



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN®



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico Matemáticas
Licenciatura en Seguridad en Tecnologías de
Información

Diseño Orientado a Objetos

Maestro: Miguel Salazar

Patrones de diseño

Nombre: Daniel de la Rosa Rodríguez

Matrícula: 1666385

27 de octubre de 2017, Cd. Universitaria, San Nicolás
de los Garza, Nuevo León

¿Qué es un patrón de diseño y para qué sirven?

Son soluciones o herramientas para problemas típicos y recurrentes, enfocados al desarrollo de software, o en este caso aplicaciones web. Deben ser reusables permitiendo así que sean adaptados a diferentes condiciones o problemas reiterativos en el desarrollo, donde la solución se encuentra identificada mediante la aplicación de una serie de pasos, adaptando el sistema a una estructura definida por un patrón, garantizando que esta solución pueda ser aplicada cuantas veces sea necesario en circunstancias similares.

Existen 3 tipos de patrones:

- Creacionales: en torno a creación de objetos
- Estructurales: estructura de clases y objetos que las componen
- Comportamiento: clases y objetos son relacionados

Los patrones de diseño ayudan a identificar y documentar la experiencia en el diseño orientado a objetos, capturan soluciones recurrentes a sus *trade-offs* a problemas similares en un contexto dado. Proveen un vocabulario común para el diseño y la arquitectura, clarifican y documentan arquitecturas existentes. Hacen avanzar el diseño e influyen en las decisiones de dicho diseño.

Transfer Object (TO)

Es necesario transferir datos entre las diferentes capas de una aplicación, el TO es un componente que encapsula los valores de los atributos y provee métodos de acceso. Si la implementación puede ser:

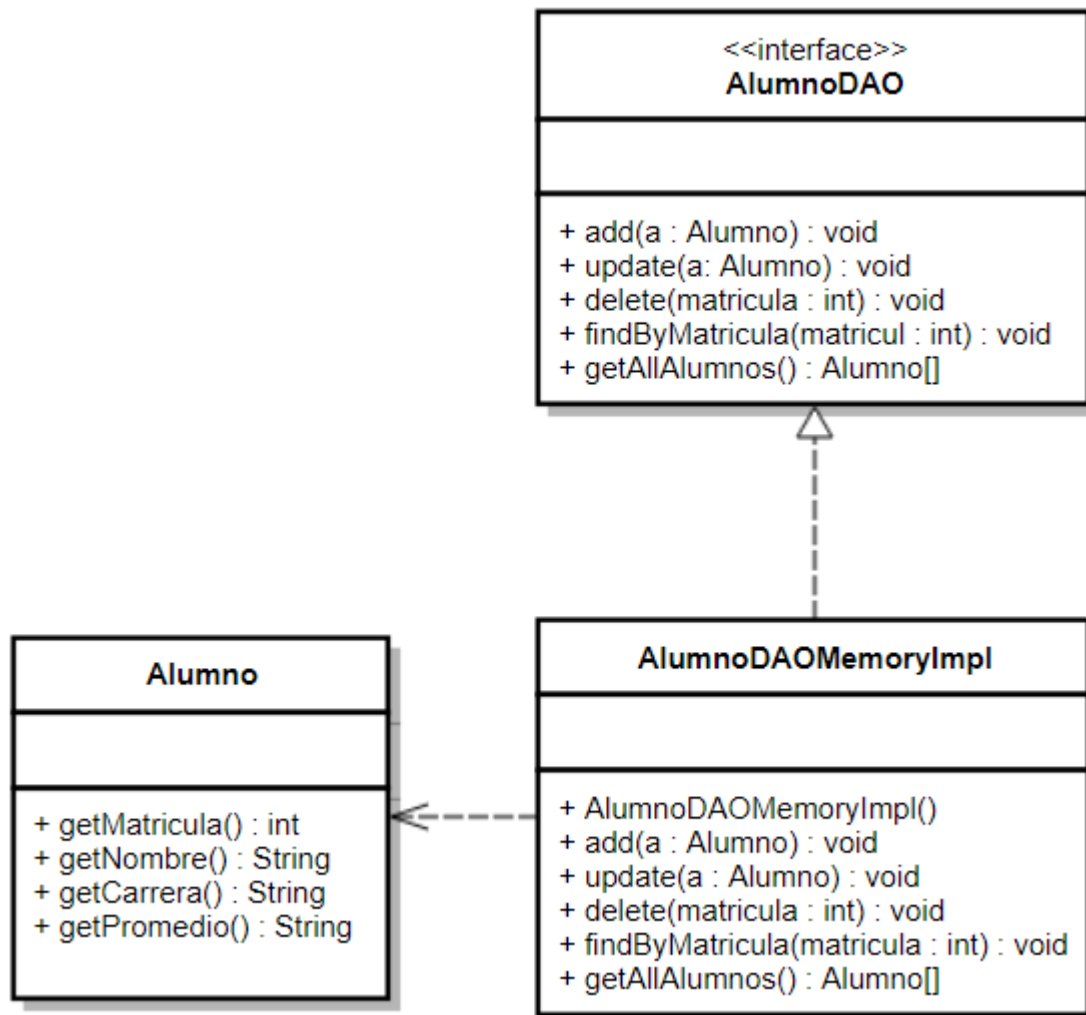
- JavaBean
- Plain Old Java Object (POJO)

Ejemplo:

```
public class Alumno {  
    private int matricula;  
    private String nommbre;  
    public int getMatricula() {  
        return matricula;  
    }  
    public String getNombre() {  
        return nombre;  
    }  
    public Alumno(int mat, String n) {  
        matricula = mat; nombre = n; } }  
}
```

Data Acces Object (DAO)

Se usa cuando se crea una aplicación que debe persistir información, separa el dominio del problema del mecanismo de persistencia. Usa una interfaz para definir los métodos usados para persistencia. La interfaz permite a la implementación de persistencia ser reemplazada con DAOs basados en memoria, archivos, JDBC, JPA, etc.



La lógica de negocio y de acceso a datos están separados.

Los objetos de acceso a datos promueven la reutilización del código y flexibilidad al hacer cambios en el sistema. Este diseño hace más fácil el cambio de tecnologías de front-end y back-end

Factory

Depende del uso de interfaces para definir una abstracción. Si se usa un constructor de implementación del DAO, se usaría siempre esa misma implementación. Usando un tipo de interfaz, las líneas subsecuentes no estarán ligadas a una única implementación.

La invocación del constructor se liga a una implementación específica y aparecerá en muchos lugares de la aplicación.

Usar Factory previene a la aplicación acoplarse fuertemente a una implementación DAO específica. La implementación del DAO permanece oculta.

```
public class AlumnoDAOFactory {  
    public AlumnoDAO crearAlumnoDAO() {  
        return new AlumnoDAOMemoryImpl();  
    }  
}
```

```
AlumnoDAOFactory factory = new AlumnoDAOFactory();  
AlumnoDAO dao = factory.crearAlumnoDAO();
```

Bibliografía

Romeo Sánchez. (2016). DOO, UANL.