

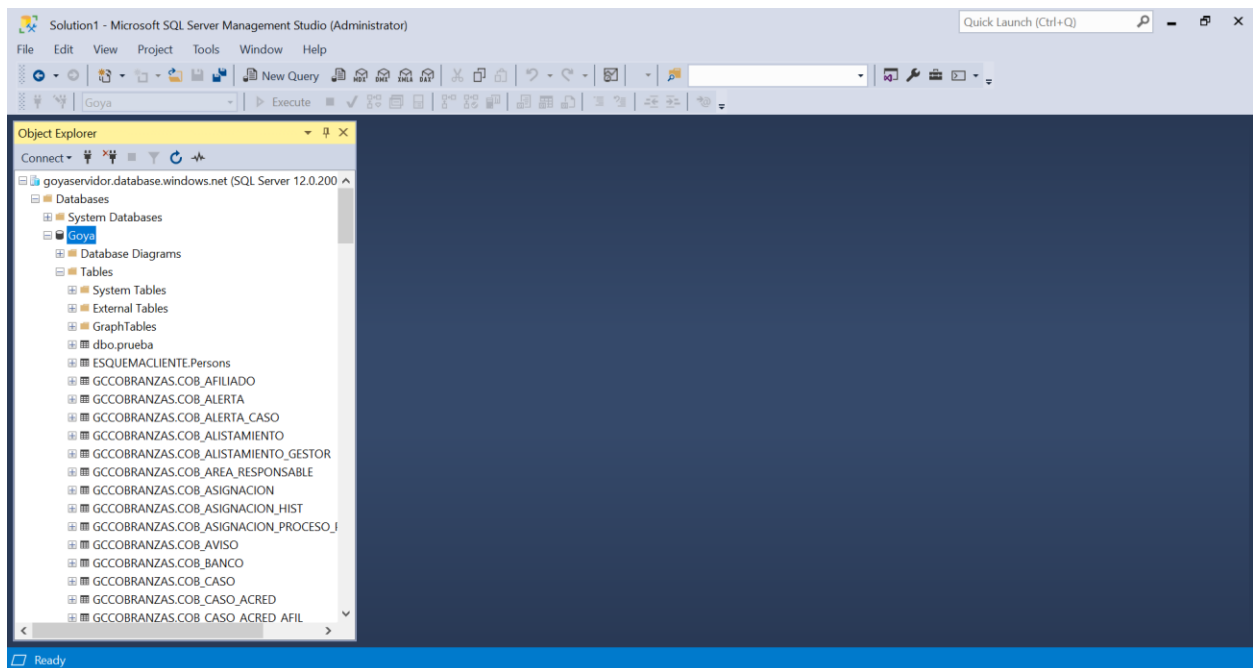
Manual del Reto técnico Desarrollo de un Microservicio en Java – Springboot

Yeison Chacón Fúquene

28/12/2021

1. Ingresar a la base de datos sql server con el siguiente usuario y contraseña

Para la primer parte del reto instalé en mi máquina Microsoft SQL Server Management Studio 18 e ingresé a la base de datos remotamente con los datos suministrados de host, usuario, password y database.



2. Consulta la tabla GCCOBRANZAS. COB_GESTOR y GCINTEGRIDAD.GEN_LOCALIDAD

Para la siguiente parte del reto explore las tablas para ver cuántos registros tenía cada una, los campos y tipos de datos que las mismas manejan

USUARIO...	PRIMER...	SEGUND...	PRIMER...	SEGUND...	TARJETA...	DIRECCI...	TELEFO...	CELULAR	CORREO...	APROBA...	DIRECT...	ACTIVO
insertes...	Diana		Cardenas	Gomez	16368413	Calle 100...	1234567	3151234...	diana@p...	NULL	NULL	S
NuevoGe...	Diana		Cardenas	Gomez	16368413	Calle 100...	1234567	3151234...	diana@p...	NULL	NULL	S
NuevoGe...	Diana		Cardenas	Gomez	16368413	Calle 100...	1234567	3151234...	diana@p...	NULL	NULL	S
NuevoGe...	Diana		Cardenas	Gomez	16368413	Calle 100...	1234567	3151234...	diana@p...	NULL	NULL	S
NuevoGe...	Diana		Cardenas	Gomez	16368413	Calle 100...	1234567	3151234...	diana@p...	NULL	NULL	S
NuevoGe...	Diana		Cardenas	Gomez	16368413	Calle 100...	1234567	3151234...	diana@p...	NULL	NULL	S
NuevoGe...	Diana		Cardenas	Gomez	16368413	Calle 100...	1234567	3151234...	diana@p...	NULL	NULL	S
POR0001	Diana		Cardenas	Gomez	16368413	Calle 100...	1234567	3151234...	diana@p...	NULL	NULL	S
POR0002	Daniel	Leonardo	Gonzales	Perez	7894561...	Calle 93 ...	7894561	3154789...	daniel@p...	NULL	NULL	S
POR0003	Luis	Antonio	Luis	Ruiz	3577535	Calle 89 ...	3577535	3157894...	luis@por...	NULL	NULL	S
POR0004	Andres	Eduardo	Contreras	Valderra...	1525252	Apartado...	0558342...	0142544...	nonum...	'NULL'	'NULL'	S
POR0005	Camila		Maldona...	Everett	3525252		3525252	0921708...	Vestibul...	'NULL'	'NULL'	S
POR0006	Julian	Andres	Gomez	Suarez	5525252	797-430...	5525252	0561412...	Aenean...	'NULL'	'NULL'	S
POR0007	Ricardo	Ruiz	Combita	Valentine	7525252	5779 Elit...	0601538...	0160771...	Sed@ma...	'NULL'	'NULL'	S
POR0008	Jorge	Quintero	Laurens	Mckay	9525252	5134 Pla...	0879906...	0893462...	ut.lacus...	'NULL'	'NULL'	S
POR0009	Gray	Dolan	Barton	Rollins	11525252	973-569...	0741111...	0532467...	mauris.l...	'NULL'	'NULL'	S
POR0010	Victor	Lewis	Howell	Campos	13525252	9318 Et. C/	0447370...	0893785...	dolor.Qui...	'NULL'	'NULL'	S
POR0011	Hector	Sawyer	Wilson	Small	15525252	8381 Fa...	0770688...	0521353...	malesua...	'NULL'	'NULL'	S
POR0011...	Mario	Alfonso	Bros	Bros	12313132	Cra 123 ...	12313132	3112001...	marioalf...	NULL	NULL	S
POR0012	Martin	Silas	McLadden	Pena	17525252	Apdo.:51...	0784889...	0943203...	ullamco...	'NULL'	'NULL'	S
POR00123	Maria	Fernanda	Cagliostro	Gonzalez	4564564	Cra 123	4564564	3201232...	mariafer...	NULL	NULL	S
POR0013	Fulton	Amos	Ellis	Chapman	19525252	4089 Pha...	0210405...	0360266...	at.auctor...	'NULL'	'NULL'	S
POR0014	Nancy	Adriana	Rodriguez	Cacer	7894561	Calle 78...	6546454	3152456	adriana@...	NULL	NULL	S

LOCALID...	CODIGO...	NOMBRE	DESCRIP...	TIPO	LOCALID...	REGISTR...	USUARI...	FECHA...	USUARI...	FECHA...
1	00	Colombia	Colombia	PAIS	0	S	carcastr	2020-10...	carcastr	2020-10...
2	05	ANTIOQ...	ANTIOQ...	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
3	08	ATLANTI...	ATLANTI...	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
4	11	BOGOTA	BOGOTA	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
5	13	BOLIVAR	BOLIVAR	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
6	15	BOYACA	BOYACA	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
7	17	CALDAS	CALDAS	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
8	18	CAQUETA	CAQUETA	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
9	19	CAUCA	CAUCA	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
10	20	CESAR	CESAR	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
11	23	CORDOBA	CORDOBA	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
12	25	CUNDIN...	CUNDIN...	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
13	27	CHOCO	CHOCO	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
14	41	HUILA	HUILA	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
15	44	LA GUAJ...	LA GUAJ...	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
16	47	MAGDAL...	MAGDAL...	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
17	50	META	META	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
18	52	NARIÑO	NARIÑO	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
19	54	N. DE SA...	N. DE SA...	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
20	63	QUINDIO	QUINDIO	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
21	66	RISARAL...	RISARAL...	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
22	68	SANTAN...	SANTAN...	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...
23	70	SUCRE	SUCRE	DEPARTA...	1	S	carcastr	2020-10...	carcastr	2020-10...

3. Construya un servicio rest/json en spring boot que consulte esta tabla y traiga los nombre de los gestores concatenados y que cuente cuantos ciudades hay en la tablas. El método debe ser POST (puede definir el servicio como usted lo considere)

Para esta etapa del reto utilicé JAVA 11 con Springboot con el gestor de dependencias Maven y algunas dependencias como: Spring Data JPA, MS SQL Server Driver entre otras.

En la primera parte del reto (Listado de Gestores) establecí la comunicación del proyecto con la base de datos a través del archivo application.properties generado del proyecto.

```

1 spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver
2 spring.datasource.url=jdbc:sqlserver://goyaservidor.database.windows.net;databaseName=Goya
3 spring.datasource.username=pruebas2
4 spring.datasource.password=1234567A%
5 spring.jpa.show-sql=true
6 spring.jpa.properties.hibernate.format_sql = true
7
8 ## Hibernate Properties
9 # The SQL dialect makes Hibernate generate better SQL for the chosen database
10 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.SQLServer2012Dialect
11
12 # Hibernate ddl auto (create, create-drop, validate, update)
13 spring.jpa.hibernate.ddl-auto = update
14

```

The screenshot shows the IDE's Project Explorer on the left with the 'application.properties' file selected under 'resources'. The main editor displays the configuration for the database connection and Hibernate settings. The Run console at the bottom shows the application running successfully.

Una vez hecho esto construí el modelo y el controlador para solucionar la primera parte del reto, cree una clase por cada uno

Modelo

```

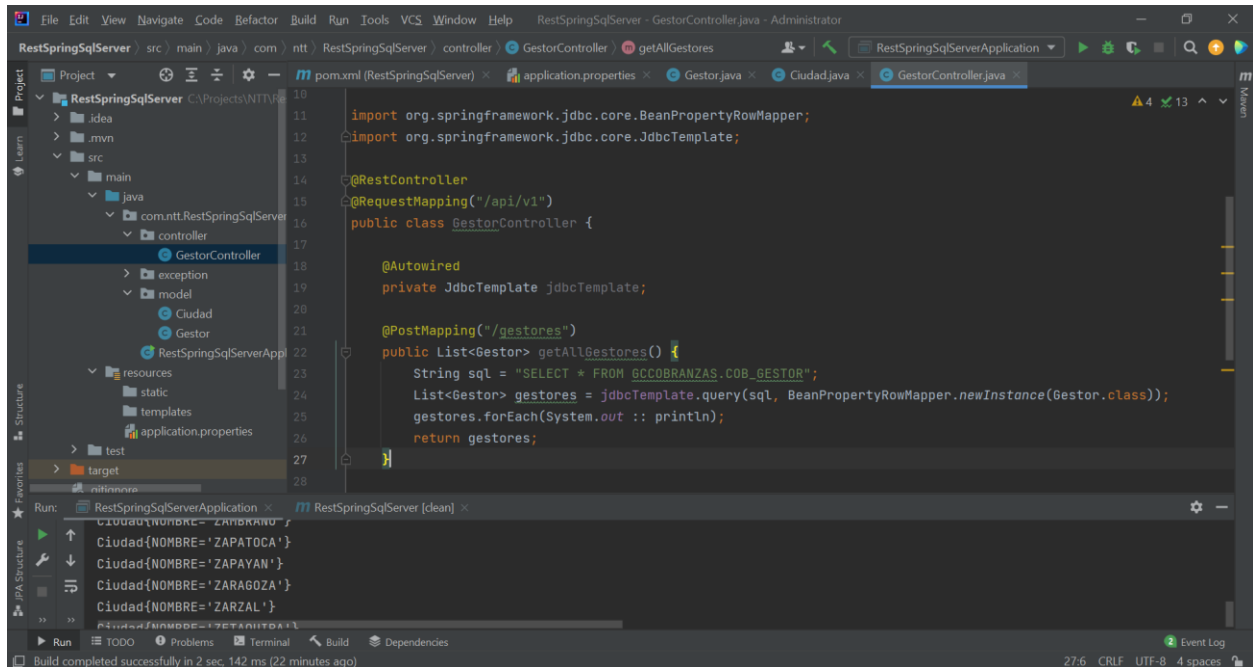
1 package com.ntt.RestSpringSqlServer.model;
2
3 public class Gestor {
4     private String PRIMER_NOMBRE;
5
6     public String getPRIMER_NOMBRE() {
7         return PRIMER_NOMBRE;
8     }
9
10    public void setPRIMER_NOMBRE(String PRIMER_NOMBRE) {
11        this.PRIMER_NOMBRE = PRIMER_NOMBRE;
12    }
13
14    @Override
15    public String toString() {
16        return PRIMER_NOMBRE;
17    }
18 }

```

The screenshot shows the IDE's Project Explorer on the left with the 'Gestor' class selected under 'model'. The main editor displays the code for the Gestor class, which includes a private attribute, getters, setters, and a toString method. The Run console at the bottom shows the application running successfully.

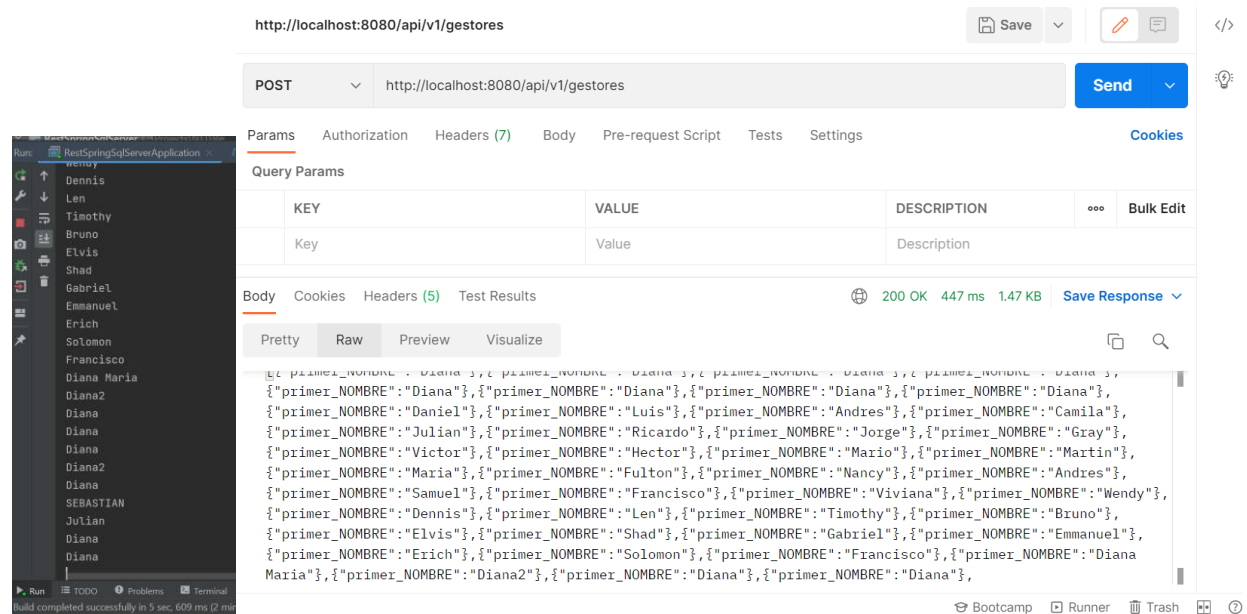
Dentro del modelo como el ejercicio es poder retornar a través del servicio Rest los nombres de los Gestores, cree variables, getters and setters con el primer nombre del Gestor obtenido de la base de datos.

Controlador



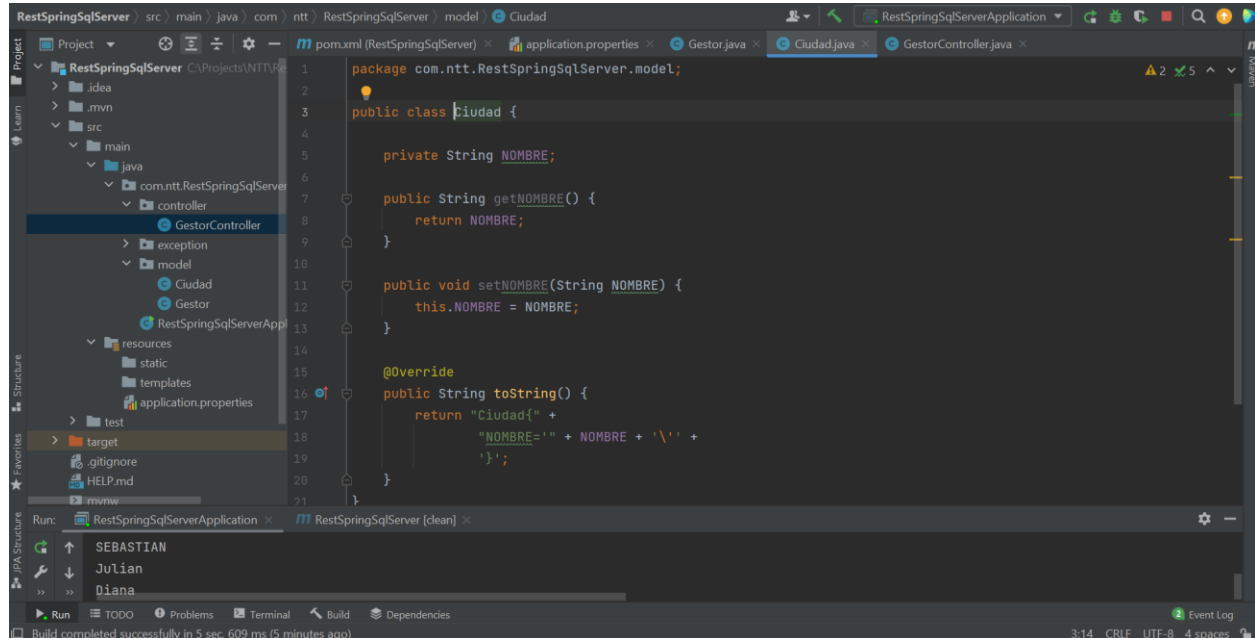
Dentro del Controlador se encuentra la lógica del servicio Rest y de la operación solicitada en el reto (Post) también dentro de este se declaro el query SQL mediante el cual obtenemos la información de la Base de Datos, finalmente se imprime el resultado mediante la línea de comandos y se retorna el resultado a la petición realizada en este caso en Postman

Resultado en la línea de comandos y Postman:



Finalmente para la última parte del reto se implemento de igual forma el modelo y el controlador necesarios para el funcionamiento del servicio rest de conteo de Ciudades, para el modelo también tomamos el nombre de cada ciudad para de esta forma verificar el funcionamiento del proyecto.

Modelo



```

package com.ntt.RestSpringSqlServer.model;

public class Ciudad {

    private String NOMBRE;

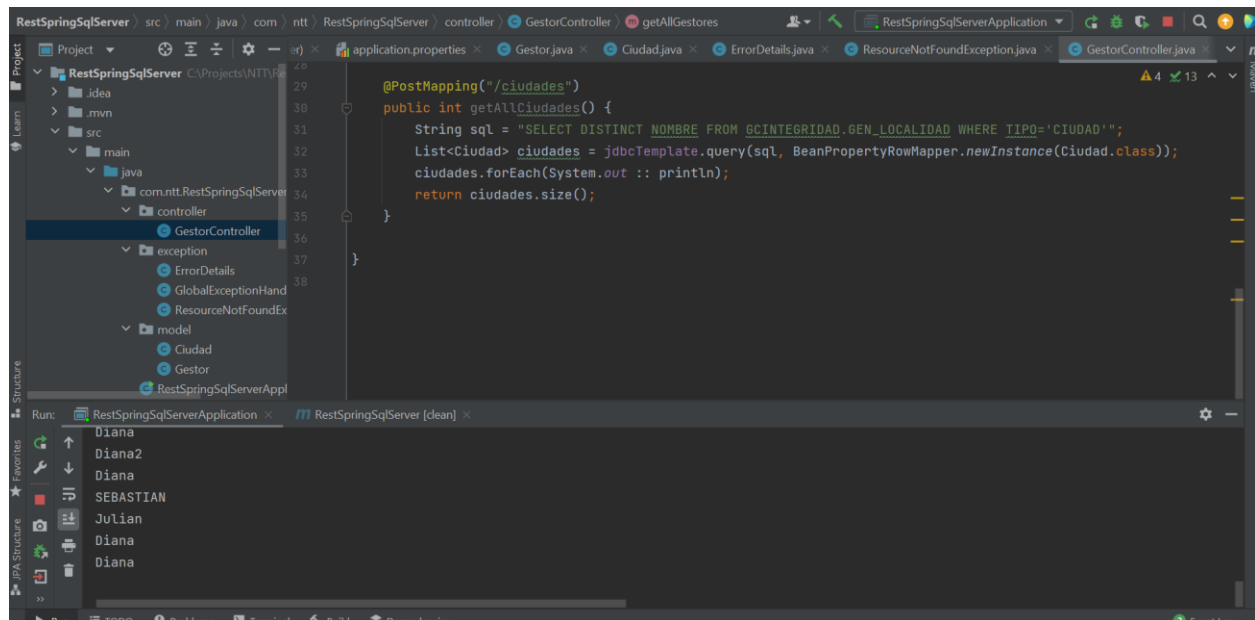
    public String getNOMBRE() {
        return NOMBRE;
    }

    public void setNOMBRE(String NOMBRE) {
        this.NOMBRE = NOMBRE;
    }

    @Override
    public String toString() {
        return "Ciudad{" +
            "NOMBRE='" + NOMBRE + '\'' +
        '}';
    }
}

```

Dentro del Controlador se encuentra la lógica del servicio Rest y de la operación solicitada en el reto (Post) también dentro de este se declaro el query SQL que en este caso debía tener algunas consideraciones, como el reto solicitaba el conteo solo de ciudades se debía filtrar por tipo 'Ciudad' y además de esto se debía seleccionar la operación Distinct en el Query con la cuál garantizamos que no se repita el conteo de una ciudad y tengamos un total confiable



```

@PostMapping("/ciudades")
public int getAllCiudades() {
    String sql = "SELECT DISTINCT NOMBRE FROM GCINTEGRIDAD.GEN_LOCALIDAD WHERE TIPO='CIUDAD'";
    List<Ciudad> ciudades = jdbcTemplate.query(sql, BeanPropertyRowMapper.newInstance(Ciudad.class));
    ciudades.forEach(System.out :: println);
    return ciudades.size();
}

```

Finalmente se imprime el resultado mediante la línea de comandos y se retorna el resultado a la petición realizada en este caso en Postman en donde se observa que el total de ciudades en la tabla de GCINTEGRIDAD.GC_LOCALIDAD son 1032.

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/api/v1/ciudades`. The response status is **200 OK** with a response time of 230 ms and a body size of 168 B. The response body is `1032`.

Finalmente se realizaron algunas clases de excepciones para el manejo de errores en donde imprimimos el tipo de error si es por lado del cliente o del servidor y un timestamp

Ejemplo:

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/api/v1/ciudade`. The response status is **404 Not Found** with a response time of 62 ms and a body size of 363 B. The response body is a JSON object:

```
{
  "timestamp": "2021-12-28T23:06:26.788+00:00",
  "status": 404,
  "error": "Not Found",
  "path": "/api/v1/ciudade"
}
```