



## Objetivos

- Afianzar el manejo de datos persistentes
- Profundizar en el patrón MVC
- Aprender a identificar la necesidad del uso del patrón DAO
- Comprender y utilizar el patrón DAO para resolver una problemática

## Conceptos

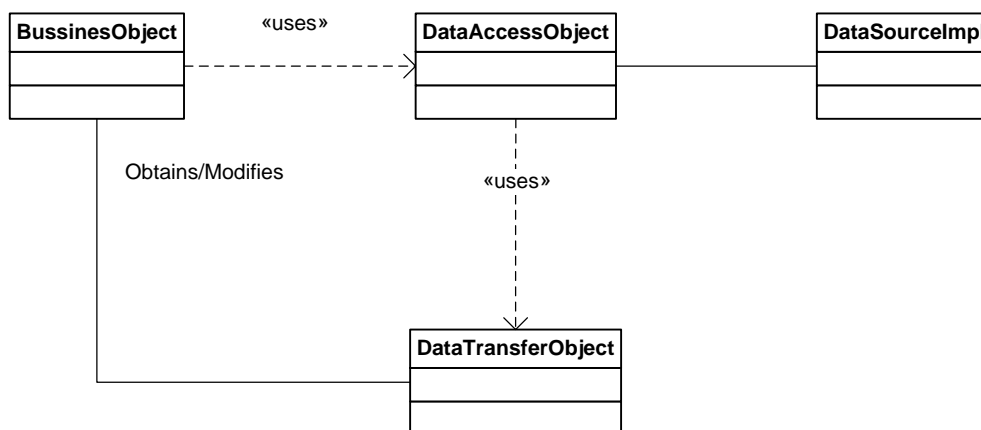
### 1. Problemática

En la mayoría de los desarrollos software de la actualidad es necesario el uso de datos persistentes en algún momento, o incluso el acceso a datos residentes en sistemas diferentes, como por ejemplo un repositorio documental o una base de datos. Es fácil comprender que esto representa un problema en los desarrollos más complejos debido a la gran dificultad de mantener el código de la aplicación en capas.

Como solución a este problema surge el patrón DAO (Data Access Object). Este patrón encapsula la forma de acceder a la fuente o fuentes de datos; DAO es el encargado de implementar el mecanismo requerido para trabajar con los datos persistentes de la aplicación. Esta implementación es completamente transparente para los clientes del DAO que se limitarán a llevar a cabo una invocación a sus servicios. Así mismo, la interfaz del DAO no debe sufrir cambio ninguno, con lo que permite a sus clientes conectarse a varias fuentes de datos (si fuese necesario) sin que se vean afectado los componentes de negocio.

### 2. Estructura

La siguiente figura muestra el diagrama de clases que representa las relaciones para el patrón DAO:



*Ilustración 1*

**BussinesObject:** Es el objeto de la lógica de negocio que quiere acceder, modificar o guardar los datos persistentes.

**DataAccessObject:** Es el objeto que encapsula todas las operaciones con los datos persistentes, abstrayendo así al **BussinesObject**.

**DataSourceImpl:** Es la implementación concreta de la interfaz del **DataAccessObject**, esta implementación es dependiente de la fuente de datos.

**DataTransferObject:** Es un objeto utilizado para el transporte de los datos desde la capa de negocio a la de persistencia.



### 3. Participantes y responsabilidad

La siguiente figura muestra el comportamiento de los integrantes del patrón:

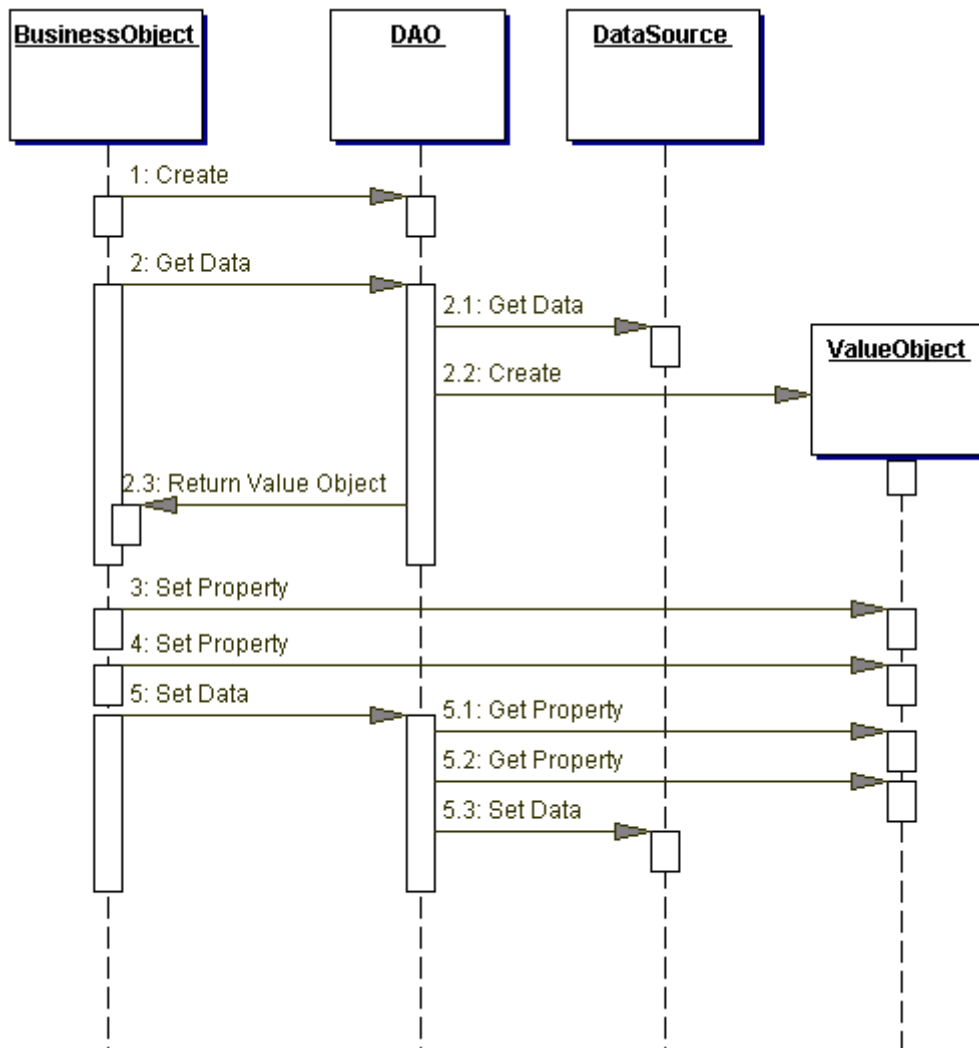


Ilustración 2

En este gráfico el TransferObject se denomina ValueObject. Puede observarse las llamadas que recibe y genera el DAO para una consulta y actualización de datos:

1. El **DAO** es creado por el cliente (**BusinessObject**).
2. A continuación, el cliente solicita los datos al **DAO**.
3. El **DAO** responde a la llamada pidiendo los datos a la fuente de datos.
4. Para cada fila recibida, el **DAO** crea un **DataTransferObject**.
5. El **DAO** devuelve al cliente el(los) **DataTransferObject**.
6. A continuación, el cliente define un **DataTransferObject** mediante llamadas a `setProperty`.
7. En `DAO.setData()` se solicita (5.1 y 5.2) al **DataTransferObject** realizar el acceso a datos (`dataSource.setData()`), (5.3).



## Experimentos

---

En la práctica anterior se detalló un problema sobre gestión de contactos. La aplicación debía permitir el alta, baja, actualización, borrado y listado de unos contactos a partir de una interfaz gráfica. En esta experimentación desarrollaremos la interconexión de las diferentes capas del patrón arquitectónico MVC así como implementar detalladamente la capa del modelo que usará MySQL para hacer los datos persistentes. Para ello siga los siguientes pasos:

1. Cree un proyecto vacío y cree la estructura de paquetes mínima.
2. Especifique e implemente la lógica de negocio de la aplicación.
3. Especifique la interfaz por cada entidad fuerte y capa de la aplicación.
4. Implemente cada una de las interfaces definidas anteriormente.
5. Implemente la capa de persistencia detallada. Para ello use el script facilitado en WebCT.

## Problemas

---

Suponga que la biblioteca de su barrio necesita llevar un control básico de préstamos de sus libros. Para ello, se requiere una base de datos que almacene información acerca de los libros (título, autor y signatura) y los usuarios (dni, nombre, teléfono, población), manteniendo un registro activo de todos y cada uno de los préstamos (libro, usuario, fecha de préstamo y fecha de devolución).

Se pide:

- Diseñar y generar dicha base de datos en MySql
- Genere la estructura MVC.
- Implemente la capa del Modelo según la siguiente descripción:
  - o Implementar en java las siguientes consultas usando el patrón DAO:
    - Libros de un autor determinado (título y signatura).
    - Libros que se encuentran prestados (título del libro y dni y nombre del usuario que lo tiene prestado).
    - Libros que llevan prestados más de un mes (título del libro y dni y nombre del usuario que lo tiene prestado).
  - o Además, se requiere que el sistema pueda introducir nuevos préstamos y renovaciones de préstamos existentes
- Implemente la capa Vista de manera que la interacción con el usuario se a través de línea de comandos.



## Datos de la Práctica

**Autor del documento:** Norberto Díaz Díaz (Mayo 2012).

**Revisiones:**

- Norberto Díaz Díaz (Mayo 2013): Revisión de formato, estimación temporal y modificación de ejercicios.

**Estimación temporal:**

- Parte presencial: 120 minutos.
  - Experimentos: 120 minutos.
- Parte no presencial: 270 minutos.
  - Lectura y estudio del guión y Bibliografía: 90 minutos
  - Problema: 180 minutos