|  |
| --- |
| **AD-4. Tarea en equipo. Microservicios web con Spring Boot** |

|  |
| --- |
|  |

|  |  |
| --- | --- |
| GRUPO 12  Jose San Pastor  David Menéndez  Raúl Moreno  Github: <https://github.com/Davasda/AD-4-GRUPO> |  |

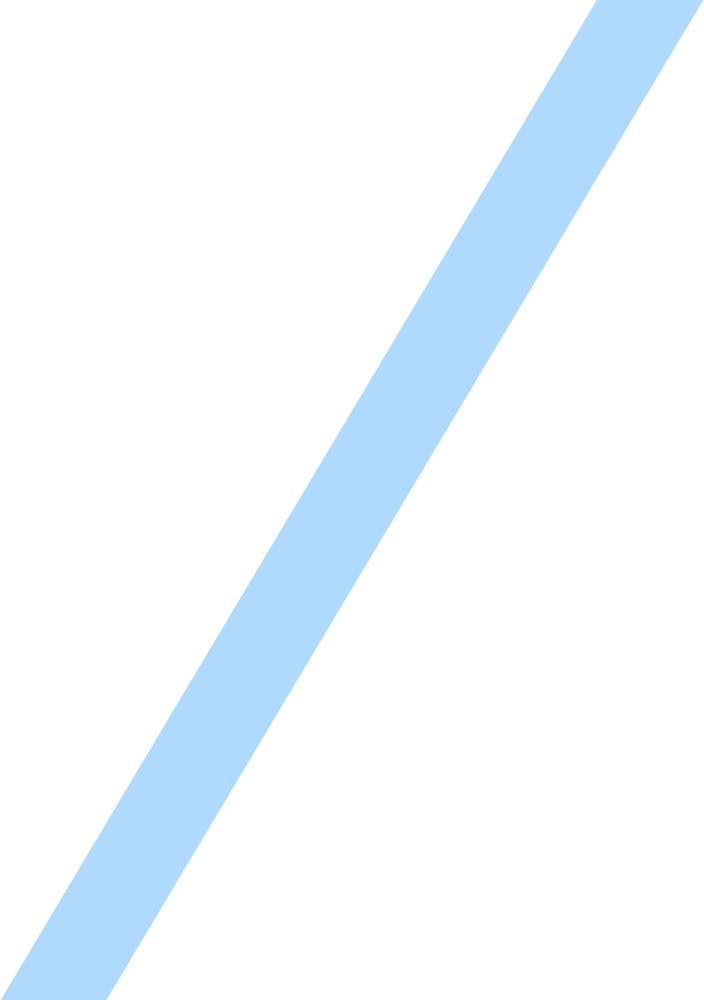


Tabla de contenido

[ENUNCIADO 2](#_Toc126407202)

[EJERCICIO A REALIZAR 3](#_Toc126407203)

[RESULTADOS 4](#_Toc126407204)

[1. Crear @RequestMapping(“/comercial”) 4](#_Toc126407205)

[2. /alta para Dar de alta el comercial 5](#_Toc126407206)

[3. /eliminar Eliminar de la bbdd el comercial cuyo id coincida. 7](#_Toc126407207)

[4. /uno/{id} Devolver los datos del comercial cuyo id coincida 9](#_Toc126407208)

[5. /bycliente/{id} Devolver la lista de los comerciales que han atendido pedidos del cliente que coincida con ese id. 11](#_Toc126407209)

[6. /conpedidos Devolver la lista de los comerciales que han atendido algún pedido 13](#_Toc126407210)

[7. /pedidos/{id} Devolver la lista de pedidos gestionados por el comercial que coincida con ese id. 15](#_Toc126407211)

|  |
| --- |
| 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.  Tablas:  Interfaz de usuario gráfica, Diagrama  Descripción generada automáticamente  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. | | |
|  | | |
| 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. | |  |  |

|  |  |
| --- | --- |
| EJERCICIO A REALIZAR |  |

|  |  |
| --- | --- |
| RESULTADOS |  |

Estructura del proyecto:

Texto

Descripción generada automáticamente

La aplicación se ejecuta en puerto 8088: <http://localhost:8088>

## Crear @RequestMapping(“/comercial”)

Se crea dentro de ComercialRestController.java el correspondiente RequestMapping

|  |  |
| --- | --- |
|  |  |

## /alta para Dar de alta el comercial

* 1. Código realizado
     1. En el Interface ComercialService (no volvemos a poner este código en este pdf ya que se usa para todos los métodos de comercial)

Texto

Descripción generada automáticamente

* + 1. En la clase ComercialServiceImplMy8

Texto

Descripción generada automáticamente

* + 1. En la clase controlador ComercialRestController

Texto

Descripción generada automáticamente

* 1. Resultado /alta

|  |  |
| --- | --- |
| Resultado Postman | Resultado Mysql |
|  |  |

## /eliminar Eliminar de la bbdd el comercial cuyo id coincida.

* 1. En la clase ComercialServiceImplMy8

Texto

Descripción generada automáticamente

* 1. En la clase controlador ComercialRestController

Texto

Descripción generada automáticamente

* 1. Resultado /eliminar

|  |  |
| --- | --- |
| Resultado postman | Resultado Mysql |
|  |  |

## /uno/{id} Devolver los datos del comercial cuyo id coincida

* 1. En la clase ComercialServiceImplMy8

Texto

Descripción generada automáticamente

* 1. En la clase controlador ComercialRestController

Texto

Descripción generada automáticamente

* 1. Resultado /uno/{id}

|  |
| --- |
| Resultado Postman |

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

* 1. En en interface ComercialRespository hemos creado una Query con nativeQuery.

Texto

Descripción generada automáticamente

* 1. En la clase ComercialServiceImplMy8

Sitio web

Descripción generada automáticamente con confianza baja

* 1. En la clase ComercialRestController

Texto

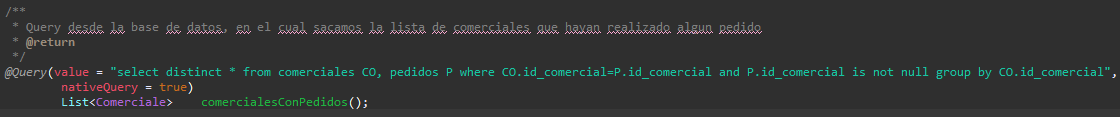
Descripción generada automáticamente

* 1. Resultado /bycliente/{id}

|  |
| --- |
| Resultado Postman  Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente |

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

* 1. En en interface ComercialRespository hemos creado una Query con nativeQuery.



* 1. En la clase ComercialServiceImplMy8

Texto

Descripción generada automáticamente

* 1. En la clase ComercialRestController

Texto

Descripción generada automáticamente

* 1. Resultado /conpedidos

|  |  |
| --- | --- |
| Resultado Postman | Resultado Texto Json  [      {          "idComercial": 2,          "apellido1": "Gómez",          "apellido2": "López",          "comision": 0.13,          "nombre": "Juan"      },      {          "idComercial": 5,          "apellido1": "Carretero",          "apellido2": "Ortega",          "comision": 0.12,          "nombre": "Antonio"      },      {          "idComercial": 1,          "apellido1": "Sáez",          "apellido2": "Vega",          "comision": 0.15,          "nombre": "Daniel"      },      {          "idComercial": 3,          "apellido1": "Flores",          "apellido2": "Salas",          "comision": 0.11,          "nombre": "Diego"      },      {          "idComercial": 6,          "apellido1": "Domínguez",          "apellido2": "Hernández",          "comision": 0.13,          "nombre": "Manuel"      },      {          "idComercial": 7,          "apellido1": "Vega",          "apellido2": "Hernández",          "comision": 0.11,          "nombre": "Antonio"      }  ] |

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

Para este método teníamos dudas de si crear un Restcontroller de pedidos, ya que lo que realmente nos devuelve es una lista de pedidos y no una lista de comerciales ,por lo que que requestmapping de comerciales quizá no sería el sitio.

Como la actividad tiene claramente el requestmapping de comercial lo hemos dejado ahí aunque si hemos creado los servicios y repositorios de pedidos para que se haga la consulta y métodos y luego sea el controlador del comercial quien lo llame.

* 1. En en interface PedidoRepository hemos creado una Query con nativeQuery.

Texto

Descripción generada automáticamente

* 1. En el interface PedidoService

Texto

Descripción generada automáticamente

* 1. En la clase PedidolServiceImplMy8

Texto

Descripción generada automáticamente

* 1. En la clase ComercialRestController

Texto

Descripción generada automáticamente

* 1. Resultado /pedidos/{id}

|  |  |
| --- | --- |
| Resultado Postman | Resultado Texto Json  [      {          "idPedido": 4,          "fecha": "2016-08-17",          "total": 110.5,          "cliente": {              "idCliente": 8,              "apellido1": "Ruiz",              "apellido2": "Santana",              "categoria": 200,              "ciudad": "Huelva",              "nombre": "Pepe"          },          "comerciale": {              "idComercial": 3,              "apellido1": "Flores",              "apellido2": "Salas",              "comision": 0.11,              "nombre": "Diego"          }      },      {          "idPedido": 9,          "fecha": "2016-10-10",          "total": 2480.4,          "cliente": {              "idCliente": 8,              "apellido1": "Ruiz",              "apellido2": "Santana",              "categoria": 200,              "ciudad": "Huelva",              "nombre": "Pepe"          },          "comerciale": {              "idComercial": 3,              "apellido1": "Flores",              "apellido2": "Salas",              "comision": 0.11,              "nombre": "Diego"          }      }  ] |