

enunciado.pdf



SakiSap



Introducción a la Ingeniería del Software y los Sistemas de Información I



2º Grado en Ingeniería Informática - Tecnologías Informáticas



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Sevilla**

MÁSTER

**Inteligencia Artificial
& Data Management**

MADRID

Conquista el mundo de la IA
en 10 meses



Ahora
25%
DE DESCUENTO

Aprenderás:

- Datos a IA generativa
- Big Data, ML, LLMs
- MLOps + cloud
- Visión estratégica

EOI Escuela de
organización
industrial



Info y descuentos



FICHA, ALINEA, COMPITE

Y GANA

IISSI-1
Prueba de Laboratorio.

Curso 2023-24
Diciembre 2023

Instrucciones

- Descargue el proyecto del examen del siguiente repositorio: <https://github.eii.us.es/IISSI-2223/examen-lab-iissi.git>.
- Prepare el proyecto creando la BD ModeloExamen en HeidiSQL, revisando settings.py, y ejecutando "silence createdb", "silence createapi", "silence createtests" y finalmente "silence run".
- Implemente la solución a los ejercicios en los archivos **sql/solucion.sql** (para código sql) y **tests/solucion.http** (para peticiones http). Estos ejercicios están basados en unos requisitos adicionales que se suministran a continuación.
- Incluya capturas de pantalla HeidiSQL/vsCode que muestre el resultado de ejecutar el ejercicio al final de este documento.
- Debe subir a la actividad de EV este documento con las capturas de pantallas, y los archivos sql/solucion.sql y tests/solucion.http. Antes de subir la actividad, avise a un profesor para que valide la misma.
- Es necesario completar un nivel para que se valoren los siguientes niveles.

Catálogo de requisitos

RI-1-01 Requisito de información

Como: Profesor de la asignatura

Quiero: Tener disponible información sobre **Eventos**: Un evento es organizado por un usuario para que asista una cantidad máxima de personas. Sus atributos son: el usuario que lo organiza, el título del evento, la capacidad del evento, y la fecha en la que tendrá lugar. Todos los atributos son obligatorios.

Para: Evaluar al estudiante

RN-1-01 Primera regla de negocio

Como: Profesor de la asignatura

Quiero: Un mismo usuario no puede organizar varios eventos el mismo día.

Para: Evaluar al estudiante

RN-1-02 Segunda regla de negocio

Como: Profesor de la asignatura

Quiero: La capacidad debe ser superior o igual a 2 e inferior o igual a 1000.

Para: Evaluar al estudiante

RN-1-03 Tercera regla de negocio

Como: Profesor de la asignatura

Quiero: El año de la fecha debe ser posterior a 2000.

Para: Evaluar al estudiante



WUOLAH

IISSI-1 Prueba de Laboratorio.	Curso 2023-24 Diciembre 2023
-----------------------------------	---------------------------------

PA- 1-01 Inserción de nuevas entidades	
Como:	Profesor de la asignatura
Quiero:	<ul style="list-style-type: none"> ✓ Insertar Evento titulado "Evento 1" organizado por el usuario con ID=2, con capacidad para 10, que tiene lugar el 01/04/2021. ✓ Insertar Evento titulado "Evento 2" organizado por el usuario con ID=2, con capacidad para 10, que tiene lugar el 05/06/2018. ✓ Insertar Evento titulado "Evento 3" organizado por el usuario con ID=1, con capacidad para 100, que tiene lugar el 04/02/2020. ✗ Insertar Evento titulado "Evento 4" organizado por el usuario con ID=1, con capacidad para 70 que tiene lugar el 05/05/1990 (RN-1-03) ✗ Insertar Evento titulado "Evento 5" organizado por el usuario con ID=15, con capacidad para 200, que tiene lugar el 04/01/2020. (referencia no existente) ✗ Insertar Evento titulado "Evento 6" organizado por el usuario con ID=2, con capacidad para 200, que tiene lugar el 01/04/2021. (RN-1-01)
Para:	Evaluar al estudiante.

Nivel C. Máximo 7 puntos

- 1.(2)- Implemente los requisitos proporcionados (RI/RN) en MariaDB. No es necesario usar triggers.
- 2.(1)- Use Insert para implementar las pruebas de aceptación. **Incluya captura de pantalla con los datos de la tabla después de las pruebas.**

Use Delete para eliminar el elemento con ID=1. **Incluya captura de pantalla con los datos de la tabla después de la llamada.**

Use Update para actualizar los elementos con IDs 2 y 3 para que tengan una capacidad de 30. **Incluya captura de pantalla con los datos de la tabla después de la llamada.**
- 3.(0.5)- Escriba y realice peticiones HTTP para obtener el listado de géneros de la base de datos.
- 4.(1)- Cree una consulta que devuelva el precio medio de los videojuegos de plataforma con ID=2:

Videogames (1r x 1c)
averageVideogamePricePlatform2
35,33

- 5.(1.5)- Cree un disparador que, antes de actualizarse un Evento, si la capacidad está por encima de la media existente en la tabla, la establezca como 1000 en su lugar. Ejecute Update con un valor de 5000 para comprobar que funciona el disparador. Pruebe el disparador. **Incluya captura de pantalla con los datos de la tabla después de ejecutar la llamada.**
- 6.(0.5)- Cree una función que, a partir del identificador de un Evento, devuelva la cantidad de eventos de la tabla con una capacidad menor que la de dicho Evento. Use la función para obtener el resultado a partir del elemento con ID=2. Incluya capturas de pantallas.
- (0.5)- Cree un procedimiento que, a partir del identificador de un Evento, lo borre solo si el resultado de la función anterior es mayor a 1.

Nivel B. Máximo 9 puntos

7.(1)- Cree una consulta que devuelva el precio medio de los videojuegos de cada plataforma, ordenados por su precio medio de mayor a menor:

idPlatform	avgPrice
4	58,49
6	39,95
2	35,33
3	27,495
7	20,0
1	9,144286
8	5,0

8.(1)- Implemente las pruebas de aceptación como peticiones POST HTTP. **Recuerde limpiar los datos de la tabla insertados los apartados anteriores.** Las peticiones POST se pueden realizar con o sin autenticación. Incluya capturas de pantallas.

Nivel A. Máximo 10 puntos.

9.(0.5)- Cree una consulta que devuelva la descripción de los géneros con videojuegos valorados de media en más de 8, así como esta media:

description	avgScore
Shooter	9,0
Plataformas	9,0
Aventura	8,5

10.(0.5)- Cree un procedimiento **pAddTwoEvents(...)** que, dentro de una transacción, inserte dos Eventos a partir de los datos suministrados de los dos.

Realice dos llamadas: una que inserte dos correctamente, y una en la que el segundo rompa alguna restricción y aborte la transacción. Incluya capturas de pantallas



EL CICLO VITAL ES MARAVILLOSO... ¡LOS EXÁMENES SON SOLO UNA ETAPA MÁS!

Disney
EL REY LEÓN
TEATRO LOPE DE VEGA, MADRID

IISI-1
Prueba de Laboratorio.

Curso 2023-24
Diciembre 2023

Capturas de pantalla

Inserte aquí las capturas de pantalla COMPLETA que muestre la ejecución de los ejercicios. Para realizar una captura de pantalla puede usar la tecla "impr pant" del teclado y pulsar "ctrl+v" para pegar la captura de pantalla.

-Tras los 3 primeros Inserts, los demás dan error:

The screenshot shows the HeidiSQL interface with the 'modeloxamen' database selected. The 'eventos' table is highlighted, showing 3 rows. The SQL console displays the following commands and errors:

```
83 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 3 consultas: 0,093 seg. */
84 INSERT INTO eventos(iduser,title,capacity,date) VALUES (2, "Evento 1", 10, "2021-04-01");
85 /* Error de SQL (1054): Unknown column 'user' in 'field list' */
86 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 3 consultas: 0,000 seg. */
87 INSERT INTO eventos(iduser,title,capacity,eventDate) VALUES (2, "Evento 1", 10, "2021-04-01");
88 INSERT INTO eventos(iduser,title,capacity,eventDate) VALUES (2, "Evento 2", 10, "2018-06-05");
89 INSERT INTO eventos(iduser,title,capacity,eventDate) VALUES (1, "Evento 3", 100, "2020-02-04");
90 /* Filas afectadas: 3 Filas encontradas: 0 Advertencias: 0 Duración para 3 consultas: 0,016 seg. */
91 SHOW CREATE TABLE `modeloxamen`.`eventos`;
92 SELECT * FROM `modeloxamen`.`eventos` LIMIT 1000;
```

-Tras el primer DELETE:

The screenshot shows the HeidiSQL interface after the first DELETE operation. The 'eventos' table now contains 2 rows. The SQL console displays the following commands and errors:

```
98 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 1 consulta: 0,000 seg. */
99 INSERT INTO eventos(iduser,title,capacity,eventDate) VALUES (2, "Evento 1", 10, "2021-04-01");
100 /* Error de SQL (1062): Duplicate entry '2-2021-04-01' for key 'PRIMARY' */
101 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 1 consulta: 0,000 seg. */
102 DELETE FROM eventos WHERE idEvent = 1;
103 /* Error de SQL (1064): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '' FROM eventos WHERE idE
104 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 1 consulta: 0,000 seg. */
105 DELETE FROM eventos WHERE idEvent = 1;
106 /* Filas afectadas: 1 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
107 SELECT * FROM `modeloxamen`.`eventos` LIMIT 1000;
```

WUOLAH

-Tras el UPDATE:

The screenshot shows the HeidiSQL interface with the 'modeloexamen' database selected. The 'eventos' table is displayed with the following data:

idEvent	idUser	title	capacity	eventDate
2	2	Evento 2	30	2018-06-05
3	1	Evento 3	30	2020-02-04

The SQL editor shows the following commands:

```

127 /* Entrando a la sesión "iissi_user" */
128 SELECT * FROM `modeloexamen`.`eventos` LIMIT 1000;
129 UPDATE eventos SET capacity = 30 WHERE idEvent = 2 AND idEvent = 3;
130 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
131 SELECT * FROM `modeloexamen`.`eventos` LIMIT 1000;
132 UPDATE eventos SET capacity = 30 WHERE idEvent = 2 AND idEvent = 3;
133 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
134 UPDATE eventos SET capacity = 30 WHERE idEvent = 2 OR idEvent = 3;
135 /* Filas afectadas: 2 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
136 SELECT * FROM `modeloexamen`.`eventos` LIMIT 1000;
  
```

-Tras el trigger del ejercicio 5:

The screenshot shows the HeidiSQL interface with the 'modeloexamen' database selected. The 'eventos' table is displayed with the following data:

idEvent	idUser	title	capacity	eventDate
2	2	Evento 2	1.000	2018-06-05
3	1	Evento 3	1.000	2020-02-04

The SQL editor shows the following commands:

```

196 SELECT *, EVENT_SCHEMA AS `db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`="modeloexamen";
197 UPDATE eventos SET capacity = 5000;
198 /* Filas afectadas: 2 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,016 seg. */
199 SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA="modeloexamen" AND TABLE_NAME="eventos" ORDER BY ORDINAL_POSITION;
200 SHOW INDEXES FROM `eventos` FROM `modeloexamen`;
201 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA="modeloexamen" AND TABLE_NAME="eventos" AND REFERENCED_TABLE_NAME IS NOT NULL;
202 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA="modeloexamen" AND TABLE_NAME="eventos" AND REFERENCED_TABLE_NAME IS NOT NULL;
203 SHOW CREATE TABLE `modeloexamen`.`eventos`;
204 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA="modeloexamen" AND TABLE_NAME="eventos";
205 SELECT * FROM `modeloexamen`.`eventos` LIMIT 1000;
  
```

PRODUCIDO POR

Stage
ENTERTAINMENT
SPAIN

**SIMBA Y NALA.
TIMÓN Y PUMBAA.**

¡TUS AMIGOS Y TÚ!

El combo perfecto para un
planazo de 10 • **Desde 25€***

© Disney

Disney
**EL
REY
LEÓN**

TEATRO LOPE DE VEGA,
MADRID

**¡CONSIGUE YA
TUS ENTRADAS!**

*Iva incluido. Gastos de gestión no incluidos.



-Antes del procedimiento del ejercicio 6:

The screenshot shows the HeidiSQL interface connected to a MySQL database named 'modeloxamen'. The left sidebar displays the database structure, including tables like 'eventos', 'genres', 'platforms', 'users', and 'videogames'. The main window shows the 'eventos' table with the following data:

idEvent	idUser	title	capacity	eventDate
3	1	Evento 3	100	2020-02-04
4	2	Evento 1	10	2021-04-01
5	2	Evento 2	1.000	2018-06-05

The bottom pane shows SQL queries for inserting and updating data in the 'eventos' table.

-Despues del procedimiento del ejercicio 6 se borra el evento con id = 5.

-Capturas de peticiones POST HTTP (cada captura es una petición) :

The screenshot shows the VS Code interface with the 'BD Examen' project. The 'solution.http' file contains the following HTTP requests:

```

1  ## Referencia a la base de datos
2  @BASE = http://127.0.0.1:8888/api/v1
3
4  ## Petición para obtener generos
5  GET {{BASE}}/genres
6
7  ## Pruebas de aceptación
8  POST {{BASE}}/eventos
9  Content-Type: application/json
10
11  {
12    "idUser": 2,
13    "title": "Evento 1",
14    "capacity": 10,
15    "eventDate": "2021-04-01"
16  }
17
18  ##
19  POST {{BASE}}/eventos
20  Content-Type: application/json
21
22  {
23    "idUser": 2,
24    "title": "Evento 2",
25    "capacity": 10,
26    "eventDate": "2018-06-05"
27  }
28
29  ##

```

The 'Response(34ms)' pane shows the response for the first POST request:

```

1  HTTP/1.0 200 OK
2  Content-Type: application/json
3  Content-Length: 13
4  Access-Control-Allow-Origin: *
5  Server: Werkzeug/2.0.3 Python/3.9.7
6  Date: Wed, 20 Dec 2023 11:23:31 GMT
7
8  {
9    "lastid": 1
10 }

```




LOS GRANDES REYES TE MIRAN DESDE
LAS ESTRELLAS... ¡Y TE DARÁN SUERTE!

Disney
EL REY LEÓN
TEATRO LOPE DE VEGA, MADRID

IISII-1
Prueba de Laboratorio.

Curso 2023-24
Diciembre 2023

The screenshot shows a web application interface with a REST client. The client is configured to send a POST request to the endpoint `/api/v1/eventos` with a JSON body. The response is a 200 OK status with a JSON body containing the event details.

```
ModeloExamen > tests > E solucion.http > POST /api/v1/eventos
1  ## Referencia a la base de datos
2  @BASE = http://127.0.0.1:8080/api/v1
3
4  ## Petición para obtener generos
5  GET {{BASE}}/genres
6
7  ## Pruebas de aceptación
8  POST {{BASE}}/eventos
9  Content-type: application/json
10
11  {
12    "idUser": 2,
13    "title": "Evento 1",
14    "capacity": 10,
15    "eventDate": "2021-04-01"
16  }
17
18  ##
19  Send Request
20  POST {{BASE}}/eventos
21  Content-type: application/json
22
23  {
24    "idUser": 2,
25    "title": "Evento 2",
26    "capacity": 10,
27    "eventDate": "2018-06-05"
28  }
29  ##
30  Send Request
```

Response(35ms)

```
1  HTTP/1.0 200 OK
2  Content-Type: application/json
3  Content-Length: 13
4  Access-Control-Allow-Origin: *
5  Server: Werkzeug/2.0.3 Python/3.9.7
6  Date: Wed, 20 Dec 2023 11:24:44 GMT
7
8  {
9    "lastId": 2
10 }
```

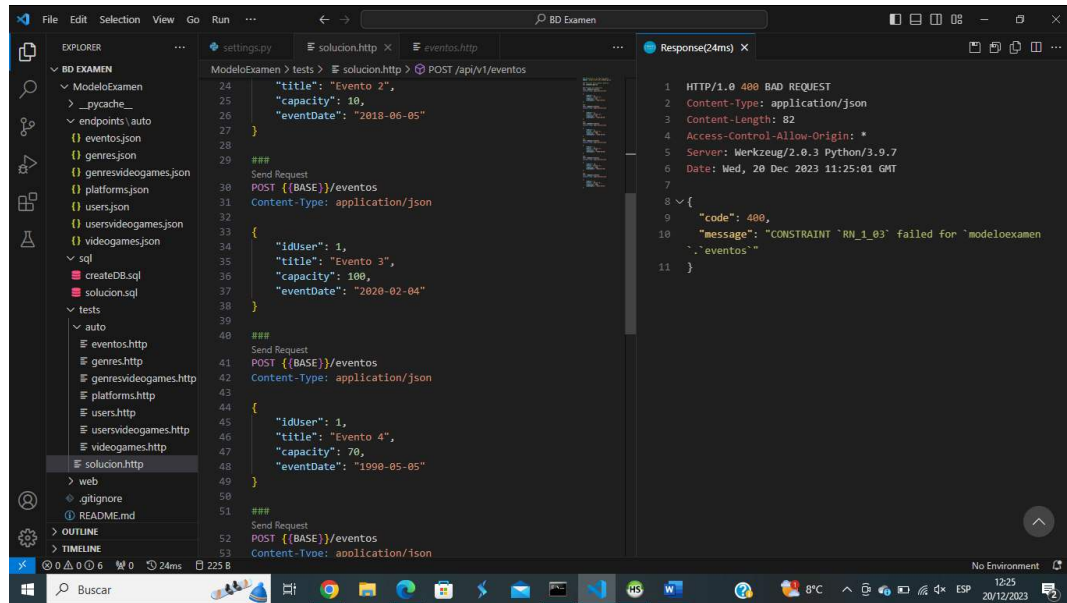
The screenshot shows a web application interface with a REST client. The client is configured to send a POST request to the endpoint `/api/v1/eventos` with a JSON body. The response is a 200 OK status with a JSON body containing the event details.

```
ModeloExamen > tests > E solucion.http > POST /api/v1/eventos
12  {
13    "idUser": 2,
14    "title": "Evento 1",
15    "capacity": 10,
16    "eventDate": "2021-04-01"
17  }
18  ##
19  Send Request
20  POST {{BASE}}/eventos
21  Content-type: application/json
22
23  {
24    "idUser": 2,
25    "title": "Evento 2",
26    "capacity": 10,
27    "eventDate": "2018-06-05"
28  }
29  ##
30  Send Request
31  POST {{BASE}}/eventos
32  Content-type: application/json
33
34  {
35    "idUser": 1,
36    "title": "Evento 3",
37    "capacity": 100,
38    "eventDate": "2020-02-04"
39  }
40  ##
41  Send Request
```

Response(33ms)

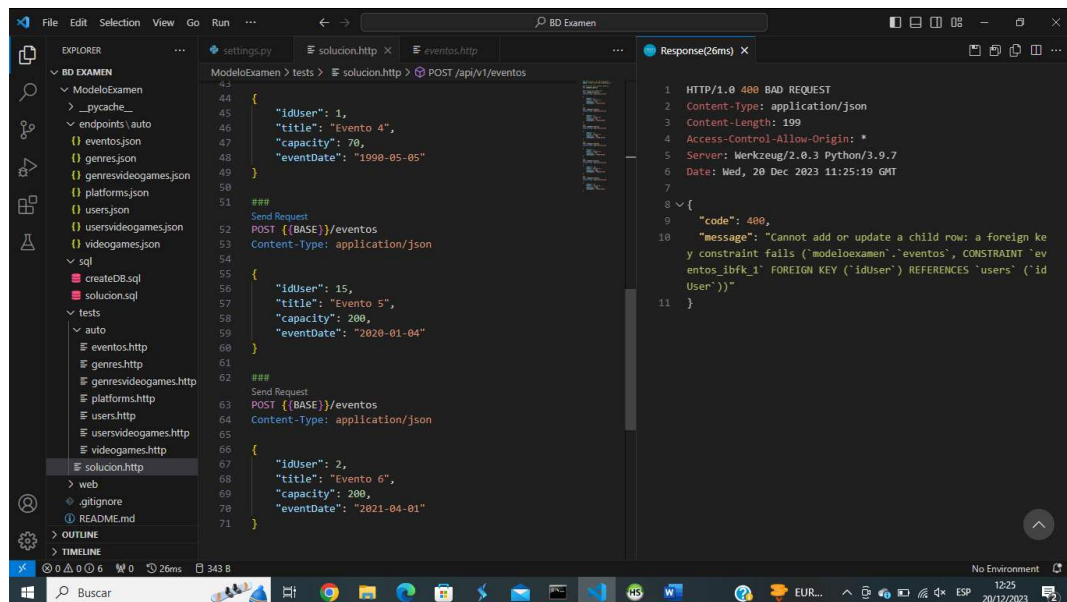
```
1  HTTP/1.0 200 OK
2  Content-Type: application/json
3  Content-Length: 13
4  Access-Control-Allow-Origin: *
5  Server: Werkzeug/2.0.3 Python/3.9.7
6  Date: Wed, 20 Dec 2023 11:24:53 GMT
7
8  {
9    "lastId": 3
10 }
```

WUOLAH



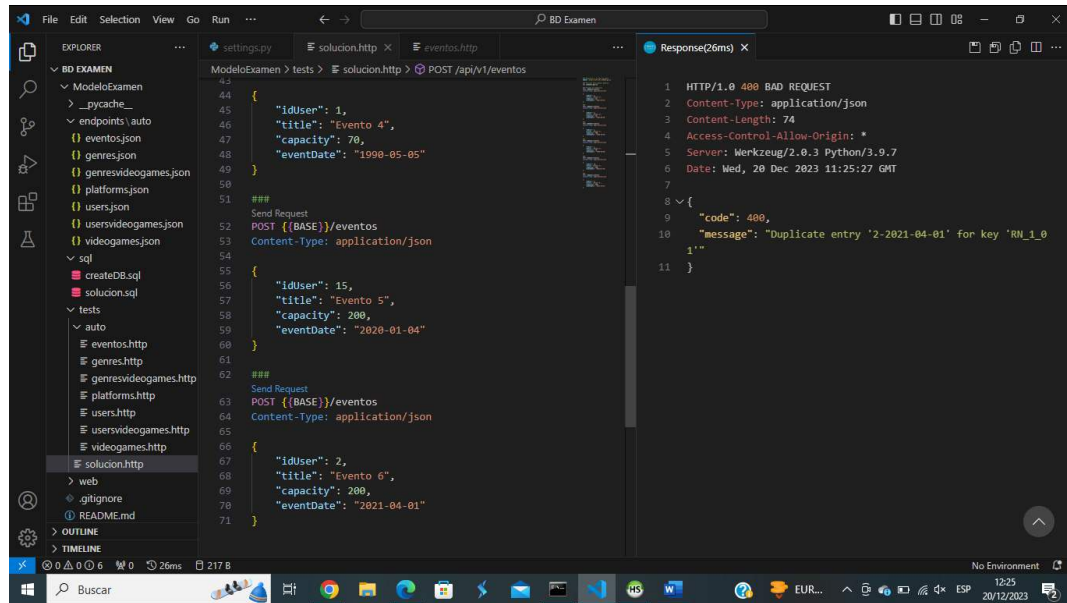
```
24 {
25   "title": "Evento 2",
26   "capacity": 10,
27   "eventDate": "2018-06-05"
28 }
29
30 ###
31 Send Request
32 POST {{BASE}}/eventos
33 Content-Type: application/json
34 {
35   "idUser": 1,
36   "title": "Evento 3",
37   "capacity": 100,
38   "eventDate": "2020-02-04"
39 }
40
41 ###
42 Send Request
43 POST {{BASE}}/eventos
44 Content-Type: application/json
45 {
46   "idUser": 1,
47   "title": "Evento 4",
48   "capacity": 70,
49   "eventDate": "1990-05-05"
50 }
51
52 ###
53 Send Request
54 POST {{BASE}}/eventos
55 Content-Type: application/json
```

```
1 HTTP/1.0 400 BAD REQUEST
2 Content-Type: application/json
3 Content-Length: 82
4 Access-Control-Allow-Origin: *
5 Server: Werkzeug/2.0.3 Python/3.9.7
6 Date: Wed, 20 Dec 2023 11:25:01 GMT
7
8 {
9   "code": 400,
10  "message": "CONSTRAINT 'RN_1_03' failed for 'modeloexamen'
11  'eventos'"
12 }
```



```
44 {
45   "idUser": 1,
46   "title": "Evento 4",
47   "capacity": 70,
48   "eventDate": "1990-05-05"
49 }
50
51 ###
52 Send Request
53 POST {{BASE}}/eventos
54 Content-Type: application/json
55 {
56   "idUser": 15,
57   "title": "Evento 5",
58   "capacity": 200,
59   "eventDate": "2020-01-04"
60 }
61
62 ###
63 Send Request
64 POST {{BASE}}/eventos
65 Content-Type: application/json
66 {
67   "idUser": 2,
68   "title": "Evento 6",
69   "capacity": 200,
70   "eventDate": "2021-04-01"
71 }
```

```
1 HTTP/1.0 400 BAD REQUEST
2 Content-Type: application/json
3 Content-Length: 199
4 Access-Control-Allow-Origin: *
5 Server: Werkzeug/2.0.3 Python/3.9.7
6 Date: Wed, 20 Dec 2023 11:25:19 GMT
7
8 {
9   "code": 400,
10  "message": "Cannot add or update a child row: a foreign key
11  constraint fails ('modeloexamen','eventos', CONSTRAINT 'ev
12  entos_ibfk_1' FOREIGN KEY ('idUser') REFERENCES 'users' ('id
13  User'))"
14 }
```



The screenshot shows a Visual Studio Code editor with a REST client configuration. The Explorer sidebar on the left shows a project named 'BD EXAMEN' with various files including JSON data, SQL scripts, and HTTP request files. The main editor area displays a REST client configuration for a POST request to 'POST /api/v1/eventos'. The request body is a JSON object with fields: 'idUser': 1, 'title': 'Evento 4', 'capacity': 70, and 'eventDate': '1990-05-05'. The response pane on the right shows an 'HTTP/1.0 400 BAD REQUEST' status with a JSON body: {'code': 400, 'message': 'Duplicate entry '2-2021-04-01' for key 'RN_1_0 1''}. The status bar at the bottom indicates 'No Environment' and the date '20/12/2023'.

```
44 {
45   "idUser": 1,
46   "title": "Evento 4",
47   "capacity": 70,
48   "eventDate": "1990-05-05"
49 }
50
51 ###
52 Send Request
53 POST {{BASE}}/eventos
54 Content-Type: application/json
55
56 {
57   "idUser": 15,
58   "title": "Evento 5",
59   "capacity": 200,
60   "eventDate": "2020-01-04"
61 }
62
63 ###
64 Send Request
65 POST {{BASE}}/eventos
66 Content-Type: application/json
67
68 {
69   "idUser": 2,
70   "title": "Evento 6",
71   "capacity": 200,
72   "eventDate": "2021-04-01"
73 }
```

```
1 HTTP/1.0 400 BAD REQUEST
2 Content-Type: application/json
3 Content-Length: 74
4 Access-Control-Allow-Origin: *
5 Server: Werkzeug/2.0.3 Python/3.9.7
6 Date: Wed, 20 Dec 2023 11:25:27 GMT
7
8 {
9   "code": 400,
10  "message": "Duplicate entry '2-2021-04-01' for key 'RN_1_0 1'"
11 }
```