15ª Emisión

DIPLOMADO

Desarrollo de Sistemas con Tecnología Java

Módulo 5-6

Desarrollo de aplicaciones empresariales con Jakarta EE

Uriel Hernández

DGTIC

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Dirección General de Cómputo y de Tecnologías de información y Comunicación
Dirección de Docencia en TIC

50 AÑOS
Educación Continua
1971 - 2021

Validator Framework

Colonia - Validator

```java
@Data
@NoArgsConstructor
@Entity
public class Colonia {

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer id;

    @NotBlank(message="Nombre es requerido")
    private String nombre;

    @Pattern(regexp="^(\\d{5})$",
            message="Formato no válido para Código Postal")
    private String cp;

    @ManyToOne(targetEntity=Municipio.class)
    @JoinColumn(name="id_municipio", nullable=false)
    @NotNull(message="La colonia debe estar asociada a un municipio")
    private Municipio municipio;
```

Práctica No. 2

APIs Colonia

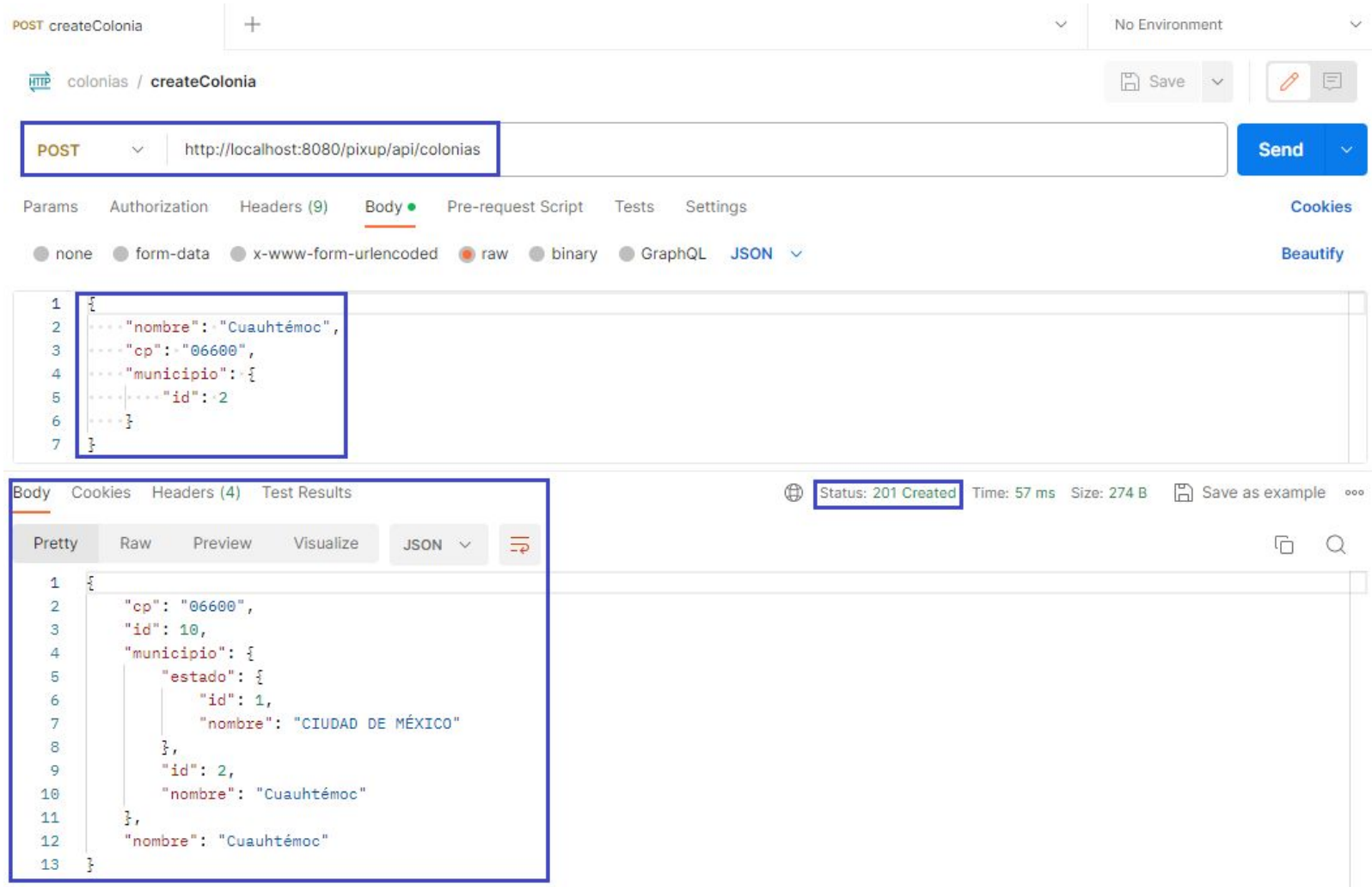
Capa API: ColoniaApi

```java
@DELETE
@Path("{id}")
void deleteColonia(@PathParam("id") Integer id);

@POST
Response createColonia(@NotNull @Valid Colonia colonia);

@PUT
@Path("{id}")
Colonia updateColonia(@PathParam("id") Integer id, Colonia colonia);
```

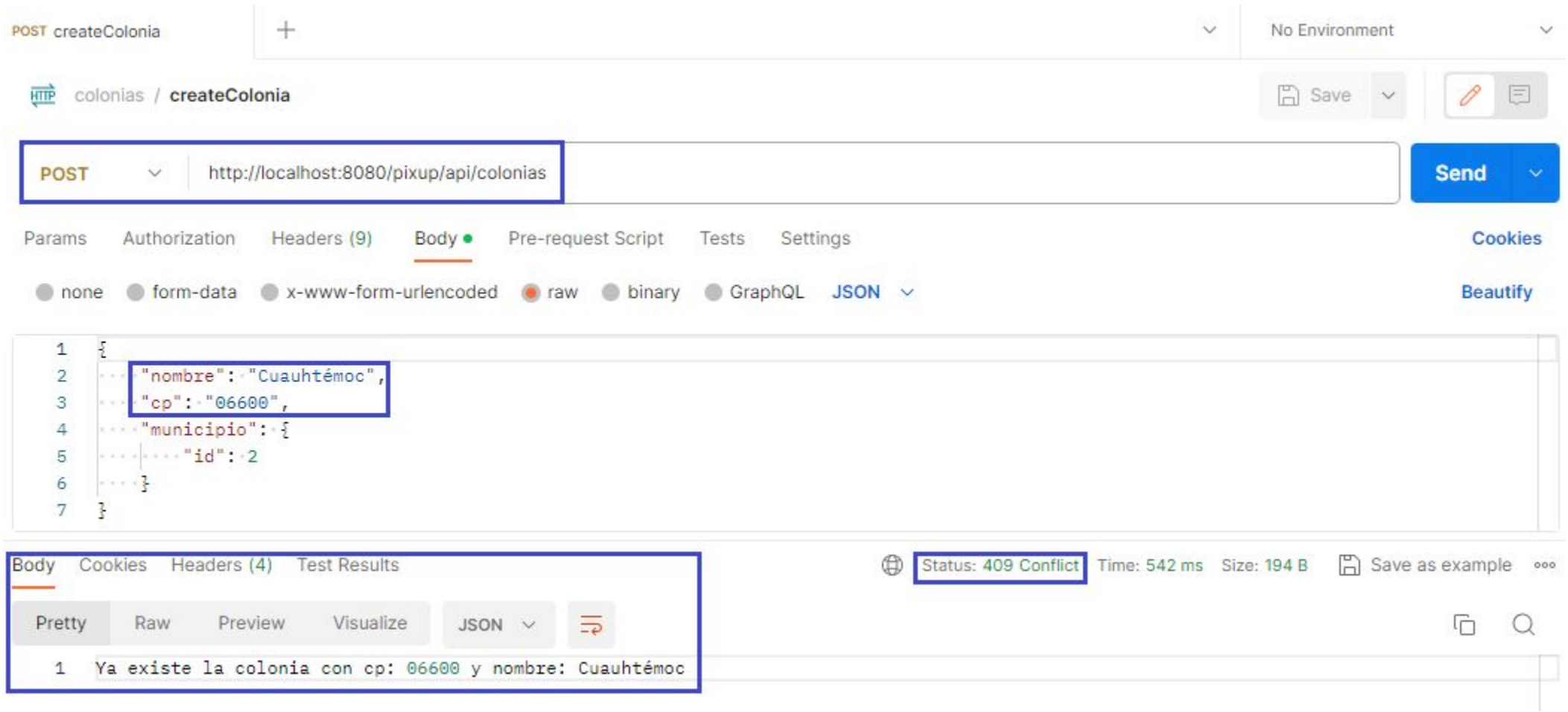
Capa API: ColoniaResource

```java
@Override
public Response createColonia(Colonia colonia) {
    try {
        Colonia coloniaCreada = coloniaService.crearColonia(colonia);
        return Response
                .status(Response.Status.CREATED)
                .entity(coloniaCreada)
                .build();
    } catch (Exception e) {
        if (e.getCause() instanceof ColoniaAlreadyExistsException) {
            return Response
                    .status(Response.Status.CONFLICT)
                    .entity(e.getCause().getMessage())
                    .build();
        }
        return Response
                .status(Response.Status.PRECONDITION_REQUIRED)
                .entity(e.getCause().getMessage())
                .build();
    }
}
```

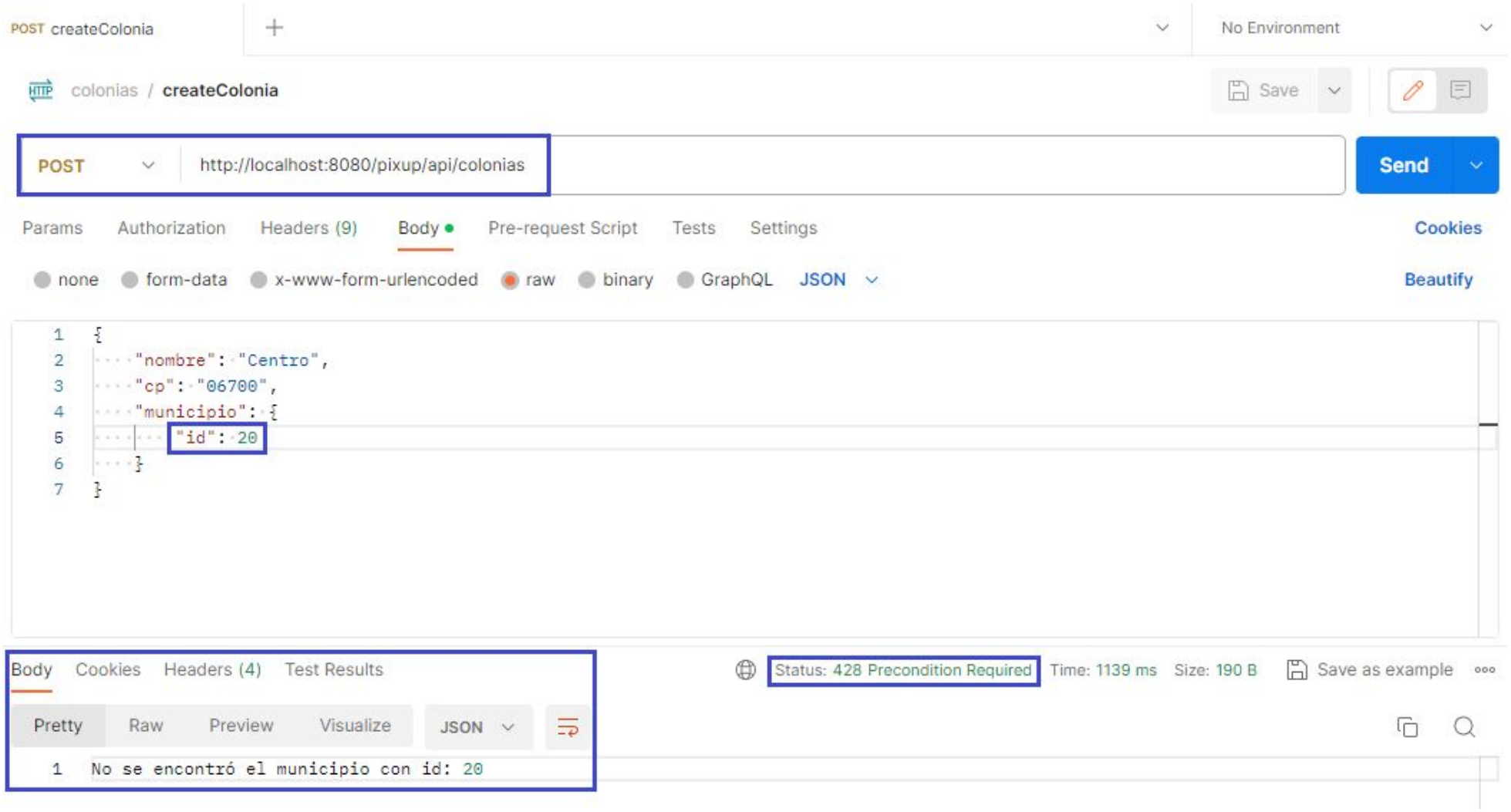
Postman: ColoniaResource POST createColonia



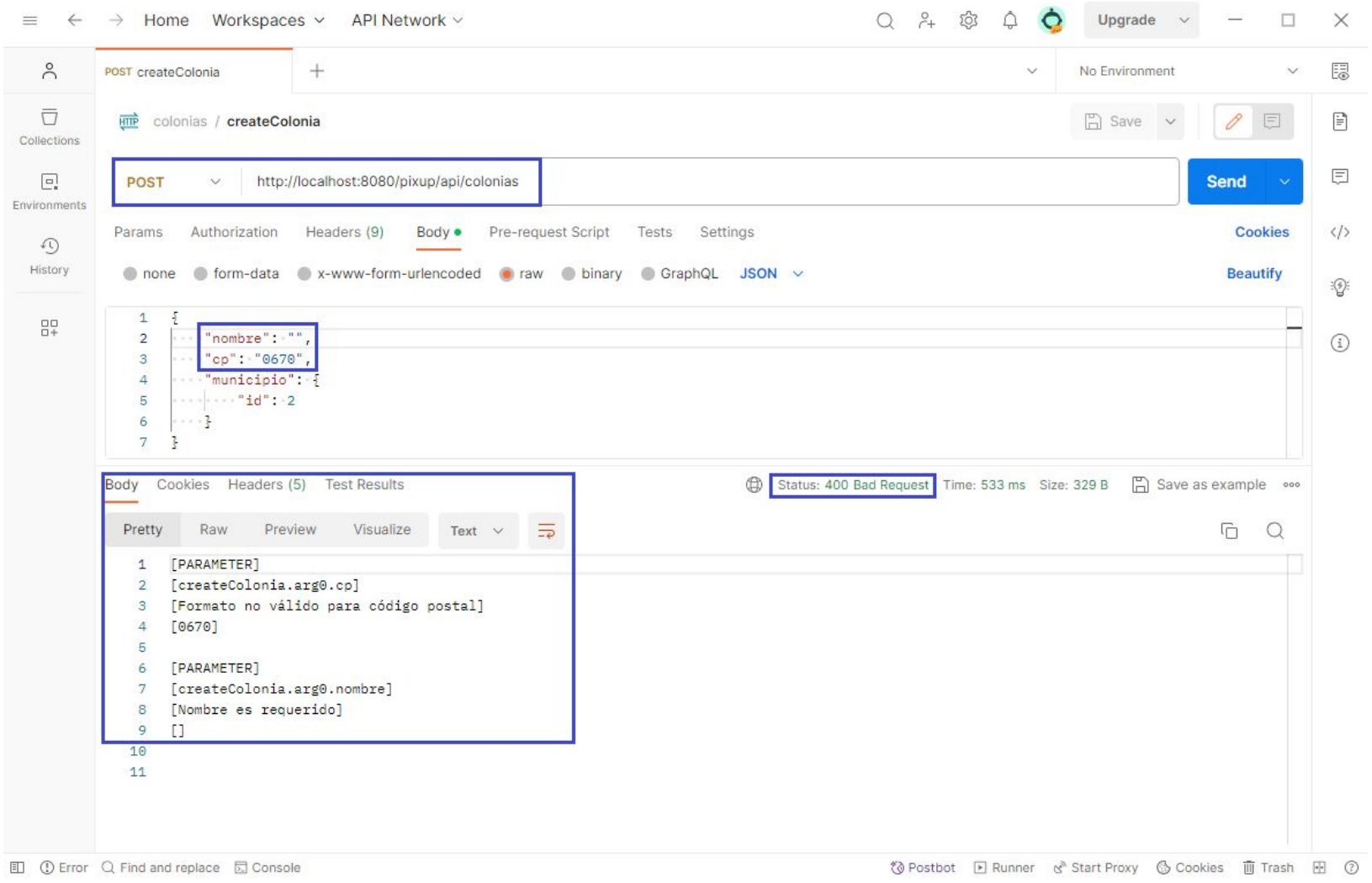
Postman: ColoniaResource POST createColonia – UK duplicated



Postman: ColoniaResource POST createColonia – Municipio inexistente



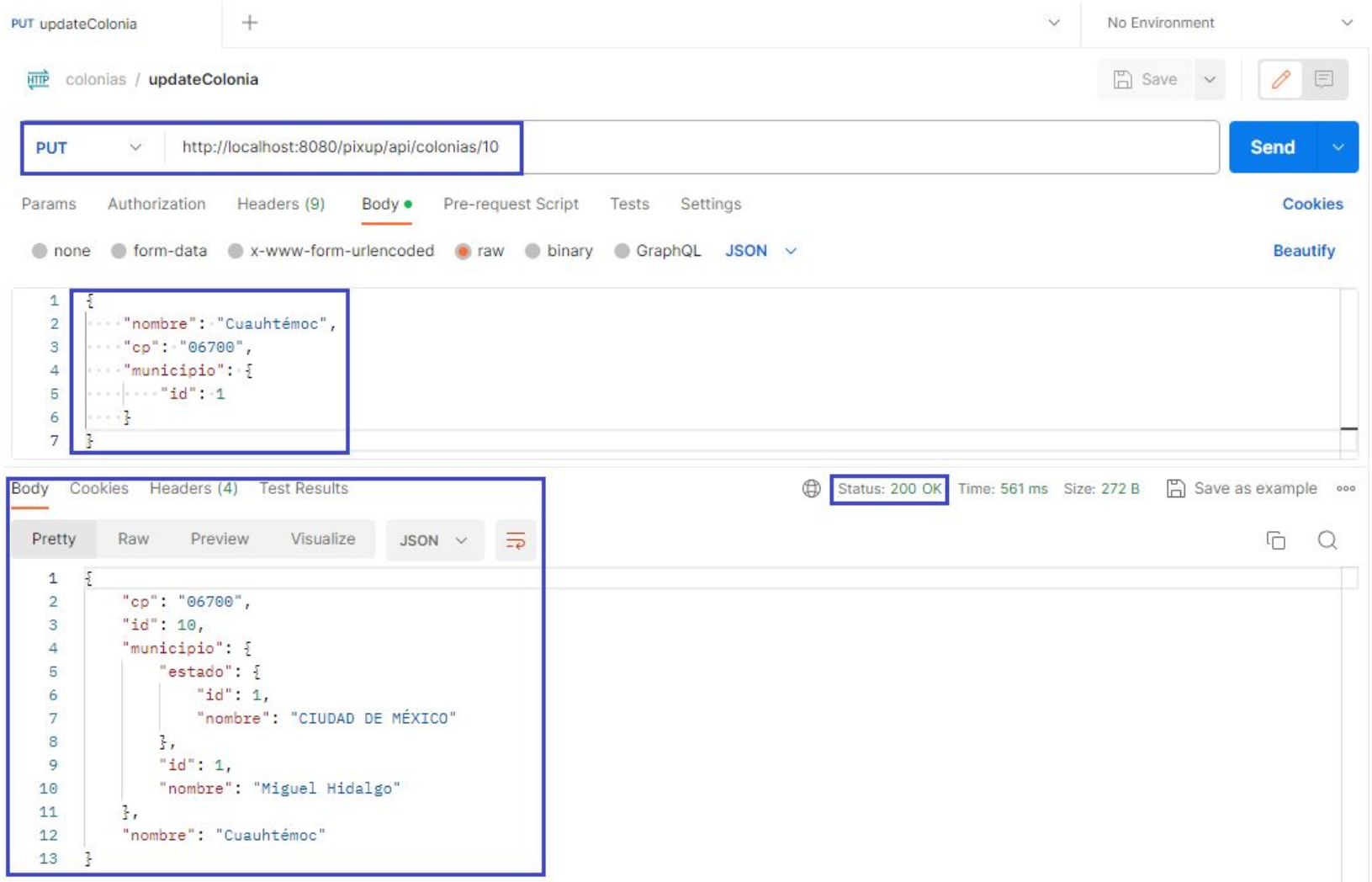
Postman: ColoniaResource POST createColonia - Validator



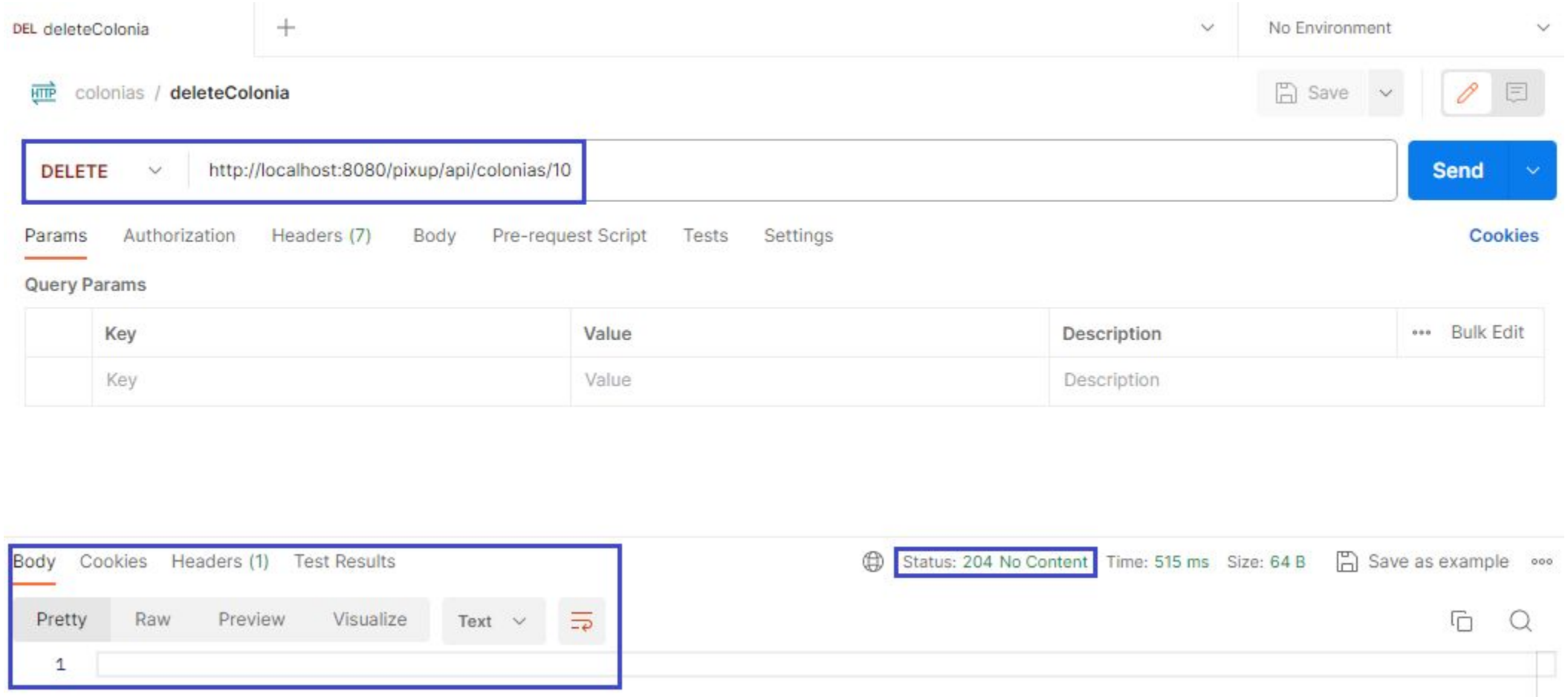
Gestión de Excepciones Centralizada

```java
@Override
public Colonia getColoniaById(Integer id) {
    return coloniaService.obtenerColoniaPorId(id);
}

@Override
public Response createColonia(Colonia colonia) {
    Colonia coloniaCreada = coloniaService.crearColonia(colonia);
    return Response.status(Response.Status.CREATED).entity(coloniaCreada).build();
}
```

Postman: ColoniaResource PUT updateColonia



Postman: ColoniaResource DELETE deleteColonia



DTO (Data Transfer Object)

- Patrón de diseño introducido por Martin Fowler.
- Permite desacoplar el modelo de dominio de la capa de presentación, **permitiendo que puedan evolucionar de manera independiente**.
- Los DTOs son POJOs, es decir, estructuras de datos que no contienen lógica de negocio.
- Su principal propósito es reducir el número de peticiones al servidor agrupando varios parámetros en una sola llamada, esto reduce la sobrecarga de red en operaciones remotas.
- En el diseño de APIs el DTO representa el modelo enviado desde/para el cliente API (request/response).
- Los datos son mapeados desde el modelo de dominio hacia el DTO (y viceversa) a través de un componente mapper en la capa de presentación o fachada.

Postman: ColoniaResource GET getColoniasbyCp – sin DTO


```
[
    {
        "cp": "06400",
        "id": 1,
        "municipio": {
            "estado": {
                "id": 1,
                "nombre": "CIUDAD DE MÉXICO"
            },
            "id": 2,
            "nombre": "Cuauhtémoc"
        },
        "nombre": "Santa María la Ribera"
    },
    {
        "cp": "06400",
        "id": 2,
        "municipio": {
            "estado": {
                "id": 1,
                "nombre": "CIUDAD DE MÉXICO"
            },
            "id": 2,
            "nombre": "Cuauhtémoc"
        },
        "nombre": "San Rafael"
    }
]
```

ColoniaDTO

```java
@Data
@AllArgsConstructor
public class ColoniaDTO {

    private Integer id;
    private String nombre;
    private String cp;
    private String municipio;
    private String estado;

}
```

ColoniaMapper

```java
@Singleton
public class ColoniaMapper {

    public ColoniaDTO toDto(Colonia colonia){
        return new ColoniaDTO(
                colonia.getId(), colonia.getNombre(), colonia.getCp(),
                colonia.getMunicipio().getNombre(),
                colonia.getMunicipio().getEstado().getNombre());
    }

}
```

ColoniaMapper Aplicado

```java
@Override
public Collection<ColoniaDTO> getColoniasByCp(String cp) {
    return coloniaRepository.findByCp(cp)
            .stream()
            .map(colonia -> coloniaMapper.toDto(colonia))
            .toList();
}
```

Postman: ColoniaResource GET getColoniasbyCp - con DTO


```
[
    {
        "cp": "06400",
        "estado": "CIUDAD DE MÉXICO",
        "id": 1,
        "municipio": "Cuauhtémoc",
        "nombre": "Santa María la Ribera"
    },
    {
        "cp": "06400",
        "estado": "CIUDAD DE MÉXICO",
        "id": 2,
        "municipio": "Cuauhtémoc",
        "nombre": "San Rafael"
    }
]
```

Contacto

Uriel Hernández

Solution Architect

urielhdezorozco@yahoo.com.mx

Redes sociales:

https://www.linkedin.com/in/juho-mex