Patrones Básicos

Gateway

Encapsular lógica de acceso
Ej: API

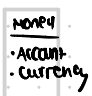
Layer Supertype

crear supertipos y subtipos

Registry

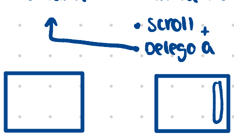
contenedor
Registra y almacena obj necesarios

Money



Plugin

Extender f(x) → sin mod código Base



Mapper

lógica negocio ←→ lógica datos

Transforma obj dominio en obj almacenamiento

Separated Interface

≠ interfaces para que puedan evol. indep.

Value object

obj del ATR, no por entidad
↳ obj = campos ; = objeto

Special Case

clases q manejan excepciones de forma bonita

Service Stub

simular comport → externos (q aún no disponible)

Record Set

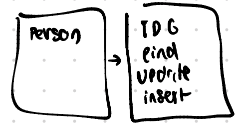
Representar csv

Data Source Arq Patterns

Persistencia datos

Table Data Gateway

1 instancia maneja toda la tabla



Row Data Gateway

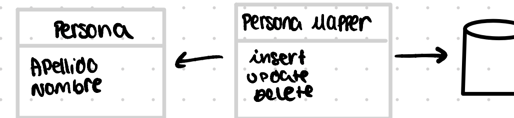
Pasarela a un registro 1 tabla
Transcription Script

Active Record

datos + comportam ⇒ 1 clase

Data Mapper

Traducir / adaptar obj a la base de datos



Web Representation Patterns

Page Controller

1 obj cada página (URL)

Front Controller

1 controller → genere URLs
Trata comportamiento común para request
↓
delega

Aplication Controller

Centraliza en un sitio lo q es común para todas las request

controller → delega en APP de controles

Template View


→ Book Helper → HTML (rellena agujeros)

Transform View

no partes HTML sino q lo produces

Two Steps

1. Def. objetos + disparar vistas
2. Combinarlo

Single Table INHERITANCE

nombre	tiros	goles	paradas	tipo
Carlos	NULL	40	NULL	futbolista
Paloma	33	NULL	NULL	baloncestista
Orianna	NULL	1	123	portero

No hace falta Joins

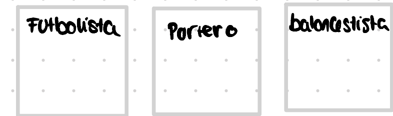
Class Table Inheritance

Tablas:

evita uniones complejas

Concrete Class Table Inheritance

Tablas:



¿select * from jugador?
⇓
complicado

Cada clase accede solo a su tabla
select * from Futbolista (fácil, no Joins)

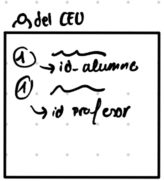
Object Relational Structural

Identity Field

identificador único / exclusivo

DNI → semántica
€ 𝒜, puede cambiarlo ✗✗

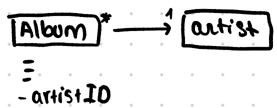
Alumnos	Profesores
1 ~	1 ~
2 ~	2 ~
3 ~	3 ~



ya no es un identif único

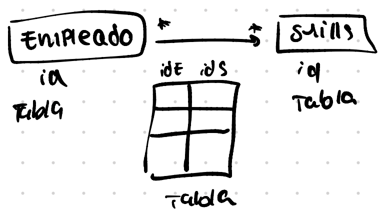
Foreing key Mapper

Relaciones 1-N

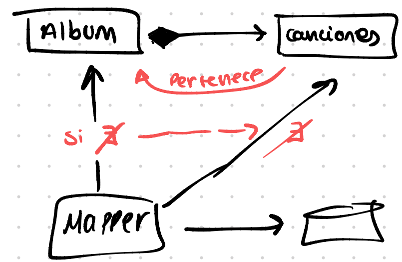


complejidad

Association Table Mapping



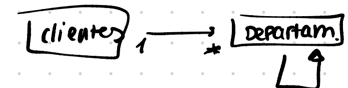
Dependent Mapping



Delete on cascade idAlbum

Inheritance Mapper

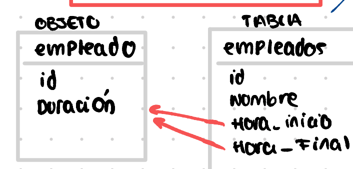
Serialized LOB



¿cómo haces select de un depart hijo?
↓
difícil
⇓
Encapsulas BLOB / CLOB
↓
xml

Embedded Value

1 ✗ N



Mapea un objeto a varios campos Tabla

Metadata Mapping

Metadatos → datos q describen otros datos < origen, contenido, formato, ...

evitar inconsistencias

Asociación

Valores / Campos Metadatos — Sist1 ⟷ Sist2

ORM

code generation

Programa genera código

crear clases → Metadatos (mapeados)

Reflective Programming

Programa va mod + inspecc su propia estructura y comportamiento

setters → a partir metadatos

Query - object

no escribir SQL

Representar consulta en la BBDD

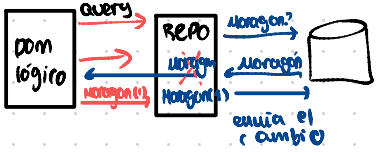
NO te tienes q preocupar de las PARTICULARIDADES del motor / BD

Interpreter (especif.) — Identity Map — Domain Model

Metadata Mapping (necesita relación entre objeto BD)

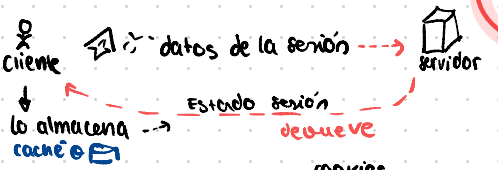
Data Mapper

Repository



object Relational Metadata

Client Session State

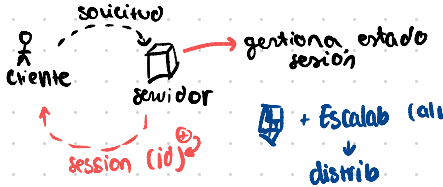


para mantener sesión < cookies, tokens autenticación

Session state patterns

Server Session State

Gestiona estado sesión



+ Escalab (almacen nº solicitudes) → distrib carga

Database Session

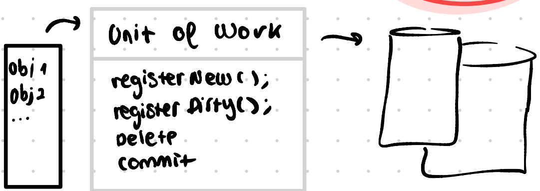
necesita id de la sesión para recuperar la info.

escalabilidad
durabilidad
Δ nº servidores
Flexibilidad
Control

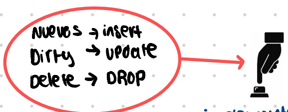
Unity of Work

Object-relational Behavioral

Transacciones



Lectora
Escritura > organiza



implementación típica de una sesión

incluso puede haber obj que no ∃ en la BD

Registro por llamada

Auto-Registro

UoW controlador BD

Identity Map

Garantiza q cada objeto se cargue solo 1 vez

Elección clave primaria ⚠

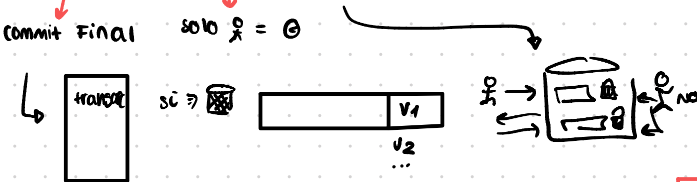
Lazy load

optimiza carga aplicaciones del usuario

Optimistic/Pesimistic offline lock

commit final

solo



¿Hay una versión nueva?
NO → commit

Coarse grained lock

Bloquear secciones código/datos

∇ Sobrecarga
ineficiente

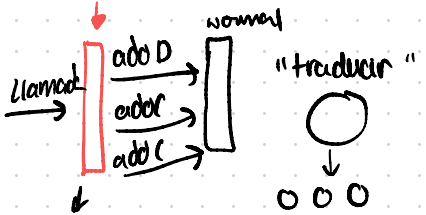
online concurrency

Implicit lock

Bloqueo activa → accede recurso

verUsuario()
↳ BloquearUsuario()
← usuario

Remote Façade



encapsular interacc complejas

Transcription Script

separa Fcs) en métodos = clase

checkAvailability
calculatePrice
BookRoom
CommitTransaction

→ bookRoom()

organizar lógica en procedimientos

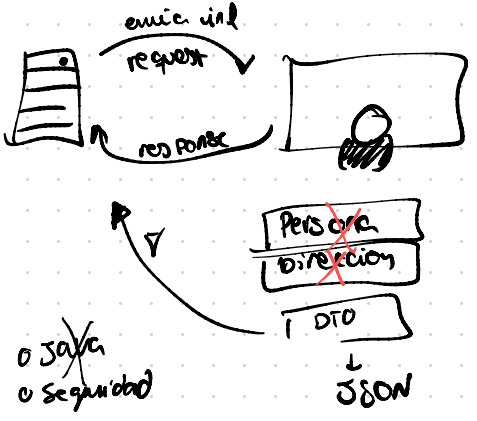
Refactor ⇒ Template Method

⇒ Especif. Composite

Domain Logic

dominios con poca funcionalidad ✓

Distrib. Patterns



Domain Model

incorporar → comportam. (Fcs) (Attr) } encapsular
→ Datos

No lo entiendo

DTO (Data Transfer Object)

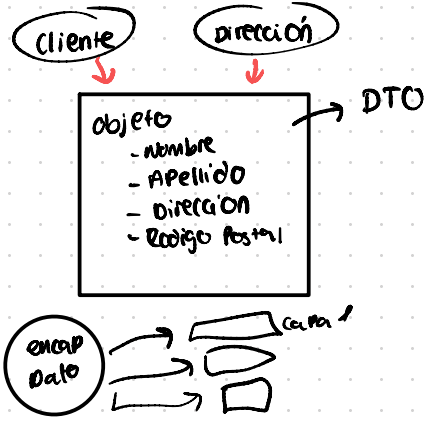


Table Module

active record con un tabla

instancia → tabla/vista

1 clase → tabla

Service Layer

un Façade para accesos externos?!