

# Laboratorio II

## Introducción

El introducirnos en un nuevo lenguaje siempre es importante tener una vista básica de lo que los desarrolladores en tal plataforma han producido o suelen realizar para facilitar la implementación y eficiencia de su código. Los patrones de diseño no son más que una serie de costumbres, tradiciones o incluso implementaciones populares que permiten producir soluciones eficientes de código.

## POJO (Plain Old Java Object)

### ¿Qué es?

Es un objeto de notación sencilla y rápida, sin funcionalidades complejas, su objetivo es representar un objeto que interactúa en el sistema y sus atributos.

## DAO (Data Access Object)

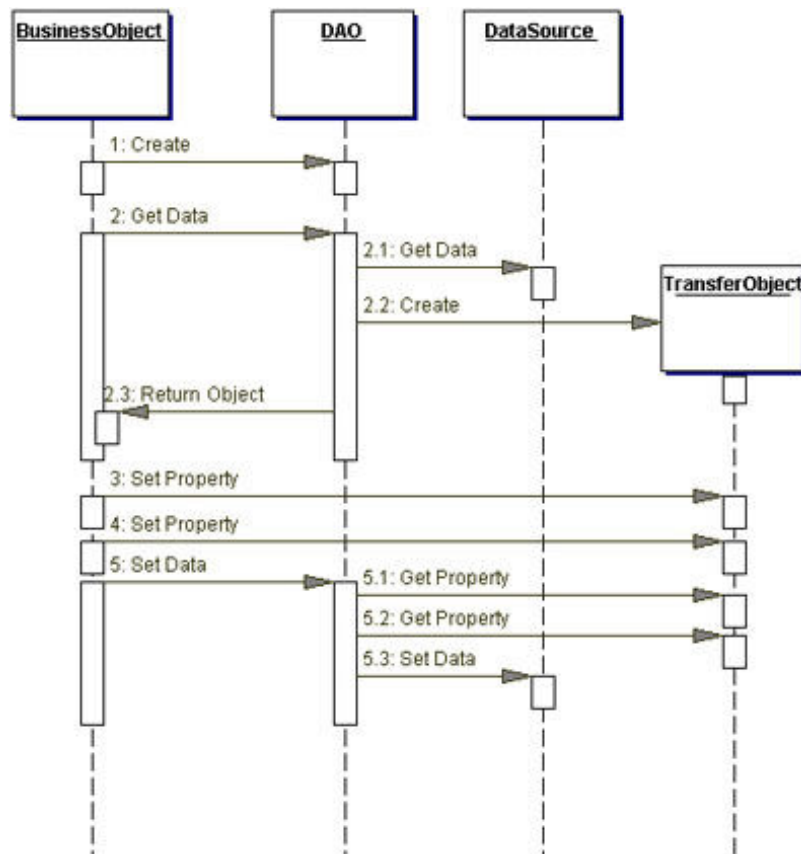
### ¿Qué es?

Una interfaz que provee métodos de interacción entre proveedores de datos (bases de datos, archivos) y una aplicación.

### ¿Para qué sirve?

El patrón DAO nace para facilitar el acceso a datos, desde una simple interfaz se proveen todos los métodos necesarios para solicitar información a un proveedor de datos, de esta forma se implementa la reutilización del código y se logra reducir la complejidad del mismo.

### Diagrama de implementación



## Singleton

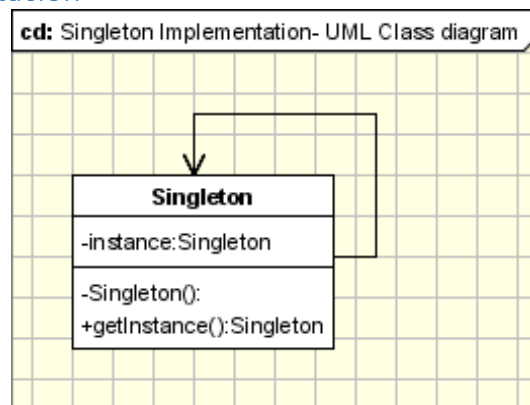
### ¿Qué es?

Es un patron cuyo diseño permite la reutilizacion de objetos previamente instanciados en memoria. Consiste en mantener una unica instancia de una clase, y asegurarse que esta sea accesible a todos los sectores necesarios.

### ¿Para qué sirve?

Facilita un punto de acceso a multiples sectores de una aplicacion a un solo objeto sin tener instanciar multiples al en cada ejecucion y generando mayor flexibilidad.

### Diagrama de implementacion



## Data Mapper

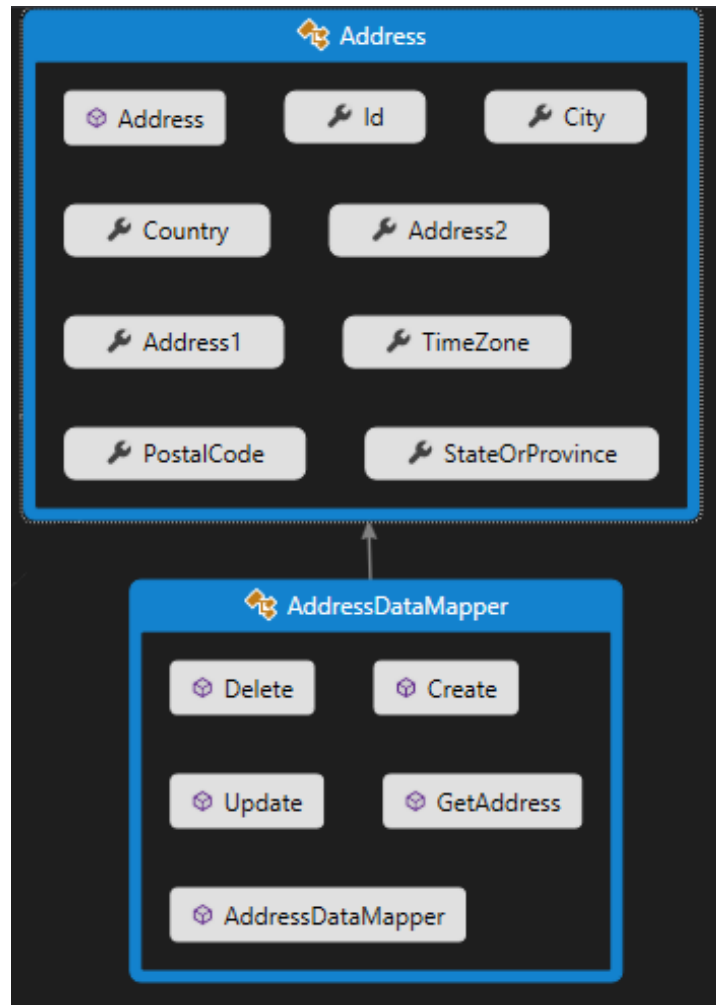
### ¿Qué es?

Es un patrón arquitectónico, el cual permite darle sentido a los valores provenientes de la base de datos además es responsable de enviar datos entre la memoria y la base de datos y también de aislar cada uno.

### ¿Para qué sirve?

Data mapper facilita una capa de abstracción la cual le da sentido a los datos provenientes de la base de datos y facilita su conversión en objetos en memoria.

### Diagrama de implementación



## Data Mapper

A. (n.d.). Data Mapper Pattern. Retrieved January 27, 2018, from <https://www.codeproject.com/Articles/821803/Data-Mapper-Pattern>

Braithwaite, B. (2013, August 15). Repository vs. Domain Model vs. Data Mapper. Retrieved from <http://www.bradoncode.com/blog/2013/08/repository-vs-domain-model-vs-data.html>

B. (n.d.). Using Statement (C# Reference). Retrieved January 27, 2018, from <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/using-statement>

Harrison, N. (2016, July 28). A Defense of Reflection in .NET. Retrieved January 27, 2018, from <https://www.red-gate.com/simple-talk/dotnet/.net-framework/a-defense-of-reflection-in-.net/>

Michael, P. (2013, July 23). Connecting to a SQL Server Database with C#. Retrieved from <https://www.youtube.com/watch?v=EiUSK5-sv4Q>

On Design Patterns: When to use the Singleton? (n.d.). Retrieved January 27, 2018, from <https://stackoverflow.com/questions/228164/on-design-patterns-when-to-use-the-singleton>