

# Data Transfer Object (DTO)

Aplicaciones Distribuidas

Curso 2025/2026

# ¿Qué es un DTO?

- Data Transfer Object (Objeto de Transferencia de Datos).
- Es un objeto simple que contiene datos y es usado para transferir información entre distintas capas de una aplicación.
- Normalmente se usa en sistemas multicapa como aplicaciones web o aplicaciones empresariales.

# Características clave de un DTO

- Reducen la cantidad de datos transferidos entre capas.
- Permiten que solo los datos necesarios sean accesibles en una capa específica, mejorando la seguridad y privacidad.
- Mejoran la eficiencia de las consultas al incluir solo los datos relevantes para una tarea específica.
- Contienen solo datos: no tienen lógica de negocio.
- Son inmutables o tienen solo métodos getter y setter.
- A menudo son serializables para facilitar su transmisión, especialmente en aplicaciones distribuidas.

## ¿Por qué usar DTOs?

- Seguridad: Expone solo los datos necesarios.
- Reducción de datos: Menos información se transfiere entre capas.
- Separación de capas: Desacopla la capa de presentación y la capa de datos.
- Control sobre el formato de datos: Permite estructurar y validar los datos que viajan entre las capas.

# Cuándo usar DTOs en un proyecto

- Aplicaciones multicapa: Cuando una aplicación tiene capas de datos, negocio y presentación separadas.
- Aplicaciones distribuidas: Como servicios REST, donde se necesita controlar la información que se envía a los clientes.
- Optimización de transferencias: En escenarios donde el rendimiento es clave y se deben enviar solo los datos esenciales.

# Implementación simple de DTO

Entidad: `UserEntity`

```
public class UserEntity {  
    private Long id;  
    private String username;  
    private String email;  
    private String password;  
    // Constructor, Getters y Setters  
}  
  
public class UserDTO {  
    private Long id;  
    private String username;  
    private String email;  
    // Constructor, Getters y Setters  
}
```

# Métodos de conversión en la capa de servicio

```
public class UserService {  
    public UserDTO convertToDto(UserEntity userEntity) {  
        UserDTO dto = new UserDTO(userEntity.getId(),  
            userEntity.getUsername(),  
            userEntity.getEmail());  
        return dto;  
    }  
}
```

# Conversión por JPQL

- El mapeo puede ser manual, como en el ejemplo anterior, usar una ayuda como [MapStruct](#) o si las entidades se obtienen con JPQL, se puede hacer la conversión directamente en la consulta:

```
String queryString =  
"SELECT new um.aadd.UserDTO(u.id,  
u.username, u.email)  
FROM UserEntity u ";
```

[Artículo interesante de Medium](#)