GUÍAS DE DISEÑO CON UML

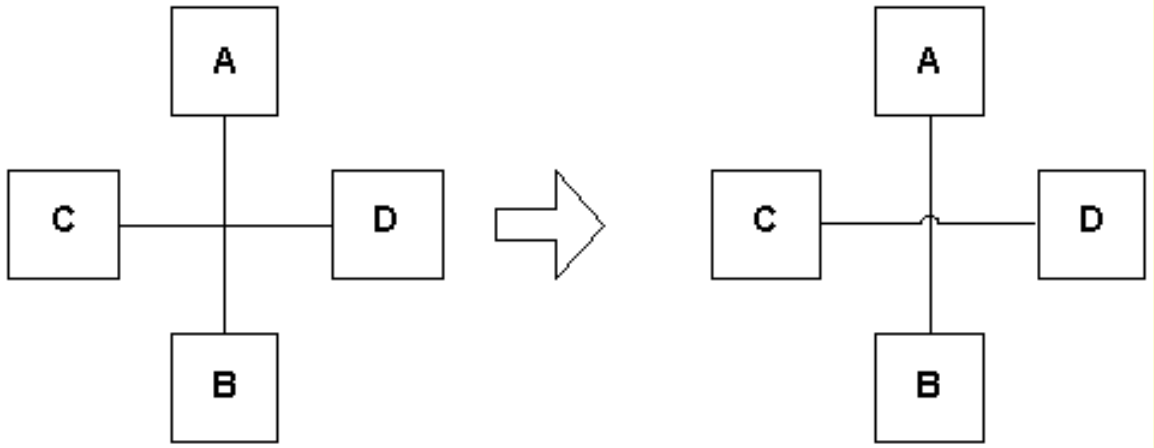
Técnicas para creación de diagramas de software óptimos en UML

ÍNDICE

- Recomendaciones de diseño generales
- Guías para diagramas de Casos de Uso
- Guías para diagramas de Actividad
- Guías para diagramas de Secuencia
- Guías para diagramas de Clases

Recomendaciones De Diseño Generales

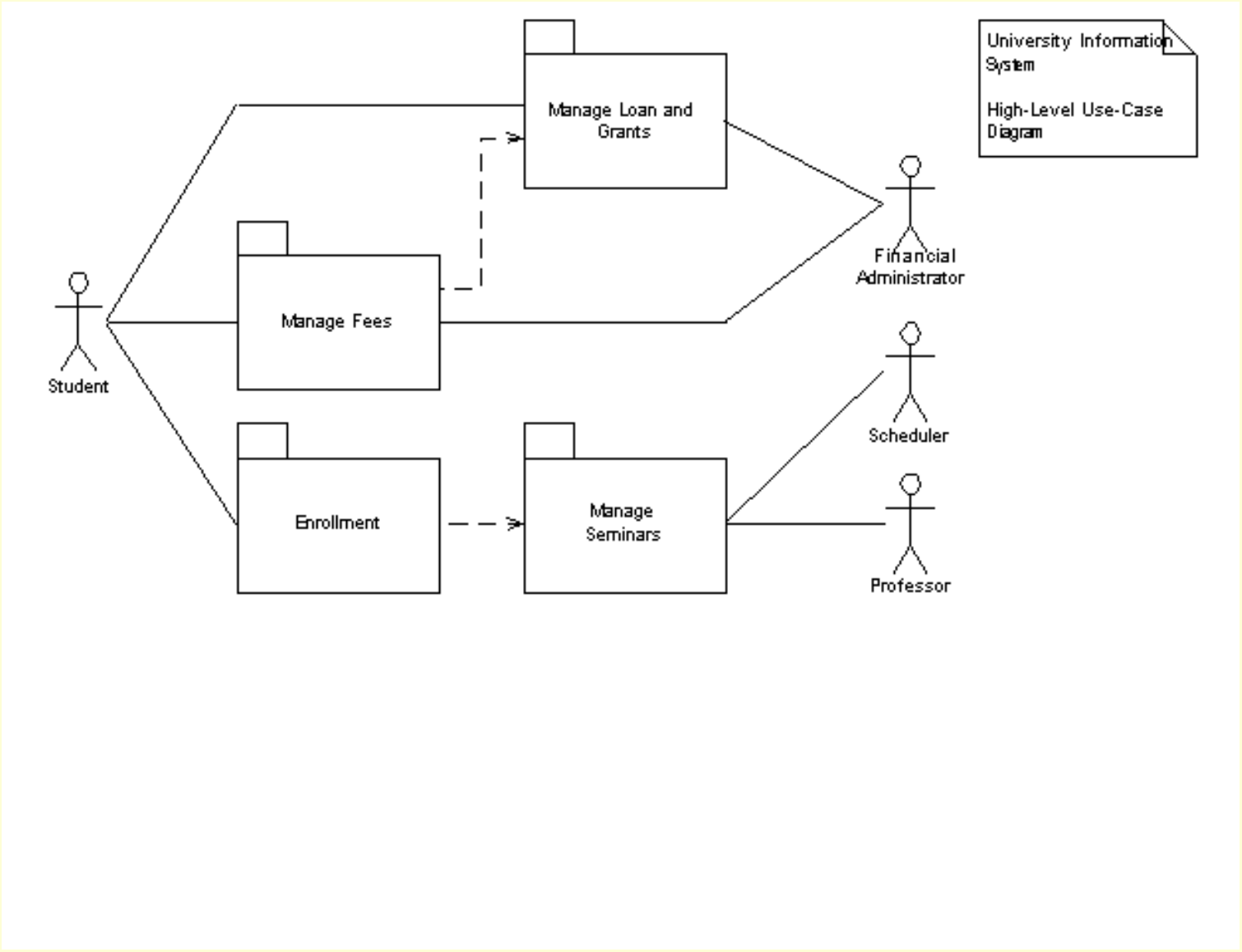
- Líneas:
  - Evitar líneas curvas.
  - Indicar adecuadamente líneas cruzadas.



Recomendaciones De Diseño Generales

- Organizar los diagramas de izquierda a derecha, de arriba a abajo.
- Usar notas para describir diagramas.

- Reorganizar grandes diagramas en varios: Usar Diagramas de Paquetes


University Information System
High-Level Use-Case Diagram
Manage Loan and Grants
Manage Fees
Enrollment
Manage Seminars
Student
Financial Administrator
Scheduler
Professor

Recomendaciones De Diseño Generales

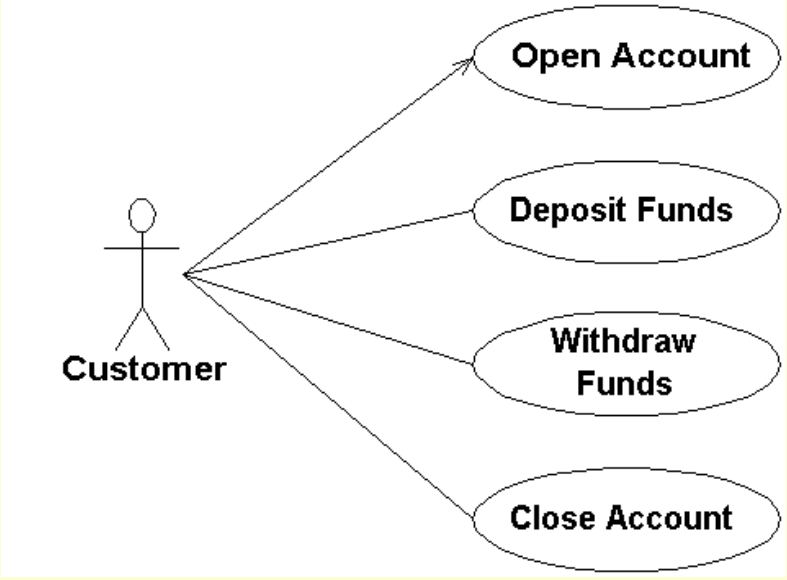
- Establecer y seguir una convención de nombres.
- Considerar añadir colores aclarativos a los diagramas.

Diagramas De Casos De Uso

- Comenzar cuando sea posible los nombres de casos de uso con verbos aclarativos. Ej: Usar *registrar*, *entregar*, *retirar* en vez de *procesar*, *hacer*.
- Proporcionar significado inmediato en los nombres de casos de uso. Ej: Usar “Entregar pedido” en vez de “Enviar paquete vía empresa de transporte”.

Diagramas De Casos De Uso

- Emplazar casos de uso principales arriba y a la izquierda.
- Usar consideraciones temporales al apilar casos de uso.



Diagramas De Casos De Uso

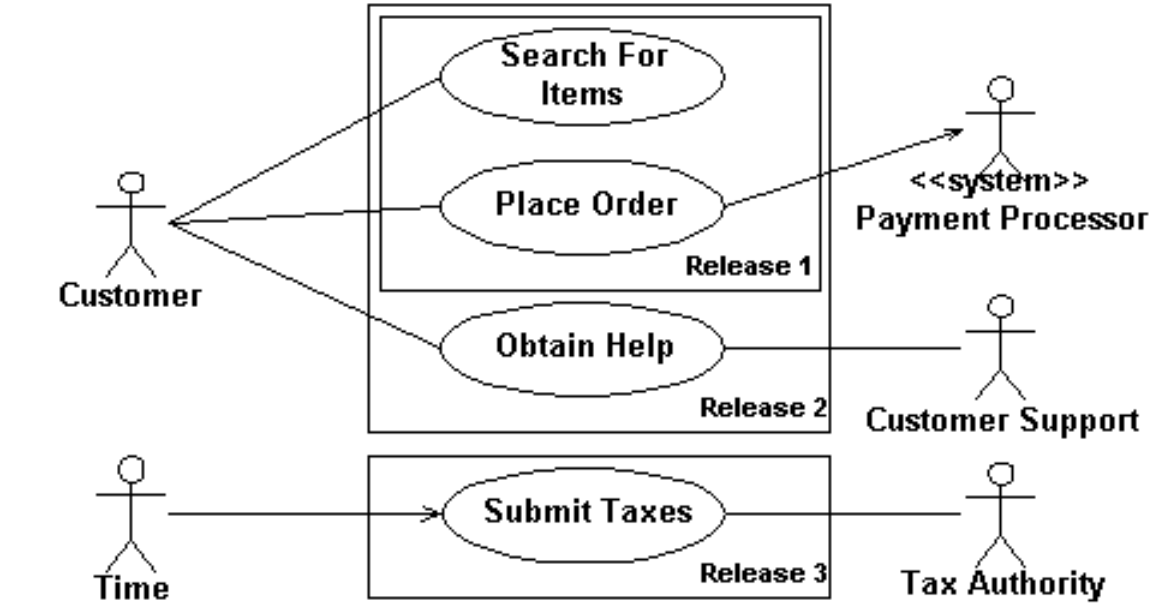
- Dibujar los actores principales arriba y a la izquierda. Ubicar los actores en la parte externa del diagrama.
- Los actores modelan roles, no puestos.
- Usar el actor “Tiempo” para iniciar eventos periódicos.
- Aplicar <<incluye>> cuando un caso de uso invoque a otro.
- Aplicar <<extiende>> cuando un caso de uso sea una variación de otro.

Diagramas De Casos De Uso

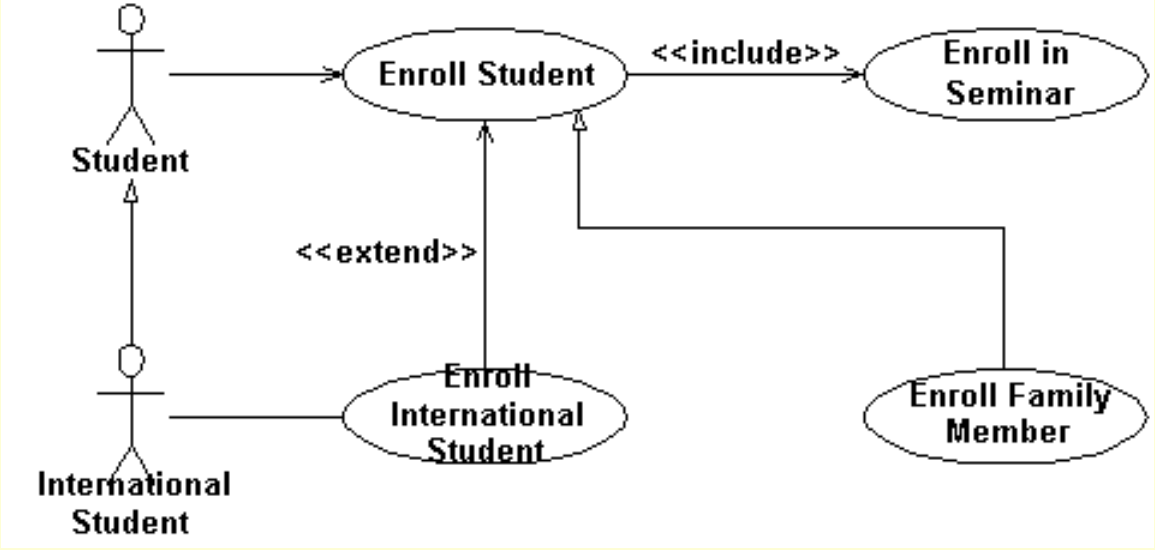
- Generalizar casos de uso cuando haya claramente nueva lógica de negocio.
- Evitar más de dos niveles de asociación entre casos de uso.
- Casos de uso “incluidos”: A la izquierda del invocador.
- Aplicar la regla “es como” para la generalización.

Diagramas De Casos De Uso

- Ubicar el actor que hereda bajo el actor padre.



Diagramas De Casos De Uso

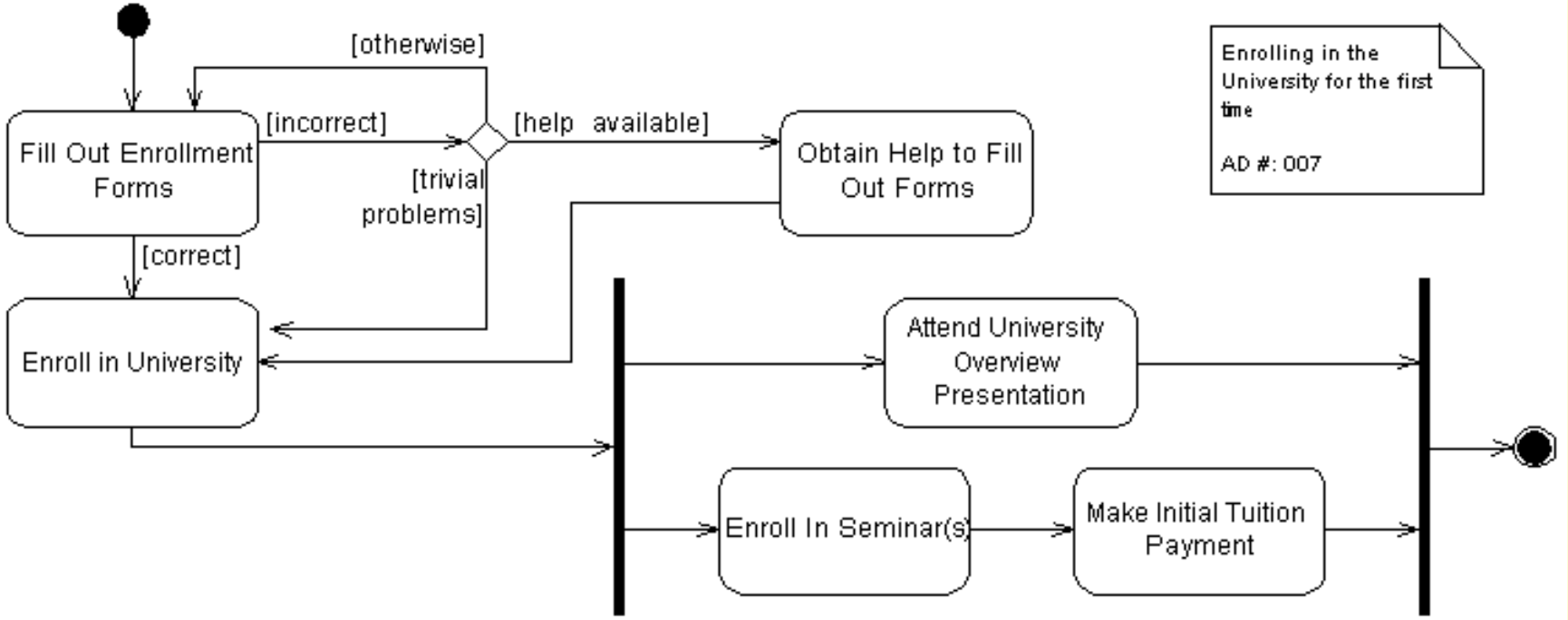


Diagramas De Actividad

- Colocar el punto de inicio arriba y a la izquierda. Siempre incluir un punto de finalización.
- Actividades “agujero negro”: transiciones entrantes pero no salientes.
- Actividades “milagro”: transiciones salientes pero no entrantes.

Diagramas De Actividad

- Evitar puntos de decisión superfluos.



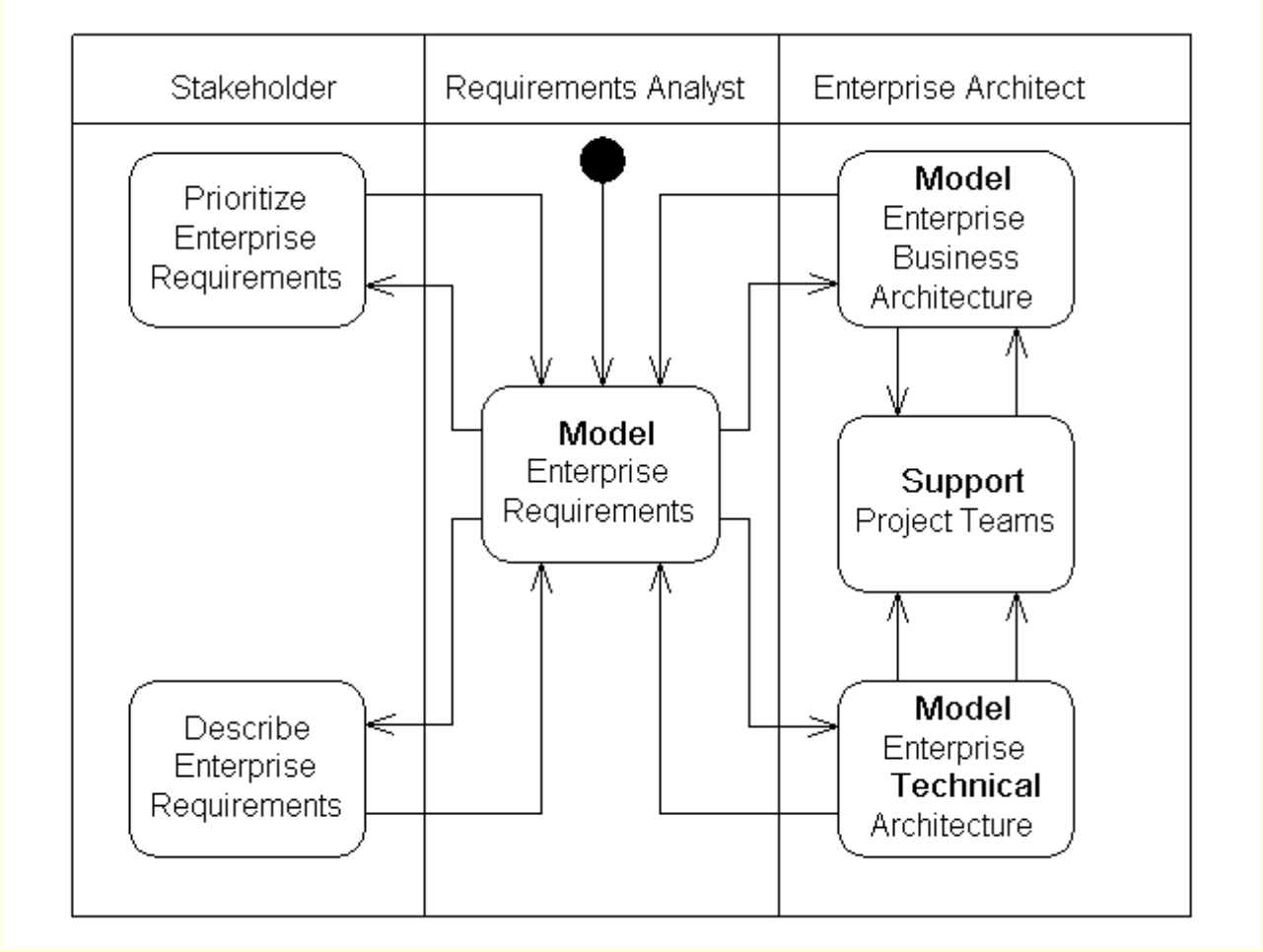
Diagramas De Actividad

- Toda transición saliente de un punto de decisión debe tener una condición.
- Las condiciones no deben solaparse.
- Las condiciones deben formar un conjunto completo.
- Toda bifurcación debe tener su unión correspondiente.

Diagramas De Actividad

- Las bifurcaciones sólo tienen una transición de entrada.
- Las uniones sólo tienen una transición de salida.
- Usar “swimlanes” para agrupar actividades realizadas por el mismo actor (como mucho cinco).

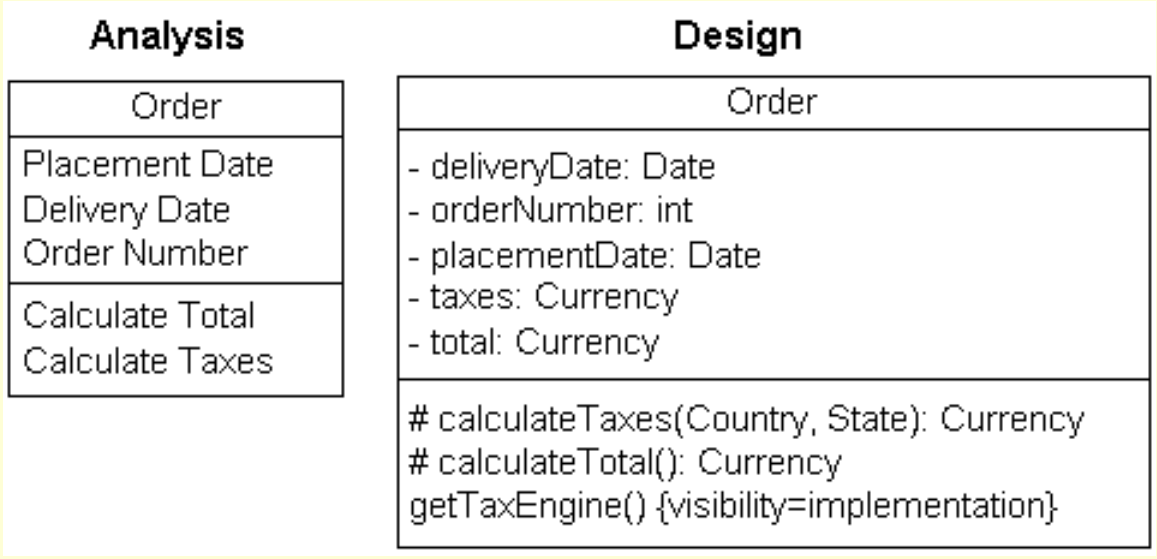
Diagramas De Actividad



Diagramas De Clases

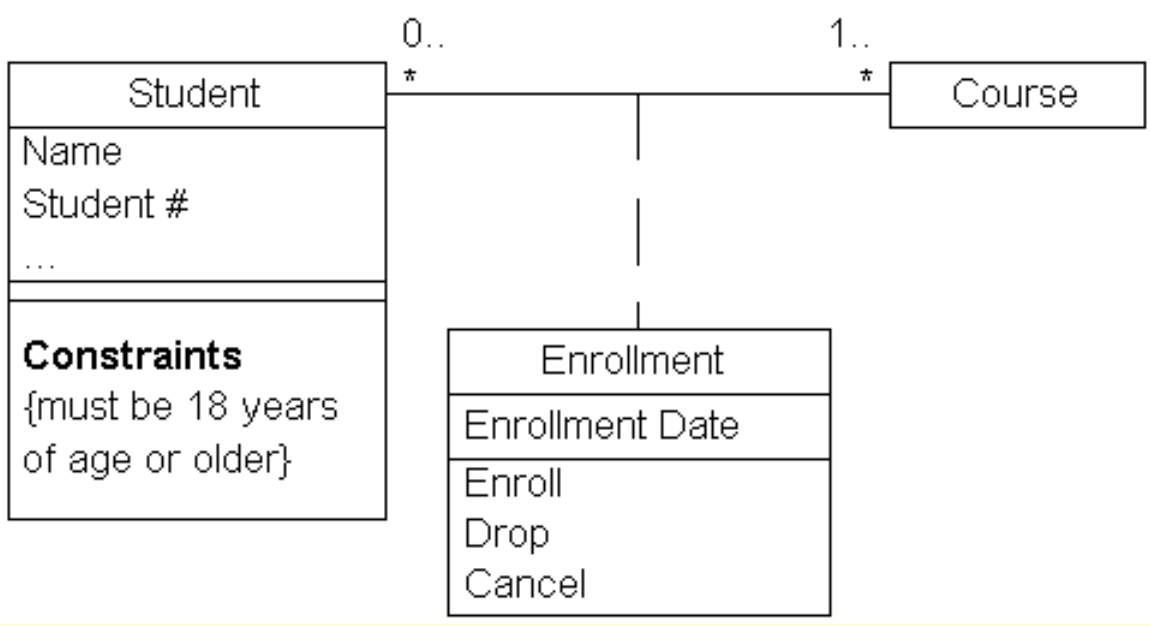
- Indicar visibilidad sólo en modelos de diseño.
- Indicar tipos en modelos de análisis sólo cuando sean un requerimiento.
- Reflejar la nomenclatura adoptada en la implementación. Seguir la convención de nombres adoptada.

Diagramas De Clases



Diagramas De Clases

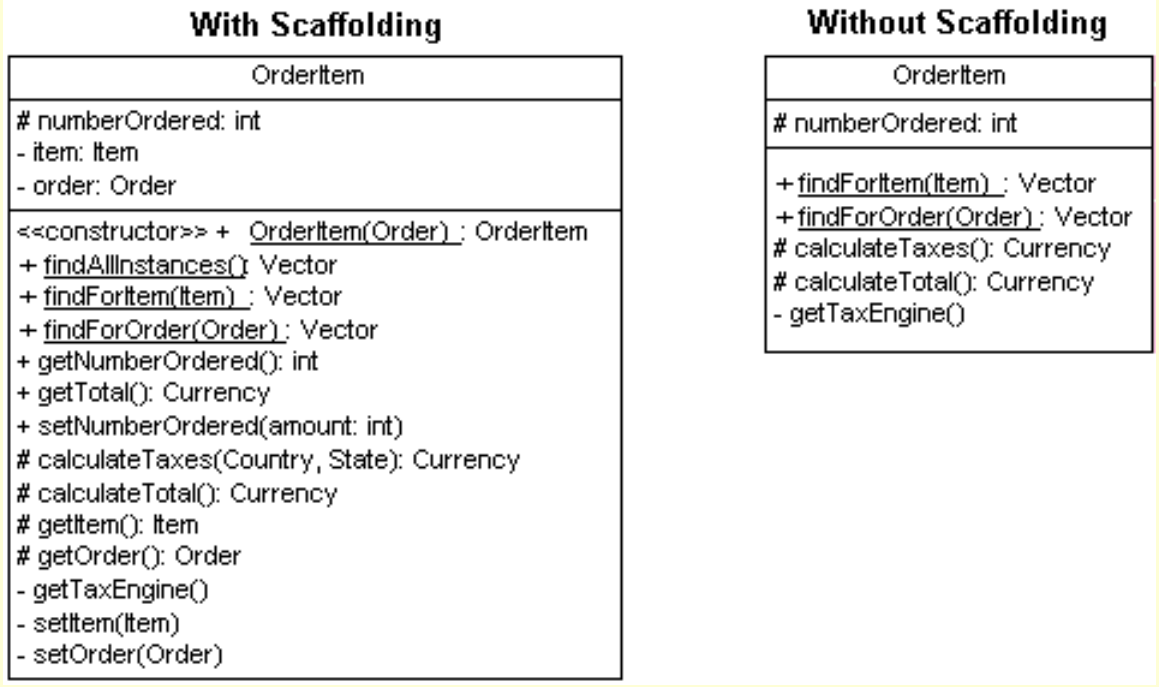
- Modelar clases de asociación en los diagramas de análisis



Diagramas De Clases

- No nombrar asociaciones que tienen clases de asociación.
- Usar nombres completos y en singular para nombrar las clases.
- Nombrar métodos y atributos con nombres autoexplicativos.
- No modelar en las clases los atributos o métodos de funcionalidad básica.Ej: código requerido para implementar relaciones con otras clases, set y get).

Diagramas De Clases



- Mostrar clases siempre con los tres compartimentos.

Diagramas De Clases

- Etiquetar los compartimentos de clase no comunes.
- Incluir (...) para indicar listas incompletas.
- Listar atributos/métodos estáticos primero.
- Listar atributos/métodos en orden de visibilidad decreciente.

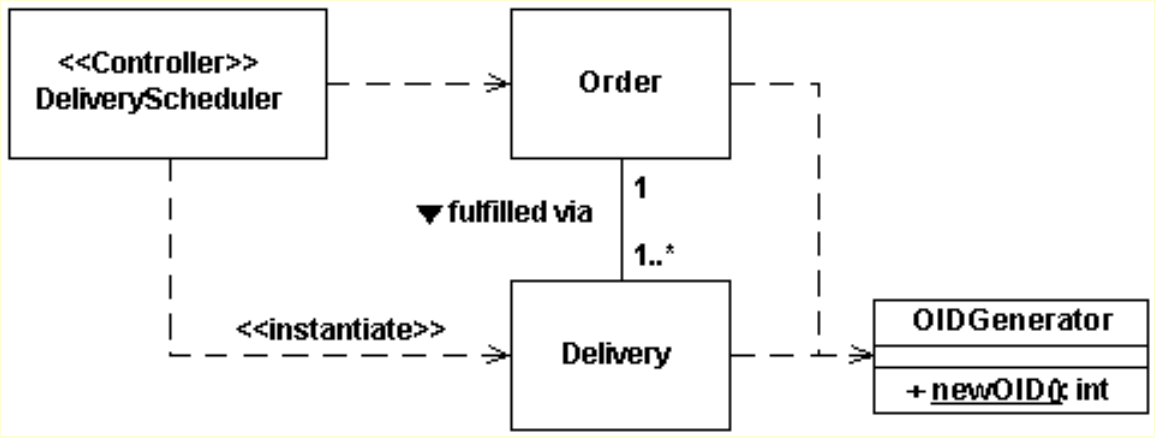
Diagramas De Clases

- En parámetros que son objetos, sólo listar el nombre.
- Indicar excepciones con una etiqueta de propiedad de operación.

```
+ findAllInstances(): Vector
{exceptions=NetworkFailure, DatabaseError}
```

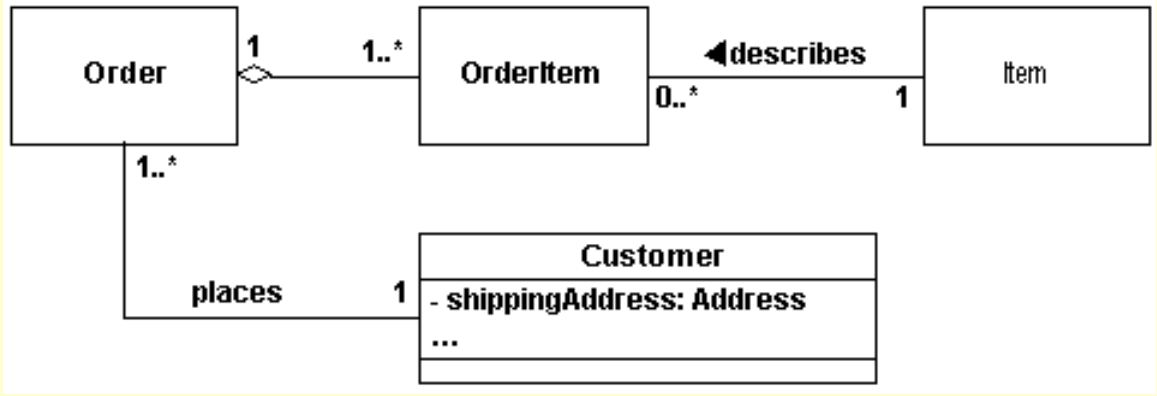
Diagramas De Clases

- Tratar de modelar las relaciones de forma horizontal, excepto la herencia.



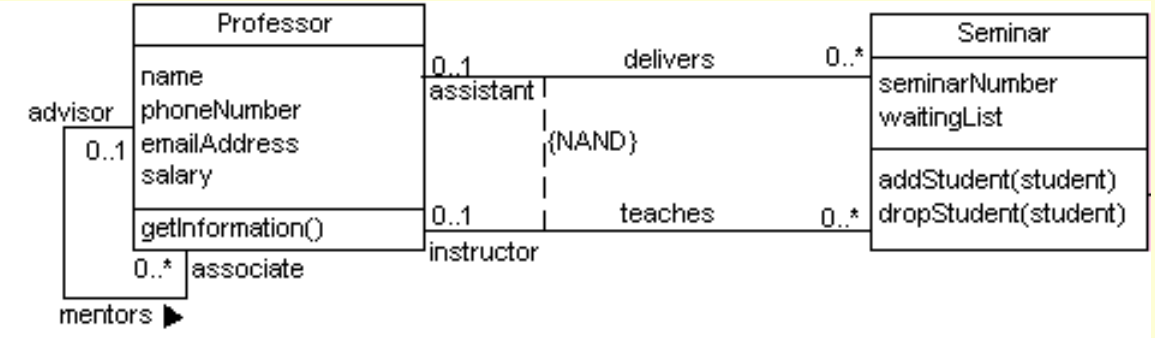
Diagramas De Clases

- Unir relaciones similares con una clase común.
- Indicar siempre la multiplicidad.



Diagramas De Clases

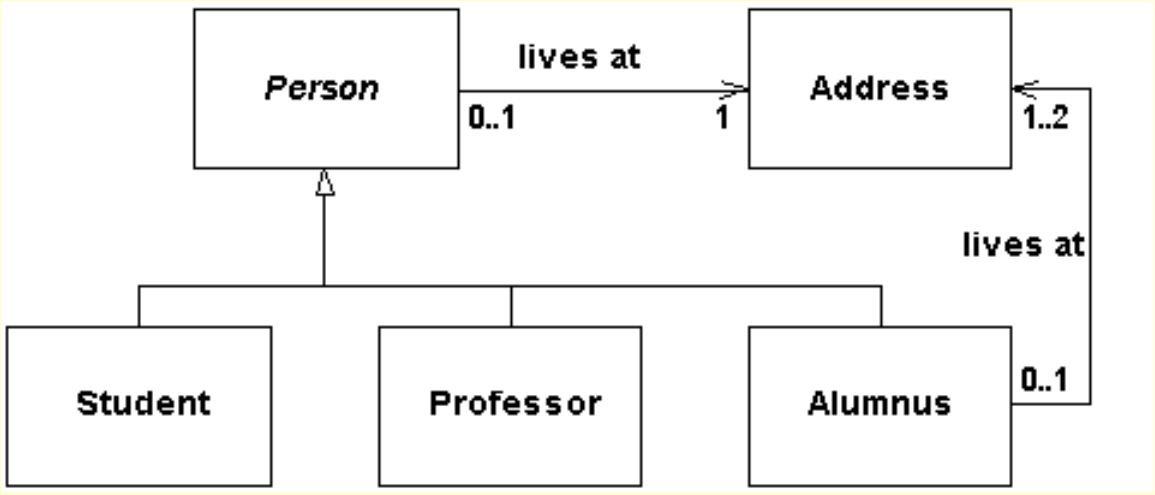
- Elimina relaciones indicando tipos en atributos.
- Indicar direccionalidad para clarificar un nombre de asociación.



Diagramas De Clases

- Indicar roles cuando haya varias asociaciones entre las mismas clases.
- Indicar roles en asociaciones recursivas.
- Dibujar asociaciones heredadas sólo cuando algo cambie.

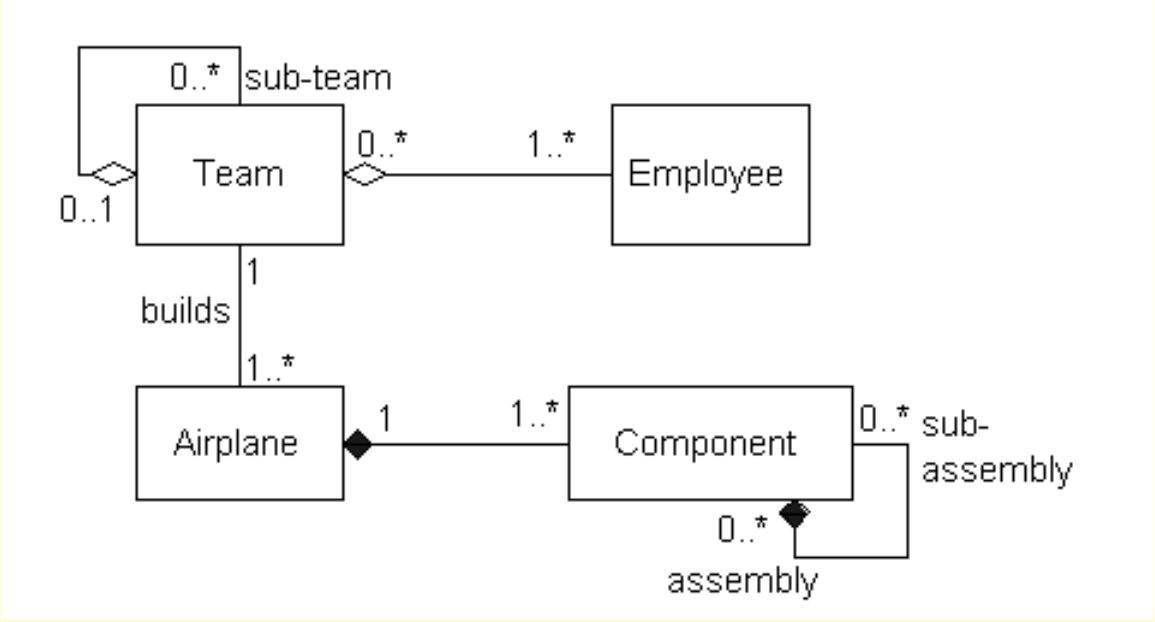
Diagramas De Clases



- Regla para indicar herencia: “la subclase A ES UNA superclase B” o “la subclase A ES COMO una superclase B”.
- Colocar subclases bajo superclases.

Diagramas De Clases

- Las agregaciones representan relaciones “es parte de”.

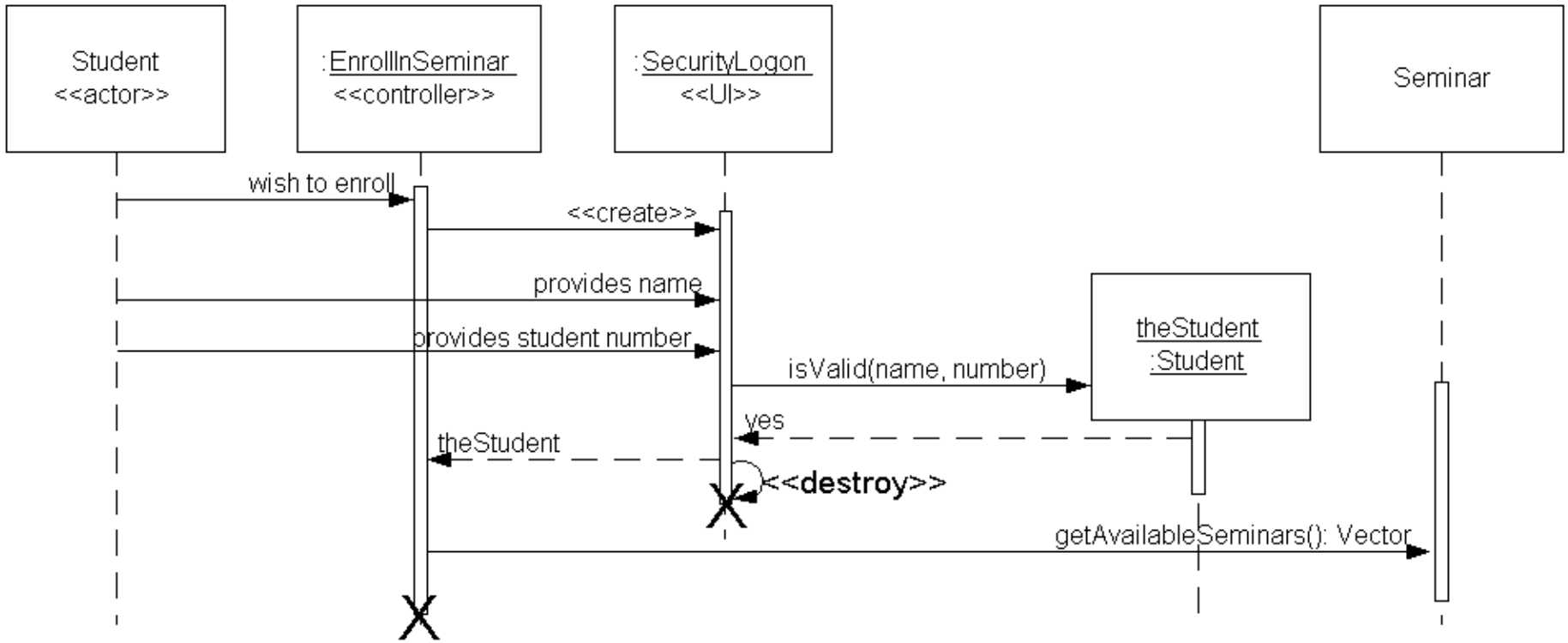


Diagramas De Clases

- La composición es una agregación fuerte en la cual el tiempo de vida del total y de las partes coinciden.
- Aplicar composiciones a agregaciones de objetos físicos.
- Dibujar el total a la izquierda de las partes.

Diagramas De Secuencia

- Divide en capas los clasificadores.

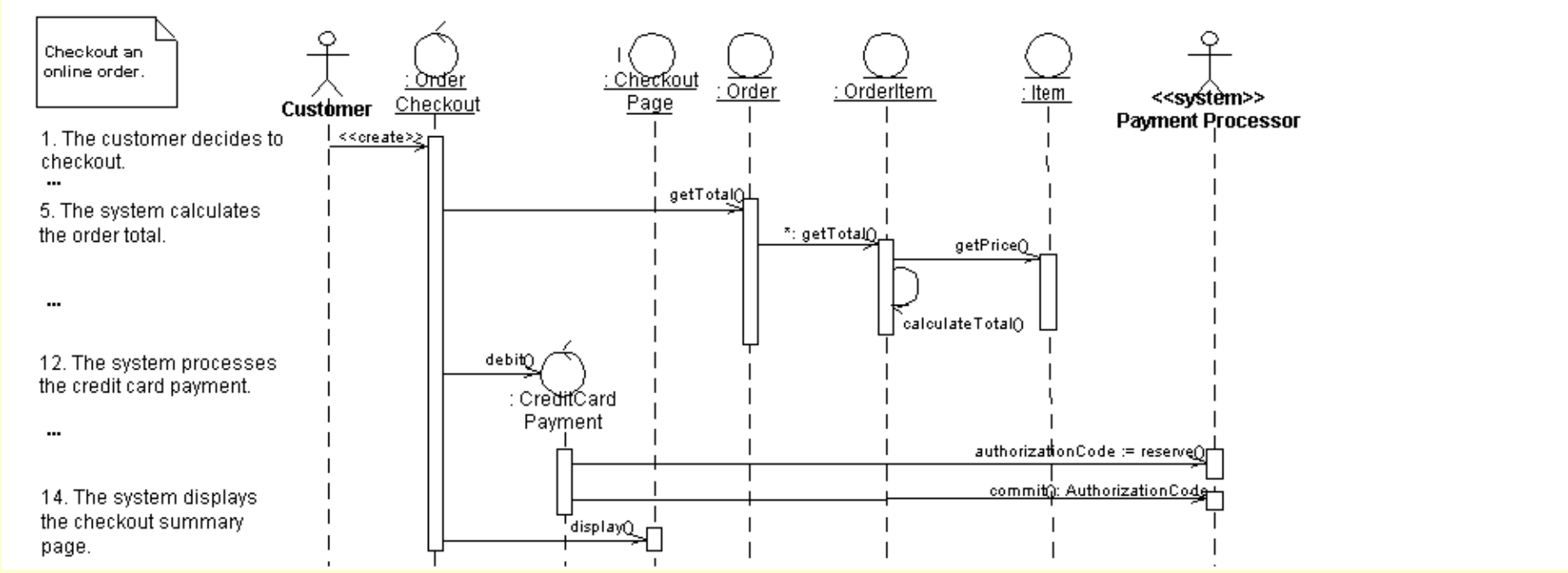


Diagramas De Secuencia

- Nombra los actores de forma consistente con los diagramas de casos de uso.
- Nombra las clases de forma consistente con los diagramas de clase.
- Un actor puede tener el mismo nombre que una clase.

Diagramas De Secuencia

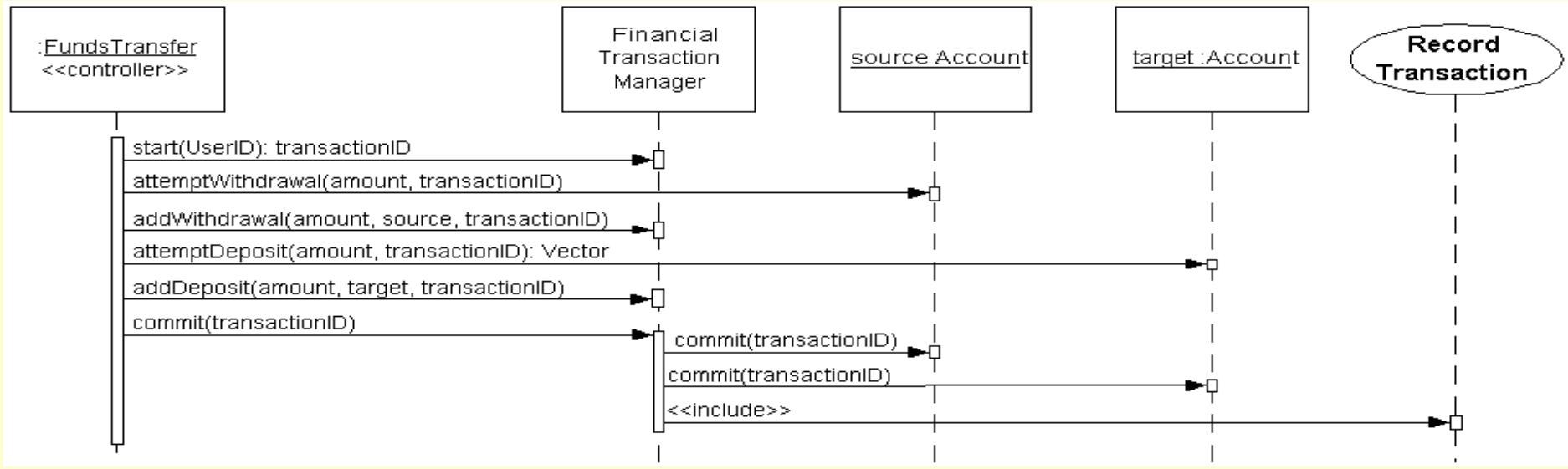
- Es conveniente incluir una descripción de la lógica seguida.



Diagramas De Secuencia

- Evitar modelar la destrucción de objetos.
- Nombrar los objetos cuando sean referidos en mensajes.
- Nombrar objetos cuando existan varios del mismo tipo.

Diagramas De Secuencia



- Aplicar estereotipos consistentemente.
- Mostrar la creación de los objetos de forma directa.

Diagramas De Secuencia

- Usar descripciones para mensajes implicando actores humanos u organizacionales.
- Optar por nombres en vez de tipos en los parámetros. Ej:
  hacerDeposito(cantidad,cuenta,IDtransaccion) en vez de hacerDeposito(Moneda,Cuenta,int)

Diagramas De Secuencia

- Los mensajes a las clases se implementarán como métodos estáticos. Consistencia con diagramas de clases.
- No modelar un valor de retorno cuando éste sea obvio.
- Modelar valores de retorno como parte de una invocación de método.