

PATRONES DE INTERACCIÓN CON BASES DE DATOS

OBJETIVOS

- ① Conocer cuáles son los patrones para la interacción con una base de datos.
- ① Seleccionar adecuadamente el patrón de interacción con una base de datos de acuerdo con los requerimientos del proyecto.

Dennis S. Cohn MSc, Mag. Ing.
Pontificia Universidad Católica del Perú



1

PATRONES DE INTERACCIÓN

Patrones comúnmente utilizados ...

INTERACCIÓN ENTRE OBJETOS Y TABLAS

Definen los objetos responsables para la lectura y escritura de datos en las tablas.

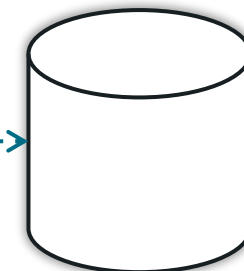
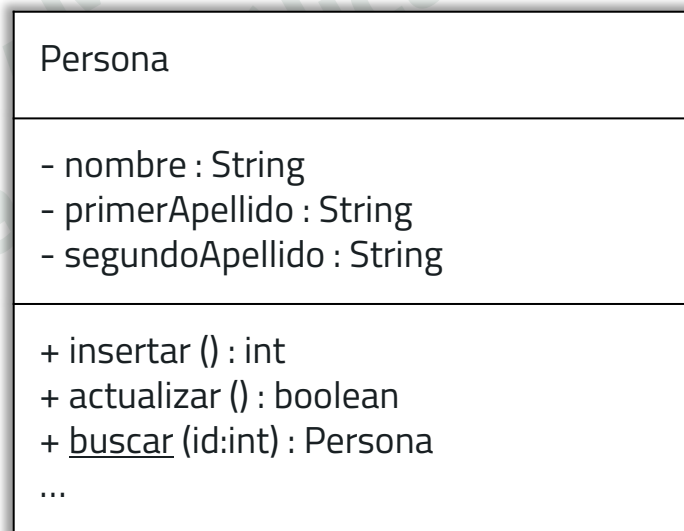


PATRÓN: ACTIVE RECORD

Un mismo objeto gestiona la **lógica** de la entidad y la **interacción** con el repositorio de datos.

Consideraciones:

- Correspondencia atributos - columnas.
- Contienen los SQL.
- Métodos estáticos de búsqueda que retornan "Active Records"
- Métodos de actualización e inserción utilizan datos de los atributos.



¿CUÁNDO USAR ACTIVE RECORD?

Ventajas:

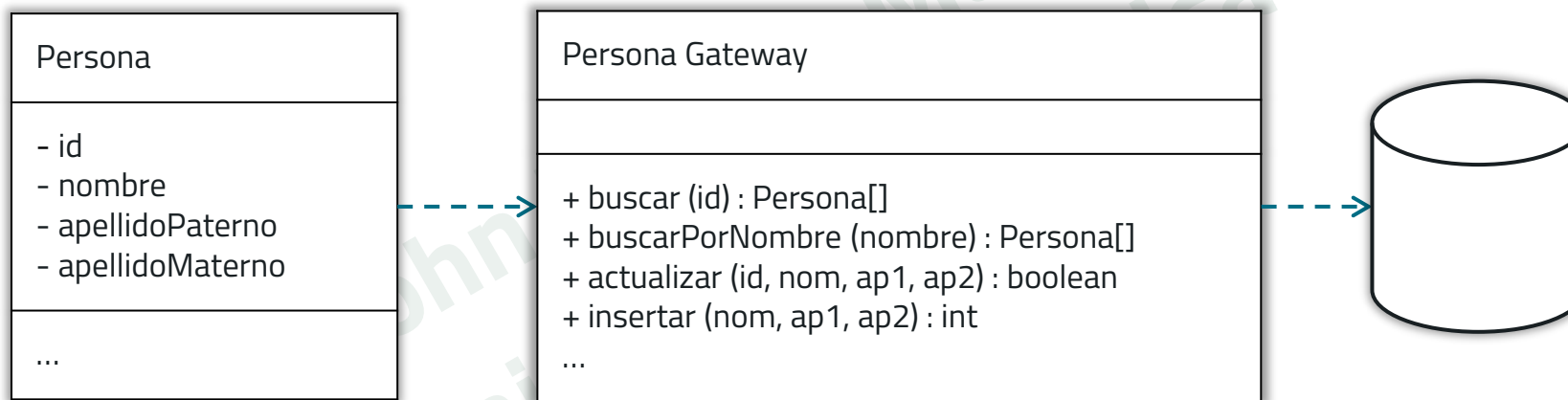
- Fácil de implementar.
- Correspondencia objeto-tabla.
- Eficaz cuando el modelo del dominio es simple.

Desventajas:

- Complejidad cuando no hay correspondencia objeto - tabla.
- Dificultad para trabajar con herencias y colecciones.
- Fuerte acoplamiento al diseñar la clase y la tabla.
- El objeto conoce la existencia del repositorio y su estructura.

PATRÓN: TABLE DATA GATEWAY

El objeto **Gateway** funge como **intermediario** con el repositorio de datos.



Consideraciones:

- Correspondencia atributos - columnas.
- Gateway contiene los SQL.
- Métodos de búsqueda deben retornar listas.
- Métodos mapean los parámetros a las sentencias SQL.

¿CUÁNDO USAR TABLE DATA GATEWAY?

Ventajas:

- Fácil de implementar.
- Encapsula la lógica para interactuar con el repositorio.
- Se puede utilizar de forma combinada con consultas y procedimientos almacenados.

Desventajas:

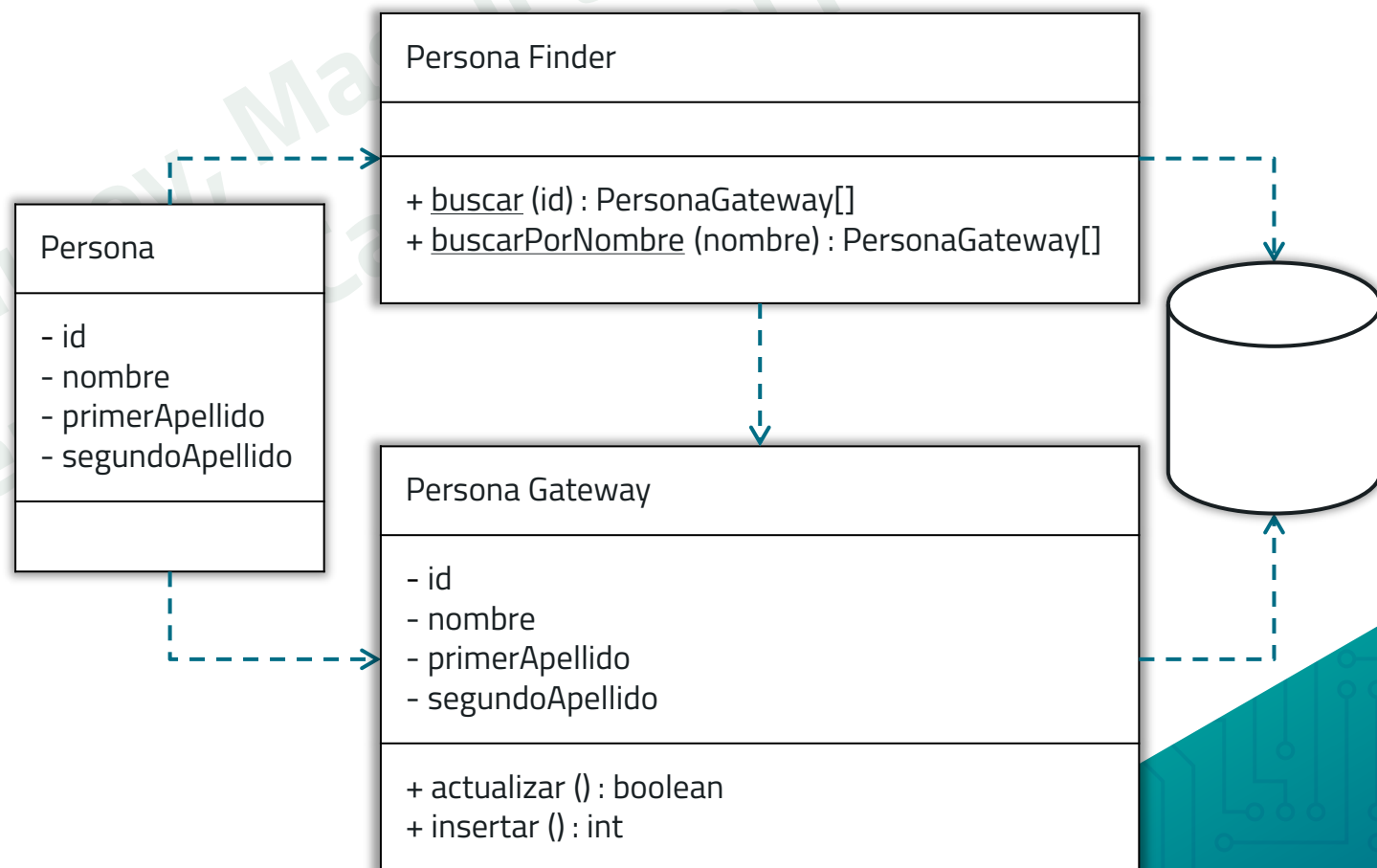
- Dependencia circular entre el Gateway y la Entidad.
- El objeto entidad conoce la existencia un repositorio.

PATRÓN: ROW DATA GATEWAY

El objeto **Gateway** representa la estructura de la tabla y funge como **intermediario** con el repositorio de datos.

Consideraciones:

- Gateway contiene los SQL.
- Métodos de búsqueda deben retornar listas.



¿CUÁNDO USAR ROW DATA GATEWAY?

Ventajas:

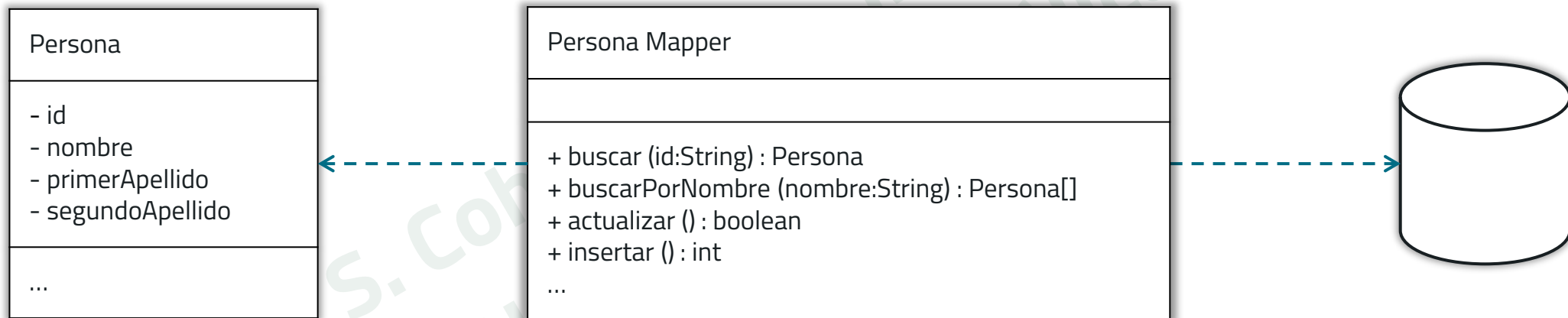
- Encapsula la lógica para interactuar con el repositorio.
- Independiza la estructura de la entidad de la estructura de la tabla.
- Se puede utilizar de forma combinada con consultas y procedimientos almacenados.

Desventajas:

- El objeto entidad conoce la existencia un repositorio.
- Complejidad de mantenimiento de las estructuras: entidad, gateway, repositorio.

PATRÓN: DATA MAPPER

Transfiere datos entre el objeto entidad y el repositorio de datos.



Consideraciones:

- Mapear atributos de la entidad con las columnas del repositorio.
- Mapper contiene los SQL.
- Métodos de búsqueda pueden retornar una entidad o una lista.

¿CUÁNDO USAR DATA MAPPER?

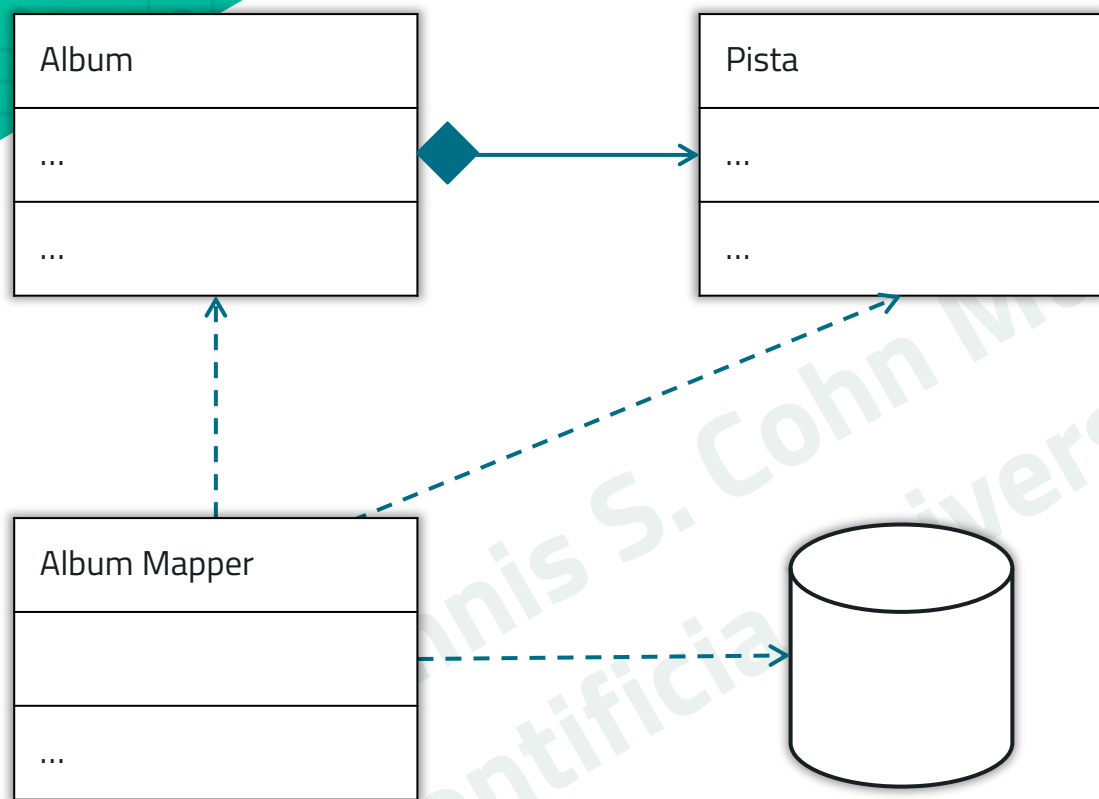
Ventajas:

- Soporta lógica compleja como colecciones y herencia.
- Independiza la estructura de la entidad de la estructura de la tabla.
- El objeto entidad no conoce la existencia de un repositorio

Desventajas:

- Dada la complejidad, se recomienda el uso de librerías y frameworks.

PATRÓN: DEPENDENT MAPPING



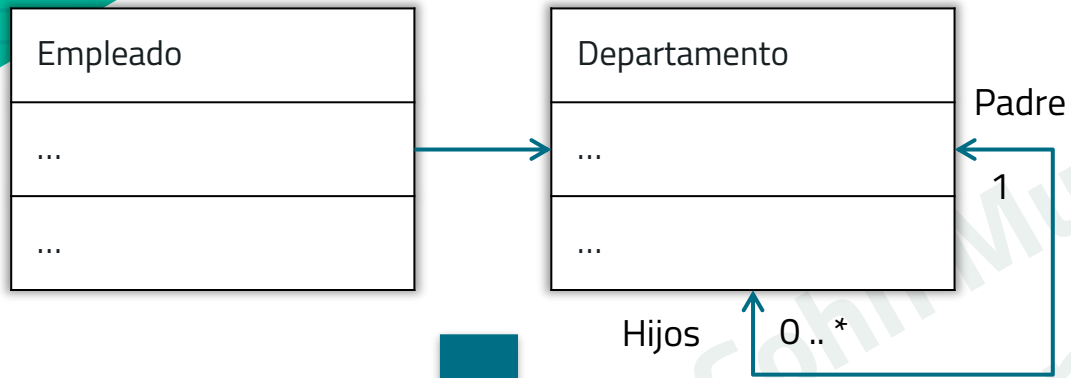
Cuando un objeto aparece en el contexto de otro; éste **delega su mapeo** al otro objeto.

Consideraciones:

- La clase "dependiente" depende de la clase "dueña" para gestionar su persistencia.
- La clase "dependiente" debe tener una sola clase "dueña".
- La clase "dependiente" solo debe ser referenciada desde la clase "dueña".

PATRÓN: OBJETO GRANDE (LOB) SERIALIZADO

Objeto



Tabla

Empleado
...
departamentos : BLOB

Almacena una estructura de objetos **serializándolos** en un único objeto.

Consideraciones:

- La serialización puede ser:
 - Binaria (BLOB).
 - Texto (CLOB).
- Cuidado de no duplicar data; salvo que se almacenen "snapshots".
- No es compatible con sentencias SQL.

¿CUÁNDO USAR BLOB Y CLOB?

BLOB

(Binary Large Object)

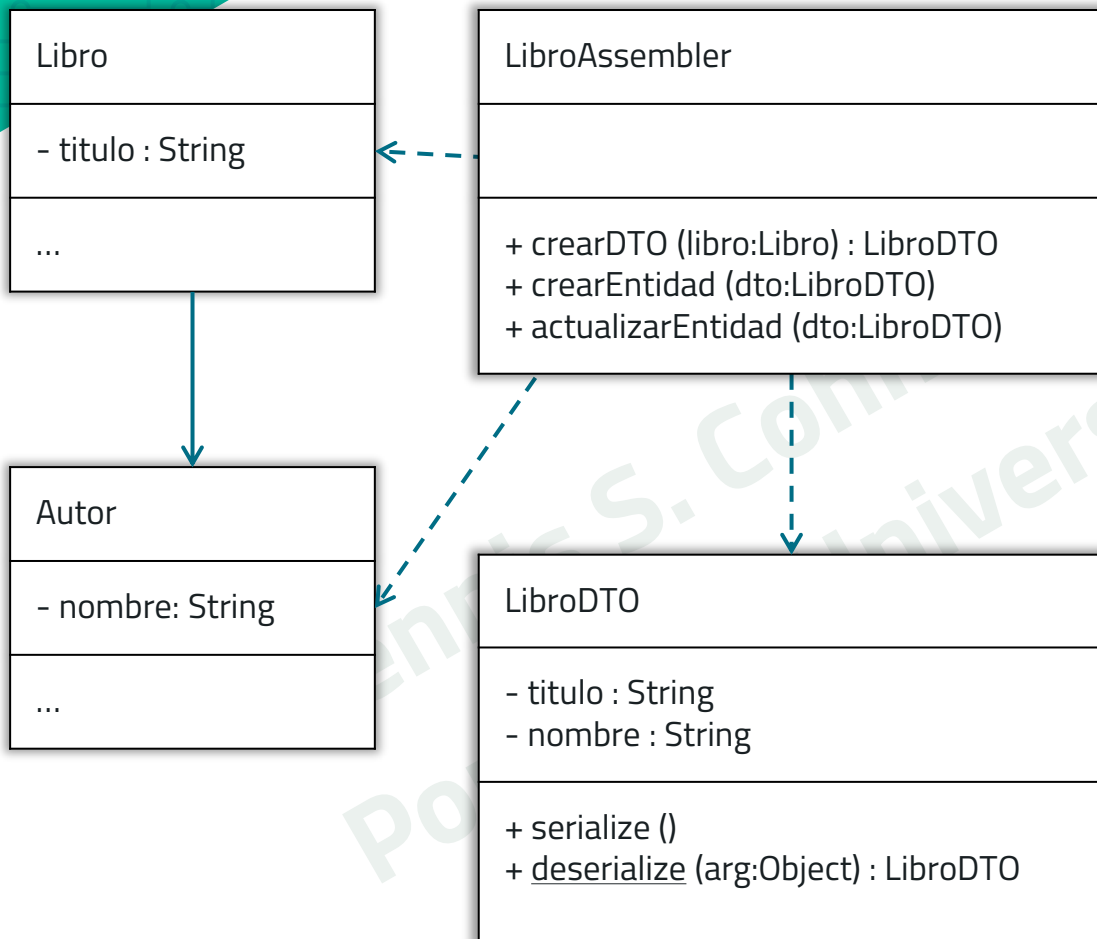
- Fácil de programar.
- Utiliza poco espacio.
- No puede reconstruirse los datos si no se cuenta con el objeto original.
- Un cambio en el objeto puede impedir la lectura de objetos previamente guardados.

CLOB

(Character Large Object)

- Serializa el objeto en una cadena de caracteres (JSON, XML).
- El texto es entendible a simple vista.
- Ocupa más espacio.
- Requiere implementar el serializador/deserializador.
- Más lento que el BLOB.

PATRÓN: DATA TRANSFER OBJECT (DTO)



Objeto que **traslada información** entre sistemas a fin de **reducir** el número de **llamadas** a métodos.

Consideraciones:

- Envía y recibir un gran número de registros con una sola llamada.
- El DTO debe ser un objeto serializable.
- Cliente y servidor deben contar con el mismo algoritmo para serializar/deserializar los datos.

2

REFERENCIAS

BIBLIOGRAFÍA

- Fowler, M. (2012). Patterns of Enterprise Application Architecture. Addison-Wesley.

Créditos:

- Plantilla de la presentación por [SlidesCarnival](#)
- Fotografías por [Unsplash](#)
- Diseño del fondo [Hero Patterns](#)