

Controller

```
> > CRUD [boot] [crud-java-mysql main]
v demo [boot]
  v src/main/java
    v com.SegundoExamenJuanFlores.demo
      > JuanFloresApplication.java
    v com.SegundoExamenJuanFlores.demo.Controller
      > EventoController.java
    > com.SegundoExamenJuanFlores.demo.DTO
    > com.SegundoExamenJuanFlores.demo.Mapper
    > com.SegundoExamenJuanFlores.demo.Model
    > com.SegundoExamenJuanFlores.demo.Repository
    > com.SegundoExamenJuanFlores.demo.Service
    > com.SegundoExamenJuanFlores.demo.ServiceImpl
  > src/main/resources
  > src/test/java
  > JRE System Library [JavaSE-21]
  > Maven Dependencies
  > target/generated-sources/annotations
  > target/generated-test-sources/test-annotations
  > src
  > target
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
  > Juan_Carlos [boot]
  > JuanFlores [boot]
    onetmany
  > > programacionv (in SistemasClases2025) [boot] [SistemasClases2025 feature/02]
  > record [boot]
```

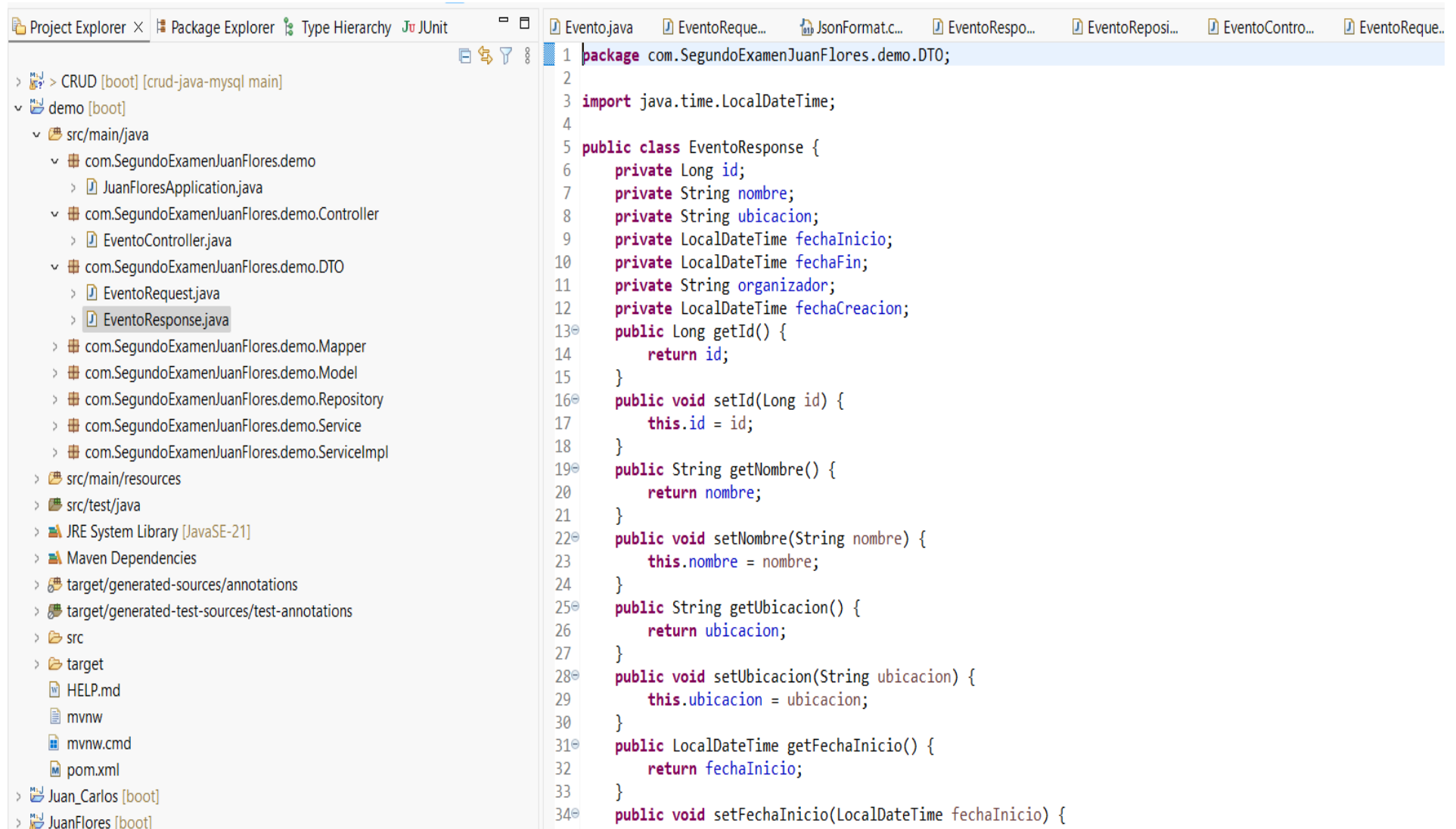
```
--
22
23 @RestController
24 @RequestMapping("/api")
25 public class EventoController {
26
27     @Autowired
28     EventoService eventoService;
29
30     @GetMapping("/eventos/all")
31     public ResponseEntity<Collection<EventoResponse>> getEventos(){
32         return ResponseEntity.ok(eventoService.findAllCliente());
33     }
34
35     @GetMapping("/eventos/{id}")
36     public ResponseEntity<EventoResponse> getEventosById(@PathVariable Long id){
37         return ResponseEntity.ok(eventoService.findByIdEvento(id));
38     }
39
40     @PostMapping("/eventos/guardar")
41     public void saveEventosById(@RequestBody EventoRequest request){
42         eventoService.saveEvento(request);
43     }
44
45
46     @PutMapping("/update/eventos/{id}")
47     public void updateEventosById(@PathVariable Long id, @RequestBody EventoRequest request){
48         eventoService.updateEvento(id, request);
49     }
50
51     @DeleteMapping("/delete/eventos/{id}")
52     public void deleteEventosById(@PathVariable Long id){
53         eventoService.deleteEvento(id);
54     }
55 }
```

DTOS request

The screenshot shows an IDE with a project explorer on the left and a code editor on the right. The project explorer displays a project named 'demo' with a package structure: `com.SegundoExamenJuanFlores.demo`. The package `demo.DTO` is expanded, showing `EventoRequest.java` selected. The code editor shows the following Java code for `EventoRequest`:

```
1 package com.SegundoExamenJuanFlores.demo.DTO;
2
3 import java.time.LocalDateTime;
4
5
6
7 public class EventoRequest {
8
9
10 private Long id;
11 private String nombre;
12 private String ubicacion;
13
14 @JsonFormat(pattern = "yyyy-MM-dd'T'HH:mm:ss")
15 private LocalDateTime fechaInicio;
16
17 @JsonFormat(pattern = "yyyy-MM-dd'T'HH:mm:ss")
18 private LocalDateTime fechaFin;
19
20 private String organizador;
21
22
23
24 public Long getId() {
25     return id;
26 }
27 public void setId(Long id) {
28     this.id = id;
29 }
30 public String getNombre() {
31     return nombre;
32 }
33 public void setNombre(String nombre) {
34     this.nombre = nombre;
35 }
36 public String getUbicacion() {
37     return ubicacion;
38 }
```

Dtos Response



The screenshot displays an IDE interface with two main panels. The left panel, titled 'Project Explorer', shows the project structure. The right panel, titled 'Package Explorer', shows the package structure. The main editor area displays the code for 'EventoResponse.java'.

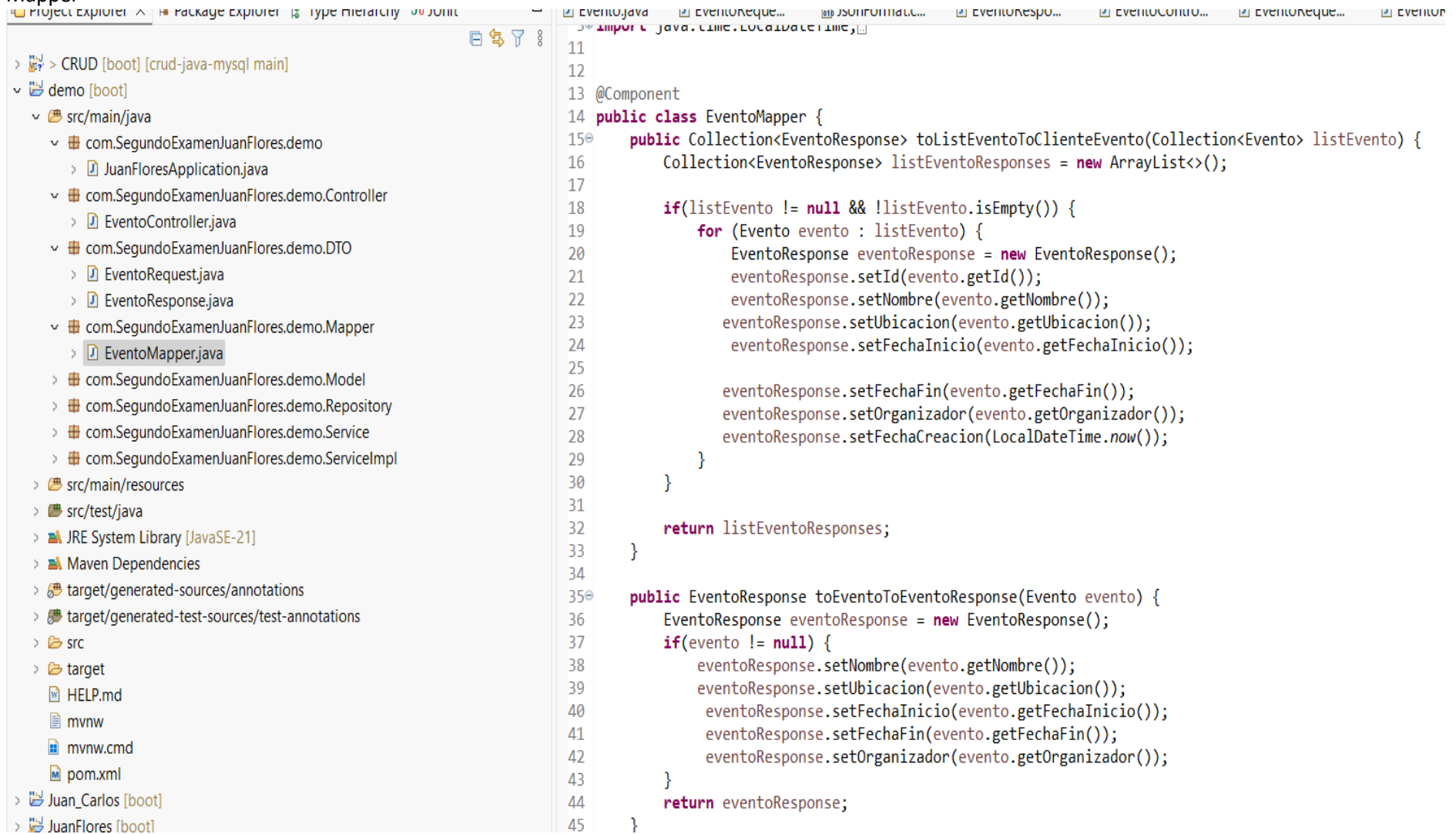
Project Explorer Structure:

- > crud [boot] [crud-java-mysql main]
- ▼ demo [boot]
 - src/main/java
 - com.SegundoExamenJuanFlores.demo
 - JuanFloresApplication.java
 - com.SegundoExamenJuanFlores.demo.Controller
 - EventoController.java
 - com.SegundoExamenJuanFlores.demo.DTO
 - EventoRequest.java
 - EventoResponse.java
 - com.SegundoExamenJuanFlores.demo.Mapper
 - com.SegundoExamenJuanFlores.demo.Model
 - com.SegundoExamenJuanFlores.demo.Repository
 - com.SegundoExamenJuanFlores.demo.Service
 - com.SegundoExamenJuanFlores.demo.ServiceImpl
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-21]
 - Maven Dependencies
 - target/generated-sources/annotations
 - target/generated-test-sources/test-annotations
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - Juan_Carlos [boot]
 - JuanFlores [boot]

Code Editor Content (EventoResponse.java):

```
1 package com.SegundoExamenJuanFlores.demo.DTO;
2
3 import java.time.LocalDateTime;
4
5 public class EventoResponse {
6     private Long id;
7     private String nombre;
8     private String ubicacion;
9     private LocalDateTime fechaInicio;
10    private LocalDateTime fechaFin;
11    private String organizador;
12    private LocalDateTime fechaCreacion;
13    public Long getId() {
14        return id;
15    }
16    public void setId(Long id) {
17        this.id = id;
18    }
19    public String getNombre() {
20        return nombre;
21    }
22    public void setNombre(String nombre) {
23        this.nombre = nombre;
24    }
25    public String getUbicacion() {
26        return ubicacion;
27    }
28    public void setUbicacion(String ubicacion) {
29        this.ubicacion = ubicacion;
30    }
31    public LocalDateTime getFechaInicio() {
32        return fechaInicio;
33    }
34    public void setFechaInicio(LocalDateTime fechaInicio) {
```

Mapper



The image shows an IDE window with two main panes. The left pane is the Project Explorer, showing a Maven project structure. The right pane is the Code Editor, showing the implementation of the `EventoMapper` class.

Project Explorer (Left Pane):

- > `demo [boot]`
 - src/main/java
 - com.SegundoExamenJuanFlores.demo
 - JuanFloresApplication.java
 - EventoController.java
 - EventoRequest.java
 - EventoResponse.java
 - EventoMapper.java**
 - com.SegundoExamenJuanFlores.demo.Model
 - com.SegundoExamenJuanFlores.demo.Repository
 - com.SegundoExamenJuanFlores.demo.Service
 - com.SegundoExamenJuanFlores.demo.ServiceImpl
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-21]
 - Maven Dependencies
 - target/generated-sources/annotations
 - target/generated-test-sources/test-annotations
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
- > Juan_Carlos [boot]
- > JuanFlores [boot]

Code Editor (Right Pane):

The code editor shows the `EventoMapper` class, which is annotated with `@Component`. It implements two methods: `toListEventoToClienteEvento` and `toEventoToEventoResponse`.

```
11 import java.time.LocalDate;
12
13 @Component
14 public class EventoMapper {
15     public Collection<EventoResponse> toListEventoToClienteEvento(Collection<Evento> listEvento) {
16         Collection<EventoResponse> listEventoResponses = new ArrayList<>();
17
18         if(listEvento != null && !listEvento.isEmpty()) {
19             for (Evento evento : listEvento) {
20                 EventoResponse eventoResponse = new EventoResponse();
21                 eventoResponse.setId(evento.getId());
22                 eventoResponse.setNombre(evento.getNombre());
23                 eventoResponse.setUbicacion(evento.getUbicacion());
24                 eventoResponse.setFechaInicio(evento.getFechaInicio());
25
26                 eventoResponse.setFechaFin(evento.getFechaFin());
27                 eventoResponse.setOrganizador(evento.getOrganizador());
28                 eventoResponse.setFechaCreacion(LocalDate.now());
29             }
30         }
31
32         return listEventoResponses;
33     }
34
35     public EventoResponse toEventoToEventoResponse(Evento evento) {
36         EventoResponse eventoResponse = new EventoResponse();
37         if(evento != null) {
38             eventoResponse.setNombre(evento.getNombre());
39             eventoResponse.setUbicacion(evento.getUbicacion());
40             eventoResponse.setFechaInicio(evento.getFechaInicio());
41             eventoResponse.setFechaFin(evento.getFechaFin());
42             eventoResponse.setOrganizador(evento.getOrganizador());
43         }
44         return eventoResponse;
45     }
46 }
```

Model

```
> > CRUD [boot] [crud-java-mysql main]
v demo [boot]
  v src/main/java
    v com.SegundoExamenJuanFlores.demo
      > JuanFloresApplication.java
    v com.SegundoExamenJuanFlores.demo.Controller
      > EventoController.java
    v com.SegundoExamenJuanFlores.demo.DTO
      > EventoRequest.java
      > EventoResponse.java
    v com.SegundoExamenJuanFlores.demo.Mapper
      > EventoMapper.java
    v com.SegundoExamenJuanFlores.demo.Model
      > Evento.java
    v com.SegundoExamenJuanFlores.demo.Repository
    v com.SegundoExamenJuanFlores.demo.Service
    v com.SegundoExamenJuanFlores.demo.ServiceImpl
  > src/main/resources
  > src/test/java
  > JRE System Library [JavaSE-21]
  > Maven Dependencies
  > target/generated-sources/annotations
  > target/generated-test-sources/test-annotations
  > src
  > target
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
```

```
2
3+import java.time.LocalDateTime;
10
11 @Entity
12
13 @Table(name="evento")
14 public class Evento {
15
16     @Id
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private Long id;
19     private String nombre;
20     private String ubicacion;
21     private LocalDateTime fechaInicio;
22     private LocalDateTime fechaFin;
23     private String organizador;
24     private LocalDateTime fechaCreacion;
25
26     public Long getId() {
27         return id;
28     }
29     public void setId(Long id) {
30         this.id = id;
31     }
32     public String getNombre() {
33         return nombre;
34     }
35     public void setNombre(String nombre) {
36         this.nombre = nombre;
37     }
38     public String getUbicacion() {
39         return ubicacion;
40     }
```

Repository

> > CRUD [boot] [crud-java-mysql main]

▼ demo [boot]

- ▼ src/main/java
 - ▼ com.SegundoExamenJuanFlores.demo
 - > JuanFloresApplication.java
 - ▼ com.SegundoExamenJuanFlores.demo.Controller
 - > EventoController.java
 - ▼ com.SegundoExamenJuanFlores.demo.DTO
 - > EventoRequest.java
 - > EventoResponse.java
 - ▼ com.SegundoExamenJuanFlores.demo.Mapper
 - > EventoMapper.java
 - ▼ com.SegundoExamenJuanFlores.demo.Model
 - > Evento.java
 - ▼ com.SegundoExamenJuanFlores.demo.Repository
 - > EventoRepository.java

```
1 package com.SegundoExamenJuanFlores.demo.Repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5
6
7 public interface EventoRepository extends JpaRepository<Evento, Long>{
8
9 }
10
```

Service

The screenshot displays an IDE with two main panels. The left panel shows the Project Explorer with a tree structure of the project 'crud-java-mysql'. The right panel shows the code editor with the implementation of the 'EventoService' interface.

Project Explorer Structure:

- CRUD [boot] [crud-java-mysql main]
 - demo [boot]
 - src/main/java
 - com.SegundoExamenJuanFlores.demo
 - JuanFloresApplication.java
 - com.SegundoExamenJuanFlores.demo.Controller
 - EventoController.java
 - com.SegundoExamenJuanFlores.demo.DTO
 - EventoRequest.java
 - EventoResponse.java
 - com.SegundoExamenJuanFlores.demo.Mapper
 - EventoMapper.java
 - com.SegundoExamenJuanFlores.demo.Model
 - Evento.java
 - com.SegundoExamenJuanFlores.demo.Repository
 - EventoRepository.java
 - com.SegundoExamenJuanFlores.demo.Service
 - EventoService.java
 - com.SegundoExamenJuanFlores.demo.ServiceImpl
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-21]
 - Maven Dependencies
 - target/generated-sources/annotations

Code Editor Content (EventoService.java):

```
1 package com.SegundoExamenJuanFlores.demo.Service;
2
3 import java.util.Collection;
4
5
6
7
8
9
10
11
12
13
14
15 @Service
16 public interface EventoService {
17
18     Collection<EventoResponse> findAllCliente ();
19
20     //Para buscar por id del cliente
21     EventoResponse findByIdEvento(Long idEvento);
22     //Para crear cliente
23     void saveEvento (EventoRequest request);
24     //Para Actualizar cliente
25     void updateEvento(Long id, EventoRequest request);
26
27     void deleteEvento (Long idEvento);
28
29 }
30
```

ServiceImpl

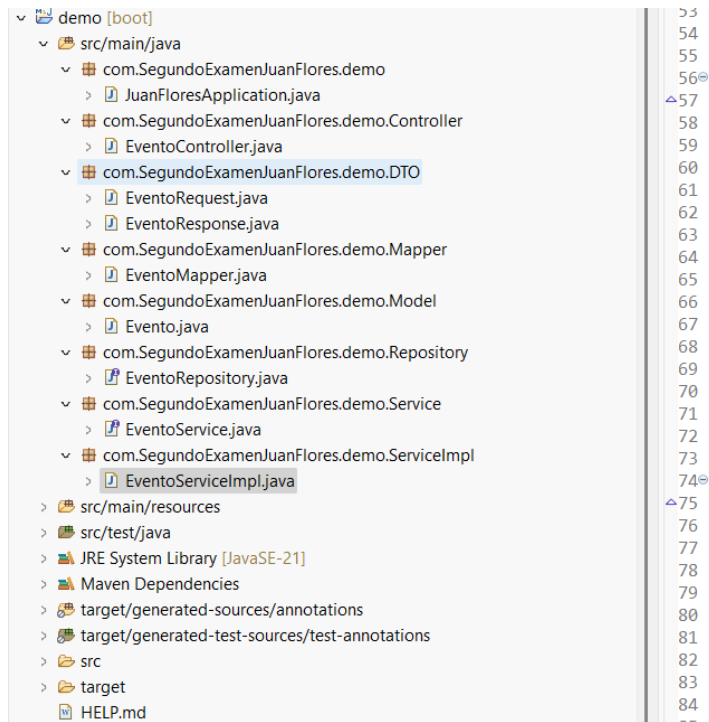
The screenshot shows an IDE with two main panels. The left panel displays the project structure for a Spring Boot application named 'demo'. The right panel shows the implementation of the 'EventoService' interface in the 'EventoServiceImpl' class.

Project Structure (Left Panel):

- demo [boot]
 - src/main/java
 - com.SegundoExamenJuanFlores.demo
 - JuanFloresApplication.java
 - com.SegundoExamenJuanFlores.demo.Controller
 - EventoController.java
 - com.SegundoExamenJuanFlores.demo.DTO
 - EventoRequest.java
 - EventoResponse.java
 - com.SegundoExamenJuanFlores.demo.Mapper
 - EventoMapper.java
 - com.SegundoExamenJuanFlores.demo.Model
 - Evento.java
 - com.SegundoExamenJuanFlores.demo.Repository
 - EventoRepository.java
 - com.SegundoExamenJuanFlores.demo.Service
 - EventoService.java
 - com.SegundoExamenJuanFlores.demo.ServiceImpl
 - EventoServiceImpl.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-21]
 - Maven Dependencies
 - target/generated-sources/annotations
 - target/generated-test-sources/test-annotations
 - src
 - target
 - HELP.md

EventoServiceImpl.java (Right Panel):

```
21 @Service
22 public class EventoServiceImpl implements EventoService {
23
24     @Autowired
25     private EventoRepository eventoRepository;
26
27     @Autowired
28     private EventoMapper eventoMapper;
29
30     @Override
31     public Collection<EventoResponse> findAllCliente() {
32         Collection<Evento> ListEventoResponses=eventoRepository.findAll();
33         return eventoMapper.toListEventoToClienteEvento(ListEventoResponses);
34     }
35
36     @Override
37     public EventoResponse findByIdEvento(Long idEvento) {
38         Evento evento=eventoRepository.findById(idEvento).orElse(null);
39         return eventoMapper.toEventoToEventoResponse(evento);
40     }
41
42     @Override
43     public void saveEvento(EventoRequest request) {
44         Evento newEvento = new Evento();
45         System.out.println("que llega" + request);
46         newEvento.setId(request.getId());
47         newEvento.setNombre(request.getNombre());
48         newEvento.setUbicacion(request.getUbicacion());
49         newEvento.setFechaInicio(request.getFechaInicio());
50         newEvento.setFechaFin(request.getFechaFin());
51         newEvento.setOrganizador(request.getOrganizador());
52         newEvento.setFechaCreacion(LocalDateTime.now());
53         eventoRepository.save(newEvento);
54     }
55 }
```

```
53 eventokepository.save(newEvento);
54 }
55
56 @Override
57 public void updateEvento(Long id, EventoRequest request) {
58     Evento evento = eventoRepository.findById(id).orElse(null);
59     if (evento != null) {
60         evento.setId(request.getId());
61         evento.setNombre(request.getNombre());
62         evento.setUbicacion(request.getUbicacion());
63         evento.setFechaInicio(request.getFechaInicio());
64         evento.setFechaFin(request.getFechaFin());
65         evento.setOrganizador(request.getOrganizador());
66         evento.setFechaCreacion(LocalDate.now());
67
68         eventoRepository.save(evento);
69     }
70 }
71
72 @Override
73 public void deleteEvento(Long idEvento) {
74     Evento evento = eventoRepository.findById(idEvento).orElse(null);
75     if (evento != null) {
76         eventoRepository.delete(evento);
77     }
78 }
79
80 }
81
82
83
84
85
```

prueba

```
demo [boot]
src/main/java
├── com.SegundoExamenJuanFlores.demo
│   ├── JuanFloresApplication.java
│   └── com.SegundoExamenJuanFlores.demo.Controller
│       ├── EventoController.java
│       ├── com.SegundoExamenJuanFlores.demo.DTO
│       │   ├── EventoRequest.java
│       │   ├── EventoResponse.java
│       │   └── com.SegundoExamenJuanFlores.demo.Mapper
│       │       ├── EventoMapper.java
│       │       ├── com.SegundoExamenJuanFlores.demo.Model
│       │       │   ├── Evento.java
│       │       ├── com.SegundoExamenJuanFlores.demo.Repository
│       │       │   ├── EventoRepository.java
│       │       ├── com.SegundoExamenJuanFlores.demo.Service
│       │       │   ├── EventoService.java
│       │       └── com.SegundoExamenJuanFlores.demo.ServiceImpl
│       │           ├── EventoServiceImpl.java
│       └── src/main/resources
│       ├── src/test/java
│       ├── JRE System Library [JavaSE-21]
│       ├── Maven Dependencies
│       ├── target/generated-sources/annotations
│       ├── target/generated-test-sources/test-annotations
│       ├── src
│       ├── target
│       ├── HELP.md
│       ├── mvnw
│       ├── mvnw.cmd
│       └── pom.xml
├── Juan_Carlos [boot]
├── JuanFlores [boot]
├── onetmany
└── > programacionv (in SistemasClases2025) [boot] [SistemasClases2025 feature]
```

```
23 @RestController
24 @RequestMapping("/api")
25 public class EventoController {
26
27     @Autowired
28     EventoService eventoService;
29
30     @GetMapping("/eventos/all")
31     public ResponseEntity<Collection<EventoResponse>> getEventos(){
32         return ResponseEntity.ok(eventoService.findAllCliente());
33     }
34
35     @GetMapping("/eventos/{id}")
36     public ResponseEntity<EventoResponse> getEventosById(@PathVariable Long id){
37         return ResponseEntity.ok(eventoService.findByIdEvento(id));
38     }
39
40     @PostMapping("/eventos/guardar")
41     public void saveEventosById(@RequestBody EventoRequest request){
42         eventoService.saveEvento(request);
43     }
44 }
```

Console × Debug Shell Boot Dashboard

demo - JuanFloresApplication [Spring Boot App] C:\Program Files\Java\jdk-21\bin\javaw.exe (24 may. 2025 11:05:55 elapsed: 0:06:18) [pid: 18496]

```
2025-05-24T11:06:00.464-05:00 INFO 18496 --- [JuanFlores] [main] org.hibernate.orm.connections.pooling : HHH10001005: Database
info:
    Database JDBC URL [Connecting through datasource 'HikariDataSource (HikariPool-1)']
    Database driver: undefined/unknown
    Database version: 8.0.40
    Autocommit mode: undefined/unknown
    Isolation level: undefined/unknown
    Minimum pool size: undefined/unknown
    Maximum pool size: undefined/unknown
2025-05-24T11:06:00.989-05:00 INFO 18496 --- [JuanFlores] [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA
platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2025-05-24T11:06:01.065-05:00 INFO 18496 --- [JuanFlores] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA
EntityManagerFactory for persistence unit 'default'
2025-05-24T11:06:01.232-05:00 WARN 18496 --- [JuanFlores] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-
view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view
```

postman eventos id

Collections

Environments

Flows

History

+

localhost:8080

POST /api/eventos/guardar

GET /api/eventos/1

GET /api/eventos/all

HTTP localhost:8080 / /api/eventos/guardar

GET

http://localhost:8080/api/eventos/7

Params

Authorization

Headers (7)

Body

Scripts

Settings

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK

156 ms

359 B

{ } JSON

Preview

Visualize

1 {

2 " id": null,

3 " nombre": "romian",

4 " ubicacion": "docteles",

5 " fechaInicio": "2025-06-15T10:00:00",

6 " fechaFin": "2025-06-20T18:00:00",

7 " organizador": "Municipalidad Provincial de Huamanga",

8 " fechaCreacion": null

9 }

Eventos save

Collections

Environments

Flows

History

localhost:8080

POST /api/eventos/guardar

GET /api/eventos/1

GET /api/eventos/all

localhost:8080 / api/eventos/guardar

GET http://localhost:8080/api/eventos/8

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "nombre": "dolores",
3   "ubicacion": "pichar",
4   "fechaInicio": "2025-06-15T10:00:00",
5   "fechaFin": "2025-06-20T18:00:00",
6   "organizador": "Municipalidad Provincial de Huamanga"
7 }
```

Body Cookies Headers (5) Test Results

200 OK • 234 ms • 358

{ } JSON Preview Visualize

```
1 {
2   "id": null,
3   "nombre": "dolores",
4   "ubicacion": "pichar",
5   "fechaInicio": "2025-06-15T10:00:00",
6   "fechaFin": "2025-06-20T18:00:00",
7   "organizador": "Municipalidad Provincial de Huamanga",
8   "fechaCreacion": null
9 }
```


Put mapping actualiza

lections

localhost:8080

PUT /api/eventos/guardar

GET /api/eventos/1

GET /api/eventos/all

ronments

Flows

history

PUT

http://localhost:8080/api/update/eventos/8

ParamsAuthorizationHeaders (9)BodyScriptsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1 {

2 "nombre": "Romina",

3 "ubicacion": "pichar",

4 "fechaInicio": "2025-06-15T10:00:00",

5 "fechaFin": "2025-06-20T18:00:00",

6 "organizador": "Municipalidad Provincial de Huamanga"

7 }

BodyCookiesHeaders (4)Test Results

200 OK • 18 ms • 123 B •

RawPreviewVisualize

1

delete

Collections

localhost:8080

PUT /api/eventos/guardar

GET /api/eventos/1

GET /api/eventos/all

Environments

Flows

History

DELETE

http://localhost:8080/api/delete/eventos/8

Send

Params

Authorization

Headers (7)

Body

Scripts

Settings

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

Body

Cookies

Headers (4)

Test Results

200 OK

84 ms

123 B

Save Response

Raw

Preview

Visualize

1

ironments

Flows

History

PUT /api/eventos/guardar

GET /api/eventos/1

GET /api/eventos/all

GET http://localhost:8080/api/eventos/8

Send

ParamsAuthorizationHeaders (7)BodyScriptsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

BodyCookiesHeaders (5)Test Results

200 OK • 44 ms • 281 B • Save Response

{ } JSONPreviewVisualize

id	null
nombre	null
ubicacion	null
fechaInicio	null
fechaFin	null
organizador	null
fechaCreacion	null

