uso real
- Se incluyen tareas de marketing, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.
- Los manuales de usuario se completan y refinan con la información anterior
- Estas tareas se realizan también en iteraciones

44

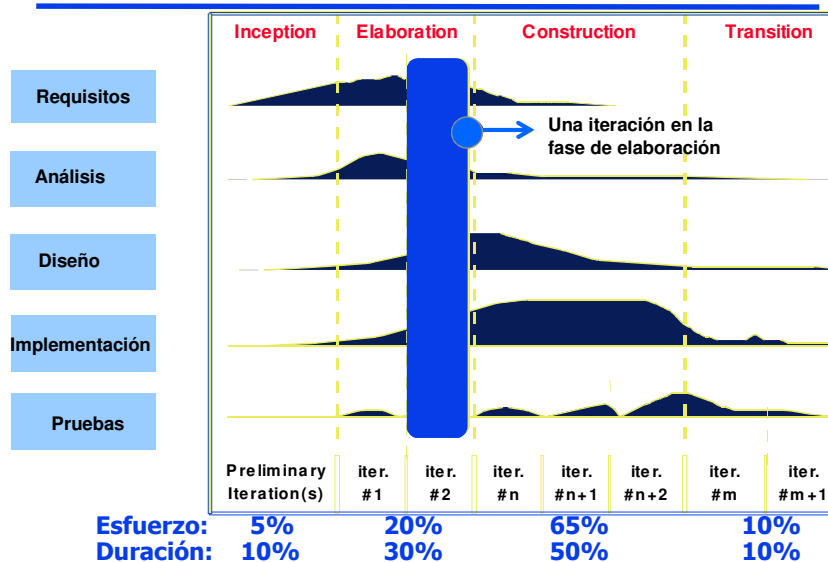
Esfuerzo respecto de las Workflows



45

IV. Proceso de Desarrollo de SW basado en UML

...Esfuerzo respecto de las Fases



46

Conclusiones

47

V. Conclusiones

Claves en el Desarrollo de SI

Notación
UML



Herramientas
p.e. Rational Rose

Proceso
p.e. Rational Unified Process

48

Contexto de Desarrollo: Grado de Complejidad



49

Modelado de SI: Algunas Reflexiones

- Modelar para la concebir el sistema y/o para la documentarlo
- Pragmatismo, los modelos deben ser útiles
- Sencillez y Elegancia
- Distintos nivel de abstracción, diferentes modelos
- Seguimiento de transformaciones durante el proceso (*Traceability*)
- Sincronización de modelos
- Dificultades para la introducción de técnicas y herramientas de modelado

50

... Finalmente

- Apostar por enfoque Orientado a Objetos usando notación UML
- Problemas actuales en implementación, al usar entornos de programación visual y/o bases de datos relacionales
- Posibles mejoras a mediano plazo
 - Evolución: Uso de BDOO y/o mejoras en los LPOO
 - Revolución: Generación Automática de Código a partir de Modelos OO (Compilación de Modelos)