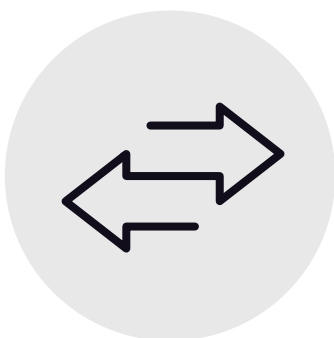


# DTOs VS DAOs

**Data Transfer Object:** los DTO son objetos utilizados para encapsular datos y transferirlos entre procesos de la aplicación, estos suelen ser clases simples que contienen propiedades que representan los datos que se desean transferir entre diferentes partes de la aplicación. Los DTO suelen contener únicamente los datos necesarios para una operación específica, sin incluir lógica de negocio.

## Casos de uso:



Transmitir datos entre el frontend y el backend de una aplicación.



Evitar el acoplamiento excesivo entre las capas de una aplicación al transferir datos de una capa a otra.



Validar los datos antes de ser transferidos para garantizar su integridad.

## Ejemplo:

```
class UserDTO {
  constructor(
    public id: number,
    public name: string,
    public email: string
  ) {}
}
```

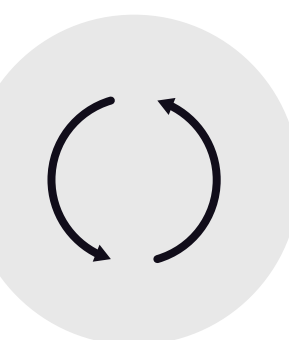


**Data Access Object:** los DAO son un patrón que provee una interfaz para encapsular el acceso a una fuente de datos, como una base de datos o una API. De esta manera provee métodos para realizar operaciones (CRUD) sin exponer los detalles de la fuente de datos y a su vez manteniendo separando la logica de negocio.

## Casos de uso:



Abstraer la lógica de acceso a la fuente de datos para facilitar el mantenimiento y la escalabilidad de la aplicación.



Facilita la reutilización del código relacionado con el acceso a la fuente de datos.



Cada entidad del dominio tiene su propio DAO para gestionar su acceso a la fuente de datos.

## Ejemplo:

```
import { UserDTO } from './userDTO';
class UserDAO {
  getUsers() : UserDTO[] {
    ...
  }
  getUserById(id : number) : UserDTO {
    ...
  }
  createUser(user : UserDTO) : void {
    ...
  }
  //Other methods
}
```



Links de referencia

1

2



Cursos relacionados



Más recursos

Síguenos:



/Fernando\_Her85



/DevTalles



/DevTalles

www.devtales.com

