Sprint 3

Universidad Nacional de Colombia

Docente a cargo:
Brayan Steven Garcia

Integrantes:
Maria Paula Torres
Anderson Daza
Santiago Rodiguez
Laura Patarroyo
Julian Mora

Grupo:
P14

Grupo de trabajo:
3

Repositorio:
https://github.com/4a-docs/4a-ms2

Modelos:
Se definen los atributos del modelo Agenda, con datos como fecha y hora, al cual tendrá acceso el rol Doctor.

Agenda.java

```java
package com.example.agenda.models;
import org.springframework.data.annotation.Id;
import java.sql.Date;
import java.time.LocalDate;
import java.time.LocalTime;

public class Agenda {

    @Id
    private String id;
    private String doctor;
    private LocalTime start_hour;
    private LocalDate start_date;

    public Agenda(String id, String doctor, LocalTime start_hour, LocalDate start_date) {
        this.id = id;
        this.doctor = doctor;
        this.start_hour = start_hour;
//        this.end_hour = end_hour;
        this.start_date = start_date;
//        this.end_date = end_date;
    }

    public String getId() { return id; }

    public String getDoctor() { return doctor; }

    public LocalTime getStart_hour() { return start_hour; }

    public void setStart_hour(LocalTime start_hour) { this.start_hour = start_hour; }

    public LocalDate getStart_date() { return start_date; }

    public void setStart_date(LocalDate start_date) { this.start_date = start_date; }

}
```

Repositorio:
Define el método AgendaRepository que nos permite el acceso a la base de datos en Mongo.

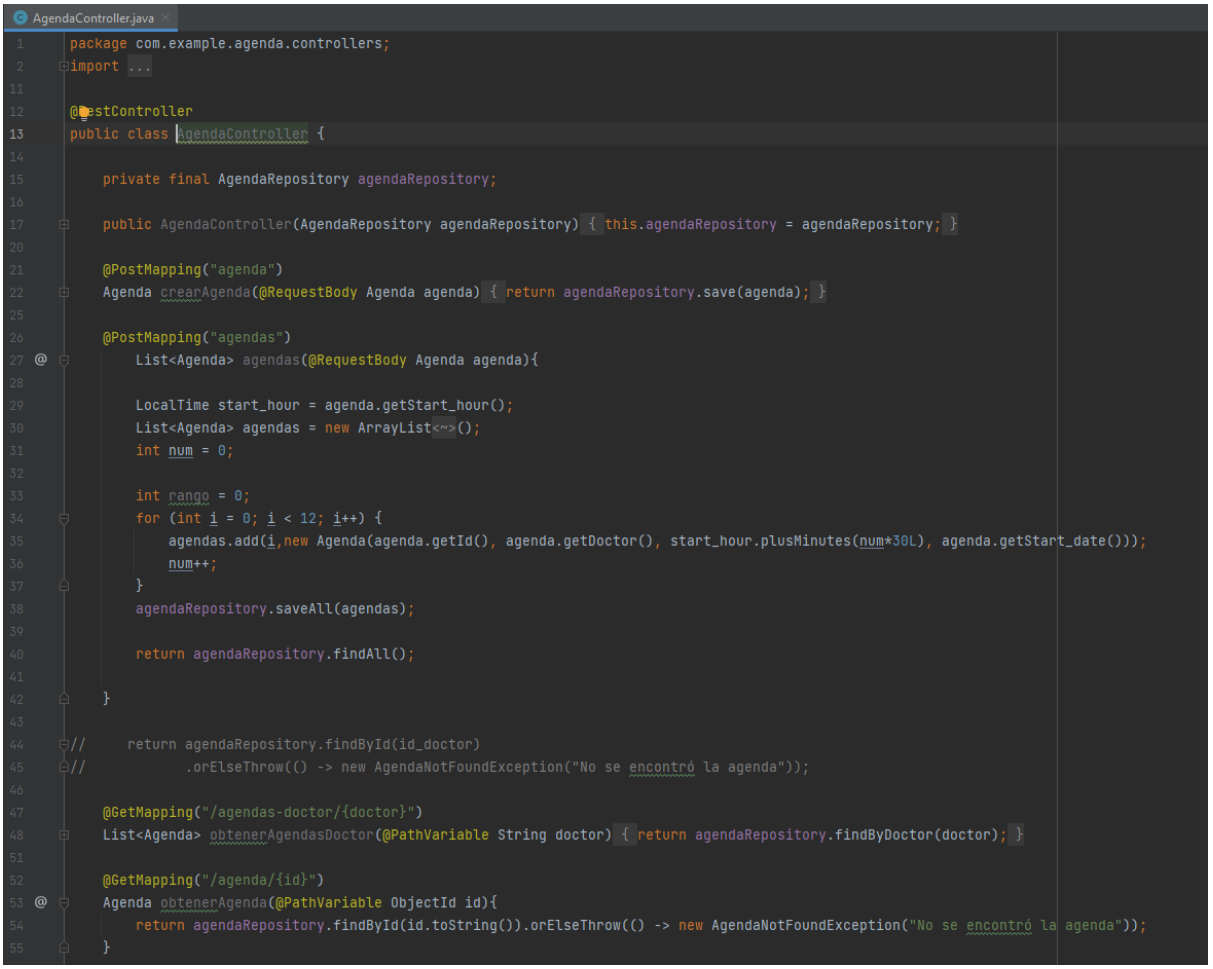
AgendaRepository.java

```java
package com.example.agenda.repositories;

import com.example.agenda.models.Agenda;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;

import java.time.LocalDate;
import java.util.List;
import java.util.Optional;

public interface AgendaRepository extends MongoRepository<Agenda, String> {
    List<Agenda> findByDoctor (String doctor);
}

```

Controladores:

En AgendaController se define el CRUD con ayuda de los decoradores:

- Post agenda, permite la creación de la agenda del doctor.
- Post agendas, realiza la inserción múltiple de 12 registros con intervalos de 30 minutos desde la hora inicial que se agrega en el post.
- Get agendas doctor, permite visualizar la agenda creada previamente filtrada por el ID del doctor.
- Delete agenda, borra la agenda del doctor.
- Put agenda, permite editar los datos como fecha y hora.



```java
package com.example.agenda.controllers;
import ...

@RestController
public class AgendaController {

    private final AgendaRepository agendaRepository;

    public AgendaController(AgendaRepository agendaRepository) { this.agendaRepository = agendaRepository; }

    @PostMapping("agenda")
    Agenda crearAgenda(@RequestBody Agenda agenda) { return agendaRepository.save(agenda); }

    @PostMapping("agendas")
    List<Agenda> agendas(@RequestBody Agenda agenda){

        LocalTime start_hour = agenda.getStart_hour();
        List<Agenda> agendas = new ArrayList<~>();
        int num = 0;

        int rango = 0;
        for (int i = 0; i < 12; i++) {
            agendas.add(i,new Agenda(agenda.getId(), agenda.getDoctor(), start_hour.plusMinutes(num*30L), agenda.getStart_date()));
            num++;
        }
        agendaRepository.saveAll(agendas);

        return agendaRepository.findAll();

    }

//    return agendaRepository.findById(id_doctor)
//            .orElseThrow(() -> new AgendaNotFoundException("No se encontró la agenda"));

    @GetMapping("/agendas-doctor/{doctor}")
    List<Agenda> obtenerAgendasDoctor(@PathVariable String doctor) { return agendaRepository.findByDoctor(doctor); }

    @GetMapping("/agenda/{id}")
    Agenda obtenerAgenda(@PathVariable ObjectId id){
        return agendaRepository.findById(id.toString()).orElseThrow(() -> new AgendaNotFoundException("No se encontró la agenda"));
    }
```

```java

    @DeleteMapping("/agenda/{id}")
    String deleteAgenda(@PathVariable String id){
        Agenda user = agendaRepository.findById(id).orElseThrow(() -> new AgendaNotFoundException("No se encontró la agenda para eliminar"));
        agendaRepository.deleteById(id);
        return "La agenda se elimino correctamente agenda del doctor con id: "+user.getDoctor();
    }

    @PutMapping("/agenda/{id}")
    Agenda updateAgenda(@PathVariable String id, @RequestBody Agenda agenda){
        Agenda old_agenda = agendaRepository.findById(id).orElse( other: null);
        old_agenda.setStart_hour(agenda.getStart_hour());
        old_agenda.setStart_date(agenda.getStart_date());
        return agendaRepository.save(old_agenda);
    }
}
```

Excepciones:
En las excepciones colocamos las plantillas y sobreescribimos los métodos que nos van a ayudar a poder cambiar los mensajes de respuesta.

AgendaNotFoundAdvice.java

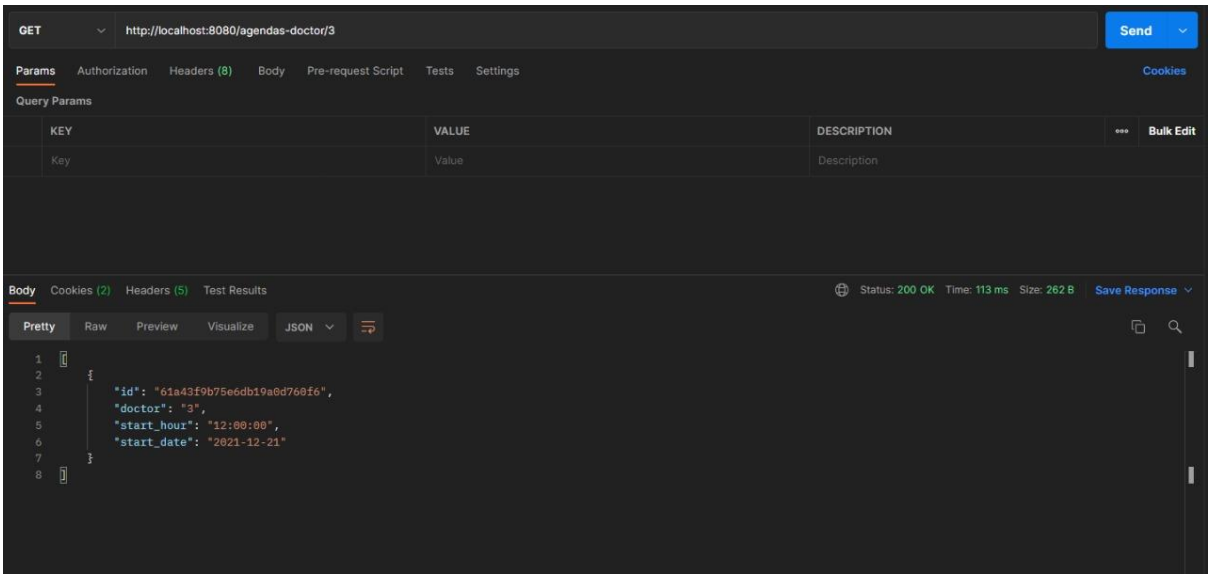
```java
package com.example.agenda.exceptions;

import ...

@ControllerAdvice
@ResponseBody

public class AgendaNotFoundAdvice {
    @ResponseBody
    @ExceptionHandler(AgendaNotFoundException.class)
    @ResponseStatus(HttpStatus.NOT_FOUND)
    String EntityNotFoundAdvice(AgendaNotFoundException ex) { return ex.getMessage(); }
}
```

AgendaNotFoundAdvice.java | AgendaNotFoundException.java

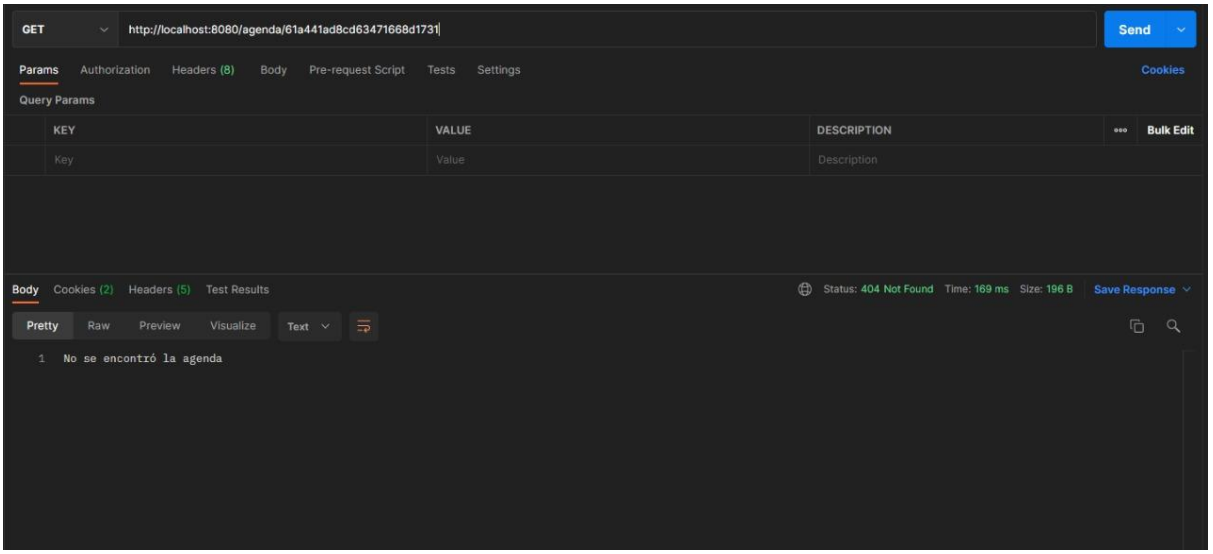
```java
package com.example.agenda.exceptions;

public class AgendaNotFoundException extends RuntimeException {
    public AgendaNotFoundException(String message) { super(message); }
}
```

Pruebas Postman:

GetByIdDoctor:
En la imagen se puede apreciar la búsqueda por el id de doctor, se pueden observar todas las agendas programadas



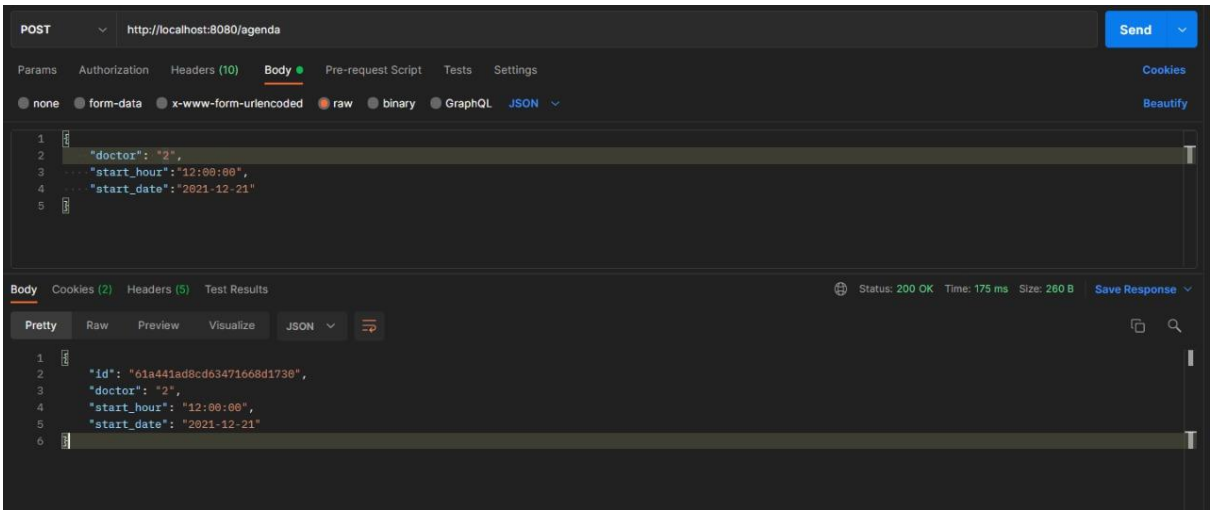
GetByIdAgenda(Excepciones):
En la imagen se puede apreciar como al filtrar por el id de una agenda que no existe o fue eliminada el programa nos arroja una excepción.



POST

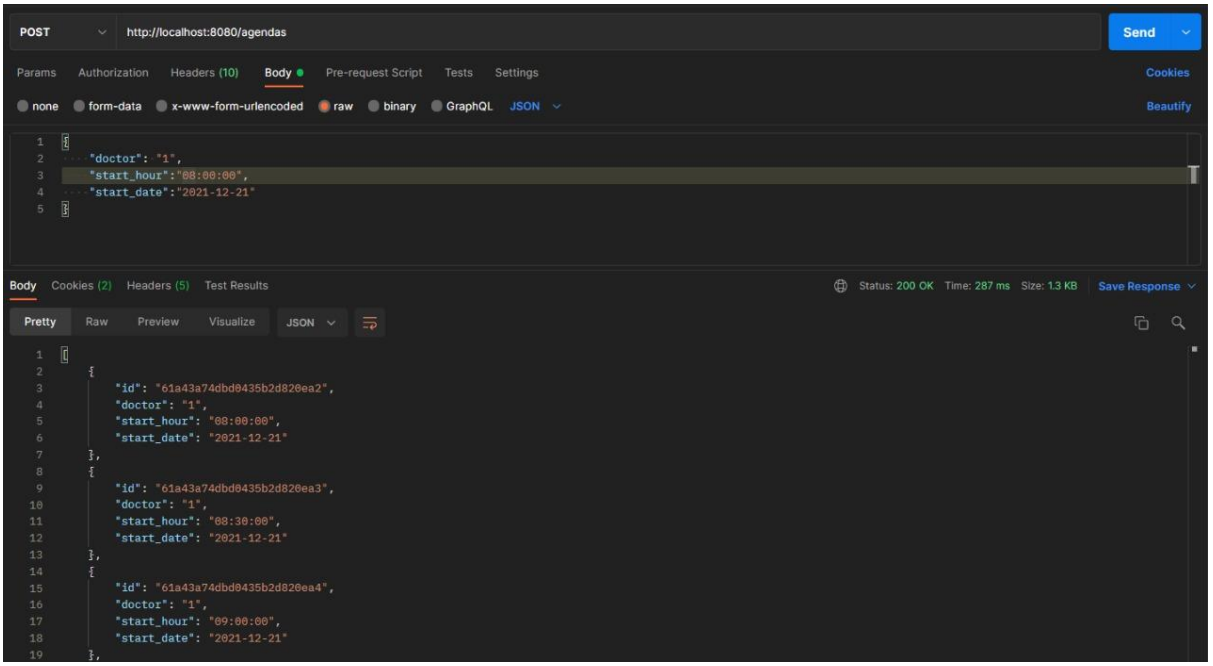
Agenda:

En la imagen se puede ver como se hace un post simple de una sola agenda

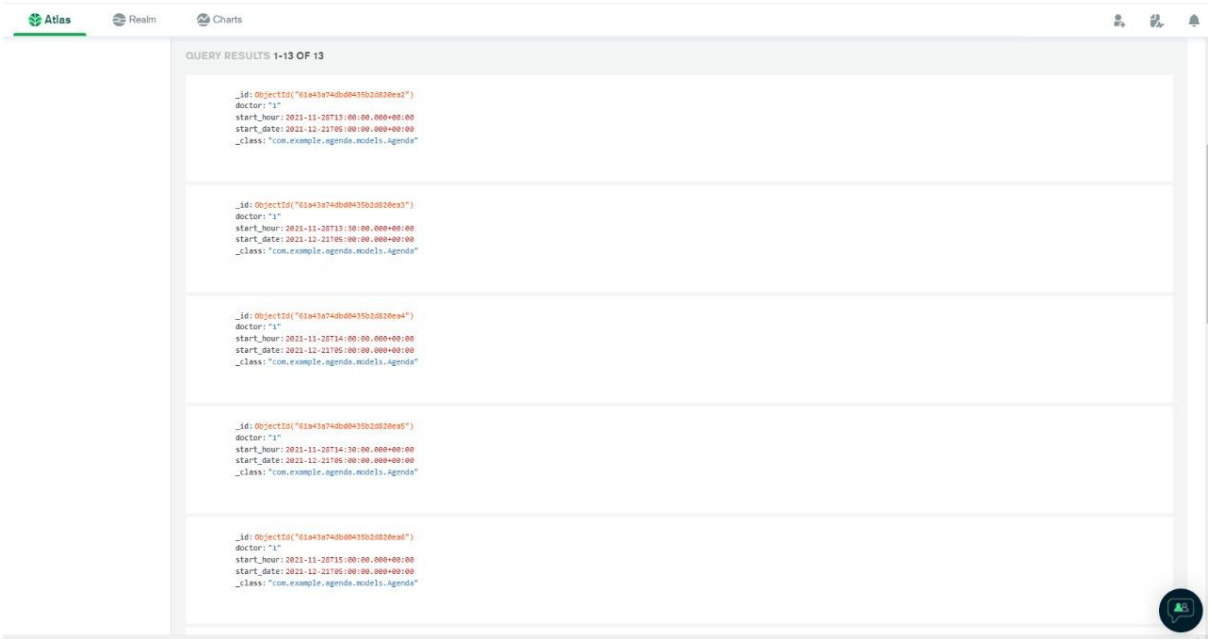


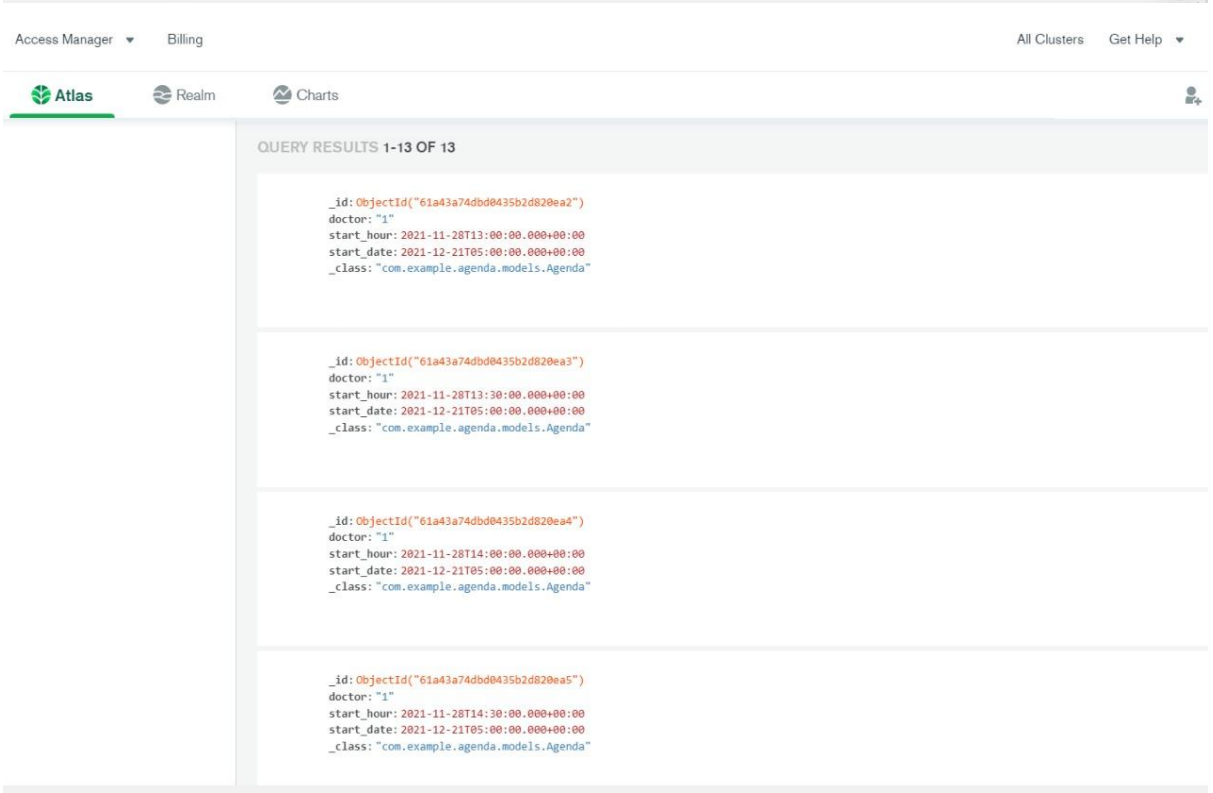
Agendas(Inserción Múltiple):

En la imagen se puede ver la inserción múltiple al momento de enviar un post con una hora inicial, el programa crea 12 registros con intervalos de 30 minutos para cada cita

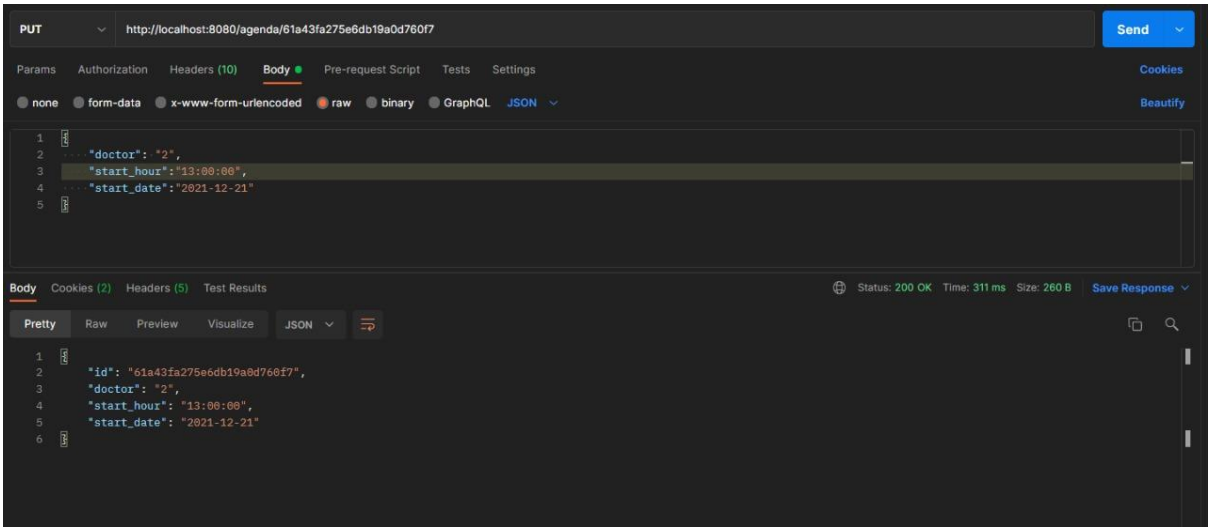


Pruebas de creación de Documentos en Mongodb(Atlas):
Se aprecian las inserciones de 13 elementos, 12 de inserción múltiple y 1 de inserción simple.

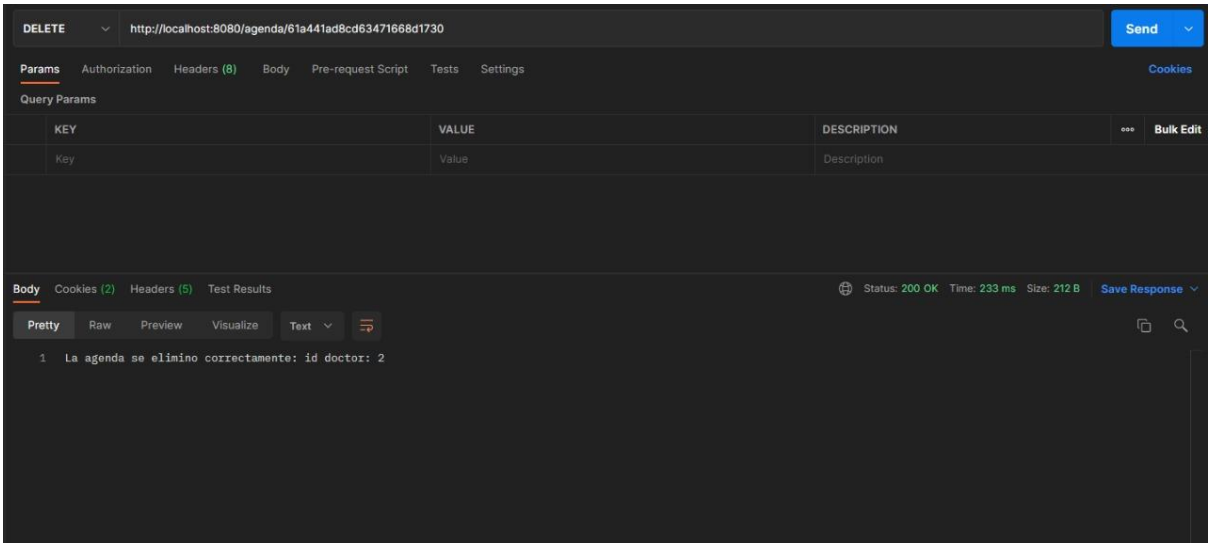




PUT:
En la imagen se puede apreciar como al enviar un PUT cambiando la hora inicial de un registro se cambia de 12:30 pm a 13:00 pm



Delete:
En la imagen se puede apreciar la eliminación por ID agenda con su respectiva respuesta tipo String



Delete Excepción:
Se puede apreciar que pasa si se intenta eliminar un registro inexistente o que ya había sido eliminado el programa arroja una excepción