

# Tesina de Grado

“Una herramienta para medición, control y seguimiento de proyectos”

por Andrés Romero  
Directora: Claudia Pons

# Agenda

- Motivación
- Conceptos previos
- Acerca de la herramienta desarrollada
  - Características de la herramienta
  - Descripción de la herramienta
  - Vista de la herramienta en funcionamiento
- Aportes realizados
- Extensiones a la herramienta

# Motivación



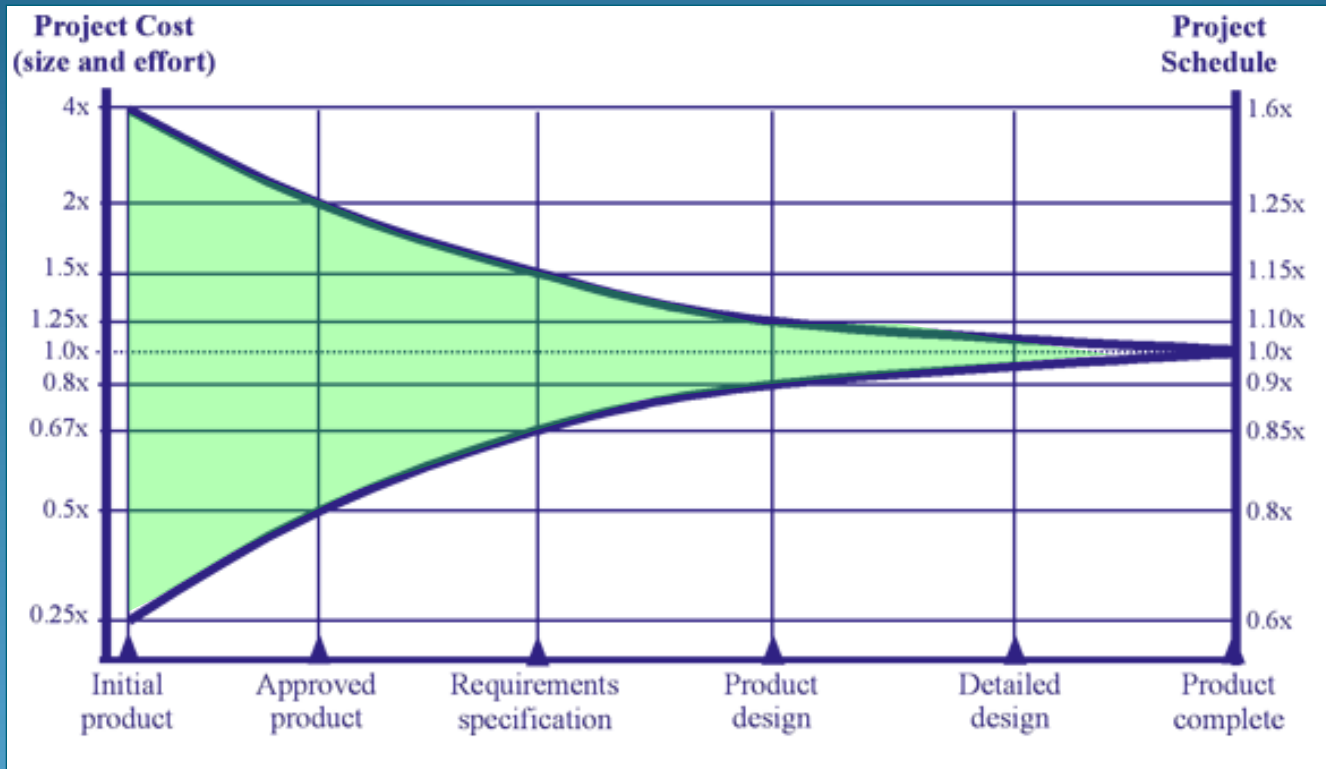
*“No se puede controlar lo que no se puede medir”*

*De Marco.*

# Motivación

- **¿Para que realizar control?**
  - Evaluar calidad
  - Evaluar productividad
  - **Conocer el estado actual en el avance**
    - Reaccionar a tiempo
    - Tomar mejores decisiones
    - Contrastar estimaciones

# Motivación



# Motivación

- El objetivo entonces es desarrollar una herramienta que implemente:
  - Un conjunto de características deseables detectado
  - Seguimientos de proyectos en cuanto a tamaño
  - Seguimiento y evaluación de proyectos y productos, en cuanto a calidad

Los dos últimos puntos están enmarcados en los conceptos definidos por Gabriela Robiolo en su Tesis de Magíster y Doctorado

# Conceptos previos

- Unified Modeling language (UML)
- XML Metadata Interchange (XMI)
- Conceptos de la tesis de G. Robiolo
  - Transacciones
- Productos sobre los que se desarrolló
  - Eclipse
  - Rational Software Architect

## Conceptos previos

# Unified Modeling Language (UML)

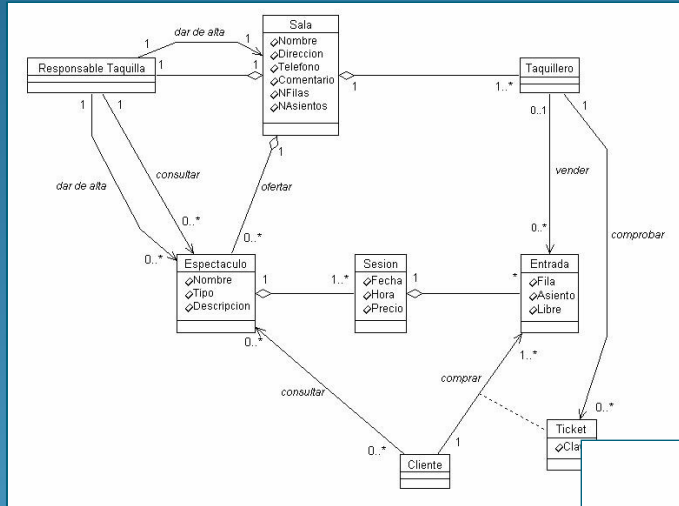
- Lenguaje para especificar, construir, visualizar y documentar artefactos de un sistema de software OO.
- Diagramas complementarios
- UML es un metamodelo



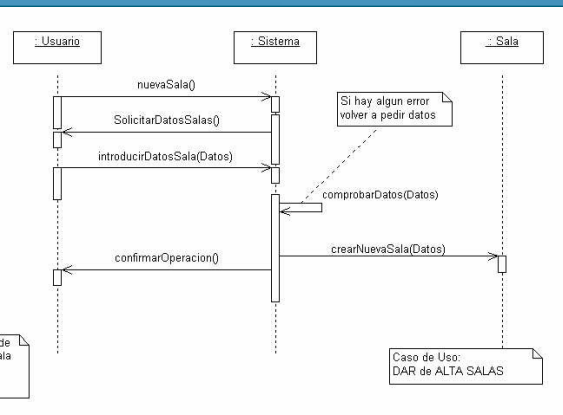
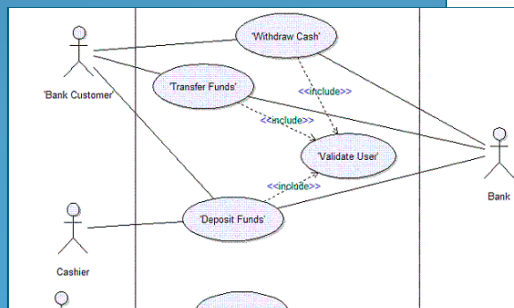


# Conceptos previos

## Unified Modeling Language (UML)

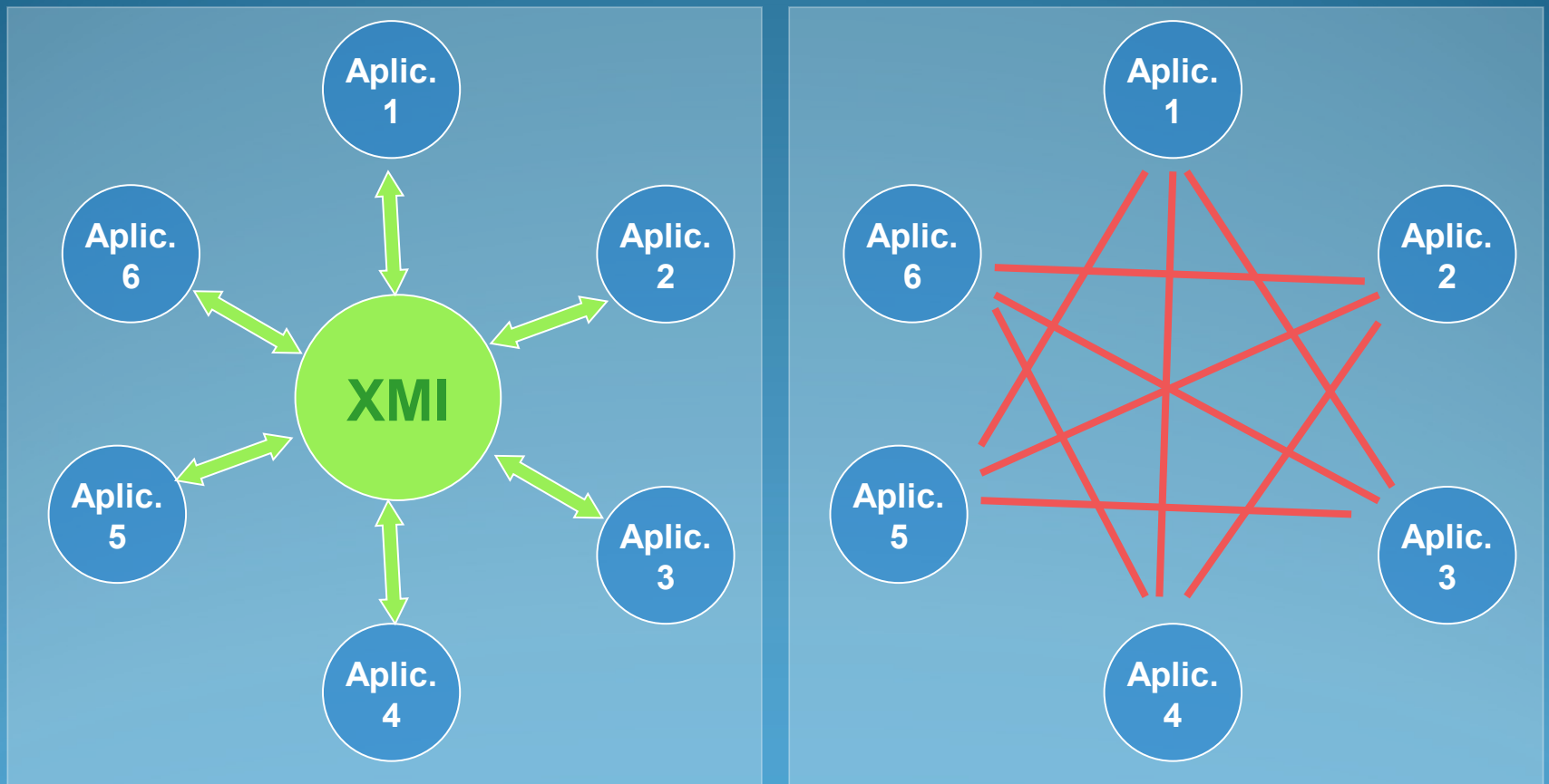


Los diagramas de UML que son importantes para esta tesina son los de Clases, Secuencia y Casos de Uso



# Conceptos previos

## XML Metadata Interchange (XMI)



# Conceptos previos

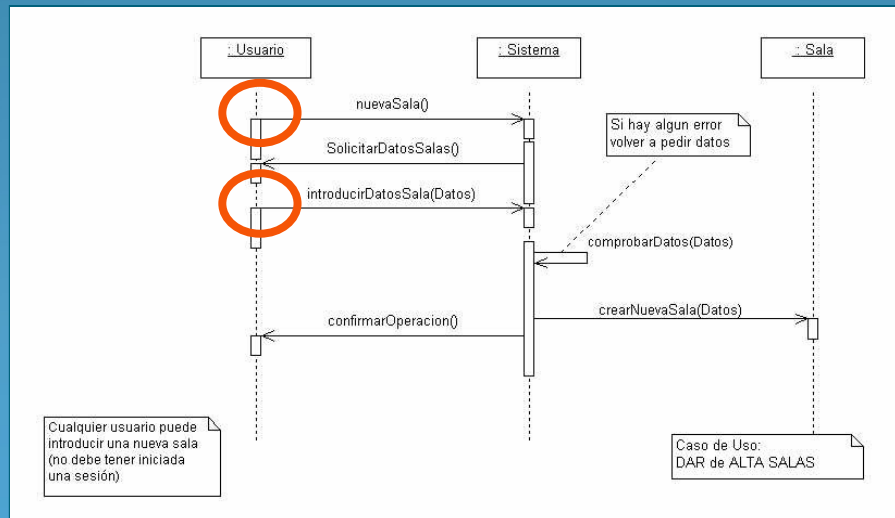
## XML Metadata Interchange (XMI)

```
<XMI version="1.1" xmlns:UML="org.omg/UML 1.3" >
... (header)
<XMI.content>
<UML:Class name="Departamento"
xmi.id="Departamento"/>
<UML:Class name="Instructor"
xmi.id="Instructor"/>
<UML:Class name="Professor"
xmi.id="Professor" generalization="Instructor"/>
<UML:Class name="Catedrático"
xmi.id="Catedrático" generalization="Instructor"/>
<UML:Class name="Monitor"
xmi.id="Monitor" generalization="Instructor"/>
<UML:Association>
<UML:Association.connection>
<UML:AssociationEnd name="instructores"
type="Instructor"/>
<UML:AssociationEnd name="miembroDe"
type="Departamento"/>
</UML:Association.connection>
</UML:Association>
</XMI.content>
</XMI>
```

# Conceptos previos

## Transacciones

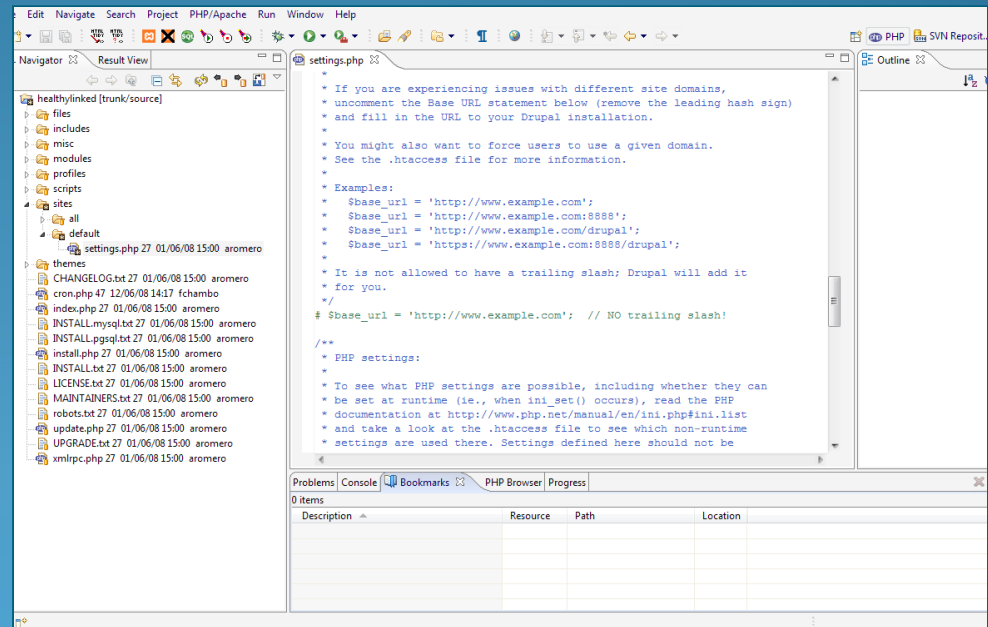
- Estímulos del actor hacia el sistema
- Un diagrama de secuencia puede tener una o varias transacciones, dependiendo del criterio de quien los escribe



# Conceptos previos

# Eclipse

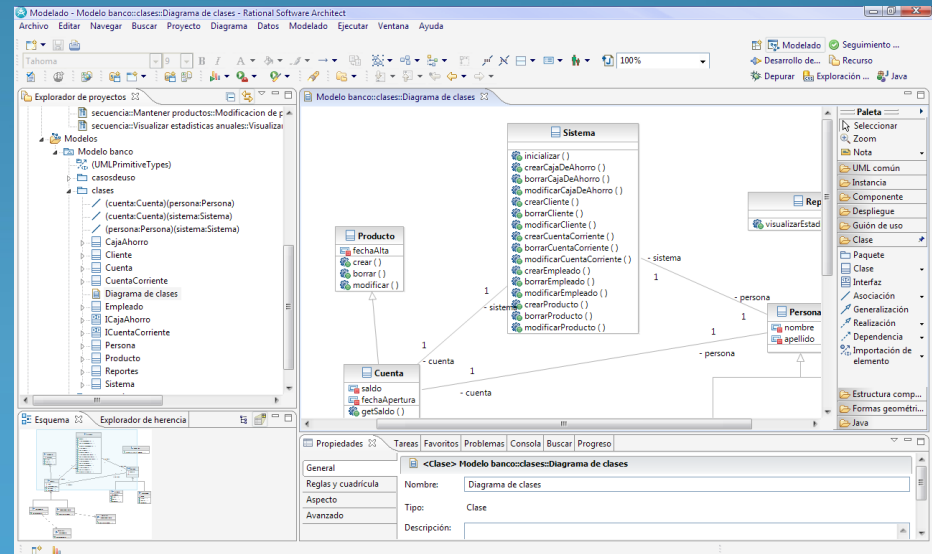
- IDE y Framework para desarrollo de aplicaciones
- Muy utilizado en el mercado
- Multiplataforma
- Soporta múltiples lenguajes



# Conceptos previos

# IBM Rational Software Architect

- Eclipse + 700 Plugins
- Ambiente integrado para el desarrollo de proyectos
- Soporte UML, XMI.



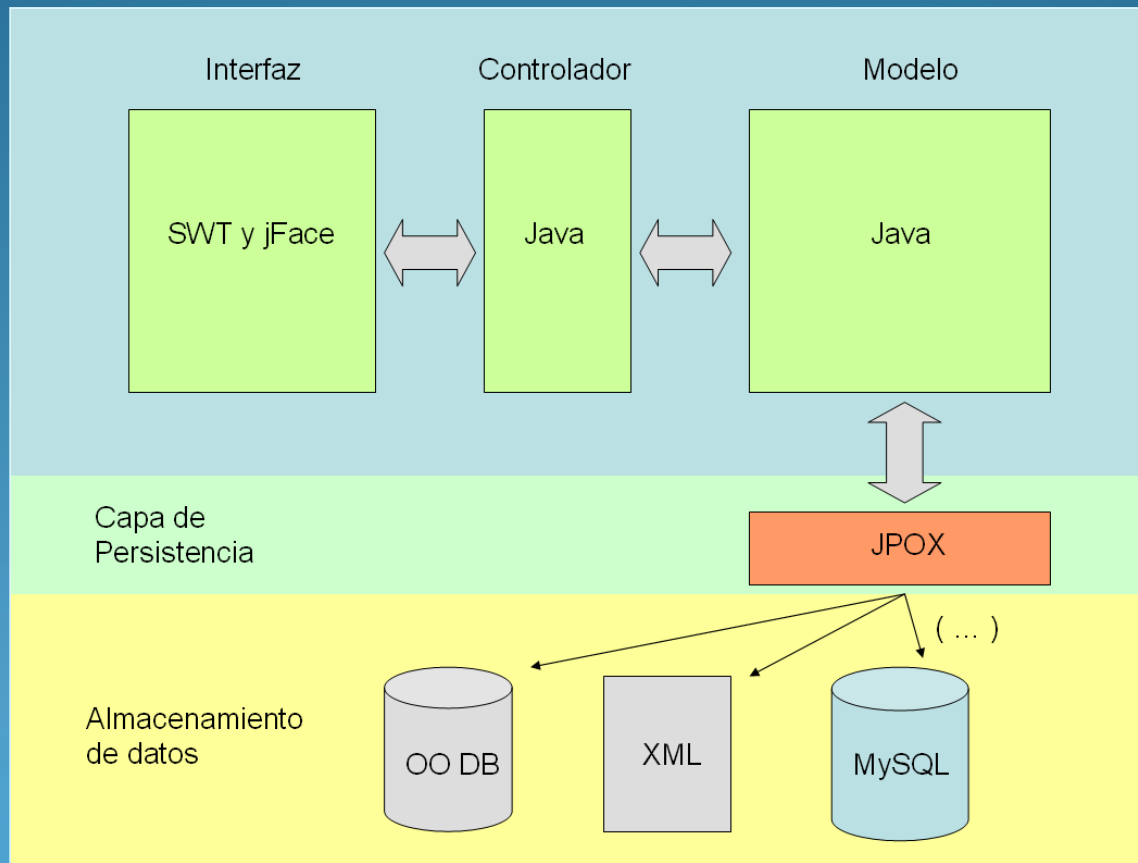
# Acercas del sistema desarrollado

## Características de la herramienta

- Capacidad de obtención de información desde modelos UML
- Separación entre la extracción y el procesamiento de los datos
- Integración con el ambiente de desarrollo habitual
- Seguimiento de proyectos en el tiempo
- Monitorear múltiples proyectos
- Resultados usables
- Portabilidad
- Capacidad de definir métricas personalizadas

# Acercas del sistema desarrollado

## Descripción de la herramienta

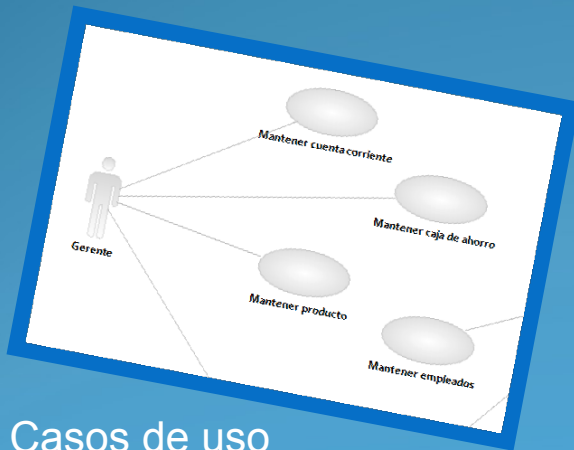




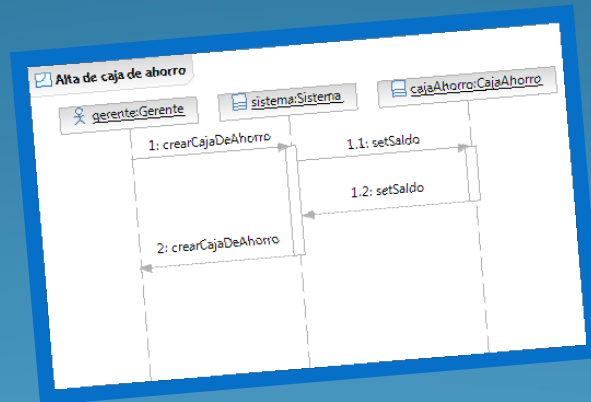


# Seguimiento en tamaño

- ¿Cómo se mide el avance?



Casos de uso



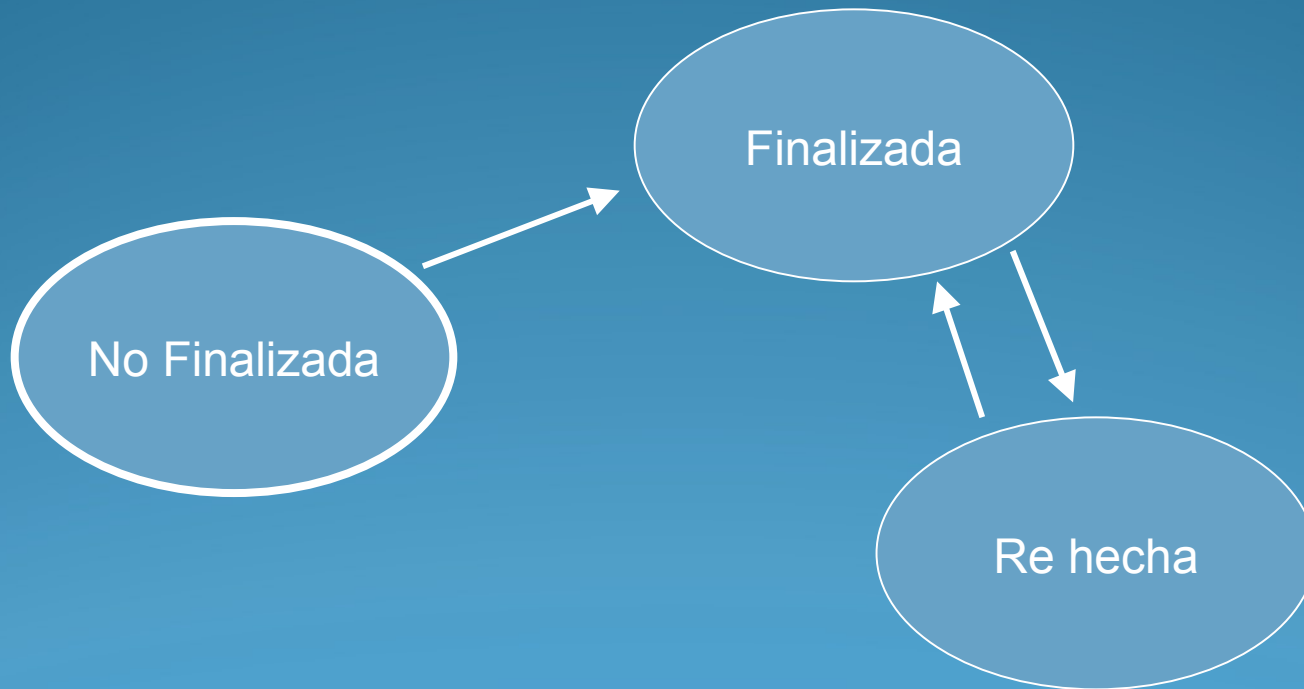
Diag. de secuencia



Diag. de clases

# Seguimiento en tamaño

- Posibles estados de las clases



# Seguimiento en tamaño

- El estado de las clases implica el estado de las transacciones
  - Al menos 1 clase en estado “Rehecha”, entonces toda la transacción se considera Rehecha
  - Al menos 1 clase en estado “No-Finalizada”, entonces toda la transacción se considera “No Finalizada”
  - Todas las clases “Finalizadas”, entonces la transacción se considera finalizada

En el caso en que se encuentre con una clase Rehecha y una clase No Finalizada, pesa mas esta ultima y hace que la transacción quede como No Finalizada.



# Seguimiento en tamaño

Se obtienen del sistema

$CT\ NF + CT\ RH + CT\ F =$   
Total de transacciones

SSCP - Consultar avance de construcción de producto

**Consultar avance de construcción de producto**  
Resultado de la consulta de fechas seleccionada. 'CT' = Cantidad de transacciones

16/07/2008 10:05:41

Caso de uso	CT pactadas	CT incorporadas	CT no finalizadas	CT rehechas	CT finalizadas	Productividad	% de avance
Mantener caja de ahorro	3	0	0	0	3	1,5	1
Mantener clientes	3	0	0	0	3	0,75	1
Mantener cuenta corriente	3	0	0	0	3	0,5	1
Mantener empleados	3	0	0	0	3	0,38	1
Mantener productos	3	0	0	0	3	0,3	1
Visualizar estadísticas anuales	1	0	0	0	1	0,08	1
<b>Total</b>	<b>16</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>16</b>	<b>0,38</b>	<b>1</b>

Volver

$CT\ Incorporadas =$   
Total de transacciones - CT  
pactadas

$Productividad =$   
 $CT\ F / Cant.\ Hs$   
trabajadas

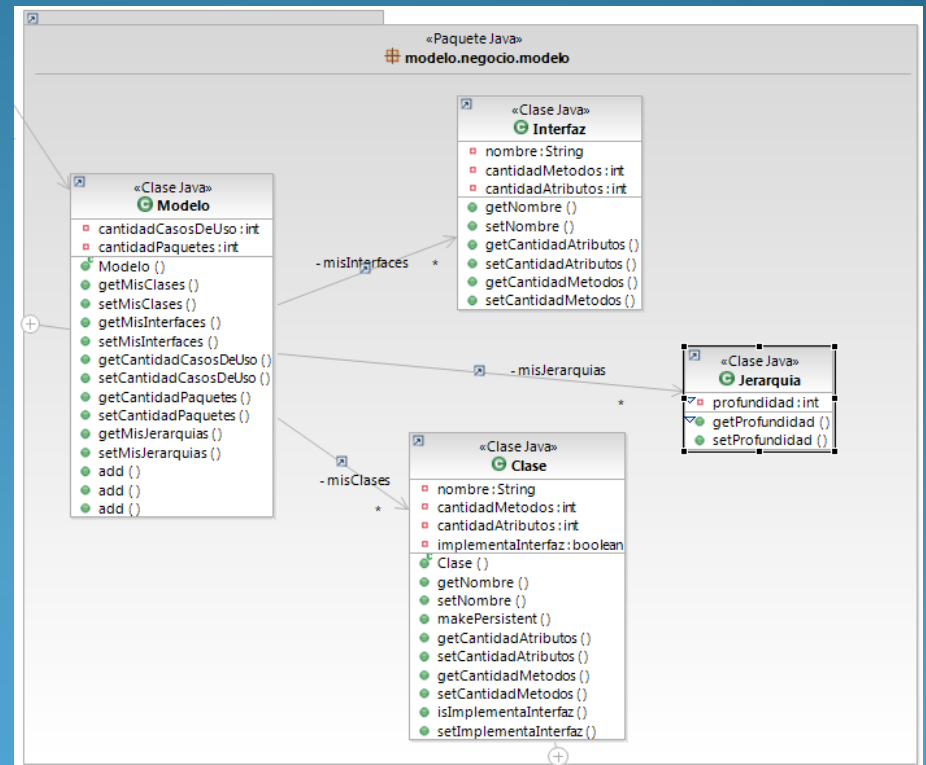
$\% \text{ de avance} =$   
 $CT\ F / (CT\ Pactadas + CT$   
Incorporadas)



# Seguimiento en calidad

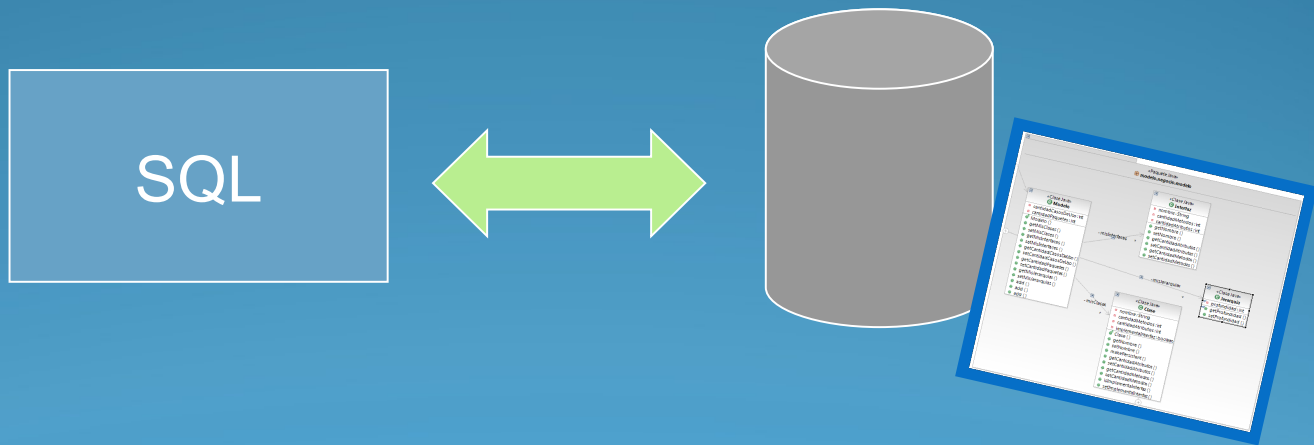
- ¿Como se resolvió las métricas personalizadas?
  - Se realizan 2 pasos:

1) Obtener el modelo desde la fuente de datos XMI y guardarlo en la base de datos



# Seguimiento en calidad

2) ejecutar una consulta SQL  
Sobre el modelo  
generado







# Seguimiento en calidad

- Ventajas
  - Se tiene mucha flexibilidad para definir las métricas
  - Se puede seguir extendiendo el modelo desde el cual se extraen las métricas
- Desventajas
  - No es adecuado para un usuario sin conocimientos de SQL
  - Requiere conocer parte del diseño de la base de datos

# Sobre la herramienta

- Desarrollada 100% orientada a objetos
- ORM JDO JPOX
- UML2
- Plugin de todo producto derivado de Eclipse
- Portable
- Con ligeras modificaciones se podría hacer standalone
- Apego a estándares UML, XMI

# Aportes realizados

- Se completo el desarrollo de la herramienta y se logró:
  - La implementación del concepto de 'Transacción' como indicador del 'peso funcional' de un sistema
  - La separación del proceso de extracción y procesamiento de datos mediante el estándar XML
  - La implementación de un conjunto de buenas características en herramienta CASE para seguimiento de proyectos
- Referencia para desarrollo de futuras herramientas para seguimiento de proyectos

# Extensiones a la herramienta

- Formas alternativas para definir métricas
  - Usuario experto
    - OQL, JDOQL
  - Usuario no experto
    - XML, Interfaces
- Aumentar el potencial conjunto de métricas a generar
- Considerar la medición de los ‘Objetos de Entidad’ como indicadores de tamaño de datos de un sistema
- Cuestiones de usabilidad
  - Mejorar la visualización de los datos
  - Reacción a cambios en el código. Cambiar estado de clases automáticamente.

**¡Muchas gracias!**  
**¿Preguntas?**

