

## EXAMEN 1EVAL DWS

Dada la bbdd adjunta, realizar los siguientes endpoints con la arquitectura vista en clase:

- **GET localhost:8080/api/orders (5 pts)**

Listado de todos los pedidos. La información de cada pedido será:

```
{  
  "id": 1,  
  "orderDate": "2023-11-30",  
  "deliveryDate": "2023-12-05",  
  "status": "DELIVERED",  
  "total": 85.50  
}
```

- **GET localhost:8080/api/orders/{id} (5 pts)**

Detalle de un pedido, con la siguiente información (**la imagen no tiene porqué coincidir con los datos reales**):

```
{  
  "id": 5,  
  "orderDate": "2024-03-20",  
  "deliveryDate": "2024-03-25",  
  "status": "DELIVERED",  
  "total": 61.40,  
  "user": {  
    "name": "Ana Martínez Sánchez",  
    "address": "Calle User2",  
    "admin": false  
  },  
  "orderDetails": [  
    {  
      "quantity": 1,  
      "price": 11.50,  
      "book": {  
        "id": 3,  
        "isbn": "9786074213485",  
        "title": "La insorportable levedad del ser"  
      }  
    },  
  ],  
}
```

## EXAMEN

**MÒDUL: DWS**  
**NOM:**

**DATA: 04/12/2024**

```
{
  "quantity": 2,
  "price": 20.30,
  "book": {
    "id": 4,
    "isbn": "9788466338141",
    "title": "La isla del día de antes"
  }
},
{
  "quantity": 1,
  "price": 9.30,
  "book": {
    "id": 5,
    "isbn": "9788448022440",
    "title": "Buenos presagios"
  }
}
]
```

- **POST localhost:8080/api/orders/{idOrder}/books/{idBook} (1 pt)**

Insertar un libro a un pedido. La información que se le pasará será la cantidad y el precio:

```
{
  "quantity": 2,
  "price": 23.40
}
```

### Notas:

- El campo status en la bbdd es de tipo numérico. La correspondencia será:
  - 0 → Pending
  - 1 → Actuated
  - 2 → Cancelled
  - 3 → In transit
  - 4 → Delivered
- No hace falta implementar los métodos que no se vayan a usar (por ejemplo, el método save() de OrderRepository no implementar el insert).
- Podéis crear los modelos de dominio o presentación como queráis (siempre que tenga lógica). No hace falta que penséis en funcionalidad futura.