



## Ciclo 2

# Semana 4

*Data Access Object (DAO) y Data Transfer Object (DTO),  
manejo de excepciones y documentación de código*

---

### Lectura 1 - DAO Y DTO

Semana 4

Data Access Object (DAO) y Data Transfer Object (DTO),  
manejo de excepciones y documentación de código.

## | DAO Y DTO



El patrón **DAO (DATA ACCESS OBJECT)**, en su traducción objeto de acceso a datos, es un componente de software que provee una interface entre la aplicación y sus dispositivos de almacenamiento de datos ya sean bases de datos o simples archivos.

Es un objeto cuya tarea es encapsular el acceso a la base de datos, diferenciando la interface cliente de los mecanismos de acceso, normalmente a través de **DTO (DATA TRANSFER OBJECT)** en su traducción **Objeto de Transferencia de Datos**.

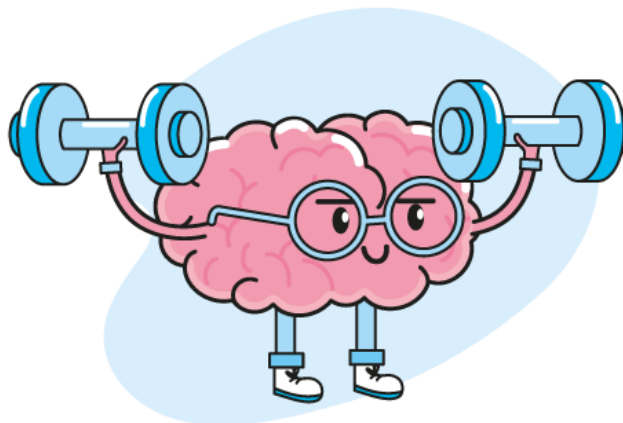
Al utilizar DAO como objetos de acceso a datos, permite que los otros objetos, los propios de la aplicación y que contienen los detalles de la operación, solo se encarguen de ejercer su labor, sin necesitar conocimiento del destino de la información manipulada. En java se usan también para apartar la aplicación de la API de persistencia Java haciendo que esta tecnología de persistencia subyacente se actualice o modifique sin cambiar las otras partes de la aplicación.

¿Pero qué es persistencia?, se conoce como persistencia a aquella operación que se realiza para resguardar la información de un objeto permanentemente y también de recuperar su información para que esta sea reutilizada.



## Semana 4

Data Access Object (DAO) y Data Transfer Object (DTO),  
manejo de excepciones y documentación de código.



En las aplicaciones, normalmente se suele hacer persistencia de datos en algún punto, ya sea serializando la información o almacenándola en bases de datos relacionales o ficheros XML, entre otros, para lo cual la aplicación debe interactuar con la base de datos. Esta interacción no debe corresponder a la capa lógica de negocio sino a la capa de persistencia, encargada de interactuar con la base de datos. Teniendo en cuenta esto se puede decir que un DAO es un patrón de diseño que se usa para crear esta capa de persistencia.

El DAO encapsula el acceso a la base de datos y gestiona la conexión que se tiene, para obtener la información solicitada a través de la API que da la DAO, así si se cambia la base de datos de almacenamiento se modificaría solo la capa DAO. Esta API generalmente corresponde a métodos Create, Read, Update y Delete (CRUD). DAO se encargará de crear, leer, actualizar o eliminar un dato, esto independiente de la capa lógica del método, ya que es solo tarea de DAO la implementación del método que corresponda y su interacción con el almacenamiento, sea este base de datos, fichero, etc.

El DAO se relaciona normalmente con los patrones transfer object, en donde la información que se envía o recibe del DAO se empaqueta en objetos, y el patrón Factory que instancia los DAOs para tener independencia del almacén de datos.

Entonces, el DAO da el acceso a los datos a través de métodos definidos en el DAO, dando una abstracción al modelo de datos, pero existe el DTO que son objetos que transporta los datos desde un origen en una o más entidades de información y son incorporados a una instancia para su acceso.

El DTO no ofrece la lógica del negocio ni hace validaciones, solo provee acceso a las propiedades del BEAN, entendiendo este término como una clase que almacena datos, encapsulando la información. Es necesario definir un DTO por cada bean de la aplicación. Este modelo aparta el

**Semana 4**

Data Access Object (DAO) y Data Transfer Object (DTO),  
manejo de excepciones y documentación de código.

controlador del modelo, por tanto, si se requiere un cambio en la forma de acceso a los datos, el modelo no se modifica pues recibe el mismo DTO.

Un transfer object es un objeto que empaqueta datos para que viajen entre capas y tendrá todos los datos que nos interesen de forma accesible. Se debe tener en cuenta que, aunque los transfer objects se relacionan directamente con objetos del modelo, ellos no son de los mismos objetos, ya que los objetos del dominio contienen lógica de negocio, pero los transfer objects solo son almacenes de datos.