Grupo: Jonathan Ruiz Velázquez, Alan Cajigas Arroyo, Cristian Maldonado Filomeno

16 de mayo de 2024

COMP 3400

Prof. Javier Dastas

Evidencia de Documentación, base de datos, Tablas, conexiones.

**CODIGO FRONT-END**  (Codificación hecha por Cristian Maldonado)

# Index.html

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title> Datos</title>

<!-- Bootstrap CSS -->

<link

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"

rel="stylesheet"

/>

<!-- DataTables CSS -->

<link

href="https://cdn.datatables.net/1.10.24/css/jquery.dataTables.min.css" rel="stylesheet"

/>

</head>

<body>

<div class="container mt-5">

<h1>Tabla de Datos</h1>

<table id="tablaDatos" class="table">

<thead>

<tr>

<th>Nombre</th>

<th>Telefono</th>

<th>Email</th>

<th>Direccion</th>

<th>Fecha de Evento</th>

<th>Fecha de Registro</th>

<th>Lugar de Evento</th>

<th>Tipo de Evento</th>

<th>Nombre de Evento</th>

</tr>

</thead>

<tbody>

<!-- Aquí se llenará la tabla con datos -->

</tbody>

</table>

</div>

<!-- jQuery y Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>

<!-- DataTables JS -->

<script src="https://cdn.datatables.net/1.10.24/js/jquery.dataTables.min.js"></script>

<!-- Script personalizado -->

<script src="./js/script.js"></script>

</body>

</html>

# Event.list

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Lista de Eventos</title>

<!-- Bootstrap CSS -->

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet" />

<!-- DataTables CSS -->

<link href="https://cdn.datatables.net/1.10.24/css/jquery.dataTables.min.css" rel="stylesheet" /> </head>

<body>

<div class="container mt-5">

<h1>Lista de Eventos</h1>

<table id="tablaeventos" class="table">

<thead>

<tr>

<th>ID Evento</th>

<th>Fecha Registro</th>

<th>Fecha Evento</th>

<th>Lugar</th>

<th>Tipo</th>

<th>Descripcion</th>

</tr>

</thead>

<tbody>

<!-- Aquí se llenará la tabla con datos -->

</tbody>

</table>

</div>

<!-- jQuery y Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script> <!-- DataTables JS -->

<script src="https://cdn.datatables.net/1.10.24/js/jquery.dataTables.min.js"></script>

<!-- Script personalizado -->

<script src="./js/script.js"></script>

</body>

</html>

# User-list

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8" />

<title>Lista de Usuarios</title>

<link href="https://cdn.datatables.net/1.11.3/css/jquery.dataTables.min.css" rel="stylesheet"

/>

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet" />

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrapicons@1.11.3/font/bootstrapicons.min.css"> <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdn.datatables.net/1.11.3/js/jquery.dataTables.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script> </head>

<body>

<div class="container mt-5">

<h2>Usuarios Registrados</h2>

<table id="tablaUsuarios" class="display" style="width: 100%">

<thead>

<tr>

<th>ID Usuarios</th>

<th>Nombre</th>

<th>Telefono</th>

<th>Email</th>

<th>Direccion Fisica</th>

<th class="text-center">

<button type="button" id="btn-add-new" class="btn btn-primary btn-sm">

Crear

</button>

</th>

</tr>

</thead>

</table>

</div>

<!-- Modal para Añadir, Editar y Borrar -->

<div class="modal" id="actionModal">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="modalTitle">Acción</h5>

<button type="button" class="close" data-dismiss="modal">

&times;

</button>

</div>

<div class="modal-body">

<form>

<form id="miFormulario">

<input type="hidden" name="id\_usuario" id="id\_usuario">

<div class="mb-2">

<label for="nombre" class="form-label">Nombre</label>

<input type="text" class="form-control" name="nombre" id="nombre" placeholder="Nombre" />

</div>

<div class="mb-2">

<label for="apellido" class="form-label">Telefono</label>

<input type="text" class="form-control" name="telefono" id="telefono" placeholder="Telefono" />

</div>

<div class="mb-2">

<label for="email" class="form-label">Email</label>

<input type="email" class="form-control" name="email" id="email" placeholder="Email"/>

</div>

<div class="mb-2">

<label for="direccion fisica" class="form-label">Direccion Fisica</label>

<input type="direccion fisica" class="form-control" name="direccion fisica" id="direccion fisica" placeholder="Direccion Fisica" />

</div>

</form>

</form>

<div id="modalMessages"></div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">

Cerrar

</button>

<button type="button" id="btn-modal-crud-action" class="btn" data-dismiss="modal">

Acción

</button>

</div>

</div>

</div>

</div>

<script src="./js/users.js"></script>

<script src="./js/functions-tools.js"></script>

</body>

</html>

# Script.js

$(document).ready(function () {

//Configuración de DataTables y llamada al API

$("#tablaDatos").DataTable({

ajax: {

url: "http://linux-vm14:5000/api/get\_inscripciones",

},

error: function (xhr, status, error) { console.log("Error al obtener los datos del API:", error);

}, columns: [

{ data: "nombre" },

{ data: "telefono" },

{ data: "email" },

{ data: "direccion\_fisica" },

{ data: "fecha\_registro" },

{ data: "fecha\_evento"},

{ data:"lugar"},

{ data:"tipo"},

{ data:"descripcion"}

],

});

$("#tablaeventos").DataTable({

ajax: {

url: "http://linux-vm14:5000/api/get\_eventos",

},

error: function (xhr, status, error) { console.log("Error al obtener los datos

del API:", error);

},

columns: [

{ data: "id\_evento" },

{ data: "fecha\_registro" },

{ data: "fecha\_evento" },

{ data: "lugar" },

{ data: "tipo"},

{ data:"descripcion"},

],

});

});

# User.js

$(document).ready(function () { let crudAction

= "edit";

var table = $("#tablaUsuarios").DataTable({

ajax: {

url: "http://linux-vm14:5000/api/get\_users", // Cambia esto por la URL de tu API

},

columns: [

{ data: "id\_usuarios" },

{ data: "nombre" },

{ data: "telefono" },

{ data: "email" },

{ data: "direccion\_fisica" },

{

data: null, defaultContent:

"<div class='text-center'>" +

"<button type='button' class='btn btn-info btn-sm btnEdit'><i class='bi bipencil'></i></button>&nbsp;" +

"<button type='button' class='btn btn-danger btn-sm btnDelete'><i class='bi bitrash3'></i></button>" +

"</div>",

},

],

});

$("#btn-add-new").click(function () { crudAction

= "new";

$("form").get(0).reset();

$("#nombre").prop("readonly", false);

$("#telefono").prop("readonly", false);

$("#email").prop("readonly", false);

$("#direccion fisica").prop("readonly", false);

updateModal("Añadir Usuario");

});

$("#tablaUsuarios tbody").on("click", ".btnEdit", function () { var data = table.row($(this).parents("tr")).data(); // Get ROW DATA in TABLE setDataElementsReadOnly(data, false);

fillForm(data);

crudAction = "edit";

// $("#modalTitle").text("Editar Usuario");

// $("#modalBody").text(

// `¿Deseas editar al usuario ${data.nombre} ${data.apellido}?`

// );

updateModal("Editar Usuario");

});

$("#tablaUsuarios tbody").on("click", ".btnDelete", function () { var data = table.row($(this).parents("tr")).data(); // Get ROW DATA in TABLE setDataElementsReadOnly(data, true); fillForm(data); crudAction

= "delete";

$("#modalTitle").text("Eliminar Usuario");

$("#modalMessages").html(

'<div class="alert alert-danger" role="alert">' +

`¿Estás segur@ que quieres eliminar al usuario <strong>${data.nombre} ${data.apellido}</strong>?` +

"</div>"

);

$("#btn-modal-crud-action").addClass("btn-danger");

$("#btn-modal-crud-action").removeClass("btn-primary");

$("#btn-modal-crud-action").text("Borrar");

$("#actionModal").modal("show");

});

$("#btn-modal-crud-action").click(function () {

// alert(crudAction); if

(crudAction === "new") {

// Execute REST API to INSERT

} else if (crudAction === "edit") {

// Execute REST API to EDIT

} else {

// if crudAction === "delete";

// Execute REST API to DELETE

}

$("#actionModal").modal("hide");

});

function updateModal(title) {

$("#modalTitle").text(title);

$("#btn-modal-crud-action").addClass("btn-primary");

$("#btn-modal-crud-action").removeClass("btn-danger");

$("#btn-modal-crud-action").text("Guardar");

$("#actionModal").modal("show");

}

});

Function.js

// formData is a JSON Object with all values

// from de REST API function fillForm(formData) {

$.each(formData, function (key, valor) { var elemento =

$('[name="' + key + '"]'); if (elemento.length > 0) { if (elemento.is(":checkbox") || elemento.is(":radio")) { elemento.prop("checked", elemento.val() === valor);

} else if (elemento.is("select")) { elemento.val(valor);

} else {

elemento.val(valor);

}

}

});

}

function setDataElementsReadOnly(formData, isReadOnly) { $.each(formData, function

(key) { var elemento = $('[name="'

+ key + '"]'); elemento.prop("readonly", isReadOnly ?

"readonly" : "");

});

}

**PROGRAMACION BACK-END** (Codificación hecha por Jonathan Ruiz)

from flask import Flask, jsonify, request import mariadb import

sys

# Importa la configuración de acceso a la base de datos from config import plataforma\_de\_gestion\_de\_eventos app

= Flask(\_\_name\_\_)

try:

conn = mariadb.connect(\*\*plataforma\_de\_gestion\_de\_eventos) except mariadb.Error as e:

print(f"Error con conexion: {e}") sys.exit(1)

cursor = conn.cursor()

# Ruta de prueba

@app.route('/api/hello', methods=['GET']) def hello\_world(): return jsonify({'message': '¡Hola, mundo con Flask!'})

# Ruta para obtener información sobre el usuario

@app.route('/api/misdatos/', methods=['GET']) def mis\_datos():

# Aquí puedes devolver información específica sobre el usuario, como su nombre, teléfono, etc. return jsonify({'datos': 'Juan Perez, 879901010, juanperez@gmail.com, Calle 123'})

# Ruta para obtener todos los usuarios Y eventos

@app.route('/api/get\_users', methods=['GET']) def get\_users():

cursor.execute("SELECT \* FROM Usuarios") users = cursor.fetchall() user\_list

= [] for user in users: user\_list.append({

"id\_usuarios": user[0],

"nombre": user[1],

"telefono": user[2],

"email": user[3],

"direccion\_fisica": user[4]

})

response = jsonify({"data": user\_list}) response.headers.add("Content-type", 'application/json') response.headers.add('Access-Control-Allow-Origin', '\*') return response

# Ruta para obtener todos los eventos

@app.route('/api/get\_eventos', methods=['GET']) def get\_eventos():

cursor.execute("SELECT \* FROM Eventos") eventos = cursor.fetchall() eventos\_list = [] for evento in eventos: eventos\_list.append({ "id\_evento": evento[0],

"fecha\_registro": evento[1],

"fecha\_evento": evento[2],

"lugar": evento[3],

"tipo": evento[4],

"descripcion": evento[5]

})

response = jsonify({"data": eventos\_list}) response.headers.add("Contenttype", 'application/json') response.headers.add('Access-Control-Allow-Origin', '\*') return response

# Ruta para obtener toda la data de usuarios y eventos

@app.route('/api/get\_inscripciones', methods=['GET']) def get\_inscripciones():

cursor.execute("SELECT \* FROM v\_all\_tables") inscripciones

= cursor.fetchall() inscripciones\_list

= [] for inscripcione in inscripciones: inscripciones\_list.append({

"id\_usuarios": inscripcione[0],

"nombre": inscripcione[1],

"telefono": inscripcione[2],

"email": inscripcione[3],

"direccion\_fisica": inscripcione[4],

"id\_evento": inscripcione[5],

"fecha\_registro": inscripcione[6],

"fecha\_evento": inscripcione[7],

"lugar": inscripcione[8],

"tipo": inscripcione[9],

"descripcion": inscripcione[10]

})

response = jsonify({"data": inscripciones\_list}) response.headers.add("Contenttype", 'application/json') response.headers.add('Access-Control-Allow-Origin', '\*') return response

# Ruta para obtener un usuario por su ID

@app.route('/api/get\_user/<int:id>', methods=['GET']) def get\_user(id):

if id == 0:

return jsonify({"error": "ID inválido"}), 404 else:

cursor.execute("SELECT \* FROM usuarios WHERE ID\_registro = ?", (id,)) user = cursor.fetchone() if user: user\_data = {

"ID\_registro": user[0],

"nombre": user[1],

"telefono": user[2],

"correo\_electronico": user[3],

"direccion\_fisica": user[4],

"ID\_evento": user[5],

"tipo\_de\_evento": user[6],

"descripcion\_de\_evento": user[7],

"lugar\_de\_evento": user[8],

"fecha\_de\_evento": user[9],

"fecha\_de\_registro": user[10] } return jsonify(user\_data) else:

return jsonify({"message": "Usuario no encontrado"}), 404

# Ruta para obtener un evento por su ID

@app.route('/api/get\_event/<int:id>', methods=['GET']) def get\_event(id):

if id == 0:

return jsonify({"error": "ID inválido"}), 404 else:

cursor.execute("SELECT \* FROM Eventos WHERE ID\_evento = ?", (id,)) evento = cursor.fetchone() if evento:

evento\_data = {

"ID\_evento": evento[0],

"fecha\_registro": evento[1],

"fecha\_evento": evento[2],

"lugar": evento[3],

"tipo": evento[4],

"descripcion": evento[5]

} return jsonify(evento\_data)

else:

return jsonify({"message": "Evento no encontrado"}), 404

# Ruta para agregar un nuevo usuario y evento

@app.route('/api/new\_user', methods=['POST']) def new\_user(): datos

= request.json nombre = datos.get('nombre') telefono

= datos.get('telefono') correo\_electronico = datos.get('correo\_electronico') direccion\_fisica = datos.get('direccion\_fisica')

ID\_evento = datos.get('ID\_evento') tipo\_de\_evento =

datos.get('tipo\_de\_evento') descripcion\_de\_evento = datos.get('descripcion\_de\_evento') lugar\_de\_evento = datos.get('lugar\_de\_evento') fecha\_de\_evento = datos.get('fecha\_de\_evento') fecha\_de\_registro = datos.get('fecha\_de\_registro')

strQry = 'INSERT INTO usuarios '

strQry += "(nombre, telefono, correo\_electronico, direccion\_fisica, ID\_evento, tipo\_de\_evento, descripcion\_de\_evento, lugar\_de\_evento, fecha\_de\_evento, fecha\_de\_registro) "

strQry += f"VALUES ('{nombre}', '{telefono}', '{correo\_electronico}',

'{direccion\_fisica}', '{ID\_evento}', '{tipo\_de\_evento}', '{descripcion\_de\_evento}',

'{lugar\_de\_evento}', '{fecha\_de\_evento}', '{fecha\_de\_registro}')"

cursor.execute(strQry) conn.commit() response = {"message": "Registro insertado

exitosamente"}

return jsonify(response), 200

# Ruta para actualizar usuario

@app.route('/api/update\_user', methods=['PUT']) def update\_user(): datos = request.json ID\_registro = datos.get('ID\_registro') nombre = datos.get('nombre') telefono

= datos.get('telefono') correo\_electronico = datos.get('correo\_electronico') direccion\_fisica = datos.get('direccion\_fisica')

strQry = 'UPDATE usuarios ' strQry += f"SET nombre =

'{nombre}', " strQry += f"telefono = '{telefono}', " strQry

+= f"correo\_electronico = '{correo\_electronico}', " strQry

+= f"direccion\_fisica = '{direccion\_fisica}' " strQry

+= f"WHERE ID\_registro = {ID\_registro}" cursor.execute(strQry) conn.commit() response = {"message":

"Registro de usuario actualizado exitosamente"}

return jsonify(response), 200

# Ruta para actualizar evento

@app.route('/api/update\_event', methods=['PUT']) def update\_event(): datos = request.json

ID\_evento = datos.get('ID\_evento') fecha\_evento

= datos.get('fecha\_evento') lugar = datos.get('lugar') tipo =

datos.get('tipo') descripcion = datos.get('descripcion') fecha\_registro

= datos.get('fecha\_registro')

strQry = 'UPDATE Eventos ' strQry += f"SET fecha\_evento = '{fecha\_evento}', " strQry += f"lugar

= '{lugar}', " strQry += f"tipo = '{tipo}', " strQry

+= f"descripcion = '{descripcion}', " strQry

+= f"fecha\_registro = '{fecha\_registro}' " strQry += f"WHERE

ID\_evento = {ID\_evento}"

cursor.execute(strQry) conn.commit() response = {"message": "Registro

de evento actualizado exitosamente"} return

jsonify(response), 200

# Ruta para eliminar un usuario por su ID @app.route('/api/del\_user/<int:id>', methods=['DELETE']) def

del\_user(id): if id == 0:

return jsonify({"error": "ID inválido"}), 404 else:

cursor.execute("DELETE FROM usuarios WHERE ID\_registro = ?", (id,)) conn.commit() return jsonify({"message": "Usuario eliminado correctamente"}), 200

# Ruta para eliminar un evento por su ID

@app.route('/api/del\_event/<int:id>', methods=['DELETE']) def del\_event(id):

if id == 0:

return jsonify({"error": "ID inválido"}), 404 else:

cursor.execute("DELETE FROM Eventos WHERE ID\_evento = ?", (id,)) conn.commit() return jsonify({"message": "Evento eliminado correctamente"}),

200

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

**CODIGO DE BASE DE DATOS** (Hecho por Alan Cajigas)

**CREATE DATABASE plataforma\_de\_gestion\_de\_eventos;**

**USE plataforma\_de\_gestion\_de\_eventos;**

**CREATE TABLE Usuarios (**

**id\_usuarios INT PRIMARY KEY AUTO\_INCREMENT,**

**nombre VARCHAR(255),**

**telefono INT,**

**email VARCHAR(255),**

**direccion\_fisica VARCHAR(255)**

**);**

**CREATE TABLE Eventos (**

**id\_evento INT PRIMARY KEY AUTO\_INCREMENT,**

**fecha\_registro DATE,**

**fecha\_evento DATE,**

**lugar VARCHAR(255),**

**tipo VARCHAR(255),**

**descripcion TEXT**

**);**

**CREATE TABLE Inscripciones (**

**id\_usuarios INT,**

**id\_evento INT,**

**PRIMARY KEY (id\_usuarios, id\_evento)**

**);**

**create or replace view v\_all\_tables as**

**(select u.id\_usuarios, nombre, telefono, email, direccion\_fisica,**

**e.id\_evento, fecha\_registro, fecha\_evento, lugar, tipo, descripcion**

**from Usuarios u join Inscripciones i on u.id\_usuarios = i.id\_usuarios**

**JOIN Eventos e on e.id\_evento = i.id\_evento);**

**INSERT INTO Usuarios (nombre, telefono, email, direccion\_fisica)**

**VALUES**

**('Juan Perez', '879901010', 'juanperez@gmail.com', 'Calle 123'),**

**('Maria Gomez', '874578569', 'mariagomez@gmail.com', 'Av. Principal'),**

**('Carlos Sanchez', '875472893', 'carlossanchez@gmail.com', 'Calle 987'),**

**('Ana Marilu', '875556547', 'anamari@gmail.com', 'Plaza Principal'),**

**('Laura Martinez', '939987896', 'lauramartinez@hotmail.com', 'Calle 741'),**

**('Jose Lopez', '939333213', 'joselo@outlook.com', 'Calle 555'),**

**('Ana Gonzalez', '878882589', 'anagonza@gmail.com', 'Av Libertad'),**

**('Luis Torres', '939741234', 'josetorres@hotmail.com', 'Calle 111'),**

**('David Perez', '939951456', 'davidperez@gmail.com', 'Plaza Central'),**

**('Marta Garcia', '874120002', 'mariagarcia@yahoo.com', 'Plaza Principal');**

**INSERT INTO Eventos (fecha\_registro, fecha\_evento, lugar, tipo, descripcion)**

**VALUES**

**('2024-05-01', '2024-05-10', 'Centro de convenciones', 'Conferencia', 'Conferencia sobre tecnología'),**

**('2024-05-02', '2024-05-15', 'Teatro Municipal', 'Concierto', 'Concierto de música clásica'),**

**('2024-05-03', '2024-05-20', 'Estadio Olímpico', 'Deporte', 'Partido de fútbol'),**

**('2024-05-04', '2024-05-25', 'Parque Central', 'Festival', 'Festival de comida'),**

**('2024-05-05', '2024-05-30', 'Plaza Mayor', 'Conferencia', 'Conferencia sobre negocios'),**

**('2024-05-06', '2024-06-05', 'Auditorio Nacional', 'Concierto', 'Concierto de rock'),**

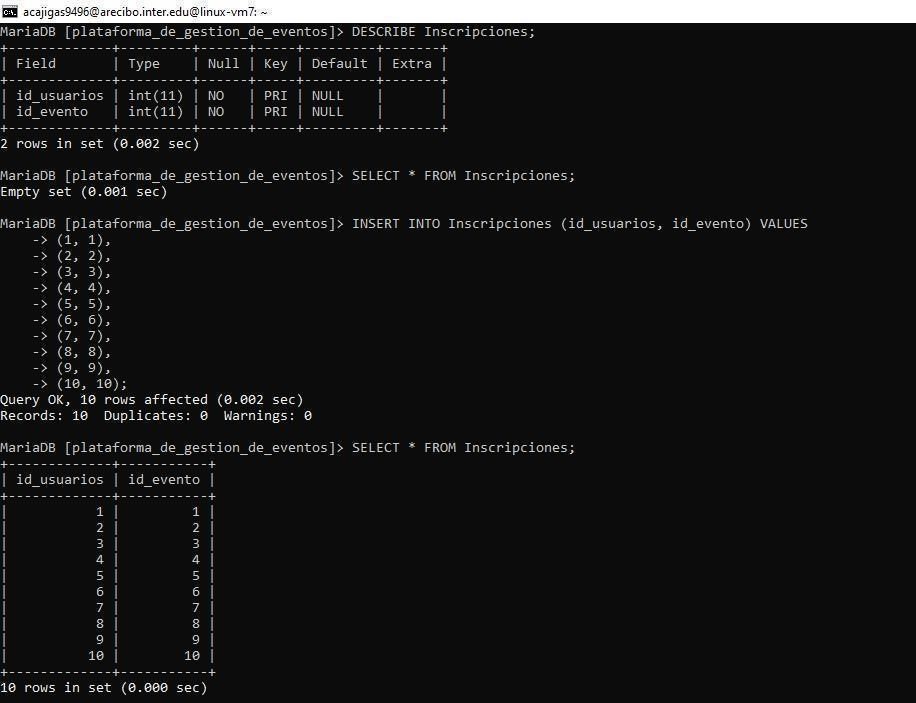
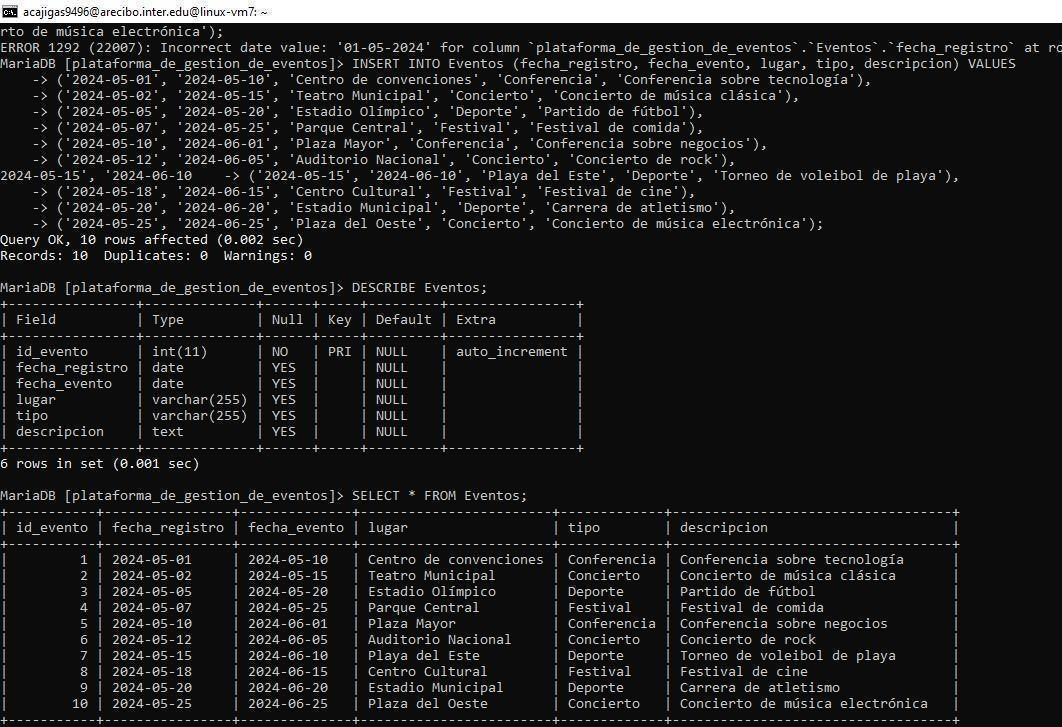
**('2024-05-07', '2024-06-07', 'Playa del Este', 'Deporte', 'Torneo de voleibol de playa'),**

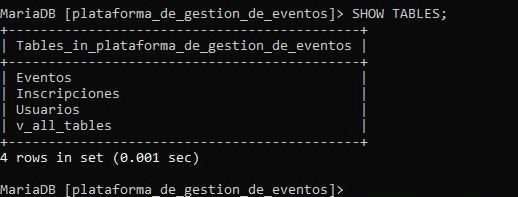
**('2024-05-08', '2024-06-10', 'Centro Cultural', 'Festival', 'Festival de cine'),**

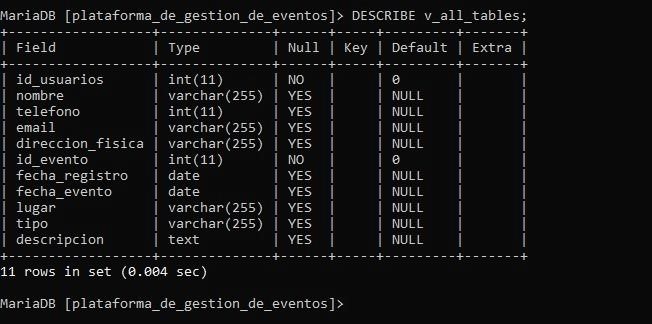
**('2024-05-09', '2024-06-15', 'Estadio Municipal', 'Deporte', 'Carrera de atletismo'),**

**('2024-05-10', '2024-06-20', 'Plaza del Oeste', 'Concierto', 'Concierto de música electrónica');**

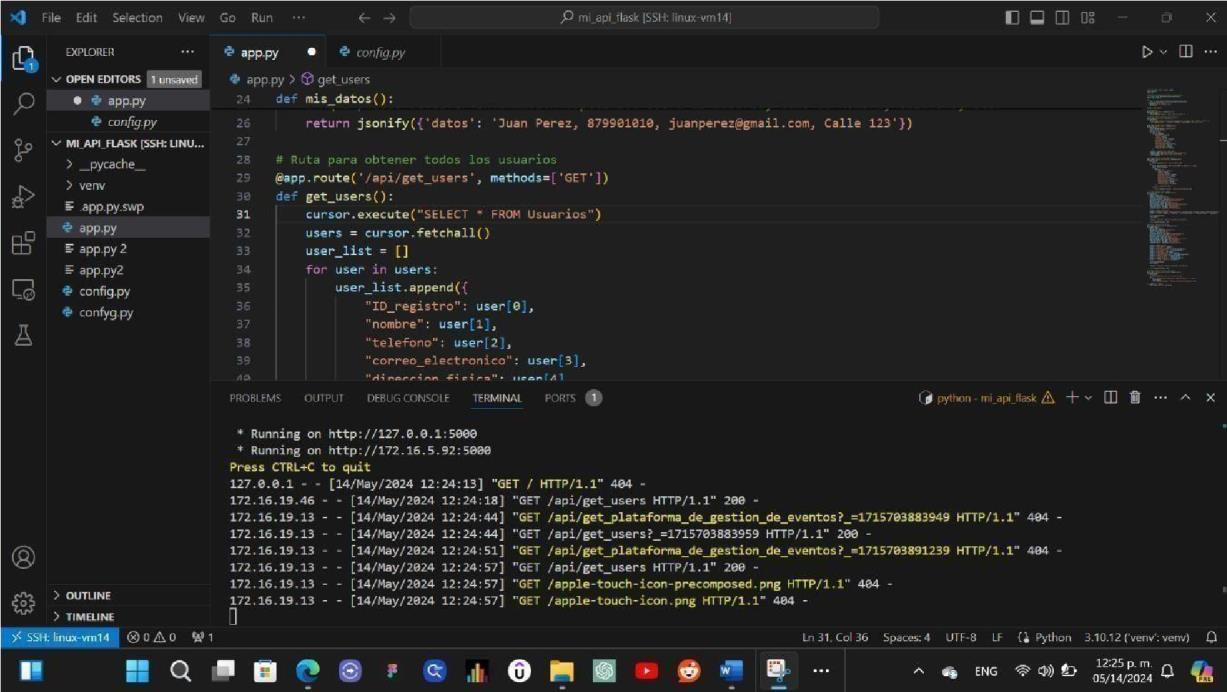
**BASE DE DATOS CREADA**

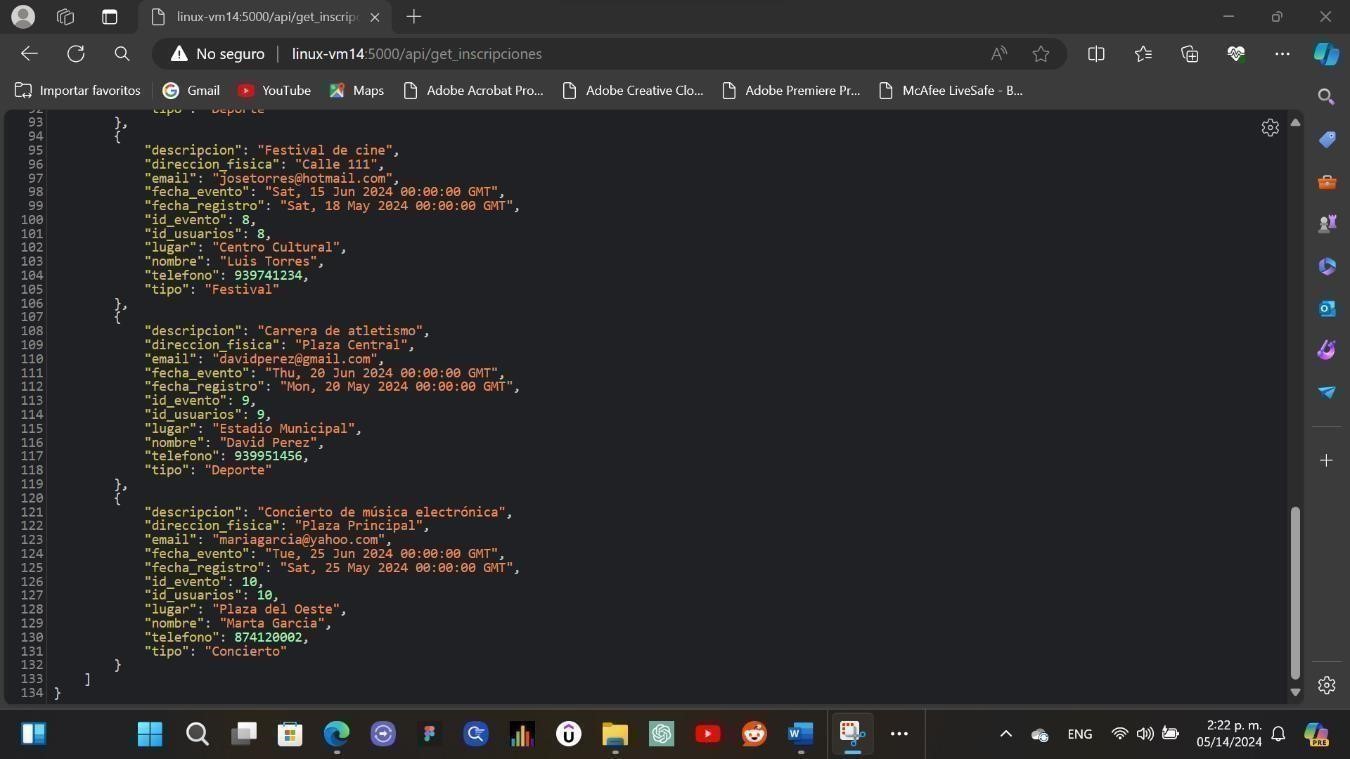
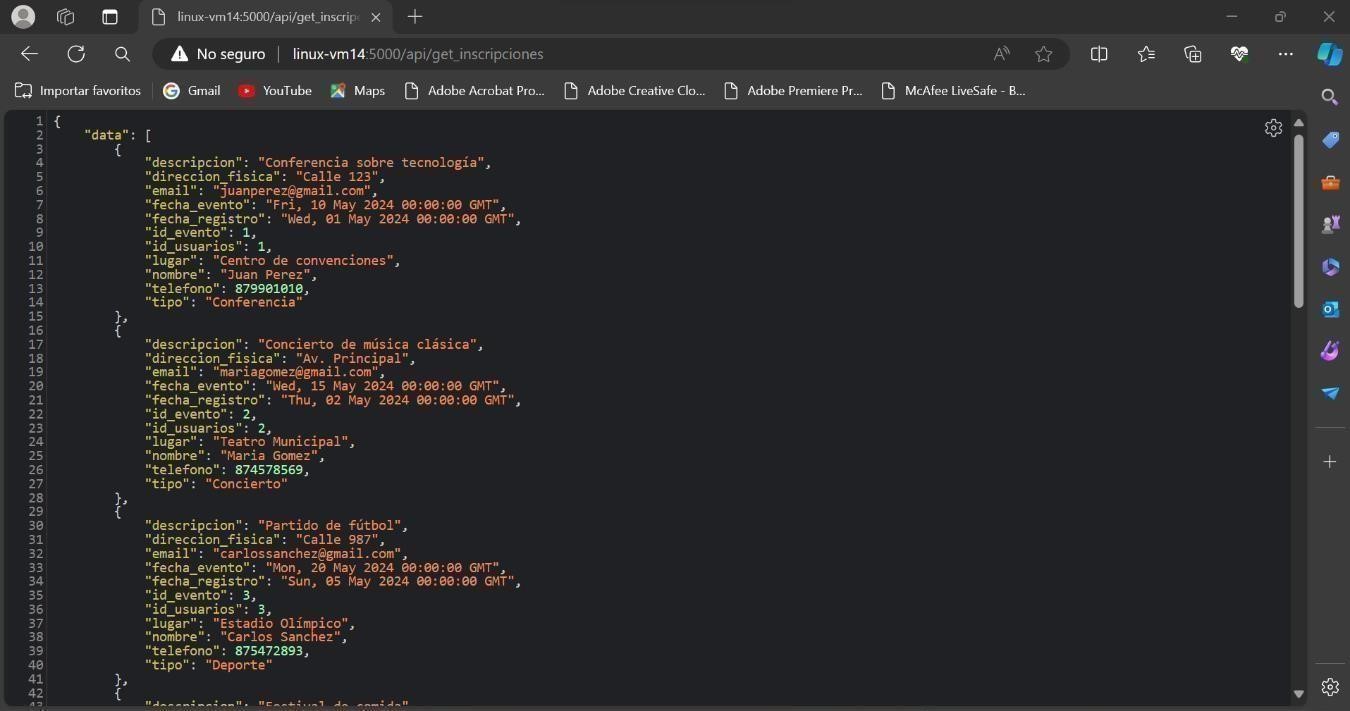
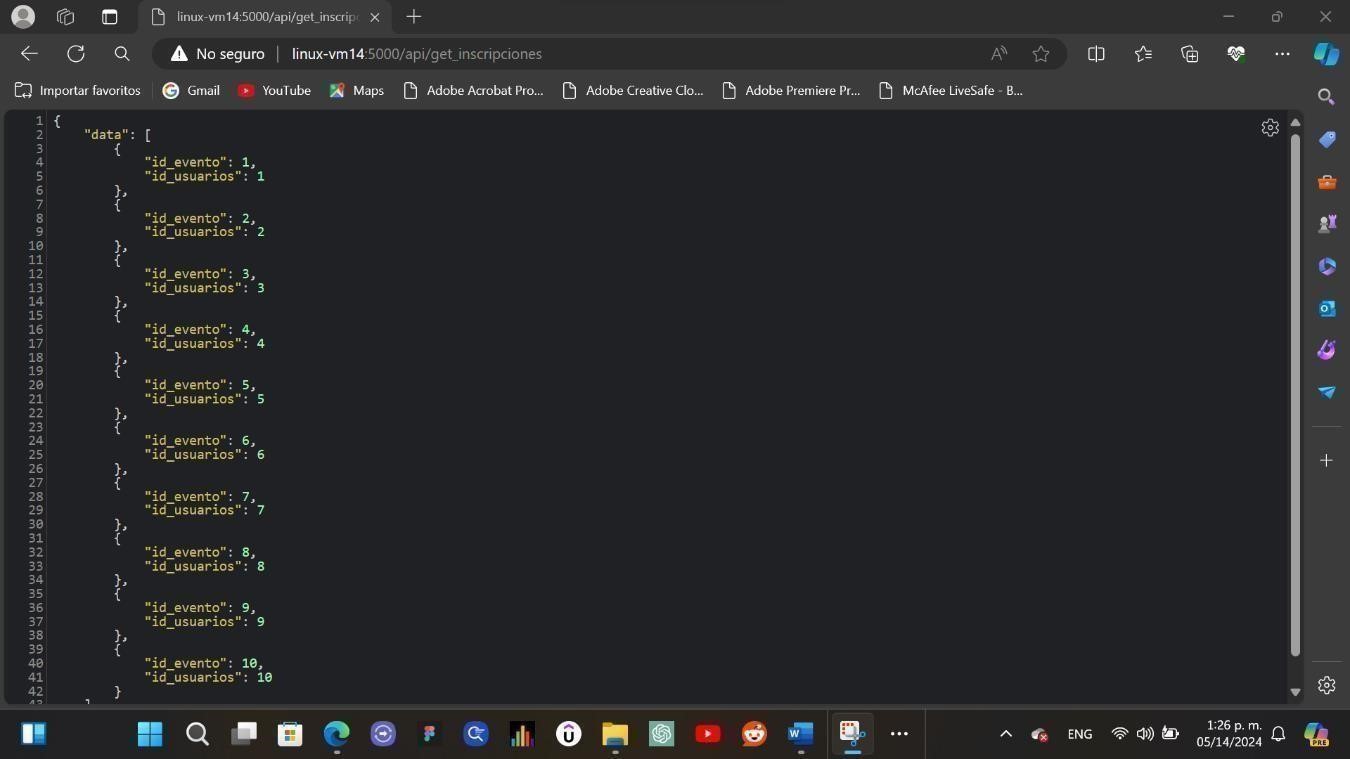


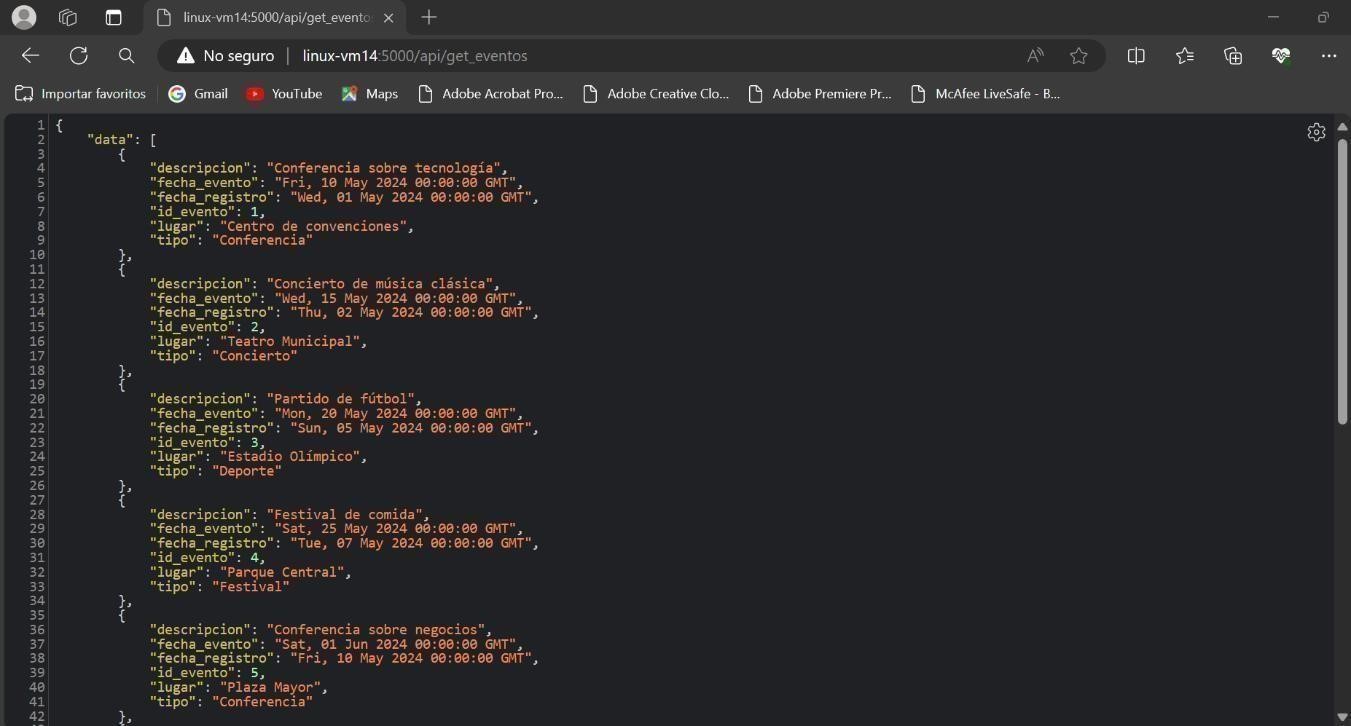


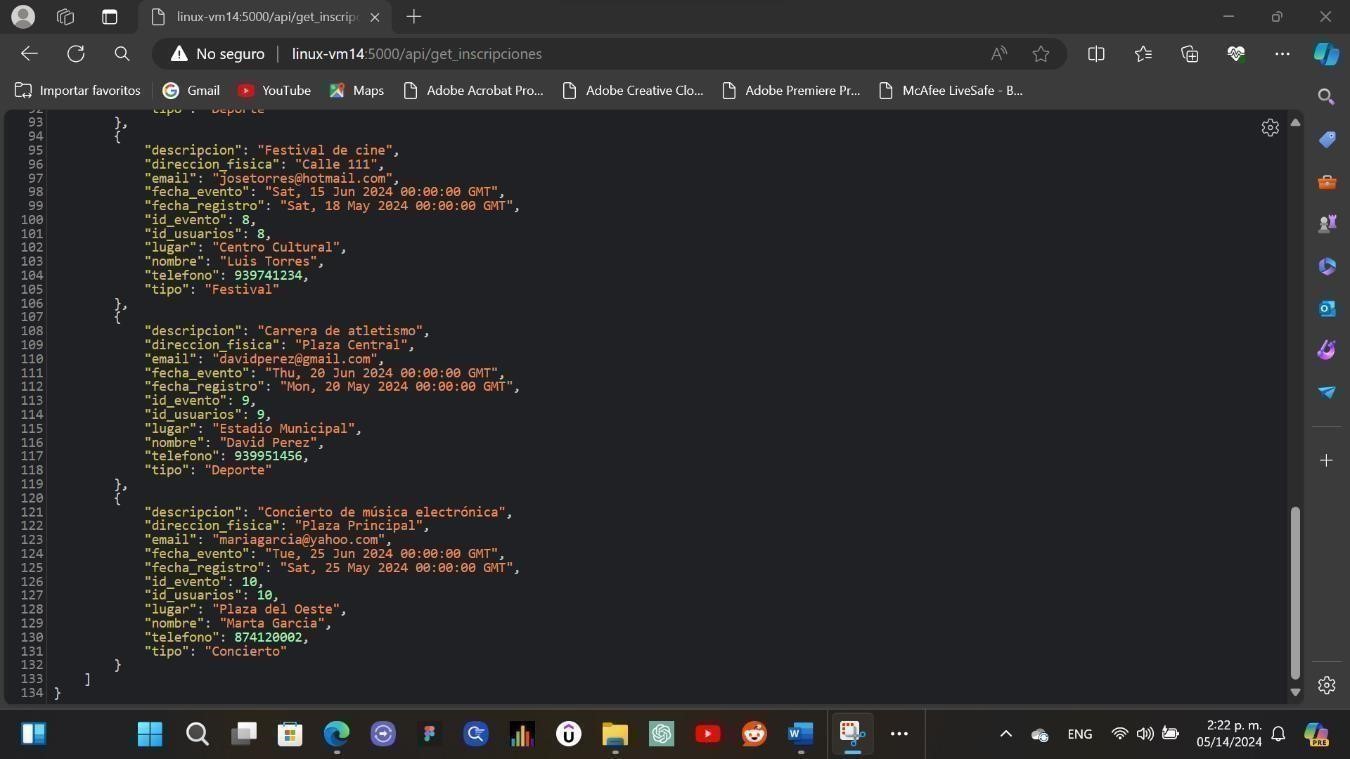


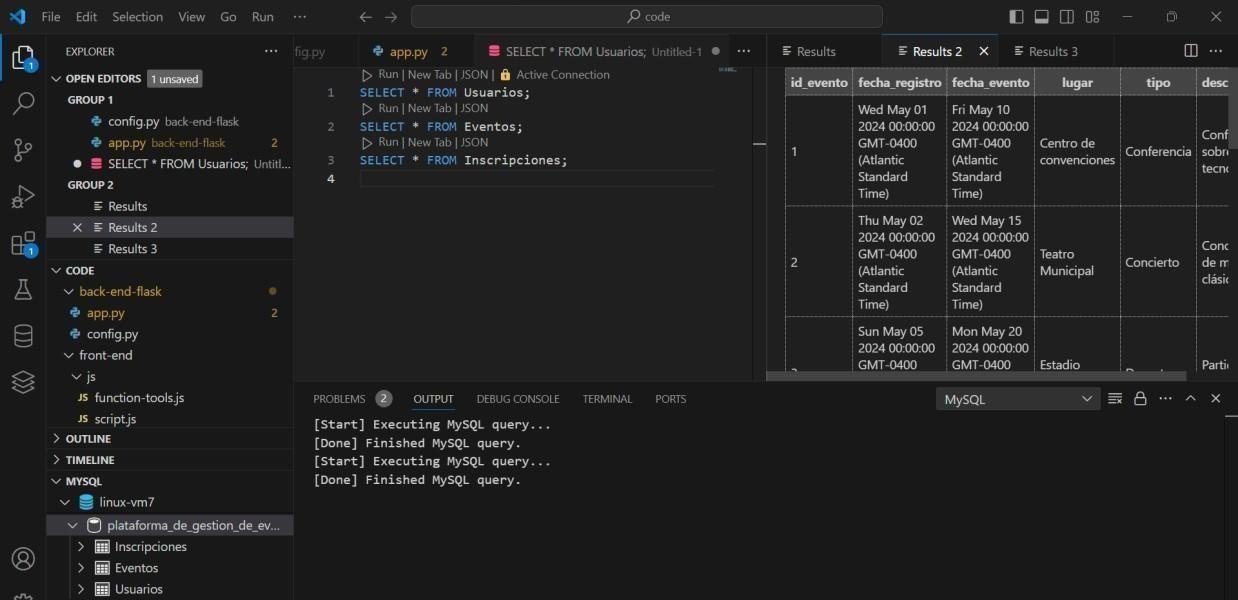
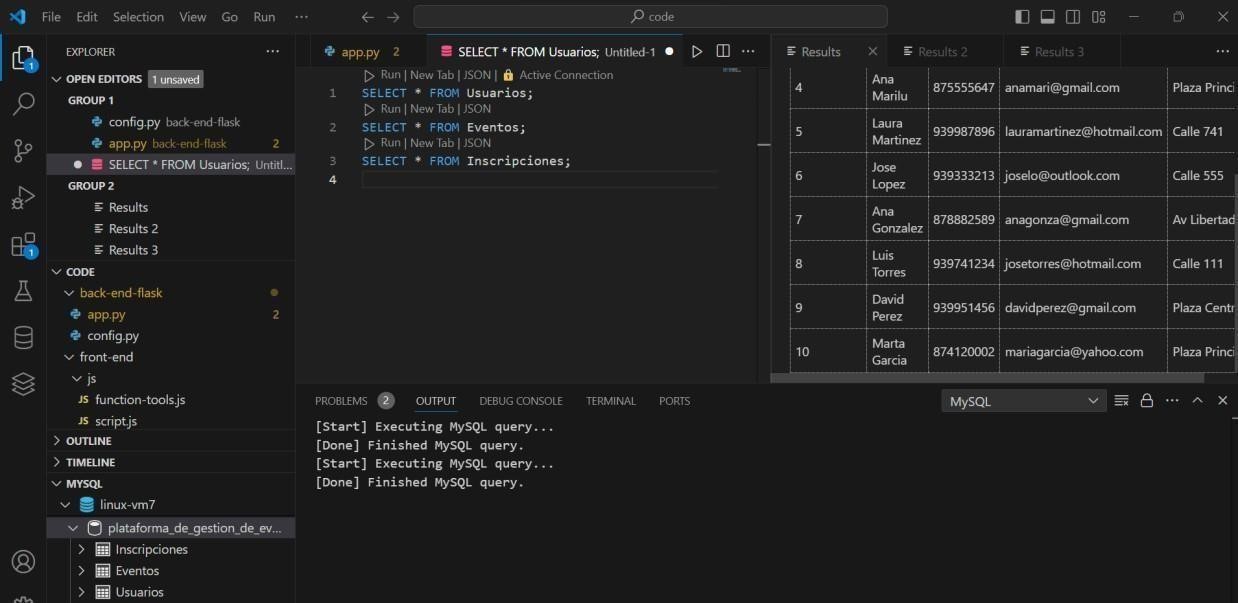
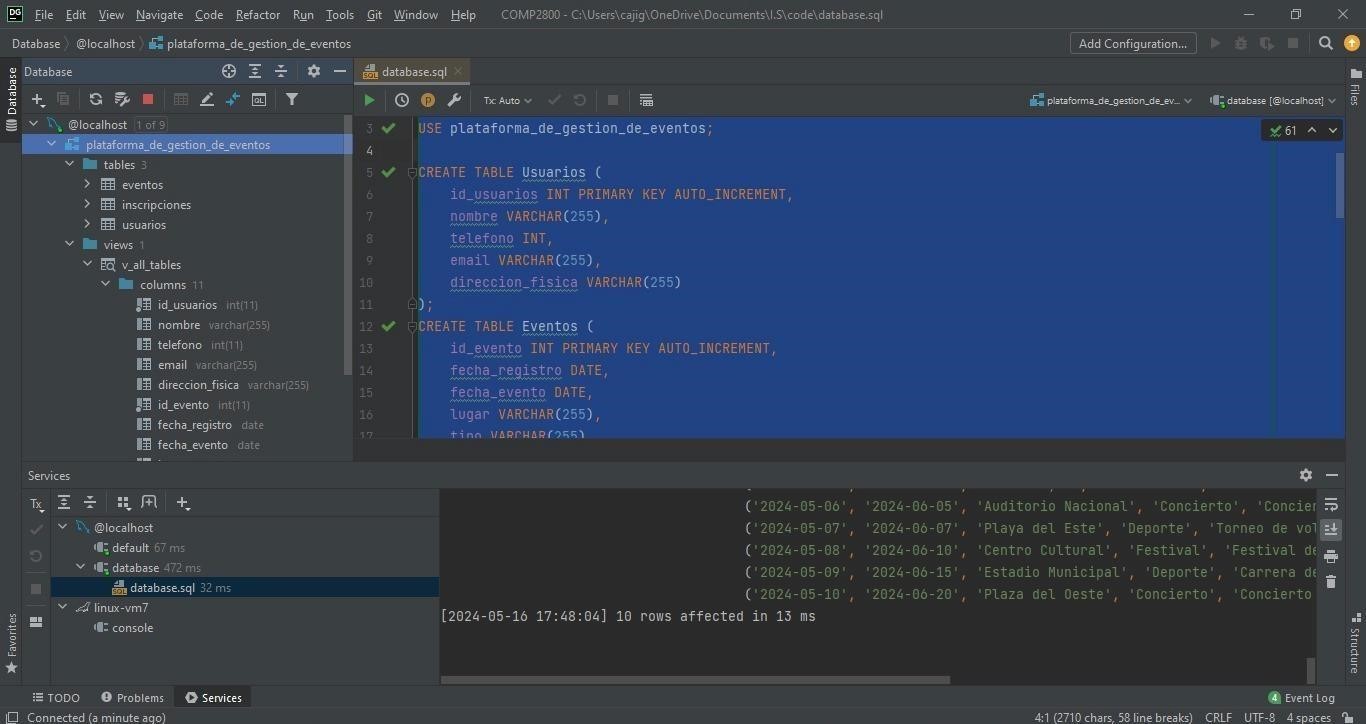
**APLICACION CORRIENDO Y BASE DE DATOS**











**TABLAS DESDE FRONT-END**

