

PREGRADO



UNIDAD 3 | HIGH-LEVEL SOFTWARE DESIGN & PATTERNS

ENTERPRISE APPLICATION DESIGN

SI720 | Diseño y Patrones de Software



Al finalizar la unidad, el estudiante elabora y comunica artefactos de diseño de software aplicando principios básicos y patrones de diseño para un dominio y contexto determinados.

AGENDA

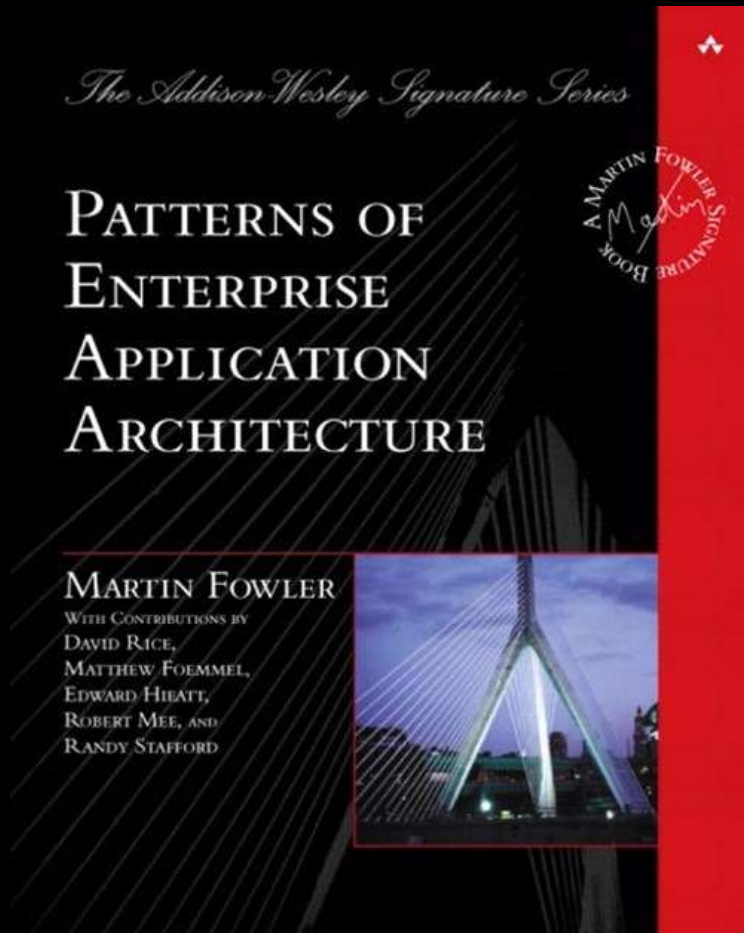
INTRO

ENTERPRISE SOFTWARE DESIGN



Software Design Patterns

2002



Martin Fowler
2002



Enterprise App Design Patterns

2002

Domain Logic

Data Source
Architectural
Patterns

Object-Relational
Behavioral

Object-Relational
Structural

Object-Relational Metadata
Mapping

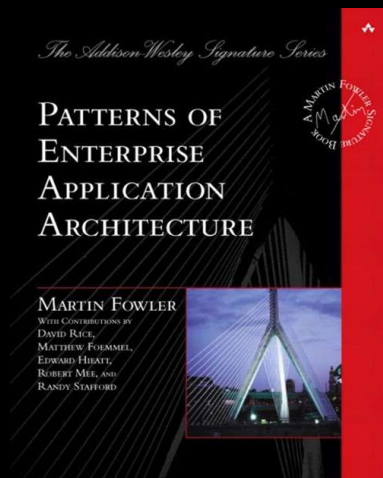
Transaction
Script
Domain Model
Table Module
Service Layer

Table Data Gateway
Row Data Gateway
Active Record
Data Mapper

Unit of Work
Identity Map
Lazy Load

Identity Field
Foreign Key
Mapping
Association Table
Mapping
Dependent
Mapping
Embedded Value
Serialized LOB
Single Table
Inheritance
Class Table
Inheritance
Concrete Table
Inheritance
Inheritance
Mappers

Metadata Mapping
Query Object
Repository



Enterprise App Design Patterns

2002

Web Presentation

Model View
Controller
Page Controller
Front Controller
Template View
Transform View
Two-Step View
Application Controller

Distribution

Remote Façade
Data Transfer Object

Offline Concurrency

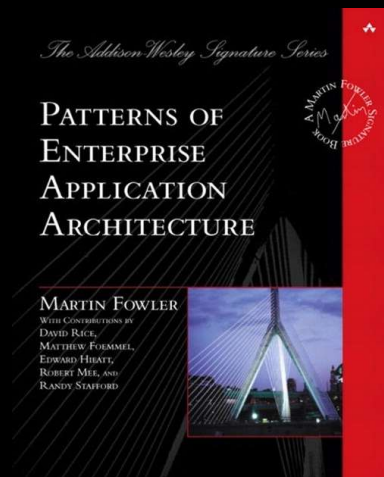
Optimistic Offline
Lock
Pessimistic Offline
Lock
Coarse Grained Lock
Implicit Lock

Session State

Client Session
State
Server Session
State
Database Session
State

Base

Gateway
Mapper
Layer Supertype
Separated Interface
Registry
Value Object
Money
Special Case
Plugin
Service Stub
Record Set



AGENDA

INTRO

ENTERPRISE SOFTWARE DESIGN

REST DESIGN PATTERNS



AGENDA

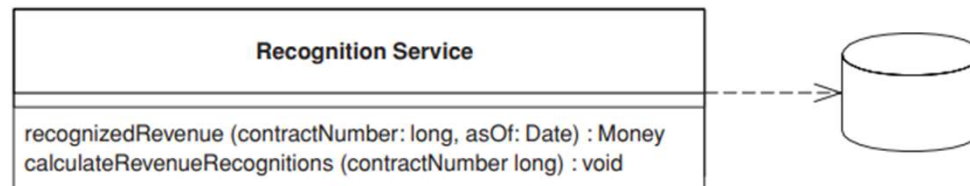
Intro

Patterns of Enterprise Application Architecture

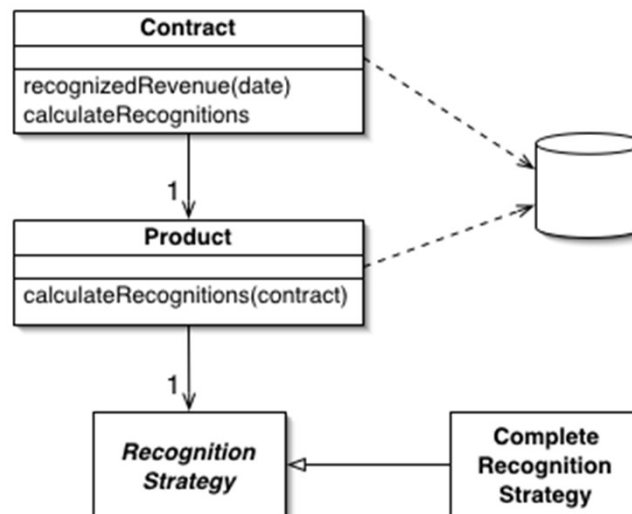
REST Architectural Design Pattern

Domain Logic Patterns

Transaction Script. Organiza la lógica principalmente como un único procedimiento, haciendo llamadas directamente o a través de un thin database wrapper. Cada transacción tendrá su propio Transaction Script, aunque subtareas comunes se pueden descomponer en subprocedimientos.

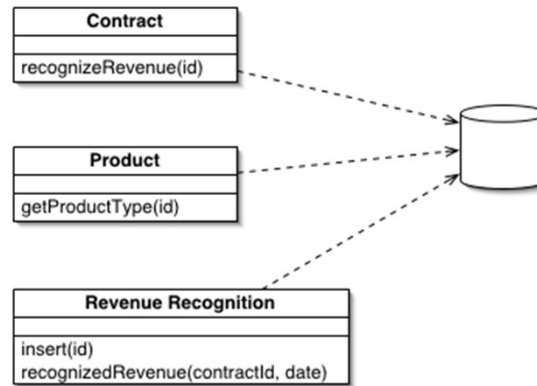


Domain Model. Aborda la complejidad formando una red de objetos que se interconectan, donde cada objeto representa una parte individual, que podría abarcar desde una compañía hasta una línea de un formato de orden.

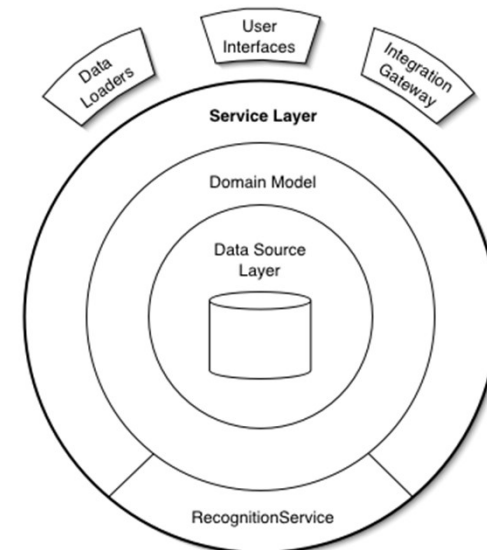


Domain Logic Patterns

Table Module. Organiza la domain logic con una clase por tabla en la base de datos y una sola instancia de la clase contiene los diversos procedimientos que actúan sobre los datos.

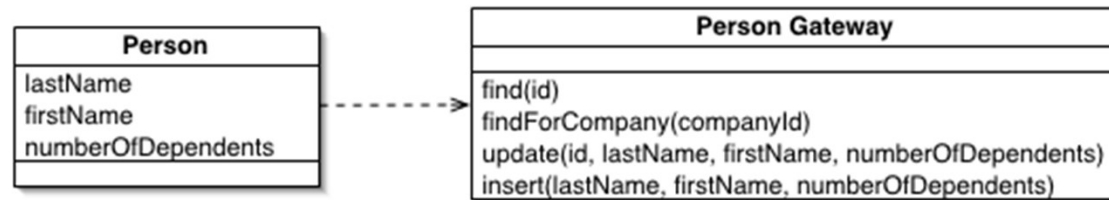


Service Layer. Define límites para una aplicación y forma un conjunto de operaciones disponibles desde la perspectiva de la interacción con las capas clientes. Encapsula la business logic, controlando las transacciones y coordinando respuestas en la implementación de operaciones.

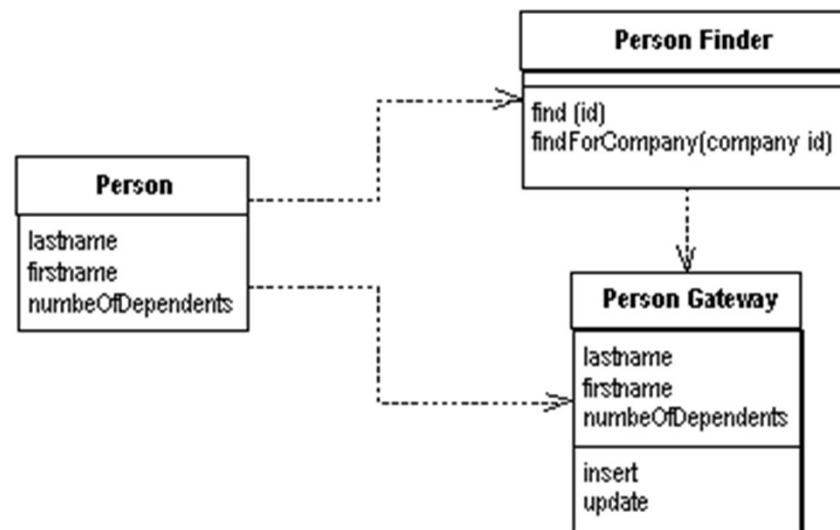


Data Source Architectural Patterns

Table Data Gateway. Un objeto que actúa como una puerta de enlace a una tabla de base de datos. Una instancia maneja todas las filas de la tabla.

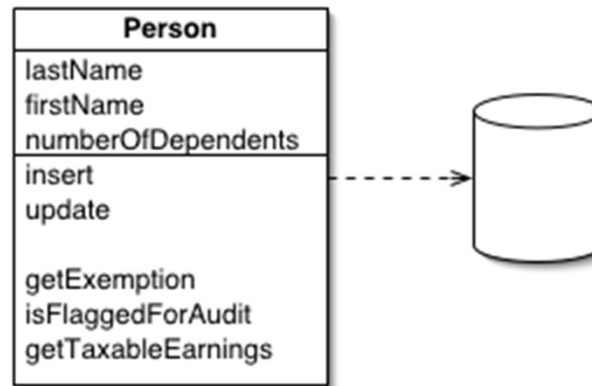


Row Data Gateway. Un objeto que actúa como una puerta de enlace a un único registro en una fuente de datos. Hay una instancia por fila.

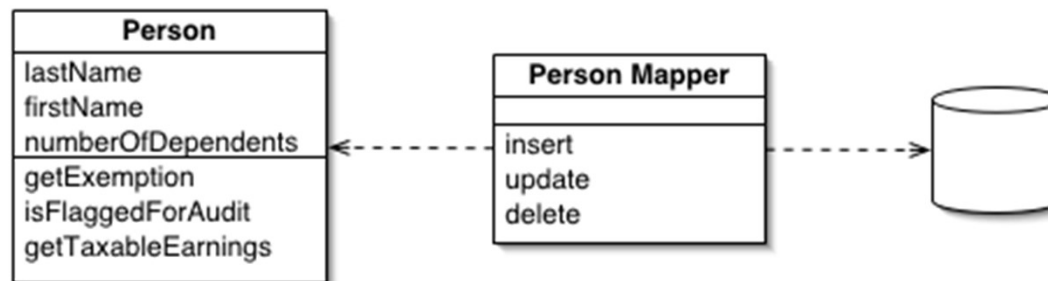


Data Source Architectural Patterns

Active Record. Un objeto que envuelve una fila en una tabla o vista de base de datos, encapsula el acceso a la base de datos y agrega lógica de dominio en esos datos..

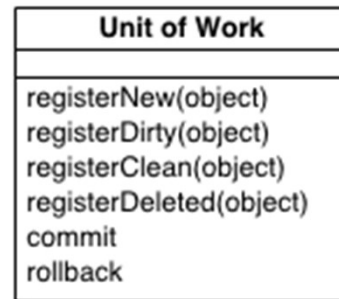


Data Mapper. Una capa de Mappers que mueve datos entre objetos y una base de datos mientras los mantiene independientes unos de otros y del mapeador.

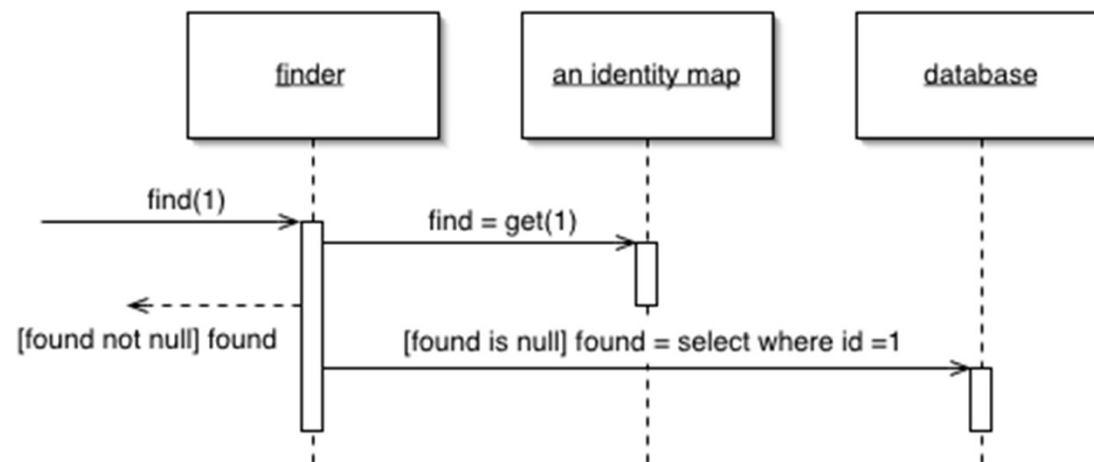


Object-Relational Behavioral Patterns

Unit of Work. Mantiene una lista de objetos afectados por una transacción comercial y coordina la redacción de los cambios y la resolución de problemas de concurrencia..

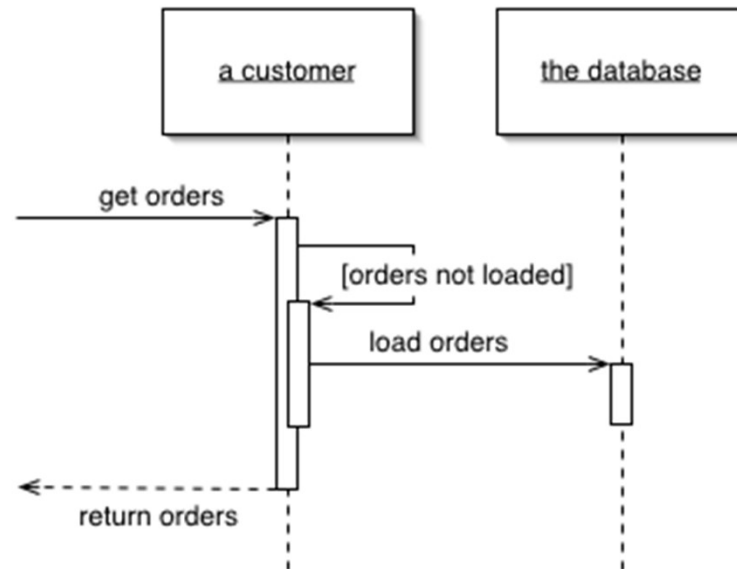


Identity Map. Asegura que cada objeto se cargue solo una vez al mantener cada objeto cargado en un mapa. Busca objetos usando el mapa cuando se refiere a ellos.



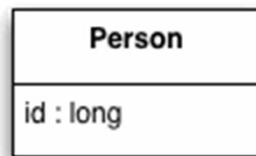
Object-Relational Behavioral Patterns

Lazy Load. Un objeto que no contiene todos los datos que necesita pero que sabe cómo obtenerlos..

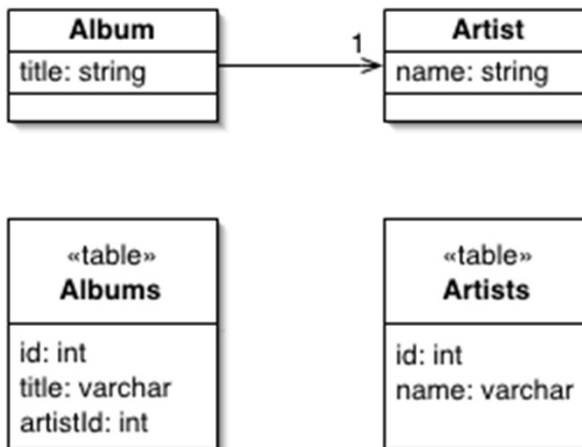


Object-Relational Structural Patterns

Identity Field. Guarda un campo de ID de base de datos en un objeto para mantener la identidad entre un objeto en memoria y una fila de la base de datos..

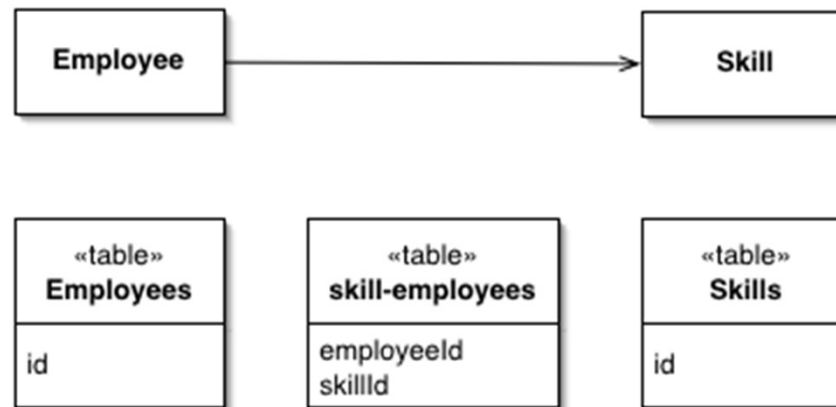


Foreign Key Mapping. Asigna una asociación entre objetos a una referencia de clave externa entre tablas..

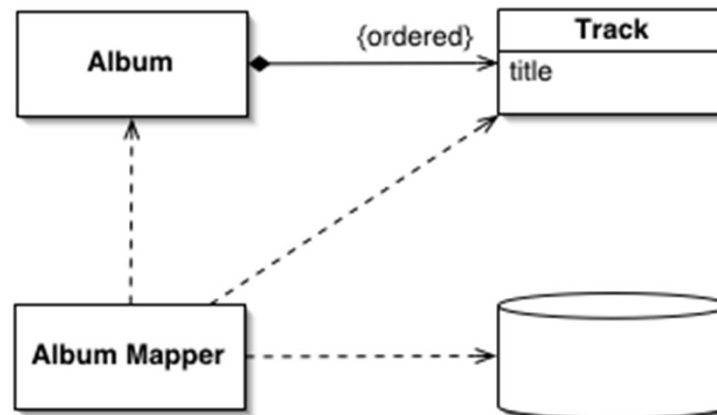


Object-Relational Structural Patterns

Association Table Mapping. Guarda una asociación como una tabla con claves foráneas en las tablas vinculadas por la asociación..

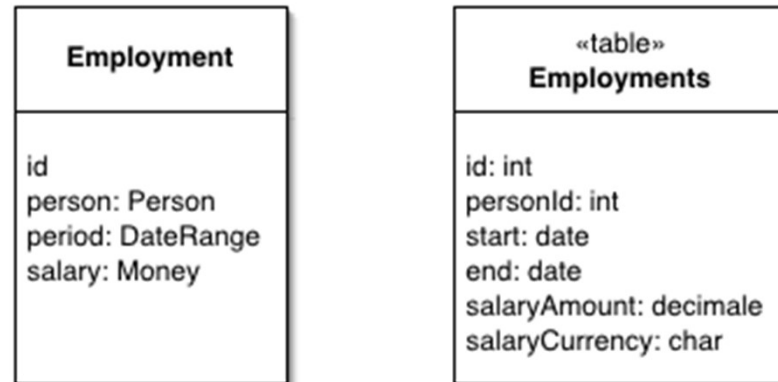


Dependent Mapping. Tiene una clase que realiza la asignación de la base de datos para una clase secundaria.

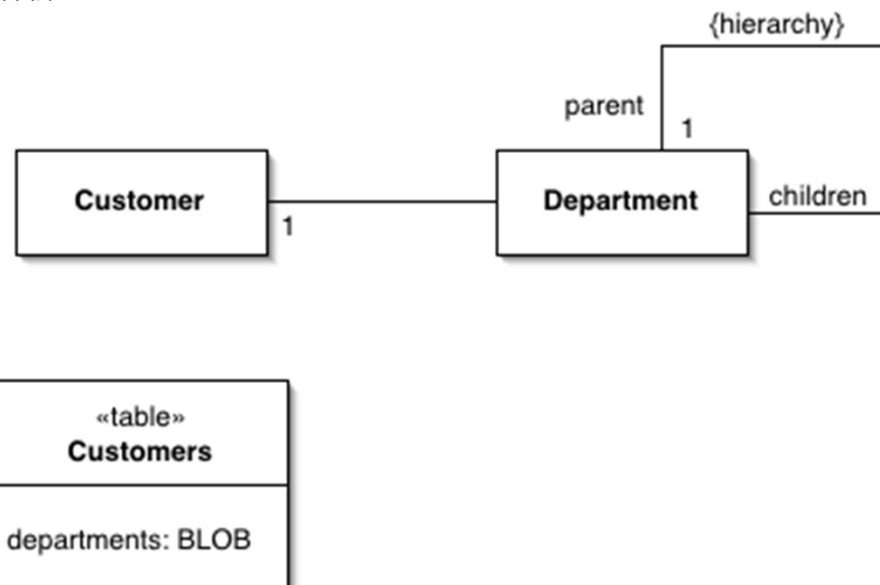


Object-Relational Structural Patterns

Embedded Value. Mapea un objeto en varios campos de la tabla de otro objeto..

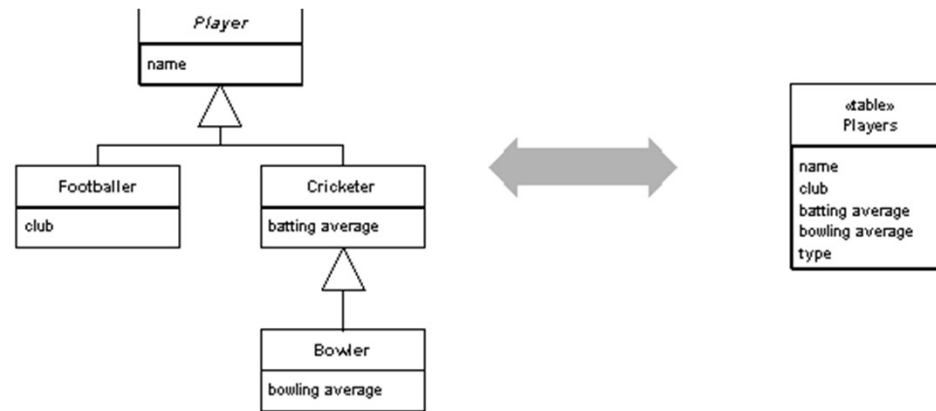


Serialized LOB. Guarda un gráfico de objetos serializándolos en un solo objeto grande (LOB), que almacena en un campo de base de datos

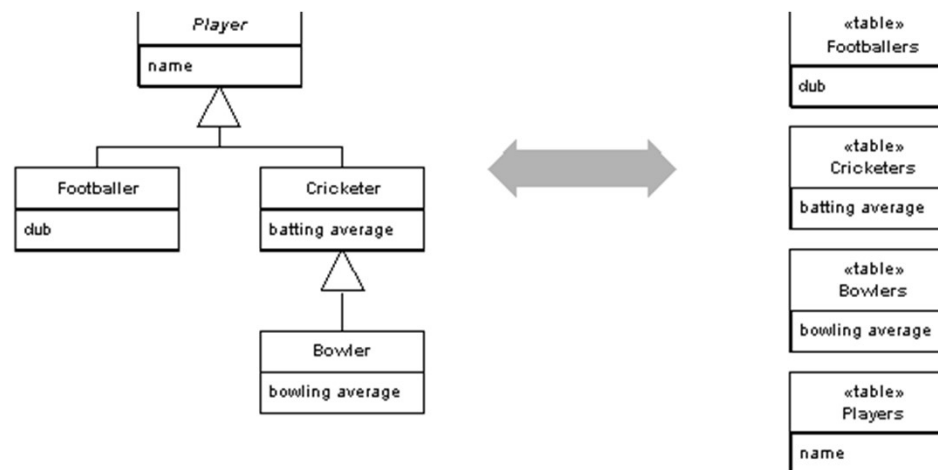


Object-Relational Structural Patterns

Single Table Inheritance. Representa una jerarquía de clases de herencia como una tabla única que tiene columnas para todos los campos de las diversas clases..

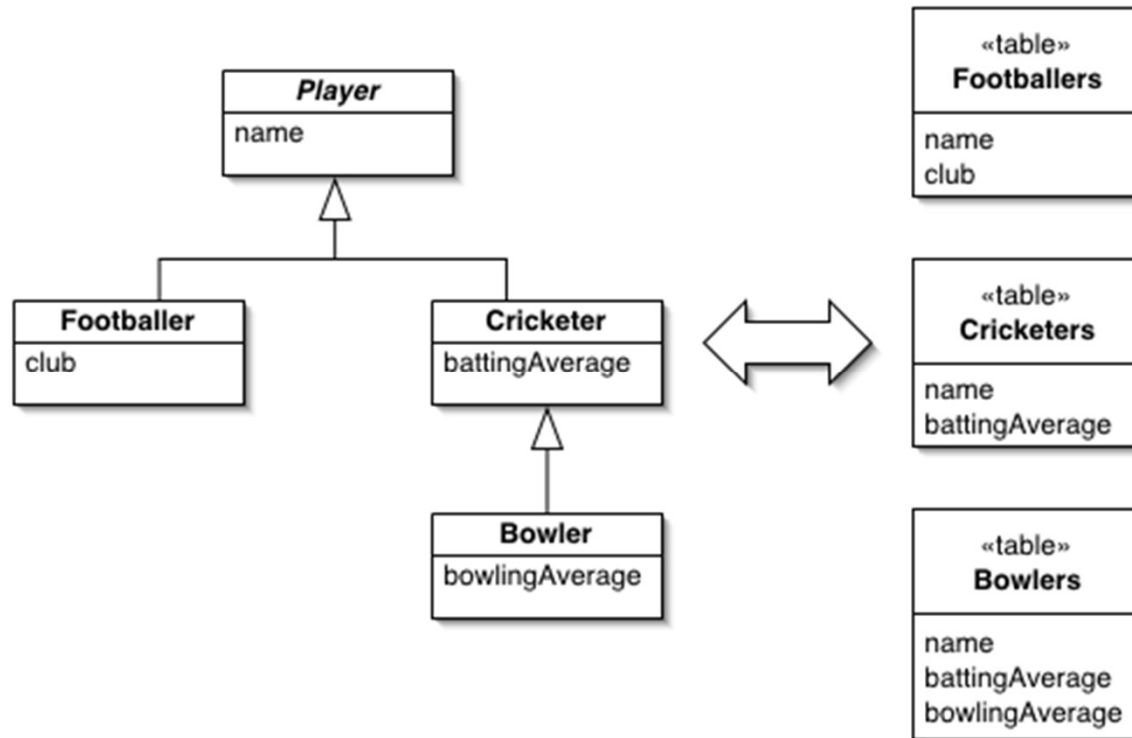


Class Table Inheritance. Represents an inheritance hierarchy of classes with one table for each class..



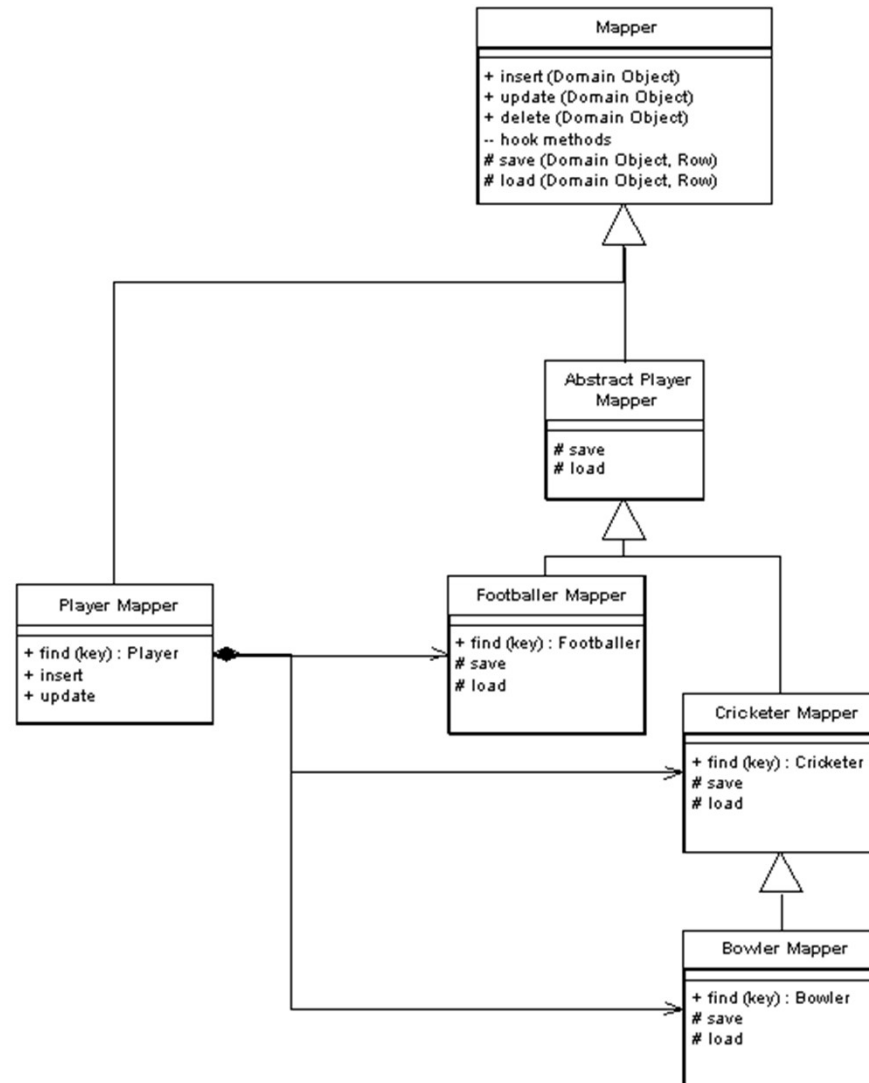
Object-Relational Structural Patterns

Concrete Table Inheritance. Representa una jerarquía de clases de herencia con una tabla por clase concreta en la jerarquía..



Object-Relational Structural Patterns

Inheritance Mappers. A structure to organize database mappers that handle inheritance hierarchies..

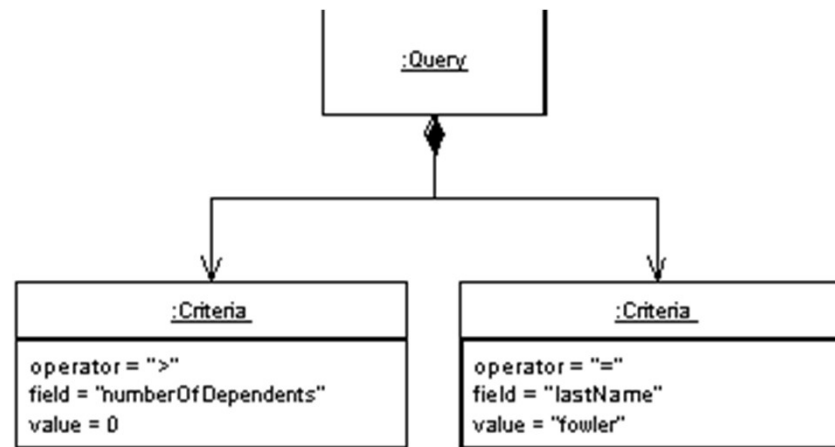


Object-Relational Metadata Mapping Patterns

Metadata Mapping. Contiene detalles de mapeo relacional de objetos en metadatos..

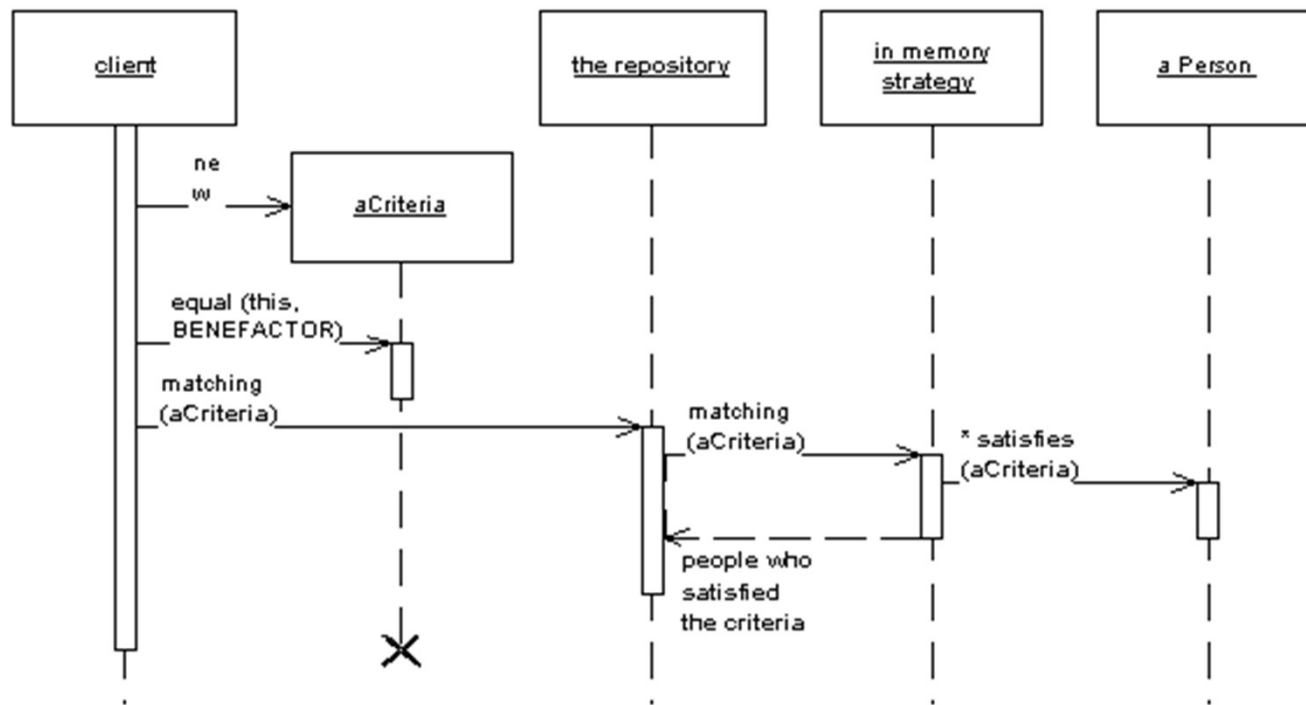


Query Object. Un objeto que representa una consulta de base de datos..



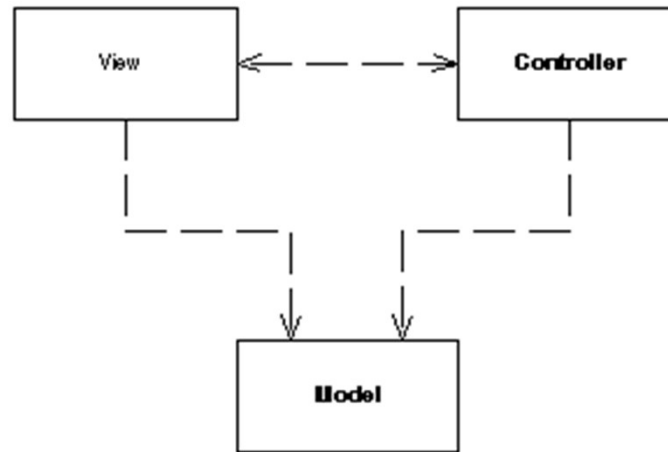
Object-Relational Metadata Mapping Patterns

Repository. Media entre el dominio y las capas de mapeo de datos utilizando una interfaz similar a una colección para acceder a objetos de dominio..

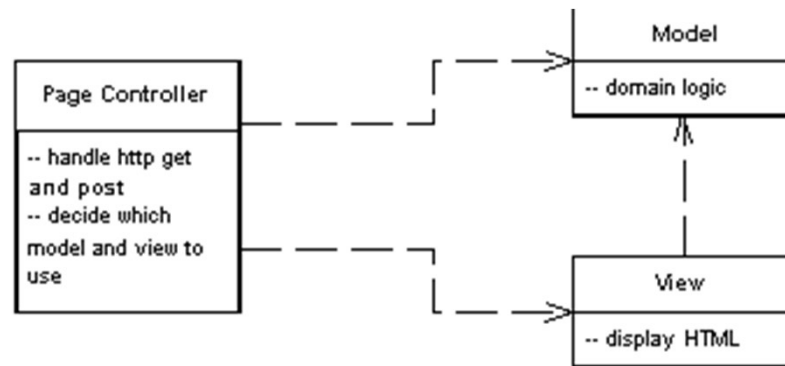


Web Presentation Patterns

Model View Controller. Divide la interacción de la interfaz de usuario en tres roles distintos..

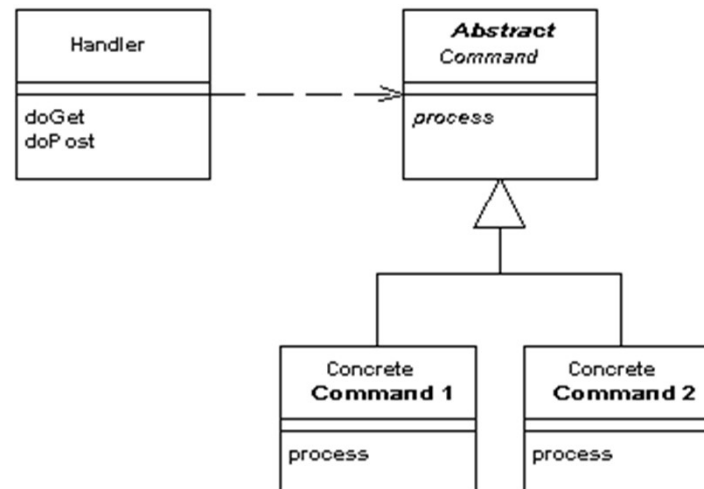


Page Controller. Un objeto que maneja una solicitud de una página o acción específica en un sitio web..

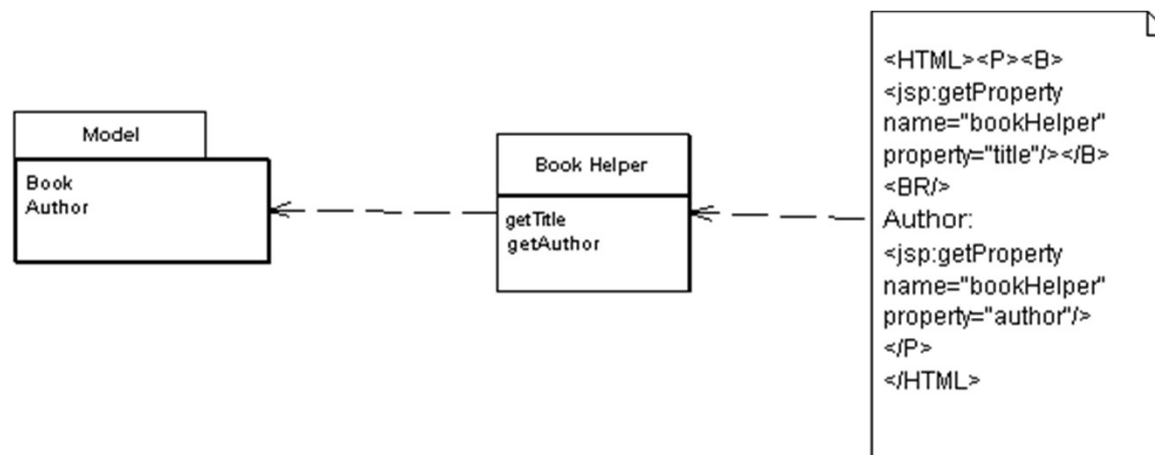


Web Presentation Patterns

Front Controller. A controller that handles all requests for a Web site.

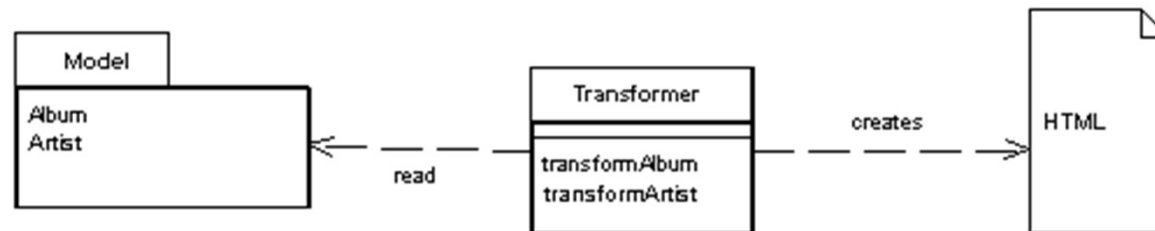


Template View. Representa información en HTML mediante la inserción de marcadores en una página HTML..

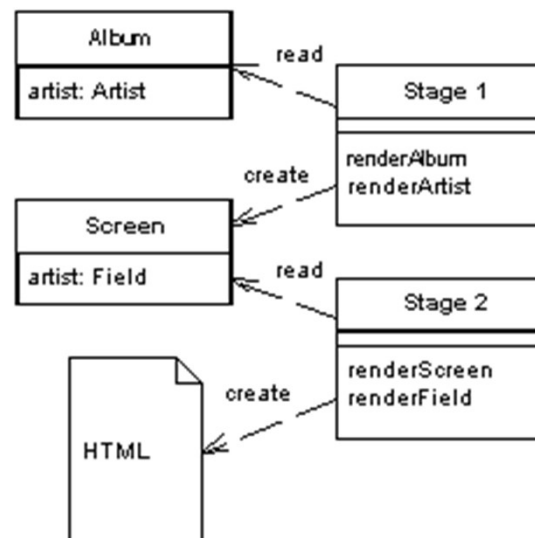


Web Presentation Patterns

Transform View. Una vista que procesa datos de dominio elemento por elemento y los transforma en HTML.

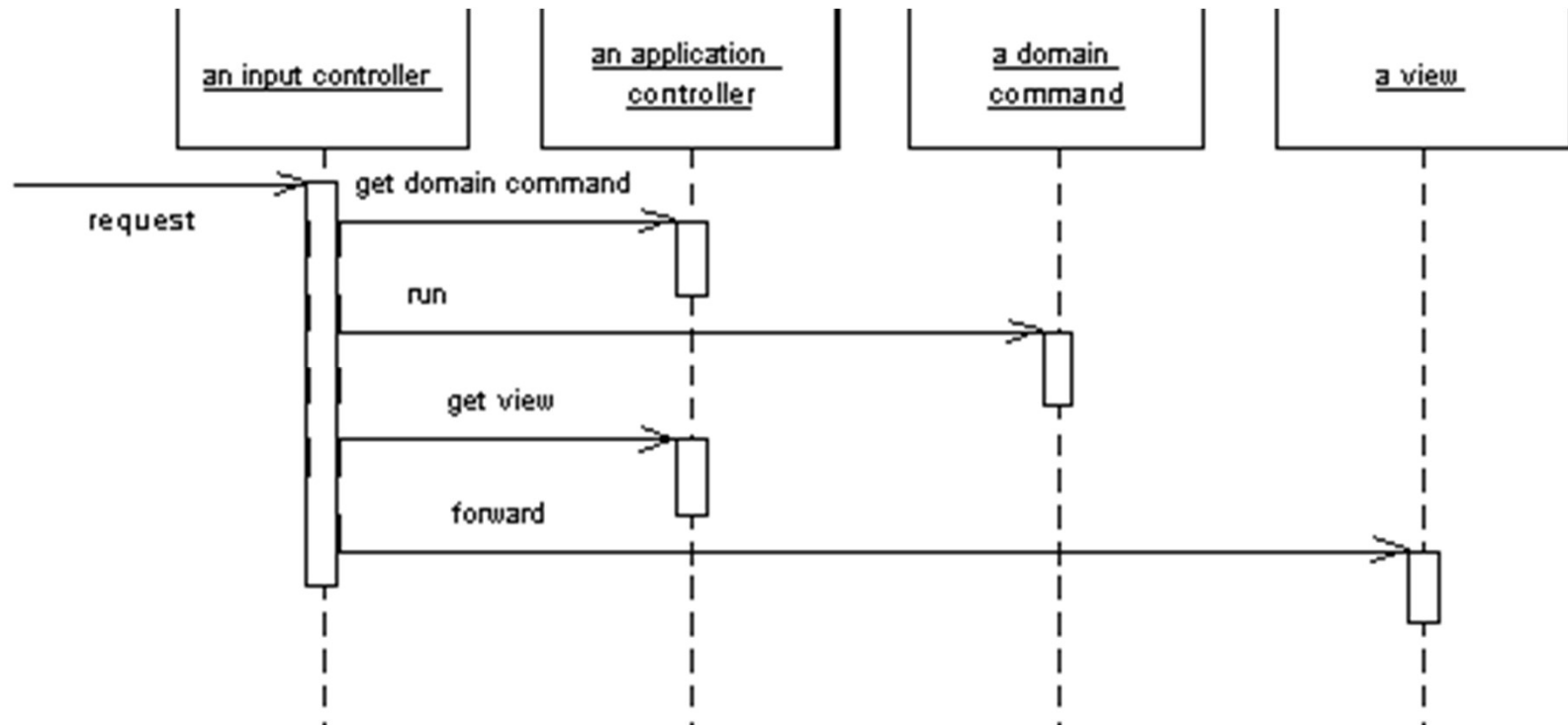


Two Step View. Convierte los datos del dominio en HTML en dos pasos: primero formando algún tipo de página lógica, luego renderizando la página lógica en HTML.



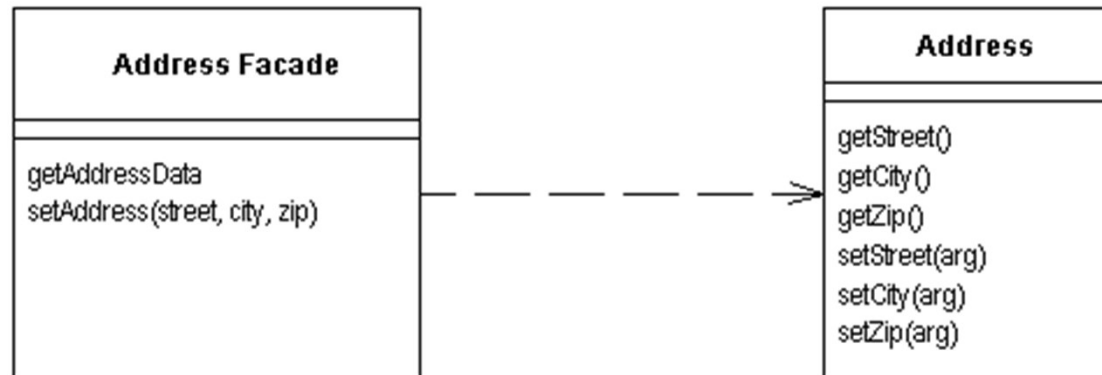
Web Presentation Patterns

Application Controller. Un punto centralizado para manejar la navegación de pantalla y el flujo de una aplicación.

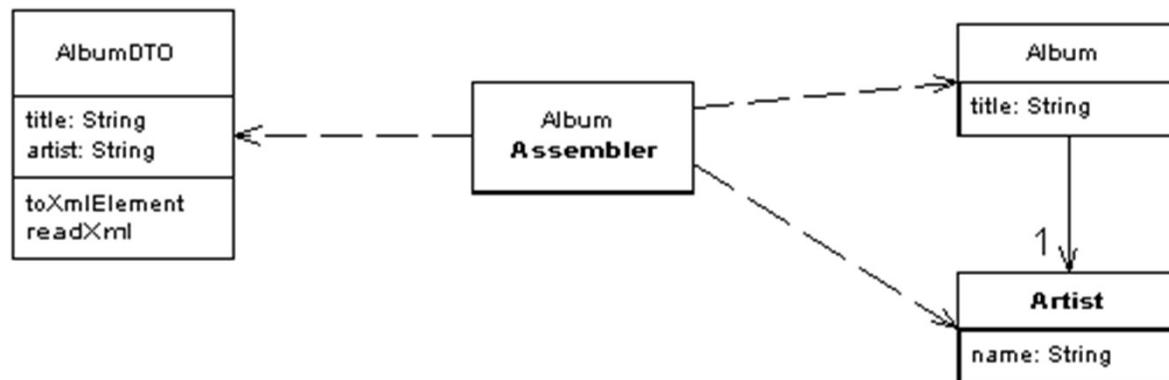


Distribution Patterns

Remote Facade. Proporciona una fachada de grano grueso en objetos de grano fino para mejorar la eficiencia en una red.

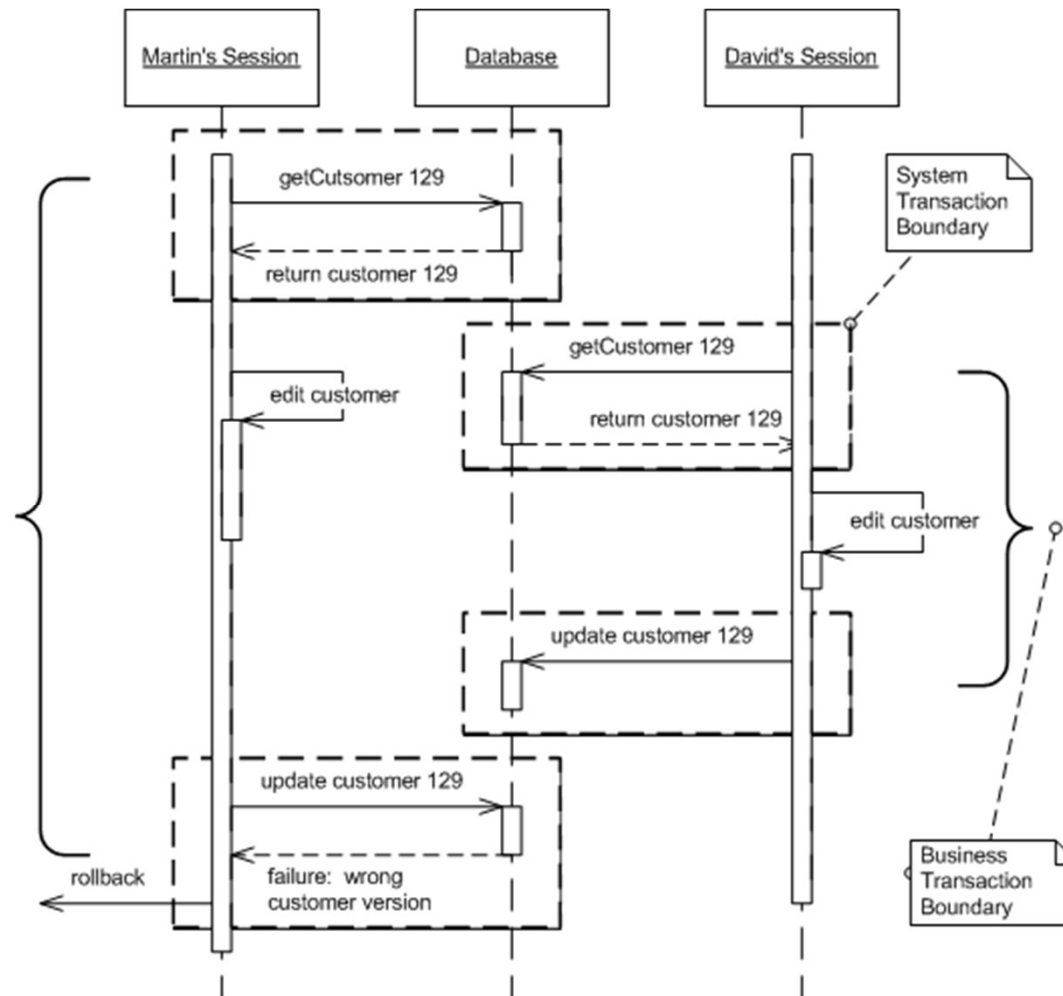


Data Transfer Object. Un objeto que transporta datos entre procesos para reducir la cantidad de llamadas a métodos.



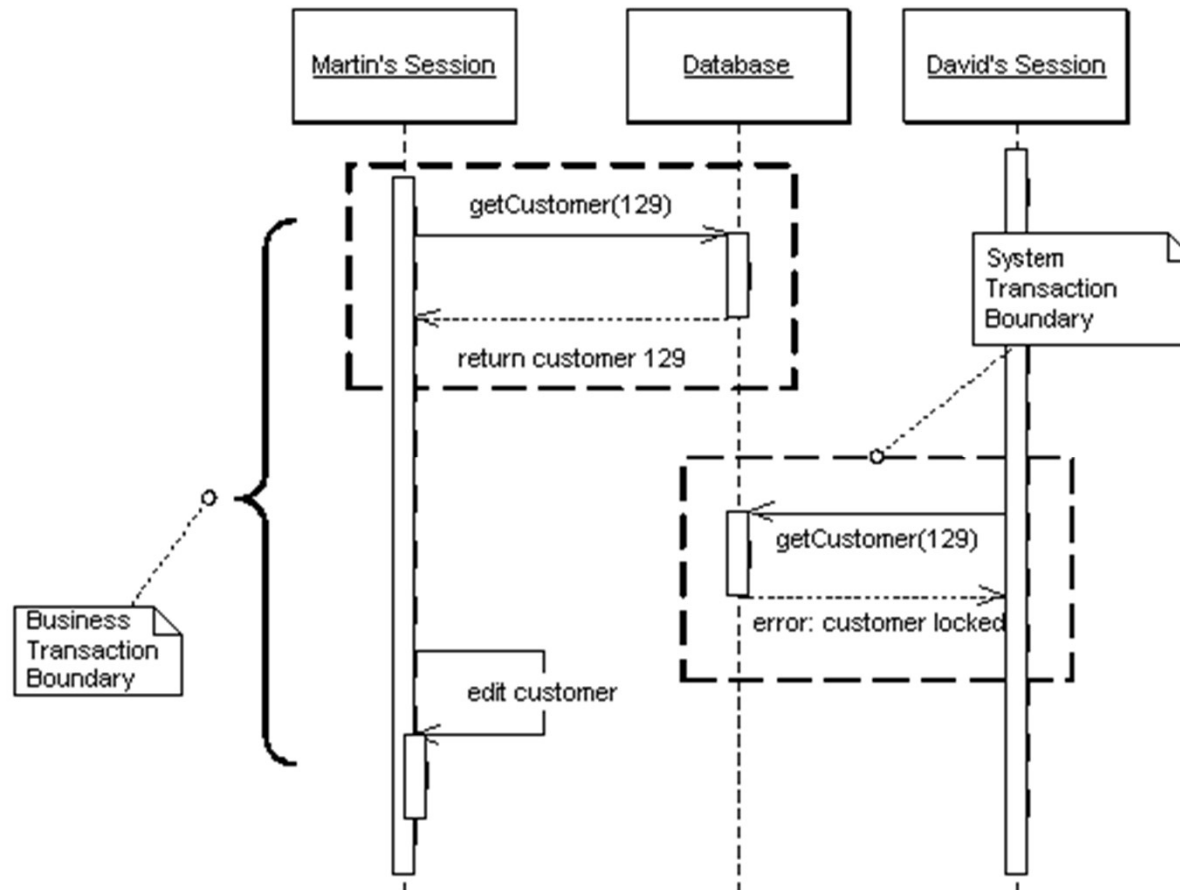
Offline Concurrency Patterns

Optimistic Offline Lock. Previene conflictos entre transacciones comerciales concurrentes al detectar un conflicto y revertir la transacción.



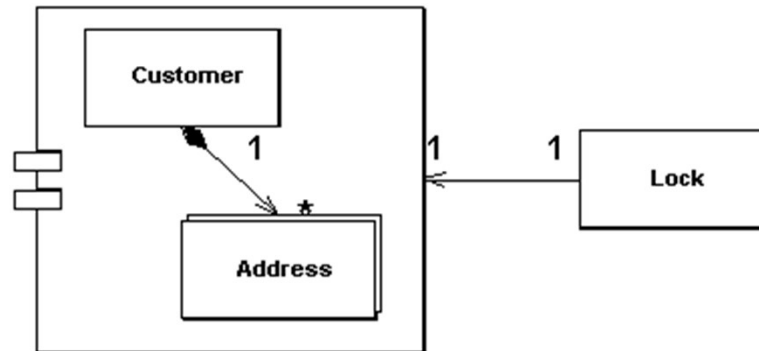
Offline Concurrency Patterns

Pessimistic Offline Lock. Evita conflictos entre transacciones comerciales concurrentes al permitir que solo una transacción comercial a la vez acceda a los datos.

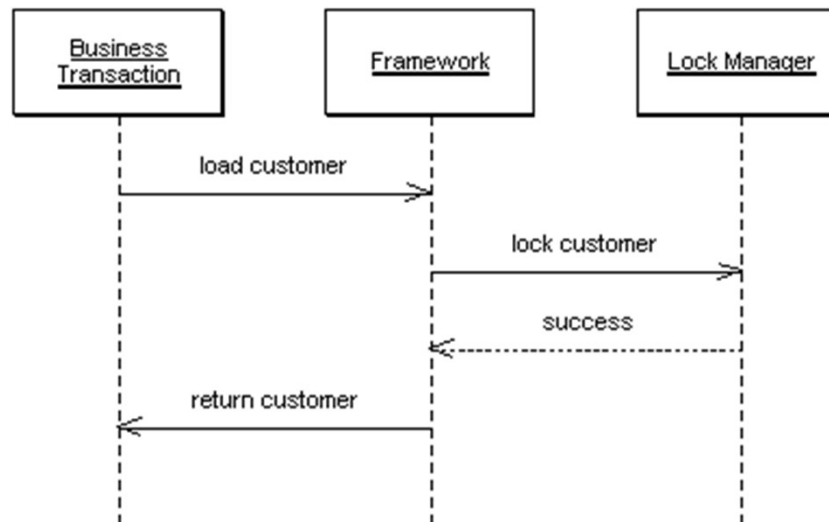


Offline Concurrency Patterns

Coarse-Grained Lock. Bloquea un conjunto de objetos relacionados con un solo bloqueo..



Implicit Lock. Permite que el código de supertipo de marco o capa adquiriera bloqueos fuera de línea.



Session State Patterns

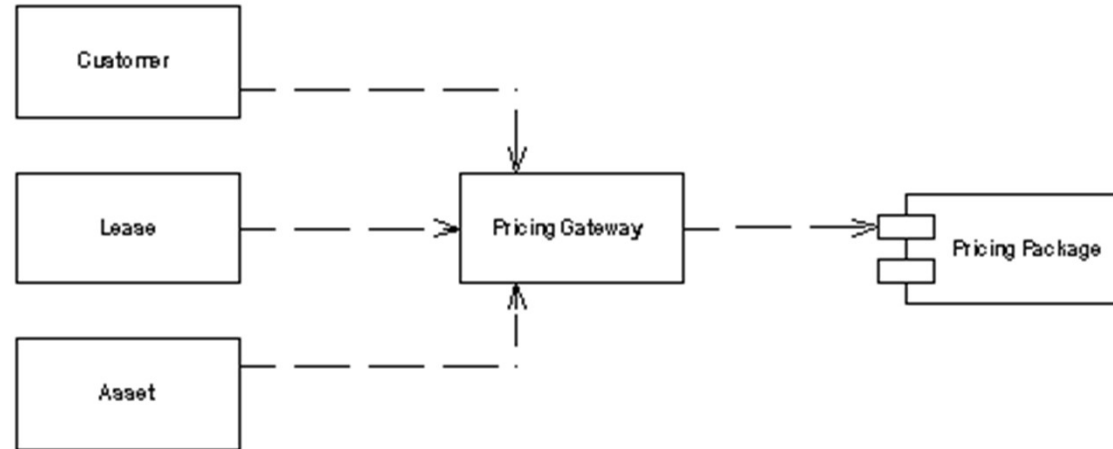
Client Session State. Almacena el estado de la sesión en el cliente.

Server Session State. Mantiene el estado de la sesión en un sistema de servidor en forma serializada.

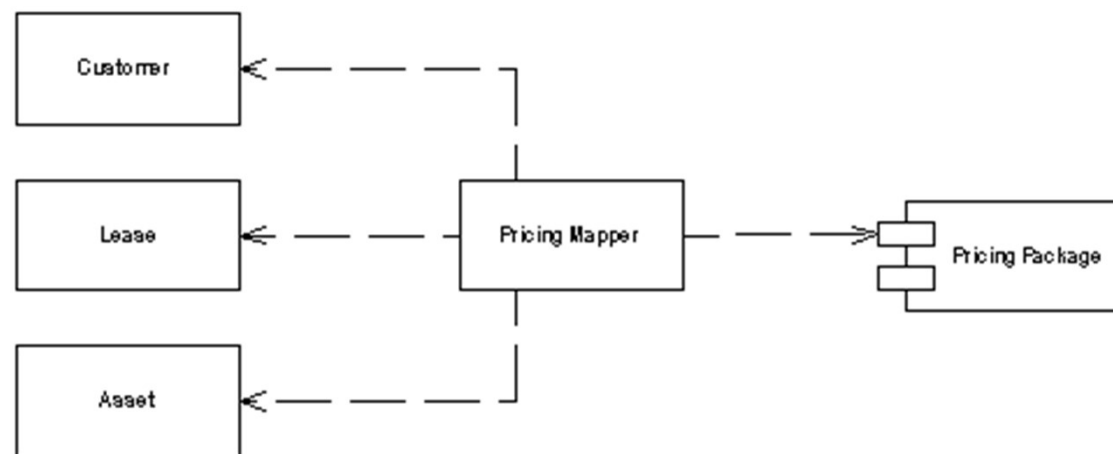
Database Session State. Almacena datos de sesión como datos confirmados en la base de datos.

Base Patterns

Gateway. Un objeto que encapsula el acceso a un sistema o recurso externo.



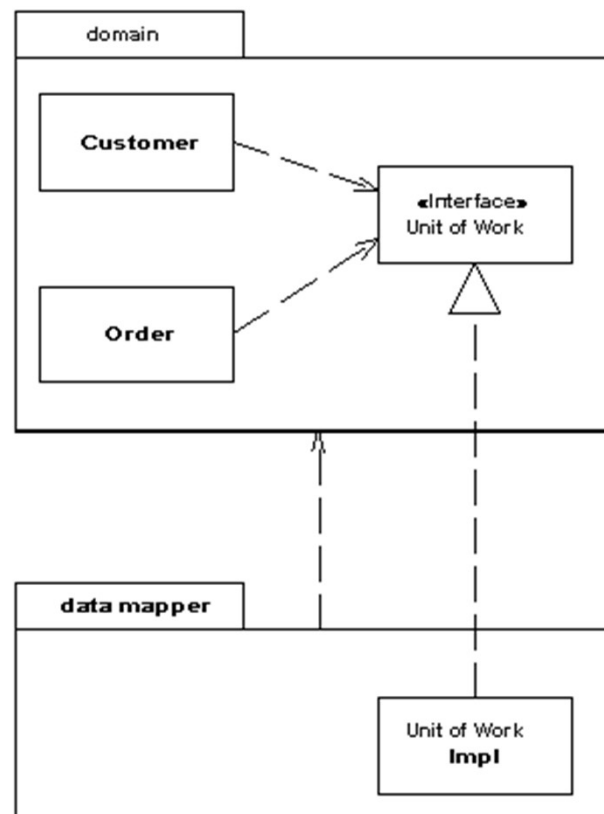
Mapper. Un objeto que establece una comunicación entre dos objetos independientes.



Base Patterns

Layer Supertype. Un tipo que actúa como el supertipo para todos los tipos en su capa.

Separated Interface. Define una interfaz en un paquete separado de su implementación.



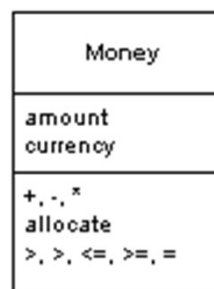
Base Patterns

Registry. Un objeto bien conocido que otros objetos pueden usar para encontrar objetos y servicios comunes.



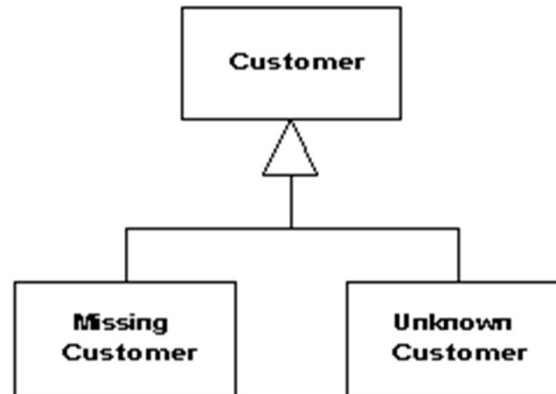
Value Object. Un objeto pequeño y simple, como el dinero o un rango de fechas, cuya igualdad no se basa en la identidad.

Money. Representa un valor monetario.



Base Patterns

Special Case. Una subclase que proporciona un comportamiento especial para casos particulares.

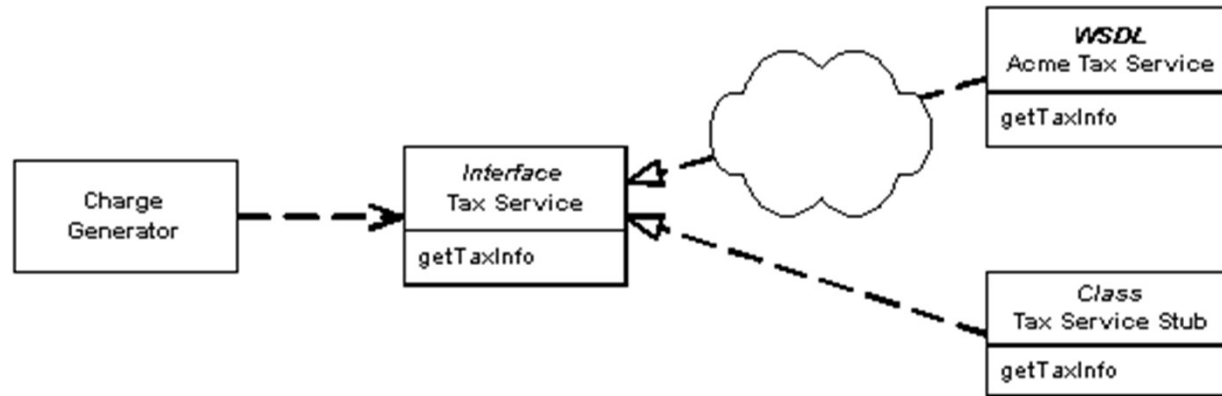


Plugin. Vincula las clases durante la configuración en lugar de la compilación..

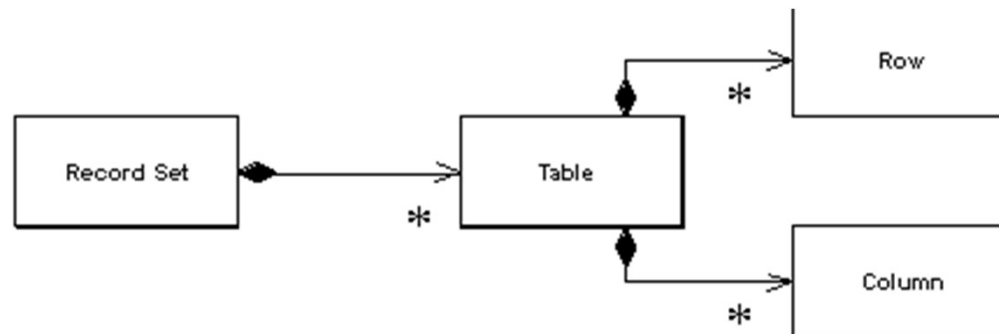


Base Patterns

Service Stub. Elimina la dependencia de servicios problemáticos durante las pruebas. WSDL



Record Set. Una representación en memoria de datos tabulares..



RESUMEN

Recordemos

Patterns of Enterprise Application Architecture



REFERENCIAS

Para profundizar

<https://martinfowler.com/eaCatalog/index.html>



PREGRADO

Ingeniería de Software

Escuela de Ingeniería de Sistemas y Computación | Facultad de Ingeniería



UPC

Universidad Peruana
de Ciencias Aplicadas

Prolongación Primavera 2390,
Monterrico, Santiago de Surco
Lima 33 - Perú
T 511 313 3333
<https://www.upc.edu.pe>

exígete, innova