

**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

FACULTAD DE INGENIERÍA

PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS DE INFORMACIÓN

PACIENTE DE CLÍNICA – GRUPO 04

**Desarrollo para Sistemas Distribuidos**

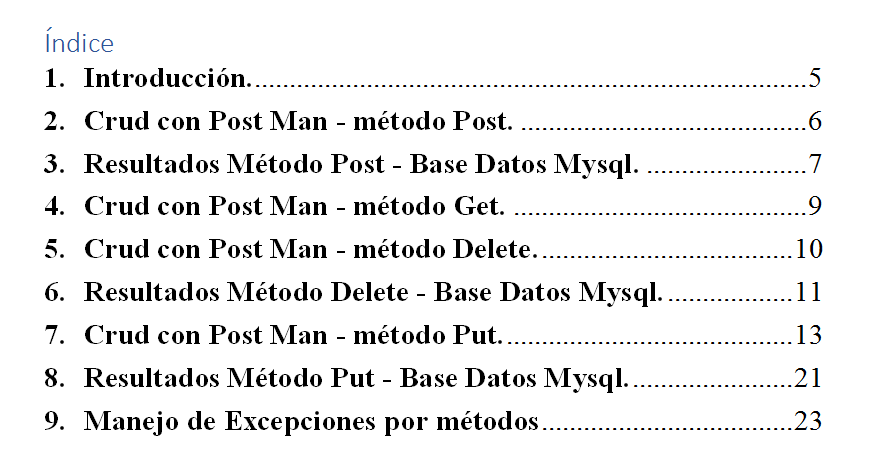
Integrantes:

* Christian Arturo Cabrera Pajares- U202020431
* Oscar David Hospinal Roman - U202021214
* Edgard Daniel Castañeda De La Cruz- U201724395
* Juan Pablo Cánepa Alvarez- U201816287
* Jose Daniel Orosco Figueroa- U201501207
* Lucas Daniel Mosquera Llosa – U202020951

**PROFESOR**

Flores Orihuela Carlos Alberto

**Lima, 30 de Octubre del 2020**

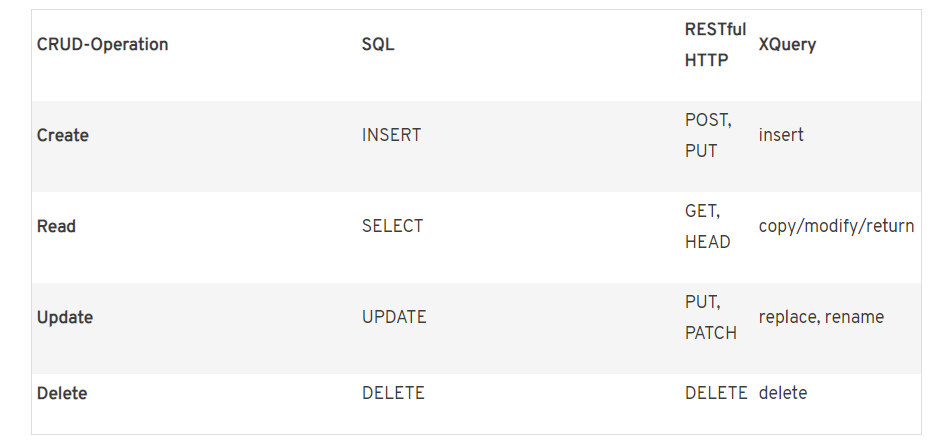
****

**Introducción**

A continuación el presente trabajo presentará la creación de un “CRUD”, utilizando métodos en HTTP (PostMan), para “Paciente de clínica”, utilizando el lenguaje de programación Java, usando la herramienta Intellij Idea.

Esta a su vez utilizará una base de datos (Mysql) donde se guardará la data en general.

En pocas palabras, CRUD resume las funciones requeridas por un usuario para crear y gestionar datos. Varios procesos de gestión de datos están basados en CRUD, en los que dichas operaciones están específicamente adaptadas a los requisitos del sistema y de usuario, ya sea para la gestión de bases de datos o para el uso de aplicaciones. Para los expertos, las operaciones son las herramientas de acceso típicas e indispensables para comprobar, por ejemplo, los problemas de la base de datos, mientras que para los usuarios, CRUD significa crear una cuenta (créate) y utilizarla (read), actualizarla (update) o borrarla (delete) en cualquier momento. Dependiendo de la configuración regional, las operaciones CRUD pueden implementarse de diferentes maneras, como lo muestra la siguiente tabla:



**Crud con Post Man - método Post**

A continuación se muestra la data ingresada al Postman, utilizando el método post, luego llamando a la conexión :” <http://localhost:8080/home/patient>” y en el Body insertando la data con los atributos creados en la clase “Patient”.

La cual arroja como resultado la misma data ingresada.

Ingreso 01:

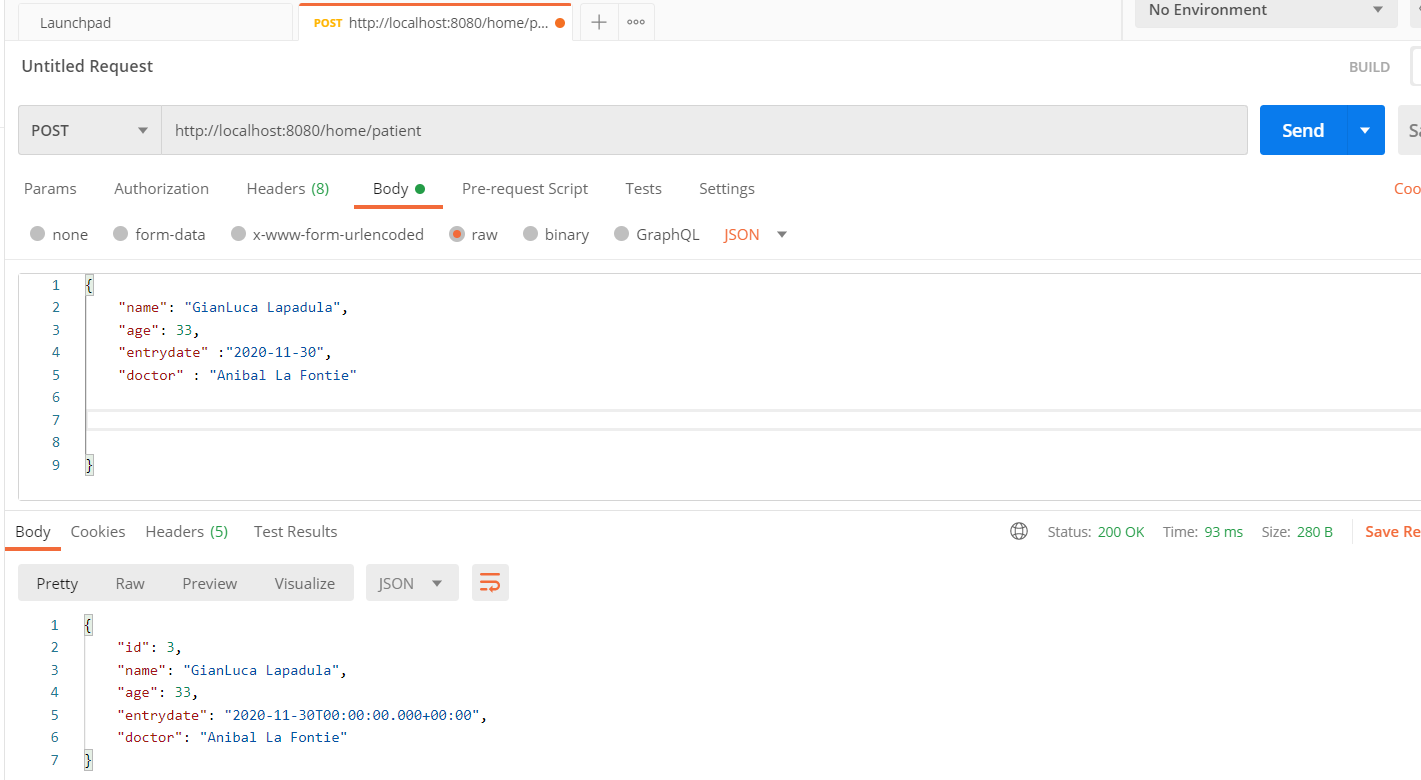
"name": "GianLuca Lapadula",

    "age": 33,

    "entrydate" :"2020-11-30",

    "doctor" : "Anibal La Fontie"

Resultado 01:

****

Ingreso 02:

"name": "Renato Tapia",

    "age": 26,

    "entrydate" :"2020-12-05",

    "doctor" : "Alonso la Valle"

Resultado 02:

{

    "id": 4,

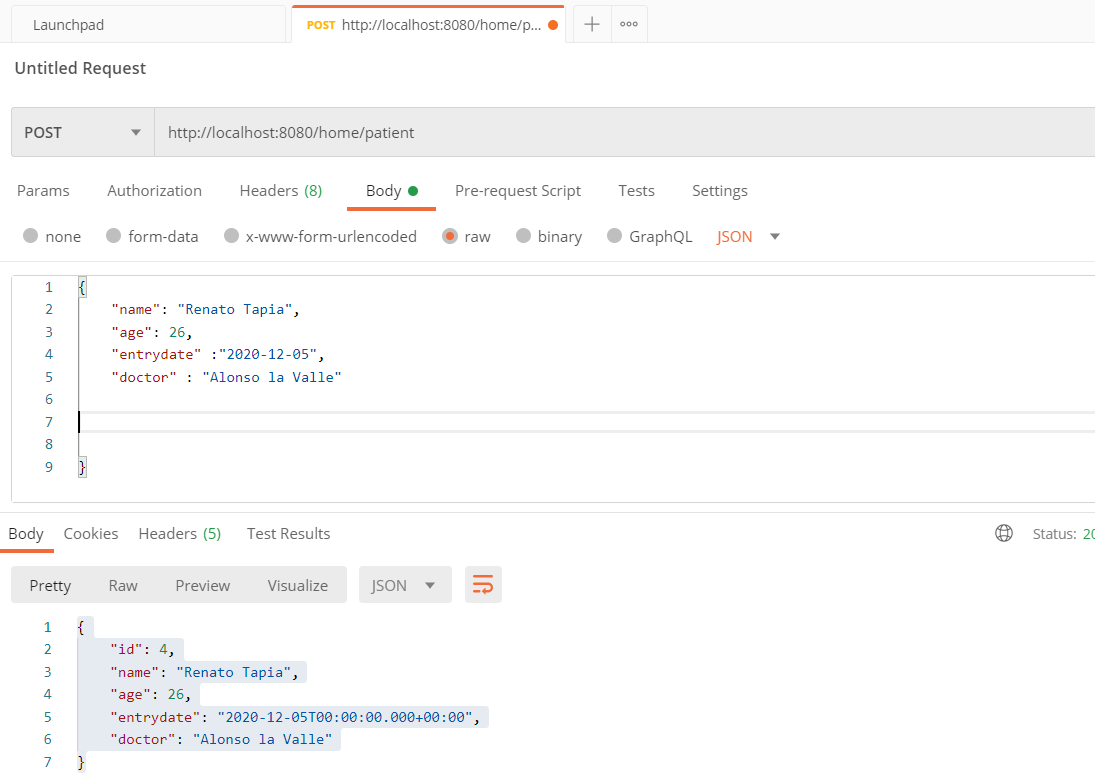
    "name": "Renato Tapia",

    "age": 26,

    "entrydate": "2020-12-05T00:00:00.000+00:00",

    "doctor": "Alonso la Valle"

}



Ingreso 03:

{

    "name": "Paolo Guerrero",

    "age": 38,

    "entrydate" :"2021-07-13",

    "doctor" : "Jose la Marque Ildenfonso"

}

Resultado 03:

{

    "id": 5,

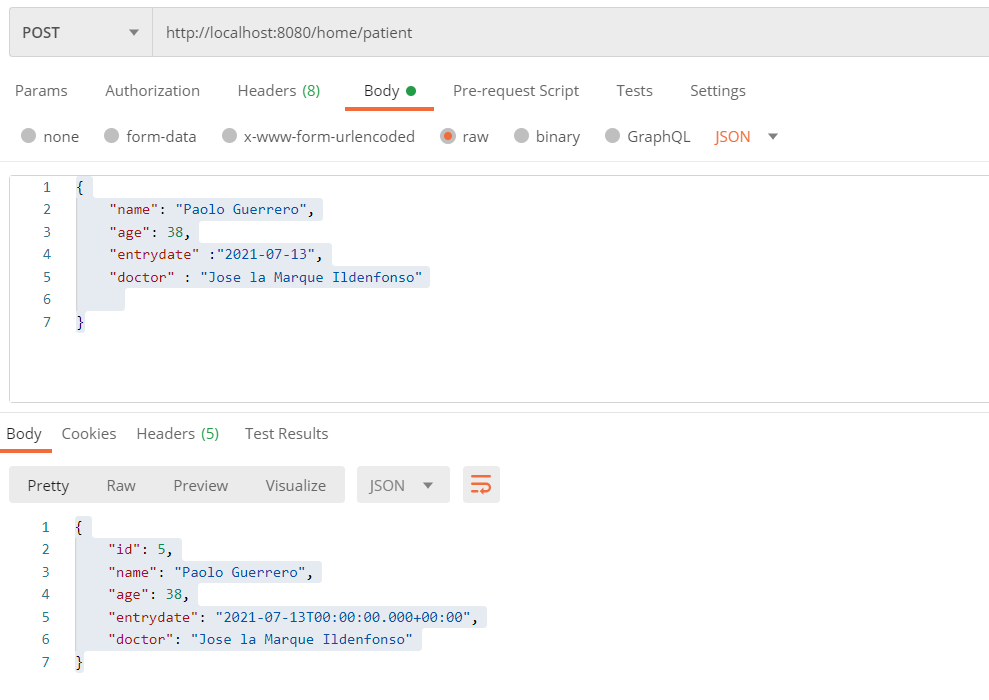
    "name": "Paolo Guerrero",

    "age": 38,

    "entrydate": "2021-07-13T00:00:00.000+00:00",

    "doctor": "Jose la Marque Ildenfonso"

}

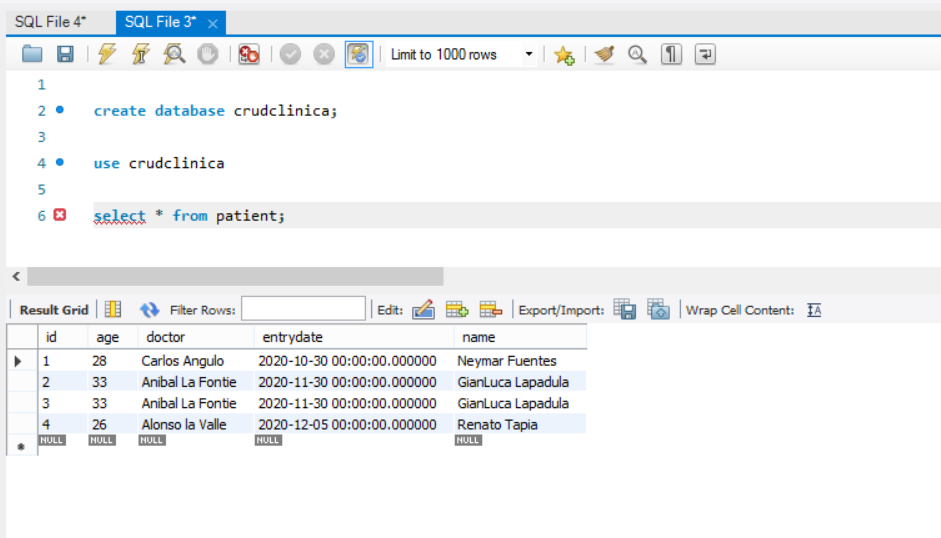


**Resultados Método Post - Base Datos Mysql**

A continuación se muestra en el Mysql , la actualización de de la base de datos “crudclinica” y el ingreso de la data en la tabla “patient”.

Para ello previamente en application.properties del Intellij se hizo la conexión.

## Spring DATASOURCE  
#Cadena de conexion jdbc a la base de datos  
spring.datasource.url=jdbc:mysql://localhost:3306/crudclinica?serverTimezone=UTC  
#usuario de la base de datos  
spring.datasource.username=root  
#clave de la base de datos  
spring.datasource.password=mysql  
## Hibernate Properties  
# The SQL dialect makes Hibernate generate better SQL for the chosen database  
#tipo o nombre de la base de datos  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect  
# Hibernate ddl auto (create, create-drop, validate, update)  
#update sino existe las tablas Entidad las creará, sino las actualizará  
spring.jpa.hibernate.ddl-auto=update  
#Desactiva el atributo trace de la Excepciones tipo ResponseStatusException en los servicios REST  
server.error.include-stacktrace=*on\_trace\_param*



**Crud con Post Man - método Get**

A continuación se muestra la consulta , utilizando el método get, luego llamando a la conexión :” <http://localhost:8080/home/patients>”, la cual al final me devuelve un arreglo de Json.

Por otro lado también podemos listar llamando por código:” <http://localhost:8080/home/patient/5>”, la cual al final me devolverá la data con el código consultado.

Resultado 01 – Listando:

[

    {

        "id": 1,

        "name": "Neymar Fuentes",

        "age": 28,

        "entrydate": "2020-10-30T00:00:00.000+00:00",

        "doctor": "Carlos Angulo"

    },

    {

        "id": 2,

        "name": "GianLuca Lapadula",

        "age": 33,

        "entrydate": "2020-11-30T00:00:00.000+00:00",

        "doctor": "Anibal La Fontie"

    },

    {

        "id": 3,

        "name": "GianLuca Lapadula",

        "age": 33,

        "entrydate": "2020-11-30T00:00:00.000+00:00",

        "doctor": "Anibal La Fontie"

    },

    {

        "id": 4,

        "name": "Renato Tapia",

        "age": 26,

        "entrydate": "2020-12-05T00:00:00.000+00:00",

        "doctor": "Alonso la Valle"

    },

    {

        "id": 5,

        "name": "Paolo Guerrero",

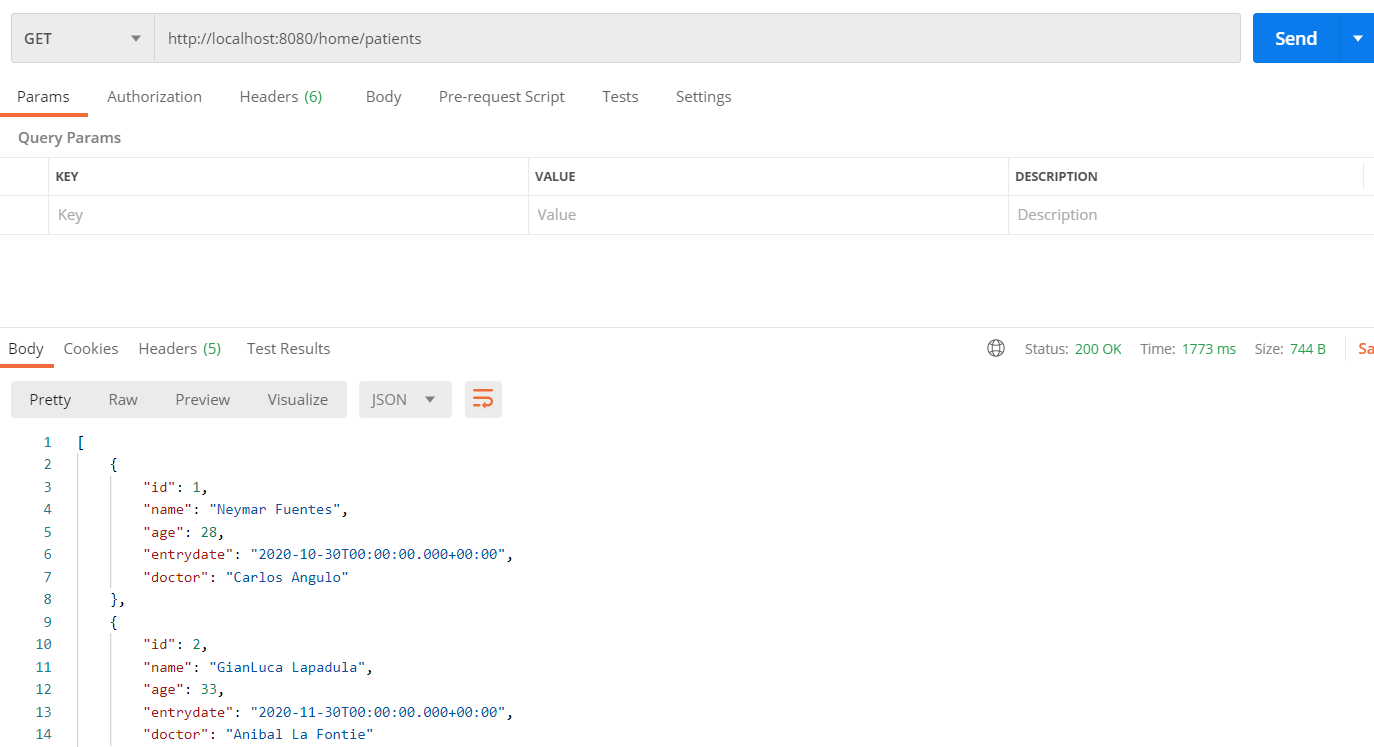
        "age": 38,

        "entrydate": "2021-07-13T00:00:00.000+00:00",

        "doctor": "Jose la Marque Ildenfonso"

    }

]



Resultado 02 – Listando por código:

{

    "id": 5,

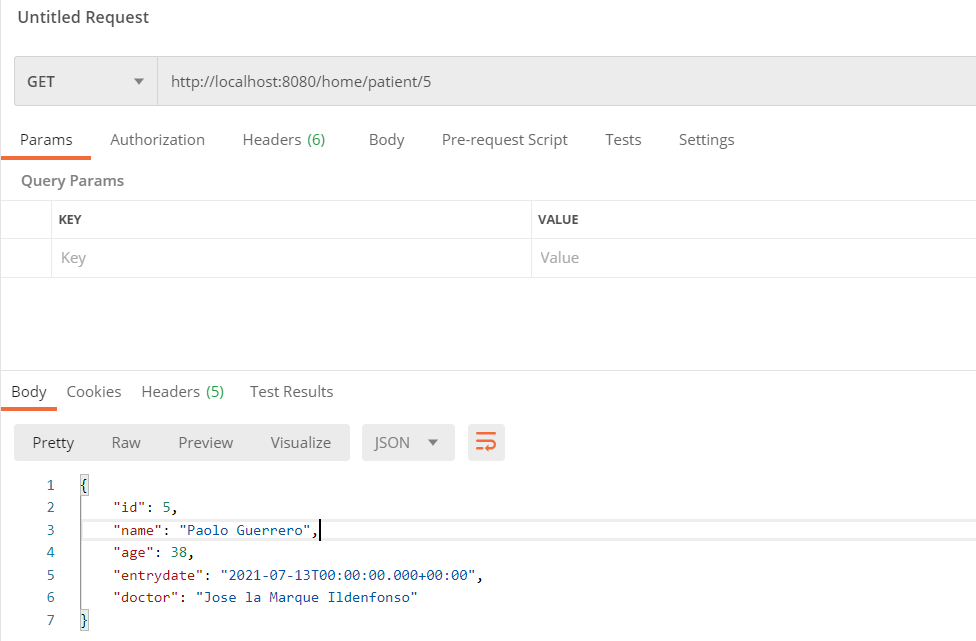
    "name": "Paolo Guerrero",

    "age": 38,

    "entrydate": "2021-07-13T00:00:00.000+00:00",

    "doctor": "Jose la Marque Ildenfonso"

}



**Crud con Post Man - método Delete**

A continuación se muestra el método delete la cual se ingresa un código para eliminar la data de esa fila de la tabla en nuestra base de datos mysql, utilizando el método delete en el Post, luego llamando a la conexión:” http://localhost:8080/home/patient/3”.

Resultado 01 – Eliminando por código:

{

    "id": 3,

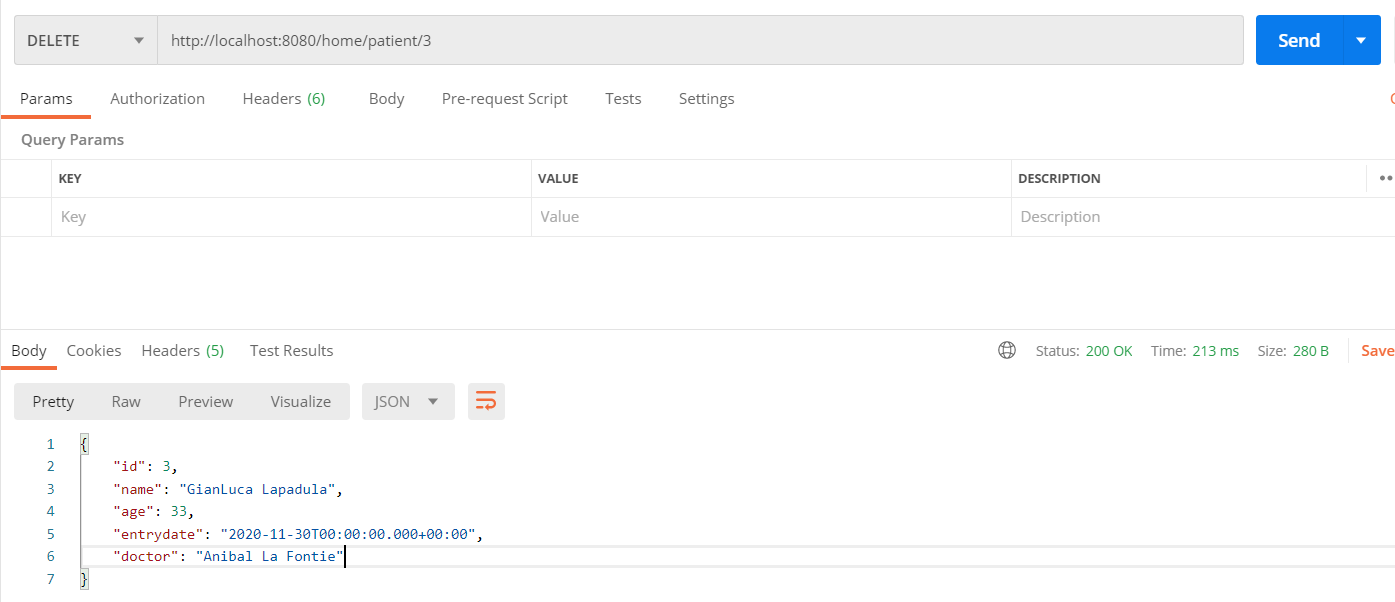
    "name": "GianLuca Lapadula",

    "age": 33,

    "entrydate": "2020-11-30T00:00:00.000+00:00",

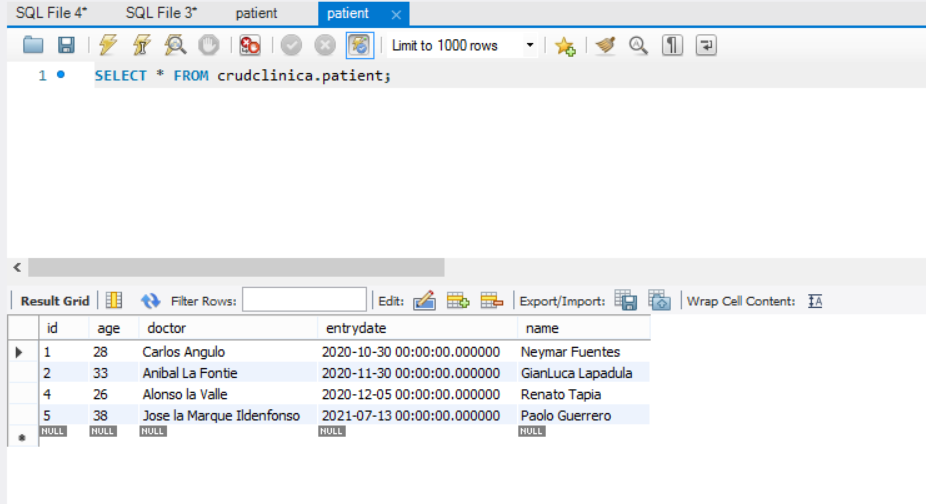
    "doctor": "Anibal La Fontie"

}



**Resultados Método Delete - Base Datos Mysql**

A continuación se muestra en el Mysql , la eliminación de la tabla por el código solicitado en la base de datos “crudclinica”.



**Crud con Post Man - método Put**

A continuación se muestra el método put la cual se ingresa un código para actualizar la data de esa fila de la tabla en nuestra base de datos mysql, utilizando el método put en el Post, luego llamando a la conexión:” <http://localhost:8080/home/patient>”.

Resultado 01 – Actualizar por código:

{

      "id": 4,

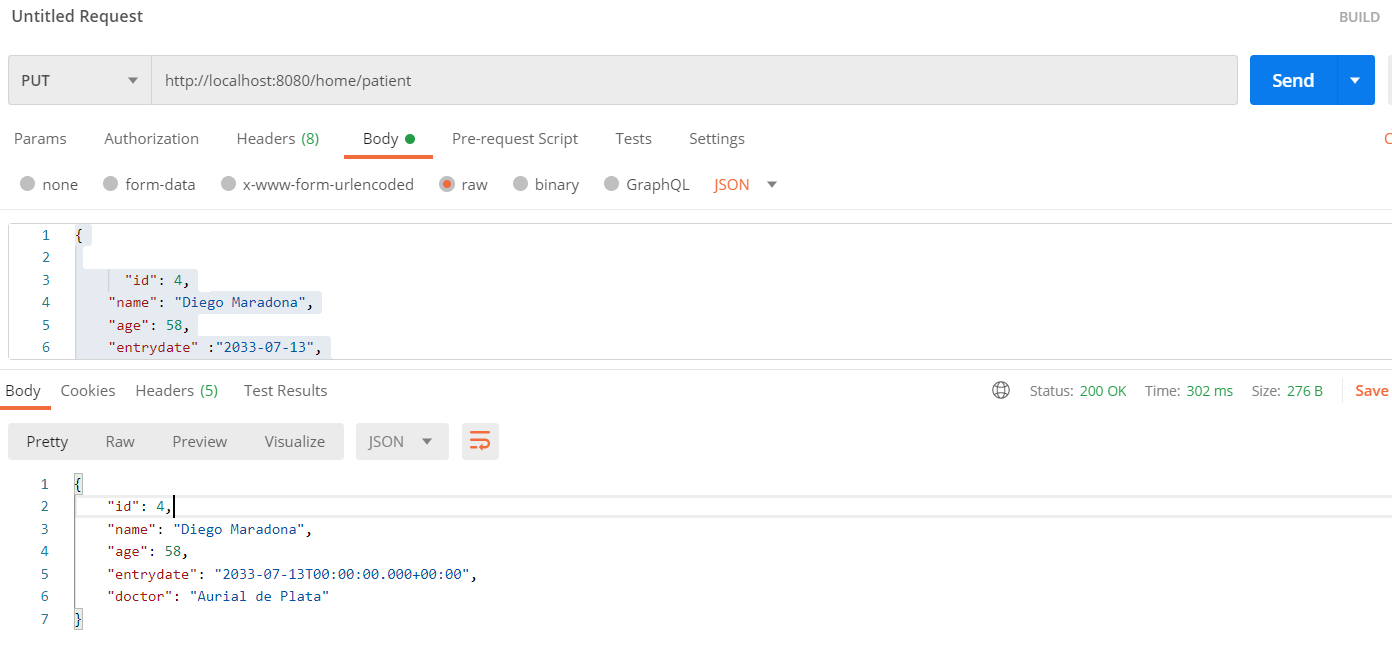
    "name": "Diego Maradona",

    "age": 58,

    "entrydate" :"2033-07-13",

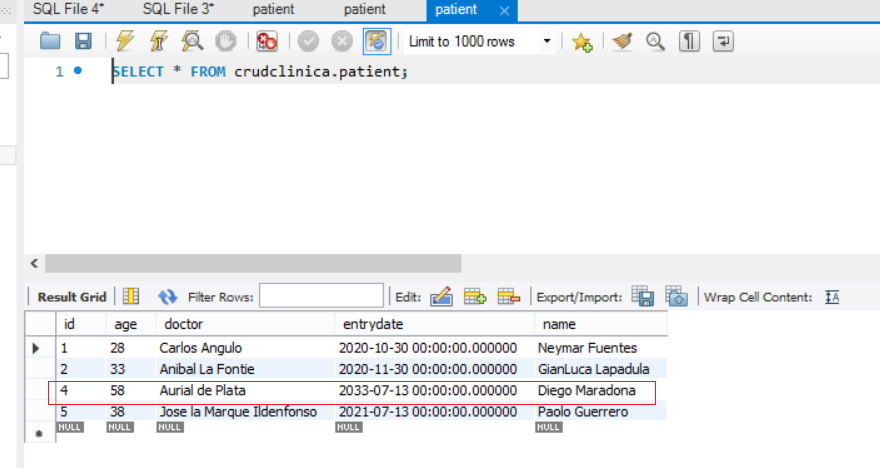
    "doctor" : "Aurial de Plata"

}



**Resultados Método Put - Base Datos Mysql**

A continuación se muestra en el Mysql , la actualización de la tabla por el código solicitado en la base de datos “crudclinica”.

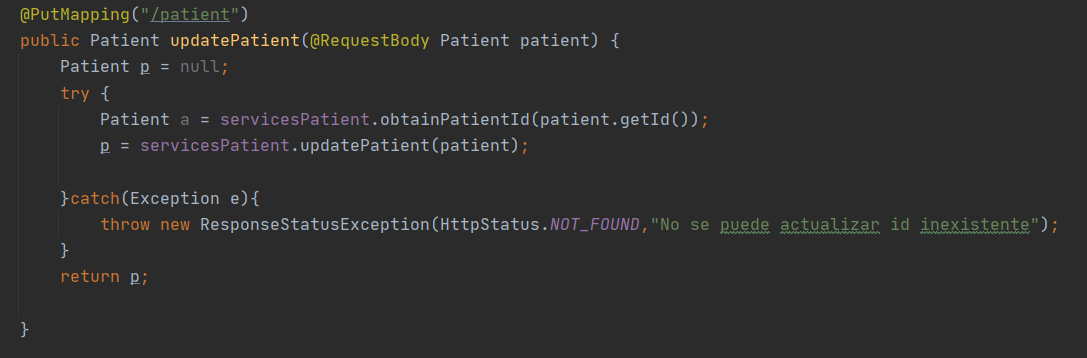


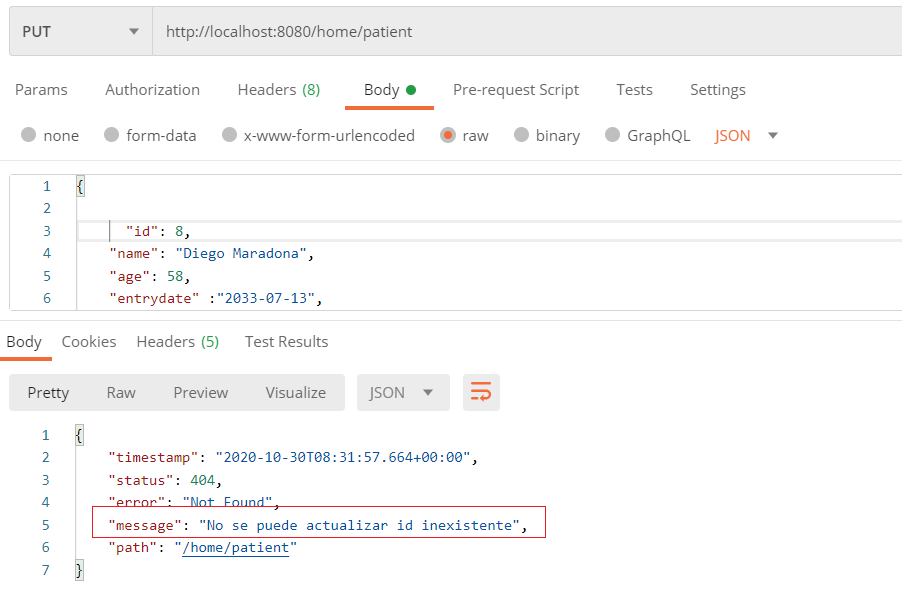
**Manejo de Excepciones por métodos**

En ello utilizaremos excepciones a los métodos usados, para poder controlar cualquier tipo de error.

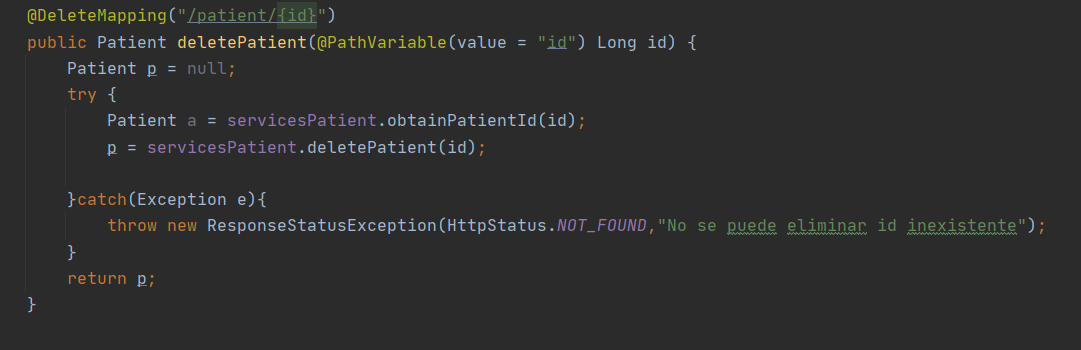
Es una técnica de programación que permite al programador controlar los errores ocasionados durante la ejecución de un programa, este caso aplicados al java.

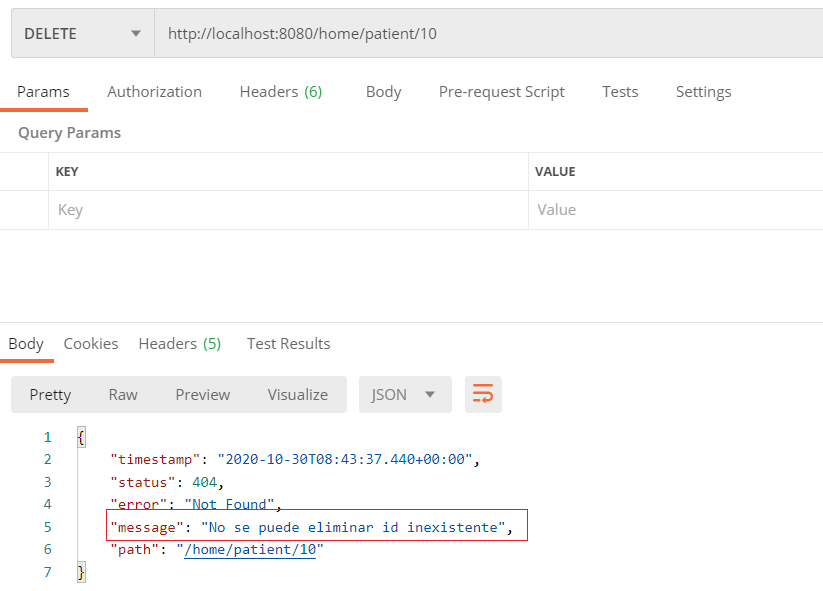
Resultado 01 –Manejo de excepciones en Java- Método put:



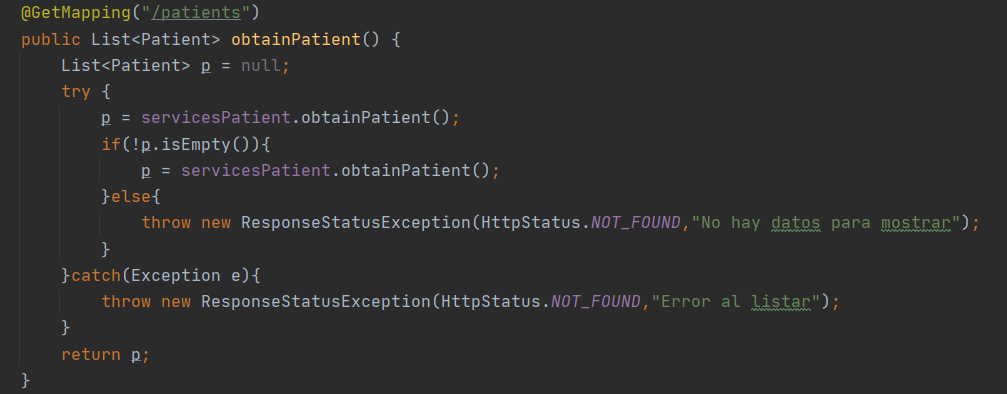


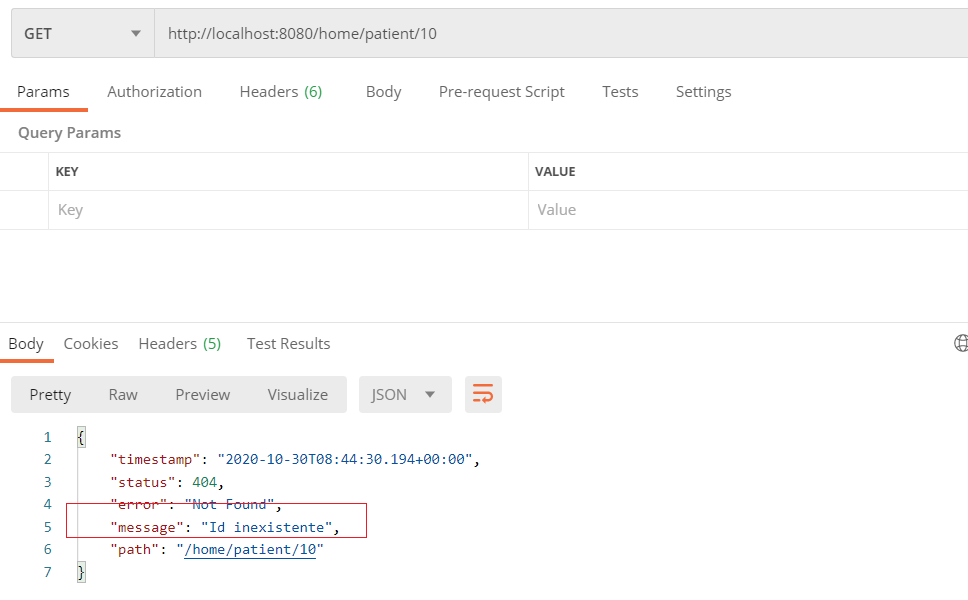
Resultado 02 –Manejo de excepciones en Java- Método delete:





Resultado 03 –Manejo de excepciones en Java- Método get:





Resultado 04 –Manejo de excepciones en Java- Método post:

