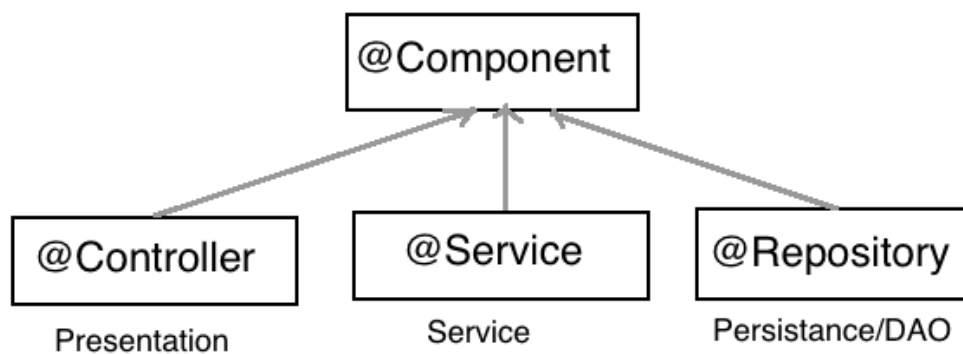


AD-4. Microservicios web con Spring Boot



Grupo 11:

Silvia Piñel Fañanás

Benjamin Ruiz de Casas

Enunciado

Una Empresa tiene recogido en una base de datos los pedidos solicitados por Clientes y el Comercial asociado a la venta. La tabla pedidos recoge el importe total del pedido realizado.

La aplicación la van a usar los jefes de comerciales para ver la información, tanto de clientes como de sus comerciales.

Se adjunta el script del sql (en la unidad formativa UF6) con la creación de la base de datos (ventasbbdd_2023) de las tablas, insert para pruebas y creación del usuario uventas_2023 con la password uventas.

Ejercicio

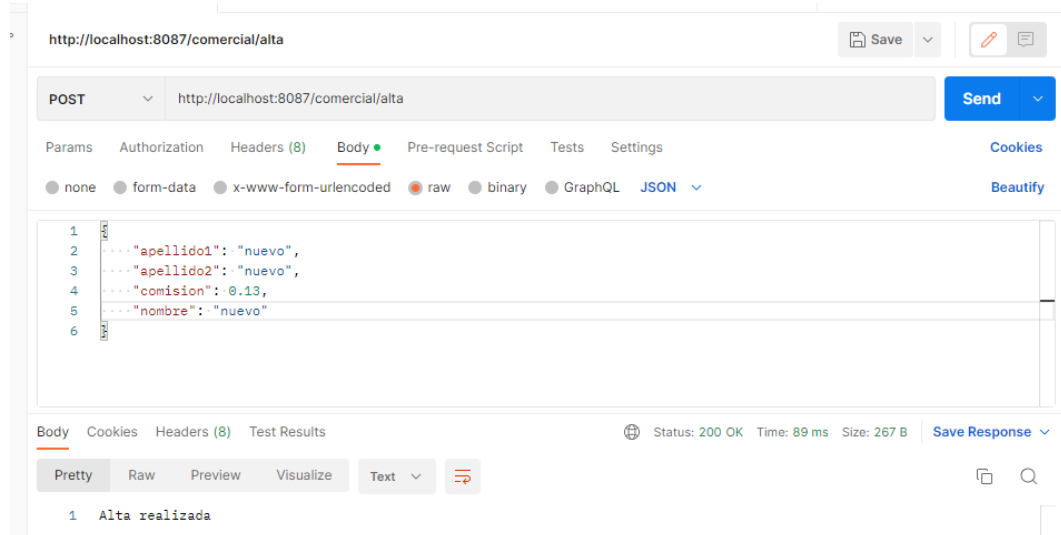
Crear un proyecto Spring web, con Spring data Jpa y MySql 8 para los siguientes servicios web restfull, y probar con postman:

url	Descripción
/comercial	@RequestMapping("/comercial")
/alta	Dar de alta el comercial
/eliminar/{id}	Eliminar de la bbdd el comercial cuyo id coincida
/uno/{id}	Devolver los datos del comercial cuyo id coincida
/bycliente/{id}	Devolver la lista de los comerciales que han atendido pedidos del cliente que coincida con ese id.
/conpedidos	Devolver la lista de comerciales que han atendido algún pedido
/pedidos/{id}	Devolver la lista de pedidos gestionados por el comercial que coincida con ese id.

/alta – Dar de alta un comercial

Prueba con Postman de la funcionalidad alta de un comercial, en este caso damos de alta el comercial “nuevo”.

Adjuntamos captura de la base de datos donde se ha creado el comercial, con el id_comercial 9.



The screenshot shows a database viewer displaying the 'comerciales' table. The table has 9 rows, including a new entry with id_comercial 9. Below the table, the output of a SQL query is shown.

	id_comercial	nombre	apellido1	apellido2	comision
	4	Marta	Herrera	Gil	0.14
	5	Antonio	Carretero	Ortega	0.12
	6	Manuel	Domínguez	Hernández	0.13
	7	Antonio	Vega	Hernández	0.11
	8	Alfredo	Ruiz	Flores	0.05
	9	nuevo	nuevo	nuevo	0.13
*	NULL	NULL	NULL	NULL	NULL

comerciales 1 x

Output

Action Output

#	Time	Action	Message
1	16:58:28	SELECT * FROM ventasbdd_2023.comerciales LIMIT 0, 1000	9 row(s) returned

AD-4. Microservicios web con Spring Boot

/eliminar/{id} Eliminar de la BBDD el comercial cuyo id coincida con el dado por parámetro.

Prueba con Postman de la funcionalidad eliminar un comercial, en este caso eliminamos el comercial creado en la prueba anterior.

Adjuntamos captura de la base de datos donde se ha eliminado el comercial, con el id_comercial 9.

The image shows two screenshots. The top screenshot is from Postman, showing a DELETE request to `http://localhost:8087/comercial/eliminar/9`. The response status is 200 OK, and the body contains the message: "Se ha eliminado el comercial correctamente".

The bottom screenshot is from a database client showing a table named `comerciales` with the following data:

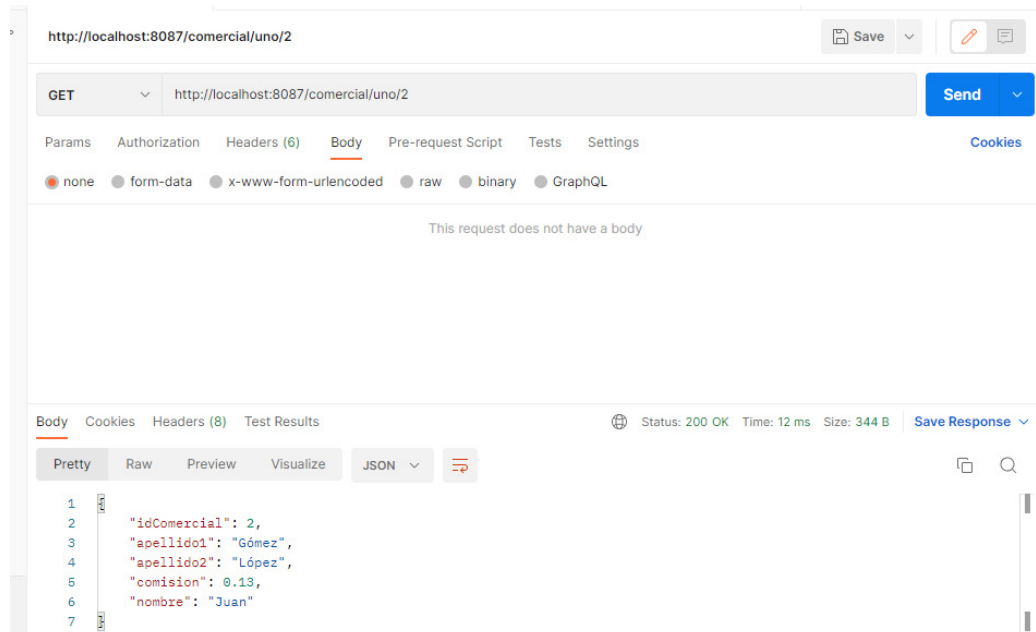
id_comercial	nombre	apellido1	apellido2	comision
3	Diego	Flores	Salas	0.11
4	Marta	Herrera	Gil	0.14
5	Antonio	Carretero	Ortega	0.12
6	Manuel	Dominguez	Hernández	0.13
7	Antonio	Vega	Hernández	0.11
8	Alfredo	Ruiz	Flores	0.05
9	NULL	NULL	NULL	NULL

Below the table, the database client shows the output of a SQL query: `SELECT * FROM ventasbdd_2023.comerciales LIMIT 0, 1000`, which returned 9 rows.

AD-4. Microservicios web con Spring Boot

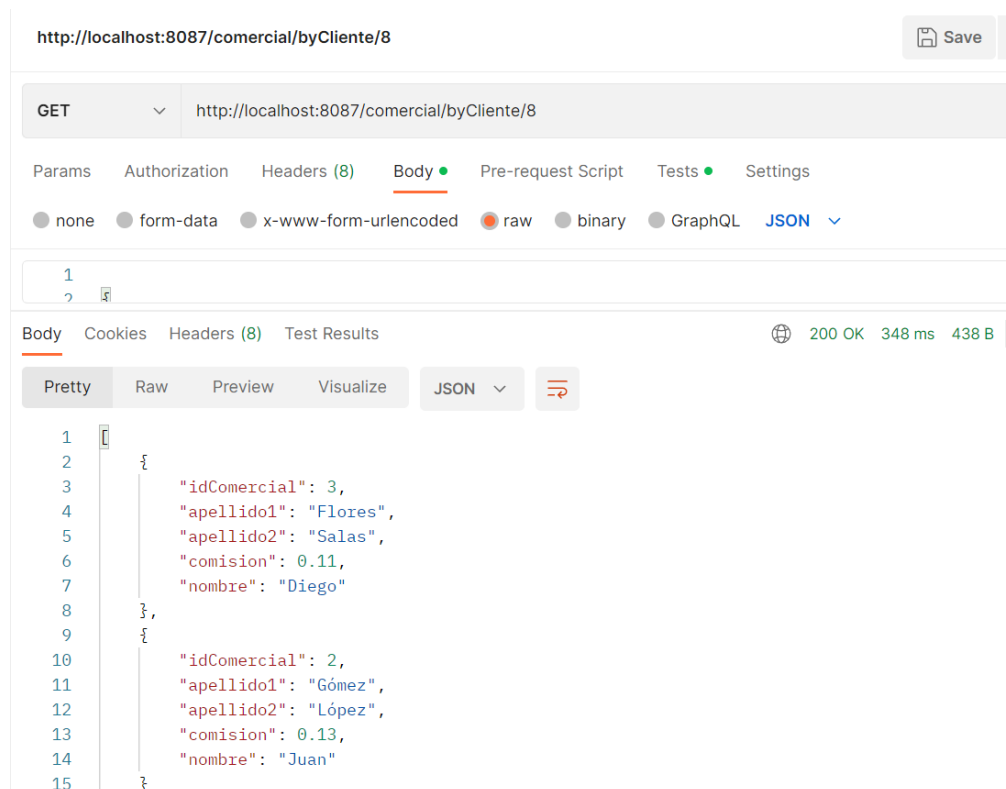
/uno/{id} Devolver los datos del comercial cuyo id coincida con el que pasamos por parámetro.

Prueba con Postman de la funcionalidad encontrar a un comercial según su id_comercial.



/bycliente/{id} Devolver la lista de los comerciales que han atendido pedidos del cliente que coincida con ese id

Prueba con Postman de la funcionalidad de mostrar los comerciales que tienen pedido con un determinado cliente, en este ejemplo hemos probado con el Id_cliente 8.



AD-4. Microservicios web con Spring Boot

Si lo vemos en sql:

```
1 select * from comerciales c inner join pedidos p where c.id_comercial=p.id_comercial and p.id_cliente=8 group by c.id_comercial;
```

id_comercial	nombre	apellido1	apellido2	comision	id_pedido	total	fecha	id_cliente	id_comercial
3	Diego	Flores	Salas	0.11	4	110.5	2016-08-17	8	3
2	Juan	Gómez	López	0.13	10	250.45	2015-06-27	8	2

/conpedidos Devolver la lista de comerciales que han atendido algún pedido

Prueba con Postman de la funcionalidad de mostrar los comerciales que han atendido algún pedido.

http://localhost:8087/comercial/conpedidos

GET http://localhost:8087/comercial/conpedidos

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Body Cookies Headers (8) Test Results 200 OK 15 ms

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "idComercial": 2,
4     "apellido1": "Gómez",
5     "apellido2": "López",
6     "comision": 0.13,
7     "nombre": "Juan"
8   },
9   {
10    "idComercial": 5,
11    "apellido1": "Carretero",
12    "apellido2": "Ortega",
13    "comision": 0.12,
14    "nombre": "Antonio"
15  },
16  {
17    "idComercial": 1,
18    "apellido1": "Sáez",
19    "apellido2": "Vega",
20    "comision": 0.15,
21    "nombre": "Daniel"
22  },
23  {
24    "idComercial": 3,
25    "apellido1": "Flores",
26    "apellido2": "Salas",
27    "comision": 0.11,
28    "nombre": "Diego"
29  },
30 ]
```

AD-4. Microservicios web con Spring Boot

```
30     {
31         "idComercial": 6,
32         "apellido1": "Dominguez",
33         "apellido2": "Hernández",
34         "comision": 0.13,
35         "nombre": "Manuel"
36     },
37     {
38         "idComercial": 7,
39         "apellido1": "Vega",
40         "apellido2": "Hernández",
41         "comision": 0.11,
42         "nombre": "Antonio"
43     }
44 }
```

Ahora lo vemos en sql:

The screenshot shows a SQL query: `1 • select * from comerciales c inner join pedidos p where c.id_comercial=p.id_comercial group by c.id_comercial`. Below the query, a table displays the results of the join operation, grouped by commercial ID.

	id_comercial	nombre	apellido1	apellido2	comision	id_pedido	total	fecha	id_cliente	id_comercial
2	Juan	Gómez	López		0.13	1	150.5	2017-10-05	5	2
5	Antonio	Carretero	Ortega		0.12	2	270.65	2016-09-10	1	5
1	Daniel	Sáez	Vega		0.15	3	65.26	2017-10-05	2	1
3	Diego	Flores	Salas		0.11	4	110.5	2016-08-17	8	3
6	Manuel	Dominguez	Hernández		0.13	8	1983.43	2017-10-10	4	6
7	Antonio	Vega	Hernández		0.11	11	75.29	2016-08-17	3	7

/pedidos/{id} Devolver la lista de pedidos gestionados por el comercial que coincida con ese id

Prueba con Postman de la funcionalidad para obtener un listado de pedidos gestionados por un comercial, según su `id_comercial` que pasamos por parámetro.

The screenshot shows a Postman interface with a GET request to `http://localhost:8087/pedidos/2`. The response is a JSON object with the following structure:

```
1 {
2   "idPedido": 1,
3   "fecha": "2017-10-05",
4   "total": 150.5,
5   "cliente": {
6     "idCliente": 5,
7     "apellido1": "Loyola",
8     "apellido2": "Méndez",
9     "categoria": 200,
10    "ciudad": "Almería",
11    "nombre": "Marcos"
12  },
13  "comerciales": {
14    "idComercial": 2,
15    "apellido1": "Gómez",
16    "apellido2": "López",
17    "comision": 0.13,
18  }
```