

# Diseño

# Análisis vs Diseño

- ▶ En modelo de Cascada tradicional luego de levantamiento de requisitos viene una etapa de análisis
- ▶ La etapa de diseño vendría después de concluir la etapa de análisis
- ▶ Bajo enfoque ágil análisis y diseño no tienen una separación tan nítida
- ▶ El paso de análisis a diseño es suave y casi imperceptible
- ▶ En ambos casos lo que se busca es construir un modelo
- ▶ modelo de análisis se parece mucho a un modelo de datos
- ▶ modelo de diseño incluye aspectos relacionados con el "como"

# Modelo de Dominio (Análisis)

- ▶ Identificación de las clases de dominio
- ▶ Nivel de detalle en clases es mucho menor
- ▶ Por lo general se comienza con atributos (sin métodos)
- ▶ Modelo incluye clases de dominio y no objetos relacionados con implementación
- ▶ Se parece mucho a un modelo E-R pero pueden aparecer clases no persistentes e incluso algunos métodos

# Ejemplo Modelo de Dominio

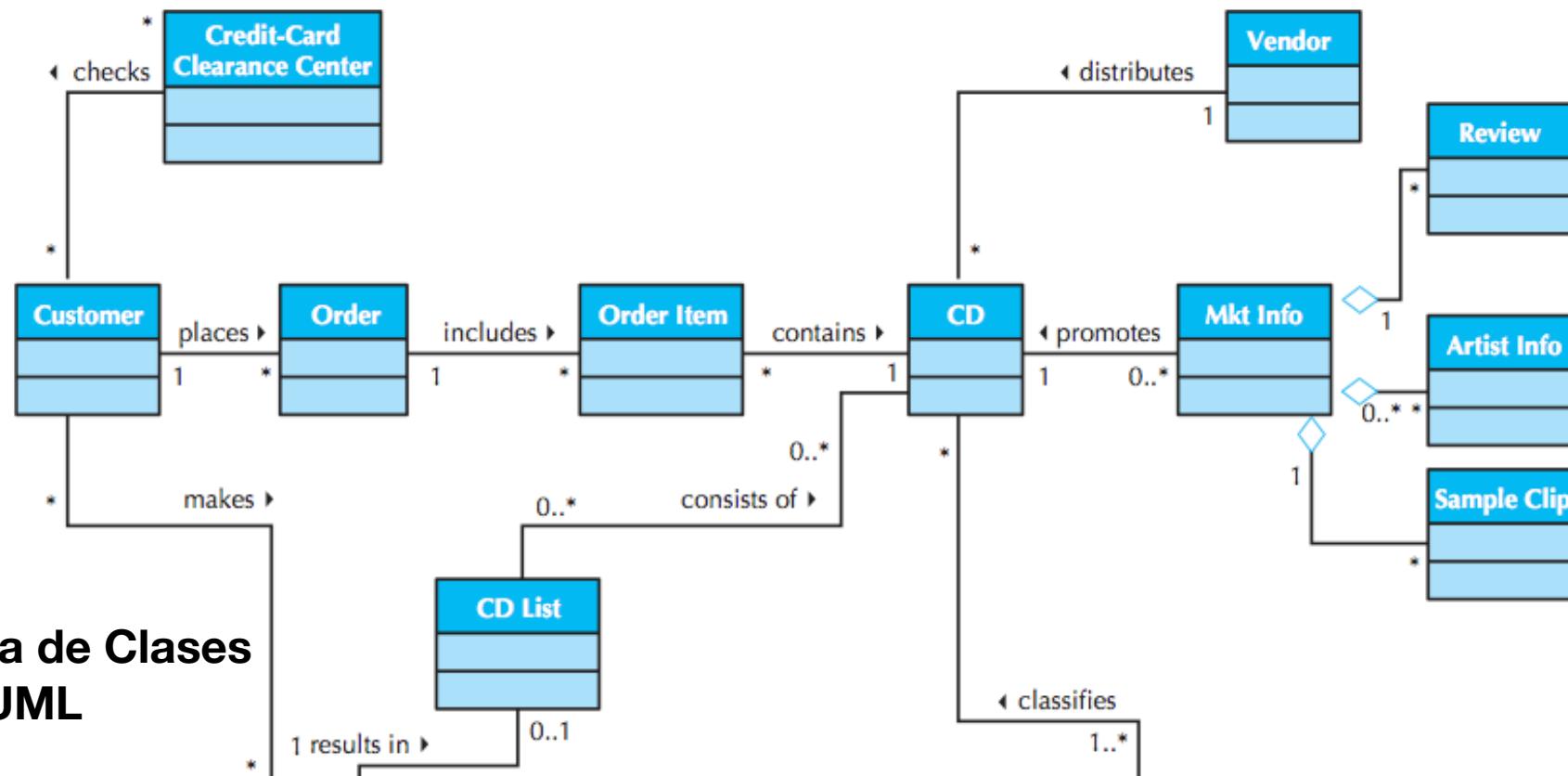
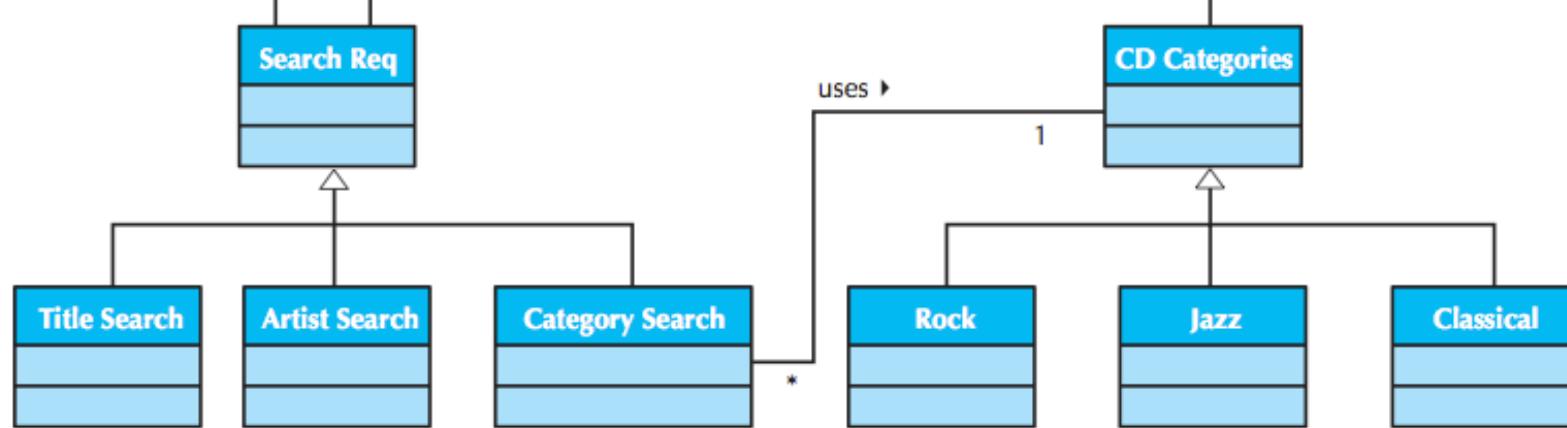


Diagrama de Clases  
UML



# Descubrimiento de Objetos

- ▶ Buscar candidatos en
    - ▶ descripción de requisitos
    - ▶ casos de uso
    - ▶ relatos de usuario
  - ▶ Buscar adicionalmente en
    - ▶ espacio general del problema
    - ▶ dispositivos físicos con los que interactúa
    - ▶ roles
    - ▶ eventos que deben ser registrados
    - ▶ ubicaciones geográficas
    - ▶ unidades organizacionales
- Algunos son objetos candidatos
  - Algunos son solo atributos

# Análisis del texto

- A common or improper noun implies a class of objects.
- A proper noun or direct reference implies an instance of a class.
- A collective noun implies a class of objects made up of groups of instances of another class.
- An adjective implies an attribute of an object.
- A doing verb implies an operation.
- A being verb implies a classification relationship between an object and its class.
- A having verb implies an aggregation or association relationship.
- A transitive verb implies an operation.
- An intransitive verb implies an exception.
- A predicate or descriptive verb phrase implies an operation.
- An adverb implies an attribute of a relationship or an operation.

Source: These guidelines are based on Russell J. Abbott, "Program Design by Informal English Descriptions," *Communications of the ACM* 26, no. 11 (1983): 882–894; Peter P-S Chen, "English Sentence Structure and Entity-Relationship Diagrams," *Information Sciences: An International Journal* 29, no. 2–3 (1983): 127–149; and Graham, *Migrating to Object Technology*.

Alice and the other **student evaluated the instructor**.

A stack **is a structure** where you can **push** or **pop elements** in a lifo manner.

This implementation of a stack **has an array** and an index

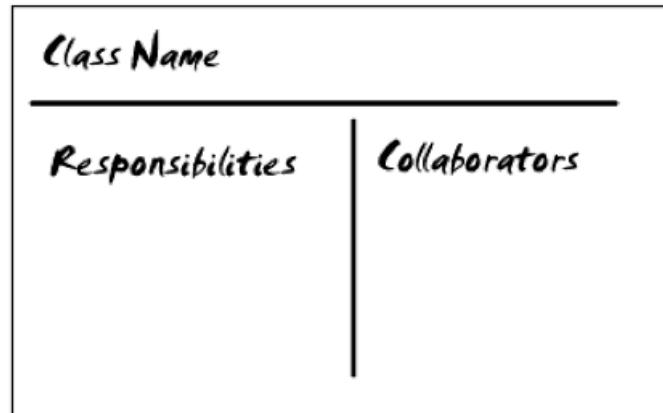
# Evaluar Candidatos

- ▶ cada objeto debe guardar algún dato
- ▶ más de 1 atributo (o de lo contrario atributo)
- ▶ tiene métodos ?
- ▶ todas las instancias mismos atributos
- ▶ todas las instancias mismos métodos

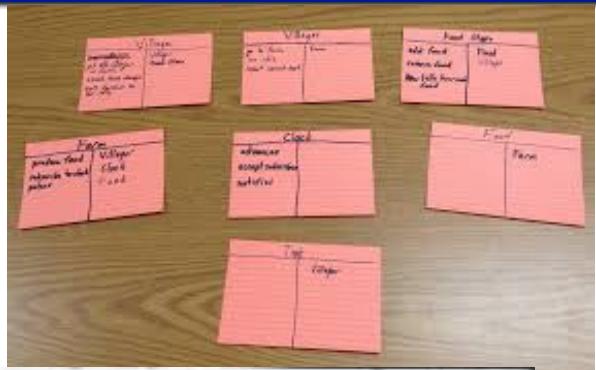
# Responsabilidades y Colaboradores

- ▶ Responsabilidades
  - ▶ qué es lo que el objeto sabe
  - ▶ qué es lo que el objeto es capaz de hacer
- ▶ Colaboradores
  - ▶ otras clases que participan

# Uso de Tarjetas CRC



- ▶ permiten explorar las interacciones entre objetos
- ▶ CRC clase - responsabilidad - colaborador
- ▶ responsabilidades - que es lo que sabe y que hace con esa información
- ▶ colaboradores - clases que colaboran para lograr el propósito

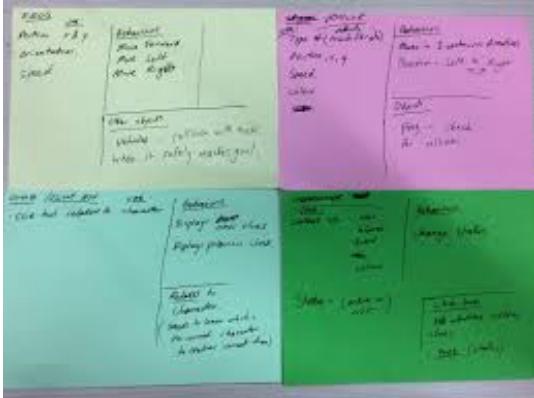


## Drawing

Holds Figures.

Accumulates updates,  
refreshes on demand.

Figure  
Drawing View  
Drawing Controller

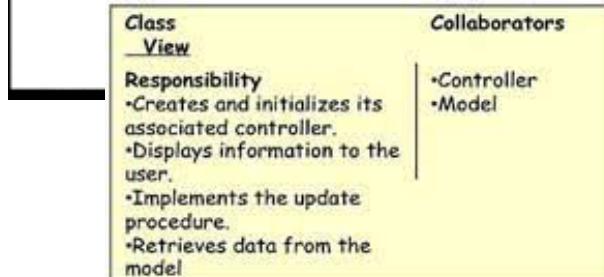


## Class: AbstractWidgetManagerFactory

### Responsibilities

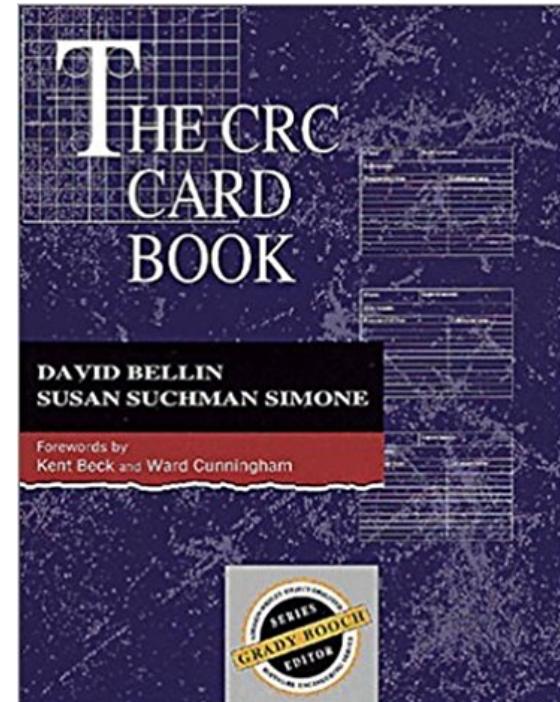
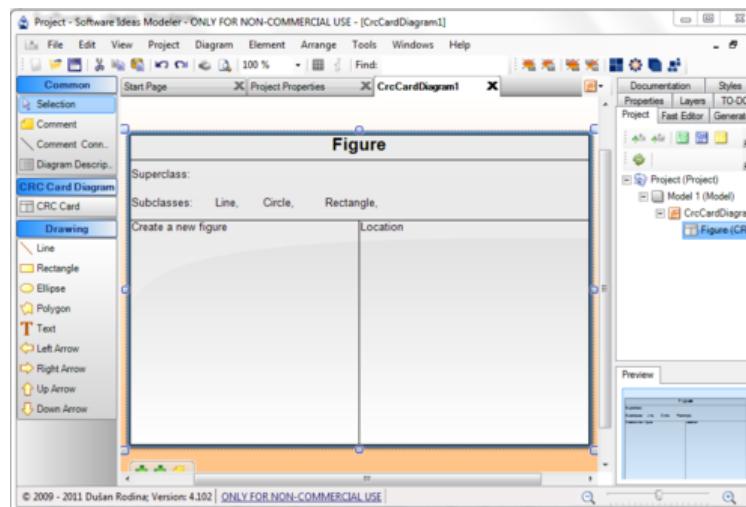
### Collaborators

- Widget
- AbstractWidget
- Factory Manager

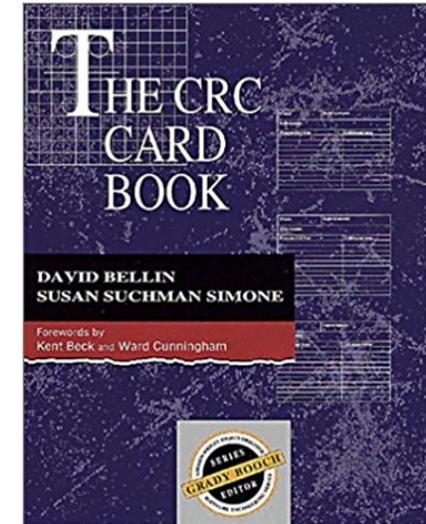


Class CardReader	
Responsibilities	Collaborators
Tell ATM when card is inserted	ATM
Read information from card	Card
Eject card	
Retain card	

CRC Card	
Class:	StudentRecord
Responsibilities:	Collaborators:
contains description of student	Description
has a collection of Assignments	Assignment
changes grades and Curve Status for assignments when asked by Gradebook	GradeBook, Assignment



- ▶ el mayor valor es hacer pensar a los desarrolladores en términos de objetos
- ▶ no más de 4 a 5 participantes
- ▶ informal
- ▶ foco en un escenario
- ▶ se suele descubrir la necesidad de otras clases, agregar responsabilidades o ajustar algunas

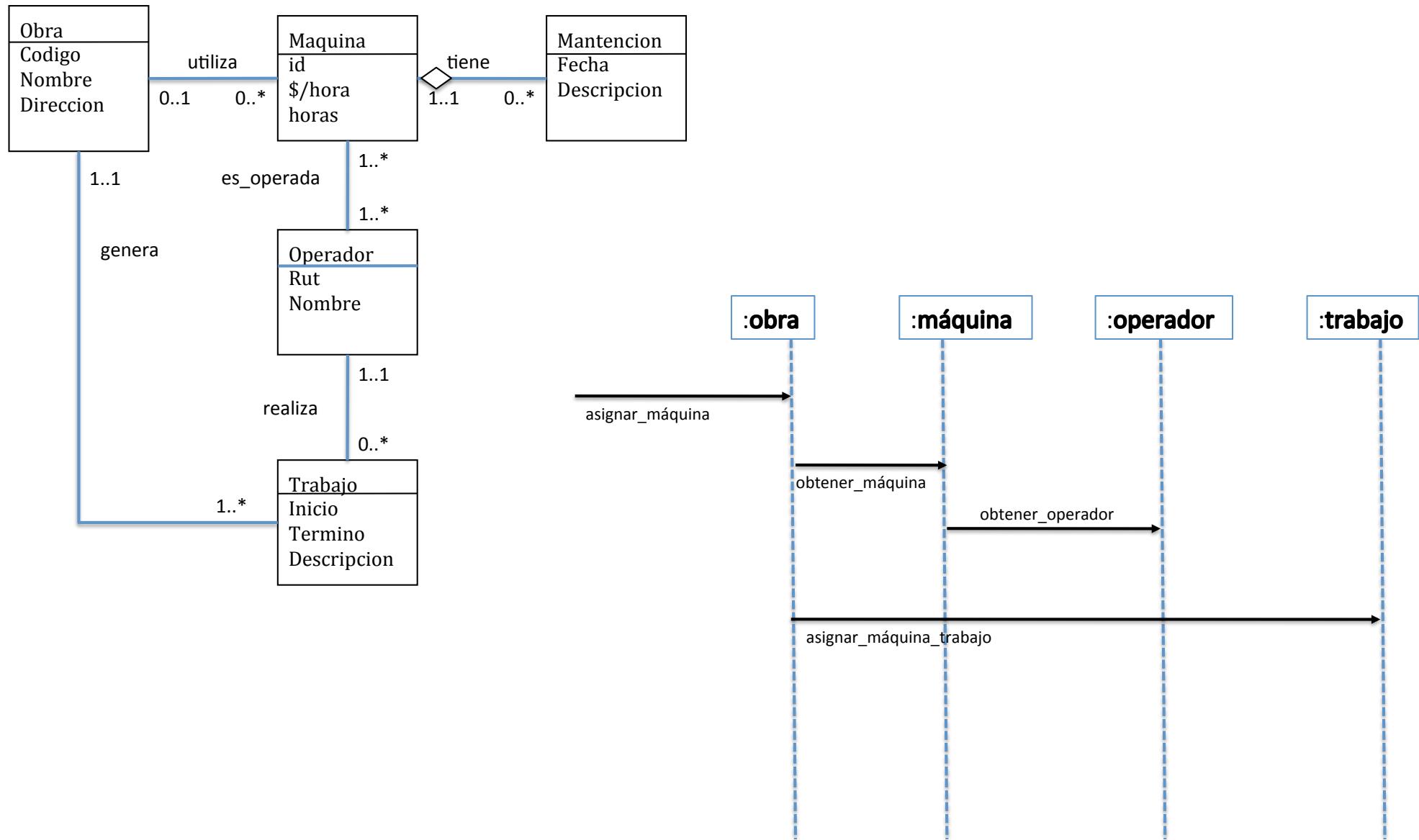


<http://www.math-CS.gordon.edu/courses/cs211/ATMExample/>

# Ejemplo: Empresa Constructora

Una empresa constructora que mantiene normalmente entre 4 y 8 obras en ejecución simultáneamente maneja un pool de **máquinas** comunes que son usadas en las distintas **obras**. Cada una de las máquinas tiene un nombre y un identificador además de otra información asociada: características, costo de operación por hora, horas de uso, horas desde última mantención, etc. Cada una de estas máquinas puede ser operada por un cierto número de **operadores** que la empresa emplea en una modalidad por hora. Cada fin de semana se debe pagar a cada operador de acuerdo al trabajo realizado para lo cual se emite un detalle de cada **trabajo** realizado en este lapso (obra, fecha, hora inicio, hora término, máquina asociada). Para cada máquina se mantiene una historia con el detalle de las **mantenciones** periódicas que se le han realizado.

# Modelo de Dominio



# Ejemplo: Reserva Aviones

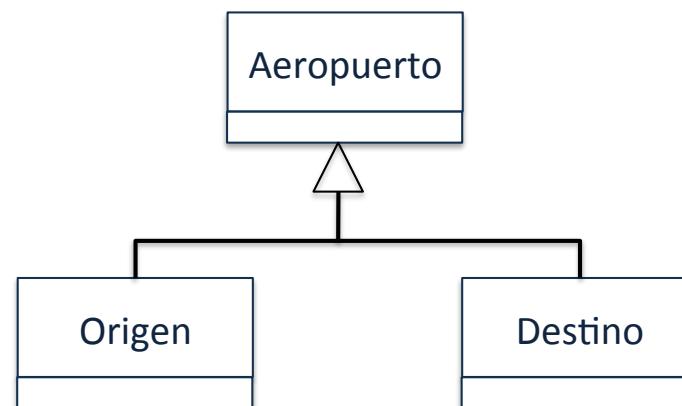
Considera un sistema de reserva de vuelos para una agencia de viajes, con las siguientes características. Las líneas aéreas ofrecen varios vuelos. Un vuelo es abierto para hacer reservas y nuevamente cerrado por orden de la compañía. Un cliente puede reservar uno o más vuelos y para diferentes pasajeros. Una reserva se refiere a un único vuelo y a un único pasajero, y puede ser cancelada o confirmada. Los vuelos tienen aeropuerto, fecha y hora de salida y de llegada, y pueden incluir escalas en aeropuertos; cada escala tiene una hora de llegada y una hora de salida. Cada aeropuerto atiende a una o más ciudades.

# Clases

- ▶ aerolíneas
- ▶ vuelos
- ▶ reservas
- ▶ aeropuertos
- ▶ pasajeros
- ▶ clientes
- ▶ ciudades

# Opciones

- ▶ Aeropuertos de origen y destino como subclases



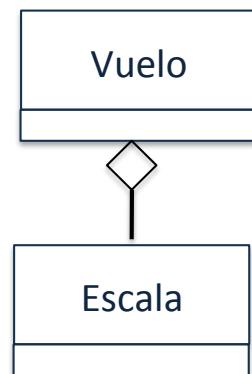
# Opciones

- ▶ Escala como subclase de aeropuerto



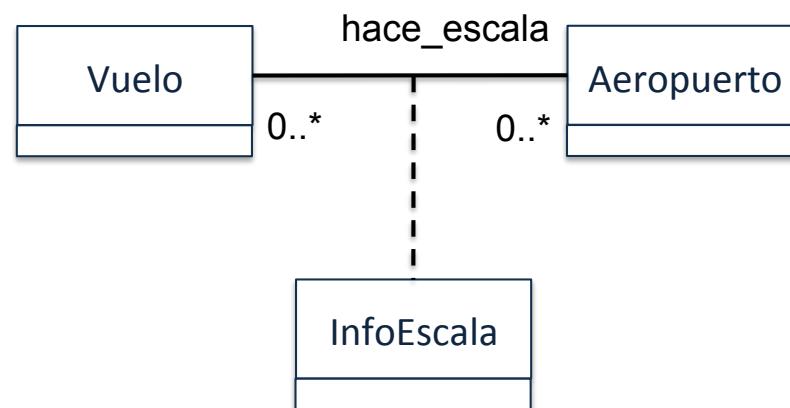
# Opciones

- ▶ Vuelo contiene número de escalas
- ▶ escalas pueden compartirse



# Opciones

- ▶ Escala como clase asociativa



# Modelo Elegido

