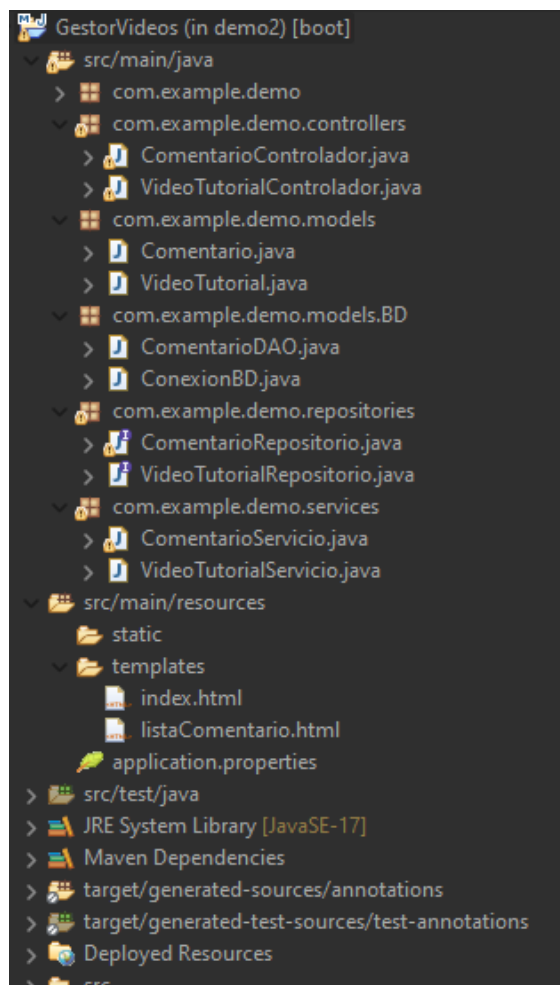


Informe: prueba perfil tecnólogo número 3

Nombre: Cristian Andres Torres Chaparro

- Tecnologías utilizadas:
 - Lenguajes de programación: Java 11 con el framework spring boot.
 - Gestor de bases de datos: MySQL.
 - Lenguajes informáticos: HTML5, CSS3, Bootstrap.
- Estructura empleada: La arquitectura empleada fue Modelo-Vista-Controlador (MVC) para organizar y separar las responsabilidades del aplicativo. Se dividió la estructura de la aplicación en tres componentes principales: el Modelo, la Vista y el Controlador.



- Microservicios:

La aplicación contiene dos microservicios para dividir responsabilidades, el primer microservicio gestiona la información asociada a los videos (crear video, eliminar video, modificar video, filtrar videos) y el segundo gestiona los comentarios asociados a los videos (crear comentario, eliminar comentario, modificar comentario, listar comentarios).

```

@Service
public class ComentarioServicio {
    @Autowired
    private ComentarioRepositorio comentarioRepositorio;

    //Retorna una lista de todos los videos
    public ArrayList<Comentario> obtenerComentario(){
        return (ArrayList<Comentario>) comentarioRepositorio.findAll();
    }

    //Retorna una lista de todos los comentarios
    public ArrayList<Comentario> obtenerListaComentario(){
        return (ArrayList<Comentario>) comentarioRepositorio.findAll();
    }

    //Modificar un comentario
    //Permite guardar un video tutorial

```

```

@Service
public class VideoTutorialServicio {
    @Autowired
    private VideoTutorialRepositorio videoTutorialRepositorio;

    //Retorna una lista de todos los videos
    public ArrayList<VideoTutorial> obtenerListaVideos(){
        return (ArrayList<VideoTutorial>) videoTutorialRepositorio.findAll();
    }

    //Permite guardar un video tutorial
    public VideoTutorial guardarVideo(VideoTutorial video) {
        return videoTutorialRepositorio.save(video);
    }

    //Permite modificar un video
    public Optional<VideoTutorial> obtenerVideoTutorialId(Integer idvideotutorial){
        return videoTutorialRepositorio.findById(idvideotutorial);
    }

    //Permite eliminar video tutorial
    public boolean eliminarVideo(Integer idvideotutorial) {
        try {
            videoTutorialRepositorio.deleteById(idvideotutorial);
            return true;
        } catch (Exception e) {
            return false;
        }
    }
}

```

- API REST: Se utilizaron las operaciones HTTP estándar (GET, POST, PUT, DELETE) para realizar acciones en los recursos asociados a los vídeos y comentarios.

```

@RestController
@RequestMapping("/comentario")
public class ComentarioControlador {
    private @Autowired ComentarioServicio comentarioServicio;

    //Permite controlar la obtencion del listado de todos los videos
    @GetMapping()
    public ArrayList<Comentario> obtenerListaComentario(){
        return this.comentarioServicio.obtenerListaComentario();
    }

    //Permite controlar el modificar un video
    @PostMapping()
    public Comentario modificarComentario(@RequestBody Comentario comentario) {
        return this.comentarioServicio.modificarComentario(comentario);
    }

    //Permite guardar un comentarios asociado a un libro
    @PostMapping("/guardarComentario")
    public String guardarComentario(@Validated Comentario comentario, Model modelo) {
        ComentarioDAO comentarioDAO = ComentarioDAO.getComentarioDAO();
        comentarioDAO.guardarComentario(comentario);
        return "redirect://listarCitasMedicas";
    }

    //Permite controlar el filtrado de un comentario por id
    @GetMapping("path =("/{idComentario}")")
    public Optional<Comentario> obtenerComentarioId(@PathVariable("idComentario") Integer idComentario){

```

```

@Controller
public class VideoTutorialControlador {
    @Autowired
    private VideoTutorialServicio videoTutorialServicio;

    //Permite controlar la obtencion del listado de todos los videos
    @RequestMapping("/")
    public String obtenerListaVideos(Model modelo) {
        List<VideoTutorial> listaVideoTutorial = this.videoTutorialServicio.obtenerListaVideos();
        modelo.addAttribute("misVideoTutorial", listaVideoTutorial);
        return "index";
    }

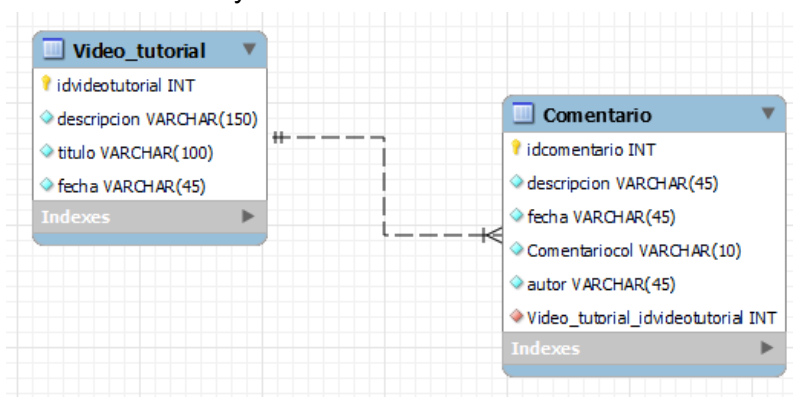
    //Permite controlar el guardado de un video
    @PostMapping()
    public VideoTutorial guardarVideo(@RequestBody VideoTutorial video) {
        return this.videoTutorialServicio.guardarVideo(video);
    }

    //Permite controlar el filtrado de un videotutorial por id
    @GetMapping(path =("/{idvideotutorial}") )
    public Optional<VideoTutorial> obtenerVideoTutorialId(@PathVariable("idvideotutorial") Integer id) {
        return this.videoTutorialServicio.obtenerVideoTutorialId(idvideotutorial);
    }

    //Permite controlar la eliminacion de un videotutorial por id
    @DeleteMapping(path =("/{id}") )
    public String eliminarVideoTutorial(@PathVariable("id") Integer idvideotutorial) {
        boolean ok = this.videoTutorialServicio.eliminarVideo(idvideotutorial);
        return ok ? "Video eliminado" : "Error al eliminar video";
    }
}

```

- Patrones de diseño utilizados:
 - a. Patrones de diseño de inyección de dependencias.
 - b. Patrones de diseño singleton para tener un único ejemplar a la conexión a la base de datos.
 - c. Patrón de diseño DAO.
- Base de datos: se empleó una base de datos relacional haciendo uso del gestor de bases de datos MySQL.



```

DROP SCHEMA IF EXISTS `BDaulaVirtual`;
CREATE SCHEMA IF NOT EXISTS `BDaulaVirtual` DEFAULT CHARACTER SET utf8 ;

use BDaulaVirtual;

CREATE TABLE IF NOT EXISTS `BDaulaVirtual`.`Video_tutorial` (
  `idvideotutorial` int NOT NULL AUTO_INCREMENT,
  `descripcion` VARCHAR(150) NOT NULL,
  `titulo` VARCHAR(100) NOT NULL,
  `fecha` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`idvideotutorial`))
ENGINE = InnoDB;

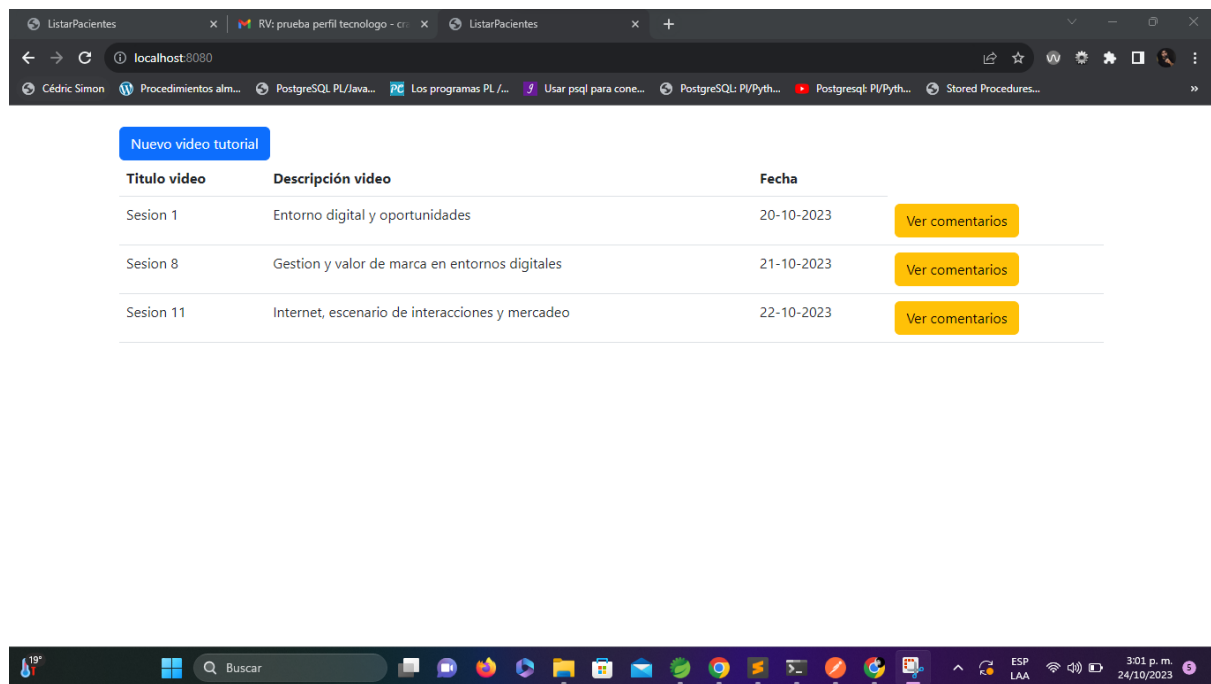
Insert into Video_tutorial values(1,"Entorno digital y oportunidades", "Sesion 1","20-10-2023");
Insert into Video_tutorial values(2,"Gestion y valor de marca en entornos digitales", "Sesion 8","21-10-2023");
Insert into Video_tutorial values(3,"Internet, escenario de interacciones y mercadeo", "Sesion 11", "22-10-2023");

CREATE TABLE IF NOT EXISTS `BDaulaVirtual`.`Comentario` (
  `idcomentario` int NOT NULL AUTO_INCREMENT,
  `descripcion` TEXT NOT NULL,
  `fecha` VARCHAR(10) NOT NULL,
  `autor` VARCHAR(45) NOT NULL,
  `fk_idvideotutorial` int NOT NULL,
  PRIMARY KEY (`idcomentario`))
ENGINE = InnoDB;

Insert into Comentario values(1,"Comentario 1...", "20-10-2023", "Cristian Torres", 1);
Insert into Comentario values(2,"Comentario 2...", "21-10-2023", "Andres Chaparro", 1);
Insert into Comentario values(3,"Comentario 3...", "22-10-2023", "Camilo Torres", 1);

```

- Funcionamiento de la aplicación:
 1. Mostrar el listado de todos los videotutoriales.

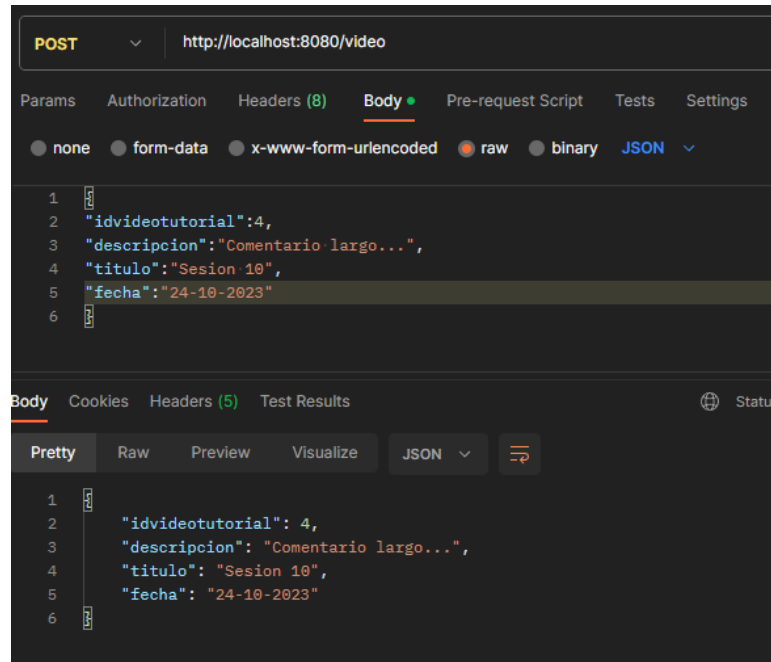


```

mysql> select * from Video_tutorial;
+-----+-----+-----+-----+
| idvideotutorial | descripcion | titulo | fecha |
+-----+-----+-----+-----+
| 1 | Entorno digital y oportunidades | Sesion 1 | 20-10-2023 |
| 2 | Gestion y valor de marca en entornos digitales | Sesion 8 | 21-10-2023 |
| 3 | Internet, escenario de interacciones y mercadeo | Sesion 11 | 22-10-2023 |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)

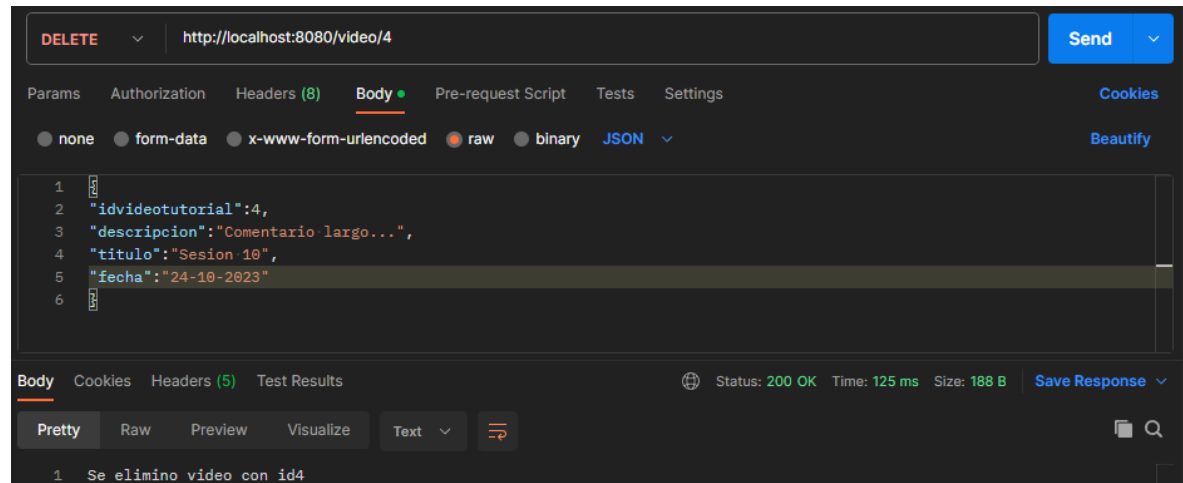
```

2. Crear un nuevo video tutorial:



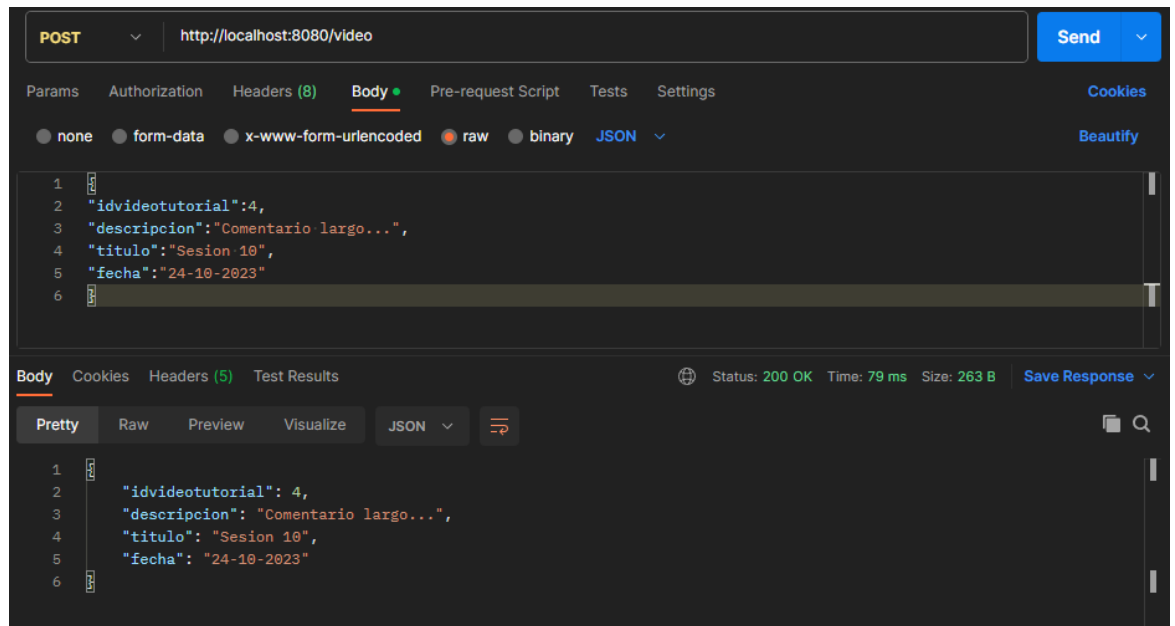
```
mysql> select * from Video_tutorial;
+-----+-----+-----+-----+
| idvideotutorial | descripcion | titulo | fecha |
+-----+-----+-----+-----+
| 1 | Entorno digital y oportunidades | Sesion 1 | 20-10-2023 |
| 2 | Gestion y valor de marca en entornos digitales | Sesion 8 | 21-10-2023 |
| 3 | Internet, escenario de interacciones y mercadeo | Sesion 11 | 22-10-2023 |
| 4 | Comentario largo... | Sesion 10 | 24-10-2023 |
+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

3. Eliminar un video:

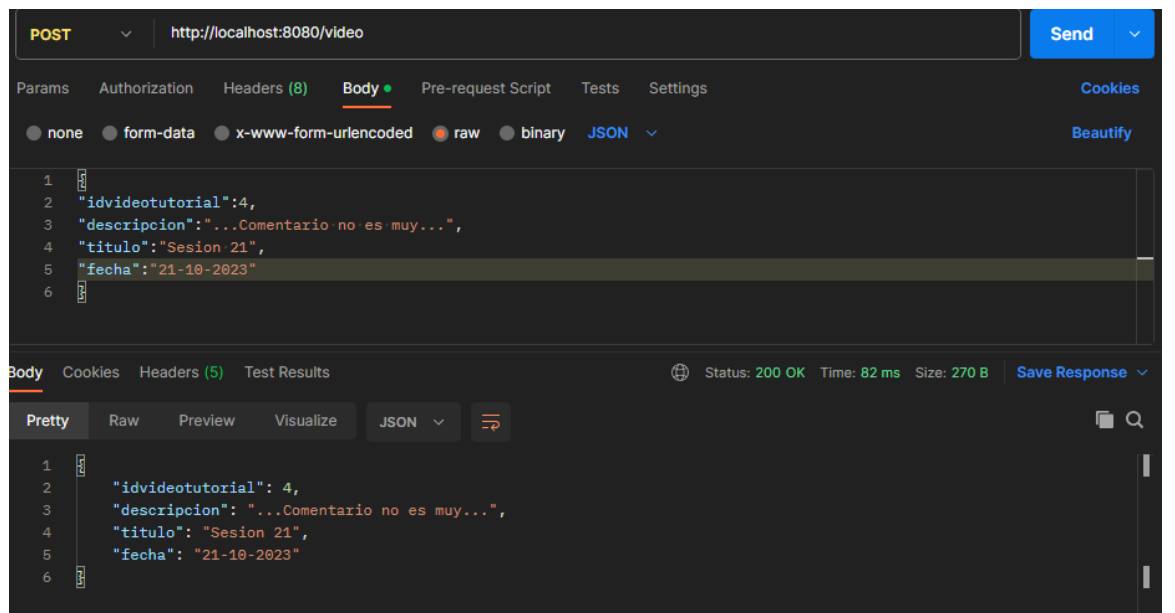


```
mysql> select * from Video_tutorial;
+-----+-----+-----+-----+
| idvideotutorial | descripcion | titulo | fecha |
+-----+-----+-----+-----+
| 1 | Entorno digital y oportunidades | Sesion 1 | 20-10-2023 |
| 2 | Gestion y valor de marca en entornos digitales | Sesion 8 | 21-10-2023 |
| 3 | Internet, escenario de interacciones y mercadeo | Sesion 11 | 22-10-2023 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

4. Modificar un video:
- Se crea un nuevo registro de video.



- Se modifican todos los datos.



```
mysql> select * from Video_tutorial;
```

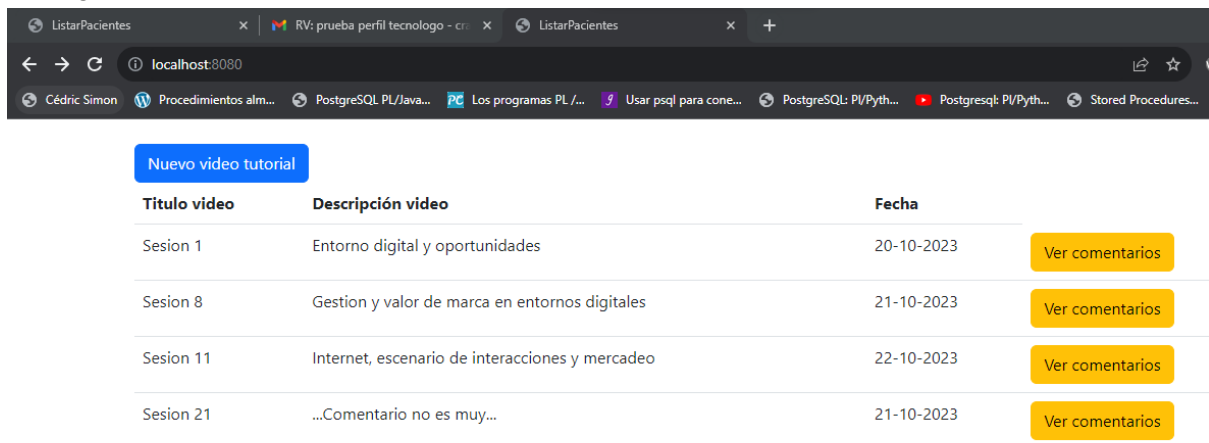
	idvideotutorial	descripcion	titulo	fecha
1	1	Entorno digital y oportunidades	Sesion 1	20-10-2023
2	2	Gestion y valor de marca en entornos digitales	Sesion 8	21-10-2023
3	3	Internet, escenario de interacciones y mercadeo	Sesion 11	22-10-2023
4	4	Comentario largo...	Sesion 10	24-10-2023

4 rows in set (0.02 sec)

```
mysql> select * from Video_tutorial;
+-----+-----+-----+-----+
| idvideotutorial | descripcion | titulo | fecha |
+-----+-----+-----+-----+
| 1 | Entorno digital y oportunidades | Sesion 1 | 20-10-2023 |
| 2 | Gestion y valor de marca en entornos digitales | Sesion 8 | 21-10-2023 |
| 3 | Internet, escenario de interacciones y mercadeo | Sesion 11 | 22-10-2023 |
| 4 | ...Comentario no es muy... | Sesion 21 | 21-10-2023 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

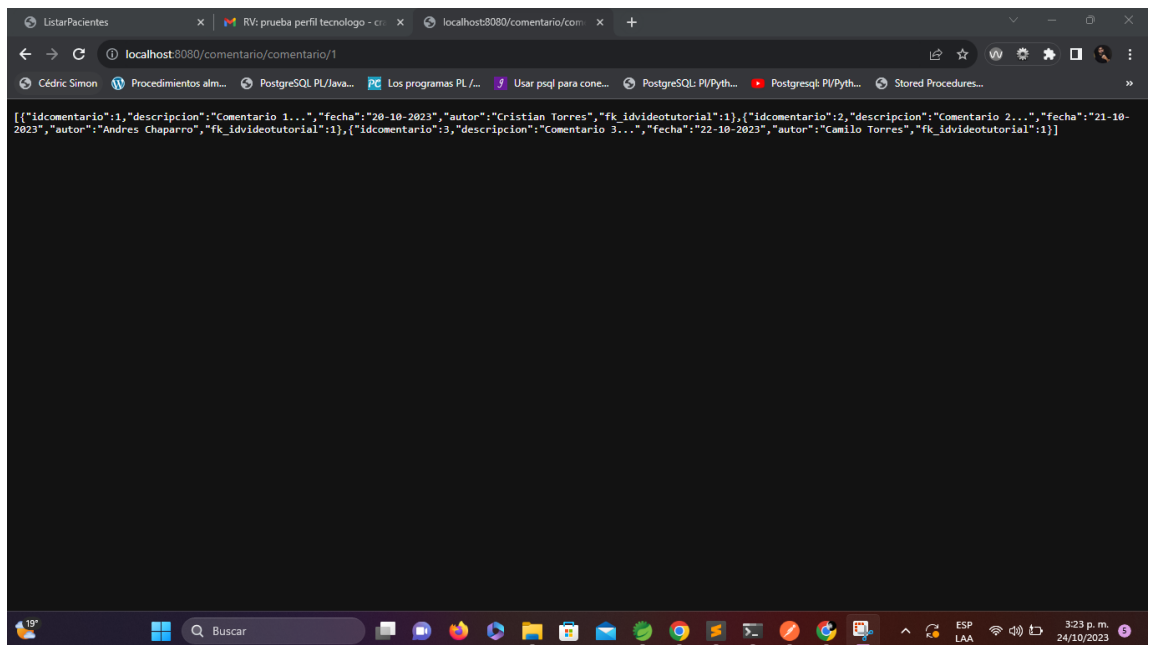
mysql>
```

- Mostrar el listado de todos los comentarios asociados a un video.
 - Vista general, se selecciona un video para ver sus comentarios asociados.



The screenshot shows a web browser with the URL `localhost:8080`. The page has a header with a button labeled "Nuevo video tutorial". Below the header is a table with three columns: "Título video", "Descripción video", and "Fecha". There are four rows of data, each with a "Ver comentarios" button to its right.

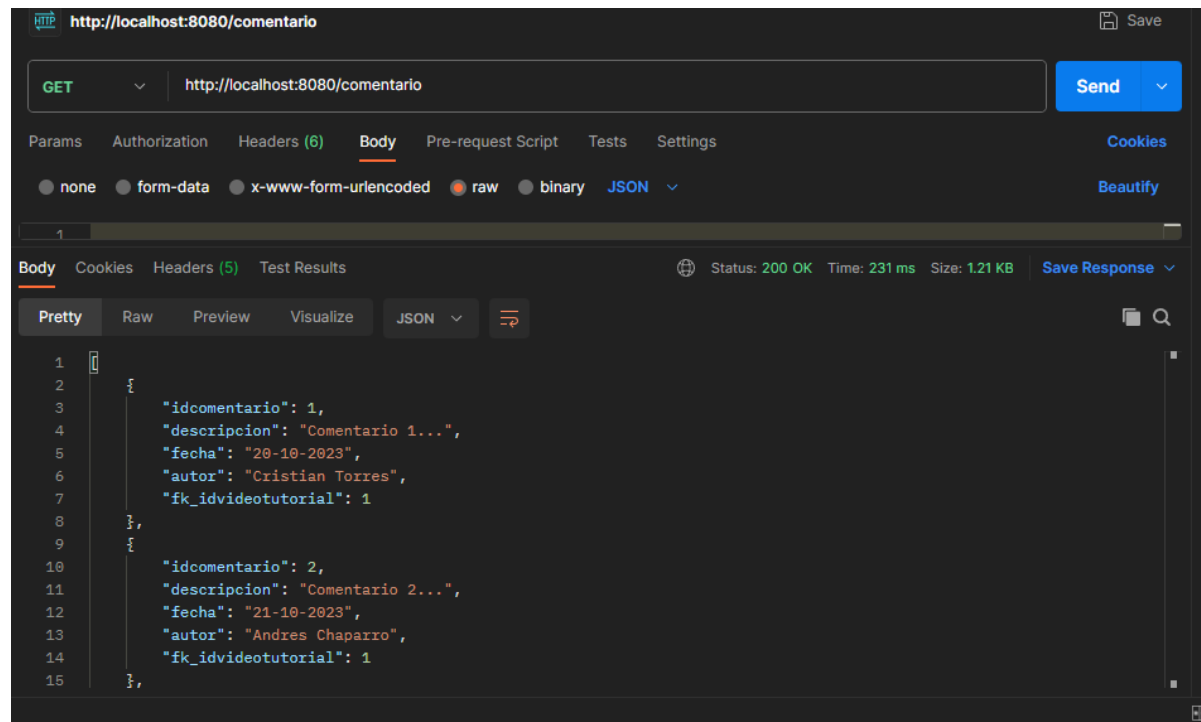
Título video	Descripción video	Fecha	
Sesion 1	Entorno digital y oportunidades	20-10-2023	Ver comentarios
Sesion 8	Gestion y valor de marca en entornos digitales	21-10-2023	Ver comentarios
Sesion 11	Internet, escenario de interacciones y mercadeo	22-10-2023	Ver comentarios
Sesion 21	...Comentario no es muy...	21-10-2023	Ver comentarios



The screenshot shows a web browser with the URL `localhost:8080/comentario/comentario/1`. The page displays a JSON array of comment objects. The JSON is as follows:

```
[{"idcomentario":1,"descripcion":"Comentario 1...", "fecha":"20-10-2023", "autor":"Cristian Torres", "fk_idvideotutorial":1}, {"idcomentario":2, "descripcion":"Comentario 2...", "fecha":"21-10-2023", "autor":"Andres Chaparro", "fk_idvideotutorial":1}, {"idcomentario":3, "descripcion":"Comentario 3...", "fecha":"22-10-2023", "autor":"Camilo Torres", "fk_idvideotutorial":1}]
```

- Ver todos los comentarios almacenados en la base de datos:



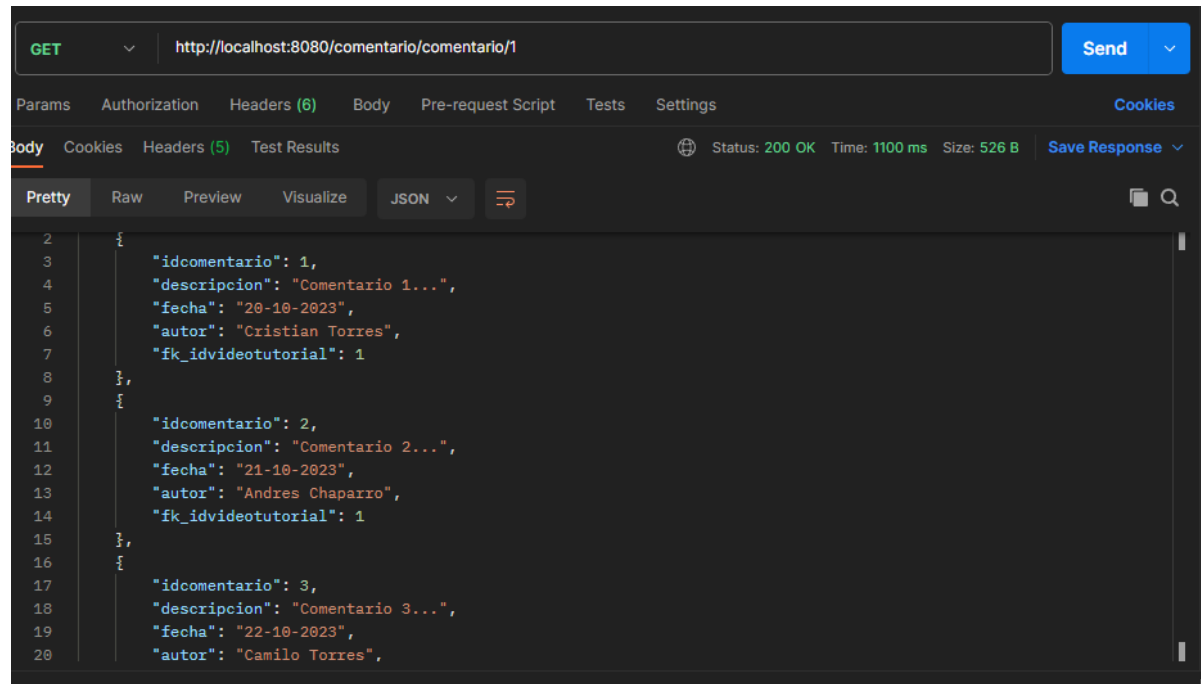
```
mysql> select * from comentario;
```

idcomentario	descripcion	fecha	autor	fk_idvideotutorial
1	Comentario 1...	20-10-2023	Cristian Torres	1
2	Comentario 2...	21-10-2023	Andres Chaparro	1
3	Comentario 3...	22-10-2023	Camilo Torres	1
4	Comentario 4...	23-10-2023	Andres Guti	2
5	Comentario 5...	24-10-2023	Jhojan Rizo	2
6	Comentario 6...	24-10-2023	Miguel Guzman	2
7	Comentario 4...	1-10-2023	Estefania Torres	3
8	Comentario 5...	2-10-2023	Laura Duegnaz	4
9	Comentario 6...	3-10-2023	Angui Vanessa	5

```
9 rows in set (0.02 sec)
```

```
mysql>
```

- c. Filtrar comentarios a partir del video al que está asociado:



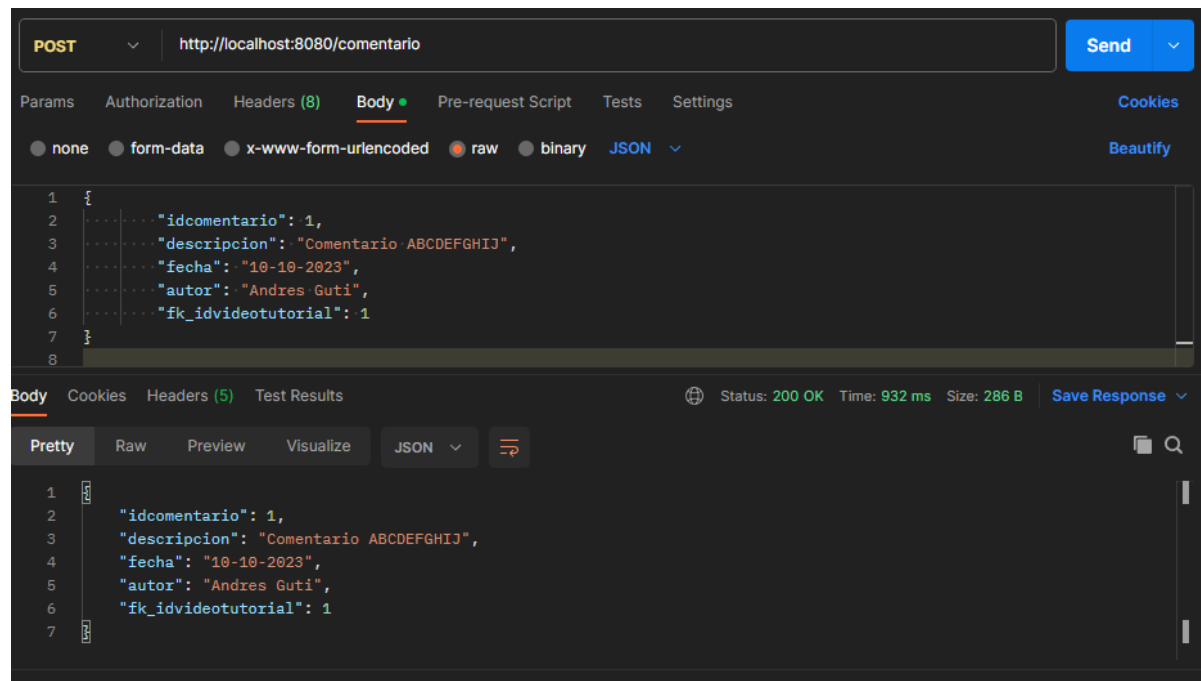
```
mysql> select * from comentario where fk_idvideotutorial = 1;
```

idcomentario	descripcion	fecha	autor	fk_idvideotutorial
1	Comentario 1...	20-10-2023	Cristian Torres	1
2	Comentario 2...	21-10-2023	Andres Chaparro	1
3	Comentario 3...	22-10-2023	Camilo Torres	1

```
3 rows in set (0.00 sec)
```

```
mysql>
```

d. Modificar comentario:



```
mysql> select * from comentario where fk_idvideotutorial = 1;
```

idcomentario	descripcion	fecha	autor	fk_idvideotutorial
1	Comentario 1...	20-10-2023	Cristian Torres	1
2	Comentario 2...	21-10-2023	Andres Chaparro	1
3	Comentario 3...	22-10-2023	Camilo Torres	1

3 rows in set (0.00 sec)

```
mysql>
```

```
mysql> select * from comentario where fk_idvideotutorial = 1;
```

idcomentario	descripcion	fecha	autor	fk_idvideotutorial
1	Comentario ABCDEFGHIJ	10-10-2023	Andres Guti	1
2	Comentario 2...	21-10-2023	Andres Chaparro	1
3	Comentario 3...	22-10-2023	Camilo Torres	1

3 rows in set (0.00 sec)

```
mysql>
```

e. Eliminar comentario:

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** http://localhost:8080/comentario/1
- Body:** A JSON object with the following fields:

```
{  "idcomentario": 1,  "descripcion": "Comentario ABCDEFGHIJ",  "fecha": "10-10-2023",  "autor": "Andres Guti",  "fk_idvideotutorial": 1}
```
- Status:** 200 OK
- Time:** 196 ms
- Size:** 193 B
- Response:** Se elimino comentario con id1

```
mysql> select * from comentario where fk_idvideotutorial = 1;
```

idcomentario	descripcion	fecha	autor	fk_idvideotutorial
1	Comentario ABCDEFGHIJ	10-10-2023	Andres Guti	1
2	Comentario 2...	21-10-2023	Andres Chaparro	1
3	Comentario 3...	22-10-2023	Camilo Torres	1

```
3 rows in set (0.00 sec)
```



```
mysql> select * from comentario where fk_idvideotutorial = 1;
```

idcomentario	descripcion	fecha	autor	fk_idvideotutorial
2	Comentario 2...	21-10-2023	Andres Chaparro	1
3	Comentario 3...	22-10-2023	Camilo Torres	1

```
2 rows in set (0.00 sec)
```

```
mysql>
```

f.